T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BLOK ŞİFRELERDE KULLANILAN İKİLİ VE MDS MATRİSLER İÇİN YENİ TASARIM TEKNİKLERİ

GÖKHAN TUNCAY

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Prof. Dr. M. Tolga SAKALLI

EDİRNE-2021

GÖKHAN TUNCAY 'ın hazırladığı "**BLOK ŞİFRELERDE KULLANILAN İKİLİ VE MDS MATRİSLER İÇİN YENİ TASARIM TEKNİKLERİ**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir **Doktora tezi** olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. M. Tolga SAKALLI	.....
Doç. Dr. Vedat TAŞKIN	.....
Dr. Öğr. Üyesi Andaç ŞAHİN MESUT	.....
Doç. Dr. Sedat AKLEYLEK	.....
Dr. Öğr. Üyesi Meltem KURT PEHLİVANOĞLU	.....

Tez Savunma Tarihi: 24.09.2021

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

Tez Danışmanı	İmza
Prof. Dr. M. Tolga SAKALLI	.....

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

.....
Prof. Dr. H.R. Ferhat KARABULUT
Fen Bilimleri Enstitüsü Müdürü

T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ DOKTORA PROGRAMI
DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

24.09.2021
Gökhan TUNCAY

İmza

Doktora Tezi
Blok Şifrelerde Kullanılan İkili ve MDS Matrisler İçin Yeni Tasarım Teknikleri
T.Ü. Fen Bilimler Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Maksimum Uzaklıkla Ayrılabilen (MDS) matrisler, maksimum difüzyon sağladıkları ve maksimum dal sayısına sahip oldukları için blok şifreleme ve özet fonksiyonları gibi kriptografik ilkellerin tasarımında difüzyon katmanlarının temel bileşeni olarak kullanılmaktadır. Ancak bu matrislerin uygulamaları genellikle yüksek maliyet gerektirmektedir.

Bu tez çalışmasında, $\mathbb{F}_{2^m}$ cismi üzerinde 4×4 boyutlu involutif MDS matrislerin tümünü oluşturmak için doğrudan tasarım ve arama tabanlı yöntemleri birleştiren yeni bir melez teknik önerilmektedir. Önerilen teknik, n, bir kaba kuvvet araması ile elde edilebilecek $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı tüm 4×4 boyutlarındaki MDS matrislerin sayısını göstermek üzere, arama uzayı karmaşıklığını $\sqrt{n}$ düzeyinde azaltmaktadır. Dolayısıyla, $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cisimleri üzerinde 4×4 boyutlu involutif MDS matrislerin tamamı oluşturulabilmektedir. Bu bağlamda, donanım uygulama maliyetini hesaplamak için kullanılan XOR sayısı metriği açısından, $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cisimleri üzerinde bilinen en iyi involutif/involutif olmayan MDS matrisler oluşturulmaktadır. Ardından, elde edilen bu MDS matrislere bir global optimizasyon tekniği olan Boyar-Peralta algoritması uygulanmaktadır. Ayrıca önerilen teknik, $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cisimleri üzerinde üretilen 4×4 boyutlu involutif MDS/Almost-MDS matrislerden yararlanarak 12×12 ve 16×16 boyutlu MDBL matrislerin tasarımında da kullanılabilmektedir. Son olarak, yeni bir Hadamard matris özelliği tanımlanmaktadır. Bu özellik, Hadamard matrislerin aslında tüm $2^k \times 2^k$ $(k > 1)$ boyutlu involutif MDS matrislerin küçük bir miktarını oluşturmak için kullanılabilen temsili bir matris formu olduğunu göstermektedir.

Yıl : 2021

Sayfa Sayısı : 134

Anahtar Kelimeler : Blok şifreler, MDS matrisler, ikili matrisler, dal sayısı, yayılım katmanları, XOR sayısı, global optimizasyon

Doctor of Philosophy (PhD) Thesis
New Design Techniques for Binary and MDS Matrices Used in Block Ciphers
Trakya University Institute of Natural Sciences
Department of Computer Engineering

ABSTRACT

Maximum Distance Separable (MDS) matrices are used as the core component of diffusion layers in the design of cryptographic primitives such as block ciphers and hash functions because they provide the maximum diffusion and have maximum branch number. However, the implementations of these matrices require high-cost, generally.

In this thesis, we propose a new hybrid technique combining the direct construction methods and the search-based methods to generate all 4×4 involutory MDS matrices over $\mathbb{F}_{2^m}$. The proposed technique reduces the search space complexity at the level of $\sqrt{n}$, where n represents the number of all 4×4 matrices needed in the case of a brute-force search over $\mathbb{F}_{2^m}$. Therefore, this allows us to generate all involutory MDS matrices over $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$. In this regard, the lightest involutory/non-involutory MDS matrices known over $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$ in terms of XOR count metric are generated to estimate hardware implementation cost. Afterward, the Boyar-Peralta algorithm, a global optimization technique, was applied to these MDS matrices obtained. Moreover, the proposed technique can also be used to construct 12×12 and 16×16 MDBL matrices with 4×4 involutory MDS/Almost-MDS matrices generated over $\mathbb{F}_{2^3}$ and $\mathbb{F}_{2^4}$. Finally, a new property of Hadamard matrix is defined. This property indicates that Hadamard matrices are, in fact, a representative matrix form that can be used to generate a small amount of all $2^k \times 2^k$ $(k > 1)$ involutory MDS matrices.

Year : 2021

Number of Pages : 134

Keywords : Block ciphers, MDS matrices, binary matrices, branch number, diffusion layers, XOR count, global optimization

TEŞEKKÜR

Tüm akademik yaşamım boyunca bilgisi, tecrübesi ve dostluğuyla bana daima destek olan danışmanım Sayın Prof. Dr. Muharrem Tolga SAKALLI'ya,

Çalışmaya yaptığı bilimsel katkıları için, Sayın Doç Dr. Sedat AKLEYLEK'e, deneysel sonuçların elde edilmesi aşamasında sağladığı sonsuz destek için, Sayın Dr. Öğr. Üyesi Meltem KURT PEHLİVANOĞLU'na,

Tez çalışmama değerli bilgi ve tavsiyeleri ile katkıda bulunan tez izleme jüri üyelerim Sayın Doç. Dr. Vedat TAŞKIN ve Sayın Dr. Öğr. Üyesi Andaç ŞAHİN MESUT'a,

Bugünlere gelmemi sağlayan ve hep yanımda olan aileme,

Yol arkadaşım Telay'a,

Sonsuz teşekkürler. . .

Gökhan TUNCAY

İÇİNDEKİLER

ÖZET iv

ABSTRACT v

TEŞEKKÜR vi

SİMGELER DİZİNİ x

KISALTMALAR DİZİNİ xi

ŞEKİLLER DİZİNİ xii

ÇİZELGELER DİZİNİ xiii

1 GİRİŞ 1

1.1 Kriptoloji 1

1.1.1 Kriptografi 2

1.1.2 Kriptoanaliz 3

1.2 Modern Kriptografi 3

1.3 Blok Şifreler 5

1.4 Blok Şifrelerin Kriptografik Bileşenleri 7

1.4.1 Yer Değiştirme Kutuları 7

1.4.2 Doğrusal Dönüşümler 8

1.4.3 Anahtar Planlama Algoritmaları 10

1.5 Tezin Ana Hatları, Gerekçesi ve Önemi 10

2 MATEMATİKSEL ALTYAPI 12

3 LİTERATÜRDE KULLANILAN TASARIM VE OPTİMİZASYON TEKNİKLERİ 25

3.1 MDS Matrislerin Tasarım Teknikleri 25
3.1.1 Cauchy ve Vandermonde Matrislerden MDS Matris Tasarımı . . . 27
3.1.2 3×3 Boyutunda İnvolutif MDS Matrislerin Tümünün Elde Edilmesini Sağlayan Matris Formu 31
3.1.3 Companion Matrisler ve Çeşitli Kodlar Kullanılarak MDS Matris Tasarımı 32
3.1.4 Dairesel Matris ve Türevlerinden MDS Matris Tasarımı 34
3.1.5 Toeplitz ve Hankel Matrislerden MDS Matris Tasarımı 37
3.1.6 Hadamard ve GHadamard Matrislerden MDS Matris Tasarımı . . 38
3.2 MDBL Matrislerin Tasarım Teknikleri 40
3.2.1 Özel Matris Formları Kullanılarak MDBL Matris Tasarımı 40
3.2.2 Cebirsel Yöntemler Kullanılarak MDBL Matris Tasarımı 43
3.2.3 Devirli Matris Grupları Kullanılarak MDBL Matris Tasarımı . . . 45
3.3 XOR Sayısı 46
3.4 Optimizasyon Teknikleri 49
3.4.1 Yerel Optimizasyon 49
3.4.1.1 Doğrudan XOR Sayısı 49
3.4.1.2 Sıralı XOR Sayısı 50
3.4.2 Global Optimizasyon 51
3.4.2.1 PAAR Algoritması 53
3.4.2.2 Boyar-Peralta (BP) Algoritması 54

4 4×4 BOYUTUNDA TÜM İNVOLUTİF MDS MATRİSLERİN ELDE EDİLMESİNİ SAĞLAYAN YENİ BİR TEKNİK 59
4.1 Tekniğin Matematiksel İspatı 60
4.2 Tekniğin Algoritması 64
4.3 Materyal ve Metot 66

5 ÖNERİLEN TEKNİKLE ELDE EDİLEN MDS MATRİSLERE AİT DENEYSEL BULGULAR 73
5.1 Çeşitli Özel Matris Tiplerine Ait Deneysel Bulgular 74
5.1.1 GHadamard Matris Tipine Ait Deneysel Bulgular 74
5.1.2 Dairesel Matris Tipine Ait Deneysel Bulgular 76
5.1.3 Toeplitz Matris Tipine Ait Deneysel Bulgular 78
5.2 $\mathbb{F}_{2^3}$ Cismine Ait Deneysel Bulgular 80
5.3 $\mathbb{F}_{2^4}$ Cismine Ait Deneysel Bulgular 83
5.4 $\mathbb{F}_{2^8}$ Cismine Ait Deneysel Bulgular 89

6 ÖNERİLEN TEKNİKLE ELDE EDİLEN MDBL MATRİSLERE AİT DENEYSEL BULGULAR **92**
6.1 12×12 Boyutunda İkili Matrislere Ait Deneysel Bulgular 92
6.2 16×16 Boyutunda İkili Matrislere Ait Deneysel Bulgular 95

7 SONUÇ VE TARTIŞMA **99**

KAYNAKLAR **102**

EKLER **109**

A TEZ ÇALIŞMASINDA KULLANILAN MAGMA KODU **109**

B BP GLOBAL OPTİMİZASYON DETAYLARI **114**

ÖZGEÇMİŞ **119**

TEZ SIRASINDA GERÇEKLEŞTİRİLEN BİLİMSEL FAALİYETLER **120**

SİMGELER DİZİNİ

β	Dal Sayısı
$\mathbb{F}_2$	Elemanları 0 ve 1 olan sonlu cisim
$\mathbb{F}_{2^m}$	Elemanları 0 ve 1 olan m elemanlı vektörleri temsil eden sonlu cisim
$\oplus$	eXclusive OR (Özel Veya)
$\mathcal{M}_n(\mathbb{R})$	Girişleri $\mathbb{R}$ halkasından seçilen $n \times n$ boyutunda matrisler kümesi
wt	Hamming ağırlığı
$+$	Herhangi bir sonlu cisim üzerinde toplama
$*$	Herhangi bir sonlu cisim üzerinde çarpma
$\mathbb{Q}$	Rasyonel sayılar kümesi
$\mathbb{Z}$	Tam sayılar kümesi

KISALTMALAR DİZİNİ

AES	Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
BP	Boyar-Peralta Algoritması
Circ	Dairesel Matris
DES	Data Encryption Standard (Veri Şifreleme Standardı)
FP	Fixed Point (Sabit Nokta)
GF	Galois Field (Galois Cismi)
Ghad	Genel Hadamard Matris
Had	Hadamard Matris
IoT	Internet of Things (Nesnelerin İnterneti)
MD5	Message Digest Algorithm (Mesaj Özeti Algoritması)
MDBL	Maximum Distance Binary Linear Codes (Maksimum Uzaklıkta İkili Doğrusal Kodlar)
MDS	Maximum Distance Separable (Maksimum Uzaklıkta Ayrılabilen)
S-box	Substitution-box (Yer Değiştirme Kutusu)
SHA	Secure Hash Algorithm (Güvenli Özet Algoritması)
SLP	Shortest Linear Program (En Kısa Doğrusal Yol Programı)
SPN	Substitution–Permutation Network (Yer Değiştirme Permütasyon Ağı)
Vand	Vandermonde Matris
WEP	Wireless Equivalent Privacy (Kablosuz Eşdeğer Gizlilik)
WPA	Wi-Fi Protected Access (Wi-Fi Korumalı Erişim)
XOR	Exclusive OR (Özel Veya)

ŞEKİLLER DİZİNİ

Şekil 1.1. Simetrik Şifreleme Algoritmaları 4
Şekil 1.2. Asimetrik Şifreleme Algoritmaları 4
Şekil 1.3. SPN Ağının Bir Döngüsü 6
Şekil 1.4. Feistel Ağının Bir Döngüsü 7
Şekil 1.5. 4×4 Boyutunda Bir S-kutusunun Şematik Gösterimi 8
Şekil 3.1. MDS Matrisler İçin Kullanılan Tasarım Tekniklerinin Sınıflandırılması . 28
Şekil 3.2. $y_1 \oplus y_2 \oplus y_3$ İşleminin Derinlik-4 İçin Devre Şeması 57
Şekil 3.3. $y_1 \oplus y_2 \oplus y_3$ İşleminin Derinlik-5 İçin Devre Şeması 57

ÇİZELGELER DİZİNİ

Çizelge 2.1. $\mathbb{F}_{2^4}/x^4+x+1$ Sonlu Cisminin Elemanları 16
Çizelge 2.2. İkili $[2n, n]$ Kodların Maksimum Dal Sayıları 22
Çizelge 2.3. XOR Kapısı Doğruluk Tablosu 23
Çizelge 3.1. $\mathbb{F}_{2^4}$ Sonlu Cisim Elemanları Çarpımlarının İndirgenemez Polinoma Bağlı XOR Sayıları 48
Çizelge 5.1. GHadamard Matris Tipi Kullanılarak Elde Edilen Deneysel Bulgular . 76
Çizelge 5.2. Dairesel Matris Tipi Kullanılarak Elde Edilen Deneysel Bulgular . . . 77
Çizelge 5.3. Toeplitz Matris Tipi Kullanılarak Elde Edilen Deneysel Bulgular . . . 79
Çizelge 5.4. Çeşitli Matris Tiplerine Ait Deneysel Bulguların Literatür ile Karşılaştırılması 79
Çizelge 5.5. $\mathbb{F}_{2^3}$ Cismi Üzerinde Tanımlı $0xb$ Polinomuna Ait Deneysel Bulgular . 81
Çizelge 5.6. $\mathbb{F}_{2^3}$ Cismi Üzerinde Tanımlı $0xd$ Polinomuna Ait Deneysel Bulgular . 82
Çizelge 5.7. İnvolutif Matrisler İçin $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı $0x13$ Polinomuna Ait Deneysel Bulgular 86
Çizelge 5.8. Non-involutif Simetrik Matrisler İçin $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı $0x13$ Polinomuna Ait Deneysel Bulgular 87
Çizelge 5.9. Non-involutif Antidiyagonal Simetrik Matrisler İçin $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı $0x13$ Polinomuna Ait Deneysel Bulgular 87
Çizelge 5.10. $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ Cismi Üzerinde 4×4 Boyutunda İnvolutif MDS Matrislerin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması 87
Çizelge 5.11. $\mathbb{F}_{2^4}$ Cismi Üzerinde 4×4 Boyutunda İnvolutif Olmayan MDS Matrislerin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması 88
Çizelge 5.12. $\mathbb{F}_{2^3}$ Cismi Üzerinde Tanımlı 4×4 İnvolutif MDS Matrislerdeki 1'lerin Sayısı 88

Çizelge 5.13. $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı 4×4 İnvolutif MDS Matrislerdeki 1'lerin Sayısı 88
Çizelge 5.14. $\mathbb{F}_{2^8}$ Cismi Üzerinde Tanımlı $0x12d$ Polinomuna Ait Deneysel Bulgular 91
Çizelge 5.15. $\mathbb{F}_{2^8}$ Cismi Üzerinde 4×4 Boyutunda İnvolutif MDS Matrislerin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması 91
Çizelge 6.1. $001-011-111$ Üreteç Matrisi ile Elde Edilen Devirli Matris Grubu 93
Çizelge 6.2. 3×3 Boyutlarında Devirli Matris Grupları 94
Çizelge 6.3. 12×12 Boyutlarında MDBL Matrislere Ait Deneysel Bulgular 95
Çizelge 6.4. 16×16 Boyutlarında MDBL Matrislere Ait Deneysel Bulgular 97
Çizelge 6.5. 16×16 Boyutunda İnvolutif MDBL M_{8B} Matrisinin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması 98
Çizelge B.1. M_1 Matrisinin BP Global Optimizasyon Sonuçları 114
Çizelge B.2. M_2 Matrisinin BP Global Optimizasyon Sonuçları 115
Çizelge B.3. M_3 Matrisinin BP Global Optimizasyon Sonuçları 115
Çizelge B.4. M_6 Matrisinin BP Global Optimizasyon Sonuçları 116
Çizelge B.5. $\mathbb{F}_{2^4}/0x13$ Üzerinde BP Optimizasyon Sonucu 36 XOR Olan Matrisler 117
Çizelge B.6. $\mathbb{F}_{2^8}/0x12d$ Üzerinde BP Optimizasyon Sonucu 88 XOR Olan Matrisler 118

BÖLÜM 1

GİRİŞ

Bilgi, insanlık tarihi boyunca ilerlemelerin şekillendirilmesi, savaşların kazanılması ve ticari başarılar elde edilmesi gibi pek çok konuda medeniyetlerin en önemli aracı olmuştur. Yazının keşfi ise bilgiyi biriktirilebilir hale getirmesinin yanında, iletilmesi sırasında gizlenmesinin de kaçınılamaz bir gereklilik olduğunu ortaya çıkarmıştır. Bu gereklilik kriptoloji biliminin doğmasına olanak sağlamış, gelişen teknolojilerin her zaman bilginin güvenlik ve gizliliğine dair endişeleri arttırması da kullanımının yaygınlaşmasına uygun ortamı hazırlamıştır.

20. yüzyılın ortalarına kadar çoğunlukla hükümetler ve askeri birimler tarafından kullanılan kriptoloji, 1950'lerin sonlarından itibaren modern toplumumuzu hızla değiştirmeye başlayan dijital çağ ile birlikte evrensel boyutta kullanılır olmuştur.

Günümüzde bir çevrimiçi sisteme kullanıcı adı ve parola ile giriş gerçekleştirmek, telefonla görüşmek, anlık mesajlaşma uygulamalarını kullanmak ve elektronik ticaret platformları üzerinden alışveriş yapmak gibi pek çok işlem sırasında kriptoloji biliminin bize sağladıklarından faydalanılmaktadır. Bunun ötesinde, yapay zeka ve makine öğrenmesi kullanılarak gerçekleştirilen veri analizleri, bulut hesaplama, nesnelerin interneti gibi son dönem teknolojilerin geniş çapta yayılması, toplu gözetim ve veri ihlalleri kaygılarının artmasına neden olmakta, bu da kriptoloji bilimini hiç olmadığı kadar önemli bir konuma getirmektedir.

1.1 Kriptoloji

Türk Dil Kurumu (2021) güncel Türkçe sözlüğüne göre kriptoloji, "Gizli yazılar, şifreli belgeler bilimi ya da incelemesi" şeklinde tanımlanmaktadır. Kriptografi ve kriptoanaliz olarak adlandırılan iki alt disiplinden oluşmaktadır.

1.1.1 Kriptografi

Genel olarak bilginin işlenmesi, depolanması ve iletilmesi sırasında yetkisiz kişi veya sistemlerin saldırılarına karşı güvence altına alınmasını sağlayan matematiksel teknikler bütünü kriptografi olarak tanımlanmaktadır.

Kriptografinin kapsamı, hızla gelişen teknolojiyle paralel olarak sürekli genişleme ve değişme eğiliminde olsa da temel hedefleri aşağıdaki gibi sıralanmaktadır.

- **Gizlilik:** Bilgiler, yetkisi olmayan herkesten gizli tutulmalıdır.
- **Bütünlük:** Bilginin, sadece yetkisi olanlar tarafından değiştirilmesine (ekleme, silme veya güncelleme gibi) izin verilmesidir.
- **Kimlik Doğrulama:** Mesaj kimlik doğrulaması ve varlık kimlik doğrulaması olmak üzere iki kısma ayrılmaktadır.
  - **Mesaj Kimlik Doğrulaması:** Mesaj herhangi bir kötü niyetli müdahaleye karşı korunmalı ve alıcı taraf mesajın kaynağını doğrulayabilmelidir.
  - **Varlık Kimlik Doğrulaması:** Bir taraf, iletişime dahil olan ikinci bir tarafın kimliğinden ve ikinci tarafın fiilen katıldığından emin olmalıdır.
- **İnkar Edememe:** Bir varlığın (kişi veya işletme gibi) önceki taahhütlerini veya eylemlerini reddetmesi engellenmelidir.

Bu hedeflere ulaşmak için, farklı taraflar arasında yüksek seviyeli kriptografik protokoller tasarlanmakta ve uygulanmaktadır. Protokoller, temel olarak kriptografik bileşenlerden oluşmaktadır. Bu bileşenler aşağıdaki şekilde sınıflandırılmaktadır (Wenbo, 2004).

- Alfabetik karakterlerden oluşan bir açık metin uzayı,
- Şifreli karakterlerden oluşan bir şifreli metin uzayı,
- Olası şifreleme anahtarları kümesinden oluşan bir şifreleme anahtar uzayı,
- Olası şifre çözme anahtarları kümesinden oluşan bir şifre çözme anahtar uzayı,
- Etkin bir şifreleme algoritması,
- Etkin bir şifre çözme algoritması.

Söz konusu kriptografik bileşenler ise çoğunlukla çeşitli matematiksel yapı taşlarından elde edilmektedir.

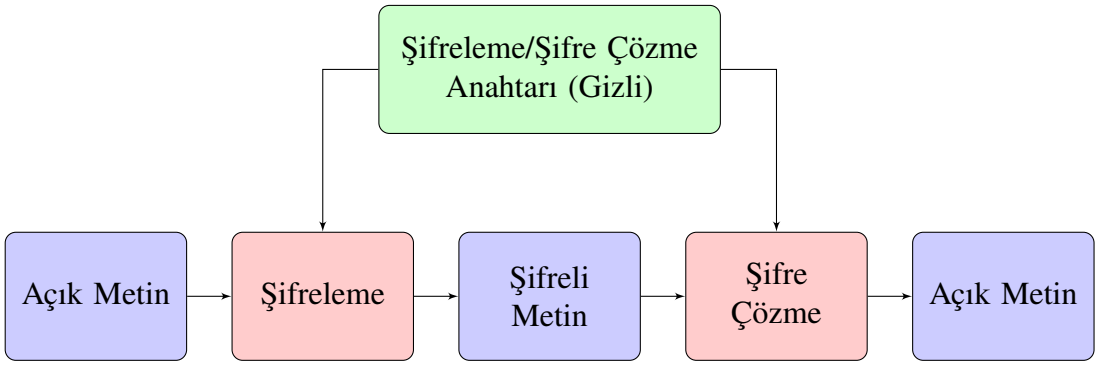
1.1.2 Kriptoanaliz

Kriptografik bir algoritmanın ya da yapının güvenilirliğinin denetlenmesi kriptoanalizin çalışma alanına girmektedir. Kriptografik sistemlerin zayıf yönlerini ortaya çıkarmak ve zaaflarını göstermek açısından oldukça önemlidir. Kriptoanalistler, kriptografik bileşenlerin çözümünü araştırırken çeşitli saldırı modelleri kullanmaktadır. Yaygın kullanılan saldırı modelleri aşağıda listelenmektedir (Englund vd., 2007; Rijmen, 1997; Stinson, 2002). Bu modeller, tek başına ya da modellerin herhangi bir kombinasyonu şeklinde de kullanılabilmektedir.

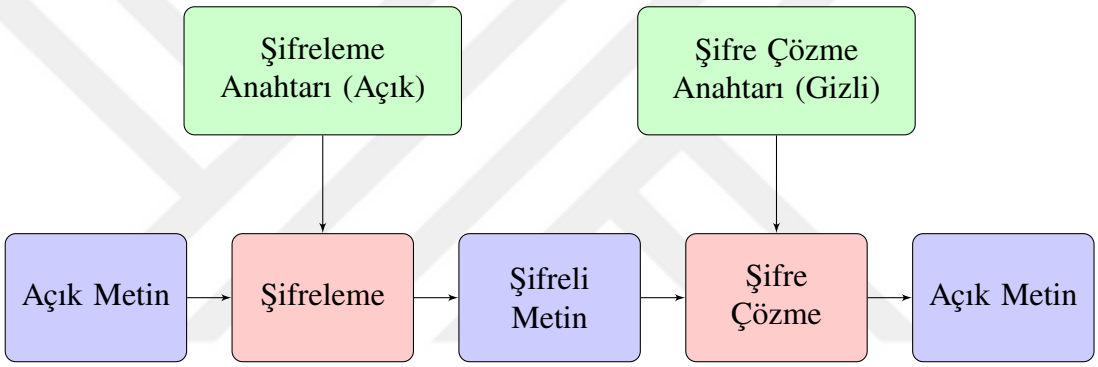
- **Yalnızca Şifreli Metin Saldırısı:** Saldırgan tamamen pasiftir. Sadece şifreli metinlere erişimi bulunur. Bazen şifresiz metin üzerindeki bazı istatistiksel bilgilere de erişebilir.
- **Bilinen Açık Metin Saldırısı:** Saldırgan, aynı anahtar kullanılarak şifrelenmiş bazı açık metin-şifreli metin çiftlerine erişebilir.
- **Seçilmiş Açık Metin Saldırısı:** Saldırganın şifreleme cihazına erişimi vardır ve uygun şekilde seçilmiş açık metinleri göndererek bu metinlere karşılık gelen şifreli metinleri elde edebilir.
- **Seçilmiş Şifreli Metin Saldırısı:** Saldırganın şifre çözme cihazına erişimi vardır ve uygun şekilde seçilmiş şifreli metinleri deşifre ederek bu metinlere karşılık gelen açık metinleri elde edebilir.
- **Seçilmiş Başlatma Vektörü Saldırısı:** Kriptosistem girdi olarak bir başlatma vektörüne (Initialization Vector-IV) sahipse, saldırgan açık metnin yanı sıra IV'nin de kontrolüne sahip olur. Böylece, şifreleme cihazını kullanarak uygun şekilde seçilmiş IV ve açık metinler göndererek bu girdilere karşılık gelen şifreli metinleri elde edebilir.

1.2 Modern Kriptografi

Günümüzde kriptografik yapılar; simetrik, asimetrik (ya da açık anahtarlı) ve kriptografik hash (özet) fonksiyonları olmak üzere üç kısıma ayrılmaktadır. Simetrik kriptografide, gönderen ve alıcı hem şifreleme hem de şifre çözme işlemleri için aynı gizli anahtarı paylaşırken, asimetrik kriptografide, gönderen ve alıcı biri gizli diğeri açık anahtardan oluşan bir anahtar çiftine sahiptir. Şifreleme işlemi açık anahtar ile yapılırken, şifre çözme işlemi için gizli anahtar kullanılmaktadır. Şekil 1.1 ve 1.2'de, söz konusu sistemlerin basit şematik gösterimleri verilmektedir.



Şekil 1.1. Simetrik Şifreleme Algoritmaları



Şekil 1.2. Asimetrik Şifreleme Algoritmaları

Her iki kriptografik yapı için de genel varsayım, gizli anahtar dışında sistemle ilgili her şey halka açık olsa bile bir şifreleme sisteminin güvenli olması gerektiğidir. Bu prensip, Kerckhoffs ilkesi (Kerckhoffs, 1883) olarak anılmaktadır. Asimetrik kriptografi bileşenleri esas olarak sayı teorisindeki matematiksel yapılar üzerine inşa edilirken, simetrik kriptografi çoğunlukla Boole fonksiyonları gibi ayrık yapılara dayanmaktadır. Simetrik şifreleme algoritmaları, asimetrik şifreleme algoritmalarına göre daha hızlıdır ve büyük boyutlardaki verilerin şifrelenmesinde kullanımı daha uygundur. Öte yandan asimetrik şifreleme algoritmaları da, simetrik algoritmaların en büyük problemi olan anahtar dağıtımına çözüm getirmeleriyle öne çıkmaktadır. Ayrıca simetrik şifreleme algoritmaları, açık metinin şifrelenme şekline göre de blok şifreler ve akış (stream) şifreleri olmak üzere iki alt gruba ayrılmaktadır. Blok şifreler açık metini belirli uzunlukta bloklara bölerek şifreleme yaparken, akış şifreler açık metini bit veya byte karakterler olarak şifreler. Literatürde iyi bilinen blok şifrelere, DES (Data Encryption Standard) (D. E. Standard vd., 1999), AES (Advanced Encryption Standard) (A. E. Standard, 2001), Blowfish (Schneier, 1993) ve PRESENT (Bogdanov vd., 2007) örnek olarak verilebilir.

Akış şifrelerde ise WPA(Wi-Fi Protected Access) ve WEP (Wired Equivalent Privacy) gibi kablosuz ağ güvenlik algoritmalarında kullanılan RC4 (ARCFOUR) (R. L. Rivest, 1992), Trivium (De Canniere & Preneel, 2008) ve HC-256 (Wu, 2004) en bilindik örneklerdendir. Asimetrik şifreleme algoritmalarına baktığımızda ise Diffie-Helman (Diffie & Hellman, 1976) ve RSA (R. L. Rivest vd., 1978) algoritmaları öne çıkmaktadır.

Kriptografik özet (Hash) fonksiyonları ise simetrik ve asimetrik şifreleme algoritmalarının yanında, günümüzde kullanılan önemli kriptografik yapılardan biridir. Bu fonksiyonlar, verinin sabit uzunlukta (bit değerlerinden oluşan) özetini oluşturur. Verideki en küçük değişiklik bile özetin değerini önemli ölçüde değiştirir. Sayısal imza, blok zincir teknolojileri gibi önemli kullanım alanlarına sahiptir. MD5 (R. Rivest & Dusse, 1992) ve SHA (Pub, 2012) günümüzde yaygın olarak kullanılan kriptografik özet fonksiyonlarıdır.

1.3 Blok Şifreler

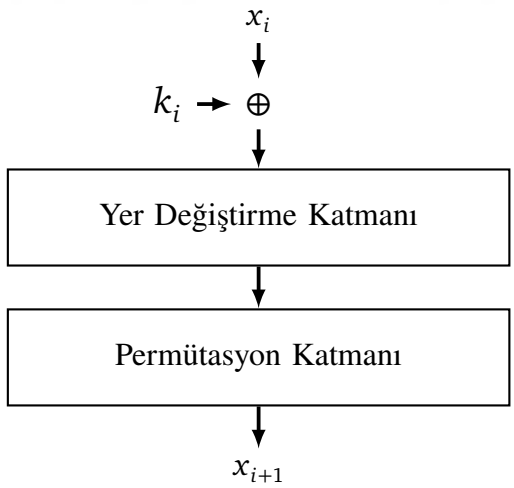
Bir blok şifre temel olarak, anahtar uzayı $\mathcal{K}$, şifreli ve açık metin uzayları $\mathcal{X}$ olan bir çift $\mathcal{E}$ =(Şifrele, ŞifreÇöz) fonksiyonundan ibarettir. Seçilen her sabit $K \in \mathcal{K}$ anahtarı için, Şifrele_K ile gösterilen bir $\text{Şifrele}(K,\cdot) : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ anahtarlı permütasyonu tanımlanır. ŞifreÇöz_K ile gösterilen $\text{ŞifreÇöz}(K,\cdot) : \mathcal{K} \times \mathcal{X} \to \mathcal{X}$ şifre çözme fonksiyonu ise herhangi bir $K \in \mathcal{K}$ ve $M \in \mathcal{X}$ için $\text{ŞifreÇöz}_K(\text{Şifrele}_K(M)) = M$ çözümünü sağlar. Bu iki fonksiyonun güvenilirliği, K anahtarını bilmeyen bir düşmanın, $\text{Şifrele}_K(\cdot)$ fonksiyonunun girdi/çıktı davranışını, tüm $|\mathcal{X}|!$ kümesinden rastgele seçilen bir $\mathcal{X}$ permütasyonundan ayırt edememesi yaklaşımına dayanır.

Shannon (1949) tarafından ortaya atılan karıştırma ve yayılma, blok şifrelerin tasarımında yaygın olarak kullanılan iki temel ilkedir. Karıştırma, şifreli metin ile şifre anahtarı arasındaki ilişkiyi gizlemeyi amaçlar ve yer değiştirme yapıları (Substitution boxes: S- boxes) ile sağlanır. Yayılma ise, doğrusal dönüşüm ya da dönüşümler yoluyla gerçeklenir ve açık metindeki izlerin, şifreli metinde sezilmemesini sağlar. Bu özelliklerin istatistiksel saldırılara ve çeşitli kriptoanaliz yöntemlerine karşı savunma sağladığı bilinmektedir. Çoğu modern blok şifreleme, iyi tasarlanmış karıştırma ve yayılma katmanlarına sahiptir.

Pratikte kullanılan blok şifrelerin hemen hemen hepsi, yinelenen şifreleme modeli adı verilen aynı temel çerçeveyi kullanır. Bu çerçeve, aynı şifreleme adımının tekrar tekrar uygulanmasıdır ve her bir adımda tekrar eden matematiksel işlemlere *döngü fonksiyonları*, bu işlemlerin bütününe ise *döngü* adı verilir. Modern blok şifreler, bu döngüleri yeterli sayıda yineleyerek güvenliği sağlarlar. Bir döngüde, basit aritmetik işlemlerinin yanı sıra hem yer değiştirme yapıları, hem de doğrusal dönüşümler kullanılabilir. Bu yapıların her döngüde art arda kullanılması, istenmeyen simetrilerin oluşmasına neden olabilir. Bunu

engellemek adına, bir anahtar genişletme algoritması yardımıyla gizli anahtardan türetilen farklı alt anahtarlar döngüye ilave edilir. Bu modeli kullanan blok şifrelerin tasarımında, Yer Değiştirme-Permütasyon ağları (Substitution Permutation Networks-SPN) ve Feistel ağları olmak üzere iki tasarım mimarisinden yararlanılmaktadır. Bu mimarilerin arasındaki en temel fark döngü içerisinde bir bloğun işlenmesi sırasında ortaya çıkmaktadır. Feistel mimarisinde bir döngüde o anki bloğun yarısı işlenirken, SPN mimarisinde o anki bloğun tümü işlenir.

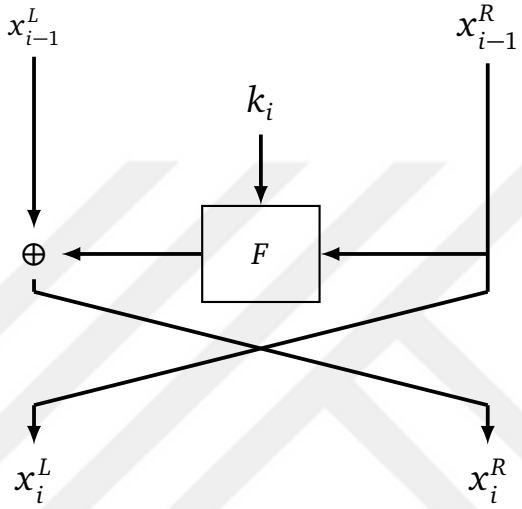
SPN, yine Claude Shannon tarafından önerilmiş ve pek çok blok şifre ile özet fonksiyonunun tasarımında popüler olmuştur. Bir SPN, hem doğrusal olmayan hem de doğrusal bileşenlerin yinelemeli uygulamasının özel bir şeklidir. SPN döngü fonksiyonu, bir anahtar ekleme işlemi, bir yer değiştirme katmanı (S-kutusu katmanı) ve ardından bir doğrusal permütasyon katmanından (P katmanı) oluşur. S-kutusu katmanı genellikle küçük boyutlu S-kutularının paralel bir uygulaması olarak uygulanır. Permütasyon katmanı, S-kutusu katmanının sonuçlarını verimli bir şekilde karıştırmak için doğrusal bir yayılım matrisinin uygulanmasını sağlar. Her döngünün çıktısı, bir sonraki döngüye girdi olarak beslenir. Son döngünün ardından, son bir anahtar ekleme adımı gerçekleştirilir ve şifrenin çıktısı edilir. En bilindik SPN blok şifresi, ABD için Federal Bilgi İşleme Standardı olan AES şifresidir. SPN ağına ait bir döngünün genel gösterimi, Şekil 1.3'de verilmektedir.



Şekil 1.3. SPN Ağının Bir Döngüsü

Feistel (1973) tarafından IBM'de geliştirilen Feistel ağları, blok şifreleri oluşturmak için başka bir yaklaşım sunar. Feistel ağlarının SPN'ye göre bir avantajı, temeldeki işlevlerin tersine çevrilebilir olmasını gerektirmemesidir. Böylece bir Feistel ağı, tersinir olmayan bileşenlerden ters çevrilebilir bir döngü fonksiyonunun oluşturulabilmesini sağlar. Feistel ağı, yinelenen bir şifre olarak çalışır. Her döngüde

girdi, sağ(R) ve sol(L) olmak üzere iki kola ayrılır. i. döngüde, sağ koldaki x_{i-1}^R değeri ve döngü anahtarı k_i, anahtarlanmış döngü fonksiyonu F'e girdi olarak verilir. Sonuç, sağ koldaki x_i^R çıktısını elde etmek üzere, sol koldaki x_{i-1}^L değeri ile birleştirilir. Sol kolun çıktısı olan X_i^L ise, x_{i-1}^R değerinin aynısını alır. Son döngüde bu değişim işlemi uygulanmaz. Döngü fonksiyonları tipik olarak S-kutuları ve doğrusal dönüşümler gibi bileşenlerden oluşturulur. Feistel ağına dayanan en dikkate değer blok şifreleme, DES şifreleme algoritmasıdır. Feistel ağına ait bir döngünün genel gösterimi, Şekil 1.4'de görülmektedir.



Şekil 1.4. Feistel Ağının Bir Döngüsü

1.4 Blok Şifrelerin Kriptografik Bileşenleri

Bir blok şifre 3 ana bileşenden oluşmaktadır.

- **Yer Değiştirme Katmanı:** Şifrenin doğrusal olmayan katmanıdır. Yer değiştirme kutuları (Substitution-boxes) ya da diğer adıyla S-kutuları (S-boxes) kullanılarak oluşturulur.
- **Yayılım Katmanı:** Şifrenin doğrusal katmanıdır. Permütasyon katmanı olarak da bilinir. Doğrusal dönüşüm veya dönüşümler kullanılarak oluşturulur.
- **Anahtar Planlama Katmanı:** Gizli anahtardan, her döngüde farklı alt anahtarların üretilmesini sağlayan bir algoritma kullanılarak oluşturulur.

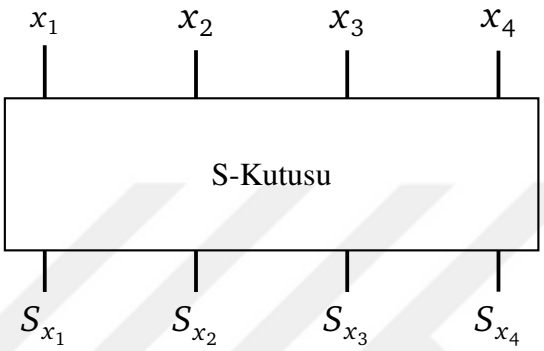
1.4.1 Yer Değiştirme Kutuları

$S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$ şeklinde gösterilen bir yer değiştirme kutusu, $n-bit$ giriş ve $m-bit$ çıkışa sahip vektörel bir Boole fonksiyonudur. Bir başka deyişle, $n \times m$ değişkenli

$f_0, f_1, \cdots, f_{m-1}$ fonksiyonlarına sahip y, ifade 1.1'deki gibi temsil edilebilir.

$$S(\mathbf{x}) = (f_0(\mathbf{x}), f_1(\mathbf{x}), \cdots, f_{m-1}(\mathbf{x})) \tag{1.1}$$

Buradaki her f fonksiyonu da S-kutusunun koordinat fonksiyonu adı verilir (Carlet vd., 2010). Pek çok blok şifrede çoğunlukla 4×4 ya da 8×8 gibi aynı giriş ve çıkış bitlerine sahip S-kutuları tercih edilse de farklı boyutların kullanılması da mümkündür. Şekil 1.5'de, şematik olarak bir S-kutusu gösterilmektedir.



Şekil 1.5. 4×4 Boyutunda Bir S-kutusunun Şematik Gösterimi

S-kutuları, blok şifrelerin tasarımında doğrusal olmama kriterini sağlamak için yaygın olarak kullanılmaktadır. Genellikle, S-kutuları bir tasarımdaki tek doğrusal olmayan bileşendir. Bu nedenle, şifrenin güvenliği büyük ölçüde S-kutularının özelliklerine bağlıdır.

1.4.2 Doğrusal Dönüşümler

Doğrusal dönüşümler, sabit uzunluktaki bir giriş bloğunu doğrusal olarak karıştırıp aynı uzunlukta bir çıkış bloğu elde edilmesini sağlayarak şifreye yayılım özelliğini kazandıran kriptografik bileşenlerdir. Şifrenin doğrusal ve diferansiyel kriptoanalize karşı dayanıklı olabilmesi için, çoğunlukla S-kutuları gibi doğrusal olmayan yapılar ile birlikte kullanılırlar. Bir doğrusal dönüşümün kriptografik bir yapıda kullanılmasını belirleyen önemli ölçütler aşağıdaki gibi verilebilir:

- **Çığ Etkisi:** Giriş bloğundaki 1 bitin değiştirilmesi, çıkış bloğunun yarısının değişmesine neden olmalıdır (Feistel, 1973).
- **Katı Çığ Etkisi:** Giriş bloğundaki tek 1 bitin değiştirilmesi, her çıkış bitinin %50 olasılıkla değişmesini sağlamalıdır (Webster & Tavares, 1985).
- **Bütünlük özelliği:** Çıkış bitlerinin her biri, giriş bitlerine bağımlı olmalıdır (Kam & Davida, 1979).

- **Dal Sayısı:** İkili doğrusal dönüşümler için, 0 giriş vektörü hariç diğer giriş vektörlerinin her birinin Hamming ağırlığı ile çıkış vektörlerinin her birinin Hamming ağırlığının toplamının minimum değeridir (Deamen & Rijmen, 2002). Bu kriter, bir blok şifrede azami yayılıma ulaşılmasını sağlayan en önemli kriptografik özelliktir. Doğrusal ve diferansiyel dal sayısı olmak üzere iki şekilde hesaplanır. Doğrusal dal sayısı bir blok şifrenin doğrusal kriptoanalize, diferansiyel dal sayısı ise diferansiyel kriptoanalize karşı dayanıklılığının ölçüsüdür.

  Modern blok şifrelerin büyük bir kısmı, doğrusal dönüşüm katmanı ya da bu katmanın esas bileşeni olarak maksimum uzaklıkta ayrılabilen (Maximum Distance Separable-MDS) ve maksimum uzaklıkta ikili doğrusal (Maximum Distance Binary Linear-MDBL) matrisler kullanırlar. Bu matrisler, genellikle yayılım ya da difüzyon matrisleri olarak anılırlar. Örneğin,

  - AES: $\mathbb{F}_{2^8}$ cismi üzerinde tanımlı 4×4 boyutunda dairesel bir MDS matris,
  - Khazad (P. Barreto & Rijmen, 2000): $\mathbb{F}_{2^8}$ cismi üzerinde tanımlı 8×8 boyutunda tersi kendisine eşit Hadamard bir MDS matris,
  - Camellia (Aoki vd., 2000): $\mathbb{F}_{2^8}$ cismi üzerinde tanımlı 8×8 boyutunda bir MDBL matris,
  - ARIA (Kwon vd., 2003): $\mathbb{F}_{2^8}$ cismi üzerinde tanımlı tersi kendisine eşit 16×16 boyutunda bir MDBL matris

  doğrusal dönüşüm katmanının ana bileşeni olarak kullanılmıştır.

