T.C
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MODERN BİR BLOK ŞİFRE TASARIMI

Selma BULUT
BÜYÜKGÖZE

Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Yrd. Doç. Dr. M. Tolga SAKALLI
EDİRNE-2012

T.C.
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MODERN BİR BLOK ŞİFRE TASARIMI

Selma BULUT BÜYÜKGÖZE

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Bu tez 13 / 01 / 2012 tarihinde aşağıdaki jüri tarafından kabul edilmiştir.

Jüri

Yrd. Doç. Dr. M. Tolga SAKALLI
Danışman
Jüri Başkanı

Doç. Dr. Yılmaz KILIÇASLAN
Üye

Yrd. Doç. Dr. Hilmi KUŞÇU
Üye

T.C
TRAKYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENTSTİTÜSÜ

MODERN BİR BLOK ŞİFRE TASARIMI

Selma BULUT BÜYÜKGÖZE

Yüksek Lisans Tezi
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. M. Tolga SAKALLI

EDİRNE-2012

Yüksek Lisans Tezi
Trakya Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

ÖZET

Bu tez, simetrik şifreleme algoritmalarından blok şifreler ile ilgilidir. Literatürde bulunan AES, ARIA, KHAZAD blok şifrelerinin mimarileri incelenmiş ve bu şifrelerin incelenmesinden elde edilen tecrübe ile AES ve Khazad blok şifrelerine dayanan modern bir blok şifre geliştirilmiştir.

Tezin giriş bölümünde temel simetrik şifreleme teknikleri olan blok ve akış şifreler ve bu şifrelere karşı yapılan kriptanaliz saldırılarının tanımı yapılmıştır.

Tezin 2. bölümünde sonlu cisimler teorisi ile ilgili matematik alt yapı verilmiştir.

3. bölümde AES, ARIA ve KHAZAD blok şifreleme algoritmaları ve tasarım stratejileri incelenmiştir.

4. bölümde blok şifrelerde kullanılan kriptografik yapılar incelenmiş ve geliştirilen şifrede kullanılan kriptografik dönüşümlerin tasarımı yapılmış ve bu dönüşümlerin kriptografik özellikleri irdelenmiştir.

5. bölümde geliştirilen blok şifrenin mimarisi ve örnek test değerleri ile şifrenin çalışması gösterilmiştir.

6. bölümde tezde elde edilen sonuçların değerlendirilmesi ve blok şifrenin daha büyük genişlikte giriş bloğunu işleyecek şekilde tasarlanması için neler yapılabileceği tartışılmaktadır.

Anahtar Sözcükler: Blok Şifreler, S-kutusu tasarımı, Doğrusal Dönüşüm, Anahtar Planlama, AES (Advanced Encryption Standard), ARIA, Khazad.

Yıl: 2012
Sayfa: 110

Msc. Thesis
Trakya University Graduate School of
Natural and Applied Sciences
Department of Computer Engineering

ABSTRACT

This thesis is concerned with block ciphers which fall under the category of symmetric ciphers. In this thesis, three block ciphers found in the literature, AES, ARIA and Khazad and their design strategies, are examined and then, with the experience gained from this examination, a new block cipher is developed.

In the introduction section of the thesis, symmetric key ciphers, which are block ciphers and stream ciphers, and also cryptanalytic attacks against these ciphers, are examined.

In the second chapter of the thesis, a background for the finite field theory is given.

In the third chapter of the thesis, three important block ciphers, AES, ARIA and Khazad, and their design strategies are examined.

In the fourth chapter of the thesis, cryptographic components used in block ciphers and their cryptographic properties are examined and suitable cryptographic components with good cryptographic properties to be used in the developed block cipher are designed. Also, cryptographic properties of these components are given.

In the fifth chapter of the thesis, the architecture of the developed block cipher is given. In addition, a test example for the software implementation of the developed block cipher is presented.

In the last chapter of the thesis, evaluation of the results and expansion of the block cipher to be used with larger block sizes are discussed.

Keywords: Blok ciphers, AES (Advanced Encryption Standard), ARIA, Khazad.

Year: 2012
Page: 110

TEŞEKKÜR

Bu tez için gerçekleştirdiğim araştırmalar sırasında birçok kişinin bana katkısı olmuştur. Bu kişilere burada destekleri için teşekkür etmek isterim.

İlk olarak tez çalışmam sırasında bana yardımlarından ve katkılarından dolayı değerli hocam ve danışmanım Sayın Yrd. Doç. Dr. M. Tolga SAKALLI ve değerli eşi Fatma BÜYÜKSARAÇOĞLU SAKALLI'ya sonsuz teşekkürlerimi sunmak isterim.

Bu tezin izleme komitesinde yer alan ve bana verdikleri destek ve değerli katkılarından dolayı Doç Dr. Yılmaz KILIÇASLAN ve Yrd. Doç. Dr. Hilmi KUŞÇU'ya sonsuz teşekkürlerimi sunarım.

Tez çalışmamın ilerlemesinde büyük emeği olan, bilgi ve paylaşımlarının yanı sıra dostluklarını da esirgemeyerek bana destek veren Öğr. Gör. Bora ASLAN ve değerli eşi Füsun YAVUZER ASLAN'a sonsuz teşekkür ederim.

Çalışma ortamında paylaştıkları dostluk ve çalışmalarıma destek oldukları için başta Kırklareli Üniversitesi Pınarhisar Meslek Yüksek Okulu Müdürü Bahtiyar DURSUN ve tüm çalışma arkadaşlarıma teşekkür ederim.

Eski mesai arkadaşım olmasına rağmen desteklerini hala hissettiğim Öğr. Gör. Ali MÜLAYİM 'e teşekkür ederim.

Bu tezin ortaya çıkmasında desteğini hiç esirgemeyen sevgili eşim Simge BÜYÜKGÖZE ve değerli ailesine teşekkür ederim.

Yüksek Lisans sürecinde beni sürekli motive eden ve her türlü desteği veren değerli arkadaşlarım Emel KAHRAMAN ve Gülçin CANDEMİR'e teşekkür ederim.

Son olarak da tez süresince beni motive eden, ellerinden gelen yardımı esirgemeyen öğrencim Seda Örnek ve diğer öğrencilerime teşekkür ederim.

Ocak 2012 Selma BULUT BÜYÜKGÖZE

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

1. GİRİŞ

Teknolojinin geliştiği ve gelişmeye çok hızlı bir şekilde devam edeceği bu bilgi çağında, bilgisayarlar ve internet hayatımızın vazgeçilmez birer unsuru haline gelmiştir. Böyle bir ortamda, bilginin korunması ve bir noktadan diğerine güvenli bir şekilde iletilmesi çok büyük önem kazanmıştır. Verilerin güvenli bir biçimde aktarımı ve elde edilmesi için, kriptografi bilimi aracılığı ile çeşitli şifreleme ve deşifreleme sistemleri, algoritmaları ve yaklaşımları geliştirilmektedir. Bu yeni oluşturulan şifreleme algoritmaları ve yapıları gelişen teknolojiye uygun bir şekilde tasarlanmalı ve çağın ihtiyaçlarına cevap verebilmelidir (Kayış H., 2006).

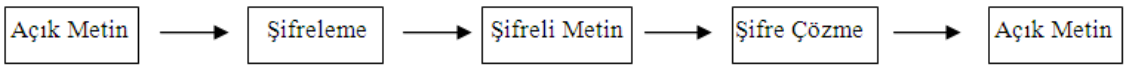
1.1 Şifreleme (Kriptografi)

Yunanca gizli anlamına gelen "kript" ve yazı anlamına gelen "graf" kelimelerinden türetilen kriptografi anlaşılır bir mesajı anlaşılmaz hale dönüştürme ve anlaşılmaz mesajı tekrar anlaşılır hale geri dönüştürme işlemlerini kapsayan bir bilimdir (Aslan B., 2008).

Kriptolojinin (kriptografi + kriptanaliz) tarihi 4000 yıl öncesine hiyeroglif kodları duvarlara yazan Mısırlılara kadar dayanmaktadır (Diffie ve M. Hellman., 1976). O zamanlardan bu zamana birçok kripto sistem geliştirilmiş ve kullanılmıştır. Geliştirilen eski şifrelerin çoğu bugünkü uygulamalarda kullanılabilecek kadar yeterince güçlü olmadığı görülmüştür (Knudsen L. R., 1994, 1994).

Güvenlik için kullanılan yaygın bir yaklaşım olan kriptografi biliminde, karmaşık işlemlerden oluşan matematiksel fonksiyonlar olan algoritmalar, şifreleme (encryption) ve şifre çözme (decryption) için kullanılırlar. Şifreleme, açık metni anlaşılamayacak bir forma dönüştürme işlemidir. Bu işlem bir matematiksel fonksiyon ve bir anahtar veya anahtar çiftinin biri kullanılarak yapılır. Şifre çözme (deşifreleme)

ise, şifrelenmiş mesajı, şifrelemede kullanılan fonksiyonun tersini ve bir anahtar veya anahtar çiftinin diğerini kullanarak açık metne dönüştürme işlemi olarak tarif edilebilir (Başkök M. D., 2007). Bu işlemleri gösteren temel kriptografik mekanizma Şekil 1.1'de verilmiştir.



Şekil 1.1. Şifreleme ve Deşifreleme Yöntemi

Şifrelenmemiş bir bilgiye açık metin (plaintext) denir. Açık metin, bir insanın okuyabileceği bir yazı ya da bir bilgisayarın anlayabileceği çalıştırılabilir (.exe, .com) bir program ya da bir veri dosyası (.txt) olabilir. Bir kripto algoritması kullanılarak, herkesin okuyamayacağı bir şekilde kodlanmış bilgiye ise şifreli metin (ciphertext) denir (Başkök M. D., 2007).

Verinin anlaşılmaz hale dönüştürme işlemi şifreleme ile sağlanır. Şekil 1.1, şifreleme ve deşifreleme işlemlerini göstermektedir. Verilen şekle göre veri gönderilmeden önce açık metin şifreleme işlemine girerek şifreli metin oluşur. Veri gönderildikten sonra alıcı tarafında şifreli metin şifre çözme işlemlerine tabi tutularak açık metin elde edilir.

Modern şifreleme algoritmaları genel olarak üç guruba ayrılmaktadır: simetrik, asimetrik ve hash algoritmaları. Simetrik algoritmalar şifreleme ve deşifreleme işlemlerinde aynı anahtarı (gizli anahtarı) kullanır. Diğer yandan kriptografi de blok şifreleme ve akan (stream) şifreleme olmak üzere iki temel simetrik algoritma tipi vardır. Bunlardan blok şifreleme algoritmaları orijinal metni veya şifreli metni bloklara bölerek şifreleme/deşifreleme işlemini yaparken; akan şifreleme algoritmaları bir bit veya byte üzerinde şifreleme ve deşifreleme işlemlerini gerçekleştirirler. Hash algoritmaları ise verinin sıkı bir temsilini oluşturmak için kullanılırlar ve kimlik

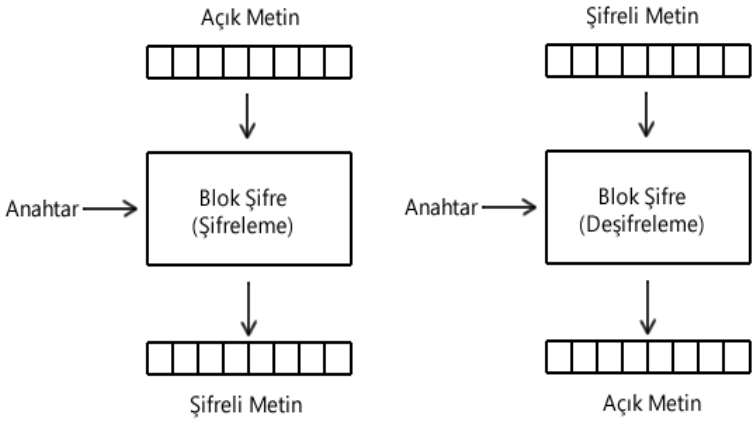
denetimin sağlanmasında büyük rol oynarlar (Aslan B., 2008). Tablo 1.1, bu üç gruptaki şifreleme algoritmalarına örnekler göstermektedir.

Tablo 1.1. Üç gruba göre şifreleme algoritmalarına örnekler (Aslan B.,2008)

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

1.2 Blok Şifreler

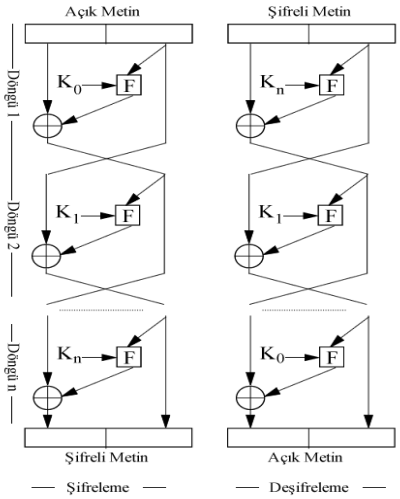
Blok şifreleme algoritmaları açık metni sabit uzunluklu blok adı verilen bit grupları halinde işler. Bloklar bir anahtar aracılığı ile şifrelenerek şifreli metin elde edilir. Deşifreleme işleminde yine aynı anahtar kullanılarak şifreli metin açık metin haline dönüştürülür.



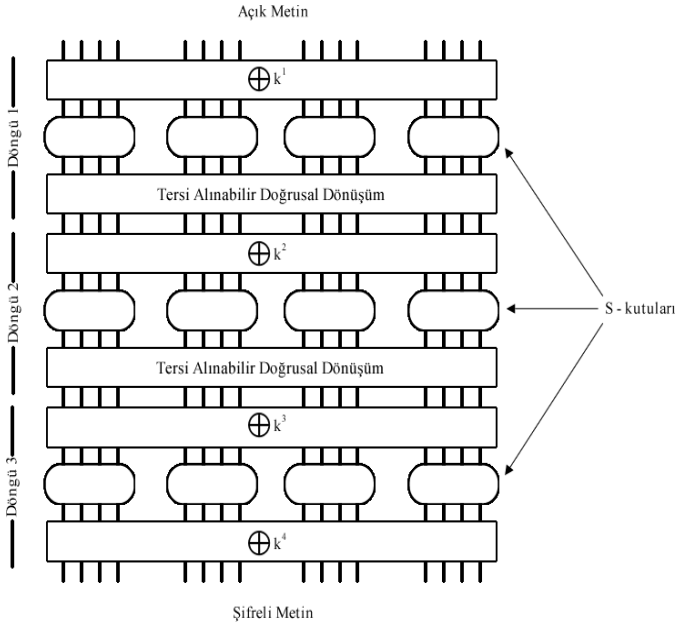
Şekil 1.2. Blok Şifrede Şifreleme ve Deşifreleme İşlemi

Blok şifreler, Shannon'un (Shannon C.E., 1949) önerdiği karıştırma (confusion) ve yayılım (diffusion) tekniklerine dayanır. Karıştırma, şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılım açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Karıştırma ve yayılım, sırasıyla yer değiştirme kutuları (S-kutuları) ve doğrusal dönüşüm işlemleri ile gerçekleştirilir.

Bir blok şifrenin tasarımında Feistel ve SPN (Substitution Permutation Networks) mimarisi olmak üzere iki mimari kullanılır. DES (Data Encryption Standard) (FIPS 46-3, *Data Encryption Standard*, Federal Information Processing Standard (FIPS), 1999) ve AES (Advanced Encryption Standard) (Schneier B., 1996) sırasıyla Feistel ve SPN mimarisine örnek olarak verilebilir. Bu iki mimaride şifreleme ve deşifreleme işlemleri döngü adı verilen şifreleme adımlarının birleşmesi ile gerçekleştirilir. Diğer yandan bu mimarilerin arasındaki en temel fark döngü içerisinde bir bloğun işlenmesinde ortaya çıkmaktadır. Örneğin, Feistel mimarisinde bir döngüde o anki bloğun yarısı işlenirken SPN mimarisinde o anki bloğun tümü işlenir. Feistel mimarisinin bir avantajı Feistel ağı döngü fonksiyonunun tersi alınabilir olma ihtiyacının olmamasıdır. Bu da şifrenin tasarımında büyük bir esneklik sağlar (Sakallı F. B., 2011) Buna ek olarak bir blok şifrenin genel tasarımında bir döngü içinde yer değiştirme S-kutuları ile, yayılım ise doğrusal dönüşüm veya dönüşümler ile sağlanır ve her döngüde döngünün sonunda anahtar planlamadan gelen o döngü için elde edilen bir anahtar değeri ile XOR'lama işlemi gerçekleştirilir. Şekil 1.3 ve Şekil 1.4'te bu mimarilere örnekler verilmiştir.



Şekil 1.3. Feistel ağı



Şekil 1.4. 16 bit giriş-çıkışlı 3 döngülük bir örnek SPN ağı

Blok şifrelerin gücünü belirleyen bazı faktörler aşağıdaki gibidir:

- **Anahtar**: Blok şifrelerde anahtarın uzunluğu saldırılara karşı güçlü olacak şekilde seçilmelidir. DES algoritması 56-bit anahtar uzunluğu kullanırken, AES algoritması 128, 192, 256 bit anahtar uzunluklarını seçenekli olarak

sunmaktadır. Bunun sayesinde şifrenin kaba kuvvet (brute-force) saldırısına karşı dayanıklılığı arttırılmaktadır.

- **Döngü sayısı**: Blok şifreleme algoritmalarında döngü sayısı iyi seçilmelidir. Böylelikle doğrusal dönüşüm ve yer değiştirme işlemleri ile şifreleme algoritması daha da güçlenmektedir. Ayrıca şifrenin karmaşıklığının arttırılmasında da çok önemli bir etkendir. Böylelikle saldırılara karşı açık metin iyi derecede korunabilir. Döngü sayısını belirleyebilmek için belirli bir teorik hesaplama olmamasına rağmen Lars Knudsen'e göre kabaca döngü sayısı $r \geq \frac{d.n}{w}$ ifadesindeki gibi olmalıdır (Knudsen L. R., 2000). Bu ifade de *r* döngü sayısını, *d* yer değiştirme durumda bir word'u almak için gerekli maksimum döngü sayısını, *n* blok genişliğini ve *w* ise tüm şifrede yer değiştirme durumuna giriş olan minimum word genişliğini temsil etmektedir. Tablo 1.2'de verilen şifreleme algoritmalarının döngü sayılarının (Knudsen L. R., 2000) olması gereken değerleri verilmiştir. Diğer yandan döngü sayısını belirleyen en önemli unsur şifreye olan saldırılardır. Yapılan saldırıların başarımına göre bir blok şifre için döngü sayısı belirlenebilir.

Tablo 1.2. Döngü sayılarına göre bazı şifreleme algoritmaları

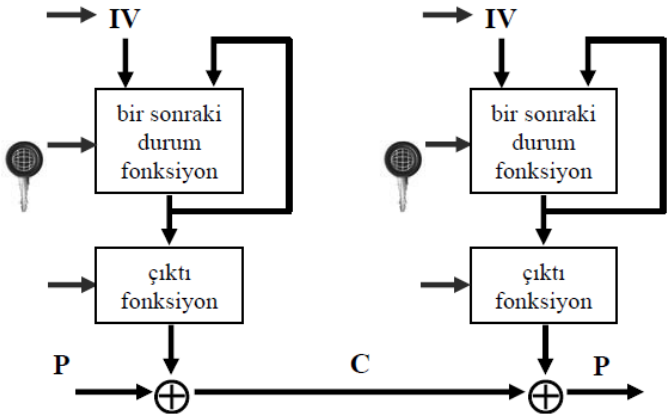
Algoritma	Döngü Sayısı	Lars Knudsen'e göre olması gereken döngü sayısı
DES	16	21
IDEA	8	8
BlowFish	16	16
AES	10	16

- **S-kutuları (Yer değiştirme kutuları):** Blok şifreleme algoritmalarının en önemli elemanı S-kutularıdır. Algoritmanın tek doğrusal olmayan elemanıdır. Bu yüzden iyi bir S-kutusu seçimi şifrenin karmaşıklığını doğrudan etkiler.

1.3 Akış Şifreler

Akış şifreleme (stream ciphers) algoritmaları, orijinal veri olarak bit dizilerini almakta, çıktı olarak da bit dizileri üretmektedirler.

Akış şifreler tek bir seferde bit/word tabanlı işlem yapan sözde rastgele diziler (pseudo random sequences) oluşturan anahtarlı üreteçlerdir (Şekil 1.5). Birçoğu her bir saat çevriminde (clock cycle) bir bit çıktı üretir. Bununla birlikte byte ya da daha büyük birimleri tek seferde şifreleyen akış şifre algoritmaları da vardır. Akış şifre algoritmaları genellikle blok şifre algoritmalarından hızlıdır ve donanım gereksinimleri daha azdır.

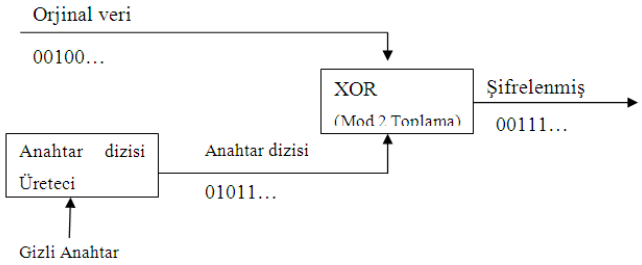


Şekil 1.5. Akan Şifreleme Algoritması (Saran N.,2009)

Akan şifreleme işleminde ilk olarak gizli anahtar ve üreteç yardımı ile bir anahtar dizisi oluşturulmakta daha sonra bu dizi ve girdi mesajının her bir biti ayrı ayrı XOR (mod 2 toplama) işlemine tabi tutulmaktadır. Çözme işlemi de aynı şekilde

gerçekleştirilmektedir. Akış şifreleme algoritmalarının güvenliği anahtar dizisi üreteci tarafından yaratılan dizinin ne kadar rastsal olduğu ile ilişkilidir. Bu nedenle anahtar dizisi olarak tamamıyla rastsal verinin kullanılması en ideal durumdur. Gerçek uygulamalarda tam anlamı ile rastsal anahtar dizisi yaratılması imkânsız olduğundan, anahtar üreteci ve onun girdisi olan gizli anahtar yardımı ile anahtar dizileri yaratılmaktadır. Diğer bir deyişle, bir akış şifreleme algoritmasının en önemli bileşeni kullandığı anahtar dizisi üretecidir. Yaratılan anahtar dizisinin kendini tekrarlamaması ve sonraki anahtar bitlerinin öncekiler yardımı ile elde edilememesi anahtar üreteçlerinin sağlaması gereken önemli özelliklerindendir (Shamir A., 2004).

Aşağıdaki şekil akış şifreleme algoritmasını göstermektedir.



Şekil 1.6. Akış Şifreleme Algoritması

Akış şifreleme algoritmaları, açık metnin bir karakterini bir seferde zamanla değişen bir şifreleme fonksiyonu kullanarak ayrı ayrı şifreler. Akış şifreler eş zamanlı ve eş zamansız olmak üzere temelde ikiye ayrılırlar. Eş zamanlı akış şifrelerde anahtar dizisi, açık metin ve gizli anahtardan bağımsız olarak üretilir. Her iki şifreleme tipi de sonlu durum otomatıdır ancak eş zamansız akış şifrelerde anahtar dizisi, sabit uzunluktaki bir önceki şifreli metinlerin ve anahtarın bir fonksiyonu ile elde edilir.

Akış şifrelerin kullanımı için Shamir (Shamir A., 2004) ve Babbage (Babbage S., 2004) iki tür uygulama önermektedir:

1. Yazılımsal olarak yüksek hızda çalışan uygulamalar (örn. yönlendiriciler (routers) için).
2. Donanımsal olarak kısıtlı kaynaklarda (örn. mantık kapıları, güç tüketimi, vs.) çalışabilen uygulamalar (örn. RFID için) (Saran N., 2009).

Avrupa Komisyonu 6. Çerçeve Programı tarafından desteklenen ve 4 yıl süren Avrupa Kriptografide Mükemmeliyet Ağı – European Network of Excellence for Cryptology (ECRYPT) – tarafından yeni bir akan şifre tasarımı projesi (eStream) çağrısı yapılmıştır (ECRYPT, 2008). Bu çalışmada algoritmalar yazılıma ve donanıma yönelik tasarımlar olarak iki bölümde incelenmiştir. Yazılımsal tasarımlardan yüksek hızda çalışması, donanımsal tasarımlardan da sınırlı kaynaklarla çalışması beklenmiştir. Karşılaştırmalar hem önerilen şifreler arasında hem de AES'le yapılmıştır. İlk çağrı Ekim 2004'te yapılmış ve Nisan 2005'de 33 algoritma önerilmiştir. Eylül 2008'e kadar süren uzun çalışmalar sonunda 7 algoritma finalist olarak ilan edilmiştir.

Tablo 1.3. eStream Projesi Finalistleri

Profil 1 (Yazılım)	**Profil 2 (Donanım)**
Sosemanuk	Trivium
Salsa20/12	MICKEY v2
Rabbit	Grain v1
HC-128	

Akış şifrelerin ayrıldığı diğer bir kategori de bu şifrelerin word tabanlı ya da bit tabanlı olup olmamaları ile ilgilidir. HC–256 word tabanlı iken Trivium bit tabanlı bir akış şifredir.

Akış şifreleme algoritmaları, hem hızlı olmaları hem de basit bir yapıya sahip olmaları nedeni ile birçok uygulama alanına sahiptir. 256 byte uzunluğuna kadar gizli anahtarlarla çalışabilen RC4, akış şifreleme algoritmaları içinde en yaygın olarak kullanılanıdır (Sakallı M. T., 2006).

1.4 Kriptanaliz

Kriptanaliz kısaca şifre kırma bilimi olarak ifade edilebilir. Diğer bir deyişle açık metni ya da anahtarı elde etme bilimidir. Bir şifreleme sisteminin zayıf ve güçlü yanlarını ortaya çıkarmak için kullanılabileceği gibi bir şifrenin kırılarak açık metni ya da anahtarı elde etme şeklinde kötü niyetli olarak da kullanılabilir.

Bir kriptografik yapının tasarımı için ilk ve en önemli kural Kerckhoffs prensibidir. Bu prensibe göre: "Bir kriptografik yapının güvenliği sadece anahtarın gizliliğine bağımlı olmalıdır. Algoritmanın gizliliğine bağımlı olmamalıdır". Diğer bir deyişle kriptografik yapının bileşenleri herkese açık olmalıdır. Buna ek olarak kriptanaliz saldırısı için saldırganın ya da kriptanalistin sahip olabileceği veriler olabilir. Bu sahip olabileceği verilere göre saldırı modellerinden birini seçebilir. Bu saldırı modellerinden en yaygın olanları aşağıda verilmiştir (Sakallı M. T., 2006):

1. *Sadece şifreli metin saldırısı (Ciphertext only attack): Sadece şifreli metin saldırısında, saldırgan sadece bazı şifreli metinlerin erişimine sahiptir. O şifreli metne ilişkili anahtarı ve açık metni elde etmeye çalışır. Saldırganın bu saldırı için sadece şifreli metinlere ihtiyacı vardır. Bir şifreli mesajın düşman tarafından çözülmesini etkisiz hale getirmek için bir şifrenin bu saldırı tipine karşı oldukça dirençli olması gerekir.*
2. *Bilinen açık metin saldırısı (Known plaintext attack): Bilinen açık metin saldırısında saldırgan şifreli metinlere ve bazı açık metin ve şifreli metin çiftlerine erişime sahiptir.*
3. *Seçilmiş açık metin saldırısı (Chosen plaintext attack): Seçilmiş açık metin saldırısı bilinen açık metin saldırısına benzerdir. Fakat açık metin şifreli metin çiftleri saldırgan tarafından seçilmiştir.*
4. *Seçilmiş şifreli metin saldırısı (Chosen ciphertext attack): Seçilmiş şifreli metin saldırısı seçilmiş açık metin saldırısına benzerdir. Bu saldırıdan farklı olarak saldırgan bazı şifreli metinleri seçebilir. Buna ek olarak bir açık metin ve şifreli metin çifti elde edebilmek için deşifreleme yapabilir.*

Kriptanalitik saldırılar yukarıda verilen saldırı modellerinden birini kullanabilir. Önemli bazı kriptanalitik saldırı tipleri doğrusal kriptanaliz, diferansiyel kriptanaliz,

kesik diferansiyel kriptanaliz, imkânsız diferansiyel kriptanaliz, çoklu set saldırıları, interpolasyon saldırısı gibi cebirsel saldırılar, boomerang saldırısı, kare (rectangular) saldırısı, yan kanal saldırısı şeklinde verilebilir (Z'aba M. R., 2010). Bir şifrenin gücü değerlendirilirken verilen saldırılar geniş anahtar arama saldırısı (exhaustive key search) ile karşılaştırılır. Geniş anahtar arama saldırısı tüm olası anahtarları deneyerek doğru olanı bulmaya dayanır. Dolayısıyla şifrenin erişilebilecek maksimum gücü tüm anahtarların denenmesi için gerekli hesaplama gücü ile ilişkilidir.

