T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KRİPTOGRAFİK YAYILIM TABAKALARINDA OPTİMİZASYON TEKNİKLERİ

NİHAL TAN KAÇAR

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Prof. Dr. Muharrem Tolga SAKALLI

EDİRNE-2023

NİHAL TAN KAÇAR'ın hazırladığı "**KRİPTOGRAFİK YAYILIM TABAKALARINDA OPTİMİZASYON TEKNİKLERİ**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **BİLGİSAYAR MÜHENDİSLİĞİ** Anabilim Dalında bir **Yüksek lisans tezi** olarak kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. M. Tolga SAKALLI	……
Prof. Dr. Erdinç UZUN	……
Dr. Öğr. Üyesi Andaç ŞAHİN MESUT	……

Tez Savunma Tarihi: 18.01.2023

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Prof. Dr. Muharrem Tolga SAKALLI
Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

…….
Prof. Dr. Hüseyin Rıza Ferhat KARABULUT
Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS 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.

18 / 01 / 2023

Nihal TAN KAÇAR

Yüksek Lisans Tezi

Kriptografik Yayılım Tabakalarında Optimizasyon Teknikleri

T.Ü. Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Simetrik şifreleme, şifreleme ve şifre çözme evrelerinde aynı anahtarın kullanıldığı şifreleme algoritmalarını içeren bir şifreleme yöntemidir. Bu şifreleme tekniği grubuna blok şifreler ve akış şifreler girmektedir. Blok şifreler büyük verilerin hızlı bir şekilde şifrelenmesini sağlayan ve bloklar şeklinde şifreleme yapan algoritmalardır. Akış şifreler ise veriyi karakter tabanlı (byte veya bit) olarak şifreler. Son yıllarda hafif sıklet algoritmaların geliştirilmesi büyük bir önem kazanmıştır. Bu bağlamda hafifsıklet blok şifrelerin tasarımı ve bu algoritmaların kullandığı yapıların olabildiğince düşük maliyete sahip olması literatürde önemli bir araştırma konusudur.

Bir blok şifrede kullanılan temel yapılar doğrusal olmayan yapılar (yer değiştirme katmanı) ve doğrusal yapılar (yayılım katmanı) şeklindedir. Bu tür şifrelerde maliyet daha çok şifrede kullanılan yayılım tabakası ile ilgilidir. Bu tez çalışması özellikle son yıllarda ilgi çeken ve gelişme gösteren bu tabakalarda (Maksimum Uzaklıkta Ayrılabilen Matrisler (Maximum Distance Separable Matrisler-MDS) ve Maksimum Uzaklıkta Ayrılabilen İkili Matrisler (Maximum Distance Binary Linear-MDBL) kullanılan optimizasyon teknikleri üzerinedir. Bu çalışmanın düşük maliyetli kriptografik yapıların geliştirilmesine katkı sağlayacağı düşünülmektedir.

Yıl : 2023
Sayfa Sayısı : 74
Anahtar Kelimeler : Kriptografi, MDS Matrisler, MDBL Matrisler, Optimizasyon Teknikleri, XOR Sayısı

Master Thesis

Optimization Techniques in Cryptographic Diffusion Layers

Trakya University Institute of Natural and Applied Sciences

Department of Computer Engineering

ABSTRACT

Symmetric encryption is an encryption method including encryption algorithms in which the same key is used in encryption and decryption phases. Block ciphers and stream ciphers fall into the group of symmetric encryption techniques. Block ciphers provide fast encryption of big data and encrypt in blocks. Stream ciphers, on the other hand, encrypt data character-wise (byte or bit). The development of light weight algorithms has gained a great importance in recent years. In this context, the design of light weight block ciphers and the low cost of the structures used by these algorithms are an important research subject in the literature.

The basic components used in a block cipher are nonlinear structures (confusion layer) and linear structures (diffusion layer). In block ciphers, the cost of implementation is more related to the diffusion layer used in their round function. This thesis is on the optimization techniques used in diffusion layers (Maximum Distance Separable (MDS) matrices and Maximum Distance Binary Linear (MDBL) matrices), which have attracted attention and development in recent years. We believe that this study will contribute to the development of low-cost cryptographic components.

Year : 2023

Number of Pages : 74

Keywords : Cryptography, MDS matrices, MDBL matrices, Optimization techniques, XOR Count

TEŞEKKÜR

Tez çalışmam sırasında bilgisi, tecrübesi ile bana yol gösteren ve daima destek olan, gerek lisans gerek lisansüstü eğitimim süresince öğrencisi olmaktan gurur duyduğum danışmanım Sayın Prof. Dr. Muharrem Tolga SAKALLI'ya,

Çalışma boyunca yapılan deneysel bulguların elde edilmesinde sonsuz destek veren, her sorumu sabırla yanıtlayan sayın Dr. Öğr. Üyesi Gökhan TUNCAY'a, Yüksek Lisans yapmam konusunda beni cesaretlendiren ablam Nilgün TAN TABAKOĞLU'na,

Tez yazım süresince yaşından büyük bir olgunlukla bana anlayış gösteren kızım Nermin Naz KAÇAR 'a ve her zaman desteğini hissettiğim eşim Şahin KAÇAR 'a

Saygı ve teşekkürlerimi sunarım.

Nihal TAN KAÇAR

İÇİNDEKİLER

ÖZET iv
ABSTRACT v
TEŞEKKÜR vi
İÇİNDEKİLER vii
SİMGELER DİZİNİ ix
KISALTMALAR DİZİNİ x
ÇİZELGELER DİZİNİ xi
ŞEKİLLLER DİZİNİ xii
1. GİRİŞ 1
1.1 Kriptoloji 1
1.2 Kriptoanaliz 2
1.2.1 Doğrusal Kriptoanaliz 2
1.2.2 Diferansiyel Kriptoanaliz 2
1.3 Kriptografi 3
1.4 Kriptografide Kullanılan Algoritmalar 3
1.4.1 Asimetrik Şifreleme Algoritmaları 4
1.4.2 Simetrik Şifreleme Algoritmaları 4
1.5 Tezin Ana Hatları, Gerekçesi ve Önemi 11
2. MATEMATİKSEL ALTYAPI 13
3. MATRİS TASARIM TEKNİKLERİ 20
3.1 Maksimum Uzaklıkta Ayrılabilen (MDS) Matrisler 20
3.1.1 Circulant (Dairesel) Matrisler 21
3.1.2 Toeplitz Matrisler 22
3.1.3 Hadamard Matrisler 22
3.1.4 GHadamard Matrisler 23
3.1.5 Maximum Distance Binary Linear (MDBL) Matrisler 23
3.1.6 MDBL Matrislerin Tasarım Yöntemleri 25

3.1.7 Hadamard ve Dairesel Matrisler Gibi Özel Matrislerden ve Bu Matrislerin Kombinasyonlarından Yararlanılarak MDBL Matrislerin Tasarımı 26
3.1.8 Devirli Matris Gruplarından Yararlanılarak MDBL Matrislerin Tasarımı 27
3.1.9 Cebirsel Yöntemlerle MDBL Matrislerin Tasarımı 28
3.2 XOR Sayısı..... 29
3.3 Derinlik 30
4. OPTİMİZASYON TEKNİKLERİ 31
4.1 Yerel Optimizasyon 31
4.2 Global Optimizasyon 32
4.2.1 Shortest Linear Straight Line Program-SLP 32
4.2.2 Paar Algoritması..... 34
4.2.3 Boyar-Peralta (BP) Algoritması 35
5. METOT, MATERYALVE DENEYSEL BULGULAR..... 39
5.1 MDBL Matrislere Ait Deneysel Bulgular..... 40
6. DEĞERLENDİRME VE SONUÇ 57
KAYNAKLAR..... 59
ÖZGEÇMİŞ..... 62

SİMGELER DİZİNİ

β	Dal Sayısı
$\mathbb{F}_2/GF(2)$	Elemanları 0 ve 1 olan sonlu cisim
$\mathbb{F}_{2^m}$	Elemanları 0 ve 1 olan m elemanlı vektörleri işaret eden sonlu cisim
$\oplus$	Özel VEYA (eXclusive OR)
wt	Hamming Ağırlığı
$\mathbb{Z}$	Tam Sayılar Kümesi
$\mathbb{Z}^+$	Pozitif Tam sayılar Kümesi
$+$	Herhangi bir sonlu cisim üzerinde toplama işlemi

KISALTMALAR DİZİNİ

AES	Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
BP	Boyar-Peralta Algoritması
Circ	Dairesel Matris
DES	Data Encryption Standard (Veri Şifreleme Standardı)
GF	GaloisField (Galois Cismi)
Ghad	Genel Hadamard Matris
Had	Hadamard Matris
MDBL	Maximum DistanceBinaryLinearCodes (Maksimum Uzaklıkta İkili Doğrusal Kodlar)
MDS	Maximum DistanceSeparable (Maksimum Uzaklıkta Ayrılabilen)
S-box	Substitution-box (Yer Değiştirme Kutusu)
SHA	SecureHashAlgorithm (Güvenli Özet Algoritması)
SLP	ShortestLinear Program (En Kısa Doğrusal Yol Programı)
SPN	Substitution–Permutation Network (Yer Değiştirme Permütasyon Ağı)
XOR	Exclusive OR (Özel Veya)

ÇİZELGELER DİZİNİ

Çizelge 3. 1. İkili Kodların Maksimum Dal Sayıları 24
Çizelge 5. 1.Literatürde Yer Alan Sonuçlarla, Dal Sayısı 8, 16×16 Boyutlu MDBL M Matrisinin Farklı Derinlik Değerlerine Ait XOR Sayılarının Karşılaştırılması 53
Çizelge 5. 3. 16×16 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası Bazı XOR Değerleri (Hadamard) 53
Çizelge 5. 4. 16×16Boyutlu MDBL Matrislere Ait Bazı XOR Değerleri (GHadamard) 54
Çizelge 5. 5. 20×20 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası XOR Değerleri (Dairesel) 55
Çizelge 5. 6. 20×20 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası XOR Değerleri (Toeplitz) 55
Çizelge 5. 7. 24×24 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası XOR Değerleri 56

ŞEKİLLLER DİZİNİ

Şekil 1.1. Kriptolojiyi Oluşturan Alt Bilimler 2
Şekil 1. 2. Şifreleme Algoritmaları 3
Şekil 1. 3. Şifreleme ve Şifre Çözme 4
Şekil 1. 4. Şifreleme Algoritmaları 5
Şekil 1. 5. SPN Ağının Bir Döngü Şeması 7
Şekil 1. 6. ARIA Şifresinde Kullanılan 16 x 16 Boyutunda İkili Matris 11
Şekil 3. 1. MDS Matris Tasarım Yöntemleri (Tuncay, 2021) 21
Şekil 3. 2. MDBL Matris Tasarım Yöntemleri 26
Şekil 5. 1. 4×4 boyutlu A ve 16×16 Boyutlu Dal Sayısı 5 Olan $A1B$ Matrislerine Ait MAGMA Çıktısı 41
Şekil 5. 2. Test Ortamı XOR Ölçütü Hesaplama ve Optimizasyon Yazılımı 42
Şekil 5. 3. Test Ortamına eklenen MAGMA Kodu ve İlgili Alanların Düzenlenmesi 43
Şekil 5. 4. Test Ortamında Yapılan Düzenleme Sonrası BP Algoritmasına Uygun Hale Gelen $A1B$ Matrisi 44
Şekil 5. 5. 16×16 Boyutlu Involutif MDBL $A1B$ Matrisine Ait Ham XOR Sonucunun Test Ortamında Gösterimi 45
Şekil 5. 6. Ham XOR Sayıları Hesaplanan Matrisler ve Ham XOR Değerleri 46
Şekil 5. 7. 16×16 Boyutlu Involutif MDBL $A1B$ Matrisine Ait Paar Optimizasyonu Sonucunun Test Ortamında Gösterimi 47
Şekil 5. 8. Ham XOR Sayıları Hesaplanan Matrisler ve Paar Optimizasyonu Sonrası XOR Sayıları 48
Şekil 5. 9. 16×16 Boyutlu Involutif MDBL $A1B$ Matrisine Ait BP Optimizasyonu Sonucunun Test Ortamında Gösterimi 49
Şekil 5. 10. 16×16 Boyutlu Involutif MDBL $A1B$ Matrisine Ait BP Optimizasyonu Detayları 51

BÖLÜM 1

1. GİRİŞ

Günümüze kadar olan süreçte her zaman değerli bir olgu olan bilgi, yazının keşfi ile iletilebilir ve saklanabilir hale gelmiştir. Bilginin saklanabilir ve iletilebilir olması gelişen medeniyetlerde gizlilik ve bilginin korunmasına ilişkin bir ihtiyacı da beraberinde getirmiştir.

İçinde bulunduğumuz dönemde bilgisayar kullanımının ve bilişim teknolojilerinin gerek dünyada gerekse ülkemizde kullanımının giderek yaygınlaşması bilginin gizliliğinin sağlanması ve güvenli şekilde iletilmesine ilişkin yöntemlerin geliştirilmesini gerekli hale getirmiş ve bu alanda yapılan çalışmaların önemini ortaya çıkarmıştır. (Internet World Stats, 2022)

Modern kriptografi, 1970'li yıllarda açık anahtarlı bir şifreleme yöntemi geliştirilmesi fikriyle ortaya çıkmıştır (Diffie & Hellman, 1976). Bu fikri 1978 yılında IBM tarafından geliştirilen gizli anahtarlı şifreleme algoritması olan Data Encryption Standard (DES) izlemiştir. Modern kriptografideki önemli yapılar/kavramlar arasında asimetrik şifreleme (açık anahtarlı), simetrik şifreleme (gizli anahtarlı), özet fonksiyonları (örneğin Secure Hash Algorithm (SHA)), e-imzalar (Furth vd., 2005) ve kripto paralar bulunmaktadır.

1.1 Kriptoloji

Kriptoloji; güvenilir olmayan iletişim ağları üzerinden bilgiyi, iletilmek istenen yere güvenli bir şekilde iletebilmesini sağlayan yöntemler ile ilgilidir. Kriptoloji bilimi iki alt dalın birleşiminden oluşur. Bu alt dallar, verinin matematiksel algoritmalar aracılığı ile anlaşılamaz şekilde değiştirilmesine olanak sağlayan kriptografi bilimi ve değiştirilmiş metin üzerinde uygulanmış olan algoritmaların zayıf noktalarının yakalanmasıyla tekrar anlaşılır hale getirilmesi olarak tanımlanan ve kriptografik yapıların kırılması anlamına da gelen kriptoanaliz bilimidir. Şekil 1.1'de Kriptolojinin ana bileşenleri gösterilmiştir.



Şekil 1.1. Kriptolojiyi Oluşturan Alt Bilimler

1.2 Kriptoanaliz

Kriptografik yapıların saf dışı bırakılmasıyla açık metne ulaşmayı amaçlayan bir şifre kırma bilimidir. Kriptanalizde kullanılan yöntemler şifrenin gücünü test etmek için kullanılabileceği gibi açık metni ele geçirmek amacıyla art niyetli kişilerce de kullanılabilmektedir. Şifrenin gücü test edilmek istendiğinde "Geniş anahtar arama saldırısı" yöntemi diğer saldırıların başarımı için de referans alınmaktadır. Bu yöntemde *k* bit anahtarlı bir şifre için olası tüm anahtarların (2^k) denenmesi ve ardından anlamlı bir açık metne ulaşılıp ulaşılamamasına bakılarak anahtara ulaşma yöntemidir (Pehlivanoğlu, 2018).

1.2.1 Doğrusal Kriptoanaliz

Kerckhoff'un prensibi (Kerckhoffs, 1883) gereği saldıranın şifreleme sistemi hakkında bilgisinin olduğu ve belli miktarda açık metin-şifreli metin çiftlerine sahip olduğunun farz edildiği blok şifrenin yer değiştirme kutularının (S-kutusu) (Substitution Boxes-S-box) doğrusal ifadelere dönüştürüp anahtar bitlerini ele geçirmeyi amaçlayan kriptografik saldırı türüdür. Şifreye saldırıların zorlaştırılması için şifre tasarımı, aktif S-kutularının sayısının arttırılmasına ve kullanılan S-kutularının doğrusal sapmalarının minimum olmasına yönelik yapılmalıdır (Pehlivanoğlu, 2018).

1.2.2 Diferansiyel Kriptoanaliz

Doğrusal Kriptoanaliz saldırısında da ifade edildiği gibi Kerckhoff'un prensibi gereği şifreleme sistemi hakkında bilgi sahibi olan saldırganın birden fazla açık ve şifreli metnin üretildiği ve üretilen açık metin ikililerinden özel farkların, şifreli metin ikililerinde oluşturduğu etkileri/farkları incelediği saldırı türüdür. Bu inceleme sonucunda görülen farklara göre mümkün olabilecek tüm anahtarların ihtimallerini bulunur ve bunları deneyerek ihtimali en yüksek anahtarı/anahtarları şifrenin gizli anahtarı olarak kabul edilir.

Bu saldırı türüne karşı koyabilmek için tıpkı doğrusal kriptanalizde olduğu gibi diferansiyel kriptanalizde de aktif S-kutusu miktarı olabildiğince fazla olmalıdır. Fakat burada bakılması gereken diğer durum S-kutularının diferansiyel sapmalarının yine mümkün olduğunca minimum olması gerektiğidir (Pehlivanoğlu, 2018).

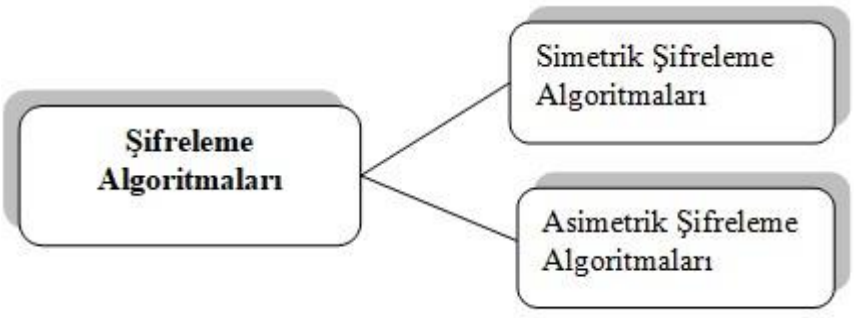
1.3 Kriptografi

Bilgiyi güvenli şekilde iletmek için sıkça başvurulan iki yöntem vardır. Bunlardan ilki nesnelere verinin gizlenmesi ile uygulanan stenografi diğeri ise gönderilen mesajın içerdiği bilgiyi anlaşılamaz şekilde değiştirme yöntemine başvuran kriptografidir.

Kriptografi, veriyi değiştirme ve dolayısıyla verinin güvenli şekilde iletilmesini sağlama döngüsünü çeşitli algoritmalar yardımı ile gerçekleştirir. Bu algoritmaların kullanımı milattan önceki (M.Ö.) yıllara dayansa da o dönemlerde kullanılan algoritmalar bugünkü teknolojiye bakıldığında çok yetersiz kalmıştır. Bu nedenle bir zorunluluk olarak ortaya yeni algoritmalar çıkmıştır.

1.4 Kriptografide Kullanılan Algoritmalar

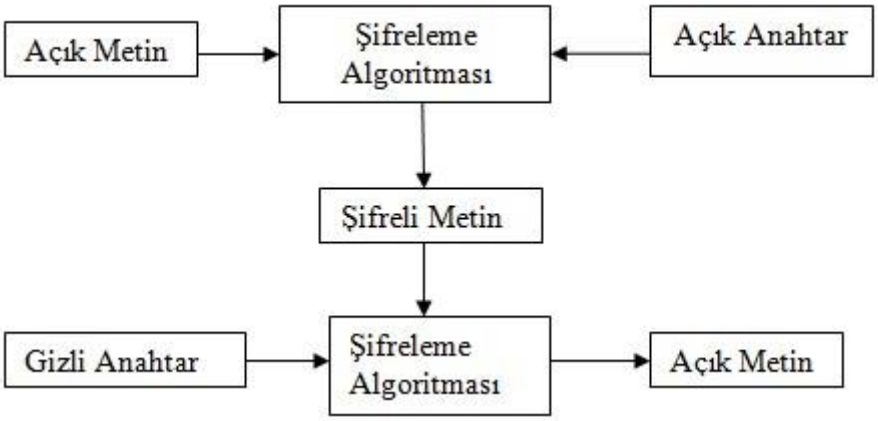
Kriptografide şifreleme ve şifre çözme işlemleri uygulanırken verinin saldırganlar tarafından elde edilmesini önlemek ve verinin güvenliği açısından oluşabilecek riskleri en aza indirmek amacıyla şifreleme algoritmaları geliştirilmiştir. Şifreleme algoritmalarında anahtar kullanılması ise bu riski en aza indirmede önemli bir etken olmuştur. Anahtar kullanımının, saldırganların kullanılan anahtarı elde etmeden şifreli metnin çözülmesini zorlaştırması, güvenli bilgi akışına olanak vermiştir. Algoritmalarda şifreleme ve şifre çözme işlemlerinde kullanılan anahtara göre şifreleme algoritmaları da çeşitlilik göstermektedir. Kullanılan anahtar çeşidine göre algoritmalar simetrik ve asimetrik şifreleme algoritmaları Şekil 1. 2'deki gibi ikiye ayrılırlar.