- **Sabit Nokta Sayısı:** Bir doğrusal dönüşüm, giriş bloğundaki bitleri çıkış bloğuna değiştirmeden aktarıyorsa, bu giriş bloğuna o doğrusal dönüşümün sabit noktası denir (Z'aba, 2010). Kriptografik özellikleri iyi bir doğrusal dönüşümde sabit nokta sayısının 1 olması beklenir ve bu sayının aşılması doğrusal dönüşümün zayıf yayılıma sahip olduğunu gösterir.

Bu kriterlere ek olarak, bir MDS ya da MDBL matrisin donanım uygulaması bağlamında önemli bir ölçüt olan XOR sayısından da (Khoo vd., 2014) bahsetmek gerekmektedir. Özellikle Nesnelerin İnterneti (Internet of Things-IoT) teknolojisinin günden güne gelişmesi, akıllı telefonlar, tabletler, arabalar, saatler, kahve makinaları, süpürgeler ve dahi elektrik ampülleri gibi akla gelebilecek hemen her nesnenin çevrimiçi yetenekler kazanmasını sağlamaktadır. Bu cihazların çoğunlukla, asgari güç tüketimine sahip ve düşük maliyetli donanımlardan oluşması istenmektedir. Güç tüketimindeki bu hassasiyet ve kısıtlı donanım seçenekleri, IoT cihazların haberleşmesinin güvenilirliğini sağlayan kriptografik sistemlerin de bu yönde evrilmesine neden olmaktadır. Hafif sıklet

(Lightweight) kriptografi olarak bilinen bu alan, kısıtlı donanımlarda çalışabilecek düşük maliyetli kriptografik bileşenlerin araştırılması üzerine çalışmakta ve önemi giderek artmaktadır. XOR sayısı metriği de bu bileşenlerden yayılım matrislerinin belirlenmesinde kullanılan en önemli kriterlerden biridir.

1.4.3 Anahtar Planlama Algoritmaları

Anahtar planlama algoritmaları, bir blok şifrenin her döngüsünde birbirinden farklı alt anahtarların kulllanılmasını sağlamaktadır. Bu alt anahtarlar, gizli ana anahtardan söz konusu algoritmalar ile elde edilmektedir. Tasarımında zaaflar bulunan anahtar planlama algoritmaları yavaş yayılım ve bit sızıntısına (bit leakage) (May vd., 2002) neden olabilmektedir. Bu da ilişkili anahtar saldırıları olarak bilinen saldırılara sebep vermektedir (Biham, 1994). Yavaş yayılım zaafiyetini test etmek için katı çığ kriteri kullanılmaktadır. Herhangi bir saldırı sonucu ele geçirilen bir alt anahtardan, gizli ana anahtar veya diğer alt anahtarların ya da bunların bir kısmının elde edilebilmesi ise bit sızıntısı olarak tanımlanmaktadır. Bit sızıntısı zaafiyeti, alt anahtarların birbirinden bağımsız olarak üretilmesiyle giderilebilmektedir.

1.5 Tezin Ana Hatları, Gerekçesi ve Önemi

Literatürde, blok şifrelerin doğrusal dönüşüm katmanında kullanılabilecek, kriptografik özellikleri iyi ve verimli yayılım matrislerinin tasarımı her zaman güncel bir problem olmuştur. Özellikle sınırlı yazılım ve donanım bileşenlerine sahip, düşük güç tüketmesi beklenen kısıtlı sistemler üzerinde güvenlik ve verimlilik arasındaki dengenin sağlanabilmesi gerekir. Bu sebeple, hafif sıklet blok şifreler için MDS veya MDBL matrisleri kullanan involutif (tersi kendisine eşit) yayılım tabakalarının tasarımı gün geçtikçe önem kazanmaktadır. İnvolutif yayılım tabakaları, şifreleme ve şifre çözme işlemlerinde aynı yapıyı kullandığından hem yer maliyetlerini düşürebilir, hem de şifreleme ve şifre çözme işlemlerinin aynı ya da yakın maliyete sahip olmasını sağlarlar. Hafif sıklet şifrelere uygun involutif MDS veya MDBL yayılım katmanlarının tasarlanması, şifrenin çalışacağı donanımın sahip olduğu kısıtlamalar da göz önüne alındığında oldukça zorlu bir süreç haline gelmektedir (Sarkar & Syed, 2016). Bu duruma alandaki kriptografik algoritmaların eksikliğinin de eklenmesi, hafif sıklet kriptografiyi popüler bir çalışma alanı haline getirmiştir.

Söz konusu sorunların göz önüne alındığı bu tez, blok şifrelerin doğrusal dönüşüm katmanında kullanılabilecek MDS ve MDBL matrisleri üreten yeni bir tasarım tekniği üzerine bir çalışma içermektedir. Yeni teknik, maksimum yayılıma ve düşük XOR maliyetine sahip yayılım katmanlarının üretilmesinde önem taşımaktadır.

Tez yedi bölüm olarak planlanmıştır. Kriptoloji ve alt sistemleriyle ilgili genel kavramları içeren bu bölümün ardından, ikinci bölümde tezde kullanılan matematiksel kavramların anlaşılabilmesi adına temel bir altyapı verilecektir. Üçüncü bölüm, literatürde MDS ve MDBL matrislerin elde edilmesinde kullanılan tasarım tekniklerinin açıklamalarıyla birlikte, XOR sayısı değerinin önemini, detaylandırılmasını ve iyileştirilmesiyle ilgili yapılan çalışmaları içermektedir. Dördüncü bölümde, blok şifrelerde doğrusal dönüşüm katmanı görevi üstlenebilecek, kriptografik özellikleri iyi, XOR maliyeti düşük MDS ve MDBL yayılım matrislerinin tasarımında kullanılabilecek yeni bir teknik tanıtılmaktadır. Beşinci ve altıncı bölüm sırasıyla, söz konusu teknik ile $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$ ve $\mathbb{F}_{2^8}$ cisimleri üzerinde elde edilen MDS matrislere ait sonuçlar ile, $\mathbb{F}_2$ cismi üzerinde elde edilen 12×12 ve 16×16 boyutlarında MDBL matrislere ait sonuçları içermektedir. Yedinci ve son bölüm ise, tüm sonuçların değerlendirilmesini, yeni tekniğin literatürdeki diğer tekniklerle karşılaştırılmasını ve gelecekte ortaya çıkabilecek açık problemlerin tartışılmasını kapsamaktadır.

BÖLÜM 2

MATEMATİKSEL ALTYAPI

Bu bölümde, tezde kullanılan tekniği oluşturmak için faydalanılan ve kriptografik alanda geniş kullanıma sahip olan çeşitli matematiksel ifadelerin kapsamlı açıklamaları verilmektedir. Bu bölümde verilen tanım, teorem, önerme ve açıklamaların ispatlarına Forouzan ve Mukhopadhyay (2015), Lidl ve Niederreiter (1994), Ling ve Xing (2004), Nakahara Jr ve Abrahao (2009), Stallings (2017) ve Stinson (2002)'den ulaşılabilir.

Tanım 2.1 (Grup)**.** Bir G kümesi üzerinde $\oplus$ işaretiyle gösterilen bir ikili işlem tanımlı olsun. Eğer bu işlem, grup aksiyomları adı verilen aşağıdaki dört özelliği sağlıyor ise G yapısına bir grup denir ve $\{G, \oplus\}$ notasyonu ile gösterilir.

(G1) *Kapalılık Özelliği:* $a, b \in G$ ise $a \oplus b \in G$'dir.

(G2) *Birleşme Özelliği:* Her $a, b, c \in G$ için $(a \oplus b) \oplus c = a \oplus (b \oplus c)$'dir.

(G3) *Birim Eleman:* Her $a \in G$ için $a \oplus e = e \oplus a$ olacak şekilde bir $e \in G$ vardır.

(G4) *Ters Eleman:* Her $a \in G$ için öyle bir $a' \in G$ elemanı vardır ki $a \oplus a' = a' \oplus a = e$ olur.

Tanım 2.2 (Değişmeli Grup)**.** Tanım 2.1'de verilen dört aksiyoma ek olarak aşağıdaki beşinci aksiyomu da sağlayan G kümesine değişmeli (abelyan) grup adı verilir.

(G5) *Değişme Özelliği:* Her $a, b \in G$ için $a \oplus b = b \oplus a$'dir.

Tanım 2.3 (Devirli Grup)**.** Bir grubun tüm elemanları, bu grubun bir elemanına grup işleminin devamlı uygulanmasıyla elde edilebiliyorsa bu gruba devirli (cylic) grup, bu elemana ise üreteç (generator) eleman adı verilmektedir. Bununla birlikte bir devirli grup aynı zamanda bir değişmeli gruptur.

Tanım 2.4 (Halka)**.** Bir H kümesi üzerinde $\oplus$ ve $\otimes$ işaretleriyle gösterilen iki adet ikili işlem tanımlı olsun. Eğer bu işlem, halka aksiyomları adı verilen aşağıdaki dört özelliği sağlıyor ise H yapısına bir halka denir ve $\{H, \oplus, \otimes\}$ notasyonu ile gösterilir.

(H1) $\oplus$ *İşlemi Üzerinde Grup Özelliği:* $\{H, \oplus\}$ bir değişmeli gruptur. Diğer bir deyişle; H kümesi, $\oplus$ işlemi üzerinde Tanım 2.1 ve Tanım 2.2'de verilen beş aksiyomu $(G1 - G5)$ da sağlar.

(H2) $\otimes$ *İşlemi Üzerinde Kapalılık Özelliği:* $a, b \in H$ ise $a \otimes b \in H$'dir.

(H3) $\otimes$ *İşlemi Üzerinde Birleşme Özelliği:* Her $a, b, c \in H$ için $(a \otimes b) \otimes c = a \otimes (b \otimes c)$'dir.

(H4) $\otimes$ *İşleminin* $\oplus$ *İşlemi Üzerinde Dağılma Özelliği:* Her $a, b, c \in H$ için $a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$ ve $(a \oplus b) \otimes c = a \otimes c \oplus b \otimes c$'dir.

Tanım 2.5 (Değişmeli Halka)**.** Tanım 2.4'de verilen dört aksiyoma ek olarak aşağıdaki beşinci aksiyomu da sağlayan H kümesine değişmeli halka adı verilir.

(H5) $\otimes$ *İşlemi Üzerinde Değişme Özelliği:* Her $a, b \in H$ için $a \otimes b = b \otimes a$'dir.

Tanım 2.6 (Tamlık Bölgesi)**.** Bir T kümesi üzerinde $\oplus$ ve $\otimes$ işaretleriyle gösterilen iki adet ikili işlem tanımlı olsun. Eğer bu işlem, tamlık bölgesi aksiyomları adı verilen aşağıdaki üç özelliği sağlıyor ise T yapısına bir tamlık bölgesi denir ve $\{T, \oplus, \otimes\}$ notasyonu ile gösterilir.

(T1) *Değişmeli Halka Özelliği:* $\{T, \oplus, \otimes\}$ bir değişmeli halkadır. Diğer bir deyişle; T kümesi, $\oplus$ ve $\otimes$ işlemi üzerinde Tanım 2.4 ve Tanım 2.5'de verilen beş aksiyomu $(H1 - H5)$ da sağlar.

(T2) $\otimes$ *İşlemi Üzerinde Birim Eleman:* Her $a \in T$ için $a \otimes e = e \otimes a = e$ olacak şekilde bir $e \in T$ vardır.

(T3) $\otimes$ *İşlemi Üzerinde Sıfır Böleni İçermeme Özelliği:* Eğer $a, b \in T$ ve $a \otimes b = 0$ ise a ya da b elemanlarından biri mutlaka 0 olur.

Örnek 2.1. *$\mathbb{Z}$ tam sayılar kümesi, $+$ (toplama) ve $*$ (çarpma) işlemine göre bir tamlık bölgesidir.*

Tanım 2.7 (Cisim)**.** Bir F kümesi üzerinde $\oplus$ ve $\otimes$ işaretleriyle gösterilen iki adet ikili işlem tanımlı olsun. Eğer bu işlem, cisim aksiyomları adı verilen aşağıdaki iki özelliği sağlıyor ise F yapısına bir cisim denir ve $\{F, \oplus, \otimes\}$ notasyonu ile gösterilir.

(F1) *Tamlık Bölgesi Özelliği:* F bir tamlık bölgesidir. Diğer bir deyişle; F kümesi, Tanım 2.6'da verilen üç aksiyomu $(T1-T3)$ da sağlar.

(F2) $\otimes$ *İşlemi Üzerinde Ters Eleman:* $a \neq 0$ olmak üzere, her $a \in F$ için $a \otimes a^{-1} = a^{-1} \otimes a = 1$ eşitliğini sağlayan bir $a^{-1} \in F$ vardır.

Örnek 2.2. $\mathbb{Q}$ *rasyonel sayılar kümesi,* $+$ *(toplama) ve* $*$ *(çarpma) işlemine göre bir cisimdir.*

Teorem 2.1. *Eğer* p *bir asal sayı ise* $\mathbb{Z}_p$ *bir cisim oluşturur.*

Kanıt. p asal bir sayı olsun. Herhangi bir sıfır olmayan eleman $a \in \mathbb{Z}_p$ ve $0 < a < p$ için, a ve p aralarında asal olur. Böylece $ax + py = 1$ olacak şekilde $x(0 \leq x \leq p-1)$ ve y tam sayıları vardır. Buna ek olarak $ax \equiv 1 mod p$ olduğundan Tanım 2.7'de verilen $(F2)$ aksiyomu sağlanmış olur. Dolayısıyla $\mathbb{Z}_p$ bir cisimdir. ■

Tanım 2.8 (Sonlu Cisim)**.** Eğer bir cisim sonlu sayıda elemana sahipse sonlu cisim olarak adlandırılmaktadır. Sonlu cisimler, p bir asal sayı, n ise bir pozitif tam sayı olmak üzere $\mathbb{F}_{p^n}$ ya da $GF(p^n)$ (Galois Cismi) notasyonları ile gösterilmektedirler. Sonlu bir cisimdeki elemanların sayısına ise cismin derecesi veya mertebesi denilmektedir.

Örnek 2.3. *Bazı sonlu cisimlere ait örnekler aşağıdaki gibi verilebilir.*

- $\mathbb{F}_2$*: Elemanları* $\{0,1\}$ *olan sonlu cisim*
- $\mathbb{F}_{2^4}$*: Elemanları* $\{0,1\}$ *olan 4 bitlik genişletilmiş sonlu cisim*
- $\mathbb{F}_3$*: Elemanları* $\{0,1,2\}$ *olan sonlu cisim*

Tanım 2.9 (Polinom)**.** $n \geq 0$ şeklinde bir tam sayı olmak üzere bir polinom, Denklem 2.1'de gösterildiği gibi tanımlanmaktadır.

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i \tag{2.1}$$

Burada $a_n \neq 0$ olmak üzere a_i, S notasyonu ile gösterilen katsayılar kümesini temsil etmektedir.

Sıfır dereceli bir polinom, sabit polinom olarak adlandırılır ve basitçe, S katsayılar kümesinin bir öğesini göstermektedir. Eğer $a_n = 1$ ise, bu polinoma da monik polinom denilmektedir. Katsayıların bir halka veya cismin elemanı olduğu polinomlara ise sırasıyla polinom tabanlı halka ve polinom tabanlı cisim adı verilmektedir.

Tanım 2.10 (Polinom Aritmetiği)**.** Kriptografide, x gibi tek bir değişkene sahip polinomlarla işlem yapılmaktadır ve polinom aritmetiği bağlamında üç farklı sınıftan söz etmek mümkündür.

i) Cebirin temel kurallarını kullanan sıradan polinom aritmetiği.

ii) Katsayıların $\mathbb{F}_p$ cinsinden olduğu ve işlemlerin $mod p$ üzerinden gerçekleştirildiği polinom aritmetiği.

iii) Katsayıların $\mathbb{F}_p$ cinsinden olduğu ve en yüksek derecesi n şeklinde bir tam sayı olan $m(x)$ polinomunun mod işlemlerinde kullanıldığı polinom aritmetiği.

x değişkeninin S katsayılar kümesinin doğal bir elemanı olduğu düşünülerek polinom aritmetiğinde toplama, çıkarma ve çarpma işlemleri tanımlanabilmektedir. Ancak bölme işleminin tanımlanabilmesi için S katsayılar kümesinin bir cisim olması gerekmektedir.

Tanım 2.11 (İndirgenebilir ve İndirgenemez Polinomlar)**.** Bir $f(x)$ polinomu, $deg(g(x)) < deg(f(x))$ ve $deg(h(x)) < deg(f(x))$ olmak üzere $f(x) = g(x).h(x)$ şeklinde ifade edilebiliyorsa indirgenebilir polinom, aksi taktirde ise indirgenemez (asal) polinom olarak tanımlanmaktadır.

Teorem 2.2. *$f(x)$ derecesi 1'den büyük olmak üzere bir $\mathbb{F}$ cismi üzerine polinom olsun. O zaman $\mathbb{F}[x]/f(x)$ toplama ve çarpma işlemi ile birlikte bir polinom tabanlı halka oluşturur. Bunun ötesinde eğer $f(x)$ bir indirgenemez polinom ise $\mathbb{F}[x]/f(x)$ bir polinom tabanlı cisim oluşturur.*

Bu teoreme göre $\mathbb{F}[x]/f(x)$, $mod f(x)$'e göre indirgeme sonucu oluşan sistemi ifade etmektedir.

Örnek 2.4. *$\mathbb{Z}_2/(1+x^2)$ için toplama ve çarpma işlemleri aşağıdaki gibidir.*

+	0	1	x	1+x
0	0	1	x	1+x
1	1	0	1+x	x
x	x	1+x	0	1
1+x	1+x	x	1	0

*	0	1	x	1+x
0	0	0	0	0
1	0	1	x	1+x
x	0	x	1	1+x
1+x	0	1+x	1+x	0

$1+x^2$ polinomu indirgenebilir bir polinom olduğundan çarpma işlemine göre, $1+x$ elemanının tersi bulunmamaktadır. Dolayısıyla bu yapı bir polinom tabanlı halkadır.

Örnek 2.5. $\mathbb{Z}_2/(1+x+x^2)$ *için toplama ve çarpma işlemleri aşağıdaki gibidir.*

+	*0*	*1*	*x*	*1+x*
0	*0*	*1*	*x*	*1+x*
1	*1*	*0*	*1+x*	*x*
x	*x*	*1+x*	*0*	*1*
1+x	*1+x*	*x*	*1*	*0*

*	*0*	*1*	*x*	*1+x*
0	*0*	*0*	*0*	*0*
1	*0*	*1*	*x*	*1+x*
x	*0*	*x*	*1+x*	*1*
1+x	*0*	*1+x*	*1*	*x*

$1+x+x^2$ *polinomu indirgenemez bir polinom olduğundan çarpma işlemine göre,* 0 *elemanı hariç her elemanın tersi bulunmaktadır. Bu sebeple söz konusu yapı bir polinom tabanlı cisimdir.*

Tez çalışması boyunca, $\mathbb{F}_{2^m}/p(x)$ notasyonu ile ifade edilen indirgenez polinomlar, gösterim kolaylığı sağlaması açısından onaltılık gösterimle verilmektedir. Örneğin; $\mathbb{F}_{2^3}/x^3+x+1$ polinomu, $4-bit$ ikili vektör dizisi olarak (1011) şeklinde gösterilmektedir. (1011) ikili gösteriminin onaltılık tabanda karşılığı ise $0xb$ olmaktadır. Benzer şekilde;

$$\mathbb{F}_{2^4}/x^4+x+1 \text{ polinomu} \Rightarrow (10011) \Rightarrow 0x13,$$
$$\mathbb{F}_{2^8}/x^8+x^5+x^3+x^2+1 \text{ polinomu} \Rightarrow (100101101) \Rightarrow 0x12d$$

şeklinde ifade edilmektedir. İndirgenemez polinomların yanında, α ilkel elemanı kullanılarak elde edilen bir $\mathbb{F}_{2^m}/p(x)$ sonlu cisminin elemanlarının gösterimi de çalışma boyunca onaltılık tabanda yapılmaktadır. Örnek olarak, Çizelge 2.1, $\mathbb{F}_{2^4}/x^4+x+1$ sonlu cisminin elemanlarını göstermektedir.

Çizelge 2.1. $\mathbb{F}_{2^4}/x^4+x+1$ Sonlu Cisminin Elemanları

Onaltılık Tabanda Gösterim	**Sonlu Cisim Elemanı**
1_h	$\alpha^0 = 1$
2_h	$\alpha^1 = \alpha$
3_h	$\alpha^4 = \alpha + 1$
4_h	$\alpha^2 = \alpha^2$
5_h	$\alpha^8 = \alpha^2 + 1$
6_h	$\alpha^5 = \alpha^2 + \alpha$
7_h	$\alpha^{10} = \alpha^2 + \alpha + 1$
8_h	$\alpha^3 = \alpha^3$
9_h	$\alpha^{14} = \alpha^3 + 1$
A_h	$\alpha^9 = \alpha^3 + \alpha$
B_h	$\alpha^7 = \alpha^3 + \alpha + 1$
C_h	$\alpha^6 = \alpha^3 + \alpha^2$
D_h	$\alpha^{13} = \alpha^3 + \alpha^2 + 1$
E_h	$\alpha^{11} = \alpha^3 + \alpha^2 + \alpha$
F_h	$\alpha^{12} = \alpha^3 + \alpha^2 + \alpha + 1$

$\mathbb{F}_{2^m}/p(x)$ sonlu cisminin elemanları ile toplama işlemi yapılırken $\bmod 2$, çarpma işlemi yapılırken ise $\bmod p(x)$ kullanılarak indirgeme yapılmaktadır.

Örnek 2.6. $\mathbb{F}_{2^4}/0x13$ *sonlu cisminin* 8_h *ve* C_h *elemanlarının toplamı;* $8_h = \alpha^3$ *ve* $C_h = \alpha^3 + \alpha^2$ *olduğundan,* $\alpha^3 + \alpha^3 + \alpha^2 = \alpha^2 = 4_h$ *olarak bulunur.*

Aynı cisim üzerinde 8_h *ve* C_h *elemanlarının çarpımı ise;* $(\alpha^3) \times (\alpha^3 + \alpha^2) = \alpha^6 + \alpha^5 = \alpha^3 + \alpha^2 + \alpha^2 + \alpha = \alpha^3 + \alpha = A_h$ *olarak bulunur.*

Tanım 2.12 (Doğrusal Dönüşüm)**.** Elemanları $A : (\{0,1\}^m)^n \rightarrow (\{0,1\}^m)^n$ olmak üzere bir doğrusal dönüşüm, 2.2 ifadesindeki gibi tanımlanmaktadır.

$$A(x) = A.x \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{2.2}$$

(a_{ij}) doğrusal dönüşüm elemanları için $m = 1$ alınırsa, diğer bir deyişle elemanlar $\mathbb{F}_2$ cisminden seçilirse, bu doğrusal dönüşüme ikili doğrusal dönüşüm denilmektedir.

Tanım 2.13 (Hamming Ağırlığı)**.** $wt(C)$ notasyonu ile gösterilen Hamming ağırlığı, herhangi bir C kod kelimesindeki 1'lerin sayısı olarak tanımlanmaktadır. Örneğin $C = \{0,1,1,1,1,0,0\}$ kod kelimesinin Hamming ağırlığı 4 olmaktadır.

Tanım 2.14 (Hamming Uzaklığı)**.** $\mathbb{F}_{2^m}$ cisminden seçilen n boyutlu iki kod kelimesinin farklı elemanlarının sayısı olarak tanımlanmaktadır. Örneğin $C_1 = \{1,0,\mathbf{1},1,\mathbf{1},0,1\}$ ile $C_2 = \{1,0,\mathbf{0},1,\mathbf{0},0,1\}$ kod kelimelerinin Hamming uzaklığı 2 olmaktadır.

Tanım 2.15 (Diferansiyel Dal Sayısı)**.** $n \times n$ boyutunda bir A matrisinin diferansiyel dal sayısı (differential branch number) β_d, ifade 2.3'deki gibi tanımlanmaktadır.

$$\beta_d(A) = min\{wt(x) + wt(A.x^T) | x \in (\{0,1\}^m)^n, x \neq 0\} \tag{2.3}$$

Tanım 2.16 (Doğrusal Dal Sayısı)**.** $n \times n$ boyutunda bir A matrisinin doğrusal dal sayısı (linear branch number) β_l, ifade 2.4'deki gibi tanımlanmaktadır.

$$\beta_l(A) = min\{wt(x) + wt(A^T.x^T) | x \in (\{0,1\}^m)^n, x \neq 0\} \tag{2.4}$$

2.3 ve 2.4 ifadeleri uyarınca, bir ikili doğrusal dönüşümün dal sayısı, 0 giriş vektörü hariç diğer tüm giriş vektörleri ile tüm çıkış vektörlerinin her birinin Hamming ağırlıkları

toplamlarının minimum değerine eşit olmaktadır (Deamen & Rijmen, 2002). Ayrıca, bir doğrusal dönüşümün dal sayısı, ardışık iki döngüde bulunan aktif S-kutularının minimum sınırıdır.

Literatürde maksimum dal sayısına sahip doğrusal dönüşüm katmanları tasarlamak için özel matris formlarından faydalanılmaktadır. Üçüncü bölümde detaylandırılacak olan tasarım tekniklerinde kullanılan bu özel matris formları, Tanım 2.17 − 2.31 arasında verilmektedir.

Tanım 2.17 (Birim Matris)**.** Ana köşegeni oluşturan elemanları 1, diğer tüm elemanları 0 olan $n \times n$ boyutunda kare matrislere birim matris denilmektedir ve I_n ya da I notasyonu ile gösterilmektedirler.

Tanım 2.18 (Köşegen Matris)**.** Eğer $i \neq j$ olmak üzere $a_{ij} = 0$ ise $n \times n$ boyutlarında $A = [a_{ij}]$ matrisine köşegen (diyagonal) matris denilmektedir. Köşegen bir matriste ana köşegenin elemanı olmayan tüm elemanlar 0'dır.

Tanım 2.19 (Simetrik Matris)**.** Tüm terimleri reel olan A matrisi için; $A^T = A$ eşitliği yazılabiliyorsa bu A matrisine simetrik matris adı verilmektedir.

Tanım 2.20 (İnvolutif Matris)**.** Eğer bir A matrisinin tersi kendisine eşit ise ($A = A^{-1}$), bu A matrisine involutif matris adı verilmektedir.

Tanım 2.21 (Ortogonal Matris)**.** $n \times n$ boyutlarında bir kare matris olan A matrisinin kendisi ile transpozunun çarpımı birim matrisi veriyorsa ($A \cdot A^T = I$), bu A matrisine ortogonal matris denilmektedir.

Tanım 2.22 (Singüler Olmayan Matris)**.** Eğer $AB = BA = I$ olacak şekilde $n \times n$ boyutunda bir B matrisi mevcut ise $n \times n$ boyutundaki A matrisi, singüler olmayan (nonsingular) matris olarak isimlendirilmektedir.

Tanım 2.23 (Dairesel Matris)**.** Dairesel olarak matrisin her satırının 1 pozisyon sağa ötelenmesi ile elde edilmektedirler. $a_{ij} = x_0, x_1, \cdots, x_{n-1}$ olmak üzere, bir Dairesel matris olan C, ifade 2.5'deki gibi gösterilmektedir (Davis, 1979).

$$C = circ(x_0, x_1, \cdots, x_{n-1}) = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-1} \\ x_{n-1} & x_0 & x_1 & \cdots & x_{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{bmatrix} \tag{2.5}$$

Tanım 2.24 (Hadamard Matris)**.** A_0 ve A_1, $2^{n-1} \times 2^{n-1}$ sonlu cisim Hadamard matrisleri olmak üzere, $2^n \times 2^n$ boyutunda bir sonlu cisim Hadamard matris olan H, ifade 2.6'daki

gibi gösterilmekte ve kısaca Hadamard matris olarak adlandırılmaktadır. (P. S. Barreto, 2000).

$$H = had(A_0, A_1) = \begin{bmatrix} A_0 & A_1 \\ A_1 & A_0 \end{bmatrix} \tag{2.6}$$

Tanım 2.25 (Genel Hadamard Matris)**.** $\mathbb{F}_{2^m}$ cisminde tanımlı ve b_1, $\mathbb{F}_{2^m}$ cisminin sıfır olmayan bir elemanı olmak üzere, Tanım 2.24'da belirtilen H matrisine karşılık gelen 2×2 boyutlarında Genel Hadamard matrisi $Ghad(a_0, a_1; b_1)$, 2.7'deki gibi gösterilmektedir (Pehlivanoğlu, Sakallı, Akleylek, Duru ve Rijmen, 2018).

$$GH = Ghad(a_0, a_1; b_1) = \begin{bmatrix} a_0 & a_1 b_1 \\ a_1 b_1^{-1} & a_0 \end{bmatrix} \tag{2.7}$$

Tanım 2.24'da belirtilen H matrisine karşılık gelen 4×4 boyutlarında Genel Hadamard matrisi $Ghad(a_0, a_1; b_1, a_2; b_2, a_3; b_3)$ ise 2.8'deki gibi gösterilmektedir.

$$GH = Ghad(a_0, a_1; b_1, a_2; b_2, a_3; b_3) = \begin{bmatrix} a_0 & a_1 b_1 & a_2 b_2 & a_3 b_3 \\ a_1 b_1^{-1} & a_0 & a_3 b_1^{-1} b_2 & a_2 b_1^{-1} b_3 \\ a_2 b_2^{-1} & a_3 b_2^{-1} & a_0 & a_1 b_2^{-1} b_3 \\ a_3 b_3^{-1} & a_2 b_3^{-1} b_1 & a_1 b_3^{-1} b_2 & a_0 \end{bmatrix} \tag{2.8}$$

Tanım 2.26 (Vandermonde Matris)**.** $a_{ij} = x_j^i$ olmak üzere, n mertebesinde bir kare Vandermonde matris olan V, ifade 2.9'daki gibi gösterilmektedir.

$$V = vand(x_0, x_1, x_2, \cdots, x_{n-1}) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_{n-1} \\ x_0^2 & x_1^2 & \cdots & x_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{n-1} & x_1^{n-1} & \cdots & x_{n-1}^{n-1} \end{bmatrix} \tag{2.9}$$

Tanım 2.27 (Cauchy Matris)**.** $\{x_0, x_1, \cdots, x_{n-1}\}, \{y_0, y_1, \cdots, y_{n-1}\} \in \mathbb{F}_{2^m}$ ve $a_{ij} = \frac{1}{x_i + y_i}$ veya $a_{ij} = \frac{1}{x_i - y_i}$ olmak üzere, tüm $0 \leq i, j \leq n-1$ değerleri için $x_i + y_i \neq 0$ ise bir Cauchy matris olan $C_1(a_{i,j})$ ve $C_2(a_{i,j})$, ifade 2.10'daki gibi gösterilmektedir.

$$C_1 = \begin{bmatrix} \frac{1}{x_0+y_0} & \frac{1}{x_0+y_1} & \cdots & \frac{1}{x_0+y_{n-1}} \\ \frac{1}{x_1+y_0} & \frac{1}{x_1+y_1} & \cdots & \frac{1}{x_1+y_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{x_{n-1}+y_0} & \frac{1}{x_{n-1}+y_1} & \cdots & \frac{1}{x_{n-1}+y_{n-1}} \end{bmatrix} C_2 = \begin{bmatrix} \frac{1}{x_0-y_0} & \frac{1}{x_0-y_1} & \cdots & \frac{1}{x_0-y_{n-1}} \\ \frac{1}{x_1-y_0} & \frac{1}{x_1-y_1} & \cdots & \frac{1}{x_1-y_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{x_{n-1}-y_0} & \frac{1}{x_{n-1}-y_1} & \cdots & \frac{1}{x_{n-1}-y_{n-1}} \end{bmatrix} \tag{2.10}$$

Tanım 2.28 (Toeplitz Matris)**.** $a_{ij} = x_{i-j}$ olmak üzere, $n \times n$ boyutunda bir Toeplitz matris olan T, ifade 2.11'deki gibi gösterilmektedir.

$$T = \begin{bmatrix} x_0 & x_{-1} & x_{-2} & \cdots & x_{-(n-1)} \\ x_1 & x_0 & x_{-1} & \cdots & \vdots \\ x_2 & x_1 & x_0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & \cdots & \cdots & \cdots & x_0 \end{bmatrix} \tag{2.11}$$

Tanım 2.29 (Hankel Matris)**.** $n \times n$ boyutunda bir Hankel matris olan H, ifade 2.12'deki gibi gösterilmektedir.

$$H = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-2} & x_{n-1} \\ x_1 & x_2 & x_3 & \cdots & x_{n-1} & x_n \\ x_2 & x_3 & x_4 & \cdots & x_n & x_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1} & x_n & x_{n+1} & \cdots & x_{2n-3} & x_{2n-2} \end{bmatrix} \tag{2.12}$$

Tanım 2.30 (Companion Matris)**.** k mertebesinde bir monik polinom olan $g(x) = a_0 + a_1 x + \cdots + a_{k-1}x^{k-1} + x^k \in \mathbb{F}_q[x]$ şeklinde tanımlandığında, $\mathbb{F}_q$ cismi üzerinde tanımlı matrisler kümesinin bir elemanı olan C_g Companion matrisi 2.13'deki gibi gösterilmektedir.

$$C_g = Companion(-a_0, -a_1, \cdots, -a_{k-1}) = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 1 \\ -a_0 & -a_1 & \cdots & \cdots & -a_{k-1} \end{bmatrix} \tag{2.13}$$

Tanım 2.31 (DSI Matris)**.** $1 \leq i \leq k$ ve $1 \leq j \leq k-1$ için $a_i, b_i \in \mathbb{F}_{2^m}$, $a = [a_1, a_2, \cdots, a_k]$ ve $b = [b_1, b_2, \cdots, b_{k-1}]$ olmak üzere bir Köşegen-Seri-Tersinir (Diagonal-Serial-Invertible-DSI) matris $D = (d_{i,j}) \in \mathbb{F}_{2^m}$, a ve b vektörlerinin durumuna

bağlı olarak 2.14'deki gibi tanımlanmaktadır (Toh vd., 2018).

$$d_{i,j} = \begin{cases} a_1 & i = 1, j = k \\ a_i, & i = j + 1 \\ b_i, & i = j \leq k - 1 \\ 0, & \text{diğer durumlar.} \end{cases} \tag{2.14}$$

Tanım 2.32 (Singleton Sınırı)**.** $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı bir $[n, k, d]$ kod için;

k : $(\mathbb{F}_{2^m})^n$ cisminin alt uzayının boyutunu,

n : Bu alt uzaydan seçilen vektörlerin eleman sayısını,

d : Seçilen iki vektör arasındaki Hamming uzaklığının minimum değerini göstermektedir.

Bu bağlamda doğrusal $[n, k, d]$ kodlar, $d \leq n - k + 1$ şeklinde gösterilen ve Singleton sınırı olarak bilinen eşitsizliği sağlamaktadırlar.

Tanım 2.33 (MDS Kodlar)**.** $d = n - k + 1$ eşitliğini sağlayan ve bir doğrusal $[n, k, d]$ kod olan C, maksimum uzaklıkta ayrılabilen (Maximum Distance Seperable-MDS) kod olarak adlandırılmaktadır.

Tanım 2.34 (Üreteç Matris)**.** Doğrusal bir $[n, k, d]$ kod olan C için G üreteç matrisi, $k \times n$ boyutunda öyle bir matristir ki satırları C koduna bir taban oluşturmaktadır.

Teorem 2.3. *Bir MDS matris, üreteç matrisi* $G = [I, A]$ *olan bir* $[n, k, d]$ *parametreli kodun* $d = n - k + 1$ *Singleton sınırını sağlayan A matrisidir.*

Tanım 2.35 (MDS Matrisler)**.** MDS matrisler, MDS kodlardan elde edilirler ve bu matrislerin özellikleri aşağıdaki gibi verilebilir:

i) $n \times n$ boyutunda bir A matrisinin tüm kare alt matrislerinin determinantları 0'dan farklı (nonsingular) ise A matrisi bir MDS matristir.

ii) $n \times n$ boyutunda bir MDS matrisin tüm girişleri 0'dan farklı olmak zorundadır.

iii) Eğer $n \times n$ boyutunda bir A matrisi MDS ise A matrisinin dal sayısı $n + 1$ olacaktır.

iv) MDS olduğu bilinen $n \times n$ boyutunda bir A matrisinin transpozesi de bir MDS matristir.

v) $n \times n$ boyutlarında bir MDS matrisin satır veya sütunları $\mathbb{F}_{2^m}$ cisminin 0'dan farklı bir elemanı ile çarpıldığında elde edilen yeni matris de MDS matris olmaktadır. Diğer bir deyişle matrisin MDS olma özelliği korunmaktadır.

Tanım 2.36 (MDBL Kodlar)**.** Elemanları $\mathbb{F}_2$ cisminden olan matrislere ikili (binary) matrisler adı verilmektedir ve $n \times n$ boyutunda ikili matrislerin maksimum dal sayısı, ikili $[2n, n]$ doğrusal kodların maksimum uzaklığına eşittir (Kwon vd., 2005). Bu tanıma ek olarak, ikili bir doğrusal dönüşümün maksimum dal sayısına sahip olabilmesi için bu kodların maksimum uzaklığının alt ve üst sınırlarının birbirine eşit olması gerekmektedir ve bu kodlara maksimum uzaklıkta ikili doğrusal kodlar (Maximum Distance Binary Linear Codes – MDBL) adı verilmektedir.

Çizelge 2.2. İkili $[2n, n]$ Kodların Maksimum Dal Sayıları

n	Erişilebilir Değer	Teorik Sınır	n	Erişilebilir Değer	Teorik 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

Ancak, Çizelge 2.2'den de görülebileceği gibi $n \times n$ boyutundaki MDBL matrislerde, $n > 18$ olduğu durumlar için maksimum dal sayısı tam olarak bilinmemektedir. Bu sebeple, söz konusu boyutlara sahip MDBL matrislerde maksimum dal sayısının erişilebilir değeri ve teorik sınırlarından bahsetmek gerekmektedir. Örneğin; $[64, 32]$ doğrusal kodların $(n = 32)$ maksimum dal sayısının erişilebilir değeri 12 iken teorik sınırı 16'dır. Bir başka deyişle 32×32 bir ikili matrisin erişilebilir dal sayısı 12 olmaktadır.

Tanım 2.37 (Sabit Nokta Sayısı)**.** A bir doğrusal dönüşüm ve x bu doğrusal dönüşüme ait bir giriş bloğu olmak üzere, $A.x = x$ ise, x bu doğrusal dönüşümün sabit noktasıdır

denilmektedir. Bir başka deyişle, A doğrusal dönüşümü giriş vektörünü aynı çıkış vektörüne haritalıyorsa bu giriş vektörüne sabit nokta adı verilir.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{2.15}$$

Doğrusal dönüşüm matrisi A ile birim matris I'nın $n \times n$ boyutunda olduğu ve giriş bitlerinin $m-bit$ değerlerden oluştuğu düşünüldüğünde, A doğrusal dönüşüm matrisinin tüm sabit noktalarının sayısı ifade 2.16'deki gibi elde edilmektedir.

$$(A+I)x^T = 0 \tag{2.16}$$

İfade 2.16, matris rankı bağlamında ifade 2.17'ye dönüştürülerek A doğrusal dönüşümünün sabit nokta sayısı hesaplanmaktadır.

$$FP_A = 2^{m(rank(A)-rank(A+I))} = 2^{m(n-rank(A+I))} \tag{2.17}$$

Tanım 2.38 (XOR Kapısı)**.** Özel VEYA (eXclusive OR) olarak da bilinen XOR kapısı, giriş değerlerinin birbirinin aynısı olduğu durumlarda çıkış olarak 0, giriş değerlerinin birbirinden farklı olduğu durumlarda ise çıkış olarak 1 değerini veren bir mantık kapısıdır ve $\oplus$ notasyonu ile gösterilmektedir.

Çizelge 2.3. XOR Kapısı Doğruluk Tablosu

a	**b**	**Y**
0	0	0
0	1	1
1	0	1
1	1	0

XOR kapısının Boole cebri eşitliği ise $Y = a'b + ab'$ şeklindedir.

Tanım 2.39 (XOR Sayısı)**.** $XOR(\alpha)$, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı $p(x)$ polinomunun bir elemanı olan α ile yine $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı $p(x)$ polinomundan keyfi olarak seçilen β elemanının çarpımını gerçekleştirmek için gerekli olan XOR kapılarının sayısı olarak tanımlanmaktadır (Khoo vd., 2014).

Tanım 2.40 (Matrisler Özelinde XOR Sayısı)**.** $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı $n \times n$ boyutundaki M matrisinin bir satırının XOR sayısı, ifade 2.18'de görüldüğü gibi

tanımlanmaktadır (Khoo vd., 2014).

$$M \text{ matrisinin bir satırının } XOR \text{ sayısı } = (\gamma_1, \gamma_2, \cdots, \gamma_k) + (n-1) \cdot m \qquad (2.18)$$

İfadede γ_i, M matrisinin hesaplama yapılacak satırındaki i. elemanın XOR sayısını, n aynı satırdaki sıfır olmayan elemanların sayısını, m ise sonlu cismin mertebesini temsil etmektedir. Söz konusu ifade tüm satırlar için uygulanarak, matrisin toplam XOR sayısı elde edilmektedir.

Tezde kullanılan matematiksel ifadelerin açıklanmasına ayrılan bu bölümün ardından, üçüncü bölüm, MDS ve MDBL matrislerin tasarımı için literatürde kullanılan tekniklerin yanında, XOR sayısı metriğinin detaylandırılmasını ve iyileştirilmesi için geliştirilen optimizasyon algoritmalarını içermektedir.

BÖLÜM 3

LİTERATÜRDE KULLANILAN TASARIM VE OPTİMİZASYON TEKNİKLERİ

Literatür taramasına ayrılan bu bölüm, iki alt kısımdan oluşmaktadır. Birinci kısımda, blok şifrelerde doğrusal dönüşüm katmanı olarak kullanılan MDS ve MDBL matrislerin elde edilmesinde yararlanılan tekniklerden bahsedilmektedir. İkinci kısım ise MDS ve MDBL matrislerinin donanıma uygulanması sırasında önemli bir maliyet kriteri olan XOR sayısının detaylandırılması ve örneklendirilmesinin yanı sıra bu maliyet ölçütünün iyileştirmesi için geliştirilen yeni yaklaşım, teknik ve algoritmaları kapsamaktadır.

3.1 MDS Matrislerin Tasarım Teknikleri

Blok şifrelerin tasarımında yaygın olarak kullanılan iki temel ilkeden biri yayılma (difüzyon) ilkesidir. Yayılma, açık metnin ve gizli anahtarın her bitinin şifreli metnin birden fazla bitini etkilemesini ve birkaç döngüden sonra tüm çıkış bitlerinin tüm giriş bitlerine bağlı olmasını sağlamaktadır. Başarılı bir difüzyon katmanı oluşturmak için, çoklu permütasyon tekniği (Schnorr & Vaudenay, 1994; Vaudenay, 1994) veya maksimum uzaklıkta ayrılabilen (MDS) matrisler kullanılmaktadır. Heys ve Tavares (1994, 1995, 1996) çalışmalarında, yer değiştirme permütasyon ağlarındaki (SPN'ler) permütasyon katmanının, iyi difüzyon özelliği gösteren bir doğrusal dönüşüm matrisiyle değiştirilmesinin blok şifrenin çığ özelliklerini geliştirerek, doğrusal ve diferansiyel kriptoanalize karşı direncini arttırdığını göstermişlerdir. 1996 yılında ise ilk kez, SHARK (Rijmen vd., 1996) blok şifresinin doğrusal dönüşüm katmanında bir MDS matris kullanılmıştır. Ardından geliştirilen SQUARE (Daemen vd., 1997), AES (Deamen & Rijmen, 2002), Twofish (Schneier vd., 1998) gibi blok şifreler, MDS matrislerin doğrusal

dönüşüm katmanında kullanımının maksimum yayılıma ulaşılmasını sağladığını ortaya çıkarmıştır. İlerleyen yıllarda Maelstorm (GD Filho, 2006), Whirlpool (P. Barreto, Rijmen vd., 2000; Shirai & Shibutani, 2003), SPN-Hash (Choy vd., 2012) gibi pek çok özet fonksiyonunda da difüzyon katmanı olarak MDS matrislerden yararlanılmıştır. 2011 yılına gelindiğinde, LED (J. Guo, Peyrin, Poschmann & Robshaw, 2011) blok şifresi ve PHOTON (J. Guo, Peyrin & Poschmann, 2011) özet fonksiyonunda kullanılan MDS matrislerin Companion matrislerden oluşturulması, hafif sıklet kriptografide de MDS matrislere kullanım alanı açmıştır. Günümüzde ise MDS matrisler, modern blok şifrelerin ve özet fonksiyonlarının en önemli bileşenlerinden biri haline gelmiştir.