Tezin Önemi ve Gerekçesi: Bu tez, blok şifreler üzerine yapılan incelemelerden elde edilen sonuçlara göre yazılım tabanlı 64-bit girişli 64-bit çıkışlı ve 64-bit anahtar kullanan bir blok şifre geliştirilmesi üzerinedir. Literatürde bulunan önemli blok şifreleme algoritmalarından AES, ARIA, Khazad şifreleme algoritmaları incelenmiş ve bu şifrelerden edinilen tecrübe ile AES ve Khazad blok şifresi tabanlı modern bir blok şifre geliştirilmesi hedeflenmiştir. Özellikle AES şifreleme algoritmasının S-kutusu incelenmiş ve bu S-kutusunun diğer kriptografik özelliklerini değiştirmeden polinomsal ifadesi daha iyi bir S-kutusunun kullanılabileceği anlaşılmıştır. Bunun yanında incelenen şifreleme algoritmalarının doğrusal dönüşümlerine bakıldığında AES'in 4×4 MDS byte matris, KHAZAD'ın 8×8 involutif (tersi kendisi, $A = A^{-1}$) MDS byte matris, ARIA'nın 16×16 boyutunda involutif ikili matris kullandığı görülmüştür. Geliştirilecek şifrede de AES şifresinde kullanılan 4×4 MDS matris yerine 8×8 involutif bir MDS matris kullanmanın, şifreleme algoritmasını güçlendirilebileceği ve şifreleme ile deşifreleme arasındaki performans farkını yok edeceği açıktır. Yayılım katmanlarında kullanılan dönüşümlerden için kriptografik bir kriter olan dallanma sayısı (branch number) değeri AES için 5, KHAZAD için 9, ARIA blok şifresi içinse 8'dir. Dolayısıyla çalışmamızda geliştirilen blok şifrede kullanılacak doğrusal dönüşümün yüksek dallanma sayısı değerine sahip olması amaçlanmıştır. Tez çalışmasında doğrusal dönüşüm olarak involutif bir MDS matris kullanıldığından bu değer 9'dur. AES şifresinin incelemesi sonucu gözlenen anahtar planlama evresindeki zayıflıklar da geliştirilen şifrede yok edilmeye çalışılmıştır. Buna ek olarak 64-bit giriş 64-bit çıkışlı geliştirilen şifre kullanılarak 128-bit girişli 128-bit çıkışlı ve 128-bit anahtar kullanan bir blok şifre geliştirilebilir.

2. SONLU CİSİMLER TEORİSİ

Sonlu cisimler teorisi, hata düzeltme kodları, sayısal sinyal işleme ve kriptografi gibi alanlarda kullanılan bir teoridir. Bu bölümde tez esnasında kullanılan ve tezin anlaşılması açısından önemli olan bazı matematiksel tanım ve teorilere yer verilecektir. Bu tanım ve teorilerin ispatlarına (Stinson D. R., 2002) (Ling S., Xing C., 2004) ve (Lidl R. ve H., 1994) kaynaklarından elde edilebilir.

Tanım 2.1: a, b tamsayı ve m pozitif tamsayı olsun. Eğer m, $b-a$'yı bölüyorsa $a \equiv b \pmod{m}$ şeklinde yazılabilir. $a \equiv b \pmod{m}$ ifadesine denklik denir ve a, b 'ye mod m'ye göre denktir denir. Tamsayı m 'ye de modulo denir.

Aritmetik modulo m : Z_m $\{0,1,...,m\text{-}1\}$ kümesi üzerinde iki işlem toplama ve çarpma tabanlı tanımlanır. Z_m'de toplama ve çarpma işlemleri bilinen toplama ve çarpma işlemleridir. Bunun yanın sonuçlar modulo m 'ye göre indirgenir.

Z_m'de toplama ve çarpmanın tanımları bilinen bir çok aritmetik kuralı sağlar. Bu aksiyomlar aşağıdaki gibi listelenebilir:

1. Toplamada kapalılık: $a,b \in \mathrm{Z}_m$ için $a+b \in \mathrm{Z}_m$
2. Toplamada değişme: $a,b \in \mathrm{Z}_m$ için $a+b=b+a$
3. Toplamada geçişme: $a,b \in \mathrm{Z}_m$ için $(a+b)+c=a+(b+c)$
4. Toplama etkisiz eleman 0: $a,b \in \mathrm{Z}_m$ için $a+0=0+a=a$
5. $a \in \mathrm{Z}_m$ için a'nin toplamaya göre tersi $m-a$ 'dır.
6. Çarpmada kapalılık: $a,b \in \mathrm{Z}_m$ için $a.b \in \mathrm{Z}_m$
7. Çarpmada değişme: $a,b \in \mathrm{Z}_m$ için $a.b=b.a$
8. Çarpmada geçişme $a,b \in \mathrm{Z}_m$ için $(a.b).c=a.(b.c)$
9. Çarpma işleminde etkisiz eleman 1'dir. $a \in \mathrm{Z}_m$ için $a.1=1.a=a$
10. Dağılma özelliği sağlanır. $a,b,c \in \mathrm{Z}_m$ olmak üzere $(a+b).c=ac+bc$ ve $a.(b+c)=ab+ac$

Tanım 2.2: Yukarıda verilen aksiyomlardan 1, 3, 4, 5 aksiyomlarını sağlayan Z_m cebirsel yapısına **grup** denir. Eğer bahsedilen özelliklerle beraber aksiyom 2'yi de sağlıyorsa **abelian grup** adını alır.

Tanım 2.3: Verilen aksiyomlardan 1, 2,.., 10 aksiyomlarını sağlayan Z_m cebirsel yapısına **halka** denir. Örnek olarak tamsayılar, reel sayılar ve karmaşık sayılar halka örneklerindendir.

Tanım 2.4: Verilen aksiyomlara ek olarak toplama ve çarpma işlemine göre ters alma işlemini sağlayan cebirsel yapıya **cisim** adı verilir.

Örnek 2.1. Z_4'ü düşünelim. Bu yapının sonlu bir cisim oluşturup oluşturmadığını inceleyelim. $Z_4 = \{0,1,2,3\}$ elemanlarına sahip bir küme olduğuna göre çarpma ve toplama işlemleri için aşağıdaki tablo elde edilebilir:

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

X	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	0	2
3	0	3	2	1

Tablolar incelendiğinde Z_4 (mod 4 işlemine göre kalanların oluşturduğu küme) bir cisim oluşturmaz. Bunun nedeni olarak bir cismin oluşabilmesi için gereken ve yukarıda verilen tüm aksiyomlar sağlanmaz. Çünkü 2 değerinin çarpma işlemine göre tersi yoktur.

Örnek 2.2. Z_5'i düşünelim. Bu yapının sonlu bir cisim oluşturup oluşturmadığını inceleyelim. $Z_5 = \{0,1,2,3,4\}$ elemanlarına sahip bir küme olduğuna göre çarpma ve toplama işlemleri için aşağıdaki tablo elde edilebilir:

+	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

x	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 ($\mod 5$ işlemine göre kalanların oluşturduğu küme) Tanım 2.4 gereği bir cisim oluşturur. Yukarıda verilen toplama ve çarpma tabloları Z_5 de toplama ve çarpma işlemleri için bunu doğrulamaktadır.

Tanım 2.5. Sonlu bir cisim F için, $\alpha \in F$ elemanının derecesi $\alpha^m = 1$ olacak şekilde en küçük pozitif m sayısıdır.

Teorem 2.1. Her tamsayı $n > 1$ asal sayıların bir ürünü olarak yazılabilir. Diğer bir deyişle her tamsayı $n = {p_1}^{m_1}.{p_2}^{m_2}...{p_r}^{m_r}$ şeklinde ifade edilebilir ve buna n'nin cononical faktorizasyonu adı verilir.

Not: İfadedeki m_i'ler pozitif ve $p_1 < p_2 < ... < p_r$ şeklindedir.

Teorem 2.2. $m = \prod_{i=1}^{n} {p_i}^{m_i}$ p_i'ler farklı tamsayılar ve $m_i > 0,\ 1 \le i \le n$ olmak üzere $\phi(m) = \prod_{i=1}^{n} ({p_i}^{e_i} - {p_i}^{e_i - 1})$ şeklindedir. $\phi(m)$'e Euler Phi fonksiyonu denir ve bu fonksiyon 1 ile m arasında m ile aralarında asal olan sayıların sayısını verir.

Teorem 2.3. Bir Z_p kümesinin, eğer p asal sayı ise, bir elemanının en yüksek derecesi $\phi(p) = p - 1$'dir ve bu dereceye sahip elemana ilkel (primitive) eleman adı verilir. Buna ek olarak herhangi bir elemanın derecesi $p - 1$ değerini böler.

Teorem 2.4. p asal sayı ve $\alpha \in Z_p^*$ olsun. O zaman sadece ve sadece $\frac{p-1}{q}$ olacak şekilde tüm asal q'lar için $\alpha^{\frac{p-1}{q}} \neq 1 \bmod p$ şeklinde ise $\alpha \bmod p$'ye göre ilkel elemandır.

Teorem 2.5. Eğer p bir asal sayı ve $\alpha \bmod p$'ye göre ilkel eleman ise $\beta \in Z_p^*$ ve $\beta = \alpha^i, 0 \leq i \leq p-2$ şeklinde yazılabilir. $\beta = \alpha^i$'nin derecesi $\frac{p-1}{\gcd(p-1,i)}$ şeklinde elde edilebilir.

Verilen teorileri kullanarak Örnek 2.3'de Z_{11} sonlu cismi için ilkel elemanlar gösterilmektedir.

Örnek 2.3. Z_{11} de kaç tane ilkel eleman vardır ve bu ilkel elemanları bulunuz. İlk önce bir ilkel eleman elde edelim:

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

Yukarıda gösterildiği gibi 2 elemanı Z_{11}'de ilkel bir elemandır ve $\gcd(10,i) = 1$ olacak şekilde i değerlerine sahip 2^i değerleri ilkel eleman olacaktır. $\gcd(10,i)$ değerleri 1, 3, 7, 9 olacağından diğer ilkel elemanlar; $2^1 \bmod 11 = 2$, $2^3 \bmod 11 = 8$,

$2^7 \bmod 11 = 7,\ 2^9 \bmod 11 = 6,\ 2^7 \bmod 11 = 7,\ 2^9 \bmod 11 = 6$ şeklinde hesaplanabilir. Dolayısıyla Z_{11}'te ilkel elemanlar 2, 6, 7 ve 8 dir.

Tanım 2.6. F bir cisim olsun. Küme $F[x] := \left\{ \sum_{i=0}^{n} a_i x^i : a_i \in F, n \geq 0 \right\}$ F üstüne polinom halka olarak isimlendirilir. $F[x]$'in bir elemanına F üstüne bir polinom adı verilir. Bir $f(x) = \sum_{i=0}^{n} a_i x^i$ polinomu için tamsayı n $f(x)$'in derecesi olarak adlandırılır ve $\deg(f(x))$ ile tanımlanır. Bunun ötesinde $n.$ dereceden $f(x) = \sum_{i=0}^{n} a_i x^i$ sıfır olmayan polinomu için $a_n = 1$ ise monic olarak isimlendirilir. Bir $f(x)$ polinomu $\deg(g(x)) < \deg(f(x))$ ve $\deg(h(x)) < \deg(f(x))$ olacak şekilde $f(x) = g(x).h(x)$ şeklinde yazılabiliyorsa indirgenebilir aksi halde pozitif dereceli $f(x)$ polinomu indirgenemez polinom olarak adlandırılır.

Teorem 2.6. $f(x)$ derecesi 1'den büyük olmak üzere bir F cisminin üzerine bir polinom olsun. Yani $f(x)$ polinomunun elemanları F cisminin elemanlarından oluşsun. O zaman $F[x]/f(x)$ toplama ve çarpma işlemi ile birlikte bir halka oluşturur. Bunun ötesinde $f(x)$ indirgenemez bir polinom ise $F[x]/f(x)$ bir cisim oluşturur.

Yukarıdaki Teoreme göre $F[x]/f(x)$ mod $f(x)$'e göre indirgeme sonucu oluşan sistemi ifade etmektedir. Sonuç olarak bir indirgenemez polinoma göre mod alma işlemi sonucunda cisim elde edilebilir ve bu cisimler F_{p^n} ya da $GF(p^n)$ cisimleri olarak isimlendirilir ve p^n eleman içerir.

Örnek 2.4: $Z_2[x]/(1+x+x^3)$ cisminin elemanlarını yani F_{2^3} cismini oluşturmak için x'in üslerini $\bmod(1+x+x^3)$ işlemi ile elde edelim.

$$x^0 \equiv 1$$
$$x^1 \equiv x$$
$$x^2 \equiv x^2$$
$$x^3 \equiv x+1$$
$$x^4 \equiv x^2+x$$
$$x^5 \equiv x^2+x+1$$
$$x^6 \equiv x^2+1$$
$$x^7 \equiv 1$$

Aynı tabloyu üç boyutlu vektör $\alpha=[0,1,0]$ kullanılarak da aşağıdaki gibi elde edilebilir.

$$\alpha^0 = 1$$
$$\alpha^1 = \alpha$$
$$\alpha^2 = \alpha^2$$
$$\alpha^3 = \alpha+1$$
$$\alpha^4 = \alpha^2+\alpha$$
$$\alpha^5 = \alpha^2+\alpha+1$$
$$\alpha^6 = \alpha^2+1$$
$$\alpha^7 = 1$$

Böylece α 'nin ilk 7 kuvveti F_{2^3}'te birbirinden farklıdır ve sadece F_{2^3}'te birbirinden ve sıfırdan farklı 7 eleman vardır.

Örnek 2.5: $Z_2[x]/(1+x+x^3)$ cisminde $a.b=(110).(111)$ sonucunu elde edelim. $a=\alpha^4, b=\alpha^5$ şeklinde yazılabileceğinden $a.b=\alpha^4.\alpha^5=\alpha^9=\alpha^2.\alpha^7=\alpha^2.1=(100)$ olarak elde edilebilir (α'nın üsleri mod 7'ye göre indirgeme yapılarak elde edilir çünkü $\alpha^7=1$'dir).

Örnek 2.5' de bir cisimde iki polinomun çarpımı için bir tablo oluşturma yolu ile elde edilmesi gösterilmiştir. Diğer yandan bilgisayar uygulamalarında iki polinomun çarpımının sonucunun elde edilebilmesi için etkin bir algoritma aşağıdaki gibi verilebilir:

1. x ile çarpma çarpılacak polinomun 1 bit sola ötelemesi demektir,

2. Ancak bir önceki sonucun en anlamlı biti bir ise 1 bit sola öteleme işleminden sonra elde edilen değer indirgenemez polinomun kalanı ile XOR işlemine tabi tutulmalıdır.

Örneğin bir değerin x^2 ile çarpımı söz konusu ise o zaman değerin iki defa arka arkaya x ile çarpımı ile istenilen sonuç elde edilebilir. Diğer yandan eğer $x^2 + x$ ile çarpım söz konusu ise x^2 ve x ile çarpım sonuçları elde edilir. Daha sonra bu elde edilen değerler XOR işlemine tabi tutulur.

Örnek 2.6: $P_1 = (x^4 + x^2 + x)$ ile $P_2 = (x^7 + x^6 + x^4 + x + 1)$ polinomlarının $GF(2^8)$ de modulo $(x^8 + x^4 + x^3 + x + 1)$ kullanılarak çarpımını elde edelim.

Üsler	İşlem	Yeni Sonuç	İndirgeme
$x^0 \otimes P_1$		$(x^7 + x^6 + x^4 + x + 1)$	yok
$x^1 \otimes P_1$	$x \otimes (x^7 + x^6 + x^4 + x + 1)$	$(x^7 + x^5 + x^4 + x^3 + x^2 + 1)$	var
$x^2 \otimes P_1$	$x \otimes (x^7 + x^5 + x^4 + x^3 + x^2 + 1)$	$(x^6 + x^5 + 1)$	var
$x^3 \otimes P_1$	$x \otimes (x^6 + x^5 + 1)$	$(x^7 + x^6 + x)$	yok
$x^4 \otimes P_1$	$x \otimes (x^7 + x^6 + x)$	$(x^7 + x^4 + x^3 + x^2 + x + 1)$	var
$P_1 \times P_2 = (x^7 + x^5 + x^4 + x^3 + x^2 + 1) + (x^6 + x^5 + 1) + (x^7 + x^4 + x^3 + x^2 + x + 1) = x^6 + x + 1$			

Örnek 2.6' da verilen indirgenemez polinom $(x^8 + x^4 + x^3 + x + 1)$ AES blok şifresinde kullanılmaktadır. Bu indirgenemez polinom ilkel bir polinom olmadığı için tablo oluşturarak çarpma işlemi yapılmak istendiğinde Örnek 2.4'de verildiği gibi önce cismin ilkel bir eleman kullanılarak üretilmesi gerekir. Ancak $\alpha = [0,0,0,0,0,0,1,0]$ elemanı bu indirgenemez polinom ile tanımlı F_{2^8} ya da $GF(2^8)$ cisminde ilkel eleman olmadığı için tüm cismin elemanları bu eleman kullanılarak elde edilemez. Örneğin $\alpha + 1 = [0,0,0,0,0,0,1,1]$ elemanı ilkel bir elemandır ve bu eleman yolu ile tüm cismin elemanları üretilebilir ve daha sonra bu cisim üzerinde çarpma işlemi, ters alma işlemi gibi işlemler kolaylıkla yapılabilir.

Bir $GF(2^n)$ cismi üzerinde toplama işlemi bu cismin karakteristiği 2 olduğu için XOR işlemi ile kolaylıkla gerçekleştirilebilinir. Buna ek olarak dördüncü dereceden $GF(2^4)$ cismini tanımlayabilecek $x^4 + x + 1, x^4 + x^3 + 1$ ve $x^4 + x^3 + x^2 + x + 1$ olmak üzere 3 adet indirgenemez polinom varken $GF(2^8)$ cismini tanımlayabilecek 30 adet indirgenemez polinom bulunmaktadır. Aşağıda örnekte $(x^8 + x^4 + x^3 + x + 1)$ indirgenemez polinomu ile tanımlı $GF(2^8)$ cisminde çarpma işlemi verilen algoritma ile elde edilmektedir.

Örnek 2.7: "1*B*" hexadecimal byte değeri ile "03" hexadecimal byte değerinin çarpımını hem tablo yöntemi hem de verilen algoritma ile elde edelim.

$\alpha + 1 = [0,0,0,0,0,0,1,1]$ ilkel elemanı kullanılarak;

$$'1B' = 00011011 = (\alpha + 1)^{200}$$

$$'03' = 00000011 = (\alpha + 1)^{1}$$

olarak elde edilir. Bu iki byte değerinin çarpımı $(\alpha + 1)^{200}.(\alpha + 1)^{1} = (\alpha + 1)^{201}$ şeklinde elde edilebileceğinden çarpım sonucu Hexedecimal "2D" ya da "00101101" şeklindedir.

Eğer çarpım sonucunun üs değeri 255 değerinden yüksek bir değer elde edilirse, mod 255'e göre indirgeme yapılarak sonuç elde edilir. Çünkü $(\alpha+1)^{255}=1$ dir. Verilen algoritmayı kullanarak $'1B'\otimes'03'$ işleminin sonucunu $"1B"$ hexadecimal değerinin 1 sola ötelenmesi ve kendisi ile elde edilen sonucun XOR işlemine tabi tutulması ile elde edilebilir.

$$'1B'=00011011 \xrightarrow{1\,sola\,öteleme} 00110110\oplus 00011011\rightarrow 00101101="2D"$$

Bölüm 2 de anlatılanlara göre $GF(2^n)$ cisminde toplama, çarpma ve ters alma işlemleri ile ilgili örnekler aşağıda verilmiştir.

2.1 Sonlu Cisimlerde 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 katsayılarının basitçe modulo 2 aritmetiğine (XOR) göre toplamıdır denilebilir.

Örnek 2.8: a=(01110111) ve b=(10110101) olsun. O zaman a+b=11000010 olacaktır. Polinomsal olarak göstermek gerekirse $a=x^6+x^5+x^4+x^2+x+1$ ve b= $x^7+x^5+x^4+x^2+1$ olarak ifade edilir. Buradan a+b= x^7+x^6+x olarak bulunacaktır.

Çarpma

Sonlu cisim aritmetiğinde çarpma polinomların birbirleri ile aritmetik çarpımı şeklindedir. Fakat çarpma sonucunda doğal olarak sonlu cismin derecesinden daha

yüksek dereceli elemanlar oluşabilir. O zaman bu elemanları sonlu cismin derecesinden küçük olacak şekilde cismi oluşturan indirgenemez polinom aracılığı ile indirgemek gerekir. Dolayısı ile bu işlem indirgenemez polinoma göre indirgeme ya da mod alma işlemidir.

Örnek 2.9: $a = 1101$ ve $b = 0101$ ve indirgenemez polinom $x^4 + x + 1$ seçilsin. Bu değerlere göre;

$$
\begin{aligned}
a \cdot b &= \left(x^3 + x^2 + 1\right).\left(x^2 + 1\right) \\
&= \left(x^5 + x^4 + x^3 + x^2 + x^2 + 1\right) \\
&= x^5 + x^4 + x^3 + 1 \\
&= x^2 + x + x + 1 + x^3 + 1 \\
&\vdots \\
&= x^3 + x^2
\end{aligned}
\qquad
\begin{aligned}
x^1 &= x \\
&\vdots \\
x^4 &= x + 1 \\
x^5 &= x^2 + x \\
&\vdots \\
x^{15} &= 1
\end{aligned}
$$

şeklinde olacaktır.

2.2 Sonlu Cisimlerde 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.

Örnek 2.10: $GF\left(2^4\right)$ 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 $a^{-1} = \beta^{(-i) \bmod (2^n - 1)}$ 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 2.10'da n (örnek için 4) küçük 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. Bu tezde tablo yöntemi

kullanılarak cisim elemanları $P(x) = x^8 + x^4 + x^3 + x + 1$ indirgenemez polinomu kullanılarak elde edilmiştir. Sonlu cisimde ters alma işlemi için ikinci yöntem ise ikili Euclidean algoritmasını kullanmaktır. Sonlu cisimde ters alma işlemi için ikili Euclidean algoritması Algoritma 2.1 de gösterilmiştir.

Giriş:	$a \in GF(2^m), a \neq 0$
Çıkış:	$a^{-1} \bmod f$
Adım1:	$u \leftarrow a, v \leftarrow f, g_1 \leftarrow 1, g_2 \leftarrow 0$
Adım2:	X, u'yu tam böldüğü sürece aşağıdaki işlemleri gerçekle
	Adım 2.1: $u \leftarrow u / X$
	Adım 2.2: Eğer x , g_1'i tam bölerse $g_1 \leftarrow g_1 / X$ yap aksi takdirde $g_1 \leftarrow (g_1 + f) / X$ yap
Adım3:	Eğer $u = 1$ ise (g_1) değerini döndürür
Adım4:	Eğer derece (u) < derece (v) ise $u \leftrightarrow v, g_1 \leftrightarrow g_2$ yap
Adım5:	$u \leftarrow u + v, g_1 \leftarrow g_1 + g_2$
Adım6:	Adım 2'ye git

Algoritma 2.1. Ters Alma İşlemi için İkili Euclidean Algoritması (Sakallı M.T.,2006)

Algoritma 2.1 de gösterilen ikili Euclidean algoritması (1110)'ın tersini Örnek 2.10'da gösterildiği gibi $P_1(x) = x + 1$ ya da (0011) şeklinde bulacaktır. Bu sonuç, derece$(P_1(x)) < 4$ ve derece $(P_2(x)) < 3$ olmak üzere; $P_1(x).(x^3 + x^2 + x) + P_2(x).(x^4 + x + 1) = 1$ ifadesinde $P_1(x)$, $(x^3 + x^2 + x)$ polinomunun çarpmaya göre tersi olacak şekilde gösterilebilir. Yukarıdaki ifadede $P_1(x)$ ve $P_2(x) \in Z_2[x]$'tir.

3. İNCELENEN ŞİFRELER VE TASARIM STRATEJİLERİ

Bu bölümde incelenen AES, ARIA ve KHAZAD blok şifreleme algoritmalarının kısa bir tanıtımı yapılmaktadır.

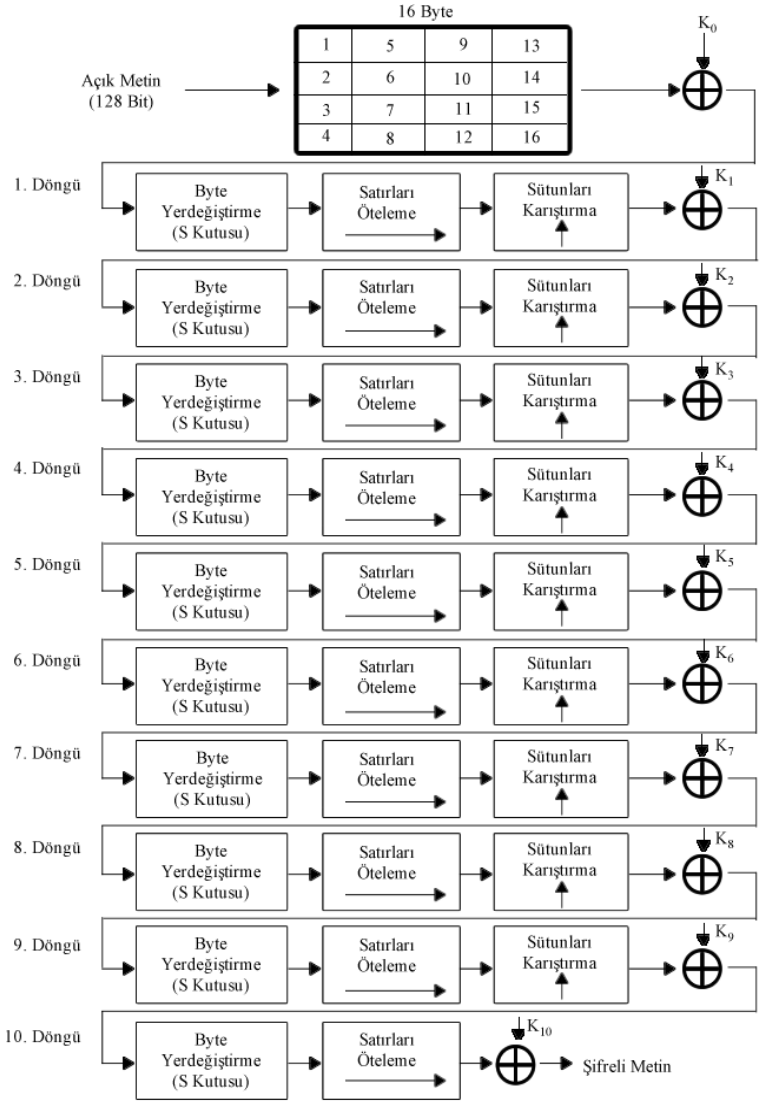
3.1 AES Blok Şifresi

AES (Advanced Encryption Standard, *Gelişmiş Şifreleme Standardı*), uluslararası olarak kullanılan bir şifreleme (kripto) sistemidir. Belçikalı Vincent Rijmen ve Joan Daemen tarafından geliştirilmiş, DES'in ve diğer olası algoritmaların zayıf yönlerini tamamen gidererek, sonlu cisimler, kodlama teorisi gibi önemli teorilerin ışığı altında oluşturulmuş olan bir algoritmadır.