Şekil 1. 2. Şifreleme Algoritmaları

1.4.1 Asimetrik Şifreleme Algoritmaları

Diffie ve Helman (1976) tarafından bulunan ve modern şifreleme yöntemlerinden biri olan, açık anahtarlı şifreleme algoritmaları da denilen asimetrik şifreleme algoritmalarında şifreleme ve şifre çözme işlemleri sırasında farklı iki anahtar kullanılır. Anahtarların üretimini sağlayan algoritmalar her bir kullanıcı için birbirinden farklı açık ve özel anahtarlar üretirler (Kodaz & Botsalı, 2010). Asimetrik şifreleme algoritmaları simetrik şifreleme algoritmalarına göre daha yavaştırlar. Bu yavaşlığın nedenleri asimetrik şifreleme algoritmalarında (örneğin RSA), açık ve özel anahtarların üretilmesinde kullanılan asal sayıların çok büyük olması ve asimetrik şifreleme algoritmalarının, verinin şifrelenmesi ve şifrenin çözülmesinde farklı anahtar kullanmasından kaynaklanan simetrik şifreleme algoritmalarına göre daha fazla hesaplamaya ihtiyaç duymasıdır.



Şekil 1. 3. Şifreleme ve Şifre Çözme Şeması

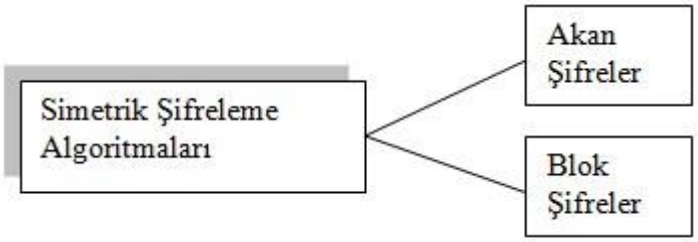
1.4.2 Simetrik Şifreleme Algoritmaları

Simetrik şifreleme algoritmalarında şifreleme ve şifre çözme sırasında aynı anahtar kullanılır. Simetrik şifreleme algoritması, açık metin ile gizli anahtarı girdi olarak alır ve şifreli metni çıktı olarak döndürür. Söz konusu gizli anahtar şifrelemeyi yapan ve metnin çözüleceği taraflarca bilinir. Şifreli metinden açık metne dönüşüm için gizli anahtarın bilinmesi elzemdir aksi durumda şifreli metinden açık metne dönüştürme

gerçekleşmez. Simetrik şifreleme algoritmalarının tercih nedeni olmalarındaki en önemli faktör hızlı çalışmalarıdır.

Asimetrik şifreleme algoritmalarına kıyasla simetrik şifreleme algoritmalarının hız bakımından sunduğu avantajların yanı sıra donanımsal olarak kullanımı da asimetrik algoritmalara göre basittir. Simetrik şifreleme algoritmalarında kullanılan anahtarın bit sayısının asimetrik şifreleme algoritmalarına göre daha düşük olması ve dolayısıyla anahtar boyunun küçüklüğü simetrik şifreleme algoritmalarının hızına etki eden bir diğer faktördür. (Şahin, 2015).

Simetrik şifreleme algoritmaları akış ve blok şifreler olmak üzere Şekil 1. 4'de gösterilen şekilde ikiye ayrılırlar.



Şekil 1. 4. Şifreleme Algoritmaları

1.4.2.1 Blok Şifreler

Blok şifreleme algoritmaları açık metinleri uzunlukları sabit olan bloklar şeklinde şifreler. Bloklar halinde aldıkları (örneğin 64-bit) açık metinleri birbirinden bağımsız ya da birbirlerine bağımlı şekilde şifreleyebilirler. Blok şifreleme algoritmaları simetrik şifreleme algoritmalarındandır ve bu nedenle şifreleme ve şifre çözme işlemlerinde aynı anahtarı kullanırlar. Aldıkları veriyi belirli uzunlukta bloklara bölerler ve her blokta ayrı şifreleme işlemi yaparlar.

Blok şifreleme algoritmaları kullanılırken algoritmanın güvenliğini etkileyen farklı çalışma kiplerine sahiptirler (Ferguson vd., 2011; Stallings W. , 2010; Knudson, 2000).

Bu çalışma kipleri aşağıdaki gibi verilebilir:

Elektronik Kod Kitabı (Electronic Code Book-ECB): Bu çalışma kipinde bloklar halinde bulunan her veri bloğu aynı anahtar kullanılarak diğer bloklardan bağımsız olarak

şifrelenmektedir. Bu durumda şifrelenen metinden elde edilen bloklarda aynı veri bloğunun birden fazla kez görünmesi, aynı olan bu veri bloklarının her defasında aynı şekilde şifreleneceği anlamını taşımaktadır. Bu çalışma kipi diğer çalışma kiplerine kıyasla daha az güvenilir kabul edilir.

Şifre Bloğu Zincirleme (Cipher Block Chaining-CBC): Bu çalışma kipi, veri bloklarını şifrelerken her bir veri bloğu için, anahtar ve bir önceki veri bloğunu kullanır. Kullanımı için bir başlangıç vektörünün gerektiği bu prensip şifrelediği her bir veri bloğunu kendisinden bir önceki veri bloğuna bağımlı kıldığından özellikle ECB çalışma kipinde çalışan bir blok şifreyi kırmak için kullanılan düz metin saldırısı Yinelenen anahtar XOR (repeating-key XOR)' a karşı daha güvenlidir.

Şifre Geri Besleme (Cipher Feedback-CFB): CFB çalışma kipinde ise CBC kipinde olduğu gibi şifrelenecek her bir veri bloğu şifrelenirken, anahtar ve bir önceki veri bloğundan yararlanılır fakat CFB kipinin CBC kipinden farkı bir önceki veri bloğunun kendinden sonra gelen veri bloğunu şifrelemek için değil anahtar oluşturmak için kullanılmasıdır. CFB kipinin başlatılması için de tıpkı CBC kipinde olduğu gibi başlangıç vektörü gereklidir.

Çıkış Geri Besleme (Output Feedback-OFB): Her bir veri bloğunun şifrelenmesinde anahtar ve önceki veri bloğu kullanılarak oluşturulan rastgele sayı dizisinden yararlanılan çalışma kipidir. Bu çalışma kipinin başlatılabilmesi için bir başlangıç vektörü gereklidir.

Sayaç (Counter): Veri bloklarının şifrelenmesinde ve şifrenin çözülmesinde kullanılan rastgele sayı akışının sayaç yardımıyla oluşturulduğu çalışma kipidir.

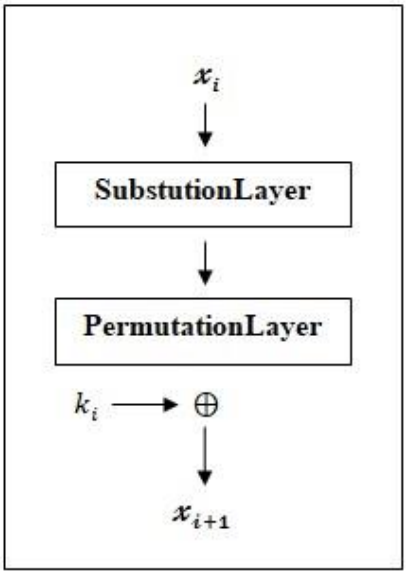
Blok şifreleme algoritmaları Shannon (1949)'un önermiş olduğu karıştırma ve yayılım tekniklerini esas alır. Karıştırma tekniği şifreli ve açık metin arasında bulunan ilişkiyi gizlerken, yayılma tekniği ise şifrelenen açık metindeki izlerin yayılımını ve bu sayede de şifreli metinde bu izlerin takip edilememesini amaçlar (Keliher, 2002).

Karıştırma ve yayılma tekniklerini uygulamak için yer değiştirme (substitution) ve doğrusal dönüşüm (linear transformation) işlemlerinden yararlanılır. Bu işlemleri uygularken blok şifrelemede kullanılan tasarım mimarileri Feistel ağları (Feistel & Notz, 1975) ve Yerdeğiştirme-Permütasyon (Substitution-permutation Networks-SPN) (Feistel, 1973) ağları şeklindedir. Feistel ağları ve SPN mimarileri arasındaki en belirgin

farklılık ise blok şifrelerin tasarımındaki en önemli parça olan döngülerin (round) işlenmesinde ortaya çıkmaktadır. Feistel mimarisinde bir döngüdeki o anki bloğun sadece yarısını işlenirken, SPN mimarisinde ise o anki bloğun tamamı işlenmektedir.

1.4.2.1.1 SPN Mimarisi

SPN mimarisi, blok şifrelerin tasarımında kullanılan bağlantılı bir dizi matematiksel işlemleri gerçekleştiren bir kriptografik mimaridir: açık metin bloğunu ve anahtarı girdi olarak alarak yer değiştirme katmanı (S-kutuları) ve permütasyon/doğrusal dönüşüm katmanından oluşan döngüler yardımıyla şifreli metin bloğunu oluşturur. *R* adet döngüden meydana gelen bir SPN algoritması için (*R*+1) adet *n*-bit anahtar gerekir. Yer değiştirme, doğrusal dönüşüm ve anahtar ekleme katmanlarının uygulandığı döngü adımlarından oluşmaktadırlar. S-kutusu katmanında mevcut blok alt bloklara bölünür ve bu alt bloklar genellikle eşit bir giriş ve çıkışa sahip S-kutusuna girdi olur. S-kutusu şifrenin çalışmasında kullanılan doğrusal olmayan tek yapıdır. Doğrusal dönüşüm katmanı, yer değiştirme katmanından gelen bitlerin doğrusal olarak karıştırma işlemini gerçekleştirir. Bu karıştırma işlemi için doğrusal yayılım matrisinin uygulanması ile sağlanır. Anahtar ekleme katmanında alt anahtar ile *n*-bit döngü girişi XOR işleminden geçer. Her döngünün sonucu diğer bir döngüde giriş verisi olarak kullanılır (*R*+1). Yani son döngüye ulaşıldığında son döngü anahtarı ile anahtar ekleme işlemi yapılır ve şifrenin çıktısı elde edilir (Sakallı vd., 2004).



Şekil 1. 5. SPN Ağının Bir Döngü Şeması

1.4.2.1.2 Feistel Mimarisi

Fiestel Mimarisi, kendisini geliştiren Horst Fiestel (Feistel, 1973)'ın adını alan ve döngü içerisinde o anki bloğu ikiye bölerek bu iki parçadan birini işleyen bir mimaridir. Bu sayede orijinal mesajı karıştırarak saldırganlar tarafından çözülmesini zorlaştıran bir tasarım ilkesidir.

SPN mimarisinde verilerin şifrelenmesi ve şifrelerin çözülmesinde kullanılan anahtarın olası bir saldırgan tarafından kolaylıkla tahmin edilmesine neden olabilecek sınırlı anahtar alanı olasılığı, şifreleme işlemleri sırasında anahtar alanının arttırılmasına olanak sağlayan Fiestel Mimarisi'ni güvenlik açısından avantajlı hale getirmiştir. Anahtar alanının arttırılması şifrelemenin güvenliğin arttırırken şifrenin verimliliğini azaltacağından Fiestel Mimarisi uygulanırken güvenlik ve verimlilik arasındaki dengeye dikkat edilmelidir. (Avanzi, 2017).

Fiestel ağları tekrarlanan şifrelerdir ve her bir döngüde girdi sağ ve sol olmak üzere iki ayrı koldan yapılır. Bloğun sağ tarafına doğrusal olmayan *F* döngü fonksiyonu girdi olarak uygulanır. Bu fonksiyonun uygulanmasından edilen işlem çıktısı ile bloğun sol tarafı XOR işleminden geçirilir. Bu işlemin sonucu bir sonraki döngünün sağ bloğunun girdisini oluştururken ilk döngünün sağ bloğu ise bir sonraki döngünün sol bloğunun girdisini oluşturur. Bu işlemlerde her döngüde bloğun yarısı değişir.

Bu uygulamada karıştırma, yayılma (*F* fonksiyonu) ve XOR işleminin uygulanmasından oluşan üç temel adım tekrarlanır.

Blok şifreler üç ana katman altında incelenebilirler. Bunlar:

- Yer değiştirme katmanı (S-Boxes)
- Yayılım Katmanı (Permutation Layer)
- Her Döngü için Alt Anahtarların Üretildiği Anahtar Planlama Katmanı

1.4.2.1.3 Doğrusal Dönüşümler ve S-Kutuları (S-Boxes)

Blok şifrelerde yayılma işleminin gerçekleşmesi için kullanılan yapılar Error! Reference source not found.'te bahsedilen doğrusal dönüşüm yapılarıdır ve bu yapılarda sabit boyuttaki giriş bloğu doğrusal olarak karıştırılarak giriş bloğuna eşit uzunlukta olan çıkış bloğu elde edilir (Aslan, 2013). Doğrusal dönüşümün kullanıldığı önemli şifreleme

algoritmalarına AES ve Khazad şifreleme algoritmalarını örnek verilebilir (Barreto & Rijmen, 2000).

Yine blok şifreleme algoritmalarında genellikle, düz metin ve şifreli metin arasında gerekli olan karmaşıklığın sağlanması için yer değiştirme amacıyla bir ya da birden fazla S-kutusu (S-Boxes) kullanılır. Bir blok şifre tasarımda S-kutusu (S-Boxes), doğrusal olmayan (nonlinear) katmandır. Bu durumda tasarlanan blok şifrenin gücü de doğrudan S-kutusunun gücünden etkilenir. Bu da demektir ki güçlü bir blok şifre tasarlanmak isteniyorsa, güçlü bir S-kutusu tasarımı kullanmak olmazsa olmazdır. Güçlü bir S-kutusu tasarlamak için S-kutularına ait bazı kriptografik kriterleri göz önünde bulundurmak gerekir. Bu kriterlerin belirlenmesinde doğrusal kriptanaliz, diferansiyel kriptanaliz gibi saldırılar etkin rol oynamıştır. Bu kriterler aşağıdaki sıralanabilir (Kam & Davida, 1979):

- **Bütünlük Kriteri:** Kam ve Davida tarafından belirlenen bütünlük özelliğine göre bir S-Kutusunda her bir çıkış bitinin, giriş bitlerinin tamamına bağımlı olmasıdır. (Kam & Davida, 1979)

- **Çığ Etkisi:** Feistel'ın tanımladığı bu özelliğe göre giriş bitlerinden bir tanesinin değiştirilmesi durumunda çıkış bloğunun yarısı değişmelidir (Feistel, 1973).

- **Katı Çığ Etkisi:** Bütünlük ve çığ etkisi özelliklerinin bir arada düşünülmesi ile ortaya çıkan bu kriter Webster ve Tavares (1985) tarafından ortaya konmuştur. Bu kritere göre giriş bitlerinden bir tanesinin değiştirilmesi sonucunda her bir çıkış bitinin yarısı değişir.

- **Sabit Nokta Sayısı:** Doğrusal dönüşüm yapılarında giriş bloğunun doğrusal olarak karıştırılarak giriş bitine eşit boyutta çıkış biti elde edilir. Eğer giriş bloğunda yer alan bitler, çıkış bloğuna değiştirilmeden aktarılıyorsa bu durumda giriş bloğuna doğrusal dönüşümün sabit noktası adı verilir (Tanım 2.23).

1.4.2.1.4 AES (Advanced Encryption Standard) Şifreleme Algoritması

SPN blok şifrelerinin en bilineni olan AES şifreleme algoritması, 128-bit uzunluğunda olan veri bloklarını 128-192-256 bitlik anahtar uzunluğu ile şifreler. Algoritmada yer alan döngü sayısı anahtar genişliğine bağlı olarak değişir. Örneğin 128-bit uzunluğundaki bir anahtar için şifreleme işlemi 10 döngüde tamamlanırken, bahsi geçen döngü sayısı 192 ve 256 bitlik anahtar genişliği söz konusu olduğunda 12 ve 14 olarak değişir.

Her bir döngünün kendi içinde dört katmandan meydana geldiği AES şifreleme algoritmasında şifreleme yapılmadan önce ilk açık metin 128-bit bloklara ayrılarak 16-bytelık (128 bit) mesaj 4×4-byte matrisine dönüştürülür. Daha sonraki adımlarda ise 4×4-byte matrisine sırayla (SubBytes)-byte'ların yer değiştirmesi (ShiftRows)-satırların ötelenmesi, (MixColumns)-sütunların karıştırılması ve son olarak (AddRoundKey) - anahtarın döngüye dahil edilmesi işlemleri uygulanır. Burada döngünün son adımında yer alan anahtarın döngüye dahil edilmesi anahtar planlama algoritmasından gelen döngü anahtarının sütunların karıştırılması adımından sonra oluşan durum matrisi ile XOR işlemine tabi tutulması anlamına gelir. AES algoritmasında bahsi geçen her döngü için yukarıda anlatılmış olan 4 temel döngü adımı uygulanır, algoritma son döngüye ulaştığında ise satırların ötelenmesi (MixColumns) adımı uygulanmaz (Sakallı, 2006).

1.4.2.1.5 ARIA Blok Şifreleme Algoritması

ARIA blok şifreleme algoritmasında 128-bit veriler bloklar halinde işlenir ve kullanılan anahtarın uzunluğu 128-196-256 şeklinde değişebilir. Bu şifreleme algoritmasına ait doğrusal dönüşümde Şekil 1.6'da görülen 16×16 boyutunda tersi kendisine eşit (involutif) MDBL matris kullanılır (Yavuzer-Aslan vd., 2012). Bu matrisin dal sayısı (branch number) 16×16 boyutunda bir ikili matrisin alabileceği en yüksek değer olan 8 değerine eşittir.

Şifreleme aşamasında kullanılan doğrusal dönüşüm katmanı ile 128-bit'lik veri blokları 16-bit'lik dikey bölümlere ayrılarak Şekil 1.6'da verilen dönüşüm matrisi ile çarpma işlemine sokulur. Bu çarpma işlemleri sonlu cisim $\mathbb{F}_2$'de yapılır ve bu çarpma işlemlerinin sonucunda elde edilen veriler giriş bitlerinden bazılarının XOR toplamı olacak şekilde doğrusal fonksiyon olarak tanımlanabilir (Akkanat, 2018).

$$A_{Aria} = \begin{bmatrix} 0&0&0&1&1&0&1&0&1&1&0&0&0&1&1&0 \\ 0&0&1&0&0&1&0&1&1&1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1&0&0&0&1&1&1&0&0&1 \\ 1&0&0&0&0&1&0&1&0&0&1&1&0&1&1&0 \\ 1&0&1&0&0&1&0&0&1&0&0&1&0&0&1&1 \\ 0&1&0&1&1&0&0&0&0&1&1&0&0&0&1&1 \\ 1&0&1&0&0&0&0&1&0&1&1&0&1&1&0&0 \\ 0&1&0&1&0&0&1&0&1&0&0&1&1&1&0&0 \\ 1&1&0&0&1&0&0&1&0&0&1&0&0&1&0&1 \\ 1&1&0&0&0&1&1&0&0&0&0&1&1&0&1&0 \\ 0&0&1&1&0&1&1&0&1&0&0&0&0&1&0&1 \\ 0&0&1&1&1&0&0&1&0&1&0&0&1&0&1&0 \\ 0&1&1&0&0&0&1&1&0&1&0&1&1&0&0&0 \\ 1&0&0&1&0&0&1&1&1&0&1&0&0&1&0&0 \\ 1&0&0&1&1&1&0&0&0&1&0&1&0&0&1&0 \\ 0&1&1&0&1&1&0&0&1&0&1&0&0&0&0&1 \end{bmatrix}$$

Şekil 1. 6. ARIA Şifresinde Kullanılan 16 x 16 Boyutunda İkili Matris

1.5 Tezin Ana Hatları, Gerekçesi ve Önemi

Blok şifreleme algoritmalarının dönüşüm katmanlarında kullanılmak üzere kriptografik açıdan verimli ve düşük maliyetli matrislerin tasarlanması, literatürde yer alan problemlerdendir. Bir dönüşüm katmanında kullanılacak matrisin hem maliyet açısından düşük olması hem yazılım ve donanımsal olarak sınırlı imkanlara sahip sistemlerde güvenlikten, hızdan ödün vermeden kullanılabilmesi ve bu unsurlar arasında gereken verimi sağlaması beklenmektedir. Hız ve verimlilik açısından kullanışlı olan Maximum Distance Binary Linear- Maksimum Uzaklıkta Ayrılabilen İkili Doğrusal (MDBL) matrisler ile yayılım tabakalarının tasarımı ve sistemsel kısıtlamalar değerlendirildiğinde, tezin yukarıda bahsi geçen problemlere yönelik çalışma motivasyonu oluşmuştur.

