T.C
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BLOK ŞİFRELERDE KULLANILAN DOĞRUSAL DÖNÜŞÜM YAPILARININ İNCELENMESİ

FÜSUN YAVUZER ASLAN

YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
DANIŞMAN: YRD. DOÇ. DR. M. TOLGA SAKALLI
EDİRNE-2012

T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

BLOK ŞİFRELERDE KULLANILAN DOĞRUSAL DÖNÜŞÜM YAPILARININ İNCELENMESİ

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

Füsun YAVUZER ASLAN

Bu tez 01/08/2012 tarihinde Aşağıdaki Jüri Tarafından Kabul Edilmiştir.

(İmza)	(İmza)	(İmza)
.....	.....	.....
Yrd. Doç. Dr. M. Tolga SAKALLI	Doç. Dr. Yılmaz ÇAN	Yrd. Doç. Dr. Andaç ŞAHİN MESUT

Yüksek Lisans Tezi
Blok Şifrelerde Kullanılan Doğrusal Dönüşüm Yapılarının İncelenmesi
Trakya Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Simetrik şifreleme sınıfına giren blok şifrelerin içyapısında üç önemli eleman bulunmaktadır. Bunlar sırasıyla yer değiştirme kutuları (S-kutuları), doğrusal dönüşüm yapıları ve anahtar planlama safhasıdır. Bu tez, bu yapılardan doğrusal dönüşümlerin kriptografik özelliklerinin incelenmesi, bilgisayarda uygulaması etkin ve kriptografik özellikleri iyi doğrusal dönüşümlerin tasarımı üzerinedir.

Literatürde blok şifreler Feistel ve SPN (Substitution-Permutation Networks) mimarisi tabanlıdır. Günümüzde blok şifrelerde kullanılan doğrusal dönüşümler, blok şifrelerin performansını belirleyen en önemli unsurlardır. Dolayısıyla SPN tabanlı blok şifreleme algoritmalarında şifreleme ve deşifreleme aşamalarında involutif (tersi kendisi) yapıların kullanılması bu iki işlem arasında oluşacak hız farkını da önleyecektir. Buna ek olarak şifreleme ve deşifreleme hızı birbirine yakın doğrusal dönüşüm yapılarını kullanmak bu hız farkını kullanan saldırıları da önleyecektir.

Tez şu şekilde organize edilmiştir. Tezin 1. bölümünde kriptografi bilimine kısa bir giriş yapılmıştır. Önemli blok şifreler ve kullandıkları doğrusal dönüşümler hakkında bilgi verildikten sonra günümüzde halen güvenli bir şifreleme algoritması olan AES (Advanced Encryption Standard) tanıtılmıştır.

2. bölümünde, tez sırasında kullanılan sonlu cisimler teorisi ile ilgili matematiksel alt yapı verilmiştir.

3. bölümde, literatürde bulunan şifreleme algoritmalarında kullanılan doğrusal dönüşüm yapıları, belirlenen kriptografik özelliklere göre ayrıntılı olarak incelenmiştir.

Tezin 4. bölümünde, 4×4 ve 8×8 boyutunda elemanları GF(2^8) cismine ait olan tersi kendisi (involutif) MDS (Maximum Distance Seperable) matris tasarımı gerçekleştirilmiştir.

5. bölümde, elemanları GF(2^m) cismine ait n×n boyutunda MDS matrisleri arayan ve bu doğrusal dönüşümlerin sabit nokta sayısını elde etmek için geliştirilen yazılım tanıtılmaktadır.

6. bölümde ise tez çalışmasında elde edilen sonuçlar sıralanmaktadır.

Yıl **:** 2012

Sayfa **:** 86

Anahtar Kelimeler **:** Doğrusal dönüşümler, MDS matrisler, Dallanma sayısı, Sabit noktalar, Blok şifreler, Kriptografi

Msc. Thesis
An Examination of Linear Transformation Structures used in Block Ciphers
Trakya University Graduate School of
Natural and Applied Sciences
Department of Computer Engineering

ABSTRACT

Block ciphers, which are symmetric ciphers, are composed of three important components: S-box (Substitution-box), linear transformation and key schedule. This thesis is related with cryptographic properties of linear transformations and linear transformations having good cryptographic and implementation properties.

In the literature, Feistel networks and SPNs (Substitution-Permutation Networks) are the two main structures in designing block ciphers. On the other hand, today, the most important consideration in determining the performance of block cipher is linear transformations. Therefore, involutional linear transformations are of interest in the use of SPN based block cipher design since they provide equal cost of encryption and decryption operations. In addition, the use of linear transformations that are implemented with close cost of encryption and decryption operations will prevent the attacks, which use the performance difference of the encryption and decryption operations.

This thesis proceeds as follows. In Section 1, an introduction to cryptography is given. Also, some important block ciphers and linear transformations they use are introduced and AES (Advanced Encryption Standard), which is still secure block cipher, is discussed in detail. In Section 2, an introduction to finite fields, required mathematical background for the thesis, is given. In Section 3, the linear transformations of important block ciphers with respect to two important cryptographic

properties are examined in detail. In Section 4, the construction of 4×4 ve 8×8 involutional MDS (Maximum Distance Separable) matrices with the elements of GF(2^8) is given. In Section 5, the developed software, which can be used to search for n×n MDS matrices the elements of GF(2^m), is given. In Section 6, the thesis is concluded by giving obtained results.

Year **:** 2012

Number of Pages **:** 86

Keywords **:** Linear transformations, MDS matrices, Branch number, The number of fixed points, Block Ciphers, Cryptography

TEŞEKKÜR

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

Öncelikle tanıştığıma çok kere sevindiğim değerli hocam ve danışmanım Yrd. Doç. Dr. M. Tolga SAKALLI' ya ve sevgili eşi Yrd. Doç. Dr. Fatma BÜYÜKSARAÇOĞLU SAKALLI' ya bana olan desteklerinden ve dostluklarından dolayı çok teşekkür ederim.

Bu tezin izleme komitesinde yer alan Doç. Dr. Yılmaz Çan' a ve Yrd. Doç. Dr. Andaç ŞAHİN MESUT' a değerli katkılarından dolayı teşekkürlerimi sunarım.

Tez sırasında yaptığı paylaşımlarından dolayı Osman KARAAHMETOĞLU' na teşekkür ederim.

Yüksek Lisans sürecinde beni sürekli motive eden ve her türlü desteği veren canım annem İlknur YAVUZER' e ve canım babam Hasan YAVUZER' e sonsuz sevgi ve teşekkürlerimi sunarım.

Son olarak, hayatım boyunca attığım her adımda yanımda olan, desteği ve emeği ile beni hiçbir zaman yalnız bırakmayan sevgili eşim, yoldaşım, herşeyim Bora ASLAN' a sonsuz kere teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

AES : Advanced Encryption Standard
DES : Data Encryption Standard
ECC : Elliptic Curve Cryptography
FIPS : Federal Information Processing Standard
FPN : Sabit Nokta Sayısı (Fixed Point Number)
IDEA : International Data Encryption Algorithm
MD : Message-Digest Algorithm
MDBL : Maximum Distance Binary Linear
MDS : Maximum Distance Seperable
MOBIC : Maximum Order Bit Independence Criterion
MOSAC : Maximum Order Strict Avalanche Criterion
NIST : National Institute of Standards and Technology
NTT : Nippon Telegraph and Telephone Corporation
SHA : Secure Hash Algorithm
SPN : Substitution-Permutation Network
XOR : Exclusive Or
$GF(2)$: 2 elemanlı Galois Cismi
$GF(2^n)$: n elemanlı Galois Cismi
$\{0,1\}^n$: Elemanları 0 veya 1 Olan n-bit Vektör
$wt()$: Hamming Ağırlığı
$Had()$: Hadamard Matris
$circ()$: Dairesel Matris
$\beta()$: Dallanma Sayısı (Branch Number)
xtime : 02_h veya α ile Çarpma

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

BÖLÜM 1

GİRİŞ

Geçmişten günümüze, insanoğlu haberleşmede gizliliği her zaman ön planda tutmuştur. Bunun için yeni yöntemler geliştirmiş, iletilerin gizliliğini korumak için hep bir çaba ve arayış içerisinde olmuştur. Yani insanlık bilgi ve haberleşme güvenliğinin yüksek oranda sağlanmasının nasıl gerçekleşeceğine odaklanmıştır.

Şifreleme biliminin ilk yıllarında, bilgi iletimindeki gizlilik sadece kağıt ve kalemle yapılan tekniklere dayanırken, 1970'lerde bilgisayar ve matematiğin etkin bir şekilde kullanılmaya başlanması ile yeni bir boyut kazanmış, teknoloji ve internetin gelişimi ile daha kompleks algoritmaların ortaya çıkması zorunlu hale gelmiştir.

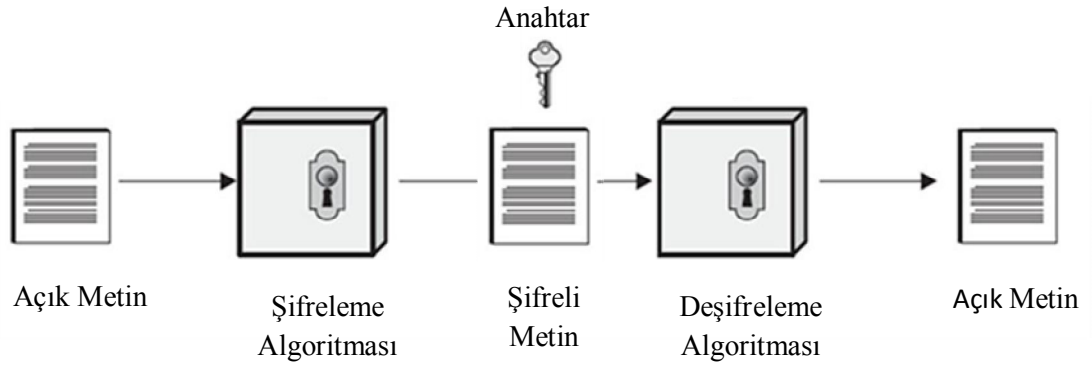
Tüm çaba, günümüz rekabet ortamında gitgide daha değerli bir konu halini alan bilgiyi en güvenli şekilde saklamak, korumak ve iletmek için yeni teknikler geliştirmektir.

1.1. Kriptoloji

Köken olarak Yunanca gizli anlamına gelen "kryptos (kript)" ve yazı anlamına gelen "graphein (graf)" kelimelerinden türetilen kriptografi, anlaşılır bir mesajı anlaşılmaz hale dönüştürme ve anlaşılmaz mesajı anlaşılır hale geri dönüştürme işlemlerini kapsayan bir bilimdir. Kriptografi, bilgi güvenliğini sağlamak için çalışan matematiksel yöntemler bütünüdür. Bu yöntemler bir bilginin iletimi sırasında karşılaşılabilecek saldırılardan bilgiyi, göndereni ve alıcıyı korumayı amaçlar. Yani kriptografi, verinin güvenli bir şekilde iletilmesi ile ilgilidir. Dolayısıyla güvenli bir şifreleme algoritması tasarımı kriptografide çok önemli bir yer tutar. Kriptografi ile ilgilenen bilim adamlarına kriptograf adı verilir. Kriptanaliz ise ele geçirilen şifreli metinleri bazı teknikler kullanılarak açık metinleri elde etme işlemidir. Kriptanaliz ile

ilgilenen kişilere kriptanalist denir. İyi bir kriptanalist aynı zamanda kriptografi alanında da bilgi sahibi olmalıdır çünkü bir şifrenin kırılabilmesi için şifre tasarımı hakkında yeterli bilgiye ihtiyaç vardır. Tarih boyunca kriptograflar ile kriptanalistler arasında çekişmeli bir yarış olduğunu söylemek yanlış olmayacaktır. Kriptanalistlerin başarıları kriptografları daha güçlü şifreler tasarlamaya zorlamıştır. Bu iki alanın birleşmesi ile kriptoloji bilimi ortaya çıkmıştır.

Bir kripto sistem; şifreleme algoritması, açık metin, şifreli metin ve anahtardan oluşmaktadır. Şifreleme algoritmaları kripto sistemin en önemli parçasıdır. Şekil 1.1'de bir kripto sistemin çalışma mantığı gösterilmiştir.



Şekil 1.1. Bir kripto sistemin çalışma mantığı

Temel olarak şifreleme algoritmaları simetrik, asimetrik ve hash algoritmaları olmak üzere üç gruba ayrılır. Bu şifreleme algoritmalarından simetrik algoritmalar, şifreleme ve deşifreleme işlemlerinde aynı anahtarı (gizli anahtarı) kullanır. Asimetrik şifreleme algoritmaları ise şifreleme için herkesin ulaşabileceği açık bir anahtar kullanılırken deşifreleme işlemi için gizli bir anahtar kullanır. Son olarak Hash algoritmaları, verinin özetini oluşturmak için kullanılırlar ve kimlik denetiminin sağlanmasında büyük rol oynarlar (Aslan, 2008). Buna göre bazı şifreleme algoritmaları Tablo 1.1'de gösterilmiştir.

Tablo 1.1. Şifreleme algoritmalarına örnekler (Aslan, 2008)

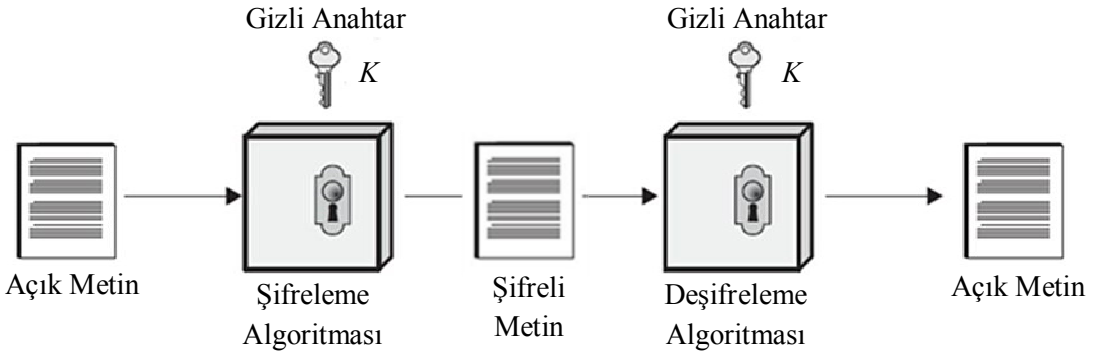
Simetrik Şifreleme Algoritmaları		Asimetrik Şifreleme Algoritmaları	Hash Algoritmaları
Blok Şifreler	Akan Şifreler		
-DES -IDEA -Square -AES -Camellia -ARIA -Khazad	-RC4 -Trivium - HC-256	- RSA - ElGamal - ECC	- MD4 - MD5 - SHA -RIPEMD-160

Kriptolojide kullanılan karmaşık şifreleme algoritmaları, sonlu cisimler teorisi, sayı teorisi ve kodlama teorisi gibi matematiğin önemli teorilerinin birleşmesi ile ortaya çıkmıştır.

1.2. Simetrik Şifreleme Algoritmaları

Simetrik şifreleme algoritmaları şifreleme ve deşifreleme için tek anahtar (gizli anahtar) kullanır. Açık metin gizli anahtar kullanılarak şifrelenir ve karşı tarafa iletilir. Şifreli mesaj tekrar aynı gizli anahtar kullanılarak deşifre edilir.

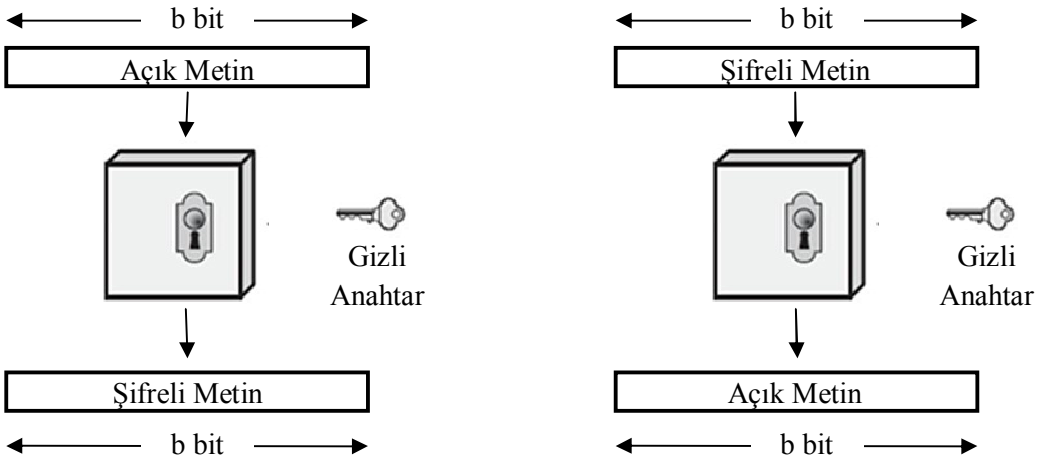
Simetrik şifreleme algoritmaları blok şifreler ve akan şifreler olmak üzere 2 kategoride incelenebilir. Bu tezin konusu olan doğrusal dönüşümler blok şifrelerin önemli elemanlarındandır. Simetrik şifreleme algoritmalarının çalışma biçimi Şekil 1.2'deki gibidir.



Şekil 1.2. Simetrik şifreleme algoritmasının çalışma mantığı

1.2.1. Blok Şifreler

Blok şifreleme algoritmaları açık metni sabit uzunluklu blok adı verilen bit grupları halinde işler. Bloklar bir anahtar aracılığı ile şifrelenerek şifreli metin ortaya çıkar. Deşifreleme işleminde yine aynı anahtar sayesinde şifreli metin açık metin haline getirilir. Blok şifreler için şifreleme ve deşifreleme aşamaları Şekil 1.3'teki gibidir.



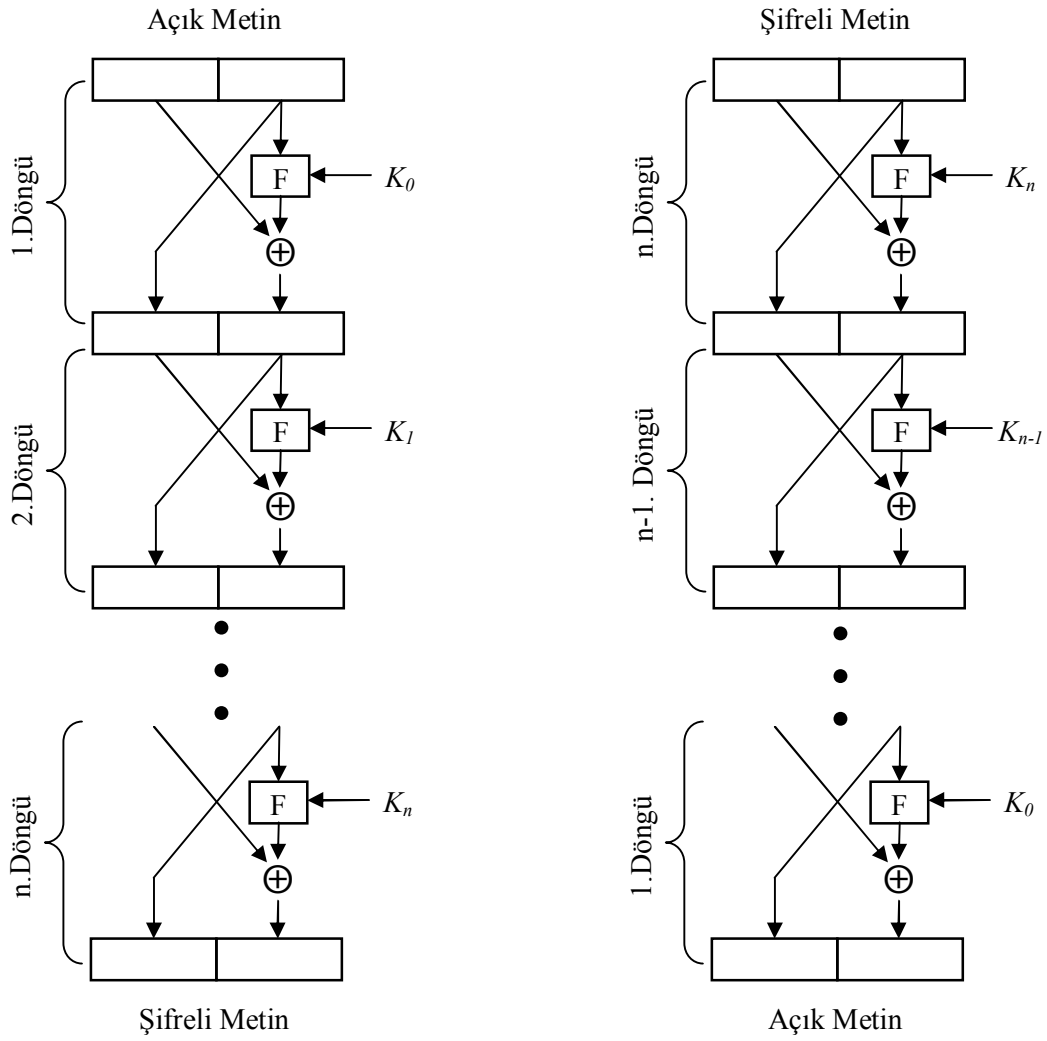
(a) Şifreleme (b) Deşifreleme

Şekil 1.3. Blok şifreleme ve deşifreleme

Blok şifreler, Shannon'un (Shannon, 1949) önerdiği karıştırma (confusion) ve yayılma (diffusion) teknikleri üzerine kuruludur. Karıştırma şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılma açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Bir blok şifrede karıştırma yer değiştirme işlemleri (S-kutuları) ile gerçekleştirilirken yayılma ise doğrusal dönüşüm işlemleri ile gerçekleştirilir.

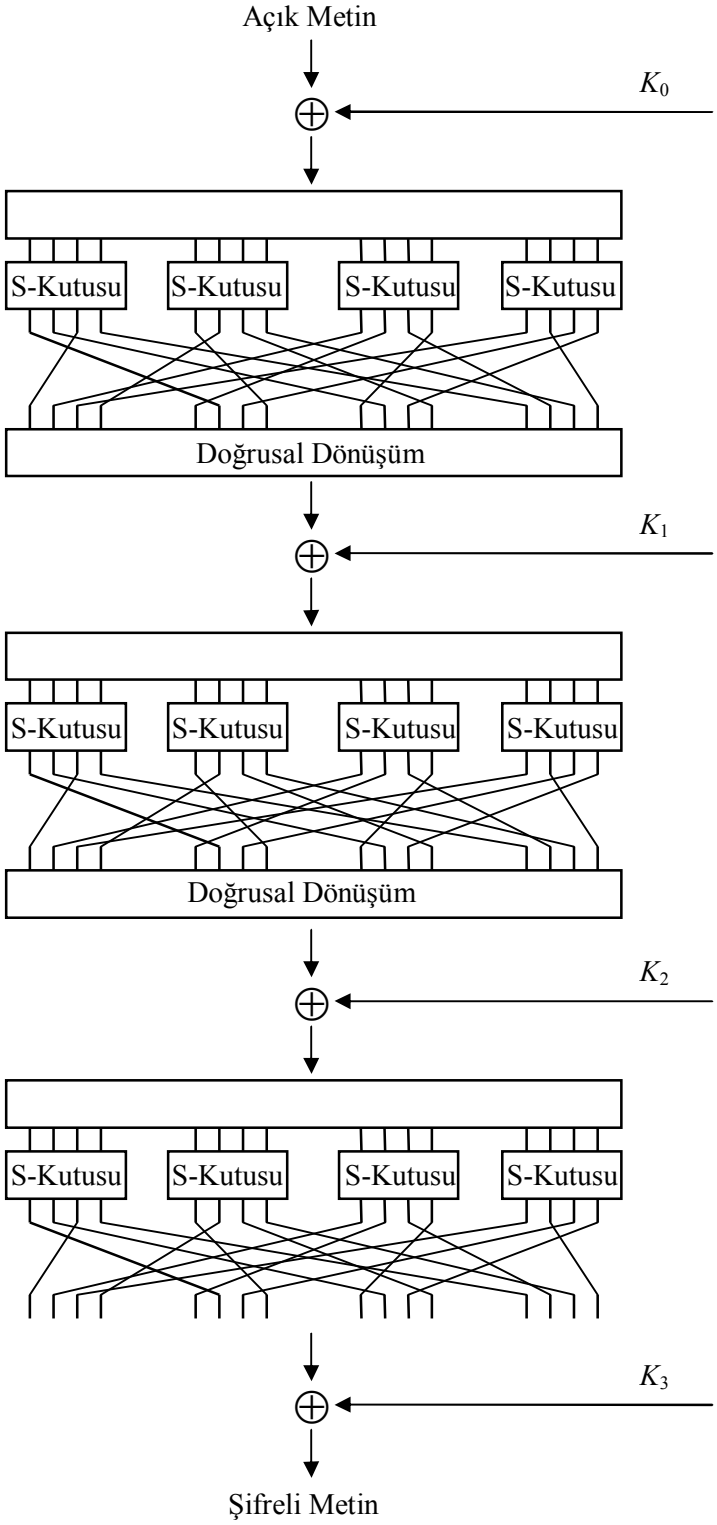
Blok şifrelerin tasarımında iki mimari türünden bahsedilebilir. DES (Data Encryption Standard) (FIPS 46-3, 1999) algoritmasının tasarımında Feistel mimarisi kullanılırken, AES (Advanced Encryption Standard) (FIPS 197, 2001) algoritmasının tasarımında ise SPN (Yer değiştirme-Permütasyon) mimarisi kullanılmaktadır (Keliher, 2003). Her iki mimari de yer değiştirme ve doğrusal dönüşüm yapılarını kullanır. Ayrıca her iki mimari ürün şifrelerinin örneklerindendir. Yani birden fazla şifreleme işleminin birleşmesi ile oluşturulurlar. Tekrarlanan şifreler yine ürün şifreleridir ve aynı

şifreleme adımının tekrarlanan uygulamasını içerir ve her şifreleme adımına döngü denir. Bir döngü birden fazla şifreleme adımı içerebilir. Genellikle her döngüde farklı anahtar materyali kullanılmaya özen gösterilir. Diğer yandan bu mimarilerin arasındaki en temel fark döngü içerisinde bir bloğun işlenmesinde ortaya çıkmaktadır. Örneğin Feistel mimarisinde (Şekil 1.4) bir döngüde o anki bloğun yarısı işlenirken SPN mimarisinde (Şekil 1.5) o anki bloğun tümü işlenir. Buna ek olarak bir blok şifrenin genel tasarımında bir döngü içindeki yer değiştirme S-kutuları ile yayılım ise doğrusal dönüşüm veya dönüşümler ile sağlanır. Her döngüde döngünün sonunda anahtar planlamadan gelen o döngü için elde edilen bir anahtar değeri ile XOR'lama işlemi gerçekleştirilir.