AES şifreleme algoritması 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. İlk olarak 128-bit veri 4×4 byte matrisine dönüştürülür. Daha sonra her döngüde sırasıyla byte'ların yer değiştirmesi, satırları öteleme, sütunları karıştırma ve anahtar planlamadan gelen o döngü için belirlenen anahtar ile XOR' lama işlemleri yapılmaktadır. Byte'ların yer değiştirilmesinde 16-byte değerinin her biri 8 bit girişli ve 8 bit çıkışlı S-kutusuna sokulur. S-kutusu değerleri, Galois cismi (Galois Field-*GF*) $GF(2^8)$ de, 8 bitlik değerlerin $P(X) = x^8 + x^4 + x^3 + x + 1$ indirgenemez polinom tabanlı sonlu cisimde tersi alındıktan sonra doğrusal bir dönüşüme sokularak elde edilmektedir. Satırların

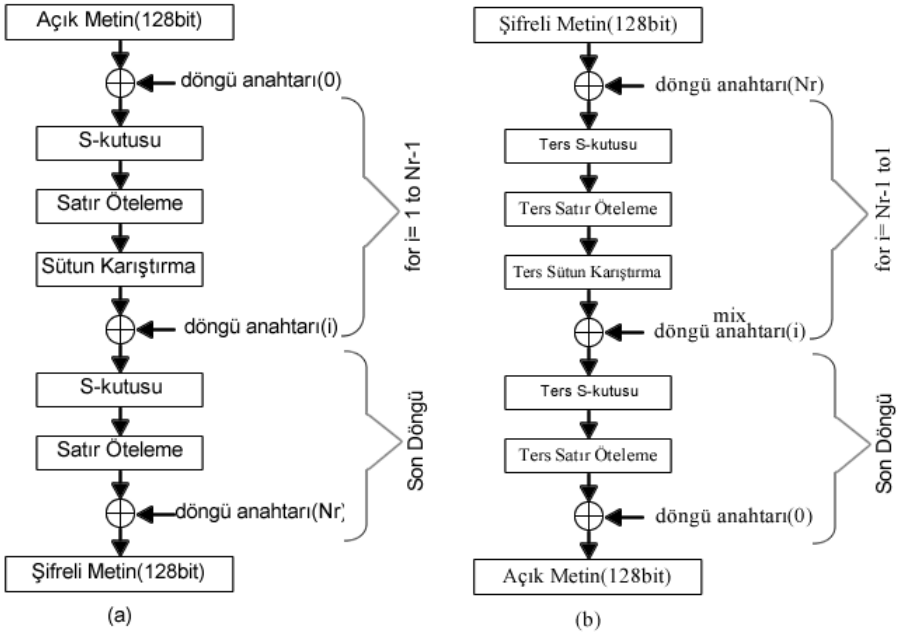
ötelenmesi işleminde 4×4 byte matrisinde satırlar ötelenmektedir ve sütunların karıştırılması işleminde herhangi bir sütun için o sütundaki değerler karıştırılmaktadır. Sütun karıştırma işleminde Galois cisminde iki sayının çarpım kavramı kullanılmaktadır. Döngünün son katmanında ise o döngüye ait anahtar ile XOR'lama işlemi yapılmaktadır.



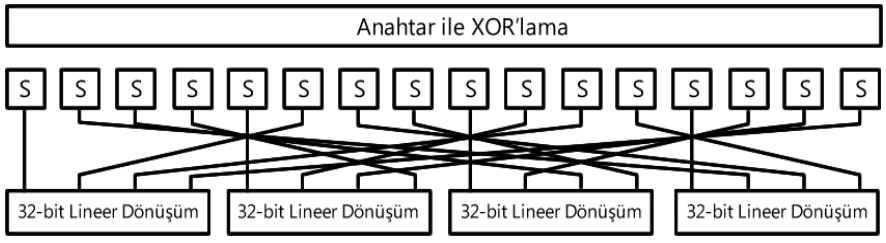
Şekil 3.1.1. 10 Döngü için AES Algoritması

Şekil 3.1.1, 10 döngülük AES algoritmasını göstermektedir. AES algoritmasında S kutularının tasarımında sonlu cisimde ters alma işlemi, doğrusal kriptanaliz için

kullanılan doğrusal yaklaşım tablolarına ve diferansiyel kriptanaliz için kullanılan fark (difference) dağılım tablolarına girişlerin olabildiğince uniform'a yakın olmasını sağlarken (diferansiyel ve doğrusal kriptanalize karşı etkin olması demek), sütunları karıştırma (doğrusal dönüşüm) işlemi saldırılarda az sayıda aktif S kutusu (doğrusal ve diferansiyel kriptanaliz de az sayıda olması daha az açık metin/şifreli metin kullanılması demek) kullanmayı imkânsız hale getirir. AES şifresine, imkânsız diferansiyel saldırısı gibi çeşitli saldırlar yapılmıştır. Ancak bu saldırılar azaltılmış döngü sayısına sahip AES algoritmalarına karşı gerçekleştirilmiştir.



Şekil 3.1.2. AES Algoritmasında a) Şifreleme yapısı b) Deşifreleme yapısı



Şekil 3.1.3. AES Algoritmasında tek döngülük şifreleme (Keliher L., 2003)

Şekil 3.1.2 AES şifresinin şifreleme ve deşifreleme adımını gösterirken, Şekil 3.1.3 SPN mimarisi tabanlı olarak şifrenin bir döngüsünü göstermektedir. AES birden fazla döngü işlemi yapar. Her döngü de birden fazla adımdan oluşmaktadır. Her adımdan önce ve sonra veri bloğu durum (state) adını alır.

3.1.1 AES Blok Şifresinde Döngü Yapısı

AES blok şifresindeki döngü özellikleri aşağıdaki gibi verilebilir:

- Her döngü tersi alınabilir dönüşümler kullanır.
- Her döngü, son döngü hariç, 4 dönüşüm kullanır: Byte Yerdeğiştirme (SubBytes), Satırların Ötelenmesi (ShiftRows), Sütunların Karıştırılması (MixColumns) ve Anahtar ile XOR'lama (AddRoundKey).
- Son döngü de Sütunları Karıştırma dönüşümü göz ardı edilir.
- Her döngüde farklı anahtar materyali kullanılır.
- Farklı anahtar materyalleri anahtar planlama evresinde gelen anahtarlardır. Gizli (Master) anahtardan farklı anahtarlar elde edilerek blok şifrede kullanılır.
- Deşifreleme kısmında ters dönüşümler kullanılır: InvSubByte, InvShiftRows, InvMixColumns ve AddRounKey (tersi kendisidir- XOR işlemi).

3.1.2 Byte Yerdeğiştirme (SubByte) Dönüşümü

AES blok şifresi, her byte (8-bit) değere karşılık farklı byte yer değiştirmesi işlemini şifreye doğrusal olmama özelliğini katmak için uygulamaktadır. AES şifresinin

S-kutusu $GF(2^8) = Z_2(x) \big/ x^8 + x^4 + x^3 + x + 1$ sonlu cisminde $x \to x^{-1}$ ters haritalama işleminden sonra aşağıdaki verilen ikili doğrusal dönüşüm uygulanarak elde edilmiştir.

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

Bu doğrusal dönüşümün eklenmesi S-kutusunun kriptografik özellikleri açısından herhangi bir iyileştirme yapmamakla beraber cebirsel ifadesini daha karmaşık hale getirmektedir. $GF(2^8)$ de AES S-kutusunun cebirsel ifadesi x^{-1} veya x^{254} olarak ifade edilirken ters alma işleminin çıkışına uygulanan ikili doğrusal dönüşümden sonra bu ifade aşağıdaki şekli almıştır:

$$S(x) = "05"x^{254} + "09"x^{253} + "f9"x^{251} + "25"x^{247} + "f4"x^{239} + "01"x^{223} + "b5"x^{191} + "8f"x^{127} + "63"$$

Diğer bir ifadeyle S-kutusunun cebirsel ifadesi daha karmaşık hale getirilmiştir. Yine S-kutusunun sonlu bir cisimde ters alma işlemi ile tasarlanmasındaki amaç Nyberg'in (Nyberg K., 1994) çalışmasında da gösterdiği gibi doğrusal kriptanaliz (Matsui M.,1994) ve diferansiyel kriptanaliz (Biham E. ve Shamir A.,1991) saldırılarına karşı dayanıklı bir S-kutusu tasarlamaktır. Örneğin, AES S-kutusu doğrusal olmama değeri 112 ve fark dağılım tablosundaki en büyük değer 4'tür. AES blok şifresinin S-kutusu aşağıda Tablo 3.1.1'de verilmiştir:

Tablo 3.1.1. AES algoritmasında kullanılan S-kutusu

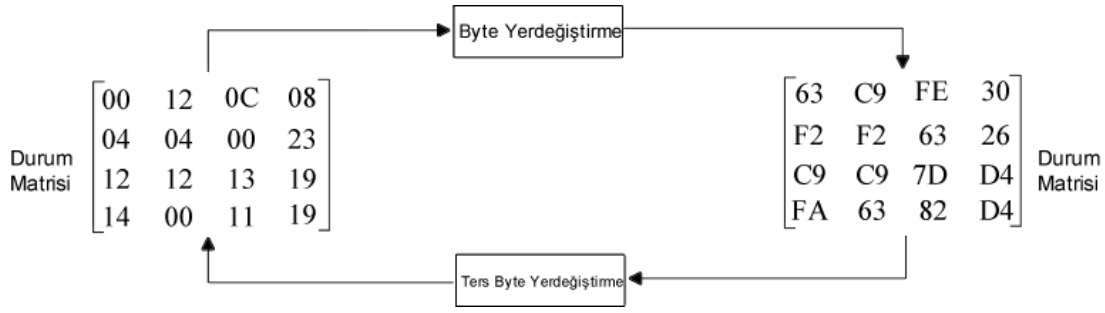
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	f4	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	07	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

AES blok şifresinin S-kutusunun tersi Tablo 3.1.2 de verilmiştir.

Tablo 3.1.2. AES algoritmasındaki 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	do	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	2	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

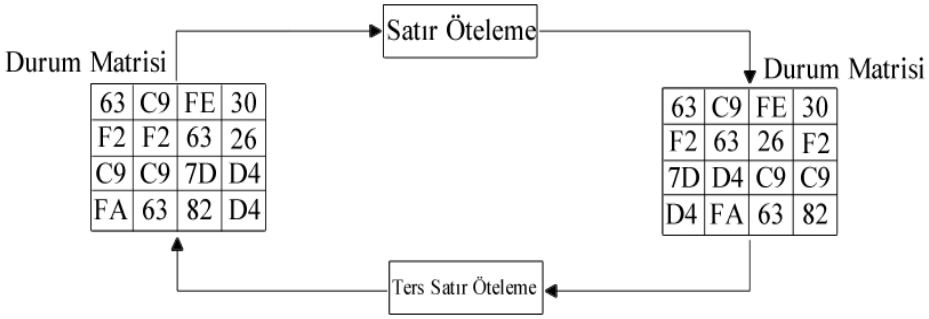
Bir durum matrisinde SubByte dönüşümü Şekil 3.1.4'te örnek bir durum matrisi üzerinde verilmiştir.



Şekil 3.1.4. AES algoritmasındaki SubByte işleminin tersi

3.1.3 Satırları Öteleme (ShiftRows)

Bu dönüşüm ilk satırın sola ötelenmemesi, ikinci satırın sola bir ötelenmesi, üçüncü satırın sola iki kere ötelenmesi, ve son satırın sola üç kere ötelenmesi işlemidir. Ters dönüşüm ise anlatılan şekilde sağa öteleme işlemidir. Şekil 3.1.5'te ShiftRows dönüşümüne bir örnek durum matrisi üzerinde verilmiştir.



Şekil 3.1.5. AES algoritmasında ShiftRow ve InvShiftRow işlemi

3.1.4 Sütunları Karıştırma (MixColumns)

Rijndael (AES) şifrenin içyapısında 32-bitten 32-bite dönüşüm yapan bir doğrusal dönüşüm içerir ve bu dönüşüm **MixColumns (**sütun karıştırma**)** olarak isimlendirilir. Bu dönüşüm doğrusal ve diferansiyel kriptanalizi zorlaştırıcı etki yapma amacındadır ve sonlu cisimde çarpma tabanlıdır. Bir MDS matris tabanlı olan bu doğrusal dönüşüm aşağıdaki gibi gösterilebilir:

$$\begin{bmatrix} a & b & 1 & 1 \\ 1 & a & b & 1 \\ 1 & 1 & a & b \\ b & 1 & 1 & a \end{bmatrix}$$

Yukarıdaki dönüşümde $a = x$ ve $b = x + 1$ olarak seçilmiştir. Buna ek olarak matrisin elemanları 8-bit değerler ya da $GF(2^8)$ cisminden elemanlardır.

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Sonuç olarak 32 bitlik dönüşüm $y_0,...,y_3,\ a_0,...,a_3$ 8-bit değerleri yani 1 byte değerleri temsil etmek üzere;

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \cdot \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

şeklinde gösterilebilir. Sonlu cisimde çarpma ve XOR işlemleri ile temsil edilebilecek bu dönüşüm indirgeme işlemleri için $x^8 + x^4 + x^3 + x + 1$ indirgenemez polinomunu kullanır.

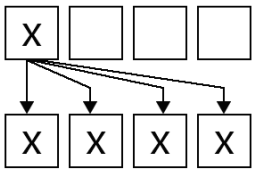
Aşağıdaki şekilde MixColumns işleminde kullanılan sabit matris ile deşifreleme işleminde kullanılan matris verilmiştir:

$$\underset{A}{\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}} \overset{tersi}{\leftrightarrow} \underset{A^{-1}}{\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ OB & 0D & 09 & 0E \end{bmatrix}}$$

Şekil 3.1.6. AES algoritmasında MixColumns dönüşümünde kullanılan sabit matris ve tersi

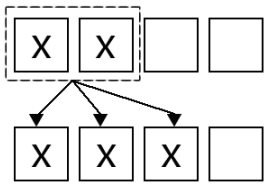
Yayılım elemanları şifre tasarımında kullanılan önemli elemanlardır. Yayılım özelliği doğrusal ve diferansiyel saldırılar düşünüldüğünde diferansiyel ve doğrusal yaklaşımların elde edilmesi ve saldırıların gerçeklenmesi açısından önemli kavramlardır. Yayılım özelliğinin çığ etkisi özelliğini karakterize etmenin bir yolu da dallanma sayısı olarak adlandırılan Daemen tarafından Ph.D tezinde ortaya atılan bir kavramdır.

Örneğin AES şifresindeki dallanma sayısı 5 tir. Bunun anlamı; 1 byte değişmesi sonucunda bu durumdan etkilenecek minimum byte sayısı 4 tür. Şekil 3.1.7 de bu durum gösterilmiştir.



Şekil 3.1.7. AES şifresinde MixColumns dönüşümünde giriş yapan bir byte'ın değişmesi sonucunda çıkışın 4 byte'ı etkilemesi

Eğer 2 byte değişirse bu durumda etkilenecek byte sayısı minimum 3 olmalıdır. Şekil 3.1.8 de bu durum gösterilmiştir.



Şekil 3.1.8. AES şifresindeMixColumns dönüşümüne giriş yapan iki byte'ın değişmesi sonucunda çıkışın üç byte'ının etkilenmesi

3.1.5 AES Şifresinde Anahtar Planlama

AES şifresi daha önceden de bahsedildiği gibi 128-bit veri bloklarını 128, 192, 256-bit anahtar seçenekleri ile şifreleyen bir algoritmadır. SPN algoritmasının geniş bir çeşididir. Döngü sayısı anahtar genişliğine göre değişmektedir. 128-bit anahtar için 10 döngüde şifreleme yaparken 192 ve 256-bit anahtarlar için sırasıyla 12 ve 14 döngüde şifreleme yapmaktadır. Her döngüye döngü anahtarı yaratmak için AES bir anahtar genişletme işlemi kullanır. Eğer döngü sayısı N_r ise anahtar genişletme işlemi $(N_r + 1)$ adet 128-bit döngü anahtarını tek bir 128-bit şifre anahtarından elde eder. Şifre anahtar döngü başlamadan önce kullanılırken geri kalan döngü anahtarları her döngünün sonundaki son dönüşüm olarak kullanılır.

Anahtar genişletme rutini döngü anahtarlarını word word (32-bit 32-bit) yaratır. Rutin aşağıdaki gibi tanımlanan $4 \times (N_r + 1)$ adet word yaratır:

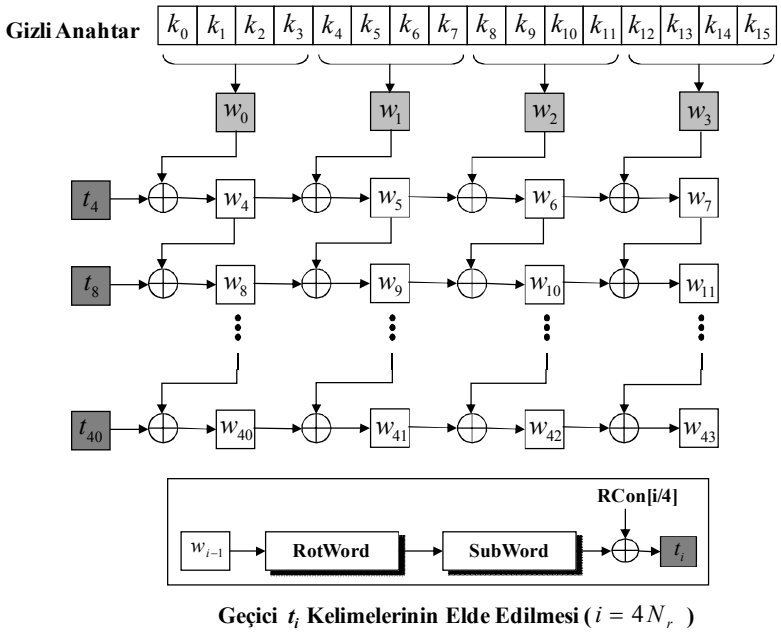
$w_0, w_1, w_2, \ldots, w_{4(N_r+1)-1}$.

Diğer bir deyişle AES-128 şifresinde 44 kelime(word), AES-192 şifresinde 52 kelime(word) ve AES-256 şifresinde 60 kelime(word) vardır. Tablo 3.1.3 döngüler ve wordler arasındaki ilişkiyi göstermektedir:

Tablo 3.1.3. AES şifresindeki Döngüler ve Wordler arasındaki ilişki

Döngü	Word'ler
Döngüye hazırlık	w_0 w_1 w_2 w_3
1	w_4 w_5 w_6 w_7
2	w_8 w_9 w_{10} w_{11}
……	……..
N_r	w_{4N_r} w_{4N_r+1} w_{4N_r+2} w_{4N_r+3}

AES-128 de anahtar genişletme işlemi ile şifre anahtarından 44 kelime'nin nasıl yaratıldığı Şekil 3.1.9'da gösterilmektedir. Şifrenin diğer iki versiyonu için de anahtar planlama algoritması bazı küçük değişiklikler ile birlikte aynıdır.



Şekil 3.1.9. AES-128 de Anahtar Genişletme Algoritması(Stinson D. R., 2002)

Anahtar planlama işlemi aşağıdaki gibidir:

1- İlk 4 word (w_0, w_1, w_2, w_3) şifre anahtarından elde edilir. Şifre anahtarı k_0 dan k_{15}'e kadar 16 byte bir dizi olarak düşünülür. İlk 4 byte (k_0 dan k_3'e) w_0, ikinci 4 byte (k_4'ten k_7'ye) w_1 ve bu şekilde diğer word'ler w_2 ve w_3'te şifre anahtarının wordler şeklinde yan yana konması ile elde edilir.
2- Diğer wordler (w_i i=4 den 43'e kadar) aşağıdaki şekilde elde edilir:
   a- Eğer i (mod 4) $\neq$ 0 ise $w_i = w_{i-1} \oplus w_{i-4}$ şeklinde tablodan da görüldüğü gibi soldan ve üstten bir değerden elde edilir.
   b- Eğer i (mod 4) = 0 ise $w_i = t \oplus w_{i-4}$ şeklinde elde edilir. Burada t geçici bir bellek ve iki rutinin w_{i-1} üzerindeki uygulama sonucudur: SubWord ve RotWord. t'nin elde edilme süreci bir döngü sabiti RCon ile XOR lama işlemi ile sonlanır. Diğer bir deyişle;

$$t = SubWord(RotWord(w_{i-1})) \oplus RCon_{i/4}.$$

RotWord

ShiftRows dönüşümüne benzemektedir. Ancak sadece 1 satıra uygulanır. Bu rutin bir word'ü 4 byte'ın bir dizisi olarak alır ve her byte'ı sola öteler.

SubWord

Bu rutin SubBytes dönüşümüne benzemektedir. Ancak sadece 4 byte'a uygulanır. Bu döngü worddeki her byte değerini alır ve diğer bir byte ile yerdeğiştirir.

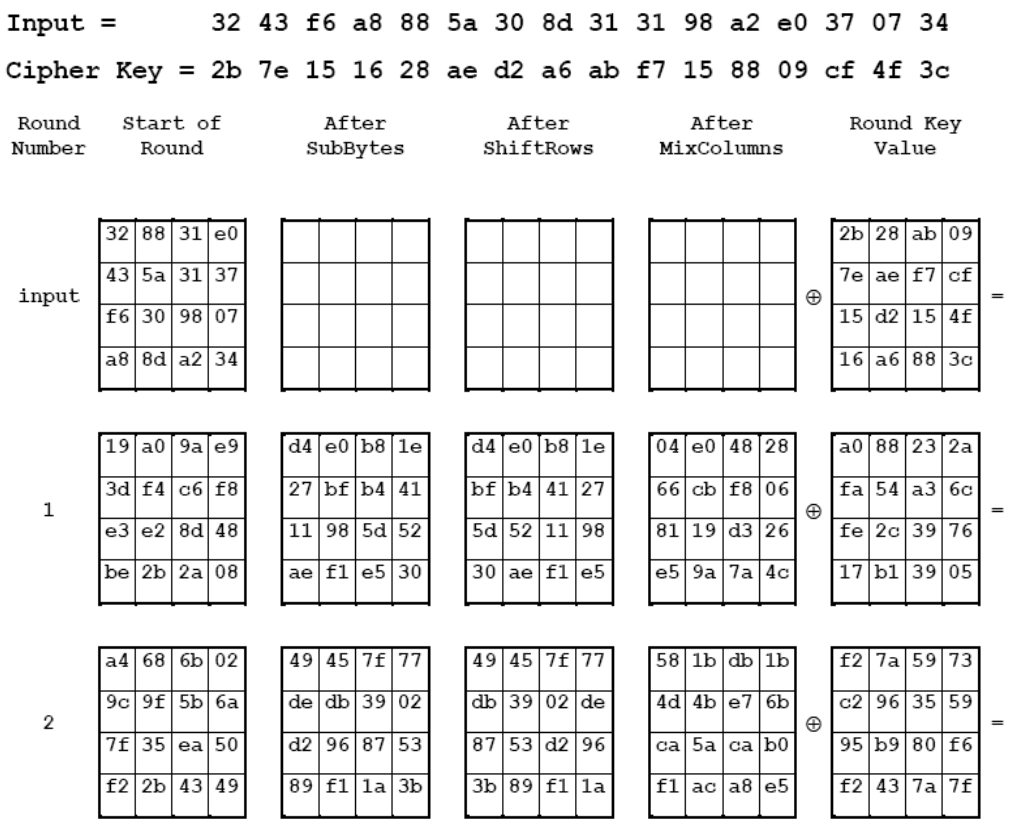
Döngü Sabitleri (Round Constants)

Her döngü sabiti, RCon, 4 byte değerinde ve en sağdaki 3 byte'ı 0 olan bir değerdir. Aşağıdaki tablo AES-128 için (10 döngü) değerleri göstermektedir.

Round	Constant(RCon)	Round	Constant(RCon)
1	(**01** 00 00 00)$_{16}$	6	(**20** 00 00 00)$_{16}$
2	(**02** 00 00 00)$_{16}$	7	(**40** 00 00 00)$_{16}$
3	(**04** 00 00 00)$_{16}$	8	(**80** 00 00 00)$_{16}$
4	(**08** 00 00 00)$_{16}$	9	(**1B** 00 00 00)$_{16}$
5	(**10** 00 00 00)$_{16}$	10	(**36** 00 00 00)$_{16}$

Anahtar genişletme rutini word değerlerini hesaplarken ya yukarıdaki tabloyu ya da en soldaki byte'ı dinamik olarak hesaplamak GF(2^8) cismini kullanır.

AES blok şifresinin verilen giriş ve anahtar ile 2 döngülük şifreleme işlemi Şekil 3.1.10'da gösterilmektedir



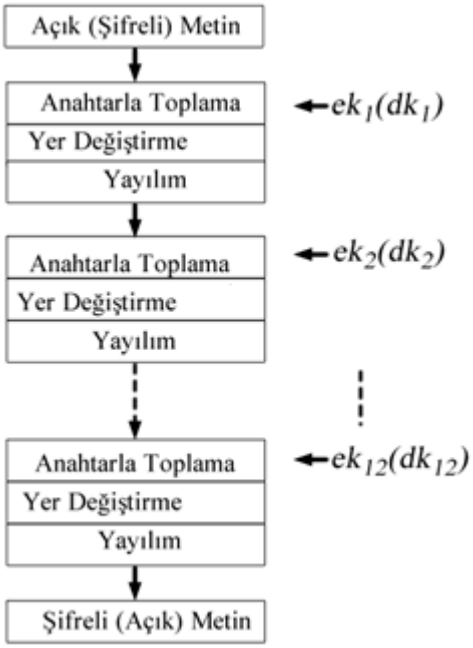
Şekil 3.1.10. AES-128 algoritmasında bir şifreleme örneği (AES-128'in 2 döngülük çalışması) (FIPS, 1999)

3.2 ARIA Blok Şifresi

ARIA blok şifresi Güney Koreli araştırmacılar tarafından 2004'te tasarlanmıştır ve Kore Teknoloji Ajansı tarafından standart şifreleme tekniği olarak seçilmiştir (Kwon D., Kim J. vd, 2004).