MDS matrislerinin oluşturulmasında, özyinelemeli (Recursive) ve özyinelemeli olmayan (Nonrecursive) şeklinde ifade edilen iki temel yaklaşım bulunmaktadır. Özyinelemeli yapılarda genellikle, n mertebesinde bir M companion matris kullanılmaktadır. Çalışılan cisimdeki karakteristik polinomun uygun katsayılarının seçimiyle, bu companion matrisin üsleri alınarak M^n olacak şekilde bir MDS matrise ulaşılmaya çalışılır. Ayrıca, Companion matrislerden MDS matrisler elde edilirken kodlama teorisi alanındaki devirli kodlar (MacWilliams & Sloane, 1977), eşlik denetim kodları (Blaum & Roth, 1999), kısaltılmış BCH kodlar (Bose & Ray-Chaudhuri, 1960; Hocquenghem, 1959), Gabidulin kodlar (Berger, 2013; Berger & Ourivski, 2009) gibi bazı özel kodlardan yararlanılmaktadır. Yakın zamanda önerilen DSI (Diagonal-Serial-Invertible) matrisler (Toh vd., 2018) de, MDS matris tasarımında kullanılan özyinelemeli yapılara verilebilecek örneklerdendir. Özyinelemeli yapılar hafif sıklet uygulamalar için oldukça popülerdir. Özyinelemeli olmayan yapılarda MDS matrisleri oluşturmak içinse genellikle kendisi de MDS olan matrislerden yararlanılmaktadır.

MDS matrisleri tasarlamak için kullanılan teknikleri sınıflandırmanın başka bir yolu da, matrislerin doğrudan tasarlanmasına, bir arama uzayı oluşturacak arama yöntemlerinin kullanılmasına veya bu iki tekniği birleştiren melez yöntemlere dayanmaktadır. Doğrudan tasarım teknikleri, MDS matrisleri oluşturmak için cebirsel özellikleri kullanmaktadır. Arama yöntemlerinde ise, matrisin elemanları arama uzayını daraltacak kriterler göz önünde bulundurularak mantıklı bir şekilde seçilmekte ve matrisin MDS olup olmadığı kontrol edilmektedir. Bir matrisin MDS olup olmadığının doğrulanması NP-tam zorluğunda bir problemdir. Bu sebeple, bir matrisin MDS olup olmadığı sonucunu polinom zamanda veren bir algoritma henüz bulunamamıştır. Dolayısıyla, arama tekniği yalnızca küçük mertebeye sahip MDS matrislerin tasarlanmasında kullanışlı olmaktadır.

Özyinelemeli olmayan MDS matrisleri oluşturmak için, iki ana doğrudan tasarım

tekniği kullanılmaktadır. Bunlardan ilkinde Cauchy matrisinden, diğerinde ise iki adet Vandermonde matrisinden faydalanılmaktadır. Bu teknikler herhangi bir mertebedeki MDS matrisleri tasarlamakta başarılı olmakla beraber, elde edilen matrisler genellikle uygulama tarafında verimli değildir. Bu nedenle, verimli bir şekilde uygulanabilir MDS matrisleri üreten özyinelemeli olmayan yapılarda, arama yöntemleri tercih edilmektedir. Bu tür yapılar için bilinen teknikler; dairesel, dairesel benzeri (Gupta & Ray, 2015), genelleştirilmiş dairesel (Liu & Sim, 2016) ve Hadamard matrislerin elemanlarını aramaktır. Toeplitz ve Hankel matrisleri de dairesel matrislerle yakından ilişkilidir ve son zamanlarda MDS matrisleri tasarlamak için kullanılmaktadırlar (Sarkar & Syed, 2016, 2017). Dairesel matris ve türevlerinde olduğu gibi, Toeplitz ve Hankel matrislerden MDS matrisleri tasarlamak için de aramaya dayanan tekniklerden yararlanılmaktadır. Arama yöntemlerinde herhangi bir rastlantısal matris yerine bu özel matris formlarının kullanılmasının bazı avantajları bulunmaktadır. Bu avantajlar;

- Rastlantısal bir matrise kıyasla MDS matris bulma olasılığının daha yüksek olması,
- Elde edilen MDS matrislerin daha düşük uygulama maliyetine sahip olma ihtimalinin bulunması,

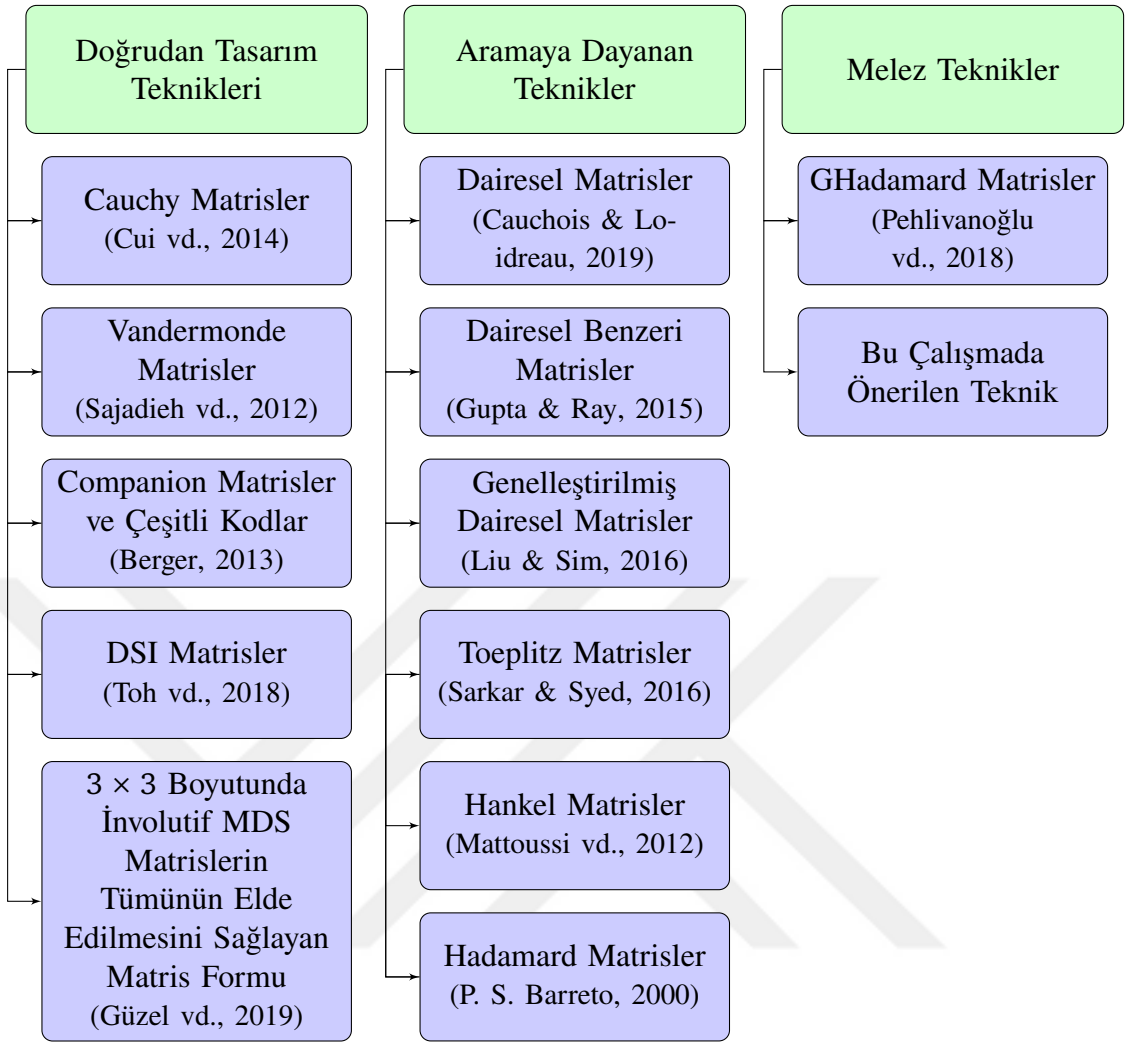
şeklinde verilebilir.

Özyinelemeli olmayan yapılara benzer şekilde, özyinelemeli yapılarda da birkaç doğrudan tasarım tekniği bulunmaktadır. Ancak bu teknikler, özyinemeli olmayan yapılarda olduğu gibi uygulama açısından o kadar verimli değillerdir. Dolayısıyla arama yöntemleri, düşük mertebeden verimli MDS matrislerin tasarlanmasında daha ön plana çıkmaktadırlar.

Tüm bunlara ek olarak blok şifrelerin uygulanmasında, şifreleme işlemi için kullanılan verimli bir MDS matris olan M, eğer involutif (tersi kendisine eşit) ise, şifre çözme işlemi için kullanılan M^{-1} matrisinin de verimli olacağı söylenebilmektedir. Bu nedenle, tasarlanan verimli MDS matrislerin aynı zamanda involutif olmasının istenilen bir özellik olduğunu ve tasarım tekniklerinde de bu kriterin göz önünde bulundurulduğunu belirtmek gerekmektedir. MDS matrislerin tasarımında sıklıkla kullanılan teknikler, Şekil 3.1'de görülmektedir.

3.1.1 Cauchy ve Vandermonde Matrislerden MDS Matris Tasarımı

Cauchy matrisler MDS matrislerin tasarımında yoğun olarak kullanılmaktadır (Cui vd., 2014; Gupta & Ray, 2015; MacWilliams & Sloane, 1977; Mattoussi vd., 2012; Roth & Lempel, 1989; Roth & Seroussi, 1985; Sim vd., 2015; Youssef vd., 1997).



Şekil 3.1. MDS Matrisler İçin Kullanılan Tasarım Tekniklerinin Sınıflandırılması

Benzer şekilde, Vandermonde matrislerden MDS matrislerin tasarımı da literatürde geniş kullanım alanına sahiptir (Gupta & Ray, 2013; Lacan & Fimes, 2004; Lacan & Fimes, 2003; Mattoussi vd., 2012; Sajadieh vd., 2012). Özyinelemeli olmayan doğrudan tasarım teknikleri sınıfına giren bu iki yöntem, Cauchy ve Vandermonde matrislerin bazı benzer cebirsel özelliklerinden faydalanmaktadır. Aralarındaki en dikkat çekici fark, Cauchy matrislerden tasarım yapılırken tek bir Cauchy matris, Vandermonde matrislerden tasarım yapılırken ise $V_1 = vand(x_0, x_1, \cdots, x_{n-1})$ ve $V_2 = vand(y_0, y_1, \cdots, y_{n-1})$ olmak üzere, $V_1^{-1}V_2$ ya da $V_2^{-1}V_1$ şeklinde tanımlanan iki adet Vandermonde matris kullanılmasıdır.

2.26'da verilen Vandermonde matris ile 2.27'de verilen Cauchy matris tanımları göz önüne alınarak oluşturulan V_1 Vandermonde matrisinin elemanları x_i, V_2 Vandermonde matrisinin elemanları y_i, C Cauchy matrisinin elemanları ise x_i ve y_i olsun. Bu elemanların çeşitli koşullar altında tanımlanmasına bağlı olarak dört farklı tip tasarım tekniğinden söz etmek mümkündür. Aşağıda bu dört tasarım tekniğinin hem Cauchy hem

de Vandermonde matrisler için uygulanması ve elde edilen MDS matrislerin özellikleri örneklerle anlatılmaktadır.

- **Tip-1:** $\{x_0, x_1, \cdots, x_{n-1}\}, \{y_0, y_1, \cdots, y_{n-1}\} \in \mathbb{F}_{2^m}$ ve $a_{ij} = \frac{1}{x_i + y_i}$ olmak üzere $C_{(a_{i,j})}$ Cauchy matrisi ile $V_1 = vand(x_0, x_1, \cdots, x_{n-1})$ ve $V_2 = vand(y_0, y_1, \cdots, y_{n-1})$ Vandermonde matrislerinin tüm x_i ve y_i girişlerinin birbirinden farklı olması bu tipin ana koşulunu oluşturmaktadır.

  - *Cauchy Matrisler* için $C_{(a_{i,j})}$ matrisi, yukarıdaki ana koşul dışında tüm $0 \leq i, j \leq n-1$ değerleri için $x_i + y_i \neq 0$ koşulunu da sağlıyorsa, tüm kare alt matrislerinin determinantı 0'dan farklı (singüler olmayan) bir matris, başka bir deyişle MDS olur. Elde edilen MDS matris seçilen girişlere bağlı olarak involutif veya non-involutif olabilir.
  - *Vandermonde Matrisler* için ilave bir koşul gerekmemektedir. Ana koşulu gerçekleyen $V_1^{-1}V_2$ ya da $V_2^{-1}V_1$ matrisleri, tüm kare alt matrislerinin determinantı 0'dan farklı olan bir MDS matristir. Sadece V_1 ve V_2 şeklinde iki matris kullanıldığından eğer bu matrisler n mertebesinde ise, $2n$ adet farklı giriş seçilmesi gerektiği unutulmamalıdır. Yine, elde edilen MDS matris seçilen girişlere bağlı olarak involutif veya non-involutif olabilir.

- **Tip-2:** Tip-1'de verilen koşula ek olarak, l, $\mathbb{F}_{2^m}$ cisminin 0'dan farklı ve keyfi olarak seçilen herhangi bir elemanı olmak üzere, y_i girişi $l + x_i$ şeklinde ifade edilebiliyorsa elde edilen $C_{(a_{i,j})}$ ve $V_1^{-1}V_2$ ya da $V_2^{-1}V_1$ matrisleri MDS olur.

  - *Cauchy Matrisler* için herhangi bir ilave koşul olmamasına rağmen bu tip ile elde edilen $C_{(a_{i,j})}$ matrisi involutif değildir. Eğer bir involutif MDS matris elde edilmek isteniyorsa, $D_1 = (x_{0,n-1}, x_{1,n-1}, x_{2,n-1}, \cdots, x_{n-1,n-1})$ ve $D_2 = (\prod_{k=0}^{n-1}(x_k + y_0), \prod_{k=0}^{n-1}(x_k + y_1), \prod_{k=0}^{n-1}(x_k + y_2), \cdots, \prod_{k=0}^{n-1}(x_k + y_{n-1}))$ olmak üzere iki adet singüler olmayan köşegen matris tanımlanır. Sonuç olarak elde edilen $D_1 C_{(a_{i,j})} D_2$ matrisi involutif MDS olur (Gupta, Pandey, Ray vd., 2019).
  - *Vandermonde Matrisler* için yine ilave bir koşul gerekmemektedir. Elde edilen $V_1^{-1}V_2$ ya da $V_2^{-1}V_1$ matrisleri ise involutif ve MDS'dir.

- **Tip-3:** Tip-1 ve Tip-2'de verilen koşullara ek olarak, x_i girişleri, $\mathbb{F}_{2^m}$ cisminin n mertebesinde toplamsal bir alt grubu olan $G = \{x_0, x_1, x_2, \cdots, x_{n-1}\}$ kümesinin elemanlarından seçilmek üzere, bir M matrisi en fazla n tane farklı eleman içeriyorsa ve $l \notin G$ ise, M matrisi bir MDS matristir (Gupta & Ray, 2013).

  - *Cauchy Matrisler* kullanıldığında bu tip ile elde edilen $C_{(a_{i,j})}$ matrisi yine involutif değildir. Bir involutif MDS matris elde etmek için r, $C_{(a_{i,j})}$ matrisinin

herhangi bir satırının elemanları toplamını temsil etmek üzere, $\frac{1}{r}C_{(a_{i,j})}$ işlemi gerçekleştirilmelidir.

  - *Vandermonde Matrisler* için yine ek bir koşul yoktur. Elde edilen $V_1^{-1}V_2$ ya da $V_2^{-1}V_1$ matrisleri ise involutif ve MDS'dir.

- **Tip-4:** Tip-1, Tip-2 ve Tip-3'de verilen koşullara ek olarak, x_i girişleri $x_i + x_j = x_{i \oplus j}$ olacak şekilde ve $\mathbb{F}_{2^m}$ cisminin n mertebesinde toplamsal bir alt grubu olan $G = \{x_0, x_1, x_2, \cdots, x_{n-1}\}$ kümesinin elemanlarından seçilmek üzere, bir M matrisi en fazla n tane farklı eleman içeriyorsa ve $l \notin G$ ise, M matrisi bir Hadamard MDS matristir (Gupta & Ray, 2013).

  - *Cauchy Matrisler* kullanıldığında bu tip ile elde edilen $C_{(a_{i,j})}$ matrisi yine involutif değildir. Bir involutif MDS matris elde etmek için Tip-3'e benzer şekilde r, $C_{(a_{i,j})}$ matrisinin herhangi bir satırının elemanları toplamını temsil etmek üzere, $\frac{1}{r}C_{(a_{i,j})}$ işlemi gerçekleştirilmelidir.
  - *Vandermonde Matrisler* kullanıldığında ek koşul olmaksızın elde edilen $V_1^{-1}V_2$ ya da $V_2^{-1}V_1$ matrisleri ise involutif ve Hadamard MDS olmaktadır.

Örnek 3.1. *α, $\mathbb{F}_{2^4}/0x13$ polinomunun bir kökü olmak üzere, x_i, y_i elemanları $x_0 = 0, x_1 = \alpha^3, x_2 = \alpha^5$ ve $y_0 = 1, y_1 = \alpha^4, y_2 = \alpha^8$ şeklinde seçildiğinde bir Cauchy matris olan M_{C1} aşağıdaki gibi oluşturulabilir:*

$$M_{C1} = \begin{bmatrix} 1 & \frac{1}{\alpha^4} & \frac{1}{\alpha^8} \\ \frac{1}{(1+\alpha^3)} & \frac{1}{(\alpha^3+\alpha^4)} & \frac{1}{(\alpha^3+\alpha^8)} \\ \frac{1}{(1+\alpha^5)} & \frac{1}{(\alpha^4+\alpha^5)} & \frac{1}{(\alpha^5+\alpha^8)} \end{bmatrix} = \begin{bmatrix} 1 & \alpha^3+\alpha^2+\alpha & \alpha^3+\alpha+1 \\ \alpha & \alpha^2+1 & \alpha^2 \\ \alpha^2+\alpha & \alpha^3+\alpha+1 & \alpha^3+\alpha^2+\alpha \end{bmatrix}.$$

Tip-1 tasarım tekniği kullanılarak elde edilen M_{C1} matrisi MDS ve non-involutiftir.

Örnek 3.2. *α, $\mathbb{F}_{2^4}/0x13$ polinomunun bir kökü olmak üzere, x_i elemanları $x_0 = \alpha^2, x_1 = \alpha^4, x_2 = \alpha^5$ ve $l = 1$'e karşılık $y_i = l + x_i$ elemanları $y_0 = 1+\alpha^2, y_1 = 1+\alpha^4, y_2 = 1+\alpha^5$ şeklinde seçildiğinde bir Cauchy matris olan M_{C2} aşağıdaki gibi oluşturulabilir:*

$$M_{C2} = \begin{bmatrix} 1 & \frac{1}{(1+\alpha^2+\alpha^4)} & \frac{1}{(1+\alpha^2+\alpha^5)} \\ \frac{1}{(1+\alpha^2+\alpha^4)} & 1 & \frac{1}{(1+\alpha^4+\alpha^5)} \\ \frac{1}{(1+\alpha^2+\alpha^5)} & \frac{1}{(1+\alpha^4+\alpha^5)} & 1 \end{bmatrix} = \begin{bmatrix} 1 & \alpha^2+\alpha+1 & \alpha^3+\alpha^2+\alpha \\ \alpha^2+\alpha+1 & 1 & \alpha^3+\alpha^2+1 \\ \alpha^3+\alpha^2+\alpha & \alpha^3+\alpha^2+1 & 1 \end{bmatrix}.$$

Tip-2 tasarım tekniği kullanılarak elde edilen M_{C2} matrisi yine MDS ve non-involutiftir.

Örnek 3.3. *α, $\mathbb{F}_{2^4}/0x13$ polinomunun bir kökü olmak üzere, $y_0 = \alpha^2$ ve $G = \{0, \alpha, \alpha^3, \alpha + \alpha^3\}$ olarak seçildiğinde $x_0 + y_0 = \alpha^2$ olur. Birer Vandermonde matris olan V_1 ile V_2 matrisleri, $V_1 = vand(0, \alpha, \alpha^3, \alpha + \alpha^3)$ ve $V_2 = vand(\alpha^2, \alpha + \alpha^2, \alpha^3 + \alpha^2, \alpha + \alpha^3 + \alpha^2)$ şeklinde tanımlandığında $M_{Vand} = V_1^{-1}V_2$ matrisi aşağıdaki gibi oluşturulabilir:*

$$M_{Vand} = V_1^{-1}V_2 = \begin{bmatrix} \alpha^3 + \alpha & \alpha^3 + \alpha^2 & \alpha^2 + \alpha & 1 \\ \alpha^3 + \alpha^2 & \alpha^3 + \alpha & 1 & \alpha^2 + \alpha \\ \alpha^2 + \alpha & 1 & \alpha^3 + \alpha & \alpha^3 + \alpha^2 \\ 1 & \alpha^2 + \alpha & \alpha^3 + \alpha^2 & \alpha^3 + \alpha \end{bmatrix}.$$

Tip-4 tasarım tekniği kullanılarak elde edilen M_{Vand} matrisi bir Hadamard MDS matris olup aynı zamanda involutiftir(Gupta, Pandey, Ray vd., 2019).

3.1.2 3×3 Boyutunda İnvolutif MDS Matrislerin Tümünün Elde Edilmesini Sağlayan Matris Formu

Son dönem doğrusal tasarım tekniklerinden biri de Güzel vd. (2019) çalışmasında verilmektedir. Söz konusu çalışmada yazarlar, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı 3×3 boyutlarında tüm involutif ve MDS matrisleri elde etmek için yeni bir matris formu önermektedirler. Önerilen matris formu, $IM_{3\times3}(a_{11}, a_{22}, b_0, b_1)$ notasyonu ile gösterilir ve $a_{11} \neq a_{22}, a_{11}, a_{22} \neq 0, a_{11}, a_{22} \neq 1, a_{11} + a_{22} \neq 1$ ve $b_0, b_1 \in \mathbb{F}_{2^m} - \{0\}$ olmak üzere aşağıdaki şekilde tanımlanabilir:

$$IM_{3\times3} = \begin{bmatrix} a_{11} & (a_{11}+1)b_0 & (a_{11}+1)b_1 \\ (a_{22}+1)b_0^{-1} & a_{22} & (a_{22}+1)b_0^{-1}b_1 \\ (a_{11}+a_{22})b_1^{-1} & (a_{11}+a_{22})b_1^{-1}b_0 & a_{11}+a_{22}+1 \end{bmatrix}.$$

Böylece, a_{11}, a_{22}, b_0 ve b_1 şeklinde verilen 4 adet parametre yardımıyla 3×3 boyutlarında tüm involutif ve MDS matrisler elde edilmektedir.

a_{11}, a_{22}, b_0 ve b_1 parametreleri için yukarıda verilen kısıtlamalar dikkate alındığında, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı 3×3 boyutunda tüm involutif ve MDS matrislerin sayısı ise $m > 2$ olmak üzere $(2^m - 1)^2 \cdot (2^m - 2) \cdot (2^m - 4)$ eşitliği ile hesaplanmaktadır.

3×3 boyutunda involutif ve MDS matrislerin tümünü oluşturmak için kullanılan ve b_0, b_1 parametreleri ile bunların terslerini içermeyen $IMR_{3\times3}(a_{11}, a_{22})$ matrisi, temsili matris formu olarak adlandırılmakta ve aşağıdaki gibi gösterilmektedir:

$$IMR_{3\times3}(a_{11}, a_{22}) = \begin{bmatrix} a_{11} & a_{11}+1 & a_{11}+1 \\ a_{22}+1 & a_{22} & a_{22}+1 \\ a_{11}+a_{22} & a_{11}+a_{22} & a_{11}+a_{22}+1 \end{bmatrix}.$$

Bir diğer temsilci matris formu ise $IMR_{3\times3}(a_{11}, a_{22})$ matrisinin transpozu olarak aşağıdaki gibi tanımlanabilir:

$$IMR^{T}_{3\times3}(a_{11}, a_{22}) = \begin{bmatrix} a_{11} & a_{22}+1 & a_{11}+a_{22} \\ a_{11}+1 & a_{22} & a_{11}+a_{22} \\ a_{11}+1 & a_{22}+1 & a_{11}+a_{22}+1 \end{bmatrix}.$$

Son adımda bu iki temsilci matris formuna b_0, b_1 parametreleri ile bunların tersleri uygulanarak, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı 3×3 boyutunda involutif ve MDS matrislerin tümü elde edilmektedir.

3.1.3 Companion Matrisler ve Çeşitli Kodlar Kullanılarak MDS Matris Tasarımı

Özyinelemeli yaklaşımlar kullanarak MDS matrislerin tasarımına olanak veren Companion matrisler Tanım 2.30'da verilmekte ve $companion(-a_0, -a_1, \cdots, -a_{k-1})$ notasyonu ile gösterilmektedir. Söz konusu tanım dikkate alındığında, eğer $a_0 \neq 0$ ise C_g matrisi singüler olmayan matris özelliği taşımakta, $a_0 = 1$ ise de C_g matrisinin son satırının elemanları ile C_g^{-1} matrisinin ilk satırının elemanları aynı olmaktadır. Companion matrisler aynı zamanda seyrek matris oldukları için hafif sıklet kriptografi alanında sıklıkla kullanılmaktadır. Seyrek matris elemanların çoğunun sıfır olduğu bir matristir (Gloub & Van Loan, 1996). Bir matrisin seyrek olarak nitelendirilmesi için sıfır değerine sahip elemanların oranıyla ilgili kesin bir tanım yoktur, ancak ortak bir kriter, sıfır olmayan elemanlarının sayısının kabaca matrisin satır veya sütun sayısına eşit olmasıdır.

Özyinelemeli yaklaşımın temelinde, MDS olmayan bir companion matrisin tekrarlı bir şekilde üslerinin alınmasıyla bir MDS matrise ulaşılması düşüncesi bulunmaktadır. PHOTON (J. Guo, Peyrin & Poschmann, 2011) özet fonksiyonu ve LED (J. Guo, Peyrin, Poschmann & Robshaw, 2011) blok şifresinde bu yaklaşıma dayanan ve kapsamlı arama yöntemleriyle elde edilen companion matrisler kullanılmaktadır. Aramaya dayanan teknikler genellikle matris boyutu ve cisim alanının küçük olduğu durumlarda uygulanmaktadır. 8×8 boyutundan büyük matrislerde ve $\mathbb{F}_{2^8}$ cisminden geniş arama uzaylarında ise verimli olmamaktadır. Bu sebeple, daha büyük boyutta özyinelemeli MDS matrisleri elde etmek için kodlama teorisi alanından yararlanılarak oluşturulan doğrudan tasarım teknikleri kullanılmaktadır. Berger (2013)'de, özyinelemeli MDS matrisleri elde etmek için Gabidulin (Berger & Ourivski, 2009) kodlarının nasıl kullanılacağını göstermektedir. Augot ve Finiasz (2014)'de ise kısaltılmış BCH kodları kullanılarak özyinelemeli MDS matrisler elde edilmektedir. Bu iki yöntem, herhangi bir mertebede özyinelemeli bir MDS matrisi oluşturmak için yeterli olsa da, Gupta vd. (2017)'de Companion matrisin tanımlı olduğu karakteristik polinomun köklerini kullanarak daha

genelleştirilmiş bir tasarım ortaya koymaktadır.

Bu tasarımda elde edilecek özyinelemeli MDS matrisi, bir Companion matrisin üssü olarak ifade edilebilen bir MDS matris olarak tanımlanmaktadır. Diğer bir deyişle k dereceli monik polinom $g(x) \in \mathbb{F}_q[x]$ olmak üzere $m \geq k$ ise bir MDS matrisi olan $M = C_g^m$ şeklinde ifade edilmektedir. Eğer $g(x) \in \mathbb{F}_q[x]$ polinomu için böyle bir m tam sayısı varsa $g(x)$'in özyinelemeli bir MDS matris oluşturabildiği söylenmektedir. Dikkat edilmesi gereken nokta, MDS matrisin boyutunun 1'den büyük olması gerektiğidir ve bu nedenle her zaman özyinelemeli MDS matrisleri elde etmek için düşünülen $g(x) \in \mathbb{F}_q[x]$ polinomlarının derecesi $k = deg(g) \geq 2$ olarak seçilmektedir. Örneğin k adet satırdan oluşan Companion matris C_g aşağıdaki gibi tanımlanmaktadır:

$$C_g = \begin{bmatrix} & x & \\ & x^2 & \\ & \vdots & \\ & x^{k-1} & \\ x^k & \bmod g(x) & \end{bmatrix}.$$

C_g matrisinin m derecesine kadar üsleri alındığında,

$$C_g^2 = \begin{bmatrix} & x^2 & \\ & x^3 & \\ & \vdots & \\ x^k & \bmod g(x) & \\ x^{k+1} & \bmod g(x) & \end{bmatrix}, \cdots, C_g^m = \begin{bmatrix} x^m & \bmod g(x) & \\ x^{m+1} & \bmod g(x) & \\ & \vdots & \\ x^{m+k-2} & \bmod g(x) & \\ x^{m+k-1} & \bmod g(x) & \end{bmatrix}$$

elde edilen C_g^m matrisi, ancak ve ancak $G = [C_g^m | I]$ ya da $G' = [-C_g^m | I]$ üreteç matrislerinin herhangi bir k sütununun $\mathbb{F}_q$ cismi üzerinde doğrusal olarak bağımsız olması durumunda MDS'dir. Benzer bir durum kodlama teorisindeki özel kodlar kullanılarak MDS matrisler tasarlanırken de geçerli olmaktadır. G ve G' üreteç matrislerinin herhangi bir k sütununun kullanılan özel kod ile doğrusal olarak bağımsız olması durumunda elde edilen matris yine MDS olmaktadır. Ancak, doğrudan tasarım teknikleri yardımıyla $\mathbb{F}_2$ cismi üzerinde tanımlı $M = \begin{bmatrix} 1 \end{bmatrix}$ matrisi dışında özyinelemeli involutif MDS matris elde edilememektedir (Gupta, Pandey & Venkateswarlu, 2019).

MDS matrisleri özyinelemeli olarak elde etmek için DSI (Diagonal-Serial-Invertible) matrisler olarak bilinen yeni bir tür matris tipi de literatürde son zamanlarda öne çıkmaktadır. Tanım 2.31'de verilen DSI matrisler de, tıpkı Companion matrisler gibi seyrek tipte matrislerdir ve MDS olmayan bir matrisin tekrarlı bir şekilde üslerinin alınmasıyla bir MDS matrise ulaşılması yaklaşımını kullanmaktadır.

Örnek 3.4. $\mathbb{F}_2[x]/0x13$ *polinomunun bir kökü* α *olmak üzere* $a = [1, 1, 1, \alpha]$ *ile* $b = [1, 0, \alpha^3 + 1]$ *vektörleri kullanılarak oluşturulan DSI matrisi D, aşağıdaki gibi gösterilebilir:*

$$D = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & \alpha^3 + 1 & 0 \\ 0 & 0 & \alpha & 0 \end{bmatrix}.$$

D matrisinin 4 *kez kuvveti alındığında oluşan* D^4 *matrisi bir MDS matristir.*

$$D^4 = \begin{bmatrix} \alpha + 1 & \alpha + 1 & \alpha^3 + \alpha & 1 \\ 1 & \alpha & \alpha + 1 & 1 \\ \alpha^2 + 1 & \alpha^3 + \alpha^2 + \alpha + 1 & \alpha^3 + \alpha^2 & \alpha^3 \\ \alpha + 1 & \alpha^3 + 1 & \alpha^3 + \alpha^2 + 1 & \alpha \end{bmatrix}$$

Yukarıda bahsedilen DSI matrisi D'nin, 4×4 boyutları için şimdiye kadar elde edilen en iyi Companion matrisinden daha düşük uygulama maliyetine sahip olduğu belirtilmektedir (Toh vd., 2018). Hafif sıklet özyinelemeli MDS matrisleri oluşturmak için ciddi bir potansiyel aday olmasına rağmen, çok yakın zamanda önerilmesi nedeniyle teorik altyapı anlamında Companion matrislerin gerisindedir. Çünkü, herhangi bir mertebe için Companion matrislerden özyinelemeli MDS matrislerin elde edilmesinde birçok doğrudan tasarım tekniği bulunmasına karşılık, DSI matrisler söz konusu olduğunda, henüz en iyi bilinen yöntem kapsamlı arama teknikleridir.

Companion ve DSI matrisler gibi seyrek matrisler, alandan tasarruf etmek için iyi bir yol sağlasa da matrisi uygulamak için artan sayıda üs alma işleminin gerekmesi maliyeti arttırmaktadır. Genel olarak, döngü tabanlı ürün şifreleri veya düşük gecikmeli uygulamalar için pek uygun olmadığı belirtilmektedir (Sim vd., 2015).

3.1.4 Dairesel Matris ve Türevlerinden MDS Matris Tasarımı

Modern kriptografi çalışmalarında, doğrudan tasarım teknikleri kullanılarak dairesel MDS matrisler elde etmek için bir yöntem bulunmamaktadır. Dolayısıyla dairesel matris ve türevlerinden MDS matrisler tasarlamak için aramaya dayanan yöntemlerden faydalanılmaktadır. Dairesel matris ve türevleri genellikle tekrarlanan girişlere sahip olması sebebiyle hafif sıklet kriptografi alanında da tercih edilmektedir. Örneğin literatürde çok iyi bilinen AES (Deamen & Rijmen, 2002) blok şifresinin difüzyon matrisi, $\mathbb{F}_{2^8}/0x11b$ polinomunun bir kökü α olmak üzere tanımlanan 4×4 boyutunda dairesel

bir MDS matristir. Arama yöntemi ile elde edilen bu matris ifade 3.1'de görülmektedir.

$$M_{AES} = circ(\alpha, \alpha+1, 1, 1) = \begin{bmatrix} \alpha & \alpha+1 & 1 & 1 \\ 1 & \alpha & \alpha+1 & 1 \\ 1 & 1 & \alpha & \alpha+1 \\ \alpha+1 & 1 & 1 & \alpha \end{bmatrix} \tag{3.1}$$

İlk satırında iki adet 1 bulunan bu matris, 1 ile çarpma işleminin herhangi bir maliyet gerektirmemesi nedeniyle uygulama avantajı sağlamaktadır. Aramaya dayanan teknikler çok büyük olmayan arama uzaylarında küçük mertebeye sahip verimli MDS matrislerin elde edilmesinde tercih edilse de, daha yüksek mertebeli matrisler için büyük arama uzaylarında başarısız olmaktadır. Dolayısıyla bu teknikler kullanılırken arama uzayını daraltabilecek kriterler saptamak büyük önem taşımaktadır. Dairesel matrisler ve türevleri ile ilgili öne çıkan çalışmalar Cauchois ve Loidreau (2019), Gupta ve Ray (2014, 2015), Junod ve Macchetti (2009), Liu ve Sim (2016) olarak örneklendirilebilir.

Her satır vektörünün bir önceki satır vektörüne göre bir eleman sağa ötelenmesiyle elde edilen dairesel matrislerin tanımına 2.23'den ulaşılabilir ve bir $circ(x_0, x_1, \cdots, x_{n-1})$ dairesel matrisinin (i, j) indisli girişleri $(C)_{i,j} = x_{(j-i) \bmod n}$ ifadesi ile kontrol edilebilir. $n \times n$ boyutlarında bir dairesel matrisin en fazla n farklı girişi bulunmaktadır ve 1 elemanı gibi maliyetsiz ya da düşük maliyetli elemanların tekrarını içerebilir. Bu durum uygulamada, Hadamard ve Cauchy matrisler gibi MDS olabilmesi için en az n farklı girişe ihtiyaç duyan matris tiplerine karşı dairesel matrisleri daha avantajlı konuma getirmektedir. Ayrıca, aramaya dayanan tekniklerde rastgele herhangi bir matris yerine dairesel matrisler kullanmanın, MDS matris elde etme olasılığını arttırdığı bilinmektedir (Daemen vd., 1997). Öte yandan bu tekniğin en büyük dezavantajı, SPN ağ mimarisini kullanan blok şifrelerde difüzyon matrisinin involutif olmasının istenilen bir özellik olduğu dikkate alındığında, dairesel matrislerin aynı anda hem MDS hem de involutif özellik gösterememesi olarak kabul edilmektedir. Bu dezavantajı ortadan kaldırmak için dairesel matrislerin çeşitli türevlerinden yararlanılmaktadır.

Dairesel matris ve türevlerinden MDS matris tasarımı yapılırken temel yaklaşım, düşük maliyete sahip elemanların maksimum sayıda tekrarlanmasını sağlayacak kriterler veya dairesel matris formları belirleyerek arama uzayını daraltmaktır. İfade 3.1'de verilen dairesel MDS $circ(\alpha, \alpha+1, 1, 1)$ matrisi, düşük hamming ağırlığına sahip öğeler içermektedir, ancak bu matristeki 1'lerin sayısı 8'dir. Junod vd. (2004), 4×4 boyutundaki bir MDS matrisin maksimum 1 sayısının 9 olduğunu göstererek alt matrisleri dairesel matrisler olan yeni ve verimli bir MDS matris sınıfı oluşturmuşlardır. Ardından Gupta vd. (2015), bu yeni tip matrisleri *Tip-I dairesel benzeri* matrisler olarak isimlendirmişler

ve dahası Tip-I dairesel benzeri matrislerin tersinin de hemen hemen aynı forma sahip olduğunu göstererek bu matrisleri de *Yakın Tip-I dairesel benzeri* matrisler olarak sınıflandırmışlardır.

Tanım 3.1 (Tip-I Dairesel Benzeri Matrisler)**.** x_i girişleri ve a, tanımlı cismin 1 dışında sıfır olmayan herhangi bir elemanı olmak üzere $n \times n$ boyutlarında Tip-I dairesel benzeri matris, ifade 3.2'deki gibi tanımlanmaktadır.

$$\begin{bmatrix} a & 1 \\ 1^T & A \end{bmatrix} \tag{3.2}$$

İfadede A, $n-1$ adet 1 elemanı içeren $A = circ(1, x_1, \cdots, x_{n-2})$ matrisini temsil etmekte ve Tip-I dairesel benzeri matris, $TypeI(a, circ(1, x_1, \cdots, x_{n-2}))$ notasyonu ile gösterilmektedir.

Tanım 3.2 (Yakın Tip-I Dairesel Benzeri Matrisler)**.** a, b ve x_i girişleri, tanımlı cismin 1 dışında sıfır olmayan herhangi bir elemanı olmak üzere $n \times n$ boyutlarında Yakın Tip-I dairesel benzeri matris, $n-1$ adet b elemanı içermek koşulu ile ifade 3.3'deki gibi tanımlanmaktadır.

$$\begin{bmatrix} a & b \\ b^T & A \end{bmatrix} \tag{3.3}$$

İfadede $A = circ(x_0, x_1, \cdots, x_{n-2})$ matrisini temsil etmekte ve Yakın Tip-I dairesel benzeri matris, $AlmostTypeI(a, b, circ(x_0, \cdots, x_{n-2}))$ notasyonu ile gösterilmektedir.

Tanım 3.1 ve 3.2'de verilen dairesel matris türevleri, düşük maliyete sahip elemanların maksimum sayıda tekrarlanmasını sağlayacak şekilde MDS matris tasarımına olanak verse de bu yöntemlerle elde edilen matrisler involutif değildir. Buna yönelik olarak, Gupta ve Ray (2015) ile Liu ve Sim (2016) çalışmalarında hem MDS hem de involutif olan matrislerin tasarlanmasını sağlayan iki yeni tip dairesel matris türevinden bahsetmektedir. Sırasıyla *Tip-II dairesel benzeri* ve *Sol dairesel* adlarıyla bilinen bu matrisler Tanım 3.3 ve 3.4'de verilmektedir.

Tanım 3.3 (Tip-II Dairesel Benzeri Matrisler)**.** $n \times n$ boyutlarında bir A matrisi, $A = circ(x_0, x_1, \cdots, x_{n-1})$ olmak üzere $2n \times 2n$ boyutlarında Tip-II dairesel benzeri matris, ifade 3.4'deki gibi tanımlanmakta ve $TypeII(circ(x_0, \cdots, x_{n-1}))$ notasyonu ile gösterilmektedir.

$$\begin{bmatrix} A & A^{-1} \\ A^3 + A & A \end{bmatrix} \tag{3.4}$$

Tanım 3.4 (Sol Dairesel Matrisler)**.** n mertebesinde bir sol dairesel matris olan L, her satır vektörünün bir önceki satır vektörüne göre bir eleman sola ötelenmesiyle elde edilmekte ve genelleştirilmiş dairesel matris adıyla da bilinmektedir. x_i matrisin ilk satırının girişlerini temsil etmek üzere $L = l - circ(x_0, x_1, \cdots, x_{n-1})$ notasyonu ile gösterilmektedir.

Kriptografik çalışmalarda difüzyon katmanı olarak $2^n \times 2^n$ boyutlarındaki matrisler sıklıkla tercih edilmektedir (Nakahara Jr & Abrahao, 2009). Tanım 3.3 ve 3.4'de verilen dairesel matris türevleri ile $2^n \times 2^n$ boyutlarında olmasa da aşağıda örnekleri verilen bazı boyutlar için hem MDS hem de involutif matrislerin elde edilmesi mümkün olmaktadır.

Örnek 3.5. *Tip-II dairesel benzeri matrisler kulanılarak,*

- $\mathbb{F}_{2^8}/0x11b$ *polinomunun bir kökü* α *olmak üzere* 6×6 *boyutunda* $TypeII(circ(\alpha, 1, 1+\alpha^2))$ *matrisi (Gupta & Ray, 2015),*

Sol dairesel matrisler kullanılarak ise,

- α, $\mathbb{F}_{2^8}/0x165$ *polinomunun bir kökü olmak üzere* 5×5 *boyutunda* $l-circ(1, \alpha, \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1, \alpha^7 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1, \alpha^3 + \alpha)$ *matrisi,*
- α, $\mathbb{F}_{2^8}/0x165$ *polinomunun bir kökü olmak üzere* 6×6 *boyutunda* $l-circ(1, 1, \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1, \alpha^5 + \alpha^3 + \alpha^2, \alpha^2, \alpha^7 + \alpha^4 + \alpha^3 + \alpha)$ *matrisi (Liu & Sim, 2016),*

involutif MDS olarak elde edilmektedir.

3.1.5 Toeplitz ve Hankel Matrislerden MDS Matris Tasarımı

Bir Toeplitz matris veya sabit köşegenli matris, esas köşegeninin tüm elemanları aynı olan özel bir matris türüdür ve ilk satırı ile ilk sütununun elemanları kullanılarak gösterilmektedir. Örneğin Tanım 2.28'de verilen Toeplitz matrisininin notasyonu $T = toep = (x_0, x_{-1}, \cdots, x_{-(n-1)}; x_1, x_2, \cdots, x_{n-1})$ şeklindedir. T matrisinin herhangi bir (i, j) elemanına ise $(T)_{i,j} = x_{j-i}$ ifadesi ile ulaşılabilir. Toeplitz matrisler, dairesel matrislerle yakından bağlantılıdır (Gray, 2005; Kra & Simanca, 2012). Dolayısıyla, Toeplitz matrislerden MDS matrisler elde edilirken tıpkı dairesel matris ve türevlerinde olduğu gibi aramaya dayanan tekniklerden faydalanılmaktadır (Sarkar & Syed, 2017). Bu teknikler dairesel, Tip-I dairesel benzeri ve Yakın Tip-I dairesel benzeri matrislerden MDS matrisler tasarlanırken kullanılan tekniklere oldukça yakın benzerlikler içermektedir. Bu benzerlikler sebebiyle de Toeplitz matrisler kullanılarak tasarlanan difüzyon katmanlarının aynı anda hem MDS hem de involutif olması mümkün değildir (Sarkar & Syed, 2016).

Hankel matrisler ise sol dairesel matrislerin özel bir formudur. Aynı zamanda simetrik matris özelliği gösteren Hankel matrisler, ilk satırı ile son sütununun elemanları kullanılarak gösterilmektedir. Tanım 2.29'de verilen Hankel matrisinin notasyonu $H = hank = (x_0, x_{-1}, \cdots, x_{(n-1)}; x_n, x_{n+1}, \cdots, x_{2n-2})$ şeklindedir. H matrisinin herhangi bir (i, j) elemanına ise $(H)_{i,j} = x_{i+j}$ ifadesi ile ulaşılabilir. Sol dairesel matrislere olan benzerliğinden dolayı $2^n \times 2^n$ boyutlarında hem MDS hem de involutif olan örneği bulunmamaktadır. $2n \times 2n$ ve $(2n+1) \times (2n+1)$ boyutlarında ise aramaya dayanan teknikler yardımıyla involutif MDS matrisler tasarlanabilmektedir. Örneğin; $\mathbb{F}_{2^8}/0x163$ polinomunun bir kökü α olmak üzere, 5×5 boyutunda $hank = (1, \alpha, \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1, \alpha^7 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1, \alpha^3 + \alpha; 1, \alpha, \alpha^7 + \alpha^5 + \alpha^4 + \alpha + 1, \alpha^7 + \alpha^5 + \alpha^4 + \alpha^3 + \alpha + 1)$ matrisi involutif ve MDS'dir (Gupta, Pandey, Ray vd., 2019).

