GELİŞMİŞ ŞİFRELEME STANDARDI

Abdullah ULUTÜRK

YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NİSAN 2010
ANKARA

Abdullah ULUTÜRK tarafından hazırlanan GELİŞMİŞ ŞİFRELEME STANDARDI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Yrd.Doç. Dr. Nursel AKÇAM
Tez Danışmanı, Elektrik-Elektronik Müh. Anabilim Dalı

Bu çalışma, jürimiz tarafından oy birliği ile Elektrik-Elektronik Mühendisliği

Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Prof. Dr. Cengiz TAPLAMACIOĞLU …….....…….
Elektrik-Elektronik Müh. Anabilim Dalı, G.Ü.
Yrd. Doç. Dr. Hasan Şakir BİLGE …….....…….
Bilgisayar Müh. Anabilim Dalı, G.Ü.
Yrd. Doç. Dr. Nursel AKÇAM ….....……..
Elektrik-Elektronik Müh. Anabilim Dalı, G.Ü.

Tarih: 20/04/2010

Bu tez ile G.Ü. Fen Bilimleri Enstitüsü Yönetim Kurulu Yüksek Lisans derecesini onamıştır.

Prof. Dr. Bilal TOKLU …….

Fen Bilimleri Enstitüsü Müdürü

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

Abdullah ULUTÜRK

GELİŞMİŞ ŞİFRELEME STANDARDI

(Yüksek Lisans Tezi)

Abdullah ULUTÜRK

GAZİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Nisan 2010

ÖZET

Bu tez çalışmasında genel olarak kriptolojiye bakış, geçmişten günümüze kadar kullanılmış kripto sistemleri, haberleşmeyi engelleyecek veya bilgileri elde etmeye çalışacak elektronik saldırılar, bilgi taşımadaki güvenlik prensipleri, açık anahtarlı, kapalı anahtarlı ve hibrit sistemler hakkında detaylı bilgi verilmiştir. Daha sonra ulusal standartlar ve teknoloji enstitüsü tarafından oluşturulan ve bilgi güvenliğinin önemli olduğu birçok yerlerde kullanılan Gelişmiş Şifreleme Standardındaki matematiksel işlemler ve aşamalar hakkında örnekle beraber bilgi verilmiştir. Uygulama olarak MATLAB ile Gelişmiş Şifreleme Standardı kullanılarak simulasyon geliştirilmiş ve bir adet örneğe uygulanmıştır.

Bilim Kodu : 905.1.011
Anahtar Kelimeler : AES, Kriptografi, Gelişmiş şifreleme standardı
Sayfa Adedi : 111
Tez Yöneticisi : Yrd. Doç. Dr. Nursel AKÇAM

ADVANCED ENCRYPTION STANDARD

(M.Sc. Thesis)

Abdullah ULUTÜRK

GAZİ UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

April 2010

ABSTRACT

In this thesis study, in addition to general view to cryptology, detailed information regarding crypto systems used from past to present, electronic attacks that will block the communication or try to gain the information, security principles on data transmission, symmetric, asymmetric and hybrid systems is given. Subsequently, examples and information regarding mathematical processing's and stages at the Advanced Encryption Standard that is constructed by National Institute of Standards and Technology and used in many areas where the data security is important. Advanced Encryption Standard as an application using MATLAB simulation was developed and applied to a sample.

Science Code : 905.1.011
Key Words : AES, Cryptography, Advanced Encryption Standard
Page Number : 111
Adviser : Assist. Prof. Dr. Nursel AKÇAM

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren hocam Yrd. Doç. Dr. Nursel AKÇAM'a teşekkürü borç bilirim. Ayrıca çalışmalarımda yardımını esirgemeyen arkadaşım Ertuğrul AKSOY'a teşekkürlerimi sunarım. Son olarak, hayatımın her aşamasında olduğu gibi çalışma süresi boyunca da ilgi ve desteklerini esirgemeyen çok değerli aileme teşekkür ederim.

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

Şekil **Sayfa**

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler	**Açıklama**
a(x)	GF(28) 'de 'a' elemanının polinomsal gösterimi
GF(p)	P elemanlı Galois cismi
GF(p^n)	P^n elemanlı Galois cismi
GF(2^n)	2^n elemanlı Galois cismi
GF(2^8)	2^8 elemanlı Galois cismi
N^b	Blok uzunluğu
N^k	Anahtar uzunluğu
N^r	Tur sayısı
p	Asal sayı
RCon	Anahtar Üreteci tur sabiti
Z	Tamsayılar kümesi
$\oplus$	Exclusive-OR (XOR)

Kısaltmalar	**Açıklama**
ABD	Amerika Birleşik Devletleri
AES	Gelişmiş Şifreleme Standardı (Advanced Encryption Standard)
A5/1	GSM şifrelemede kullanılan 1. standart
A5/2	GSM şifrelemede kullanılan 2. standart
CBC	Kapalı Metin Zincirleme (Cipher Block Chaining)
CFB	Şifreyi Geri Besleme (Cipher Feedback)

Kısaltmalar	Açıklama
DES	Veri Şifreleme Standardı (Data Encryption Standard)
EC	Eliptik Eğri (Elliptic Curve)
ECC	Eliptik Eğri Kriptografisi (Elliptic Curve Cryptography)
ECDLP	Eliptik Eğri Ayrık Logaritma Problemi
FBI	Federal Bureau of Investigation
FIPS	Federal Bilgi İşlem Standardı (Federal Information Processing Standard)
ftp	Dosya transferi protokolü (File Transfer Protocol)
IDEA	Uluslararası Veri Şifreleme Algoritması (International Data Encryption Algorithm)
http	Hipermetin Aktarma İletişim Kuralı (Hypertext Transfer Protocol)
MAC	Ortam Erişim Yönetimi (Media Access Control)
MARS	Çarpma, Toplama, Döndürme ve Yerine koyma (Multiplication, Addition, Rotation and Substitution)
MD5	Mesaj Özetleme Algoritması 5 (Message Digest Algorithm 5)
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology)
NSA	ABD Ulusal Güvenlik Kurumu (National Security Agency)
PGP	Pretty Good Privacy
PKI	Açık Anahtar Altyapısı (Public Key Infrastructure)
RC4	Ron’s Code 4

Kısaltmalar	Açıklama
RC6	Ron’s Code 6
RSA	Rivest-Shamir-Adleman
SAC	Strict Avalance Criteria
SHA	Güvenli Özetleme Algoritması (Secure Hash Algorithm)
SO	Sertifika Otoriteleri
SOBER – 128	SOBER Ailesi Şifreleme
SSH	Güvenli Kabuk (Secure Shell)
SSL	Soket Düzeyi Güvenlik (Secure Sockets Layer)
3DES	Üçlü Veri Şifreleme Standardı (Triple Data Encryption Standard)

1. GİRİŞ

Özellikle 1990'lı yıllardan başlayarak yaşanan hızlı teknolojik gelişmeler ve internetin yaygınlaşmasının bir sonucu olarak bilişim güvenliği son yıllarda giderek önem kazanan bir konu haline gelmiştir. Konunun önümüzdeki dönemde kurumların öncelik listesinde giderek artan bir öneme sahip olacağı ve kurumların bilişim güvenliği alanına gereken önemi vermeye başladıkları, ilgili önlemleri alma çabası içine girdikleri bilinmektedir [1].

Kriptografik algoritmalar gün geçtikçe daha yaygın kullanım alanları bulmaktadır. İlk zamanlarda yüksek hız ve yüksek işlem gücü kapasitesine sahip devreler tasarlanmaya çalışılırken, enerji ve alan kısıtına sahip ortamlarda kullanım alanlarının artmasıyla güç ve alan tasarruflu gerçeklemeler büyük önem kazanmıştır. Bu konu üzerine artarak devam eden araştırmalarda düşük güç tasarrufu için farklı yöntemler önerilmekte ve gerçeklemelerin güç harcamaları azaltılmaya çalışılmaktadır [2].

NIST (Ulusal Standartlar ve Teknoloji Enstitüsü - National Institute of Standards and Technology) çağın gereklerine uygun ve güvenilir şifreleme standartları oluşturan bir kurumdur. NIST kullandığı standart artık güvenilir değilse yerine daha güvenilir ve kırılması zamanın şartlarına göre çok daha zor olan yeni bir şifreleme standardı geliştirir. NIST 2000 yılında AES (Gelişmiş Şifreleme Standardı - Advanced Encryption Standard)'nı kabul etmiş ve bu standart daha önce kullanılmakta olan DES (Veri Şifreleme Standardı - Data Encryption Standard)'nın yerini almıştır.

Belçikalı bilim adamları Joan Daemen ve Vincent Rijmen tarafından geliştirilen AES, DES 'e hız, güvenlik ve tasarım kolaylığı bakımından üstünlük sağlayıp kabul edilmiştir.