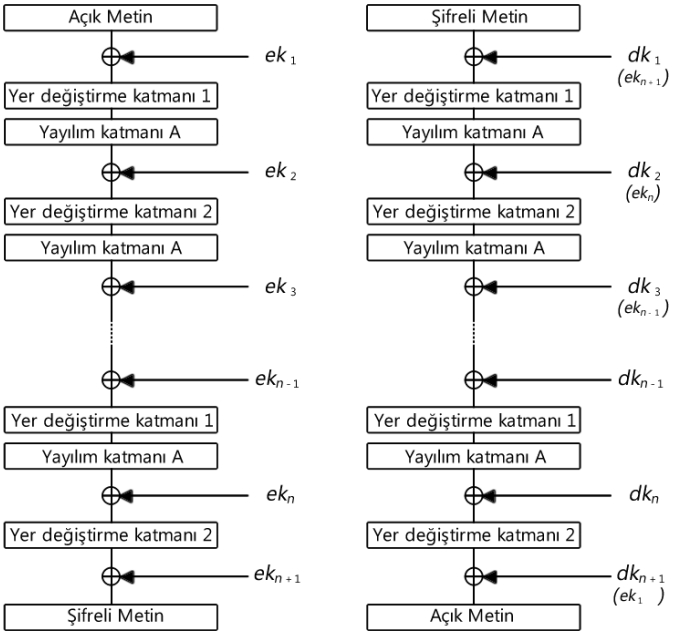
ARIA blok şifresi 128-bit blokları 128, 192 ve 256-bit anahtar seçenekleri ile şifreler. Döngü sayısı verilen anahtar seçeneklerine göre sırasıyla 10, 12 ve 14 tür. SPN tabanlı olmasına ek olarak yayılım katmanında 16×16 boyutunda involutif bir ikili doğrusal dönüşüm kullanır. Yer değiştirme katmanı, $GF(2^8)$ üzerine 8×8'lik S-kutusu ile 16 byte değeri farklı 16 byte değerlerine haritalanır (Kwon D., Kim J. vd, 2004).

ARIA blok şifresinin tek döngüsü anahtar İle XOR'lama (AddRoundKey), yer değiştirme (substitution) ve yayılım (diffusion) olmak üzere 3 katmandan oluşur. Şekil 3.2.1 ARIA şifresinin şifreleme adımlarını göstermektedir.



Şekil 3.2.1. ARIA Algoritması

Anahtarla toplama safhasında 128 bit anahtarla XOR işlemi yapılır. Yer değiştirme aşamasında ise 16 S-kutusu çıkışları yeni durumu belirler. Yayılım katmanında ise 16×16'lık ikili matris ile 128-bit durum verisi 128-bit yeni durum verisine dönüştürülür. ARIA'da Type1 ve Type 2 olmak üzere 2 farklı yer değiştirme katmanı vardır.



Şekil 3.2.2. ARIA Şifresinde şifreleme ve deşifreleme safhaları

3.2.1 Yer Değiştirme Katmanı (Substitution Layer)

ARIA algoritmasında yer değiştirme katmanında S_1, S_2 ve onların tersleri S_1^{-1} ve S_2^{-1} olmak üzere 4 tip S-kutusu kullanılır. Bu S kutuları Tablo 3.2.5, Tablo 3.2.6, Tablo 3.2.7 ve Tablo 3.2.8 de gösterilmiştir. Şekil 3.2.3 ve Şekil 3.2.4 de görüldüğü gibi Type 1 ve Type 2 olarak ARIA şifresinde yer değiştirme katmanları kullanılmaktadır. Type 1 tek döngülerde, Type 2 ise çift döngülerde kullanılır.

S_1	S_2	S_1^{-1}	S_2^{-1}	S_1	S_2	S_1^{-1}	S_2^{-1}	S_1	S_2	S_1^{-1}	S_2^{-1}	S_1	S_2	S_1^{-1}	S_2^{-1}

Şekil 3.2.3. ARIA şifresinde Type1 Yer değiştirme (Substitution) katmanı

S_1^{-1}	S_2^{-1}	S_1	S_2	S_1^{-1}	S_2^{-1}	S_1	S_2	S_1^{-1}	S_2^{-1}	S_1	S_2	S_1^{-1}	S_2^{-1}	S_1	S_2

Şekil 3.2.4. ARIA şifresinde Type 2 Yer değiştirme (Substitution) katmanı

Tablo 3.2.1. ARIA algoritmasında ki S_1 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	f4	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	07	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tablo 3.2.2. ARIA algoritmasındaki S_1^{-1} S-kutusu

	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	do	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	2	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	C0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	B1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	C9	9c	ef
e	A0	e0	3b	4d	ae	2a	F5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Tablo 3.2.3. ARIA algoritmasındaki S2 S-kutusu

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	e2	4e	54	fc	94	C2	4a	cc	62	0d	6a	46	3c	4d	8b	d1
1	5e	fa	64	cb	b4	97	be	2b	bc	77	2e	03	d3	19	59	c1
2	1d	06	41	6b	55	f0	99	69	ea	9c	18	ae	63	df	e7	bb
3	00	73	66	fb	96	4c	85	e4	3a	09	45	aa	0f	ee	10	eb
4	08	7f	f4	29	ac	cf	ad	91	8d	78	C8	95	f9	2f	ce	cd
5	ff	7a	88	38	5c	83	2a	28	47	db	b8	c7	93	a4	12	53
6	b7	87	0e	31	36	21	58	48	01	8e	37	74	32	ca	e9	b1
7	ec	ab	0c	d7	C4	56	42	26	07	98	60	d9	b6	b9	11	40
8	d8	20	8c	bd	a0	c9	84	04	49	23	f1	4f	50	1f	13	dc
9	15	c0	9e	57	e3	c3	7b	65	3b	02	8f	3e	e8	25	92	e5
a	a7	dd	fd	17	a9	bf	d4	9a	7e	c5	39	67	fe	76	9d	43
b	30	e1	d0	f5	68	f2	1b	34	70	05	a3	8a	d5	79	86	a8
c	e6	c6	51	4b	1e	a6	27	f6	35	d2	6e	24	16	82	5f	da
d	90	75	a2	ef	2c	b2	1c	9f	5d	6f	80	0a	72	44	9b	6c
e	ed	0b	5b	33	7d	5a	52	f3	61	a1	f7	b0	d6	3f	7c	6d
f	8c	14	e0	a5	3d	22	b3	f8	89	de	71	1a	af	ba	b5	81

Tablo 3.2.4. ARIA algoritmasındaki S_2^{-1} S-kutusu

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	30	68	99	1b	87	b9	21	78	50	39	db	e1	72	09	62	3c
1	3e	7e	5e	8e	f1	a0	cc	A3	2a	1d	fb	b6	d6	20	c4	8d
2	81	65	f5	89	cb	9d	77	c6	57	43	56	17	d4	40	1a	4d
3	c0	63	6c	e3	b7	c8	64	6a	53	aa	38	98	0c	f4	9b	ed
4	7f	22	76	af	dd	3a	0b	58	67	88	06	c3	35	0d	01	8b
5	8c	c2	e6	5f	02	24	75	93	66	1e	e5	e2	54	d8	10	ce
6	7a	e8	08	2c	12	97	32	ab	b4	27	0a	23	df	ef	ca	d9
7	b8	fa	dc	31	6b	d1	ad	19	49	bd	51	96	ee	e4	a8	41
8	da	ff	cd	55	86	36	be	61	52	f8	bb	0e	82	48	69	9a
9	e0	47	9e	5c	04	4b	34	15	79	26	a7	de	29	ae	92	d7
a	84	e9	d2	ba	5d	f3	c5	b0	bf	a4	3b	71	44	46	2b	fc
b	eb	6f	d5	f6	14	fe	7c	70	5a	7d	fd	2f	18	83	16	a5
c	91	1f	05	95	74	a9	C1	5b	4a	85	6d	13	07	4f	4e	45
d	b2	0f	c9	1c	a6	bc	ec	73	90	7b	cf	59	8f	a1	f9	2d
E	f2	b1	00	94	37	9f	d0	2e	9c	6e	28	3f	80	f0	3d	d3
f	25	8a	b5	e7	42	b3	c7	ea	f7	4c	11	33	03	a2	ac	60

3.2.2 Yayılım katmanı (Diffusion Layer)

ARIA' nın yayılım katmanı $(x_0, x_1, ..., x_{15})$ byte girişlerini 16 byte' lık $(y_0, y_1, ..., y_{15})$ çıkışlarına haritalar. Bu doğrusal dönüşümün dallanma sayısı 8 (MDBL kod) ve sabit nokta sayısı ise 2^{72} dir.

$$y_0 = x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_9 \oplus x_{13} \oplus x_{14} \qquad y_8 = x_0 \oplus x_1 \oplus x_4 \oplus x_7 \oplus x_{10} \oplus x_{13} \oplus x_{15}$$

$$y_1 = x_2 \oplus x_5 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{12} \oplus x_{15} \qquad y_9 = x_0 \oplus x_1 \oplus x_5 \oplus x_6 \oplus x_{11} \oplus x_{12} \oplus x_{14}$$

$$y_2 = x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15} \qquad y_{10} = x_2 \oplus x_3 \oplus x_4 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15}$$

$$y_3 = x_0 \oplus x_5 \oplus x_7 \oplus x_{10} \oplus x_{11} \oplus x_{13} \oplus x_{14} \qquad y_{11} = x_2 \oplus x_3 \oplus x_4 \oplus x_7 \oplus x_9 \oplus x_{12} \oplus x_{14}$$

$$y_4 = x_0 \oplus x_2 \oplus x_5 \oplus x_8 \oplus x_{11} \oplus x_{14} \oplus x_{15} \qquad y_{12} = x_1 \oplus x_2 \oplus x_6 \oplus x_7 \oplus x_9 \oplus x_{11} \oplus x_{12}$$

$$y_5 = x_1 \oplus x_3 \oplus x_4 \oplus x_9 \oplus x_{10} \oplus x_{14} \oplus x_{15} \qquad y_{13} = x_0 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{10} \oplus x_{13}$$

$$y_6 = x_0 \oplus x_2 \oplus x_7 \oplus x_9 \oplus x_{10} \oplus x_{12} \oplus x_{13} \qquad y_{14} = x_0 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_9 \oplus x_{11} \oplus x_{14}$$

$$y_7 = x_1 \oplus x_3 \oplus x_6 \oplus x_8 \oplus x_{11} \oplus x_{12} \oplus x_{13} \qquad y_{15} = x_1 \oplus x_2 \oplus x_4 \oplus x_5 \oplus x_8 \oplus x_{10} \oplus x_{15}$$

Bu haritalama işlemi aşağıda verilen ikili matris şeklinde de gösterilebilir:

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

3.2.3 Key Expansion (Anahtar Üretimi)

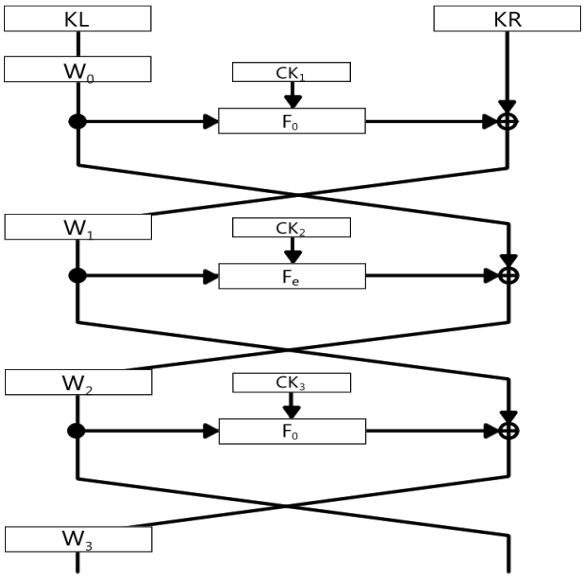
ARIA algoritmasında anahtar üretimi başlangıç ve döngü anahtar üretimi olmak üzere 2 kısımdan oluşur.

3.2.2.1 Başlangıç

Başlangıç kısmı; 4 tane 128 bit değer W_0, W_1, W_2, W_3 şifre anahtarından (master) M_K anahtardan 3 döngülü 256-bit Feistel şifresiyle üretilmesi ile oluşur. M_K şifre anahtarı 128,192 ya da 256-bit olabilir.

3.2.2.2 Round key generation (Döngüden Gelen Anahtar Üretimi)

Bu safhada 4 tane W_i değeri ile şifreleme anahtarı ek_i ve deşifreleme anahtarı dk_i üretilir.



Şekil 3.2.5. ARIA algoritmasında anahtar genişletme için başlangıç

3.3 KHAZAD Blok Şifresi

KHAZAD şifresi 128-bit anahtarla çalışan 64-bit bir blok şifredir NESSIE projesine aday blok şifre olarak sunulmuştur.

Her ne kadar KHAZAD SPN tabanlı bir mimari kullanıyor olsa da tüm döngü dönüşümlerinin tersinin kendisi olması, şifrenin tersi işlemlerinde sadece anahtar planlamada farklılık olacak şekilde tasarımına izin vermektedir. Diğer bir deyişle Khazad involutif bir blok şifredir.

KHAZAD, Wide Trail (geniş iz) stratejisine göre tasarlanmıştır (J. Daemen, 1995). Wide Trail stratejisinde, bir blok şifrenin döngü dönüşümü tersi alınabilir farklı dönüşümlerin birleşiminden 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 (geniş iz) stratejinin diğer bir avantajı da farklı bileşenlerin birbirlerinden tamamen farklı bir şekilde belirlenebilmesidir.

3.3.1 Doğrusal Yayılım Katmanı

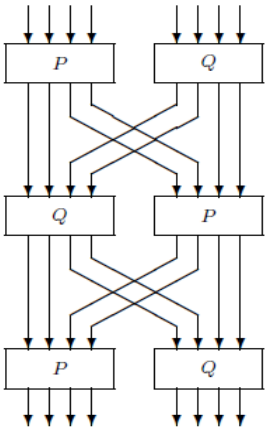
KHAZAD blok şifresinin yayılım katmanı θ, G_H üreteç matrisi kullanılarak $[16, 8, 9]$ MDS kodu ile üretilmiş $GF\left(2^8\right)^8 \rightarrow GF\left(2^8\right)^8$ haritalama yapan involutif ve Hadamard formda bir doğrusal dönüşümdür.

$$H = Had(01_h, 03_h, 04_h, 05_h, 06_h, 08_h, 0B_h, 07_h)$$

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

3.3.2 Khazad S-kutusu

Khazad sözde (pseudo) rastsal şekilde üretilmiş bir S-kutusu kullanmaktadır. Bu tasarım sırasında P ve Q olmak üzere iki kutu kullanılmıştır ve sonuç olarak involutif ve kriptografik özellikleri iyi bir S-kutusu elde edilmiştir. Şekil 3.3.1 de KHAZAD S-kutusunun 4-bit giriş ve 4-bit çıkışa sahip P ve Q kutuları ile nasıl üretildiği gösterilmektedir. Tablo 3.3.1 ve Tablo 3.3.2'de kullanılan P ve Q kutuları, Tablo 3.3.3'te ise sonuçta üretilen 8×8 S-kutusu verilmektedir.



Şekil 3.3.1. KHAZAD S-kutusunun yapısı

Tablo 3.3.1. Q kutusu

u	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
$Q(u)$	9_x	E_x	5_x	6_x	A_x	2_x	3_x	C_x	F_x	0_x	4_x	D_x	7_x	B_x	1_x	8_x

Tablo 3.3.2. P kutusu

u	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
$P(u)$	3_x	F_x	E_x	0_x	5_x	4_x	B_x	C_x	D_x	A_x	9_x	6_x	7_x	8_x	2_x	1_x

Tablo 3.3.3. KHAZAD S-kutusu

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	ba	54	2f	74	53	d3	d2	4d	50	ac	8d	bf	70	52	9a	4c
10	ea	d5	97	d1	33	51	5b	a6	de	48	a8	99	db	32	b7	fc
20	e3	9e	91	9b	e2	bb	41	6e	a5	cb	6b	95	A1	F3	B1	02
30	cc	c4	1d	14	c3	63	da	5d	5f	dc	7d	cd	7f	5a	6c	5c
40	f7	26	ff	ed	e8	9d	6f	8e	19	a0	f0	89	0f	07	af	fb
50	08	15	0d	04	01	64	df	76	79	dd	3d	16	3f	37	6d	38
60	b9	73	e9	35	55	71	7b	8c	72	88	f6	2a	3e	5e	27	46
70	0c	65	68	61	03	c1	57	d6	d9	58	d8	66	d7	3a	c8	3c
80	fa	96	a7	98	ec	b8	c7	ae	69	4b	ab	a9	67	0a	47	f2
90	b5	22	e5	ee	be	2b	81	12	83	1b	0e	23	f5	45	21	ce
A0	49	2c	f9	e6	b6	28	17	82	1a	8b	fe	8a	09	c9	87	4e
B0	e1	2e	e4	e0	eb	90	a4	1e	85	60	00	25	f4	f1	94	0b
C0	e7	75	ef	34	31	d4	d0	86	7e	ad	fd	29	30	3b	9f	f8
D0	c6	13	06	05	c5	11	77	7c	7a	78	36	1c	39	59	18	56
E0	b3	b0	24	20	b2	92	a3	c0	44	62	10	b4	84	43	93	c2
F0	4a	bd	8f	2d	bc	9c	6a	40	cf	a2	80	4f	1f	ca	aa	42

3.4 AES, ARIA ve KHAZAD Blok Şifrelerinin Karşılaştırılması

Geliştirilecek şifre incelenen şifrelerden esinleneceği için bu şifrelerin arasındaki benzerlikler ya da farklılıklar bu alt bölümde ortaya konmaktadır. İncelenen blok şifrelerin benzerlikleri ya da farklılıkları verilmektedir:

AES ve ARIA blok şifreleri 128-bit blokları sırasıyla 128-bit, 192-bit ve 256-bit anahtar seçenekleri ile şifreler ve deşifreler. Khazad blok şifresi ise 64-bit blokları 128-bit anahtar seçeneği ile şifreler ve deşifreler.

AES şifresi SPN mimarisine uygun olarak involutif bir yapı olarak tasarımı gerçekleştirilmemiştir. Buna karşın diğer iki blok şifre ARIA ve Khazad tasarımı involutif olacak şekilde geliştirilmiştir.

ARIA ve Khazad blok şifrelerinin involutif yapıda geliştirilme sebebi şifreleme ve deşifreleme performansının aynı olmasını sağlama amacı gütmektedir. Ancak bu iki blok şifrede involutif yapı sağlanırken tasarım farklılıkları göze çarpmaktadır. Örneğin her iki şifre involutif doğrusal dönüşümler kullanmaktadır. Ancak bu dönüşümler Khazad blok şifresinde 8×8 büyüklüğünde elemanları $GF(2^8)$ den olan involutif bir MDS matris iken ARIA blok şifresinde kullanılan doğrusal dönüşüm elemanları GF(2) den olan 16×16 büyüklüğünde involutif bir doğrusal dönüşümdür. Buna ek olarak ARIA blok şifresinde kullanılan doğrusal dönüşüm yazılım ve donanım performansını maksimize edecek şekilde geliştirilmiş özel bir dönüşümdür. Ancak bu iyileştirme doğrusal dönüşümün sabit nokta sayısı da 2^{72} tane olacak şekilde yüksek bir değer göstermesine neden olmaktadır. Bunun doğrusal dönüşümün kriptografik özelliği olarak çok ta iyi bir seçim olmadığını düşündürmektedir.

Khazad blok şifresinin involutif özellik gösterebilmesi için kullandığı S-kutusunun tasarımını özel bir yöntemle elde edilmiştir. Ancak bu S-kutusu kriptografik özellikleri (doğrusal olmama, LAT tablosu, DDT tablosu) açısından AES S-kutusuna göre kötü özellikler göstermektedir. Diğer yandan ARIA blok şifresi iki S kutusu ve

bunların tersleri olmak üzere 4 S-kutusu kullanarak farklı bir tasarım stratejisi ile involutif şifre tasarımı amacını gütmektedir.

AES ve ARIA blok şifrelerin anahtar genişletme algoritmaları incelendiğinde her ikisinde de bit sızıntısı probleminin olduğu söylenebilir. Ancak ARIA blok şifresinin AES blok şifresine göre anahtar genişletme algoritması istatistiksel özellikler açısından çok daha iyi sonuçlar göstermektedir.

4. BLOK ŞİFRELERDE KULLANILAN KRİPTOGRAFİK YAPILAR

Modern blok şifreler köklerini Shannon'ın dönüm noktası olan makalesinde (C.E. Shannon, 1949) sunulan karıştırma (confusion) ve yayılım (diffusion) prensiplerinden almaktadır (Keliher L., 2003). Karıştırma şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılım açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Karıştırma ve yayılım, sırasıyla yer değiştirme kutuları (S-kutuları) ve doğrusal dönüşüm işlemleri ile gerçekleştirilir (Sakallı F. B., 2011).

Bir blok şifrenin tasarımında kullanılan 3 genel yapı aşağıdaki gibi verilebilir:

- S-kutuları (Substitution Boxes),
- Doğrusal Dönüşümler,
- Anahtar Genişletme algoritmaları.

4.1. S-kutuları (Yer değiştirme Kutuları-Substitution Boxes)

S-kutuları simetrik şifreleme algoritmalarının temel bileşenlerindendir. Blok şifreleme algoritmalarında karıştırma işlemi yapan elemandır. Şifreleme algoritmasında doğrusal olmayan tek eleman S-kutularıdır. Şifreleme algoritmasına yapılan saldırılardan doğrusal (Matsui M., 1994) ve diferansiyel saldırılara (Biham E. ve Shamir A., 1991) karşı blok şifreleme algoritmasını güvenli kılmak için kriptografik özellikleri iyi olan S-kutuları seçilmelidir.

Bir $n \times n$ S-kutusu $f : \{0,1\}^n \rightarrow \{0,1\}^n$ şeklinde n-bit girişi farklı n-bit çıkışa haritalayan bir fonksiyondur. S-kutuları bir blok şifrenin içerisinde bulunan tek doğrusal olmayan yapıdır.

S-kutularının tasarım tekniklerine örnek olarak;

- pseudo-random üretim,
- sonlu cisimde ters haritalama,
- sonlu cisimde üs haritalama,
- heuristik teknikler verilebilir (Aslan B., Sakallı M. T., Buluş E., 2008).

Bu yöntemlerin en çok kullanılanları sonlu cisimde ters alma ve üssel fonksiyon tekniğidir. Nitekim AES algoritmasında kullanılan S-kutusu sonlu cisimde ters alma yöntemi ile oluşturulmuş 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 bir statik özelliktir. Dolayısıyla S-kutuları için kriptografik özelliklerden biri olan doğrusal olmama özelliği önemli bir özelliktir. S-kutusunun karakteristik yapısının saklandığı kriptografik özellikler dinamik olanlardır.

S-kutuları için kriptografik özellikler aşağıdaki gibi sıralanabilir:

- Bütünlük (Completeness) kriteri,
- Çığ (Avalanche) kriteri,
- Katı çığ kriteri (Strict Avalanche Criterion),
- Bit bağımsızlık kriteri (Bit Independence Criterion),
- MOSAC ve MOBIC özellikleri,
- Doğrusal olmama kriteri,
- S-kutularının doğrusal yaklaşım tablosu,
- S-kutularının XOR tablosu (Fark Dağılım Tablosu),
- S-kutularında doğrusal eşitlik.

4.1.1 Doğrusal Olmama Kriteri

Doğrusal olmama (nonlinearity) S-kutuları için oldukça önemlidir. Şifrede kullanılan S-kutularının doğrusal olmaması istenir. Böylelikle açık metnin tahmini veya bulunması imkânsız hale gelir.

Bir şifrenin doğrusal olmama parametresi $\mathrm{NLM_f(z)}$'dir. $f : Z_2^n \to Z_2^m$ ve $z = (a, w, c) \in Z_2^{n+m+1}$ olmak üzere, tüm giriş değerleri için $P \in Z_2^n$ doğrusal fonksiyon $(\mathrm{w.P} \oplus \mathrm{c})$ ve sıfır haricindeki doğrusal kombinasyonları $\left(\mathrm{a.f\left(P\right)}\right)$ birbirinden farklılaşması doğrusal olmama olarak tanımlanır. Burada $a \in Z_2^m, w \in Z_2^n$ ve $c \in Z_2$ dir. Buna göre doğrusal olmama ölçüsü, $\mathrm{NLM_f\left(z\right)}$, (4.1) ve (4.2) ifadelerindeki gibi tanımlanabilir (Kam J. B., Davida G. I., 1979), (Ferguson N., Kelsey J., vd., 2000).

$$\mathrm{NLM_f\left(z\right)} = \#\left\{P \mid a.f(P) \neq w.P \oplus c\right\} \tag{4.1}$$

$$\mathrm{NLM_f} = \min_{z} \mathrm{NLM_f\left(z\right)} \tag{4.2}$$

$\mathrm{NLM_f}$ değerinin alabileceği maksimum değer $2^{n-1} - 2^{\frac{n}{2}-1}$ dir. Şifrenin doğrusal olmaması ve doğrusal kriptanalize karşı başarılı olması için $\mathrm{NLM_f}$ değerinin bu maksimum değere yakın olması gerekir. $\mathrm{NLM_f}$ değerinin 0'a yakın olması istenmeyen bir özelliktir. Bu gibi durumlarda şifrenin doğrusal kriptanaliz ile kırılması olasıdır.

4.1.2 Doğrusal Yaklaşım Tablosu

Doğrusal yaklaşım tablosu (Lineer Approximation Table) (LAT) (Biryukov A., Dunkelman O., vd., 2009),(Rimoldi A., 2009),(Phan R. C.-W., 2004) doğrusal kriptanalize karşı S-kutularının gücünü test etmeye yarayan önemli bir ölçüttür.

Şifreleme algoritması için doğrusal yaklaşım tablosunda bulunan maksimum değerin küçük olması doğrusal saldırıların başarımını zorlaştıracaktır.

$S : GF(2^n) \to GF(2^n)$ olmak üzere n bit giriş ve n bit çıkışa sahip bir S-kutusu olsun. O zaman herhangi verilen a, b, Γ_a, Γ_b $GF(2^n)$ için $N_L(\Gamma_a, \Gamma_b)$,herhangi $\Gamma_a \setminus 0$ ve Γ_b için $x \in GF(2^n)$ olmak üzere $\Gamma_a \bullet x = \Gamma_b \bullet S(x)$ denklemini sağlayan değerlerin sayısını tanımlar ve (4.3) ifadesindeki gibi gösterilebilir (May L., Henricksen M., vd., 2002).S için (4.3) ifadesinde Γ_a ve Γ_b değerleri sırasıyla giriş maskesi ve çıkış maskesi olarak isimlendirilir.(4.4) ifadesinde herhangi bir giriş ve çıkış maskesi değerine göre LAT tablosu değerinin nasıl elde edileceği verilmiştir.

$$N_L(\Gamma_a, \Gamma_b) = \#\left\{x \in GF(2^n) | \Gamma_a \bullet x = \Gamma_b \bullet S(x)\right\} \tag{4.3}$$

$$LAT(\Gamma_a, \Gamma_b) = \#\left\{x \in GF(2^n) | \Gamma_a \bullet x = \Gamma_b . S(x)\right\} - 2^{n-1} \tag{4.4}$$

Diğer yandan bir S-kutusu için doğrusal olmama ölçüsü NLM_s değeri LAT değeri ile ilişkili olarak (4.5)'te verilmiştir.

$$NLM_S = 2^{n-1} - maks\,|\,LAT_S(\Gamma_a, \Gamma_b)\,| \tag{4.5}$$

Örnek 4.1. Tablo 4.1 de 4×4 boyutunda bir S-kutusu gözükmektedir. Bu S-kutusu $x^4 + x + 1$ indirgenemez polinomu ve $x \to x^7$ üs haritalama fonksiyonu ile üretilmiştir. S-kutusunun doğrusal yaklaşım tablosu Tablo 4.2'de ki gibi verilebilir.