3.1.6 Hadamard ve GHadamard Matrislerden MDS Matris Tasarımı

Blok şifrelerde, şifreleme ve şifre çözme adımları için tamamen aynı difüzyon matrisinin kullanılabilmesi değerli bir avantaj olarak kabul edilmektedir. Bu avantajı sağlamak için de difüzyon matrisinin involutif özelliğe sahip olması gerekmektedir. Şu ana kadar incelenen tasarım teknikleri ile hem MDS hem de involutif olan hafif sıklet matrisleri oluşturmanın halen önemli bir problem olduğu görülmektedir. Vandermonde veya Cauchy matrisleri kullanan tasarımlar büyük matris boyutları için involutif MDS matrisleri oluşturmaya izin verse de, arama uzayı büyüdüğünde uygun maliyetli involutif difüzyon matrisi adaylarını bulmak zor olmaktadır. Hadamard matrisler ise taşıdığı özellikler nedeniyle, kapsamlı arama yöntemleri kullanılarak 4×4 ve 8×8 gibi küçük boyutlarda involutif ve MDS difüzyon matrislerinin tasarlanmasında oldukça başarılı sonuçlar vermektedir (P. S. Barreto, 2000; P. Barreto & Rijmen, 2000; Sim vd., 2015). Elemanları $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı $n \times n$ boyutlarında bir Hadamard matris 2.24'deki gibi tanımlanmakta ve $H = had = (a_0, a_1, \cdots, a_{n-1})$ notasyonu ile gösterilmektedir. H matrisinin bazı önemli özellikleri ise aşağıdaki gibi verilmektedir.

i) a_i, H Hadamard matrisinin ilk satırının elemanları olmak üzere $H_{i,j} = a_{i \oplus j}$'dir.

ii) $J_{i,n-,+1} = 1$ ve diğer elemanları 0 olan $n \times n$ boyutunda bir değişim matrisi ise $HJ = JH$ ve $H = H^T$ olur. Diğer bir deyişle H bisimetrik bir matristir.

iii) $c = \bigoplus_{i=0}^{n-1} a_i$ ve I $n \times n$ boyutunda bir birim matris olmak üzere $H^2 = c^2 I$'dır.

Üçüncü özellik açıkça H matrisinin kendisiyle çarpımının, matrisin ilk satır elemanlarının kareleri toplamının birim matrisle çarpılmasına eşit olduğunu söylemektedir. O halde, H matrisinin ilk satır elemanları toplamının 1 olması, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı bu Hadamard matrisin involutif olması sonucunu ortaya çıkarmaktadır.

Dolayısıyla, $\mathbb{F}_{2^m}$ cismi üzerindeki bir Hadamard matrisin ilk satır elemanları toplamı sıfırdan farklıysa, söz konusu matris ilk satırın toplamına bölünerek kolaylıkla involutif hale getirilebilmektedir. Bu fikirden yola çıkıldığında da tüm olası Hadamard matrisler üzerinde kapsamlı arama yöntemleri kullanılarak küçük boyutlu (4×4 ve 8×8 gibi) involutif MDS difüzyon matrisleri elde edilmektedir (Sim vd., 2015). Ancak boyut arttıkça, MDS özelliğini ve kapsamlı arama uzayını doğrulamaya yönelik hesaplama katlanarak büyümekte ve daha yüksek boyutlu Hadamard matrisleri aramak pratik olmamaktadır. Bu probleme bir çözüm olarak, Pehlivanoğlu vd. (2018) çalışmasında hem doğrudan tasarım hem de kapsamlı arama tekniklerinden faydalanılan bir yöntem önerilmektedir. Söz konusu melez yöntem, Hadamard matrislerin genişletilmiş ve genelleştirilmiş bir formu olan Genel Hadamard (GHadamard) matrisleri kullanmaktadır.

2×2 boyutlarında bir Genel Hadamard matris olan $GH = Ghad = (a_0, a_1; b_1)$ matrisi, 2.24'de verilen $\mathbb{F}_{2^m}$ cisminde tanımlı H matrisine $\mathbb{F}_{2^m}$ cisminin sıfır olmayan herhangi bir elemanını temsil eden b_1 parametresinin uygulanmasıyla elde edilmekte ve ifade 2.7'de gösterilmektedir. 4×4 boyutlarında bir Genel Hadamard matrisi elde etmek içinse bu kez ifade 2.8'de görülebileceği gibi $\mathbb{F}_{2^m}$ cisminin sıfır olmayan herhangi bir elemanını temsil eden b_1, b_2, b_3 şeklinde üç adet parametre kullanılmaktadır. Bu yeni doğrudan tasarım tekniği ile elde edilen GH matrisi, aynı zamanda yukarıda verilen Hadamard matris özelliklerinden üçüncüsünü ($c = \bigoplus_{i=0}^{n-1} a_i$ ve I $n \times n$ boyutunda bir birim matris olmak üzere $H^2 = c^2 I$'dır) de sağlayan bir matris türüdür. Öte yandan bir Hadamard matrise, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı 0'dan farklı elemanlar kullanılarak hem satır hem sütun çarpımı uygulandığında matrisin Hadamard olma özelliği korunmaktadır. Dahası, eğer H matrisi involutif bir MDS matris ise, bir MDS matrisin satır veya sütunlarının $\mathbb{F}_{2^m}$ cisminin 0'dan farklı bir elemanı ile çarpımı MDS olma özelliğini değiştirmediğinden GH matrisi de involutif bir MDS matris olacaktır. b_i parametrelerinin yardımıyla da, $\mathbb{F}_{2^m}$ cisminde tanımlı $n \times n$ boyutlarında bir Hadamard involutif MDS matrisine karşılık, GHadamard matris formu kullanılarak $(2^m - 1)^{n-1}$ farklı involutif MDS matris üretilebilmektedir (Pehlivanoğlu vd., 2018).

Örnek 3.6. $\mathbb{F}_{2^4}/0x13$ *polinomunun bir kökü* α *olmak üzere* 4×4 *boyutlarındaki involutif ve MDS hadamard matris* $H_1 = had(1, \alpha, \alpha^8, \alpha^{10})$ *aşağıdaki gibidir:*

$$H_1 = \begin{bmatrix} 1 & \alpha & \alpha^8 & \alpha^{10} \\ \alpha & 1 & \alpha^{10} & \alpha^8 \\ \alpha^8 & \alpha^{10} & 1 & \alpha \\ \alpha^{10} & \alpha^8 & \alpha & 1 \end{bmatrix}.$$

$b_1 = \alpha^2, b_2 = \alpha^7$ *ve* $b_3 = \alpha^8$ *parametreleri kullanıldığında* H_1 *matrisine karşılık gelen*

$GH_1 = Ghad(1, \alpha; \alpha^2, \alpha^8; \alpha^7, \alpha^{10}; \alpha^8)$ *matrisi aşağıdaki gibi tanımlanmaktadır:*

$$GH_1 = \begin{bmatrix} 1 & \alpha^3 & 1 & \alpha^3 \\ \alpha^{14} & 1 & 1 & \alpha^{14} \\ \alpha & \alpha^5 & 1 & \alpha^2 \\ \alpha^2 & \alpha^2 & 1 & 1 \end{bmatrix}.$$

GH_1 *matrisi involutif ve MDS bir matristir.*

3.2 MDBL Matrislerin Tasarım Teknikleri

Maksimum uzaklıkta ikili doğrusal matrisler (Maximum Distance Binary Linear-MDBL) ARIA (Kwon vd., 2003), PRIDE (Albrecht vd., 2014) ve Camellia (Aoki vd., 2000) gibi modern blok şifrelerde doğrusal dönüşüm katmanı veya bu katmanın ana bileşeni olarak kullanılmaktadır. MDBL matrisler, bir $[2n, n]$ doğrusal kodun minimum Hamming uzaklığının maksimum değerine eşit olan, diğer bir deyişle maksimum dal sayısına sahip $n \times n$ boyutunda ikili matrislerdir. Uygulama safhasında MDS matrislerin XOR işleminin yanı sıra tablo arama ve öteleme gibi çeşitli ölçeklendirme işlemlerine ihtiyaç duymasına rağmen (Nakahara Jr & Abrahao, 2009), MDBL matrislerin sadece XOR işlemi ile uygulanabilir olmaları bir blok şifrede tercih edilmelerinin en önemli sebeplerindendir. Bu avantajlarına karşın MDBL matrislerin aynı boyuttaki maksimum dal sayısı değerleri MDS matrislerin sahip olduğu değerlere göre daha düşüktür. Örneğin bir MDS matrisin maksimum dal sayısı değerleri 4×4 boyutu için 5, 8×8 boyutu için 9 iken aynı boyutlardaki bir MDBL matrisin maksimum dal sayısı değerleri aynı boyutlar için sırasıyla 4 ve 5 olmaktadır.

Literatürde MDBL matrisleri tasarlamak için kullanılan tekniklere, kapsamlı arama yöntemleri (Koo vd., 2006), Feistel mimarisi üzerinde döngü fonksiyonları olarak bit permütasyonlarının kullanıldığı yöntemler (Z. Guo vd., 2015), dairesel ve Hadamard matrisler gibi özel matris tiplerinden veya bunların çeşitli kombinasyonlarından yararlanılan yöntemler (Akleylek vd., 2017; Albrecht vd., 2014; Sakallı vd., 2014), cebirsel inşa yöntemleri (Aslan & Sakallı, 2014; Sakallı & Aslan, 2014) ve devirli matris gruplarının kullanıldığı yöntemler (Akleylek vd., 2016) örnek olarak verilebilmektedir.

3.2.1 Özel Matris Formları Kullanılarak MDBL Matris Tasarımı

Maksimum uzaklıkta ikili matrislerin tasarımı için çoğunlukla dairesel ve Hadamard matrisler ya da bu matrislerin kombinasyonları ile özyinelemeli türevleri kullanılmaktadır. Akleylek vd. (2017) çalışmasında bu hibrit yapılar yardımıyla, yüksek/maksimum dal sayısına sahip bir ikili matris oluşturmak için arama uzayını

büyük ölçüde azaltan bir yöntem kullanmaktadır. Yöntem, $(BHwC)$ olarak adlandırılan Hadamard ve dairesel matrislerin bir kombinasyonu ile $(BCwC)$ olarak adlandırılan dairesel bir matrisin özyinelemeli türevini içermektedir.

Tanım 3.5 (Binary Hadamard with Circulant Matrices-BHwC)**.** t ve m birer pozitif tam sayı olmak üzere $2^t \times 2^t$ boyutunda bir Hadamard blok matris içinde $m \times m$ boyutunda dairesel alt matrislerden oluşan $(2^t \cdot m) \times (2^t \cdot m)$ boyutundaki ikili matrislere, dairesel alt matrislerle oluşturulan ikili Hadamard matris denilmekte ve $(BHwC)_{t,m}$ notasyonu ile gösterilmektedir. Örneğin; C_0, C_1, C_2 ve C_3, 4×4 boyutunda dairesel ikili matrisler olsun. O halde, 4×4 boyutunda bu dairesel alt matrislerden oluşan $2^2 \times 2^2$ boyutundaki Hadamard matris bloğundan elde edilen 16×16 boyutunda $(BHwC)_{2,4}$ ikili matrisi aşağıdaki gibi gösterilmektedir (Akleylek vd., 2017):

$$(BHwC)_{2,4} = had(C_0, C_1, C_2, C_3) = \begin{bmatrix} C_0 & C_1 & C_2 & C_3 \\ C_1 & C_0 & C_3 & C_2 \\ C_2 & C_3 & C_0 & C_1 \\ C_3 & C_2 & C_1 & C_0 \end{bmatrix}.$$

Tanım 3.6 (Binary Circulant with Circulant Matrices-BCwC)**.** q ve m birer pozitif tam sayı olmak üzere $q \times q$ boyutunda bir dairesel blok matris içinde $m \times m$ boyutunda dairesel alt matrislerden oluşan $(q \cdot m) \times (q \cdot m)$ boyutundaki ikili matrislere, dairesel alt matrislerle oluşturulan ikili dairesel matris denilmekte ve $(BCwC)_{q,m}$ notasyonu ile gösterilmektedir. Örneğin; C_0, C_1 ve C_2, 4×4 boyutunda dairesel ikili matrisler olsun. O halde, 4×4 boyutunda dairesel alt matrislerden oluşan 3×3 boyutundaki dairesel matris bloğundan elde edilen 12×12 boyutunda $(BHwC)_{3,4}$ ikili matrisi aşağıdaki gibi gösterilmektedir (Akleylek vd., 2017):

$$(BCwC)_{3,4} = circ(C_0, C_1, C_2) = \begin{bmatrix} C_0 & C_1 & C_2 \\ C_2 & C_0 & C_1 \\ C_1 & C_2 & C_0 \end{bmatrix}.$$

Sezgisel bir arama tekniği olan bu yöntem maksimum dal sayısına sahip $n \times n$ boyutlarında MDBL matrislerin oluşturulması için aşağıdaki adımları kullanmaktadır:

- Hamming ağırlıkları toplamı tek sayı olan uygun vektörlerin bulunması,
- Bu vektörleri kullanarak dairesel matrislerin oluşturulması,

- Bu dairesel matrislerin Hadamard veya dairesel matrisleri kullanarak $(BHwC)_{t,m}$ ve $(BCwC)_{q,m}$ formlarına dönüştürülmesi ve $n \times n$ boyutunda ikili matrislerin elde edilmesi.

Örnek 3.7. $C_0 = (1000)$, $C_1 = (1000)$, $C_2 = (1000)$ *ve* $C_3 = (1111)$ *ikili vektörleri seçilmiş olsun. İkili gösterimle* (1000100010001111) *olarak ifade edilebilen bu vektörlerin Hamming ağırlığı* 7 *olmaktadır. Bu vektörlerden* 4×4 *boyutunda dairesel matrisler aşağıdaki gibi oluşturulur.*

$$(C_0)_{circ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (C_1)_{circ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$(C_2)_{circ} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} (C_3)_{circ} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

4×4 *boyutunda Hadamard matrisin kombinasyonu olan* 16×16 *boyutundaki ikili matris ise* $(BHwC)_{2,4}$ *notasyonu ile gösterilir ve aşağıda verildiği gibi üretilir. Burada* 4×4 *boyutunda Hadamard matrisin her bir elemanı,* 4×4 *boyutunda dairesel bir matristir. Üretilen ikili matrisin her satır ve sütununun hamming ağırlığı* 7*'dir. Ayrıca involutif olan bu matrisin dal sayısı Çizelge 2.2.'den de görülebileceği gibi* 16×16 *boyutu için maksimum değer olan* 8*'e eşittir (Akleylek vd., 2017).*

$$\begin{bmatrix} C_0 & C_1 & C_2 & C_3 \\ C_1 & C_0 & C_3 & C_2 \\ C_2 & C_3 & C_0 & C_1 \\ C_3 & C_2 & C_1 & C_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Tanım 3.6 dikkate alınarak ve aynı adımlar uygulanarak, $(BHwC)_{q,m}$ matrisleri de benzer şekilde elde edilmektedir.

3.2.2 Cebirsel Yöntemler Kullanılarak MDBL Matris Tasarımı

$\mathbb{F}_{2^4}$ cisminin elemanlarının $0x13$, $0x19$ ve $0x1f$ polinomları altında, sonlu cisim elemanı x ile çarpımı sonucu 4×4 boyutunda ikili matris karşılıklarını oluşturan cebirsel bir inşa tekniği de MDBL matrislerin elde edilmesinde kullanılabilmektedir (Aslan & Sakallı, 2014; Sakallı & Aslan, 2014). Oluşturulan bu ikili matrisler daha sonra 2×2, 4×4 ve 8×8 boyutunda rastlantısal, dairesel veya Hadamard matrislerin sonlu cisim elemanları ile yer değiştirilmekte ve sırasıyla 8×8, 16×16 ve 32×32 boyutunda ikili matrisler elde edilmektedir. Elde edilen ikili matrisler söz konusu boyutlarda yüksek/maksimum dal sayısına sahip olmaktadır.

Örnek 3.8. $\mathbb{F}_{2^4}/0x13$ *polinomunun bir kökü* α *olmak üzere herhangi bir* $x \in \mathbb{F}_{2^4}$ *için,*

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

eşitliği yazılabilir. Burada $(\alpha_3, \alpha_2, \alpha_1, \alpha_0) = (\alpha^3, \alpha^2, \alpha, 1)\mathbb{F}_2$ *üzerine* $\mathbb{F}_{2^4}$*'ün bir polinom tabanıdır ve* $(x_3, x_2, x_1, x_0) \in \mathbb{F}_2$*'dir. Dolayısıyla sonlu cisim* $\mathbb{F}_{2^4}$*'ün bir elemanı olan*

α'nın x ile çarpımı,

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

şeklinde elde edilir. Böylece α sonlu cisim elemanı, ifade 3.5'de görülebileceği gibi 4×4 boyutunda bir ikili matris ile temsil edilebilir.

$$\begin{bmatrix} x_0' \\ x_1' \\ x_2' \\ x_3' \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{3.5}$$

Bu işlemler $\mathbb{F}_{2^4}$ cisminin diğer elemanlarına da uygulanarak her elemanın 4×4 boyutunda ikili matris karşılıkları oluşturulur ve aynı yaklaşım $\mathbb{F}_{2^4}/0x19$ ve $\mathbb{F}_{2^4}/0x1f$ polinomları için de tekrarlanır. Son adımda 4×4 boyutunda ikili matrisler, elemanları $\mathbb{F}_{2^4}$ cisminden olan 2×2, 4×4 ve 8×8 boyutunda matrislerin elemanları ile yer değiştirilir ve sırasıyla 8×8, 16×16 ve 32×32 boyutunda ikili matrisler elde edilir. Elemanları $\mathbb{F}_{2^4}$ cisminden olan matrisler, elde edilecek MDBL matrislerde bulunması istenen bazı özelliklere göre seçilmektedir.

Örnek 3.9. *Elemanları $\mathbb{F}_{2^4}/0x13$ polinomu ile tanımlı 2×2 boyutunda involutif ve MDS olan $A = \begin{bmatrix} \alpha^{11} & \alpha^8 \\ \alpha^5 & \alpha^{11} \end{bmatrix}$ matrisi seçilmiş olsun. Matrisin $\alpha^5, \alpha^8, \alpha^{11}$ sonlu cisim elemanları 4×4 boyutunda ikili matris karşılıkları ile yer değiştirildiğinde, 8×8 boyutundaki B ikili matrisi aşağıdaki gibi elde edilir:*

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

Seçilen A matrisi involutif olduğundan elde edilen B matrisi de involutiftir. B matrisinin dal sayısı ise Çizelge 2.2.'den de görülebileceği gibi 8×8 boyutu için maksimum değer olan 5'e eşittir. Diğer bir deyişle B matrisi bir MDBL matristir (Aslan, 2013).

3.2.3 Devirli Matris Grupları Kullanılarak MDBL Matris Tasarımı

Akleylek vd. (2016) çalışmasında devirli matris grupları kullanılarak oluşturulan arama tekniği, Aslan ve Sakallı (2014) yönteminin genelleştirilmiş hali olarak önerilmektedir. Söz konusu teknik için kullanılan yaklaşım aşağıdaki gibi özetlenmektedir:

- $\mathbb{F}_2$ cismi üzerinde tanımlı $k \in \{3, 4\}$ olacak şekilde üretilmiş tüm $k \times k$ boyutundaki matrislerin mertebesi hesaplanır.
- $\mathbb{F}_2$ cismi üzerinde, derecesi k olan indirgenemez polinomun bir kökü ve mertebesi $2^k - 1$ olan matrisler tespit edilir. Bu noktada, 3×3 ve 4×4 boyutundaki ikili matrisler için sırasıyla $0xb$ ve $0x13$ indirgenemez polinomları kullanılmaktadır.
- $k \times k$ boyutunda birbirinden farklı tüm devirli matris grupları elde edilir.

Bu yaklaşım takip edilerek, 3×3 boyutunda $0xb$ polinomunun bir kökü ve mertebesi 7 olan 8 adet, 4×4 boyutunda ise $0x13$ polinomunun bir kökü ve mertebesi 15 olan 336 adet devirli matris grubu elde edilmektedir. Elde edilen bu üreteç matrislerin $2^k - 1$ mertebesine kadar kuvvetlerinin alınmasıyla da $\mathbb{F}_2$ cismi üzerinde tanımlı elemanların ikili matris karşılıkları oluşturulmaktadır.

İkinci aşamada ise, 3×3 boyutunda devirli matris grup elemanları, 2×2 ve 4×4 boyutlarında Hadamard matrislerin $\mathbb{F}_{2^3}$ cisminde tanımlı elemanlarıyla yer değiştirilerek sırasıyla 6×6 ve 12×12 boyutlarında ikili matrisler, 4×4 boyutunda devirli matris grup elemanları ise 2×2 ve 4×4 boyutlarında Hadamard matrislerin $\mathbb{F}_{2^4}$ cisminde tanımlı elemanlarıyla yer değiştirilerek sırasıyla 8×8 ve 16×16 boyutlarında ikili matrisler elde edilmektedir. Son olarak, elde edilen $n \times n$ boyutundaki singüler olmayan ikili matrislerden yüksek/maksimum dal sayısına sahip olanlar tespit edilmektedir. Ayrıca bu teknik, farklı boyutlardaki Hadamard matrislerin kullanımıyla 32×32 ve 64×64 gibi boyutlarda ikili matrislerin oluşturulmasına da olanak sağlarken, $n = k \cdot m$ olmak üzere $m \times m$ boyutunda dairesel matrisler kullanılarak $n \times n$ boyutunda ikili matrislerin elde edilmesinde de uygulanabilir olmaktadır.

Örnek 3.10. *$\mathbb{F}_{2^3}/0xb$ polinomunun bir kökü olan $g = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ matrisi için $g^7 = I$ eşitliği yazılabiliyorsa, g üreteç matrisi kullanılarak oluşturulan devirli matris grubu $\mathbb{F}_{2^3}^*$ cismi ile izomorfik olur. Elemanları $\mathbb{F}_{2^3}/0xb$ polinomu ile tanımlı 4×4 boyutlarında involutif $H = had(I, g, g^2, g^4)$ matrisi seçilmiş olsun. H matrisinin elemanları, devirli*

matris grubunun 3×3 *boyutundaki ikili matris karşılıkları ile yer değiştirildiğinde,* 12×12 *boyutundaki ikili matris aşağıdaki gibi elde edilir:*

$$H = \begin{bmatrix} I & g & g^2 & g^4 \\ g & I & g^4 & g^2 \\ g^2 & g^4 & I & g \\ g^4 & g^2 & g & I \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$

Elde edilen matris involutif olup, matrisin dal sayısı Çizelge 2.2.'den de görülebileceği gibi 12×12 *boyutu için maksimum değer olan* 8*'e eşittir (Akleylek vd., 2016).*

3.3 XOR Sayısı

$\mathbb{R}$ rastgele bir halka, $\mathcal{M}_n(\mathbb{R})$ ise girişleri $\mathbb{R}$ halkasından seçilen tüm $n\times n$ boyutunda matrisler kümesi olmak üzere $\mathcal{M}_n(\mathcal{M}_l(\mathbb{F}_2))$, elemanları $\mathcal{M}_l(\mathbb{F}_2)$ kümesinden seçilen tüm $n \times n$ boyutunda matrisler kümesini göstermektedir ve $\mathcal{M}_n(\mathcal{M}_l(\mathbb{F}_2))$ kümesinin elemanı olan herhangi bir M matrisi, $nl \times nl$ boyutunda bir ikili matris olarak temsil edilmektedir. $M \in \mathcal{M}_l(\mathbb{F}_2)$ olan herhangi bir difüzyon matrisi de sonlu sayıda XOR kapısı ile uygulanabilmektedir.

Düşük XOR sayısı, donanım alanının en aza indirilmesiyle güçlü bir şekilde bağlantılıdır (Khoo vd., 2014). Özellikle MDS matrisler sahip oldukları maksimum dal sayısı ölçütü ile mükemmel difüzyon tabakası bileşenleri olsalar da uygulanma sırasında oldukça maliyetli olabilirler. Yazılım performansları, bellek kısıtı olmadığından ve tablo tabanlı uygulamalar sonlu cisim çarpımlarını doğrudan depolanan değerlere dahil ettiğinden bu maliyetten çok fazla etkilenmez. Ancak donanım uygulamaları, sonlu cisim çarpımlarından dolayı önemli bir alan gereksinimine ihtiyaç duyacaktır. Bu sebeple iyi donanım uygulaması verimliliği, başka bir ifadeyle minimum XOR sayısına sahip uygulamayı bulmak, modern kriptografide önemli bir tasarım trendi haline gelmektedir. Genel olarak, düşük Hamming ağırlığının uygulamalarda genellikle daha az donanım kaynağı gerektirdiği bilinmektedir. Dolayısıyla bu durum, bir difüzyon

matrisini belirlemek için önemli bir seçim kriteri olarak kabul görmektedir. Örneğin, AES (Deamen & Rijmen, 2002) blok şifresinin difüzyon matrisinin katsayıları, az maliyetli bir uygulama sağlayacağı düşünüldüğü için $0x1$, $0x2$ ve $0x3$ şeklinde seçilmiştir. Ancak, Khoo vd. (2014)'de bu sezgisel yöntemin her zaman doğru olmadığı gösterilmektedir. Sonlu cismi tanımlayan indirgenemez polinoma bağlı olarak ve çeşitli indirgeme etkileri nedeniyle, yüksek Hamming ağırlığına sahip bazı katsayıların çok düşük XOR maliyeti ile uygulanabildiği belirtilmektedir.

Sıradaki üç örnek, 2.39 ve 2.40 tanımlarından faydalanarak, tek bir sonlu cisim elemanının farklı indirgenemez polinomlar altındaki XOR maliyetinin ve 4×4 boyutunda bir matrisin toplam XOR sayısının hesaplanmasını göstermektedir.

Örnek 3.11. (b_3, b_2, b_1, b_0), $\mathbb{F}_{2^4}/0x13$ *cismi üzerinde tanımlı ve rastgele seçilen bir* β *elemanının ikili gösterimini temsil etmek üzere yine aynı cisim üzerinde tanımlı* $\alpha = 6_h$ *elemanının XOR sayısı aşağıdaki gibi hesaplanmaktadır:*

$$\begin{aligned}(0,1,1,0).(b_3,b_2,b_1,b_0) &= (b_1, b_0 \oplus b_3, b_2 \oplus b_3, b_2) \oplus (b_2, b_1, b_0 \oplus b_3, b_3) \\ &= (b_1 \oplus b_2, b_0 \oplus b_1 \oplus b_3, b_0 \oplus b_2, b_2 \oplus b_3)\end{aligned}$$

$\mathbb{F}_{2^4}/0x13$ *cismi üzerinde tanımlı* $\alpha = 6_h$ *elemanının çarpma işlemi maliyetinin* 5 XOR *olduğu görülebilmektedir.*

Örnek 3.12. (b_3, b_2, b_1, b_0), $\mathbb{F}_{2^4}/0x19$ *cismi üzerinde tanımlı ve rastgele seçilen bir* β *elemanının ikili gösterimini temsil etmek üzere yine aynı cisim üzerinde tanımlı* $\alpha = 6_h$ *elemanının XOR sayısı ise aşağıdaki gibi hesaplanmaktadır:*

$$\begin{aligned}(0,1,1,0).(b_3,b_2,b_1,b_0) &= (b_1 \oplus b_2 \oplus b_3, b_0, b_3, b_2 \oplus b_3) \oplus (b_2 \oplus b_3, b_1, b_0, b_3) \\ &= (b_1, b_0 \oplus b_1, b_0 \oplus b_3, b_2)\end{aligned}$$

$\mathbb{F}_{2^4}/0x19$ *cismi üzerinde tanımlı* $\alpha = 6_h$ *elemanının çarpma işlemi maliyetinin* 2 XOR *olduğu görülebilmektedir.*

Bu örneklerden de açıkça anlaşılabileceği gibi indirgenemez polinomun seçimine bağlı olarak, bir sonlu cisim elemanı ile çarpma işlemi için gereken XOR sayısı farklılık göstermektedir. Sim vd. (2015) çalışmasında, farklı indirgenemez polinom kullanımın sonlu cisim elemanları arasında farklı XOR dağılımına neden olduğu belirtilmektedir. Bu dağılım standart sapma ile ölçülmekte ve yüksek bir standart sapma, bu polinomun düşük XOR sayısına sahip daha fazla eleman içerdiği anlamına gelmektedir. Dolayısıyla standart sapma değeri yüksek olan bir indirgenemez polinom kullanımı, daha düşük XOR sayısına sahip bir difüzyon matrisi bulma şansını arttırmaktadır. Çizelge 3.1, $\mathbb{F}_{2^4}$ cismi

üzerinde tanımlı $0x13$, $0x19$ ve $0x1f$ indirgenemez polinomları kullanıldığında, her bir sonlu cisim elemanı ile çarpma işlemi için gereken XOR sayısını ve bu XOR sayısı dağılımlarının standart sapma değerlerini göstermektedir (Sim vd., 2015).

Çizelge 3.1. $\mathbb{F}_{2^4}$ Sonlu Cisim Elemanları Çarpımlarının İndirgenemez Polinoma Bağlı XOR Sayıları

Onaltılık Tabanda Gösterim	**Sonlu Cisim Elemanı**	**XOR Sayıları**		
		0x13	**0x19**	**0x1f**
0	0	0	0	0
$0x1$	1	0	0	0
$0x2$	α	1	1	3
$0x3$	$\alpha+1$	5	3	5
$0x4$	α^2	2	3	3
$0x5$	α^2+1	6	5	5
$0x6$	$\alpha^2+\alpha$	5	2	6
$0x7$	$\alpha^2+\alpha+1$	9	6	6
$0x8$	α^3	3	6	3
$0x9$	α^3+1	1	8	5
$0xa$	$\alpha^3+\alpha$	8	5	6
$0xb$	$\alpha^3+\alpha+1$	6	9	6
$0xc$	$\alpha^3+\alpha^2$	5	1	6
$0xd$	$\alpha^3+\alpha^2+1$	3	5	6
$0xe$	$\alpha^3+\alpha^2+\alpha$	8	6	5
$0xf$	$\alpha^3+\alpha^2+\alpha+1$	6	8	3
Standart Sapma		2,68	2,68	1,71

Örnek 3.13. $\mathbb{F}_{2^4}/0x13$ *cismi üzerinde tanımlı* $H = had(0x1, 0x4, 0x9, 0xd)$ *matrisinin XOR sayısını hesaplamak içinse Tanım 2.40 gereği aşağıdaki adımlar izlenmektedir:*

- *Çizelge 3.1.'den faydalanılarak H matrisinin ilk satır elemanlarının XOR sayısı toplamı* $(0+2+1+3) = 6$ *olarak elde edilir,*
- *((Satırın* 0 *Olmayan Elemanlarının Sayısı*-1*) × Tanımlı Cismin Mertebesi), ilk satır için* $3 \times 4 = 12$ *XOR olarak hesaplanır,*
- *İlk satırın toplam XOR sayısı* $(0+2+1+3)+3\times 4 = 18$ *olarak bulunur,*
- *İlk üç adım matrisin diğer satırları için tekrarlanır ve bulunan değerlerin hepsi toplanarak matrisin toplam XOR maliyeti elde edilmiş olur. Örneği verilen H matrisi Hadamard bir matris olduğundan, tüm satırları XOR sayısı cinsinden eşdeğerdir. Dolayısıyla H matrisinin toplam XOR sayısı* $18 \times 4 = 72$ *olarak bulunur.*

3.4 Optimizasyon Teknikleri

XOR sayısı işlenmesi kolay ve basitleştirilmiş bir metrik olmasına rağmen minimum XOR sayısına sahip uygulamayı elde etmek zorlu bir problemdir. Bunun en önemli sebebi, XOR sayısı ile ilişkili mevcut ölçütlerin sadece üst sınırları belirlemesi, minimum sınırları tanımlayan bir ölçüt bulunmamasıdır. $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı $n \times n$ boyutundaki matrislerin XOR maliyetini hesaplamanın genel bir yolunu Khoo vd. (2014) çalışmasında önerdikten sonra, literatürde uygulanmaları için gereken XOR işlemlerinin sayısını en aza indiren difüzyon matrislerini bulmak amacıyla iki farklı yaklaşımın öne çıktığı görülmektedir. Bunlardan ilki, difüzyon matrisinin her bir elemanının ayrı ayrı XOR maliyetinin hesaplanmasına ve yine eleman bazında XOR işlemleri sayısının iyileştirilmesine odaklanan yerel (local) optimizasyon teknikleridir. Küresel (global) optimizasyon teknikleri olarak bilinen ikinci yaklaşımda ise eleman bazında XOR maliyetine bakılmaksızın difüzyon matrisi bir bütün olarak ele alınmaktadır. Bu difüzyon matrisi, öncelikle ikili bir matrise dönüştürülmekte ve ardından problem, $mn \times mn$ boyutunda bir ikili matrisin optimizasyonu olarak düşünülmektedir.

3.4.1 Yerel Optimizasyon

Yerel optimizasyon teknikleri, XOR sayısı ile bağlantılı iki farklı ölçüt içermektedir. Khoo vd. (2014) çalışmasında önerilen ve Jean vd. (2017)'de doğrudan XOR sayısı olarak adlandırılan ilk ölçüt, bir M matrisinin her satırının eleman eleman işlem gördüğü ham (naive) bir donanım uygulamasına karşılık gelmektedir. XOR sayısının hesaplanmasında en çok kullanılan yöntemdir ve bir matrisin ham XOR sayısı denildiğinde, doğrudan XOR sayısı kastedilmektedir.

Jean vd. (2017) çalışmasında önerilen ve sıralı XOR sayısı olarak adlandırılan ikinci yerel optimizasyon ölçütü, ekstra bellek depolaması kullanılmadan ve yerinde işlemlerle gerçekleştirilen sıralı bir uygulamadaki minimum XOR sayısı olarak tanımlanmaktadır.

3.4.1.1 Doğrudan XOR Sayısı

Doğrudan XOR sayısı (Direct XOR Count), bir M matrisi için $DXC(M)$ ya da $d-XOR(M)$ notasyonu ile gösterilmektedir. Aslında, Örnek 3.13'deki H matrisi için hesapladığımız XOR sayısı, $d-XOR$ sayısını işaret etmektedir. Diğer taraftan sonlu bir $\mathbb{F}_{2^m}$ cismi verildiğinde, cismin elemanları, Örnek 3.11 ve 3.12'de olduğu gibi $\mathbb{F}_2$ cismi üzerinde tanımlı m boyutlu ikili vektörler halinde temsil edilebilmektedir. Bu durumda $\mathbb{F}_{2^m}$'deki sıfır olmayan bir α elemanının çarpımı, $\mathbb{F}_2$ cismine göre m dereceli bir sol çarpım matrisi ile gösterilmektedir. Genellikle çarpma matrisi olarak bilinen ve M_α ile

gösterilen bu matrisin XOR sayısı da α sonlu cisim elemanının $d-XOR$ sayısı olarak ifade edilmektedir (Beierle vd., 2016; Sarkar & Sim, 2016; Sim vd., 2015).

Bu açıklamalara dayanarak $d-XOR$ ölçütünün biçimsel tanımı aşağıdaki gibi verilmektedir.

Tanım 3.7 (Doğrudan XOR Sayısı)**.** $\mathbb{F}_{2^m}/p(x)$ cismi üzerinde tanımlı bir sonlu cisim elemanı olan α'nın $d-XOR$ değeri, $\alpha : \beta \rightarrow \alpha\beta$ şeklindeki sonlu cisim çarpımını uygulamak için gereken XOR işlemlerinin sayısıdır ve w, M_α çarpma matrisinin Hamming ağırlığı, r ise M_α çarpma matrisinin satır sayısı olmak üzere;

$$d-XOR(M_\alpha) = wM_\alpha - r \tag{3.6}$$

şeklinde ifade edilmektedir (Khoo vd., 2014).

$d-XOR$ değerinin yalnızca sonlu cisim elemanlarının uygulama maliyetinin üst sınırının tahmin edilmesini sağladığına dikkat edilmelidir. Sıralı XOR sayısı ölçütünde de görülebileceği gibi pratikte gereken XOR işlemlerinin sayısı daha düşük olabilir.

3.4.1.2 Sıralı XOR Sayısı

Sıralı XOR sayısı (Sequential XOR Count), bir sonlu cisim elemanı (Beierle vd., 2016) ile çarpmanın optimize edilmiş uygulamasını elde etmek için kullanılmakta ve bir M matrisi için $SXC(M)$ ya da $s-XOR(M)$ notasyonu ile gösterilmektedir. Doğrudan XOR sayısını iyileştirmek için önerilmektedir (Jean vd., 2017). Pratikte, rastgele bir giriş vektörü verildiğinde, sadece vektörün bileşenleri güncellenerek herhangi bir sonlu cisim çarpması yerinde uygulanabilmektedir. Yerinde algoritmalar yürütülürken, elde edilen çıktı ya tekrar girdinin üzerine yazılır ya da giriş elemanlarının yerlerinin değişmesine neden olur. Dolayısıyla bu tip algoritmalar genellikle ekstra bellek depolaması kullanmamaktadır. $s-XOR$ ölçütünde bu yerinde algoritma bir dizi sıralı XOR işlemiyle gerçekleştirilmektedir. Yerinde algoritmalarla ilgili çeşitli kısıtlamalara rağmen, $s-XOR$ sayısının $d-XOR$ sayısından önemli ölçüde düşük olabileceğini gösteren örnekler bulunmaktadır. Bununla birlikte, 32×32 ve daha büyük boyutlu ikili matrislerin $s-XOR$ sayısını belirlemek mümkün olmamaktadır (Duval & Leurent, 2018).

Bu açıklamalara dayanarak $s-XOR$ ölçütünün biçimsel tanımı aşağıdaki gibi verilmektedir.

Tanım 3.8 (Sıralı XOR Sayısı)**.** $\mathbb{F}_{2^m}/p(x)$ cismi üzerinde tanımlı bir sonlu cisim elemanı olan α'nın $s-XOR$ değeri, bir çarpma matrisi olan M_α matrisinin sol çarpımını uygulamak için gereken minimum XOR işlemi sayısıdır. Burada minimum değer, tüm uygulama dizileri dikkate alınarak hesaplanır ve $s-XOR(M_\alpha)$ ile gösterilir (Jean vd., 2017).

Örnek 3.14. *(b_2, b_1, b_0), $\mathbb{F}_{2^3}/0xb$ cismi üzerinde tanımlı ve rastgele seçilen bir β elemanının ikili gösterimini temsil etmek üzere yine aynı cisim üzerinde tanımlı $\alpha = 6_h$ elemanının $d-XOR$ ve $s-XOR$ sayıları aşağıdaki gibi hesaplanmaktadır:*

$$\begin{aligned}(1,1,0).(b_2,b_1,b_0) &= (b_0 \oplus b_2, b_1 \oplus b_2, b_1) \oplus (b_1, b_0 \oplus b_2, b_2) \\ &= (b_0 \oplus b_1 \oplus b_2, b_0 \oplus b_1, b_1 \oplus b_2)\end{aligned}$$

$\mathbb{F}_{2^3}/0xb$ cismi üzerinde tanımlı $\alpha = 6_h$ elemanının çarpma işlemi maliyeti $d-XOR(\alpha) = 4\ XOR$ olarak bulunur. M_α çarpım matrisi kullanılarak hesaplama yapılmak istendiğinde ifade 3.6 uyarınca;

$$\begin{bmatrix}1 & 1 & 1\\ 0 & 1 & 1\\ 1 & 1 & 0\end{bmatrix}\begin{bmatrix}b_2\\ b_1\\ b_0\end{bmatrix} = \begin{bmatrix}b_0 \oplus b_1 \oplus b_2\\ b_0 \oplus b_1\\ b_1 \oplus b_2\end{bmatrix}$$

M_α matrisinin Hamming ağırlığı $w = 7$, satır sayısı $r = 3$ olduğundan, $d-XOR(M_\alpha) = wM_\alpha - r = 7-3 = 4\ XOR$ değerinin tekrar elde edilebildiği görülebilir.

α sonlu cisim elemanının $s-XOR$ değerini elde etmek içinse, $b_0 \oplus b_1$ ve $b_1 \oplus b_2$ bileşenlerinin ilk bileşende de bulunuyor olması dikkate alınır. Dolayısıyla pratikte, istenen çıktıyı elde etmek için giriş vektör bileşenleri birbirine dönüştürülebilir. İlk olarak, üçüncü bileşenle ikinci bileşen XOR işlemine tabi tutularak $b_0 \oplus b_2$, ardından bu bileşenle birinci bileşen XOR işlemine tabi tutularak b_1 elde edilir. Böylece, çarpma matrisini uygulamak için böyle bir sıralı işlemin $2\ XOR$ maliyeti ile gerçekleştirilebildiği, bir başka deyişle $s-XOR(\alpha) = 2\ XOR$ olduğu doğrulanabilir.

3.4.2 Global Optimizasyon

Difüzyon matrisinin ikili matrise dönüştürülerek bir bütün olarak ele alındığı global optimizasyon teknikleri, bu özel amaç için tasarlanmış çeşitli araçların kullanımına dayanmaktadır (Duval & Leurent, 2018; Kranz vd., 2017) ve temel amaç, en kısa doğrusal düz çizgi programı (Shortest Linear Straight Line Program-SLP) olarak bilinen probleme çözüm aramaktır.

Tanım 3.9 (En Kısa Doğrusal Düz Çizgi Programını Bulma Problemi)**.** M ve x sırasıyla $\mathbb{F}_2$ cismi üzerinde tanımlı $m \times n$ boyutunda bir sabitler matrisi ve n değişkenlerinin bir vektör temsili olsun. Her program satırı belirli bir formda olmak üzere $M \cdot x$'i hesaplayan en küçük satır sayısına sahip programın bulunması $\mathbb{F}_2$ cismi üzerinde bir SLP problemi olarak tanımlanmaktadır.

$y_i = x_i$ ve $y_i, y_j \in V$ olmak üzere V, başlangıçta $\{y_0, y_1, \cdots, y_{n-1}\}$ girdi

değişkenlerini içeren, $\mathbb{F}_2$ cismi üzerinde tanımlı bir değişkenler kümesi olsun. Bu durumda her bir program satırı $y' \leftarrow y_i \oplus y_j$ formuna sahip olur. Bu program satırını çalıştırdıktan sonra, kümeye yeni y' değişkeni eklenir ve $V \leftarrow V \cup \{y'\}$ elde edilir. Yeni değişken y' bu nedenle bir sonraki program satırında kullanılabilir. Sonuç olarak, $0 \leq j \leq n-1$ olmak üzere bir $y_{i_j} \in V$ için $M \cdot x = (y_{i_0}, y_{i_1}, \cdots, y_{i_{n-1}})$ ifadesi yazılabiliyorsa, programın $M \cdot x$ değerini hesaplayabildiği söylenir (S. Li vd., 2019).

Yukarıdaki tanımdan da anlaşılacağı gibi SLP problemi, aslında doğrusal bir denklem sistemini tamamen ifade etmek için gereken minimum 2 girişli XOR kapısı sayısına eşit olmaktadır. Dolayısıyla global optimizasyonda amaç, bir difüzyon matrisinin minimum sayıda XOR kapısı ile uygulanması olduğundan, bunun SLP probleminin bir örneğini $\mathbb{F}_2$ cismi üzerinden çözmeye eşdeğer olduğu görülmektedir.