Tezde blok şifrelerde doğrusal dönüşüm katmanlarında kullanılmak üzere mevcut farklı tekniklerle MDBL Matrisler üretilmiş ve üretilen bu matrislerden düşük XOR sayısına sahip olanlar üzerinde çalışılarak optimizasyon teknikleri uygulanmıştır. Optimizasyon algoritmaları sonucu elde edilen yeni XOR sayıları değerlendirilmiş, uygulanan teknikler karşılaştırılmış ve maliyet ve yayılım açısından verimli matrisler elde

edilmeye çalışılmıştır. Bu tezde Paar ve Boyar Peralta optimizasyon yöntemlerine odaklanılmıştır.

Tezin amacının temelini oluşturan Kriptoloji bilimi ile ilgili temel kavramlardan yukarıda yer alan ilk bölümde bahsedilmiştir. Bundan sonraki bölüm olan matematiksel kavramlar bölümü, tezde kullanılmış olan tanımları ve altyapıyı içermektedir. Üçüncü bölüm literatürde yer alan Maximum Distance Separable- Maksimum Uzaklıkta Ayrılabilen (MDS) ve MDBL matrisler ve tasarım tekniklerini ve XOR sayısı ile ilgili temel bilgileri içermektedir. Dördüncü bölümde ham XOR sayısı ile elde edilen matrislere uygulanacak olan optimizasyon teknikleri hakkında detaylı bilgileri verilirken beşinci bölümde ise tezde materyal olarak kullanılan matrislerin elde edilme yöntemleri, bu yöntemler uygulanırken kullanılan kodları ve elde edilen deneysel sonuçlara ait çizelgeler verilmiştir.

Tezin son bölümü olan altıncı bölümde beşinci bölümde elde edilen tüm verilerin değerlendirilmelerini ve ortaya çıkması muhtemel açık problemlere ait tartışma ve sonucu kapsamaktadır.

BÖLÜM 2

2. MATEMATİKSEL ALTYAPI

Tezin bu bölümünde kriptografik yayılım tabakalarında kullanılan optimizasyon tekniklerinin uygulanmasında yararlanılan ve bu alanda yaygın kullanılan matematiksel ifadelere ait açıklamalar tanımlanmıştır. Matematiksel altyapı bölümüne ait tanımların ispatları Lidl ve Niederreiter (1994), Roman (1992), Stallings (2017), Nakahara ve Abrahao (2009) ve Forouzan ve Mukhopadhyay (2015)' a ait yayınlarda yer almaktadır.

Tanım 2.1 Sonlu Cisim: Sonlu sayıda elemana sahip cisimdir. q, bir asal sayı ve $n \in \mathrm{Z}^{+}$ olmak üzere sonlu cisim $\mathbb{F}_{q^n}$ $veya$ $GF(q^n)$ şeklinde ifade edilir. Sonlu cismin eleman sayısına ise o cismin mertebe ya da derecesi adı verilir.

Örnek 2.1. *Sonlu cisim tanımlamalarına ait birkaç örnek aşağıda verilmiştir.*

- $\mathbb{F}_2$: $Elemanları$ $\{0,1\}$ $olan$ $sonlu$ $cisim$
- $\mathbb{F}_{2^3}$: $Elemanları$ $\{0,1\}$ $olan$ 3 $bitlik$ $sonlu$ $cisim$
- $\mathbb{F}_4$: $Elemanları$ $\{0,1,2,3\}$ $olan$ $sonlu$ $cisim$

Tanım 2.2 Polinom: $n \in Z$ ve $n \geq 0$ olmak şartıyla tanımlanan bir polinom denklem 2.1'de gösterilmiştir.

$$f(x) = a_n x^n + a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0 = \sum_{i=0}^{n} a_i x^i \qquad (2.1)$$

Denklem 2.1 'de yer alan $a \neq 0$ olmak şartıyla a_i katsayılar kümesidir.

Derecesi 0 olan bir polinom sabit polinom olarak tanımlanırken, 0'a eşit olan bir polinoma ise sıfır polinom denilmektedir ve sıfır polinomun derecesi belirsizdir.

Tanım 2.3 İndirgenebilir ve İndirgenemez Polinom: F bir cisim olmak üzere $F[x] \coloneqq \left\{ \sum_{i=0}^{n} \alpha_i x^i : \alpha_i \in F, n \geq 0 \right\}$ F üstüne bir polinom halkadır. $F[x]$'in bir elemanı ise F üstüne bir polinom olarak isimlendirilir. Bir $f(x) \coloneqq \sum_{i=0}^{n} \alpha_i x^i$ polinomu için n ($n \in Z$), $f(x)$ polinomunun derecesidir ve $deg\big(f(x)\big)$ şeklinde tanımlanmaktadır. Bu $f(x)$ polinomu için;

$$deg\big(g(x)\big) < deg\big(f(x)\big) \; ve \; deg\big(h(x)\big) < deg\big(f(x)\big) \; iken$$

$f(x) = g(x) \cdot h(x)$ eşitliğini sağlıyorsa bu polinomlara indirgenebilir polinomlar adı verilir. Bu eşitliği sağlamayan polinomlar ise indirgenemez polinomlar olarak adlandırılır (Tuncay, 2021; Pehlivanoğlu vd., 2021).

Error! Reference source not found. Doğrusal Dönüşüm: Elemanları $A: (\{0.1\}^m)^n \rightarrow (\{0.1\}^m)^n$ olan doğrusal dönüşümün ifadesi 2.2'de yer almaktadır.

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

Doğrusal dönüşüme ait elemanlar $\mathbb{F}_2$ cisminden seçildiğinde, bu doğrusal dönüşüm ikili doğrusal dönüşüm olarak adlandırılmaktadır.

Error! Reference source not found. Hamming Ağırlığı: *C* herhangi kod kümesi olsun. Bu kod kümesinde yer alan 1'lerin sayısı ilgili kod kümesinin Hamming ağırlığıdır ve $wt(C)$ ile gösterilir. Denklem 2.3'te gösterilen kod kümesinin Hamming ağırlığı 3'tür.

$$C = \{0,0,1,1,1,0\} \tag{2.3}$$

Tanım 2.6 Hamming Uzaklığı: C_1 ve C_2, $(\mathbb{F}_{2^m})^n$ vektör uzayında tanımlı olan n boyutlu iki adet kod kümesi olmak üzere iki kod kümesinin farklı elemanlarının sayısı Hamming ağırlığı olarak tanımlanır. Örnek 2.3.'de yer alan kod kümelerinin Hamming uzaklığı 3'tür.

Örnek 2.3. $C_1 = \{\mathbf{0}, 0,1, \mathbf{1}, \mathbf{0}, 0\}, C_2 = \{\mathbf{1}, 0,1, \mathbf{0}, \mathbf{1}, 0\}$

Tanım 2.7 Birim Matris: $n \times n$ boyutlu, ana köşegen elemanları 1, diğer elemanları 0 olan ve I_n / I ile gösterilen kare matrislere birim matris adı verilir.

Tanım 2.8 Simetrik Matris: M matrisinin transpozu M^T ile gösterilsin.

$M = M^T$ eşitliği sağlanıyorsa bu matrisler simetrik matrisler olarak adlandırılır.

Tanım 2.9 Köşegen Matris (Diyagonal Matris): $i \neq j$ ve $a_{ij} = 0$ şartlarını sağlayan $n \times n$ boyutlu $M = [a_{ij}]$ matrisi köşegen matris olarak adlandırılır. Köşegen matrislerin ana köşegenleri üzerinde bulunan elemanları dışında bulunan tüm elemanları 0'dır.

Tanım 2.10 Involutif Matris: Tersi kendisine eşit olan matrisler involutif matris olarak tanımlanır ve $M = M^{-1}$ ile ifade edilir.

Tanım 2.11 Ortogonal Matris: M, $n \times n$ boyutlu bir kare matris olmak üzere $M \cdot M^T = I$ eşitliğini sağlayan matrisler ortogonal matrisler adı verilir.

Tanım 2.12 Dairesel Matris: Her bir satır vektörünün kendisinden bir önce gelen satır vektörüne göre bir elemanla döndürülmesi ile oluşan $n \times n$ boyutlu matrise dairesel matris adı verilir ve Örnek 2.4'teki şekilde ifade edilir.

Örnek 2.4. $\mathrm{C} = (\alpha_{ij}), \alpha_{ij} \in \mathbb{C}$ olmak üzere, matris elemanları $j - i \equiv k(mod\, n)$'de tanımlı $n \times n$ boyutlu dairesel matris

$$C = \begin{bmatrix} a_0 & a_1 & a_2 \cdots a_{n-1} \\ a_{n-1} & a_0 & a_1 \cdots a_{n-2} \\ a_{n-2} & a_{n-1} & a_0 \cdots a_{n-3} \\ \vdots & \vdots & \vdots \ddots \vdots \\ a & a_2 & a_3 \cdots a_0 \end{bmatrix}$$

şeklinde açık olarak gösterilebilir (Gupta & Ray, 2013). $n \times n$ boyutlu bir dairesel matriste en çok *n* farklı adette giriş bulunmaktadır.

Tanım 2.13 Hadamard Matris: A_0 ve A_1, $2^{n-1} \times 2^{n-1}$ sonlu cisim Hadamard matrisler olsun, $2^n \times 2^n$ boyutlu sonlu cisim Hadamard matris olan *H*, denklem 2.4'te yer alan ifadedeki gibi gösterilir.

$$H = had(A_0, A_1) = \begin{bmatrix} A_0 & A_1 \\ A_1 & A_0 \end{bmatrix} \tag{2.4}$$

Örnek 2.5. $\mathbb{F}_{2^m}$ sonlu cisminin sıfırdan farklı elemanı a_i, ve $H = had(\, a_0, a_1, a_2, a_3\,)$, 4×4 boyutlu bir Hadamard matris olmak üzere *H* matrisinin gösterimi aşağıdaki gibidir (Pehlivanoğlu vd., 2018):

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

Tanım 2.14 Genel Hadamard Matris: $\mathbb{F}_{2^m}$ cisminin 0'dan farklı bir elemanı b_1 olmak üzere Denklem 2.4'te verilen Hadamard Matrisine karşılık gelen 2×2 boyutlu Genel Hadamard Matrisi Ghad($a_0, a_1; b_1$), Denklem 2.5'da gösterildiği şekilde ifade edilir.

$$GH = Ghad(a_0, a_1; b_1) = \begin{bmatrix} a_0 & a_1 b_1 \\ a_1 b_1^{-1} & a_0 \end{bmatrix} \tag{2.5}$$

Tanım 2.15 Toeplitz Matris: $n \times n$ boyutlu, $a_{ij} = x_{i-j}$ şartını sağlayan T matrisi Denklem 2.6'daki gibi ifade edilir. Bu matrislere Toeplitz matrisler denir.

$$T = \begin{bmatrix} x_0 & x_{-1} & x_{-2} & \cdots & x_{-(n-1)} \\ x_1 & x_0 & x_{-1} & \cdots & \vdots \\ x_2 & x_1 & x_0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{n-1} & \cdots & \cdots & \cdots & x_0 \end{bmatrix} \tag{2.6}$$

Örnek 2.6. $T = Toep(a_1, a_2, a_3, a_4, a_5, a_6)$, 4×4 boyutlu bir Toeplitz matris olmak üzere. *T* matrisinin gösterimi aşağıdaki gibidir (Sakallı vd.,2020):

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

Tanım 2.16 Vandermonde Matris: $n \times n$ boyutlu, $a_{ij} = x_j^i$ şartını sağlayan matrislere Vandermonde matrisler denir. Denklem 2.8'de Vondermonde matrisinin gösterimi yer almaktadır.

$$V = vand(x_0, x_1, x_2, \ldots, x_{n-1}) = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ x_0 & x_1 & \cdots & x_{n-1} \\ x_0^2 & x_1^2 & \cdots & x_{n-1}^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_0^{n-1} & x_1^{n-1} & \cdots & x_{n-1}^{n-1} \end{bmatrix} \tag{2.7}$$

Tanım 2.17 Cauchy Matris: $\{x_0, x_1, \dots, x_{n-1}\}, \{y_0, y_1, \dots, y_{n-1}\}$ kümeleri $\mathbb{F}_{2^m}$ sonlu cisminde tanımlı ve$a_{ij} = \frac{1}{x_i + y_j}$ veya $a_{ij} = \frac{1}{x_i - y_j}$ olmak üzere, $0 \leq i, j \leq n-1$ değerleri için $x_i + y_i \neq 0$ ise $C_1(a_{i,j})$ ve $C_2(a_{\mathrm{i,j}})$ ile Denklem 2.8'de gösterilen matrisler Cauchy Matris olarak tanımlanır.

$$C_1 = \begin{bmatrix} \frac{1}{x_0+y_0} & \frac{1}{x_0+y_1} & \cdots & \frac{1}{x_0+y_{n-1}} \\ \frac{1}{x_1+y_0} & \frac{1}{x_1+y_1} & \cdots & \frac{1}{x_1+y_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{x_{n-1}+y_0} & \frac{1}{x_{n-1}+y_1} & \cdots & \frac{1}{x_{n-1}+y_{n-1}} \end{bmatrix} C_2 = \begin{bmatrix} \frac{1}{x_0-y_0} & \frac{1}{x_0-y_1} & \cdots & \frac{1}{x_0-y_{n-1}} \\ \frac{1}{x_1-y_0} & \frac{1}{x_1-y_1} & \cdots & \frac{1}{x_1-y_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{x_{n-1}-y_0} & \frac{1}{x_{n-1}-y_1} & \cdots & \frac{1}{x_{n-1}-y_{n-1}} \end{bmatrix} \quad (2.8)$$

Tanım 2.18 Hankel Matris: Simetrik bir matristir. *H* ile ifade edilen Hankel matrisinin herhangi bir elemanına $(H)_{i,j} = x_{i+j}$ eşitliği ile bulunabilir. *H*, Denklem 2.9'da gösterildiği şekilde tanımlanırken 2.10'da H Hankel matris formu gösterilmiştir.

$$H = hank(x_0, x_{-1}, \cdots, x_{(n-1)}; x_n, x_{n+1}, \cdots, x_{(2n-2)}) \quad (2.9)$$

$$H = \begin{bmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-2} & x_{n-1} \\ x_1 & x_2 & x_3 & \cdots & x_{n-1} & x_n \\ x_2 & x_3 & x_4 & \cdots & x_n & x_{n+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{n-1} & x_n & x_{n+1} & \cdots & x_{2n-3} & x_{2n-2} \end{bmatrix} \quad (2.10)$$

Örnek 2.7. *H=Hank*(a_0, a_1, a_2, a_3), 4×4 boyutlu bir Hankel matris olmak üzere. *H* matrisinin gösterimi aşağıdaki gibidir.

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

Tanım 2.19 Üreteç Matris: Doğrusal bir kod *C = [n, k, d]* olmak üzere, *G*, *C* doğrusal kodunun baz vektörlerinden oluşan $n \times k$ boyutlu bir üreteç matristir ve Denklem 2.11'deki eşitliği sağlar ve *G* matrisinin satırları *C* doğrusal kodunun tabanıdır.

$$C = \{xG | x \in V(k, q)\} \quad (2.11)$$

Tanım 2.20 Maksimum Uzaklıkta Ayrılabilen Matrisler (Maximum Distance Separable-MDS): Doğrusal kod olan *C=[n, k, d]*, $d = n - k + 1$ eşitliğini sağlıyorsa, *C* kodu MDS koddur. Yukarıda tanımı verilen MDS kodlar, MDS matrisleri üretirler. *M*, $n \times n$ boyutlu bir kare matris olmak üzere *M* matrisine ait kare alt matrislerin determinantlarının tamamı

0'dan farklı ise M matrisi bir MDS matristir. $n \times n$ boyutlu MDS matrisin tüm girdilerinin 0'dan farklı olması zorunludur ve bu matrisin dal sayısı $n + 1$'dir. M matrisine ait satır ve sütunların, $\mathbb{F}_{2^m}$ cisminin 0'dan farklı herhangi bir elemanı ile çarpılması sonucu oluşan yeni matris ve yine *M* matrisinin transpozesi olan matris olmak üzere oluşan matrislerin tamamı MDS özelliklerini korurlar.

Bir MDS matris olan *M* matrisinin doğrusal dönüşümünün yayılma gücü, dal sayısı ile tanımlanır.

Tanım 2.21 Doğrusal Dal Sayısı (Linear Branch Number): *M*, $n \times n$ boyutlu bir matris olsun. Bu matrisin doğrusal dal sayısı β_l, Denklem 2.12'te ifade edildiği şekilde tanımlanır.

$$\beta_l(M) = \min\{wt(x) + wt(A^T.x^T) | x \in (\{0.1\}^m)^n, x \neq 0\} \tag{2.12}$$

Tanım 2.22 Diferansiyel Dal Sayısı (Differential Branch Number): *M*, $n \times n$ boyutlu bir matris olsun. Bu matrisin diferansiyel dal sayısı β_d, Denklem 2.13'te ifade edildiği şekilde tanımlanır.

$$\beta_d(M) = \min\{wt(x) + wt(A \cdot x^T) | x \in (\{0.1\}^m)^n, x \neq 0\} \tag{2.13}$$

Tanım 2.23 Sabit Nokta Sayısı: A bir doğrusal dönüşüm, x ise A doğrusal dönüşümüne ait bir giriş bloğu olsun. $A \cdot x = x$ eşitliği sağlanıyorsa bu doğrusal dönüşümün bir sabit noktası x'tir. A doğrusal dönüşümüne ait sabit nokta sayısının hesaplanması için I birim matrisi, A'nın $n \times n$ boyutlu olduğunu ve *m-bit*'lik giriş değerlerine sahip olduğunu düşünelim. Bu durumda $n \times n$ boyutlu A doğrusal dönüşüm matrisinin tüm sabit noktaların kümesi Denklem 2.14 denkleminin çözümü ile elde edilebilir.

$$(A + I)x^T = 0 \tag{2.14}$$

A doğrusal dönüşümüne ait tüm sabit nokta sayısı FP_A, Denklem 2.15'te verilmiştir.

$$FP_A = 2^{m(rank(A) - rank(A+I))} = 2^{m(n - rank(A+I))} \tag{2.15}$$

Tanım 2.24 XOR Sayısı: Giriş değerlerinin birbirinden farklı durumlarda 1, diğer hallerde 0 sonucunu veren özel VEYA/eXclusive OR olarak da adlandırılan mantık kapılarına XOR kapısı denilir. $\mathbb{F}_{2^m}$ cisminde tanımlı indirgenemez $p(x)$ polinomunun elemanı olan a $(a(\in px))$ ve $\mathbb{F}_{2^m}$'de tanımlı olan $p(x)$ polinomundan rastgele seçilen bir elemanla çarpımı için gerekli olan XOR kapısı sayısıdır denilir ve XOR(a) şeklinde ifade edilir.

$\mathbb{F}_{2^m}$ cisminde tanımlı olan $n \times n$boyutlu bir A matrisine ait bir satırın XOR sayısının hesaplanması Denklem 2.16'da gösterilmiştir. Bu ifadede yer alan $\lambda_i A$ matrisinin *i.* satırına ait toplam XOR sayısıdır. *m* sonlu cismin derecesiyken *n,* aynı satırda bulunan 0'dan farklı elemanların sayısıdır (Khoo vd., 2014).

$$XOR\ Sayısı(A) = (\lambda_1, \lambda_2, \dots, \lambda_k) + (n-1).m \tag{2.16}$$

BÖLÜM 3

3. MATRİS TASARIM TEKNİKLERİ

3.1 Maksimum Uzaklıkta Ayrılabilen (MDS) Matrisler

Tanım 2.20'de tanımlanan MDS matrisler veya çoklu permütasyon tekniği Schnorr (1994) tarafından verimli bir difüzyon tabakası oluşturulmak için kullanılmıştır.