Tablo 4.1 4×4 Boyutundaki Bir S-kutusu

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Giriş	0000	0001	0010	0011	0100	0101	00110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Çıkış	0011	1101	1010	0010	0001	0111	1011	0101	1100	1110	1111	00110	1001	1000	0000	1001
Hex	3	D	A	2	1	7	B	5	C	E	F	6	9	8	0	9

Tablo 4.2 Doğrusal Yaklaşım Tablosu (LAT)

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

Örnek olarak LAT(7,C) değerinin elde edilişi aşağıda gösterilmektedir:

$$(0111).(0000) = (0111).(0000) \rightarrow 0 = 0 \quad *$$

$$(0111).(0001) = (1100).(0011) \rightarrow 1 \neq 0$$

$$(0111).(0010) = (1100).(1101) \rightarrow 1 = 1 \quad *$$

$$(0111).(0011) = (1100).(1010) \rightarrow 0 = 0 \quad *$$

$$(0111).(0100) = (1100).(1110) \rightarrow 1 \neq 0$$

$$(0111).(0101) = (1100).(0001) \rightarrow 0 \neq 1$$

$$(0111).(0110) = (1100).(1111) \rightarrow 0 \neq 1$$

$$(0111).(0111) = (1100).(1011) \rightarrow 1 = 1 \quad *$$

$$(0111).(1000) = (1100).(0101) \rightarrow 0 = 0 \quad *$$

$(0111).(1001) = (1100).(1100) \rightarrow 1 \neq 0$

$(0111).(1010) = (1100).(1110) \rightarrow 1 \neq 0$

$(0111).(1011) = (1100).(1111) \rightarrow 0 \neq 1$

$(0111).(1100) = (1100).(0110) \rightarrow 1 = 1$

$(0111).(1101) = (1100).(1000) \rightarrow 0 \neq 1$

$(0111).(1110) = (1100).(0000) \rightarrow 0 = 0$

$(0111).(1111) = (1100).(0100) \rightarrow 1 = 1$

(4.3) ifadesine göre $\mathrm{LAT}(7,\mathrm{C})$ için eşitliği sağlayan (* ile işaretlenmiş olanlar) 8 değer bulunmaktadır. (4.4)'teki ifadeyi kullanarak $\mathrm{LAT}(7,\mathrm{C}) = 8 - 2^{4-1} = 8 - 8 = 0$ şeklinde elde edilir. S-kutusunun (4.5) ifadesine göre $\mathrm{NLM_s}$ değeri ise tüm LAT elemanlarının en büyük mutlak değeri göz önüne alınarak $\mathrm{NLM_s} = 2^{4-1} - 4 = 4$ şeklinde elde edilir.

4.1.3. Fark Dağılım Tablosu (Difference Distribution Table)

Diferansiyel kriptanaliz bir blok şifreleme algoritmasına karşı kullanılan bir saldırı yöntemidir (Stinson D. R., 2002). S-kutularının fark dağılım tablosu (XOR tablosu veya DDT) bu saldırıya karşı şifrenin gücü ile ilgili fikirler vermektedir. n × m boyutunda bir S-kutusu için XOR tablosu (Stinson D. R.,2002) , (Phan R. C.-W., 2004) $2^n \times 2^m$ matrise denk düşer.

$S : \mathrm{GF}(2^n) \rightarrow \mathrm{GF}(2^n)$ olmak üzere n bit giriş ve n bit çıkışa sahip bir S-kutusu olmak üzere herhangi verilen $a, b \in \mathrm{GF}(2^n)$ için $\mathrm{XOR}(a,b)$ herhangi a \ 0 ve b için

$S(x) + S(x \oplus a) = b$ denklemindeki b değerlerinin sayısını tanımlar ve (4.6)'daki gibi gösterilebilir.

Bir S-kutusu için denklem (4.6)'da a ve b değerleri sırasıyla giriş farkı ve çıkış farkı olarak isimlendirilir.

$$XOR(a,b) = \#\left\{x \in GF(2^n) | S(x) + S(x \oplus a) = b\right\} \quad (4.6)$$

Örnek 4.2: Örnek 4.1 de kullanılan S-kutusunun XOR tablosu Tablo 4.3 teki gibidir.

Tablo 4.3 Fark Dağılım Tablosu (XOR tablosu)

		ÇIKIŞ FARKI(b)															
GİRİŞ FARKI(a)		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	2	2	0	2	0	2	0	2	2	0	0	2	0	4	0
	2	0	0	2	2	0	0	0	0	2	4	2	0	2	0	0	2
	3	0	4	0	0	2	2	0	2	2	0	2	0	0	2	0	0
	4	0	2	4	0	0	0	2	2	0	0	2	0	2	0	0	2
	5	0	0	0	0	4	2	2	2	0	2	0	2	2	0	0	2
	6	0	0	0	0	0	0	2	0	4	0	2	0	0	2	2	0
	7	0	0	0	2	0	0	4	2	2	0	0	2	0	0	2	2
	8	0	2	0	2	2	2	0	0	2	0	0	0	2	4	0	4
	9	0	2	0	0	0	2	0	0	0	2	0	2	4	2	2	2
	A	0	2	2	2	0	0	2	0	0	0	0	2	2	0	0	0
	B	0	0	2	2	2	4	0	2	0	0	0	0	0	2	2	0
	C	0	0	0	2	2	2	2	0	0	2	4	0	0	2	0	0
	D	0	0	2	0	2	0	0	0	0	0	2	4	0	2	2	2
	E	0	2	0	4	0	0	0	2	0	2	2	2	0	0	2	0
	F	0	0	2	0	0	2	0	4	2	2	0	2	0	2	0	0

(4.6) ifadesi gereği elde edilen $XOR_s(8,F) = 4$ sonraki sayfada verilmektedir.

$S(0) \oplus S(8) = 3 \oplus C \rightarrow F$ *

$S(1) \oplus S(9) = D \oplus E \rightarrow 3$

$S(2) \oplus S(A) = A \oplus F \rightarrow 5$

$S(3) \oplus S(B) = 2 \oplus 6 \rightarrow 4$

$S(4) \oplus S(C) = 1 \oplus 9 \rightarrow 8$

$S(5) \oplus S(D) = 7 \oplus 8 \rightarrow F$ *

$S(6) \oplus S(E) = B \oplus 0 \rightarrow B$

$S(7) \oplus S(F) = 5 \oplus 4 \rightarrow 1$

$S(8) \oplus S(0) = C \oplus 3 \rightarrow F$ *

$S(9) \oplus S(1) = E \oplus D \rightarrow 3$

$S(A) \oplus S(2) = F \oplus A \rightarrow 5$

$S(B) \oplus S(3) = 6 \oplus 2 \rightarrow 4$

$S(C) \oplus S(4) = 9 \oplus 1 \rightarrow 8$

$S(D) \oplus S(5) = 8 \oplus 7 \rightarrow F$ *

$S(E) \oplus S(6) = 0 \oplus B \rightarrow B$

$S(F) \oplus S(7) = 4 \oplus 5 \rightarrow 1$

Yukarıdaki işlemlerden de görüldüğü gibi $S(x) + S(x+8) = F$ eşitliğini sağlayan (* ile işaretli olanlar) 4 durum vardır.

Sonlu cisimde ters alma işlemi (Nyberg K., 1994) üs haritalama işleminin özel bir durumu olarak görülebilir ve bu iki teknik ile doğrusal olmama ölçüsü yüksek ve diğer kriptografik özellikleri iyi S-kutuları elde edilebilir. Bunun yanında bu tasarım teknikleri kullanılarak tasarlanan S-kutuları monomial tabanlı polinomlara dayalı üs haritalama ve ters alma gibi cebirsel işlemler olduğu için doğrusal denklik (Fuller J., Millan W., 2003), (Youssef A. M., Tavares S. E., 2005) ve S-kutularının cebirsel

ifadesinde bazı basit cebirsel yaklaşımlar (Youssef A. M., Tavares S., Gong G., 2006) gibi istenmeyen özellikleri de beraberinde getirmektedir.

Günümüzde blok şifrelerin içyapısında kullanılan S-kutuları genellikle 4×4 (4-bit giriş 4-bit çıkış) ya da 8×8 (8-bit giriş 8-bit çıkış) büyüklüğündedir. Örneğin AES blok şifresi sonlu cisimde ters haritalama yöntemiyle elde edilen 8×8 büyüklüğünde bir S-kutusu Tablo 4.4 de gösterildiği gibi kullanmaktadır.

4.2. Geliştirilen Şifrede Kullanılan S kutusunun Kriptografik Özellikleri

Yapılan incelemelerden S-kutuları ile ilgili olarak literatürdeki çalışmalarda cebirsel S-kutularının kullanıldığını görülmüştür. Örneğin AES S-kutusu, ARIA şifresinde kullanılan S-kutuları ve Camellia (Aoki, K., Ichikawa, T. vd, 2001) şifresinde kullanılan S-kutusu cebirsel olarak tasarlanmıştır. Özellikle sonlu cisimde ters haritalama kullanılarak üretilen S-kutuları kriptografik özellikler açısından (doğrusal olmama, LAT dağılımı, DDT tablosu vb.) iyi sonuçlar vermektedir (Aslan B., Sakallı M. T. 2008).

$GF(2^8)$ cisminde ($x^8+x^4+x^3+x+1$ ile tanımlı) ters haritalama tabanlı AES S-kutusunun bu kriterler açısından özellikleri aşağıdaki gibi verilebilir:

- 255×255 boyutundaki Doğrusal Yaklaşım Tablosunda (LAT- Linear Approximation Table) en büyük elemanın mutlak değeri 16'dır.
- Doğrusal olmama değeri (4.5) ifadesinden faydalanılarak $2^{n-1}-maks\left|LAT(\Gamma_A,\Gamma_B\right|$ hesaplanırsa 128-16 = 112 olarak elde edilebilir. 8×8 büyüklüğünde bir S-kutusunun doğrusal olmama değeri en fazla $2^{n-1}-2^{\frac{n}{2}-1}$ ifadesinden 120 olacağından AES S-kutusu % 93 doğrusal olmama değerine sahiptir.

- 255×255 boyutundaki Fark Dağılım Tablosundaki (DDT-Difference Distribution Table-Fark Dağılım Tablosu) en büyük elemanın değeri 4'tür.
- Polinomsal ifadesindeki terim sayısı 9'dur.

Özellikle AES S-kutusu incelendiğinde, polinomsal cebirsel ifadesi daha iyi olabilecek S-kutularının geliştirilebileceği literatürde bulunan çalışmalar yoluyla gözlenmiştir (Sakallı M.T., Aslan B. vd., 2010)(Aslan B., 2008). Bu nedenle kendi geliştireceğimiz şifre için kullanılacak S-kutusu bahsedilen kriterler ile birlikte interpolasyon saldırıları gibi cebirsel saldırılara karşı kuvvetli bir S-kutusu seçilecektir. AES S-kutusu ile aynı kriptografik özelliklere sahip, AES S-kutusundan polinomsal ifadesi terim sayısı açısından daha iyi olan ve geliştirilecek şifrede kullanılacak S-kutusu Tablo 3.1 de verilmiştir (M. T. Sakallı, B. Aslan vd, 2010). Buna ek olarak verilen S-kutusunun cebirsel ifadesi de Ek-A'da gösterilemektedir.

Tablo 4.4. Geliştirilecek şifrede kullanılan S-kutusu

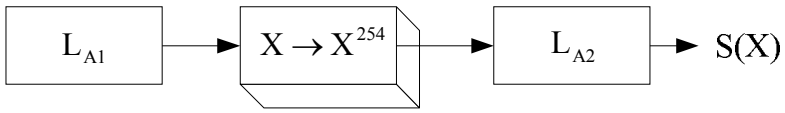
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	c3	18	27	80	15	34	fd	f7	2b	fe	6b	77	f0	ca	d4	72
1	1a	1b	e3	d6	cf	6a	d1	b1	21	10	9d	40	85	d0	f9	9f
2	66	48	c1	57	8a	e8	78	b4	e9	ce	d9	98	68	8c	99	bb
3	0a	49	95	ac	08	6c	c8	4e	14	de	2a	4f	17	cd	a7	19
4	89	e6	b0	0f	28	1e	e1	94	74	bd	1c	2e	f6	3e	61	9e
5	13	97	64	3d	0b	ee	60	88	f4	7a	8d	6d	24	32	c2	79
6	c9	59	9c	af	ab	01	63	c5	e5	d8	36	26	05	c7	07	75
7	aa	4d	50	7f	f3	b6	51	f5	be	4c	20	ed	5a	83	52	84
8	e7	a9	ae	56	91	62	3a	06	c4	73	44	0c	22	dc	b8	5e
9	ba	c6	8b	dd	86	b9	b5	03	41	16	42	a1	69	11	87	55
a	53	5b	58	cb	29	b3	2c	6e	45	a8	33	ef	92	8f	da	ff
b	b7	cc	31	a5	eb	e2	23	96	ad	c0	47	82	f2	7b	67	d7
c	a3	38	d2	bc	3c	02	fb	43	3b	2f	a0	09	fc	00	39	4a
d	7c	6f	76	30	a4	a2	7d	fa	12	b2	9a	04	3f	93	f1	71
e	81	90	db	46	5d	7e	ec	5f	d3	e4	5c	e0	d5	37	ea	65
f	f8	8e	df	9b	54	2d	0d	bf	35	1d	0e	70	a6	25	1f	4b

Tablo 4.5 de verilen ve geliştirilen blok şifrede kullanılacak S-kutusu, AES S-kutusunda olduğu gibi $x \to x^{254}$ üs haritalaması kullanılarak tasarlanmıştır. AES S-kutusundan farklı olarak (4.7) ve (4.8) ifadelerinde verilen doğrusal dönüşümler sırasıyla bu haritalamadan önce ve bu haritalamadan sonra uygulanarak S-kutusu tasarımı sonuçlandırılmıştır. Doğrusal dönüşümlerde kullanılan hexadecimal notasyonda gösterilen sabitler sırası ile $L_{A_{S1}}$ ="33" ve $L_{A_{S1}}$ ="63" şeklindedir. S-kutusunun tasarımı aşağıdaki maddeler halinde verilebilir:

Adım 1. $P = L_{A1}(x) = L_{A1(8\times8)}.(x_0, x_1, ..., x_7)^T \oplus L_{A_{S1}}$.

Adım 2. $K = (P)^{254}$.

Adım 3. $S(x) = L_{A2(8\times8)}.(K_0, K_1, ..., K_7)^T \oplus L_{A_{S2}}$.



Şekil 4.1 Kullanılan S-kutusunun Tasarım Yapısı

Bu doğrusal dönüşümlerin uygulanması $X \to X^{254}$ üs haritalamasının sağladığı iyi kriptografik özellikleri değiştirmediği gibi AES S-kutusunun polinomsal ifadesindeki terim sayısını 9'dan 255'e çıkarmaktadır. Bu doğrusal dönüşümler tersi alınabilir ikili dönüşümlerdir. 8-bit giriş 8-bit çıkışlı bir S-kutusu üretilmek istendiğinden dolayı 8×8'lik doğrusal dönüşüm kullanılmıştır. Dönüşümlerdeki $x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7$ değerleri ikili biçimde yazılmış bitlerdir. Buna ek olarak geliştirilen şifrede kullanılan S-kutusu AES şifresinde kullanılan S-kutusunda olduğu gibi sabit nokta içermemektedir.

$$L_{A1}(x) = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.7}$$

$$L_{A2}(x) = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \tag{4.8}$$

Elde edilen S-kutusunun polinomsal ifadesi 2 farklı yöntem ile elde edilebilir:

- Lagrange interpolasyonu,
- Sonlu cisimler aritmetiği.

Geliştirilen şifrede kullanılan S-kutusunun polinomsal cebirsel ifadesi EK A'da verilmiştir. Bu ifade bir yazılım vasıtasıyla Lagrange interpolasyonu kullanılarak elde edilmiş ve (M. T. Sakallı, B. Aslan, E. Buluş, A. Ş. Mesut, F. Büyüksaraçoğlu, Osman Karaahmetoğlu, 2010) çalışmasında verilen ile karşılaştırılarak doğrulanmıştır. Lagrange İnterpolasyonu ile cebirsel olarak tasarlanan bir S-kutusunun polinomsal ifadesinin elde edilmesi Bölüm 3.2.1 de kısaca gösterilmektedir. Diğer yandan sonlu cisimler aritmetiği ile S-kutusunun cebirsel ifadesinin hesabı ile ilgili detaylı bilgi (Aslan B., 2008) çalışmasından elde edilebilir.

4.2.1. Lagrange İnterpolasyonu

Lagrange interpolasyonu, n noktadan geçen $n-1$'inci dereceden polinomu bulmak için kullanılır. Dolayısı ile $(x_0, y_0), (x_1, y_1), ..., (x_{n-1}, y_{n-1})$ değerleri biliniyor ise (4.10) ifadesi ile polinomun cebirsel açılımı elde edilebilir.

$$P(X) = \sum_{j=0}^{n} y_j \ell_j(X) \tag{4.10}$$

(4. 10) ifadesindeki $\ell_j(X)$değerinin açılımı (4.11) ifadesindeki gibidir.

$$\ell_j(X) = \prod_{i=0, i\neq j}^{k} \frac{x - x_i}{x_j - x_i} = \frac{x - x_0}{x_j - x_0} \cdot \cdots \cdot \frac{x - x_{j-1}}{x_j - x_{j-1}} \cdot \frac{x - x_{j+1}}{x_j - x_{j+1}} \cdot \cdots \cdot \frac{x - x_k}{x_j - x_k} \tag{4.11}$$

(4.12) ifadesi Lagrange interpolasyonunun nasıl hesaplandığını açık şekilde göstermektedir:

$$\begin{aligned} P(X) = {} & \frac{(X-x_2)(X-x_3)(X-x_4)...(X-x_n)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)...(x_1-x_n)} y_1 + \\ & \frac{(X-x_1)(X-x_3)(X-x_4)...(X-x_n)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)...(x_2-x_n)} y_2 + \\ & \frac{(X-x_1)(X-x_2)(X-x_4)...(X-x_n)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)...(x_3-x_n)} y_3 + \\ & \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots + \\ & \frac{(X-x_1)(X-x_2)(X-x_4)...(X-x_{n-1})}{(x_n-x_1)(x_n-x_2)(x_n-x_4)...(x_n-x_{n-1})} y_n \end{aligned} \tag{4.12}$$

Lagrange interpolasyonu ile bir S-kutusunun cebirsel ifadesinin açılımı bulunabilir. Fakat bu yapılırken sonlu cisim aritmetiği dikkate alınmalıdır ve sonlu cisim üzerinde işlem yapılmalıdır.

4.3. Doğrusal Dönüşümler

Doğrusal dönüşümler bir blok şifreye yayılım eklemek için kullanılan elemanlardır. Blok şifre mimarilerinin (Feistel ve SPN) her ikisinde de kullanılabilirler. Doğrusal dönüşümler sabit uzunluktaki bir giriş bloğunu doğrusal olarak karıştırarak aynı uzunlukta bir çıkış bloğu elde etmeyi sağlar (Z'aba M. R., 2010). Doğrusal dönüşümlerin sağladığı yayılımın ölçülmesi için var olan teknikler aşağıdaki gibi sıralanabilir:

- Çığ Etkisi (Avalanche ciriterion) (Feistel H., 1973),
- Katı çığ etkisi (Strict avalance criterion) (Webster A. F., Tavares S. E., 1986),
- Bütünlük (Completeness) (Kam J. B., Davida G. I., 1979),
- Dallanma sayısı (Branch number) (Daemen J., Rijmen V., 2002),
- Sabit noktalar (Fixed points) (Z'aba M. R., 2010).

Dallanma sayısı: Bir blok şifrede iki ardışık döngüde aktif S-kutularının minimum sayısını temsil eder. Bu sayı, bir blok şifreye karşı uygulanacak doğrusal ve diferansiyel saldırıların başarımını ölçmek için kullanılır. Çoğu yayılım elemanı doğrusal dönüşümlerdir ve matrisler ile temsil edilirler. Dolayısıyla bir yayılım elemanını $A:\left(\{0,1\}^m\right)^n \rightarrow \left(\{0,1\}^m\right)^n$ şeklinde tanımlanabilir ve (4.13) ifadesindeki gibi gösterilebilir:

$$A(x) = A.x^T = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ \dots \\ x_n \end{bmatrix} \tag{4.13}$$

Yukarıdaki verilen (4.13) ifadesinde $x = (x_1, x_2, \dots x_n)^T$, $x_i \in \{0, 1\}^m$, $i = 1, \dots, n$ şeklindedir. Buna ek olarak n bir yayılım elemanındaki S-kutularının sayısını temsil

eder ve her S-kutusunun giriş ve çıkış genişliği *m*-bittir. *A* matrisinin elemanları $GF(2^m)$ cisminin elemanlarıdır (Özellikle sonlu cisim ya da $GF(2)$ nin elemanları olabilir). Örneğin AES'in doğrusal dönüşümü 4×4 boyutundaki MixColumns (Sütunları karıştırma) dönüşümünün elemanları sonlu cisim $GF(2^8)$'in elemanları iken ARIA (Kwon D., Kim J., vd., 2004.) şifresinde kullanılan 16×16 boyutundaki doğrusal dönüşümün elemanları $GF(2)$'nin elemanlarıdır.

Diğer yandan bir $n\times n$ boyutundaki A matrisinin dallanma sayısı (4.14) ifadesinde gösterildiği gibi tanımlanabilir:

$$\beta(A)=\min\left\{wt(x)+wt(A.x^T)\,|\,x\in\left(\{0,\,1\}^m\right)^n,\ x\neq0\right\} \tag{4.14}$$

Blok şifrelerin tasarımında optimal dallanma sayısına sahip yayılım elemanları tercih edilmektedir. $GF(2^m)$ cisim elemanlarından oluşturulan doğrusal dönüşüm yapılarının optimal dallanma sayınsa sahip olması da bu matrislerin MDS (Maximum Distance Seperable) özelliğine sahip olması ile sağlanmaktadır. Örneğin, AES blok şifresi elemanları $\mathrm{GF}\left(2^8\right)$ den olan 4×4 MDS matris kullanarak 32-bitten 32-bite dönüşüm yapan bir yayılım elemanı kullanırken, Khazad (Barreto P. S. L. M., RijmeN V., 2000) blok şifresi yine elemanları $\mathrm{GF}\left(2^8\right)$ den olan 8×8 MDS matris kullanarak 64-bitten 64-bite dönüşüm yapan bir yayılım elemanı kullanır. Bu şifreler için dallanma sayısı AES'in yayılım elemanı için 5 iken Khazad blok şifresi için 9 dur (Daemen J., Rijmen V., 2002), (Barreto P. S. L. M., RijmeN V., 2000).

Yardımcı Önerme 4.1. Bir doğrusal $[n,k,d]$-kod $d\le n-k+1$ Singleton sınırını karşılıyorsa bu koda MDS kod adı verilir. Alternatif olarak, bir matrisin MDS matris olabilmesi için satır ve sütunlarından oluşturulan tüm alt matrislerinin determinantının 0'dan farklı olması gerekir (Nakahara Jr. J., Abrahao E., 2009), (Youssef A. M., Mister S., Tavares S. E., 1997).

Yardımcı Önerme 4.1 den yola çıkarak elemanları $GF(2^m)$'den olan bir $n \times n$ matrisin tüm hesaplanması gereken alt matrislerinin determinantlarının sayısı:

$$\sum_{k=1}^{n-2}\left[C\binom{n}{n-k} \right]^2 \tag{4.15}$$

(4.15) ifadesindeki gibi verilebilir. Örneğin 4×4 bir matrisin hesaplanması gereken alt determinantlarının sayısı 2×2 alt determinantlarının sayısı ile 3×3 alt determinantlarının sayısının toplamı şeklinde elde edilebilir. Bu da yukarıdaki ifadeden $\left(C\binom{4}{2} \right)^2 + \left(C\binom{4}{3} \right)^2 = 36 + 16 = 52$ şeklinde elde edilebilir.

AES şifreleme algoritmasında kullanılan MDS matrisi ve tersi Şekil 4.2' de verilmiştir. AES şifreleme algoritmasında $x^8 + x^4 + x^3 + x + 1$ indirgenemez polinomu kullanılarak tüm alt matris determinantları hesaplanarak bu determinantların hiçbirinin 0 olmadığı gözlenmiştir.

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

A

Şekil 4.2. AES şifreleme Algoritmasında kullanılan MDS matris

Diğer yandan blok şifrelerin tasarımında kullanılan doğrusal dönüşümlerde iki önemli matris tipi bulunmaktadır. Bunlar dairesel (circulant) ve Hadamard matrislerdir. Örneğin AES blok şifresinde kullanılan sütunları karıştırma dönüşümü dairesel matris türüne girerken Khazad şifresinde kullanılan matris, Hadamard matris türüne girmektedir. Her iki matris tipi ile MDS matrisler Yardımcı Önerme 1 ile elde

edilebilirken bu tür matrisler arasındaki temel fark matrislerin involutif matris (şifreleme ile deşifreleme de aynı elemanın kullanılması) olup olmamasında ortaya çıkmaktadır. Örneğin dairesel matrisler ile hem MDS hem de involutif yapılar elde edilemez. Ancak Hadamard matrisler ile bu mümkün olmaktadır. Hadamard matrislerin kullanılmasının bir avantajı şifreleme ve deşifreleme performansının aynı olduğu şifreleme algoritmalarının tasarımının yapılabilmesine olanak sağlamasıdır.

Yardımcı Önerme 4.2. $a_1, a_2, ..., a_t$ $GF(2^m)$ cisminin elemanları olsun. O zaman

$$(a_1 + a_2 + ... + a_t)^{2^k} = {a_1}^{2^k} + {a_2}^{2^k} + ... + {a_t}^{2^k} \tag{4.16}$$

şeklinde ifade edilebilir (Aslan B., Sakallı M. T., Buluş E., 2008).

Yardımcı Önerme 4.3. $Had(a_1, a_2, a_3, a_4) = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}_{4\times 4}$ elemanları $GF(2^m)$ sonlu cisminden olan Hadamard formunda 4×4 büyüklüğünde bir matris olsun. Eğer $\bigoplus_{i=1}^{4} a_i = 1$, ya da diğer bir deyişle kullanılan elemanların XOR toplamı 1, ise verilen matris involutif bir matristir.

İspat: Aşağıdaki denklemde gösterildiği gibi A^2 matrisi $\bigoplus_{i=1}^{4} a^2{}_i = 1$ olduğu durumda birim matris olarak elde edilebilir. Buna ek olarak Yardımcı Önerme 4.2'yi kullanarak $\bigoplus_{i=1}^{4} a^2{}_i = \bigoplus_{i=1}^{4} a_i = 1$ olduğundan verilen matris $\bigoplus_{i=1}^{4} a_i = 1$ koşulu altında involutif bir matristir.

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

$$= \begin{bmatrix} \bigoplus_{i=1}^{4} a_i^2 & 0 & 0 & 0 \\ 0 & \bigoplus_{i=1}^{4} a_i^2 & 0 & 0 \\ 0 & 0 & \bigoplus_{i=1}^{4} a_i^2 & 0 \\ 0 & 0 & 0 & \bigoplus_{i=1}^{4} a_i^2 \end{bmatrix}$$

Yardımcı Önerme 4.1 ve Yardımcı Önerme 4.3 kullanılarak involutif ve MDS matrisler elde edilebilir. Örnek 4.3' te 4×4 büyüklüğünde ve elemanları $GF(2^8)$ den olan involutif bir MDS matris verilen önermeler ışığı altında elde edilmiştir.