SLP, bilgisayar bilimlerinde iyi bilinen zor bir problemdir. Literatürde yıllar içinde, SLP problemini çözmek için çeşitli yaklaşım algoritmaları öneren birçok girişimde bulunulmuştur. Boolean tatmin edilebilirlik problemi (Boolean Satisfiability Problem-SAT) olarak bilinen ve belirli bir Boole formülünün değişkenlerinin, formülün *Doğru* olarak değerlendirileceği şekilde, *Doğru* veya *Yanlış* değerleriyle tutarlı bir şekilde değiştirilip değiştirilemeyeceğine cevap arayan SAT çözücülerini kullanmak bu girişimlerden biridir (Fuhs & Schneider-Kamp, 2010). SAT tabanlı çözümler, benzer yaklaşımlar kullanılarak S-kutularının kapı karmaşıklıklarını en aza indirmek için daha önce kullanılmıştır (Stoffelen, 2016). Ancak, çözücünün kendisinin çalışma süresi girdi boyutunda üsteldir ve 10×10 boyutundan büyük ikili matrisler için çözücüden makul sürede bir çözüm elde etmek oldukça zordur. Bununla birlikte, daha büyük boyuttaki ikili matrislerin verimli bir şekilde uygulanması için çeşitli sezgisel algoritmalar önerilmiştir. Bunlardan ilki Paar (1997) çalışmasıdır. Reed-Solomon kodlayıcıları tarafından kullanılan aritmetiğin nasıl optimize edileceğini inceleyen ve iki adet sezgisel algoritma içeren söz konusu çalışma esasen, $\mathbb{F}_2$ cismi üzerinde tanımlı sabit bir çarpan için gerekli olan XOR kapılarının sayısını azaltmakla ilgilidir. İlk algoritma her aşamada denklemler kümesinde en sık görülen terim çiftini bulur ve bunları yeni bir değişkenle değiştirir. İşlem, tüm terimler tam olarak bir kez görünene kadar devam eder. İkinci algoritma ilkine benzer olmakla birlikte, birden fazla terim çifti eşit sayıda görüldüğünde farklılık içerir. Sadece ilk çifti almak yerine, hepsini tekrar tekrar dener. Algoritma bu nedenle çok daha yavaştır, ancak biraz daha iyi sonuçlar verebilir. Paar algoritmaları, SLP problemine yalnızca *cancellation-free* olarak bilinen yaklaşımı kullanarak çözümler üretir. Bu yaklaşım temel olarak herhangi iki ara terim ifadesinin, giriş değişkenlerinin doğrusal bir denklemi olarak yazılabilmesi halinde, iki ifadenin herhangi bir ortak terim içermemesi olarak tanımlanır. Ham XOR sayısıyla karşılaştırıldığında Paar algoritmaları, $\mathbb{F}_{2^4}$ ve $\mathbb{F}_{2^8}$ cisimleri

üzerinde tanımlı matrisler için XOR sayısında sırasıyla $\%17,5$ ve %40'lık bir azalma kaydetmiştir. Ancak, Boyar vd. (2008)'de *cancellation-free* tekniklerin optimal çözümler üretmesinin beklenmediğini göstermiş, Boyar ve Peralta (2010) çalışmasında da SLP probleminin NP-Zor sınıfında olduğunu kanıtlayarak *cancellation-free* olmayan yeni bir sezgisel yöntem geliştirmişlerdir (Boyar vd., 2013). Boyar-Peralta (BP) algoritması olarak bilinen bu yöntemin ana fikri, bir çıkış bitinin hedeflenen her ifadesi için gerekli olan ve halihazırda hesaplanmış ara değerlerin minimum ekleme sayısını içeren bir mesafe vektörünü bulmaktır. Hangi değerlerin ekleneceğine karar vermek için yeni mesafelerin toplamını en aza indiren çift seçilir. Ek olarak eğer iki değer, eklenmesi hedeflenen bir çıkış değişkeninin elde edilmesini sağlarsa, aynı çıkış değerini verecek diğer olasılıkların aranmasına gerek kalmaz. Algoritma, ancak ve ancak mesafe vektörü 0 olursa, diğer bir deyişle sinyal kümesi, doğrusal yüklemler kümesini hesaplarsa yürütmeyi durdurur. BP algoritması çoğu senaryoda Paar algoritmalarından daha verimli sonuçlar üretir ancak, 32×32 ve daha büyük boyutlar söz konusu olduğunda Paar algoritması daha hızlı çalışmaktadır. Paar ve BP algoritmalarının detayları ve algoritmalara dair örnekler aşağıda incelenmektedir.

3.4.2.1 PAAR Algoritması

Paar'ın algoritması bir açgözlü (greedy) algoritma örneğidir. Açgözlü yaklaşım basitçe, bir seçim yapılması gerektiğinde sonuca en çok yaklaştıracak olan seçimin yapılmasını önerir (Black, 2012). Ancak tahmin edilebileceği üzere, bu seçimin her zaman için en iyi seçim olması mümkün değildir. Bu sebeple Paar algoritmasının optimal çözümler üretmesi beklenmez fakat, BP algoritmasının pratik zamanda bir çözüm getiremediği daha büyük boyutlu matrislerin XOR sayısını azaltmak için bu algoritmayı kullanmak her zaman yararlı olmaktadır.

M, XOR sayısı minimize edilmek istenen bir matris olsun. Bu durumda Paar algoritması aşağıdaki adımlarda belirtilen işlemleri gerçekleştirir:

- Bit düzeyinde AND işlemiyle en büyük Hamming ağırlığına sahip sütunlar tespit edilir. Bu işlem temelde, XOR işlemleri içinde en çok tekrar edilen x_i ve x_j terimlerini bulmamızı sağlar.
- İlk adımda elde edilen çarpım sütunu *"yeniSütun"* adı ile matrise eklenerek M matrisi genişletilir.

Bu iki adım, hem XOR işlemlerine $v = x_i \oplus x_j$ işlemini ekler hem de çarpım sütununu M matrisine ekleyerek yeni bir giriş değişkeni olan v'yi oluşturur.

- Eski sütunlar için $eskiStn \leftarrow eskiStn \cdot \overline{yeniStn}$ işlemi gerçekleştirilir.

Bu işlem de matris yapısına yeni sütunun eklenmesinden sonra gerekli olmayan fazladan XOR işlemlerini kaldırır.

Örnek 3.15. *Aşağıda gibi bir doğrusal sistem verilmiş olsun.*

$$\begin{aligned} x_2 \oplus x_3 \oplus x_4 &= y_1 \\ x_2 \oplus x_3 \oplus x_4 &= y_2 \\ x_1 \oplus x_2 \oplus x_3 \oplus x_4 &= y_3 \\ x_1 \oplus x_2 \oplus x_3 &= y_4 \end{aligned} \quad \Longrightarrow \quad \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

Bu sistemde, ikinci ve üçüncü sütunun çarpımı en büyük Hamming ağırlığına sahip olduğundan $v_1 = x_2 \oplus x_3$ *değeri ve* $(1\ 1\ 1\ 1) \cdot (1\ 1\ 1\ 1) = (1\ 1\ 1\ 1)$ *şeklindeki yeni çarpım sütunu elde edilir. Bu çarpım sütunu beşinci sütun olarak matrise,* v_1 *ise giriş değişkenlerine eklenir. Ardından aynı çarpım sütununun değili ile ikinci ve üçüncü sütunlara* $eskiStn \leftarrow eskiStn \cdot \overline{yeniStn}$ *işlemi uygulanır. Oluşan son durum aşağıdaki gibidir:*

$$\begin{aligned} x_4 \oplus v_1 &= y_1 \\ x_4 \oplus v_1 &= y_2 \\ x_1 \oplus x_4 \oplus v_1 &= y_3 \\ x_1 \oplus v_1 &= y_4 \end{aligned} \quad \Longrightarrow \quad \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ v_1 \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

İşleme tüm terimler tam olarak bir kez görününceye kadar devam edilir. Bu esnada gerçekleştirilen XOR işlemleri sayısının toplamı, matrisin XOR maliyetini oluşturur (Banik vd., 2019).

3.4.2.2 Boyar-Peralta (BP) Algoritması

BP algoritması, $\mathbb{F}_2$ cismi üzerinde tanımlı $m \times n$ boyutunda bir M matrisi için $f(x) = M \cdot x$ değerini hesaplayan bir en kısa doğrusal düz çizgi programı bulmayı amaçlar. Bunun için ilk önce bilinen doğrusal fonksiyonları içeren temel bir S kümesi oluşturulur. S kümesi başlangıçta sadece $\{x_0, x_1, \cdots, x_{n-1}\}$ değerlerini içeren giriş değişkenleri kümesidir. Ardından S kümesinden M matrisinin satırlarına Hamming uzaklıklarının kümesi olan bir mesafe vektörü tanımlanır ve $Msf[.]$ notasyonu ile gösterilir. Eğer f_i, M matrisinin i satırından elde edilebilen bir doğrusal fonksiyon ise S kümesine söz konusu f_i değerini ekleyebilmek için gerekli olan minimum fonksiyon sayısı $Msf[i]$ ile temsil edilir. Bir diğer deyişle başlangıç durumunda $Msf[i]$ vektörü, i satırının Hamming ağırlığından sadece 1 azdır. Bu işlemlerin devamında bir döngü halinde aşağıdaki adımlar uygulanır:

- S kümesinden aşağıdaki kriterlere göre iki eleman seçilip XOR işlemine tabi tutulur ve elde edilen yeni eleman S kümesine eklenir.
- S kümesine yeni eklenen bu eleman da göz önüne alınarak $Msf[i]$ vektörü güncellenir.
- Tüm i değerleri için $Msf[i] = 0$ olana kadar yukarıdaki iki adım tekrar edilir.

Algoritmanın herhangi bir aşamasında S kümesinin boyutu t ise yeni bir temel eleman seçmek için $\binom{t}{2}$ adet seçenek bulunur ve yeni temel elemanı seçme kriterleri aşağıdaki gibi tanımlanır:

1. Güncellenen $Msf[i]$ vektörünün öğelerinin toplamını en aza indiren eleman seçilir.
2. Yeni elemanın iki seçeneği arasında eşitlik varsa, o zaman güncellenmiş $MSf[i]$ vektörünün Öklid normunu (bir vektör dizisi elemanlarının mutlak kareleri toplamının karekökü) maksimize eden eleman seçilir.

Bu eşitlik sorununu çözme kriteri ilk bakışta mantık dışı gibi görünebilir. Ancak temel fikir, $[0,0,3,1]$ gibi bir uzaklık vektörünün, $[1,1,1,1]$ gibi bir uzaklık vektörüne tercih edilmesidir. Çünkü ikinci durumda, uzaklık vektörünü 0'a eşitlemek için 4 adet XOR kapısına daha ihtiyaç varken, ilkinde, 3 adet XOR kapısı ile bu işlem yapılabilir. BP sezgisel yönteminin zaman maliyetinin çok büyük kısmını yeni temel öğeyi seçme işlemi oluşturur.

BP algoritmasının anlaşılmasına dair Boyar ve Peralta (2010) çalışmasından alınan bir örnek aşağıda verilmektedir.

Örnek 3.16. *Aşağıdaki şekilde tanımlanan denklem sistemini hesaplayan bir devreye ihtiyacımız olduğunu varsayalım. Bu doğrusal sistemin çözümü, sağ tarafta verilen* 6×5 *boyutunda M matrisiyle çarpma için bir devre bulmaya eşdeğerdir.*

$$\begin{aligned} x_0 \oplus x_1 \oplus x_2 &= y_0 \\ x_1 \oplus x_3 \oplus x_4 &= y_1 \\ x_0 \oplus x_2 \oplus x_3 \oplus x_4 &= y_2 \\ x_1 \oplus x_2 \oplus x_3 &= y_3 \\ x_0 \oplus x_1 \oplus x_3 &= y_4 \\ x_1 \oplus x_2 \oplus x_3 \oplus x_4 &= y_5 \end{aligned} \quad \Longrightarrow \quad \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \end{bmatrix}$$

Hesaplanacak çıkış fonksiyonları M matrisin satırlarını temsil etmektedir. Başlangıç temel kümesi $S = \{10000, 01000, 00100, 00010, 00001\}$*'e karşılık gelen*

$\{x_0, x_1, x_2, x_3, x_4\}$ *değerleridir. Buna göre mesafe vektörü* $Msf = [2,2,3,2,2,3]$ *şeklinde tanımlanır.*

Adım1: *Algoritma, toplamı S kümesine eklendiğinde yeni mesafelerin toplamını en aza indiren iki temel vektör bulur. Tüm olasılıklar incelendiğinde doğru seçimin* $t_5 = x_1 \oplus x_3$ *olduğu görülebilir. Dolayısıyla S kümesi,* 01010 *sinyalini içerecek şekilde genişletilir ve yeni mesafe vektörü* $Msf = [2,1,3,1,1,2]$ *olur.*

Adım2: *Algoritmaya göre* $x_1 \oplus x_2$ *veya* $x_0 \oplus x_5$ *vektörleri seçilebilir. Bu seçimler için güncellenmiş* Msf *vektörleri sırasıyla* $Msf = [1,1,3,1,1,2]$ *ve* $Msf = [2,1,3,1,0,2]$ *olacaktır. Eşitlik söz konusu olduğundan vektör dizilerinin Öklid norm değerleri hesaplanır. İlk seçenek* $\sqrt{17}$*, ikinci seçenek ise* $\sqrt{19}$ *olarak bulunur. İkinci seçeneğin Öklid normu maksimize ettiği görüldüğünden* $t_6 = x_0 \oplus x_5 = y_4$ *değeri seçilir.*

Adım3: $t_7 = x_2 \oplus t_5 = y_3$*, yeni* $Msf = [2,1,3,0,0,1]$

Adım4: $t_8 = x_4 \oplus t_5 = y_1$*, yeni* $Msf = [2,0,3,0,0,1]$

Adım5: $t_9 = x_2 \oplus t_8 = y_5$*, yeni* $Msf = [2,0,2,0,0,0]$

Adım6: $t_{10} = x_0 \oplus x_1$*, yeni* $Msf = [1,0,1,0,0,0]$

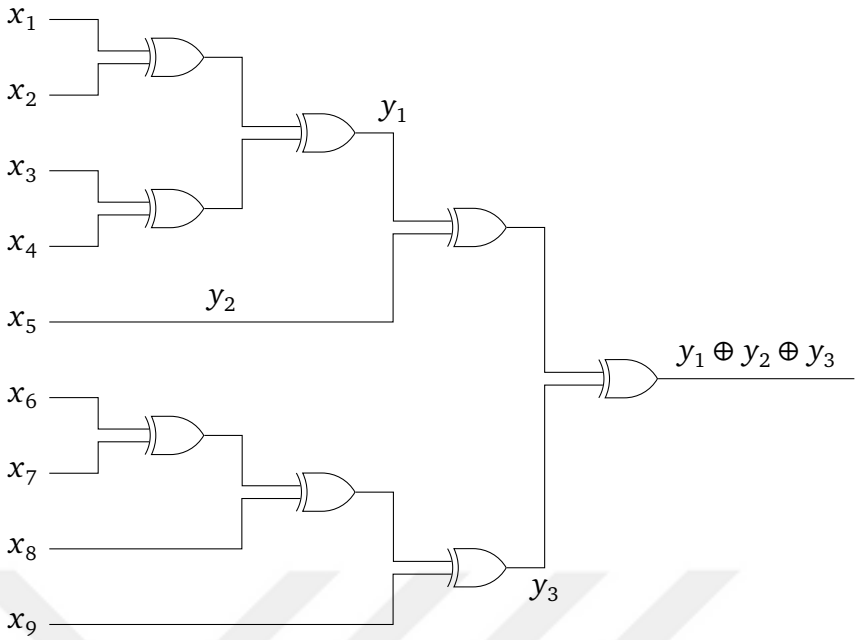
Adım7: $t_{11} = x_2 \oplus t_{10} = y_0$*, yeni* $Msf = [0,0,1,0,0,0]$

Adım8: $t_{12} = t_8 \oplus t_{11} = y_2$*, yeni* $Msf = [0,0,0,0,0,0]$

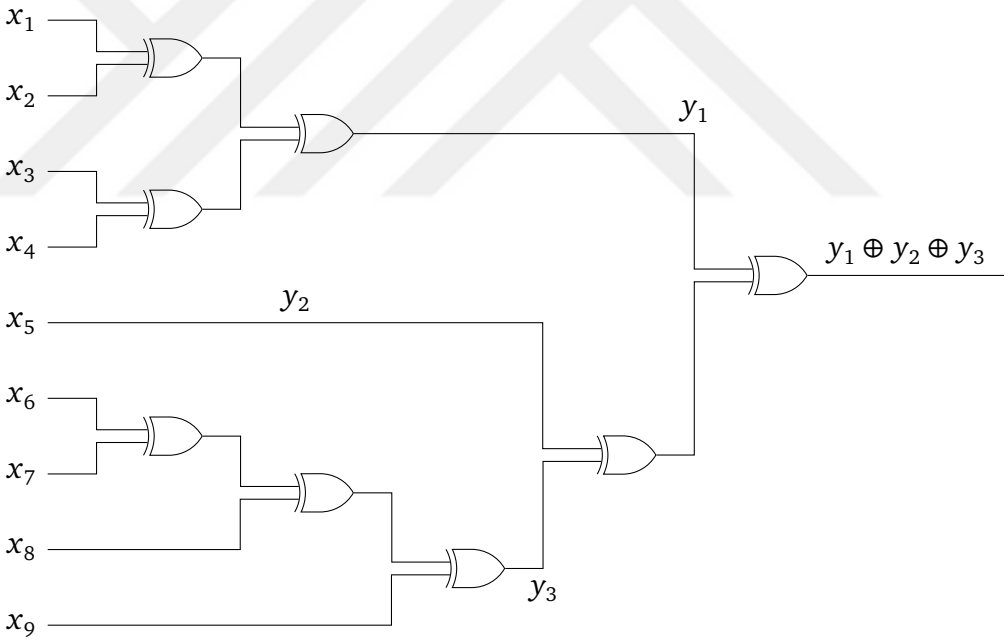
Sonuç olarak 8 XOR *işlemi ile matris uygulaması tamamlanmış olur.*

Bu noktaya kadar bahsedilen optimizasyon teknikleri, bir difüzyon matrisinin uygulanması için gereken XOR kapılarının sayısı ile ölçülen devre alanını iyileştirmek için kullanılan özelleştirilmiş araçlar olsa da bu uygulamaların diğer bir önemli ölçütü, devrenin çalışabileceği saat frekansına kısıtlama getiren gecikmedir. Bir uygulamanın gecikmesi, devre derinliği ile karakterize edilmektedir. Şekil 3.2 ve 3.3'de, $(y_1 \oplus y_2 \oplus y_3)$ işlemi için farklı derinliklere sahip devre şemaları verilmektedir (Pehlivanoğlu & Kavun, 2020).

Tanım 3.10 (Devre Derinliği)**.** Doğrusal bir katmanın uygulamasının kritik yolu, bir giriş ve çıkış arasındaki maksimum sayıda XOR kapısı içeren yol olarak tanımlanmakta ve uygulamanın derinliği, kritik yolda yer alan XOR kapılarının sayısını ifade etmektedir.



Şekil 3.2. $y_1 \oplus y_2 \oplus y_3$ İşleminin Derinlik-4 İçin Devre Şeması



Şekil 3.3. $y_1 \oplus y_2 \oplus y_3$ İşleminin Derinlik-5 İçin Devre Şeması

S. Li, Sun, C. Li, Wei ve Hu (2019) çalışmasında, Boyar-Peralta algoritmasını devre derinliği farkındalığı ile geliştirmiştir. Temel olarak, BP algoritmasını belirli bir derinlik sınırını aşmayacak sinyalleri seçmesini sağlayacak şekilde düzenlemişler, ayrıca H-Uzaklığı olarak adlandırdıkları ve devre derinliğini hesaba katan yeni bir mesafe kavramı tanımlamışlardır.

Tanım 3.11 (H-Uzaklığı)**.** S bir sinyal kümesi olsun. Herhangi bir doğrusal ifade olan f için, $\delta_H(S,f)$ ifadesi, uygulamanın derinliği H'den büyük olmayacak şekilde, S kümesinden gelen giriş sinyalleriyle f'i uygulamak için gereken minimum XOR kapıları sayısı olarak tanımlanmakta ve S'den f'ye H-Uzaklığı olarak adlandırılmaktadır (S. Li vd., 2019).

Örnek 3.17. $S = [x_1, x_2, x_3, x_4, x_5]$ *ve* $f = x_2 \oplus x_3 \oplus x_4 \oplus x_5$ *olsun. O zaman* $\delta(S,f) = \delta_2(S,f) = 3$ *olur ve* f, $x_6 = x_2 \oplus x_3$, $x_7 = x_4 \oplus x_5$ *ve* $x_8 = x_6 \oplus x_7$ *olarak uygulanabilir. Burada* x_8 *teriminin, derinliği* 2 *olan* f *değerini hesapladığı görülebilir.*

Bu tez çalışmasında, önerilen yeni teknikle tasarlanan MDS ve MDBL matrislerin optimizasyonu için, çoğu senaryoda daha verimli sonuçlar verdiği deneysel bulgularla da desteklenen BP algoritması öncelikli olarak tercih edilmiştir. BP algoritmasının makul sürede bir çözüm üretmekte zorlandığı 32×32 ve daha büyük boyutlarda ise matrisler öncelikle, Paar algoritması kullanılarak XOR maliyeti bağlamında bir ön değerlendirmeye tabi tutulmuştur. Bu ön değerlendirme sonucunda, XOR maliyeti belirli bir eşik değerin altında olan matrislere tekrar BP algoritması uygulanmıştır. Ayrıca optimizasyonlar yapılırken, devre derinliği de dikkate alınan kriterlerden biri olmuştur.

BÖLÜM 4

4×4 BOYUTUNDA TÜM İNVOLUTİF MDS MATRİSLERİN ELDE EDİLMESİNİ SAĞLAYAN YENİ BİR TEKNİK

Bu bölümde, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı 4×4 boyutunda tüm involutif ve MDS matrisleri oluşturmak için yeni bir teknik tanıtılmaktadır. Önerilen teknik, hibrit bir tasarım tekniği olup, hem arama tabanlı hem de doğrudan tasarım tabanlı tekniklerin birleşimine dayanmaktadır. Tekniğin ana fikri, tüm 4×4 boyutunda involutif ve MDS matris temsilcilerini arama yoluyla üretmek ve ardından bu temsilci matrislere 3 yeni parametre (sıfır değerine sahip olmayan) ve bu parametrelerin terslerini de uygulayarak 4×4 boyutunda involutif MDS matrislerin tümünü elde etmektir. Söz konusu parametreler, 4×4 boyutunda belirli bir involutif MDS matris temsilcisinin MDS ve involutif özelliklerini korumakta, ayrıca involutif bir matrisi tanımlamak için kullanılan tüm denklemleri sağlamaktadır. Bu durum, involutif MDS matrislerin arama uzayında ayrı sınıflar olarak oluşturulduğu anlamına gelmektedir. İnvolutif ve MDS temsilci matrisleri oluştururken, involutif olan bir Hadamard matrisin herhangi bir satırındaki veya sütunundaki elemanların XOR toplamının 1'e, ana köşegendeki elemanların XOR toplamının ise 0'a eşit olması özelliğinden yararlanılmaktadır. Bu özellik ayrıca ters köşegendeki elemanların XOR toplamının da 0 olmasını sağlamaktadır. Bu bağlamda, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı tüm temsilci involutif MDS matrisleri aramak için kullanılan R_1 matris formu aşağıdaki gibi ifade edilmektedir:

$$R_1 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{11}+r_{12}+r_{13}+1 \\ r_{21} & r_{22} & r_{12}+r_{13}+r_{21}+r_{31}+r_{32} & r_{12}+r_{13}+r_{22}+r_{31}+r_{32}+1 \\ r_{31} & r_{32} & r_{33} & r_{31}+r_{32}+r_{33}+1 \\ r_{11}+r_{21}+r_{31}+1 & r_{12}+r_{22}+r_{32}+1 & r_{12}+r_{21}+r_{31}+r_{32}+r_{33}+1 & r_{11}+r_{22}+r_{33} \end{bmatrix}.$$

Bölüm 4.1'de, tez çalışmasında kullanılan tekniğin çeşitli teoremler yardımıyla kanıtlaması yapılacaktır.

4.1 Tekniğin Matematiksel İspatı

İnvolutif ve MDS matris temsilcilerini bulmak için kullanılan R_1 matris formu, $\mathbb{F}_{2^m} - \{0\}$ cismi üzerinde 8 parametre $(r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{31}, r_{32}, r_{33})$ ile tanımlanmaktadır. Matris formu R_1 tanımlanırken, bir MDS matrisin tüm girişlerinin sıfırdan faklı olması gerektiği gerçeğinden hareketle, sonlu cisim elemanı 0 dikkate alınmamaktadır. Böylece, $\mathbb{F}_{2^m}$ sonlu cismi üzerinde tanımlı, involutif ve MDS matris temsilcilerini bulmak için arama uzayı $(2^m - 1)^8$ olarak elde edilebilir. Örnek olarak, $\mathbb{F}_{2^4}$ sonlu cismi üzerinde tanımlı, 4×4 boyutlarında involutif ve MDS matris temsilcilerinin tümünü bulmak için arama uzayı $(2^4 - 1)^8 \approx 2^{31.25}$ olarak hesaplanabilir.

Önerilen tekniğin arama tabanlı kısmını oluşturan ve temsilci matrisleri bulmak için kullanılan R_1 matris formunu tanımlamak için gereken matematiksel alt yapı Önerme 4.1'de verilmektedir.

Önerme 4.1. *Tüm $r_{ij} \neq 0$ ve $r_{ij} \in \mathbb{F}_{2^m}$ olmak üzere $R = \left[r_{ij} \right]$, 4×4 boyutunda bir kare matris olsun. Eğer R matrisi involutif bir temsilci matris ise;*

- *R matrisinin ana köşegenini oluşturan tüm elemanların toplamı 0'a eşittir. (Diğer bir deyişle $\sum_{i=1}^{4} r_{ii} = 0$'dır.)*
- *R matrisinin herhangi bir satır veya sütununu oluşturan elemanların toplamı 1'e eşittir.*

Kanıt. 4×4 boyutunda bir kare matris olan $R = \left[r_{ij} \right]$ için, tüm $r_{ij} \neq 0$ olsun ve $c_{ij} = \sum_{k=1}^{4} r_{ik} r_{kj} (i, j = 1, 2, 3, 4$ olmak üzere) R^2 matrisinin elemanlarını temsil etsin. Eğer $R^2 = I$ (4×4 boyutlarında birim matris) ise $i = j$ olduğunda $c_{ij} = 1$, aksi takdirde $c_{ij} = 0$ olur. Böylece aşağıdaki denklemler sağlanmaktadır.

$$r_{11}^2 + r_{12} r_{21} + r_{13} r_{31} + r_{14} r_{41} = 1 \tag{4.1}$$

$$r_{21} r_{12} + r_{22}^2 + r_{23} r_{32} + r_{24} r_{42} = 1 \tag{4.2}$$

$$r_{31} r_{13} + r_{32} r_{23} + r_{33}^2 + r_{34} r_{43} = 1 \tag{4.3}$$

$$r_{41} r_{14} + r_{42} r_{24} + r_{43} r_{34} + r_{44}^2 = 1 \tag{4.4}$$

$$r_{11} r_{12} + r_{12} r_{22} + r_{13} r_{32} + r_{14} r_{42} = 0 \tag{4.5}$$

$$r_{11}r_{13} + r_{12}r_{23} + r_{13}r_{33} + r_{14}r_{43} = 0 \tag{4.6}$$

$$r_{11}r_{14} + r_{12}r_{24} + r_{13}r_{34} + r_{14}r_{44} = 0 \tag{4.7}$$

$$r_{21}r_{11} + r_{22}r_{21} + r_{23}r_{31} + r_{24}r_{41} = 0 \tag{4.8}$$

$$r_{21}r_{13} + r_{22}r_{23} + r_{23}r_{33} + r_{24}r_{43} = 0 \tag{4.9}$$

$$r_{21}r_{14} + r_{22}r_{24} + r_{23}r_{34} + r_{24}r_{44} = 0 \tag{4.10}$$

$$r_{31}r_{11} + r_{32}r_{21} + r_{33}r_{31} + r_{34}r_{41} = 0 \tag{4.11}$$

$$r_{31}r_{12} + r_{32}r_{22} + r_{33}r_{32} + r_{34}r_{42} = 0 \tag{4.12}$$

$$r_{31}r_{14} + r_{32}r_{24} + r_{33}r_{34} + r_{34}r_{44} = 0 \tag{4.13}$$

$$r_{41}r_{11} + r_{42}r_{21} + r_{43}r_{31} + r_{44}r_{41} = 0 \tag{4.14}$$

$$r_{41}r_{12} + r_{42}r_{22} + r_{43}r_{32} + r_{44}r_{42} = 0 \tag{4.15}$$

$$r_{41}r_{13} + r_{42}r_{23} + r_{43}r_{33} + r_{44}r_{43} = 0 \tag{4.16}$$

Denklem (4.1), (4.2), (4.3) ve (4.4) taraf tarafa toplandığında aşağıdaki denklem elde edilebilir:

$$r_{11}^2 + r_{22}^2 + r_{33}^2 + r_{44}^2 = 0 \tag{4.17}$$

$\mathbb{F}_{2^m}$ cismi üzerinde çalışıldığından denklem (4.17) aşağıdaki gibi düzenlenebilir:

$$r_{11} + r_{22} + r_{33} + r_{44} = 0 \tag{4.18}$$

Böylelikle Önerme 4.1'in ilk ifadesi kanıtlanmış olur.

Önerme 4.1'in ikinci ifadesinin kanıtlanabilmesi için R matrisinin herhangi bir satırının elemanları toplamının 1'e eşit olduğu varsayılmaktadır. Ardından bu varsayıma

karşılık gelen denklemler aşağıdaki gibi oluşturulabilir:

$$r_{11} + r_{12} + r_{13} + r_{14} = 1 \tag{4.19}$$

$$r_{21} + r_{22} + r_{23} + r_{24} = 1 \tag{4.20}$$

$$r_{31} + r_{32} + r_{33} + r_{34} = 1 \tag{4.21}$$

$$r_{41} + r_{42} + r_{43} + r_{44} = 1 \tag{4.22}$$

Denklem (4.5), (4.6) ve (4.7) taraf tarafa toplandığında aşağıdaki denklem elde edilebilir:

$$r_{11}(r_{12}+r_{13}+r_{14})+r_{12}(r_{22}+r_{23}+r_{24})+r_{13}(r_{32}+r_{33}+r_{34})+r_{14}(r_{42}+r_{43}+r_{44}) = 0 \tag{4.23}$$

Denklem (4.19), (4.20), (4.21) ve (4.22) varsayımları da aşağıdaki gibi yazılabilir:

$$r_{11} + 1 = +r_{12} + r_{13} + r_{14} \tag{4.24}$$

$$r_{21} + 1 = r_{22} + r_{23} + r_{24} \tag{4.25}$$

$$r_{31} + 1 = r_{32} + r_{33} + r_{34} \tag{4.26}$$

$$r_{41} + 1 = r_{42} + r_{43} + r_{44} \tag{4.27}$$

Denklem (4.23)'de bulunan $(r_{12} + r_{13} + r_{14})$, $(r_{22} + r_{23} + r_{24})$, $(r_{32} + r_{33} + r_{34})$ ve $(r_{42} + r_{43} + r_{44})$ ifadelerinin yerine sırasıyla $r_{11} + 1$, $r_{21} + 1$, $r_{31} + 1$ ve $r_{41} + 1$ ifadeleri yazılırsa denklem (4.28) ve (4.29) aşağıda görüldüğü gibi elde edilebilir:

$$r_{11}(r_{11} + 1) + r_{12}(r_{21} + 1) + r_{13}(r_{31} + 1) + r_{14}(r_{41} + 1) = 0 \tag{4.28}$$

$$r_{11}^2 + r_{11} + r_{12}r_{21} + r_{12} + r_{13}r_{31} + r_{13} + r_{14}r_{41} + r_{14} = 0 \tag{4.29}$$

Denklem (4.29), denklem (4.19) kullanılarak düzenlendiğinde ise denklem (4.1)'e

aşağıda görüldüğü gibi ulaşılabilir:

$$r_{11}^2 + r_{12}r_{21} + r_{13}r_{31} + r_{14}r_{41} = 1 \tag{4.30}$$

Daha sonra, (4.8)'den (4.16)'ya kadar olan denklemler ile (4.19), (4.20), (4.21) ve (4.22) varsayımları kullanılarak denklem (4.2), (4.3) ve (4.4) denklemleri de benzer şekilde elde edilebilir.

Önerme 4.1'in herhangi bir sütun elemanlarının toplamı ile ilgili olan ikinci kısmı ispatlanırken de aşağıdaki varsayımlar yapılmaktadır:

$$r_{11} + r_{21} + r_{31} + r_{41} = 1 \tag{4.31}$$

$$r_{12} + r_{22} + r_{32} + r_{42} = 1 \tag{4.32}$$

$$r_{13} + r_{23} + r_{33} + r_{43} = 1 \tag{4.33}$$

$$r_{14} + r_{24} + r_{34} + r_{44} = 1 \tag{4.34}$$

Kolaylıkla görülebileceği gibi, (4.5)'den (4.16)'ya kadar olan denklemler ile (4.31), (4.32), (4.33) ve (4.34) varsayımları kullanılarak, denklem (4.1), (4.2), (4.3) ve (4.4) ifadelerine ulaşılabilir. Örneğin, (4.31), (4.32), (4.33) ve (4.34) varsayımları ile (4.8), (4.11), (4.14) denklemleri kullanılarak ve önermenin ilk kısmı ispatlanırken gösterilen işlemlere benzer işlemler uygulanarak denklem (4.1) elde edilebilir. Dolayısıyla, Önerme 4.1'in ikinci ifadesi de kanıtlanmış olmaktadır. ■

Açıklama 4.1. Önerme 4.1'de verilen varsayımlar tüm involutif matris temsilcilerini bulmayı garanti etmekte, ancak varsayımları karşılayan herhangi bir matrisin her zaman involutif bir matris olduğunu garanti etmemektedir.

Tekniğin doğrudan tasarım tarafında, 4×4 boyutunda involutif olan bir MDS matrisin, involutif ve MDS özelliğini koruyacak şekilde sıfır olmayan 3 adet parametre ile genişletilmesi fikri bulunmaktadır. Bu fikrin matematiksel arka planı ise Teorem 4.1'de verilmektedir.

Teorem 4.1. *$R = \left[r_{ij}\right]$, 4×4 boyutunda involutif ve MDS özelliğine sahip bir temsilci matris ve $i = 1, 2, 3$ için b_i, $\mathbb{F}_{2^m}$ cisminin sıfır olmayan bir elemanı olmak üzere, yine involutif ve MDS olan $A = \left[a_{ij}\right]$ matrisi aşağıdaki gibi tanımlanmaktadır:*

$$A = \left[a_{ij}\right] = \begin{bmatrix} r_{11} & r_{12}b_1 & r_{13}b_2 & r_{14}b_3 \\ r_{21}b_1^{-1} & r_{22} & r_{23}b_1^{-1}b_2 & r_{24}b_1^{-1}b_3 \\ r_{31}b_2^{-1} & r_{32}b_2^{-1}b_1 & r_{33} & r_{34}b_2^{-1}b_3 \\ r_{41}b_3^{-1} & r_{42}b_3^{-1}b_1 & r_{43}b_3^{-1}b_2 & r_{44} \end{bmatrix}.$$

Kanıt. R matrisi involutif olduğundan, Önerme 4.1'de verilen (4.1) ile (4.16) arasındaki tüm denklemleri sağlamaktadır. Söz konusu önermede verilen denklemlerde, r_{ij} değerleri yerine a_{ij} değerleri konulduğunda A matrisinin de tüm denklemleri sağladığı görülebilmektedir. Dolayısıyla, R matrisinin involutif özelliği korunmaktadır. Öte yandan, A matris formundaki b_i parametreleri, $\mathbb{F}_{2^m}$ cisminin sıfırdan farklı c_1, c_2, c_3, c_4 sabit elemanlarının matrisin sırasıyla birinci, ikinci, üçüncü ve dördüncü sütunlarıyla, $c_1^{-1}, c_2^{-1}, c_3^{-1}, c_4^{-1}$ sabit elemanlarının ise matrisin sırasıyla birinci, ikinci, üçüncü ve dördüncü satırlarıyla çarpımından elde edilebilmektedir. Tanım 2.35'de verilen, "$n \times n$ boyutlarında bir MDS matrisin satır veya sütunları $\mathbb{F}_{2^m}$ cisminin 0'dan farklı bir elemanı ile çarpıldığında elde edilen yeni matrisin MDS olma özelliği korunmaktadır" ifadesi uyarınca da, R matrisinin MDS özelliğinin korunduğu söylenebilmektedir. ■

4.2 Tekniğin Algoritması

Bir önceki başlıkta verilen ispat ve bilgiler ışığında önerilen teknik, iki aşama olarak aşağıdaki gibi özetlenebilir:

- Önerme 4.1 kullanılarak, R_1 matris formu yardımıyla olası tüm adaylar aranarak involutif ve MDS matris temsilcileri bulunmaktadır. Bu durum, R_1 matrisinde gösterildiği gibi arama aşamasında 16 parametre yerine 8 parametre $(r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{31}, r_{32}, r_{33})$ kullanmamızı sağlamaktadır. Böylece $n, \mathbb{F}_{2^m}$ cismi üzerinde tanımlı tüm 4×4 boyutunda matrisleri temsil etmek üzere, arama uzayı karmaşıklığını yaklaşık olarak $\sqrt{n}$ düzeyinde azaltmamıza olanak vermektedir.
- Ardından, Teorem 4.1 kullanılarak, b_i parametreleri yardımıyla 4×4 boyutunda tüm involutif MDS matrisler doğrudan tasarım ile oluşturulmaktadır.

Önerilen tekniğin ana fikri, involutif MDS matrislerin arama uzayında ayrı sınıflar olarak oluşturulmasıdır ki bu da öncelikli olarak involutif MDS matris temsilcilerine odaklanarak, tüm involutif matrisleri tüm involutif olmayan matrislerden daha kolay üretmemizi sağlamaktadır. Örnek 4.1 ve 4.2 daha önce literatürde verilen iki farklı matrisin, bu çalışmada önerilen teknik ile elde edilen bir temsilci matris sınıfına ait olduğunu göstermektedir.

Örnek 4.1. *Cauchois ve Loidreau (2019) çalışmasında verilen ve* $\mathbb{F}_{2^4}$ *cismi üzerinde tanımlı* $0x13$ *polinomunun bir kökü* α *olmak üzere* 4×4 *boyutlarında bir* $\theta-$ *dairesel involutif MDS matris olan* M_A *aşağıdaki gibi gösterilmektedir:*

$$M_A = \begin{bmatrix} \alpha & 1 & \alpha^{14} & \alpha^7 \\ \alpha^{14} & \alpha^2 & 1 & \alpha^{13} \\ \alpha^{11} & \alpha^{13} & \alpha^4 & 1 \\ 1 & \alpha^7 & \alpha^{11} & \alpha^8 \end{bmatrix}.$$

M_A *matrisi,* $b_1 = \alpha^8, b_2 = \alpha^9$ *ve* $b_3 = \alpha^{11}$ *parametrelerinin* RIM_A *involutif MDS matris temsilcisine uygulanmasıyla kolayca elde edilebilmektedir.*

$$RIM_A = \begin{bmatrix} \alpha & \alpha^7 & \alpha^5 & \alpha^{11} \\ \alpha^7 & \alpha^2 & \alpha^{14} & \alpha^{10} \\ \alpha^5 & \alpha^{14} & \alpha^4 & \alpha^{13} \\ \alpha^{11} & \alpha^{10} & \alpha^{13} & \alpha^8 \end{bmatrix}$$

Dikkat edilecek olursa RIM_A *matris temsilcisi, simetrik ve* $\theta-$ *dairesel involutif bir MDS matristir.* M_A *matrisi, matris temsilcisi* RIM_A *olan bir sınıfa ait olmaktadır.*

Örnek 4.2. *Pehlivanoğlu vd. (2018) çalışmasında verilen ve* $\mathbb{F}_{2^4}$ *cismi üzerinde tanımlı* $0x13$ *polinomunun bir kökü* α *olmak üzere* 4×4 *boyutlarında bir involutif MDS matris olan* M_B *aşağıdaki gibi gösterilmektedir:*

$$M_B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & \alpha & \alpha^2 & \alpha^5 \\ \alpha^7 & \alpha^{10} & \alpha & \alpha^5 \\ \alpha^9 & \alpha^2 & \alpha^{10} & 1 \end{bmatrix}.$$

M_B *matrisi,* $b_1 = \alpha^9, b_2 = \alpha^{13}$ *ve* $b_3 = \alpha^{12}$ *parametrelerinin* RIM_B *involutif MDS matris temsilcisine uygulanmasıyla kolayca elde edilebilmektedir.*

$$RIM_B = \begin{bmatrix} 1 & \alpha^6 & \alpha^2 & \alpha^3 \\ \alpha^9 & \alpha & \alpha^{13} & \alpha^2 \\ \alpha^5 & \alpha^{14} & \alpha & \alpha^6 \\ \alpha^6 & \alpha^5 & \alpha^9 & 1 \end{bmatrix}$$

RIM_B *matris temsilcisi bir involutif MDS matristir.* M_B *matrisi ise matris temsilcisi* RIM_B *olan bir sınıfa ait olmaktadır.*

4.3 Materyal ve Metot

Önerilen tekniğin ilk aşamasında, R_1 matris formu yardımıyla involutif MDS temsilci matrisleri oluşturmak ve bu temsilci matrislere $i = 1, 2, 3$ olmak üzere, $\mathbb{F}_{2^m}$ cisminin sıfırdan farklı herhangi bir elemanını temsil eden b_i parametrelerinin uygulanmasıyla tüm involutif ve MDS matrisleri elde etmek için MAGMA ("Magma Computational Algebra System", 2021) yazılımından faydalanılmıştır.

MAGMA; cebir, sayı teorisi ve cebirsel geometri için tasarlanmış, Avustralya Sidney Üniversitesi Cebirsel Hesaplama Grubu tarafından dağıtılan bir yazılım paketidir. Gruplar, halkalar, cisimler, cebirler, şemalar, eğriler, grafikler, kodlar, modüler aritmetik ve benzeri pek çok yapıyı tanımlamak ve bunlarla çalışmak için matematiksel bir ortam sağlar. Ayrıca cebirsel alanlardaki hesaplamalı araştırmalara yardımcı olmak için tasarlanmış bir dizi veri tabanını da destekler. Ticari bir yazılım olan MAGMA'nın kısıtlı bir ücretsiz sürümüne http://magma.maths.usyd.edu.au/calc/ adresinden ulaşılabilir. Aşağıda, MAGMA dilinin söz dizimine ait bazı örnekler verilmiştir. Tez çalışmasında kullanılan MAGMA kodunun tamamına ise çalışmanın $EK-A$ bölümünden ulaşılabilir.

Örnek 4.3. $\mathbb{F}_{2^4}/0x13$ *üzerinde tanımlı,* 4×4 *boyutunda* $A = \begin{bmatrix} x^9 & x^6 & x^5 & 1 \\ x^6 & x^9 & 1 & x^5 \\ x^5 & 1 & x^9 & x^6 \\ 1 & x^5 & x^6 & x^9 \end{bmatrix}$ *matrisi, MAGMA yazılımı ile aşağıdaki gibi ifade edilmektedir:*

```
P<z> := PolynomialRing(GF(2));
p := z^4 + z + 1;
F<x> := ext <GF(2) | p >;
A:=Matrix(F, 4, 4, [
x^9, x^6, x^5, 1,
x^6, x^9, 1, x^5,
x^5, 1, x^9, x^6,
1, x^5, x^6, x^9]);
```

Örnek 4.4. *Örnek 4.3'de verilen A matrisinin dal sayısının bulunması ve involutif özelliğinin kontrol edilmesi ise aşağıdaki kod parçacığı ile yapılmaktadır:*

```
P<z> := PolynomialRing(GF(2));
p := z^4 + z + 1;
F<x> := ext <GF(2) | p >;
A:=Matrix(F, 4, 4, [
x^9, x^6, x^5, 1,
x^6, x^9, 1, x^5,
```

```
x^5, 1, x^9, x^6,
1, x^5, x^6, x^9]);
I := IdentityMatrix(F, 4);
C := LinearCode(HorizontalJoin(I, A));
printf("Dal Sayisi: "); MinimumWeight(C);
A^2 eq I;
```

Örnek 4.5. $\mathbb{F}_2$ *cismi üzerinde tanımlı,* 8×8 *boyutunda* B *ikili matrisi aşağıdaki gibi verilsin.*

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

Bu ikili matrisin MAGMA yazılımı ile kodlanması aşağıdaki gibi yapılmaktadır:

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

Örnek 4.6. *Örnek 4.5'de verilen* B *ikili matrisinin ve transpozunun* (B^T) *dal sayısını bulmak içinse aşağıdaki koddan yararlanılmaktadır:*

```
C:=HorizontalJoin(IdentityMatrix(GF(2), 8), B);
D:=LinearCode(C);
printf("Dal Sayisi: ");
MinimumWeight(D);
C:=HorizontalJoin(IdentityMatrix(GF(2), 8), Transpose(B));
D:=LinearCode(C);
printf("Transpoz Dal Sayisi: ");
MinimumWeight(D);
```

MAGMA kullanılarak elde edilen matrislere, PAAR ve BP algoritmalarının uygulanabilmesi için aşağıda örnekleri verilen çeşitli dönüşümlerin gerçekleştirilmesi gerekmektedir. Bu sebeple tez kapsamında, Python ("Python Programming Language", 2021) programlama dili kullanılarak bir ara yazılım geliştirilmiştir. Geliştirilen bu ara yazılımda, özel matris tiplerini kullanabilmek ve matrisler üzerinde çeşitli işlemleri kolaylaştırmak için NumPy ("Fundamental Package for Scientific Computing with Python", 2021) ve SciPy ("Scientific Computing Tools for Python", 2021) isimli Python kitaplıklarından faydalanılmıştır. Ayrıca sonlu cisimler üzerinde hızlıca çalışabilmek adına da yine Python tabanlı açık kaynak bir sistem olan SageMath ("Mathematics Software System", 2021) aracından yararlanılmıştır.