(a) Feistel Şifreleme Algoritması (b) Feistel Deşifreleme Algoritması

Şekil 1.4. Feistel ağı



Şekil 1.5. 16-bit giriş-çıkışlı 3 döngülük bir örnek SPN ağı

İyi bir şifrenin gücünü belirleyen çeşitli faktörler vardır. Blok şifrelerin gücünü belirleyen anahtar büyüklüğü, S-kutuları ve doğrusal dönüşümler hakkında detaylı bilgi aşağıdaki başlıklarda anlatılmaktadır.

1.2.1.1. Anahtar Büyüklüğü

Blok şifrelerde anahtarın uzunluğu saldırılara karşı güçlü olacak şekilde seçilmelidir. Bunun için anahtar uzunluğu önemlidir. Anahtar uzunluğu sayesinde şifrenin kaba kuvvet (bruteforce) saldırısına karşı kırılabilirliği zorlaşmaktadır. Bazı blok şifrelerin kullandıkları anahtar uzunlukları Tablo 1.2'de verilmiştir.

Tablo 1.2. Bazı blok şifrelerin anahtar uzunlukları

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

1.2.1.2. S-Kutuları

Blok şifreleme algoritmalarının en önemli elemanı S-kutularıdır ve karıştırma işlevini üstlenirler. Algoritmanın tek doğrusal olmayan elemanıdır. Bu yüzden iyi bir S-kutusu seçimi şifrenin karmaşıklığını yani gücünü doğrudan etkiler (Aslan, 2008).

S-kutuları şifre içerisinde bit bloklarının yer değiştirmesinde kullanılır. Bit blokları S-kutusundan geçirilerek farklı bit bloklarına haritalanır.

Bir S-kutusu vektörel boolean fonksiyonu $f_0, f_1, ..., f_{m-1}$ ile temsil edilebilir. f_i boolean fonksiyonları F_2^n 'den F_2 'ye tanımlanır ve S-kutusunun çıkış fonksiyonları olarak isimlendirilir. S-kutuları tasarlanırken aşağıdaki çeşitli yöntemler kullanılmaktadır:

- Pseudo-random üretim,
- Sonlu cisimde ters alma,

- Sonlu cisimde üs alma tekniği,
- Heuristic teknikler.

Bu yöntemlerden en çok kullanılanları sonlu cisimde ters alma ve üssel fonksiyon tekniğidir. Bu iki teknik ile doğrusal olmama ölçüsü yüksek ve diğer kriptografik özellikleri iyi S-kutuları elde edilebilir. Nitekim AES algoritmasında kullanılan S-kutusu sonlu cisimde ters alma yöntemi ile elde edilmiş bir S-kutusudur.

Bir S-kutusunun kriptografik özellikleri statik özellikler ve dinamik özellikler başlıkları altında işlenebilir. Statik özellikler açık metin, şifreli metin ve anahtar arasındaki ilişkiler ile ilgilidir. Örneğin doğrusal olmama özelliği bir statik özelliktir. S-kutusunun karakteristik yapısının saklandığı kriptografik özellikler dinamik olanlardır. S-kutuları için kriptografik özellikler aşağıdaki gibi sıralanabilir:

- Bütünlük (Completeness) kriteri (Kam ve Davida, 1979),
- Çığ (Avalanche) kriteri (Feistel, 1973),
- Katı çığ kriteri (Strict Avalanche Criterion) (Webster ve Tavares, 1986),
- Bit bağımsızlık kriteri (Bit Independence Criterion) (Webster ve Tavares, 1986),
- MOSAC ve MOBIC özellikleri (Aras ve Yücel, 2001) (Mister ve Adams, 1996),
- Doğrusal olmama kriteri (Meier ve Staffelbach, 1989),
- S-kutularının doğrusal yaklaşım tablosu (Matsui, 1994),
- S-kutularının XOR tablosu (Fark Dağılım Tablosu) (Biham ve Shamir, 1991),
- Boole fonksiyonlarının cebirsel ifadesindeki terimin en yüksek derecesi ve terimlerinin sayısı (Sakallı vd, 2010),
- Cebirsel S-kutuların polinomsal ifadesindeki cebirsel derece ve terim sayısı (Sakallı vd, 2010).

1.2.1.3. Doğrusal Dönüşümler

Blok şifrelerin önemli bir özelliği olan yayılma işlemini sağlayan yapılar bu tezin de konusu olan doğrusal dönüşümlerdir. Blok şifrelerde farklı doğrusal dönüşüm mekanizmaları kullanılmaktadır. Doğrusal dönüşümler sabit uzunluktaki bir giriş bloğunu doğrusal olarak karıştırarak aynı uzunlukta bir çıkış bloğu elde etmeyi sağlar. Örneğin AES şifresi 128-bit girişi 128-bit çıkışa haritalayan doğrusal bir dönüşüme sahiptir. Aşağıda verilen AES şifresinde kullanılan doğrusal dönüşüm, 4×4 byte matrisi

ile 4 byte verinin çarpımı sonucunda başka bir 4 byte veriye dönüşümü sağlar. Burada *h* alt indisi Hexadecimal notasyonu ifade etmektedir.

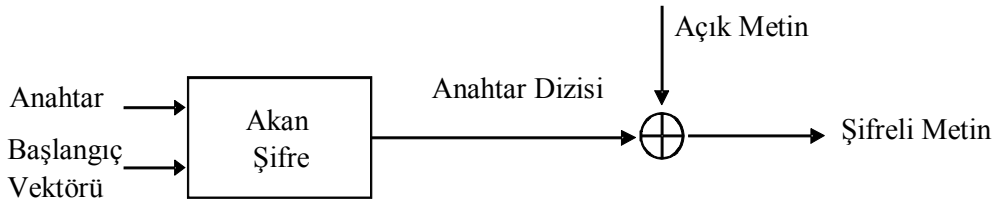
$$\begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix}$$

1.2.2. Akan Şifreler

Akan şifreleme algoritmaları, açık metnin bir karakterine bir seferde zamanla değişen bir şifreleme fonksiyonu kullanarak açık metnin karakterlerini ayrı ayrı şifreler.

Girdi olarak alınan bir anahtar ve başlangıç vektörü ile mümkün olduğu kadar uzun periyotlu ve rastgele gözüken anahtar dizilerini üretir. Elde edilen anahtar bir fonksiyona sokulur. Bu fonksiyon genellikle XOR işlemidir. İşlem sonucunda şifreli metin elde edilir.

Şekil 1.6'da bir akan şifrenin şifreli metin üretme safhası ile beraber örnek gösterimi verilmiştir.



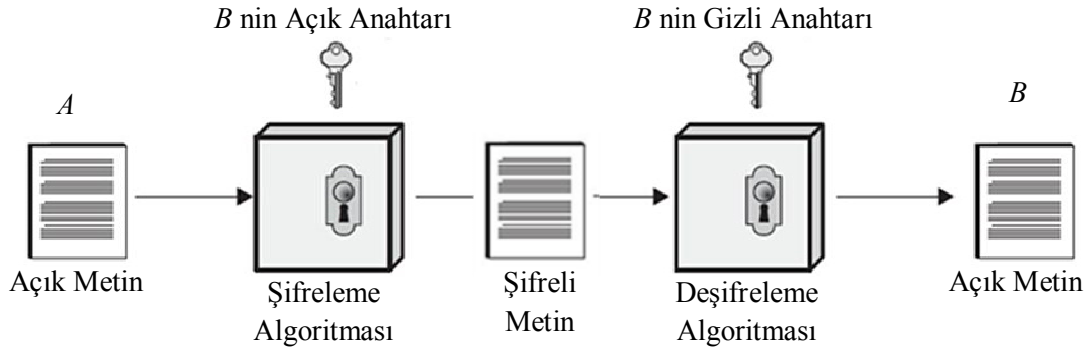
Şekil 1.6. Akan şifrenin XOR fonksiyonu ile gösterimi

Akan şifreleme algoritmalarına örnek olarak en yaygın olarak kullanılan 256 byte uzunluğuna kadar gizli anahtarlarla çalışabilen RC4 verilebilir.

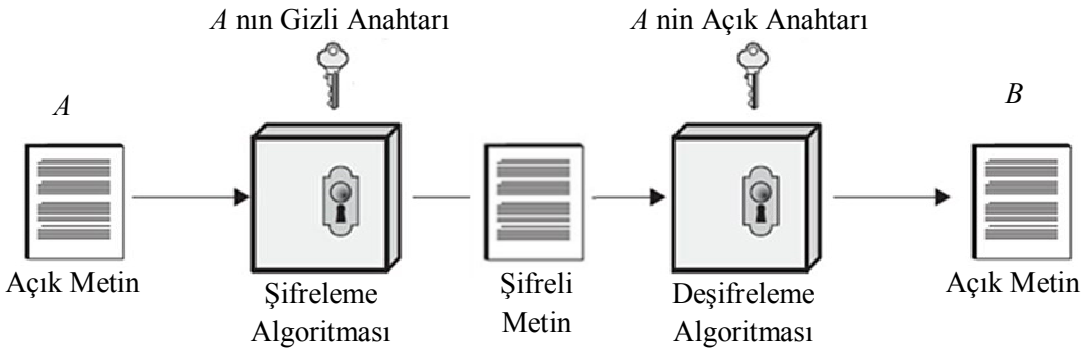
1.3. Asimetrik Şifreleme Algoritmaları

Asimetrik şifreleme algoritmalarında açık anahtar ve özel anahtar olmak üzere iki çeşit anahtar kullanılmaktadır. Asimetrik şifreleme ile iletişime geçecek taraflardan

her birinin iki anahtarı mevcuttur. Bunlardan açık anahtar karşı tarafa iletilirken herkesin erişimine açıktır, özel anahtar ise kişiye özel olup sadece o kişinin erişiminde gizlidir. Bu anahtarlar birbirine matematiksel bir ilişkiyle bağlanmıştır. Asimetrik şifreleme algoritmalarına örnek olarak RSA (Rivest vd., 1978) , ECC (Koblitz, 1987) ve ElGamal (ElGamal, 1985) verilebilir. Temel olarak bir asimetrik şifreleme algoritması Şekil 1.7'deki gibi çalışır.



Açık Anahtar ile Asimetrik Şifreleme (a)



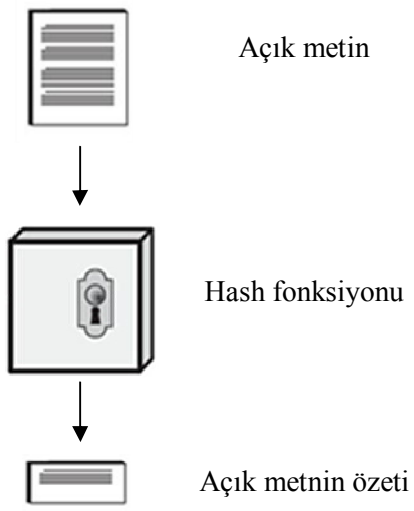
Gizli Anahtar ile Asimetrik Şifreleme (b)

Şekil 1.7. Asimetrik şifreleme algoritmalarının çalışma mantığı

1.4. Hash Algoritmaları

Veri bütünlüğü ve kimlik doğrulaması gibi uygulamalarda kullanılan Hash algoritmaları, değişik uzunluktaki bit dizilerini sabit uzunluklu bit dizilerine taşır. Kolay hesaplanabilen hash algoritmaları için gönderilecek mesajdan matematiksel yollarla sabit uzunlukta sayısal bilgi üretme işlemidir denilebilir. Üretilen bu anlamsız bilgiye mesaj özeti denir.

Hash algoritmaları geri dönüşümü olmayan, tek yönlü algoritmalardır. Algoritma da amaçlanan, aynı özeti veren iki farklı mesajın bulunmasının mümkün olmamasıdır. Hash algoritmalarının çalışma biçimi Şekil 1.8'de gösterilmiştir.



Şekil 1.8. Hash algoritmalarının çalışma mantığı

1.5. Örnek Bir Blok Şifre: AES (Advanced Encryption Standart)

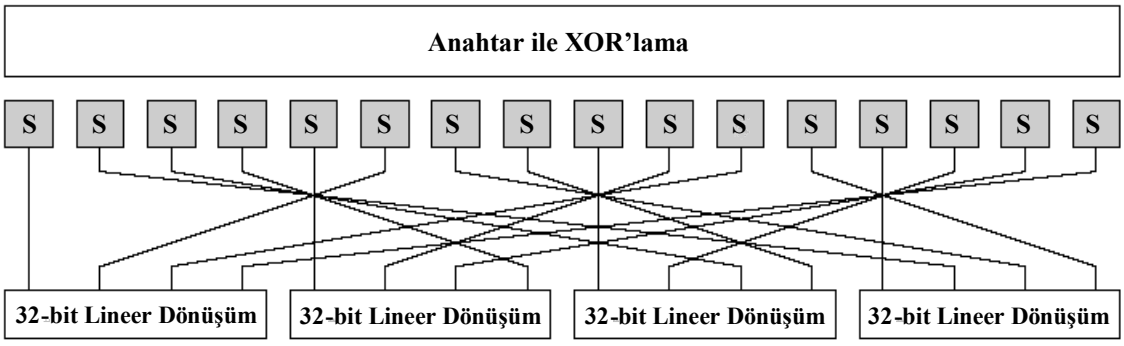
1990'lı yıllara gelindiğinde DES (Data Encryption Standart) NIST (National Institute of Standards and Technology -Ulusal Standartlar ve Teknoloji Enstitüsü) tarafında güvenli bir şifreleme algoritması olarak görülmekteydi. 1991 yılında Biham ve Shamir tarafından yapılan diferansiyel saldırı (Biham ve Shamir, 1991) ile DES sorgulanmaya başlandı. Fakat DES algoritmasının asıl çöküşünü sağlayan darbe 1993 yılında Japonya'dan geldi. Mitsuri Matsui doğrusal kriptanaliz yöntemini (Matsui, 1994) keşfetti ve DES'in kırılabileceğini gösterdi. NIST, 1997'de yeni bir şifreleme standardı için bir yarışma başlattı. Yarışmaya katılan 15 aday şifreleme algoritması arasından Belçikalı kriptograflar Vincent Rijmen ve Joan Daemen tarafından tasarlanan Rijndael adındaki algoritma kazandı. 2001 yılında Rijndael şifreleme algoritması AES (Advanced Encryption Standard-Gelişmiş Şifreleme Standardı) adıyla yeni şifreleme standardı olarak kabul gördü. Günümüzde AES hala bütün dünyada yaygın olarak kullanılan güvenli bir şifreleme algoritmasıdır.

AES 128-bit veri bloklarını 128, 192, 256-bit anahtar seçenekleri ile şifreleyen bir algoritmadır. SPN mimarisi tabanlıdır. Döngü sayısı anahtar uzunluğuna göre değişmektedir. 128-bit anahtar uzunluğu için AES şifreleme algoritması 10 döngüde şifreleme yaparken 192 ve 256-bit anahtar uzunlukları için sırasıyla 12 ve 14 döngüde şifreleme yapmaktadır.

AES algoritmasında her döngü dört katmandan oluşur. Her döngü dört adım içerir:

- Byte Yerdeğiştirme (SubBytes),
- Satırları Öteleme (ShiftRows),
- Sütunları Karıştırma (MixColumns),
- Döngü Anahtarı Ekleme (AddRoundKey).

İlk olarak 128-bit veri 4×4 byte matrisine dönüştürülür. Bu matrise durum (state) matrisi denir. Her döngüde sırasıyla gerçekleştirilen yukarıdaki adımlardan byte yerdeğiştirme adımında 8-bit (byte) değerler farklı 8-bit (byte) değerler ile yer değiştirilir. Bu dönüşüm doğrusal olmayan bir dönüşümdür ve GF(2^8) sonlu cisminde ters haritalama tabanlıdır. Satırları öteleme adımında byte değerlerinin permütasyonu ile byte değerlerinin sırası değiştirilirken, sütunları karıştırma doğrusal dönüşümde 32-bit giriş değerlerinden sabit bir matris çarpımı yardımıyla 32-bit çıkış değerleri elde edilmektedir. Diğer yandan son adım olan döngü anahtarı ekleme evresinde 128-bit anahtar seçeneği ile şifreleme yapan AES şifresi için anahtar planlama evresinden gelen 128-bit anahtar değeri ile o anki blok XOR'lama işlemine alınır. Tek döngülük SPN mimarisine uygun AES algoritması Şekil 1.9'da gösterilmektedir.



Şekil 1.9. AES blok şifresindeki tek döngülük SPN mimarisi

AES şifresinde döngü yapısına ait özellikler aşağıda verilmiştir:

1. Her döngüde tersi alınabilir dönüşümler kullanır,
2. Son döngü hariç her döngüde SubBytes, ShiftRows, MixColumns ve AddRoundKey dönüşümleri kullanır. Sadece son döngüde MixColumns dönüşümü kullanılmaz,
3. Anahtar planlama evresinde gizli anahtar kullanılarak döngü sayısı kadar farklı anahtar üretilir,
4. Deşifreleme kısmında ters dönüşümler kullanılır: InvSubByte, InvShiftRows, InvMixColumns ve AddRounKey (tersi kendisidir-XOR işlemi).

1.5.1. Byte Yerdeğiştirme (SubBytes) Dönüşümü

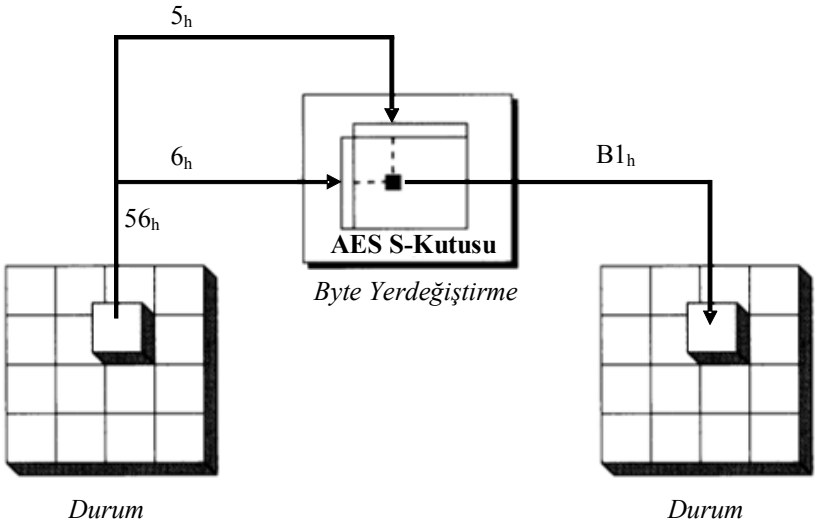
AES şifresi her byte (8-bit) değere karşılık farklı bir byte değerine dönüşümü yapan ve şifreye doğrusal olmama özelliğini katan bir S-kutusu kullanır. S-kutusunun tasarımı iyi kriptografik özellikler veren sonlu cisimde ters haritalama işlemi kullanılarak yapılmıştır. AES şifresinde kullanılan S-kutusu Tablo 1.3'te verilmiştir. Tablo 1.3'teki tüm değerler hexadecimal notasyondadır. Daha ayrıntılı bilgi (FIPS 197, 2001)'den elde edilebilir.

Tablo 1.3. AES S-kutusu

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

S-kutusunun sonlu bir cisimde ters alma işlemi ile tasarlanmasındaki amaç doğrusal kriptanaliz ve diferansiyel kriptanalize karşı dayanıklı bir S-kutusu seçme

amaçlıdır. Örneğin AES S-kutusu doğrusal olmama değeri 112 ve fark dağılım tablosundaki en büyük değer 4'tür. AES S-kutusu Tablo 1.3'de Hexadecimal formda verilmiştir. Örneğin tabloya göre 56_h değeri $B1_h$ değerine haritalanır. Bu işlem Şekil 1.10'da gösterilmiştir.



Şekil 1.10. AES SubBytes işlemi

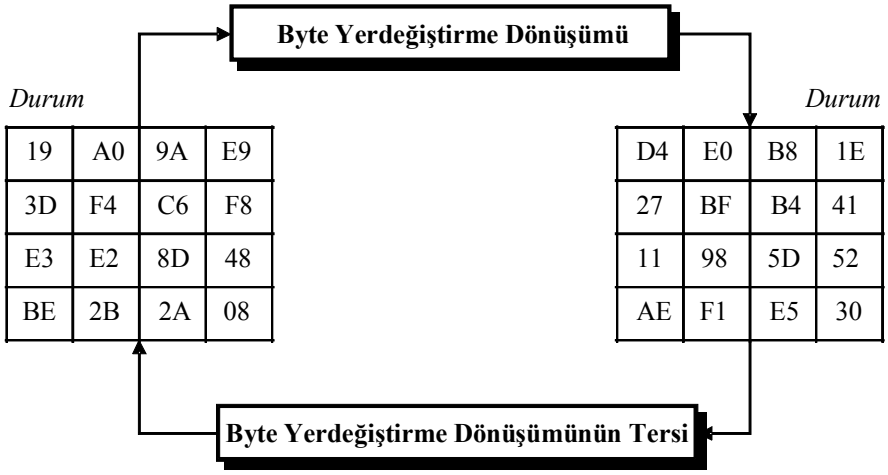
Deşifreleme yapılırken AES S-kutusunun tersi kullanılır. Bu aşama InvSubByte olarak isimlendirilmiştir. AES S-kutusunun tersi Tablo 1.4'te verilmiştir. Tabloya göre örneğin $B1_h$ değerinin tersi 56_h olarak bulunabilir.

Tablo 1.4. AES S-kutusunun tersi

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

Örnek 1.1.

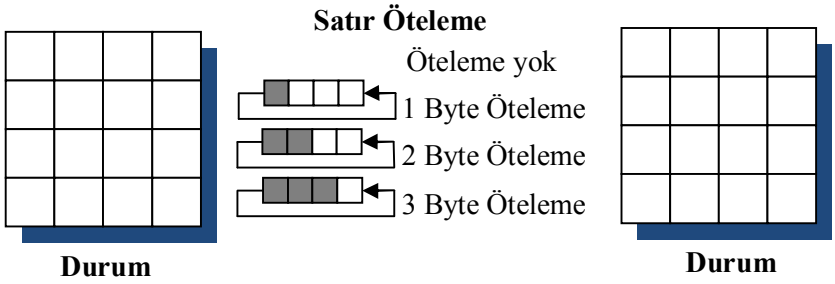
$19_h, 3D_h, E3_h, BE_h, A0_h, F4_h, E2_h, 2B_h, 9A_h, C6_h, 8D_h, 2A_h, E9_h, F8_h, 48_h, 08_h$ 128-bit değerin SubByte ve InvSubByte işlemleri Şekil 1.11'deki gibi gösterilebilir.



Şekil 1.11. AES SubBytes işlemi örneği

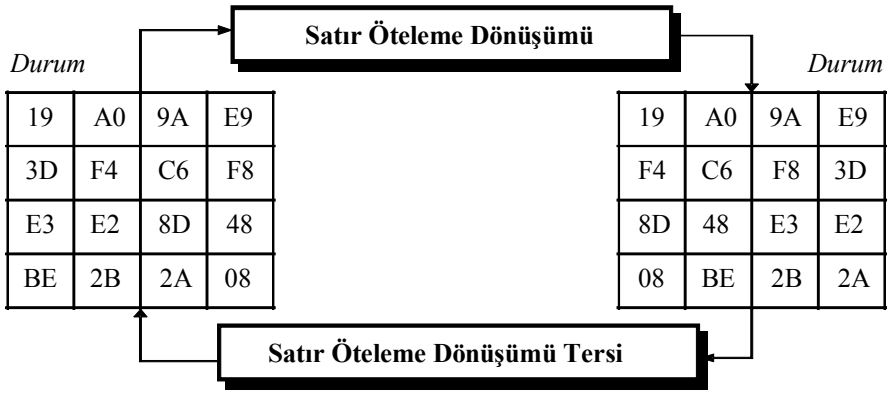
1.5.2. ShiftRows (Satırları Öteleme) Dönüşümü

AES şifresi 4×4 boyutunda bir durum matrisi şeklinde değerlendirilirse satırları öteleme dönüşümü byte değerlerinin sola öteleme işlemidir. İlk satırda sola öteleme yapılmaz iken ikinci, üçüncü ve dördüncü satırlar sırasıyla 1 defa, 2 defa ve 3 defa sola ötelenir. Bu işlem Şekil 1.12'de gösterilmiştir.



Şekil 1.12. AES ShiftRows işlemi

Satırları öteleme dönüşümü 4×4 durum matrisi üzerinde Şekil 1.13'de verilmiştir.



Şekil 1.13. AES ShiftRows işlemi örneği

Diğer yandan AES şifresindeki satırları öteleme işlemi, Şekil 1.9'daki SPN mimarisi düşünüldüğünde 1. byte 1. byte'a, 2. byte 14. byte'a, 3. byte 11. byte'a ve 4. byte 8. byte'a ve bu şekilde devam eden byte değerlerinin permütasyonudur.

1.5.3. Sütunları Karıştırma (MixColumns) Dönüşümü

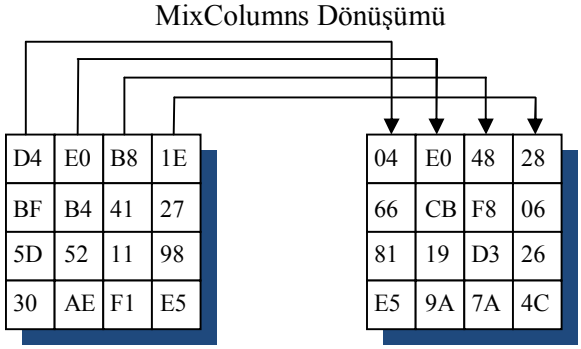
AES şifresinin içyapısında 32-bit'ten 32-bite dönüşüm yapan bir doğrusal dönüşüm bulunur ve bu dönüşüm MixColumns (sütunları karıştırma) olarak isimlendirilir. Bu dönüşüm doğrusal ve diferansiyel kriptanalizi zorlaştırıcı etki yapma amacındadır ve sonlu cisimde çarpma tabanlıdır. Temel olarak bu dönüşüm bir matris çarpımı şeklinde işlemektedir. AES şifresinin doğrusal dönüşümü, $y_0,...,y_3$, $a_0,...,a_3$ 8-bit değerleri yani 1 byte değerleri temsil etmek üzere aşağıdaki şekilde gösterilebilir.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 02_h & 03_h & 01_h & 01_h \\ 01_h & 02_h & 03_h & 01_h \\ 01_h & 01_h & 02_h & 03_h \\ 03_h & 01_h & 01_h & 02_h \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

AES şifreleme algoritmasında kullanılan MixColumns dönüşümü tersi kendisi bir dönüşüm değildir. Bu sebeple deşifreleme yapılırken InvMixColumns dönüşümünde, yukarıda verilen matrisin tersi olan aşağıdaki dönüşüm kullanılmaktadır.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0E_h & 0B_h & 0D_h & 09_h \\ 09_h & 0E_h & 0B_h & 0D_h \\ 0D_h & 09_h & 0E_h & 0B_h \\ 0B_h & 0D_h & 09_h & 0E_h \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Örnek 1.2. $D4_h, BF_h, 5D_h, 30_h, E0_h, B4_h, 52_h, AE_h, B8_h, 41_h, 11_h, F1_h, 1E_h, 27_h, 98_h, E5_h$ 128-bit değeri şifreleme yapılırken MixColumns aşamasında aşağıdaki gibi işleme sokulur.