Blok şifreler modern şifreleme algoritmaları için oldukça önemlidir. Bundan dolayı veri iletişimdeki güvenlik düşünüldüğünde şifreleme algoritmalarının

gücü oldukça önemli bir kriterdir. Diğer yandan, blok şifreler onlara gücünü veren önemli özelliklere sahiptir. AES, günümüzde kullanılan en önemli blok şifreleme algoritmalarından biridir. Gücünü şifreleme algoritmasının en önemli parçalarından biri olan S kutularından almaktadır ve AES S kutusu 8 bit ve 8 bit çıkışa sahiptir [3, 4].

Türkiye'de kriptografi ile ilgili çalışmaların daha da genişlemesi ve gelecekte ülkenin kendi simetrik şifreleme algoritması tasarlamasının da gerekliliği düşünüldüğünde şifrenin içyapısında güvenli doğrusal olmayan ve AES S kutusunda olduğu gibi cebirsel S kutuları kullanılabilir [5].

Bu çalışmada AES detaylı bir şekilde incelenmiş ve MATLAB ile simule edilen çalışma anlatılmıştır.

İkinci bölümde, kripto bilimi ile ilgili genel bilgiler verilmiş, şifreleme biliminin tarihçesi anlatılmış, bilgi taşımada güvenlik prensiplerinden bahsedilmiştir.

Üçüncü bölümde, kripto sistemleri hakkında bilgi verilmiş, gizli anahtarlı sistemler, açık anahtarlı sistemler ve hibrit sistemler anlatılmıştır.

Dördüncü bölümde, genel olarak kriptolojide kullanılan matematiksel işlemlerden bahsedilmiştir.

Beşinci bölümde, AES şifreleme standardı hakkında genel bilgi verilmiş, şifreleme katmanları anlatılmış, şifrelemeyi ve şifre çözmeyi nasıl yaptığına dair bir örnekle açıklama yapılmıştır.

Altıncı bölümde, AES şifreleme standardının Matlab uygulamasıyla yapılan çalışma detaylı olarak anlatılmıştır.

Son bölümde de bu çalışmayla ilgili yapılan değerlendirmelere yer verilmiştir.

2. KRİPTOLOJİYE GENEL BAKIŞ

2.1. Kriptolojiye Giriş

Türkçe adı şifre yazımıdır. Kriptografi yunanca gizli anlamına gelen *kriptos* ve yazı anlamına gelen *graphi*'den türetilmiştir. Kriptoloji ise şifre bilimidir.

Hayatımızın birçok alanında bilgisayar ve bilgisayar ağı teknolojileri olmazsa olmaz bir şekilde yer almaktadır. İletişim, para transferleri, kamu hizmetleri, askeri sistemler, elektronik bankacılık, savunma sistemleri, bu alanlardan sadece birkaçıdır. Teknolojideki bu gelişmeler, bilgisayar ağlarını ve sistemlerini, aynı zamanda, bir saldırı aracı haline, kullandığımız sistemleri de açık birer hedef haline getirmiştir.

Bilişim sistemlerine ve bu sistemler tarafından işlenen verilere yönelik güvenlik ihlalleri inanılmaz bir hızla artmaktadır [1].

Kriptografi, veriyi yalnızca okuması istenen şahısların okuyabileceği bir şekilde saklamak ve göndermek amacıyla kullanılan bir teknolojidir. Kriptografide veri, matematiksel yöntemler kullanılarak kodlanır ve başkalarının okuyamayacağı hale getirilir. Bu matematiksel kodlamaya kripto algoritması adı verilir. Şekil 2.1.'de kripto adımları gösterilmiştir.

Şekil 2.1. Kriptolama adımları

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

bilgiye ise şifreli metin denir. Açık metinden şifreli metne geçme işlemi şifreleme, şifreli metinden açık metne geçme işlemi ise şifre çözme olarak adlandırılır.

Şifreleme ve şifre çözme yapan bir sistem de kripto sistem olarak adlandırılır. Bir kripto sistemin, şifreleme ve şifre çözme yapan hem donanım hem de yazılım bileşenleri olabilir. Algoritmalar, açık metin üzerinde yapılan karmaşık işlemlerden oluşan matematiksel formüllerdir. Bir algoritma, hem yazılımla hem de donanım bileşenleri ile gerçekleştirilebilir. Birçok algoritma, şifreleme ve şifre çözme işlemini gerçekleştirmek amacıyla, açık metinden başka, anahtar denen bir değer de kullanır. Anahtar 0 ve 1'lerden oluşan uzun bir bit dizisidir. Her algoritmanın kullandığı anahtar boyları farklıdır. Genellikle anahtar boyu arttıkça, olası anahtar sayısı arttığından, saldırganın bu şifreyi çözmesi güçleşir, ama aynı zamanda da şifreleme ve şifre çözme hızı yavaşlar. Bir algoritmanın olası tüm anahtarlar olasılıklarının oluşturduğu topluluğa anahtar uzayı denir.

2.2. Kriptolojinin Tarihçesi

Kriptoloji insanlığın yaratılışından çeşitli evreler geçirerek günümüze ulaşmıştır. Kısacası insanlık ne zaman var olmuşsa kriptolojide o zaman var olmuştur. İlk başlarda insanlar sadece gizlilik kavramını gerçekleştirmek için bu bilime ihtiyaç duymuşlar devir değiştikçe kriptoloji bölümünü alanları da değişmiştir. Özellikle web teknolojisinin gelişmesiyle kriptolojiye daha çok ihtiyaç duyulmuştur.

MÖ.1900 dolaylarında bir Mısırlı kâtip yazdığı kitabelerde standart dışı hiyeroglif işaretleri kullanmıştı.

MÖ.60–50 dolaylarında Julius Caesar normal alfabedeki harflerin yerini değiştirerek oluşturduğu şifreleme yöntemini devlet haberleşmesinde kullanmıştır.

725–790 dolaylarında Abu Abd al-Rahman al-Khalil ibn Ahmad ibn Amr ibn Tammam al Farahidi al-Zadi al Yahmadi, kriptografi hakkında bir kitap yazdı. Kitabı yazmasına ilham kaynağı olan, Bizans imparatoru için Yunanca yazılmış bir şifreli metni çözmesidir. Abu Abd al-Rahman, bu metni çözmek için ele geçirdiği şifreli mesajın başındaki açık metni tahmin etme yöntemini kullanmıştır.

1000–1200 dolaylarında Gazneliler'den günümüze kalan bazı dokümanlarda şifreli metinlere rastlanmıştır. Bir tarihçinin dönemle ilgili yazdıklarına göre yüksek makamlardaki devlet görevlilerine yeni görev yerlerine giderken şahsa özel şifreleme bilgileri verilmiştir.

1586'da Blaise de Vigenère (1523–1596) şifreleme hakkında bir kitap yazdı. İlk kez bu kitapta açık metin ve şifreli metin için otomatik anahtarlama yönteminden bahsedilmişti. Günümüzde bu yöntem hala DES, CBC (Kapalı Metin Zincirleme - Cipher Block Chaining) ve CFB (Şifreyi Geri Besleme - Cipher Feedback) kiplerinde kullanılmaktadır.

1623'de Sir Francis Bacon, 5 bit ikili kodlamayla karakter tipi değişikliğine dayanan stenografı bulmuştu.

1790'da Thomas Jefferson, Strip Cipher makinesini geliştirmişti. Bu makineyi temel alan M–138-A, ABD (Amerika Birleşik Devletleri) donanmasının 2.Dünya savaşında da kullanılmıştır.

1917'de Joseph Mauborgne ve Gilbert Vernam mükemmel şifreleme sistemi olan "one time pad"i bulmuşlar.

1920 ve 1930'larda FBI (Federal Bureau of Investigation) içki kaçakçılarının haberleşmesini çözebilmek bir araştırma ofisi kurdu. William Frederick Friedman, Riverbank Laboratuarlarını kurmuş, ABD için kriptoanaliz yapmış,

2. Dünya savaşında Japonlar'ın Purple Machine şifreleme sistemini çözülmüştür.

2. Dünya savaşında Almanlar Arthur Scherbius tarafından icat edilmiş olan Enigma makinesini kullanmışlardır. Bu makine Alan Turing ve ekibi tarafından çözülmüştü.

1952'de ABD'de Resmi olarak NSA (ABD Ulusal Güvenlik Kurumu - National Security Agency) kuruldu. Ayrıca bilgi toplamak için internet, telefon görüşmeleri ve e-postaları da izlerler. Dünyadaki en büyük telefon görüşmeleri arşivine bu teşkilat sahiptir. İllegal olarak sivillerin telefon görüşmelerini kaydettikleri ve telekomünikasyon şirketlerinden telefon kayıtlarını istedikleri ortaya çıkmıştır.

