T.C

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SİMETRİK ŞİFRELEME TEKNİKLERİNDE ANAHTAR PLANLAMA

HÜSEYİN VURAL

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: YRD. DOÇ. DR. MUHARREM TOLGA SAKALLI

EDİRNE-2014

T.Ü. Fen Bilimleri Enstitüsü onayı

Prof. Dr. Mustafa ÖZCAN

Fen Bilimleri Enstitüsü Müdürü

Bu tezin Yüksek Lisans tezi olarak gerekli şartları sağladığını onaylarım.

Prof. Dr. Yılmaz KILIÇASLAN
Anabilim Dalı Başkanı

Bu tez tarafımca okunmuş, kapsamı ve niteliği açısından bir Yüksek Lisans tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Muharrem Tolga SAKALLI

Tez Danışmanı

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

Juri Üyeleri İmza

Yrd. Doç. Dr. Muharrem Tolga SAKALLI (Danışman)

Yrd. Doç. Dr. Vedat TAŞKIN

Yrd. Doç. Dr. Deniz TAŞKIN

Tarih: 10/07/2014

T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ YÜKSEK LİSANS PROGRAMI

DOĞRULUK BEYANI

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

10/07/2014

Hüseyin VURAL

YÜKSEK LİSANS TEZİ

SİMETRİK ŞİFRELEME TEKNİKLERİNDE ANAHTAR PLANLAMA

T.Ü. Fen Bilimleri Enstitüsü

BİLGİSAYAR MÜHENDİSLİĞİ Anabilim Dalı

ÖZET

Bu tez, simetrik şifreleme algoritmalarından blok şifrelerin anahtar genişletme algoritmaları ile ilgilidir. Literatürde bulunan AES ve ARIA blok şifrelerinin anahtar genişletme algoritmaları incelenmiş ve bu anahtar genişletme algoritmalarının incelenmesinden elde edilen tecrübe ile AES anahtar genişletme algoritmasına dayanan yeni bir anahtar genişletme algoritması 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 Blok şifrelerde kullanılan anahtar genişletme algoritmalarının zaafları incelenmiştir. 4. bölümde AES ve ARIA blok şifreleme algoritmaları ve tasarım stratejileri incelenmiştir. 5. bölümde geliştirilen yeni anahtar genişletme algoritması ve örnek test değerleri ile yeni anahtar genişletme algoritmasının çalışması gösterilmiştir.

Yıl : 2014

Sayfa Sayısı : 98

Anahtar Kelimeler : Kriptografi, blok şifreler, anahtar genişletme algoritmaları, kriptanaliz

MSC. THESİS

KEY EXPANSION ALGORITHMS IN SYMMETRIC ENCRYPTION TECHNIQUES

Trakya University Institute of Natural Sciences

DEPARTMENT OF COMPUTER ENGİNEERİNG

ABSTRACT

This thesis is related to key scheduling in Symmetric Encryption Techniques. In this thesis, the key expansion algorithms of AES and ARIA block ciphers are examined and a new key expansion algorithm based on the AES key expansion algorithm is developed with the experience gained from the examination of these algorithms. In the introduction part of the thesis, an introduction to the symmetric key ciphers, block and stream ciphers and cryptanalysis attacks against these ciphers are given. In the second part of the thesis, a mathematical background of the theory of finite fields is given. In Chapter 3, an overview of weaknesses of key expansion algorithms used in these block ciphers is given. In Chapter 4, the AES and ARIA block ciphers and design strategies of them are examined. In Chapter 5, a new expansion algorithm based on the AES key expansion algorithm is developed and the sample test values are shown to work with the new key expansion algorithm.

Year : 2014

Page : 98

Key Words : Cryptography, Block ciphers, Key expansion algorithms, Cryptanalysis

TEŞEKKÜR

Bu tezin ortaya çıkmasında desteklerini esirgememiş olan yakınlarıma, dostlarıma, hocalarıma teşekkürü bir borç bilirim.

Tezin ortaya çıkmasındaki katkılarından ve emeklerinden dolayı değerli hocam ve danışmanım Sayın Yrd. Doç Dr. M. Tolga SAKALLI hocama teşekkürlerimi sunarım.

Bu konuda bana çok büyük destek vermiş aileme; babam M. Üryan VURAL'a, annem Hanım VURAL' a ve kardeşim Giray VURAL' a teşekkür ederim.

Güzel bir çalışma ortamı sağladıkları ve dostlukları için başta Adıyaman Üniversitesi Kahta Meslek Yüksek Okulu Müdür'ü Doç Dr. A. Zafer TEL olmak üzere değerli çalışma arkadaşlarıma teşekkür ederim.

Bu tezi 2 yıldan önce aramızdan ayrılmış olan ve bana eğitim alanında her zaman en iyiyi hedeflememi tavsiye etmiş çok sevdiğim saygıdeğer dedem Ali VURAL' a atfediyorum.

ÖZET ..... **İV**

ABSTRACT ..... **V**

TEŞEKKÜR ..... **Vİ**

ŞEKİLLER LİSTESİ ..... **İX**

TABLOLAR LİSTESİ ..... **Xİ**

BÖLÜM 1 ..... **1**

GİRİŞ **1**

1.1 KRİPTOLOJİ 2

1.1.1 Kriptografi *3*

1.1.2 Kriptanaliz *4*

1.2 SİMETRİK YAPILAR 5

1.2.1 Blok Şifreler *6*

1.2.1.1 Feistel Mimari 7

1.2.1.2 SPN Ağları 9

1.2.2 Blok Şifrelerde Kullanılan Önemli Yapılar *10*

1.2.2.1 S Kutuları: 10

1.2.2.2 Doğrusal Dönüşümler 11

1.2.2.3 Anahtar Genişletme Algoritmaları 11

1.2.3 Akan Şifreler *12*

1.3 TEZİN ÖNEMİ VE GEREKÇESİ 13

BÖLÜM 2 ..... **14**

SONLU CİSİMLER TEORİSİNE GİRİŞ **14**

2.1 SONLU CİSİM 15

2.2 HALKA 18

2.3 GF(2^N) (GALOİS) CİSMİ 18

2.4 GF(2^N) CİSMİNDE TOPLAMA VE ÇIKARMA 19

2.5 GF(2^N) CİSMİNDE ÇARPMA 19

2.6 GF(2^N) CİSMİNDE ÇARPIM İŞLEMİNE GÖRE TERS ALMA 20

BÖLÜM 3 ..... **23**

BLOK ŞİFRELEMEDE KULLANILAN ANAHTAR GENİŞLETME ALGORİTMALARINDAKİ ZAAFLAR 23

3.1 BLOK ŞİFRELERDE KULLANILAN ANAHTAR GENİŞLETME ALGORİTMALARININ ÖZELLİKLERİ 23

3.1.1 Tek Yönlü Fonksiyon 24

3.1.2 Minimum Karşılıklı İlişki 25

3.1.3 Etkin Uygulama 27

3.2 AES ANAHTAR GENİŞLETMESİNDE BLOKLAR ARASI GEÇİŞ 27

3.2.1 İlişkili-Anahtar Saldırısı 27

BÖLÜM 4 29

ÖNEMLİ İKİ BLOK ŞİFRE ALGORİTMASI 29

4.1 AES 29

4.1.1 SubByte (Byte Yerdeğiştirme) Dönüşümü 34

4.1.2 ShiftRows (Satırları Öteleme) Dönüşümü 38

4.1.3 MixColumns (Sütunları Karıştırma) Dönüşümü 38

4.1.4 AES Anahtar Genişletmesi 40

4.2 ARIA 44

4.2.1 Yer Değiştirme Adımı 46

4.2.2 Difüzyon adımı 49

4.2.3 ARIA Anahtar Genişletme 50

BÖLÜM 5 54

AES ANAHTAR GENİŞLETMESİ İÇİN YENİ GELİŞTİRİLEN ANAHTAR GENİŞLETME ALGORİTMASI 54

5.1 YENİ GELİŞTİRİLEN ANAHTAR GENİŞLETME ALGORİTMASININ ÖZELLİKLERİ 54

BÖLÜM 6 79

SONUÇLAR 79

KAYNAKLAR 80

ÖZGEÇMİŞ 85

TEZ ÖĞRENCİSİNE AİT TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER 86

ŞEKİLLER LİSTESİ

Şekil 1.1. Kriptografinin kayda geçen ilk kullanımı Hiyerografik yazılardır.....2

Şekil 1.2. Blok Şifreleme ve Akan Şifreleme.....6

Şekil 1.3. Blok Şifrede şifreleme ve şifre çözme işlemi.....7

Şekil 1.4. Feistel mimarisi.....8

Şekil 1.5. 1-döngülük ve 16 bit giriş-çıkışlı bir SPN ağı.....9

Şekil 3.1. AES Anahtar Genişletme Algoritması.....26

Şekil 4.1. AES Algoritmasının genel yapısı.....33

Şekil 4.2. Bir döngülük AES genişletmesi.....34

Şekil 4.3. S-kutusu için Bitsel Doğrusal Dönüşüm.....34

Şekil 4.4. AES algoritmasında bir byte Yer Değiştirme aşaması.....37

Şekil 4.5. ShiftRows dönüşümü.....38

Şekil 4.6. Sütunları Karıştırma Dönüşümü.....39

Şekil 4.7. AES Anahtar Genişletme.....41

Şekil 4.8. AES-128 bit Anahtar Genişletme.....43

Şekil 4.9. ARIA algoritması şifreleme ve şifre çözme işlemi.....45

Şekil 4.10. Tekli döngülerde kullanılan S-kutuları.....46

Şekil 4.11. Çiftli döngülerde kullanılan S-kutuları.....46

Şekil 4.12. ARIA W_i değerlerinin elde edilmesi.....52

Şekil 5.1. AES-128 bit anahtar genişletme tasarımı S-D_1-S.....58

Şekil 5.2. AES-128 bit anahtar genişletme tasarımı D_1-S-D_1.....63

Şekil 5.3. AES-128 bit anahtar genişletme tasarımı D_1-S-D_2.....68

Şekil 5.4. AES-256 bit anahtar genişletme tasarımı S-D_1-S.....73

TABLOLAR LİSTESİ

Tablo 2.1. Standart cebirsel yol ve GF(2^n) sonlu cismi ile polinom toplamaları.....19

Tablo 3.1. İstenilen Anahtar Genişletme Algoritmasının Özellikleri.....24

Tablo 4.1. NIST Aday Şifreler.....30

Tablo 4.2. Anahtar-Blok-Döngü Kombinasyonları.....32

Tablo 4.3. AES blok şifresinde kullanılan S-Kutusu.....35

Tablo 4.4. AES blok şifresinde kullanılan S-kutusunun tersi.....36

Tablo 4.5. Döngü Sabitleri (RCON değerleri).....42

Tablo 4.6. ARIA algoritmasında kullanılan S_1 S kutusu.....47

Tablo 4.7. ARIA algoritmasında kullanılan S_1^{-1} S kutusu.....47

Tablo 4.8. ARIA algoritmasında kullanılan S_2 S kutusu.....48

Tablo 4.9. ARIA algoritmasında kullanılan S_2^{-1} S kutusu.....48

Tablo 4.10. Gizli anahtar uzunluğuna bağlı sabit değerler.....51

Tablo 5.1. 128-bit S-D_1-S tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi işleminin adım adım sonuçları.....59

Tablo 5.2. AES-128 için önerilen anahtar genişletme algoritmalarından S-D_1-S yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 128 farklı bit pozisyonu için ortalama bit değişimleri.....61

Tablo 5.3. 128-bit D_1-S-D_1 tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi.....64

Tablo 5.4. AES-128 için geliştirilen anahtar genişletme algoritmalarından D_1-S-D_1 yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 128 farklı bit pozisyonu için ortalama bit değişimleri.....66

Tablo 5.5. 128-bit D_1-S-D_2 tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi.....69

Tablo 5.6. AES-128 için geliştirilen anahtar genişletme algoritmalarından D_1-S-D_2 yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 128 farklı bit pozisyonu için ortalama bit değişimleri.....71

Tablo 5.7. 256-bit S-D_1-S tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi.....74

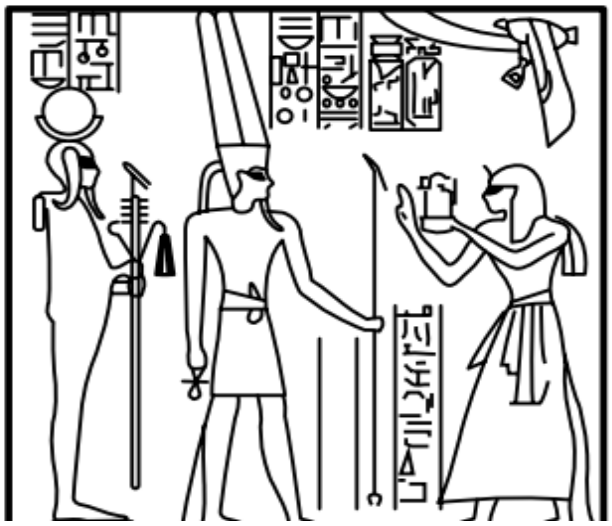
Tablo 5.8. AES-256 için geliştirilen anahtar genişletme algoritmasının S-D_1-S yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 256 farklı bit pozisyonu için ortalama bit değişimleri.....77

BÖLÜM 1

GİRİŞ

İletilen mesajların içeriğini gizleme geleneği çok eskilere dayanmaktadır. Yüzyıllar boyunca insanlar kendi aralarında iletişim kurarken, gönderdikleri bilginin istenmeyen kişilerin eline geçmesine veya iletişim sırasında değiştirilmemesine de dikkat ve özen göstermişlerdir [1]. Bu konudaki kanıtlar Mısır tarihinin ilk zamanlarına kadar uzanır. İçeriği gizlenerek gönderilen ilk metin yaklaşık 4000 yıl önce eski Mısır'da küçük bir şehir olan Menet Khufu'da bir kral olan KHNUMHOTEP II 'nin mezar anıtındaki hiyeroglifsel olarak yazılmış olan metin olarak kabul edilir. Şekil 1.1'de bunla ilgili örnek gösterilmektedir.

Geçmişten günümüze dek birçok şifreleme tekniği kullanılmıştır. Günümüzde ki modern iletişim tekniklerinden bir tanesi olan bilgisayarlar ağları aracılığıyla bilgiler iletir ve bu iletişim şekli güvenlik açıklarını da beraberinde getirmiştir [2].



Şekil 1.1. Kriptografi'nin kayda geçen ilk kullanımı Hiyerografik yazılardır [1]

1.1 Kriptoloji

Kriptoloji yunanca *cruptos* (gizli) ve *logos* (bilim, çalışma) kelimelerinin birleşiminden meydana gelmiştir. Bu nedenle kelime anlamı olarak gizleme bilimi anlamına gelir. Bilgilerin belli bir sisteme göre şifrelenip güvenli bir yol ile alıcıya iletilmesi ve alıcıya iletilmiş olan şifrelenmiş mesajların şifrelerinin çözülmesi (açık metne dönüştürülmesi) işlemleri kriptoloji biliminin alanları arasındadır [3].

Kriptolojiyi temel olarak iki alana ayrılmaktadır: Kriptografi ve Kriptanaliz. *Kriptografi* kripto sistemlerinin tasarımı ile ilgili çalışmaları, *Kriptanaliz* ise kripto sistemlerine yönelik saldırı metotlarını içerir. Bu iki alan birbiriyle yakın ilişki içindedir. Bir kripto sistem oluşturulurken tasarım ve güvenlik sistemin iki önemli parçasıdır [1].

Kripto sistemler aşağıda belirtilen bazı nedenlerden dolayı kullanılır:

Gizlilik: Bilgi iletimi sırasında, bilgiyi ileten taraf ilettiği bilginin istenmeyen erişimlere karşı korunaklı ve güvenli olmasını ister. Aynı konu bilginin saklanması içinde geçerlidir.

Kimlik Doğrulama: Bu özellik imza ile eşdeğerdir. Mesajı alan kişi, mesajın doğru taraftan geldiğini veya daha sonra göndericinin iletimini inkâr etmeye çalışması durumunda ilgili yerden geldiğini kanıtlamak ister.

Bütünlük: Bilgiyi alan kişi gönderilen bilgi kendisine eriştiğinde bu bilginin iletim sırasında üçüncü taraflarca değiştirilmediğinden emin olmak ister [1].
(Bilginin sadece yetkili kullanıcılar tarafından değiştirilebilmesini temin etmeyi amaç edinir)
İnkar Edememe: Bir kişinin bir önceki taahhüdünü veya eylemini inkar edememesini temin etmeyi amaç edinir.

Yüzyıllar boyunca kripto sistemler askeri ve diplomatik iletişimde kullanılmıştır [4]. Bugün bilgisayarlar sanayide ve sivil olarak çok büyük bir iletişim aracı olarak kullanılmaktadır ve bu yapılırken yukarıda belirtilen amaçların sağlanması için kripto sistemler kullanılmaktadır [1].

1.1.1 Kriptografi

Kriptografi; güvenlik mühendisliğinin matematikle buluşmasıdır. Birçok modern güvenlik protokollerinin temelinin oluşmasında Kriptografi'nin katkısı olmuştur [5]. Oxford sözlüğünde Kriptografi kısaca kod yazma veya çözme sanatı olarak tanımlanmıştır. Bu tanım tarihsel olarak doğru olabilir ama modern Kriptografi'nin niteliklerini kapsamamaktadır. Bunun birinci sebebi; sadece gizli iletişimin problemlerine odaklanıyor olmasıdır. Bu, tanımda bulunan "kod" kelimesine açıkça vurgu yapılmasından anlaşılmaktadır. İkinci sebebi; Kriptografi'nin sanatın bir formu olarak tanımlanmasından kaynaklanmaktadır. Gerçektende 20. yüzyıla kadar Kriptografi bir sanat olarak kabul edilmiştir. Kişisel yetenek ve bilgiyle iyi kodlar üretmek veya mevcut kodlardan birini kırmak mümkün olmuştur. 20. yüzyılın sonunda Kriptografi alanında radikal değişimler yaşanmıştır. Çok sayıda teori ortaya çıkmış ve Kriptografi alanındaki özenli çalışmalar bir bilim dalı olarak kabul görmüştür. Buna ek olarak Kriptografi şu anda gizli iletişimden çok daha fazlasını teşkil etmektedir. Örnek olarak mesaj doğrulama, sayısal imza, gizli anahtarları değiştirmek için gereken protokoller, protokol doğrulama, elektronik ihale ve seçimler, sayısal hesaplar verilebilir. Kısaca modern Kriptografi gizli iletişim ile ilgili bir sanat dalı olmaktan çıkıp dünya üzerindeki tüm insanların kullanabileceği güvenli sistemler üretmeye yönelik bir bilim olmuştur. Bunun anlamı Kriptografi'nin bilgisayar biliminde giderek daha merkezi bir konuma gelmesidir [5].

Kriptografi de temel amaç iki birim mesajlaşırken dışarıdan birilerinin bu mesajı anlaşılır formda elde edememesidir. Bunu sağlanması için bir çok yöntem mevcuttur. Kriptografi bu yöntemler arasından açık metni şifreli metne çevirme ve şifreli metni de açık metne çevirme yöntemleri ile ilgilidir. Açık metni şifreli metne çevirme işlemine şifreleme, tersi işlemede şifre çözme denir. Eğer şifrelenmiş bir mesaj içeriği ve şifreleme yöntemi bilinmeden okunursa bu işleme "şifre kırma" adı verilir.

Genelde, şifreli metin sayısının artması şifreyi kırmayı daha basit bir hale getirir. Bu yüzden şifreleme mekanizmasının düzenli olarak değiştirilmesi şifrenin kırılmasına karşı iyi bir önlemdir [6].

1.1.2 Kriptanaliz

Kriptanaliz yunanca c*ruptos* (gizli) ve *analyein* (çözme) kelimelerinin birleşiminden meydana gelmiştir. Şifreli metinden açık metin elde edilmesi amacıyla yapılan çalışmalar kriptanaliz olarak bilinmektedir. Kriptanaliz matematiksel ve karmaşıklık açısından kriptografi ile benzer zorluk seviyesindedir [6].

Doğrusal [7] ve diferansiyel kriptanaliz [8] blok şifreler üzerinde uygulanan iki önemli kriptanaliz çeşididir. Doğrusal kriptanaliz; EUROCRYPT '93 konferansında Matsui tarafından Data Encryption Standard (DES)'e [9] karşı bir kuramsal saldırı olarak sunulmuştur ve daha sonra DES'e karşı yapılan kriptanaliz çalışmalarında başarılı bir şekilde kullanılmıştır. Diferansiyel kriptanaliz ilk olarak CRYPTO '90 konferansında Biham ve Shamir tarafından DES'e karşı bir sadırı çeşidi olarak sunulmuştur. Her iki saldırının ilk başlardaki hedefi DES olmasına karşın çok sayıdaki diğer blok şifrelere karşı uygulanabilirliğinin olması, her iki saldırı çeşidini tüm blok şifrelerin güvenliği ile ilgili çalışmalarda dikkate alınması gereken önemli bir etken haline getirmektedir. Örneğin, Ulusal Standartlar ve Teknoloji Enstitüsü (The National Institute of Standards and Technology - NIST)) tarafından düzenlenen Gelişmiş Şifreleme Standardı (Advanced Encrytion Standard - AES) [10] yarışmasında, aday katılımcılar tarafından sunulacak şifrelerin tasarımlarında doğrusal ve diferansiyel kriptanaliz saldırıları engelleyecek tekniklerin bulunması şartı konulmuştur [11].

Kriptografik yapıların tasarımında birincil ve en önemli kural Kerckhoff prensibidir. Kerchkoff bir kripto sistemin *gizli anahtar dışında* tüm ayrıntılarının bilinmesi gerektiğini belirtmiştir. Bu prensibe göre bir saldırganın şifreye ait tüm

ayrıntılarına sahip olduğu varsayımı ile aşağıda verilen yaygın saldırı modellerinden birini kullanarak saldırı yaptığı kabul edilir [12].

Kripto sistemlerine açık metin, şifreli metin veya farklı yönlerden yaklaşımlara göre çok sayıda kriptanaliz modeli vardır. Saldırgan bu modellerden birini kullanan saldırı tipini seçerek şifreye saldırabilir. Aşağıda en çok kullanılan kriptanaliz modelleri belirtilmiştir:

Sadece şifreli metin saldırısı (Ciphertext-only attack): Bu yöntemde kriptanalist açık metin ile ilgili her hangi bir bilgiye sahip değildir sadece şifreli metin bilgisine sahiptir ve bu bilgi üzerinde çalışır.

Bilinen açık metin saldırısı (Known-plaintext attack): Bu yöntemde kriptanalist şifreli metnin karşılığı olan açık metnin bir kısmını bilir ve bu bilgiyi kullanarak açık metni şifreli metine dönüştüren algoritmada kullanılan anahtarı tahmin etmeye çalışır.

Seçilmiş açık metin saldırısı (Chosen-plaintext attack): Bu yöntemde kriptanalist bilinen bir açık metne ve bu metnin şifrelenmiş şekline sahiptir ama açık metnin şifrelenmesi için gereken anahtar bilgisinden yoksundur. Kriptanalist açık metin ile şifrelenmiş metni karşılaştırarak anahtarı tahmin etmeye çalışır.