Şekil 1.14. AES MixColumns işlemi örneği

Oluşturulan durum matrisinin ilk sütunu olan $D4_h, BF_h, 5D_h, 30_h$ 32-bitlik değeri şifreleme için kullanılan doğrusal dönüşüm ile sonlu cisimde aşağıdaki gibi çarpılarak $04_h, 66_h, 81_h, E5_h$ değerine ulaşılır.

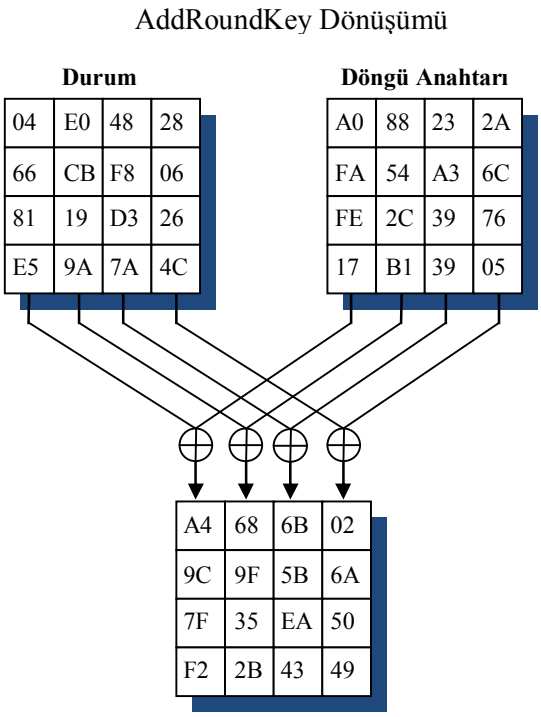
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} D4 \\ BF \\ 5D \\ 30 \end{bmatrix} = \begin{bmatrix} 04 \\ 66 \\ 81 \\ E5 \end{bmatrix}$$

Örnekten de anlaşılacağı gibi 128-bit blokları şifreleyen AES algoritmasında her döngüde 4 defa 32-bit'ten 32-bite doğrusal dönüşüm işlemi gerçekleştirilmektedir. Son döngüde MixColumns dönüşümü kullanılmaz.

1.5.4. AddRoundKey (Döngü Anahtarı Ekleme)

Anahtar planlama aşamasında gizli anahtar kullanılarak her döngü için farklı bir anahtar üretilir. AddRoundKey aşamasında döngü anahtarı ile MixColumns dönüşümünden çıkan durum matrisi XOR işlemine tabi tutulur. Bu işleme beyazlatma (whitening) adı da verilir. Bu işlem şifrede güvenliğin artırılması amacıyla uygulanır.

Örnek 1.3. $04_h,66_h,81_h,E5_h,E0_h,CB_h,19_h,9A_h,48_h,F8_h,D3_h,7A_h,28_h,06_h,26_h,4C_h$ 128-bit değeri şifreleme yapılırken MixColumns dönüşümünün çıkışı olsun ve anahtar planlama evresinden $A0_h,FA_h,FE_h,17_h,88_h,54_h,2C_h,B1_h,23_h,A3_h,39_h,39_h,2A_h,6C_h,76_h,05_h$ 128-bit değeri ilgili döngü için oluşturulmuş olsun. Bu durumda AddRoundKey işlemi Şekil 1.15'deki gibi özetlenebilir.



Şekil 1.15. AES AddRoundKey işlemi örneği

128-bit verinin $32_h,88_h,31_h,E0_h,43_h,5A_h,31_h,37_h,F6_h,30_h,98_h,07_h,A8_h,8D_h,$ $A2_h,34_h$ 128-bitlik anahtar $2B_h,28_h,AB_h,09_h,7E_h,AE_h,F7_h,CF_h,15_h,D2_h,15_h,4F_h,16_h,$ $A6_h,88_h,3C_h$ ile 10 döngüde şifrelenmesi işlemi aşağıdaki gibidir.

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

1.5.5. Tezin Önemi ve Gerekçesi

Bu tezde simetrik şifreleme algoritma sınıfına giren blok şifreleme algoritmalarında kullanılan doğrusal dönüşüm yapılarının kriptografik özellikleri incelenecektir. Literatürde bulunan şifreleme algoritmalarının içyapısındaki doğrusal dönüşüm yapıları incelenerek bir blok şifrede kullanılmak üzere kriptografik özellikleri iyi doğrusal dönüşüm yapılarının tasarımı araştırılacaktır. Buna ek olarak bu yapıların uygulamasında ortaya çıkan şifreleme ve deşifreleme işlemlerindeki hız farkını gidermek için bir uygulama yönteminden bahsedilecektir.

BÖLÜM 2

MATEMATİKSEL ALT YAPI

Bu bölümde şifreleme algoritmalarının tasarımı açısından önemli olan bazı matematiksel tanım ve teorilere yer verilecektir. Bu tanım ve teorilerin ispatları (Lidl ve Niederreiter, 1994), (Stinson, 2002) ve (Ling, 2004)'den elde edilebilir.

2.1. Sonlu Cisim Teorisi

Tanım 2.1. a, b tamsayı ve m pozitif tamsayı olsun. Eğer m, $b-a$'yı bölüyorsa $a \equiv b \ (mod\ m)$ şeklinde yazabiliriz. $a \equiv b \ (mod\ m)$ ifadesine denklik denir ve a, b'ye $(mod\ m)$'e göre denktir denir. Tamsayı m'ye de modulo denir.

Biz aritmetik modulo m : $Z_m \{0,1,...,m-1\}$ seti ile iki işlem toplama ve çarpma tabanlı tanımlayabiliriz. Z_m 'de toplama ve çarpma işlemleri gerçek toplama ve çarpma işlemleri olacak sadece sonuçlar modulo m'ye göre indirgenecektir.

Tanım 2.2. Cisim, toplama ve çarpma işlemleri ile aşağıdaki aksiyomları sağlayan elemanları boş olmayan bir Z setidir.

1. Toplamada kapalılık özelliği;
   $a,b \in Z_m \rightarrow a+b \in Z_m$
2. Çarpmada kapalılık özelliği;
   $a,b \in Z_m \rightarrow a.b \in Z_m$
3. Toplamada değişme özelliği;
   $a,b \in Z_m \rightarrow a+b=b+a$
4. Çarpmada değişme özelliği;

$a,b \in Z_m \rightarrow a.b = b.a$

5. Toplamada geçişme özelliği;

   $a,b,c \in Z_m \rightarrow (a+b)+c = a+(b+c)$

6. Çarpmada geçişme özelliği;

   $a,b,c \in Z_m \rightarrow (a.b).c = a.(b.c)$

7. Çarpmada dağılma özelliği;

   $a,b,c \in Z_m \quad \rightarrow \quad (a.b).c = a.(b.c)$

   $a.(b+c) = a.b + a.c$

İki farklı birim elemanı (identity) 0 ve 1 (sırası ile toplamaya ve çarpmaya göre) aşağıdakileri sağlayan Z_m 'in içinde olmak zorundadır.

8. $a+0 = a, \forall a \in Z_m$
9. $a.1 = a$ ve $a.0 = 0, \forall a \in Z_m$
10. $a \in Z_m$ için a nın toplamaya göre tersi *m-a* 'dır.
11. a $\in Z_m$ için a nın çarpmaya göre tersi a^{-1} dir ve $a^{-1}.a = 1$ olmalıdır.

Tanım 2.3. Eğer Z_m seti yukarıdaki 1, 2, 5, 6 ve 7 numaralı aksiyomları sağlıyor ise toplama ve/veya çarpma işlemlerine göre gruptur denir. Buna ek olarak 3 ve 4 numaralı aksiyomlar da sağlanıyor ise abelyan grup adı verilir.

Tanım 2.4. Eğer Z_m seti yukarıdaki aksiyomlardan 1'den 9'a kadar olan aksiyomları sağlıyor ise bu sete halka denir.

Tanım 2.5. Eğer Z_m seti yukarıdaki aksiyomların hepsini sağlıyor ise cisim olarak adlandırılır. Sonlu elemana sahip cisimlere sonlu cisim adı verilir.

Örnek 2.1. Z_4'ü düşünelim.

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

×	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Görüldüğü gibi Z_4 bir cisim oluşturmaz. Çünkü çarpma işlemine göre 2'nin tersi yoktur.

Örnek 2.2. Z_5'i inceleyelim.

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	0
2	2	3	4	0	1
3	3	4	0	1	2
4	4	0	1	2	3

×	0	1	2	3	4
0	0	0	0	0	0
1	0	1	2	3	4
2	0	2	4	1	3
3	0	3	1	4	2
4	0	4	3	2	1

Z_5, Tanım 2.2' de verilen tüm aksiyomları sağladığı ve sonlu elemana sahip olduğu için sonlu cisimdir.

Teorem 2.1. Eğer p asal sayı ise Z_p bir cisimdir.

Teorem 2.2. Z_p^*, Z_p'nin bir alt seti olup, çarpmaya göre tersi olan elemanları içerir. Eğer p asal sayı ise çevrimsel bir gruptur.

Tanım 2.6. Z_p cisminin 0 olmayan bir elemanı olan α'nın derecesi $\alpha^k = 1$ olmak üzere en küçük k değeridir.

Tanım 2.7. $\bmod p$'ye göre $(p-1)$ derecesine sahip bir α elemanına asal eleman denir.

Tanım 2.8. $a \geq 1$ ve $m \geq 1$ olsun. Eğer $OBEB(a,m) = 1$ ise a ve m için aralarında asal denir. Z_m de m ile aralarında asal olan tamsayıların sayısı genellikle $\phi(m)$ ile tanımlanır ve bu fonksiyona *Euler phi fonksiyonu* denir.

Teorem 2.3. $\phi(m)$, m'nin asal üslerinin çarpanlarına ayrılması ile bulunabilir. p_i'ler farklı asal sayılar olmak üzere $e_i > 0$ ve $1 \leq i \leq n$ için sırasıyla m ve $\phi(m)$ifade (2.1)'deki gibi gösterilebilir.

$$m = \prod_{i=1}^{n} p_i^{e_i}, \qquad \phi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i-1}). \tag{2.1}$$

Örnek 2.3. $\phi(60)$değerini bulalım. Yani 60'dan küçük 60 ile aralarında asal olan tamsayıların sayısını bulalım.

$$m = 60 = 2^2.3.5 \Rightarrow \phi(60) = (2^2 - 2^1).(3^1 - 3^0).(5^1 - 5^0) = 2.2.4 = 16$$

Tanım 2.9. p asal ve α $\bmod p$'ye göre asal eleman olsun. Herhangi bir $\beta \in Z_p^*$, $\beta = \alpha^i (0 < i < p-2)$ olmak üzere yazılabilir. $\beta = \alpha^i$'nin derecesi $\dfrac{p-1}{OBEB(p-1,i)}$'dir. Böylece eğer $OBEB(p-1,i) = 1$ ise β asal bir elemandır. Dolayısıyla $\bmod p$'ye göre asal elemanların sayısı $\phi(p-1)$'dir.

Teorem 2.4. p asal ve $\alpha \in Z_p^*$ olsun. Eğer $(p-1)$'i bölen tüm asal q değerleri için $\alpha^{\frac{p-1}{q}} \pmod p \neq 1$ ise o zaman α $\bmod p$'ye göre asaldır.

Örnek 2.4. $p = 11$ için asal elemanları bulalım. Asal elemanları elde edebilmek için önce en küçük asal elemanı bulmamız gerekmektedir. Çünkü p değerinin asal elemanlarını bulmak için aşağıda gösterildiği gibi, en küçük böleninden başlayarak üsler

şeklinde yazıldığında 1'den $p-1$'e kadar tüm tamsayıları veren en küçük değer bulunur. En küçük asal eleman aşağıda gösterildiği gibi 2'dir.

$$2^0 \bmod 11 = 1$$
$$2^1 \bmod 11 = 2$$
$$2^2 \bmod 11 = 4$$
$$2^3 \bmod 11 = 8$$
$$2^4 \bmod 11 = 5$$
$$2^5 \bmod 11 = 10$$
$$2^6 \bmod 11 = 9$$
$$2^7 \bmod 11 = 7$$
$$2^8 \bmod 11 = 3$$
$$2^9 \bmod 11 = 6$$

Bir sonraki adım $OBEB(p-1, i) = 1$ şartını sağlayan sayıları bulmaktır. Bu değerler bulunduktan sonra $2^i \bmod p$'de karşılık gelen değerler asal elemanlar olarak karşımıza çıkar. $OBEB(i, 10) = 1$ için i değerleri sırasıyla 1,3,7,9 olduğu için asal elemanları sırasıyla 2,8,7,6 olarak buluruz. Bu asal elemanlara cismin üreteci denir. Dolayısı ile $p = 11$ için 2, 8, 7 ve 6 üreteç elemanları ile sonlu cisim oluşturulabilir.

$8^0 \bmod 11 = 1$ $7^0 \bmod 11 = 1$ $6^0 \bmod 11 = 1$
$8^1 \bmod 11 = 8$ $7^1 \bmod 11 = 7$ $6^1 \bmod 11 = 6$
$8^2 \bmod 11 = 9$ $7^2 \bmod 11 = 5$ $6^2 \bmod 11 = 3$
$8^3 \bmod 11 = 6$ $7^3 \bmod 11 = 2$ $6^3 \bmod 11 = 7$
$8^4 \bmod 11 = 4$ $7^4 \bmod 11 = 3$ $6^4 \bmod 11 = 9$
$8^5 \bmod 11 = 10$ $7^5 \bmod 11 = 10$ $6^5 \bmod 11 = 10$
$8^6 \bmod 11 = 3$ $7^6 \bmod 11 = 4$ $6^6 \bmod 11 = 5$
$8^7 \bmod 11 = 2$ $7^7 \bmod 11 = 6$ $6^7 \bmod 11 = 8$
$8^8 \bmod 11 = 5$ $7^8 \bmod 11 = 9$ $6^8 \bmod 11 = 4$
$8^9 \bmod 11 = 7$ $7^9 \bmod 11 = 8$ $6^9 \bmod 11 = 2$

Tanım 2.10. Z_m bir cisim olmak üzere set $Z_m[x]:=\left\{\sum_{i=0}^{n} a_i x^i : a_i \in Z_m, n \geq 0\right\}$, Z_m üzerine bir polinom halka olarak isimlendirilir. $Z_m[\mathrm{x}]$'in bir elemanı Z_m üzerine polinom olarak isimlendirilir. Pozitif dereceli bir polinom $f(x)=\sum_{i=0}^{n} a_i x^i$ için derece$(\mathrm{g(x)})<$derece$(\mathrm{f(x)})$, derece$(\mathrm{h(x)})<$derece$(\mathrm{f(x)})$ ve $\mathrm{f(x)=g(x)h(x)}$ şartlarını sağlayacak şekilde iki polinom varsa *f(x)* polinomu Z_m üzerine indirgenebilir aksi takdirde pozitif dereceli *f(x)* polinomu Z_m üzerine indirgenemez polinom olarak tanımlanabilir.

Teorem 2.5. *f(x)*, derecesi 1'den büyük Z_m cismi üzerine bir polinom olmak üzere $Z_m[x]/f(x)$ sadece ve sadece *f(x)* indirgenemez ise cisimdir.

Örnek 2.5.

a) $f(x)=x^4+2x^6 \in Z_3[x]$ polinomu 6. dereceden bir polinomdur ve $f(x)=x^4(1+2x^2)$ şeklinde yazılabileceğinden indirgenebilir bir polinomdur.

b) $g(x)=1+x+x^2 \in Z_2[x]$ polinomu 2. dereceden bir polinomdur ve indirgenemezdir.

c)Aynı şekilde Z_2 üzerine $1+x+x^3$ ve$1+x^2+x^3$ polinomlarının doğrusal çarpanları olmadığı için indirgenemez olduğu gösterilebilir.

Tanım 2.10 ve Teorem 2.5'ten yola çıkarak, p asal ve $n \geq 1$ (n, *f(x)* polinomunun derecesi) olmak üzere $q=p^n$ elemana sahip bir sonlu cisim vardır diyebiliriz. Diğer bir ifade ile $Z_m[x]/f(x)$ sonlu cisim ise bu sonlu cisim F_{P^n} ya da $GF(p^n)$ şeklindeki ifade ile tanımlanabilir ve bu özel sonlu Galois cismi (galois field) $GF(p)$ cisminin *n*. dereceden genişletilmiş cismi olarak adlandırılır. Örneğin $GF(2^3)$, $GF(2)$ cisminin 3. dereceden genişletilmiş cismi olarak isimlendirilir ve $GF(2)$ cismine, $GF(2^3)$ cisminin taban cismi adı verilir.

Tanım 2.11. Asal polinom (primitive polynomial) taban cisminden genişletilmiş cismin tüm elemanlarını üretebilen polinomdur.

Teorem 2.6. Herhangi bir asal ya da asal üs q ve pozitif n için $GF(q)$ üzerine *n.* dereceden bir asal polinom vardır ve bu asal polinomların sayısı $s_q(n) = \dfrac{\phi(q^n - 1)}{n}$ şeklinde bulunabilir.

Örnek 2.6. $GF(2^3)$ sonlu cismi için 3. dereceden asal polinomların sayısını bulalım.

$$s_q(n) = \frac{\phi(q^n - 1)}{n} \Rightarrow s_{2^3}(3) = \frac{\phi(2^3 - 1)}{3} = \frac{\phi(7)}{3} = \frac{(7^1 - 7^0)}{3} = 2$$

Dolayısı ile $GF(2^3)$ cismindeki asal polinom sayısı 2'dir. Derecesi 1 ile 5 arasında olan asal polinomlar aşağıda verilmiştir.

n	Asal Polinomlar
1	$x+1$
2	x^2+x+1
3	$x^3+x+1,\ x^3+x^2+1$
4	$x^4+x+1,\ x^4+x^3+1$
5	$x^5+x^2+1,\ x^5+x^3+x^2+x+1,\ x^5+x^3+1,$ $x^5+x^4+x^3+x+1,\ x^5+x^4+x^3+x^2+1,\ x^5+x^4+x^2+x+1$

Örnek 2.7. $Z_2[x]/(x^3+x+1)$ bir sonlu cisim olmak üzere, bu sonlu cismin eleman sayısı 8'dir ve $GF(2^3)$ şeklinde gösterilebilir. Aşağıda 0 haricindeki $GF(2^3)$'ün elemanları verilmiştir.

$$x^1 = x$$
$$x^2 = x^2$$
$$x^3 = x+1$$
$$x^4 = x^2+x$$
$$x^5 = x^2+x+1$$
$$x^6 = x^2+1$$
$$x^7 = 1$$

Bu cismin toplam ve çarpım tabloları aşağıda verilmiştir.

+	1	x	x^2	$x+1$	x^2+x	x^2+x+1	x^2+1
1	0	$x+1$	x^2+1	x	x^2+x+1	x^2+x	x^2
x	$x+1$	0	x^2+x	1	x^2	x^2+1	x^2+x+1
x^2	x^2+1	x^2+x	0	x^2+x+1	x	$x+1$	1
$x+1$	x	1	x^2+x+1	0	x^2+1	x^2	x^2+x
x^2+x	x^2+x+1	x^2	x	x^2+1	0	1	$x+1$
x^2+x+1	x^2+x	x^2+1	$x+1$	x^2	1	0	x
x^2+1	x^2+x	x^2+x+1	1	x^2+x	$x+1$	x	0

×	1	x	x^2	$x+1$	x^2+x	x^2+x+1	x^2+1
1	1	x	x^2	$x+1$	x^2+x	x^2+x+1	x^2+1
x	x	x^2	$x+1$	x^2+x	x^2+x+1	x^2+1	1
x^2	x^2	$x+1$	x^2+x	x^2+x+1	x^2+1	1	x
$x+1$	$x+1$	x^2+x	x^2+x+1	x^2+1	1	x	x^2
x^2+x	x^2+x	x^2+x+1	x^2+1	1	x	x^2	$x+1$
x^2+x+1	x^2+x+1	x^2+1	1	x	x^2	$x+1$	x^2+x
x^2+1	x^2+1	1	x	x^2	$x+1$	x^2+x	x^2+x+1

2.1.1. Sonlu Cisimde Toplama İşlemi

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

Örnek 2.8. $a=x^6+x^4+x+1$ ve $b=x^7+x^6+x^3+x$ polinomları verilmiştir. Buna göre $a+b=(x^6+x^4+x+1)+(x^7+x^6+x^3+x)=x^7+x^4+x^3+1$ olarak bulunur.

Örnek 2.9. $a = x^6 + x^5 + x^4 + x^2 + x + 1$ ve $b = x^7 + x^5 + x^4 + x^2 + 1$ ise $a + b = x^7 + x^6 + x$ olarak bulunur. Vektörel olarak düşünüldüğünde a = (01110111) ve b = (10110101) olarak yazılabilir. O zaman $a + b$ = (11000010) şeklinde de gösterilebilir. Aynı şekilde hexadecimal olarak bu toplamın sonucu $77_h + B5_h = C2_h$'dir.

2.1.2. Sonlu Cisimde Çarpma İşlemi

Sonlu cisim aritmetiğinde çarpma, polinomların birbirleri ile aritmetik çarpımı şeklindedir. Fakat çarpma sonucunda doğal olarak sonlu cismin derecesinden daha yüksek dereceli elemanlar oluşabilir. Bu durumda elemanları, cismi oluşturan indirgenemez polinoma göre indirgemek gerekir. Dolayısı ile sonlu cisim aritmetiğinde çarpma işlemi indirgenemez polinoma göre indirgeme ya da mod alma işlemidir.

Örnek 2.10. GF(2^4) sonlu cisminde $a = 1101$ ve $b = 0101$ ve indirgenemez polinom $x^4 + x + 1$ seçilsin. $a \otimes b$ sonlu cisim çarpmasını hesaplamak için öncelikle cismi oluşturalım.

$$
\begin{aligned}
x^1 &= x & x^9 &= x^3 + x \\
x^2 &= x^2 & x^{10} &= x^2 + x + 1 \\
x^3 &= x^3 & x^{11} &= x^3 + x^2 + x \\
x^4 &= x + 1 & x^{12} &= x^3 + x^2 + x + 1 \\
x^5 &= x^2 + x & x^{13} &= x^3 + x^2 + 1 \\
x^6 &= x^3 + x^2 & x^{14} &= x^3 + 1 \\
x^7 &= x^3 + x + 1 & x^{15} &= 1 \\
x^8 &= x^2 + 1
\end{aligned}
$$

$a \otimes b$ işlemi polinom olarak $a \otimes b = (x^3 + x^2 + 1)(x^2 + 1) = (x^5 + x^4 + x^2 + x^3 + x^2 + 1)$ şeklinde bulunur. Fakat çarpma sonucu elde edilen x^5, x^4 gibi elemanlar indirgenemez polinoma göre indirgendiğinde $x^5 = x^2 + x$ ve $x^4 = x + 1$ olacaktır. Dolayısı ile çarpma işlemi sonucu $a \otimes b = (x^2 + x + x + 1 + x^2 + x^3 + x^2 + 1) = x^3 + x^2$ olur. Vektörel olarak (1100) veya hexadecimal olarak C_h şeklinde yazılabilir.

Yukarıda anlatılan indirgeme işleminin yanı sıra mod alma işlemi ile de çarpma yapılabileceği aynı örnek için aşağıda verilmiştir.

$a \otimes b = (x^3 + x^2 + 1)(x^2 + 1) = (x^5 + x^4 + x^2 + x^3 + x^2 + 1) = x^5 + x^4 + x^3 + 1$ sonucu indirgenemez polinoma göre mod işlemine alınır.

$$x^5 + x^4 + x^3 + 1 \mod x^4 + x + 1$$

$$\begin{array}{l} 111001 \bmod 10011 \\ \oplus 10011 \\ \quad 011111 \\ \oplus \quad 10011 \\ \quad\quad 01100 \end{array}$$ olarak bulunur.

Şifreleme algoritmalarında kullanılan birçok bileşen (doğrusal dönüşümler ve S-kutuları gibi) sonlu cisimde çarpma tabanlıdır. Örneğin Bölüm 1'de anlatıldığı gibi AES şifreleme algoritmasında kullanılan S-kutusu ters haritalama işlemi ile tasarlanmıştır. Aynı şekilde AES'te kullanılan doğrusal dönüşüm sonlu cisimde çarpma yaparak 32-bit veriyi başka bir 32-bit veriye dönüştürür.

GF(2^8) cisminin $a(x)$ elemanı bir polinom ve $a_i \in GF(2)$ olmak üzere;

$$a_7 x^7 + a_6 x^6 + a_5 x^5 + a_4 x^4 + a_3 x^3 + a_2 x^2 + a_1 x + a_0$$

şeklinde verilebilir ve yukarıdaki ifade sonlu cisim elemanlarının polinomsal gösterimidir. Polinom $a(x)$ için $\{a_7, a_6, a_5, a_4, a_3, a_2, a_1, a_0\}$ elemanları $\{0,1\}$ setine ait bir vektöre karşılık gelir.

Literatürde sonlu cisimde çarpma işlemi için genel olarak kullanılan 2 yaklaşım mevcuttur (Lidl ve Niederreiter, 1994). Bunlardan ilki α ile çarpma (xtime) işlemi diğeri ise tablo okuma (table lookup) işlemidir. Bu bölümde bu iki işlem hakkında ayrıntılı bilgi verilecektir.