1970'de Horst Feistel (IBM) DES'in temelini oluşturan Lucifer algoritmasını geliştirmiştir.

1976'da DES, ABD tarafından FIPS 46 (Federal Bilgi İşlem Standardı Federal Information Processing Standard) standardı olarak açıklanmıştır.

1976'da Whitfield Diffie ve Martin Hellman açık anahtar sistemini anlattıkları makaleyi yayınlamışlardır.

1978'de Ronald L. Rivest, Adi Shamir ve Leonard M. Adleman: RSA (Rivest-Shamir-Adleman)algoritmasını bulmuşlardır.

1985'de Neal Koblitz ve Victor S.Miller ayrı yaptıkları çalışmalarda ECC (Eliptik Eğri Kriptografisi - Elliptic Curve Cryptography) sistemlerini tarif etmişlerdir.
1990'da Xuejia Lai ve James Massey IDEA (Uluslararası Veri Şifreleme Algoritması - International Data Encryption Algorithm) algoritmasını bulmuşlardır.

1991’de Phil Zimmerman PGP (Pretty Good Privacy) sistemini geliştirmiş ve yayınlamıştır.

1995’de SHA–1 (Güvenli Özetleme Algoritması - Secure Hash Algorithm) özet algortiması NIST tarafından standart olarak yayınlanmıştır.

1997’de ABD’nin NIST DES’in yerini alacak bir simetrik algoritma için yarışma açmıştır.

2001’de NIST’in yarışmasını kazanan Belçikalı Joan Daemen ve Vincent Rijmen’e ait Rijndael algoritması, AES adıyla standart haline getirildi.

2005’de Çin’li bir ekip tarafından SHA–1 algoritmasının kırıldığı duyurulmuştu. Buna göre 2^{80} gücündeki algoritma, 2^{63}'e kadar indirilmişti. Bunun üzerine Amerikan Hükümeti ve Microsoft, Sun gibi birçok büyük firma artık kullanmayacaklarını açıklamıştı.

2007’de artık her programcı algoritma geliştirebiliyor. Ve bu algoritmayı programlayabiliyor ve kırılması zor diyor lakin her seferin de kırılıyor.

2009’da kriptografi konusunda dünyanın ilk olimpiyatları Belçika’nın Katholieke Üniversitesinde 25–28 Şubat tarihleri arasında yapılacak olan ön eleme ile başladı. Amaç SH1’lerin kırılmasından sonra yeni bir güvenli algoritma oluşturmaktı.

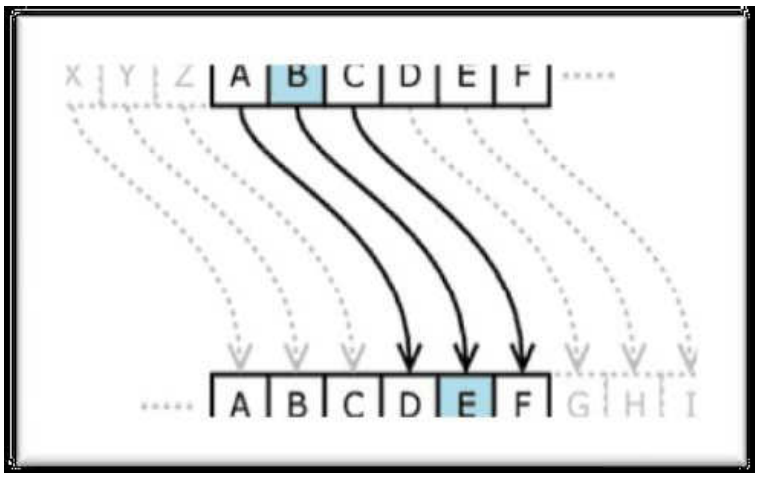
Tarihsel süreci bundan ibarettir. Amerikan hükümeti kriptoloji biliminin ne kadar önemli olduğunu farkında olduğundan dolayı NSA‘yı kurmuştur [6].

Aşağıda geçmişte kullanılıp en çok bilinen kriptolama methodlarından bahsedeceğiz.

2.2.1. Sezar şifreleme algoritması

Tarihin ilk kriptolojik fikirleri ingilizcede transposition and substitution cipher adını taşır, yani yer değiştirme ve harf değiştirme şifrelemesidir. Bu yöntemlerden ilki, bir yazıdaki harflerin yerlerini değiştirerek, ikincisi ise harfleri başka harflerle değiştirerek elde edilir. Bu şifrelemeyi kullanan belki de en ünlü teknik sezar şifresidir. Bu şifrede, her harf o harften birkaç sonraki harf kullanılarak yazılır. Örneğin, 3 harf atlamalı sezar şifresinde *deneme* yerine *ghrhph* yazılır.

Öte yandan, sezar şifresini kırmak kolay olmaktadır. Bir filolog bir dilde en çok geçen harfleri bulabilir. O harfler ile mesajda en sık geçen harfleri karşılaştırarak hangi harfin hangi harf ile değiştirildiği bulunabilir. Bu adımların ardından, mesaj çözülmüş olur. Şekil 2.2.'de sezar şifreleme yöntemi gösterilmiştir [7].

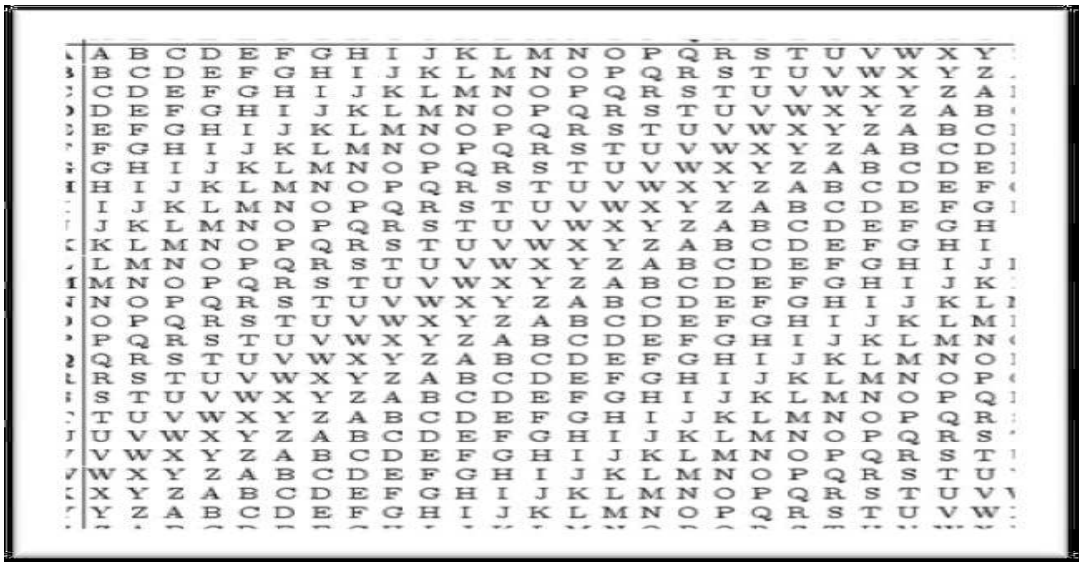


Şekil 2.2. Sezar şifreleme yöntemi

2.2.2. Vigenere şifresi

İlk olarak 1553 yılında Giovan Batista Belasa tarafından tanıtılmış 16. yüzyılın sonlarında Blaise De Vigenere bu yöntemi düzenleyip kullanmıştır ve bu yöntemin adı vigenere şifresi olarak kalmıştır.

Vigenere şifreleme yönteminden önce anlattığımız şifreleme yöntemlerini kısaca yeniden hatırlarsak, şifreler genel olarak bir alfabede yer değiştirme fonksiyonları ile oluşturuluyordu. Buna en basit örnek olarak sezar şifreleme yöntemini verebiliriz. Tabiî bu yer değiştirme algoritmaları ile yazılan şifrelerde olasılık düşüktü ve şifre deneme yanılma yöntemiyle kırılabilecek tarzdaydı. Vigenere yöntemi ile bu olasılık biraz daha büyültülmüştür. Bu yöntemin en büyük özelliği çoklu alfabe kullanmasıdır. Bundan kastımızı da Şekil 2.3.'e bakarak anlayabiliriz.