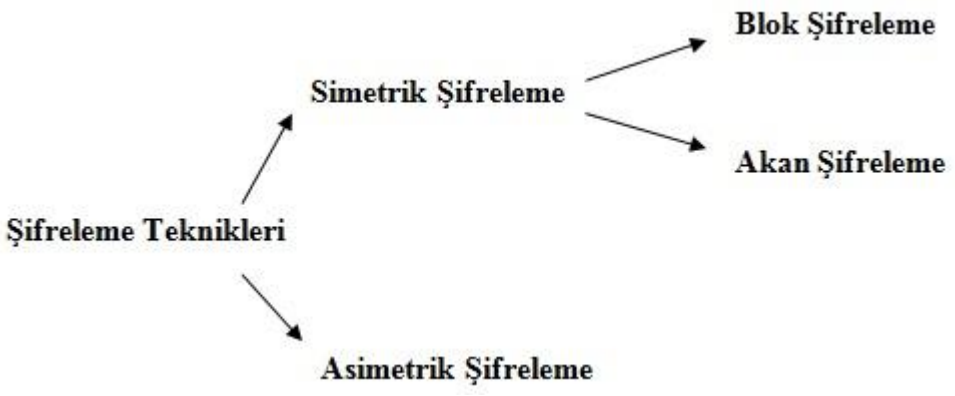
Seçilmiş şifreli metin saldırısı (Chosen-ciphertext attack): Bu saldırı modelinde kriptanalist istediği şifreli metni açık metine çevirebilir [5].

Günümüzde kriptanaliz'in çok geniş kullanım alanları vardır: devletler diğer devletlerin diplomatik ve askeri iletişimlerini elde etmek için kriptanaliz tekniklerini kullanırlar, şirketler geliştirdikleri güvenlik ürünlerinin güvenlik özelliklerini test etmeleri için kriptanalistlere test ettirirler. Kriptanalistler ile kriptologlar arasında süre giden mücadele kriptoloji bilimini ileriye taşımaktadır.

1.2 Simetrik Yapılar

Simetrik şifrelemede hem şifreleme hem de şifre çözme işlemlerinde aynı anahtar kullanılır. Simetrik şifreleme algoritması; açık metni ve gizli anahtarı veri girişi olarak alıp, çıktı olarak şifreli metni üretir. Algoritmadaki amaç şifreli metnin açık metne dönüşümünün gizli anahtar kullanılarak sağlanmasıdır dolayısıyla gizli anahtarın bilinmediği durumlarda dönüşüm gerçekleşemez. Açık metine erişim sadece gizli

anahtar bilindiği durumlarda mümkün olmalıdır. Simetrik şifreleme teknikleri blok şifreleme ve akan şifreleme olmak üzere ikiye ayrılır [13]. Şekil 1.2'de genel şifreleme tekniklerinin yapısı gösterilmiştir.



Şekil 1.2. Blok Şifreleme ve Akan Şifreleme [14]

1.2.1 Blok Şifreler

Simetrik anahtar şifreleme işlemlerinin sabit uzunluktaki bit gruplarına (blok) uygulanmasına blok şifreleme denir. Bir blok şifreleme algoritması ile sabit uzunluktaki metin şifreli metne çevrilir. Blokların uzunluğu genellikle 64-bit veya 128-bittir. 128-bit uzunluğundaki açık metnin 128-bit uzunluğundaki şifreli metne dönüşümünde 128-bit uzunluğundaki açık metin *giriş verisi*, 128-bit uzunluğundaki şifreli metin ise *çıkış verisi* olarak işlem görür. Açık metnin şifreli metne dönüşümünde gizli anahtar da kullanılır. Gizli anahtar blok şifreleme algoritmasında açık metinden sonraki ikinci giriş verisidir [15].

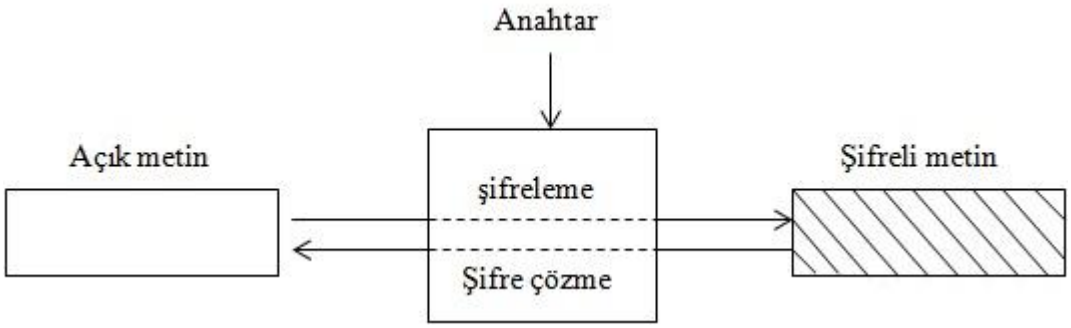
Blok şifrelerin güvenliğinin sağlanmasında Shannon'un önerdiği aşağıdaki prensiplerden yararlanılmaktadır. Bu prensipler açık metnin bilinen statiksel özelliklerini temel alarak yapılacak kriptanalizleri engellemeyi amaçlamaktadır.

- *Confusion* (Karıştırma): Bu prensibe göre karıştırma, şifreli metin ile anahtar arasındaki ilişkiyi mümkün olduğu kadar karmaşık yapmaktadır. Anahtarda

bulunan her harf şifreli metin bloğundaki her karakteri etkilemektedir. Bununla beraber, iyi bir karıştırma ancak şifreli metindeki her karakterin anahtar üzerindeki bir kaç kısma bağlı olması ile başarılabilmektedir ve bu bağımlılık gözlemciye göre rastgele olmalıdır. Karıştırmayı çok fazla sağlamayan şifreler frekans analizini kullanan saldırılar için iyi birer adaydır.

- *Diffusion* (Yayılım): Karıştırma, açık metnin statiksel yapısının şifreli metnin içerisine geniş bir ölçekte yayılması özelliğini belirtmektedir. Yayılım ise karıştırmanın tersine açık metnin bir karakterin şifreli metin üzerindeki birçok karakteri etkilemesini sağlamaktadır. Açık metindeki karakterlerin istatistiksel sıklıkları göstermektedir ki yayılım bu karakterleri şifreli metindeki karakterleri rastgele etkileyecek şekilde yaymaktadır. Bu sayede şifreye bir istatistiksel saldırı yapmak için ihtiyaç duyulan şifreli metin sayısı artmaktadır [12].

Blok şifreleme akan şifrelemeye oranla daha geniş alanlarda kullanılabilir. Ağ tabanlı simetrik kriptografik uygulamalarının büyük çoğunluğunda blok şifreler kullanır. Şekil 1.3'te blok şifreler için şifreleme ve şifre çözme işlemlerinin genel bir hali gösterilmektedir.

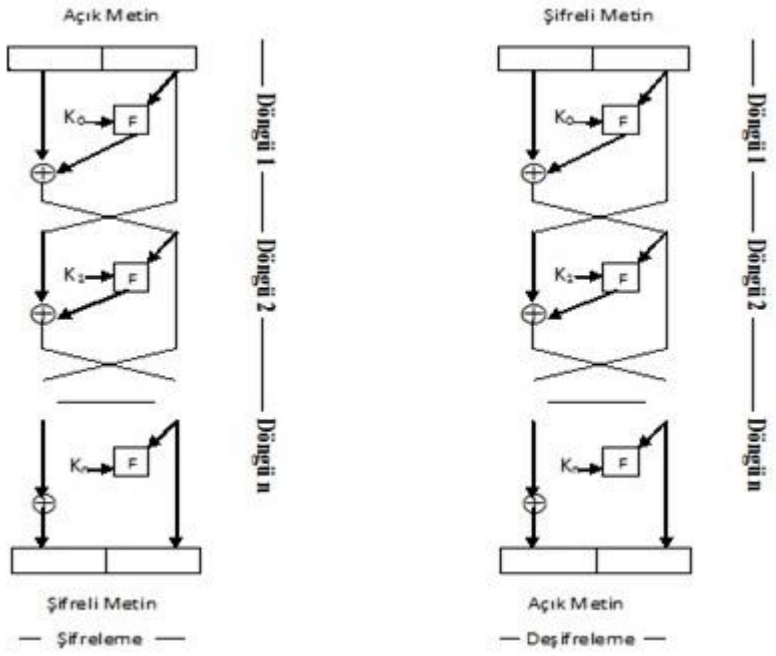


Şekil 1.3. Blok Şifrede şifreleme ve şifre çözme İşlemi

1.2.1.1 Feistel Mimari

Feistel ağı özel bir yapıya sahip olan bir blok şifredir. IBM kriptologu Horst Feistel tarafından isimlendirilmiştir ve daha çok Feistel şifresi olarak bilinir. Blok

şifrelerin büyük bir kısmı bu yapıyı kullanmaktadır. Feistel şifresinin avantajı şifreleme ve şifre çözme evrelerinin birbirine çok benzemesidir, bazı durumlarda aynı olmasıdır, farklı olan tek yan ihtiyaç duydukları anahtar genişletme algoritmasının birbirlerinin tersi olmasıdır. Bu nedenle böyle bir şifreyi kodlamak için gereken kod uzunluğu yarı yarıya azalmaktadır. Feistel mimarisi Şekil 1.3'de gösterilmiştir.



Şekil 1.4. Feistel mimarisi [11]

Feistel ağları aşağıda belirtilen işlemlerin birçok döngüde tekrar edilmesi ile oluşurlar.

- Bit-karıştırma (genelde permütasyon kutuları veya P-kutuları olarak bilinir)
- Basit doğrusal olmayan fonksiyonlar (genelde yer değiştirme kutuları veya S-kutuları olarak bilinir)
- XOR yardımıyla doğrusal karıştırma

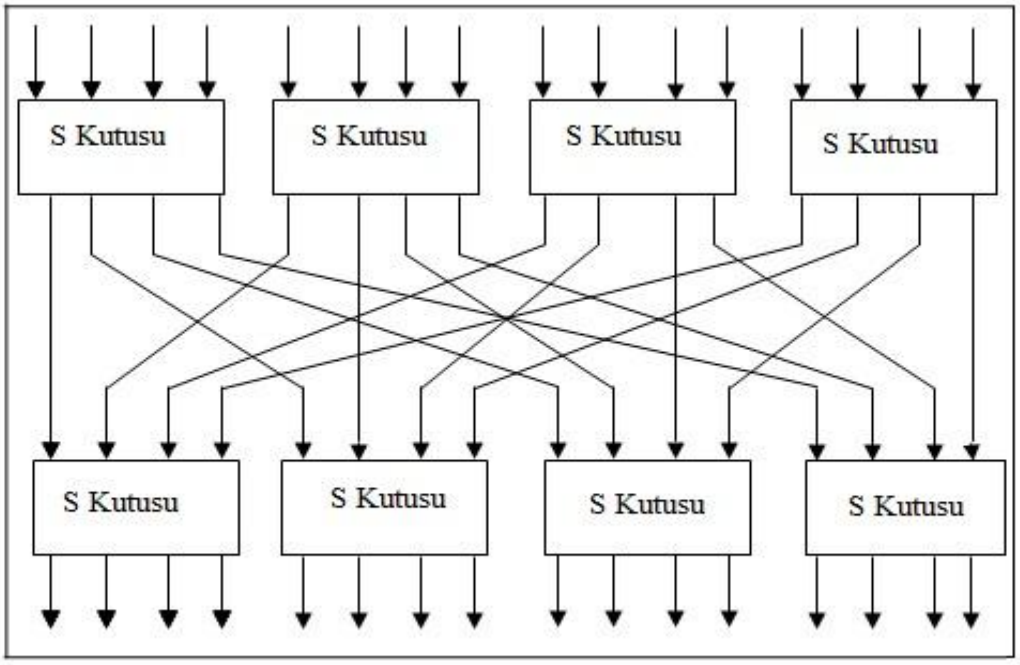
Bu işlemler Shannon'un ilkeleri olan karıştırma ve yayılımı büyük oranda üretir. Bit-karıştırma yayılımı sağlarken, yer değiştirme karıştırmayı sağlar. Birçok modern

blok şifreleme algoritması, örneğin DES algoritması, Feistel ağlarını temel almıştır. Feistel ağlarının yapısı ve özellikleri ayrıntılı bir şekilde kriptanalistler tarafından araştırılmıştır [13].

1.2.1.2 SPN Ağları

Shannon 1940'lı yıllarda güçlü şifrelerin yer değiştirme ve transpozisyon-yayılım (transposition) işlemlerinin tekrar tekrar bir arada birleşimi ile olabileceğini ileri sürmüştür. Bir şifrenin yayılım ve yer değiştirme özelliklerine sahip olması gerektiğini belirtmiştir ek olarak bilinmeyen anahtar değerlerinin eklenmesi ile saldırganın şifreye yönelik saldırısında açık metindeki sembollerde karmaşıklığın artacağını belirtmiştir.

İlk blok şifreler yer değiştirme ve permütasyon döngülerinden oluşan basit ağlardı ve SP-ağları olarak isimlendirilmişlerdi. Aşağıdaki Şekil 1.4'de 16 bit girişli bir SP-ağını göstermektedir [16].



Şekil 1.5. 1-döngülük ve 16 bit giriş-çıkışlı bir SPN ağı

1.2.2 Blok Şifrelerde Kullanılan Önemli Yapılar

Bölüm 1.2.1.2 de belirtildiği gibi modern blok şifreler köklerini Shannon'ın dönüm noktası olan makalesinde [17] sunulan karıştırma (confusion) ve yayılım (diffusion) prensiplerinden almaktadır [18]. 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 [19].

1.2.2.1 S Kutuları:

S-kutuları bitlerin karıştırılmasını sağlayan ve doğrusal olmayan bir dönüşümdür. Modern şifreleme algoritmalarının tasarımında doğrusal olmayan dönüşümlerin kullanılması gerekli bir özelliktir. Bu dönüşümler doğrusal kriptanaliz'e [7] ve diferansiyel kriptanaliz'e [8] karşı şifrenin dirençli olmasını sağlamaktadır. Doğrusal olmayan dönüşümler S-kutuları ile uygulanmaktadır. Giriş verisi p bitleri ve çıkış verisi q bitleri olan bir S-kutusu $p \times q$ ile gösterilmektedir. DES algoritması 8 tane 6×4 S-kutusu kullanmaktadır. S-kutuları 8-bit işlemciler üzerinde çalışan yazılım uygulamaları için tasarlanmaktadır. 8×8 S-kutularına sahip blok şifreler SAFER, SHARK ve AES'tir. S-kutuları rastgele (random), düzensiz bir haritalama kullanılarak (chaotic) veya sonlu Galois cisminde matematiksel yapılar (sonlu cisimde üs alma, ters alma) kullanılarak oluşturulabilir.

AES şifreleme algoritmasında, S-kutusu GF(2) ve GF(2^8) Galois cisimlerinde iki dönüşüm kullanılarak tasarlanmıştır. S-kutusu o anki veri bloğu üzerinde her byte'ın yer değiştirme tablosu kullanılarak başka bir byte ile yer değiştirmesi sayesinde doğrusal olmayan bir dönüşüm olarak kullanılmaktadır. S-kutusunun tasarımında kullanılan birinci dönüşüm S-kutusu GF(2^8) cisminde byte değerinin çarpma işleminde göre tersini elde etmektedir. Bu cebirsel işlem, cebirsel saldırılara karşı risk yaratabileceğinden dolayı bu dönüşümü ikinci bir dönüşüm (bit bazında) affine (doğrusal) dönüşüm izlemektedir. Doğrusal dönüşümün seçilmesinin sebebi doğrusal olmama özelliğinin korunarak Bölüm 3'te gösterilen SubByte (Byte Yer Değiştirme) aşamasını cebirsel olarak karmaşık hale getirmektir [20].

1.2.2.2 Doğrusal Dönüşümler

Doğrusal dönüşümler blok şifrelerde yayılımı sağlamak için kullanılmaktadır. Yayılım katmanı blok şifrelerin bilinen en önemli saldırı çeşitleri olan doğrusal kriptanaliz ve diferansiyel kriptanaliz'e karşı dayanıklı olmasında etkin bir rol oynamaktadır. 1994 yılında Vaudenay [21,22] çoklu permütasyon'un (multipermutation) sağlanması için kriptografik yapılarda MDS (Maximum Distance Separable) matrislerinin kullanılması önerisinde bulunmuştur. Çoklu permütasyon özelliği taşımayan bir yayılım katmanı için kriptanalitik saldırıların nasıl yapılacağını da göstermiştir. Bu kavram daha sonra Daemen [23] tarafından dallanma sayısı (branch number) olarak ifade edilmiştir. Düşük dallanma sayısı değerine sahip yayılım katmanını kullanan blok şifreler güçlü doğrusal olmama özellikleri taşıyan S-kutuları kullansalar dahi diferansiyel ve doğrusal kriptanaliz saldırılarına karşı kritik derecede zayıflıklara sahip olmaktadırlar [24].

Tanım 1.1. Bir $n \times n$ boyutundaki A: $(\{0,1\}^m)^n \rightarrow (\{0,1\}^m)^n$ matrisinin diferansiyel dallanma sayısı aşağıda verildiği gibi tanımlanabilir:

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

Tanım 1.2. Bir $n \times n$ boyutundaki A: $(\{0,1\}^m)^n \rightarrow (\{0,1\}^m)^n$ matrisinin doğrusal dallanma sayısı aşağıda verildiği gibi tanımlanabilir:

$$\beta(\mathrm{A}) = \min \{wt(x) + wt(A^T \cdot x^T) \mid x \in (\{0,1\}^m)^n,\ x \neq 0\} \qquad (2)$$

Tanım 1.1 ve Tanım 1.2'de n yayılım katmanı A'daki S-kutularının sayısını ve m bu S-kutularının giriş ve çıkış büyüklüğünü temsil eder [25].

1.2.2.3 Anahtar Genişletme Algoritmaları

Blok şifrelerin güvenliğinde anahtar genişletme algoritmaları önemli bir yer tutmaktadır. Aşağıda verilen saldırı çeşitlerine karşı şifrenin dirençli olmasında önemli bir rol oynamaktadır:

- Şifre anahtarının bir kısmının saldırgan tarafından bilindiği saldırılar
- Şifre anahtarının bilindiği veya seçilebildiği saldırılar
- İlişkili-anahtar saldırıları

Ek olarak şifrenin simetrik yapısının elimine edilmesinde önemli bir rol oynamaktadır:

- Döngüler arasındaki simetri: Tüm döngülerdeki döngü dönüşüm yapısı aynıdır. Bu eşit yapılar arasındaki simetri, döngüden bağımsız döngü sabitlerinin anahtar genişletme algoritmasında kullanılması ile giderilmektedir.

Anahtar genişletme aşağıdaki kriterlere göre seçilmelidir [26]:

- Tersinir dönüşümler kullanması
- Geniş bir yelpazedeki işlemciler üzerinde hızlı olması
- Simetrik yapının yok edilmesi için döngü sabitleri kullanması
- Yeterli doğrusal olmama özelliğinin sağlanması
- Şifre anahtarının bir kısmının veya döngü anahtarının bitlerinin bilinmesinin diğer döngü anahtarlarının bitlerinin bilinmesine imkân vermemesi

Örneğin AES algoritmasında şifre anahtarı anahtar genişletme algoritmasına giriş olarak alınmaktadır ve anahtar genişletme algoritması şifreleme algoritmasının her döngüsünde farklı anahtarın kullanılması amacıyla farklı anahtarlar üretmektedir. Bu işlem sonrası elde edilen anahtarlar şifrenin her döngüsünde, açık metnin şifreli metne dönüşümünde veya şifreli metnin açık metne dönüşümünde, aynı döngülerde aynı fonksiyonların kullanılması nedeniyle oluşacak simetrinin bozulması için kullanılmaktadır [27].

1.2.3 Akan Şifreler

Akan şifreler şifreleme algoritmalarının önemli bir parçasıdır. Zamana bağlı olarak değişen şifreleme dönüşümleri kullanılarak açık metindeki karakterler tek tek şifrelenir. Akan şifreler blok şifrelerden donanımsal olarak genelde daha hızlıdırlar ve donanımsal olarak daha az karmaşıklığa sahiptirler. Arabelleğe alınmanın limitli veya karakterlerin tek tek alınıp işlenmesi gereken zamanlarda (örneğin bazı

telekomünikasyon uygulamalarında) akan şifrelerin kullanılması daha uygun ve bazı durumlarda zorunludur [28].

Akan şifrelemenin yaygınlaşmasının en büyük nedeni Shannon'un *tek kullanımlık şerit* çalışmasıdır, bu çalışmanın orijinali Vernam şifresi olarak bilinir. *Tek kullanımlık şerit*; tamamen rastgele bitlerden veya karakterlerden seçilmiş uzun bir akan anahtar metni kullanır. Bu akan anahtar, açık metin bitleri ya da karakterleri ile modüler olarak eklenerek (günümüzde çoğunlukla XOR işlemi) işlem görür. Akan anahtar, mesaj ile aynı uzunlukta olup sadece bir kez kullanılır; bazı durumlarda çok geniş akan anahtar gerekebilir [29].

Akan şifrelerde 5 farklı tasarım stratejisinin olduğu söylenebilir [19]. Bunlar;

- LFSR (Linear Feedback Shift Registers- Doğrusal Geribeslemeli Ötelemeli Saklayıcılar) Tabanlı Akış Şifreler,
- NFSR (Nonlinear Feedback Shift Registers- Doğrusal Olmayan Geribeslemeli Ötelemeli Saklayıcılar) Tabanlı Akış Şifreler,
- Blok Şifre Tabanlı Akış Şifreler,
- Karıştırma Tabanlı Akış Şifreler,
- Hash Fonksiyon Tabanlı Akış Şifreler şeklindedir.

1.3 Tezin Önemi ve Gerekçesi

Bu tezde günümüz iki önemli blok şifreleme algoritmasında kullanılan anahtar genişletme algoritmaları incelenerek bu algoritmalardaki önemli zaaflar belirlenmiş ve bu zaafları gidermeye yönelik yeni bir anahtar genişletme algoritması tasarlanmıştır. Bu bağlamda tasarlanan ve AES anahtar genişletme algoritmasından esinlenerek geliştirilen yeni anahtar genişletme algoritması her hangi bir 128-bit, 256-bit blok şifrenin 128-bit, 256-bit anahtar genişletme algoritmalarında sırasıyla kullanılabilir. Ayrıca geliştirilen bu tasarımlar üzerinde yapılacak bazı küçük değişiklikler ile 512-bit'lik blok şifreler için anahtar genişletme algoritmalarının tasarımı mümkündür.

BÖLÜM 2

SONLU CİSİMLER TEORİSİNE GİRİŞ

Sonlu cisimler teorisi, hata düzeltme kodları, sayısal sinyal işleme ve kriptografi gibi alanlarda kullanılan bir teoridir. Bu bölümde bu teori kriptografide kullanımı açısından incelenecektir. Bu teori ile ilgili daha detaylı bilgi [19]'ten elde edilebilir.

Bir cisim aşağıda verilen aksiyomları toplama ve çarpma işlemine göre sağlayan boş olmayan bir F kümesidir.

$\forall a, b, c \in F$ olmak üzere

a-) a ve b F kümesinin elemanı olmak üzere, kapalılık özelliğinin sağlanması gereğince (a+b) değeri de F kümesinin elemanı olmalıdır

$\forall a,b \in F : (a+b) \in F$

b-) a ve b F kümesinin elemanı olmak üzere, değişme özelliğinin sağlanması gereğince (a+b)=(b+a) eşitliği sağlanmalıdır.