Örnek 4.3: Elemanları $x^8+x^4+x^3+x+1$ ile tanımlı $GF(2^8)$ sonlu cisminden olan 4×4 genişliğinde biri dairesel formda ve diğeri Hadamard formda 32-bitten 32-bite dönüşüm yapan iki doğrusal dönüşümü örnek olarak gösterelim.

Dairesel formdaki matrisler ilk satırdaki elemanların birer sağa ötelenerek diğer satırların oluşturulduğu matrislerdir. Örneğin;

$$D(01_h,02_h,04_h,06_h) = \begin{bmatrix} 01_h & 02_h & 04_h & 06_h \\ 06_h & 01_h & 02_h & 04_h \\ 04_h & 06_h & 01_h & 02_h \\ 02_h & 04_h & 06_h & 01_h \end{bmatrix}_{4\times4}$$

elemanları $GF(2^8)$ sonlu cisminden dairesel formda bir matristir. Yine daha önce verilen Yardımcı Önerme 4.1 kullanılarak tüm alt matrislerin alt determinantları tanımlanan indirgenemez polinomuna göre incelendiğinde 52 alt determinanttan $\begin{vmatrix} 02_h & 04_h \\ 01_h & 02_h \end{vmatrix} = 0$, $\begin{vmatrix} 04_h & 02_h \\ 02_h & 01_h \end{vmatrix} = 0$, $\begin{vmatrix} 01_h & 02_h \\ 02_h & 04_h \end{vmatrix} = 0$ üç alt determinantın değeri 0 olduğu için

verilen örnek dairesel matris MDS matris değildir. Diğer yandan Hadamard matris formu 4×4 büyüklüğündeki bir matris için aşağıdaki gibi verilebilir:

$$Had(a_1,a_2,a_3,a_4)=\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_2 & a_1 & a_4 & a_3 \\ a_3 & a_4 & a_1 & a_2 \\ a_4 & a_3 & a_2 & a_1 \end{bmatrix}_{4\times 4}$$

Aynı elemanlar ve indirgenemez polinom ile tanımlı Hadamard formdaki 4×4 büyüklüğündeki matrisi düşünelim:

$$Had(01_h,02_h,04_h,06_h)=\begin{bmatrix} 01_h & 02_h & 04_h & 06_h \\ 02_h & 01_h & 06_h & 04_h \\ 04_h & 06_h & 01_h & 02_h \\ 06_h & 04_h & 02_h & 01_h \end{bmatrix}_{4\times 4}$$

Daha önce verilen Yardımcı Önerme 4.1' i kullanılarak verilen matrisin tüm alt determinantları tanımlanan indirgenemez polinomuna göre incelendiğinde 52 alt determinantın tümünün 0'dan farklı olduğu gözlenebilir. Dolayısıyla bu yayılım elemanı MDS bir matristir ve dallanma sayısı optimal değer olan 5'tir. Buna ek olarak Hadamard formundaki bu 4×4 matrisin elemanlarının XOR toplamı da 1dir. Yardımcı Önerme 4.3 gereği verilen matris aynı zamanda involutif bir matristir.

Diğer yandan Camellia (Aoki K., Ichikawa T., Kanda M., vd., 2001) ve ARIA şifresinde olduğu gibi yüksek dallanma sayısına sahip MDBL (Maximum Distance Binary Linear Codes) kodlarda yayılım elemanı olarak kullanılmaktadır. Bu ikili doğrusal dönüşümler verilen şifrelerde sırasıyla 8×8 ve 16×16 boyutunda matrislerdir. Bunun yanında bu boyutlardaki matrisler için optimal dallanma sayısı değerleri sırasıyla 5 ve 8 olduğu (Kang J-S, Hong S., Lee S., vd., 2001), (Kwon D., Sung S. H., vd., 2005) çalışmalarında verilmiştir.

Sabit Noktalar (Fixed Points): Doğrusal dönüşümlerde *sabit noktaların* önemi (Z'aba M.R., 2010) çalışmasında verilmiştir. Bu çalışmada doğrusal dönüşümdeki sabit nokta sayısı bir rastsal doğrusal dönüşümde olması gerekenden çok daha fazla sayıda olursa bunun yayılım elemanının kötü bir yayılım sağladığının göstergesidir denmektedir. Rastsal bir doğrusal dönüşümde beklenen sabit nokta sayısı 1'dir.

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 (4.17) ifadesindeki denklemin çözülmesi ile elde edilebilir:

$$(A-I)x^T = 0 \tag{4.17}$$

(4.17) ifadesinde 0, n uzunluğunda tüm elemanları 0 olan vektörü temsil etmektedir. Buradan yola çıkarak sabit noktaların sayısı (4.18) ifadesindeki gibi verilebilir:

$$F_A = 2^{m(rank(A)-rank(A-I)} = 2^{m(n-rank(A-I))} \tag{4.18}$$

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

4.4. Geliştirilen Blok Şifrede Kullanılan Doğrusal Dönüşüm

İncelenen doğrusal dönüşümler literatürde farklılıklar göstermektedir. Örneğin, AES şifresi, MDS (Maximum Distance Separable) ve 4×4 byte matrisi şeklinde bir doğrusal dönüşüm kullanırken, ARIA şifresi 16×16 ikili bir matris kullanmaktadır.

Aşağıda bu şifrelerin kullandığı dönüşümler verilmiştir:

Tablo 4.6. AES,KHAZAD,Camellia ve ARIA şifrelerinde kullanılan yayılım katmanları

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

Bu kullanılan dönüşümlerde kriptografik kriter olan *branch number* (dallanma sayısı) değeri AES şifresi için 5, Khazad şifresi için 9, Camellia şifresi için 5, ARIA şifresi için 8'dir.

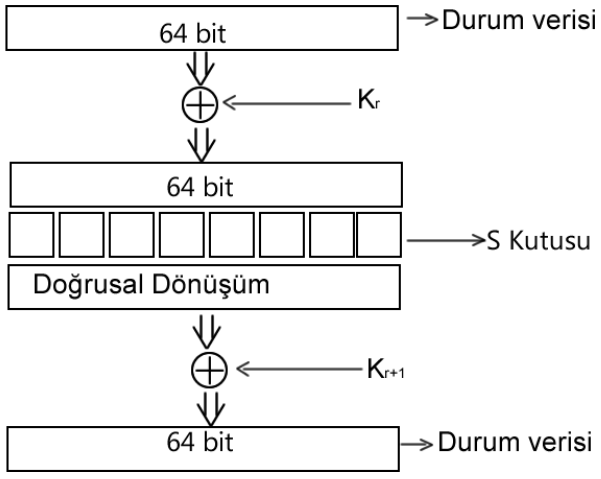
Geliştirilecek olan şifrede kullanılması düşünülen doğrusal bileşenin özellikleri;

- İndirgenemez polinom $x^8+x^4+x^3+x+1$ ile tanımlı $GF(2^8)$ üzerine 8×8 MDS matris,
- İnvolutif matris.

Aşağıda verilen doğrusal dönüşüm Yardımcı Önerme 4.3 kullanılarak elde edilmiştir ve doğrusal dönüşüm MDS bir matris olduğu için dallanma sayısı değeri 9'dur.

$$A_1 = \begin{bmatrix} 01_h & 02_h & 05_h & 04_h & 06_h & 0B_h & 09_h & 07_h \\ 02_h & 01_h & 04_h & 05_h & 0B_h & 06_h & 07_h & 09_h \\ 05_h & 04_h & 01_h & 02_h & 09_h & 07_h & 06_h & 0B_h \\ 04_h & 05_h & 02_h & 01_h & 07_h & 09_h & 0B_h & 06_h \\ 06_h & 0B_h & 09_h & 07_h & 01_h & 02_h & 05_h & 04_h \\ 0B_h & 06_h & 07_h & 09_h & 02_h & 01_h & 04_h & 05_h \\ 09_h & 07_h & 06_h & 0B_h & 05_h & 04_h & 01_h & 02_h \\ 07_h & 09_h & 0B_h & 06_h & 04_h & 05_h & 02_h & 01_h \end{bmatrix}.$$

A_1 matrisi geliştirilirken Yardımcı Önerme 4.1 gereği matrisin MDS matris olduğundan emin olmak için tüm alt matrislerinin determinantları incelenmiş ve herhangi birinin 0 olmadığı bir program aracılığı ile test edilmiştir. Bu test esnasında A_1 matrisi için 12804 alt matris determinantı incelenmiştir. Buna ek olarak (A_1-I) matrisinin rank değeri de test edilmiş ve 4 olarak elde edilmiştir. Dolayısıyla (4.18) ifadesi gereği bu doğrusal dönüşüm 2^{32} adet sabit nokta içermektedir. Diğer bir deyişle bu dönüşümle yapılan 2^{32} haritalamadan 1'inde bir sabit noktaya rastlanacaktır. Aşağıdaki Şekil 4.3'te 8×8 MDS matrisin (doğrusal dönüşümün) geliştirilen blok şifrede tek döngü adımındaki kullanımı gösterilmektedir.



Şekil 4.3. Geliştirilen Şifrenin Bir Döngü için Blok Diyagramı

4.5. Anahtar Genişletme Algoritmaları

Bir blok şifre daha önce de belirtildiği gibi döngülerden ve döngülerdeki aynı adımlardan oluşmaktadır. Dolayısıyla döngülerdeki simetriyi bozmak için her döngüde farklı bir anahtar materyalinin kullanılması gereklidir. Anahtar genişletme algoritmaları gizli anahtardan her döngüde kullanılacak farklı anahtarların (alt anahtarların) elde edilmesini sağlayan algoritmalardır. Her blok şifrede farklı algoritmalar kullanılabilmektedir ve şifreleme algoritmasında kullanılan yapılar tercih edilerek bu algoritmalar geliştirilebilir. Lars Knudsen (L. R. Knudsen, 2000) güçlü bir anahtar planlamanın özelliklerini aşağıdaki gibi vermektedir:

- Çarpışmaya dayanıklı tek yönlü fonksiyon (one-way function) olma,
- Tüm alt anahtarlar ve gizli anahtar arasında minimum karşılıklı ilişki bulunma,
- Uygulama etkinliği.

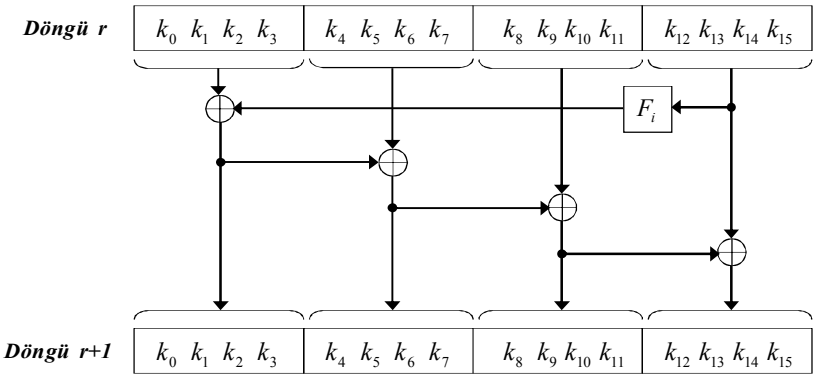
Bir blok şifre, gizli anahtar bilinmediği zaman tek yönlü bir fonksiyon olarak düşünülebilir. Dolayısıyla alt anahtarların üretilmesinde şifreleme algoritmasının kullanımı alt anahtarlarının tersinirlik (invertibility) özelliğinin olamaması için yeterli bir teknik olarak düşünülebilir (May L., Henricksen M., vd., 2002).

Tüm alt anahtarlar ve gizli anahtar arasında minimum karşılıklı ilişki özelliği blok şifreler üzerine saldırı senaryolarının karmaşıklığını azaltarak saldırgana yardımcı olacak ilişkileri yok edecektir (May L., Henricksen M., vd., 2002). Bu tür ilişkilerin kullanıldığı saldırılara örnekler DES blok şifresinde karşı doğrusal kriptanaliz, diferansiyel kriptanaliz gibi saldırılar ile AES blok şifresine karşı olan çeşitli saldırılar verilebilir. (Ferguson N., Kelsey J., vd., 2000) çalışmasının yazarları "Bazı saldırıların genişletilen anahtar bitleri arasındaki ilişkileri kullandıklarını ve bu ilişkilerin olamaması durumunda saldırıların daha yüksek karmaşıklık gerektireceğini" belirtmişlerdir.

Şifreleme algoritması ve anahtar genişletme algoritması güvenlik açısından olduğu kadar uygulama yönüyle de birbirlerini tamamlamalıdırlar. Bu açıdan bakıldığında anahtar planlama algoritmasında, şifreleme algoritmasında kullanılan optimize edilen elemanların tekrar kullanılması bir avantaj olarak kabul edilebilir (May L., Henricksen

M., vd., 2002). Anahtar genişletme algoritmaları ile elde edilen alt anahtarların üzerinde yürütülen iki önemli test, frekans testi ve katı çığ kriteri testidir. Frekans testi, bit karıştırma özelliğinin ölçülmesinde (Shannon'ın karıştırma özelliğinin ölçülmesinde temel teşkil eder) kullanılırken katı çığ kriteri testi, bit yayılım özelliğinin ölçülmesinde kullanılır. Bu test, giriş bloğunda bir bit değişimin çıkış bloğundaki bitlerin yarısının değişimini kontrol eder (Shannon'un yayılım özelliğinin ölçümünü sağlar).

AES-128 blok şifresinin anahtar genişletme algoritması düşünüldüğünde, Şekil 4.4' te genel formu verilmiştir, yukarıda verilen özelliklerden sadece üçüncü özelliği sağladığı belirtilmiştir (May L., Henricksen M., vd., 2002). Bunun yanında AES'in anahtar genişletme algoritmasının kötü yayılım özelliği ilişkili anahtar saldırıları gibi bazı saldırılarda etkin olarak kullanılmaktadır. Bu tür saldırılar gerçek hayatta her ne kadar pratik olmasalar da AES-192 (192-bit anahtar kullanan AES blok şifresi) ve AES-256 (256-bit anahtar kullanan AES blok şifresi) için ilişkili anahtar saldırıların ne kadar faydalı olduğu (Biryukov A., Khovratovich D., 2009), (Biryukov A., Khovratovich D., Nikolic I., 2009) çalışmalarında gösterilmiştir. Bunun temel nedeni olarak AES-192 ve AES-256 versiyonlarındaki anahtar planlama algoritmasının AES-128 (128-bit anahtar kullanan AES blok şifresi) versiyonuna göre daha yavaş yayılım özelliği sağlaması olarak verilebilir. Ayrıca zaman karmaşıklığı açısından Biryukov vd. (Biryukov A., Dunkelman O., vd., 2009) 10 döngüye kadar bir AES algoritmasına pratik bir saldırıyı göstermişlerdir.



Şekil 4.4. AES-128 Blok Şifresinin Anahtar Genişletme Algoritmasının Genel Formu (Rimoldi A. , 2009)

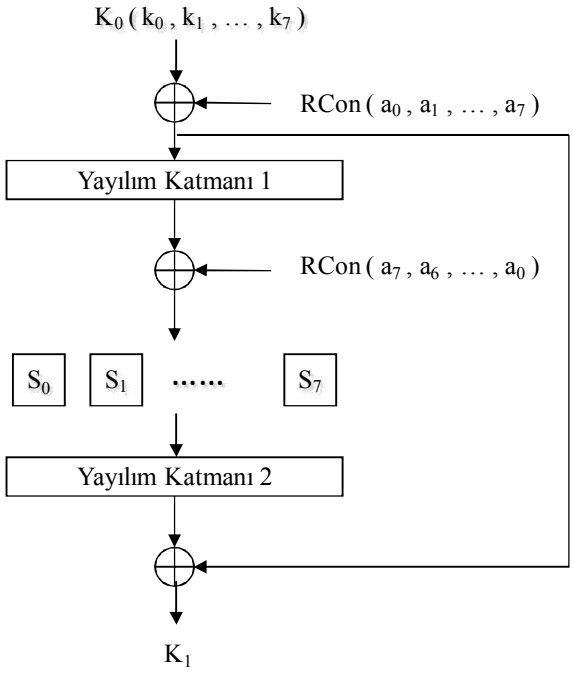
Diğer yandan AES'in anahtar genişletme algoritmasında bit sızıntısı (bit leakage) problemi bulunmaktadır. Bu problem kullanılarak çeşitli saldırılarda bir alt anahtardan faydalanarak diğer alt anahtardan parçalar elde edilebilmektedir. Örneğin (Phan R. C.-W., 2004) çalışmasında bu sızıntı problemini kullanmıştır. Bu problemin önüne geçmek için alt anahtarların birbirinden bağımsız olarak üretilmesi bir yöntem olarak kullanılabilir.

4.6. Geliştirilen Şifrede Kullanılan Anahtar Planlama Evresi

Blok şifre tasarımında kullanılacak üçüncü bileşen anahtar planlama safhasıdır. Bu bileşen için literatürdeki şifreler incelendiğinde bu evre için farklı tasarım stratejilerinin olduğu gözlenmiştir. Bu tasarım stratejilerinden ilki, anahtar planlama evresinin tasarımını Feistel mimarisi ile gerçekleştirmek, ikincisi ise SPN mimarisi ile gerçekleştirmektir. Geliştirilen şifrede tercih edilen anahtar planlama evresi SPN tabanlı bir mimaridir ve AES şifresinde kullanılan anahtar planlama evresinden daha etkin bir planlama evresinin geliştirilmesi hedeflenmiştir.

Bir anahtar genişletme algoritması daha önce de belirtildiği gibi bazı önemli özelliklere sahip olması gerekir. Özellikle AES blok şifresinin anahtar genişletme algoritması katı çığ kriteri testi açısından kötü özellikler göstermektedir. Buna ek olarak bu algoritmanın herhangi bir saldırıda kullanılabilecek bit sızdırma özelliği de bulunmaktadır. Bu gibi problemleri yok etme amacıyla geliştirilen şifrede kullanılacak anahtar planlama evresinde alt anahtarlar birbirlerinden bağımsız şekilde üretilmektedir. Ayrıca yüksek yayılım özelliği gösterecek ve herhangi bir blok şifreye adapte edilebilecek şekilde tasarımı yapılmaktadır. Bu mimari kısaca yayılım-yer değiştirme-yayılım (*Diffusion-Substitution-Diffusion*) şeklinde isimlendirilebilir. Geliştirilen şifrede kullanılan anahtar genişletme mimarisinin en önemli özelliği alt anahtarların elde edilmesi birbirinden bağımsız olması ve yazılım uygulamasının tamamen XOR

işlemleri ve tablo okuma işlemleri ile gerçekleştirilebilmesidir. Şekil 4.5' de de bu mimari 64-bit gizli anahtar için bir alt anahtarın elde edilmesi için gösterilmektedir.



Şekil 4.5. Kullanılan Anahtar Planlama Evresinin Blok Diyagramı

Geliştirilen blok şifre için aynı mimari ve farklı döngü sabitleri kullanılarak döngü sayısı kadar farklı alt anahtarlar üretilebilmektedir. Yine Şekil 4.5' de gösterilen yayılım katmanı 1 ve 2, yüksek yayılım özelliği gösteren doğrusal dönüşümlerdir. Yayılım katmanı 2, Bölüm 4.4' de verilen ve geliştirilen blok şifrenin döngü fonksiyonunda kullanılan 8×8 boyutunda involutif byte matristir. Bu dönüşüm daha önce de belirtildiği gibi bir MDS matris olduğundan dallanma sayısı 9'dur ve $(A - I)$ matrisinin rank değeri 4 olduğundan 2^{32} tane sabit nokta içermektedir. Diğer yandan yayılım katmanı 1 ise 8×8 boyutunda ikili bir dönüşümüdür. Bu dönüşüm aşağıda verilmiştir:

$$YK_1 = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

Bu dönüşüm bir MDBL koddur. Diğer bir deyişle dallanma sayısı 5 olan ikili bir doğrusal dönüşümdür. Bu ikili matrisin $(A-I)$ matrisinin rank değeri 8 olduğundan bu ikili dönüşüm 1 tane sabit nokta içermektedir. Giriş değerleri $GF(2^8)$'in elemanları ya da byte değerler seçildiğinden 64-bit'ten 64-bit'e dönüşüm gerçekleştirmektedir.

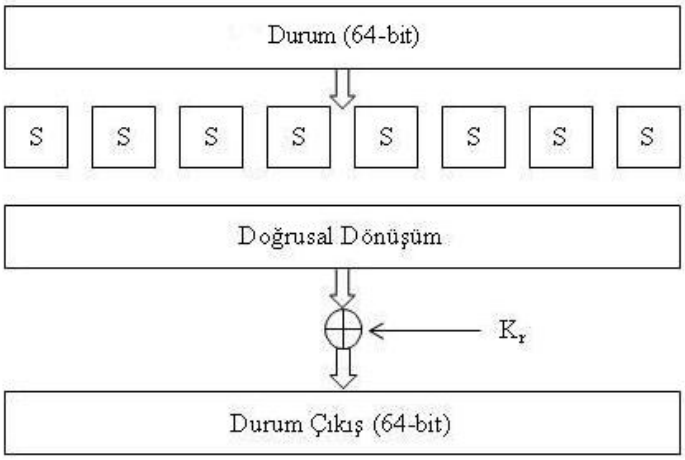
Yayılım katmanı 1 (YK_1) (Aslan B.,2010) çalışmasında gösterildiği gibi cebirsel olarak tasarlanmıştır ve x^4+x+1 ilkel polinomu ile tanımlı $GF(2^4)$'te $\begin{bmatrix} 7_h & C_h \\ C_h & 7_h \end{bmatrix}_{2\times 2}$ matrisinin ikili forma dönüşümünden sonra her satırının 1 birim dairesel olarak sağa döndürülmesi ile elde edilmiştir.

Anahtar planlamada kullanılan yer değiştirme katmanı (S-kutusu) Bölüm 4.2 de verilen ve şifrenin döngü fonksiyonunda kullanılan 8-bit giriş ve 8-bit çıkışa sahip S-kutusudur. Bir alt anahtarı elde ederken döngü sabiti (RCon) bu mimaride $RCon(a_0, a_1, ..., a_7)$ ve $RCon(a_7, a_6, ..., a_0)$ şeklinde 2 defa kullanılmaktadır. Bunun nedeni olarak yayılım katmanı 1'in sahip olduğu 1 tane sabit noktayı da yok etmek olarak gösterilebilir.

5. BLOK ŞİFRE TASARIMININ GERÇEKLEŞTİRİLMESİ VE BLOK ŞİFRENİN ÇALIŞMA ÖRNEĞİ

Bu bölümde Bölüm 4'te verilen kriptografik yapılar kullanılarak blok şifrenin tasarımı gerçekleştirilerek bir uygulama örneği verilecektir. Geliştirilen şifre 64-bit anahtarla çalışan 64-bit bir blok şifredir. SPN tabanlıdır ve Bölüm 4'te gösterildiği gibi yayılım katmanında 8×8 boyutunda $GF(2^8)$ üzerine involutif bir MDS matris kullanmaktadır ve $x^8 + x^4 + x^3 + x + 1$ indirgenemez polinom tabanlı olarak çarpma işlemini gerçekleştirmektedir.

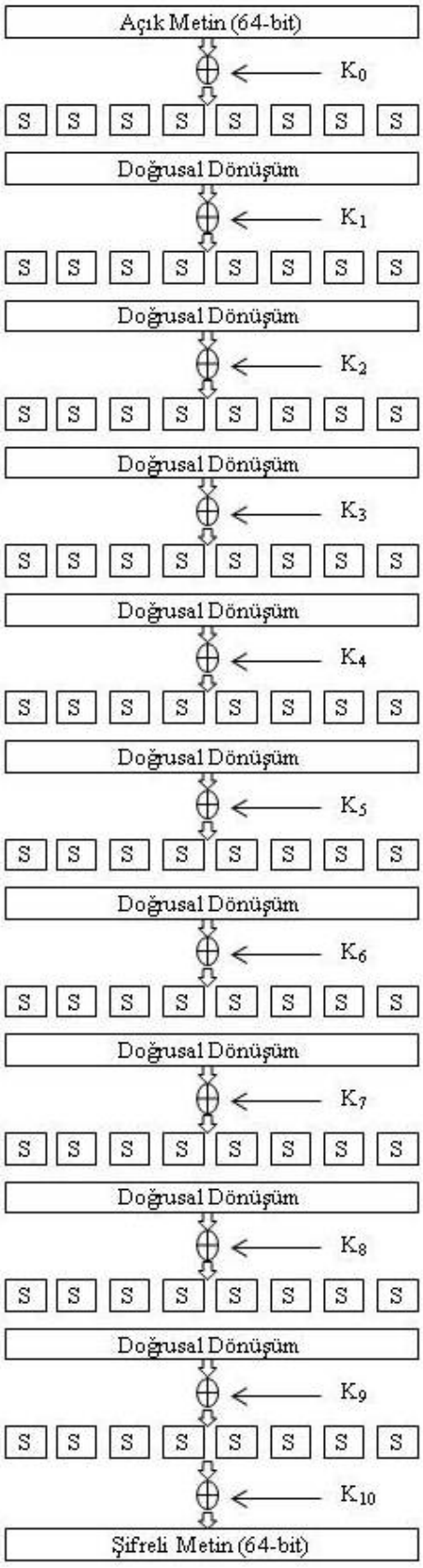
Yer değiştirme katmanı, $GF(2^8)$ üzerinde 8×8 boyutunda bir S-kutusu içerir ve tasarımı $GF(2^8)$ cisminde ters haritalama tabanlıdır. Döngü sayısı 10 olarak seçilmiştir. Her döngü yer değiştirme (substitution), yayılım (diffusion) ve anahtar ekleme adımlarından oluşmaktadır. Son döngüde doğrusal dönüşüm işlemi deşifreleme işleminde şifreleme genel diyagramının aynısının kullanılması amacıyla göz ardı edilmiştir. Şekil 5.1 geliştirilen şifrenin tek döngü için şifreleme adımlarını göstermektedir.



Şekil 5.1 Geliştirilen 64-bit Blok Şifrenin Tek Döngüsü

Anahtar planlama evresi *yayılım-yer değiştirme-yayılım* mimarisine uygun bir şekilde tasarlanmıştır (Sakallı B. F., 2011). Anahtar planlama evresinde geliştirilen blok şifrede kullanılan yapılara ek olarak XOR işlem tabanlı ve 8×8 boyutunda giriş elemanları $GF(2^8)$ cisminden olan ikili bir matris (dallanma sayısı değeri 5 ve sabit nokta sayısı 1) kullanılmaktadır. Döngülerde kullanılacak anahtarlar birbirinde bağımsız şekilde üretilmektedir. Böylelikle AES şifresinin anahtar planlama evresindeki iki önemli problem olan bit sızıntı ve yavaş yayılım özellikleri geliştirilen anahtar planlama evresinde yok edilmeye çalışılmıştır.

Geliştirilen blok şifrenin deşifreleme algoritması Şekil 5.2 de verilen şifreleme algoritması ile aynı yapıyı kullanmaktadır. Sadece S-kutusunun tersi deşifreleme algoritmasında kullanılmaktadır. Şekil 5.3'te S-kutusunun tersi verilmektedir. Diğer yandan anahtar planlamasından gelen anahtarlar ters sırada kullanılmaktadır ve ilk uygulanan gizli anahtar ve son alt anahtar hariç diğer alt anahtarlar blok şifrede kullanılan doğrusal dönüşüme sokularak deşifreleme anahtarları elde edilir.