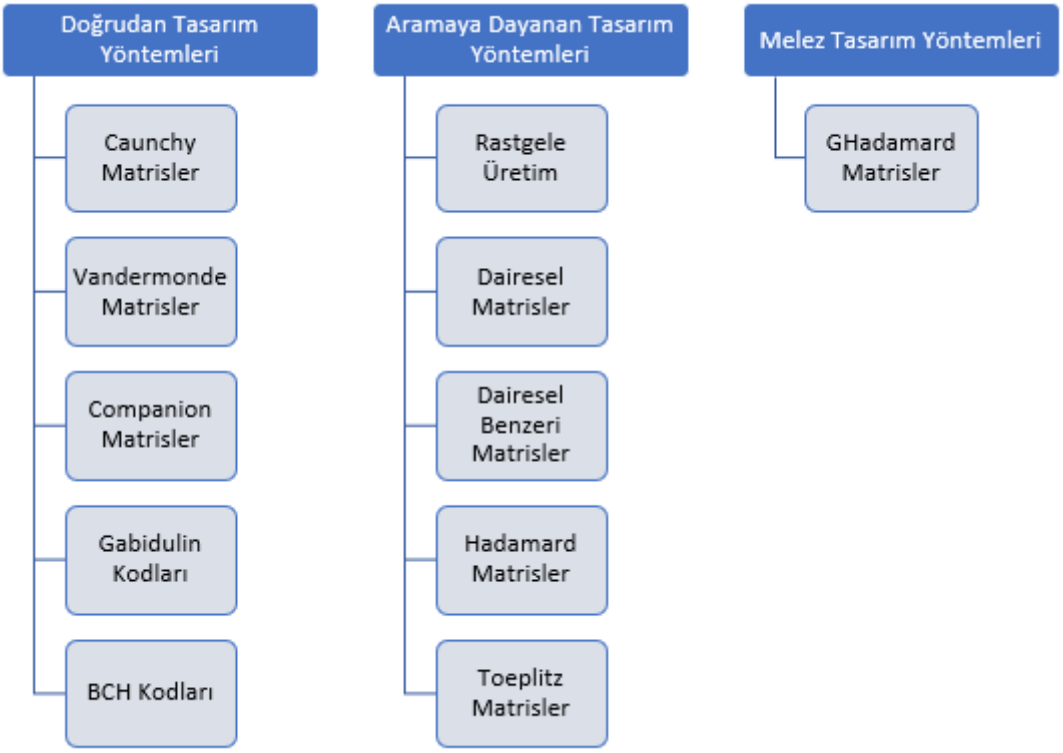
Yayılım katmanı blok şifrelerin tasarımında kullanılması gereken temel bileşenlerden biridir. Yayılım, her bir döngüde açık metin ve gizli anahtarda bulunan her bitin, şifreli metindeki birden fazla biti etkilemesini ve döngü sayısı ilerledikçe çıkıştaki tüm bitlerin girişteki bitlere bağlı olmasını sağlar. MDS matrislerin önemi burada ortaya çıkmaktadır. Yayılım tabakasında başarılı bir sonuç elde edebilmenin ve belli ölçüde güvenlik için ihtiyaç duyulan alanı azaltmanın yolu MDS matrisler ya da çoklu-permütasyonların kullanımı ile mümkündür (Tuncay, 2021).

MDS matrisler elde edilirken şimdiye dek kullanılan teknikleri 3 temel başlıkta tanımlamamız gerekirse, bunlar; doğrudan tasarım, arama ve melez tasarım teknikleridir. Bu tekniklerden doğrudan tasarım tekniği özel matrislerin ve kodların kullanıldığı ve bu bileşenlerle MDS matrislerin doğrudan elde edildiği bir MDS matris üretim tekniğidir.

Kaba kuvvet arama tekniği ise yine özel matrisleri kullanıp rastgele arama yaparak MDS matrislere ulaşmaya çalışılan tekniktir.

MDS matrislerde şifreleme ve şifre çözme sırasında kullanılan maliyetlerde farklılıklar olabilmektedir. Bu durumda involutif MDS matrislerin hafif sıklet şifreleme sistemlerinde kullanılması donanımsal maliyeti düşürmek açısından avantajlıdır.

Ancak Cauchy ve Vandermonde gibi matrislerin kullanılmasıyla ortaya çıkan katsayılar oldukça karmaşıktır ve ortaya çıkan bu durum yapılan uygulamaların verimliliğini düşürür.



Şekil 3. 1. MDS Matris Tasarım Yöntemleri (Tuncay, 2021)

3.1.1 Circulant (Dairesel) Matrisler

Tanım 2.12'de yer alan dairesel matrislerden MDS matrisler oluşturulurken doğrudan tasarım tekniklerinden yararlanmak mümkün değildir, ancak dairesel matrislerden de MDS matris tasarımları yapılabilir. Bunun için aramaya dayalı yöntemlerden yararlanmak gerekir. Dairesel matrisler, MDS matris tasarımında yararlanılmak üzere Hadamard matrisler ile maliyet açısından kıyaslandığında bir avantaja sahiptir. Bu avantaj; 1 elemanının tekrarını içerebiliyor olmalarıdır. Bilindiği gibi hafif-siklet kriptografide tekrarlı girişlere sahip matrisler maliyet avantajı sağlarlar. Buna en güzel örnek AES şifreleme algoritmasında kullanılan difüzyon matrisidir (Daemen & Rijmen, 2002). Bu matriste yer alan 1'lerin sayısı ve 1 elemanının çarpma işlemi sırasında maliyetsiz olması durumu nispeten küçük dereceli arama uzaylarında dairesel matrisleri avantajlı konuma getirir. Fakat söz konusu SPN ağ mimarisinin devreye girdiği blok şifrelerde difüzyon matrisinin hem MDS hem de involutif olması şifreleme ve şifre çözme evrelerinde avantaj sağlayacağından ve dairesel matrislerin aynı anda bu iki özelliği taşıyamaması söz konusu olduğundan MDS matrislerin tasarımı için dairesel matrislerin farklı türevleri oluşturulmuş ve kullanılmıştır (Tuncay, 2021).

3.1.2 Toeplitz Matrisler

Tanım 2.15'te tanımlanan Toeplitz matris $T= (t_{ij}), t_{ij} \in \mathbb{C}$ olmak üzere, matris elemanları $t_{ij} = t_{i+1,j+1}$ ve i,j = 1,2,…,n-1 olarak belirlenir. Bu ifadeden anlaşıldığı üzere dairesel matrisler aynı zamanda Toeplitz matrislerdir denilebilirken Toeplitz matrislerin her zaman Dairesel matrisler olduğu söylenemez (Uslu, 2008).

3.1.3 Hadamard Matrisler

Tanım 2.13'de anlatılan Hadamard matrisler James Sylvester tarafından 1987 yılında ilk kez tanımlanmışlardır. Ancak James Sylvester bu matris türünü tanımlarken matrislerin mertebelerinin 2'nin katları (H_1,H_2,H_4,H_8,H_{16},H_{32},…) şeklinde olabileceğini söylemiştir. Daha sonra Jacques Hadamard, bu matris formuna "Hadamard" ismini verdiren ve günümüzde hala geçerliliğini koruyan yeni bir tanımlama yapmış ve 1. ve 2. mertebeden sonraki matrislerin 4'ün kuvvetleri şeklinde olduğunu 12 ve 20 boyutlu yapıları oluşturarak göstermiştir (H_1, H_2, H_3 , H_4, H_8, H_{12}, H_{20}, …).

Hadamard Matrisler kodlar teorisi ve mobil iletişim başta olmak üzere çeşitli alanlarda rol oynamaktadırlar. Hadamard Matrislerin tasarımında kullanılan birçok yöntem vardır bunlara Kronecker çarpımı ve Konferans matris yöntemleri örnek verilebilir. n boyutlu Hadamard matrisinin oluşturulabilmesi için n 'in 1,2 veya $4k$ ($k \in N$) olması zorunludur.

MDS matrislerin Tasarımında Sonlu Cisim Hadamard matrisler kullanılır. Tez çalışmasında kolaylık olması açısında Sonlu Cisim Hadamard matrisler, Hadamard matris olarak ifade edilmektedir. Hadamard matrislere MDS matrislerin tasarımında başvurulması, involutif MDS matrislerin tasarımında arama uzayının büyüklüğüne bağlı olarak maliyet açısından uygun matrislerin bulunmasının Vandermonde ve Cauchy (Tanım 2.17) matrislerinden yararlanılan tekniklerde zorlaşmasıyla gerçekleşmiştir. $\mathbb{F}_{2^m}$ üzerine tanımlı bir $k \times k$ Hadamard matrisin önemli bazı özellikleri aşağıdaki gibi verilebilir (Pehlivanoğlu M. vd., 2018) :

1. $H_{i,j} = \alpha_{i \oplus j}$, Bu eşitlikte α_i, , $k \times k$ boyutlu H Hadamard matrisinin ilk satırının girişlerini temsil etmektedir.

2. $J_{i,k-i+1} = 1$ ve diğer elemanları 0 olan $n \times n$ boyutlu bir değişim matrisi olmak üzere $H \cdot J = J \cdot H \; ve \; H = H^T$ 'dir ve H matrisi bisimetrik matristir.

3. $c = \oplus_{i=0}^{n-1} \alpha_i$ ve $I, k \times k$ boyutlu birim matris olmak üzere $H^2 = c^2 I$ dir.

Bir Hadamard matrisin karesi yani kendi ile çarpımı, söz konusu matrisin ilk satırda bulunan elemanlarının karelerinin toplamı ve Tanım 2.7'de tanımlanan birim matrisin çarpımına eşittir. Bu durumda Hadamard matrisin ilk satırda yer alan elemanlarının XOR toplamı 1 olduğunda $\mathbb{F}_{2^m}$'de tanımlanmış olan matrisin involutif matris olduğu anlamını taşır. Ayrıca bir Hadamard matrisin ilk satır elemanlarının toplamı sıfırdan farklı olduğunda bu matrisin ilk satır elemanlarının toplamına bölünerek bir involutif matris (Tanım 2.10) elde edilir (Tuncay, 2021).

3.1.4 GHadamard Matrisler

Cauchy ve Vandermonde matris yöntemlerinin boyut büyüdükçe artan maliyetleri nedeniyle Hadamard matrislerin kullanılması tercih edilmesine benzer şekilde Hadamard matrislerde de uzay ve boyut büyüdükçe maliyet artmakta ve bu durum farklı yöntemlere yönelimi kaçınılmaz kılmaktadır. Bu arayış sonucunda Tanım 2.14'te verilen GHadamard matrisler ortaya çıkmıştır. GHadamard (Genel Hadamard) matrisler Pehlivanoğlu (2018)'de anlatılan yöntemle, involutif MDS matrislerin yeni türlerini hem küçük boyutlu hem de daha büyük boyutlarda oldukça verimli şekilde meydana getirmektedirler.

3.1.5 Maximum Distance Binary Linear (MDBL) Matrisler

Elemanları $\mathbb{F}_2$ cisminin elemanları olan matrislere ikili matrisler adı verilir. İkili matrislerin $n \times n$ boyutlu olanları, maksimum dal sayıları $[2n, n]$ ikili linear kodların maksimum uzaklığına eşittir (Kwon vd., 2005).

Bu tanımdan yola çıktığımızda MDBL (Maximum Distance Binary Linear) kodlarda, diğer deyişle maksimum uzaklıkta ikili doğrusal kodlarda, ikili doğrusal dönüşümün maksimum dal sayısına sahip olabilmesi ancak doğrusal kodların maksimum uzaklığının alt ve üst sınırlarının birbirine eşit olması ile mümkündür.

Çizelge 3. 1. İkili 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

İkili $[2n, n]$ kodların maksimum dal sayıları incelendiğinde $n \times n$ boyutlu MDBL kodların (Nakahara & Abrahao, 2009) $n \leq 18$ olduğu durumlar için maksimum dal sayılarının erişilebilen değerlerinin ve teorik sınırların birbirine eşit olduğu görülürken, $n > 18$ olan durumlarda maximum dal sayısının tam olarak bilinmemesi erişilebilen değer ve teorik sınırdan ayrı ayrı söz edilmesini gerektirir.

Çizelge 3.1'den yola çıkarak örnek vermek gerekirse ikili [52,26] değerlerini alan (*n=26)* doğrusal kodların maksimum dal sayıları incelendiğinde "erişilebilir değerin" 10, "teorik sınırın" ise 12 olduğu görülür. Bu durumda 26×26 boyutlu bir ikili matris için erişilebilen dal sayısı 10'dur.

Maksimum uzaklıkta ayrılabilen ya da kısaca MDBL matrisler maksimum dal sayısına sahip olan $n \times n$ boyutlu ikili matrislerdir. Bu matrislere verilebilecek en bilinen örnekler ARIA, Camellia, PRIDE gibi modern blok şifrelerin tasarımında kullanılan ikili matrislerdir (Kwon vd., 2003; Aoki vd., 2000; Albrecht vd., 2014).

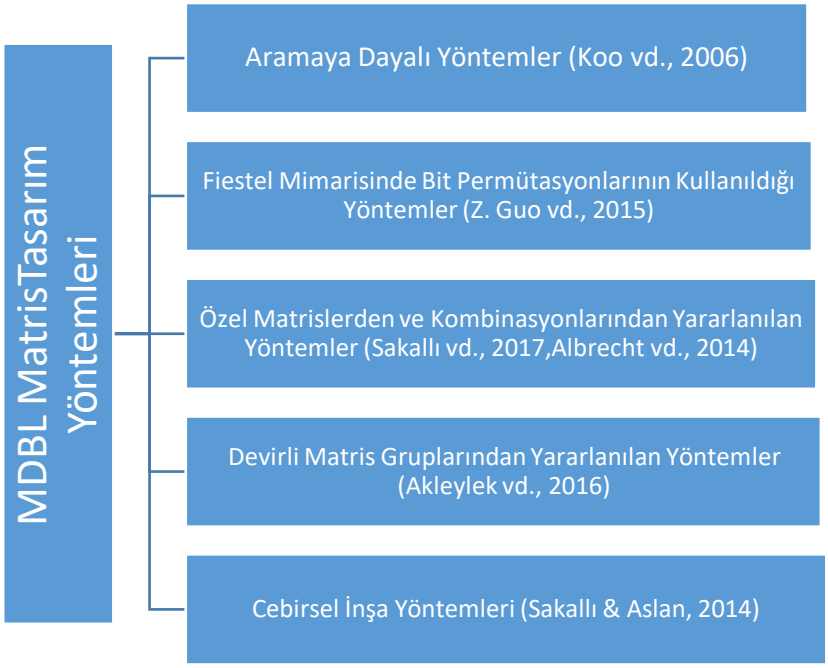
MDBL matrisler, modern blok şifreleme yöntemlerinde doğrusal dönüşüm katmanının kendisi veya temel bileşeni olarak yer alan matrislerdir. Genellikle bu yapı içinde difüzyon matrisleri olarak kullanılırlar. Difüzyon katmanında yer alan MDBL matrislere verilebilecek en bilinen örnekler:

- **ARIA:** 16 × 16 boyutlu ve aynı zamanda involutif MDBL matris (Kwon vd., 2003).
- **Camellia:** 8 × 8 boyutlu MDBL matris (Aoki vd., 2000).

MDBL matrisler pratikte blok şifrelerde kullanıldıklarında yalnızca XOR işlemiyle uygulanırlar. Bu durum MDBL matrisleri, XOR işlemine ek olarak tablo arama, öteleme gibi ölçeklendirme işlemleri de gerektiren MDS matrislerden daha avantajlı hale getirir. Fakat sadece XOR işlemine ihtiyaç duyan MDBL matrislerin eşit boyutlu bir MDS matrise kıyasla maksimum dal sayısının daha küçük olması MDS matrislere göre bir dezavantajdır (Nakahara & Abrahao, 2009).

3.1.6 MDBL Matrislerin Tasarım Yöntemleri

MDS matrislerde olduğu gibi MDBL matrisleri tasarlarken de yararlanılan çeşitli tasarım teknikleri mevcuttur. Bu teknikler, Şekil 3.2'deki gibi sınıflandırılabilir.



Şekil 3. 2. MDBL Matris Tasarım Yöntemleri

3.1.7 Hadamard ve Dairesel Matrisler Gibi Özel Matrislerden ve Bu Matrislerin Kombinasyonlarından Yararlanılarak MDBL Matrislerin Tasarımı

Bu bölümde Hadamard ve dairesel matrisler, bu matrislerin kombinasyonları (*BHwC matris yönteminde kullanılır*) ve dairesel matrislerin tekrarlayan türevleri (*BCwC matris yönteminde kullanılır*) kullanılarak MDBL matrislerin ve yüksek dal sayısına sahip matrislerin elde edilmesine yönelik tasarım yöntemlerine değinilecektir (Akleylek vd., 2017).

(BHwC-Binary Hadamard with Circulant) olarak adlandırılan bu yöntemde (k, $n \in \boldsymbol{Z^+}$) k ve n pozitif tam sayılar olmak üzere $2^k \times 2^k$ boyutlu Hadamard matrisinde $n \times n$ boyutlu dairesel alt matrislerden oluşan $(2^k \cdot n) \times (2^k \cdot n)$ boyutlu ikili matrislere, dairesel alt matrislerle oluşturulan Binary Hadamard matris denilir ve $(BHwC)_{k,n}$ ile gösterilir.

Örnek olarak C_0, C_1, C_2 ve C_3 , 4×4 boyutundaki ikili dairesel matrisler olsun. Bu durumda, 4×4 boyutundaki dairesel alt matrislerden oluşan $2^2 \times 2^2$ boyutlu Hadamard matris bloğu kullanılarak elde edilen 16×16 boyutlu $(BHwC)_{2,4}$ binary-ikili matrisi aşağıdaki şekilde gösterilebilir (Akleylek vd., 2017):

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

Bir diğer yöntem olan (*BCwC-Binary Circulant with Circulant*) yönteminde ise (t, $m \in \boldsymbol{Z^+}$) t ve m pozitif tam sayılar olmak üzere $t \times t$ boyutlu dairesel matrisin içinde $m \times m$ boyutlu olan dairesel alt matrislerin meydana getirdiği $(t \cdot m) \times (t \cdot m)$ boyutlu olan binary-ikili matrislere, dairesel alt matrislerle oluşturulan binary dairesel matris denilmektedir. Bu matrisler$(BHwC)_{t,m}$ ile ifade edilir.

Bu duruma örnek olarak $C_0, C_1, C_2 \; ve \; C_3$, 4×4 boyutundaki ikili dairesel matrisler olsun. Bundan yola çıkarsak 4×4 boyutundaki dairesel alt matrislerden oluşan 3×3 boyutlu dairesel matris bloğunun kullanılması ile elde edilen 12×12 boyutundaki $(BHwC)_{3,4}$ binary-ikili matrisi şu şekilde gösterilebilir (Akleylek vd., 2017):

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

3.1.8 Devirli Matris Gruplarından Yararlanılarak MDBL Matrislerin Tasarımı

Bu bölümde tanıtılan yöntemde ilk olarak $\mathbb{F}_2$ cisminde tanımlı $t \in \{3,4\}$ şeklinde üretilen $t \times t$ boyutlu matrislerin tamamının dereceleri hesaplanır. Daha sonra yine $\mathbb{F}_2$ cisminde tanımlı olan ve t dereceli indirgenemez polinoma ait köklerden biri ve derecesi $(2^t - 1)$'e kadar olan kuvvetlerin alınması ile $\mathbb{F}_2$'nin elemanları olan ikili matris karşılıkları oluşturulur. Bu yöntemle $t \times t$ boyutlu ve hepsi birbirinden farklı olan devirli matris gruplarının tamamı elde edilebilir.

Bu ikili matrislerin oluşturulmasının ardından 3×3 boyutlu ve $x^3 + x + 1$ indirgenemez polinomuna ait köklerinden biri ve derecesi 7 olan 8 adet, 4×4 boyutlu ve $x^4 + x + 1$ indirgenemez polinomuna ait köklerden biri ve derecesi 15 olan 336 tane devirli matris grubuna ulaşılabilmektedir. Ulaşılan bu matrisler üreteç matrislerdir (Tanım 2.19) ve dereceleri $(2^t - 1)$ olana dek kuvvetleri alınır. Bu yöntem basamakları takip edilerek $\mathbb{F}_2$ cisminde tanımlı olan elemanlara karşılık gelen ikili matrisler elde edilir.

Yukarıda belirtilen yöntemle elde edilen 3×3 boyutlu devirli matris grubuna ait elemanlar $\mathbb{F}_{2^3}$ cisminde yer alan $2 \times 2, 4 \times 4$ boyutlu Hadamard matrislerinin elemanlarıyla yer değiştirilir ve böylece 6×6 , 12×12 boyutlarında ikili matrisler oluşturulur. Bu yöntem, 4×4 boyutlu devirli matris grubuna ait elemanlar üzerinde uygulanırsa elemanları $\mathbb{F}_{2^4}$ cisminde yer alan 2×2 , 4×4 boyutlu Hadamard matrislerinin elemanlarıyla, yer değiştirilmesi gerekir. Yer değiştirme işleminden sonra 8×8 ve 16×16 boyutlu ikili matrisler elde edilir.