$\forall a,b \in F : (a+b)=(b+a)$

c-) a ve b F kümesinin elemanı olmak üzere, birleşme özelliğinin sağlanması gereğince (a+b)+c=a+(b+c) eşitliği sağlanmalıdır.

$\forall a,b,c \in F : (a+b)+c=a+(b+c)$

d-) a F kümesinin elemanı olmak üzere F kümesinde a+0=a eşitliğini sağlayan bir tane etkisiz eleman vardır

$\forall a \in F$, $\exists 0 \in F$: a+0 =a

e-) a ve b F kümesinin elemanı olmak üzere bir b elemanı vardır ki a+b=0 olsun.

$\forall a \in F$, $\exists$b $\in F$: a+b =0

2.1 Sonlu Cisim

Sonlu cisimde yapılan hesaplamalar standart tamsayı hesaplamalarından farklıdır. Sonlu cisimde sonlu sayıda eleman vardır ve sonlu cisimde yapılan tüm işlemler sonucunda elde edilecek değerler, bu sonlu sayıdaki elemanlardan bir tanesi olmalıdır. Sonsuz sayıda farklı sonlu cisim vardır ve bu farklı sonlu cisimlerin her birinin eleman sayısının p^n formunda olması gerekmektedir. Bu formdaki p asal sayıyı temsil ederken n pozitif bir tamsayıyı temsil etmektedir. p asal sayısı cismin karakteristiği olarak adlandırılmaktadır, n pozitif tam sayısı ise cismin boyutu olarak adlandırılmaktadır.

Eleman sayısı p^n olan bir sonlu cisim GF(p^n) olarak ifade edilmektedir ve aynı zamanda *Galois Cismi* olarak adlandırılmaktadır. GF(p) sonlu cismi p asal sayı olmak üzere mod p'ye göre tamsayıların halkasıdır. Bu nedenle tamsayılar üzerinde olağan işlemler (toplama, çıkarma, çarpma) mod p'ye göre yapılmakatadır. Örneğin, GF(5) sonlu cisminde 4+3=7 işlemi mod 5'e göre 2'ye indirgenir. Bölme işlemi, mod p'nin tersi ile çarpıma eşittir ve bu işlem Genişletilmiş Öklid Algoritması (Extended Euclidean Algorithm) [30] ile hesaplanmaktadır. GF(2) sonlu cismi, sonlu cisimler arasında özel bir durumdur çünkü toplama işlemi *dışlamalı ya da* (exclusive OR, $\oplus$) yapısında ve çarpma işlemi *ise* (AND) yapısındadır. Sadece 1 sayısı tersinir olduğundan bölme işlemi *birim fonksiyona* (identitiy function) eşittir.

GF(p^n) sonlu cisminin elemanları GF(p) sonlu cisminde n'den daha küçük dereceli polinomlar ile ifade edilebilmektedir. Bu durumda işlemler GF(p) sonlu cisminde derecesi n olan ve indirgenemez bir polinom olan R ile mod R'ye göre yapılmaktadır.

GF(2) sonlu cisminin elemanları (p asal sayısı 2 olduğu için) *ikili sayılar* (binary

numbers) olarak gösterilebilmektedir. Örneğin, karakteristiği 2 olan bir sonlu cisimde aşağıda gösterilen ifadeler aynı değeri ifade etmektedir.

Polinom: $x^6 + x^4 + x + 1$

İkili eleman: {01010011}

Hexadecimal: {53}

Teorem 1.1. Z_p (mod p işlemine göre kalanların oluşturduğu küme) p değeri asal sayı ise bir cisim oluşturur [31].

Örnek 1.1. Z_3 (mod 3 işlemine göre kalanların oluşturduğu küme) bir cisim oluşturur. Aşağıdaki tablolarda Z_3 için toplama ve çarpma değerler kümesi gösterilmiştir.

Z_3 Toplama:

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

Z_3 Çarpma:

x	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

Örnek 1.2. Z_6 (mod 6 işlemine göre kalanların oluşturduğu küme) bir cisim oluşturmaz çünkü 2 ve 4 değerlerinin çarpma işlemine göre tersi yoktur. Z_6 için toplama ve çarpma değerlerinin bulunduğu tablolar aşağıda gösterilmiştir.

Z_6 Toplama:

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

Z_6 Çarpma:

x	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

2.2 Halka

Tanım 2.1. *R* boş olmayan bir küme olsun. *R* üzerinde " + " ve " · " ikili işlemleri verilsin. Eğer;

I. $(R , +)$ bir değişmeli grup ise

II. *R* çarpma işlemine göre birleşme özelliğine sahipse; yani her $a,b,c \in R$ için

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

ise,

III. *R* üzerinde dağılma özellikleri sağlanırsa; yani her $a,b,c \in R$ için

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

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

ise $(R , + , .)$ sıralı üçlüsüne bir halka denir.

Eğer her $a,b \in R$ için a.b = b.a ise halkaya değişmeli halka denir. *R* nin toplamsal birimi 0_R ile gösterilir ve buna *R* nin sıfırı denir. Eğer her $a \in R$ için $a.1_R = 1_R.a = a$ olacak şekilde $1_R \in R$ varsa 1_R elemanına halkanın birim elemanı ve halkaya da birimli halka denir [32].

2.3 GF(2^n) (Galois) Cismi

Bu tez, GF(2) cismi ve bu cismi taban olarak kullanan yapılar üzerinedir. Karakteristiği 2 olan $GF(2^n)$ sonlu cisminin kriptografide kullanımı yaygındır. 2^n adet eleman içermektedir. 2 karakteristikli *Galois cisimlerinde* gösterim olarak polinomsal gösterim yaygın olarak kullanılmaktadır. GF(2^n) için polinomsal baz, $\{x^{n-1}, x^{n-2},..., x^2, x, 1\}$ kümesinden oluşur. GF(2^n)'nin bir elemanının polinomsal gösterimi, polinomsal baz vektörünün her bir elamanının GF(2)'ye ait bir elemanla çarpılması ile elde edilir [33]. Örneğin 8-bit ikili değer {10111010} Hexadecimal notasyonda {BA} polinomsal olarak şu şekilde gösterilir:

$$a(x) = x^7 + x^5 + x^4 + x^3 + x.$$

2.4 GF(2^n) Cisminde Toplama ve Çıkarma

Karakteristiği 2 olan bir sonlu cisimde, mod 2'ye göre toplama, mod 2'ye göre çıkarma ve XOR işlemleri birbirine denktir. Bu nedenle;

Polinom: $(x^7 + x^6 + x^2 + x) + (x^6 + x^4 + x^3 + x+1) = x^7 + x^4 + x^3 + x^2 + 1$
İkili: {11000110} + {01011011} = {10011101}
Hexadecimal: {c6} + {5b} = {9d}

Yukarıdaki işlemler eğer normal GF(2) sonlu cisminde yapılmasaydı $x^6 + x^6 = 2x^6$ sonucunu üretecekti ancak bu sonuç mod 2'ye göre indirgendiğinde $0x^6$ sonucunu üretecek ve işlemden çıkarılacaktır. Aşağıdaki tabloda bir kaç polinomun toplamları; normal cebirsel toplam ve karakteristiği 2 olan bir sonlu cisim ile yapılarak karşılaştırılmıştır.

Tablo 2.1. Standart cebirsel yol ve GF(2^n) sonlu cismi ile polinom toplamaları

p_1	p_2	p_1+p_2 (standart cebirsel işlem ile)	p_1+p_2 (GF(2^n) sonlu cismi ile)
x^3+x+1	x^3+x^2	$2x^3+x^2+x+1$	x^2+x+1
x^4+x^2	x^6+x^2	$x^6+x^4+2x^2$	x^6+x^4
$x+1$	x^2+1	x^2+x+2	x^2+x
x^3+x	x^2+1	x^3+x^2+x+1	x^3+x^2+x+1
x^2+x	x^2+x	$2x^2+2x$	0

2.5 GF(2^n) Cisminde Çarpma

GF(2^n) sonlu cisminde iki polinomun çarpımı, iki polinomun aritmetik çarpımının alınmasıyla elde edilir. Bu iki polinomun çarpımı sonucunda elde edilen polinom'un derecesi GF(2^n) sonlu cisminin derecesinden daha yüksek dereceli olabilir. Bu nedenle çarpım sonucunda oluşan polinomu GF(2^n) sonlu cismini oluşturan

indirgenemez polinomu kullanarak indirgemek gerekmektedir.

Örneğin AES algoritmasında karakteristiği 2 olan 256 elemana sahip bir sonlu cisim kullanmıştır ve bu sonlu cisim (Galois cismi) $GF(2^8)$ cismi olarak adlandırılmaktadır. AES algoritmasında kullanılan $GF(2^8)$ cisminin tanımlanmasında (dolayısıyla çarpma işlemlerinde kullanılmak üzere) 8. dereceden 30 indirgenemez polinom arasından aşağıda verilen indirgenmez polinom seçilmiştir:

$$x^8 + x^4 + x^3 + x + 1$$

Örneğin $GF(2^8)$ sonlu cisminde {a7}•{16} = {61} 'dir (• işlemi yukarıda belirtildiği gibi iki *n*-bit değerin bir indirgenemez polinom yardımıyla çarpımını tanımlamaktadır) Bu işlem aşağıda gösterilmiştir:

$$(x^7 + x^5 + x^2 + x + 1)\bullet(x^4 + x^2 + x) =$$

$$(x^{11} + x^9 + x^6 + x^5 + x^4) + (x^9 + x^7 + x^4 + x^3 + x^2) + (x^8 + x^6 + x^3 + x^2 + x) =$$

$$x^{11} + 2x^9 + x^8 + x^7 + 2x^6 + x^5 + 2x^4 + 2x^3 + 2x^2 + x =$$

$$x^{11} + x^8 + x^7 + x^5 + x$$

ve

$$x^{11} + x^8 + x^7 + x^5 + x \mod (x^8 + x^4 + x^3 + x + 1) =$$

$$x^6 + x^5 + 1 = \{01100001\} = 61.$$

2.6 $GF(2^n)$ Cisminde Çarpım İşlemine Göre Ters Alma

$GF(2^n)$ sonlu cisminde ters alma işlemi için genel olarak iki farklı yöntem verilebilir [34,35]. Bu yöntemlerden ilki $GF(2^n)$ cisminin üretilerek tablo şeklinde saklanması ile ters alma işleminin uygulanmasına dayanır. İkinci yöntem ise ikili öklid

algoritmasının (binary euclidean algoritm) uygulanması ile gerçekleştirilir. Bu yöntemlerden cismin elde edilmesi ile ters alma işleminin gerçekleştirilmesi bu bölümde kısaca aşağıdaki gibi verilmektedir. GF(2^n) cisminin üretilmesi herhangi bir elemanın üslerinin alınması ile gerçekleştirilebilir. Ancak üs alınan elemanın tüm cisim elemanlarını üretebilmesi için ilkel eleman olması gerekir.

Tanım 2.2. Sonlu bir çarpma grubu G için, $g \in G$ elemanının derecesi $g^m = 1$ olacak şekilde en küçük pozitif m sayısıdır.

Tanım 2.3. GF(p) sonlu cisminin bir elemanının derecesi p-1 ise bu elemana ilkel eleman denir.

Örnek 1.3. Z_{13}'te (mod 13 işlemine göre oluşturulan cisim) ilkel elemanlardan biri aşağıda gösterildiği gibi 2'dir:

$$2^0 \bmod 13 = 1 \qquad 2^8 \bmod 13 = 9$$
$$2^1 \bmod 13 = 2 \qquad 2^9 \bmod 13 = 5$$
$$2^2 \bmod 13 = 4 \qquad 2^{10} \bmod 13 = 10$$
$$2^3 \bmod 13 = 8 \qquad 2^{11} \bmod 13 = 7$$
$$2^4 \bmod 13 = 3 \qquad 2^{12} \bmod 13 = 1$$
$$2^5 \bmod 13 = 6$$
$$2^6 \bmod 13 = 12$$
$$2^7 \bmod 13 = 11$$

Örnek 1.4. GF(2^3) cisminin $x^3 + x + 1$ indirgenemez polinomu ile tanımlansın. O zaman 8 elemanlı cisim (0 elemanı hariç) aşağıdaki gibi elde edilebilir:

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

GF(2^n) cisminde herhangi bir elemanın tersi, (tersi alınacak eleman x^m şeklinde ifade edilirse) aşağıdaki gibi verilebilir:

$$x^{2^n-1-m} \bmod 2^n - 1.$$

Örneğin bu cisimdeki 5 Hexadecimal ($x^6 = x^2 + 1$) değerinin tersi $x^{7-6} = x$ ya da Hexadecimal 2 olarak elde edilir.

BÖLÜM 3

BLOK ŞİFRELEMEDE KULLANILAN ANAHTAR GENİŞLETME ALGORİTMALARINDAKİ ZAAFLAR

Bu bölümde blok şifrelerin anahtar genişletme algoritmalarından ve özelde AES blok şifresinin anahtar genişletme algoritmasından ve zafiyetlerinden bahsedilecektir.

3.1 Blok Şifrelerde Kullanılan Anahtar Genişletme Algoritmalarının Özellikleri

AES (Advanced Encryption Standard) şifresi blok şifreler için en önemli standarttır. Bu nedenle AES şifresinin güvenliği yüksek önemdedir. Blok şifreleme algoritmalarının tasarımına çok fazla dikkat ve özen gösterilmişken aynı dikkat ve özen blok şifrelerin anahtar genişletme algoritmalarında gösterilmemiştir [36, 37, 38]. AES şifresinin anahtar genişletme algoritması bazı saldırı yöntemlerinin(ör. Kare saldırısı [39]) etkili olmasına sebebiyet veren zafiyetlere sahiptir. Bu zafiyetler yavaş yayılım ve bit sızıntısıdır [39].

Blok şifrelerde güçlü anahtar genişletme algoritması olmasının amacı, şifreye karşı teorikte veya pratikte yapılabilecek saldırıların, anahtar genişletme algoritmasının zafiyetlerinden faydalanılarak yapılmasını engellemektir. Bu nedenle blok şifrelerde kullanılacak anahtar genişletme algoritmaları saldırılara karşı güçlü olacak şekilde tasarlanmalıdır.

Tasarımcıların blok şifreleme algoritmalarında kullandıkları Shannon metotlarını

[17] (bit karıştırma ve difüzyon özelliklerini sağlamak için), anahtar genişletme algoritmalarında da kullanmaları ile anahtar genişletme algoritmalarının güvenliğini blok şifre algoritmalarına benzer şekilde arttıracaktır [39].

1993 yılında Bilham [36], dikkatli tasarlanmamış bir anahtar genişletme algoritması ile üretilen anahtarların, birbirleri ile ilişkili olabileceğini ve anahtarlar arasındaki bu ilişkinin bazı saldırılara karşı şifrenin güvenliğini azaltacağını belirtmiştir. Aynı yıl Knudsen, Feistel tabanlı şifrelerin güvenli olması için yeterli olmamakla beraber mutlaka olması gereken 3 özellikten bahsetmiştir [40]. Tablo 3.1'de, Knudsen tarafından güvenli bir anahtar genişletme algoritmasının sahip olması gereken özellikler gösterilmektedir.

Tablo 3.1 İstenilen Anahtar Genişletme Algoritmasının Özellikleri

Özellik 1: çarpışmaya dayanıklı tek-yönlü fonksiyon (fonksiyon tersine çevrilebilir olmamalıdır)
Özellik 2: minimum karşılıklı ilişki (tüm alt anahtar bitleri ve ana anahtar bitleri arasında)
Özellik 3: etkin uygulama

3.1.1 Tek Yönlü Fonksiyon

Şifreleme anahtarı bilinmeyen bir blok şifreleme algoritması tek yönlü bir fonksiyon olarak değerlendirilebilir. Tersinir olmayan anahtar genişletme evresi blok şifreleme algoritmasına tek yönlü fonksiyon özelliğini kazandırır. Alt anahtarlarından bir tanesi bilinen bir anahtar genişletme evresinin bu alt anahtar bilgisi ile önceki alt anahtar(lar)ı elde edilemiyorsa anahtar genişletme evresi için tersinir özellikte değildir, elde edilebiliyorsa anahtar genişletme evresi için tersinir özelliktedir denilir [41, 42,43,44].

AES blok şifresinde bu özelliğe dikkat edilmemiştir ve AES blok şifresinin

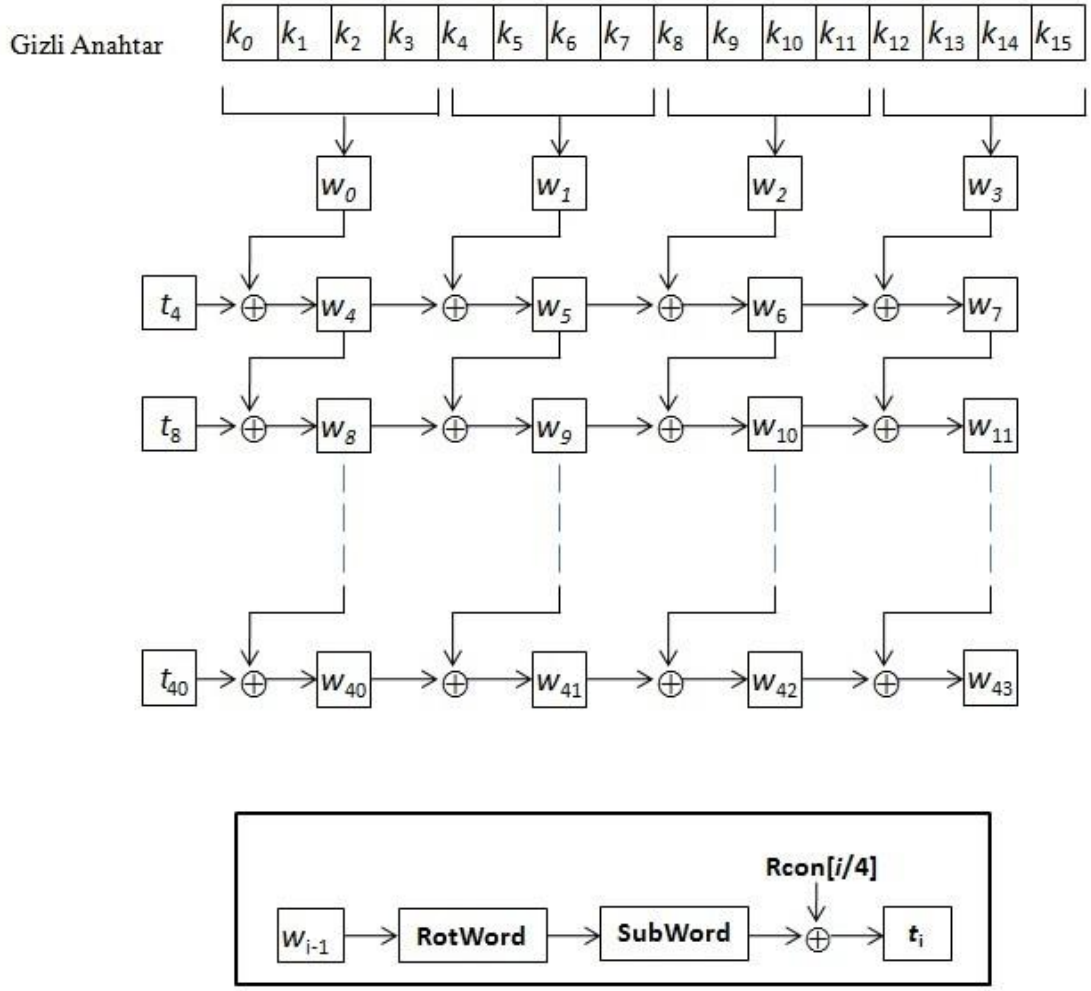
kullandığı anahtar genişletme algoritmasında bir alt anahtarın elde edilmesi halinde diğer alt anahtarların elde edilmesi mümkündür.

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 [45] çalışmasında AES blok şifresine karşı imkansız diferansiyel saldırısının uygulanmasında bu sızıntı problemi kullanılmıştır. Bu problemin önüne geçmek için alt anahtarların birbirinden bağımsız olarak üretilmesi bir yöntem olarak kullanılabilir [19].

3.1.2 Minimum Karşılıklı İlişki

Bu özellik ana anahtar ile alt anahtarlar arasında bit sızıntısını gidermeyi ve şifredeki zaafları ortadan kaldırarak şifreye karşı yapılacak saldırıların karmaşıklığını arttırmayı amaçlar. Tablo 3.1'de ki özelliklerden birincisi; *i* anahtarı ile kendisinden sonraki anahtar olan *i* + 1 anahtarı veya kendisinden önceki anahtar olan *i* - 1 anahtarı arasındaki bit sızıntısını doğrudan önleyerek anahtarlar arasında minimum karşılıklı ilişki sağlamaya yöneliktir.

AES anahtar genişletme algoritmasının tasarımı ve alt anahtarlar arasındaki ilişki, 2 anahtar (w[4] - w[5] - w[6] - w[7] ve w[8] - w[9] - w[10] - w[11]) için aşağıdaki Şekil 3.1'de AES blok şifresinin anahtar genişletme algoritması üzerinde gösterilmiştir:



Şekil 3.1. AES Anahtar Genişletme Algoritması

- Birinci alt anahtar : w[4] - w[5] - w[6] - w[7]
- İkinci alt anahtar : w[8] - w[9] - w[10] - w[11]

w[8] = w[4] ⊕ t8

w[9] = w[5] ⊕ w[8]

w[10] = w[6] ⊕ w[9]

w[11] = w[7] ⊕ w[10]

Yukarıdaki iki alt anahtardan ikincisinin bilindiğini varsayalım. AES anahtar genişletme algoritmasının tasarımından dolayı bir bilinmeyenli denklem ile w[5] - w[6]

- w[7] kelimelerinin değerleri elde edilebilir. Geriye kalan w[4] kelimesinin değeri;

$$t_8 = SubWord(RotWord(w_{8-1})) \oplus RCon_{8/4}$$

denkleminden t_8 değeri hesaplanarak bu değerin w[8] = w[4] $\oplus$ t8 denkleminde kullanılması ile elde edilebilir. Sonuç olarak, ikinci alt anahtarın bilinmesi yukarıdaki örnekte gösterildiği gibi birinci alt anahtarın bilinmesine imkan verebilir. Bu istenmeyen bir durumdur ve minimum karşılıklı ilişki ilkesi ile ters düşmektedir. Yukarıdaki örnek ile AES blok şifresinin anahtar genişletme algoritmasının tersinir olduğu gösterilmiştir.

3.1.3 Etkin Uygulama

Şifreleme algoritması ve anahtar genişletme algoritması uygulanabilirlik ve güvenlik açılarından birbirlerini tamamlayıcı olmalıdırlar. Şifreleme algoritmasının en uygun bileşenlerinin tekrar kullanılması ile fazladan kod yazma maliyetine gerek kalmadan hızlı bir uygulama elde edilebilir [46].

3.2 AES Anahtar Genişletmesinde Bloklar Arası Geçiş

AES'in 1997 yılında DES (Data Encryption Standard) yerine kullanılacak şifreleme algoritması seçilmesinden itibaren, dünya üzerindeki birçok araştırmacı AES'e karşı çeşitli kriptanaliz saldırıları gerçekleştirmiştir [47]. Bu kriptanaliz saldırıları arasında en etkili olanlar kare saldırısı (square attack) [48,49] ve imkansız diferansiyel kriptanalizi'dir (impossible lineer cryptanaliz) [48,50]. İmkansız diferansiyel saldırısı, AES anahtar genişletme algoritmasının bit sızıntısı problemini nedeniyle yapılabilmektedir.