Şekil 2.3. Vigenere şifreleme yöntemi

Bu tabloya vigenere tablosu denilmektedir. Bu yöntemle şifreleme yaparken vigenere tablosunda periyodik döngüler ile yer değiştirme işlemleri yapılır. Sezar şifreleme yöntemini göz önünde bulundurup tablonun ilk satırına bakarsak arada bir benzerlik göreceğiz. Sezar yöntemine göre burada öteleme sayımız 0, son satıra göre de 25'dir ve ya 26x26'lık bir matris şeklinde de düşünebiliriz. Bunu birinci satırı bir şerit gibi düşünürsek her seferinde bir yana kaymasıyla oluşan tablo bize vigenere tablosunu verir. Böylelikle alfabe ile yer değiştirme arasında bire bir değil de bire çok bir ilişki sağlamış oluruz. Yani 26 karakterlik her harfe karşı bir tane 26 karakterlik

harf şeridi karşılık gelir. Aslında bu şeridi oluşturan harfler hep aynıdır fakat yerleri kaydırılarak değiştirilmiştir [7].

2.2.3. Afin şifreleme yöntemi

Sezar şifreleme yöntemi ile yazı üzerinde 27 farklı dönüşüm yapılabiliyordu. Biz İngiliz alfabesini göz önünde tutarsak 25 harften söz etmemiz gerekir. Bu da simetrik şifreleme yönteminin güvenli olmadığını gösterir. Afin yöntemi ile simetrik şifreleme yöntemi biraz daha genelleştirilmiştir ve güvenlik azda olsa simetrik şifreleme yöntemine göre daha güçlüdür. Tabi bu kâğıt kalem kriptolojisinin bir örneği olduğundan bunu günümüz koşullarına göre düşünürsek çok zayıf bir yöntem olduğunu görürüz. Fakat bu bize kriptolojinin temel mantığını kavratmak için güzel bir örnek teşkil etmektedir.

y= ax+b (mod m) fonksiyonunu göz önüne alalım.

Burada x düz metindeki harflerin sayısal karşılığı, m düz metinde kullanılan alfabenin karakter sayısı, a ve b gizli sayılarımız ve y de fonksiyonumuzun işlem sonucunda aldığı değerdir. y'nin x'e geri dönüşümü ise

x=ters a(y-b) (mod m) formülü yardımıyla hesaplanır.

Ters a, a ile çarpımının mod m'e göre sonucu 1 olan sayıdır. Bunun kısaca şöyle ifade edebiliriz.

a∗ters a (mod m) =1

Farz edelim ki mesaj, y= 11x+4 (mod 26) fonksiyonu ile şifrelensin. Şifreli metnimiz MONEY olsun. Öncelikle düz metnimizdeki her bir karakterin aşağıda verilen listedeki olduğu gibi 0 ile 25 arasındaki sayısal değerlerini bulmalıyız.

A-0 B-1 C-2 D-3 E-4 . . . Y-24 Z-25

Böylece MONEY metnimizin uygun sayısal değerleri 12, 14, 13, 4 ve 24'tür. Buradaki her bir değer için daha önce belirlediğimiz y=11x+4 (mod 26) fonksiyonunu kullanırsak,

M: y = 11∗12+4 (mod 26)=6 ---- G
O: y = 11∗14+4 (mod 26)= 2 --- C
N: y= 11∗13+4 (mod 26)=17 --- R
E: y= 11∗4+4 (mod 26) =22 ---- W
Y: y= 11∗24+4 (mod 26) =8 ----- I

Böylece bulduğumuz şifreli metnimiz GCRWI olur [7].

2.2.4. Enigma

Enigma sözlük literatüründe bilmece anlamına gelmektedir. 1920′li yıllarda devlet kurumları, ordular bilgilerin şifrelenmesi için enigmayı kullanıyorlardı ama pek bir rağbet söz konusu değildi. Enigma daha çok 2. Dünya Savaşı yıllarında Nazi'ler tarafından kullanılmıştır.

Kullanılma sebebi savaşta belirtilen koordinatların, bilgilerin, emirlerin vs. düşmanlar tarafından bilinmemesidir. Enigma makinesi verilerin şifrelenmesi ve deşifre edilmesi için kullanılmıştır. Nazi'ler sayesinde enigma ismini duyurmuştur. Basit bir anlatımla enigma bir şifreleme makinesidir.

Elektromekanik mantığıyla çalışan bir rotor makinedir. Temel olarak, rotor mekanizması sayesinde olasılık üreten bir mekanizmadır. Daktilo klavyesine benzer her bir klavye tuşuna basıldığında, rotorlar döner. Belirgin olarak tüm enigma sistemlerinde öncelikle en sağdaki rotor döner, daha sonra ona komşu olan rotorlar bir veya daha fazla adım atabilir. Rotorun dişli mekanizması her algoritma programlanmadan önce sökülür ve farklı bir konumda takılırdı. Ayrıca her mesaj çekiminden önce operatör tarafından alt bölümdeki elektrik soketlerini farklı şekilde dizerek şifrenin çözümünü daha

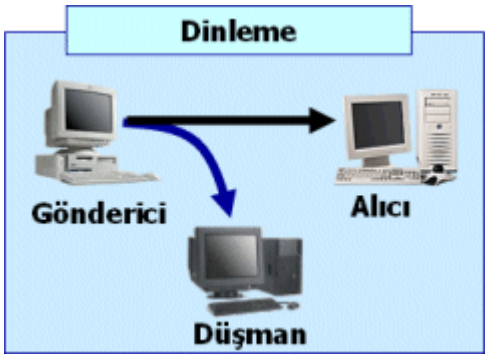
da zorlaştırırdı. Mekanik sisteme bağlı elektrik sistemi, operatöre gösterge bölümünde hangi harfin basıldığını ışıklı olarak gösterirdi [7].

2.3. Bilgi Taşımada Güvenlik Prensipleri

2.3.1. Gizlilik

Bilginin yetkisiz kişilerin eline geçmesinin engellenmesidir. Gizlilik, hem kalıcı ortamlarda (disk, tape, vb.) saklı bulunan veriler, hem de ağ üzerinde bir göndericiden bir alıcıya gönderilen veriler için söz konusudur. Saldırganlar, yetkileri olmayan verilere birçok yolla erişebilirler: Parola dosyalarının çalınması, sosyal mühendislik, bilgisayar başında çalışan bir kullanıcının, ona fark ettirmeden özel bir bilgisini ele geçirmesidir. Bunun yanında trafik analizinin, yani hangi gönderici ile hangi alıcı arası haberleşmenin olduğunun belirlenmesine karşı alınan önlemler de gizlilik hizmeti çerçevesinde değerlendirilir. Şekil 2.4. ve Şekil 2.5.'te veri transferindeki gizlilik gösterilmiştir[1,28].

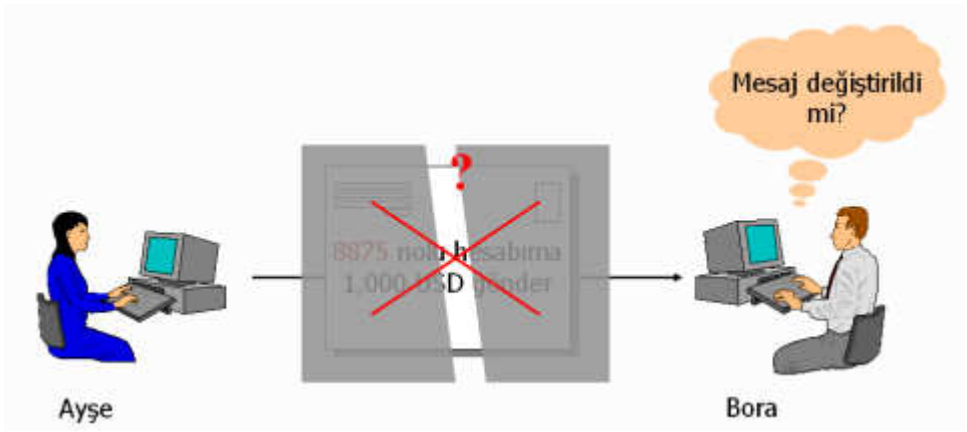
Şekil 2.4. Veri transferinde gizlilik



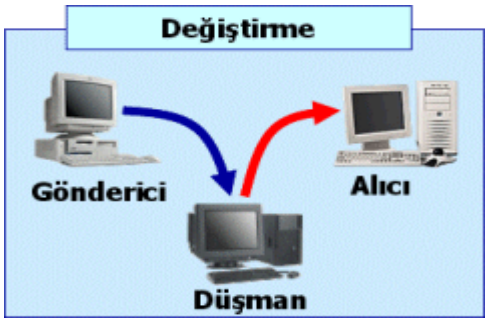
Şekil 2.5. İzinsiz erişim

2.3.2. Veri bütünlüğü

Bu hizmetin amacı, veriyi göndericiden çıktığı haliyle alıcısına ulaştırmaktır. Bu durumda veri, haberleşme sırasında izlediği yollarda değiştirilmemiş, araya yeni veriler eklenmemiş, belli bir kısmı ya da tamamı tekrar edilmemiş ve sırası değiştirilmemiş şekilde alıcısına ulaşır. Bu hizmeti, geri dönüşümü olan ve olmayan şekilde verebiliriz. Şöyle ki; alıcıda iki tür bütünlük sınaması yapılabilir: bozulma sınaması ya da düzeltme sınaması. Bozulma sınaması ile verinin göndericiden alıcıya ulaştırılması sırasında değiştirilip değiştirilmediğinin sezilmesi hedeflenmiştir. Düzeltme sınamasında ise, bozulma sınamasına ek olarak eğer veride değişiklik sezildiyse bunu göndericiden çıktığı haline döndürmek hedeflenmektedir. Şekil 2.6. ve Şekil 2.7.'de veri bütünlüğü gösterilmiştir [1,28].