Öncelikle, MAGMA koduyla elde edilen matrislerin elemanları onaltılık gösterime dönüştürülür. Aşağıdaki kod parçacığı, sadece $\mathbb{F}_{2^4}$ cisminde tanımlı $0x13$ polinomunun tüm elemanlarının onaltılık karşılıklarının bir döngü yardımıyla elde edilmesini örneklendirse de çalışma boyunca kullanılan yazılımda cisim ve polinom isteğe bağlı olarak seçilebilmektedir.

```
F.<b> = GF(2)[]
S.<x> = GF(2**4, modulus = b**4 + b + 1)

for k in range(0, 16):
    print(hex((x**k).integer_representation()))
```

Ardından, *sonlu_cisimden_ikiliye* fonksiyonu bir sonlu cisim elemanını, uygun çarpma işlemini gerçekleştirerek ikili matrise dönüştürürken, *sonlu_cisim_matrisinden_ikiliye* isimli fonksiyon ise sonlu cisim üzerindeki bir matrisi, uygun çarpma işlemini gerçekleştirerek ikili matrise dönüştürmektedir.

```
def sonlu_cisimden_ikiliye(sonlu_cisim_elemani):
    F = sonlu_cisim_elemani.parent()
    C = companion_matrix(F.modulus(), format='right')
    B = Integer(sonlu_cisim_elemani.integer_representation()).digits(
   base=2, padto=F.degree())
    X = 0
    I = identity_matrix(GF(2), F.degree())
    for i in B:
        if i == 1:
            X = X + I
        I = I * C
    return X
```

```
def sonlu_cisim_matrisinden_ikiliye(sonlu_cisim_matrisi):
    B = []
    for i in sonlu_cisim_matrisi:
```

```
        for j in i:
            B.append(sonlu_cisimden_ikiliye(j))
    return block_matrix(sonlu_cisim_matrisi.nrows(),
    sonlu_cisim_matrisi.ncols(), B)
```

ham_xor_sayisi fonksiyonu ise herhangi bir matrisin ham (naive) XOR maliyetinin hesaplanması amacıyla kullanılmaktadır.

```
def ham_xor_sayisi(matrix):
    if matrix[0][0].parent() != GF(2):
        matrix = sonlu_cisim_matrisinden_ikiliye(matrix)
    satir, sutun = matrix.dimensions()
    return matrix.density() * satir * sutun - satir
```

sutun_notasyonu fonksiyonu, matrisin Paar algoritmasında kullanılabilmesi için Bölüm 3'de anlatılan gerekli sütun dönüşümlerinin yapılmasını sağlamaktadır.

```
def sutun_notasyonu(matrix):
    if matrix[0][0].parent() != GF(2):
        matrix = sonlu_cisim_matrisinden_ikiliye(matrix)

    sutunlar = []
    for sutun in matrix.sutunlar():
        sutunlar.append(reduce(lambda x, y: Integer(x)*2 + Integer(y),
    sutun))
    return sutunlar
```

Paar algoritmasının uygulanmasına olanak veren kod parçacığı ise aşağıda verilmektedir.

```
def paar_algoritmasi(matrix):
    if matrix[0][0].parent() != GF(2):
        matrix = sonlu_cisim_matrisinden_ikiliye(matrix)

    boyutlar = matrix.dimensions()[0]
    xor_sayisi = ham_xor_sayisi(matrix)
    matris_sutunlari = sutun_notasyonu(matrix)

    while True:
        maksimum_hamming = 0

        for i, stn_i in enumerate(matris_sutunlari):
            for j, stn_j in enumerate(matris_sutunlari[i+1:], i+1):
                hamming_agirligi = Integer(stn_i & stn_j).popcount()
                if hamming_agirligi > maksimum_hamming:
                    maksimum_hamming = hamming_agirligi
                    i_maksimum = i
```

```
                    j_maksimum = j

        if maksimum_hamming > 1:
            yeni_sutun = matris_sutunlari[i_maksimum] &
    matris_sutunlari[j_maksimum]
            matris_sutunlari.append(yeni_sutun)
            matris_sutunlari[i_maksimum] = (yeni_sutun^^((1 <<
    boyutlar)-1)) & matris_sutunlari[i_maksimum]
            matris_sutunlari[j_maksimum] = (yeni_sutun^^((1 <<
    boyutlar)-1)) & matris_sutunlari[j_maksimum]
            xor_sayisi -= maksimum_hamming - 1

        if maksimum_hamming <= 1:
            break

    return xor_sayisi
```

Örnek 4.7. *$\mathbb{F}_{2^4}$ cismi üzerinde tanımlı $0x13$ polinomunun bir kökü α olmak üzere 4×4 boyutundaki A matrisi aşağıdaki gibi verilsin.*

$$A = \begin{bmatrix} 1 & \alpha^8 & \alpha^{14} & \alpha^{13} \\ 1 & 1 & \alpha^5 & \alpha^{14} \\ \alpha^{14} & \alpha^{13} & 1 & \alpha^8 \\ \alpha^5 & \alpha^{14} & 1 & 1 \end{bmatrix}$$

A matrisinin onaltılık gösterimine ve matrisin tanımlı olduğu cisim ile polinoma ait kod örneği aşağıdaki gibi verilebilir.

```
F_0x13 = GF(2**4, name="x", modulus=PolynomialRing(GF(2), name="x")("x
    **4 + x + 1"))
A = matrix(F_0x13, map(lambda x: map(F_0x13.fetch_int, x), [[0x1, 0x5,
    0x9, 0xd], [0x1, 0x1, 0x6, 0x9], [0x9, 0xd, 0x1, 0x5], [0x6, 0x9, 0
    x1, 0x1]]))
ham_xor_sayisi(A)
paar_algoritmasi(A)
```

Kodun son iki satırında çağırılan ham_xor_sayisi ve paar fonksiyonları sonucunda elde edilen çıktı ise aşağıdaki gibi olmaktadır. Söz konusu iki fonksiyona ait kod parçacıklarında görülen diğer fonksiyon çağrılarına, kod bloğunu uzatmamak adına bu örnekte yer verilmemiştir.

```
> Ham XOR Sayisi: 80
> Paar Sonrasi XOR Sayisi: 52
```

matrisi_BP_formatina_cevir fonksiyonu ise ikili bir matrisi Boyar-Peralta formatına dönüştürmektedir.

```
def matrisi_BP_formatina_cevir(matrix):
    if matrix[0][0].parent() != GF(2):
        matrix = sonlu_cisim_matrisinden_ikiliye(matrix)

    matrix = matrix(GF(2), matrix.dimensions()[0], matrix.dimensions()
    [1], matrix.rows())
    cikti = "1\n%d %d\n" % matrix.dimensions()
    for row in matrix.str().split("\n"):
        cikti += row[1:-1] + "\n"
        print(cikti)
    return cikti
```

Bu fonksiyon kullanılarak Örnek 4.7'de verilen A matrisinin BP formatı aşağıda görüldüğü gibi oluşturulmaktadır. Elde edilen formatın ilk satırı matris adedini, ikinci satırı matrisin boyutlarını belirtmektedir. Diğer satırlar ise matrisin ikili gösterimini içermektedir.

```
1
16 16
1 0 0 0 1 0 1 0 1 1 0 0 1 1 1 0
0 1 0 0 0 1 1 1 0 0 1 0 0 0 0 1
0 0 1 0 1 0 1 1 0 0 0 1 1 0 0 0
0 0 0 1 0 1 0 1 1 0 0 0 1 1 0 0
1 0 0 0 1 0 0 0 0 0 1 1 1 1 0 0
0 1 0 0 0 1 0 0 1 0 1 0 0 0 1 0
0 0 1 0 0 0 1 0 1 1 0 1 0 0 0 1
0 0 0 1 0 0 0 1 0 1 1 0 1 0 0 0
1 1 0 0 1 1 1 0 1 0 0 0 1 0 1 0
0 0 1 0 0 0 0 1 0 1 0 0 0 1 1 1
0 0 0 1 1 0 0 0 0 0 1 0 1 0 1 1
1 0 0 0 1 1 0 0 0 0 0 1 0 1 0 1
0 0 1 1 1 1 0 0 1 0 0 0 1 0 0 0
1 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0
1 1 0 1 0 0 0 1 0 0 1 0 0 0 1 0
0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 1
```

Yukarıdaki gibi Boyar-Peralta gösterimleri oluşturulan matrisler, son adımda, Kranz, Leander, Stoffelen ve Wiemer (2017) çalışmasında verilen BP algoritması kullanılarak optimize edilmektedir. Öncelikli olarak bu algoritmanın kullanılma sebebi, algoritmanın derinlik kriterini göz ardı ederek mümkün olan en düşük XOR maliyetine ulaşmayı hedeflemesidir. Ardından, optimizasyon sonuçları iyi olan matrislere bu kez, S. Li vd. (2019) çalışmasında verilen ve derinlik kriterini de dikkate alan BP algoritması

kullanılarak tekrar optimizasyon işlemi yapılmakta ve sonuçlar elde edilmektedir.

Beşinci ve altıncı bölüm sırasıyla, önerilen teknik ile $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$ ve $\mathbb{F}_{2^8}$ cisimleri üzerinde elde edilen 4×4 boyutlu MDS matrislere ait sonuçlar ile $\mathbb{F}_2$ cismi üzerinde elde edilen 12×12 ve 16×16 boyutlarında MDBL matrislere ait sonuçları içermektedir.

BÖLÜM 5

ÖNERİLEN TEKNİKLE ELDE EDİLEN MDS MATRİSLERE AİT DENEYSEL BULGULAR

Bu bölümde, $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cismi üzerinde tanımlı, 4×4 boyutunda tüm involutif MDS matris temsilcilerinin arama yöntemiyle elde edilmesi ve bu matris temsilcilerini kullanarak aynı sonlu cisimler üzerindeki 4×4 boyutlarında involutif MDS matrislerinin tümünün tasarlanmasına olanak sağlayan önerilen tekniğe ait deneysel bulgular verilecektir. Ayrıca, önerilen tekniğin $\mathbb{F}_{2^8}$ sonlu cismi üzerinde uygulanmasıyla elde edilen çeşitli verilerden ve literatürde bilinen GHadamard, Dairesel ve Toeplitz gibi çeşitli özel matris formlarına ait sonuçlardan da bahsedilecektir. Buna ek olarak önerilen teknik, her ne kadar belirtilen sonlu cisim alanlarındaki tüm involutif MDS matrisleri tasarlamak amacıyla geliştirilmiş olsa da hem non-involutif MDS hem de diyagonal veya antidiyagonal köşegenlerine göre simetrik özellik gösteren non-involutif MDS matrislerin tasarlanmasında da kullanılabildiğinden bu matrislere ait deneysel verilere de yer verilmektedir.

Çalışmanın optimizasyon tarafında, en düşük XOR sayısına ve minimum derinliğe sahip yeni involutif MDS matrislerin aranması temel hedef olarak belirlenerek, Yer değiştirme-Permütasyon Ağı (SPN) şifrelerinde kullanılabilecek, 4×4 boyutlarında involutif ve MDS matrisler ile tasarlanan doğrusal dönüşüm katmanlarına odaklanılmaktadır. İkili cisim uzantıları üzerinde tanımlı, 4×4 boyutlarında involutif ve MDS matrislerin tümünü oluşturmak için sunduğumuz yeni melez teknik yardımıyla önce düşük ham XOR sayılarına sahip hafif sıklet involutif MDS matrisler oluşturulmaktadır. Ardından global optimizasyon için $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı, 4×4 boyutlarında yeni involutif MDS matrislerden dönüştürülmüş ikili matrisler üzerinde BP algoritması çalıştırılmakta ve sonuçlar elde edilmektedir.

5.1 Çeşitli Özel Matris Tiplerine Ait Deneysel Bulgular

GHadamard, Dairesel ve Toeplitz gibi literatürde sık kullanılan özel matris formlarına $i = 1, 2, 3$ olmak üzere, $\mathbb{F}_{2^m}$ cisminin sıfırdan farklı herhangi bir elemanını temsil eden b_i parametrelerinin uygulanmasıyla, belirli bir eşik ham XOR değerinin altında elde edilen ve global optimizasyonları yapılan matrisler aşağıda incelenmektedir.

5.1.1 GHadamard Matris Tipine Ait Deneysel Bulgular

4×4 boyutlarında bir Hadamard matris olan $H = had(a_0, a_1, a_2, a_3)$ aşağıdaki gibi tanımlanmaktadır:

$$H = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_1 & a_0 & a_3 & a_2 \\ a_2 & a_3 & a_0 & a_1 \\ a_3 & a_2 & a_1 & a_0 \end{bmatrix}.$$

Teorem 4.1 uyarınca H matrisi b_1, b_2, b_3 ve bunların tersleri olacak şekilde 3 parametre kullanılarak Genel Hadamard (GHadamard) olarak adlandırılan matris formuna dönüştürülmekte ve $GH = Ghad(a_0, a1; b_1, a_2; b_2, a_3; b_3)$ notasyonu ile gösterilip aşağıdaki gibi tanımlanmaktadır:

$$GH = \begin{bmatrix} a_0 & a_1 b_1 & a_2 b_2 & a_3 b_3 \\ a_1 b_1^{-1} & a_0 & a_3 b_1^{-1} b_2 & a_2 b_1^{-1} b_3 \\ a_2 b_2^{-1} & a_3 b_2^{-1} b_1 & a_0 & a_1 b_2^{-1} b_3 \\ a_3 b_3^{-1} & a_2 b_3^{-1} b_1 & a_1 b_3^{-1} b_2 & a_0 \end{bmatrix}.$$

Örnek 5.1. *$\mathbb{F}_{2^4}$ cismi üzerinde tanımlı $0x13$ polinomunun bir kökü olan α kullanılarak üretilen 4×4 boyutlarında involutif MDS Hadamard matris $H_1 = had(1, \alpha, \alpha^8, \alpha^{10})$ aşağıdaki gibidir:*

$$H_1 = \begin{bmatrix} 1 & \alpha & \alpha^8 & \alpha^{10} \\ \alpha & 1 & \alpha^{10} & \alpha^8 \\ \alpha^8 & \alpha^{10} & 1 & \alpha \\ \alpha^{10} & \alpha^8 & \alpha & 1 \end{bmatrix}.$$

$b_1 = \alpha^2, b_2 = \alpha^7, b_3 = \alpha^8$ parametreleri kullanılarak oluşturulan ve H_1 matrisine karşılık

gelen $GH_1 = Ghad(1, \alpha; \alpha^2, \alpha^8; \alpha^7, \alpha^{10}; \alpha^8)$ *matrisi ise aşağıdaki gibidir:*

$$GH_1 = \begin{bmatrix} 1 & \alpha^3 & 1 & \alpha^3 \\ \alpha^{14} & 1 & 1 & \alpha^{14} \\ \alpha & \alpha^5 & 1 & \alpha^2 \\ \alpha^2 & \alpha^2 & 1 & 1 \end{bmatrix}.$$

İnvolutif ve MDS olan GH_1 *matrisinin ham XOR sayısı* 68*, BP optimizasyonu sonrası XOR sayısı ise* 39 *olarak elde edilmiştir.*

Örnek 5.2. $\mathbb{F}_{2^4}$ *cismi üzerinde tanımlı* $0x13$ *polinomunun bir kökü olan* α *kullanılarak üretilen* 4×4 *boyutlarında involutif MDS Hadamard matris* $H_2 = had(1, \alpha^2, \alpha^5, \alpha^{14})$ *aşağıdaki gibidir:*

$$H_2 = \begin{bmatrix} 1 & \alpha^2 & \alpha^5 & \alpha^{14} \\ \alpha^2 & 1 & \alpha^{14} & \alpha^5 \\ \alpha^5 & \alpha^{14} & 1 & \alpha^2 \\ \alpha^{14} & \alpha^5 & \alpha^2 & 1 \end{bmatrix}.$$

$b_1 = \alpha^2, b_2 = \alpha^2, b_3 = 1$ *parametreleri kullanılarak oluşturulan ve* H_2 *matrisine karşılık gelen* $GH_2 = Ghad(1, \alpha^2; \alpha^2, \alpha^5; \alpha^2, \alpha^{14}; 1)$ *matrisi ise aşağıdaki gibidir:*

$$GH_2 = \begin{bmatrix} 1 & \alpha^4 & \alpha^7 & \alpha^{14} \\ 1 & 1 & \alpha^{14} & \alpha^3 \\ \alpha^3 & \alpha^{14} & 1 & 1 \\ \alpha^{14} & \alpha^7 & \alpha^4 & 1 \end{bmatrix}.$$

Non-involutif ve MDS olan GH_2 *matrisinin ham XOR sayısı* 80*, BP optimizasyonu sonrası XOR sayısı ise* 38 *olarak elde edilmiştir.*

Önerilen teknik kullanılarak $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı hem involutif hem de involutif olmayan GHadamard matrisler, 80 *XOR* ham değeri eşik olacak şekilde MAGMA yazılımı ile oluşturulmuş ve BP optimizasyonları yapılmıştır. Elde edilen GHadamard matrislerin BP optimizasyonları sonucunda ulaşılan en düşük *XOR* maliyetine sahip matris sayıları Çizelge 5.1'de özetlenmektedir.

Çizelge 5.1. GHadamard Matris Tipi Kullanılarak Elde Edilen Deneysel Bulgular

			BP Sonuçları	
Matris Tipi	**Ham XOR Aralığı**	**Matris Sayısı**	**39 XOR**	**38 XOR**
İnvolutif	$58-70$	576	24	0
	$71-75$	1296	6	0
	$76-78$	2136	12	0
	$79-80$	3240	0	0
TOPLAM		**7248**	**42**	**0**
Non-involutif	$58-70$	6192	0	0
	$71-75$	20712	6	0
	$76-78$	33527	36	0
	$79-80$	40944	2	72
TOPLAM		**101375**	**44**	**72**
GENEL TOPLAM		**108623**	**86**	**72**

5.1.2 Dairesel Matris Tipine Ait Deneysel Bulgular

4×4 boyutlarında bir Dairesel (Circulant) matris olan $C = circ(a_0, a_1, a_2, a_3)$ aşağıdaki gibi tanımlanmaktadır:

$$C = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_3 & a_0 & a_1 & a_2 \\ a_2 & a_3 & a_0 & a_1 \\ a_1 & a_2 & a_3 & a_0 \end{bmatrix}.$$

Teorem 4.1 uyarınca C matrisi b_1, b_2, b_3 ve bunların tersleri olacak şekilde 3 parametre daha kullanılarak $CP = (a_0, a1; b_1, a_2; b_2, a_3; b_3)$ notasyonu ile gösterilip aşağıdaki gibi tanımlanmaktadır:

$$CP = \begin{bmatrix} a_0 & a_1 b_1 & a_2 b_2 & a_3 b_3 \\ a_3 b_1^{-1} & a_0 & a_1 b_1^{-1} b_2 & a_2 b_1^{-1} b_3 \\ a_2 b_2^{-1} & a_3 b_2^{-1} b_1 & a_0 & a_1 b_2^{-1} b_3 \\ a_1 b_3^{-1} & a_2 b_3^{-1} b_1 & a_3 b_3^{-1} b_2 & a_0 \end{bmatrix}.$$

Örnek 5.3. $\mathbb{F}_{2^4}$ *cismi üzerinde tanımlı* $0x13$ *polinomunun bir kökü olan* α *kullanılarak üretilen* 4×4 *boyutlarında MDS Dairesel matris* $C_1 = circ(1, \alpha^{11}, \alpha, \alpha^9)$ *aşağıdaki gibidir:*

$$C_1 = \begin{bmatrix} 1 & \alpha^{11} & \alpha & \alpha^9 \\ \alpha^9 & 1 & \alpha^{11} & \alpha \\ \alpha & \alpha^9 & 1 & \alpha^{11} \\ \alpha^{11} & \alpha & \alpha^9 & 1 \end{bmatrix}.$$

$b_1 = \alpha^7, b_2 = 1, b_3 = \alpha^8$ parametreleri kullanılarak oluşturulan ve C_1 matrisine karşılık gelen $CP_1 = (1, \alpha^{11}; \alpha^7, \alpha; 1, \alpha^9; \alpha^8)$ matrisi ise aşağıdaki gibidir:

$$CP_1 = \begin{bmatrix} 1 & \alpha^3 & \alpha & \alpha^2 \\ \alpha^2 & 1 & \alpha^4 & \alpha^2 \\ \alpha & \alpha & 1 & \alpha^4 \\ \alpha^3 & 1 & \alpha & 1 \end{bmatrix}.$$

Non-involutif ve MDS olan CP_1 matrisinin ham XOR sayısı 74, BP optimizasyonu sonrası XOR sayısı ise 38 olarak elde edilmiştir.

Önerilen teknik kullanılarak $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı involutif olmayan Dairesel matrisler, 80 *XOR* ham değeri eşik olacak şekilde MAGMA yazılımı ile oluşturulmuş ve BP optimizasyonları yapılmıştır. Elde edilen Dairesel matrislerin BP optimizasyonları sonucunda ulaşılan en düşük *XOR* maliyetine sahip matris sayıları Çizelge 5.2'de verilmektedir.

Çizelge 5.2. Dairesel Matris Tipi Kullanılarak Elde Edilen Deneysel Bulgular

			BP Sonuçları	
Matris Tipi	**Ham XOR Aralığı**	**Matris Sayısı**	**39 XOR**	**38 XOR**
Non-involutif	XOR ≤ 64	824	2	0
	$65-69$	3808	10	0
	$70-79$	59936	32	2
	80	18352	0	0
GENEL TOPLAM		**82920**	**44**	**2**

5.1.3 Toeplitz Matris Tipine Ait Deneysel Bulgular

4×4 boyutlarında bir Toeplitz matris olan $T = toep(a_0, a_1, a_2, a_3, a_4, a_5, a_6)$ aşağıdaki gibi tanımlanmaktadır:

$$T = \begin{bmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_0 & a_1 & a_2 \\ a_5 & a_4 & a_0 & a_1 \\ a_6 & a_5 & a_3 & a_0 \end{bmatrix}.$$

Teorem 4.1 uyarınca T matrisi b_1, b_2, b_3 ve bunların tersleri olacak şekilde 3 parametre daha kullanılarak $TP = (a_0, a1; b_1, a_2; b_2, a_3; b_3, a_4, a_5, a_6)$ notasyonu ile gösterilip aşağıdaki gibi tanımlanmaktadır:

$$TP = \begin{bmatrix} a_0 & a_1 b_1 & a_2 b_2 & a_3 b_3 \\ a_4 b_1^{-1} & a_0 & a_1 b_1^{-1} b_2 & a_2 b_1^{-1} b_3 \\ a_5 b_2^{-1} & a_4 b_2^{-1} b_1 & a_0 & a_1 b_2^{-1} b_3 \\ a_6 b_3^{-1} & a_5 b_3^{-1} b_1 & a_4 b_3^{-1} b_2 & a_0 \end{bmatrix}.$$

Örnek 5.4. $\mathbb{F}_{2^4}$ *cismi üzerinde tanımlı* $0x13$ *polinomunun bir kökü olan* α *kullanılarak üretilen* 4×4 *boyutlarında MDS Toeplitz matris* $T_1 = toep(1, 1, \alpha^5, \alpha^4, \alpha^{13}, \alpha^{12}, \alpha^9)$ *aşağıdaki gibidir:*

$$T_1 = \begin{bmatrix} 1 & 1 & \alpha^5 & \alpha^4 \\ \alpha^{13} & 1 & 1 & \alpha^5 \\ \alpha^{12} & \alpha^{13} & 1 & 1 \\ \alpha^9 & \alpha^{12} & \alpha^{13} & 1 \end{bmatrix}.$$

$b_1 = \alpha^{13}, b_2 = \alpha^{12}, b_3 = \alpha^{10}$ *parametreleri kullanılarak oluşturulan ve* T_1 *matrisine karşılık gelen* $TP_1 = (1, 1; \alpha^{13}, \alpha^5; \alpha^{12}, \alpha^4; \alpha^{10}, \alpha^{13}, \alpha^{12}, \alpha^9)$ *matrisi ise aşağıdaki gibidir:*

$$TP_1 = \begin{bmatrix} 1 & \alpha^{13} & \alpha^2 & \alpha^{14} \\ 1 & 1 & \alpha^{14} & \alpha^2 \\ 1 & \alpha^{14} & 1 & \alpha^{13} \\ \alpha^{14} & 1 & 1 & 1 \end{bmatrix}.$$

Non-involutif ve MDS olan TP_1 *matrisinin ham XOR sayısı* 62*, BP optimizasyonu sonrası XOR sayısı ise* 38 *olarak elde edilmiştir.*

Önerilen teknik kullanılarak $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı involutif olmayan ve simetrik Toeplitz matrisler, eşik XOR değerleri seçilerek MAGMA yazılımı ile

oluşturulmuş ve BP optimizasyonları yapılmıştır. Ayrıca BP optimizasyonu ile 38 XOR değerine inen matris formuna tekrar b_i parametreleri uygulanarak yeni matrisler türetilmiş ve bir kez daha optimize edilmişlerdir. Elde edilen Toeplitz matrislerin BP optimizasyonları sonucunda ulaşılan en düşük XOR maliyetine sahip matris sayıları Çizelge 5.3'de gösterilmektedir.

Çizelge 5.3. Toeplitz Matris Tipi Kullanılarak Elde Edilen Deneysel Bulgular

			BP Sonuçları	
Matris Tipi	**Ham XOR Aralığı**	**Matris Sayısı**	**39 XOR**	**38 XOR**
Non-involutif	XOR ≤ 62	10320	60	30
	$63-64$	33315	135	60
Simetrik	$66-74$	348	4	4
38XOR'dan Türetilenler	XOR ≤ 66	11232	24	12
	$67-70$	29416	28	0
	$71-74$	46712	17	0
	$75-77$	30384	0	0
	$78-80$	16480	12	0
GENEL TOPLAM		**178207**	**280**	**106**

$\mathbb{F}_{2^4}$ sonlu cismi üzerinde tanımlı çeşitli özel matris tiplerinin kullanılmasıyla elde edilen, 4×4 boyutundaki MDS matrislerin literatürde bulunan diğer matrislerle karşılaştırılması ise Çizelge 5.4'de verilmektedir.

Çizelge 5.4. Çeşitli Matris Tiplerine Ait Deneysel Bulguların Literatür ile Karşılaştırılması

Matris Tipi	**BP Optimizasyonu Sonrası XOR Sayıları**	**Referans**
İnvolutif Hadamard	48	(Sim vd., 2015)
Non-involutif Hadamard	48	(Sim vd., 2015)
İnvolutif	47	(Jean vd., 2017)
Dairesel	44	(Liu & Sim, 2016)
Toeplitz	43	(Sarkar & Syed, 2016)
Dairesel	42	(Y. Li & Wang, 2016)
İnvolutif	42	(Sarkar & Syed, 2016)
İnvolutif GHadamard	**39**	**Örnek 5.1**
Non-involutif GHadamard	**38**	**Örnek 5.2**
CP	**38**	**Örnek 5.3**
TP	**38**	**Örnek 5.4**

5.2 $\mathbb{F}_{2^3}$ Cismine Ait Deneysel Bulgular

$\mathbb{F}_{2^3}$ sonlu cismi üzerindeki tüm involutif MDS matris temsilcilerini arayarak aynı cisim üzerinde tanımlı, 4×4 boyutlarında involutif ve MDS matrislerin tamamının üretilmesi sonucu elde edilen optimizasyon sonuçlarına ait veriler bu kısımda incelenecektir. Deneysel sonuçlar, $\mathbb{F}_{2^3}$ sonlu cismi üzerinde tanımlı $0xb$ ve $0xd$ polinomları için, 48'er adet involutif ve MDS matris temsilcisi olduğunu göstermektedir. b_i parametrelerinin involutif MDS matris temsilcilerine uygulanmasının ardından da toplam $48 \cdot (2^3 - 1)^3 = 16.464$ adet, 4×4 boyutlarında involutif ve MDS matris elde edilmektedir. Aynı işlemler, Hadamard matris formu kullanılarak yapıldığında yalnızca 24 adet involutif MDS matris temsilcisi elde edilebilmekte, bu matris temsilcilerine Pehlivanoğlu vd. (2018) çalışmasında verilen GHadamard matris formu uygulandığında da toplam $24 \cdot (2^3 - 1)^3 = 8.232$ adet, 4×4 boyutlu involutif ve MDS matris oluşturulabilmektedir.

Örnek 5.5'de önerilen teknik ile elde edilen ve literatürde bilinen matrislerden daha üstün olan, $\mathbb{F}_{2^3}/0xb$ üzerinde tanımlı yeni bir 4×4 boyutlarında involutif MDS matris verilmektedir. Devre derinliği 6 olan söz konusu matris, sadece 28 XOR işlemi ile uygulanabilmektedir.

Örnek 5.5. *$\mathbb{F}_{2^3}$ cismi üzerinde tanımlı $0xb$ polinomunun bir kökü α olmak üzere 4×4 boyutlarında bir involutif MDS matris olan M_1, aşağıdaki gibi gösterilmektedir:*

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \alpha^2 & \alpha^6 & 1 & \alpha^5 \\ \alpha & \alpha^6 & \alpha^3 & 1 \\ \alpha^4 & 1 & \alpha^3 & \alpha^5 \end{bmatrix}.$$

M_1 matrisi, $b_1 = \alpha^6, b_2 = \alpha^3$ ve $b_3 = \alpha^5$ parametrelerinin aşağıda verilen RIM_1 involutif MDS matris temsilcisine uygulanmasıyla elde edilmektedir:

$$RIM_1 = \begin{bmatrix} 1 & \alpha & \alpha^4 & \alpha^2 \\ \alpha & \alpha^6 & \alpha^3 & \alpha^6 \\ \alpha^4 & \alpha^3 & \alpha^3 & \alpha^5 \\ \alpha^2 & \alpha^6 & \alpha^5 & \alpha^5 \end{bmatrix}.$$

Bir involutif MDS matris olan M_1 matrisinin ham XOR sayısı, $(23+4 \cdot 3 \cdot 3) = 59$ XOR'dur. Matrise BP optimizasyonunun uygulanmasının ardından derinlik 3, derinlik 4, derinlik 5 ve derinlik 6 değerleri için sırasıyla 35 XOR, 30 XOR, 29 XOR ve 28 XOR maliyetlerine ulaşılmıştır.

Önerilen teknik kullanılarak $\mathbb{F}_{2^3}/0xb$ üzerinde tanımlı involutif, non-involutif ve antidiyagonal köşegenine göre simetrik olan matrislerin optimizasyon sonuçlarının tamamı Çizelge 5.5'de verilmektedir.

Çizelge 5.5. $\mathbb{F}_{2^3}$ Cismi Üzerinde Tanımlı $0xb$ Polinomuna Ait Deneysel Bulgular

Matris Tipi		Matris Sayısı	BP Sonuçları		
			30 XOR	**29 XOR**	**28 XOR**
İnvolutif		16464	962	122	24
Non-involutif	1. Grup	16464	434	43	0
	2. Grup	16463	359	24	0
	3. Grup	16464	541	77	0
	4. Grup	16464	404	69	0
	5. Grup	16464	327	0	0
	6. Grup	16464	583	32	0
TOPLAM		**115247**	**3610**	**367**	**24**
Antidiyagonal Simetrik	İlk Eleman 1	16170	740	76	0
	İlk Eleman α	16170	585	116	0
	İlk Eleman α^2	16170	483	45	0
	İlk Eleman α^3	16464	329	44	0
	İlk Eleman α^4	16170	301	29	0
	İlk Eleman α^5	16464	365	41	0
	İlk Eleman α^6	16464	583	79	0
TOPLAM		**114072**	**3386**	**430**	**0**
GENEL TOPLAM		**229319**	**6996**	**797**	**24**

Örnek 5.6'da ise Örnek 5.5'e benzer şekilde önerilen teknik ile elde edilen ve literatürde bilinen matrislerden daha üstün olan, $\mathbb{F}_{2^3}/0xd$ üzerinde tanımlı yeni bir 4×4 boyutlarında involutif MDS matris verilmektedir. M_1 matrisinde olduğu gibi bu matrisin de devre derinliği 6 olup, 28 XOR işlemi ile uygulanabilmektedir.

Örnek 5.6. *$\mathbb{F}_{2^3}$ cismi üzerinde tanımlı $0xd$ polinomunun bir kökü α olmak üzere 4×4 boyutlarında bir involutif MDS matris olan M_2, aşağıdaki gibi gösterilmektedir:*

$$M_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ \alpha^3 & \alpha^2 & 1 & \alpha^4 \\ \alpha^5 & \alpha^2 & \alpha & 1 \\ \alpha^6 & 1 & \alpha & \alpha^4 \end{bmatrix}.$$

M_2 matrisi, $b_1 = \alpha^2, b_2 = \alpha$ ve $b_3 = \alpha^4$ parametrelerinin aşağıda verilen RIM_2 involutif MDS matris temsilcisine uygulanmasıyla elde edilmektedir:

$$RIM_2 = \begin{bmatrix} 1 & \alpha^5 & \alpha^6 & \alpha^3 \\ \alpha^5 & \alpha^2 & \alpha & \alpha^2 \\ \alpha^6 & \alpha & \alpha & \alpha^4 \\ \alpha^3 & \alpha^2 & \alpha^4 & \alpha^4 \end{bmatrix}.$$

Bir involutif MDS matris olan M_2 matrisinin ham XOR sayısı, $(23+4\cdot3\cdot3) = 59$ XOR'dur. Matrise BP optimizasyonunun uygulanmasının ardından derinlik 3, derinlik 4, derinlik 5 ve derinlik 6 değerleri için sırasıyla 35 XOR, 30 XOR, 29 XOR ve 28 XOR maliyetlerine ulaşılmıştır.

Önerilen teknik kullanılarak $\mathbb{F}_{2^3}/0xd$ üzerinde tanımlı involutif, non-involutif ve antidiyagonal köşegenine göre simetrik olan matrislerin optimizasyon sonuçlarının tamamı Çizelge 5.6'da verilmektedir.

Çizelge 5.6. $\mathbb{F}_{2^3}$ Cismi Üzerinde Tanımlı $0xd$ Polinomuna Ait Deneysel Bulgular

			BP Sonuçları		
Matris Tipi		**Matris Sayısı**	**30 XOR**	**29 XOR**	**28 XOR**
İnvolutif		16464	950	128	24
Non-involutif	1. Grup	16464	532	32	0
	2. Grup	16464	383	0	0
	3. Grup	16464	477	37	0
	4. Grup	16464	539	65	0
	5. Grup	16464	351	24	0
	6. Grup	16464	488	43	0
TOPLAM		**115248**	**3720**	**329**	**24**
Antidiyagonal Simetrik	İlk Eleman 1	16170	812	74	0
	İlk Eleman α	16464	640	82	0
	İlk Eleman α^2	16464	398	26	0
	İlk Eleman α^3	16170	338	26	0
	İlk Eleman α^4	16464	322	46	0
	İlk Eleman α^5	16170	531	43	0
	İlk Eleman α^6	16170	575	113	0
TOPLAM		**114072**	**3616**	**410**	**0**
GENEL TOPLAM		**229320**	**7336**	**739**	**24**

Örnek 5.5 ve 5.6'da verilen, M_1 ve M_2 matrislerinin global optimizasyon sonuçlarının detaylarına ise $EK-B$'den ulaşılabilmektedir.

5.3 $\mathbb{F}_{2^4}$ Cismine Ait Deneysel Bulgular

$\mathbb{F}_{2^4}$ sonlu cismi üzerindeki tüm involutif MDS matris temsilcilerini arayarak aynı cisim üzerinde tanımlı, 4×4 boyutlarında involutif ve MDS matrislerin tamamının üretilmesi sonucu elde edilen optimizasyon sonuçlarına ait veriler bu kısımda incelenecektir. Deneysel sonuçlar, $\mathbb{F}_{2^4}$ sonlu cismi üzerinde tanımlı $0x13$ polinomu için 71.856 adet involutif ve MDS matris temsilcisi olduğunu göstermektedir. b_i parametrelerinin involutif MDS matris temsilcilerine uygulanmasının ardından da toplam $71.856 \cdot (2^4-1)^3 = 242.514.000 \approx 2^{27.85}$ adet, 4×4 boyutlarında involutif ve MDS matris elde edilmektedir. Aynı işlemler, Hadamard matris formu kullanılarak yapıldığında yalnızca 1.512 adet involutif MDS matris temsilcisi elde edilebilmekte, bu temsilci matrislere Pehlivanoğlu vd. (2018) çalışmasında verilen GHadamard matris formu uygulanarak da toplam $1.512 \cdot (2^4-1)^3 = 5.103.000 \approx 2^{22.28}$ adet, 4×4 boyutlu involutif ve MDS matris oluşturulabilmektedir.

Örnek 5.7'de önerilen teknik ile elde edilen ve literatürde bilinen matrislerden daha üstün olan, $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı yeni bir 4×4 boyutlarında involutif MDS matris verilmektedir. Literatürde bilinen en hafif matris (Sarkar & Syed, 2016), 4 devre derinliğinde 42 XOR işlemi ile uygulanabilirken, aşağıda verilen M_3 matrisi 4 devre derinliğinde yalnızca 38 XOR işlemi ile uygulanabilmektedir. Buna ek olarak aynı matris, devre derinliği 5 olarak seçildiğinde 36 XOR işlemi gerektirirken, Sarkar ve Syed (2016) çalışmasındaki matris, bu derinlik için 42 XOR işlemi ile uygulanmaktadır.

Örnek 5.7. *$\mathbb{F}_{2^4}$ cismi üzerinde tanımlı $0x13$ polinomunun bir kökü α olmak üzere 4×4 boyutlarında bir involutif MDS matris olan M_3, aşağıdaki gibi gösterilmektedir:*

$$M_3 = \begin{bmatrix} 1 & \alpha^4 & \alpha^{14} & \alpha \\ 1 & \alpha^{14} & \alpha & \alpha^{14} \\ \alpha^8 & \alpha & \alpha^{14} & \alpha^4 \\ \alpha^2 & \alpha^8 & 1 & 1 \end{bmatrix}.$$

M_3 matrisi, $b_1 = \alpha^4, b_2 = \alpha^5$ ve $b_3 = \alpha^9$ parametrelerinin aşağıda verilen RIM_3 involutif

MDS matris temsilcisine uygulanmasıyla elde edilmektedir:

$$RIM_3 = \begin{bmatrix} 1 & 1 & \alpha^9 & \alpha^7 \\ \alpha^4 & \alpha^{14} & 1 & \alpha^9 \\ \alpha^{13} & \alpha^2 & \alpha^{14} & 1 \\ \alpha^{11} & \alpha^{13} & \alpha^4 & 1 \end{bmatrix}.$$

Bir involutif MDS matris olan M_3 matrisinin ham XOR sayısı, $(31 + 4 \cdot 3 \cdot 4) = 79$ *XOR'dur. Matrise BP optimizasyonunun uygulanmasının ardından derinlik 4 ve derinlik 5 değerleri için sırasıyla* 38 *XOR ve* 36 *XOR maliyetlerine ulaşılmıştır. M_3 matrisinin global optimizasyon sonuçlarının detaylarına ise $EK-B$'den ulaşılabilmektedir.*

Örnek 5.8'de, $\mathbb{F}_{2^4}$ üzerinde tanımlı yeni bir 4×4 boyutunda involutif olmayan MDS matris verilmektedir. Bu matris, Örnek 5.7'de verilen 4×4 boyutundaki involutif MDS matrisiyle aynı elemanlara sahip olacak şekilde tüm antidiyagonal simetrik matrislerin aranmasıyla bulunmuştur. Bu non-involutif MDS matrisi ve tersi, derinlik 5 için yalnızca 36 *XOR* işlemi ile uygulanabilir.

Örnek 5.8. $\mathbb{F}_{2^4}$ *cismi üzerinde tanımlı* $0x13$ *polinomunun bir kökü* α *olmak üzere* 4×4 *boyutlarında bir antidiyagonal simetrik non-involutif MDS matris olan M_4, aşağıdaki gibi gösterilmektedir:*

$$M_4 = \begin{bmatrix} \alpha^{14} & 1 & \alpha^4 & \alpha \\ \alpha^{14} & \alpha^8 & \alpha & \alpha^4 \\ 1 & \alpha^2 & \alpha^8 & 1 \\ \alpha & 1 & \alpha^{14} & \alpha^{14} \end{bmatrix}.$$

M_4 matrisinin tersi (M_4^{-1}) *ise aşağıda gibi gösterilmektedir:*

$$M_4^{-1} = \begin{bmatrix} \alpha^8 & \alpha^{14} & \alpha^4 & \alpha \\ 1 & \alpha^{14} & \alpha & \alpha^4 \\ 1 & \alpha & \alpha^{14} & \alpha^{14} \\ \alpha^2 & 1 & 1 & \alpha^8 \end{bmatrix}.$$

Bir non-involutif MDS matris olan M_4 ve tersi M_4^{-1} matrislerinin ham XOR sayısı, $(31+4\cdot 3\cdot 4) = 79$ *XOR'dur. Matrislere BP optimizasyonunun uygulanmasının ardından derinlik 5 değeri için* 36 *XOR maliyeti ile uygulanabildikleri görülmüştür. Ayrıca hem M_4 hem de M_4^{-1} matrislerinin, aynı elemanların aynı sayıda tekrarını içerdiğine dikkat edilmelidir.*

Önerilen teknik kullanılarak $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı ham *XOR* eşik değeri 80 olacak şekilde üretilen involutif MDS matrislerin optimizasyon sonuçlarının

tamamı Çizelge 5.7'de verilmektedir. Verilen çizelgede matrisler ilk elemanlarına göre gruplandırılmış olup, 36 XOR maliyetine sahip 13 adet matrisin tümüne ise $EK - B$ bölümünden ulaşılabilir.

Çizelge 5.7. İnvolutif Matrisler İçin $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı $0x13$ Polinomuna Ait Deneysel Bulgular

			BP Sonuçları			
İlk Eleman	**Ham XOR Aralığı**	**Matris Sayısı**	**39 XOR**	**38 XOR**	**37 XOR**	**36 XOR**
1	XOR ≤ 66	3172	24	4	0	0
	$67-70$	12860	25	10	0	5
	$71-74$	35304	30	12	0	0
	$75-77$	45826	31	12	0	0
	$78-80$	60986	20	6	2	0
	TOPLAM	**158148**	**130**	**44**	**2**	**5**
α	XOR ≤ 66	1382	0	4	0	0
	$67-70$	7082	13	10	0	0
	$71-74$	22290	16	12	0	0
	$75-77$	33000	14	2	0	0
	$78-80$	49922	8	0	0	0
	TOPLAM	**113676**	**51**	**28**	**0**	**0**
α^2	XOR ≤ 66	1468	0	0	0	0
	$67-70$	5291	16	5	0	0
	$71-74$	17374	6	8	0	0
	$75-77$	26374	16	2	0	0
	$78-80$	42174	12	4	0	0
	TOPLAM	**92681**	**50**	**19**	**0**	**0**
α^4	XOR ≤ 66	172	6	2	0	0
	$67-70$	2196	4	6	0	0
	$71-74$	8790	9	8	0	0
	$75-77$	15100	15	2	0	0
	$78-80$	27462	6	4	0	2
	TOPLAM	**53720**	**40**	**22**	**0**	**2**
α^8	XOR ≤ 66	12	0	0	0	0
	$67-70$	730	2	0	0	2
	$71-74$	5590	2	0	0	0
	$75-77$	12614	8	4	0	0
	$78-80$	24518	4	2	1	1
	TOPLAM	**43464**	**16**	**6**	**1**	**3**
α^{14}	XOR ≤ 66	2314	10	2	0	0
	$67-70$	10080	7	2	0	1
	$71-74$	30316	11	8	0	0
	$75-77$	42326	32	2	0	0
	$78-80$	60946	10	4	0	2
	TOPLAM	**145982**	**70**	**18**	**0**	**3**
GENEL TOPLAM		**607671**	**357**	**137**	**3**	**13**

İnvolutif ve MDS olan M_3 matrisinin elemanları kullanılarak türetilen non-involutif matrislerden, simetrik özellik gösteren matrislerin optimizasyon sonuçları Çizelge 5.8'de, antidiyagonal köşegenine göre simetrik matrislerin optimizasyon sonuçları ise Çizelge 5.9'da verilmektedir. Her iki matris formu için de ham XOR eşik değerleri, 66 ile 85 XOR aralığında seçilmiştir.