Örnek 3.1. $\mathbb{F}_{2^3}$ cisminde $x^3 + x + 1$ indirgenemez polinomunun kökü $g = \begin{pmatrix} 001 \\ 011 \\ 111 \end{pmatrix}$ matrisi olmak üzere $g^7 = I$ ise g üreteç matrisi ile üretilen devirli matris grubu ve $\mathbb{F}_{2^3}^*$ cismi (0 elemanı hariç) izomorfiktir.

$\mathbb{F}_{2^3}/x^3 + x + 1$ üzerine tanımlı involutif 2×2 boyutlu $A = had(4_h, 5_h) = had(g^2, g^6)$ matrisini düşünelim. Bu durumda, $\mathbb{F}_{2^3}$ cisminin elemanları devirli matris grubuna karşılık gelen 3×3 boyutlu matrislerle yer değiştirilerek, dal sayısı 4 olan 6×6 boyutlu involutif ikili A matrisi aşağıdaki gibi elde edilebilir (Akleylek vd., 2016):

$$A = \begin{bmatrix} g^2 & g^6 \\ g^6 & g^2 \end{bmatrix} = \begin{bmatrix} 110010 \\ 001011 \\ 101100 \\ 010110 \\ 011001 \\ 100101 \end{bmatrix}.$$

3.1.9 Cebirsel Yöntemlerle MDBL Matrislerin Tasarımı

Bu yöntem, $\mathbb{F}_{2^4}$ sonlu cisim elemanlarının 4×4 boyutunda ikili matris karşılıkları elde edilerek bu ikili matrislerin 2×2, 4×4 ve 8×8 boyutlu matrislere uygulanması sonucu sırasıyla $8 \times 8, 16 \times 16$ ve 32×32 boyutlu maksimum dal sayısına sahip MDBL matrislerin oluşturulmasına olanak sağlayan bir yöntemdir (Aslan, 2013).

Örnek 3.2. $\mathbb{F}_{2^4}$ sonlu cismi $p(x) = x^4 + x + 1$ indirgenemez polinomu ile tanımlansın ve *x*, $p(x)$ indirgenemez polinomunun bir kökü olsun. Herhangi bir eleman $\gamma \epsilon \mathbb{F}_{2^4}$, $\gamma_i \in \mathbb{F}_2$ ve $(x_3, x_2, x_1, x_0) = (x^3, x^2, x^1, 1)$ $\mathbb{F}_{2^4}$'ün $\mathbb{F}_2$ üzerine polinom taban olmak üzere $\gamma = \gamma_3 x^3 + \gamma_2 x^2 + \gamma_1 x + \gamma_0$ şeklinde yazılabilir. 2_h ya da *x* sonlu cisim elemanının herhangi bir sonlu cisim elemanı $\gamma \epsilon \mathbb{F}_{2^4}$ ile çarpımı aşağıdaki gibi verilebilir:

$$x(\gamma_3 x^3 + \gamma_2 x^2 + \gamma_1 x^1 + \gamma_0) \bmod p(x) = \gamma_3 x^4 + \gamma_2 x^3 + \gamma_1 x^2 + \gamma_0 x$$

$$= \gamma_2 x^3 + \gamma_1 x^2 + \gamma_0 x + \gamma_3 x + \gamma_3$$

$$= \gamma_2 x^3 + \gamma_1 x^2 + x(\gamma_3 + \gamma_0) + \gamma_3.$$

Bu eşitliklerden yararlanarak aşağıda yer alan 4×4 boyutlu ikili matris aşağıdaki gibi elde edilir:

$$\begin{bmatrix} \gamma_0' \\ \gamma_1' \\ \gamma_2' \\ \gamma_3' \end{bmatrix} = \begin{bmatrix} 0\,0\,0\,1 \\ 1\,0\,0\,1 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \end{bmatrix} . \begin{bmatrix} \gamma_0 \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}.$$

Bu işlemleri $\mathbb{F}_{2^4}$'ün diğer elemanlarına da uygulayarak her eleman için 4×4'lük ikili matris karşılığı elde edilir. Aynı yöntem $\mathbb{F}_{2^4}$ üzerine tanımlı 0×19 ve $0 \times 1f$ indirgenemez polinomları için de uygulanabilir. Son olarak 4×4 'lük ikili matrisler, $\mathbb{F}_{2^4}$ cisminde tanımlı olan $2 \times 2, 4 \times 4$ ve 8×8 boyutlu matrisler ile yer değiştirilerek sırası ile $8 \times 8,\ 16 \times 16,\ 32 \times 32$ boyutlarındaki ikili matrislere ulaşılır (Aslan, 2013; Tuncay, 2021).

3.2 XOR Sayısı

Tanım 2.24'ten yola çıkarak XOR(a) değeri, ikili matrisler için, a elemanına karşılık gelen $n \times n$ boyutlu ikili matristeki 1'lerin sayısından n değerinin çıkarılması olarak ifade edilebilir.

XOR sayısının önemi özellikle donanım alanında maliyet düşünüldüğünde ortaya çıkmaktadır. Bunun nedeni yazılım uygulamalarından farklı olarak donanım uygulamalarının alan gereksinimine duydukları ihtiyaçtır. Verimlilik ve maliyet

açısından iyi bir uygulama XOR sayısının minimum değerlerde tutulması ile mümkün olabilir.

Yaygın olarak bilinmektedir ki düşük Hamming ağırlığı (Tanım 2.5), uygulamalarda maliyet açısından daha az yük getirmektedir (Örneğin AES şifreleme algoritması) (Tuncay, 2021). Ancak yüksek Hamming ağırlığına sahip katsayılarda düşük XOR maliyetinin hesaplanabildiği ve bu durumun sonlu cismi tanımlayan indirgenemez polinoma bağlı olduğu (Khoo vd., 2014)'nin çalışmasında belirtilmektedir.

Örnek 3.3. $\mathbb{F}_{2^4}/0x13$'de bir cisim elemanını $a = 6_h = a^2 + a = (0,1,1,0)$ ve rastgele bir cisim elemanı b'yi $(b = b_3, b_2, b_1, b_0)$ düşünelim. Bu durumda bu iki elemanın çarpma işlemi için gerekli XOR sayısı aşağıdaki şekilde hesaplanabilir:

$$(a^2 + a) \cdot (b_3a^3 + b_2a^2 + b_1a + b_0)$$

$$= (b_3a^5 + b_2a^4 + b_1a^3 + b_0a^2) + (b_3a^4 + b_2a^3 + b_1a^2 + b_0a)$$

$0x13$ $i$ndirgenemez polinomu için $a^5 = a^2 + a, a^4 = a + 1$ olduğundan;

$$= (b_3a^2 + b_3a + b_2a + b_2 + b_1a^3 + b_0a^2) + (b_3a + b_3 + b_2a^3 + b_1a^2 + b_0a)$$

$$= (b_1 + b_2)a^3 + (b_0 + b_1 + b_3)a^2 + (b_0 + b_2)\, a + (b_2 + b_3).$$

Başka bir gösterimle;

$$(0,1,1,0) \cdot (b_3, b_2, b_1, b_0) = (b_1 + b_2,\ b_0 + b_1 + b_3,\ b_0 + b_2,\ b_2 + b_3)$$

sonucu elde edilir. Bu sonuçtan yola çıkarak $\mathbb{F}_{2^4}/0x13$'de tanımlı olan $a = 6_h$ elamanının herhangi bir cisim elemanı ile çarpma işlemin maliyetinin 5 XOR olduğu hesaplanabilir.

3.3 Derinlik

Tanım 3.1: Bir devrenin giriş ve çıkış kapıları arasındaki en uzun yola denir. Bu yolda bulunan XOR sayıları o devrenin derinliğinin ölçütüdür. Bir şifreleme uygulamasında iki önemli bileşen olan hız ve güçlülük ilkelerinden hız ölçütü devre derinliği ile ilişkilidir. Bu nedenle hızın önemli bir kıstas olarak belirlendiği şifreleme uygulamalarında devrenin derinliğinin minimum seviyede olması ve dolayısıyla bu devrenin mümkün olan en az XOR sayısı ile tasarlanması amaçlanır. XOR sayılarının düşürülebilmesi için ise optimizasyon tekniklerinden yararlanılabilir.

BÖLÜM 4

4. OPTİMİZASYON TEKNİKLERİ

Blok şifreleme algoritmalarında difüzyon tabakasında kullanılmak üzere tasarlanan involütif MDS matrislerin donanımsal maliyetlerinin en az olması istenir. Bu durumda odaklanılması gereken noktalardan biri XOR sayısıdır. Minimum sayıda XOR 'a sahip şifreleme uygulamasının elde edilmesi kolay değildir fakat bunun yanı sıra bazı optimizasyon teknikleri kullanılarak elde edilen şifreleme uygulamasına ait XOR sayılarının ve dolayısıyla maliyetin de düşürülmesi sağlanabilmektedir.

XOR sayısının düşürülebilmesi için uygulanan optimizasyon teknikleri yerel (local) ve global optimizasyon olmak üzere iki başlık altında incelenebilir.

4.1 Yerel Optimizasyon

Yerel optimizasyon tekniğinde XOR sayılarının azaltılabilmesi için matrise ait elemanlar tek tek ele alınır ve içlerinden en az maliyete sahip elemanlar seçilerek şifreleme uygulamasında toplam XOR sayısı ile maliyetin düşürülmesi hedeflenir (Pehlivanoğlu vd., 2021).

Yerel optimizasyon teknikleri uygulanırken doğrudan ve sıralı XOR sayılarının dikkate alındığı iki farklı yöntem vardır. Bu yöntemlerden en popüler yöntem olan, doğrudan XOR sayısına odaklanan yöntemdir ki bu yöntemde matrise ait elemanların XOR maliyetleri tek tek hesaplanır ve bu yöntem bize ham XOR sayısını verir. Bunun yanı sıra, sıralı bir uygulamada bulunan elemanların, fazladan bellek alanı işgal edilmeden yerinde işlenmeleriyle en az XOR sayısının hedeflendiği diğer bir yöntemde odaklanılan ise sıralı XOR sayısıdır (Khoo vd., 2014; Jean vd., 2017).

Doğrudan XOR sayısı bize elemanları ile uygulama maliyetleri hakkında bir fikir yürütme olanağı sunması açısından kullanışlıdır fakat sıralı XOR sayısına odaklanıldığında görülür ki gerekli olan XOR sayısı aslında daha az olabilmektedir.

Doğrudan XOR sayısı, $\mathbb{F}_{2^m}$ cisminde tanımlı indirgenemez $p(x)$ polinomunun elemanı olan a $(a(\in px))$'nın, b sonlu cismi ile olan çarpımı için gerekli olan XOR sayılarını ifade eder. Örnek 3.3'de yapılan hesaplamalar sonucu elde edilen XOR sayısı da ham ya da diğer adıyla doğrudan XOR sayısıdır.

Sıralı XOR sayısının hesaplanması doğrudan XOR sayısına göre daha zor olsa bile çoğunlukla gerekli gerçek optimal XOR sayısının tahmin edilmesi açısından daha güvenilirdir.

Doğrusal bir katman uygularken, ilgili matrisin bu metriklere göre optimal olacağı şekilde bir alan temsili seçilebilir. Bu şekilde, belirli bir lineer katmanın performansı geliştirilebilir.

Kısaca anlatmak istenirse, doğrudan XOR sayısı matriste bulunan sıfır olmayan elemanların sayısı iken, sıralı XOR sayısı matrisi birim matrise dönüştürmek için gereken temel satır işlemlerinin sayısıdır denilebilir (Kölsch, 2019).

4.2 Global Optimizasyon

Yerel optimizasyonların global olarak en uygun seçenek olmamasının kesinleşmesi üzerine, araştırmalar global optimizasyon tekniklerinin uygulanmasına yönelmiştir.

Global optimizasyon teknikleri şu anda verimliliği bilinen matrislerin XOR sayıları üzerinde çalışarak mevcut XOR sayısını iyileştirmeye yönelik araçlara veya detaylı aramalara dayanır (Kölsch, 2019).

4.2.1 Shortest Linear Straight Line Program-SLP

Bir dizi doğrusal formun hesaplanmasında gerekli olan işlemlerin sayısının minimuma indirilmesi problemine En Kısa Doğrusal Program (SLP) adı verilir.

A, $\mathbb{F}_2'$de tanımlı $m \times n$ boyutlu bir sabitler matrisi ve x, $\mathbb{F}_2'$de tanımlı n değişkenin bir vektörü olsun. Bu durumda SLP, belirli bir biçimde olan her bir program satırı için $A \cdot x$'i hesaplayan en küçük satır sayısını veren programı bulmaktır (Stoffelen, 2016).

V, $\{x_0,\ldots,x_{n-1}\}$ girdisini içeren $\mathbb{F}_2{}'$de tanımlı bir değişkenler kümesi olsun, $v_i, v_j \in V$ olmak üzere her program satırı

$$v' \coloneqq v_i + v_j$$

formundadır.

Bu program satırının çalıştırılmasıyla birlikte yeni değişken v' de kümeye eklenir. Bu durum

$$V \coloneqq V \cup \{v'\}$$

Şeklinde ifade edilebilir.

Yeni değişkenimiz v' artık bir sonraki program satırında kullanılabilir durumdadır. Program $\exists(v_1, \ldots, v_m) \in V^m$ koşulu sağlandığında $A \cdot x$'i hesaplayacaktır. Bu durumda

$$A \cdot x = (v_1, \ldots, v_m)^T \text{ dir.}$$

En Kısa Doğrusal Program (SLP) kriptolojide en bilinen karmaşık ve zor problemlerdendir ve bu problem $\mathbb{F}_2{}'$de sadece XOR kapılarının kullanılmasıyla bir fonksiyonun hesaplanmasında en kısa devre yolunu bulmaya eşittir denilebilir.

SLP problemine bir örnek olarak da AES şifreleme algoritmasında bulunan şifreleme adımlarından MixColumns ve SubBytes aşamalarında doğrusal dönüşüm gibi doğrusal işlemlerin optimizasyonu verilebilir. Fakat bu teknik S-Kutuları gibi doğrusal olmayan işlemlerde işe yaramadığından hem yazılım hem de donanımda yer alan S-kutularının optimizasyonunda da SAT (Satisfiability Solver) çözücüler kullanılır.

Fuhs ve Schneider-Kamp (2010)'in sunduğu bir yönteme göre, SLP probleminin bir SAT örneği olarak kodlanabilmesi ve bu doğrultuda AES şifreleme algoritmasının adımlarından olan SubByte adımının affine dönüşümünü optimize etmek mümkündür. Bu yöntemde yine benzer bir yaklaşımla Ko Stoffelen (2016)'in sunduğu yöntemle SAT çözücüler S-Box 'ların karmaşıklıklarını minimuma indirebilmek için kullanılmıştır. Fakat bu yöntemler küçük örneklerde işe yararken büyük örnekler söz konusu olduğunda uygulanamaz hale gelirler.

Daha büyük boyutlu matrisler söz konusu olduğunda ise yaygın olarak kullanılan iki yaklaşım vardır. Bunlardan ilki Paar (Paar, 1997) algoritması iken diğeri Boyar-Peralta (Boyar vd., 2013) bir başka adıyla BP Algoritmasıdır.

4.2.2 Paar Algoritması

Açgözlü algoritmalara bir örnek olan Paar Algoritması da diğer açgözlü algoritma örneklerinde olduğu gibi sonuca varmaya en yakın olan seçime yönlenir. Fakat bu seçim her zaman en doğru seçim demek değildir bu nedenle Paar Algoritmasının en uygun sonuçları verdiği söylenemese de zamanın ön planda olduğu çalışmalarda BP algoritmasına oranla büyük ölçekli matrisler üzerinde XOR sayısının azaltılmasında işe yaradığını söylemek mümkündür.

A, XOR sayısı en aza indirilmek istenen matris olsun. Bu durumda *A* Matrisine Paar algoritması uygulanmak istendiğinde şu adımlar takip edilir:

1. İlk olarak Hamming ağırlığı en fazla olan sütunları AND işlemi kullanılarak belirlenirler.
2. Bu işlem ile XOR işlemleri sırasında en fazla tekrarlanan x_i ve x_j elemanlar tespit edilmiş olur.
3. Yukarıdaki adımlar ile elde edilmiş olan çarpım sütununa "newClmn" adı verilir ve bu sütun matrise eklenerek ilgili *A* matrisi genişletilmiş olur. Bu şekilde *A* matrisinin genişletilmesiyle yeni giriş değişkeni (v) elde edilirken diğer taraftan $v = x_i \oplus x_j$ işlemini de XOR işlemlerine dâhil eder.
4. Matristeki eski sütunlara "oldClmn" adı verilir. Bu durumda eski sütunlara oldClmn ← oldClmn**.** $\overline{\text{newClmn}}$ işlemi uygulanır.

Uygulanan bu son işlem ile gereksiz durumdaki XOR işlemleri kaldırılmış olur.

Örnek 4.1.

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

Görüldüğü üzere yukarıda yer alan sistemin birinci ve üçüncü sütunlar en büyük Hamming ağırlığına sahip olan sütunlardır. Bu durumda; $v_1 = x_1 \oplus x_3$ olur.

$(1\ 1\ 1\ 1).(0\ 0\ 0\ 0) = (0\ 0\ 0\ 0)$ işlemi ile yeni çarpım sütunu elde edilir.

Yukarıda bulunan adımlardan 4. Adımda bahsedildiği üzere v_1 yeni giriş değişkeni olarak giriş değişkenlerine eklenir ve elde edilen çarpım sütunu ile matris genişletilir. Hamming ağırlığı en fazla olan birinci ve üçüncü sütunlar için

oldClmn ← oldClmn. $\overline{\text{newClmn}}$ işlemi uygulandığında ise oluşan son durum aşağıdaki gibidir.

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

Yukarıda yer alan işlemde yer alan tüm terimler iki kez görünene dek işleme devam edilir. Bu işlemler sırasında gerçekleşen XOR sayılarının toplamı ise bize XOR maliyetinin değerini verir (Banik vd., 2019).

4.2.3 Boyar-Peralta (BP) Algoritması

Daha büyük boyutlu matrisler söz konusu olduğunda akla gelen bir diğer optimizasyon tekniği olan Boyar-Peralta ya da diğer adıyla BP algoritması, $\mathbb{F}_2$ cismi üzerinde tanımlı olan $m \times n$ boyutlu bir A matrisi için $f(x) = A \cdot x$ değerini hesaplar. Böylece en kısa linear düz çizgi programını bulmayı hedeflemektedir (Tuncay, 2021).

Bu bahsi geçen en kısa yola ulaşma söz konusu olduğunda öncelikle bilinen linear fonksiyonların bir "tabanı" olan ve başlangıç olarak yalnızca $(x_1, x_2, x_3 \ldots, x_n)$ giriş

değişkenlerinden oluşan M kümesi oluşturulmaktadır. Daha sonra, A matrisinin satırlarının M kümesine olan doğrusal fonksiyonların Hamming uzaklıklarından oluşan bir uzaklık vektörü olarak Uzk[·] tanımlanır. Bu durumda eğer f_i A matrisinin i satırı tarafından verilen doğrusal fonksiyon ise, o zaman Uzk[i], M tabanından f_i'yi elde edebilmek için gerekli olan minimum fonksiyon sayısını verir.

Başlangıçta Uzk[i], i satırının Hamming ağırlığından yalnızca 1 eksiktir ve Uzk[i] vektörünün tanımlanmasının ardından ise aşağıdaki adımlar bir döngü şeklinde uygulanır.

1. M kümesinden iki adet temel eleman belirli özelliklere bakılarak seçilir ve bu elemanların XOR işlemine tabi tutulmasının ardından elde edilen yeni eleman tekrar M kümesine eklenir.
2. Uzk[i] vektör kümesi de M kümesine eklenen yeni elemandan dolayı güncellenir.
3. i 'nin aldığı tüm değerler için Uzk[i]=0 sonucu elde edilinceye dek yukarıda yer alan adımlar tekrar edilir.

Uygulamanın herhangi bir aşamasında eğer M kümesinin boyutu t ise yeni bir temel eleman seçilmesi için $\binom{t}{2}$ adet opsiyon bulunur. Bunlar:

1. Güncellenmiş Uzk[·] dizisinin öğelerinin toplamını en aza indiren elemanlardan birisinin seçilmesi
2. Yeni temel öğenin iki seçeneği arasında eşitlik gibi bir bağ varsa, güncellenmiş Uzk[·] dizisinin Öklid normunu maksimize eden temel öğeyi seçerek bu sorunun ortadan kaldırılması.