Şekil 2.6. Veri bütünlüğü

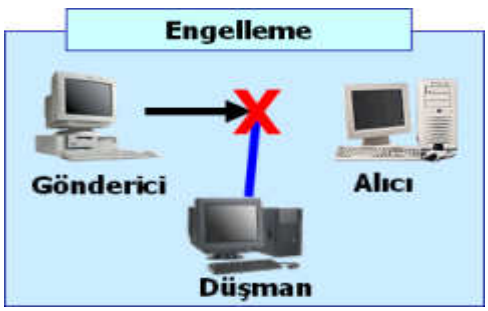


Şekil 2.7. Veri değişikliği

2.3.3. Süreklilik

Bilişim sistemleri, kendilerinden beklenen işleri gerçekleştirirken, hedeflenen bir başarım vardır. Bu başarım sayesinde müşteri memnuniyeti artar, elektronik işe geçiş süreci hızlanır. Süreklilik hizmeti, bilişim sistemlerini, kurum içinden ve dışından gelebilecek başarım düşürücü tehditlere karşı korumayı hedefler. Süreklilik hizmeti sayesinde, kullanıcılar, erişim yetkileri dâhilinde olan verilere, veri tazeliğini yitirmeden, zamanında ve güvenilir bir şekilde ulaşabilirler.

Sistem sürekliliği, yalnızca kötü amaçlı bir Hacker'in, sistem başarımını düşürmeye yönelik bir saldırısı sonucu zedelenmez. Bilgisayar yazılımlarındaki hatalar, sistemin yanlış, bilinçsiz ve eğitimsiz personel tarafından kullanılması, ortam şartlarındaki değişimler (nem, ısı, yıldırım düşmesi, topraklama eksikliği) gibi faktörler de sistem sürekliliğini etkileyebilir. Şekil 2.8.'de engelleme gösterilmiştir [1,28].



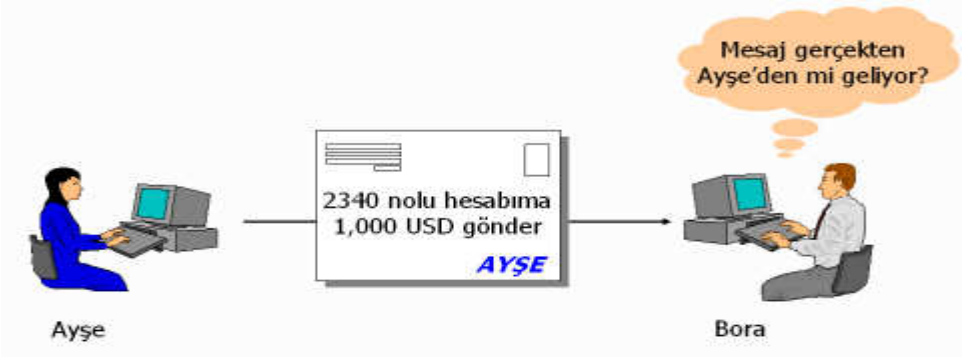
Şekil 2.8. Engelleme

2.3.4. İzlenebilirlik

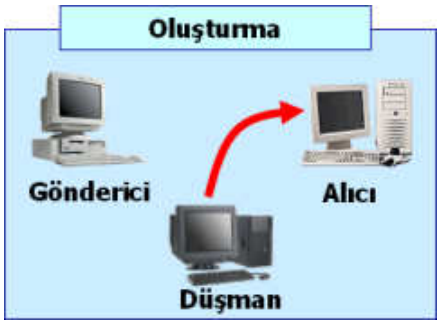
Bu hizmetin hedefi sistemde gerçekleşen olayları, daha sonra analiz edilmek üzere kayıt altına almaktır. Burada olay dendiğinde, bilgisayar sistemi ya da ağı üzerinde olan herhangi bir faaliyeti anlayabiliriz. Bir sistemde olabilecek olaylara, kullanıcının parolasını yazarak sisteme girmesi, bir web sayfasına bağlanmak, e-posta almak göndermek ya da icq ile mesaj yollamak gibi örnekler verilebilir. Toplanan olay kayıtları üzerinde yapılacak analiz sonucunda, bilinen saldırı türlerinin örüntülerine rastlanırsa ya da bulanık mantık kullanılarak önce rastlanmayan ve saldırı olasılığı yüksek bir aktivite tespit edilirse alarm mesajları üretilerek sistem yöneticileri uyarılır [1,28].

2.3.5. Kimlik sınaması

Ağ güvenliği açısından kimlik sınaması; alıcının, göndericinin iddia ettiği kişi olduğundan emin olmasıdır. Bunun yanında, bir bilgisayar programını kullanırken bir parola girmek de kimlik sınaması çerçevesinde değerlendirilebilir. Günümüzde kimlik sınaması, sadece bilgisayar ağları ve sistemleri için değil, fiziksel sistemler için de çok önemli bir hizmet haline gelmiştir. Akıllı karta ya da biyometrik teknolojilere dayalı kimlik sınama sistemleri yaygın olarak kullanmaya başlanmıştır. Şekil 2.9. ve Şekil 2.10.'da kimlik sınaması gösterilmiştir [1,28].



Şekil 2.9. Kimlik sınaması



Şekil 2.10. Veri üretimi

2.3.6. Güvenilirlik

Sistemin beklenen davranışı ile elde edilen sonuçlar arasındaki tutarlılık durumudur. Başka bir deyiş ile güvenilirlik, sistemden ne yapmasını bekliyorsak, sistemin de eksiksiz ve fazlasız olarak bunu yapması ve her çalıştırıldığında da aynı şekilde davranması olarak tanımlanabilir [1,28].

2.3.7. İnkâr edememe

Bu hizmet sayesinde, ne gönderici alıcıya bir mesajı gönderdiğini ne de alıcı göndericiden bir mesajı aldığını inkâr edebilir. Bu hizmet, özellikle gerçek zamanlı işlem gerektiren finansal sistemlerde kullanım alanı bulmaktadır ve gönderici ile alıcı arasında ortaya çıkabilecek anlaşmazlıkların en aza indirilmesini sağlamaya yardımcı olmaktadır.

Bu hizmetler, zaman içinde bilgisayar sistemlerine karşı ortaya çıkmış tehditler ve yaşanmış olaylar sonucunda ortaya konmuştur. Yani her bir hizmet, belli bir grup potansiyel tehdide karşı sistemi korumaya yöneliktir, denilebilir. Şekil 2.11.'de inkar edememe gösterilmiştir [1,28].

Şekil 2.11. İnkâr edememe

2.4. Kripto Analiz Çeşitleri

Kripto analiz; bir şifreleme sistemini veya sadece şifreli mesajı inceleyerek, şifreli mesajın açık halini elde etmeye çalışan kriptoloji disiplinidir.

2.4.1. Lineer kripto analiz

1993 yılında Matsui tarafından teorik bir saldırı olarak keşfedilmiştir. Daha sonra DES algoritmasına karşı başarı ile uygulanmıştır. Lineer kripto analiz, şifreli metin bitleri ile, açık metin bitleri arasındaki yüksek olasılıkta lineer ifadelerin meydana gelme avantajını kullanır. Bunun yolu da S kutularından geçer. Saldırganın algoritmayı bildiği ve belli sayıda açık metin ve şifreli metinlere sahip olduğu varsayılır. S kutularının büyük olması, aktif S kutularının sayısının artışı, lineer sapması ve küçük S kutularının tasarımı lineer kripto analizin uygulanmasını engelleyici faktörlerdir [8].

2.4.2. Diferansiyel kripto analiz

Diferansiyel kripto analiz açık metin çiftlerindeki özel farkların sonuçlanan şifreli metinlerde oluşturduğu farkın etkisini analiz eder. Bu farklar mümkün olan anahtarların olasılıklarını ve en yüksek mümkün anahtarı ortaya koymak için tayin edilir [9].