2000 yılında Bilham ve Keller AES üzerinde 5 döngüye kadar imkansız diferansiyel kriptanalizi gerçekleştirmişlerdir [51]. 2001 yılında Cheon AES üzerinde 6 döngüye kadar imkansız diferansiyel kriptanalizi gerçekleştirmiştir [48]. AES üzerinde

son olarak 7 döngüye kadar imkansız diferansiyel saldırısı gerçekleştirilebilmiştir. AES üzerinde gerçekleştirilen en iyi imkansız diferansiyel kriptanalizi 7 döngüye kadardır ve bu kriptanaliz yöntemi AES anahtar genişletme algoritmasının zaaflarından dolayı uygulanabilmiştir [47].

3.2.1 İlişkili-Anahtar Saldırısı

Yavaş yayılım problemi, AES-192 (192 bit anahtar kullanan AES blok şifresi) ve AES-256 (256 bit anahtar kullanan AES blok şifresi) için ilişkili anahtar saldırılarında kullanılmıştır [36]. Saldırgan, ilişkili-anahtar saldırılarını kullanarak blok şifrelerin anahtar genişletme algoritmasında değişiklikler yapma imkanı elde edebilir. [36,52]. Buna ek olarak saldırgan, iki ya da daha fazla farklı gizli anahtar arasında ve açık metinler arasında ilişki seçebilir. Bir bakış açısına göre bu saldırı çeşidinde açık metinlerin şifrelenmesi ilişkili anahtarlar ile sağlanır. İlişkili anahtar saldırısı diğer saldırı çeşitlerinden sadece bir gizli anahtar kullanması ile ayrılır. İlişkili anahtar saldırısı gerçekte çok pratik olmamasına rağmen AES-192 [53], AES-256 [54], SHACAL-1 [55] ve KASUMİ [56] gibi blok şifrelerin tüm döngülerine karşı yapılan saldırılarda fayda sağlamıştır. Ayrıca zaman karmaşıklığı açısından Biryukov [57] 10 döngüye kadar bir AES algoritmasına karşı pratik bir saldırıyı göstermiştir.

BÖLÜM 4

ÖNEMLİ İKİ BLOK ŞİFRE ALGORİTMASI

Bu bölümde AES ve ARIA blok şifreleme algoritmaları ve bu algoritmalarda kullanılan anahtar genişletme algoritmalarının tanıtımı yapılacaktır.

4.1 AES

AES blok şifresinden önce geliştirilmiş kriptografik algoritmaların çoğu çeşitli problemlere sahiptir ve bu kriptografik algoritmalar modern hesaplama sistemleriyle kolayca kırılabilmektedir. Buna ek olarak DES algoritmasının özellikle anahtar uzayının küçük olması bu algoritmanın güvenliği hakkında soru işaretleri doğurmuştur. Bu nedenlerden dolayı yeni bir şifreleme algoritmasına ihtiyaç doğmuştur ve bu yeni geliştirilecek algoritmanın bilinen tüm saldırılara karşı dirençli olması amaçlanmıştır. Bununla beraber Ulusal Standartlar ve Teknoloji Enstitüsü (The National Institute of Standards and Technology - NIST) yeni bir standart'ın oluşturulmasında yardımcı olmak istemiştir. Diğer yandan DES algoritması ile ilgili ihtilaflar ve Amerika Birleşik Devletleri'nin bazı birimlerinin yıllar boyunca güvenli kriptografinin yayılmasını engelleme çabaları bu konuda şüphelerin oluşmasına sebep olmuştur. Yine de NIST teknik birikime ve yeni şifre oluşturma çabalarına öncülük edecek gerekli kaynaklara sahip tek birim olmuştur [58].

NIST yeni bir şifre tasarımı oluşturma veya bu konuda yardımcı olmak yerine

dünya üzerinde her isteyenin katılabileceği bir şifre tasarım yarışması düzenlenmesine karar vermiştir. Yarışma 02/01/1997 tarihinde duyurulmuştur. Geliştirilecek yeni şifreleme algoritmasının birçok gereksinimi karşılayacak özelliklere sahip olmasının yanında tüm tasarım ayrıntılarının belgelenmesi gerekmekteydi. Aday algoritmalar teslim edildikten sonra bir kaç yıl boyunca kriptografik konferanslarda incelendi. Yarışmanın ilk safhasında 15 algoritma kabul edildi ve ikinci safhasında bu sayı 5'e düşürüldü. Tablo 4.1'de kabul edilen 15 algoritma gösterilmiştir ve koyu yazılanlar son 5'e kalanlardır. Algoritmalar dünyanın en şöhretli kriptocuları ve NIST'in kendisi tarafından etkinlik ve güvenlik yönlerinden test edilmiştir.

Tüm bu araştırmalardan sonra NIST Rijndael olarak bilinen algoritmayı seçmiştir. Rijndael iki Belçikalı kriptocu tarafından geliştirilmiştir. Dr. Joan Daemen ve Dr. Vincent Rijmen. 26 Kasım 2001'de AES (Rijndael algoritmasının standartlaştırılmış versiyonu) FIPS standardı olarak belirlenmiştir.

Tablo 4.1. NIST Aday Şifreler

ALGORİTMA	**SUNAN**
CAST-256	Entrust Teknoloji
CRYPTON	Gelecek Sistemleri
DEAL	Richard Outerbridge,Lars Knudsen
DFC	CNRS- Centre National pour la Recherche Scientifique
E2	NTT-Nippon Telgraf ve Telefon
FROG	TecApro
HPC	Rich Schroeppel
LOK197	Lawrie Brown,Josef Pieprzyk,Jennifer Seberry

MAGENTA	Telekom Almanya
MARS	IBM
RC 6	RSA labaratuar
Rijndael	Joaen Daemen, Vincent Rijmen
SAFER+	Cylink Şirketi
Serpent	Ross Anderson, Eli Biham, Lars Knudsen
Twofish	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

AES (Rijndael) algoritması bir simetrik blok şifredir. Bunun anlamı şifreleme ve şifre çözme işlemlerinde aynı anahtar kullanılır. AES algoritmasında veri blokları 128 bit uzunluğunda olmalıdır fakat anahtar uzunluğu 128,192 veya 256 bit uzunluğunda olabilir. Seçilen anahtarın uzunluğuna bağlı olarak AES algoritması AES-128, AES-192 ve AES-256 olarak isimlendirilir.

Rijndael, aşağıdaki özelliklere sahip bir algoritma olarak tasarlanmıştır [58]:

- Bilinen tüm saldırılara karşı dirençli olması.
- Geniş aralıktaki platformlarda hızlı ve kod uygulanabilirliğinin olması.
- Tasarımının basit olması.

AES algoritmasında; giriş bloğunun, çıkış bloğunun ve Durum'un (Durum, şifrenin işleyişi esnasında herhangi bir andaki 128-bit bloğa verilen addır ve 4×4 boyutunda byte matrisi şeklinde ifade edilir.) uzunluğu 128 bittir. Durum'da bulunan 128 bit veri ***Nb*** kelimeleri ile gösterilir, ***Nb*** kelimesinin uzunluğu 32 bit ve sayısı 4'tür.

AES algoritmasında; şifre anahtarının uzunluğu 128, 192 veya 256 bittir. Şifre anahtarı ***Nk*** kelimeleri ile gösterilir, ***Nk*** kelimesinin uzunluğu 32 bittir ve şifre

anahtarının uzunluğunun 128,192 ve 256 bit olmasına bağlı olarak ***Nk*** kelimelerinin sayısı 4,6 veya 8 olmaktadır.

AES döngü sayıları anahtar uzunluğuna bağlı olarak değişmektedir. Örneğin, anahtar uzunluğu 128 bit seçilmişse döngü sayısı 10, 192 veya 256 bit seçilmişse döngü sayısı sırasıyla 12 ve 14 olmaktadır. Kullanılan anahtar uzunluğu 128 bit ve döngü sayısı ***Nr*** ile gösterilirse ***Nk***=4 olduğunda ***Nr***=10, ***Nk***=6 olduğunda ***Nr=12*** ve ***Nk***=8 olduğunda ***Nr***=14 olmaktadır [10].

Tablo 4.2. Anahtar-Blok-Döngü Kombinasyonları [10]

	Anahtar Uzunluğu (*Nk Kelimeleri*)	**Blok Boyutu** **(***Nb kelimeleri***)**	**Döngü Sayısı** **(***Nr***)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES algoritması **Döngü anahtarı ekleme** aşaması ile başlar, her biri dört aşamadan oluşan dokuz döngü ile devam eder ve üç aşamadan oluşan onuncu döngü ile son bulur. Bu uygulama şifreleme ve şifre çözme işleminde sadece şu yönden değişir; şifre çözme işleminde, döngünün her aşamasında şifreleme algoritmasının karşılığının tersi alınır. Şifreleme algoritmasının ilk dokuz döngüsünün her biri aşağıdaki aşamaları kapsamaktadır:

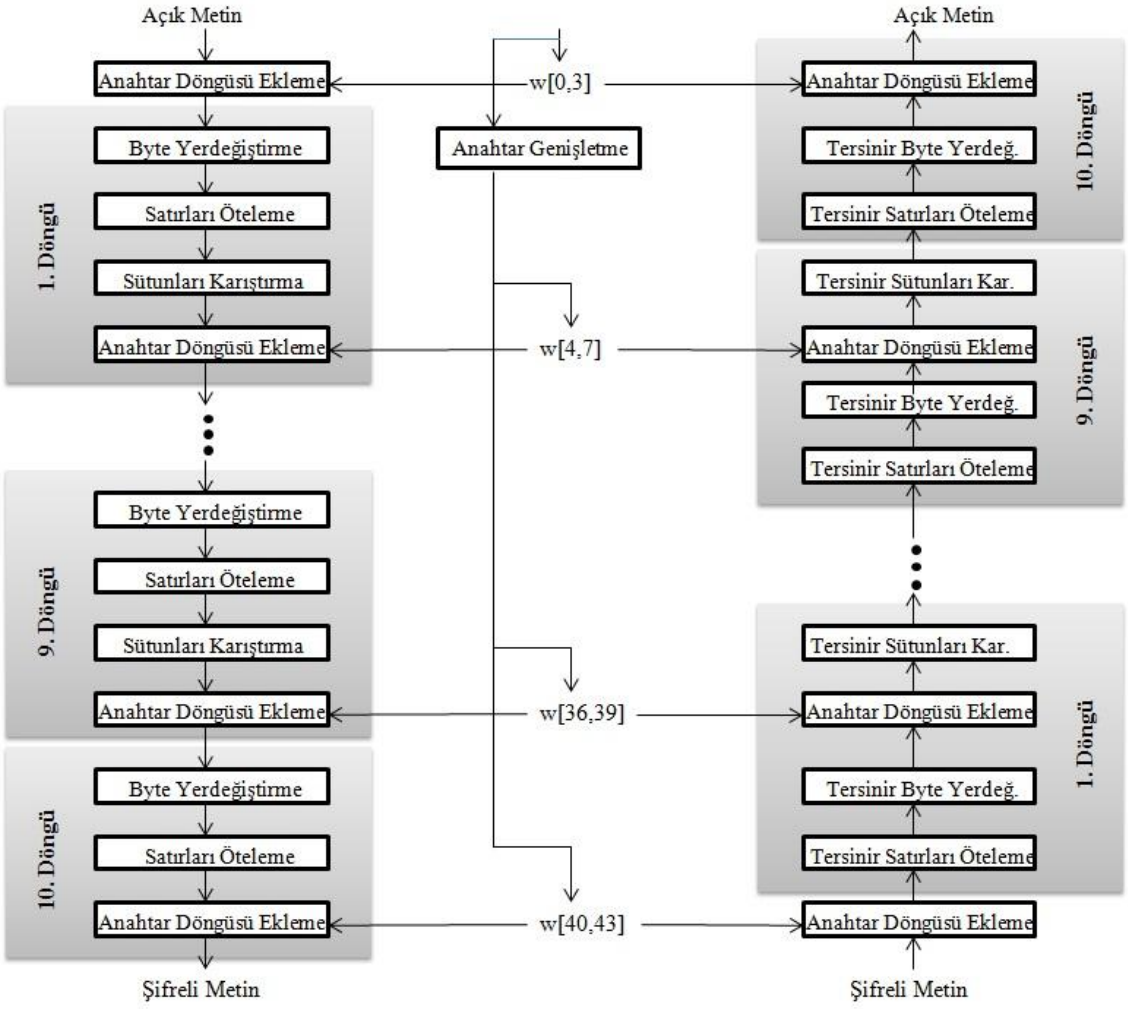
1. SubByte (Byte Yerdeğiştirme)
2. Shift Rows (Satırları Öteleme)
3. Mix Columns (Sütunları Karıştırma)

4. Add Round Key (Döngü Anahtarı Ekleme)

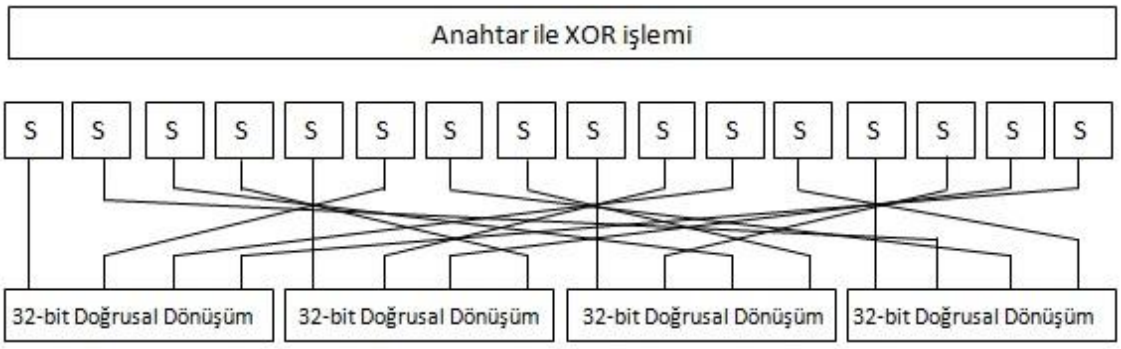
Onuncu döngü de sadece **Sütunları Karıştırma** aşaması bulunmamaktadır. Şifre çözme algoritmasının ilk dokuz döngüsünün her biri aşağıdaki aşamaları kapsamaktadır:

1. Inverse Shift Rows (Tersinir Satırları Öteleme)
2. Inverse SubByte (Tersinir Byte Yerdeğiştirme)
3. Inverse Add Round Key (Tersinir Döngü Anahtarı Ekleme)
4. Inverse Mix Columns (Tersinir Sütunları Karıştırma)

Şifreleme algoritması ile benzer şekilde şifre çözme algoritmasının onuncu döngüsünde **Tersinir Sütunları Karıştırma** aşaması bulunmamaktadır [58].



Şekil 4.1. AES Algoritmasının genel yapısı



Şekil 4.2. Bir döngülük AES genişletmesi [19]

Şekil 4.1.2'de AES şifresinin SPN yapısına uygun gösterimi verilmiştir.

4.1.1 SubByte (Byte Yerdeğiştirme) Dönüşümü

AES şifresinin işleyişinde Byte yer değiştirme işleminde her byte (8-bit) değeri farklı bir byte değeri ile yer değiştirir. Bu işlem S-kutusu (yer değiştirme kutusu) ile sağlanır ve bu sayede şifreye doğrusal olmama özelliğini kazandırılır. AES şifresinin S kutusu olası tüm 8-bit değerlere (0 değeri hariç, 0 değeri 8-bit 0 vektörünü temsil eder.) $GF(2^8) = Z_2(x)/(x^8 + x^4 + x^3 + x + 1)$ sonlu cisminde $x \to x^{-1}$ ters haritalama işleminin sonucu elde edilen 8-bit değerlerin çıkış bitlerine ($x_7, x_6, \ldots, x_0$) aşağıdaki şekilde verilen bitsel doğrusal dönüşüm uygulanarak elde edilmiştir.

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

Şekil 4.3. S-kutusu için Bitsel Doğrusal Dönüşüm

S-kutusu 16×16'lık bir byte matrisinden oluşmaktadır. Bir döngüde gerçekleşen S-kutusu işlemi aşağıdaki gibi gerçekleşmektedir:

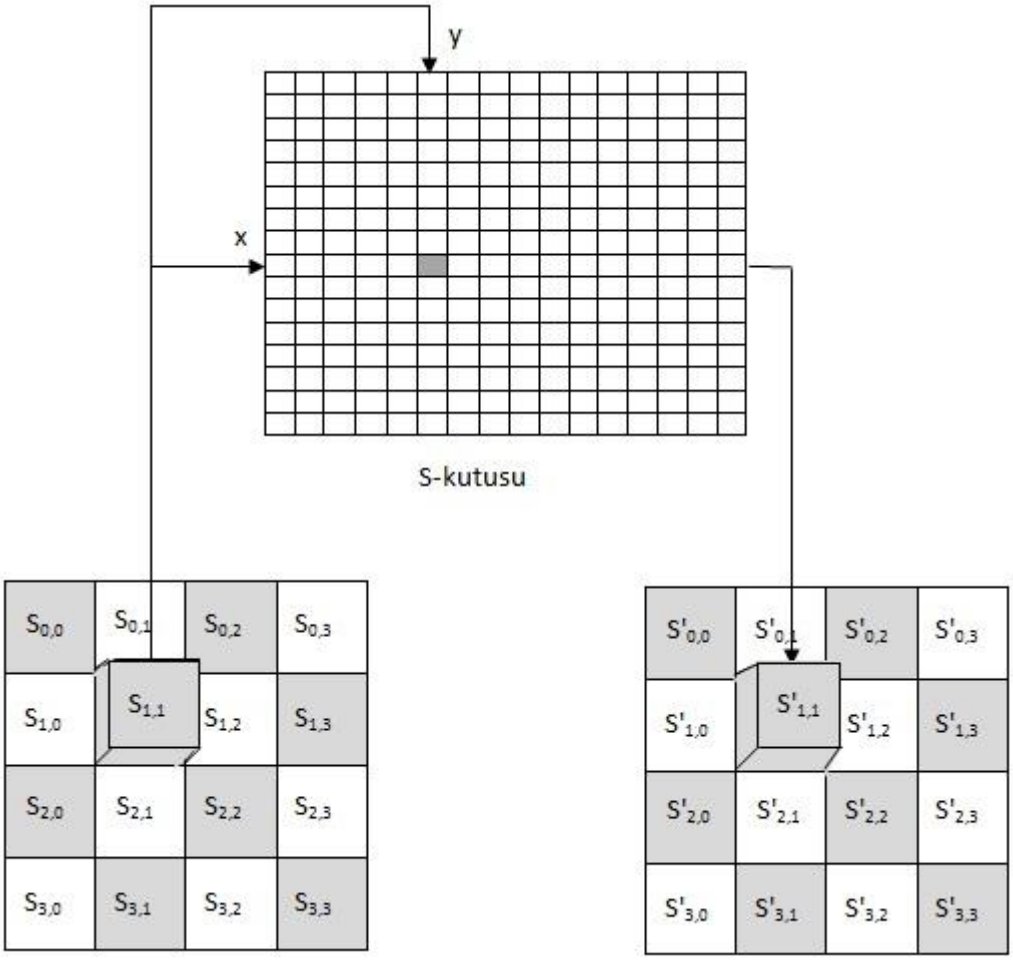
Her byte yeni bir byte ile şu şekilde eşleştirilir: byte'ın sol kısmındaki değer S-kutusunda satırı belirlemeye, byte'ın sağ kısmındaki değer ise S-kutusunda sütunu belirlemeye yarar. Örneğin, {95} (süslü parantezin içindeki değerler Hexadecimal değerlerdir) byte'ı S-kutusunun 9.satırında ve 5.sütununda bulunan değeri {2a} değerini getirir. Daha sonra Durum matrisi bu değer ile güncellenir. Bu işlem aşağıdaki şekil 4.1.5'de gösterilmiştir. Tersinir byte Yer Değiştirme işleminde tersinir S-kutusu kullanır bu sayede {2a} byte'ından {95} byte'ı elde edilir. Tablo 4.1.1, Tablo 4.1.2 ve Tablo 4.1.3'de sırasıyla AES blok şifresinde kullanılan S-kutusu, S-kutusunun tersi ve AES şifresindeki byte yer değiştirme aşaması gösterilmiştir.

Tablo 4.3. AES blok şifresinde 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	8e	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tablo 4.4. AES blok şifresinde kullanılan S-kutusunun tersi

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	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



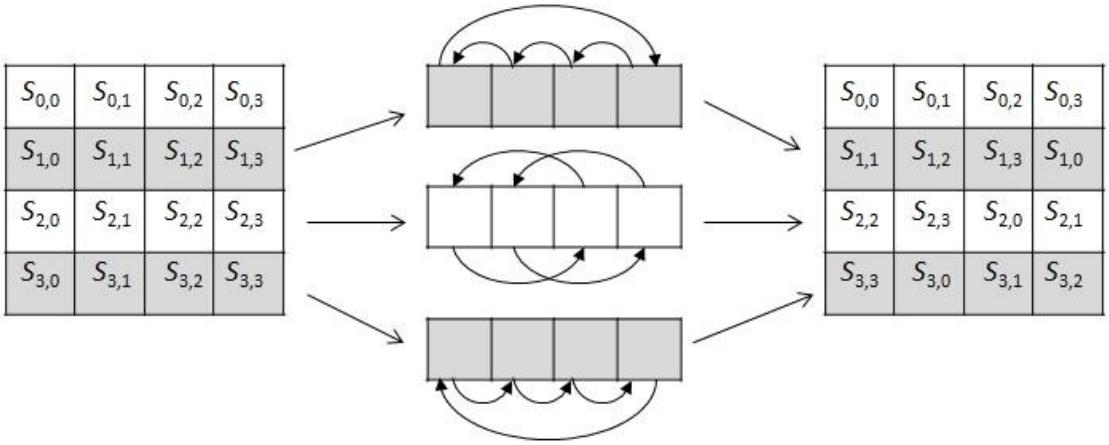
Şekil 4.4. AES algoritmasında bir byte Yer Değiştirme aşaması

S-kutusu bilinen kriptografik saldırılara karşı dirençli olması hedefiyle tasarlanmıştır. Dolayısıyla S-kutusu, kriptografik özellikleri (doğrusal olmama, doğrusal yaklaşım tablosunda ve fark dağılım tablosundaki en büyük değerin olabildiğince küçük olması vb.) bilinen en iyi özelliklere sahip olacak şekilde tasarlanmıştır. Ek olarak, S-kutusu sabit noktalara (S-kutusu(a)=a) sahip değildir [58].

4.1.2 ShiftRows (Satırları Öteleme) Dönüşümü

AES şifresinin döngü fonksiyonunda kullanılan diğer bir işlem o anki durum'un satırlarının ötelenmesi işlemidir. Bu işlem bir permütasyon işlemidir ve aşağıdaki gibi ifade edilebilir:

- Durum matrisin birinci satırında bir değişiklik olmaz.
- İkinci satırda ki byte'lar bir kez sol tarafa kaydırılır.
- Üçüncü satırdaki byte'lar iki kez sol tarafa kaydırılır.
- Dördüncü satırdaki byte'lar üç kez sol tarafa kaydırılır.



Şekil 4.5. ShiftRows dönüşümü

Bu dönüşüm ile Durum matrisinin bir sütununda bulunan byte'lar diğer 4 sütuna dağıtılırlar. Tersinir Satırları Öteleme dönüşümünde byte'lar tersi yönde yani sağa doğru kaydırılır [58].