Şekil 5.2. Geliştirilen 64-bit Blok Şifrenin Genel Diyagramı

Tablo 5.1. Geliştirilen 64-bit Blok Şifrede Kullanılan S-kutusunun Tersi

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	cd	65	c5	97	db	6c	87	6e	34	cb	30	54	8b	f6	fa	43
1	19	9d	d8	50	38	04	99	3c	01	3f	10	11	4a	f9	45	fe
2	7a	18	8c	b6	5c	fd	6b	02	44	a4	3a	08	a6	f5	4b	c9
3	d3	c2	5d	aa	05	f8	6a	ed	c1	ce	86	c8	c4	53	4d	dc
4	1b	98	9a	c7	8a	a8	e3	ba	21	31	cf	ff	79	71	37	3b
5	72	76	7e	a0	f4	9f	83	23	a2	61	7c	a1	ea	e4	8f	e7
6	56	4e	85	66	52	ef	20	be	2c	9c	15	0a	35	5b	a7	d1
7	fb	df	72	89	48	6f	d2	0b	26	5f	59	bd	d0	d6	e5	73
8	03	e0	bb	7d	7f	1c	94	9e	57	40	24	92	2d	5a	f1	ad
9	e1	84	ac	dd	47	32	b7	51	2b	2e	da	f3	62	1a	4f	1f
a	ca	9b	d5	c0	d4	b3	fc	3e	a9	81	70	64	33	b8	82	63
b	42	17	d9	a5	27	96	75	b0	8e	95	90	2f	c3	49	78	f7
c	b9	22	5e	00	88	67	91	6d	36	60	0d	a3	b1	3d	29	14
d	1d	16	c2	e8	0e	ec	13	bf	69	2a	ae	e2	8d	93	39	f2
e	eb	46	b5	12	e9	68	41	80	25	28	ee	b4	e6	7b	55	ab
f	0c	de	bc	74	58	77	4c	07	f0	1e	d7	c6	cc	06	09	af

Örnek 5.1'de hexadecimal notasyonda verilen bir gizli anahtardan şifreleme ve deşifreleme işleminde kullanılacak 10 alt anahtarın elde edilmesi gösterilmektedir.

Örnek 5.1. Aşağıda verilen şifre anahtarını kullanarak Bölüm 4'te verilen anahtar planlama evresi ile 10 adet alt anahtar üretelim:

Şifre Anahtarı: `73 65 6C 6D 61 62 75 6C`

Şifre Anahtarı	`73 65 6C 6D 61 62 75 6C`
$RCon_1$ (a_0, a_1,…,a_7)	`9C C3 5F 04 6A 78 F5 79`
$RCon_1$ (a_0, a_1,…,a_7) Çıkışı	`EF A6 33 69 0B 1A 80 15`
Yayılım 1 Çıkışı	`51 93 F8 24 BA B7 AF E4`
$RCon_1$ (a_7, a_6,…,a_0)	`79 F5 78 6A 04 5F C3 9C`
$RCon_1$ (a_7, a_6,…,a_0) Çıkışı	`28 66 80 4E BE E8 6C 78`

S-kutusu Çıkışı	E9 63 E7 61 67 D3 05 BE
Yayılım 2 Çıkışı	39 A7 EF 78 77 C6 D3 68
1. Alt Anahtar	D6 01 DC 11 7C DC 53 7D
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_2$ (a_0, a_1,…,a_7)	57 EF 42 69 CA BD DE FC
$RCon_2$ (a_0, a_1,…,a_7) Çıkışı	24 8A 2E 04 AB DF AB 90
Yayılım 1 Çıkışı	1E 2F 44 D1 41 61 4E 10
$RCon_2$ (a_7, a_6,…,a_0)	FC DE BD CA 69 42 EF 57
$RCon_2$ (a_7, a_6,…,a_0) Çıkışı	E2 F1 F9 1B 28 23 A1 47
S-kutusu Çıkışı	DB 8E 1D 40 E9 57 5B 94
Yayılım 2 Çıkışı	ED 0E FD 9D 5A C3 29 4A
2. Alt Anahtar	C9 84 D3 99 F1 1C 82 DA
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_3$ (a_0, a_1,…,a_7)	68 C9 F1 04 BD EF CC 69
$RCon_3$ (a_0, a_1,…,a_7) Çıkışı	1B AC 9D 69 DC 8D B9 05
Yayılım 1 Çıkışı	A5 FA 0C 07 67 70 EF 4A
$RCon_3$ (a_7, a_6,…,a_0)	69 CC EF BD 04 F1 C9 68
$RCon_3$ (a_7, a_6,…,a_0) Çıkışı	CC 36 E3 BA 63 81 26 22
S-kutusu Çıkışı	FC C8 46 47 AF A9 78 C1
Yayılım 2 Çıkışı	C1 B6 37 F0 A5 47 0D D5
3. Alt Anahtar	DA 1A AA 99 79 CA B4 D0
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_4$ (a_0, a_1,…,a_7)	F5 48 30 44 69 78 C9 B5
$RCon_4$ (a_0, a_1,…,a_7) Çıkışı	86 2D 5C 29 08 1A BC D9
Yayılım 1 Çıkışı	69 62 93 5D 30 2B FD 9A
$RCon_4$ (a_7, a_6,…,a_0)	B5 C9 78 69 44 30 48 F5
$RCon_4$ (a_7, a_6,…,a_0) Çıkışı	DC AB EB 34 74 1B B5 6F
S-kutusu Çıkışı	3F EF E0 08 F3 40 E2 75
Yayılım 2 Çıkışı	9E 30 33 81 F3 8A 48 31
4. Alt Anahtar	18 1D 6F A8 FB 90 F4 E8
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C

$RCon_5$ (a_0, a_1,…,a_7)	DF 57 02 49 AE FC D5 12
$RCon_5$ (a_0, a_1,…,a_7) Çıkışı	AC 32 6E 24 CF 9E A0 7E
Yayılım 1 Çıkışı	07 74 57 17 06 E1 FA E1
$RCon_5$ (a_7, a_6,…,a_0)	12 D5 FC AE 49 02 57 DF
$RCon_5$ (a_7, a_6,…,a_0) Çıkışı	15 A1 AB B9 4F E3 AD 3E
S-kutusu Çıkışı	6A 5B EF C0 9E 46 8F A7
Yayılım 2 Çıkışı	E9 31 A3 4C F9 45 D1 B4
5. Alt Anahtar	45 03 CD 68 36 DB 71 CA
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_6$ (a_0, a_1,…,a_7)	9A F5 66 7E D3 40 E6 9F
$RCon_6$ (a_0, a_1,…,a_7) Çıkışı	E9 90 0A 13 B2 22 93 F3
Yayılım 1 Çıkışı	51 F3 EA F3 21 FA 28 A1
$RCon_6$ (a_7, a_6,…,a_0)	9F E6 40 D3 7E 66 F5 9A
$RCon_6$ (a_7, a_6,…,a_0) Çıkışı	CE 15 AA 20 5F 9C DD 3B
S-kutusu Çıkışı	39 6A 33 66 79 69 93 4F
Yayılım 2 Çıkışı	D3 E7 D1 A6 A9 94 CE 7A
6. Alt Anahtar	3A 77 DB B5 1B B6 5D 89
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_7$ (a_0, a_1,…,a_7)	0F 88 2D CE 65 24 77 8A
$RCon_7$ (a_0, a_1,…,a_7) Çıkışı	7C ED 41 A3 04 46 02 E6
Yayılım 1 Çıkışı	AE 71 AB DE 3E E7 90 30
$RCon_7$ (a_7, a_6,…,a_0)	8A 77 24 65 CE 2D 88 0F
$RCon_7$ (a_7, a_6,…,a_0) Çıkışı	24 06 8F BB F0 CA 18 3F
S-kutusu Çıkışı	8A FD 5E 82 F8 A0 21 19
Yayılım 2 Çıkışı	BF 64 1D 2B FD 4C DD 4A
7. Alt Anahtar	C3 89 5C 88 F9 0A DF AC
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_9$ (a_0, a_1,…,a_7)	3A B0 E2 5D EF D5 B1 28
$RCon_9$ (a_0, a_1,…,a_7) Çıkışı	49 D5 8E 30 8E B7 C4 44
Yayılım 1 Çıkışı	EB E6 16 0A 04 37 9B 1C
$RCon_9$ (a_7, a_6,…,a_0)	28 B1 D5 EF 5D E2 B0 3A

$RCon_9$ (a_7, a_6,…,a_0) Çıkışı	C3 57 C3 E5 59 D5 2B 26
S-kutusu Çıkışı	BC 88 BC 7E 7A A2 98 78
Yayılım 2 Çıkışı	F4 52 F0 E9 50 6C 1B 56
8. Alt Anahtar	BD 87 7E D9 DE DB DF 12
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_8$ (a_0, a_1,…,a_7)	DB E0 90 8C 4E 50 77 49
$RCon_8$ (a_0, a_1,…,a_7) Çıkışı	A8 85 FC E1 2F 32 02 25
Yayılım 1 Çıkışı	6C 32 A0 AA A2 C6 D9 D9
$RCon_8$ (a_7, a_6,…,a_0)	49 77 50 4E 8C 90 E0 DB
$RCon_8$ (a_7, a_6,…,a_0) Çıkışı	25 45 F0 E4 2E 56 39 02
S-kutusu Çıkışı	E8 1E F8 5D 99 60 DE 27
Yayılım 2 Çıkışı	F9 4D E1 F3 2B 9C 6C 2E
9. Alt Anahtar	51 C8 1D 12 04 AE 6E 0B
Şifre Anahtarı	73 65 6C 6D 61 62 75 6C
$RCon_{10}$ (a_0, a_1,…,a_7)	18 7A 8F 49 36 E3 88 67
$RCon_{10}$ (a_0, a_1,…,a_7) Çıkışı	6B 1F E3 24 57 81 FD 0B
Yayılım 1 Çıkışı	9A 4E 05 87 26 C3 27 36
$RCon_{10}$ (a_7, a_6,…,a_0)	79 F5 78 6A 04 5F C3 9C
$RCon_{10}$ (a_7, a_6,…,a_0) Çıkışı	FD C6 E6 B1 6F 4C 5D 2E
S-kutusu Çıkışı	25 FB EC CC 75 F6 32 99
Yayılım 2 Çıkışı	2A 46 A3 50 E4 2B BE 38
10. Alt Anahtar	41 59 40 74 B3 AA 43 33

Sonuç olarak gizli anahtar K_0 olmak üzere geliştirilen blok şifrenin çalışması sırasında kullanılacak K_1 den K_{10}'a kadar diğer alt anahtarlar ile birlikte Şekil 5.2 de verilen diyagrama uygun deşifreleme anahtarları Tablo 5.2 de verilmiştir.

Tablo 5.2 Örnek 5.1 de Verilen Gizli Anahtardan Elde Edilen Şifreleme ve Deşifreleme İşlemlerinde Kullanılacak Alt Anahtarlar

Şifreleme Anahtarları	Deşifreleme Anahtarları
EK_0 = 73 65 6C 6D 61 62 75 6C	DK_0 = 41 59 40 74 B3 AA 43 33
EK_1 = D6 01 DC 11 7C DC 53 7D	DK_1 = 0F E3 43 D2 8D 38 AB 3A
EK_2 = C9 84 D3 99 F1 1C 82 DA	DK_2 = 79 03 93 8B 4A 21 56 0A
EK_3 = DA 1A AA 99 79 CA B4 D0	DK_3 = 91 D4 ED 14 E2 B4 EB 1F
EK_4 = 18 1D 6F A8 FB 90 F4 E8	DK_4 = E1 5C 22 52 A3 85 95 24
EK_5 = 45 03 CD 68 36 DB 71 CA	DK_5 = F8 D2 83 8E 54 2C 4E A4
EK_6 = 3A 77 DB B5 1B B6 5D 89	DK_6 = AC 65 BB 74 94 CA 55 B8
EK_7 = C3 89 5C 88 F9 0A DF AC	DK_7 = AF 2F 8C 93 B2 86 5F D0
EK_8 = BD 87 7E D9 DE DB DF 12	DK_8 = 20 DD AE 99 98 11 E7 16
EK_9 = 51 C8 1D 12 04 AE 6E 0B	DK_9 = 52 13 79 85 EC D7 1D 0F
EK_{10}=41 59 40 74 B3 AA 43 33	DK_{10}=73 65 6C 6D 61 62 75 6C

Tablo 5.2 de verilen deşifreleme anahtarları $DK_0 = EK_{10}$, $DK_1 = DD(EK_9)$, $DK_2 = DD(EK_8)$,…, $DK_9 = DD(EK_1)$ ve $DK_{10} = EK_0$ şeklinde elde edilmektedir. Denklemlerde verilen DD blok şifrenin döngü fonksiyonunda kullanılan ve Bölüm 4.4'te verilen 8×8 boyutunda involutif MDS matrisi temsil etmektedir.

Örnek 5.2. Aşağıda verilen açık metni ve Örnek 5.1 de verilen şifre anahtarını kullanarak 10 döngü için şifreleme adımlarını göstererek şifreli metni elde edelim. Tablo 5.2 de verilen deşifreleme anahtarlarını ve aynı şifreleme diyagramını kullanarak deşifreleme adımlarını 10 döngü için göstererek açık metni elde edelim.

Açık Metin: 74 6F 6C 67 61 73 61 6B

Şifre Anahtarı: 73 65 6C 6D 61 62 75 6C

Şifreleme Adımları:

Açık Metin	74 6F 6C 67 61 73 61 6B
Gizli Anahtar	73 65 6C 6D 61 62 75 6C
1. Döngü Giriş	07 0A 00 0A 00 11 14 07
1. Döngü S-kutusu Çıkışı	F7 6B C3 6B C3 1B CF F7
1 . Döngü Doğrusal Dönüşüm Çıkışı	23 70 0B 0B 9D 6A C4 B4
1. Alt anahtar	D6 01 DC 11 7C DC 53 7D
2. Döngü Giriş	F5 71 D7 1A E1 B6 97 C9
2. Döngü S-kutusu Çıkışı	2D 4D FA 9D 90 23 03 2F
2 . Döngü Doğrusal Dönüşüm Çıkışı	37 C5 E0 A6 97 FF 83 C7
2. Alt anahtar	C9 84 D3 99 F1 1C 82 DA
3. Döngü Giriş	FE 41 33 3F 66 E3 01 1D
3. Döngü S-kutusu Çıkışı	1F E6 AC 19 63 46 18 D0
3 . Döngü Doğrusal Dönüşüm Çıkışı	C5 7E 36 39 0C 5E 11 56
3. Alt anahtar	DA 1A AA 99 79 CA B4 D0
4. Döngü Giriş	1F 64 9C A0 75 94 A5 86
4. Döngü S-kutusu Çıkışı	9F AB 69 53 B6 86 B3 3A
4 . Döngü Doğrusal Dönüşüm Çıkışı	FD CF 4C B2 08 D9 F0 5A
4. Alt anahtar	18 1D 6F A8 FB 90 F4 E8
5. Döngü Giriş	E5 D2 23 1A F3 49 04 B2
5. Döngü S-kutusu Çıkışı	7E 76 57 9D 9B BD 15 31
5 . Döngü Doğrusal Dönüşüm Çıkışı	DE 7A A7 9A 40 E6 5F A0
5. Alt anahtar	45 03 CD 68 36 DB 71 CA
6. Döngü Giriş	9B 79 6A F2 76 3D 2E 6A
6. Döngü S-kutusu Çıkışı	A1 4C 36 DF 51 CD 99 36
6 . Döngü Doğrusal Dönüşüm Çıkışı	BA E9 9B 95 1C 18 AF C1
6. Alt anahtar	3A 77 DB B5 1B B6 5D 89
7. Döngü Giriş	80 9E 40 20 07 AE F2 48
7. Döngü S-kutusu Çıkışı	E7 87 89 66 F7 DA DF 74
7 . Döngü Doğrusal Dönüşüm Çıkışı	3B 66 E2 2B 3B 5C FE 04
7. Alt anahtar	C3 89 5C 88 F9 0A DF AC
8. Döngü Giriş	F8 EF BE A3 C2 56 21 A8

8. Döngü S-kutusu Çıkışı	35 65 67 CB D2 60 48 45
8 .Döngü Doğrusal Dönüşüm Çıkışı	B7 06 DD 55 4F E0 45 90
8. Alt anahtar	BD 87 7E D9 DE DB DF 12
9. Döngü Giriş	0A 81 A3 8C 91 3B 9A 82
9. Döngü S-kutusu Çıkışı	6B A9 CB 22 C6 4F 42 AE
9 . Döngü Doğrusal Dönüşüm Çıkışı	5E 2A E2 6F 1E 96 5C 63
9. Alt anahtar	51 C8 1D 12 04 AE 6E 0B
10. Döngü Giriş	0F E2 FF 7D 1A 38 32 68
10. Döngü S-kutusu Çıkışı	72 DB 4B 83 9D 14 95 E5
10. Alt anahtar	41 59 40 74 B3 AA 43 33
Şifreli Metin	33 82 0B F7 2E BE D6 D6

Deşifreleme Adımları:

Şifreli Metin	33 82 0B F7 2E BE D6 D6
1. Alt Anahtar	41 59 40 74 B3 AA 43 33
1.Döngü Giriş	72 DB 4B 83 9D 14 95 E5
1. Döngü Ters S-kutusu Çıkışı	0F E2 FF 7D 1A 38 32 68
1. Döngü Doğrusal Dönüşüm Çıkışı	64 4A 88 F0 4B 77 E9 94
2. Alt Anahtar	0F E3 43 D2 8D 38 AB 3A
2.Döngü Giriş	6B A9 CB 22 C6 4F 42 AE
2. Döngü Ters S-kutusu Çıkışı	0A 81 A3 8C 91 3B 9A 82
2. Döngü Doğrusal Dönüşüm Çıkışı	4C 66 F4 40 98 41 1E 4F
3. Alt Anahtar	79 03 93 8B 4A 21 56 0A
3.Döngü Giriş	35 65 67 CB D2 60 48 45
3. Döngü Ters S-kutusu Çıkışı	F8 EF BE A3 C2 56 21 A8
3. Döngü Doğrusal Dönüşüm Çıkışı	76 53 64 72 15 6E 34 6B
4. Alt Anahtar	91 D4 ED 14 E2 B4 EB 1F
4.Döngü Giriş	E7 87 89 66 F7 DA DF 74
4. Döngü Ters S-kutusu Çıkışı	80 9E 40 20 07 AE F2 48
4. Döngü Doğrusal Dönüşüm Çıkışı	40 10 14 8D F2 48 0C 12
5. Alt Anahtar	E1 5C 22 52 A3 85 95 24

5.Döngü Giriş	A1 4C 36 DF 51 CD 99 36
5. Döngü Ters S-kutusu Çıkışı	9B 79 6A F2 76 3D 2E 6A
5. Döngü Doğrusal Dönüşüm Çıkışı	86 A4 D4 13 CF 91 5B 95
6. Alt Anahtar	F8 D2 83 8E 54 2C 4E A4
6.Döngü Giriş	7E 76 57 9D 9B BD 15 31
6. Döngü Ters S-kutusu Çıkışı	E5 D2 23 1A F3 49 04 B2
6. Döngü Doğrusal Dönüşüm Çıkışı	33 CE D2 27 22 4C E6 82
7. Alt Anahtar	AC 65 BB 74 94 CA 55 B8
7.Döngü Giriş	9F AB 69 53 B6 86 B3 3A
7. Döngü Ters S-kutusu Çıkışı	1F 64 9C A0 75 94 A5 86
7. Döngü Doğrusal Dönüşüm Çıkışı	B0 C9 20 8A D1 C0 47 00
8. Alt Anahtar	AF 2F 8C 93 B2 86 5F D0
8.Döngü Giriş	1F E6 AC 19 63 46 18 D0
8. Döngü Ters S-kutusu Çıkışı	FE 41 33 3F 66 E3 01 1D
8. Döngü Doğrusal Dönüşüm Çıkışı	0D 90 54 04 08 32 E4 39
9.Alt Anahtar	20 DD AE 99 98 11 E7 16
9.Döngü Giriş	2D 4D FA 9D 90 23 03 2F
9. Döngü Ters S-kutusu Çıkışı	F5 71 D7 1A E1 B6 97 C9
9. Döngü Doğrusal Dönüşüm Çıkışı	A5 78 BA EE 2F CC D2 F8
10.Alt Anahtar	52 13 79 85 EC D7 1D 0F
10 .Döngü Giriş	F7 6B C3 6B C3 1B CF F7
10.Döngü Ters S-kutusu Çıkışı	07 0A 00 0A 00 11 14 07
Gizli Anahtar	73 65 6C 6D 61 62 75 6C
Açık Metin	74 6F 6C 67 61 73 61 6B

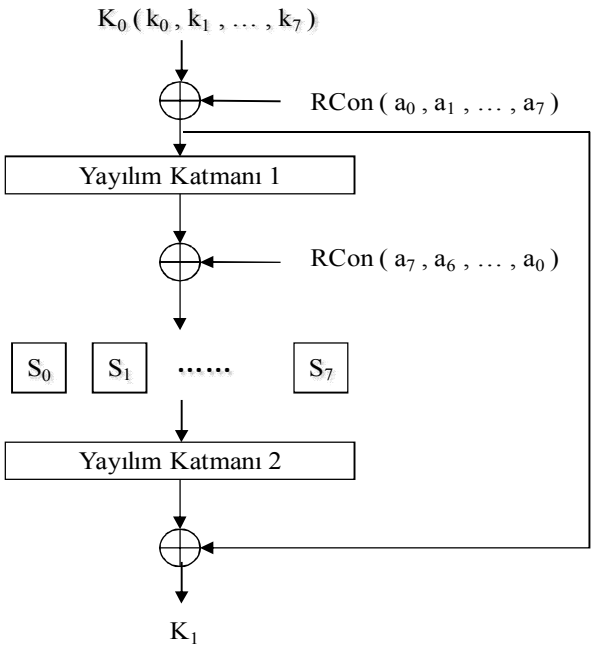
5.1 Geliştirilen Blok Şifrenin Yazılım Maliyeti

Şifrenin yazılım uygulaması geliştirilirken iki önemli byte işleminden faydalanılmıştır. Bunlar sırasıyla XOR işlemi ve tablodan okuma işlemidir. Geliştirilen blok şifrenin Şekil 5.1 de verilen tek döngüsü için S-kutusu dönüşüm safhasının yazılım uygulaması için 8 byte değerinin tablodan okunma yeterlidir. Diğer yandan aşağıdan verilen doğrusal dönüşüm safhasının yazılım uygulamasının gerçekleştirilmesi için sonlu cisimde çarpmalar için 7×256 boyutunda bir tablodan okuma ve XOR işlemleri yeterli olacaktır. Dolayısıyla 56 byte tablodan okuma ve 56 byte XOR işlemi bu doğrusal dönüşümün yazılım uygulaması maliyeti olarak karşımıza çıkar. Döngüdeki son adım olan anahtar ekleme işlemi de 8 byte XOR işlemi gerektireceğinden toplamda 64 byte tablodan okuma ve 64 byte XOR işlemi şifrenin bir döngü için yazılım uygulamasında karşımıza çıkacak byte işlem sayısını verir.

$$A_1 = \begin{bmatrix} 01_h & 02_h & 05_h & 04_h & 06_h & 0B_h & 09_h & 07_h \\ 02_h & 01_h & 04_h & 05_h & 0B_h & 06_h & 07_h & 09_h \\ 05_h & 04_h & 01_h & 02_h & 09_h & 07_h & 06_h & 0B_h \\ 04_h & 05_h & 02_h & 01_h & 07_h & 09_h & 0B_h & 06_h \\ 06_h & 0B_h & 09_h & 07_h & 01_h & 02_h & 05_h & 04_h \\ 0B_h & 06_h & 07_h & 09_h & 02_h & 01_h & 04_h & 05_h \\ 09_h & 07_h & 06_h & 0B_h & 05_h & 04_h & 01_h & 02_h \\ 07_h & 09_h & 0B_h & 06_h & 04_h & 05_h & 02_h & 01_h \end{bmatrix}.$$

Sonuç olarak Şekil 5.2 de genel diyagramı verilen ve 64-bit blokları 64-bit gizli anahtarla şifreleyen blok şifrenin yazılım uygulaması için son döngüde doğrusal dönüşümünde göz ardı edildiğini de hesaba katarak 584 byte tablodan okuma ve 592 byte XOR işlemi yeterli olacaktır. Buna ek olarak anahtar genişletme evresinde bir gizli anahtardan blok şifrenin döngülerinde kullanılması için 10 farklı anahtar elde edilmektedir. Şekil 5.3'te gizli anahtardan tek alt anahtarın elde edilmesi için kullanılan anahtar genişletme algoritması gösterilmektedir. Tek alt anahtarın elde edilmesinde yayılım katmanı 1'in yazılım uygulaması için 36 byte XOR ve daha önceden de verildiği gibi yayılım katmanı 2'nin yazılım uygulaması için 56 byte XOR ve 56 byte

tablodan okuma işlemi gerekmektedir. Dolayısıyla tek alt anahtarın elde edilmesi için toplamda 108 byte XOR ve 64 byte tablodan okuma işlemi, 10 alt anahtarın yazılım uygulamasında üretilmesi için ise toplamda 1080 byte XOR ve 640 byte tablodan okuma işlemi gerekmektedir. Deşifreleme işlemi yazılım uygulamasında şifreleme işlemi ile aynı sayıda byte işlemi gerektirmektedir. Sadece şifrelemede kullanılan S-kutusunun yerine deşifreleme işleminde ters S-kutusu kullanılmaktadır.



Şekil 5.3. Geliştirilen Blok Şifrede Kullanılan Anahtar Genişletme Algoritması

5.2 Geliştirilen Blok Şifrenin Kriptografik Saldırılara Karşı Dayanıklılığı

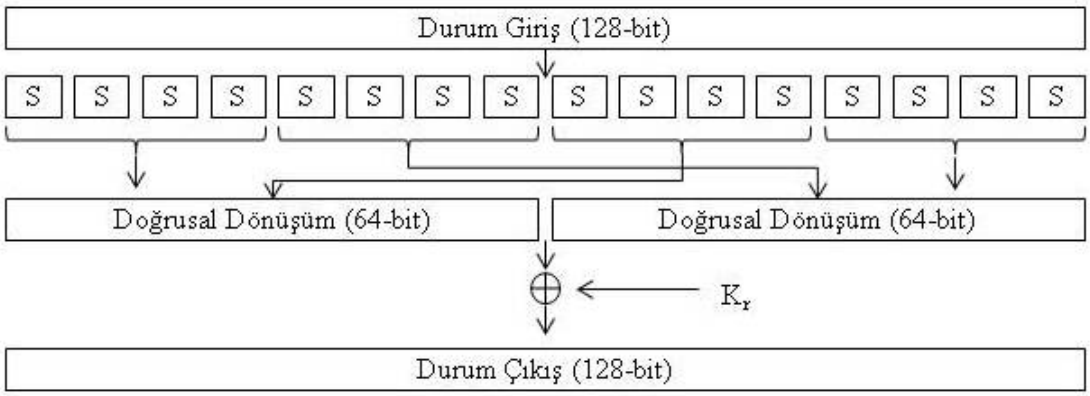
Geliştirilen şifrede kullanılan S-kutusu ve doğrusal dönüşüm yapılarının özellikleri şifreye uygulanabilecek doğrusal saldırıları (Matsui M., 1994) ve diferansiyel saldırıları (Biham E. and Shamir A., 1991) etkisiz kılmaktadır. Diğer yandan diğer saldırılara (kesik diferansiyel kriptanaliz (Knudsen L. R.,1994), imkânsız diferansiyel kriptanaliz (Phan R. C.-W.,2004) vb.) karşı literatürde bulunan ve incelenen blok

şifrelerden Khazad blok şifresi taban alınabilir. Örneğin geliştirilen blok şifre Khazad blok şifresine yapı olarak benzemektedir ancak geliştirilen blok şifrede kullanılan S-kutusunun özellikleri daha iyidir (optimal kriptografik özelliklere sahiptir). Buna ek olarak S-kutusu kriptografik özellikleri AES S-kutusunun özellikleri ile aynı olmakla beraber polinomsal ifadesindeki terim sayısı optimal değer olan 255'tir. Bu özellikte şifrenin interpolasyon saldırıları (Jakobsen T., Knudsen L.,1997) gibi cebirsel saldırılara karşı dayanıklılığını arttırmaktadır. Şifrenin yapısında kullanılan 9 dallanma sayı değerine (involutif MDS matris) sahip doğrusal dönüşüm, geliştirilen blok şifreye yapılacak diğer saldırılara karşı da şifrenin dayanıklılığını arttıracak niteliktedir. Geliştirilen anahtar planlama evresinin özellikleri incelenen iki blok şifrede bulunan bit sızıntı problemi (AES ve ARIA şifresinde bulunmaktadır) ve yavaş yayılım özelliklerini (AES şifresinde bulunmaktadır) yok etme amacındadır. Dolayısıyla bu tür problemlerin kullanıldığı saldırılar olan ilişkili anahtar saldırıları (Z'aba M. R.), imkansız diferansiyel kriptanaliz ve ilişkili anahtar saldırıları ile beraber kullanılabilecek saldırılara karşı blok şifrenin gücünün arttırılması hedeflenmiştir. Sonuç olarak geliştirilen blok şifre anahtar uzayı olan 2^{64} denemeden (kaba kuvvet saldırısı) daha az maliyetle anlamlı mesajı elde etmeyi engelleyici doğrultuda tasarlanmıştır.