Bu eşitlik durumu gibi iki eleman arasında mevcut olması muhtemel bağın çözümü için önerilen yöntem her ne kadar sezgisel görünse de bu yöntemin mantığı ve temelde yola çıkışı, [0, 0, 3, 1] gibi bir uzaklık vektörünün, [1, 1, 1, 1] şeklindeki başka bir uzaklık vektörüne tercih edilmesine dayanmaktadır. [1, 1, 1, 1] vektörünün tercih edilmesi söz konusu olduğunda fazladan 4 adet XOR kapısına gerek duyulacakken [0, 0, 3, 1] vektörü tercih edilerek 3 XOR kapısı ile Uzk[·]=0 eşitliğine ulaşılabilmektedir.

Bu durumda Boyar-Peralta yönteminin zamanının büyük kısmı yeni bir temel elemanın seçilmesinde harcanmaktadır. Bu durum zaman maliyetinin hesaplanmasında dikkate alınmalıdır.

Örnek 4.2. Aşağıdaki gibi tanımlanan denklem sistemini hesaplayan bir devreye ihtiyacımız olduğunu varsayalım. Verilen bu doğrusal sistemin çözümü sol tarafta verilmiş olan 6 × 5 boyutlu *A* matrisi ile 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 hedef fonksiyonlar *A* matrisinin satırları olarak verilmiştir. İlk temel küme *M*= {10000, 01000, 00100, 00010, 00001}'e karşılık gelen $\{x_0, x_1, x_2, x_3, x_4\}$ ile verilen değerlerdir. Bu durumda ilk uzaklık vektörü Uzk = [2, 2, 3, 2, 2, 3] şeklindedir. Algoritma toplam değerini *M* kümesine eklediğinde yeni mesafelerin toplamını en aza indiren iki temel vektör bulur. Yapılan hesaplamaların sonucunda en doğru seçimin $t_5 = x_1 + x_3$ olduğu görülür. Bu durumda yeni taban *M* kümesi, 01010'i içerecek şekilde genişletilir ve yeni uzaklık vektörü Uzk = [2, 1, 3, 1, 1, 2] 'ye evrilir.

Adım 1- Algoritmaya göre $x_1 \oplus x_2$ veya $x_0 \oplus x_5$ vektörleri arasında seçenekler mevcuttur. Bu seçenekler arasından bir seçim yapıldığından yeni Uzk vektörleri sırasıyla Uzk = [1, 1, 3, 1, 1, 2] ve Uzk = [2, 1, 3, 1, 0, 2] olur. Her iki uzaklık vektörünün toplamı 9'dur ve bu durumda eşitlik söz konusu olacaktır. Bu durumda vektörlerin Öklid normlarına bakılır. Öklid normları birinci seçenekte $\sqrt{17}$ ikinci seçenekte ise $\sqrt{19}$ olarak hesaplandığından ikinci seçenek $t_6 = x_0 \oplus x_5 = y_4$ değeri seçilir.

Adım 2- $t_7 = x_2 \oplus t_5 = y_3$, yeni Uzk = [2, 1, 3, 0, 0, 1]

Adım 3- $t_8 = x_4 \oplus t_5 = y_1$, yeni Uzk = [2, 0, 3, 0, 0, 1]

Adım 4- $t_9 = x_2 \oplus t_8 = y_5$, yeni Uzk = [2, 0, 2, 0, 0, 0]

Adım 5- $t_{10} = x_0 \oplus x_1$, yeni Uzk = [1, 0, 1, 0, 0, 0]

Adım 6-$t_{11} = x_2 \oplus t_{10} = y_0$, yeni Uzk = [0, 0, 1, 0, 0, 0]

Adım 7- $t_{12} = t_8 \oplus t_{11} = y_2$, yeni Uzk = [0, 0, 0, 0, 0, 0]

Son durumda devre 8 XOR kapısı ile tamamlanmış olur (Banik vd., 2019).

Optimizasyon teknikleri başlığı altında bahsedilen tekniklerde difüzyon matrisinin uygulanması için gerekli olan XOR sayısının minimum yapılması ve bu şekilde devrenin iyileştirilmesi için kullanılan yöntemlerden bahsedilmiştir. Fakat söz konusu matrislerde dikkat edilmesi gereken bir diğer nokta olan gecikmedir. Gecikmenin yaşanması devrenin derinliği ile ilgilidir. Derinlik bir devreyi oluşturan en uzak mesafeye sahip yolun uzunluğudur. Bu yol maksimum sayıda XOR içeren yoldur. XOR sayıları bize devrenin derinliğini vermektedir (Pehlivanoğlu & Kavun, 2020).

BÖLÜM 5

5. METOT, MATERYALVE DENEYSEL BULGULAR

Tezin bu bölümünde çeşitli kodlarla elde edilen farklı MDBL matris formlarının Ham XOR sayıları ve optimizasyon sonrası XOR sayılarının karşılaştırılması ve şifreleme algoritmalarında kullanılmak üzere uygun maliyetli MDBL matrislerin elde edilmesine yönelik yapılan çalışmalar aktarılmıştır.

Öncelikle $\mathbb{F}_{2^m}$ sonlu cismi üzerinde farklı tasarım yöntemleriyle çeşitli boyutlarda MDBL matrislerin elde edilmesinde hazırlanan kodların sayılar teorisi ve geometri alanlarındaki hesaplamalar için tasarlanan "MAGMA" yazılımı kullanılmıştır ("Magma Computational Algebra System", 2021).

Aşağıda tez yazım aşamasında MAGMA dilinde yazılan bazı kod parçaları ve bu kod parçalarının kullanım amaçları paylaşılmıştır.

Örnek 5.1. $\mathbb{F}_2$ sonlu cisminde tanımlanan 5×5 boyutlu ikili *M* matrisinin MAGMA'da tanımlanması aşağıdaki gibi olmaktadır.

```
n:=5;
M:=Matrix(GF(2), 5, 5, [
0, 1, 1, 0, 1,
1, 0, 0, 0, 0,
1, 1, 0, 0, 0,
0, 1, 1, 0, 1,
0, 0, 1, 1, 0
]);
```

Örnek 5.2. Yukarıda verilen *M* matrisinin transpozuna ait dal sayısına ulaşmak için aşağıda yer alan kod parçası kullanılmıştır.

```
I:= IdentityMatrix(GF(2),5);
C:= LinearCode(HorizontalJoin(I,M));
printf("Dal Sayisi: ");
MinimumWeight(C);
C:= HorizontalJoin(I, Transpose(B));
D:= LinearCode(C);
printf("Transpoz Dal Sayisi: ");
MinimumWeight(D);
```

Yukarıda yer alan kod parçalarından da yararlanılarak oluşturulan kodlar MAGMA aracılığı ile derlenmiş ve farklı tekniklerle elde edilmiş olan matrislerin ham XOR sayıları da hesaplatılmıştır. Bu şekilde elde edilen matrislerden ham XOR sayıları düşük ve yayılımları iyi olan matrisler seçilmiş ve seçilen matrislere BP optimizasyonu derinliğin dikkate alındığı ve alınmadığı durumlar için ayrı ayrı uygulanarak XOR sayısında meydana gelen değişimler değerlendirilmiştir.

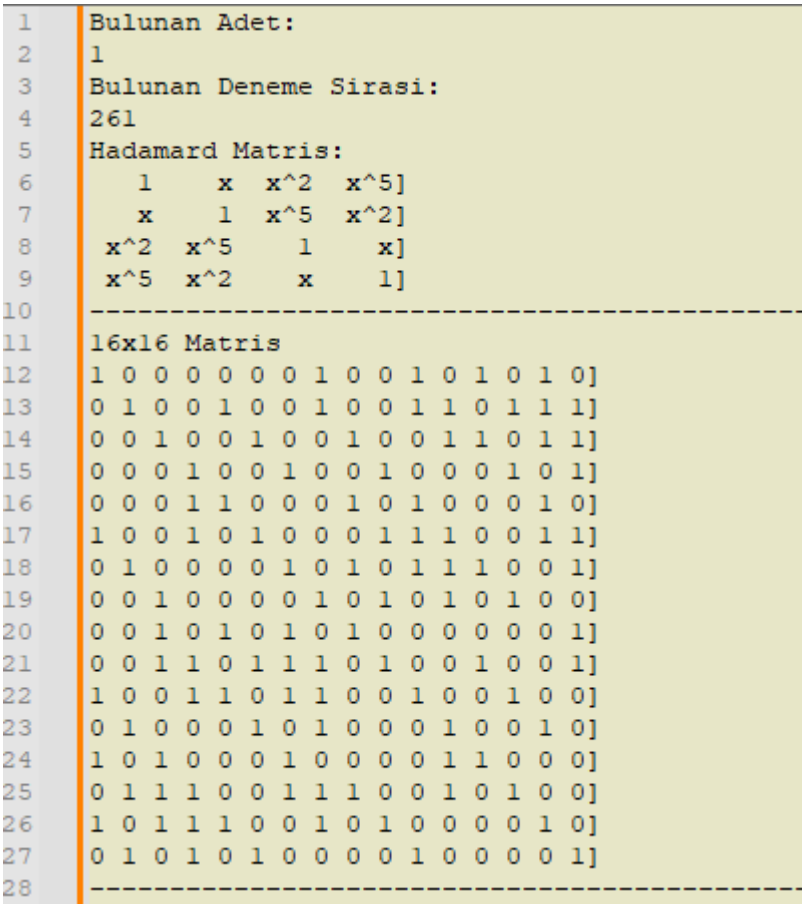
5.1 MDBL Matrislere Ait Deneysel Bulgular

Bu bölümde MDBL matrislerin tasarımında uygulanan yönteme ait detaylara, elde edilen matrislerin BP optimizasyon algoritması sonrası bulgularına ve karşılaştırmalara yer verilmiştir.

Deneyler kapsamında Dairesel, Hadamard, Ghadamard ve Toeplitz matris türlerinden yararlanılarak $16 \times 16, 20 \times 20$ ve 24×24 boyutlarında ikili MDBL matrisler üretilmiş ve kullanılmıştır.

MDBL matrislerin elde edilmesi için yürütülen çalışmalar hazırlanan test ortamı (XOR Ölçütü Hesaplama ve Optimizasyon Yazılımı) üzerinden ilerlemiştir. Kaba kuvvet arama tekniği kullanılarak hazırlanan kod, MAGMA'da derlenmiş ve BP ve Paar optimizasyon teknikleri, hazırlanan test ortamı kullanılarak elde edilen 16×16, 20×20, 24×24 boyutlu matrislere uygulanmıştır.

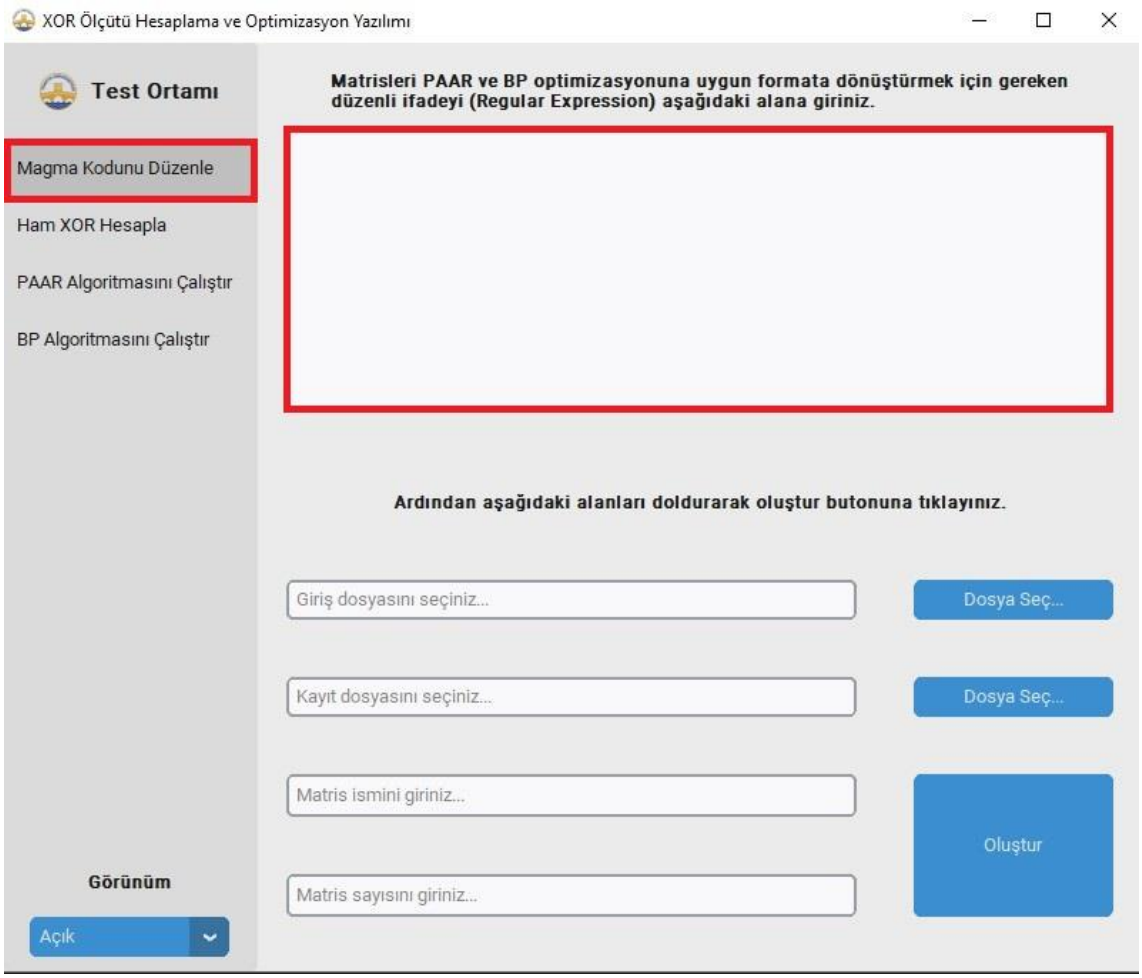
Örnek 5.3. $\mathbb{F}_{2^4}$ sonlu cisim elemanı ve $x^4 + x + 1$ indirgenemez polinomu üzerinde tanımlı elemanlar üzerinde kaba kuvvet arama tekniği ile uygun elemanlar aranmış yazılan kod MAGMA'da derlendikten sonra alınan çıktıda görülen 4×4 boyutlu A Hadamard matrisi ve söz konusu matristen elde edilen dal sayısı 5, 16×16 olan A_{1B} matrisi Şekil 5.1'de gösterilmiştir.



Şekil 5. 1. 4×4 boyutlu A ve 16×16 Boyutlu Dal Sayısı 5 olan A_{1B} Matrislerine Ait MAGMA Çıktısı

MAGMA ile elde edilen ve Şekil 5.1'de yer alan çıktı daha sonra hazırlanan test ortamı aracılığı ile PAAR ve BP algoritmalarına uygun hale getirilmiş ve ayrıca ham XOR sayıları hesaplanmıştır.

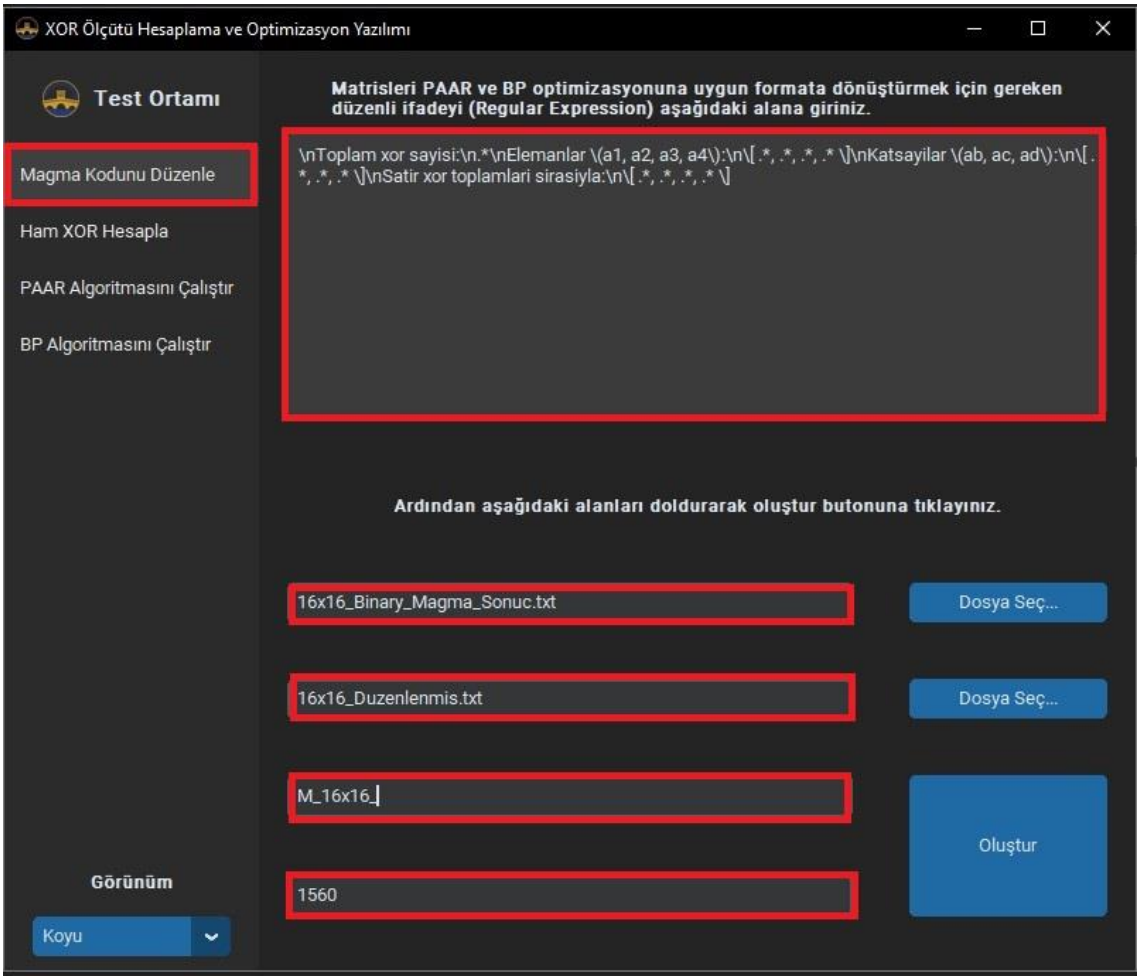
Tez çalışmasının bu kısmında hazırlanan test ortamına ilişkin görüntüler ve detaylı anlatıma yer verilecektir.



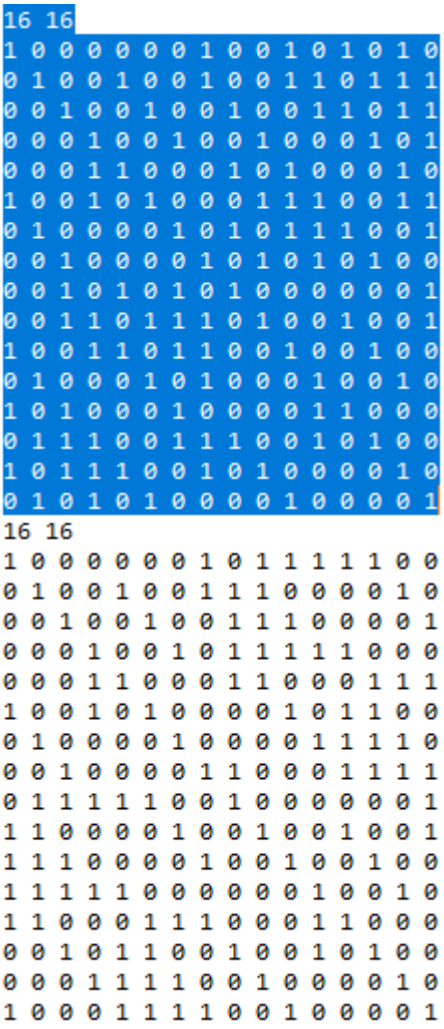
Şekil 5. 2. Test Ortamı XOR Ölçütü Hesaplama ve Optimizasyon Yazılımı

Şekil 5.2'de görülen ekran görüntüsünde yer alan "Magma Kodunu Düzenle" butonu ile Şekil 5.1'de bir kısmı görülen çıktı dosyasının Paar ve BP algoritmaları için düzenlenmesi adına "Dosya Seç" seçeneği ile test ortamına yüklenebileceği gibi, söz konusu MAGMA kodu çerçeve içinde görülen alana da direkt yazılabilir. MAGMA kodu ortama yüklendikten sonra "Kayıt dosyası", "Matris ismi" ve "Matris sayısı" gibi ilgili alanlar doldurulmalıdır. Bu işlemin ardından "Magma kodunu Düzenle" butonu kullanılır ve MAGMA kodu bu işlemin ardından Paar ve BP algoritmalarına uygun hale getirilmiş olmaktadır. Şekil 5.3.'te düzenleme öncesi giriş ve Şekil 5.4'te ise düzenleme sonrası BP algoritması için uygun hale gelen A_{1B} matrisine ait kod parçası görülmektedir.