2.1.2.1. α ile Çarpma İşlemi (xtime)

xtime işlemi temel olarak sonlu cisimde 02_h veya α ile çarpma olarak tanımlanabilir. Bu değerler ile çarpma aslında sola bir bit kaydırma anlamına gelmektedir. Örneğin 55_h değeri xtime işlemine alınırsa (01010101) değeri bir sola kaydırılarak (10101010) değerine ulaşılır. Bu işlem aşağıda gösterilmiştir.

$$\text{xtime}(55_h) = 55_h \otimes 02_h = (01010101)(00000010)$$
$$= (\alpha^6 + \alpha^4 + \alpha^2 + 1)(\alpha) = (\alpha^7 + \alpha^5 + \alpha^3 + \alpha) = AA_h$$

xtime işlemi sonucunda eğer elde meydana gelirse sonlu cismin dışına çıkılmış olur. Bu durumda sonlu cismi oluşturan indirgenemez polinoma göre indirgeme işlemi yapılmalıdır. Dolayısıyla, AES şifresindeki indirgenemez polinom $x^8 + x^4 + x^3 + x + 1$ olduğundan çıkan sonuç $1B_h$ ile XOR'lanır.

Örnek olarak DB_h değerini, indirgenemez polinomu $x^8 + x^4 + x^3 + x + 1$ olan bir sonlu cisimde xtime işlemine alalım. $\text{xtime}(DB_h) = (11011011) \xrightarrow{\text{Bir Bit sola Ötele}} (10110110) = B6_h$ kaydırma işlemi sonucunda elde olduğundan indirgenemez polinom ile XOR'lama işlemi yapılarak $B6_h \oplus 1B_h = AD_h$ sonucuna ulaşılır.

Öte yandan, iki değeri çarparken ardarda xtime işlemi yapılarak sonuca ulaşılır. İşlemden tasarruf etmek için küçük olan değerin derecesi kadar xtime işlemi yapılır. Küçük değerin 1 olan bitlerinin bulunduğu xtime değerleri XOR'lanarak sonuca ulaşılır.

Örnek 2.11. Örneğin 72_h ile 14_h sayısını xtime işlemi ile çarpalım.

$72_h \otimes 14_h = (01110010)\ (00010100)$

1. xtime	72_h sola kaydır	11100100	$D4_h$
2. xtime	$D4_h$ sola kaydır	11001000	$C8_h$ elde olduğundan $1B_h$ ile XOR'lanınca sonuç $D3_h$ (11010011)
3. xtime	$D3_h$ sola kaydır	10100110	$A6_h$ elde olduğundan $1B_h$ ile XOR'lanınca sonuç BD_h (10111101)
4. xtime	BD_h sola kaydır	01111010	$7A_h$ elde olduğundan $1B_h$ ile XOR'lanınca sonuç 61_h (01100001)

72_h (01110010) ve 14_h (00010100) değerlerinin çarpımı hesaplanırken, 14_h'ün derecesi küçük olduğundan 4 xtime uygulanır. 14_h'ün 1 olan bit değerlerine göre 2. ve 4. xtime sonuçları aşağıdaki gibi XOR'lanarak çarpım $B2_h$ olarak elde edilir.

2. xtime	$D3_h$	11010011
4. xtime	61_h	01100001
XOR	$B2_h$	10110010

2.1.2.2. Tablo Okuma (Table Lookup)

Çarpma işlemi için bir başka yaklaşım ise tablo okuma yöntemidir. Ön işlem aşamasında tüm giriş değerleri için çarpım tablosu oluşturulur. 8-bitlik bir işlem için 2^8 adet, 4-bitlik bir işlem için ise 2^4 olası giriş değeri mevcuttur. Tablo oluşturulduktan sonra çarpma aşamasında tablodan çarpılacak değerler bulunur.

4-bitlik bir sonlu cisim için 16×16 boyutunda bir çarpım tablosu oluşturmak gerekmektedir. Örneğin GF(2^4)'te x^4+x+1 indirgenemez polinomu ile oluşturulan cismin çarpım tablosu Tablo 2.1.'de verilmiştir.

Tablo 2.1. GF(2^4) te x^4+x+1 polinomu ile oluşturulan cismin çarpım tablosu

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

B_h ile 8_h değerlerinin çarpımını bulmak için tablodan B satırı ile 8 sütunun kesişim yerine bakmak yeterli olacaktır. Dolayısıyla $B_h \otimes 8_h = 7_h$ dir. $x^4 + x^3 + 1$ ve $x^4 + x^3 + x^2 + x + 1$ indirgenemez polinomlarının çarpım tablosu Ek A'da verilmiştir.

Elbette ki tüm olası giriş değerleri için bir çarpım tablosu oluşturmak maliyeti arttırır. Örneğin GF(2^8)'de tanımlı olan AES doğrusal dönüşümü işlemleri için 256×256'lık bir çarpım tablosu oluşturmak gerekmektedir. Fakat AES doğrusal dönüşümünde şifreleme için sadece 02_h ve 03_h; deşifreleme işlemleri için ise 09_h, $0B_h$, $0D_h$ ve $0E_h$ değerleri kullanıldığından bu 6 değerin çarpım tablosunu oluşturmak yeterli olacaktır. Dolayısıyla AES için 6×256 lık bir çarpım tablosu oluşturulur. Bu tablonun ilk 16 değeri aşağıdaki gibidir. Tablodaki değerler hexadecimal notasyonda yazılmış olup tablonun tümü Ek B'de verilmiştir. Örnek olarak $07_{\text{h}} \otimes 0E_h = 07_h$ işleminin sonucu Tablo 2.2'ye göre $2A_h$'dır.

Tablo 2.2. AES şifresinde kullanılan cismin ilk 16 elemanının çarpım tablosu

×	**02**	**03**	**09**	**0B**	**0D**	**0E**
01	02	03	09	0B	0D	0E
02	04	06	12	16	1A	1C
03	06	05	1B	1D	17	12
04	08	0C	24	2C	34	38
05	0A	0F	2D	27	39	36
06	0C	0A	36	3A	2E	24
07	0E	09	3F	31	23	2A
08	10	18	48	58	68	70
09	12	1B	41	53	65	7E
0A	14	1E	5A	4E	72	6C
0B	16	1D	53	45	7F	62
0C	18	14	6C	74	5C	48
0D	1A	17	65	7F	51	46
0E	1C	12	7E	62	46	54
0F	1E	11	77	69	4B	5A

2.1.2.3. Sonlu Cisimde Çarpma İçin Yeni Bir Yöntem: Nokta Ürün (Dot Product)

Literatürde bulunan xtime ve tablo okuma işlemleri çarpma için ideal yöntemlerdir. Bu yöntemlerden xtime yöntemi dinamik bir işlem iken tablo okuma yöntemi statik bir yöntemdir. Tez çalışması sırasında sonlu cisimde çarpma yapmak için nokta ürün adında yeni bir yöntem önerilmektedir. Önerilen bu yöntem xtime ve tablo okuma işlemlerinin bir karmasıdır. Çarpma işleminin cismin tüm elemanları için birbirlerine yakın sürede tamamlanmasını hedef alan nokta ürün yöntemi aşağıda tanıtılmaktadır.

GF(2^4)'te tanımlanmış bir $a(x)$ elemanı , $\{x_3, x_2, x_1, x_0\}$ elemanları $\{0,1\}$ setine ait ve basamak değerleri $\{\alpha^3, \alpha^2, \alpha, 1\}$ olmak koşuluyla ifade (2.2)'deki gibi yazılabilir.

$$\alpha^3 x_3 + \alpha^2 x_2 + \alpha x_1 + x_0 \qquad (2.2)$$

Eğer $\alpha^3 x_3 + \alpha^2 x_2 + \alpha x_1 + x_0$ polinomu 1_h(0001) ile çarpılır ise yine kendisini verir. Bit değişimi olarak bakıldığında α^3 basamak değerini x_3, α^2 basamak değerini x_2, α basamak değerini x_1 ve 1 basamak değerini x_0 etkileyecektir.

Eğer 2_h ile çarpma yapılır ise xtime işleminde olduğu gibi α ile yani (0010) ile çarpma işlemi yapılmış olur. Bu durumda $(\alpha^3 x_3 + \alpha^2 x_2 + \alpha x_1 + x_0)(\alpha) = (\alpha^4 x_3 + \alpha^3 x_2 + \alpha^2 x_1 + \alpha x_0)$ sonucuna ulaşılmış olur. Cisim $\alpha^4 + \alpha^3 + 1$ indirgenemez polinomu ile indirgendiğinde $\alpha^4 = \alpha^3 + 1$ olduğundan çarpma işlemi $(\alpha^3 + 1)x_3 + \alpha^3 x_2 + \alpha^2 x_1 + \alpha x_0$ şeklinde yazılır. İfadeyi tekrar düzenlediğimizde ifade (2.3)'teki sonuca ulaşırız.

$$\alpha^3 (x_3 + x_2) + \alpha^2 x_1 + \alpha x_0 + x_3 \qquad (2.3)$$

Yukarıdaki ifadeden anlaşılacağı gibi 2_h ile çarpma yapıldığında α^3 basamak değerini x_3 ve x_2, α^2 basamak değerini x_1, α basamak değerini x_0 ve 1 basamak

değerini x_3 etkileyecektir. Bu işlemler olası tüm değerler için tekrarlandığında sonuçlar aşağıdaki Tablo 2.3'teki gibi olacaktır.

Tablo 2.3. $\alpha^4+\alpha^3+1$ İndirgenemez polinomunun basamak çarpım değerleri

	α^3	α^2	α	1
1_h	x_3	x_2	x_1	x_0
2_h	(x_3+x_2)	x_1	x_0	x_3
3_h	x_2	(x_2+x_1)	(x_1+x_0)	(x_3+x_0)
4_h	$(x_3+x_2+x_1)$	x_0	x_3	(x_3+x_2)
5_h	(x_2+x_1)	(x_2+x_0)	(x_3+x_1)	$(x_3+x_2+x_0)$
6_h	x_1	(x_1+x_0)	(x_3+x_0)	x_2
7_h	(x_3+x_1)	$(x_2+x_1+x_0)$	$(x_3+x_1+x_0)$	(x_2+x_0)
8_h	$(x_3+x_2+x_1+x_0)$	x_3	(x_3+x_2)	$(x_3+x_2+x_1)$
9_h	$(x_2+x_1+x_0)$	(x_3+x_2)	$(x_3+x_2+x_1)$	$(x_3+x_2+x_0)$
A_h	(x_1+x_0)	(x_3+x_1)	$(x_3+x_2+x_0)$	(x_2+x_1)
B_h	$(x_3+x_1+x_0)$	$(x_3+x_2+x_1)$	$(x_3+x_2+x_1+x_0)$	$(x_2+x_1+x_0)$
C_h	x_0	(x_3+x_0)	x_2	x_1
D_h	(x_3+x_0)	$(x_3+x_2+x_0)$	(x_2+x_1)	(x_1+x_0)
E_h	$(x_3+x_2+x_0)$	$(x_3+x_1+x_0)$	(x_2+x_0)	(x_3+x_1)
F_h	(x_2+x_0)	$(x_3+x_2+x_1+x_0)$	$(x_2+x_1+x_0)$	$(x_3+x_1+x_0)$

GF(2^4)'te tanımlı $\alpha^4+\alpha^3+\alpha^2+\alpha+1$ ve $\alpha^4+\alpha+1$ indirgenemez polinomunun tablosu Ek C'de verilmiştir.

Sonlu cisimlerde çarpma işlemi için yeni bir yaklaşım olan nokta ürün yöntemi, ilgili indirgenemez polinoma göre yukarıdaki teknikle üretilen tablo verilerini kullanmaktadır. Buna göre *n*-bit $a \otimes b$ işleminde tablodan *a* değerinin $1,\alpha,\alpha^2 ... \alpha^{n-1}$ basamak değerleri $x_{n-1} ... x_2 x_1 x_0$ sırasına göre yazılır. Böylelikle *n* adet ifade oluşturulur. *b* değerinin ikili karşılığı bu *n* adet ifade ile ayrı ayrı noktasal çarpma işlemine tabi tutulur. Çıkan ikili sonuç çarpımın sonucunu verir.

Örnek 2.12. GF(2^4)'te $\alpha^4+\alpha^3+1$ indirgenemez polinomu ile oluşturulan sonlu cisimde $A_h \otimes 8_h$ işleminin sonucu nokta ürün ile aşağıdaki gibi hesaplanabilir.

Adım 1: Tablo 5.1'den A_h değerinin verileri alınır.

	α^3	α^2	α	1
A_h	$(x_1 + x_0)$	$(x_3 + x_1)$	$(x_3 + x_2 + x_0)$	$(x_2 + x_1)$

Adım 2: Tablo 5.1'den A_h değerinin $1, \alpha, \alpha^2, \alpha^3$ basamak değerleri $x_3 x_2 x_1 x_0$ sırasına göre yazılır.

	α^3	α^2	α	1
A_h	$(x_1 + x_0)$	$(x_3 + x_1)$	$(x_3 + x_2 + x_0)$	$(x_2 + x_1)$
$x_3 x_2 x_1 x_0$	(0011)	(1010)	(1101)	(0110)

Adım 3: Adım 2'deki değerler ile 8_h ile noktasal çarpımı yapılır.

(0011)•(1000)=0
(1010)•(1000)=1
(1101)•(1000)=1
(0110)•(1000)=0

Böylelikle çarpımın sonucu 6_h (0110) olarak bulunur.

Aynı teknikle 8-bit değerler için tablo hazırlamak mümkündür. Ek D'de AES şifreleme algoritmasında kullanılan $x^8 + x^4 + x^3 + x + 1$ indirgenemez polinomu için oluşturulan nokta ürün tablosu verilmiştir.

Örnek 2.13. GF(2^8)'de $\alpha^8 + \alpha^4 + \alpha^3 + \alpha + 1$ indirgenemez polinomu ile oluşturulan sonlu cisimde $FA_h \otimes 0D_h$ sonucunu nokta ürün yöntemi ile hesaplayalım.

Tablo değerleri aşağıdaki gibidir. Bitlerin sıralaması $x_7x_6x_5x_4x_3x_2x_1x_0$ şeklindedir.

	α^7	α^6	α^5	α^4	α^3	α^2	α	1
FA_h	10011110	10100011	01000110	00010011	10111001	01110011	11100111	11001111

$0D_h$ değerinin ikili karşılığı (00001101) şeklindedir. Bu değerin yukarıdaki değerler ile noktasal çarpımı aşağıdaki gibi olacaktır.

(10011110)•(00001101)=1
(10100011)•(00001101)=0
(01000110)•(00001101)=1
(00010011)•(00001101)=0
(10111001)•(00001101)=1
(01110011)•(00001101)=1
(11100111)•(00001101)=1
(11001111)•(00001101)=0

Çarpımın sonucunda AE_h (10101110) değerine ulaşılır.

2.1.3. Sonlu Cisimde Ters Alma İşlemi

n – bit iki polinomun çarpımının kalanı seçilen indirgenemez polinoma göre 1 ise o zaman iki polinom birbirinin o indirgenemez polinoma göre tersidir denir. İndirgenemez bir polinoma göre ters alma işlemi için iki yöntem önerilebilir. Bu yöntemlerden ilki $GF(2^n)$ için tablo oluşturmaktır. Eğer n değeri küçük bir değer ise bu yöntem etkili olabilir. Büyük bir n değeri için ise ikili Euclidean algoritması kullanılabilir (Lidl ve Niederreiter, 1994). Tez çalışmamız sırasında tablo yöntemi tercih edilecektir.

Örnek 2.14. $GF(2^4)$ için indirgenemez polinom olarak $x^4 + x^3 + x^2 + x + 1$ seçilsin. Bu cismin karakteristiği 2, eleman sayısı 16 ve bu cisimdeki bir üreteç eleman $\beta = (0011) = \alpha + 1$ 'dir. Bu β üreteç elemanının üslerini düşünelim.

$$\beta^{0} = (0010), \beta^{1} = (0110), \beta^{2} = (1010), \beta^{3} = (0001)$$

$$\beta^{4} = (1001), \beta^{5} = (0100), \beta^{6} = (1100), \beta^{7} = (1011)$$

$$\beta^{8} = (1110), \beta^{9} = (1101), \beta^{10} = (1000), \beta^{11} = (0111)$$

$$\beta^{12} = (0001), \beta^{13} = (0011), \beta^{14} = 0101), \beta^{15} = (1111)$$

Dolayısıyla $a \in GF(2^n)$ ve $a = \beta^i$ olmak üzere a'nın çarpmaya göre tersi $a^{-1} = \beta^{(-i) mod (2^n - 1)}$ şeklinde verilebilir. Bunu göz önüne alarak elemanların tersi ve polinomsal yazılışları aşağıdaki gibidir:

β^0	(0001)	1	*Tersi*	β^{15}	(0001)	1
β^1	(0011)	$x+1$	*Tersi*	β^{14}	(1010)	x^3+x
β^2	(0101)	x^2+1	*Tersi*	β^{13}	(0110)	x^2+x
β^3	(1111)	x^3+x^2+x+1	*Tersi*	β^{12}	(0010)	x
β^4	(1110)	x^3+x^2+x	*Tersi*	β^{11}	(1011)	x^3+x+1
β^5	(1101)	x^3+x^2+1	*Tersi*	β^{10}	(1100)	x^3+x^2
β^6	(1000)	x^3	*Tersi*	β^9	(0100)	x^2
β^7	(0111)	x^2+x+1	*Tersi*	β^8	(1001)	x^3+1
β^8	(1001)	x^3+1	*Tersi*	β^7	(0111)	x^2+x+1
β^9	(0100)	x^2	*Tersi*	β^6	(1000)	x^3
β^{10}	(1100)	x^3+x^2	*Tersi*	β^5	(1101)	x^3+x^2+1
β^{11}	(1011)	x^3+x+1	*Tersi*	β^4	(1110)	x^3+x^2+x
β^{12}	(0010)	x	*Tersi*	β^3	(1111)	x^3+x^2+x+1
β^{13}	(0110)	x^2+x	*Tersi*	β^2	(0101)	x^2+1
β^{14}	(1010)	x^3+x	*Tersi*	β^1	(0011)	$x+1$
β^{15}	(0001)	1	*Tersi*	β^0	(0001)	1

Örnek 2.14'te $n = 4$ olduğu için tablo kolay bir şekilde elde edilmiştir. Örneğin $n = 8$ için $GF(2^8)$ sonlu cisminde 0 elemanı ile birlikte 256 adet eleman mevcuttur. Bu cisim için hesaplamalar tablo ile yapılabilir.

BÖLÜM 3

DOĞRUSAL DÖNÜŞÜMLER

Blok şifrelerde kullanılan doğrusal dönüşümler bir yandan rastlantısal (random) görünüşlü permütasyonlar olacak şekilde seçilmeli iken diğer yandan şifreye yapılan saldırılara karşı şifreyi güçlü kılacak şekilde seçilmelidir. Literatürde çeşitli doğrusal dönüşümler bulunmaktadır. Bazı doğrusal dönüşümler cebirsel tabanlı iken bazıları ise rastlantısal görünüşlü olacak şekilde tasarım mekanizmalarına sahiptir. Bir doğrusal dönüşümün şifre içerisinde seçilmesini etkileyen çeşitli kriterler mevcuttur. Bunlar:

- Çığ etkisi (Avalanche) (Feistel, 1973) ve Katı çığ etkisi (Strict Avalanche) (Webster ve Tavares, 1986): Bir bitin değişiminin çıkış bitlerinde değişimin göstergesi,
- Bütünlük (Completeness): Çıkış bitlerinin her birinin giriş bitlerine bağımlılığı (Kam ve Davida, 1979),
- Dallanma sayısı (Branch number): Ardışık iki döngüdeki minimum S-kutusu sayısı (Daemen ve Rijmen, 2002),
- Sabit noktalar (Fixed points): Doğrusal bir dönüşümün çıkış bitlerinin giriş bitleri ile aynı olanlarının sayısı (Z'aba, 2010)

şeklinde verilebilir. Özellikle bazı önemli şifrelerde cebirsel tabanlı doğrusal dönüşümler kullanılmıştır. Bu şifrelere örnek olarak AES ve Khazad (Barreto ve Rijmen, 2000) verilebilir. Bunun yanında bazı şifrelerde ise rastlantısal görünüşlü doğrusal permütasyonlar kullanılmıştır. Bu şifrelere örnek olarak Serpent (Biham vd., 1998) ve Present (Bogdanov vd., 2007) verilebilir. Diğer yandan Tablo 3.1'de görüldüğü gibi AES ve Khazad şifreleri GF(2^8) üzerine sırasıyla 4×4 byte MDS matris ve 8×8 byte involutif (tersi kendisi) MDS matrisi doğrusal dönüşüm olarak kullanırken, Camellia (Aoki ve Ichikawa, 2001) ve ARIA (Kwon ve Kim, 2004) şifreleri ise

sırasıyla GF(2^8) üzerine 8×8 ikili matris ve 16×16 involutif ikili matrisleri doğrusal dönüşüm olarak şifre içerisinde kullanmaktadırlar. Buna ek olarak bu matrisler maksimum dallanma sayısı değerine sahiptirler ve MDBL kodlardır.

Tablo 3.1. İncelenen doğrusal dönüşümlerin özellikleri

Blok Şifre	**Yayılım Katmanı**
AES	$GF\left(2^8\right)$ üzerinde 4×4 MDS matris
Khazad	$GF\left(2^8\right)$ üzerinde 8×8 involutif MDS matris
Camellia	$GF\left(2^8\right)$ üzerinde 8×8 ikili matris
ARIA	$GF\left(2^8\right)$ üzerinde 16×16 involutif ikili matris

Tez çalışması sırasında Tablo 3.1'deki şifreleme algoritmalarının doğrusal dönüşümleri Bölüm 3.1'de verilen matematiksel altyapıya göre ayrıntılı olarak incelenmiştir.

3.1. Doğrusal Dönüşümler için Matematiksel Alt Yapı

Bu bölümde MDS kodların bazı önemli özellikleri verilecektir. Yayılım elemanları doğrusal dönüşümlerdir ve matrisler ile temsil edildiklerinden bir doğrusal dönüşüm $A:(\{0,1\}^m)^n \rightarrow (\{0,1\}^m)^n$ (3.1) denklemindeki gibi tanımlanabilir.

$$A(x) = A.x^T = \begin{pmatrix} a_{11} & a_{12} & . & . & . & a_{1n} \\ a_{21} & a_{22} & . & . & . & a_{2n} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ a_{n1} & a_{n2} & . & . & . & a_{nn} \end{pmatrix} . \begin{pmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_n \end{pmatrix} \qquad (3.1)$$

Denklem (3.1)'de $x=(x_1,x_2,...,x_n)^T$, $x_i \in \{0,1\}^m$, $i=1,...,n$ olmak üzere n bir yayılım elemanına giriş olan m -bit giriş ve çıkışlı S-kutularını temsil etmektedir (Kwon vd., 2005). Buna ek olarak matris elemanları GF(2^8) ya da GF(2) nin elemanları olabilir.

Tanım 3.1. $n \times n$ boyutunda bir A matrisinin dallanma sayısı ifade (3.2)'deki gibi tanımlanabilir.

$$\beta(A) = \min\{wt(x) + wt(A.x^T) \mid x \in (\{0,1\}^m)^n, x \neq 0\} \qquad (3.2)$$

Bir yayılım elemanının dallanma sayısı o elemanın en kötü durumu için bir ölçü verir. Bu ölçü de takip eden iki döngüde bir doğrusal yaklaşım ya da fark yaklaşım karakteristiğinde bulunan aktif S-kutularının en düşük sınırıdır.

Bir kod kelimesinin Hamming ağırlığı wt(c) olarak temsil edilebilir ve c kod kelimesinin 0 olmayan elamanlarının sayısı olarak tanımlanır. Buna ek olarak $(GF(2^m))^n$ vektör uzayından n boyutlu iki vektör arasındaki Hamming uzaklığı da vektörlerin farklılaştığı pozisyon sayısı olarak tanımlanır (Nakahara ve Abrahão, 2009).

Tanım 3.2. Bir $GF(2^m)$ üzerine bir $[n, k, d]$ kod, vektör uzayı $(GF(2^m))^n$ 'in k boyutlu bir alt uzayıdır ve n elemanlı iki vektör arasındaki Hamming uzaklığı minimum d dir. Bu özellik ile d en büyük değerdir. Doğrusal bir $[n, k, d]$ kod c için bir G üreteç matris satırları c için bir taban oluşturan $k \times n$ boyutunda bir matristir. Doğrusal $[n, k, d]$ kodlar Singleton sınırı olan $d \leq n - k + 1$ eşitsizliğini sağlar (Nakahara ve Abrahão, 2009).

Yardımcı Önerme 3.1. Bir doğrusal $[n, k, d]$ kod Singleton sınırı olan $d \leq n - k + 1$ eşitsizliğini sağlıyor ise MDS koddur. Alternatif olarak, bir matrisin MDS matris olabilmesi için satır ve sütunlarından oluşturulan tüm alt matrislerinin determinantının 0'dan farklı olması gerekir (Nakahara ve Abrahão, 2009).

Literatürde MDS matris tasarımı için çeşitli yöntemler ortaya atılmıştır. Özellikle büyük MDS matrislerin tasarımında tasarımcılar aşağıda verilen yaklaşımları benimsemişlerdir:

1- MDS matrislerin Cauchy matrisleri ile tasarımı (Youssef vd., 1997),
2- MDS matrislerin Vandermonde matrisleri ile tasarımı (Lacan ve Fimes, 2004).

Özellikle Cauchy matris yöntemi ile tasarlanan 16×16 boyutunda ve elemanları $GF(2^8)$ den olan ve dallanma sayısı 17 olan involutif bir MDS matris tasarımları bulunmaktadır (Nakahara ve Abrahão, 2009).

Tanım 3.3. $x_0, x_1, ..., x_{n-1}$ ve $y_0, y_1, ..., y_{n-1}$ sonlu cisim $GF(2^n)$ de birbirinden farklı değerler olsun. O zaman $p_{i,j} = \frac{1}{x_i + y_j}$ ise $P = [p_{i,j}]$ matrisi bir MDS matristir (MacWilliams ve Sloane, 1977) (Youssef vd., 1997).

Eğer x_i'ler ve y_j'ler farklı değerlere sahipse, tüm *i, j* değerleri için $x_i + y_j \neq 0$ ifadesini sağlar. Bu da bir Cauchy matrisinin herhangi bir kare alt matrisinin herhangi bir cisim üzerine determinantının sıfırdan farklı olduğunu ifade eder. Yardımcı önerme (3.1) gereği de $P = [p_{i,j}]$ matrisi bir MDS matris olur. Diğer yandan Δ bazı özelliklere sahip bir fark olmak üzere $y_i = x_i + \Delta$ olarak seçilirse $P = [p_{i,j}]$ matrisi sonlu cisimde bir Hadamard MDS matris olur ve $P^2 = c^2 I, \mathrm{c} = \sum_{i=0}^{2^n - 1} p_{0,i}$ olacağından $P' = \frac{P}{c}$ matrisi involutif MDS matris olur.