5.3 Geliştirilen Blok Şifrenin Farklı Anahtar Büyüklüğü ve Farklı Blok Uzunluğu İçin Tasarımının Genişletilmesi

İncelenen blok şifrelerden Khazad blok şifresi 64-bit blokları 128-bit anahtar seçeneği ile şifrelemektedir. Geliştirilen şifrenin anahtar genişletme evresinde yapılacak küçük düzenlemeler ile 64-bit blokları 128-bit anahtar seçeneği ile çalışan bir versiyonu kolaylıkla tasarlanabilir. Diğer yandan incelenen şifrelerden AES ve ARIA blok şifreleri 128-bit blokları 128-bit, 192-bit ve 256-bit anahtar seçenekleri ile şifrelemektedir. Geliştirilen şifre, işlediği blok büyüklüğü 128-bit ve kullandığı anahtar büyüklüğü (anahtar planlama evresinde yapılacak düzenlemeler ile) sırasıyla 128-bit, 192-bit ve 256-bit olacak şekilde tekrar düzenlenerek genişletilebilir. Şekil 5.4, geliştirilen blok

şifreden geliştirilecek ve 128-bit blokları işleyecek blok şifrenin bir döngüsü için genel diyagramını göstermektedir.



Şekil 5.4. 128-Bit Blok Şifrenin Tek Döngüsünün Genel Diyagramı

6. SONUÇLAR VE DEĞERLENDİRME

Bu çalışma blok şifreler üzerine yapılan incelemelerden elde edilen sonuçlara göre 64-bit girişli 64-bit çıkışlı simetrik bir blok şifre geliştirilmesi üzerinedir. Literatürde bulunan önemli blok şifreleme algoritmalarından AES, ARIA ve Khazad şifreleme algoritmaları incelenmiş ve bu şifrelerden edinilen tecrübe ile AESve Khazad blok şifresi tabanlı modern bir blok şifre geliştirilmiştir.

Özellikle AES şifreleme algoritmasının S-kutusu incelenmiş ve bu S-kutusunun kriptografik özellikleri değiştirilmeden polinomsal ifadesi daha iyi bir S-kutusunun kullanılabileceği anlaşılmıştır.

İncelenen blok şifreleme algoritmalarının doğrusal dönüşümlerine bakıldığında AES'in $4x4$ MDS matris, ARIA'nın $16x16$'lık involutif ikilik matris, KHAZAD'ın $8x8$ involutif MDS matris kullandığı görülmüştür. Geliştirilen şifrede AES'in $4x4$ MDS matrisi yerine $8x8$ involutif bir MDS matris kullanmanın şifreleme algoritmasını güçlendirebileceği ve şifreleme ve deşifreleme arasındaki performans farkını yok edeceği düşünülmektedir.

Yayılım katmanlarında kullanılan dönüşümlerden kriptografik bir kriter olan dallanma sayısı değeri AES blok şifresi için 5, KHAZAD blok şifresi için 9, ARIA blok şifresi içinse 8' dir. Çalışmamızda geliştirdiğimiz şifrenin daha yüksek dallanma sayısı değerine sahip olması amaçlanmıştır. Bu değer de bir MDS matris kullanıldığında 9' dur.

Bir anahtar genişletme algoritması için iki önemli özellik olan yavaş yayılım ve bit sızdırma özellikleri temel alınarak bir blok şifreden bağımsız yeni bir anahtar genişletme mimarisi kullanılmıştır. Bununla beraber bu önerilen yeni tekniğin yazılım uygulaması XOR tabanlı olduğu için performansı da iyi sonuçlar vermektedir.

EK A: Geliştirilen Şifrede Kullanılan S-Kutusunun Cebirsel İfadesi

$S(x) = \text{'}1C\text{'}\ x^{254} + \text{'}1E\text{'}\ x^{253} + \text{'}16\text{'}\ x^{252} + \text{'}98\text{'}\ x^{251} + \text{'}07\text{'}\ x^{250} + \text{'}58\text{'}\ x^{249} + \text{'}86\text{'}\ x^{248} + \text{'}E2\text{'}\ x^{247} + \text{'}B5\text{'}\ x^{246} + \text{'}11\text{'}\ x^{245} + \text{'}06'\ x^{244} + \text{'}8E'\ x^{243} + \text{'}BA'\ x^{242} + \text{'}9E\text{'}\ x^{241} + \text{'}3F\text{'}\ x^{240} + \text{'}A4\text{'}\ x^{239} + \text{'}22\text{'}\ x^{238} + \text{'}3C\text{'}\ x^{237} + \text{'}E4\text{'}\ x^{236} + \text{'}1A\text{'}\ x^{235} + \text{'}9A\text{'}\ x^{234} + \text{'}18\text{'}\ x^{233} + \text{'}DD\text{'}\ x^{232} + \text{'}99\text{'}\ x^{231} + \text{'}82\text{'}\ x^{230} + \text{'}4C\text{'}\ x^{229} + \text{'}98\text{'}\ x^{228} + \text{'}DE\text{'}\ x^{227} + \text{'}25\text{'}\ x^{226} + \text{'}F8\text{'}\ x^{225} + \text{'}75\text{'}\ x^{224} + \text{'}BB\text{'}\ x^{223} + \text{'}81\text{'}\ x^{222} + \text{'}FD\text{'}\ x^{221} + \text{'}D0\text{'}\ x^{220} + \text{'}C9\text{'}\ x^{219} + \text{'}04\text{'}\ x^{218} + \text{'}74\text{'}\ x^{217} + \text{'}F6\text{'}\ x^{216} + \text{'}B2\text{'}\ x^{215} + \text{'}39\text{'}\ x^{214} + \text{'}49\text{'}\ x^{213} + \text{'}0A\text{'}\ x^{212} + \text{'}F9\text{'}\ x^{211} + \text{'}49\text{'}\ x^{210} + \text{'}3B\text{'}\ x^{209} + \text{'}6C\text{'}\ x^{208} + \text{'}A7\text{'}\ x^{207} + \text{'}66\text{'}\ x^{206} + \text{'}E3\text{'}\ x^{205} + \text{'}90\text{'}\ x^{204} + \text{'}42\text{'}\ x^{203} + \text{'}B7\text{'}\ x^{202} + \text{'}5D\text{'}\ x^{201} + \text{'}4F\text{'}\ x^{200} + \text{'}8D\text{'}\ x^{199} + \text{'}DB\text{'}\ x^{198} + \text{'}38\text{'}\ x^{197} + \text{'}9A\text{'}\ x^{196} + \text{'}68\text{'}\ x^{195} + \text{'}E5\text{'}\ x^{194} + \text{'}82\text{'}\ x^{193} + \text{'}50\text{'}\ x^{192} + \text{'}73\text{'}\ x^{191} + \text{'}BD\text{'}\ x^{190} + \text{'}06\text{'}\ x^{189} + \text{'}A7\text{'}\ x^{188} + \text{'}F3\text{'}\ x^{187} + \text{'}1D\text{'}\ x^{186} + \text{'}28\text{'}\ x^{185} + \text{'}46\text{'}\ x^{184} + \text{'}8C\text{'}\ x^{183} + \text{'}04\text{'}\ x^{182} + \text{'}CF\text{'}\ x^{181} + \text{'}8C\text{'}\ x^{180} + \text{'}C8\text{'}\ x^{179} + \text{'}6E\text{'}\ x^{178} + \text{'}59\text{'}\ x^{177} + \text{'}32\text{'}\ x^{176} + \text{'}51\text{'}\ x^{175} + \text{'}DF\text{'}\ x^{174} + \text{'}A8\text{'}\ x^{173} + \text{'}91\text{'}\ x^{172} + \text{'}A5\text{'}\ x^{171} + \text{'}E7\text{'}\ x^{170} + \text{'}63\text{'}\ x^{169} + \text{'}D5\text{'}\ x^{168} + \text{'}A0\text{'}\ x^{167} + \text{'}1B\text{'}\ x^{166} + \text{'}96\text{'}\ x^{165} + \text{'}D3\text{'}\ x^{164} + \text{'}85\text{'}\ x^{163} + \text{'}58\text{'}\ x^{162} + \text{'}AF\text{'}\ x^{161} + \text{'}C9\text{'}\ x^{160} + \text{'}88\text{'}\ x^{159} + \text{'}5E\text{'}\ x^{158} + \text{'}2F\text{'}\ x^{157} + \text{'}A6\text{'}\ x^{156} + \text{'}9A\text{'}\ x^{155} + \text{'}27\text{'}\ x^{154} + \text{'}84\text{'}\ x^{153} + \text{'}59\text{'}\ x^{152} + \text{'}91\text{'}\ x^{151} + \text{'}C0\text{'}\ x^{150} + \text{'}83\text{'}\ x^{149} + \text{'}2B\text{'}\ x^{148} + \text{'}1B\text{'}\ x^{147} + \text{'}BC\text{'}\ x^{146} + \text{'}19\text{'}\ x^{145} + \text{'}30\text{'}\ x^{144} + \text{'}93\text{'}\ x^{143} + \text{'}96\text{'}\ x^{142} + \text{'}52\text{'}\ x^{141} + \text{'}2E\text{'}\ x^{140} + \text{'}11\text{'}\ x^{139} + \text{'}3E\text{'}\ x^{138} + \text{'}28\text{'}\ x^{137} + \text{'}E3\text{'}\ x^{136} + \text{'}E0\text{'}\ x^{135} + \text{'}95\text{'}\ x^{134} + \text{'}2C\text{'}\ x^{133} + \text{'}0F\text{'}\ x^{132} + \text{'}26\text{'}\ x^{131} + \text{'}99\text{'}\ x^{130} + \text{'}FB\text{'}\ x^{129} + \text{'}63\text{'}\ x^{128} + \text{'}7E\text{'}\ x^{127} + \text{'}88\text{'}\ x^{126} + \text{'}14\text{'}\ x^{125} + \text{'}A3\text{'}\ x^{124} + \text{'}DD\text{'}\ x^{123} + \text{'}94\text{'}\ x^{122} + \text{'}20\text{'}\ x^{121} + \text{'}B4\text{'}\ x^{120} + \text{'}70\text{'}\ x^{119} + \text{'}7E\text{'}\ x^{118} + \text{'}B1\text{'}\ x^{117} + \text{'}F6\text{'}\ x^{116} + \text{'}$

$$0D \ ' \ x^{115} + ' \ 92 \ ' \ x^{114} + ' \ 1F \ ' \ x^{113} + ' \ 0B \ ' \ x^{112} + ' \ 62 \ ' \ x^{111} + ' \ 0D \ ' \ x^{110} + ' \ 3E \ ' \ x^{109} + ' \ 16 \ ' \ x^{108} + ' \ D6 \ ' \ x^{107} + ' \ F8 \ ' \ x^{106} + ' \ E7 \ ' \ x^{105} + ' \ 47 \ ' \ x^{104} + ' \ 30 \ ' \ x^{103} + ' \ 42 \ ' \ x^{102} + ' \ CB \ ' \ x^{101} + ' \ 26 \ ' \ x^{100} + ' \ 05 \ ' \ x^{99} + ' \ 3B \ ' \ x^{98} + ' \ 26 \ ' \ x^{97} + ' \ 8C \ ' \ x^{96} + ' \ A8 \ ' \ x^{95} + ' \ 75 \ ' \ x^{94} + ' \ A1 \ ' \ x^{93} + ' \ 09 \ ' \ x^{92} + ' \ D9 \ ' \ x^{91} + ' \ 6A \ ' \ x^{90} + ' \ D1 \ ' \ x^{89} + ' \ 5A \ ' \ x^{88} + ' \ 45 \ ' \ x^{87} + ' \ 29 \ ' \ x^{86} + ' \ D1 \ ' \ x^{85} + ' \ C8 \ ' \ x^{84} + ' \ 5E \ ' \ x^{83} + ' \ 97 \ ' \ x^{82} + ' \ 28 \ ' \ x^{81} + ' \ 79 \ ' \ x^{80} + ' \ 59 \ ' \ x^{79} + ' \ C3 \ ' \ x^{78} + ' \ 48 \ ' \ x^{77} + ' \ 6F \ ' \ x^{76} + ' \ E8 \ ' \ x^{75} + ' \ 79 \ ' \ x^{74} + ' \ 3B \ ' \ x^{73} + ' \ DE \ ' \ x^{72} + ' \ A5 \ ' \ x^{71} + ' \ B5 \ ' \ x^{70} + ' \ EB \ ' \ x^{69} + ' \ 9C \ ' \ x^{68} + ' \ C3 \ ' \ x^{67} + ' \ DE \ ' \ x^{66} + ' \ 0D \ ' \ x^{65} + ' \ 23 \ ' \ x^{64} + ' \ F9 \ ' \ x^{63} + ' \ 8A \ ' \ x^{62} + ' \ F5 \ ' \ x^{61} + ' \ 5D \ ' \ x^{60} + ' \ B1 \ ' \ x^{59} + ' \ 7C \ ' \ x^{58} + ' \ 46 \ ' \ x^{57} + ' \ 5A \ ' \ x^{56} + ' \ F9 \ ' \ x^{55} + ' \ 10 \ ' \ x^{54} + ' \ EE \ ' \ x^{53} + ' \ 55 \ ' \ x^{52} + ' \ 9D \ ' \ x^{51} + ' \ 8F \ ' \ x^{50} + ' \ C8 \ ' \ x^{49} + ' \ E6 \ ' \ x^{48} + ' \ 9D \ ' \ x^{47} + ' \ C2 \ ' \ x^{46} + ' \ FE \ ' \ x^{45} + ' \ 59 \ ' \ x^{44} + ' \ 3B \ ' \ x^{43} + ' \ 1F \ ' \ x^{42} + ' \ 1F \ ' \ x^{41} + ' \ BC \ ' \ x^{40} + ' \ 02 \ ' \ x^{39} + ' \ 20 \ ' \ x^{38} + ' \ E6 \ ' \ x^{37} + ' \ E6 \ ' \ x^{36} + ' \ 8B \ ' \ x^{35} + ' \ 7C \ ' \ x^{34} + ' \ B9 \ ' \ x^{33} + ' \ 81 \ ' \ x^{32} + ' \ 56 \ ' \ x^{31} + ' \ 95 \ ' \ x^{30} + ' \ 09 \ ' \ x^{29} + ' \ 02 \ ' \ x^{28} + ' \ 4D \ ' \ x^{27} + ' \ 6D \ ' \ x^{26} + ' \ 34 \ ' \ x^{25} + ' \ 5A \ ' \ x^{24} + ' \ 1D \ ' \ x^{23} + ' \ 02 \ ' \ x^{22} + ' \ 3E \ ' \ x^{21} + ' \ FB \ ' \ x^{20} + ' \ 41 \ ' \ x^{19} + ' \ 51 \ ' \ x^{18} + ' \ E6 \ ' \ x^{17} + ' \ EF \ ' \ x^{16} + ' \ 5D \ ' \ x^{15} + ' \ C7 \ ' \ x^{14} + ' \ B1 \ ' \ x^{13} + ' \ 78 \ ' \ x^{12} + ' \ BF \ ' \ x^{11} + ' \ FC \ ' \ x^{10} + ' \ D2 \ ' \ x^{9} + ' \ 51 \ ' \ x^{8} + ' \ FA \ ' \ x^{7} + ' \ BC \ ' \ x^{6} + ' \ A5 \ ' \ x^{5} + ' \ F6 \ ' \ x^{4} + ' \ 15 \ ' \ x^{3} + ' \ 87 \ ' \ x^{2} + ' \ E7 \ ' \ x + ' \ C3 \ '$$

EK B: Geliştirilen Şifrenin Anahtar Planlama Evresinde Kullanılan Döngü Sabitleri

1. döngü sabiti ($RCon_1$)=9C C3 5F 04 6A 78 F5 79

2. döngü sabiti ($RCon_2$)=57 EF 42 69 CA BD DE FC

3. döngü sabiti ($RCon_3$)=68 C9 F1 04 BD EF CC 69

4. döngü sabiti ($RCon_4$)=F5 48 30 44 69 78 C9 B5

5. döngü sabiti ($RCon_5$)=DF 57 02 49 AE FC D5 12

6. döngü sabiti ($RCon_6$)=9A F5 66 7E D3 40 E6 9F

7. döngü sabiti ($RCon_7$)=0F 88 2D CE 65 24 77 8A

8. döngü sabiti ($RCon_8$)=3A B0 E2 5D EF D5 B1 28

9. döngü sabiti ($RCon_9$)=DB E0 90 8C 4E 50 77 49

10. döngü sabiti ($RCon_{10}$)=18 7A 8F 49 36 E3 88 67

KAYNAKLAR

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

Aslan B., Sakallı M. T., Buluş E., Üs Haritalama Tabanlı Cebirsel 8-bit giriş 8-bit çıkışlı S-kutularının Sınıflandırılması, Ağ ve Bilgi Ulusal Sempozyumu 2, Girne-Kıbrıs, 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, Proceedings of International Workshop on the Arithmetic of Finite Fields, WAIFI 2008, Lecture Notes in Computer Science, Springer-Verlag, Vol. 5130, pp. 123-133, 2008.

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 In: Stinson, D.R., Tavares, S. (eds.) SAC 2000. LNCS, vol. 2012, pp. 39–56. Springer, Heidelberg, 2001.

Babbage S., Stream ciphers –what does industry want?, SASC, 2004.

Barreto P. S. L. M., Rijmen V., The Khazad Legacy-Level Block Cipher, Proceedings First open NESSIE Workshop, Leuven, 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. and Shamir A., Differential Cryptanalysis of DES-like Cryptosystems, Journal of Cryptology, Vol 4, No 1 pp. 3-72, 1991.

Biryukov A., Khovratovich D., Related-key Cryptanalysis of the Full AES-192 and AES-256, Cryptology ePrint Archive, Report 2009/317, 2009. Available at *http://eprint.iacr.org/2009/317/*.

Biryukov A., Dunkelman O., Keller N., Khovratovich D., Shamir A., Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds, Cryptology ePrint Archive, Report 2009/374, 2009. Available at *http://eprint.iacr.org/2009/374/*.

Biryukov A., Khovratovich D., Related-key Cryptanalysis of the Full AES-192 and AES-256, Advances in Cryptology, ASIACRYPT'09, 15th International Conference on

the Theory and Application of Cryptology and Information Security, Lecture Notes in Computer Science, Vol. 5912, pp. 1-18, Springer-Verlag, 2009.

Biryukov A., Khovratovich D., Nikolic I., Distinguisher and Related-Key Attack on the Full AES-256, Advances in Cryptology-CRYPTO'09, Lecture Notes in Computer Science, Vol. 5677, pp. 231-249, Springer-Verlag, 2009.

Biryukov, C. D.Canni`ere, J. Lano, S. B. Ors, B. Preneel, Security and Performance Analysis of Aria, Lecture Notes in Computer Science, vol. 4296, 107-117, Springer-Verlag, 2006.

Cannière C. De and Preneel B., The Stream Cipher Trivium, eSTREAM, the ECRYPT Stream Project, 2005, available at: *http://www.ecrypt.eu.org/stream.*

Daemen J., Knudsen L., Rijmen V., The Block Cipher Square Fast Software Encryption (FSE), Volume 1267 of Lecture Notes in Computer Science:149–165, Haifa, Israel: Springer-Verlag. Retrieved on 2007-02-15,1997.

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

Diffie and M. Hellman, New directions in cryptography, IEEE Trans, on Information Theory, IT-22(6):644–654, 1976.

Dobbertin H., Bosselaers A., and Preneel B.. RIPEMD-160: A strengthened version of RIPEMD, Fast Software Encryption, LNCS, vol. 1039, ed. D. Gollmann. Springer-Verlag, Berlin, 71–82, 1996.

ECRYPT, The eSTREAM Project, 2008, *http://www.ecrypt.eu.org/stream*.

Feistel H., Cryptography and Computer privacy, Scientific American, Vol. 228, no. 5, pp. 15-23, 1973.

Ferguson N., Kelsey J., Lucks S., Schneier B., Stay M., Wagner D. and Whiting D., Improved Cryptanalysis of Rijndael, Proceedings of the 7th Fast Software Encryption, Lecture Notes in Computer Science, Springer-Verlag, Vol. 1978, pp. 213-230, 2000.

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., October 25, 1999.

FIPS 197, Advanced Encryption Standard, Federal Information Processing Standard (FIPS), Publication 197, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., November 26, 2001.

Fuller J., Millan W, Linear redundancy in S-boxes, Proceedings of the Fast Software Encryption (FSE 2003), Lecture Notes in Computer Science, Vol. 2887, pp. 74–86 Springer, Berlin, 2003.

Jakobsen T., Knudsen L., The interpolation attack on block ciphers, In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 28–40, Springer, Heidelberg, 1997.

Kam J. B., Davida G. I., Structured Design of Substitution-Permutation Encryption Networks. IEEE Transactions on Computers, Vol. 28, no. 10, pp. 747-753, 1979.

Kayış H., AES Uygulamasının FPGA Gerçeklemelerine Karşı Güç Analizi Saldırısı, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul, 4-7, 10-32, 2006.

Keliher L., Linear Cryptanalysis of Substitution Permutation Networks, Ph.D. Thesis, Queen's University, Kingston, Ontario, Canada, 2003.

Knudsen L. R., Block Ciphers Analysis, Design and Applications, 1994, Ph.D. Thesis, ISSN 0105-8517

Knudsen L. R., The Number of Rounds in Block Ciphers, Public Reports of Nessie Project, May 12,2000.

Knudsen L. R., Truncated and Higher Order Differentials, In Proceedings of 2nd International Workshop on Fast Software Encryption (FSE 1994), Springer-Verlag. pp. 196–211, 1994.

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

Lidl R. and 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 Pres, 2004.

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

May L., Henricksen M., Millan W., Carter G., and Dawson E., Strengthening the Key Schedule of the AES, Proceedings of the 7th Australasian Conference on Information Seucurity and Privacy (ACISP 2002), Lecture Notes in Computer Science, Springer-Verlag, Vol. 2384, pp. 226–240, 2002.

May L., Henricksen M., Millan W., Carter G., E. Dawson, Strengthening the Key Schedule of the AES, Proceedings of The 7th Australasian Conference on Information Seucurity and Privacy (ACISP 2002), Lecture Notes in Computer Science, Vol. 2384, pp. 226-240, Springer-Verlag, 2002.

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

Nyberg K., Differentially uniform mappings for cryptography, Proceedings of Eurocrypt'93, Lecture Notes in Computer Science, Vol. 765, Springer, Berlin, pp. 55-64, 1994.

Phan R. C.-W., Impossible differential cryptanalysis of 7-round Advanced Encryption Standard (AES), Information Processing Letters Vol. 91, pp. 33-38, 2004.

Rimoldi Anna, On algebraic and statistical properties of AES-like ciphers, Ph.D. Thesis, University of Trento, 2009.

Saran N., Kriptografideki Güncel Çalışmalar, 2. Mühendislik ve Teknoloji Sempozyumu, 30 Nisan - 1 Mayıs 2009 / Çankaya Üniversitesi / Ankara

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

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

Schneier B., Applied Cryptography Protocols, Algorithms, and Source code in C, John Wiley & Sons, Inc., 2nd edition, 1996.

Shamir A., Stream Ciphers: Dead or Alive?, ASIACRYPT, 2004.

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, Second Edition, CRC Press, 2002.

SÖNMEZ R., "Veri Şifreleme Standardı (DES) ve Rivest Shamir Adleman (RSA) Güvenlik Algoritmalarının VLSI Tasarımı", Yüksek Lisans Tezi, Hacettepe Üniversitesi, 2002.

Şen Ş., İndirgenmiş SPN (Substitution Permutation Network) Algoritması için Lineer Kriptanaliz Uygulaması, Yüksek Lisans Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, 2006.

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

Wu H., The Stream Cipher HC-256, eSTREAM, the ECRYPT Stream Project., 2005, available at: *http://www.ecrypt.eu.org/stream.*

Youssef A. M., Tavares S.E., Affine equivalence in the AES round function, Discrete Applied Mathematics, Elsevier, 2005.

Youssef A. M., Tavares S.E., Gong G., On Some probabilistic approximations for AES-like s-boxes, Discrete Mathematics, Elsevier, 2006.

Z'aba M. R., Analysis of Linear Relationships in Block Ciphers, Ph.D. Thesis, Queensland University, Brisbane, Australia, 2010.

ÖZGEÇMİŞ

Selma BULUT BÜYÜKGÖZE 1979 yılında Edirne'de doğdu. İlk ve orta öğrenimini Edirne'de tamamladıktan sonra 1998 yılında girdiği Trakya Üniversitesi Mühendislik-Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü'nden 2002 yılında mezun oldu. 2002 yılında Trakya Üniversitesi Kırklareli Teknik Bilimler Meslek Yüksek Okulu'nda Öğretim Görevlisi kadrosuna atandı ve aynı yıl Trakya Üniversitesi Fen Bilimleri Enstitüsü'ne bağlı olarak Yüksek Lisans eğitimine başladı. Ancak çeşitli nedenlerden ötürü Yüksek Lisansla ilişiği 2003 yılında kesildi. 2009 yılında çıkan af yasası ile Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Ana Bilim Dalı'nda Yüksek Lisansa tekrar başladı. 2007 yılında Kırklareli Üniversitesi'nin kurulması ile yapılan kadro aktarımı sonucu Kırklareli Üniversitesi Teknik Bilimler Meslek Yüksek Okulu'na atandı. 2007 yılında Kırklareli Üniversitesi Pınarhisar Meslek Yüksek Okulu'na, 2012 yılında ise Kırklareli Üniversitesi Sağlık Yüksek Okulu'na Öğretim Görevlisi olarak görevlendirildi. Selma BULUT BÜYÜKGÖZE halen Kırklareli Üniversitesi Sağlık Yüksek Okulu'nda Öğretim Görevlisi olarak çalışmaktadır.