Hazırladığımız test ortamında "Görünüm" seçeneği ile tema açık ve koyu olarak tercih edilebilmektedir. Çalışmanın sonraki aşamalarına koyu tema ile devam edilmiştir.

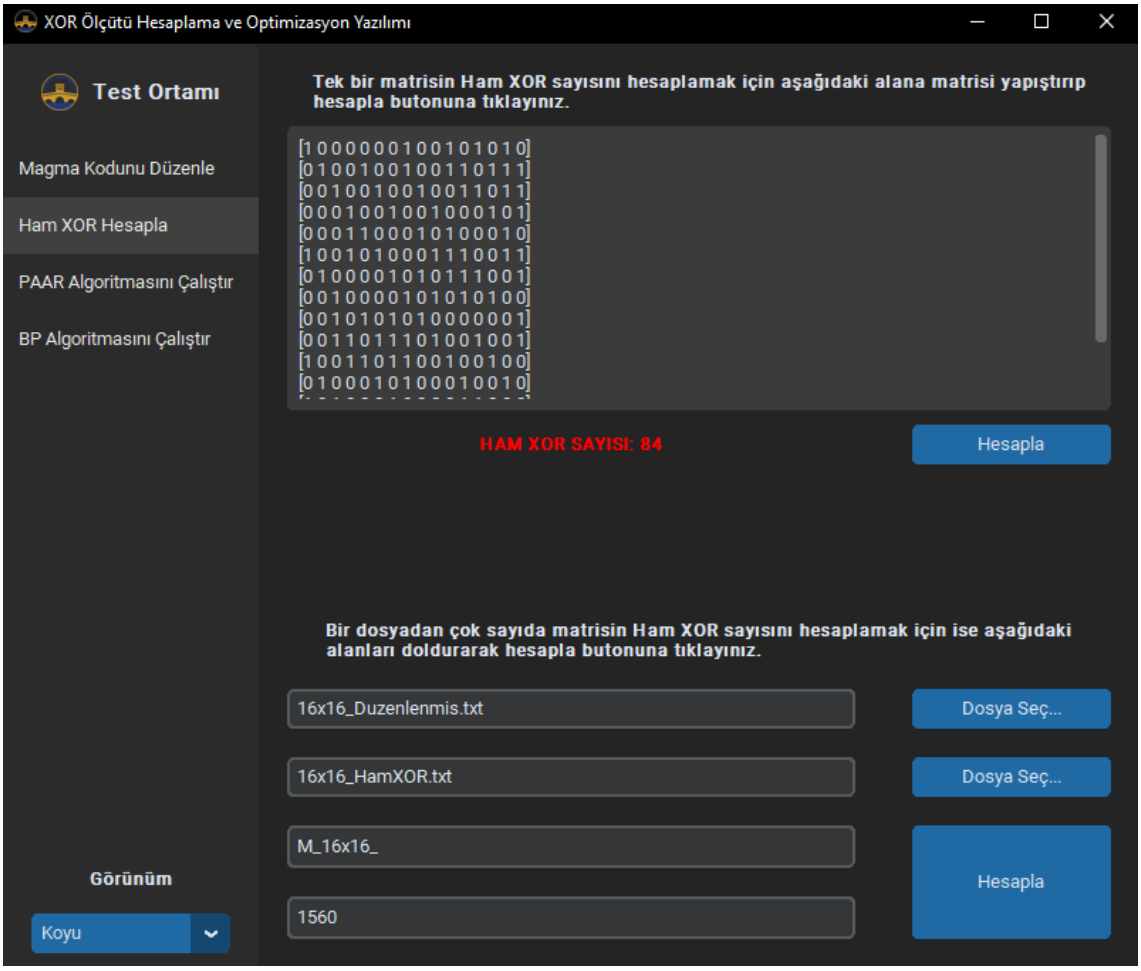


Şekil 5. 3. Test Ortamına eklenen MAGMA Kodu ve İlgili Alanların Düzenlenmesi



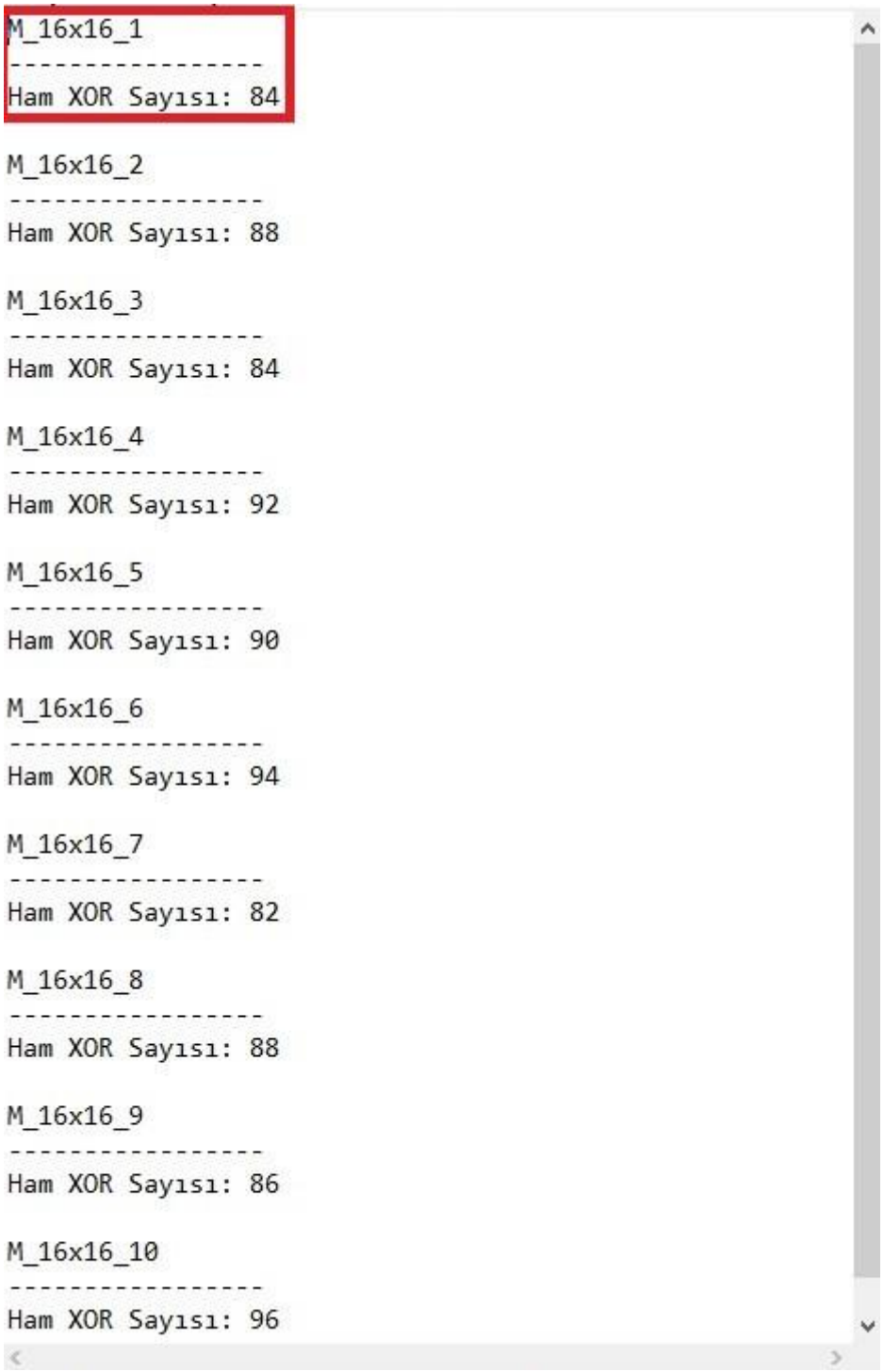
Şekil 5. 4. Test Ortamında Yapılan Düzenleme Sonrası BP Algoritmasına Uygun Hale Gelen A_{1B} Matrisi

MAGMA kodunun düzenlenmesi tamamlandıktan sonra ikili matrislerin ham XOR sayıları yine test ortamı kullanılarak hesaplanmıştır. Şekil 5.5'te ham XOR sayısının hesaplanma ekranı gösterilmiştir.



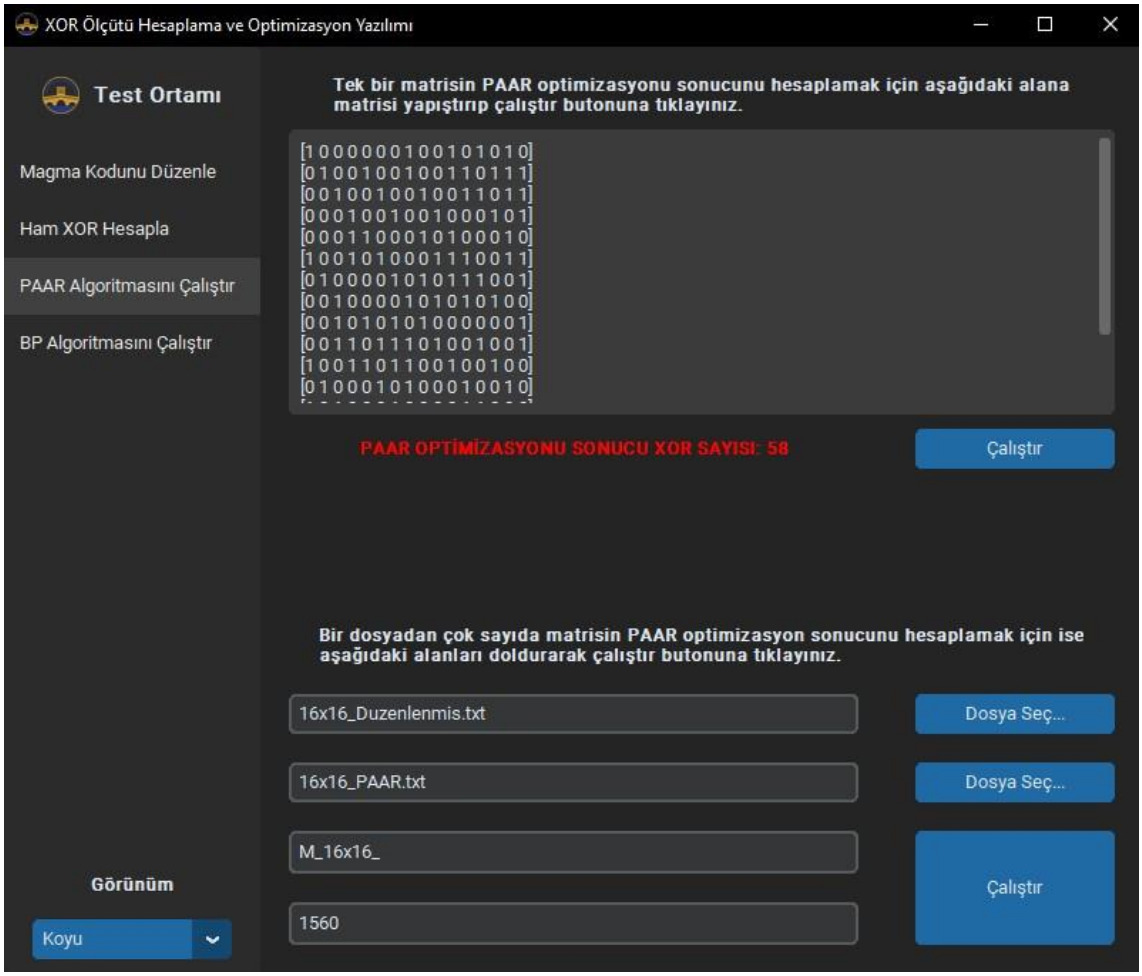
Şekil 5. 5. 16 × 16 Boyutlu Involutif MDBL A_{1B} Matrisine Ait Ham XOR Sonucunun Test Ortamında Gösterimi

Şekil 5.5'te görülen test ortamında yer alan "Dosya Seç" seçeneği ile ham XOR sayısı hesaplatılmak istenen matrislere ait dosya yüklenebildiği gibi, tek bir matrise ait ham XOR sayısı elde edilmek isteniyorsa yapılması gereken test ortamında yer alan ekrana ham XOR sayısı hesaplatılmak istenen matrisi doğrudan girmektir. Şekil 5.2'de boş halleri görülen "Kayıt dosyası", "Matris ismi" ve "Matris sayısı" gibi alanlar doldurulur ak "Hesapla" butonu ile söz konusu matrise ait ham XOR sayısı elde edilir. Şekil 5.6'da dosya yükleme seçeneği ile test ortamına yüklenen matrislerin ham XOR sonuçlarının yer aldığı ekran görülmektedir. Şekil 5.5. 'te ise tek bir matrise ait ham XOR sayısı hesaplatılmış ve dal sayısı 5 olan 16×16 boyutlu involutif MDBL A_{1B} Matrisine ait ham XOR sonucu gösterilmiştir.



Şekil 5. 6. Ham XOR Sayıları Hesaplanan Matrisler ve Ham XOR Değerleri

Ham XOR sayısı elde edilen matrisler Paar algoritmasına Şekil 5.7'de yer alan ekrandaki gibi hazırlanır.



Şekil 5. 7. 16×16 Boyutlu Involutif MDBL A_{1B} Matrisine Ait Paar Optimizasyonu Sonucunun Test Ortamında Gösterimi

Test ortamında yer alan "Dosya Seç" seçeneği ile Paar Optimizasyonu uygulanmak istenen matrislere ait dosya yüklenebildiği gibi, tek bir matrise ait Paar optimizasyonu sonucu elde edilmek isteniyorsa yapılması gereken test ortamında yer alan ekrana Paar optimizasyonu sonucu hesaplatılmak istenen matrisi doğrudan girmektir. Şekil 5.2'de boş halleri görülen "Kayıt dosyası", "Matris ismi" ve "Matris sayısı" gibi alanlar doldurularak "Çalıştır" butonu ile söz konusu matrise ait Paar optimizasyonu sonrası XOR sayısı elde edilir. Şekil 5.8'de dosya yükleme seçeneği ile test ortamına yüklenen matrislerin Paar optimizasyonu sonuçlarının yer aldığı ekran görülmektedir. Şekil 5.7. 'de ise tek bir matrise ait Paar optimizasyonu sonrası XOR sayısı hesaplatılmış ve dal sayısı 5 olan 16×16 boyutlu involutif MDBL A_{1B} Matrisine ait sonuç 58 XOR olarak elde edilmiştir.

```
M_16x16_1
-----------------
PAAR Sonrası XOR Sayısı: 58

M_16x16_2
-----------------
PAAR Sonrası XOR Sayısı: 62

M_16x16_3
-----------------
PAAR Sonrası XOR Sayısı: 58

M_16x16_4
-----------------
PAAR Sonrası XOR Sayısı: 56

M_16x16_5
-----------------
PAAR Sonrası XOR Sayısı: 60

M_16x16_6
-----------------
PAAR Sonrası XOR Sayısı: 62

M_16x16_7
-----------------
PAAR Sonrası XOR Sayısı: 54

M_16x16_8
-----------------
PAAR Sonrası XOR Sayısı: 62

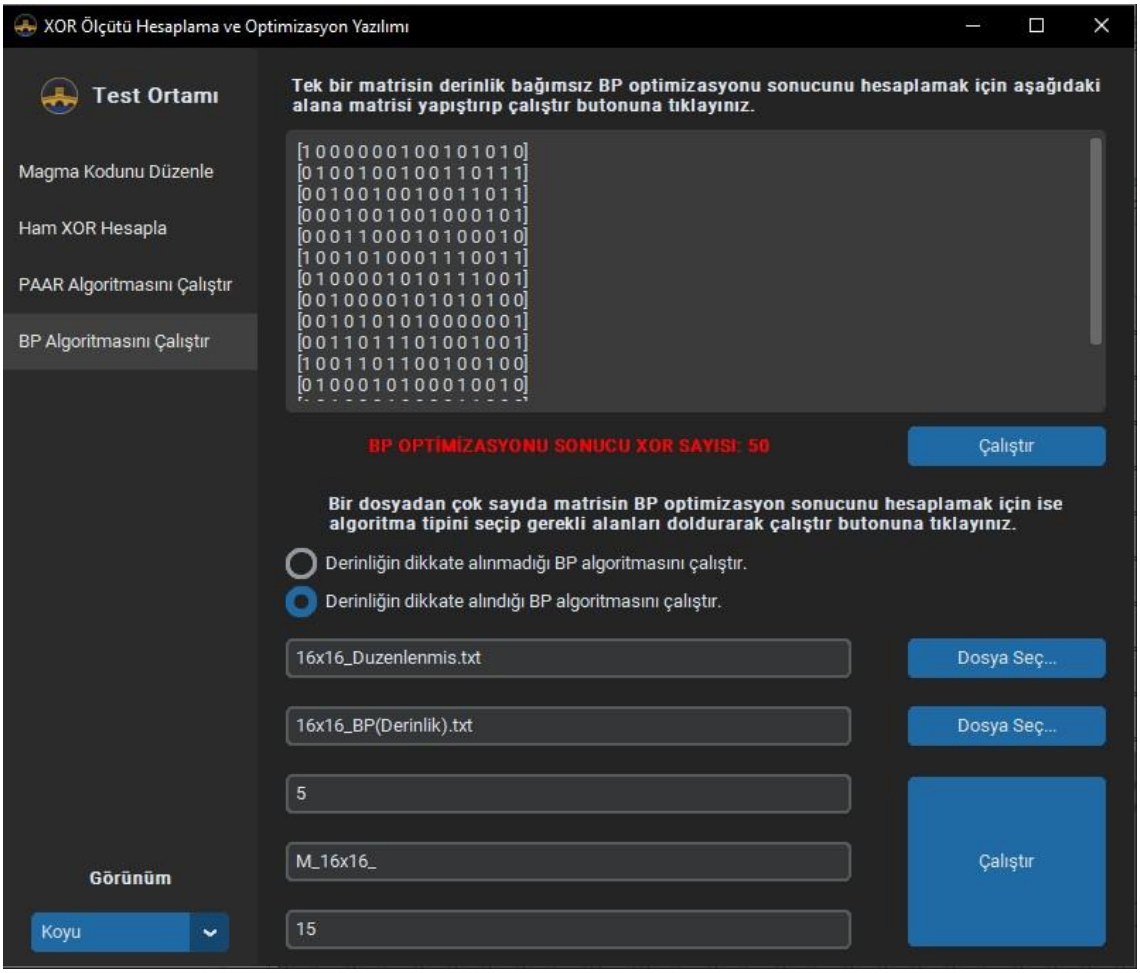
M_16x16_9
-----------------
PAAR Sonrası XOR Sayısı: 60

M_16x16_10
-----------------
PAAR Sonrası XOR Sayısı: 64
```

Şekil 5. 8. Ham XOR Sayıları Hesaplanan Matrisler ve Paar Optimizasyonu Sonrası XOR Sayıları

Paar optimizasyonun tamamlanmasının ardından MAGMA yardımı ile elde edilmiş olan Paar optimizasyonu öncesi matrislere daha sonra test ortamında BP

optimizasyonu uygulanarak derinlik bağımlı ve derinlik bağımsız XOR sayılarına bakılmıştır. Şekil 5.9'da test ortamında BP optimizasyonuna ilişkin ekran yer almaktadır.



Şekil 5. 9. 16×16 Boyutlu Involutif MDBL A_{1B} Matrisine Ait BP Optimizasyonu Sonucunun Test Ortamında Gösterimi

Test ortamında yer alan "Dosya Seç" seçeneği ile BP Optimizasyonu uygulanmak istenen matrislere ait dosya yüklenebildiği gibi, tek bir matrise ait BP optimizasyonu sonucu elde edilmek isteniyorsa yapılması gereken test ortamında yer alan ekrana BP optimizasyonu sonucu hesaplatılmak istenen matrisi doğrudan girmektir. Şekil 5.2'de boş halleri görülen "Kayıt dosyası", "Matris ismi" ve "Matris sayısı" gibi alanlar doldurur. Bu aşamadan sonra dikkat edilmesi gereken BP Optimizasyonu sonrası XOR sayısı hesaplatılırken derinliğin dikkate alınıp alınmayacağına karar verilmesi, derinliğin dikkate alınacağı durumlarda "Derinliğin dikkate alındığı BP algoritmasını çalıştır" seçeneği seçilmesidir. Derinliğin dikkate alındığı durumlar için "Derinlik Değeri" alanı

doldurulmalıdır. Daha sonra "Çalıştır" butonu ile söz konusu matrise ait BP optimizasyonu sonrası XOR sayısı hesaplatılır. Şekil 5.10'de derinlik değeri 5 olmak üzere dal sayısı 5 olan 16×16 boyutlu involutif MDBL A_{1B} Matrisine ait BP Optimizasyonu detayları gösterilmiştir.