2.4.3. İmkânsız diferansiyel kripto analiz

Kesik diferansiyel kripto analizin bir çeşididir. Önceden bir diferansiyel öngörülür ve buna göre bazı özel farklar asla meydana gelmeyecektir. Kripto analizde imkânsız durumların kullanılabileceği gerçeği eski bir fikirdir. Ortada ıskalama saldırısı ya da imkânsız diferansiyel saldırısı olarak isimlendirilen bu saldırılar bir blok şifrede imkânsız bir davranışın nasıl belirleneceği ve bunun nasıl anahtarı elde etmek için kullanılacağı ile ilişkili sistematik analizdir [10].

2.4.4. Çoklu set saldırıları (square saldırısı)

Çoklu set saldırıları ilk defa J. Daemen, V. Rijmen ve L. Knudsen, square algoritmasını ortaya koyduklarında öne sürülmüştür. Dolayısıyla diğer ismi square (kare) saldırısı olarak bilinir. O zamandan beri diğer birçok algoritmaya uygulanmıştır. Seçilmiş açık metin saldırısıdır ve iyi seçilmiş açık metin setleri ile şifrenin ileri doğru incelenmesiyle gerçekleştirilir. Bu saldırı tipinde lineer ve diferansiyel kripto analizden farklı olarak açık metin grubunun tümü düşünülerek şifre hakkında bilgi toplanır [11].

3. KRİPTO SİSTEMLERİ

3.1. Açık Anahtarlı(Asimetrik) Sistemler

3.1.1. Açık anahtarlı sistemlerin özellikleri

Gizli anahtarlı kripto sistemlerinin aksine açık anahtarlı kripto sistemlerinin kullanımı henüz çok yenidir. Açık anahtarlı kripto sistemleri üzerine ilk öneri,1976 yılında Diffie ve Hellman tarafından yapılmıştır. Ardından 1977 yılında Rivest, Shamir ve Adleman RSA Kripto sistemi adlı yeni bir açık anahtarlı kripto sistemini bulmuşlardır. 1978 yılından beri kripto dünyasına değişik teklifler yapıla gelmiştir. Bunlardan en önemlileri El-Gamal tarafından tasarlanan El-Gamal açık anahtarlı kripto algoritması ve eliptik eğri açık anahtarlı kripto sistemleridir. Temelde açık anahtarlı kripto sistemlerinin gayesi belli bir anahtar üzerinde anlaşmanın ve karşı tarafa bu anahtarı güvenli olarak ulaştırabilmenin zorluğunu ortadan kaldırmaktır. Burada tek yönlü bir mesajlaşma söz konusudur. Mesaj alıcısı sadece kendisinin bileceği gizli anahtar ve diğer kişilere dağıtabileceği bir açık anahtardan oluşan anahtar çifti belirler. Kullanılan anahtar üretim algoritmasına göre bu iki anahtar arasında matematiksel bir bağlantı mutlaka olabilecektir fakat asıl amaç, bilinen açık anahtardan gizli anahtarın hesaplanmasının polinomsal zamanda imkânsız olabilmesidir.

Mesaj göndericisi alıcıya ait herkesçe bilinen açık anahtarı kullanarak açık anahtarlı kripto algoritmasıyla göndereceği mesajı kapatır ve alıcıya gönderir, mesajın alıcısı ise yalnız kendisinin bildiği gizli anahtar ile deşifreleme algoritmasını kullanarak mesajı açabilir. Gizli anahtar yalnız alıcı tarafından bilindiği için başka birinin bu mesajı açması mümkün olamayacaktır. Gizli anahtarın açık anahtardan polinomsal zamanda türetilmesini imkânsız kılmak için Diffie ve Hellman'ın tek yönlü fonksiyon mantığı üzerine kurulu anahtar değişim protokolü vardır [12].

3.1.2. Açık anahtarlı sistemlerin avantajları

Bilgisayar bilim ve teknolojisinin eriştiği yüksek düzey göz önüne alındığında, simetrik kripto sistemlerin mutlak biçimde korumak zorunda oldukları anahtarların koruma ve dağıtım maliyetinin ne kadar yüksek ve koruma işleminin ne kadar zor olduğu kolayca görülebilir. Sırf bu nedenden ötürü, karşılıklı haberleşme içinde olan iki tarafın güvenli dağıtım kanalları oluşturması özellikle güncel bankacılık sisteminde, yaygın görülen bir örnektir.

Öte yandan, şifreleme ve deşifreleme dönüşüm fonksiyonlarının kullandıkları anahtarlar birbirinden ayrılarak anahtar güvenliği sorunu kesin biçimde çözülebilir. Asimetrik kripto sistemlerin tüm güvenliği deşifre anahtarının yalnız ve yalnız yetkili alıcı tarafından bilinmesinde yatar. Öte yandan, her ne kadar her iki anahtar birbirinden farklıysa da şifreleme anahtarından gidilerek deşifre anahtarını oluşturmak, teorik olarak olası, ancak pratikte çözümsüz bir problemdir.

Asimetrik kripto sistemlerin, simetrik kripto sistemlere göre önemli yararı da anahtar yönetimidir. Gerçekçi olunursa, n kişinin özel anahtar şifrelemesini kullanması durumunda grup içerisinde her kişi için bir farklı özel anahtar ihtiyacı olmaktadır. Böylece n(n–1) adet anahtar yönetimi olacaktır. Eğer n binlerce kullanıcı olursa o zaman milyonlarca anahtar yönetimi söz konusudur. Bundan başka gruba yeni bir kullanıcı eklemesi kolay bir iş olmamaktadır, yeni kullanıcının gruptaki herkesle iletişim kurabilmesi için n adet yeni anahtar yönetimi söz konusu olacaktır. Daha sonra, yeni anahtarların gruba yollanması gerekmektedir. Aksine, asimetrik sistemlerde, kullanıcıların n adet genel anahtarları genel bir dizinde tutulur, yeni bir kullanıcı eklemesinde kullanıcının genel anahtarı dizine eklenmesi yeterli olacaktır.

Asimetrik kripto sistemlerin en büyük avantajı, veri şifreleme ve deşifreleme işlemlerini yapmasının yanında dijital imza gerçekleştirilmede kullanılmasıdır. Aldığımız bir şifreli metinde deşifreme işlemini gerçekleştirdikten sonra karşılaşacağımız en büyük problem bu şifreli metnin bize doğru kişiden gelip gelmediğidir. Bunu da dijital imzalar sayesinde doğrulamaktayız.

Bireyleri tanımlama günümüz iletişiminde çok önemlidir. Konuştuğumuz kişinin bizi aldatmadığından ve gerçek kişiyi taklit etmediğinden emin olmamız gerekir. Bundan dolayı tanımlayıcı bir protokol kullanır. Birçok sayıda tanımlayıcı protokol vardır ve genellikle RSA veya ayrık logaritma prensiplerine bağlıdırlar. Bununla birlikte doküman doğruluğu da gereklidir. Dijital imza sayesinde bir otorite dokümanların doğruluğu denetler. İmzalama, doküman ile otoritenin girişinin belirli işlemleri ile oluşan bir kaç bitin veya MD5 (Mesaj Özetleme Algoritması 5 - Message Digest Algorithm 5) ile SHA benzeri karıştırma algoritmalarının kullanımıyla oluşan bitlerin dokümana eklenmesiyle elde edilir. Dokümana erişim yetkisi olan herhangi bir kişi, imzanın gerçekten otorite tarafından atıldığını doğrulayabilir. Bu iş için imza şemaları kullanılır. İmza şemalarının en ünlüsü El-Gamal imza şemasıdır [13, 14].

3.1.3. Açık anahtar tabanlı şifreleme algoritmalarında hız ve anahtar dağıtım sorunları

Açık anahtar tabanlı şifreleme algoritmaları ile yapılan işlemler (şifreleme, deşifreleme, sayısal imzalama ve imza doğrulama işlemleri) yavaş işlemlerdir. Kullanılan algoritma, anahtar uzunluğu ve uygulamanın koştuğu platform işlemlerin hızını belirleyen önemli faktörlerdendir. Ancak her ne şart altında olursa olsun, tek anahtarlı simetrik algoritmalar onlarca, hatta bazı durumlarda yüzlerce kat daha hızlıdır. Buna rağmen gerek sunduğu kripto analiz direnci, gerekse de anahtar dağıtım kolaylıkları açısından açık anahtar tabanlı algoritmalar tercih edilir.

İşlem süresini azaltmak için standartlaşmış hızlandırıcı mekanizmalar kullanılır. Sayısal imzalamada metnin kendisi değil de tek blok halinde özü imzalanır. Şifrelemede metin daha hızlı olan bir simetrik şifreleme algoritması ile şifrelenmekte, bu şifrelemede kullanılan anahtar ise açık anahtar tabanlı bir algoritma ile şifrelenir.