Örnek 3.1. $GF(2^8)$ cismi $x^8 + x^4 + x^3 + x + 1$ ile tanımlansın. $x_0 = 02_h,\ x_1 = 03_h,\ x_2 = 04_h,\ x_3 = 05_h$ giriş değerleri ve $\Delta = 10_h$ için Cauchy matrisi oluşturulmak istendiğinde öncelikle giriş bitleri Δ ile sonlu cisimde toplanarak aşağıdaki gibi yazılır.

$$y_0 = 02_h \oplus \Delta = 12_h$$
$$y_1 = 03_h \oplus \Delta = 13_h$$
$$y_2 = 04_h \oplus \Delta = 14_h$$
$$y_3 = 05_h \oplus \Delta = 15_h$$

4×4'lük matrisin elemanları $p_{i,j} = \frac{1}{x_i + y_j}$ ifadesi kullanılarak *P* matrisi aşağıdaki gibi oluşturulur.

$$P=\begin{bmatrix} \frac{1}{02_h \oplus 12_h} & \frac{1}{02_h \oplus 13_h} & \frac{1}{02_h \oplus 14_h} & \frac{1}{02_h \oplus 15_h} \\ \frac{1}{03_h \oplus 12_h} & \frac{1}{03_h \oplus 13_h} & \frac{1}{03_h \oplus 14_h} & \frac{1}{03_h \oplus 15_h} \\ \frac{1}{04_h \oplus 12_h} & \frac{1}{04_h \oplus 13_h} & \frac{1}{04_h \oplus 14_h} & \frac{1}{04_h \oplus 15_h} \\ \frac{1}{05_h \oplus 12_h} & \frac{1}{05_h \oplus 13_h} & \frac{1}{05_h \oplus 14_h} & \frac{1}{05_h \oplus 15_h} \end{bmatrix}=\begin{bmatrix} \frac{1}{10_h} & \frac{1}{11_h} & \frac{1}{16_h} & \frac{1}{17_h} \\ \frac{1}{11_h} & \frac{1}{10_h} & \frac{1}{17_h} & \frac{1}{16_h} \\ \frac{1}{16_h} & \frac{1}{17_h} & \frac{1}{10_h} & \frac{1}{11_h} \\ \frac{1}{17_h} & \frac{1}{16_h} & \frac{1}{11_h} & \frac{1}{10_h} \end{bmatrix}$$

$$P=\begin{bmatrix} 74_h & 64_h & 60_h & 5F_h \\ 64_h & 74_h & 5F_h & 60_h \\ 60_h & 5F_h & 74_h & 64_h \\ 5F_h & 60_h & 64_h & 74_h \end{bmatrix}$$

Üretilen P matrisi hadamard formda MDS bir matristir. P matrisinin involutif olabilmesi için Tanım 3.3'teki düzenlemeler yapılmalıdır. Buna göre $c = 74_h \oplus 64_h \oplus 60_h \oplus 5F_h = 2F_h$ olarak hesaplanır. $P' = \frac{P}{c}$ ifadesinden involutif ve MDS olan bir matris aşağıdaki şekilde üretilir.

$$P'=\begin{bmatrix} \frac{74_h}{2F_h} & \frac{64_h}{2F_h} & \frac{60_h}{2F_h} & \frac{5F_h}{2F_h} \\ \frac{64_h}{2F_h} & \frac{74_h}{2F_h} & \frac{5F_h}{2F_h} & \frac{60_h}{2F_h} \\ \frac{60_h}{2F_h} & \frac{5F_h}{2F_h} & \frac{74_h}{2F_h} & \frac{64_h}{2F_h} \\ \frac{5F_h}{2F_h} & \frac{60_h}{2F_h} & \frac{64_h}{2F_h} & \frac{74_h}{2F_h} \end{bmatrix}=\begin{bmatrix} E4_h & 70_h & 55_h & C0_h \\ 70_h & E4_h & C0_h & 55_h \\ 55_h & C0_h & E4_h & 70_h \\ C0_h & 55_h & 70_h & E4_h \end{bmatrix}.$$

Üretilen P', bir Cauchy matrisi olmakla beraber involutif ve MDS'tir.

Yardımcı Önerme (3.1) gereği, bir $n \times n$ matrisin MDS matris olduğunu doğrulamak için bu matrisin alt kare matrislerinin determinantlarının sıfırdan farklı olduğu test edilmelidir. Bu test edilecek alt matrislerin determinantlarının sayısı (3.3) denkleminden elde edilebilir (Aslan ve Sakallı, 2012).

$$\sum_{k=1}^{n-2}\left[C\binom{n}{n-k}\right]^2 \qquad (3.3)$$

(3.3)' teki denklemden, örneğin 4×4 boyutunda bir matrisin MDS matris olduğunu doğrulamak için;

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

adet alt matris determinantının incelenmesi gerekir.

16 tane 3×3 boyutunda ve 36 tane 2×2 boyutunda alt matrisin determinantının sıfırdan farklı olduğu test edilmelidir. Benzer şekilde 8×8 boyutunda bir matrisin MDS matris olduğunu doğrulamak için;

$$\left(C\binom{8}{7}\right)^2 + \left(C\binom{8}{6}\right)^2 + \ldots + \left(C\binom{8}{2}\right)^2 = 12804$$

alt matrisin determinantının sıfırdan farklı olduğu test edilmelidir.

Literatürde iki tür matris kullanılmaktadır. Bunlardan ilki dairesel (circulant) matrisler ve ikincisi Hadamard matrislerdir (Junod ve Vaudenay, 2004). Bu bahsedilen yaklaşımlardan ilki AES doğrusal dönüşümünde kullanılmıştır ve bu doğrusal dönüşüm her satırın sağa bir byte dairesel olarak ötelenmesi ile elde edilen bir dairesel matristir. İkinci yaklaşıma uygun MDS matris yani Hadamard matris kullanımı Khazad şifresinde yer almaktadır ve involutif bir doğrusal dönüşüm ile şifrede şifreleme ve deşifreleme performansının aynı olması amaçlanmıştır.

Tanım 3.4. $A = circ(a_1, a_2, ..., a_n)$ notasyonu A matrisinin dairesel ve her satırının sağa 1 pozisyon hareket ettirilerek elde edildiğini ifade eder. Dolayısıyla dairesel formdaki $n \times n$ boyutlu bir A matrisi ifade (3.4)'te verilmiştir.

$$A = circ(a_1, a_2, ..., a_n) = \begin{bmatrix} a_1 & a_2 & \ldots & a_n \\ a_n & a_1 & \ldots & a_{n-1} \\ . & . & \ldots & . \\ a_2 & a_3 & \ldots & a_1 \end{bmatrix}_{n \times n} \qquad (3.4)$$

Aşağıda verilen önerme çok iyi bilinen bir önermedir ve kolaylık sağlaması için sunulmuştur.

Yardımcı Önerme 3.2. $a_1,a_2,...,a_t \in GF(2^n)$ olsun. O zaman ifadede, ifade (3.5)'te verildiği gibi bir eşitlik söz konusu olur.

$$(a_1 + a_2 + ... + a_t)^{2^k} = a_1^{2^k} + a_2^{2^k} + ... + a_t^{2^k} \tag{3.5}$$

Tanım 3.5. $A = Had(a_1, a_2, ..., a_n)$ notasyonu A matrisinin bir Hadamard matris olduğunu ifade etmektedir. Buna göre Hadamard matris formu ifade (3.6)'da verilmiştir.

$$A = Had(a_1, a_2, a_3, a_4) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix} \tag{3.6}$$

Yardımcı Önerme 3.3. $A = Had(a_1, a_2, a_3, a_4) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}$ matrisi elemanları $GF(2^n)$ cismine ait 4×4 bir matris olsun. Buna ek olarak elemanları birbirinden farklı ve 0'dan farklı olsun. O zaman A matrisi, sadece ve sadece $\sum_{i=1}^{4} a_i = 1$ (indisler arası toplama işlemi mod 2 toplama veya XOR işlemi anlamına gelmektedir) ve her kare alt matrislerinin determinantı 0'dan farklı ise involutif MDS matristir.

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

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

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

Diğer yandan Yardımcı Önerme 2 kullanılarak $\sum_1^4 {a_i}^2 = \sum_1^4 a_i = 1$ şeklinde elde edilebilir. Dolayısıyla $A^{-1} = A$ ve $A = A^T$ şeklinde elde edilebileceğinden A matrisi involutif ve MDS matristir.

Örnek 3.2. Elemanları $GF(2^4)$ cismine ait bir involutif MDS matris aşağıdaki gibi verilebilir. $GF(2^4)$ cismi indirgenemez polinom $x^4 + x + 1$ kullanılarak tanımlanmıştır.

$$Had(1_h, 2_h, 4_h, 6_h) = \begin{bmatrix} 1_h & 2_h & 4_h & 6_h \\ 2_h & 1_h & 6_h & 4_h \\ 4_h & 6_h & 1_h & 2_h \\ 6_h & 4_h & 2_h & 1_h \end{bmatrix}.$$

Yukarıdaki matrisin (Barreto ve Rijmen, 2000) MDS matris olduğunu göstermek için 4×4 bir matris için incelenmesi gereken 52 kare alt matris determinantının 0'dan farklı olduğu bir yazılım aracılığı ile elde edilmiştir.

Matrisin MDS olup olmadığını belirlemek amacıyla geliştirilen yazılım matrisin alt determinantlarını incelemektedir. Bir matrisin alt determinant sayısı denklem (3.3) ile bulunabilir.

Örnek (3.2)'de verilen matris 4×4 boyutunda olduğundan geliştirilen yazılım ile 3×3 boyutunda 16 ve 2×2 boyutunda 36 adet matrisin determinantları incelenmiş ve tüm hesaplanan determinantlar 0'dan farklı olduğu için matrisin MDS matris olduğu

gösterilmiştir. $GF(2^8)$'de $x^8+x^4+x^3+x+1$ indirgenemez polinomu ile tanımlı $Had(1_h,2_h,4_h,6_h)$ matrisinin alt determinant hesapları Ek E'de verilmiştir.

3.2. Doğrusal Dönüşümlerde Sabit Noktalar

Bir giriş bloğu, bu bloğa bir doğrusal dönüşüm uygulandıktan sonra elde edilen çıkış bloğu ile aynı ise o zaman bu giriş bloğuna o doğrusal dönüşümün sabit noktasıdır denir. Açıkça, sabit noktalarda giriş bloğu doğrusal dönüşüm tarafından değiştirilmeden bırakıldığı için yayılım olmamaktadır. Bundan dolayı, bir doğrusal dönüşümdeki sabit nokta sayısı bir rastsal doğrusal dönüşümden beklenen sayıyı aşarsa, bu doğrusal dönüşümün kötü yayılım sağladığının göstergesidir. Rastsal bir doğrusal dönüşümden beklenen sabit nokta sayısı ise 1'dir (Z'aba, 2010).

Diğer yandan blok şifrelerin döngü fonksiyonunda var olan sabit noktaların blok şifrelere karşı birçok saldırının temelini oluşturduğu ve bu saldırıların bir döngü veya birden fazla döngü boyunca var olan sabit noktaları kullandığı belirtilmiştir (Z'aba, 2010).

Bir doğrusal dönüşüme bir giriş bloğunun $GF(2^m)$ den m-bit değerlerden oluştuğunu varsayalım. Buna ek olarak doğrusal dönüşüm matrisinin $n \times n$ boyutunda ve I matrisinin $n \times n$ boyutunda birim matris olduğunu varsayalım. O zaman doğrusal dönüşüm matrisi A (determinantı 0'dan farklı) için tüm sabit noktaların sayısı ifade (3.7)'deki denklemin çözülmesi ile elde edilebilir:

$$(A-I)x^T = 0 \qquad (3.7)$$

(3.7) ifadesindeki 0, n uzunluğunda tüm elemanları 0 olan vektörü temsil etmektedir. (3.7) ifadesinden A doğrusal dönüşümündeki sabit noktaların sayısı (3.8) ifadesindeki gibi verilebilir:

$$F_A = 2^{m(rank(A)-rank(A-I)} = 2^{m(n-rank(A-I))} \qquad (3.8)$$

(3.8) ifadesinden anlaşılacağı gibi $(A-I)$ matrisinin daha büyük bir rank değerine sahip olması A doğrusal dönüşümünün daha az sayıda sabit noktaya sahip olacağının göstergesidir

3.3. AES Şifresinde Kullanılan Doğrusal Dönüşümün İncelenmesi

AES şifreleme algoritmasında kullanılan doğrusal dönüşüm MixColumns (Sütunları karıştırma) olarak adlandırılır. Sütunları karıştırma doğrusal dönüşümü 4×4 byte matris şeklindedir. AES şifresi 32-bit'ten 32-bite dönüşüm yapan bir doğrusal dönüşüm içerir. Bu dönüşüm doğrusal ve diferansiyel kriptanalizi zorlaştırıcı etki yapma amacındadır ve sonlu cisimde çarpma tabanlıdır. $GF(2^8)$ de elemanlar içeren AES matrisi çarpma işlemleri sonucunda indirgeme işlemleri için $x^8+x^4+x^3+x+1$ polinomunu kullanmaktadır. MDS matris tabanlı olan AES doğrusal dönüşümü, şifreleme ve deşifreleme işlemi için aşağıdaki matrisleri kullanmaktadır.

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

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

AES'in dairesel formda tasarlanan MDS matrisinin dallanma sayısı 5'tir. Dolayısı ile giriş değerlerinden 1 byte'ın değişmesi sonucunda minimum 4 byte bu durumdan etkilenecektir. Kullanılan doğrusal dönüşüm matrisi involutif değildir. Bu sebeple deşifreleme işlemi için şifreleme doğrusal dönüşüm matrisinin tersi kullanılmaktadır. Sütunları karıştırma dönüşümünün $(A-I)$ matrisinin rank değeri 3 olduğundan $2^{b(rank(A)-rank(A-I)}=2^{8(4-3)}=2^8$ adet sabit nokta içerir.

AES şifreleme algoritmasının doğrusal dönüşümünün elemanları, yazılım ve donanım performansının yüksek olması açısından 01_h, 02_h, 03_h gibi GF(2^8)'in küçük

elemanlarından seçilmiştir. Bölüm 2'de anlatıldığı gibi 02_h ile çarpma işlemine xtime veya α ile çarpma denilmektedir. Aynı şekilde 03_h veya $\alpha+1$ ile çarpma, işlemlerin hızlanması için tablo okuma ile yapılabileceği gibi bir xtime ve bir XOR işlemi ile de yapılabilir. Bu durumda 4×4 byte matris olan AES doğrusal dönüşümünün her satırı için 2 xtime ve 4 XOR işlemi gerekmektedir. Dolayısı ile bir doğrusal dönüşüm işlemi toplamda 8 xtime ve 16 bit XOR işleminden oluşur. AES şifreleme algoritmasında bir döngüde 4 doğrusal dönüşüm kullanıldığından toplamda bir döngü sadece doğrusal dönüşüm için 32 xtime ve 64 XOR işlem yapar. Fakat 5 adet değişken tanımlanarak bir döngü için doğrusal dönüşüm işlem yükü 16 xtime ve 60 bit XOR işlemine indirgenebilir. Aşağıda verilen pseudo kodu AES doğrusal dönüşüm matrisinin 5 adet değişken kullanılarak optimize edilmiş halidir (Nakahara ve Abrahão, 2009).

```
/* w doğrusal dönüşüme giriş vektörü */
X0 = xtime(w[0]);
X1 = xtime(w[1]);
X2 = xtime(w[2]);
X3 = xtime(w[3]);
u = w[0] ^ w[1] ^ w[2] ^ w[3];
w[0] = u ^ w[0] ^ X1 ^ X0;
w[1] = u ^ w[1] ^ X1 ^ X2;
w[2] = u ^ w[2] ^ X2 ^ X3;
w[3] = u ^ w[3] ^ X0 ^ X3;
```

Öte yandan, AES şifreleme algoritmasında kullanılan doğrusal dönüşümün detaylı olarak incelenebilmesi için bu dönüşümün bit temelli modeli ele alınabilir (Yavuzer Aslan vd., 2010). Bu model ile AES doğrusal dönüşümünün dallanma sayısı ve performansı rahatlıkla görülebilir. Bit temelli model ortaya çıkarılırken kolaylık olması amacıyla çarpma işlemlerinin GF(2^8) yerine GF(2^4)'te olduğunu farz ederek 32-bit'ten 32-bit'e olan dönüşüm yerine 16-bit'ten 16-bit'e olacak şekilde bir doğrusal dönüşüm kullanılacaktır. Buna göre Örnek 3.2'de bu işlemler ayrıntılı olarak anlatılmıştır.

Örnek 3.3. 16-bit girişe karşılık 16-bit çıkışa sahip sütunları karıştırma işlemi 16-bit giriş $w = (w_0 w_1 w_2 w_3)$ ve her w_i 4-bit değer olmak üzere;

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \otimes \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} w_0' \\ w_1' \\ w_2' \\ w_3' \end{bmatrix}$$

şeklinde verilebilir. Buna ek olarak GF(2^4)'te çarpma işleminde indirgenemez polinom $x^4 + x + 1$ kullanılmaktadır. Bit temelli modeli elde etmek için $w_0 = (x_3 x_2 x_1 x_0)$, $w_1 = (x_7 x_6 x_5 x_4)$, $w_2 = (x_{11} x_{10} x_9 x_8)$, $w_3 = (x_{15} x_{14} x_{13} x_{12})$ 4-bit değerler şeklinde ele alınırsa bu değerler Bölüm 2'deki matematiksel kurallar kullanılarak aşağıdaki gibi ifade edilebilir:

$$\begin{aligned} w_0 &= x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0, \\ w_1 &= x_7\alpha^3 + x_6\alpha^2 + x_5\alpha + x_4, \\ w_2 &= x_{11}\alpha^3 + x_{10}\alpha^2 + x_9\alpha + x_8, \\ w_3 &= x_{15}\alpha^3 + x_{14}\alpha^2 + x_{13}\alpha + x_{12}. \end{aligned}$$

Dolayısıyla $w_0' = (f_3 f_2 f_1 f_0)$ ilk 4-bit çıkış bitleri aşağıdaki gibi elde edilebilir:

$$\begin{aligned} w_0' = \alpha\,&(x_3\alpha^3 + x_2\alpha^2 + x_1\alpha + x_0) \\ &+ \alpha + 1\,(x_7\alpha^3 + x_6\alpha^2 + x_5\alpha + x_4) \\ &+ (x_{11}\alpha^3 + x_{10}\alpha^2 + x_9\alpha + x_8) \\ &+ (x_{15}\alpha^3 + x_{14}\alpha^2 + x_{13}\alpha + x_{12}). \end{aligned}$$

Yukarıdaki ifade üzerinde, verilen indirgenemez polinom ile indirgeme gerçekleştirildikten sonra aşağıdaki gibi çıkış bitleri, giriş bitleri ve XOR işlemlerinin bir kombinasyonu olarak yazılabilir:

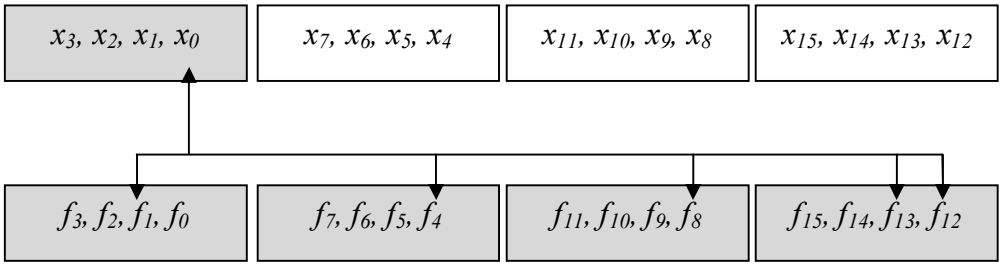
$$\begin{aligned} f_0 &= x_3 \oplus x_4 \oplus x_7 \oplus x_8 \oplus x_{12}, \\ f_1 &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_9 \oplus x_{13}, \\ f_2 &= x_1 \oplus x_5 \oplus x_6 \oplus x_{10} \oplus x_{14}, \\ f_3 &= x_2 \oplus x_6 \oplus x_7 \oplus x_{11} \oplus x_{15}. \end{aligned}$$

Bu işlem tüm 16-bit çıkış (w_0', w_1', w_2', w_3') değerleri için tekrarlanır ise bit temelli model aşağıdaki gibi yazılabilir.

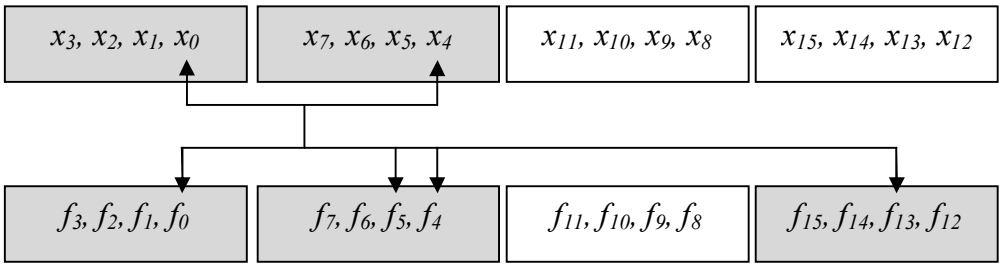
$$
\begin{aligned}
f_0 &= x_3 \oplus x_4 \oplus x_7 \oplus x_8 \oplus x_{12}, \\
f_1 &= x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_9 \oplus x_{13}, \\
f_2 &= x_1 \oplus x_5 \oplus x_6 \oplus x_{10} \oplus x_{14}, \\
f_3 &= x_2 \oplus x_6 \oplus x_7 \oplus x_{11} \oplus x_{15}, \\
f_4 &= x_0 \oplus x_7 \oplus x_8 \oplus x_{11} \oplus x_{12}, \\
f_5 &= x_1 \oplus x_4 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{11} \oplus x_{13}, \\
f_6 &= x_2 \oplus x_5 \oplus x_9 \oplus x_{10} \oplus x_{14}, \\
f_7 &= x_3 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{15}, \\
f_8 &= x_0 \oplus x_4 \oplus x_{11} \oplus x_{12} \oplus x_{15}, \\
f_9 &= x_1 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13} \oplus x_{15}, \\
f_{10} &= x_2 \oplus x_6 \oplus x_9 \oplus x_{13} \oplus x_{14}, \\
f_{11} &= x_3 \oplus x_7 \oplus x_{10} \oplus x_{14} \oplus x_{15}, \\
f_{12} &= x_0 \oplus x_3 \oplus x_4 \oplus x_8 \oplus x_{15}, \\
f_{13} &= x_0 \oplus x_1 \oplus x_3 \oplus x_5 \oplus x_9 \oplus x_{12} \oplus x_{15}, \\
f_{14} &= x_1 \oplus x_2 \oplus x_6 \oplus x_{10} \oplus x_{13}, \\
f_{15} &= x_2 \oplus x_3 \oplus x_7 \oplus x_{11} \oplus x_{14}.
\end{aligned}
$$

Yukarıdaki modelden de gözleneceği gibi sütunları karıştırma işlemi sonlu cisimde çarpma işlemi yapmadan sadece XOR işlemleri ile gerçeklenebilecek bir forma sokulmuştur. 72 bit XOR işlemi ile bu dönüşüm gerçekleştirilebilir.

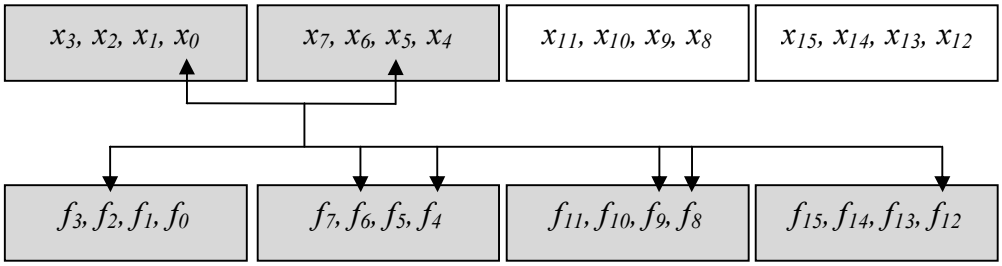
Örnek 3.3'ten elde edilen sonuçlara göre x_0 bitinin değerinin değişmesi f_1, f_4, f_8, f_{12} ve f_{13} çıkış bitlerinin etkileneceği ya da değişeceği anlamına gelir. Şekil 3.1'de x_0 bitinin değişimi ile elde edilecek dallanma sayısı 5 olarak gözlenebilir. Buna ek olarak Şekil 3.2 ve Şekil 3.3'te verilen giriş bitlerindeki değişikliklerin hangi 4'lü değerleri etkilediği gösterilmiştir. Bu da verilen giriş bitlerindeki değişliklere göre dallanma sayılarının sırasıyla 5 ve 6 olduğunu göstermektedir. Ancak Bölüm 3.1'de verilen dallanma sayısı tanımdan anlaşılacağı gibi minimum değer olan 5 değeri dönüşümün dallanma sayısını ifade eder.



Şekil 3.1. Sütunları karıştırma dönüşümü için x_0 bitinin değişimi ile elde edilen dallanma (Yavuzer Aslan vd., 2010)



Şekil 3.2. Sütunları karıştırma dönüşümü için x_0 ve x_4 bitinin değişimi ile elde edilen dallanma (Yavuzer Aslan vd., 2010)



Şekil 3.3. Sütunları karıştırma dönüşümü için x_0 ve x_5 bitinin değişimi ile elde edilen dallanma (Yavuzer Aslan vd., 2010)

Tez sırasında AES şifresinin deşifreleme aşamasında kullanılan doğrusal dönüşüm matrisinin $x^4 + x + 1$ polinomu ile tanımlı $GF(2^4)$ cismi için bit temelli modeli yine aynı yöntemler kullanılarak çıkarılmıştır. Bu model aşağıdaki gibidir.

$$
\begin{aligned}
f_0 &= x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13}, \\
f_1 &= x_0 \oplus x_1 \oplus x_4 \oplus x_6 \oplus x_7 \oplus x_{11} \oplus x_{14}, \\
f_2 &= x_0 \oplus x_1 \oplus x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_{15}, \\
f_3 &= x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{12}, \\
f_4 &= x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{11} \oplus x_{12} \oplus x_{13} \oplus x_{14}, \\
f_5 &= x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{11} \oplus x_{15}, \\
f_6 &= x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_9 \oplus x_{11} \oplus x_{12}, \\
f_7 &= x_0 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{12} \oplus x_{13}, \\
f_8 &= x_0 \oplus x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{13} \oplus x_{15}, \\
f_9 &= x_3 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{14} \oplus x_{15}, \\
f_{10} &= x_0 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{10} \oplus x_{13} \oplus x_{15}, \\
f_{11} &= x_0 \oplus x_1 \oplus x_4 \oplus x_8 \oplus x_9 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{14}, \\
f_{12} &= x_0 \oplus x_1 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14} \oplus x_{15}, \\
f_{13} &= x_0 \oplus x_2 \oplus x_3 \oplus x_7 \oplus x_{10} \oplus x_{12} \oplus x_{13}, \\
f_{14} &= x_1 \oplus x_3 \oplus x_4 \oplus x_{11} \oplus x_{12} \oplus x_{13} \oplus x_{14}, \\
f_{15} &= x_0 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{12} \oplus x_{13} \oplus x_{14} \oplus x_{15}.
\end{aligned}
$$

Öte yandan, çeşitli çalışmalar (Li ve Friggstad, 2005), (Zhang ve Parhi, 2004) sütunları karıştırma işleminin yazılım ve donanımsal performansının iyileştirilmesi üzerine yapılmıştır. Bu çalışmalardan (Venkaiah vd., 2006)'da tersi kendisi olan bir sabit matris AES şifresinde kullanılan sabit matrisin yerine öne sürülmüştür. Bu matris;

$$
\begin{bmatrix}
02 & 01 & 03 & 01 \\
01 & 02 & 01 & 03 \\
03 & 01 & 02 & 01 \\
01 & 03 & 01 & 02
\end{bmatrix}
$$

şeklindedir. Bu yayılım elemanının dallanma sayısının 4 olduğu verilen yöntemler kullanılarak kolayca test edilebilir. Örneğin verilen matrisin 2×2 boyutunda bir alt matrisi ve onun determinantı $\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} = 0$ şeklinde verileceğinden bu matris bir MDS matris değildir.