4.1.3 MixColumns (Sütunları Karıştırma) Dönüşümü

Bu dönüşüm doğrusal ve diferansiyel kriptanalizi zorlaştırıcı etki yapma amacındadır ve sonlu cisimde çarpma tabanlıdır. Durum matrisindeki her sütun ayrı ayrı işlem görür. Bir sütundaki her byte içinde bulunduğu sütundaki dört byte'ı kullanan bir

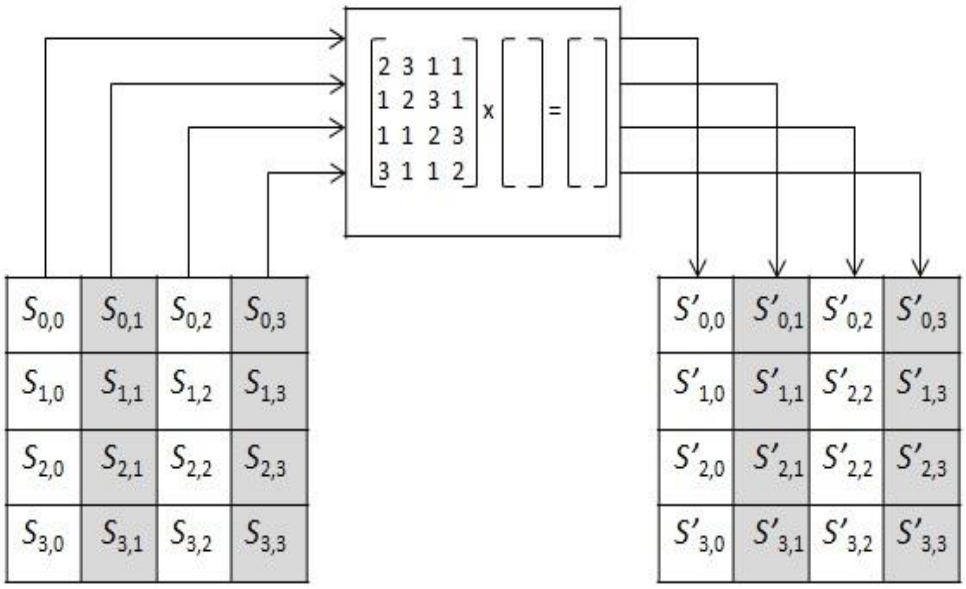
fonksiyon yardımı ile yeni bir değer ile eşleştirilir. Dönüşüm işlemi Durum ve aşağıdaki matrisin çarpımı ile belirlenir.

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix} \quad (7.1)$$

Ürün matrisindeki her eleman birinci matrisin satırı ile Durum matrisinin sütununun elemanlarının çarpımı ve XOR işlemi ile elde edilir. Bu işlemlerde toplama ve çarpma işlemlerinin her biri GF(2^8) sonlu cisminde gerçekleşir. Şekil 4.1.6'da sütunları karıştırma dönüşümünün bir temsili gösterilmektedir. Sütunları karıştırma Durum matrisinin sadece bir sütunu j ($0 \leq j \leq 3$) için aşağıdaki gibi gösterilebilir.

$$\begin{aligned} S'_{0,j} &= (2 \bullet S_{0,j}) \oplus (3 \bullet S_{1,j}) \oplus S_{2,j} \oplus S_{3,j} \\ S'_{1,j} &= S_{0,j} \oplus (2 \bullet S_{1,j}) \oplus (3 \bullet S_{2,j}) \oplus S_{3,j} \\ S'_{2,j} &= S_{0,j} \oplus S_{1,j} \oplus (2 \bullet S_{2,j}) \oplus (3 \bullet S_{3,j}) \\ S'_{3,j} &= (3 \bullet S_{0,j}) \oplus S_{1,j} \oplus S_{2,j} \oplus (2 \bullet S_{3,j}) \end{aligned} \quad (7.2)$$

• sembolü GF(2^8)'de çarpma işlemini göstermektedir.



Şekil 4.6. Sütunları Karıştırma Dönüşümü

Sütunları karıştırma dönüşümünün çalışmasını bir örnek ile açıklayacak olursak; Durum matrisinin birinci sütununun ilk elemanları $S_{0,0}$={87}, $S_{1,0}$={6E}, $S_{2,0}$={46}, $S_{3,0}$={A6} olsun. Bu örnekteki değerler ile sütunları karıştırma işleminden sonra $S_{0,0}$={87} değeri $S'_{0,0}$={47} değeri ile eşleşir. (7.2)'de ki denklemin ilk satırı ile aşağıdaki işlemler yapılır [58].

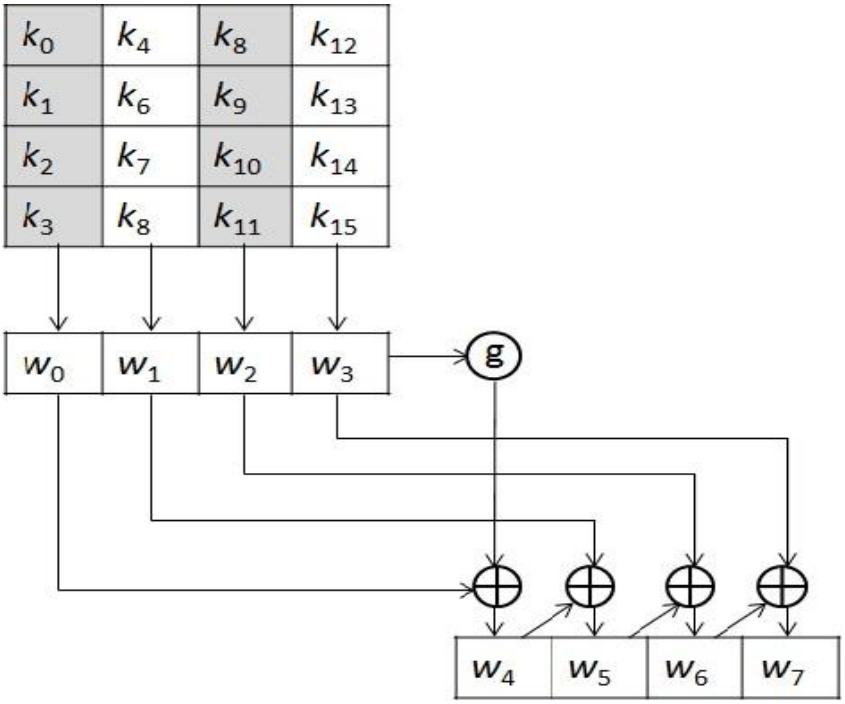
$$(02 \otimes 87) \oplus (03 \otimes 6E) \oplus 46 \oplus A6 = 47$$

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.

Yayılım özelliği doğrusal ve diferansiyel saldırılar düşünüldüğünde diferansiyel ve doğrusal yaklaşımların elde edilmesi ve saldırıların gerçeklenmesini zorlaştırıcı etkisi açısından önemli kavramlardır. Bir yayılım yapısının çığ etkisi özelliğini karakterize etmenin bir yolu da dallanma sayısı olarak adlandırılan Daemen tarafından Ph.D tezinde ortaya atılan bir kavramdır [19].

4.1.4 AES Anahtar Genişletmesi

AES Anahtar Genişletme algoritmasının 128-bit alt anahtar üreten versiyonu giriş verisi olarak 4 kelime alır ve çıkış verisi olarak 44 kelime oluşturur. Her kelime 32 byte'tan oluşur bu nedenle her alt anahtar (4 kelimeden oluşur) 128 bit uzunluğundadır. Şifre anahtarı; genişletilmiş anahtarın ilk dört kelimesine aktarılır. Genişletilmiş anahtarın geri kalan kısmı her seferinde 4 kelime doldurulacak şekilde devam eder ve bu şekilde toplamda AES-128 için 40 kelime, AES-192 için 48 kelime ve AES-256 için 56 kelime oluşturulur. Eklenen her yeni kelime **w**[*i*], kendinden bir önceki kelime **w**[*i*-1] ve dört önceki kelimeye **w**[*i*-4] bağlı olarak oluşturulur. Oluşturulan her dört kelimeden üç tanesi için basit bir XOR işlemi kullanılır. Eğer bir kelime **w** dizisinde 4'ün katı ise bu kelimenin oluşturulmasında karmaşık bir fonksiyon kullanılır. Aşağıdaki şekil 4.8'de ilk sekiz kelimenin oluşumu gösterilmiştir ve 4'ün katı olan kelimenin oluşturulması için kullanılan yukarıda bahsedilen karmaşık fonksiyon *g* ile gösterilmiştir.



Şekil 4.7. AES Anahtar Genişletme

g fonksiyonu aşağıdaki alt fonksiyonları içermektedir.

1. **RotWord** (Kelimeleri Döndürme): Kelime üzerindeki byte'lar dairesel olarak bir sola kaydırır. Bunun anlamı [b_0, b_1, b_2, b_3] kelimesi bu işlem sonrası [b_1, b_2, b_3, b_0] şeklini alır.

2. **SubWord** (Kelimeleri Yer Değiştirme): Daha önceden bahsettiğimiz S-kutusu kullanılarak her byte yeni bir byte ile yer değiştirilir.

3. Birinci ve ikinci işlemlerde elde edilen değerler döngü sabiti Rcon[*j*] ile XOR işlemine tabi tutulur.

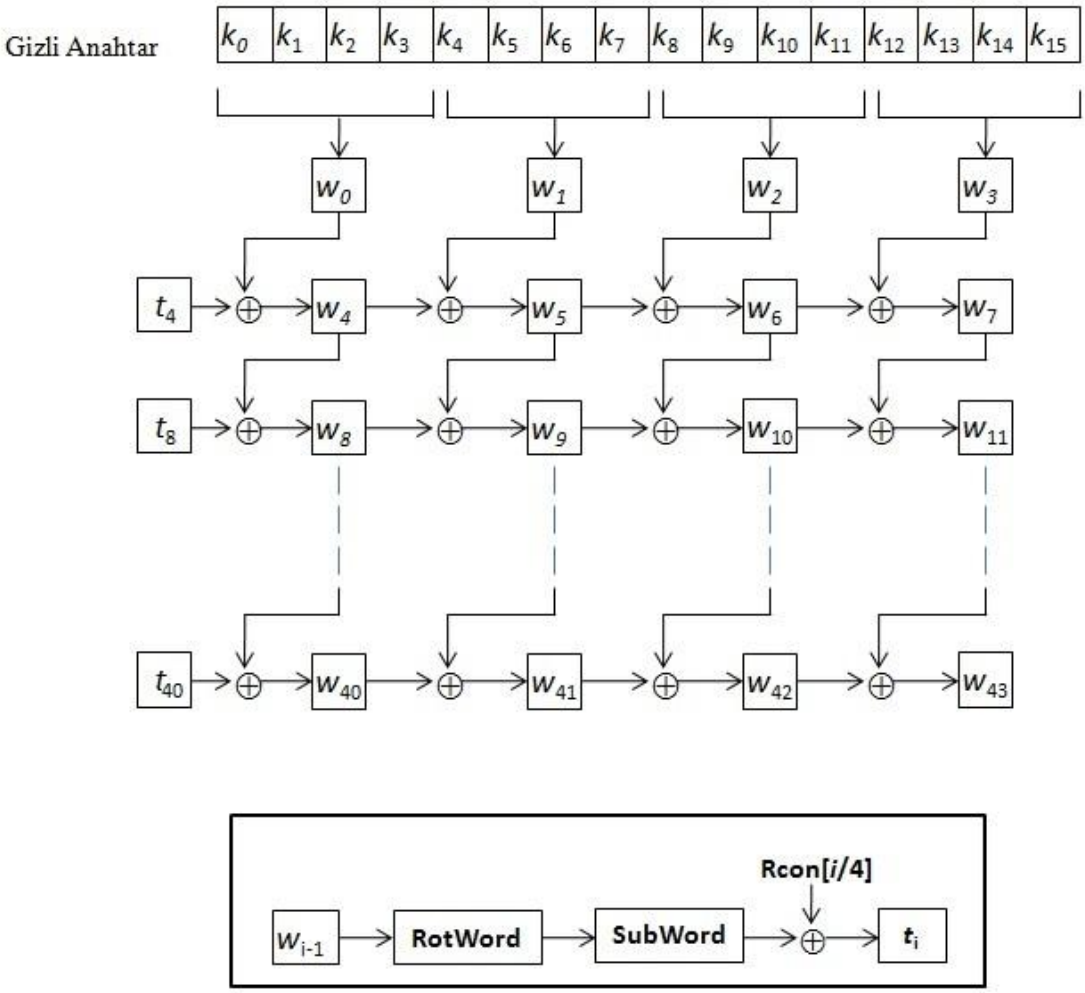
Döngü sabiti bir kelimedir ve her zaman en sağdaki 3 byte değeri 0'a eşittir. Bu nedenle Rcon ile XOR işlemine tabi tutulan bir kelimenin sadece en soldaki byte'ları etkilenir. Her döngüdeki döngü sabiti farklıdır ve şu şekilde tanımlanır Rcon[*j*] = (RC[*J*],0,0,0) yani RC[1] = 1, RC[*j*] = 2 RC[*j*-1]. Çarpma işlemi GF(2^8) alanında tanımlanmıştır. Aşağıdaki tablo 4.1.4'de döngü sabitleri gösterilmiştir.

Tablo 4.5. Döngü Sabitleri (RCON değerleri)

Döngü	Sabit Değer (RCON)	Döngü	Sabit Değer (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 bilinen kriptanalitik saldırılara karşı dirençli olması amacıyla tasarlanmıştır. Döngü sabitlerinin döngülere bağlı olması farklı döngülerde üretilen döngü anahtarları arasındaki simetriyi veya benzerliği elimine eder. Aşağıdaki sekil 4.8'de AES anahtar genişletme döngülerinin bir özeti gösterilmiştir.

AES-128 anahtar genişletme algoritmasının genel yapısı aşağıdaki Şekil 4.1.9'de gösterilmiştir. Şifrenin diğer iki versiyonu içinde bazı küçük değişiklikler ile birlikte aynıdır. Aşağıdaki şekilde orijinal anahtardan 44 kelimenin nasıl yaratıldığı gösterilmektedir. Burada ilk dizi yani k_0, k_1 , , , , k_{15} byte seti gizli anahtardır ve gizli anahtardan yeni anahtarlar üretilmektedir. Bu örnek 128-bit uzunluğundaki gizli anahtar için yapılan işlemleri göstermektedir. Anahtar uzunluğu arttıkça üretilen anahtar sayısı da artacaktır.



Şekil 4.8. AES-128 bit Anahtar Genişletme

4.2 ARIA

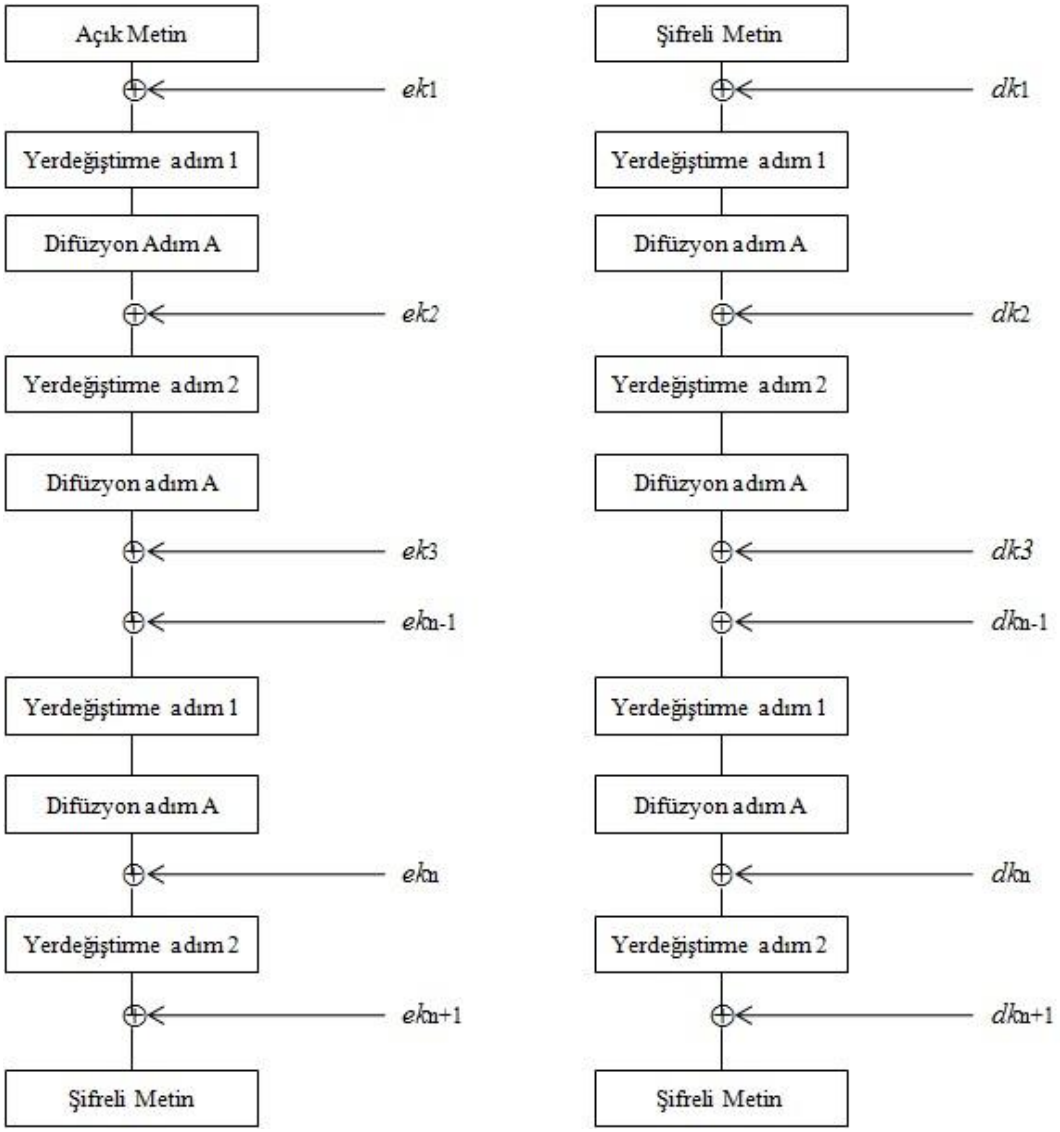
ARIA bir SPN blok şifre çeşididir. 128 bit bloklarını; 128,192 veya 256 bit anahtar uzunluklarına bağlı olarak sırasıyla 12, 14 ve 16 döngüde işler. ARIA algoritması *durum* olarak adlandırılan 128-bit uzunluğundaki diziye uygulanan bir takım işlemler olarak düşünülebilir. Durum'a, açık metin giriş verisi olarak verilmektedir ve her döngüdeki her işlem sonrasında durumun içeriğindeki veriler değişmektedir. Durum'un içeriğindeki verilerin son değerleri ARIA algoritmasının çıkışıdır. ARIA algoritmasının birçok işlemi byte tabanlıdır bu nedenle durum bazen 16 byte içeren bir dizi olarak düşünülebilir. Şifredeki her döngü aşağıdaki 3 kısmı içerir.

1. **Round Key Addition** (Döngü Anahtarı Ekleme): Durum 128 bit uzunluğundaki döngü anahtarı ile XOR işlemine tabi tutulur.

2. **Substitution Layer** (Yer Değiştirme Katmanı): Durum 16 kez S-kutusu ile işleme tabi tutulur. İki çeşit ARIA yer değiştirme katmanı vardır, Tür 1 - Tür 2 ve döngüler arasında değiştirilirler.

3. **Diffusion Layer** (Yayılım Katmanı): 16×16'lık bir matris ile durum çarpılır.

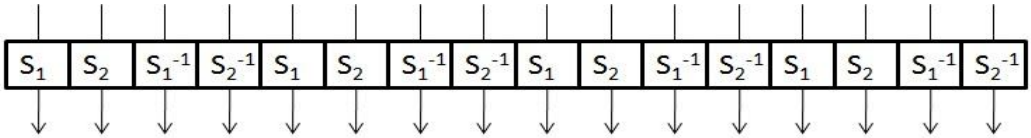
Bunlarla beraber anahtar genişletme işlemi de vardır, verilen gizli anahtarın 128,192 ve 256 bit uzunluğunda olmasına bağlı olarak sırasıyla 13,15 ve 17 döngü anahtarı oluşturulmaktadır. Aşağıdaki Şekil 4.2.1'de n = 12, 14 veya 16 için ARIA şifreleme ve şifre çözme işlemleri gösterilmiştir.



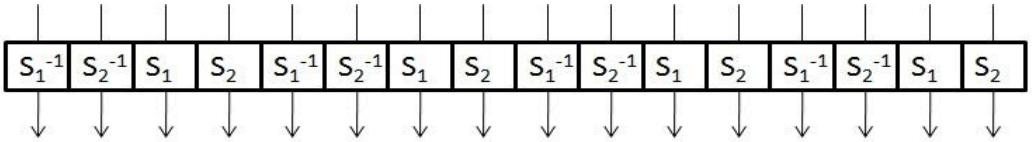
Şekil 4.9. ARIA algoritması şifreleme ve şifre çözme işlemi

4.2.1 Yer Değiştirme Adımı

ARIA Yer değiştirme adımında 2 çeşit S-kutusu S_1, S_2 ve bu S-kutularının tersleri olan S_1^{-1}, S_2^{-1} S-kutuları kullanılmaktadır. Bu S-kutuları Tablo 4.2.1, Tablo 4.2.2, Tablo 4.2.3 ve Tablo 4.2.4 de gösterilmiştir. Buna ek olarak aşağıdaki Şekil 4.2.2 ve Şekil 4.2.3'de gösterilen iki çeşit yer değiştirme katmanı kullanır. Bunlardan birincisi tek numaralı döngülerde ikincisi ise çift numaralı döngülerde kullanılır.



Şekil 4.10. Tekli döngülerde kullanılan S-kutuları



Şekil 4.11. Çiftli döngülerde kullanılan S-kutuları

Tablo 4.6. ARIA algoritmasında kullanılan 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	8e	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Tablo 4.7. ARIA algoritmasında kullanılan 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	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	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 4.8. ARIA algoritmasında kullanılan S_2 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 4.9. ARIA algoritmasında kullanılan 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

4.2.2 Difüzyon adımı

ARIA difüzyon katmanı 16 byte'lık giriş verisini (x_0,x_1,...,x_{15}) alıp 16 byte'lık çıkış verisine (y_0,y_1,...,y_{15}) dönüştüren involutif (tersi kendisine eşit) bir fonksiyondur. Aşağıdaki gösterilmiştir. Bu doğrusal dönüşümün dallanma sayısı 8 ve sabit nokta sayısı 2^{72}'dir [59]. Not edilmelidir ki doğrusal dönüşümün sahip olduğu sabit nokta sayısı 128-bit blok işlediği gerçeğine dayanarak verilmiştir. Aşağıda verilen denklemler de x_i byte giriş değerlerini ve y_i byte çıkış değerlerini temsil etmektedir.

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

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

$$y_2 = x_1 \oplus x_4 \oplus x_6 \oplus x_{10} \oplus x_{11} \oplus x_{12} \oplus x_{15}, \qquad y_{10} = x_2 \oplus x_5 \oplus x_6 \oplus x_8 \oplus x_{13} \oplus x_{15}, \oplus$$

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

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

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

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

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

Bu dönüşüm işlemi aşağıda verilen ikili matris şeklinde de gösterilir.