Çizelge 5.8. Non-involutif Simetrik Matrisler İçin $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı $0x13$ Polinomuna Ait Deneysel Bulgular

		BP Sonuçları			
Matris Tipi	**Matris Sayısı**	**39 XOR**	**38 XOR**	**37 XOR**	**36 XOR**
Non-involutif Simetrik	**206052**	**126**	**51**	**6**	**2**

Çizelge 5.9. Non-involutif Antidiyagonal Simetrik Matrisler İçin $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı $0x13$ Polinomuna Ait Deneysel Bulgular

		BP Sonuçları			
Matris Tipi	**Matris Sayısı**	**39 XOR**	**38 XOR**	**37 XOR**	**36 XOR**
Antidiyagonal Simetrik	**192680**	**71**	**62**	**10**	**6**

$\mathbb{F}_{2^3}$ sonlu cismi üzerinde tanımlı $0xb$ ve $0xd$ polinomları kullanılarak, $\mathbb{F}_{2^4}$ sonlu cismi üzerinde ise $0x13$ polinomu kullanılarak elde edilen, 4×4 boyutundaki involutif MDS matrislerin literatürde bulunan diğer matrislerle karşılaştırılması ise Çizelge 5.10'da verilmektedir.

Çizelge 5.10. $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ Cismi Üzerinde 4×4 Boyutunda İnvolutif MDS Matrislerin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması

Cisim / Polinom	**Ham XOR Sayısı**	**BP Optimizasyon Sonrası XOR Sayıları**				**Referans**
		Derinlik 3	**Derinlik 4**	**Derinlik 5**	**Derinlik 6**	
$\mathbb{F}_{2^4}/0x13$	72	51	50	50	–	(Sim vd., 2015)
$\mathbb{F}_{2^4}/0x13$	64	48	42	42	–	(Sarkar & Syed, 2016)
$\mathbb{F}_{2^4}/0x13$	68	47	47	47	–	(Jean vd., 2017)
$\mathbb{F}_{2^4}/0x13$	**79**	–	**38**	**36**	–	**Örnek 5.7**
$\mathbb{F}_{2^3}/0xb$	**59**	**35**	**30**	**29**	**28**	**Örnek 5.5**
$\mathbb{F}_{2^3}/0xd$	**59**	**35**	**30**	**29**	**28**	**Örnek 5.6**

$\mathbb{F}_{2^4}$ sonlu cismi üzerinde $0x13$ polinomu kullanılarak elde edilen, 4×4 boyutundaki involutif olmayan MDS matrislerin literatürde bulunan diğer matrislerle karşılaştırılması ise Çizelge 5.11'de verilmektedir.

Çizelge 5.11. $\mathbb{F}_{2^4}$ Cismi Üzerinde 4×4 Boyutunda İnvolutif Olmayan MDS Matrislerin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması

Cisim / Polinom	Ham XOR Sayısı	BP Optimizasyon Sonrası XOR Sayıları			Referans
		Derinlik 3	Derinlik 4	Derinlik 5	
$\mathbb{F}_{2^4}/0x13$	68	51	50	50	(Sim vd., 2015)
$\mathbb{F}_{2^4}/0x13$	60	47	46	46	(Liu & Sim, 2016)
$\mathbb{F}_{2^4}/0x13$	64	47	42	42	(Beierle vd., 2016)
$\mathbb{F}_{2^4}/0x19$	60	46	44	44	(Beierle vd., 2016)
$\mathbb{F}_{2^4}/0x19$	58	44	43	43	(Sarkar & Syed, 2016)
$\mathbb{F}_{2^4}/0x13$	61	45	44	43	(Jean vd., 2017)
$\mathbb{F}_{2^4}/0x13$	66	–	39	39	(Sajadieh & Mousavi, 2018)
$\mathbb{F}_{2^4}/0x19$	62	43	39	38	(Sajadieh & Mousavi, 2018)
$\mathbb{F}_{2^4}/0x13$	57	41	–	–	(Duval & Leurent, 2018)
$\mathbb{F}_{2^4}/0x13$	**79**	–	**38**	**36**	**Örnek 5.8**

Çizelge 5.12 ve 5.13'de ise sırasıyla $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cisimleri üzerinde tanımlı, 4×4 involutif MDS matrislerdeki 1'lerin sayısı verilmektedir.

Açıklama 5.1. Junod ve Vaudenay (2004) çalışmasında, 4×4 boyutundaki MDS matrislerde bulunan 1'lerin sayısının maksimum 9 adet olabileceğini göstermektedir.

Bu tez çalışmasında da $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cisimleri üzerinde tanımlı, 4×4 boyutundaki tüm involutif ve MDS matrisler elde edildiğinden, söz konusu bu matrisler için de 1'lerin sayısının maksimum 9 adet olduğu kanıtlanarak Junod ve Vaudenay (2004) çalışmasında verilen sonuç güncellenmiş olmaktadır.

Çizelge 5.12. $\mathbb{F}_{2^3}$ Cismi Üzerinde Tanımlı 4×4 İnvolutif MDS Matrislerdeki 1'lerin Sayısı

1'lerin Sayısı	0	1	2	3	4	5	6	7	8	9
Matris Sayısı	1.368	2.424	4.608	3.600	1.944	1.296	720	432	0	72

Çizelge 5.13. $\mathbb{F}_{2^4}$ Cismi Üzerinde Tanımlı 4×4 İnvolutif MDS Matrislerdeki 1'lerin Sayısı

1'lerin Sayısı	0	1	2	3	4	5	6	7	8	9
Matris Sayısı	73.266.816	88.442.736	53.722.608	20.148.576	5.555.760	1.146.768	206.160	21.120	3.264	192

Örnek 5.9. *$\mathbb{F}_{2^4}$ cismi üzerinde tanımlı $0x13$ polinomunun bir kökü α olmak üzere 4×4 boyutlarında bir involutif MDS matris temsilcisi olan RIM_5, aşağıdaki gibi gösterilmek-*

tedir:

$$RIM_5 = \begin{bmatrix} 1 & \alpha^4 & \alpha^{11} & \alpha^{13} \\ \alpha^{11} & \alpha^{12} & \alpha^{11} & \alpha^{11} \\ \alpha^3 & \alpha^8 & \alpha^{12} & \alpha^4 \\ \alpha^5 & \alpha^3 & \alpha^{11} & 1 \end{bmatrix}.$$

RIM_5 matris temsilcisine, $b_1 = \alpha^{11}, b_2 = \alpha^4$ ve $b_3 = 1$ parametrelerinin uygulanmasıyla elde edilen M_5 involutif MDS matrisi ise aşağıdaki gibidir:

$$M_5 = \begin{bmatrix} 1 & 1 & 1 & \alpha^{13} \\ 1 & \alpha^{12} & \alpha^4 & 1 \\ \alpha^{14} & 1 & \alpha^{12} & 1 \\ \alpha^5 & \alpha^{14} & 1 & 1 \end{bmatrix}.$$

M_5 matrisindeki 1'lerin sayısı, 4×4 boyutunda bir MDS matris için görülebilecek maksimum 1 sayısı olan 9'a eşit olmaktadır.

5.4 $\mathbb{F}_{2^8}$ Cismine Ait Deneysel Bulgular

$\mathbb{F}_{2^8}$ sonlu cisim uzayı, $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ sonlu cisim uzaylarına kıyasla çok daha büyük olduğundan, tez çalışması kapsamında $\mathbb{F}_{2^8}$ cismi ile çalışılırken birbirinden farklı iki adet yaklaşım izlenmektedir. İlk yaklaşımda, $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı, ham XOR sayıları ve optimizasyon sonuçları iyi olan elemanlar seçilerek $\mathbb{F}_{2^8}$ cisminde tanımlı, $0x11d$ ve $0x12d$ polinomları üzerinden bu iyi elemanların izomorfik karşılıkları bulunarak optimizasyonları yapılmıştır. İkinci yaklaşımda ise önerilen tekniğin kapsadığı GHadamard matris formu kullanılarak mümkün olan en düşük ham XOR sayısına sahip, involutif ve involutif olmayan MDS matrisler aranmış ve optimizasyon işlemleri gerçekleştirilmiştir.

$\mathbb{F}_{2^8}$ sonlu cismi üzerinde, optimizasyon işlemleri yapılırken ise öncelikli olarak Paar algoritması kullanılmıştır. Bunun sebebi, BP algoritmasını uygulayabilmek için $\mathbb{F}_{2^8}$ sonlu cisim matrisinden dönüştürülen ikili matrisin 32×32 boyutlarında olması ve bu boyutta BP algoritmasının Paar algoritmasına göre daha yavaş çalışmasıdır. Dolayısıyla, önce Paar algoritmasıyla XOR maliyeti 115 ve daha altında olan matrisler tespit edilmiş, ardından bu matrislere BP algoritması uygulanarak optimizasyon tamamlanmıştır. Bu optimizasyon çalışmaları sırasında yapılan gözlemlerde, $\mathbb{F}_{2^8}$ cismi üzerindeki en başarılı sonuçların $0x12d$ polinomu kullanılarak elde edilebildiği görülmüş ve bu polinoma ağırlık verilmiştir.

Sonuç olarak yukarıda verilen yaklaşımlar kullanıldığında;

- $\mathbb{F}_{2^8}/0x11d$ üzerinde tanımlı:
  - 902 adet involutif 4×4 boyutunda MDS matris,
  - 832 adet non-involutif 4×4 boyutunda MDS matris,
- $\mathbb{F}_{2^8}/0x12d$ üzerinde tanımlı:
  - 2288 adet involutif 4×4 boyutunda MDS matris,
  - 63206 adet non-involutif 4×4 boyutunda MDS matris,
- $\mathbb{F}_{2^8}/0x163, 0x165, 0x169, 0x1c3, 0x1e7$ üzerinde tanımlı:
  - 30 adet involutif 4×4 boyutunda MDS matris,

üretilerek Paar ve BP optimizasyonları yapılmıştır.

Örnek 5.10'de önerilen teknik ile elde edilen ve literatürde bilinen matrislerden daha üstün olan, $\mathbb{F}_{2^8}/0x12d$ üzerinde tanımlı yeni bir 4×4 boyutlarında involutif MDS matris verilmektedir. Verilen M_6 matrisi, 6 devre derinliğinde 88 XOR işlemi ile uygulanabilmektedir.

Örnek 5.10. $\mathbb{F}_{2^8}$ *cismi üzerinde tanımlı* $0x12d$ *polinomunun bir kökü* α *olmak üzere* 4×4 *boyutlarında bir involutif MDS matris olan* M_6*, aşağıdaki gibi gösterilmektedir:*

$$M_6 = \begin{bmatrix} 1 & \alpha^{36} & \alpha^{254} & \alpha^{38} \\ \alpha^{254} & 1 & \alpha & \alpha^{254} \\ 1 & \alpha^{39} & 1 & \alpha^{36} \\ \alpha^{2} & 1 & \alpha^{254} & 1 \end{bmatrix}.$$

M_6 *matrisi,* $b_1 = \alpha^{146}, b_2 = \alpha^{127}$ *ve* $b_3 = \alpha^{18}$ *parametrelerinin aşağıda verilen* RIM_6 *involutif MDS matris temsilcisine uygulanmasıyla elde edilmektedir:*

$$RIM_6 = \begin{bmatrix} 1 & \alpha^{145} & \alpha^{127} & \alpha^{20} \\ \alpha^{145} & 1 & \alpha^{20} & \alpha^{127} \\ \alpha^{127} & \alpha^{20} & 1 & \alpha^{145} \\ \alpha^{20} & \alpha^{127} & \alpha^{145} & 1 \end{bmatrix}.$$

Bir involutif MDS matris olan M_6 *matrisinin ham* XOR *sayısı,* $(66 + 4 \cdot 3 \cdot 8) = 162$ XOR*'dur. Paar optimizasyon sonucu* 107 XOR *olan bu matrise, BP optimizasyonunun uygulanmasının ardından derinlik 4, derinlik 5 ve derinlik 6 değerleri için sırasıyla* 98 XOR*,* 90 XOR *ve* 88 XOR *maliyetlerine ulaşılmıştır.*

$\mathbb{F}_{2^8}$ cismi üzerinde tanımlı $0x12d$ polinomu kullanılarak üretilen involutif matrislerin BP optimizasyonlarına ait sonuçlar aşağıda verilen çizelgede özetlenmektedir.

Çizelge 5.14. $\mathbb{F}_{2^8}$ Cismi Üzerinde Tanımlı $0x12d$ Polinomuna Ait Deneysel Bulgular

		BP Sonuçları			
Matris Tipi	**Matris Sayısı**	**91 XOR**	**90 XOR**	**89 XOR**	**88 XOR**
İnvolutif	**2288**	**32**	**23**	**39**	**9**

Hem M_6 matrisinin global optimizasyon sonuçlarının detaylarına hem de 88 XOR maliyetine sahip diğer matrislere ise $EK - B$'den ulaşılabilmektedir.

$\mathbb{F}_{2^8}$ sonlu cismi üzerinde $0x12d$ polinomu kullanılarak elde edilen, 4×4 boyutundaki involutif MDS matrislerin literatürde bulunan diğer matrislerle karşılaştırılması ise Çizelge 5.15'de verilmektedir.

Çizelge 5.15. $\mathbb{F}_{2^8}$ Cismi Üzerinde 4×4 Boyutunda İnvolutif MDS Matrislerin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması

Cisim / Polinom	**Ham XOR Sayısı**	**BP Optimizasyon Sonrası XOR Sayıları**			**Referans**
		Derinlik 4	**Derinlik 5**	**Derinlik 6**	
$\mathbb{F}_{2^8}/0x11d$	184	116	113	113	(P. S. Barreto, 2000)
$\mathbb{F}_{2^8}/0x165$	160	99	99	99	(Sim vd., 2015)
$\mathbb{F}_{2^8}/0x1c3$	158	113	111	111	(Pehlivanoğlu vd., 2018)
$\mathbb{F}_{2^8}/0x12d$	157	99	95	91	(Bai vd., 2020)
$\mathbb{F}_{2^8}/0x12d$	156	102	97	94	(Bai vd., 2020)
$\mathbb{F}_{2^8}/0x12d$	155	99	95	94	(Pehlivanoğlu vd., 2021)
$\mathbb{F}_{2^8}/0x12d$	157	98	94	90	(Pehlivanoğlu vd., 2021)
$\mathbb{F}_{2^8}/0x12d$	**162**	**98**	**90**	**88**	**Örnek 5.10**

Bölüm boyunca verilen deneysel bulgular incelendiğinde, önerilen teknik kullanılarak $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$ ve $\mathbb{F}_{2^8}$ cisimleri üzerinde üretilen involutif ve non-involutif MDS matrislerin BP optimizasyon sonuçlarının, literatürde bilinen matrislerden daha iyi XOR maliyetine sahip olduğu açıkça görülmektedir.

BÖLÜM 6

ÖNERİLEN TEKNİKLE ELDE EDİLEN MDBL MATRİSLERE AİT DENEYSEL BULGULAR

Bu bölümde, önerilen teknik kullanılarak Maksimum Uzaklıkta İkili Matrislerin (Maximum Distance Binary Linear-MDBL) tasarlanması için izlenen yaklaşımın detayları ve elde edilen ikili matrislerin BP optimizasyon sonuçları verilecektir. Bu kapsamda 12×12 ve 16×16 boyutlarında MDBL matrisleri oluşturmak için sırasıyla, Akleylek vd. (2016) çalışmasında verilen 3×3 ve 4×4 boyutlarında devirli (cyclic) matris gruplarından faydalanılmaktadır.

Tanım 6.1 (Devirli Matris Grupları)**.** Bir grubun tüm elemanları, grubun bir elemanı kullanılarak (elemanın üsleri alınarak) elde edilebiliyorsa bu gruba devirli veya devirsel (cyclic) grup, bu elemana ise üreteç (generator) eleman denilmektedir. Bir devirli grubun, birden fazla üreteç elemanı bulunabilmektedir.

6.1 12×12 Boyutunda İkili Matrislere Ait Deneysel Bulgular

$\mathbb{F}_{2^3}/0xb$ polinomunun bir kökü α olmak üzere herhangi bir $x \in \mathbb{F}_{2^3}$ için,

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

eşitliği yazılabilir. Burada $(\alpha_2, \alpha_1, \alpha_0) = (\alpha^2, \alpha, 1)$ $\mathbb{F}_2$ üzerine $\mathbb{F}_{2^3}$'ün bir polinom tabanıdır ve $(x_2, x_1, x_0) \in \mathbb{F}_2$'dir. Dolayısıyla sonlu cisim $\mathbb{F}_{2^3}$'ün bir elemanı olan α'nın herhangi bir sonlu cisim elemanı olan x ile çarpımı,

$$\begin{aligned} \alpha(\alpha^2 x_2 + \alpha x_1 + x_0) &= \alpha^3 x_2 + \alpha^2 x_1 + \alpha x_0 \\ &= \alpha^2 x_1 + \alpha x_2 + x_2 + \alpha x_0 \\ &= \alpha^2 x_1 + \alpha(x_2 + x_0) + x_2 \end{aligned}$$

şeklinde elde edilir. Böylece α sonlu cisim elemanı aşağıda görülebileceği gibi 3×3 boyutunda bir ikili matris ile temsil edilebilir:

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \end{bmatrix}.$$

Bu işlemler $\mathbb{F}_{2^3}$ sonlu cisminin diğer elemanlarına uygulandığında her sonlu cisim elemanının 3×3 boyutunda ikili matris karşılıkları oluşturulabilir.

Çalışma boyunca, $\mathbb{F}_2$ cismi üzerinde tanımlı 3×3 ikili matris uzayında tüm 3×3 boyutunda ikili matrisler taranarak, g, 3×3 boyutunda ikili bir üreteç matris, I ise 3×3 boyutunda birim matrisi temsil etmek üzere; $g^3 = g + I$ ve $g^7 = I$ denklemlerini sağlayan 8 adet devirli matris grubu elde edilmiştir.

Çizelge 6.1'de, $\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ üreteç matrisi $(001 - 011 - 111)$ kullanılarak oluşturulan bir devirli matris grubu örneği verilmektedir. Çizelge 6.2'de ise 3×3 boyutunda g üreteç matrisleri kullanılarak elde edilen tüm devirli matris grupları gösterilmektedir.

Çizelge 6.1. $001 - 011 - 111$ Üreteç Matrisi ile Elde Edilen Devirli Matris Grubu

$g^1 = g$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$g^5 = g^3 + g^2 = g^2 + g + I$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
$g^2 = g^2$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$g^6 = g^3 + g^2 + g = g^2 + I$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
$g^3 = g + I$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	$g^7 = g^3 + g = I$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
$g^4 = g^2 + g$	$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$		

Çizelge 6.2. 3 × 3 Boyutlarında Devirli Matris Grupları

1. Grup	$001-011-111$
2. Grup	$001-100-110$
3. Grup	$001-110-011$
4. Grup	$010-011-101$
5. Grup	$010-101-100$
6. Grup	$010-111-011$
7. Grup	$011-001-100$
8. Grup	$011-100-010$

Devirli matris gruplarının elde edilmesinin ardından, önerilen teknikte verilen temsilci matris formu yardımıyla, $\mathbb{F}_{2^3}/0xb$ üzerinde tanımlı 4×4 boyutunda involutif MDS matrisler üretilmiştir. Üretilen 4×4 boyutundaki MDS matrislerin elemanlarının, ($\mathbb{F}_{2^3}/0xb$'den elemanlar) 3×3 boyutundaki devirli matris grup elemanlarıyla (ikili matrisler) yer değiştirilmesiyle de 12×12 boyutlarında ikili matrisler oluşturulmuştur. Bu yaklaşımla, 8 adet devirli matris grubu kullanılarak toplam 384 adet 12×12 boyutunda involutif ikili matris elde edilmiştir. Elde edilen involutif ikili matrislerden 96 tanesinin MDBL (dal sayısı 8) matris olduğu tespit edilmiş ve BP optimizasyonları yapılmıştır.

Örnek 6.1. *Elemanları $\mathbb{F}_{2^3}/0xb$ üzerinde tanımlı 4×4 boyutunda involutif ve MDS olan M_7 matrisi aşağıdaki gibi verilmiş olsun:*

$$M_7 = \begin{bmatrix} 1 & \alpha^4 & \alpha & \alpha^2 \\ \alpha^4 & 1 & \alpha^2 & \alpha \\ \alpha & \alpha^2 & 1 & \alpha^4 \\ \alpha^2 & \alpha & \alpha^4 & 1 \end{bmatrix}.$$

M_7 matrisinin sonlu cisim elemanları, 3×3 boyutunda ikili matris karşılıkları ile yer değiştirildiğinde ise 12×12 boyutundaki M_{7B} ikili matrisi aşağıdaki gibi elde edilir:

$$M_{7B} = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

İnvolutif bir matris olan M_{7B} matrisinin BP optimizasyonu sonrası XOR sayısı 37'dir. Ayrıca M_{7B} matrisinin dal sayısı, Çizelge 2.2'den de görülebileceği gibi 12×12 boyutu için maksimum değer olan 8'e eşit olup, M_{7B} matrisine MDBL özelliği kazandırmaktadır.

Dal sayısı 8 olarak elde edilen, 12×12 boyutlarındaki 96 adet involutif MDBL matrisin BP optimizasyon sonuçları ise aşağıdaki çizelgede özetlenmektedir.

Çizelge 6.3. 12×12 Boyutlarında MDBL Matrislere Ait Deneysel Bulgular

		BP Sonuçları	
Matris Tipi	**Matris Sayısı**	**38 XOR**	**37 XOR**
İnvolutif MDBL	96	86	10

6.2 16×16 Boyutunda İkili Matrislere Ait Deneysel Bulgular

$\mathbb{F}_{2^4}/0x13$ polinomunun bir kökü α olmak üzere herhangi bir $x \in \mathbb{F}_{2^4}$ için,

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

eşitliği yazılabilir. Burada $(\alpha_3, \alpha_2, \alpha_1, \alpha_0) = (\alpha^3, \alpha^2, \alpha, 1)$ $\mathbb{F}_2$ üzerine $\mathbb{F}_{2^4}$'ün bir polinom tabanıdır ve $(x_3, x_2, x_1, x_0) \in \mathbb{F}_2$'dir. Dolayısıyla sonlu cisim $\mathbb{F}_{2^4}$'ün bir elemanı olan

α'nın herhangi bir sonlu cisim elemanı olan x ile çarpımı,

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

şeklinde elde edilir. Böylece α sonlu cisim elemanı aşağıda görülebileceği gibi 4×4 boyutunda bir ikili matris ile temsil edilebilir:

$$\begin{bmatrix} x'_0 \\ x'_1 \\ x'_2 \\ x'_3 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}.$$

Bu işlemler $\mathbb{F}_{2^4}$ sonlu cisminin diğer elemanlarına uygulandığında her sonlu cisim elemanının 4×4 boyutunda ikili matris karşılıkları oluşturulabilir.

Çalışma boyunca, $\mathbb{F}_2$ cismi üzerinde tanımlı 4×4 ikili matris uzayında tüm 4×4 boyutunda ikili matrisler taranarak, g, 4×4 boyutunda ikili bir üreteç matris, I ise 4×4 boyutunda birim matrisi temsil etmek üzere; $g^4 = g + I$ ve $g^{15} = I$ denklemlerini sağlayan 336 adet devirli matris grubu elde edilmiştir.

Devirli matris gruplarının elde edilmesinin ardından, önerilen teknikte verilen temsilci matris formu yardımıyla, $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı 4×4 boyutunda involutif MDS ve 4×4 boyutunda involutif Almost MDS (Dal Sayısı 4) matrisler üretilmiştir. Üretilen 4×4 boyutundaki MDS ve Almost MDS matrislerin elemanlarının, ($\mathbb{F}_{2^4}/0x13$'den elemanlar) 4×4 boyutundaki devirli matris grup elemanlarıyla (ikili matrisler) yer değiştirilmesiyle de 16×16 boyutlarında ikili matrisler oluşturulmuştur. Bu yaklaşımla, 336 adet 4×4 boyutunda devirli matris grubu kullanılarak involutif MDS matrislerden 79.968, involutif Almost MDS matrislerden ise 3.654 adet olmak üzere toplam 83.622 adet 16×16 boyutunda involutif MDBL (Dal Sayısı 8) matris elde edilerek BP optimizasyonları yapılmıştır.

Örnek 6.2. *Elemanları $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı 4×4 boyutunda involutif ve almost MDS (dal sayısı değeri 4) olan M_8 matrisi aşağıdaki gibi verilmiş olsun:*

$$M_8 = \begin{bmatrix} 1 & \alpha^{10} & \alpha^{11} & \alpha^{14} \\ \alpha^{10} & 1 & \alpha^{14} & \alpha^{11} \\ \alpha^{11} & \alpha^{14} & 1 & \alpha^{10} \\ \alpha^{14} & \alpha^{11} & \alpha^{10} & 1 \end{bmatrix}.$$

M_8 matrisinin sonlu cisim elemanları, 4×4 boyutunda ikili matris karşılıkları ile yer değiştirildiğinde ise 16×16 boyutundaki M_{8B} ikili matrisi aşağıdaki gibi elde edilir:

$$M_{8B} = \begin{bmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}.$$

İnvolutif bir matris olan M_{8B} matrisinin BP optimizasyonu sonrası XOR sayısı, derinlik 4 değeri için 49'dur. Ayrıca M_{8B} matrisinin dal sayısı, Çizelge 2.2'den de görülebileceği gibi 16×16 boyutu için maksimum değer olan 8'e eşittir. Diğer bir deyişle M_{8B} matrisi bir MDBL matristir.

Dal sayısı 8 olarak elde edilen, 16×16 boyutlarındaki involutif MDBL matrislerin BP optimizasyon sonuçları aşağıdaki çizelgede görülebilmektedir.

Çizelge 6.4. 16×16 Boyutlarında MDBL Matrislere Ait Deneysel Bulgular

			BP Sonuçları		
Matris Tipi	**Üretildiği Matris**	**Matris Sayısı**	**51 XOR**	**50 XOR**	**49 XOR**
İnvolutif MDBL	İnvolutif MDS	79968	1488	600	–
	İnvolutif Almost MDS	3654	137	53	314
GENEL TOPLAM		**83622**	**1625**	**653**	**314**

16×16 boyutundaki involutif MDBL M_{8B} matrisinin literatürde bulunan diğer ikili matrislerle derinlik kriteri dikkate alınarak yapılan BP optimizasyonu sonucu elde edilen XOR sayılarının karşılaştırılması ise Çizelge 6.5'de verilmektedir.

Çizelge 6.5. 16 × 16 Boyutunda İnvolutif MDBL M_{8B} Matrisinin Global Optimizasyon Sonrası Farklı Derinliklerdeki XOR Sayılarının Literatür ile Karşılaştırılması

Matris Tipi	Ham XOR Sayısı	BP Optimizasyon Sonrası XOR Sayıları			Referans
		Derinlik 3	Derinlik 4	Derinlik 5	
İnvolutif MDBL	96	56	49	49	(Kwon vd., 2003)
İnvolutif MDBL	96	54	49	49	(Akleylek vd., 2017)
İnvolutif MDBL	**96**	**55**	**49**	**49**	**Örnek 6.2**

Bölüm boyunca verilen deneysel bulgular incelendiğinde, önerilen teknik kullanılarak $\mathbb{F}_2$ cismi üzerinde üretilen 16 × 16 boyutlu involutif MDBL matrislerin BP optimizasyon sonuçlarının, literatürde bilinen matrislerle aynı XOR maliyetine sahip olduğu açıkça görülmektedir.

BÖLÜM 7

SONUÇ VE TARTIŞMA

Bu tez çalışmasında, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı 4×4 boyutlarında involutif MDS matrislerin tamamını oluşturmak için yeni bir melez teknik önerilmiştir. Önerilen teknik, $\mathbb{F}_{2^m}$ cismi üzerinde tanımlı, bir kaba kuvvet araması ile elde edilebilecek 4×4 boyutundaki tüm MDS matrislerin sayısı n ile ifade edildiğinde, involutif MDS temsilci matrisleri bulmak için gereken arama uzayı karmaşıklığını $\sqrt{n}$ düzeyinde azaltmaktadır. Bu bağlamda, 4×4 boyutlarındaki involutif MDS matrislerin tamamı, $\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cismi üzerinden üretilebilmektedir. $\mathbb{F}_{2^8}$ sonlu cismi üzerinde ise tüm involutif MDS matris temsilcilerini oluşturmak için arama uzayı $(2^8-1)^8 \approx 2^{64}$ olmaktadır. Bu büyüklük, $\mathbb{F}_{2^8}$ cisim uzayının tamamının aranabilmesini sağlayacak kadar düşük olmasa da önerilen teknik kullanılarak $\mathbb{F}_{2^8}$ cismi üzerinde de 4×4 boyutlarındaki involutif ve MDS matrisler kolaylıkla elde edilebilmektedir. Bununla birlikte önerilen teknik, her ne kadar belirtilen sonlu cisim alanlarındaki tüm involutif ve MDS matrisleri oluşturmak amacıyla geliştirilmiş olsa da hem involutif olmayan MDS matrislerin hem de MDBL matrislerin tasarlanmasında da kullanılabilmektedir.

Önerilen teknik yardımıyla, 4×4 boyutlarında özel matris tipleri kullanılarak yapılan çalışmalarda; 39 XOR ile involutif MDS GHadamard (GH_1) matrislerde, 38 XOR ile de involutif olmayan MDS GHadamard (GH_2), Dairesel (CP_1) ve Toeplitz (TP_1) matrislerde, literatürde bilinen en iyi optimizasyon sonuçlarına ulaşılmıştır (Bkz. Çizelge 5.4).

$\mathbb{F}_{2^3}$ ve $\mathbb{F}_{2^4}$ cisimleri göz önüne alındığında önerilen teknik kullanılarak, literatürde verilen en iyi XOR sayısına sahip matrislerden daha iyi sonuçların elde edildiği görülmüştür. $\mathbb{F}_{2^3}$ cismi üzerinde, $0xb$ ve $0xd$ polinomlarının her birinde tanımlı derinlik 3, derinlik 4 ve derinlik 5 değerleri için bilinen en düşük XOR sayısına sahip, involutif

MDS olan M_1 ve M_2 matrisleri elde edilmiştir. Söz konusu iki matris ayrıca, derinlik 6 değeri için de literatürde şimdiye kadar bilinen en düşük değer olan 28 XOR işlemi ile uygulanabilmektedir. M_3 matrisiyle ise $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı, 4×4 boyutunda involutif ve MDS matrisler için, devre derinliği 4 olmak üzere literatürde bilinen en iyi değer olan 42 XOR sayısı, 4 XOR azaltılarak 38 XOR olarak iyileştirilmiştir. Ayrıca, devre derinliği 5 olarak seçildiğinde ise literatürde bilinen en iyi değer 42 XOR iken, aynı yeni M_3 matrisi, sadece 36 adet XOR işlemi ile uygulanabilmektedir ki bu da uygulama maliyetinin 6 XOR azaltılması anlamına gelmektedir (Bkz. Çizelge 5.10). $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı, 4×4 boyutunda involutif olmayan MDS matrisler üzerinde yapılan çalışmada ise elde edilen M_4 matrisi ve tersi, derinlik 5 değeri için 36 XOR uygulama maliyeti ile yine literatürde bilinen en iyi sonucu vermiştir (Bkz. Çizelge 5.11).

$\mathbb{F}_{2^8}$ sonlu cismi söz konusu olduğunda, $0x12d$ polinomu üzerinde tanımlı 4×4 boyutlarında involutif ve MDS M_6 matrisinin, derinlik 5 değeri için 90 XOR ile literatürde bilinen en iyi değerden 4 XOR daha az, derinlik 6 değeri için ise 88 XOR ile literatürde bilinen en iyi değerden 2 XOR daha az maliyetle optimize edilebildiği görülmüştür (Bkz. Çizelge 5.15). Bununla birlikte, $\mathbb{F}_{2^8}$ sonlu cisminin arama uzayının büyüklüğünden dolayı bu cisimde tanımlı tüm involutif MDS matrislerin elde edilmemiş olduğu unutulmamalıdır.

Önerilen teknik ile MDBL matrisler tasarlanırken ise devirli matris gruplarından faydalanılmıştır. 3×3 ve 4×4 boyutlarında devirli matris grupları kullanılarak, sırasıyla 12×12 ve 16×16 boyutlarında MDBL matrisler oluşturulmuştur. Yapılan deneysel çalışmalar sonucunda ise 16×16 boyutlarındaki involutif MDBL olan M_{8B} matrisinin optimizasyon sonuçlarının literatürde bilinen en iyi değerlere sahip matrislerle aynı maliyete sahip olduğu tespit edilmiştir (Bkz. Çizelge 6.5).

MDS matrisleri global olarak optimize etmek için literatürde verilen yeni ve farklı bir yaklaşım da MDS matrisin kendisini optimize etmek yerine, bu MDS matrise karşılık gelen bir devre bulana kadar donanım maliyetine göre sıralanmış bir dizi devrede arama yapılmasına dayanmaktadır. Temel olarak, devrede tekrarlanan ara değerlerin yeniden kullanılması ve devrenin XOR kapısı sayısının azaltılması amaçlanmaktadır. Özellikle bellek kullanımı açısından maliyeti oldukça yüksek olan bu yeni yaklaşımda Yosys, ABC (Duval & Leurent, 2018) ve LIGHTER (Jean vd., 2017) gibi devre yazaçları üzerinde çalışan sentez araçlarından yararlanılmaktadır. Bu tez çalışmasında da Yosys aracı kullanılarak, $\mathbb{F}_{2^3}$, $\mathbb{F}_{2^4}$ ve $\mathbb{F}_{2^8}$ sonlu cisimleri üzerinde üretilen ve BP optimizasyon sonuçları sırasıyla 28, 36 ve 88 XOR olan matrislere tekrar optimizasyon yapılmış ancak, XOR maliyeti açısından herhangi bir iyileştirme görülmemiştir.

Gelecekte, BP algoritmasının iyileştirilmesi, yeni optimizasyon algoritmalarının geliştirilmesi ve farklı yöntemlerin ortaya çıkması ile bu çalışmada elde edilen sonuçların

iyileştirilmesi beklenebilir. Diğer taraftan, Bölüm 3'te sunulan R_1 matrisinin bazı parametrelerinin elenerek azaltılması ile yeni bir arama formu elde edilebildiği takdirde, $\mathbb{F}_{2^8}$ gibi daha büyük sonlu cisimler üzerinde involutif ve MDS matris temsilcilerini bulmak için arama uzayı azaltılabilir. Ancak bu yeni formun yapısının, yüksek olasılıkla R_1 formundan daha karmaşık olacağı düşünülmektedir.

Sonuç olarak, önerilen tekniğin sadece $\mathbb{F}_{2^m}$ sonlu cisimleri üzerinde tüm involutif MDS matrisleri oluşturmak için değil, aynı zamanda $GL(m, \mathbb{F}_2)$ genel doğrusal gruplar (General Linear Groups-GL) gibi, literatürde hafif sıklet involutif MDS matrisleri oluşturmak için kullanılan diğer yöntemlere de uygulanabileceği değerlendirilmektedir.

KAYNAKLAR

Akleylek, S., Rijmen, V., Sakallı, M. T. & Öztürk, E. (2017). Efficient methods to generate cryptographically significant binary diffusion layers. *IET Information Security*, *11*(4), 177–187.

Akleylek, S., Sakallı, M. T., Öztürk, E., Mesut, A. Ş. & Tuncay, G. (2016). Generating binary diffusion layers with maximum/high branch numbers and low search complexity. *Security and Communication Networks*, *9*(16), 3558–3569.

Albrecht, M. R., Driessen, B., Kavun, E. B., Leander, G., Paar, C. & Yalçın, T. (2014). Block ciphers–focus on the linear layer (feat. PRIDE). *Annual Cryptology Conference*, 57–76.

Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J. & Tokita, T. (2000). Camellia: A 128-bit block cipher suitable for multiple platforms—design andanalysis. *International workshop on selected areas in cryptography*, 39–56.

Aslan, B. (2013). *Blok şifreler için cebirsel ikili doğrusal dönüşüm tasarımı ve modern bir blok şifreye uygulanması* (Doktora tezi). Trakya University.

Aslan, B. & Sakallı, M. T. (2014). Algebraic construction of cryptographically good binary linear transformations. *Security and Communication Networks*, *7*(1), 53–63.

Augot, D. & Finiasz, M. (2014). Direct construction of recursive MDS diffusion layers using shortened BCH codes. *International Workshop on Fast Software Encryption*, 3–17.

Bai, J., Sun, Y. & Wang, D. (2020). On the Construction of Involutory MDS Matrices over $\mathbb{F}_{2^m}$. *Journal of Systems Science and Complexity*, *33*(3), 836–848.

Banik, S., Funabiki, Y. & Isobe, T. (2019). More results on shortest linear programs. *International Workshop on Security*, 109–128.

Barreto, P. S. (2000). The Anubis block cipher. *NESSIE*.

Barreto, P. & Rijmen, V. (2000). The Khazad legacy-level block cipher. *Primitive submitted to NESSIE*, *97*, 106.

Barreto, P., Rijmen, V. vd. (2000). The Whirlpool hashing function. *First open NESSIE Workshop, Leuven, Belgium*, *13*, 14.

Beierle, C., Kranz, T. & Leander, G. (2016). Lightweight multiplication in $GF(2^n)$ with applications to MDS matrices. *Annual International Cryptology Conference*, 625–653.

Berger, T. P. (2013). Construction of recursive MDS diffusion layers from Gabidulin codes. *International Conference on Cryptology in India*, 274–285.

Berger, T. P. & Ourivski, A. (2009). Construction of new MDS codes from Gabidulin codes. *Proceedings of ACCT*, 40–47.

Biham, E. (1994). New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, *7*(4), 229–246.

Black, P. E. (2012). Greedy algorithm, Dictionary of Algorithms and Data Structures. *US Nat. Inst. Std. & Tech Report*, *88*, 95.

Blaum, M. & Roth, R. M. (1999). On lowest density MDS codes. *IEEE Transactions on Information Theory*, *45*(1), 46–59.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Seurin, Y. & Vikkelsoe, C. (2007). PRESENT: An ultra-lightweight block cipher. *International workshop on cryptographic hardware and embedded systems*, 450–466.

Bose, R. C. & Ray-Chaudhuri, D. K. (1960). On a class of error correcting binary group codes. *Information and control*, *3*(1), 68–79.

Boyar, J., Matthews, P. & Peralta, R. (2008). On the shortest linear straight-line program for computing linear forms. *International Symposium on Mathematical Foundations of Computer Science*, 168–179.

Boyar, J., Matthews, P. & Peralta, R. (2013). Logic minimization techniques with applications to cryptology. *Journal of Cryptology*, *26*(2), 280–312.

Boyar, J. & Peralta, R. (2010). A new combinational logic minimization technique with applications to cryptology. *International Symposium on Experimental Algorithms*, 178–189.

Carlet, C., Crama, Y. & Hammer, P. L. (2010). Vectorial Boolean Functions for Cryptography.

Cauchois, V. & Loidreau, P. (2019). On circulant involutory MDS matrices. *Designs, Codes and Cryptography*, *87*(2), 249–260.

Choy, J., Yap, H., Khoo, K., Guo, J., Peyrin, T., Poschmann, A. & Tan, C. H. (2012). SPN-Hash: improving the provable resistance against differential collision attacks. *International Conference on Cryptology in Africa*, 270–286.

Cui, T., Jin, C. & Kong, Z. (2014). On compact cauchy matrices for substitution-permutation networks. *IEEE Transactions on Computers*, *64*(7), 2098–2102.

Daemen, J., Knudsen, L. & Rijmen, V. (1997). The block cipher Square. *International Workshop on Fast Software Encryption*, 149–165.

Davis, P. J. (1979). *Circulant matrices*. Wiley.

De Canniere, C. & Preneel, B. (2008). Trivium. *New stream cipher designs* (ss. 244–266). Springer.

Deamen, J. & Rijmen, V. (2002). The design of rijndael. *AES-The Advanced Encryption Standard*. Springer-Verlag.

Diffie, W. & Hellman, M. (1976). New directions in cryptography. *IEEE transactions on Information Theory*, *22*(6), 644–654.

Duval, S. & Leurent, G. (2018). MDS matrices with lightweight circuits. *IACR Transactions on Symmetric Cryptology*, *2018*(2), 48–78.

Englund, H., Johansson, T. & Turan, M. S. (2007). A framework for chosen IV statistical analysis of stream ciphers. *International Conference on Cryptology in India*, 268–281.

Feistel, H. (1973). Cryptography and computer privacy. *Scientific american*, *228*(5), 15–23.

Forouzan, B. A. & Mukhopadhyay, D. (2015). *Cryptography and network security*. Mc Graw Hill Education (India) Private Limited.

Fuhs, C. & Schneider-Kamp, P. (2010). Synthesizing shortest linear straight-line programs over GF (2) using SAT. *International Conference on Theory and Applications of Satisfiability Testing*, 71–84.

Fundamental Package for Scientific Computing with Python. (2021). 16 Temmuz 2021 tarihinde https://numpy.org adresinden erişildi.

GD Filho, B. (2006). P., Rijmen, V.: The maelstrom-0 hash function. *Proceedings of the 6th Brazilian Symposium on Information and Computer Systems Security*.

Gloub, G. H. & Van Loan, C. F. (1996). Matrix computations. *Johns Hopkins Universtiy Press, 3rd edtion*.

Gray, R. (2005). Toeplitz and Circulant Matrices: a Review, the series: Foundations and Trends in Communications and Information Theory, Vol. 2, No. 3.

Guo, J., Peyrin, T. & Poschmann, A. (2011). The PHOTON family of lightweight hash functions. *Annual Cryptology Conference*, 222–239.

Guo, J., Peyrin, T., Poschmann, A. & Robshaw, M. (2011). The LED block cipher. *International workshop on cryptographic hardware and embedded systems*, 326–341.

Guo, Z., Wu, W. & Gao, S. (2015). Constructing lightweight optimal diffusion primitives with feistel structure. *International Conference on Selected Areas in Cryptography*, 352–372.

Gupta, K. C., Pandey, S. K., Ray, I. G. & Samanta, S. (2019). Cryptographically significant MDS matrices over finite fields: A brief survey and some generalized results. *Advances in Mathematics of Communications*, *13*(4), 779.

Gupta, K. C., Pandey, S. K. & Venkateswarlu, A. (2017). On the direct construction of recursive MDS matrices. *Designs, Codes and Cryptography*, *82*(1-2), 77–94.

Gupta, K. C., Pandey, S. K. & Venkateswarlu, A. (2019). Almost involutory recursive MDS diffusion layers. *Designs, Codes and Cryptography*, *87*(2), 609–626.

Gupta, K. C. & Ray, I. G. (2013). On constructions of involutory MDS matrices. *International Conference on Cryptology in Africa*, 43–60.

Gupta, K. C. & Ray, I. G. (2014). On constructions of circulant MDS matrices for lightweight cryptography. *International Conference on Information Security Practice and Experience*, 564–576.

Gupta, K. C. & Ray, I. G. (2015). Cryptographically significant MDS matrices based on circulant and circulant-like matrices for lightweight applications. *Cryptography and Communications*, *7*(2), 257–287.

Güzel, G. G., Sakallı, M. T., Akleylek, S., Rijmen, V. & Çengellenmiş, Y. (2019). A new matrix form to generate all 3× 3 involutory MDS matrices over $\mathbb{F}_{2^m}$. *Information Processing Letters*, *147*, 61–68.

Heys, H. M. & Tavares, S. E. (1994). The design of substitution-permutation networks resistant to differential and linear cryptanalysis. *Proceedings of the 2nd ACM Conference on Computer and Communications Security*, 148–155.

Heys, H. M. & Tavares, S. E. (1995). Avalanche characteristics of substitution-permutation encryption networks. *IEEE Transactions on Computers*, *44*(9), 1131–1139.

Heys, H. M. & Tavares, S. E. (1996). Substitution-permutation networks resistant to differential and linear cryptanalysis. *Journal of cryptology*, *9*(1), 1–19.