3.4. Khazad Şifresinde Kullanılan Doğrusal Dönüşüm

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

KHAZAD şifresinin doğrusal dönüşüm katmanı $GF(2^8)$ de MDS olarak Hadamard formunda tasarlanmıştır. Bu dönüşüm aşağıda verilmiştir:

$$A = \begin{bmatrix} 01_h & 03_h & 04_h & 05_h & 06_h & 08_h & 0B_h & 07_h \\ 03_h & 01_h & 05_h & 04_h & 08_h & 06_h & 07_h & 0B_h \\ 04_h & 05_h & 01_h & 03_h & 0B_h & 07_h & 06_h & 08_h \\ 05_h & 04_h & 03_h & 01_h & 07_h & 0B_h & 08_h & 06_h \\ 06_h & 08_h & 0B_h & 07_h & 01_h & 03_h & 04_h & 05_h \\ 08_h & 06_h & 07_h & 0B_h & 03_h & 01_h & 05_h & 04_h \\ 0B_h & 07_h & 06_h & 08_h & 04_h & 05_h & 01_h & 03_h \\ 07_h & 0B_h & 08_h & 06_h & 05_h & 04_h & 03_h & 01_h \end{bmatrix}$$

KHAZAD için tasarlanan bu doğrusal dönüşüm matrisinin dallanma sayısı 9 olmakla beraber involutif olarak tasarlanmıştır. Bu sebeple şifreleme ve deşifreleme işlemlerinde aynı matris kullanılmaktadır. Böylelikle bu işlemler arasındaki fark kaldırılmış olur. Bu matrisin, $(A-I)$ matrisinin rank değeri 4 olduğundan sabit nokta sayısı 2^{32} olarak aşağıda verildiği gibi hesaplanabilir.

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

3.5. Camellia Şifresinde Kullanılan Doğrusal Dönüşüm

Camellia, 128-bit veri bloklarını 128, 192 veya 256-bit anahtar seçenekleri ile şifreleyen bir şifreleme algoritmasıdır. 2000 yılında NTT (Nippon Telegraph and Telephone Corporation) ve Mitsubishi Electric Corporation tarafından ortak geliştirilmiştir. Camellia şifresinde kullanılan doğrusal dönüşüm matrisi MDBL kod olarak tasarlanmıştır. Dallanma sayısı 5 olan dönüşüm aşağıda gösterilmiştir.

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

Yukarıda verilen Camellia doğrusal dönüşümünün $(A-I)$ matrisinin rank değeri 7 olduğundan $2^{8(8-7)} = 2^8$ adet sabit nokta içerir.

3.6. ARIA Şifresinde Kullanılan Doğrusal Dönüşüm

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

ARIA şifresinin yayılım katmanında 16x16' lık involutif ve MDBL olarak tasarlanmış bir ikili matris kullanılmaktadır. Dallanma sayısı 8 olan bu matris aşağıda verilmiştir.

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

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

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

ARIA matrisinde kullanılan bu involutif ikili matrisin, $(A - I)$ matrisinin rank değeri 7 olduğundan bu dönüşüm $2^{8(16-7)} = 2^{72}$ adet sabit noktaya içerir.

Bu bölümde, literatürde bulunan önemli blok şifrelerin kullandığı doğrusal dönüşümler incelenmiştir. Bu şifrelerde kullanılan doğrusal dönüşümler MDS ve MDBL kod tabanlıdır. İncelenen doğrusal dönüşümlerden Camellia ve ARIA şifresinde optimal dallanma sayısına sahip ikili matrisler kullanılmıştır. İkili matris kullanılması bu dönüşümlerin uygulamada sadece XOR işlemi tabanlı olarak gerçekleştirilmesini sağlamaktadır. Diğer yandan bu şifrelerin tasarımında kullanılan dönüşümlerin rastsal bir permütasyonda olması gereken sabit nokta sayısı 1 değerinden çok daha fazla sayıda sabit nokta içerdiği gözlenmiştir. İncelenen şifreleme algoritmalarında kullanılan doğrusal dönüşümlerin özet bilgileri Tablo 3.2' deki gibidir.

Tablo 3.2. Önemli blok şifrelerde kullanılan doğrusal dönüşümlerin özellikleri (Yavuzer Aslan vd., 2012)

	Özellik	**BN[1]**	**FPN[2]**
AES	$GF(2^8)$ üzerine 4×4 boyutunda MDS matris	5	2^8
KHAZAD	$GF(2^8)$ üzerine 8×8 boyutunda involutif MDS matris	9	2^{32}
CAMELLIA	$GF(2^8)$ üzerine 8×8 boyutunda MDBL matris	5	2^8
ARIA	$GF(2^8)$ üzerine 16×16 boyutunda involutif MDBL matris	8	2^{72}

[1] BN-Dallanma sayısı değeri
[2] FPN-Sabit nokta sayısı

BÖLÜM 4

AES DOĞRUSAL DÖNÜŞÜMÜNE BENZER 4×4 VE 8×8 BOYUTUNDA TERSİ KENDİSİ (INVOLUTIF) MDS MATRİS TASARIMI

Tezin bu bölümünde Bölüm 3'te incelenen doğrusal dönüşümlerin yapıları göz önüne alınarak 4×4 ve 8×8 boyutunda involutif (tersi kendisi) MDS matris tasarımı yapılacaktır. Bu matrisler sırasıyla 32-bit'ten 32-bit'e ve 64-bit'ten 64-bit'e dönüşüm yapan doğrusal dönüşümlerdir (Yavuzer Aslan vd., 2011).

4.1. AES Doğrusal Dönüşümüne benzer 4×4 Tersi Kendisi (involutif) MDS Matris Tasarımı

4×4 boyutunda involutif doğrusal dönüşüm matrisi tasarlarken aşağıda verilen 4 kriteri sağlayan elemanları $GF(2^8)$ cisminden olan matrisler aranmaktadır. Bu kriterler aşağıda sıralanmıştır:

i. 4×4 boyutundaki matris MDS olsun,
ii. 4×4 boyutundaki matris tersi kendisi bir matris olsun,
iii. 4×4 boyutundaki matrisin her elemanı $GF(2^8)$ üzerine tanımlı ve düşük Hamming ağırlığına sahip olsun,
iv. Yazılım optimizasyonu için matris çarpımında yapılacak tüm byte işlemleri AES doğrusal dönüşümüne yakın olsun.

Diğer yandan oluşturulacak olan 4×4 boyutundaki matris elemanları eğer α ve α^2 (02_h ve 04_h) olarak seçilir ise hesaplama yapılırken ardışık iki kere aynı işlemi

yapmak kolaylık sağlayacaktır. Yukarıdaki kriterler göz önüne alındığında oluşturulan AES doğrusal dönüşümü benzeri matris aşağıdadır.

$$had(01_h, 02_h, 04_h, 06_h) = \begin{bmatrix} 01_h & 02_h & 04_h & 06_h \\ 02_h & 01_h & 06_h & 04_h \\ 04_h & 06_h & 01_h & 02_h \\ 06_h & 04_h & 02_h & 01_h \end{bmatrix}.$$

Oluşturulan matris incelendiğinde tüm elemanlarının farklı olduğu görülmektedir. Tez sırasında geliştirilen bir yazılım ile test edilen matris, MDS ve involutiftir. Test sonuçları EK E'de verilmiştir. Matrisin pseudo kodu aşağıdaki gibidir.

```
/* w doğrusal dönüşüme giriş vektörü */
X0 = xtime(w[0]);
X1 = xtime(X0);
X2 = xtime(w[1]);
X3 = xtime(X2);
X4 = xtime(w[2]);
X5 = xtime(X4);
X6 = xtime(w[3]);
X7 = xtime(X6);
w[0] = w[0] ^ X2 ^ X5 ^ X6 ^ X7;
w[1] = w[1] ^ X0 ^ X4 ^ X5 ^ X7;
w[2] = w[2] ^ X1 ^ X2 ^ X3 ^ X6;
w[3] = w[3] ^ X0 ^ X1 ^ X3 ^ X4;
```

Yukarıda görüldüğü gibi 16 XOR, 8 xtime (veya 8 tablo okuma) ve 8 değişken ile matris çarpımı optimize edilebilir. Bu değerler AES şifreleme algoritmasında kullanılan doğrusal dönüşüm ile yakındır. Fakat deşifreleme aşaması düşünüldüğünde önerilen matris involutif olduğundan dolayı, AES'in deşifreleme algoritmasındaki doğrusal dönüşümünden daha iyi performans sağlayacağı açıktır. Tablo 4.1'de AES matrisi ile önerilen matrisin yazılım performanslarının karşılaştırılması verilmektedir.

Tablo 4.1. AES matrisi ile önerilen matrisin yazılım performans karşılaştırması (Yavuzer Aslan vd., 2011)

MDS Matrisler	Her döngüdeki byte işlemleri			
	xtime	Tablo okuma	XOR	Toplam
AES (Nakahara ve Abrahão, 2009)	32	-	64	96
AES (Junod ve Vaudenay, 2004)	-	16	60	76
AES (Bölüm 3.3)	16	-	60	76
AES (Bölüm 3.3)		16	60	76
Önerilen Matris	32	-	64	96
Önerilen Matris	-	32	64	96

Tablo 4.1 incelendiğinde AES doğrusal dönüşüm matrisi ile önerilen doğrusal dönüşüm matrisinin performanslarının çok yakın olduğu görülecektir. MDS matris olduğundan iki doğrusal dönüşümünde dallanma sayısı aynıdır. Ancak sabit nokta sayısı aynı değildir. AES doğrusal dönüşümünün sabit nokta sayısı 2^8 iken, önerilen doğrusal dönüşümün sabit nokta sayısı 2^{16} olarak hesaplanmıştır. Aynı zamanda önerilen doğrusal dönüşümün involutif olduğu düşünüldüğünde şifreleme ve deşifreleme arasındaki zaman farkı kaldırılmıştır.

4.2. 8×8 Tersi Kendisi (involutif) MDS Matris Tasarımı

Bölüm 3.4'te anlatıldığı gibi Khazad şifreleme algoritması 8×8 boyutunda $Had(01_h, 03_h, 04_h, 05_h, 06_h, 08_h, 0B_h, 07_h)$ formunda MDS bir doğrusal dönüşüm matrisi kullanmaktadır.

Tezin bu bölümünde, daha önce anlatılan matematiksel altyapı kullanılarak 8×8 boyutunda tersi kendisi ve MDS bir doğrusal dönüşüm tasarlanmıştır. Doğrusal

dönüşüm tasarlanırken aşağıda verilen 3 kriter göz önüne alınmıştır (Yavuzer Aslan vd., 2011):

i. 8×8 boyutundaki matris MDS olsun,
ii. 8×8 boyutundaki matris involutif olsun,
iii. 8×8 boyutundaki matrisin elemanları $GF(2^8)$ üzerine tanımlı ve düşük Hamming ağırlığına sahip olsun.

Bu kriterler doğrultusunda oluşturulan 8×8 boyutunda tersi kendisi ve MDS olan doğrusal dönüşüm $A = Had(01_h, 02_h, 05_h, 04_h, 06_h, 0B_h, 09_h, 07_h)$ formundadır. Bu matris aşağıda verilmiştir (Yavuzer Aslan vd., 2011).

$$A = \begin{bmatrix} 01_h & 02_h & 05_h & 04_h & 06_h & 0B_h & 09_h & 07_h \\ 02_h & 01_h & 04_h & 05_h & 0B_h & 06_h & 07_h & 09_h \\ 05_h & 04_h & 01_h & 02_h & 09_h & 07_h & 06_h & 0B_h \\ 04_h & 05_h & 02_h & 01_h & 07_h & 09_h & 0B_h & 06_h \\ 06_h & 0B_h & 09_h & 07_h & 01_h & 02_h & 05_h & 04_h \\ 0B_h & 06_h & 07_h & 09_h & 02_h & 01_h & 04_h & 05_h \\ 09_h & 07_h & 06_h & 0B_h & 05_h & 04_h & 01_h & 02_h \\ 07_h & 09_h & 0B_h & 06_h & 04_h & 05_h & 02_h & 01_h \end{bmatrix}.$$

Verilen doğrusal dönüşümün MDS olup olmadığı 12804 determinantın incelenmesi sonucu belirlenmiştir. Bu determinantların hiçbiri 0 değerine sahip değildir. Hesaplamalar $GF(2^8)$'de $x^8 + x^4 + x^3 + x + 1$ indirgenemez polinomu ile yapılmıştır.

4.3. Boyutu *n*×*n* olan bir matrisin MDS matris olduğunu doğrulamak için geliştirilen yazılım

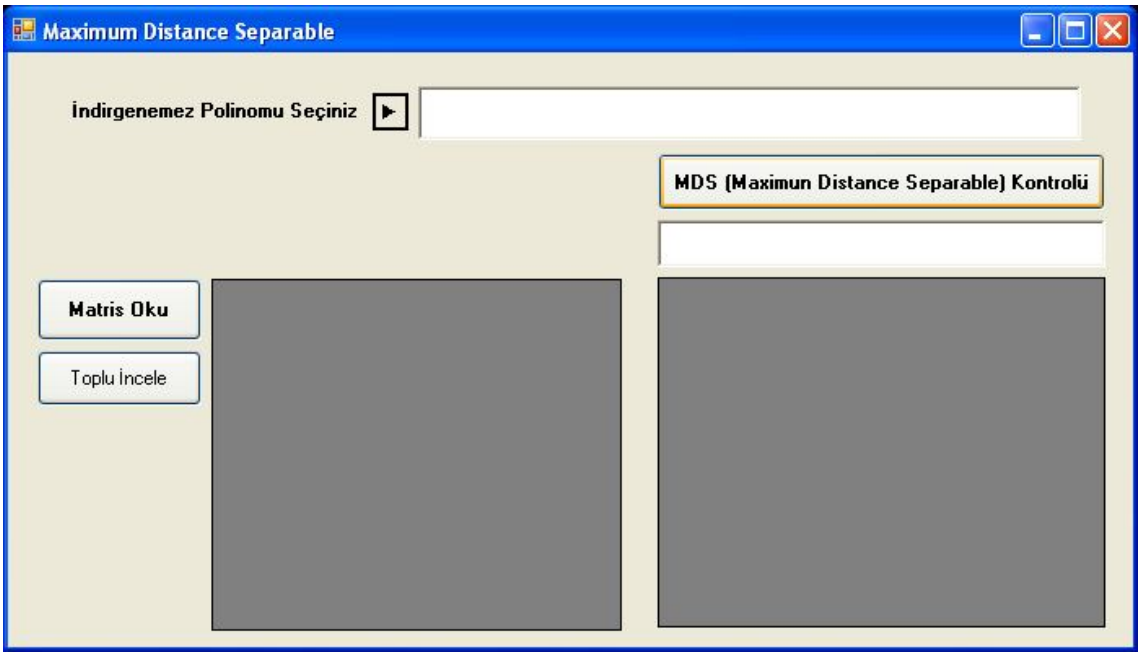
Doğrusal dönüşümlerin Bölüm 3'te anlatılan özelliklerini test etmek amacıyla tez esnasında bir yazılım aracı üretilmiştir. Buna göre bir doğrusal dönüşümün MDS matris olup olmadığını belirlemek için o matrisin alt determinantları geliştirilen yazılım ile incelenmektedir.

İncelenecek olan matrisin elemanları bir metin dosyasına aralarında boşluk bırakılarak yazılır ve yazılımdan çağırılır. Programda seçilen indirgenemez polinoma göre, matrisin tüm alt matrislerinin determinantları Bölüm 2'de anlatılan sonlu cisim aritmetiğine göre hesaplanır. Eğer alt determinantlardan herhangi bir tanesi 0 ise matris MDS değildir.

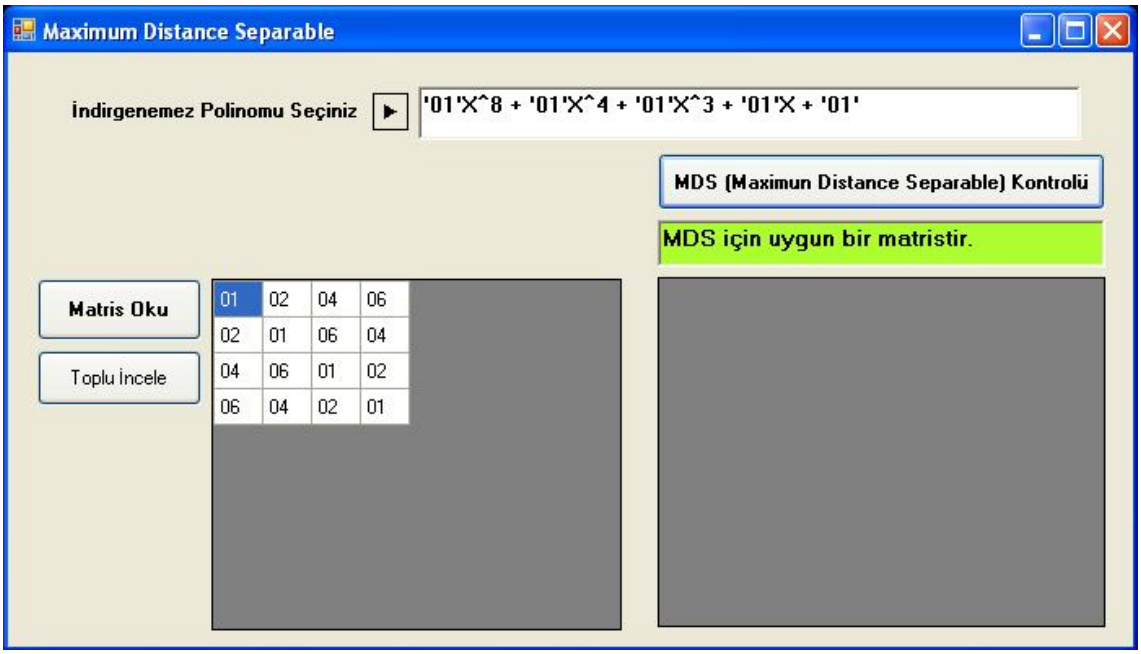
Yazılımın etkili ve hızlı çalışması açısından hesaplanan alt determinantlar bir dizide tutulmaktadır. Buna göre hesaplanacak olan determinant öncelikle bu dizide aranmaktadır. Eğer daha önceden hesaplanmış ise o determinant program tarafından göz ardı edilecektir. Örneğin Bölüm 3'te verilen ifade (3.3)'e göre 4×4 boyutunda bir matrisin, MDS matris olduğunu tespit etmek için 52 alt matrisi incelemek gerekirken yazılımdaki bu özellik sayesinde daha az sayıda matris (eğer aynı determinant hesabı var ise) incelenerek sonuca ulaşılabilir. 4×4 boyutunda AES doğrusal dönüşümü için 52 adet determinant hesabı yapılması gerekirken, program aynı determinant hesaplarını göz ardı ederek toplamda 38 determinant hesabı yapmaktadır.

Tez sırasında birçok matris test edilmiştir. Kolaylık olması açısından geliştirilen yazılıma "Toplu İncele" özelliği katılmıştır. Böylelikle tek seferde birçok matrisin testi yapılabilir hale gelmiştir.

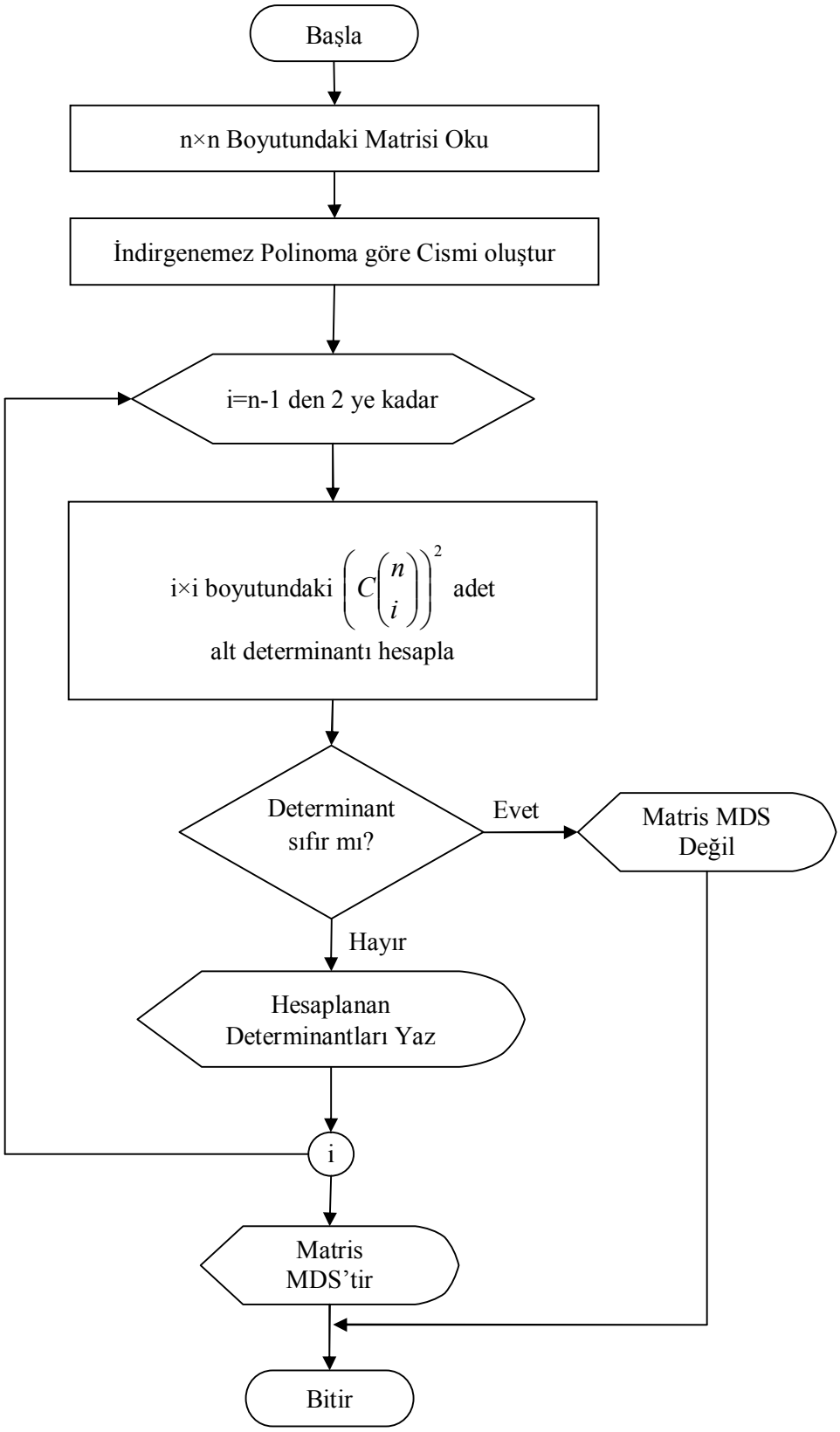
Geliştirilen yazılımın ara yüzü Şekil 4.1'de, $had(01_h, 02_h, 04_h, 06_h)$ matrisinin program ile test edilmiş görüntüsü Şekil 4.2'de ve blok şeması Şekil 4.3'de verilmektedir.



Şekil 4.1. MDS matris test ara yüzü



Şekil 4.2. $had(01_h,02_h,04_h,06_h)$ matrisinin program ile test edilmesi



Şekil 4.3. Geliştirilen yazılımın blok diyagramı

BÖLÜM 5

SONUÇLAR

Bu tezde blok şifrelerde kullanılan kriptografik doğrusal dönüşümler incelenmiştir. Modern blok şifrelerin tasarımında Maximum Distance Separable (MDS) ve Maximum Distance Binary Linear (MDBL) kodlar kriptografik doğrusal dönüşümler olarak kullanılmaktadır. Bu kriptografik dönüşümler çeşitlilik göstermektedir. Örneğin, Advanced Encryption Standard (AES) blok şifresi doğrusal dönüşüm olarak 4×4 büyüklüğünde $GF(2^8)$ üzerine 32-bit'ten 32-bit'e dönüşüm yapan bir MDS matris kullanırken, ARIA blok şifresi 16×16 boyutunda $GF(2^8)$ üzerine 128-bit'ten 128-bit'e dönüşüm yapan involutif bir ikili matris kullanmaktadır. Diğer yandan bir kriptografik doğrusal dönüşümün en önemli özelliği, blok şifrenin doğrusal ve diferansiyel saldırılara karşı güvenliğini ölçen, dallanma sayısı değeridir. Dolayısıyla blok şifrelerin tasarımında kullanılan doğrusal dönüşümler bu özelliğin optimal olduğu MDS ve MDBL kodlardır. Blok şifrelerde kullanılan doğrusal dönüşümlerin çoğunda sabit nokta sayısı açısından bir özenin gösterilmediği de tez çalışmasında ortaya konmuştur. Özellikle involutif doğrusal dönüşümlerin sabit nokta sayısının yüksek değerlere sahip olduğu da bu çalışmada gözlenmiştir. Dolayısıyla blok şifre tasarımında sabit nokta sayısı olabildiğince düşük yüksek dallanma sayısına sahip ve 8-bit, 32-bit ve 64-bit işlemci uygulamalarında yüksek performanslı doğrusal dönüşümlerin kullanılması gerekliliği bu tez için önemli bir sonuç olarak verilebilir.