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

4.2.3 ARIA Anahtar Genişletme

ARIA blok şifresinin anahtar genişletme algoritması iki aşamadan oluşur: başlangıç ve döngü anahtarları üretilmesi aşamaları.

Başlangıç aşaması:

MK'dan (Gizli anahtar-Gizli anahtar) 3-döngülü 256-bit Feistel şifre kullanılarak 128 bit uzunluğunda W_0, W_1, W_2, W_3 olmak üzere 4 anahtar üretilir. Burada gizli anahtar 128, 192 ve 256 bit uzunluğunda olabilir. Gizli anahtardan öncelikle 128-bit uzunluğundaki *KL* üretilir ve eğer gizli anahtar 128-bit'den uzun ise kalan bitlerle de 128-bit uzunluğundaki *KR* üretilir eğer kalan bitler 128 bit uzunluğunda değilse eksik kısımlar 0 bitleri ile tamamlanır. Bu işlem aşağıdaki gibi gösterilebilir:

$$KL||KR = MK||0 \cdot \cdot \cdot 0.$$

Daha sonraki adımlarda alt anahtarlar elde edilir:

$$W_0 = KL, \qquad W_2 = Fe(W_1, CK_2) \oplus W_0,$$

$$W_1 = F_0(W_0, CK_1) \oplus KR, \qquad W_3 = F_0(W_2, CK_3) \oplus W_1.$$

Yukarıda, F_0 ve Fe sırasıyla tek ve çift fonksiyonlardır. CK_i ise F_0 ve Fe fonksiyonları için kullanılan sabit değerlerdir. Sabit değerler şu şekilde bulunur. Öncelikle; 1/pi 'nin ilk 128x3 bitlik kısmı üç tane 128-bit uzunluğundaki C_i sabit değerlerine atanır.

C1 = 0x517cc1b727220a94fe12abe8fa9a6ee0

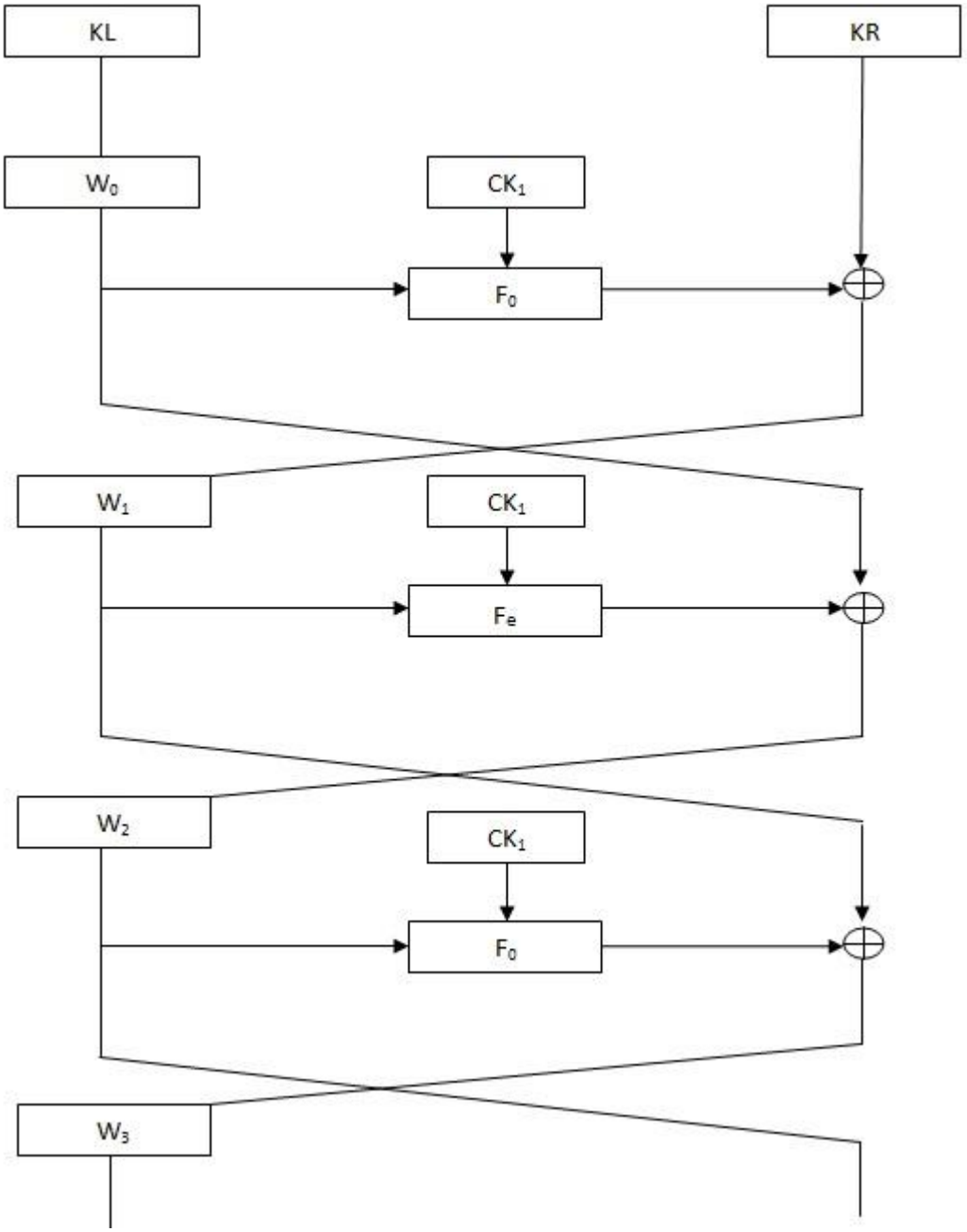
C2 = 0x6db14acc9e21c820ff28b1d5ef5de2b0

C3 = 0xdb92371d2126e970324977504e8c90e0

Daha sonra sabit değerler aşağıdaki tablo ile tanımlanır;

Tablo 4.10. Gizli anahtar uzunluğuna bağlı sabit değerler

Key Size (Anh. Uzn.)	CK_1	CK_2	CK_3
128	C_1	C_2	C_3
192	C_2	C_3	C_1
256	C_3	C_1	C_2



Şekil 4.12. ARIA W_i değerlerinin elde edilmesi

Döngü Anahtarları Üretilmesi Aşaması:

Döngü anahtarları üretilmesi aşamasında, *W*i'nin dört değeri çeşitli şekilde birleştirilerek şifreleme (ek_i) ve şifre çözme (dk_i) anahtarları üretilir.
Şifreleme anahtarları aşağıdaki şekilde hesaplanır;

$$ek_1 = (W_0) \oplus (W_1 >>>^{19}), \qquad ek_2 = (W_1) \oplus (W_2 >>>^{19}),$$

$$ek_3 = (W_2) \oplus (W_3 >>>^{19}), \qquad ek_4 = (W_0 >>>^{19}) \oplus (W_3),$$

$$ek_5 = (W_0) \oplus (W_1 >>>^{31}), \qquad ek_6 = (W_1) \oplus (W_2 >>>^{31}),$$

$$ek_7 = (W_2) \oplus (W_3 >>>^{31}), \qquad ek_8 = (W_0 >>>^{31}) \oplus (W_3),$$

$$ek_9 = (W_0) \oplus (W_1 <<<^{61}), \qquad ek_{10} = (W_1) \oplus (W_2 <<<^{61}),$$

$$ek_{11} = (W_2) \oplus (W_3 <<<^{61}), \quad ek_{12} = (W_0 <<<^{61}) \oplus (W_3),$$

$$ek_{13} = (W_0) \oplus (W_1 <<<^{31}), \quad ek_{14} = (W_1) \oplus (W_2 <<<^{31}),$$

$$ek_{15} = (W_2) \oplus (W_3 <<<^{31}), \quad ek_{16} = (W_0 <<<^{31}) \oplus (W_3),$$

$$ek_{17} = (W_0) \oplus (W_1 <<<^{19})$$

BÖLÜM 5

AES ANAHTAR GENİŞLETMESİ İÇİN YENİ GELİŞTİRİLEN ANAHTAR GENİŞLETME ALGORİTMASI

Bölüm 3'te gösterildiği gibi AES anahtar genişletme algoritması yavaş yayılım ve bit sızıntısı problemlerine sahiptir. ARIA blok şifresinde kullanılan anahtar genişletme algoritmasının yayılım özellikleri iyi olmakla beraber bit sızıntı problemi bulunmaktadır. Bu bölümde AES blok şifresinin kullandığı anahtar genişletme algoritmasının geliştirilmesi ile bahsedilen problemleri gideren bir anahtar planlama algoritması ortaya konmuştur. Bu anahtar genişletme algoritmasın 128-bit ve 256-bit alt anahtarlar üretebilmektedir. Farklı boyutlarda alt anahtarlar üretecek şekilde geliştirilmesi mümkündür. Yeni geliştirilen anahtar genişletme algoritmasında yayılımın arttırılması için iki yapı S-D-S (Substitution-Diffusion-Substitution) ve D-S-D (Diffusion-Substitution-Diffusion) kullanılmıştır.

5.1 Yeni Geliştirilen Anahtar Genişletme Algoritmasının Özellikleri

Bölüm 3'te ifade edildiği üzere AES Anahtar Genişletme Algoritmasının yavaş yayılım ve bit sızıntısı problemleri bulunmaktadır. Bu problemlerden yavaş yayılım probleminin giderilmesi için öncelikle geçici t_i değerleri elde edilirken bir yayılım elemanına daha ihtiyaç duyulduğu gözlenmektedir. Dolayısıyla bu eksik yayılım elemanı AES şifresinin döngü fonksiyonunda kullanılan MixColumns (Sütunları Karıştırma) dönüşümünün kullanılması ile giderilebilir. Diğer yandan kullanılacak

MixColumns dönüşümü yüksek maliyete sahip olacağından çalışmamızda düşük maliyetli ve maksimum dallanma sayısına sahip ikili matrisler kullanılmıştır. Şekil 5.1'de gösterilen kendi tasarımımız olan Anahtar genişletme algoritmasında, geçici t_i değerlerinin elde edilmesi için kullanılan yeni yapı yavaş yayılım problemini gidermeye yöneliktir.

Şekil 5.1'de gösterilen algoritmamızı tasarlarken sonraki kısımlarda anlatacağımız bir kaç farklı tasarım daha geliştirilmiş ve bu tasarımlar üzerinde yapılan testlerin sonuçları incelenerek Şekil 5.1'de ki tasarımımızın yeni anahtar genişletme tasarımımız olunmasına karar verilmiştir.

Aşağıdaki şekil 5.1'de gösterildiği gibi geliştirilen ikinci anahtar genişletme algoritması şifre anahtarını (k_0 dan k_{15}'e kadar 16 byte bir dizi olarak düşünülebilir) önce bir RCON değeri ile XOR işlemine tabi tutar ve bunun sonucunda elde edilen çıkış verisini her biri 32 bit uzunluğunda ve 4 byte'tan oluşan 4 kelimeye (w_0, w_1, w_2, w_3) aktarır. Diğer kelimeler (w_i ; i = 4'ten i = 19'a kadar) şu şekilde oluşturulur.

a- Eğer i (mod 4) $\neq$ 0 ise Şekil 5.1 den görüldüğü gibi kelimenin solundaki (w_{i-1}) ve üstündeki (w_{i-4}) kelimelerin XOR işlemi sonucunda oluşan değerden elde edilir.

b- Eğer i (mod 4) = 0 ise aşağıdaki denklemden geçici t_i değeri elde edilir, t_i değeri SubWord-MixWord-SubWord ya da Substitution-Diffusion-Substitution (S-D-S) işlemlerinin sırasıyla w_{i-1} kelimesinin üzerinde uygulanması sonucu elde edilir. Bu işlemler sonucunda elde edilen t_i değerinin w_{i-4} kelimesi ile XOR işlemine girilmesi sonucunda elde edilen yeni değer ile w_i kelimesi elde edilir.

$$t_i = SubWord(MixWord(SubWord(w_{i-1})))$$

Bu işlemler sonucunda elde edilen (w_{16}, w_{17}, w_{18}, w_{19}) kelimelerinin birleşimi ile oluşan 128 bit uzunluğunda 16 byte'tan oluşan çıkış verisi, RCON değerinin tersi ile XOR işlemine tabi tutulur. Bu işlem sonucunda oluşan çıkış verisinin şifre anahtarı ile XOR işlemine tabi tutulması sonucunda 1.alt anahtar elde edilmektedir. Benzer işlemlerin farklı RCON değerleri ile gizli anahtara uygulanması ile birbirinden bağımsız farklı alt anahtar elde edilmektedir. Örneğin farklı 10 RCON değeri ile 10 farklı alt anahtar elde edilebilir. Uygulamalarda kullanılan RCON ve RCON tersi değerlerinin her uygulamada farklı olması sayesinde üretilen 10 alt anahtarın farklı olması sağlanır. Alt anahtarların birbirinden bağımsız olması, AES anahtar genişletme algoritmasındaki

her hangi bir alt anahtar bilgisinden diğer alt anahtarların elde edilmesi zafiyetini ortadan kaldırmaya yöneliktir.

Anahtar genişletme algoritmaları ile elde edilen alt anahtarların üzerinde yürütülen iki önemli test, frekans testi ve çığ kriteri testidir. Frekans testi, bit karıştırma özelliğinin ölçülmesinde (Shannon'nın karıştırma özelliğinin ölçülmesinde temel teşkil eder) kullanılırken çığ kriteri testi, bit yayılım özelliğinin ölçülmesinde kullanılır. Bu test, giriş bloğunda bir bit değişimin çıkış bloğundaki bitlerin yarısının değişimini kontrol eder (Shannon'nın yayılım özelliğinin ölçümünü sağlar). Geliştirilen yeni anahtar genişletme algoritması ile elde edilen alt anahtarların (128-bit şifre anahtarı için) bit değişimi 64-bit'e yakın olarak Şekil 5.1'te gösterilen tasarım ile elde edilmiştir.

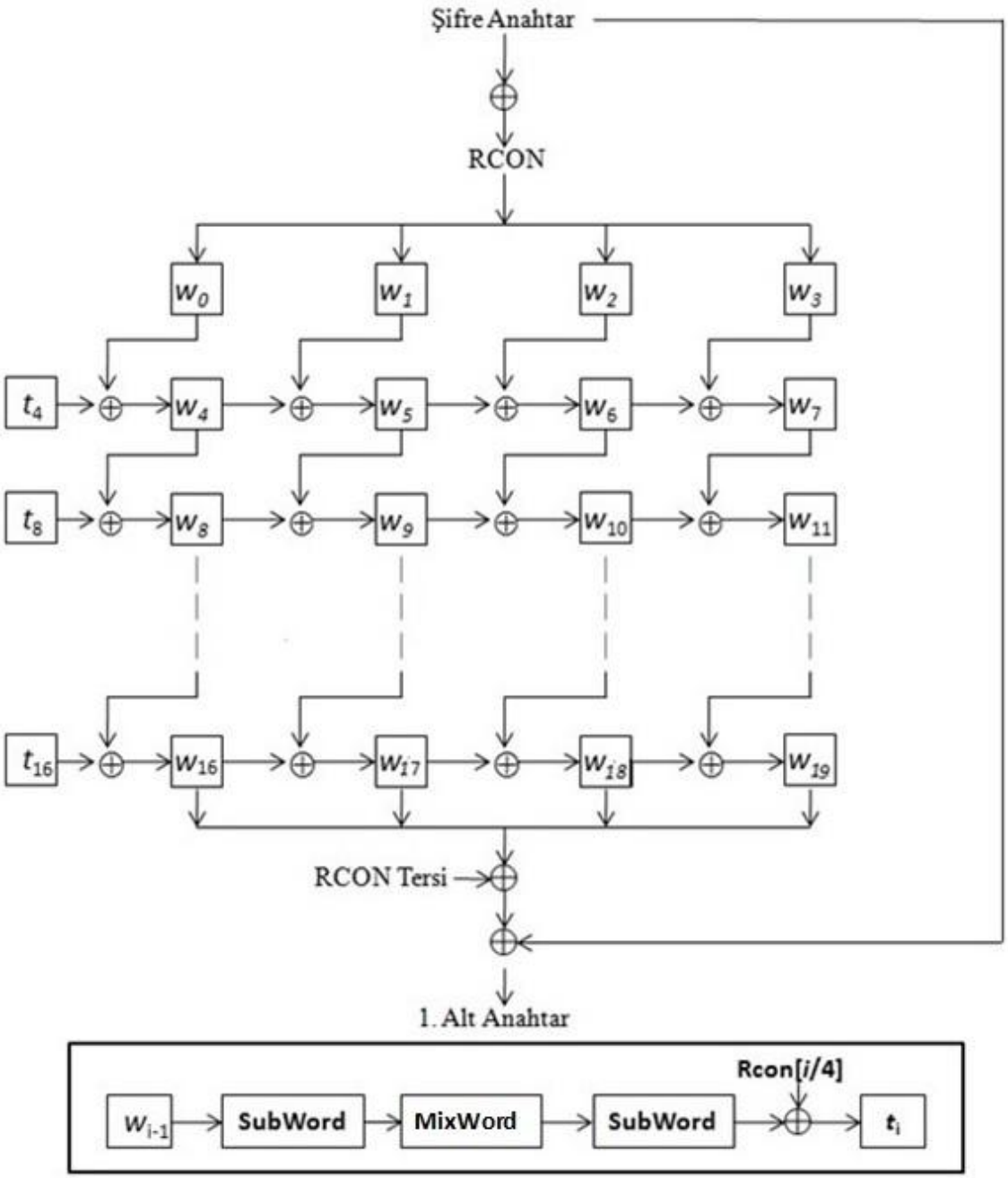
128 bit D1 matrisi	128 bit D2 matrisi
0 1 1 1	1 1 0 1
1 1 0 1	1 0 1 1
1 1 1 0	1 1 1 0
1 0 1 1	0 1 1 1

128-bit yeni anahtar genişletme algoritmasında kullanılan ve yukarıda gösterilen D1 ve D2 ikili matrisleri; dallanma sayısı maksimum ve içerdiği sabit noktalar minimum olacak şekilde tüm 4×4 matris uzayı (2^{16}) taranarak elde edilmiştir. Not edilmelidir ki verilen ikili matrislerin dallanma sayıları 4 ve içerdikleri sabit nokta sayısı 2'dir.

Birinci tasarımımız olan Şekil 5.1.'de ki tasarımımız ile ana anahtardan alt anahtarın üretilmesi aşamasında adım adım elde edilen değerler Tablo 5.1'de gösterilmiştir. Şifre anahtarı ve RCON değeri rastgele seçilmiştir.

256 bit D1 matrisi	256 bit D2 matrisi
10010111	01001101
11011100	01101011
01101110	00110101
00111111	10011010
01110011	11010100
11001010	10110110
11101101	01010011
11110110	10101001

256-bit yeni anahtar genişletme algoritmasında kullanılan ve yukarıda gösterilen D1 ve D2 ikili matrisleri; dallanma sayısı 5 ve sabit nokta sayısı 1 olacak şekilde [60] maksimum kriptografik özelliklere sahip olacak şekilde elde edilmiştir. Verilen ikili matris, elemanları GF(2^4) ($x^4 + x + 1$ indirgenemez polinomu ile tanımlı) cismine ait $\begin{bmatrix} 3_h & E_h \\ E_h & 6_h \end{bmatrix}$ 2×2 boyutundaki matristen elde edilmiştir. Elde etme işlemi sırasında 2×2 boyutundaki GF(2^4) cisim elemanlarının yerine bu cisim elemanlarına karşılık gelen 4×4 boyutundaki ikili matrislerin yer değiştirilmesi işlemi kullanılmıştır.



Şekil 5.1. AES-128 bit anahtar genişletme tasarımı S-D_1-S

Tablo 5.1'de ise AES-128 için önerilen anahtar genişletme algoritması (S-D_1-S yapısı) ile gizli bir anahtardan bir alt anahtarın elde edilmesi gösterilmektedir.

Tablo 5.1. 128 bit S-D_1-S tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi işleminin adım adım sonuçları

Şifre Anahtarı	13 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23			
RCON	3D CE 92 37 00 12 45 29 6A D2 73 54 01 73 FB AC			
RCON çıkışı	2E 8B 30 96 23 23 E1 8A D8 1E D9 60 C3 C8 8C 8F			
Kelimeler(w_0, w_1, w_2, w_3)	2E 8B 30 96	23 23 E1 8A	D8 1E D9 60	C3 C8 8C 8F
W[3] S kutusu Çıkışı	2E E8 64 73			
W[3] Difüzyon Çıkışı	FF B5 A2 39			
W[3] S kutusu Çıkışı	16 D5 3A 12			
t4 değişkeni	16 D5 3A 12			
Kelimeler(w_4, w_5, w_6, w_7)	38 5E 0A 84	1B 7D EB 0E	C3 63 32 6E	00 AB BE E1
W[7] S kutusu Çıkışı	63 62 AE F8			
W[7] Difüzyon Çıkışı	34 F9 AF 35			
W[7] S kutusu Çıkışı	18 99 79 96			
t8 değişkeni	18 99 79 96			
Kelimeler(w_8, w_9, w_{10}, w_{11})	20 C7 73 12	3B BA 98 1C	F8 D9 AA 72	F8 72 14 93

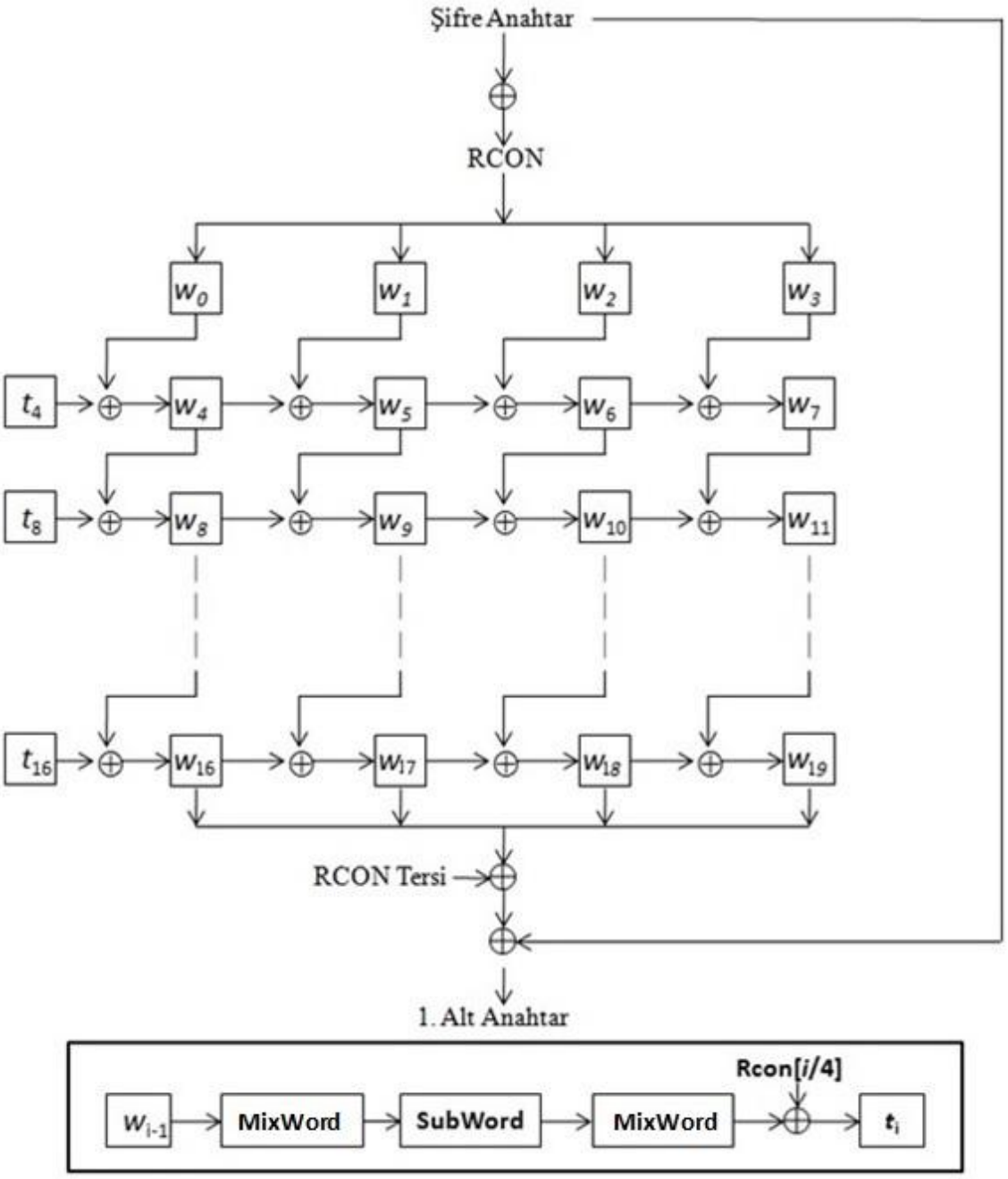
W[11] S kutusu Çıkışı	41 40 FA DC			
W[11] Difüzyon Çıkışı	66 DD FB 67			
W[11] S kutusu Çıkışı	33 C1 0F 85			
t12 değişkeni	33 C1 0F 85			
Kelimeler(w_{12}, w_{13}, w_{14}, w_{15})	13 06 7C 97	28 BC E4 8B	D0 65 4E F9	28 17 5A 6A
W[15] S kutusu Çıkışı	34 F0 BE 02			
W[15] Difüzyon Çıkışı	4C C6 7A 88			
W[15] S kutusu Çıkışı	29 B4 DA C4			
t16 değişkeni	29 B4 DA C4			
Kelimeler(w_{16}, w_{17}, w_{18}, w_{19})	3A B2 A6 53	12 0E 42 D8	C2 6B 0C 21	EA 7C 56 4B
RCON Tersi	AC FB 73 01 54 73 D2 6A 29 45 12 00 37 92 CE 3D			
RCON Tersi XOR çıkışı	96 49 D5 52 46 7D 90 B2 EB 2E 1E 21 DD EE 98 76			
Şifre anahtarı	12 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23			
Şifre Anahtarı XOR çıkışı	B8 C2 E5 C4 65 5E 71 38 33 30 C7 41 1E 26 14 F9			
1.Alt Anahtar	B8 C2 E5 C4 65 5E 71 38 33 30 C7 41 1E 26 14 F9			

Tablo 5.2'de ise AES-128 için önerilen anahtar genişletme algoritmasının S-D_1-S yapısı ile 20 gizli anahtardan üretilmiş alt anahtarların 128 farklı bit pozisyonu için ortalama bit değişimleri gösterilmiştir.