Böylelikle hem açık anahtar tabanlı sistemlerin hız sorunu kısmen de olsa aşılmış olur, hem de anahtar dağıtım avantajlarından yararlanılır.

Asimetrik şifreleme algoritmalarında iki ayrı anahtar kullanılır. Bu anahtarlar arasında bir matematiksel bağıntı bulunduğu halde açık anahtardan gizli anahtarı üretmek pratik olarak mümkün değildir. Bu durumda anahtar dağıtım sorununun ortadan kalktığı düşünülebilir. Gerçekten de simetrik anahtar tabanlı sistemlere göre anahtar dağıtım sorunu daha azdır, ama sanılanın aksine ortadan kalkmamıştır. Açık anahtarlar herkese dağıtılabilir, ancak hangi anahtarın kime ait olduğundan da emin olunmalıdır. Bu konuda kişisel beyanlara güvenilemez. O yüzden sertifikalar kullanılır. Sertifika, en basit anlatımı ile bir açık anahtar ile sahibinin kimliği arasındaki bağıntının belgesidir. Güvenilir SO (Sertifika Otoriteleri) tarafından üretilen bu sertifikaların içerdiği bilgilerin doğru olduğu varsayılır. Sertifika otoriteleri sayısal imzalarını kullanarak sertifika üretirler. Sertifika otoritesinin imzasını doğrulayan kişi aynı zamanda sertifika sahibinin açık anahtarını da öğrenmiş olur [15,16].

3.1.4. En çok kullanılan açık anahtarlı şifreleme sistemleri

Diffie-Hellman anahtar değişimi

İnsanlığa ilk duyurulan açık anahtar algoritması, Diffie ve Hellman`ın açık anahtarlı kriptografi olarak tanımladıkları 1976 yılında yayımlanmış "New Directions in Cryptography" isimli makalelerinde yer aldı ve bu algoritma

kriptografik sisteme örnek olarak Diffie-Hellman anahtar değişimi olarak geçti. Birçok ticari uygulama bu anahtar değişimini kullandı.

Algoritmanın amacı, iki kullanıcının bir anahtarı güvenli şekilde birbirlerine iletmeleri ve daha sonrasında da bu anahtar yardımı ile şifreli mesajları birbirlerine gönderebilmelerini sağlamaktı. Algoritma anahtar değişimi ile sınırlıdır.

Diffie-Hellman ortak gizli anahtar oluşturma sistemi ayrık logaritma problemini üzerine kurulmuş ve güvenirliği çok büyük asal sayıları seçmeye dayanır.

p yeteri kadar büyük bir asal sayı olsun öyle ki Zp de ayrık logaritma problemini çözmek mümkün olmasın. g'de Zp de primitif bir kök olsun. p ve g herkes tarafından bilinsin. A ve B kişileri aşağıdaki yolu izleyerek ortak bir anahtar yaratabilirler:

1. A, $0 \le a \le p-2$ eşitsizliğini sağlayan ve tesadüfî olan bir a sayısı seçer.
   $C = g_a \pmod p$'yı hesaplar ve bunu B'ye gönderir.
2. B, $0 \le b \le p-2$ eşitsizliğini sağlayan ve tesadüfî olan bir b sayısı seçer.
   $d = g_b \pmod p$'yı hesaplar ve bunu A'ya gönderir.
3. A, ortak anahtar K' yı şu şekilde hesaplar:
   $K = d_a = (g_b)_a$
4. B, ortak anahtar K' yı şu şekilde hesaplar:
   $K = c_b = (g_a)_b$

Böylelikle A ve B aralarında ortak bir anahtar olan K için anlaşmış olurlar [15,17].

El-Gamal şifreleme sistemi

El-Gamal ayrık logaritma problemlerine dayalı bir kripto sistemdir.

$Y=g^{x} \pmod p$; $p \in Z^{+}$, p asal sayı, koşuluyla verilen denklemde, x sayısının bulunmasına dayanır. El-Gamal tarafından 1985 yılında formalize edilen bu kripto sistemin ayrıntıları aşağıdaki gibidir:

A ve B kullanıcıları ortak bir anahtar oluşturmak istediklerinde:

X_a ve X_b, A ve B kullanıcılarınca bilinen gizli ifadeler, p büyük asal bir sayı ve a (mod p)'nin ilkel elementi olsun.

Bu durumda:
$y_A = a^{XA} \pmod p$ eşitliği A tarafından ve
$y_B = a^{XB} \pmod p$ eşitliği B tarafından bulunarak, değiş tokuş edilir.

Bu aşamada ortak gizli anahtar:

$$K_{AB} = a^{XA\,XB} \pmod p$$
$$= yA^{XB}$$
$$= yB^{XA}$$

olarak saptanır. Şimdi A'dan B'ye şifreli bir mesaj gönderilecekse:

A, 0 ile (p–1) arasında bir k sayısı seçerek,
$K = y_B^{k} \pmod p$ denklemiyle anahtarı oluşturur.
$Y_B = a \pmod p$, B açık şifreleme anahtarı olup, B tarafından A'ya daha önceden gönderilmiştir.

Bu iki ifade kullanılarak şifreli metin c1 ve c2 ikilisi halinde aşağıdaki gibi oluşturulur.

$c1 = a^{k} \pmod p$ ve $c2 = K_m \pmod p$

Burada dikkat edilmesi gereken nokta, kripto gram uzunluğunun, açık metin uzunluğunun iki katına çıkmış olmasıdır. B'ye varan mesaj burada iki aşamada deşifre edilir:

Birinci aşama anahtar K eldesi olup;

$K = (a^{k})^{X_B} = c^{1X_B} \pmod p$ ile gerçekleştirir.

İkinci aşamada mesaj m = (c2 / K) ile elde edilir [13].

RSA şifreleme sistemi

RSA asimetrik şifreleme algoritması 1977 yılında R. Rivest, A. Shamir ve L. Adleman tarafından bulunmuş ve daha sonra asimetrik şifreleme algoritmasına uygun biçimde geliştirilmiştir. Bu algoritma, asimetrik şifreleme sistemlerinde ve sayısal imza işlemlerinde güvenli bir şekilde kullanılır.

Günümüzde RSA şifreleme algoritması hala güvenilirliğini korumaktadır. Bunun nedeni modüler matematik üstüne kurulmuş, kripto analizi asal sayılara çarpanlara ayırmaya dayalı anlaşılması kolay, ama çözülmesi zor bir algoritma olmasıdır.

Öncelikle p ve q olmak üzere iki tane asal sayı üretilir. Bunların birbirleriyle çarpılmasıyla n=p∗q 'dan n elde edilir. Bundan sonra n sayısından küçük ve (p–1)∗(q–1) sayısıyla 1 dışında herhangi bir ortak böleni bulunmayan bir e sayısı seçilir.

Daha sonra (E∗D=1) sayısının (p–1)∗(q–1) çarpımına tam olarak bölünmesini sağlayan bir D sayısı bulunur. E ve D değerleri, sırasıyla, açık ve gizli anahtar olarak adlandırılırlar. Açık anahtarı n, E çifti, gizli anahtarı ise n, D çifti oluşturur. p ve q sayıları ya yok edilmeli, ya da gizli anahtar ile birlikte saklanmalıdır.

Gizli anahtar olan D sayısının n, E sayılarından elde edilmesi zor bir işlemdir. Eğer bir kişi n sayısını çarpanlarına ayırarak p ve q sayılarını elde edebilirse gizli anahtarı da kolaylıkla bulabilir. Bu sebeple RSA sisteminin güvenliği çarpanlarına ayırma probleminin zorluğu temeline dayanır. Çarpanlarına ayırma işleminin kolay bir yönteminin bulunması, RSA algoritmasının kırılması anlamına gelir.

RSA sisteminin kırılması birkaç değişik şekilde yorumlanabilir. Sisteme en çok zarar verecek saldırı bir kripto analistin belli bir açık anahtara karşı gelen gizli anahtarı bulmasıdır. Bunu başarabilen bir hasım hem şifrelenen bütün mesajları okuyabilir, hem de imzaları taklit edebilir. Bunu yapmanın en akla gelen yolu n'nin asal çarpanlara ayrılması yani p ve q'nun hesaplanmasıdır. p, q ve açık üs e kullanılarak d kolaylıkla hesaplanabilir. Ancak buradaki zorluk n modülünün çarpanlarına ayrılmasıdır. RSA sisteminin güvenliği çok büyük sayıların asal çarpanlarına ayrılmasının zorluğu varsayımına dayanır. Büyük sayıların çarpanlarına ayrılmasının zorluğu ispatlanmış değildir. Son üç yüzyıl içerisinde Fermat ve Legendre gibi ünlü matematikçiler bu konuda çalışmalar yapmışlardır [13].