Diğer yandan tezde involutif olmayan MDS matrislerinin uygulaması için bit tabanlı bir çarpma modeli de verilmiştir. Bu model, involutif olmayan MDS matrislerde şifreleme ve deşifreleme işlemlerinde ortaya çıkan hız farkını giderme amacındadır.

EKLER

EK A: x^4+x^3+1 **ve** $x^4+x^3+x^2+x+1$ **Polinomlarının Çarpım Tablosu**

(Bütün değerler hexadecimal notasyondadır.)

x^4+x^3+1 Polinomunun Çarpım Tablosu

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

$x^4 + x^3 + x^2 + x + 1$ Polinomunun Çarpım Tablosu

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

EK B: AES Şifresindeki Cisim için Table Lookup

(Bütün değerler hexadecimal notasyondadır.)

×	02	03	09	0B	0D	0E
01	02	03	09	0B	0D	0E
02	04	06	12	16	1A	1C
03	06	05	1B	1D	17	12
04	08	0C	24	2C	34	38
05	0A	0F	2D	27	39	36
06	0C	0A	36	3A	2E	24
07	0E	09	3F	31	23	2A
08	10	18	48	58	68	70
09	12	1B	41	53	65	7E
0A	14	1E	5A	4E	72	6C
0B	16	1D	53	45	7F	62
0C	18	14	6C	74	5C	48
0D	1A	17	65	7F	51	46
0E	1C	12	7E	62	46	54
0F	1E	11	77	69	4B	5A
10	20	30	90	B0	D0	E0
11	22	33	99	BB	DD	EE
12	24	36	82	A6	CA	FC
13	26	35	8B	AD	C7	F2
14	28	3C	B4	9C	E4	D8
15	2A	3F	BD	97	E9	D6
16	2C	3A	A6	8A	FE	C4
17	2E	39	AF	81	F3	CA
18	30	28	D8	E8	B8	90
19	32	2B	D1	E3	B5	9E
1A	34	2E	CA	FE	A2	8C
1B	36	2D	C3	F5	AF	82
1C	38	24	FC	C4	8C	A8
1D	3A	27	F5	CF	81	A6
1E	3C	22	EE	D2	96	B4
1F	3E	21	E7	D9	9B	BA
20	40	60	3B	7B	BB	DB
21	42	63	32	70	B6	D5
22	44	66	29	6D	A1	C7
23	46	65	20	66	AC	C9
24	48	6C	1F	57	8F	E3
25	4A	6F	16	5C	82	ED
26	4C	6A	0D	41	95	FF
27	4E	69	04	4A	98	F1
28	50	78	73	23	D3	AB
29	52	7B	7A	28	DE	A5

×	02	03	09	0B	0D	0E
2A	54	7E	61	35	C9	B7
2B	56	7D	68	3E	C4	B9
2C	58	74	57	0F	E7	93
2D	5A	77	5E	04	EA	9D
2E	5C	72	45	19	FD	8F
2F	5E	71	4C	12	F0	81
30	60	50	AB	CB	6B	3B
31	62	53	A2	C0	66	35
32	64	56	B9	DD	71	27
33	66	55	B0	D6	7C	29
34	68	5C	8F	E7	5F	03
35	6A	5F	86	EC	52	0D
36	6C	5A	9D	F1	45	1F
37	6E	59	94	FA	48	11
38	70	48	E3	93	03	4B
39	72	4B	EA	98	0E	45
3A	74	4E	F1	85	19	57
3B	76	4D	F8	8E	14	59
3C	78	44	C7	BF	37	73
3D	7A	47	CE	B4	3A	7D
3E	7C	42	D5	A9	2D	6F
3F	7E	41	DC	A2	20	61
40	80	C0	76	F6	6D	AD
41	82	C3	7F	FD	60	A3
42	84	C6	64	E0	77	B1
43	86	C5	6D	EB	7A	BF
44	88	CC	52	DA	59	95
45	8A	CF	5B	D1	54	9B
46	8C	CA	40	CC	43	89
47	8E	C9	49	C7	4E	87
48	90	D8	3E	AE	05	DD
49	92	DB	37	A5	08	D3
4A	94	DE	2C	B8	1F	C1
4B	96	DD	25	B3	12	CF
4C	98	D4	1A	82	31	E5
4D	9A	D7	13	89	3C	EB
4E	9C	D2	08	94	2B	F9
4F	9E	D1	01	9F	26	F7
50	A0	F0	E6	46	BD	4D
51	A2	F3	EF	4D	B0	43
52	A4	F6	F4	50	A7	51

×	02	03	09	0B	0D	0E
53	A6	F5	FD	5B	AA	5F
54	A8	FC	C2	6A	89	75
55	AA	FF	CB	61	84	7B
56	AC	FA	D0	7C	93	69
57	AE	F9	D9	77	9E	67
58	B0	E8	AE	1E	D5	3D
59	B2	EB	A7	15	D8	33
5A	B4	EE	BC	08	CF	21
5B	B6	ED	B5	03	C2	2F
5C	B8	E4	8A	32	E1	05
5D	BA	E7	83	39	EC	0B
5E	BC	E2	98	24	FB	19
5F	BE	E1	91	2F	F6	17
60	C0	A0	4D	8D	D6	76
61	C2	A3	44	86	DB	78
62	C4	A6	5F	9B	CC	6A
63	C6	A5	56	90	C1	64
64	C8	AC	69	A1	E2	4E
65	CA	AF	60	AA	EF	40
66	CC	AA	7B	B7	F8	52
67	CE	A9	72	BC	F5	5C
68	D0	B8	05	D5	BE	06
69	D2	BB	0C	DE	B3	08
6A	D4	BE	17	C3	A4	1A
6B	D6	BD	1E	C8	A9	14
6C	D8	B4	21	F9	8A	3E
6D	DA	B7	28	F2	87	30
6E	DC	B2	33	EF	90	22
6F	DE	B1	3A	E4	9D	2C
70	E0	90	DD	3D	06	96
71	E2	93	D4	36	0B	98
72	E4	96	CF	2B	1C	8A
73	E6	95	C6	20	11	84
74	E8	9C	F9	11	32	AE
75	EA	9F	F0	1A	3F	A0
76	EC	9A	EB	07	28	B2
77	EE	99	E2	0C	25	BC
78	F0	88	95	65	6E	E6
79	F2	8B	9C	6E	63	E8
7A	F4	8E	87	73	74	FA
7B	F6	8D	8E	78	79	F4
7C	F8	84	B1	49	5A	DE
7D	FA	87	B8	42	57	D0
7E	FC	82	A3	5F	40	C2

×	02	03	09	0B	0D	0E
7F	FE	81	AA	54	4D	CC
80	1B	9B	EC	F7	DA	41
81	19	98	E5	FC	D7	4F
82	1F	9D	FE	E1	C0	5D
83	1D	9E	F7	EA	CD	53
84	13	97	C8	DB	EE	79
85	11	94	C1	D0	E3	77
86	17	91	DA	CD	F4	65
87	15	92	D3	C6	F9	6B
88	0B	83	A4	AF	B2	31
89	09	80	AD	A4	BF	3F
8A	0F	85	B6	B9	A8	2D
8B	0D	86	BF	B2	A5	23
8C	03	8F	80	83	86	09
8D	01	8C	89	88	8B	07
8E	07	89	92	95	9C	15
8F	05	8A	9B	9E	91	1B
90	3B	AB	7C	47	0A	A1
91	39	A8	75	4C	07	AF
92	3F	AD	6E	51	10	BD
93	3D	AE	67	5A	1D	B3
94	33	A7	58	6B	3E	99
95	31	A4	51	60	33	97
96	37	A1	4A	7D	24	85
97	35	A2	43	76	29	8B
98	2B	B3	34	1F	62	D1
99	29	B0	3D	14	6F	DF
9A	2F	B5	26	09	78	CD
9B	2D	B6	2F	02	75	C3
9C	23	BF	10	33	56	E9
9D	21	BC	19	38	5B	E7
9E	27	B9	02	25	4C	F5
9F	25	BA	0B	2E	41	FB
A0	5B	FB	D7	8C	61	9A
A1	59	F8	DE	87	6C	94
A2	5F	FD	C5	9A	7B	86
A3	5D	FE	CC	91	76	88
A4	53	F7	F3	A0	55	A2
A5	51	F4	FA	AB	58	AC
A6	57	F1	E1	B6	4F	BE
A7	55	F2	E8	BD	42	B0
A8	4B	E3	9F	D4	09	EA
A9	49	E0	96	DF	04	E4
AA	4F	E5	8D	C2	13	F6

×	02	03	09	0B	0D	0E
AB	4D	E6	84	C9	1E	F8
AC	43	EF	BB	F8	3D	D2
AD	41	EC	B2	F3	30	DC
AE	47	E9	A9	EE	27	CE
AF	45	EA	A0	E5	2A	C0
B0	7B	CB	47	3C	B1	7A
B1	79	C8	4E	37	BC	74
B2	7F	CD	55	2A	AB	66
B3	7D	CE	5C	21	A6	68
B4	73	C7	63	10	85	42
B5	71	C4	6A	1B	88	4C
B6	77	C1	71	06	9F	5E
B7	75	C2	78	0D	92	50
B8	6B	D3	0F	64	D9	0A
B9	69	D0	06	6F	D4	04
BA	6F	D5	1D	72	C3	16
BB	6D	D6	14	79	CE	18
BC	63	DF	2B	48	ED	32
BD	61	DC	22	43	E0	3C
BE	67	D9	39	5E	F7	2E
BF	65	DA	30	55	FA	20
C0	9B	5B	9A	01	B7	EC
C1	99	58	93	0A	BA	E2
C2	9F	5D	88	17	AD	F0
C3	9D	5E	81	1C	A0	FE
C4	93	57	BE	2D	83	D4
C5	91	54	B7	26	8E	DA
C6	97	51	AC	3B	99	C8
C7	95	52	A5	30	94	C6
C8	8B	43	D2	59	DF	9C
C9	89	40	DB	52	D2	92
CA	8F	45	C0	4F	C5	80
CB	8D	46	C9	44	C8	8E
CC	83	4F	F6	75	EB	A4
CD	81	4C	FF	7E	E6	AA
CE	87	49	E4	63	F1	B8
CF	85	4A	ED	68	FC	B6
D0	BB	6B	0A	B1	67	0C
D1	B9	68	03	BA	6A	02
D2	BF	6D	18	A7	7D	10
D3	BD	6E	11	AC	70	1E
D4	B3	67	2E	9D	53	34
D5	B1	64	27	96	5E	3A
D6	B7	61	3C	8B	49	28

×	02	03	09	0B	0D	0E
D7	B5	62	35	80	44	26
D8	AB	73	42	E9	0F	7C
D9	A9	70	4B	E2	02	72
DA	AF	75	50	FF	15	60
DB	AD	76	59	F4	18	6E
DC	A3	7F	66	C5	3B	44
DD	A1	7C	6F	CE	36	4A
DE	A7	79	74	D3	21	58
DF	A5	7A	7D	D8	2C	56
E0	DB	3B	A1	7A	0C	37
E1	D9	38	A8	71	01	39
E2	DF	3D	B3	6C	16	2B
E3	DD	3E	BA	67	1B	25
E4	D3	37	85	56	38	0F
E5	D1	34	8C	5D	35	01
E6	D7	31	97	40	22	13
E7	D5	32	9E	4B	2F	1D
E8	CB	23	E9	22	64	47
E9	C9	20	E0	29	69	49
EA	CF	25	FB	34	7E	5B
EB	CD	26	F2	3F	73	55
EC	C3	2F	CD	0E	50	7F
ED	C1	2C	C4	05	5D	71
EE	C7	29	DF	18	4A	63
EF	C5	2A	D6	13	47	6D
F0	FB	0B	31	CA	DC	D7
F1	F9	08	38	C1	D1	D9
F2	FF	0D	23	DC	C6	CB
F3	FD	0E	2A	D7	CB	C5
F4	F3	07	15	E6	E8	EF
F5	F1	04	1C	ED	E5	E1
F6	F7	01	07	F0	F2	F3
F7	F5	02	0E	FB	FF	FD
F8	EB	13	79	92	B4	A7
F9	E9	10	70	99	B9	A9
FA	EF	15	6B	84	AE	BB
FB	ED	16	62	8F	A3	B5
FC	E3	1F	5D	BE	80	9F
FD	E1	1C	54	B5	8D	91
FE	E7	19	4F	A8	9A	83
FF	E5	1A	46	A3	97	8D

EK C: $x^4+x^3+x^2+x+1$ **ve** x^4+x+1 **İndirgenemez Polinomlarının Basamak Çarpım Değerleri**

$x^4+x^3+x^2+x+1$ polinomunun basamak çarpım değerleri

	α^3	α^2	α	1
1_h	x_3	x_2	x_1	x_0
2_h	x_3+x_2	(x_3+x_1)	(x_3+x_0)	x_3
3_h	x_2	$(x_3+x_2+x_1)$	$(x_3+x_1+x_0)$	(x_3+x_0)
4_h	(x_2+x_1)	(x_2+x_0)	x_2	(x_3+x_2)
5_h	$(x_3+x_2+x_1)$	x_0	(x_2+x_1)	$(x_3+x_2+x_0)$
6_h	(x_3+x_1)	$(x_3+x_2+x_1+x_0)$	$(x_3+x_2+x_0)$	x_2
7_h	x_1	$(x_3+x_1+x_0)$	$(x_3+x_2+x_1+x_0)$	(x_2+x_0)
8_h	(x_1+x_0)	x_1	(x_3+x_1)	(x_2+x_1)
9_h	$(x_3+x_1+x_0)$	(x_2+x_1)	x_3	$(x_2+x_1+x_0)$
A_h	$(x_3+x_2+x_1+x_0)$	x_3	(x_1+x_0)	$(x_3+x_2+x_1)$
B_h	$(x_2+x_1+x_0)$	(x_3+x_2)	x_0	$(x_3+x_2+x_1+x_0)$
C_h	(x_2+x_0)	$(x_2+x_1+x_0)$	$(x_3+x_2+x_1)$	(x_3+x_1)
D_h	$(x_3+x_2+x_0)$	(x_1+x_0)	(x_3+x_2)	$(x_3+x_1+x_0)$
E_h	(x_3+x_0)	$(x_3+x_2+x_0)$	$(x_2+x_1+x_0)$	x_1
F_h	x_0	(x_3+x_0)	(x_2+x_0)	(x_1+x_0)

x^4+x+1 polinomunun basamak çarpım değerleri

	α^3	α^2	α	1
1_h	x_3	x_2	x_1	x_0
2_h	x_2	x_1	x_3+x_0	x_3
3_h	x_3+x_2	x_2+x_1	$x_3+x_1+x_0$	x_3+x_0
4_h	x_1	x_3+x_0	x_3+x_2	x_2
5_h	x_3+x_1	$x_3+x_2+x_0$	$x_3+x_2+x_1$	x_2+x_0
6_h	x_2+x_1	$x_3+x_1+x_0$	x_2+x_0	x_3+x_2
7_h	$x_3+x_2+x_1$	$x_3+x_2+x_1+x_0$	$x_2+x_1+x_0$	$x_3+x_2+x_0$
8_h	x_3+x_0	x_3+x_2	x_2+x_1	x_1
9_h	x_0	x_3	x_2	x_1+x_0
A_h	$x_3+x_2+x_0$	$x_3+x_2+x_1$	$x_3+x_2+x_1+x_0$	x_3+x_1
B_h	x_2+x_0	x_3+x_{10}	$x_3+x_2+x_0$	$x_3+x_1+x_0$
C_h	$x_3+x_1+x_0$	x_2+x_0	x_3+x_1	x_2+x_1
D_h	x_1+x_0	x_0	x_3	$x_2+x_1+x_0$
E_h	$x_3+x_2+x_1+x_0$	$x_2+x_1+x_0$	x_1+x_0	$x_3+x_2+x_1$
F_h	$x_2+x_1+x_0$	x_1+x_0	x_0	$x_3+x_2+x_1+x_0$

EK D: AES Şifreleme Algoritması için Nokta Ürün Tablosu

	α^7	α^6	α^5	α^4	α^3	α^2	α	1
01	00000001	00000010	00000100	00001000	00010000	00100000	01000000	10000000
02	10000000	10000001	00000010	10000100	10001000	00010000	00100000	01000000
03	10000001	10000011	00000110	10001100	10011000	00110000	01100000	11000000
04	01000000	11000000	10000001	01000010	11000100	10001000	00010000	00100000
05	01000001	11000010	10000101	01001010	11010100	10101000	01010000	10100000
06	11000000	01000001	10000011	11000110	01001100	10011000	00110000	01100000
07	11000001	01000011	10000111	11001110	01011100	10111000	01110000	11100000
08	00100000	01100000	11000000	10100001	01100010	11000100	10001000	00010000
09	00100001	01100010	11000100	10101001	01110010	11100100	11001000	10010000
0A	10100000	11100001	11000010	00100101	11101010	11010100	10101000	01010000
0B	10100001	11100011	11000110	00101101	11111010	11110100	11101000	11010000
0C	01100000	10100000	01000001	11100011	10100110	01001100	10011000	00110000
0D	01100001	10100010	01000101	11101011	10110110	01101100	11011000	10110000
0E	11100000	00100001	01000011	01100111	00101110	01011100	10111000	01110000
0F	11100001	00100011	01000111	01101111	00111110	01111100	11111000	11110000
10	00010000	00110000	01100000	11010000	10110001	01100010	11000100	10001000
11	00010001	00110010	01100100	11011000	10100001	01000010	10000100	00001000
12	10010000	10110001	01100010	01010100	00111001	01110010	11100100	11001000
13	10010001	10110011	01100110	01011100	00101001	01010010	10100100	01001000
14	01010000	11110000	11100001	10010010	01110101	11101010	11010100	10101000
15	01010001	11110010	11100101	10011010	01100101	11001010	10010100	00101000
16	11010000	01110001	11100011	00010110	11111101	11111010	11110100	11101000
17	11010001	01110011	11100111	00011110	11101101	11011010	10110100	01101000
18	00110000	01010000	10100000	01110001	11010011	10100110	01001100	10011000
19	00110001	01010010	10100100	01111001	11000011	10000110	00001100	00011000
1A	10110000	11010001	10100010	11110101	01011011	10110110	01101100	11011000
1B	10110001	11010011	10100110	11111101	01001011	10010110	00101100	01011000
1C	01110000	10010000	00100001	00110011	00010111	00101110	01011100	10111000
1D	01110001	10010010	00100101	00111011	00000111	00001110	00011100	00111000
1E	11110000	00010001	00100011	10110111	10011111	00111110	01111100	11111000
1F	11110001	00010011	00100111	10111111	10001111	00011110	00111100	01111000
20	10001000	10011000	00110000	11101000	01011000	10110001	01100010	11000100
21	10001001	10011010	00110100	11100000	01001000	10010001	00100010	01000100
22	00001000	00011001	00110010	01101100	11010000	10100001	01000010	10000100
23	00001001	00011011	00110110	01100100	11000000	10000001	00000010	00000100
24	11001000	01011000	10110001	10101010	10011100	00111001	01110010	11100100

25	11001001	01011010	10110101	10100010	10001100	00011001	00110010	01100100
26	01001000	11011001	10110011	00101110	00010100	00101001	01010010	10100100
27	01001001	11011011	10110111	00100110	00000100	00001001	00010010	00100100
28	10101000	11111000	11110000	01001001	00111010	01110101	11101010	11010100
29	10101001	11111010	11110100	01000001	00101010	01010101	10101010	01010100
2A	00101000	01111001	11110010	11001101	10110010	01100101	11001010	10010100
2B	00101001	01111011	11110110	11000101	10100010	01000101	10001010	00010100
2C	11101000	00111000	01110001	00001011	11111110	11111101	11111010	11110100
2D	11101001	00111010	01110101	00000011	11101110	11011101	10111010	01110100
2E	01101000	10111001	01110011	10001111	01110110	11101101	11011010	10110100
2F	01101001	10111011	01110111	10000111	01100110	11001101	10011010	00110100
30	10011000	10101000	01010000	00111000	11101001	11010011	10100110	01001100
31	10011001	10101010	01010100	00110000	11111001	11110011	11100110	11001100
32	00011000	00101001	01010010	10111100	01100001	11000011	10000110	00001100
33	00011001	00101011	01010110	10110100	01110001	11100011	11000110	10001100
34	11011000	01101000	11010001	01111010	00101101	01011011	10110110	01101100
35	11011001	01101010	11010101	01110010	00111101	01111011	11110110	11101100
36	01011000	11101001	11010011	11111110	10100101	01001011	10010110	00101100
37	01011001	11101011	11010111	11110110	10110101	01101011	11010110	10101100
38	10111000	11001000	10010000	10011001	10001011	00010111	00101110	01011100
39	10111001	11001010	10010100	10010001	10011011	00110111	01101110	11011100
3A	00111000	01001001	10010010	00011101	00000011	00000111	00001110	00011100
3B	00111001	01001011	10010110	00010101	00010011	00100111	01001110	10011100
3C	11111000	00001000	00010001	11011011	01001111	10011111	00111110	01111100
3D	11111001	00001010	00010101	11010011	01011111	10111111	01111110	11111100
3E	01111000	10001001	00010011	01011111	11000111	10001111	00011110	00111100
3F	01111001	10001011	00010111	01010111	11010111	10101111	01011110	10111100
40	11000100	01001100	10011000	11110100	00101100	01011000	10110001	01100010
41	11000101	01001110	10011100	11111100	00111100	01111000	11110001	11100010
42	01000100	11001101	10011010	01110000	10100100	01001000	10010001	00100010
43	01000101	11001111	10011110	01111000	10110100	01101000	11010001	10100010
44	10000100	10001100	00011001	10110110	11101000	11010000	10100001	01000010
45	10000101	10001110	00011101	10111110	11111000	11110000	11100001	11000010
46	00000100	00001101	00011011	00110010	01100000	11000000	10000001	00000010
47	00000101	00001111	00011111	00111010	01110000	11100000	11000001	10000010
48	11100100	00101100	01011000	01010101	01001110	10011100	00111001	01110010
49	11100101	00101110	01011100	01011101	01011110	10111100	01111001	11110010
4A	01100100	10101101	01011010	11010001	11000110	10001100	00011001	00110010
4B	01100101	10101111	01011110	11011001	11010110	10101100	01011001	10110010

4C	10100100	11101100	11011001	00010111	10001010	00010100	00101001	01010010
4D	10100101	11101110	11011101	00011111	10011010	00110100	01101001	11010010
4E	00100100	01101101	11011011	10010011	00000010	00000100	00001001	00010010
4F	00100101	01101111	11011111	10011011	00010010	00100100	01001001	10010010
50	11010100	01111100	11111000	00100100	10011101	00111010	01110101	11101010
51	11010101	01111110	11111100	00101100	10001101	00011010	00110101	01101010
52	01010100	11111101	11111010	10100000	00010101	00101010	01010101	10101010
53	01010101	11111111	11111110	10101000	00000101	00001010	00010101	00101010
54	10010100	10111100	01111001	01100110	01011001	10110010	01100101	11001010
55	10010101	10111110	01111101	01101110	01001001	10010010	00100101	01001010
56	00010100	00111101	01111011	11100010	11010001	10100010	01000101	10001010
57	00010101	00111111	01111111	11101010	11000001	10000010	00000101	00001010
58	11110100	00011100	00111000	10000101	11111111	11111110	11111101	11111010
59	11110101	00011110	00111100	10001101	11101111	11011110	10111101	01111010
5A	01110100	10011101	00111010	00000001	01110111	11101110	11011101	10111010
5B	01110101	10011111	00111110	00001001	01100111	11001110	10011101	00111010
5C	10110100	11011100	10111001	11000111	00111011	01110110	11101101	11011010
5D	10110101	11011110	10111101	11001111	00101011	01010110	10101101	01011010
5E	00110100	01011101	10111011	01000011	10110011	01100110	11001101	10011010
5F	00110101	01011111	10111111	01001011	10100011	01000110	10001101	00011010
60	01001100	11010100	10101000	00011100	01110100	11101001	11010011	10100110
61	01001101	11010110	10101100	00010100	01100100	11001001	10010011	00100110
62	11001100	01010101	10101010	10011000	11111100	11111001	11110011	11100110
63	11001101	01010111	10101110	10010000	11101100	11011001	10110011	01100110
64	00001100	00010100	00101001	01011110	10110000	01100001	11000011	10000110
65	00001101	00010110	00101101	01010110	10100000	01000001	10000011	00000110
66	10001100	10010101	00101011	11011010	00111000	01110001	11100011	11000110
67	10001101	10010111	00101111	11010010	00101000	01010001	10100011	01000110
68	01101100	10110100	01101000	10111101	00010110	00101101	01011011	10110110
69	01101101	10110110	01101100	10110101	00000110	00001101	00011011	00110110
6A	11101100	00110101	01101010	00111001	10011110	00111101	01111011	11110110
6B	11101101	00110111	01101110	00110001	10001110	00011101	00111011	01110110
6C	00101100	01110100	11101001	11111111	11010010	10100101	01001011	10010110
6D	00101101	01110110	11101101	11110111	11000010	10000101	00001011	00010110
6E	10101100	11110101	11101011	01111011	01011010	10110101	01101011	11010110
6F	10101101	11110111	11101111	01110011	01001010	10010101	00101011	01010110
70	01011100	11100100	11001000	11001100	11000101	10001011	00010111	00101110
71	01011101	11100110	11001100	11000100	11010101	10101011	01010111	10101110
72	11011100	01100101	11001010	01001000	01001101	10011011	00110111	01101110