Tablo 5.2. AES-128 için önerilen anahtar genişletme algoritmalarından S-D_1-S yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 128 farklı bit pozisyonu için ortalama bit değişimleri

Gizli Anah.	Döngü Anah.1	Döngü Anah.2	Döngü Anah.3	Döngü Anah.4	Döngü Anah.5	Döngü Anah.6	Döngü Anah.7	Döngü Anah.8	Döngü Anah.9	Döngü Anah.10
Anah.1	63.45	63.64	63.74	64.20	64.13	63.10	63.60	64.57	63.07	64.08
Anah.2	64.39	64.34	63.07	64.10	63.90	63.68	63.50	63.61	63.43	63.75
Anah.3	63.21	64.44	63.39	63.49	63.44	63.96	64.68	63.72	64.74	63.74
Anah.4	64.34	63.60	63.46	64.46	64.10	64.09	63.75	63.67	64.63	63.77
Anah.5	64.21	64.13	63.90	63.78	64.01	63.52	63.65	63.88	63.97	64.06
Anah.6	64.48	64.02	64.17	63.79	64.35	63.42	63.80	64.29	62.85	63.96
Anah.7	63.57	63.53	64.42	64.46	64.41	62.84	63.70	64.03	63.88	63.93
Anah.8	63.30	63.43	63.64	64.04	63.47	63.50	64.67	63.44	63.69	64.26
Anah.9	63.82	63.80	63.72	63.93	63.73	64.0	63.17	64.02	64.25	64.07
Anah.10	64.14	63.21	64.15	63.68	64.0	64.12	63.64	63.89	63.50	63.62
Anah.11	64.09	64.47	63.22	64.23	62.97	64.35	64.19	63.60	64.04	65.11
Anah.12	64.46	64.35	63.07	63.01	64.19	64.70	64.89	63.34	63.40	63.67

Anah.13	64.83	63.89	63.89	63.71	64.66	64.60	63.92	64.32	63.97	64.69
Anah.14	65.03	64.56	63.61	64.55	63.45	63.62	63.99	63.78	63.61	64.29
Anah.15	64.46	64.35	63.07	63.01	64.19	64.70	64.89	63.34	63.40	63.67
Anah.16	63.60	63.85	63.23	64.64	64.06	63.12	64.07	62.83	64.83	63.73
Anah.17	63.68	63.32	63.53	62.99	64.55	64.71	63.89	63.53	63.70	64.14
Anah.18	63.83	64.18	64.28	63.37	63.42	63.07	63.42	64.85	64.07	62.63
Anah.19	64.33	63.42	64.31	63.89	63.42	63.97	65.23	64.09	64.22	64.97
Anah.20	64.67	64.07	64.82	63.82	63.64	64.25	63.35	64.02	63.52	63.70



Şekil 5.2. AES-128 bit anahtar genişletme tasarımı D_1-S-D_1

Tablo 5.3'de ise AES-128 için geliştirilen anahtar genişletme algoritmalarından D_1-S-D_1 yapısı ile gizli bir anahtardan bir alt anahtarın elde edilmesi gösterilmektedir.

Tablo 5.3. 128 bit D_1-S-D_1 tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi

Şifre Anahtarı	13 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23			
RCON	3D CE 92 37 00 12 45 29 6A D2 73 54 01 73 FB AC			
RCON çıkışı	2E 8B 30 96 23 23 E1 8A D8 1E D9 60 C3 C8 8C 8F			
Kelimeler(w_0, w_1, w_2, w_3)	2E 8B 30 96	23 23 E1 8A	D8 1E D9 60	C3 C8 8C 8F
W[3] Difüzyon Çıkışı	CB 84 87 C0			
W[3] S kutusu Çıkışı	1F 5F 17 BA			
W[3] Difüzyon Çıkışı	F2 FA 57 B2			
t4 değişkeni	F2 FA 57 B2			
Kelimeler(w_4, w_5, w_6, w_7)	DC 71 67 24	FF 52 86 AE	27 4C 5F CE	E4 84 D3 41
W[7] Difüzyon Çıkışı	16 21 B3 76			
W[7] S kutusu Çıkışı	47 FD 6D 38			
W[7] Difüzyon Çıkışı	A8 82 D7 12			
t8 değişkeni	A8 82 D7 12			
Kelimeler(w_8, w_9, w_{10}, w_{11})	74 F3 B0 36	8B A1 36 98	AC ED 69 56	48 69 BA 17
W[11] Difüzyon Çıkışı	C4 36 9B E5			

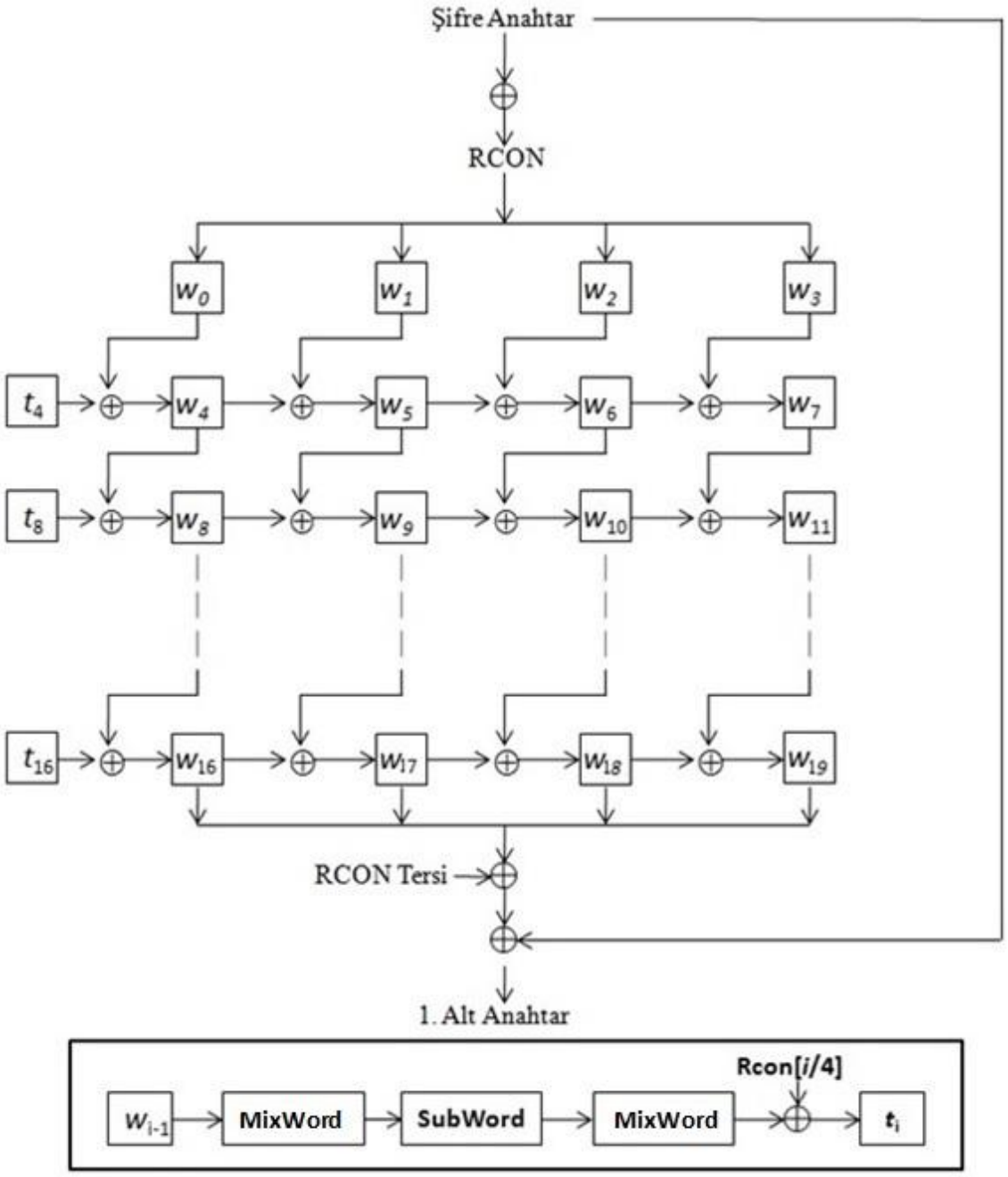
W[11] S kutusu Çıkışı	1C 05 14 D9			
W[11] Difüzyon Çıkışı	C8 C0 0D D1			
t12 değişkeni	C8 C0 0D D1			
Kelimeler(w_{12}, w_{13}, w_{14}, w_{15})	BC 33 BD E7	37 92 8B 7F	9B 7F E2 29	D3 16 58 3E
W[15] Difüzyon Çıkışı	70 FB 9D B5			
W[15] S kutusu Çıkışı	51 0F 5E D5			
W[15] Difüzyon Çıkışı	84 8B 00 DA			
t16 değişkeni	84 8B 00 DA			
Kelimeler(w_{16}, w_{17}, w_{18}, w_{19})	38 B8 BD 3D	0F 2A 36 42	94 55 D4 6B	47 43 8C 55
RCON Tersi	AC FB 73 01 54 73 D2 6A 29 45 12 00 37 92 CE 3D			
RCON Tersi XOR çıkışı	94 43 CE 3C 5B 59 E4 28 BD 10 C6 6B 70 D1 42 68			
Şifre anahtarı	12 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23			
Şifre Anahtarı XOR çıkışı	BA C8 FE AA 78 7A 05 A2 65 0E 1F 0B B3 19 CE E7			
1.Alt Anahtar	BA C8 FE AA 78 7A 05 A2 65 0E 1F 0B B3 19 CE E7			

Tablo 5.4'de ise AES-128 için D_1-S-D_1 yapısı ile 20 gizli anahtardan üretilmiş alt anahtarların 128 farklı bit pozisyonu için ortalama bit değişimleri gösterilmiştir.

Tablo 5.4. AES-128 için geliştirilen anahtar genişletme algoritmalarından D_1-S-D_1 yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 128 farklı bit pozisyonu için ortalama bit değişimleri

Gizli Anah.	Döngü Anah.1	Döngü Anah.2	Döngü Anah.3	Döngü Anah.4	Döngü Anah.5	Döngü Anah.6	Döngü Anah.7	Döngü Anah.8	Döngü Anah.9	Döngü Anah.10
Anah.1	63.53	64.07	64.01	64.54	63.46	63.97	63.78	63.41	63.85	64.45
Anah.2	63.45	63.69	64.06	63.69	63.96	63.73	64.17	63.61	64.77	64.16
Anah.3	63.44	64.42	63.33	63.81	64.32	63.71	63.42	62.96	64.91	63.81
Anah.4	64.21	64.13	63.90	63.78	64.01	63.52	63.65	63.88	63.97	64.06
Anah.5	63.71	64.76	64.51	63.27	64.39	63.36	64.26	63.71	63.77	64.18
Anah.6	64.75	63.44	64.27	64.34	64.32	64.43	64.37	63.58	62.95	64.26
Anah.7	64.67	63.09	64.11	64.34	63.62	64.28	64.32	64.82	63.68	65.02
Anah.8	64.41	62.76	64.18	63.70	64.30	64.36	64.07	63.99	64.07	62.96
Anah.9	64.57	64.18	63.62	63.96	63.85	62.92	63.47	63.85	63.80	65.24
Anah.10	63.89	64.82	64.58	64.24	63.42	63.58	64.85	65.10	64.02	63.93
Anah.11	63.87	63.54	63.77	64.84	63.13	64.10	64.49	63.50	63.31	64.10
Anah.12	64.21	63.82	64.38	64.26	63.43	63.92	64.44	63.42	63.98	63.97
Anah.13	64.95	63.26	64.53	64.03	63.57	64.50	64.74	64.00	63.55	64.13
Anah.14	63.87	63.54	63.77	64.84	63.13	64.10	64.49	63.50	63.31	64.10
Anah.15	63.93	63.74	65.07	64.04	64.69	64.73	63.59	63.33	63.47	64.47

Anah.16	63.11	63.92	64.22	63.41	64.46	63.57	64.05	64.39	63.76	63.61
Anah.17	64.75	63.89	63.55	63.17	64.51	64.48	64.44	64.12	64.64	63.97
Anah.18	64.33	63.28	65.02	63.84	64.51	63.21	63.87	64.37	64.43	63.63
Anah.19	64.61	63.86	64.35	64.21	64.24	64.75	64.44	63.12	63.72	63.93
Anah.20	64.14	63.76	64.40	63.94	63.99	64.53	64.09	64.10	63.65	64.46



Şekil 5.3. AES-128 bit anahtar genişletme tasarımı D_1-S-D_2

Tablo 5.5'de ise AES-128 için geliştirilen anahtar genişletme algoritmalarından D_1-S-D_2 yapısı ile gizli bir anahtardan bir alt anahtarın elde edilmesi gösterilmektedir.

Tablo 5.5. 128 bit D_1-S-D_2 yapısı ile şifre anahtarından 1. alt anahtarın elde edilmesi

Şifre Anahtarı	13 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23			
RCON	3D CE 92 37 00 12 45 29 6A D2 73 54 01 73 FB AC			
RCON çıkışı	2E 8B 30 96 23 23 E1 8A D8 1E D9 60 C3 C8 8C 8F			
Kelimeler(w_0, w_1, w_2, w_3)	2E 8B 30 96	23 23 E1 8A	D8 1E D9 60	C3 C8 8C 8F
W[3] Difüzyon Çıkışı	84 C0 87 44			
W[3] S kutusu Çıkışı	5F BA 17 1B			
W[3] Difüzyon Çıkışı	FE 53 F2 AD			
t4 değişkeni	FE 53 F2 AD			
Kelimeler(w_4, w_5, w_6, w_7)	D0 D8 C2 3B	F3 FB 23 B1	2B E5 FA D1	E8 2D 76 5E
W[7] Difüzyon Çıkışı	9B C0 B3 5B			
W[7] S kutusu Çıkışı	14 BA 6D 39			
W[7] Difüzyon Çıkışı	97 40 C3 D7			
t8 değişkeni	97 40 C3 D7			
Kelimeler(w_8, w_9, w_{10}, w_{11})	47 98 01 EC	B4 63 22 5D	9F 86 D8 8C	77 AB AE D2
W[11] Difüzyon Çıkışı	0E 0B 72 05			
W[11] S kutusu Çıkışı	AB 2B 40 6B			

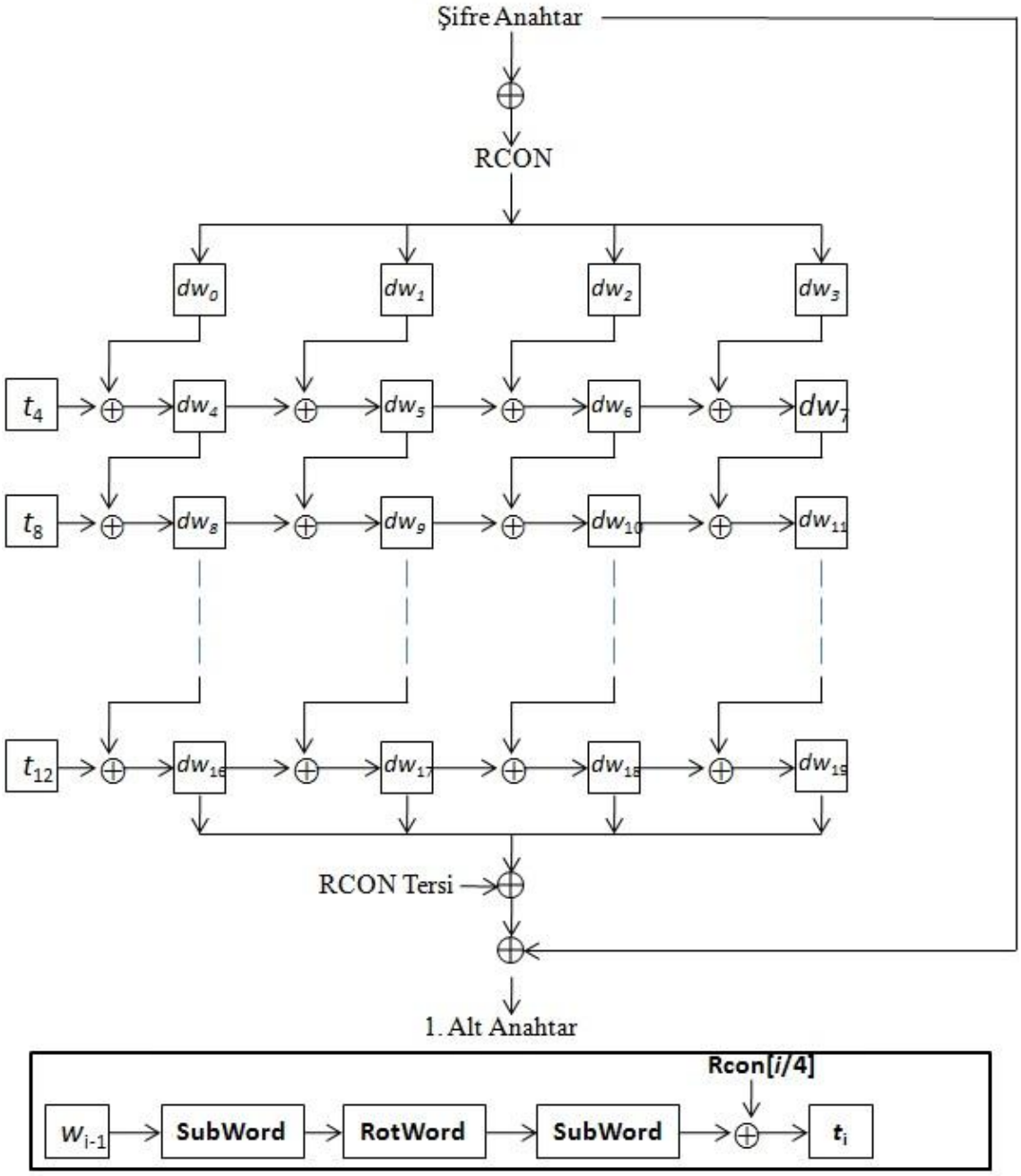
W[11] Difüzyon Çıkışı	EB 80 C0 6B			
t12 değişkeni	EB 80 C0 6B			
Kelimeler(w_{12}, w_{13}, w_{14}, w_{15})	AC 18 C1 87	18 7B E3 DA	87 FD 3B 56	F0 56 95 84
W[15] Difüzyon Çıkışı	22 E1 33 C3			
W[15] S kutusu Çıkışı	93 F8 C3 2E			
W[15] Difüzyon Çıkışı	45 7E A8 3B			
t16 değişkeni	45 7E A8 3B			
Kelimeler(w_{16}, w_{17}, w_{18}, w_{19})	E9 66 69 BC	F1 1D 8A 66	76 E0 B1 30	86 B6 24 B4
RCON Tersi	AC FB 73 01 54 73 D2 6A 29 45 12 00 37 92 CE 3D			
RCON Tersi XOR çıkışı	95 54 79 3A 01 2C 98 F2 9B 36 47 A0 49 91 AD 9E			
Şifre anahtarı	12 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23			
Şifre Anahtarı XOR çıkışı	BB DF 49 AC 22 0F 79 78 43 28 9E C0 8A 59 21 11			
1.Alt Anahtar	BB DF 49 AC 22 0F 79 78 43 28 9E C0 8A 59 21 11			

Tablo 5.6'de ise AES-128 için önerilen anahtar genişletme algoritmalarından D_1-S-D_2 yapısı ile 20 gizli anahtardan üretilmiş alt anahtarların 128 farklı bit pozisyonu için ortalama bit değişimleri gösterilmiştir.