Hocquenghem, A. (1959). Codes correcteurs d'erreurs. *Chiffers*, *2*, 147–156.

Jean, J., Peyrin, T., Sim, S. M. & Tourteaux, J. (2017). Optimizing implementations of lightweight building blocks. *IACR Transactions on Symmetric Cryptology*, 130–168.

Junod, P. & Macchetti, M. (2009). Revisiting the IDEA philosophy. *International Workshop on Fast Software Encryption*, 277–295.

Junod, P. & Vaudenay, S. (2004). Perfect diffusion primitives for block ciphers. *International Workshop on Selected Areas in Cryptography*, 84–99.

Kam, J. B. & Davida, G. I. (1979). Structured design of substitution-permutation encryption networks. *IEEE Computer Architecture Letters*, *28*(10), 747–753.

Kerckhoffs, A. (1883). La cryptographic militaire. *Journal des sciences militaires*, 5–38.

Khoo, K., Peyrin, T., Poschmann, A. Y. & Yap, H. (2014). FOAM: searching for hardware-optimal SPN structures and components with a fair comparison. *International Workshop on Cryptographic Hardware and Embedded Systems*, 433–450.

Koo, B. W., Jang, H. S. & Song, J. H. (2006). On constructing of a 32× 32 binary matrix as a diffusion layer for a 256-bit block cipher. *International Conference on Information Security and Cryptology*, 51–64.

Kra, I. & Simanca, S. R. (2012). On circulant matrices. *Notices of the AMS*, *59*(3), 368–377.

Kranz, T., Leander, G., Stoffelen, K. & Wiemer, F. (2017). Shorter linear straight-line programs for MDS matrices. *IACR Transactions on Symmetric Cryptology*, 188–211.

Kwon, D., Kim, J., Park, S., Sung, S. H., Sohn, Y., Song, J. H., Yeom, Y., Yoon, E.-J., Lee, S., Lee, J. vd. (2003). New block cipher: ARIA. *International Conference on Information Security and Cryptology*, 432–445.

Kwon, D., Sung, S. H., Song, J. H. & Park, S. (2005). Design of block ciphers and coding theory. *Trends in Mathematics*, *8*(1), 13–20.

Lacan, J. & Fimes, J. (2004). Systematic MDS erasure codes based on Vandermonde matrices. *IEEE Communications Letters*, *8*(9), 570–572.

Lacan, J. & Fimes, J. (2003). A construction of matrices with no singular square submatrices. *International conference on finite fields and applications*, 145–147.

Li, S., Sun, S., Li, C., Wei, Z. & Hu, L. (2019). Constructing low-latency involutory MDS matrices with lightweight circuits. *IACR Transactions on Symmetric Cryptology*, 84–117.

Li, Y. & Wang, M. (2016). On the construction of lightweight circulant involutory MDS matrices. *International Conference on Fast Software Encryption*, 121–139.

Lidl, R. & Niederreiter, H. (1994). *Introduction to finite fields and their applications*. Cambridge university press.

Ling, S. & Xing, C. (2004). *Coding theory: a first course*. Cambridge University Press.

Liu, M. & Sim, S. M. (2016). Lightweight MDS generalized circulant matrices. *International Conference on Fast Software Encryption*, 101–120.

MacWilliams, F. J. & Sloane, N. J. A. (1977). *The theory of error correcting codes* (C. 16). Elsevier.

Magma Computational Algebra System. (2021). 13 Haziran 2021 tarihinde http://magma.maths.usyd.edu.au/magma/ adresinden erişildi.

Mathematics Software System. (2021). 25 Temmuz 2021 tarihinde https://sagemath.org adresinden erişildi.

Mattoussi, F., Roca, V. & Sayadi, B. (2012). Complexity comparison of the use of Vandermonde versus Hankel matrices to build systematic MDS Reed-Solomon codes. *2012 IEEE 13th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 344–348.

May, L., Henricksen, M., Millan, W., Carter, G. & Dawson, E. (2002). Strengthening the Key Schedule of the AES. *Australasian Conference on Information Security and Privacy*, 226–240.

Nakahara Jr, J. & Abrahao, E. (2009). A New Involutory MDS Matrix for the AES. *IJ Network Security*, *9*(2), 109–116.

Paar, C. (1997). Optimized arithmetic for Reed-Solomon encoders. *Proceedings of IEEE International Symposium on Information Theory*, 250.

Pehlivanoğlu, M. K. & Kavun, E. B. (2020). Kriptografik Blok Şifrelerin Maksimum Uzaklıkla Ayrılabilen Yayılım Tabakalarının Tasarımı. *Siber Güvenlik ve Savunma* (ss. 295–325). Nobel Akademik Yayıncılık Eğitim Danışmanlık Tic. Ltd. Şti.

Pehlivanoğlu, M. K., Sakallı, F. B., Akleylek, S. & Sakallı, M. T. (2021). On the Construction of 4×4 Lightweight Involutory MDS Matrices over $\mathbb{F}_{2^8}$. *Proceedings of the 7th International Conference on Mathematics and Computing ICMC*.

Pehlivanoğlu, M. K., Sakallı, M. T., Akleylek, S., Duru, N. & Rijmen, V. (2018). Generalisation of Hadamard matrix to generate involutory MDS matrices for lightweight cryptography. *IET Information Security*, *12*(4), 348–355.

Pub, F. (2012). Secure hash standard (shs). *Fips pub*, *180*(4).

Python Programming Language. (2021). 3 Temmuz 2021 tarihinde https://www.python.org adresinden erişildi.

Rijmen, V. (1997). *Cryptanalysis and design of iterated block ciphers* (Doktora tezi). KU Leuven.

Rijmen, V., Daemen, J., Preneel, B., Bosselaers, A. & De Win, E. (1996). The cipher SHARK. *International Workshop on Fast Software Encryption*, 99–111.

Rivest, R. & Dusse, S. (1992). The MD5 message-digest algorithm.

Rivest, R. L. (1992). The RC4 Encryption Algorithm. RSA Data Security. *Inc., March*, *12*, 9–2.

Rivest, R. L., Shamir, A. & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, *21*(2), 120–126.

Roth, R. M. & Lempel, A. (1989). On MDS codes via Cauchy matrices. *IEEE transactions on information theory*, *35*(6), 1314–1319.

Roth, R. M. & Seroussi, G. (1985). On generator matrices of MDS codes (corresp.) *IEEE transactions on information theory*, *31*(6), 826–830.

Sajadieh, M., Dakhilalian, M., Mala, H. & Omoomi, B. (2012). On construction of involutory MDS matrices from Vandermonde Matrices in GF(2^q). *Designs, Codes and Cryptography*, *64*(3), 287–308.

Sajadieh, M. & Mousavi, M. (2018). Construction of Lightweight MDS Matrices from Generalized Feistel Structures. *IACR Cryptol. ePrint Arch.*, *2018*, 1072.

Sakallı, M. T., Akleylek, S., Aslan, B., Buluş, E. & Sakallı, F. B. (2014). On the construction of and binary matrices with good implementation properties for lightweight block ciphers and hash functions. *Mathematical Problems in Engineering*, *2014*.

Sakallı, M. T. & Aslan, B. (2014). On the algebraic construction of cryptographically good 32× 32 binary linear transformations. *Journal of Computational and Applied Mathematics*, *259*, 485–494.

Sarkar, S. & Sim, S. M. (2016). A deeper understanding of the XOR count distribution in the context of lightweight cryptography. *International Conference on Cryptology in Africa*, 167–182.

Sarkar, S. & Syed, H. (2016). Lightweight diffusion layer: Importance of toeplitz matrices. *IACR Transactions on Symmetric Cryptology*, 95–113.

Sarkar, S. & Syed, H. (2017). Analysis of Toeplitz MDS matrices. *Australasian Conference on Information Security and Privacy*, 3–18.

Schneier, B. (1993). Description of a new variable-length key, 64-bit block cipher (Blowfish). *International Workshop on Fast Software Encryption*, 191–204.

Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C. & Ferguson, N. (1998). Twofish: a 128-bit block cipher. *AES submission*.

Schnorr, C. P. & Vaudenay, S. (1994). Black box cryptanalysis of hash networks based on multipermutations. *Workshop on the Theory and Application of of Cryptographic Techniques*, 47–57.

Scientific Computing Tools for Python. (2021). 24 Temmuz 2021 tarihinde https://scipy.org adresinden erişildi.

Shannon, C. E. (1949). Communication theory of secrecy systems. *The Bell system technical journal*, *28*(4), 656–715.

Shirai, T. & Shibutani, K. (2003). On the diffusion matrix employed in the Whirlpool hashing function. *NESSIE public report*.

Sim, S. M., Khoo, K., Oggier, F. & Peyrin, T. (2015). Lightweight MDS involution matrices. *International Workshop on Fast Software Encryption*, 471–493.

Stallings, W. (2017). *Cryptography and network security, 7/E*. Pearson Education.

Standard, A. E. (2001). Federal information processing standard (fips) publication 197. *National Bureau of Standards, US Department of Commerce, Washington, DC*.

Standard, D. E. vd. (1999). Data encryption standard. *Federal Information Processing Standards Publication*, 112.

Stinson, D. (2002). Cryptography: Theory and practice. Chapman and Hall, CRC.

Stoffelen, K. (2016). Optimizing s-box implementations for several criteria using SAT solvers. *International Conference on Fast Software Encryption*, 140–160.

Toh, D., Teo, J., Khoo, K. & Sim, S. M. (2018). Lightweight MDS serial-type matrices with minimal fixed XOR count. *International Conference on Cryptology in Africa*, 51–71.

Türk Dil Kurumu: Sözlük. (2021). *Türk Dil Kurumu Sözlükleri*. 4 Mayıs 2021 tarihinde https://sozluk.tdk.gov.tr/ adresinden erişildi.

Vaudenay, S. (1994). On the need for multipermutations: Cryptanalysis of MD4 and SAFER. *International Workshop on Fast Software Encryption*, 286–297.

Webster, A. & Tavares, S. E. (1985). On the design of S-boxes. *Conference on the theory and application of cryptographic techniques*, 523–534.

Wenbo, M. (2004). Modern cryptography: theory and practice. *Yayıncı: Prentice Hall PTR*.

Wu, H. (2004). A new stream cipher HC-256. *International Workshop on Fast Software Encryption*, 226–244.

Youssef, A., Mister, S. & Tavares, S. (1997). On the design of linear transformations for substitution permutation encryption networks. *Workshop on Selected Areas of Cryptography (SAC'96): Workshop Record*, 40–48.

Z'aba, M. R. (2010). *Analysis of linear relationships in block ciphers* (Doktora tezi). Queensland University of Technology.

EKLER

EK A

TEZ ÇALIŞMASINDA KULLANILAN MAGMA KODU

Bu bölümde, 4×4 boyutunda involutif MDS matris temsilcilerini ve bu matris temsilcilerine b_i parametrelerinin uygulanmasıyla oluşturulan 4×4 boyutunda involutif MDS matrislerin tümünün elde edilmesini sağlayan MAGMA kodu verilmektedir.

```
P<z> := PolynomialRing(GF(2));
p := z^4+z+1;
F<x> := ext <GF(2) | p >;

n:=4;
msayisi1:=0;
xorke64say:=0;
xortoplam:=0;
genelxortoplam:=0;
satir1xor:=0;
satir2xor:=0;
satir3xor:=0;
satir4xor:=0;
sifirsay:=0;
birimsay:=0;
ikisay:=0;
ucsay:=0;
dortsay:=0;
bessay:=0;
altisay:=0;
yedisay:=0;
sekizsay:=0;
koc:=0;

say:=0;
birsay:=0;
for a1 in [1,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] do
for k1 in [0,1,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] do
for k2 in [0,1,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] do
for k3 in [0,1,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] do
for k4 in [0,1,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] do
for k5 in [0,1,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] do
for k6 in [0,1,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] do
for k7 in [0,1,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] do

B:=Matrix(F,4,4,[
a1,a1+k3,a1+k4,a1+k3+k4+1,
a1+k7,a1+k1,a1+k5,a1+k1+k5+k7+1,
a1+k3+k4+k5+k6+k7,a1+k6,a1+k2,a1+k2+k3+k4+k5+k7+1,
a1+k3+k4+k5+k6+1,a1+k1+k3+k6+1,a1+k2+k4+k5+1,a1+k1+k2
]);

I := IdentityMatrix(F,4);
C:= LinearCode(HorizontalJoin(I,B));

if (B^2 eq I) then
say:=say+1;
if (MinimumWeight(C) eq 5) then

for i5:=0 to 14 do
for i6:=0 to 14 do
for i7:=0 to 14 do

ab:= x^i5;
ac:= x^i6;
ad:= x^i7;

birsay:=0;
A:=Matrix(F,4,4,[
a1,(a1+k3)*ab,(a1+k4)*ac,(a1+k3+k4+1)*ad,
(a1+k7)*ab^-1,a1+k1,(a1+k5)*ac*ab^-1,(a1+k1+k5+k7+1)*ad*ab^-1,
(a1+k3+k4+k5+k6+k7)*ac^-1,(a1+k6)*ab*ac^-1,a1+k2,(a1+k2+k3+k4+k5+k7+1)
    *ad*ac^-1,
(a1+k3+k4+k5+k6+1)*ad^-1,(a1+k1+k3+k6+1)*ab*ad^-1,(a1+k2+k4+k5+1)*ac*
    ad^-1,a1+k1+k2
]);

for xx:=1 to n do
for yy:=1 to n do

if (A[xx,yy] eq 1) then xortoplam:=xortoplam+0; birsay:=birsay+1;
elif (A[xx,yy] eq x) then xortoplam:=xortoplam+1;
elif (A[xx,yy] eq x^4) then xortoplam:=xortoplam+5;
elif (A[xx,yy] eq x^2) then xortoplam:=xortoplam+2;
elif (A[xx,yy] eq x^8) then xortoplam:=xortoplam+6;
```

```
elif (A[xx,yy] eq x^5) then xortoplam:=xortoplam+5;
elif (A[xx,yy] eq x^10) then xortoplam:=xortoplam+9;
elif (A[xx,yy] eq x^3) then xortoplam:=xortoplam+3;
elif (A[xx,yy] eq x^14) then xortoplam:=xortoplam+1;
elif (A[xx,yy] eq x^9) then xortoplam:=xortoplam+8;
elif (A[xx,yy] eq x^7) then xortoplam:=xortoplam+6;
elif (A[xx,yy] eq x^6) then xortoplam:=xortoplam+5;
elif (A[xx,yy] eq x^13) then xortoplam:=xortoplam+3;
elif (A[xx,yy] eq x^11) then xortoplam:=xortoplam+8;
elif (A[xx,yy] eq x^12) then xortoplam:=xortoplam+6;
end if;

end for;

if (xx eq 1) then
satir1xor:= xortoplam+12;
xortoplam:=0;
end if;

if (xx eq 2) then
satir2xor:= xortoplam+12;
xortoplam:=0;
end if;

if (xx eq 3) then
satir3xor:= xortoplam+12;
xortoplam:=0;
end if;

if (xx eq 4) then
satir4xor:= xortoplam+12;
xortoplam:=0;
end if;

end for;

hamxor:=satir1xor+satir2xor+satir3xor+satir4xor;

K:=[ab,ac,ad];
L:=[satir1xor, satir2xor, satir3xor, satir4xor];

if (hamxor le 80) then PrintFile ("mdsinvolutoryxortoplamke80.txt",B);
     PrintFile ("mdsinvolutoryxortoplamke80.txt","-----------------");
    PrintFile ("mdsinvolutoryxortoplamke80.txt",A);

PrintFile ("mdsinvolutoryxortoplamke80.txt","Toplam xor sayisi:");
   PrintFile ("mdsinvolutoryxortoplamke80.txt",hamxor);
koc:=koc+1;

PrintFile ("mdsinvolutoryxortoplamke80.txt","Katsayilar (ab, ac, ad)
   :");PrintFile ("mdsinvolutoryxortoplamke80.txt",K);

PrintFile ("mdsinvolutoryxortoplamke80.txt","Satir xor toplamlari
   sirasiyla:");PrintFile ("mdsinvolutoryxortoplamke80.txt",L);

PrintFile ("mdsinvolutoryxortoplamke80.txt","1'lerin sayisi:");
```

```
   PrintFile ("mdsinvolutoryxortoplamke80.txt",birsay);
PrintFile ("mdsinvolutoryxortoplamke80.txt","matris sayisi");PrintFile
    ("mdsinvolutoryxortoplamke80.txt",koc);

PrintFile ("mdsinvolutoryxortoplamke80.txt","----------------------");

xorke64say:=xorke64say+1;
end if;


if (birsay eq 0) then
sifirsay:=sifirsay+1;
end if;

if (birsay eq 1) then
birimsay:=birimsay+1;
end if;

if (birsay eq 2) then
ikisay:=ikisay+1;
end if;

if (birsay eq 3) then
ucsay:=ucsay+1;
end if;

if (birsay eq 4) then
dortsay:=dortsay+1;
end if;

if (birsay eq 5) then
bessay:=bessay+1;
end if;

if (birsay eq 6) then
altisay:=altisay+1;
end if;

if (birsay eq 7) then
yedisay:=yedisay+1;
end if;

if (birsay eq 8) then
sekizsay:=sekizsay+1;
end if;


if (birsay ge 9) then PrintFile ("sonuc.txt",B);PrintFile ("sonuc.txt
   ","-----------------"); PrintFile ("sonuc.txt",A);

PrintFile ("sonuc.txt","Toplam xor sayisi:");PrintFile ("sonuc.txt",
   hamxor);
msayisi1:=msayisi1+1;

PrintFile ("sonuc.txt","Katsayilar (ab, ac, ad):");PrintFile ("sonuc.
   txt",K);
```

```
176
177 PrintFile ("sonuc.txt","Satir xor toplamlari sirasiyla:");PrintFile ("
    sonuc.txt",L);
178
179 PrintFile ("sonuc.txt","1'lerin sayisi:");PrintFile ("sonuc.txt",
    birsay);
180
181 PrintFile ("sonuc.txt","matris sayisi");PrintFile ("sonuc.txt",
    msayisi1);
182
183 PrintFile ("sonuc.txt","----------------------");
184 end if;
185
186 end for;
187 end for;
188 end for;
189
190 end if;
191 end if;
192
193 end for;
194 end for;
195 end for;
196 end for;
197 end for;
198 end for;
199 end for;
200 end for;
201
202 PrintFile ("invmdsgf24.txt",say);
203 PrintFile ("genelsonuc.txt","0'lerin sayisi:");PrintFile ("genelsonuc.
    txt",sifirsay);
204 PrintFile ("genelsonuc.txt","1'lerin sayisi:");PrintFile ("genelsonuc.
    txt",birimsay);
205 PrintFile ("genelsonuc.txt","2'lerin sayisi:");PrintFile ("genelsonuc.
    txt",ikisay);
206 PrintFile ("genelsonuc.txt","3'lerin sayisi:");PrintFile ("genelsonuc.
    txt",ucsay);
207 PrintFile ("genelsonuc.txt","4'lerin sayisi:");PrintFile ("genelsonuc.
    txt",dortsay);
208 PrintFile ("genelsonuc.txt","5'lerin sayisi:");PrintFile ("genelsonuc.
    txt",bessay);
209 PrintFile ("genelsonuc.txt","6'lerin sayisi:");PrintFile ("genelsonuc.
    txt",altisay);
210 PrintFile ("genelsonuc.txt","7'lerin sayisi:");PrintFile ("genelsonuc.
    txt",yedisay);
211 PrintFile ("genelsonuc.txt","8'lerin sayisi:");PrintFile ("genelsonuc.
    txt",sekizsay);
```

Verilen kodun $\mathbb{F}_{2^4}$ sonlu cismi üzerinde tanımlı, $0x13$ polinomu için oluşturulduğuna ve ham XOR sayısının $XOR \leq 80$ değeri ile sınırlandırılmış olduğuna dikkat edilmelidir. Örnek kod, sözü edilen tanımlamalar değiştirilerek farklı cisim, polinom ve ham XOR değerleri için de kullanılabilmektedir.

EK B

BP GLOBAL OPTİMİZASYON DETAYLARI

Bu bölümde, Örnek 5.5, 5.6, 5.7 ve 5.10'da verilen M_1, M_2, M_3 ve M_6 matrislerinin BP global optimizasyon sonuçlarının detayları sırasıyla Çizelge B.1, B.2, B.3 ve B.4'de verilmektedir.

Ayrıca,

- $\mathbb{F}_{2^4}/0x13$ üzerinde tanımlı ve BP global optimizasyon sonucu 36 XOR olarak elde edilen 13 adet,
- $\mathbb{F}_{2^8}/0x12d$ üzerinde tanımlı ve BP global optimizasyon sonucu 88 XOR olarak elde edilen 9 adet, 4×4 boyutunda involutif MDS matris de yine bu bölümde listelenmektedir.

BP global optimizasyon sonucu, derinlik 6 değeri için 28 XOR olan M_1 matrisinin optimizasyon detayları aşağıdaki çizelgede verilmektedir. Çizelgede, $x_i : [(x_0, x_1, \cdots, x_{11})]$ giriş bitlerini, $y_i : [(y_0, y_1, \cdots, y_{11})]$ çıkış bitlerini, $t_j : [(t_0, t_1, \cdots, t_{28})]$ geçici değişkenleri göstermekte, parantez içinde verilen sayısal değerler ise devre derinliğini belirtmektedir.

Çizelge B.1. M_1 Matrisinin BP Global Optimizasyon Sonuçları

No	İşlem	No	İşlem	No	İşlem
1	$t_1 = x_2 + x_8\ (1)$	**11**	$t_{11} = t_9 + t_{11}\ [y_{11}]\ (6)$	**21**	$t_{21} = t_{17} + t_{20}\ [y_3]\ (5)$
2	$t_2 = x_1 + x_7\ (1)$	**12**	$t_{12} = x_8 + t_9\ (3)$	**22**	$t_{22} = x_4 + t_1\ (2)$
3	$t_3 = x_3 + x_9\ (1)$	**13**	$t_{13} = x_{11} + t_{12}\ [y_8]\ (4)$	**23**	$t_{23} = t_4 + t_{22}\ [y_6]\ (3)$
4	$t_4 = x_6 + t_3\ (2)$	**14**	$t_{14} = x_5 + t_1\ (2)$	**24**	$t_{24} = t_{21} + t_{22}\ [y_9]\ (6)$
5	$t_5 = x_0 + t_4\ [y_0]\ (3)$	**15**	$t_{15} = x_{11} + t_{14}\ [y_2]\ (3)$	**25**	$t_{25} = x_4 + t_5\ (4)$
6	$t_6 = x_{10} + t_1\ (2)$	**16**	$t_{16} = x_4 + x_{10}\ (1)$	**26**	$t_{26} = t_{12} + t_{25}\ [y_{10}]\ (5)$
7	$t_7 = x_0 + t_6\ (3)$	**17**	$t_{17} = t_2 + t_{16}\ [y_1]\ (2)$	**27**	$t_{27} = t_3 + t_{12}\ (4)$
8	$t_8 = t_3 + t_7\ [y_5]\ (4)$	**18**	$t_{18} = x_7 + t_4\ (3)$	**28**	$t_{28} = t_{14} + t_{27}\ [y_4]\ (5)$
9	$t_9 = x_3 + t_2\ (2)$	**19**	$t_{19} = t_{10} + t_{18}\ [y_7]\ (6)$		
10	$t_{10} = x_5 + t_8\ (5)$	**20**	$t_{20} = x_{11} + t_{18}\ (4)$		

BP global optimizasyon sonucu, derinlik 6 değeri için 28 XOR olan M_2 matrisinin optimizasyon detayları aşağıdaki çizelgede verilmektedir. Çizelgede, $x_i : [(x_0, x_1, \cdots, x_{11})]$ giriş bitlerini, $y_i : [(y_0, y_1, \cdots, y_{11})]$ çıkış bitlerini, $t_j : [(t_0, t_1, \cdots, t_{28})]$ geçici değişkenleri göstermekte, parantez içinde verilen sayısal değerler ise devre derinliğini belirtmektedir.

Çizelge B.2. M_2 Matrisinin BP Global Optimizasyon Sonuçları

No	İşlem	No	İşlem	No	İşlem
1	$t_1 = x_0 + x_9$ (1)	**11**	$t_{11} = t_9 + t_{10}$ [y_3] (6)	**21**	$t_{21} = t_{17} + t_{20}$ [y_8] (5)
2	$t_2 = x_1 + x_{10}$ (1)	**12**	$t_{12} = x_8 + t_9$ (3)	**22**	$t_{22} = x_7 + t_1$ (2)
3	$t_3 = x_5 + x_8$ (1)	**13**	$t_{13} = x_3 + t_{12}$ [y_9] (4)	**23**	$t_{23} = t_{21} + t_{22}$ [y_5] (6)
4	$t_4 = x_{11} + t_3$ (2)	**14**	$t_{14} = x_6 + t_1$ (2)	**24**	$t_{24} = t_4 + t_{22}$ [y_{11}] (3)
5	$t_5 = x_2 + t_4$ [y_2] (3)	**15**	$t_{15} = x_3 + t_{14}$ [y_0] (3)	**25**	$t_{25} = x_7 + t_5$ (4)
6	$t_6 = x_4 + t_1$ (2)	**16**	$t_{16} = x_4 + x_7$ (1)	**26**	$t_{26} = t_{12} + t_{25}$ [y_4] (5)
7	$t_7 = x_2 + t_6$ (3)	**17**	$t_{17} = t_2 + t_{16}$ [y_1] (2)	**27**	$t_{27} = t_3 + t_{12}$ (4)
8	$t_8 = t_3 + t_7$ [y_6] (4)	**18**	$t_{18} = x_{10} + t_4$ (3)	**28**	$t_{28} = t_{14} + t_{27}$ [y_7] (5)
9	$t_9 = x_8 + t_2$ (2)	**19**	$t_{19} = t_{10} + t_{18}$ [y_{10}] (6)		
10	$t_{10} = x_6 + t_8$ (5)	**20**	$t_{20} = x_3 + t_{18}$ (4)		

BP global optimizasyon sonucu, derinlik 5 değeri için 36 XOR olan M_3 matrisinin optimizasyon detayları aşağıdaki çizelgede verilmektedir. Çizelgede, $x_i : [(x_0, x_1, \cdots, x_{15})]$ giriş bitlerini, $y_i : [(y_0, y_1, \cdots, y_{15})]$ çıkış bitlerini, $t_j : [(t_0, t_1, \cdots, t_{36})]$ geçici değişkenleri göstermekte, parantez içinde verilen sayısal değerler ise devre derinliğini belirtmektedir.

Çizelge B.3. M_3 Matrisinin BP Global Optimizasyon Sonuçları

No	İşlem	No	İşlem	No	İşlem
1	$t_1 = x_7 + x_{15}$ (1)	**13**	$t_{13} = t_7 + t_{12}$ [y_{11}] (4)	**25**	$t_{25} = t_{19} + t_{24}$ [y_4] (4)
2	$t_2 = x_4 + x_{12}$ (1)	**14**	$t_{14} = x_9 + t_1$ (2)	**26**	$t_{26} = t_5 + t_{24}$ [y_{10}] (4)
3	$t_3 = x_6 + x_8$ (1)	**15**	$t_{15} = x_2 + t_{14}$ [y_6] (3)	**27**	$t_{27} = x_3 + x_{11}$ (1)
4	$t_4 = x_5 + x_{11}$ (1)	**16**	$t_{16} = x_0 + x_8$ (1)	**28**	$t_{28} = t_8 + t_{10}$ (4)
5	$t_5 = x_3 + x_{14}$ (1)	**17**	$t_{17} = x_4 + t_{16}$ (2)	**29**	$t_{29} = t_{27} + t_{28}$ [y_1] (5)
6	$t_6 = x_2 + x_{13}$ (1)	**18**	$t_{18} = t_{14} + t_{17}$ [y_0] (3)	**30**	$t_{30} = t_{22} + t_{28}$ [y_9] (5)
7	$t_7 = x_1 + t_1$ (2)	**19**	$t_{19} = x_2 + t_2$ (2)	**31**	$t_{31} = x_{15} + t_{27}$ (2)
8	$t_8 = t_4 + t_7$ [y_{15}] (3)	**20**	$t_{20} = t_{18} + t_{19}$ [y_8] (4)	**32**	$t_{32} = t_{13} + t_{31}$ [y_5] (5)
9	$t_9 = x_{10} + t_2$ (2)	**21**	$t_{21} = t_3 + t_{19}$ [y_{12}] (3)	**33**	$t_{33} = x_{10} + t_{12}$ (4)
10	$t_{10} = x_3 + t_9$ [y_7] (3)	**22**	$t_{22} = t_4 + t_6$ (2)	**34**	$t_{34} = t_{17} + t_{33}$ [y_{14}] (5)
11	$t_{11} = x_7 + t_5$ (2)	**23**	$t_{23} = x_6 + t_{22}$ [y_2] (3)	**35**	$t_{35} = t_{14} + t_{23}$ (4)
12	$t_{12} = t_3 + t_{11}$ [y_3] (2)	**24**	$t_{24} = x_0 + t_{22}$ (3)	**36**	$t_{36} = t_{31} + t_{35}$ [y_{13}] (5)

BP global optimizasyon sonucu, derinlik 6 değeri için 88 XOR olan M_6 matrisinin optimizasyon detayları aşağıdaki çizelgede verilmektedir. Çizelgede, $x_i : [(x_0, x_1, \cdots, x_{31})]$ giriş bitlerini, $y_i : [(y_0, y_1, \cdots, y_{31})]$ çıkış bitlerini, $t_j : [(t_0, t_1, \cdots, t_{88})]$ geçici değişkenleri göstermekte, parantez içinde verilen sayısal değerler ise devre derinliğini belirtmektedir.

Çizelge B.4. M_6 Matrisinin BP Global Optimizasyon Sonuçları

No	İşlem	No	İşlem	No	İşlem
1	$t_1 = x_0 + x_{24}$ (1)	**31**	$t_{31} = t_{26} + t_{30}$ [y_9] (5)	**61**	$t_{61} = t_{58} + t_{60}$ [y_{15}] (3)
2	$t_2 = x_9 + x_{16}$ (1)	**32**	$t_{32} = t_6 + t_{30}$ (4)	**62**	$t_{62} = x_{12} + t_{53}$ (3)
3	$t_3 = x_{12} + x_{31}$ (1)	**33**	$t_{33} = x_{18} + t_{32}$ [y_{18}] (5)	**63**	$t_{63} = t_{60} + t_{62}$ [y_{12}] (4)
4	$t_4 = x_{15} + x_{29}$ (1)	**34**	$t_{34} = t_{19} + t_{23}$ (2)	**64**	$t_{64} = x_{17} + t_9$ (3)
5	$t_5 = x_{26} + t_3$ (2)	**35**	$t_{35} = x_{13} + t_{34}$ [y_{13}] (3)	**65**	$t_{65} = t_{16} + t_{64}$ [y_{17}] (4)
6	$t_6 = x_{14} + x_{28}$ (1)	**36**	$t_{36} = x_3 + x_{13}$ (1)	**66**	$t_{66} = x_{19} + t_5$ (3)
7	$t_7 = x_{10} + t_4$ (2)	**37**	$t_{37} = t_{12} + t_{36}$ (4)	**67**	$t_{67} = t_{37} + t_{66}$ [y_{19}] (5)
8	$t_8 = x_8 + x_{13}$ (1)	**38**	$t_{38} = x_0 + t_{24}$ (2)	**68**	$t_{68} = x_{23} + t_{48}$ (4)
9	$t_9 = x_{27} + t_8$ (2)	**39**	$t_{39} = t_{12} + t_{38}$ [y_0] (4)	**69**	$t_{69} = t_{51} + t_{68}$ [y_{11}] (5)
10	$t_{10} = x_{11} + x_{25}$ (1)	**40**	$t_{40} = x_6 + t_{38}$ (3)	**70**	$t_{70} = t_{16} + t_{22}$ (4)
11	$t_{11} = x_{30} + t_{10}$ (2)	**41**	$t_{41} = t_1 + t_{40}$ [y_{24}] (4)	**71**	$t_{71} = t_{68} + t_{70}$ [y_1] (5)
12	$t_{12} = x_{24} + t_7$ (3)	**42**	$t_{42} = t_{20} + t_{37}$ (5)	**72**	$t_{72} = x_{27} + t_{34}$ (3)
13	$t_{13} = x_7 + t_2$ (2)	**43**	$t_{43} = t_{35} + t_{42}$ [y_3] (6)	**73**	$t_{73} = t_{70} + t_{72}$ [y_{27}] (5)
14	$t_{14} = t_6 + t_{13}$ [y_7] (3)	**44**	$t_{44} = x_4 + x_{14}$ (1)	**74**	$t_{74} = x_{29} + t_{17}$ (2)
15	$t_{15} = x_7 + x_{21}$ (1)	**45**	$t_{45} = t_{10} + t_{44}$ (2)	**75**	$t_{75} = t_{36} + t_{74}$ [y_{29}] (3)
16	$t_{16} = x_1 + x_{25}$ (1)	**46**	$t_{46} = t_{22} + t_{45}$ [y_{30}] (4)	**76**	$t_{76} = x_{30} + t_{28}$ (4)
17	$t_{17} = x_6 + x_{22}$ (1)	**47**	$t_{47} = x_{18} + x_{25}$ (1)	**77**	$t_{77} = t_{51} + t_{76}$ [y_4] (5)
18	$t_{18} = t_{12} + t_{17}$ [y_{22}] (4)	**48**	$t_{48} = t_{13} + t_{47}$ [y_{25}] (3)	**78**	$t_{78} = t_2 + t_{28}$ (4)
19	$t_{19} = x_6 + x_{23}$ (1)	**49**	$t_{49} = x_{23} + t_{16}$ (2)	**79**	$t_{79} = t_{32} + t_{78}$ [y_{28}] (5)
20	$t_{20} = t_9 + t_{19}$ [y_6] (3)	**50**	$t_{50} = x_8 + t_{49}$ [y_8] (3)	**80**	$t_{80} = t_4 + t_{40}$ (4)
21	$t_{21} = x_7 + x_{23}$ (1)	**51**	$t_{51} = t_{14} + t_{45}$ (4)	**81**	$t_{81} = t_{42} + t_{80}$ [y_{10}] (6)
22	$t_{22} = t_{11} + t_{21}$ [y_{23}] (3)	**52**	$t_{52} = x_5 + x_{15}$ (1)	**82**	$t_{82} = t_9 + t_{23}$ (3)
23	$t_{23} = x_{20} + x_{30}$ (1)	**53**	$t_{53} = x_{19} + t_4$ (2)	**83**	$t_{83} = t_{45} + t_{82}$ [y_{20}] (4)
24	$t_{24} = x_8 + x_{17}$ (1)	**54**	$t_{54} = x_{16} + t_{52}$ (2)	**84**	$t_{84} = t_{14} + t_{15}$ (4)
25	$t_{25} = x_{16} + t_5$ (3)	**55**	$t_{55} = x_{31} + t_{54}$ [y_{31}] (3)	**85**	$t_{85} = t_{54} + t_{84}$ [y_{21}] (5)
26	$t_{26} = t_1 + t_{25}$ [y_{16}] (4)	**56**	$t_{56} = t_{25} + t_{53}$ (4)	**86**	$t_{86} = t_3 + t_{39}$ (5)
27	$t_{27} = x_{31} + t_{15}$ (2)	**57**	$t_{57} = x_2 + t_{56}$ [y_2] (5)	**87**	$t_{87} = t_{41} + t_{56}$ (5)
28	$t_{28} = x_{14} + t_{27}$ [y_{14}] (3)	**58**	$t_{58} = x_5 + x_{22}$ (1)	**88**	$t_{88} = t_{86} + t_{87}$ [y_{26}] (6)
29	$t_{29} = x_2 + t_3$ (2)	**59**	$t_{59} = t_5 + t_{58}$ [y_5] (3)		
30	$t_{30} = x_9 + t_{29}$ (3)	**60**	$t_{60} = t_1 + t_{52}$ (2)		

Son olarak Çizelge B.5, $\mathbb{F}_{2^4}$ sonlu cismi üzerinde tanımlı, $0x13$ polinomunun bir kökü α olmak üzere BP global optimizasyon sonucu 36 XOR olarak elde edilen 13 adet, Çizelge B.6 ise $\mathbb{F}_{2^8}$ sonlu cismi üzerinde tanımlı, $0x12d$ polinomunun bir kökü α olmak üzere BP global optimizasyon sonucu 88 XOR olarak elde edilen 9 adet, 4×4 boyutlarında involutif MDS matrisi göstermektedir.

Çizelge B.5. $\mathbb{F}_{2^4}/0x13$ Üzerinde BP Optimizasyon Sonucu 36 XOR Olan Matrisler

İlk Eleman 1

$$\begin{bmatrix} 1 & 1 & \alpha^{14} & 1 \\ 1 & \alpha^2 & 1 & \alpha^{14} \\ \alpha^8 & \alpha & \alpha^2 & 1 \\ \alpha^{14} & \alpha^8 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & \alpha^2 & \alpha^{14} & 1 \\ 1 & 1 & 1 & \alpha^{14} \\ \alpha^8 & \alpha & 1 & \alpha^2 \\ \alpha^{14} & \alpha^8 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & \alpha^8 & \alpha^2 & \alpha \\ \alpha^{14} & 1 & 1 & \alpha^2 \\ 1 & \alpha^{14} & 1 & \alpha^8 \\ 1 & 1 & \alpha^{14} & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \alpha^{14} & 1 & 1 \\ \alpha^8 & \alpha^2 & \alpha & 1 \\ 1 & 1 & \alpha^2 & \alpha^{14} \\ \alpha^{14} & 1 & \alpha^8 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & \alpha^{14} & \alpha^2 & 1 \\ \alpha^8 & 1 & \alpha & \alpha^2 \\ 1 & 1 & 1 & \alpha^{14} \\ \alpha^{14} & 1 & \alpha^8 & 1 \end{bmatrix}$$

İlk Eleman α^4

$$\begin{bmatrix} \alpha^4 & 1 & \alpha^{14} & \alpha \\ \alpha^{14} & 1 & \alpha & \alpha^{14} \\ \alpha^8 & \alpha^2 & 1 & 1 \\ \alpha & \alpha^8 & \alpha^{14} & \alpha^4 \end{bmatrix} \quad \begin{bmatrix} \alpha^4 & \alpha^{14} & 1 & \alpha \\ \alpha^8 & 1 & \alpha^2 & 1 \\ \alpha^{14} & \alpha & 1 & \alpha^{14} \\ \alpha & \alpha^{14} & \alpha^8 & \alpha^4 \end{bmatrix}$$

İlk Eleman α^8

$$\begin{bmatrix} \alpha^8 & 1 & \alpha^2 & \alpha \\ 1 & \alpha^{14} & 1 & \alpha^2 \\ 1 & 1 & \alpha^{14} & 1 \\ \alpha^{14} & 1 & 1 & \alpha^8 \end{bmatrix} \quad \begin{bmatrix} \alpha^8 & \alpha^2 & 1 & \alpha \\ 1 & \alpha^{14} & 1 & 1 \\ 1 & 1 & \alpha^{14} & \alpha^2 \\ \alpha^{14} & 1 & 1 & \alpha^8 \end{bmatrix} \quad \begin{bmatrix} \alpha^8 & \alpha^{14} & \alpha^4 & \alpha \\ 1 & \alpha^{14} & \alpha & \alpha^4 \\ 1 & \alpha & \alpha^{14} & \alpha^{14} \\ \alpha^2 & 1 & 1 & \alpha^8 \end{bmatrix}$$

İlk Eleman α^{14}

$$\begin{bmatrix} \alpha^{14} & 1 & \alpha^2 & 1 \\ 1 & \alpha^8 & \alpha & \alpha^2 \\ 1 & \alpha^{14} & \alpha^8 & 1 \\ 1 & 1 & 1 & \alpha^{14} \end{bmatrix} \quad \begin{bmatrix} \alpha^{14} & 1 & \alpha^4 & \alpha \\ \alpha^{14} & \alpha^8 & \alpha & \alpha^4 \\ 1 & \alpha^2 & \alpha^8 & 1 \\ \alpha & 1 & \alpha^{14} & \alpha^{14} \end{bmatrix} \quad \begin{bmatrix} \alpha^{14} & \alpha^4 & 1 & \alpha \\ 1 & \alpha^8 & \alpha^2 & 1 \\ \alpha^{14} & \alpha & \alpha^8 & \alpha^4 \\ \alpha & \alpha^{14} & 1 & \alpha^{14} \end{bmatrix}$$

Çizelge B.6. $\mathbb{F}_{2^8}/0x12d$ Üzerinde BP Optimizasyon Sonucu 88 XOR Olan Matrisler

$\begin{bmatrix} 1 & \alpha^{39} & 1 & \alpha^{37} \\ \alpha & 1 & \alpha^{254} & 1 \\ \alpha^{254} & \alpha^{36} & 1 & \alpha^{39} \\ \alpha^{253} & \alpha^{254} & \alpha & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \alpha^{36} & \alpha^{254} & \alpha^{38} \\ \alpha^{254} & 1 & \alpha & \alpha^{254} \\ 1 & \alpha^{39} & 1 & \alpha^{36} \\ \alpha^{2} & 1 & \alpha^{254} & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & \alpha^{39} & \alpha^{37} \\ \alpha^{254} & 1 & \alpha^{36} & \alpha^{39} \\ \alpha & \alpha^{254} & 1 & 1 \\ \alpha^{253} & \alpha & \alpha^{254} & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & \alpha^{253} & 1 & \alpha^{36} \\ \alpha^{254} & 1 & \alpha^{37} & 1 \\ \alpha^{219} & 1 & 1 & \alpha^{253} \\ \alpha & \alpha^{219} & \alpha^{254} & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \alpha^{252} & \alpha^{219} & \alpha^{254} \\ 1 & 1 & \alpha^{2} & \alpha^{219} \\ 1 & \alpha^{35} & 1 & \alpha^{252} \\ \alpha^{38} & 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \alpha^{252} & 1 & \alpha^{35} \\ 1 & 1 & \alpha^{38} & 1 \\ \alpha^{219} & \alpha^{254} & 1 & \alpha^{252} \\ \alpha^{2} & \alpha^{219} & 1 & 1 \end{bmatrix}$
$\begin{bmatrix} 1 & 1 & \alpha^{252} & \alpha^{35} \\ \alpha^{219} & 1 & \alpha^{254} & \alpha^{252} \\ 1 & \alpha^{38} & 1 & 1 \\ \alpha^{2} & 1 & \alpha^{219} & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & \alpha^{253} & \alpha^{36} \\ \alpha^{219} & 1 & 1 & \alpha^{253} \\ \alpha^{254} & \alpha^{37} & 1 & 1 \\ \alpha & \alpha^{254} & \alpha^{219} & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & \alpha^{219} & \alpha^{252} & \alpha^{254} \\ 1 & 1 & \alpha^{35} & \alpha^{252} \\ 1 & \alpha^{2} & 1 & \alpha^{219} \\ \alpha^{38} & 1 & 1 & 1 \end{bmatrix}$

ÖZGEÇMİŞ

İlk, orta ve lise öğrenimimi İstanbul'da tamamladıktan sonra, 1997 yılında Trakya Üniversitesi Fen – Edebiyat Fakültesi Kimya bölümüne başladı. 2002 yılında mezun olduktan sonra İstanbul'da özel bir ilaç firmasında çalıştı. 2003 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Kimya Öğretmenliği bölümünde tezsiz yüksek lisansa ve eş zamanlı olarak özel bir dershanede kimya öğretmenliğine başladı. 2007 yılında Sakarya Üniversitesi Adapazarı Meslek Yüksekokulu Bilgisayar Programcılığı bölümünü kazandı. 2009'da mezun olduktan sonra dershaneden ayrılarak Trakya Üniversitesi Teknik Bilimler Meslek Yüksekokulu'nda serbest öğretim elemanı olarak ders vermeye başladı. 2010'da dikey geçiş sınavına girerek Trakya Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünü kazandı. 2013 yılında bu bölümden mezun olduktan sonra, 2015 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Hesaplamalı Bilimler Anabilim Dalı Yüksek Lisans programını tamamladı. 2016 yılında başladığı Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Doktora programında halen öğrenimine devam etmektedir.

TEZ SIRASINDA GERÇEKLEŞTİRİLEN BİLİMSEL FAALİYETLER

Makale

1. F.B. Sakallı, Ö. Aydın, G. Tuncay, M.K. Pehlivanoğlu, G.G. Güzel, M.T. Sakallı, On Lightweight 4×4 MDS Matrices over Binary Field Extensions, International Journal of Information Security Science 9 (2) (2020) 94-103 (Haziran 2020)

2. A New Hybrid Method to Generate All 4×4 Involutory MDS Matrices over Binary Field Extensions (Yazım aşamasında)