73	11011101	01100111	11001110	01000000	01011101	10111011	01110111	11101110
74	00011100	00100100	01001001	10001110	00000001	00000011	00000111	00001110
75	00011101	00100110	01001101	10000110	00010001	00100011	01000111	10001110
76	10011100	10100101	01001011	00001010	10001001	00010011	00100111	01001110
77	10011101	10100111	01001111	00000010	10011001	00110011	01100111	11001110
78	01111100	10000100	00001000	01101101	10100111	01001111	10011111	00111110
79	01111101	10000110	00001100	01100101	10110111	01101111	11011111	10111110
7A	11111100	00000101	00001010	11101001	00101111	01011111	10111111	01111110
7B	11111101	00000111	00001110	11100001	00111111	01111111	11111111	11111110
7C	00111100	01000100	10001001	00101111	01100011	11000111	10001111	00011110
7D	00111101	01000110	10001101	00100111	01110011	11100111	11001111	10011110
7E	10111100	11000101	10001011	10101011	11101011	11010111	10101111	01011110
7F	10111101	11000111	10001111	10100011	11111011	11110111	11101111	11011110
80	01100010	10100110	01001100	11111010	10010110	00101100	01011000	10110001
81	01100011	10100100	01001000	11110010	10000110	00001100	00011000	00110001
82	11100010	00100111	01001110	01111110	00011110	00111100	01111000	11110001
83	11100011	00100101	01001010	01110110	00001110	00011100	00111000	01110001
84	00100010	01100110	11001101	10111000	01010010	10100100	01001000	10010001
85	00100011	01100100	11001001	10110000	01000010	10000100	00001000	00010001
86	10100010	11100111	11001111	00111100	11011010	10110100	01101000	11010001
87	10100011	11100101	11001011	00110100	11001010	10010100	00101000	01010001
88	01000010	11000110	10001100	01011011	11110100	11101000	11010000	10100001
89	01000011	11000100	10001000	01010011	11100100	11001000	10010000	00100001
8A	11000010	01000111	10001110	11011111	01111100	11111000	11110000	11100001
8B	11000011	01000101	10001010	11010111	01101100	11011000	10110000	01100001
8C	00000010	00000110	00001101	00011001	00110000	01100000	11000000	10000001
8D	00000011	00000100	00001001	00010001	00100000	01000000	10000000	00000001
8E	10000010	10000111	00001111	10011101	10111000	01110000	11100000	11000001
8F	10000011	10000101	00001011	10010101	10101000	01010000	10100000	01000001
90	01110010	10010110	00101100	00101010	00100111	01001110	10011100	00111001
91	01110011	10010100	00101000	00100010	00110111	01101110	11011100	10111001
92	11110010	00010111	00101110	10101110	10101111	01011110	10111100	01111001
93	11110011	00010101	00101010	10100110	10111111	01111110	11111100	11111001
94	00110010	01010110	10101101	01101000	11100011	11000110	10001100	00011001
95	00110011	01010100	10101001	01100000	11110011	11100110	11001100	10011001
96	10110010	11010111	10101111	11101100	01101011	11010110	10101100	01011001
97	10110011	11010101	10101011	11100100	01111011	11110110	11101100	11011001
98	01010010	11110110	11101100	10001011	01000101	10001010	00010100	00101001
99	01010011	11110100	11101000	10000011	01010101	10101010	01010100	10101001

9A	11010010	01110111	11101110	00001111	11001101	10011010	00110100	01101001
9B	11010011	01110101	11101010	00000111	11011101	10111010	01110100	11101001
9C	00010010	00110110	01101101	11001001	10000001	00000010	00000100	00001001
9D	00010011	00110100	01101001	11000001	10010001	00100010	01000100	10001001
9E	10010010	10110111	01101111	01001101	00001001	00010010	00100100	01001001
9F	10010011	10110101	01101011	01000101	00011001	00110010	01100100	11001001
A0	11101010	00111110	01111100	00010010	11001110	10011101	00111010	01110101
A1	11101011	00111100	01111000	00011010	11011110	10111101	01111010	11110101
A2	01101010	10111111	01111110	10010110	01000110	10001101	00011010	00110101
A3	01101011	10111101	01111010	10011110	01010110	10101101	01011010	10110101
A4	10101010	11111110	11111101	01010000	00001010	00010101	00101010	01010101
A5	10101011	11111100	11111001	01011000	00011010	00110101	01101010	11010101
A6	00101010	01111111	11111111	11010100	10000010	00000101	00001010	00010101
A7	00101011	01111101	11111011	11011100	10010010	00100101	01001010	10010101
A8	11001010	01011110	10111100	10110011	10101100	01011001	10110010	01100101
A9	11001011	01011100	10111000	10111011	10111100	01111001	11110010	11100101
AA	01001010	11011111	10111110	00110111	00100100	01001001	10010010	00100101
AB	01001011	11011101	10111010	00111111	00110100	01101001	11010010	10100101
AC	10001010	10011110	00111101	11110001	01101000	11010001	10100010	01000101
AD	10001011	10011100	00111001	11111001	01111000	11110001	11100010	11000101
AE	00001010	00011111	00111111	01110101	11100000	11000001	10000010	00000101
AF	00001011	00011101	00111011	01111101	11110000	11100001	11000010	10000101
B0	11111010	00001110	00011100	11000010	01111111	11111111	11111110	11111101
B1	11111011	00001100	00011000	11001010	01101111	11011111	10111110	01111101
B2	01111010	10001111	00011110	01000110	11110111	11101111	11011110	10111101
B3	01111011	10001101	00011010	01001110	11100111	11001111	10011110	00111101
B4	10111010	11001110	10011101	10000000	10111011	01110111	11101110	11011101
B5	10111011	11001100	10011001	10001000	10101011	01010111	10101110	01011101
B6	00111010	01001111	10011111	00000100	00110011	01100111	11001110	10011101
B7	00111011	01001101	10011011	00001100	00100011	01000111	10001110	00011101
B8	11011010	01101110	11011100	01100011	00011101	00111011	01110110	11101101
B9	11011011	01101100	11011000	01101011	00001101	00011011	00110110	01101101
BA	01011010	11101111	11011110	11100111	10010101	00101011	01010110	10101101
BB	01011011	11101101	11011010	11101111	10000101	00001011	00010110	00101101
BC	10011010	10101110	01011101	00100001	11011001	10110011	01100110	11001101
BD	10011011	10101100	01011001	00101001	11001001	10010011	00100110	01001101
BE	00011010	00101111	01011111	10100101	01010001	10100011	01000110	10001101
BF	00011011	00101101	01011011	10101101	01000001	10000011	00000110	00001101
C0	10100110	11101010	11010100	00001110	10111010	01110100	11101001	11010011

C1	10100111	11101000	11010000	00000110	10101010	01010100	10101001	01010011
C2	00100110	01101011	11010110	10001010	00110010	01100100	11001001	10010011
C3	00100111	01101001	11010010	10000010	00100010	01000100	10001001	00010011
C4	11100110	00101010	01010101	01001100	01111110	11111100	11111001	11110011
C5	11100111	00101000	01010001	01000100	01101110	11011100	10111001	01110011
C6	01100110	10101011	01010111	11001000	11110110	11101100	11011001	10110011
C7	01100111	10101001	01010011	11000000	11100110	11001100	10011001	00110011
C8	10000110	10001010	00010100	10101111	11011000	10110000	01100001	11000011
C9	10000111	10001000	00010000	10100111	11001000	10010000	00100001	01000011
CA	00000110	00001011	00010110	00101011	01010000	10100000	01000001	10000011
CB	00000111	00001001	00010010	00100011	01000000	10000000	00000001	00000011
CC	11000110	01001010	10010101	11101101	00011100	00111000	01110001	11100011
CD	11000111	01001000	10010001	11100101	00001100	00011000	00110001	01100011
CE	01000110	11001011	10010111	01101001	10010100	00101000	01010001	10100011
CF	01000111	11001001	10010011	01100001	10000100	00001000	00010001	00100011
D0	10110110	11011010	10110100	11011110	00001011	00010110	00101101	01011011
D1	10110111	11011000	10110000	11010110	00011011	00110110	01101101	11011011
D2	00110110	01011011	10110110	01011010	10000011	00000110	00001101	00011011
D3	00110111	01011001	10110010	01010010	10010011	00100110	01001101	10011011
D4	11110110	00011010	00110101	10011100	11001111	10011110	00111101	01111011
D5	11110111	00011000	00110001	10010100	11011111	10111110	01111101	11111011
D6	01110110	10011011	00110111	00011000	01000111	10001110	00011101	00111011
D7	01110111	10011001	00110011	00010000	01010111	10101110	01011101	10111011
D8	10010110	10111010	01110100	01111111	01101001	11010010	10100101	01001011
D9	10010111	10111000	01110000	01110111	01111001	11110010	11100101	11001011
DA	00010110	00111011	01110110	11111011	11100001	11000010	10000101	00001011
DB	00010111	00111001	01110010	11110011	11110001	11100010	11000101	10001011
DC	11010110	01111010	11110101	00111101	10101101	01011010	10110101	01101011
DD	11010111	01111000	11110001	00110101	10111101	01111010	11110101	11101011
DE	01010110	11111011	11110111	10111001	00100101	01001010	10010101	00101011
DF	01010111	11111001	11110011	10110001	00110101	01101010	11010101	10101011
E0	00101110	01110010	11100100	11100110	11100010	11000101	10001011	00010111
E1	00101111	01110000	11100000	11101110	11110010	11100101	11001011	10010111
E2	10101110	11110011	11100110	01100010	01101010	11010101	10101011	01010111
E3	10101111	11110001	11100010	01101010	01111010	11110101	11101011	11010111
E4	01101110	10110010	01100101	10100100	00100110	01001101	10011011	00110111
E5	01101111	10110000	01100001	10101100	00110110	01101101	11011011	10110111
E6	11101110	00110011	01100111	00100000	10101110	01011101	10111011	01110111
E7	11101111	00110001	01100011	00101000	10111110	01111101	11111011	11110111

E8	00001110	00010010	00100100	01000111	10000000	00000001	00000011	00000111
E9	00001111	00010000	00100000	01001111	10010000	00100001	01000011	10000111
EA	10001110	10010011	00100110	11000011	00001000	00010001	00100011	01000111
EB	10001111	10010001	00100010	11001011	00011000	00110001	01100011	11000111
EC	01001110	11010010	10100101	00000101	01000100	10001001	00010011	00100111
ED	01001111	11010000	10100001	00001101	01010100	10101001	01010011	10100111
EE	11001110	01010011	10100111	10000001	11001100	10011001	00110011	01100111
EF	11001111	01010001	10100011	10001001	11011100	10111001	01110011	11100111
F0	00111110	01000010	10000100	00110110	01010011	10100111	01001111	10011111
F1	00111111	01000000	10000000	00111110	01000011	10000111	00001111	00011111
F2	10111110	11000011	10000110	10110010	11011011	10110111	01101111	11011111
F3	10111111	11000001	10000010	10111010	11001011	10010111	00101111	01011111
F4	01111110	10000010	00000101	01110100	10010111	00101111	01011111	10111111
F5	01111111	10000000	00000001	01111100	10000111	00001111	00011111	00111111
F6	11111110	00000011	00000111	11110000	00011111	00111111	01111111	11111111
F7	11111111	00000001	00000011	11111000	00001111	00011111	00111111	01111111
F8	00011110	00100010	01000100	10010111	00110001	01100011	11000111	10001111
F9	00011111	00100000	01000000	10011111	00100001	01000011	10000111	00001111
FA	10011110	10100011	01000110	00010011	10111001	01110011	11100111	11001111
FB	10011111	10100001	01000010	00011011	10101001	01010011	10100111	01001111
FC	01011110	11100010	11000101	11010101	11110101	11101011	11010111	10101111
FD	01011111	11100000	11000001	11011101	11100101	11001011	10010111	00101111
FE	11011110	01100011	11000111	01010001	01111101	11111011	11110111	11101111
FF	11011111	01100001	11000011	01011001	01101101	11011011	10110111	01101111

EK E: Bölüm 4.1'de verilen $had(01_h, 02_h, 04_h, 06_h)$ Matrisinin MDS Test Sonuçları

$had(01_h, 02_h, 04_h, 06_h) = \begin{bmatrix} 01_h & 02_h & 04_h & 06_h \\ 02_h & 01_h & 06_h & 04_h \\ 04_h & 06_h & 01_h & 02_h \\ 06_h & 04_h & 02_h & 01_h \end{bmatrix}$ matrisi geliştirilen yazılım ile test edilmiştir. Buna göre matrisin alt determinantlarının 0 olmadığı aşağıda gösterilmektedir. Tüm işlemler GF(2^8)'de $x^8 + x^4 + x^3 + x + 1$ indirgenemez polinomu ile yapılmıştır. Tüm matris elemanları hexadecimal notasyondadır.

$$\begin{bmatrix} 01 & 06 & 04 \\ 06 & 01 & 02 \\ 04 & 02 & 01 \end{bmatrix} = 01 \quad \begin{bmatrix} 02 & 06 & 04 \\ 04 & 01 & 02 \\ 06 & 02 & 01 \end{bmatrix} = 02 \quad \begin{bmatrix} 02 & 01 & 04 \\ 04 & 06 & 02 \\ 06 & 04 & 01 \end{bmatrix} = 04 \quad \begin{bmatrix} 02 & 01 & 06 \\ 04 & 06 & 01 \\ 06 & 04 & 02 \end{bmatrix} = 06$$

$$\begin{bmatrix} 02 & 04 & 06 \\ 06 & 01 & 02 \\ 04 & 02 & 01 \end{bmatrix} = 02 \quad \begin{bmatrix} 01 & 04 & 06 \\ 04 & 01 & 02 \\ 06 & 02 & 01 \end{bmatrix} = 01 \quad \begin{bmatrix} 01 & 02 & 06 \\ 04 & 06 & 02 \\ 06 & 04 & 01 \end{bmatrix} = 06 \quad \begin{bmatrix} 01 & 02 & 04 \\ 04 & 06 & 01 \\ 06 & 04 & 02 \end{bmatrix} = 04$$

$$\begin{bmatrix} 02 & 04 & 06 \\ 01 & 06 & 04 \\ 04 & 02 & 01 \end{bmatrix} = 04 \quad \begin{bmatrix} 01 & 04 & 06 \\ 02 & 06 & 04 \\ 06 & 02 & 01 \end{bmatrix} = 06 \quad \begin{bmatrix} 01 & 02 & 06 \\ 02 & 01 & 04 \\ 06 & 04 & 01 \end{bmatrix} = 01 \quad \begin{bmatrix} 01 & 02 & 04 \\ 02 & 01 & 06 \\ 06 & 04 & 02 \end{bmatrix} = 02$$

$$\begin{bmatrix} 02 & 04 & 06 \\ 02 & 06 & 04 \\ 06 & 01 & 02 \end{bmatrix} = 06 \quad \begin{bmatrix} 01 & 04 & 06 \\ 02 & 06 & 04 \\ 04 & 01 & 02 \end{bmatrix} = 04 \quad \begin{bmatrix} 01 & 02 & 06 \\ 02 & 01 & 04 \\ 04 & 06 & 02 \end{bmatrix} = 02 \quad \begin{bmatrix} 01 & 02 & 04 \\ 02 & 01 & 06 \\ 04 & 06 & 01 \end{bmatrix} = 01$$

$$\begin{bmatrix} 01 & 02 \\ 02 & 01 \end{bmatrix} = 05 \quad \begin{bmatrix} 06 & 02 \\ 04 & 01 \end{bmatrix} = 0E \quad \begin{bmatrix} 06 & 01 \\ 04 & 02 \end{bmatrix} = 08 \quad \begin{bmatrix} 06 & 04 \\ 02 & 01 \end{bmatrix} = 0E \quad \begin{bmatrix} 01 & 04 \\ 04 & 01 \end{bmatrix} = 11$$

$$\begin{bmatrix} 01 & 06 \\ 04 & 02 \end{bmatrix} = 1A \quad \begin{bmatrix} 06 & 04 \\ 01 & 02 \end{bmatrix} = 08 \quad \begin{bmatrix} 01 & 04 \\ 06 & 02 \end{bmatrix} = 1A \quad \begin{bmatrix} 01 & 06 \\ 06 & 01 \end{bmatrix} = 15 \quad \begin{bmatrix} 04 & 02 \\ 06 & 01 \end{bmatrix} = 08$$

$$\begin{bmatrix} 04 & 01 \\ 06 & 02 \end{bmatrix} = 0E \quad \begin{bmatrix} 02 & 04 \\ 06 & 01 \end{bmatrix} = 1A \quad \begin{bmatrix} 02 & 06 \\ 06 & 02 \end{bmatrix} = 10 \quad \begin{bmatrix} 02 & 04 \\ 04 & 02 \end{bmatrix} = 14 \quad \begin{bmatrix} 02 & 06 \\ 04 & 01 \end{bmatrix} = 1A$$

$$\begin{bmatrix} 04 & 06 \\ 06 & 04 \end{bmatrix} = 04 \quad \begin{bmatrix} 02 & 01 \\ 06 & 04 \end{bmatrix} = 0E \quad \begin{bmatrix} 02 & 01 \\ 04 & 06 \end{bmatrix} = 08 \quad \begin{bmatrix} 04 & 06 \\ 02 & 01 \end{bmatrix} = 08 \quad \begin{bmatrix} 04 & 06 \\ 01 & 02 \end{bmatrix} = 0E$$

$$\begin{bmatrix} 01 & 02 \\ 06 & 04 \end{bmatrix} = 08 \quad \begin{bmatrix} 01 & 02 \\ 04 & 06 \end{bmatrix} = 0E \quad \begin{bmatrix} 02 & 06 \\ 01 & 04 \end{bmatrix} = 0E \quad \begin{bmatrix} 02 & 04 \\ 01 & 06 \end{bmatrix} = 08 \quad \begin{bmatrix} 01 & 06 \\ 02 & 04 \end{bmatrix} = 08$$

$$\begin{bmatrix} 01 & 04 \\ 02 & 06 \end{bmatrix} = 0E$$

KAYNAKLAR

Aoki K., Ichikawa T., Kanda M., Matsui M., Moriai S., Nakajima J., Tokita T., "Camellia a 128-bit block cipher suitable for multiple platforms-design and analysis", SAC 2000. LNCS, vol. 2012, pp. 39–56. Springer, Heidelberg, 2001.

Aras, E., Yücel M.D., "Performance Evaluation of Safer K-64 and S-Boxes of Safer Family", Turkish Journal of Electrical Eng. & Computer Sciences , Vol.9, No. 2, pp.161-175, August, 2001.

Aslan B., "Boole Fonksiyonları ve S-Kutularının Kriptografik Özelliklerinin İncelenmesi ve Ters Haritalama Tabanlı Cebirsel Açıdan Güçlendirilmiş Bir S-kutusu Önerisi", Trakya Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 2008.

Aslan B., Sakallı M. T., Buluş E., "Classifying 8-bit to 8-bit S-boxes based on Power Mappings from the point of DDT and LAT Distributions, International Workshop on the Arithmetic of Finite Fields", WAIFI 2008, Lecture Notes in Computer Science, Vol. 5130, , Springer-Verlag, pp.123-133, 2008.

Aslan B., Sakallı M. T., "Algebraic Construction of Cryptographically Good Binary Linear Transformations", Special Issue on the Design and Engineering of Cryptographic Solutions for Secure Information Systems-Security and Communication Networks, In publication process, 2012.

Barreto P. S. L. M., Rijmen V., "The Khazad legacy-level block cipher", First open NESSIE Workshop, Leuven, 2000.

Barreto P.S.L.M., Rijmen V., "The Anubis block cipher," NESSIE submission, 2000.

Biham E. and Shamir A., "Differential Cryptanalysis of DES-like Cryptosystems", Journal of Cryptology, Vol 4, No 1 pp. 3-72, 1991.

Biham E., Anderson R., Knudsen L., "Serpent: A New Block Cipher Proposal", Fast Software Encryption -FSE98, Lecture Notes in Computer Science, Vol. 1372, pp. 222-238, 1998.

Bogdanov A., Knudsen L.R., Leander G., Paar C., Poschmann A., Robshaw M.J.B., Seurin Y., Vikkelsoe C., " PRESENT: An Ultra-Lightweight Block Cipher". In CHES, 450-466, Springer-Verlag, 2007.

Büyüksaraçoğlu Sakallı F., "Akış Şifrelerin Tasarım Teknikleri ve Güç Analizi", Trakya Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 2011.

Çimen C., Akyıldız E., Akleylek S., "Şifrelerin Matematiği: Kriptografi", ODTÜ Geliştirme Vakfı Yayıncılık / Bilim ve Toplum Dizisi,2009.

Daemen J., Rijmen V., "The Design of Rijndael, AES- The Advanced Encryption Standard", Springer-Verlag, 2002.

ElGamal T., "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms". IEEE Transactions on Information Theory 31 (4): 469–472. DOI:10.1109/TIT.1985.1057074, 1985.

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

FIPS 197, "Advanced Encryption Standard, Federal Information Processing Standard (FIPS)", Publication 197, National Bureau of Standards, U.S. Department of Commerce, Washington D.C, 2001.

FIPS 46-3, "Data Encryption Standard, Federal Information Processing Standard (FIPS)", Publication 46-3, National Bureau of Standards, U.S. Department of Commerce, Washington D.C, 1999.

Junod P., Vaudenay S., "Perfect diffusion primitives for block ciphers – Building efficient MDS matrices", Proceedings of Selected Areas in Cryptology (SAC 2004), Lecture Notes in Computer Science, Vol. 3357, pp. 84-99, Springer-Verlag, 2004.

Kam J. B., Davida G. I., "Structured Design of Substitution Permutation Encryption Networks", IEEE Transactions on Computers, C-28(10):747-753, 1979.

Keliher L., "Linear Cryptanalysis of Substitution-Permutation Networks", Doktora Tezi, 2003.

Koblitz N., "Elliptic curve cryptosystems. Mathematics of Computation", 48 (177), 203–209, 1987.

Kwon D., Kim J., Park S., Sung S.H., Sohn Y., Song J.H., Yeom Y., Yoon E-J., Lee S., Lee J., Chee S., Han D., Hong J., "New block cipher: ARIA", Proceedings of International Conference on Information Security and Cryptology, Lecture Notes in Computer Science, Vol. 2971, pp. 432-445, Springer-Verlag, 2004.

Kwon D., Sung S.H, Song J. H., Park S., "Design of Block Ciphers and Coding Theory", Trends in Mathematics, Vol. 8, n. 1, pp. 13-20, 2005.

Lacan J., Fimes J., "Systematic MDS erasure codes based on vandermonde matrices" ,IEEE Trans. Commun. Lett. 8(9), 570–572, 2004.

Li H., Friggstad Z., "An Efficient Architecture for the AES MixColumns Operation", IEEE International Symposium on Circuits and Systems, pp. 4637-4640, 2005.

Lidl R., Niederreiter H., "Introduction to finite fields and their applications", Revised edition, Cambridge University Press, 1994.

Ling S., Xing C., "Coding Theory: A First Course", Cambridge University Press, 2004.

MacWilliams F.J., Sloane N.J.A., "The theory of error correcting codes", North-Holland, 1977.

Matsui M., "Linear Cryptanalysis Method for DES Cipher", Advances in Cryptology-EUROCRYPT' 93, Springer-Verlag, pp. 386-397, 1994.

Meier, W., Staffelbach, O., "Nonlinearity Criteria for Cryptographic Functions", Advances in Cryptology, Proc. EUROCRYPT'89, pp. 549-562, Springer Verlag, 1989.

Mister, S., Adams, C. M., "Practical S-Box Design", SAC'96- Third Annual Workshop on Selected Areas in Cryptography, pp. 61-76, Queen's Univ., Kingston, Ontario, Canada, 1996.

Nakahara Jr J., Abrahão E., "A new Involutory MDS Matrix for the AES", International Journal of Network Security, Vol. 9, n. 2, pp. 109-116, 2009.

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

Sakallı M. T., "Modern Şifreleme Yöntemlerinin Gücünün İncelenmesi", Trakya Üniversitesi Fen Bilimleri Enstitüsü, Doktora Tezi, 2006.

Sakalli,M. T., Aslan B., Buluş E., Şahin Mesut, A., Büyüksaraçoglu F., Karaahmetoglu, O., "On the Algebraic Expression of the AES S-Box Like S-Boxes", NDT (1),213-227, Prag, 2010.

Shannon C.E., "Communication Theory of Secrecy Systems, Bell System Technical Journal", No. 30, pp. 50-64, 1949.

Stinson D. R., "Cryptography, Theory and Practice", 2nd ed., Chapman & Hall/CRC Press, 2002.

Venkaiah V. C., Srinathan K., Bruhadeshwar B., "Variations to S-box and MixColumn Transformations of AES", Technical Report, Deemed University, 2006.

Webster F., Tavares S. E., "On the Design of S-boxes", Crypto 85, Lecture Notes in Computer Science, Vol. 218, pp. 523-534, Springer-Verlag, 1986.

Yavuzer Aslan F., Sakallı M. T., Aslan B., "AES Şifresinde Kullanılan MixColumns Doğrusal Dönüşümü Üzerine", ELECO 2010, Bursa, 2010.

Yavuzer Aslan F., Sakallı M. T., Aslan B., "Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi", Akademik Bilişim 2012, Uşak, 2012.

Yavuzer Aslan F., Sakallı M. T., Aslan B., Bulut S., "A New Involutory 4×4 MDS Matrix for the AES-like Block Ciphers", The International Review on Computers and Software (IRECOS), Vol 6, N 1, Italy, 2011.

Youssef A.M., Mister S., Tavares S.E., "On the design of linear transformations for substitution permutation encryption Networks", SAC'97, pp. 1–9, 1997.

Z'aba M. R., "Analysis of Linear Relationships in Block Ciphers", Queensland University of Technology, Australia, Doktora Tezi, 2010.

Zhang X., Parhi K.K., "High-Speed VLSI Architectures for the AES Algorithm", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 12, no. 9, pp. 957-967, 2004.

ÖZGEÇMİŞ

Füsun YAVUZER ASLAN, 07 Ekim 1979 yılında Zonguldak'ta doğdu. 1998 yılında Trakya Üniversitesi Fen Edebiyat Fakültesi Fizik Bölümü'nü kazandı. 2009 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı'nda Yüksek Lisans'a başladı. Halen Kırklareli Üniversitesi Lüleburgaz Meslek Yüksekokul'unda serbest öğretim görevlisi olarak çeşitli programlarda ders vermektedir.

TEZ SIRASINDA YAPILAN YAYINLAR

- Yavuzer Aslan F., Sakallı M. T., Aslan B., 2012, "Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi", Akademik Bilişim 2012, Uşak.
- Yavuzer Aslan F., Sakallı M. T., Aslan B., Bulut S., 2011, "A New Involutory 4×4 MDS Matrix for the AES-like Block Ciphers", The International Review on Computers and Software (IRECOS), Vol 6, N 1, Italy.
- Mesut A., Aslan B., Sakallı M. T., Yavuzer Aslan F., 2011, "Genlik Modülasyonu Algoritması ile Görüntü İçerisine Veri Gizleme", Akademik Bilişim 2011, Malatya, 2011
- Yavuzer Aslan F., Aslan B., 2011, "Yer Ve Zamandan Bağımsız Asenkron Öğrenme Merkezi: Luzep", 2. Uluslararası, 6. Ulusal Meslek Yüksekokulları Sempozyumu, Kuşadası, Aydın.
- Yavuzer Aslan F., Sakallı M. T., Aslan B., 2010, "AES Şifresinde Kullanılan MixColumns Doğrusal Dönüşümü Üzerine", ELECO 2010, Bursa.