Tablo 5.6. AES-128 için geliştirilen anahtar genişletme algoritmalarından D_1-S-D_2 yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 128 farklı bit pozisyonu için ortalama bit değişimleri

Gizli Anah.	Döngü Anah.1	Döngü Anah.2	Döngü Anah.3	Döngü Anah.4	Döngü Anah.5	Döngü Anah.6	Döngü Anah.7	Döngü Anah.8	Döngü Anah.9	Döngü Anah.10
Anah.1	65.06	63.90	63.92	63.42	64.32	63.72	64.50	62.55	64.17	63.40
Anah.2	64.0	63.81	63.70	63.89	64.5	63.28	63.92	63.83	63.76	64.67
Anah.3	64.10	64.60	64.38	63.44	63.75	64.39	64.14	64.47	64.63	63.87
Anah.4	63.22	64.77	64.09	63.59	63.66	64.16	63.74	63.92	63.18	63.48
Anah.5	64.14	63.75	63.67	63.43	64.23	63.34	64.62	64.33	63.70	64.41
Anah.6	63.08	64.39	64.23	64.70	65.06	63.71	63.17	64.57	63.88	64.75
Anah.7	64.28	63.75	63.23	62.84	64.23	64.42	63.92	64.20	63.96	63.09
Anah.8	64.48	64.51	64.11	63.96	64.16	63.77	63.25	64.07	63.75	63.39
Anah.9	64.22	64.07	64.62	63.57	63.24	63.17	64.01	65.07	63.68	64.30
Anah.10	63.68	65.06	64.28	64.47	63.97	64.14	64.81	63.85	63.94	63.62
Anah.11	63.69	64.36	64.54	63.96	63.28	64.60	64.63	64.52	63.99	64.02
Anah.12	64.29	63.37	64.49	63.92	65.12	63.84	64.40	63.92	64.38	63.41
Anah.13	63.10	63.59	63.78	64.20	64.37	63.81	63.45	64.51	64.67	63.57
Anah.14	63.97	63.82	63.60	62.75	63.37	63.78	64.39	63.49	65.28	63.83
Anah.15	63.42	63.71	64.37	63.69	64.14	63.38	63.67	64.69	64.06	64.56

Anah.16	64.02	63.56	64.55	64.07	64.46	64.39	64.46	63.89	63.38	63.69
Anah.17	63.80	64.26	64.92	63.64	63.85	64.54	64.66	63.86	63.52	64.20
Anah.18	63.32	64.37	63.52	63.42	64.51	62.76	63.59	64.77	64.56	63.00
Anah.19	64.35	63.92	63.85	63.85	63.60	63.93	62.99	63.79	64.35	63.77
Anah.20	62.88	63.03	64.31	64.21	64.46	64.47	62.99	63.60	62.75	63.98



Şekil 5.4. AES-256 bit anahtar genişletme tasarımı S-D_1-S

Tablo 5.7'de ise AES-256 için geliştirilen anahtar genişletme algoritmalarından S-D_1-S yapısı ile gizli bir anahtardan bir alt anahtarın elde edilmesi gösterilmektedir.

Tablo 5.7. 256-bit S-D_1-S tasarımı ile şifre anahtarından 1. alt anahtarın elde edilmesi

Şifre Anahtarı	1345A2A12331A4A3 B2CCAA34C2BB7723 1245A2A12331A4A3 B2CCAA34C2BB7723			
RCON	3DCE923700124529 6AD273540173FBAC D24ED23918624287 BAD290AFF01BAD25			
RCON çıkışı	2E8B30962323E18A D81ED960C3C88C8F C00B70983B53E624 081E3A9B32A0DA06			
Kelimeler(w_0, w_1, w_2, w_3)	2E8B30962323E18A	D81ED960C3C88C8F	C00B70983B53E624	081E3A9B32A0DA06
W[3] S kutusu Çıkışı	3072801423E0576F			
W[3] Difüzyon Çıkışı	DEE91B50B6135EFC			
W[3] S kutusu Çıkışı	1D1EAF534E7D58B0			
t4 değişkeni	1D1EAF534E7D58B0			
Kelimeler(w_4, w_5, w_6, w_7)	33959FC56D5EB93A	EB8B46A5AE9635B5	2B80363D95C5D391	239E0CA6A7650997
W[7] S kutusu Çıkışı	260BFE245C4D0188			
W[7] Difüzyon Çıkışı	92201F5F44B0A60C			
W[7] S kutusu Çıkışı	4FB7C0CF1BE724FE			

t8 değişkeni	4FB7C0CF1BE724FE			
Kelimeler(w_8, w_9, w_{10}, w_{11})	7C225F0A76B99DC4	97A919AFD82FA871	BC292F924DEA7BE0	9FB72334EA8F7277
W[11] S kutusu Çıkışı	DBA92618877340F5			
W[11] Difüzyon Çıkışı	A8BDB80419D6048F			
W[11] S kutusu Çıkışı	C27A6CF2D4F6F273			
t12 değişkeni	C27A6CF2D4F6F273			
Kelimeler(w_{12}, w_{13}, w_{14}, w_{15})	BE5833F8A24F6FB7	29F12A577A60C7C6	95D805C5378ABC26	0A6F26F1DD05CE51
W[15] S kutusu Çıkışı	67A8F7A1C16B8BD1			
W[15] Difüzyon Çıkışı	D3C4EC8C05D15380			
W[15] S kutusu Çıkışı	661CCE646B3EEDCD			
t16 değişkeni	661CCE646B3EEDCD			
Kelimeler(w_{16}, w_{17}, w_{18}, w_{19})	D844FD9CC971827A	F1B5D7CBB31145BC	646DD20E849BF99A	6E02F4FF599E37CB
RCON Tersi	25AD1BF0AF90D2BA 8742621839D24ED2 ACFB73015473D26A 294512003792CE3D			
RCON Tersi	FDE9E66C66E150C0 76F7B5D38AC30B6E C896A10FD0E82BF0			

XOR çıkışı	4747E6FF6E0CF9F6		
Şifre anahtarı	1245A2A12331A4A3 B2CCAA34C2BB7723	B2CCAA34C2BB7723	1245A2A12331A4A3
Şifre Anahtarı XOR çıkışı	D362D6FA45C2B14A 4F59DC645CAC23F0	AEE96CB3490B87E1	089DD197EBBBCDD4
1.Alt Anahtar	D362D6FA45C2B14A 4F59DC645CAC23F0	AEE96CB3490B87E1	089DD197EBBBCDD4

Tablo 5.8'de ise AES-256 için önerilen anahtar genişletme algoritmasının S-D_1-S yapısı ile 20 gizli anahtardan üretilmiş alt anahtarların 256 farklı bit pozisyonu için ortalama bit değişimleri gösterilmiştir.

Tablo 5.8. AES-256 için geliştirilen anahtar genişletme algoritmasının S-D_1-S yapısı ile 20 gizli anahtardan üretilmiş alt anahtarlarının 256 farklı bit pozisyonu için ortalama bit değişimleri

Gizli Anah.	Döngü An.1	Döngü An.2	Döngü An.3	Döngü An.4	Döngü An.5	Döngü An.6	Döngü An.7	Döngü An.8	Döngü An.9	Döngü An.10	Döngü An.11	Döngü An.12	Döngü An.13	Döngü An.14
Anah.1	127.3	128.3	127.6	127.6	128.4	127.6	127.4	128.6	128.1	127.8	127.2	127.3	127.5	127.8
Anah.2	127.3	127.8	127.8	128.2	128.6	128.2	127.5	128.0	127.0	129.2	128.2	128.3	127.7	129.2
Anah.3	127.7	127.4	127.4	127.8	127.5	128.0	127.7	128.2	127.4	128.1	127.1	127.6	127.4	128.1
Anah.4	127.6	127.4	127.6	128.0	127.5	127.9	128.9	127.2	127.9	128.8	128.6	127.9	127.5	128.8
Anah.5	128.6	127.5	128.6	128.1	128.5	127.9	127.7	127.5	127.9	127.3	128.5	128.1	127.9	127.3
Anah.6	128.9	128.1	128.5	127.7	128.1	127.7	128.0	127.2	127.4	128.2	128.7	127.2	127.8	128.2
Anah.7	128.0	128.1	127.9	127.6	128.3	128.9	128.3	129.1	128.4	128.1	128.8	127.4	129.1	128.1
Anah.8	127.9	128.1	127.9	127.4	127.1	128.2	128.5	128.2	128.1	128.2	127.9	128.7	128.1	128.2
Anah.9	127.6	128.0	127.7	127.9	127.4	127.7	128.3	128.7	128.6	127.8	128.0	127.7	127.7	127.8
Anah.10	128.0	128.4	128.3	127.7	128.0	128.2	128.4	128.8	127.9	127.7	128.1	127.4	127.6	127.7
Anah.11	127.7	128.0	127.2	128.2	127.7	127.7	127.2	127.8	129.0	128.2	128.3	128.2	127.6	128.2
Anah.12	128.3	128.5	128.5	128.3	129.5	129.0	126.7	128.1	128.5	128.2	127.4	128.3	127.1	128.2
Anah.13	128.6	126.8	127.9	128.4	128.7	129.2	127.2	127.8	127.5	127.6	128.6	127.7	128.3	127.6
Anah.14	128.1	127.4	127.0	127.9	128.8	127.0	128.2	128.6	127.9	127.9	128.2	128.4	129.5	127.9
Anah.15	128.6	128.1	128.3	127.7	128.1	128.2	127.9	127.7	127.6	128.0	127.6	129.0	126.9	128.0
Anah.16	127.7	127.3	127.9	128.1	127.1	128.1	128.0	128.5	126.4	127.6	127.8	127.6	128.4	127.6
Anah.17	127.7	128.0	127.9	127.2	128.0	128.1	128.3	128.9	128.4	127.7	128.5	127.5	126.8	127.7
Anah.18	128.2	128.5	128.5	128.4	129.0	128.9	127.8	128.1	128.1	128.2	127.7	127.4	128.2	128.2

Anah.19	127.3	127.7	128.8	128.4	127.8	129.5	127.7	128.8	127.7	128.4	128.1	128.2	127.8	128.4
Anah.20	127.1	127.9	127.8	129.0	128.0	128.2	127.5	126.8	126.9	128.5	128.5	128.6	128.5	128.5

BÖLÜM 6

SONUÇLAR

Bu tezde iki önemli blok şifresi (AES ve ARIA) ve bu şifrelerin kullandığı anahtar genişletme algoritmaları incelenmiştir. Önemli iki blok şifresi olan AES ve ARIA blok şifrelerinin anahtar genişletme algoritmalarındaki zaaflar tespit edilerek bu zaafların giderilmesi amacıyla AES blok şifresinin anahtar genişletme algoritması tabanlı yeni bir anahtar genişletme algoritması ortaya konmuştur. Not edilmelidir ki AES blok şifresinin anahtar genişletme algoritması bit sızıntısı ve yavaş yayılım özellikleri ile bazı saldırılara karşı şifreyi güçsüz kılacak önemli zaaflara sahiptir. Çalışmamızda farklı anahtar büyüklüklerine hatta bir blok şifreden bağımsız olarak tasarlanabilecek yeni anahtar genişletme algoritması verildiği gibi bu yeni anahtar genişletme algoritması ile üretilen alt anahtarların çığ etkisi özellikleri de incelemiştir. Bu özellik için gayet iyi sonuçlar elde edilmiş ve tezin 5. Bölümünde bu sonuçlara değinilmiştir. Geliştirilen anahtar genişletme algoritması farklı boyutlar ve yapılar (yer değiştirme ve yayılım tabakaları) ile tekrar geliştirilebileceğinden dolayı bir şifrenin içerisinde verilen algoritma kullanılmadan önce sonuçların (alt anahtarların) daha detaylı bir analizinin yapılması (alt anahtarların rastsallık özellikleri, katı çığ özellikleri gibi kriterler göz önüne alınarak detaylı bir analiz) tavsiye edilmektedir.

KAYNAKLAR

[1] Henk C.A. Van Tilborg, ***Fundamentals of Cryptology***, Kluwer Academic Publishers, 2000.

[2] Shon Harris, ***All-in-One CISSP***, McGraw-Hill Osborne Media, 2001.

[3] Jan C. A. Van Der Lubbe, ***Basic Methods of Cryptography***, Cambridge University Press, United Kingdom, 2000.

[4] David Khan, ***The Code Breakers***, Macmillan Company, New York, 1967

[5] Jonathan Katz,Yehuda Lindell,***Introduction to Modern Cryptography,*** Chapman & Hall/CRC Press,United States,2007

[6] C. Smith,***Basic Cryptanalysis Techniques***, SANS Institute, November, 2001.

[7] M. Matsui, ***Linear Cryptanalysis Method for DES Cipher***, Advances in Cryptology - EUROCRYPT '93 (Lecture Notes in Computer Science no. 765), Springer-Verlag, pp. 386-397, 1994.

[8] E. Biham and A. Shamir, ***Differential Cryptanalysis of DES-like Cryptosystems***, Journal of Cryptology, vol. 4, no. 1, pp. 3-72, 1991.

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

[10] 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.

[11] H. M. Heys, ***A Tutorial on Linear and Differential Cryptanalysis***, Memorial University of Newfoundland, Canada.

[12] Y. Xue, ***Symmetric Encryption Principles***, available at: file:///C:/Users/vural/Downloads/lecture7.pdf

[13] Anuj Prateek, ***A REPORT ON BLOCK CIPHERS***, National Aerospace Laboratories,Bangalore,Indian,2006

[14] Cheng. J. Kuo, ***Cryptography***, Graduate Institute of Communication Engineering,Taipei,Taiwan

[15] Blok Şifre (Block Cipher), available at: http://en.wikipedia.org/wiki/Block_cipher, 2012.

[16] R. Gens, P. Domingos, ***Learning the Structure of Sum-Product Networks***, Department of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA.

[17] Claude E. Shannon, ***Communication Theory of Secrecy System***, Bell System Technical Journal,1949

[18] A. Rashed, ***Intelligent Encryption Decryption Systems***, PhD Thesis, Computer Information System Department, Arab Academy for Banking and Financial Sciences, 2004.

[19] F. B. SAKALLI, ***AKIŞ ŞİFRELERİN TASARIM TEKNİKLERİ VE GÜÇ ANALİZİ***, Doktora tezi, Fen Bilimleri Enstitüsü, Trakya Üniversitesi, Edirne, 2011.

[20] K. KAZLAUSKAS, J. KAZLAUSKAS, ***Key-Dependent S-Box Generation in AES Block Cipher System***, Institute of Mathematics and Informatics, Vilnius, Lithunia,2008.

[21] C. Schnorr and S. Vaudenay, ***Black Box Cryptoanalysis of Hash Networks Based on Multipermutations***, In EUROCRYPT'94, volume 950, pages 47–57. Springer, 1994.

[22] S. Vaudenay, ***On the Need for Multipermutations: Cryptanalysis of MD4 and SAFER***, In FSE'94, volume 1008, pages 286–297. Springer, 1994.

[23] J. Daemen, V. Rijmen, ***The Wide Trail Design Strategy***, ProtonWorld, Brussel, Belgium

[24] M. Sajadieh, M. Dakhilalian, H. Mala, P. Sepehrdad, ***Recursive Diffusion Layers for Block Ciphers and Hash Functions***, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

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

[26] J. Daemen, V. Rijmen, ***Note on naming***, available at: http://csrc.nist.gov/archive/aes/rijndael/Rijndael-ammended.pdf

[27] N. Ajlouni, A. El-Sheikh, A. Rashed, ***A New Approach in Key Generation and***

Expansion in Rijndael Algorithm, The International Arab Journal of Information Technology, Vol.3, No.1, January, 2006.

[28] A. J. Menezes, Paul C.V. Oorschot, Scott A. Vanstone, ***Handbook of Applied Cryptography***, CRC Press, Boca Raton, USA, 1997.

[29] M. J. B. Robshaw, ***Stream Ciphers***, RSA Laboratories, Redwood, 1995.

[30] J.S. Milne, ***FIELDS AND GALOIS THEORY***, August 31, 2003 available at: http://www.galois-group.net/theory/math594fS.pdf

[31] Sonlu Cisim (Finite Field), available at: http://en.wikipedia.org/wiki/Finite_field

[32] M. Şenol, ***Grup Halkaları ve Önemi***, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, Çukurova Üniversitesi, 2011, Adana.

[33] M. Şahinoğlu, ***AES ALGORİTMASININ FPGA ÜZERİNDE GERÇEKLEMESİNE ELEKTROMANYETİK ALAN SALDIRISI***, Yüksek Lisans Tezi, Fen Bilimleri Enstitüsü, İTÜ, Aralık, 2008.

[34] B. Aslan, ***Boole Fonksiyonları ve S-kutularının Kriptografik özelliklerinin İncelenmesi ve Ters Haritalama Tabanlı Cebirsel Açıdan Güçlendirilmiş Bir S-kutusu Önerisi***, Yüksek Lisans Tez, Fen Bilimleri Enstitüsü, Trakya Üniversitesi, 2008, Edirne.

[35] S. Bulut, ***Modern Bir Blok Şifre Tasarımı***, Yüksek Lisans Tez, Fen Bilimleri Enstitüsü, Trakya Üniversitesi, 2012, Edirne.

[36] E. Biham. ***New Types of Cryptanalytic Attacks using Related Keys***, Advances in Cryptology–EUROCRYPT'93, LNCS 765, Springer-Verlag, pp 398-409, 1993.

[37] J. Daemen, R. Govaerts, J. Vandewalle, ***Weak Keys for IDEA***, Advances in Cryptology–CRYPTO'93, LNCS 773, Springer-Verlag, pp 224-231, 1993.

[38] L.Knudsen, ***New Potentially Weak Keys for DES and LOKI***, Advances in Cryptology–EUROCRYPT'94, LNCS 950, Springer-Verlag, pp 419-424, 1994,.

[39] J. Daemen, L. Knudsen, V. Rijmen, ***The Block Cipher SQUARE***, Fast Software Encryption, Fourth International Workshop, Springer-Verlag,Belgium,1997.

[40] L.Knudsen, ***Practically Secure Feistel Ciphers***, Fast Software Encryption, First International Workshop Proceedings, LNCS 809, Springer-Verlag, pp 211-221, 1993.

[41] M.Leech. ***A Feistel Cipher with Hardened Key Scheduling***, Workshop on Selected Areas in Cryptography (SAC'96), pp 15-29.

[42] V.Rijmen, J.Daemen, B.Preneel, A.Bosselaers and E.DeWin, ***The Cipher SHARK***,

Fast Software Encryption, Third International Workshop, LNCS 1039, Springer-Verlag, pp 99-111, 1996.

[43] B.Schneier. ***Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)***, Fast Software Encryption, First International Workshop, LNCS 809, Springer-Verlag, pp 191-204, 1993.

[44] K.Aoki, T.Ichikawa, M.Kanda, M.Matsui, S. Moriai, J. Nakajima and T. Tokita. *Camellia:* ***A 128-Bit Block Cipher Suitable for Multiple Platforms - Design and Analysis***, Workshop on Selected Areas in Cryptography (SAC 2000), LNCS 2012, pp 39-56.

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

[46] L. May, M. Henricksen, W. Millan, G. Carter, E. Dawson, ***Strengthening the Key Schedule of the AES,*** Information Security Research Centre, Queensland University of Technology, Brisbane, Australia.

[47] Z. Yuan, ***New Impossible Differential Attacks on AES***, Beijing Electronic Science and Technology Institute, Beijing, China, available at:http://eprint.iacr.org/2010/093.pdf

[48] E. Biham and N. Keller, "***Cryptanalysis of Reduced Variants of Rijndael***," available at: *http://csrc.nist.gov/encryption/aes/round2/conf3/aes3papers.html*

[49] C. D'Halluin, G. Bijnens, V. Rijmen, and B. Preneel, "***Attack on Six Rounds of Crypton***," Proc. of Fast Software Encryption'99, Lecture Notes in Computer

[50] H. Seki, T. Kaneko, "***Cryptanalysis of Five Rounds of CRYPTON Using Impossible Differentials***," Proc. of Asiacrypt'99, Lecture Notes in Computer Science Vol. 1716, pp.43-51,1999.

[51] H. Seki, T. Kaneko, ***Cryptanalysis of Five Rounds of CRYPTON Using Impossible Differentials***, Proc. of Asiacrypt'99, Lecture Notes in Computer Science Vol. 1716, pp.43-51,1999.

[52] J. Kelsey, B. Schneier, D. Wagner, ***Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES***, In Neal Koblitz, editor, Advances in Cryptology - CRYPTO'96, 16th Annual International Cryptology Conference, volume 1109 of Lecture Notes in Computer Science, pages 237-251. Springer-Verlag, 1996.

[53] A. Biryukov, D. Khovratovich, ***Related-key Cryptanalysis of the Full AES-192 and AES-256***, University of Luxembourg, Luxembourg, Cryptology ePrint Archive, Report 2009/317, June 2009.

[54] A, Biryukov, D. Khovratovich, I. Nikolic, ***Distinguisher and Related-Key Attack on the Full AES-256***, In Shai Halevi, editor, Advances in Cryptology - CRYPTO 2009,

volume 5677 of Lecture Notes in Computer Science, pages 231-249. Springer-Verlag, 2009.

[55] E. Biham, O. Dunkelman, N. Keller, ***A Simple Related-Key Attack on the Full SHACAL-1***, The Cryptographers Track at the RSA Conference 2007, volume 4377 of Lecture Notes in Computer Science, pages 20-30. Springer-Verlag, 2007.

[56] E. Biham, O. Dunkelman, N. Keller, ***A Related-Key Rectangle Attack on the Full KASUMI***, In Bimal K. Roy, editor, Advances in Cryptology - ASIACRYPT 2005, 11th International Conference on the Theory and Application of Cryptology and Information Security, volume 3788 of Lecture Notes in Computer Science, pages 443-461. Springer-Verlag, 2005

[57] A, Biryukov, O. D. Elman, N. Keller, D. Khovratovich, A. Shamir, ***Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds***, University of Luxembourg, Luxembourg, Cryptology ePrint Archive, Report 2009/374, August 2009.

[58] S. Mukhopadhyah, ***The Advanced Encryption Standard***, Cryptography and Network Security - MA61027, available at: http://www.facweb.iitkgp.ernet.in/~sourav/AES.pdf

[59] NSRI, Specification of ARIA, January, 2005, available at: http://edipermadi.files.wordpress.com/2008/09/aria-specification-e.pdf

[60] B. Aslan, ***Blok Şifreler İçin Cebirsel İkili Doğrusal Dönüşüm Tasarımı ve Modern Bir Blok Şifreye Uygulanması***, Doktora tezi, Fen Bilimleri Enstitüsü, Trakya Üniversitesi, 2013.

ÖZGEÇMİŞ

1984 yılında Adıyaman'da doğdu. İlk, Orta ve Lise öğrenimini Adıyaman'da tamamladı. Adıyaman Anadolu Öğretmen Lisesi mezuniyetinden sonra Bilkent Üniversitesi Bilgisayar Mühendisliği bölümünden 2008 yılında mezun oldu. 2008-2011 yılları arasında TEB İstanbul Operasyon Merkezi Bilgi Teknolojileri Bölümünde Bilgisayar Mühendisi olarak yönetici yardımcısı pozisyonunda çalıştı. 2011 yılından itibaren Adıyaman Kahta Meslek Yüksek Okulu Bilgisayar Teknolojileri bölümünde Öğretim Görevlisi olarak çalışmaya devam etmektedir. Trakya Üniversitesi Bilgisayar Mühendisliği bölümünde Yüksek Lisansını yapmaktadır.

TEZ ÖĞRENCİSİNE AİT TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER

Fatma B. SAKALLI, Ercan BULUŞ, Muharrem T. SAKALLI, Hüseyin VURAL, ***AES Blok Şifresinin Anahtar Genişletme Rutininin Geliştirilmesi ve Bir Blok Şifreden Bağımsız Anahtar Genişletme Rutininin Tasarımı***, 6. ULUSLARARASI BİLGİ GÜVENLİĞİ ve KRİPTOLOJİ KONFERANSI, ODTÜ, Ankara, 2013.