Şekil 5.9. 'da ise tek bir matrise ait BP optimizasyonu sonrası XOR sayısı hesaplatılmış ve derinlik dikkate alınarak derinliğin 5 olduğu durumlar için dal sayısı 5 olan 16×16 boyutlu involutif MDBL A_{1B} Matrisine ait sonuç 50 XOR olarak elde edilmiştir.

```
M_16x16_1

50

t1 = x2 + x11  (1)
t2 = x3 + x10  (1)
t3 = x5 + x15  (1)
t4 = x7 + x13  (1)
t5 = x6 + x15  (1)
t6 = x7 + x14  (1)
t7 = x1 + x11  (1)
t8 = x3 + x9  (1)
t9 = x0 + x12  (1)
t10 = x4 + x8  (1)
t11 = x2 + t5  (2)
t12 = t10 + t11 *  y8  (3)
t13 = x4 + t4  (2)
t14 = x14 + t2  (2)
t15 = t10 + t14 *  y4  (3)
t16 = x3 + x15  (1)
t17 = t7 + t13  (3)
t18 = t16 + t17  (4)
t19 = t14 + t18 *  y1  (5)
t20 = t12 + t18 *  y13  (5)
t21 = x8 + x12  (1)
t22 = x1 + t2  (2)
t23 = t3 + t22 *  y15  (3)
t24 = x5 + t6  (2)
t25 = t7 + t24 *  y11  (3)
t26 = x6 + t1  (2)
t27 = t9 + t26 *  y12  (3)
t28 = x9 + t1  (2)
t29 = t4 + t28 *  y7  (3)
t30 = x0 + x2  (1)
t31 = x10 + t6  (2)
t32 = t9 + t31 *  y0  (3)
t33 = x13 + t5  (2)
t34 = t8 + t33 *  y3  (3)
t35 = x0 + x6  (1)
t36 = t2 + t13  (3)
t37 = t35 + t36 *  y10  (4)
t38 = x4 + t6  (2)
t39 = t8 + t30  (2)
t40 = t38 + t39 *  y14  (3)
t41 = t1 + t3  (2)
t42 = x14 + t41  (3)
t43 = t21 + t42 *  y2  (4)
t44 = t39 + t42  (4)
t45 = x10 + t44 *  y5  (5)
t46 = t6 + t27  (4)
t47 = t44 + t46 *  y9  (5)
t48 = x6 + t21  (2)
t49 = t36 + t48  (4)
t50 = t18 + t49 *  y6  (5)
Depth is 5
```

Şekil 5. 10. 16×16 Boyutlu Involutif MDBL A_{1B} Matrisine Ait BP Optimizasyonu Detayları

A_{1B} matrisinin dal sayısı 5, derinlik bağımsız BP optimizasyon algoritması sonrası elde edilen XOR sayısı ise 37'dir.

Örnek 5.4. BHwC yöntemi ile hazırlanan MAGMA kodu ile yapılan arama sonucu 16×16 boyutlu dal sayısı 8 ve involutif MDBL matrisler elde edilmiştir. Bulunan matrislerden aşağıda yer alan *M* matrisine ait ham XOR sayısı 96 olup derinlik bağımsız çalışılan BP algoritması ile elde edilen XOR sayısı ise 49'dur. Çizelge 5.1'de *M* matrisine ait derinlik bağımsız ve derinlik bağımlı çalışılan BP algoritmasına ait sonuçlar ve literatürde yer alan da sayısı 8, 16×16 boyutlu involutif matrisler ile optimizasyon sonrası XOR sayılarına ait bilgiler paylaşılmıştır.

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

Çizelge 5. 1. Literatürde Yer Alan Sonuçlarla, Dal Sayısı 8, 16 × 16 Boyutlu MDBL *M* Matrisinin Farklı Derinlik Değerlerine Ait XOR Sayılarının Karşılaştırılması

Matris Türü	Ham XOR	BP Optimizasyonundan Sonra XOR Değerleri			Referans
		Derinlik 3	Derinlik 4	Derinlik 5	
Involütif MDBL	96	56	49	49	(Kwon vd., 2003)
Involütif MDBL	96	54	49	49	(Akleylek vd., 2017)
Involütif MDBL	96	55	49	49	(Tuncay, 2021)
Involütif MDBL	**96**	**54**	**49**	**49**	**Örnek 5.4**

Dal sayısı 8, involutif MDBL 16 × 16 boyutlu matrislere ait BP optimizasyonu derinlik bağımlı sonuçları Çizelge 5.1'e göre değerlendirildiğinde *M* matrisinin literatürde yer alan matrislerle aynı XOR maliyetine sahip olduğu görülür.

Ayrıca Çizelge 5.2. ve Çizelge 5.3.'de Hadamard ve GHadamard matris türleri ile üretilmiş olan 16 × 16 boyutlu MDBL Matrislere ait BP Optimizasyonu sonrası bazı XOR sayıları verilmiştir.

Çizelge 5. 2. 16 × 16 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası Bazı XOR Değerleri (Hadamard)

Matris Türü (Hadamard)	Toplam Matris Sayısı	BP Sonrası XOR Değerleri			
		37 XOR	47 XOR	49 XOR	50 XOR
Involütif MDBL	1512	24	24	168	124

Çizelge 5. 3. 16×16 Boyutlu MDBL Matrislere Ait Bazı XOR Değerleri (GHadamard)

Matris Türü (GHadamard)	Toplam Matris Sayısı	BP Sonrası XOR Değerleri			
		50 XOR	51 XOR	52 XOR	53 XOR
Involütif MDBL	504	24	28	8	16

20×20 ve 24×24 boyutlu involutif MDBL matrisler üzerinde çalışma yapılmak amacıyla MAGMA kodu yazılmış ve doğrudan arama tekniği ile elde edilen 20×20 ve 24×24 boyutlu involutif MDBL matrisler test ortamı kullanılarak derinlik bağımlı ve derinlik bağımsız olmak üzere her iki durum için de BP optimizasyon algoritmaları sonrası XOR sayıları hesaplatılmıştır.

Örnek 5.4. $\mathbb{F}_2$ cisminin elemanları olan 20×20 boyutlu ikili M matrisinin – dairesel matris- BP algoritması öncesi ve sonrası XOR sayıları aşağıdaki gibidir.

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

Ham XOR sayısı 160 olan M matrisinin derinlik bağımsız BP algoritması sonrası XOR sayısı 67 olarak elde edilmiştir.

Çizelge 5.4 ve Çizelge 5.5'te MAGMA ara yüzü kullanılarak Dairesel ve Toeplitz matris türleri ile elde edilen 20×20 boyutlu MDBL matrislere ait derinlik bağımlı ve derinlik bağımsız BP Optimizasyon sonrası XOR sayıları yer almaktadır.

Çizelge 5. 4. 20×20 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası XOR Değerleri (Dairesel)

Matris Türü (Dairesel)	**BP Sonrası XOR Değerleri**		
	67 XOR	**68 XOR**	**69 XOR**
Involütif MDBL	20	20	80

Çizelge 5. 5. 20×20 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası XOR Değerleri (Toeplitz)

Matris Türü (Toeplitz)	**BP Sonrası XOR Değerleri**		
	74 XOR	**76 XOR**	**77 XOR**
Involütif MDBL	1	1	4

Çizelge 5.6'da yer alan 74 XOR sayısına sahip matrise ait ham XOR sayısı 147 olarak hesaplanmıştır.

Örnek 5.5. $\mathbb{F}_2$ cisminde tanımlı 24×24 boyutlu ikili T matrisinin BP optimizasyonu öncesi Ham XOR sayısı ve derinlik bağımsız BP Algoritması sonrası XOR sayıları aşağıdaki gibidir.

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

Ham XOR sayısı 168 olan T matrisinin BP global optimizasyonu sonrası XOR sayısı 96 olarak elde edilmiştir.

Çizelge 5.6'da MAGMA ara yüzü kullanılarak dairesel ve toeplitz matris türleri ile elde edilen 24×24 boyutlu MDBL matrislere ait veriler yer almaktadır.

Çizelge 5. 6. 24×24 Boyutlu MDBL Matrislere Ait Derinlik Bağımlı ve Derinlik Bağımsız BP Algoritması Sonrası XOR Değerleri

	BP Sonrası XOR Değerleri		
Matris Türü (24 x 24)	**96 XOR**	**97 XOR**	**98 XOR**
Involütif MDBL	3	6	7

BÖLÜM 6

6. DEĞERLENDİRME VE SONUÇ

Bu tez çalışmasında kriptografik yapıların difüzyon tabakalarının önemli bileşeni olan (veya bu tabakalarda doğrudan kullanılan) MDS matrislerin ve MDBL matrisler için öne sürülen (son zamanlarda popüler olan) optimizasyon tekniklerinden PAAR algoritması ve BP algoritması incelenmiştir. Bu kapsamda bu iki algoritmanın yer aldığı çeşitli boyutlarda ve çeşitli sayıda MDS ve MDBL matrislerin optimize edilebildiği bir test ortamı (literatürde bulunan kodlardan da faydalanılarak) geliştirilmiştir. Bu test ortamı tarafından (özellikle BP algoritmasına yönelik) hem optimize edilen kriptografik yapının XOR sayısı hem de hangi derinlikte optimizasyonun gerçekleştirildiği gibi önemli bilgiler verilmektedir. Tez çalışmasında literatürde bulunan MDS ve MDBL matrislerin elde edilmesine yönelik çeşitli yöntemler de incelenerek, bu yöntemler tarafından üretilen MDS ve MDBL matrislerin geliştirilen test ortamında optimizasyonları gerçekleştirilmiştir. Özellikle $16 \times 16, 20 \times 20, 24 \times 24$ gibi farklı boyutlarda involutif ve yüksek dal sayısına sahip matrisler üretilmiş olup üretilen bu matrislere global optimizasyon tekniklerinden BP optimizasyonu uygulanmıştır. Literatürde 20×20 ve 24×24 boyutunda ve yüksek dal sayısına sahip ($n \times n$ boyutunda $n > 16$) ikili matrisler için tasarım yöntemleri kısıtlı olmasından dolayı tezde ikili matrisler ($n > 16$) için elde edilen deneysel sonuçlara literatürde daha önce rastlanmamıştır. Ayrıca bu çalışmada geliştirilen test ortamı ile 32×32 ve 64×64 boyutunda ve yüksek dal sayısına sahip ikili matrisler oluşturularak optimize edilebilir. Tez çalışmasında elde edilen diğer bir sonuç ise BP algoritmasının optimizasyonda çok başarılı bir algoritma olduğudur. Örneğin (Guo vd., 2016) çalışmasında 16×16 boyutunda MDBL matrisler için optimal XOR sayısı 48 olarak verilmiştir. Literatürde (Akleylek vd., 2017) $(BHwC)_{2,4}$ ve $(BCwC)_{2,4}$ matris tipleri aynı özellikler ile elde edilen matrislerin BP optimizasyon sonuçları tez çalışmasında 49 XOR olarak

bulunmuştur. Buna ek olarak (Guo vd., 2016) çalışmasında optimal olarak verilen matrislerin şifrenin tasarımında nasıl kullanılacağı verilmemiştir. Sonuç olarak bu iki sonuç BP algoritmasının optimizasyon algoritmasının başarım açısından iyi bir algoritma olduğunu göstermektedir. Bununla beraber bu algoritmanın gelecekte hem başarım hem de çalışma süresi olarak (özellikle büyük boyutlu matrisler için) geliştirilebilir olduğunu da göstermektedir.

KAYNAKLAR

Akkanat, K. (2018). *MDS Yayılım Matrislerinde İzomorfizmalar Üzerine Yeni Bir Çalışma.* Edirne: (Yüksek Lisans Tezi). Trakya University.

Akleylek, S., Karakaya, A., Canbay, Y., Bülbül, H. İ., Tuğlular, T., Irmak, E., . . . Takçı, H. (2020). *Siber Güvenlik ve Savunma.*

Akleylek, S., Rijmen, V., Sakallı, M. T., & Öztürk, E. (2017). Efficient Methods to Generate Cryptographically Significant Binary Diffusiom 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. *Wiley Online Library, 9*, 3558-3569.

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

Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., & Tokita, T. (2000). Camilla: A 128-Bit Block Cipher Suitable for Multiple Platforms Design and Analysis. *International workshop on selected areas in cryptography*, (s. 39-56).

Aslan, B. (2013). *Blok Şifreler İçin Cebirsel İkili Doğrusal Dönüşüm Tasarımı ve Modern Bir Blok Şifreye Uygulanması.* Edirne: (Doktora Tezi). Trakya University.

Avanzi, R. (2017). *A Salad of Block Ciphers.*

Banik, S., Funabiki, Y., & Isobe, T. (2019). International Workshop on Security. *More Results on Shortests Linear Programs*, (s. 109-128).

Barreto, P. S., & Rijmen, V. (2000). The Khazad Legacy-Level Block Cipher. *Primitive submitted to NESSIE, 97*, 106.

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

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

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

Feistel, H. (1973). Cryptography and Computer Privacy. *Scientific American, 228*(5), 15-23.

Feistel, H., & Notz, W. A. (1975, November). Some Cryptographic Techniques for Machine-to-Machine Data Communications. *Proceedings of the IEE, 63*(11), 1545-1554.

Ferguson, N., Schneier, B., & Kohno, T. (2011). *Cryptography Engineering Design Principles and Practical Applications.* Indanapolis: Wiley Publishing.

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*, (s. 71-84).

Furth, B., Muharemagic, E., & Socek, D. (2005). An Overview of Modern Cryptography. *Multimedia Encryption and Watermarking* (s. 31-51). içinde

Guo, Z., Wu, W., & Gao, S. (2016). Constructing Lightweight Optimal Diffusion Primitives with Feistel Structure. *TCA Laboratory, SKLCS, Institute of Software*, s. 352–372.

Gupta, K. C., & Ray, I. G. (2013). On Constructions of Involutory MDS Matrices In AFRICACRYPT. *Springer*, 43-60.

Internet World Stats. (2022, July 31). World Internet Usage And Population Statistic 2022 Year Estimates: https://www.internetworldstats.com/stats.htm adresinden alındı

Jean, J., Peyrin, T., Sim, S. M., & Tourteaux, J. (2017). Optimizing Implementations of Flight Weight Building Blocks. *IACR Transactions on Symmetric Cryptology*, (s. 130-168).

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

Keliher, L. (2002). Linear Cryptanalysis of Substitution-Permutation Networks. (Ph.D. Thesis). Queen's University.

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

Khoo, K., Peyrin, T., Poschmann, A. Y., & Yap, H. (2014). Searching For Hardware Optimal SPN Structure and Components With a Fair Comparison. *International Workshop on Cryptographic Hardware and Embedded Systems*, 433-450.

Knudson, L. (2000, October). Block Chaining Modes of Operation. *NIST First Modes of Operation Workshop*.

Kodaz, H., & Botsalı, F. M. (2010). Simetrik ve Asimetrik Şifreleme Algoritmalarının Karşılaştırılması. *Selçuk-Teknik Dergisi, 9*(1).

Koo, B., 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*, s. 51-64.

Kölsch, L. (2019). XOR-Counts and Lightweight Multiplication with Fixed Elements in Binary Finite Fields. *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, (s. 285-312).

Kwon, D., Kim, J., Park, S., Sung, S. H., Sohn, Y., Song, J. H., . . . Lee, J. (2003). New Blockchiper: ARIA. *International Conference on Information Security and Cryptology*, (s. 432-445).

Kwon, D., Sung, S. H., Song, J. H., & Park, S. (2005). Design of Block Chipers and Coding Theory. *Trends in Mathematics Information Center for Mathematical Sciences, 8*(1), 13-20.

Lidl, R., & Niederreiter, H. (1994). Introduction to Finite Fields and Their Applications. *Cambridge University Press*.

Menezes, A. J., Rosen, N. H., Oorschot, P. C., & Vanstone, S. A. (1997). *Handbook of Applied Cryptography*. Boca Raton.

Nakahara, J. J., & Abrahao, E. (2009). A New Involutory MDS Matrix for the AES. *9*(2), 109-116.

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

Pehlivanoğlu, M. K. (2018). *Maksimum Uzaklıkta Ayrılabilen Matrislerin Elde Edilebilmesi İçin Yeni Bir Matris Formu ve Bir Hafif Sıklet Blok Şifreleme Uygulaması*. (Doktora Tezi). Trakya University.

Pehlivanoğlu, M. K., & Kavun, E. B. (2020). Kriptografik Blok Şifrelerin Maksimum Uzaklıkla Ayrılabilen Yayılım Tabakalarının Tasarımı. *Siber Güvenlik ve Savunma* (s. 295-325). içinde

Pehlivanoğlu, M. K., Sakallı, F. B., Akleylek, S., & Sakallı, M. T. (2021). Binary Finite field Extensions For Diffusion Matrices Over the Finite Field F2m. *2021 29th Signal Processing and Communications Applications Conference (SIU)*. IEEE.

Pehlivanoğlu, M., Sakallı, M., Akleylek, S., Duru, N., & V. Rijmen. (2018, Ocak). Generalization of Hadamard Matrix to Generate Involutory MDS Matrices for Lightweight Cryptography. *IET Information Security, 12*(4).

Roman, S. (1992). Coding Information Theory, Graduate Texts in Mathematics. *Springer Verlag, 134*, 488.

Sakallı, F. B., Aydın, Ö., Tuncay, G., Pehlivanoğlu, M. K., Güzel, G. G., & Sakallı, M. T. (2020). On Lightweight 4 × 4 MDS Matrices over Binary Field Extensions. *International Journal of Information Security Science, 9*(2), s. 94-103.

Sakallı, M. T. (2006). *Modern Şifreleme Yöntemlerinin Gücünün İncelenmesi.* (Doktora Tezi). Trakya University.

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*, s. 485-494.

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., Buluş, E., & Büyüksaraçoğlu, F. (2004, Aralık). İkili Döngülü Bir Blok Şifreleme Algoritmasının Lineer Kriptanaliz Uygulaması. Bursa, Türkiye.

Schnorr, C. P., & Vaudenay, S. (1994). Black Box Cryptanalysis of Hash Networks Based on Multipermutations. *Workshop on the Theory and Application of Cryptographic Techniques*, 47-57.

Shannon, C. E. (1949). *The Bell System Technical Journal, 28*(4), 656-715.

Stallings, W. (2010). NIST Block Cipher Modes of Operation for Confidentiality. *Cryptologia*, 163-175. doi:https://doi.org/10.1080/01611190903185401

Stallings, W. (2017). Cryptography and Network Security. *Pearson Education, 7/E*.

Stoffelen, K. (2016). Optimizing S-Box Implementations for Several Criteria Using SAT Solvers. *International Conference on Fast Software Encryption*, (s. 140-160).

System, M. C. (2021). Kasım 12, 2022 tarihinde http://magma.maths.usyd.edu.au/magma/ adresinden alındı

Şahin, F. (2015). Modern Blok Şifreleme Algoritmaları. *İstanbul Aydın Üniversitesi Dergisi*(26), 23-40.

Tuncay, G. (2021). *Blok Şifrelerde Kullanılan İkili ve MDS Matrisler İçin Yeni Tasarım Teknikleri.* Edirne: (Doktora Tezi). Trakya University.

Uslu, A. (2008). *Dairesel Matrisler ve Uygulaması.* (Yüksek Lisans Tezi). Sakarya University.

Webster, A. F., & Tavares, S. E. (1985). On the Design of S- Boxes. *Conference on the theory and application of cryptographic techniques*, (s. 523-534).

Yavuzer-Aslan, F., Sakallı, M. T., & Aslan, B. (2012). Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi. *Akademik Bilişim'12 - XIV. Akademik Bilişim Konferansı.* Uşak.

ÖZGEÇMİŞ

Nihal TAN KAÇAR, 1988 yılı İzmir doğumlu. İlköğretimini ve ortaöğretimini İzmir'in Dikili ilçesinde birincilikle tamamladı. 2008 yılında Trakya Üniversitesi Bilgisayar Mühendisliği Lisans Bölümüne başladı. Mezuniyeti sonrası 2013 yılında İstanbul'da özel bir multimedia firmasında 7 ay çalıştıktan sonra 2016 yılında KPSS merkezi atama ile Ankara Çalışma ve Sosyal Güvenlik Bakanlığı'na atandı. 2019 yılında nakil ile Trakya Üniversitesi Bilgi İşlem Daire Başkanlığına geçiş yaptı. 2018 yılında giriş yaptığı Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans programına devam etmektedir.