<u>Eliptik Eğri Şifreleme Algoritması</u>

Eliptik eğri gruplarının sonlu alanlar üzerinde tanımlanmasının kripto sistem için temel alınması ilk olarak Neal Koblitz ve Victor Miller tarafından 1985' te önerilmiştir. Eliptik eğri yaklaşımı standart RSA sisteminden daha zengin matematiksel prosedürler içerir. Bu kripto sistemin temel birimleri eliptik eğri üzerindeki (x,y) noktalarıdır.

$Y^2 = x^3 + ax + b$ birlikte

$x, y, a, b \in Ifp = \{1, 2, 3, \ldots, p-2, p-1\}$

ECC algoritması ECDLP (Eliptik Eğri Ayrık Logaritma Problemi) problemi üstüne kurulmuş bir şifreleme sistemidir.

P+P+P+…....+P=NP=Q

P ve Q biliniyor olsa da n'yi hesaplamak oldukça zordur. Bu ayrık logaritma problemi birçok EC (Eliptik Eğri - Elliptic Curve) ile birleşip güvenlik sistemi olan ECC'nin temelini oluşturur.

ECC şifreleme algoritmasının en büyük özelliği diğer açık anahtar şifreleme sitemlerinin güvenliğini daha düşük anahtar değerleriyle sağlayabilmesidir. 1024 bitlik anahtar kullanan RSA şifreleme algoritmasının sağladığı güvenlik gücünü, 160 bit anahtar kullanan ECC sağlayabilmektedir. Bu açık anahtarlı algoritmalar içinde çok önemli bir avantajdır.

Yeni gelişen teknolojiyle birlikte kablosuz ağların kullanımı geniş anahtar değerlerine sahip şifreleme algoritmalarının kullanımını zorlaştırmıştır. ECC daha düşük anahtar değerlerini kullanması ve aynı güvenlik seviyesini sağlaması sayesinde kablosuz ağlarda kullanımına çok uygundur [13].

PGP

PGP güvenli bir e-posta yazılımıdır. PGP' nin başarısının ardında iki önemli etken vardır. Birincisi ücretsiz bir yazılım olması, ikincisi ise güçlü şifreleme algoritmaları içermesidir. PGP bu özelliklerini kullanarak hatırı sayılır bir kullanıcı kitlesine ulaşmıştır. Aslında kullanımı kolay bir yazılım değildir. Kendine has ve oldukça karmaşık bir güven ve sertifika modeli içerir. Sistemin yararı, isteyenin kendi güven sistemini istediği şekilde oluşturmasındadır. Ancak bunu yapabilmek için kullanıcının şifreleme ve güvenlik konularında az da olsa bilgi sahibi olması gerekir. Karmaşıklığına rağmen PGP, ücretsiz ve açık kaynak kodlu olma özelliklerini çok iyi kullanarak güvenlik konusunda hassas bir kullanıcı kitlesini kendine bağımlı kılmayı bilmiştir [16].

SSL (Soket Düzeyi Güvenlik - Secure Sockets Layer)

SSL, PGP' nin aksine, kişisel değil organize bir çaba sonucu gelişmiş bir endüstri standardıdır. Güvenli http (Hipermetin Aktarma İletişim Kuralı - Hypertext Transfer Protocol) bağlantısı sağladığı ve özellikle internet tarayıcı programlar tarafından desteklendiği için büyük bir kullanıcı kitlesine ulaşmıştır. Güvenli bir http bağlantısı kurulumu sırasında yaşanan mesaj alışverişi, son kullanıcıya saydam bir şekilde gelişmekte, kullanıcı olan biteni farkına bile varmamaktadır. Kök SO sertifikaları da yazılımla beraber geldiğinden kullanıcının herhangi bir anahtar girme zorunluluğu yoktur. Bu kolaylıklar, bazı güvenlik detaylarının gözden kaçmasına sebep olmakla beraber, son derece önemli kullanım kolaylıkları sağlamaktadır [16].

SSH (Güvenli Kabuk - Secure Shell)

Daha çok telnet ve ftp (Dosya transferi protokolü - File Transfer Protocol) gibi uzaktan erişim protokolleri yerine kullanılan ve sunucu ile istemci arasındaki iletişimi şifrelemeye yarayan SSH protokolü aslında her zaman sertifika gerektirmemektedir. İstemci, sunucuya ilk bağlantı sırasında sunucunun gönderdiği açık anahtarı çevrim dışı yollarla doğrulayıp listesine ekleyebilir. Böylelikle sertifika gerektirmeden sunucunun açık anahtarı istemci tarafından öğrenilmiş olur. Bu işlem bir seferlik bir işlemdir. Kaldı ki SSH sisteminin kullanım amacı sunucuda hesabı bulunan kısıtlı sayıdaki kullanıcıya hizmet vermektir; her internet kullanıcısı sunucunun SSH açık anahtarına ihtiyaç duymaz. O yüzdendir ki PKI (Açık Anahtar Altyapısı - Public Key Infrastructure) benzeri bir yapı çok da gerekli değildir. Ancak SSH, sertifika ve PKI sistemlerini de desteklemektedir. SSL örneğinde olduğu gibi kök sertifikalar yazılımla beraber gelmektedir. Ayrıca çevrim içi sertifika dizinlerine erişim desteği de içermektedir [16].

3.2. Gizli Anahtarlı (Simetrik) Sistemler

Simetrik kripto sistemlerde, şifreleme anahtarının deşifreleme anahtarından üretilmesi oldukça kolaydır. Çoğu simetrik algoritmalarda, şifreleme ve deşifreleme için kullanılan anahtar aynıdır. Bu algoritmaların güvenliği anahtara bağlı olduğundan anahtar gizli tutulmalıdır.

Simetrik anahtarlı şifreleme yöntemleri, yerine koyma, yer değiştirme ve cebirsel yöntemler olarak sınıflandırılabilir. Yerine koyma şifreleme yönteminde orijinal mesajın harfleri başka bir alfabenin harfleriyle değiştirilir. Açık metindeki harflerin ardışıklığı değişmez. Yer değişme şifreleme yönteminde açık metin harflerinde değişiklik olmaz. Harflerin yeniden düzenlenmesi için çeşitli yöntemler mevcuttur. Cebirsel şifreleme yöntemleri ise iki aşamadan oluşmaktadır. Birinci aşamada, açık metne yerine koyma şifreleme yöntemi, ikinci aşamada ise rasgele bir yer değişme şifreleme yöntemi uygulanır.

Simetrik algoritmaların özellikleri;

1. Aynı şifreleme algoritmaları kullanır.
2. Algoritmalar birbirine uyumludur.
3. Anahtarlar aynıdır.

Simetrik algoritmaların avantajları;

1. Algoritmalar olabildiğince hızlıdır.
2. Donanımla birlikte kullanılır.
3. Güvenlidir.

Simetrik algoritmaların dezavantajları;

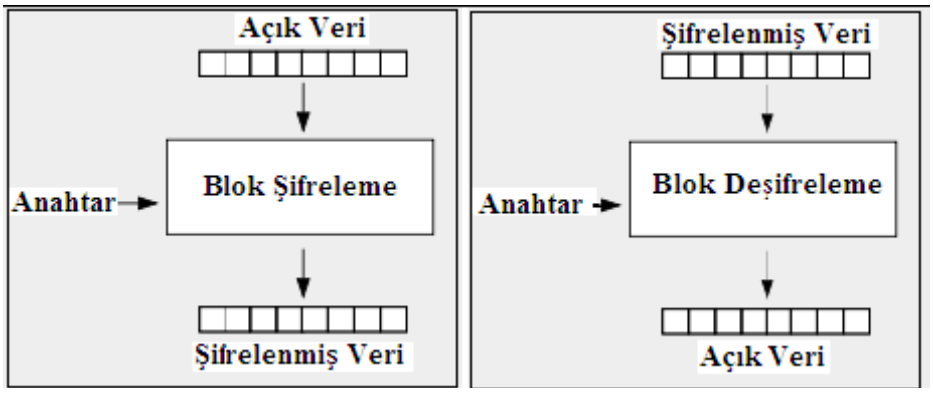
1. Güvenli anahtar dağıtımı zordur.
2. Kapasite sorunu vardır.

3. Kimlik doğrulama ve bütünlük ilkeleri hizmetlerini güvenli bir şekilde gerçekleştirmek zordur [7].

Simetrik algoritmalar blok şifreleme ve dizi şifreleme algoritmaları olarak ikiye ayrılır.

3.2.1. Blok şifreleme algoritması

Blok şifreler, Shannon'un önerdiği karıştırma ve yayılma tekniklerine dayanır. Karıştırma şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılma açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır. Karıştırma ve yayılma, sırasıyla yer değiştirme ve lineer transformasyon işlemleri ile gerçeklenir. Feistel ağları ve yer değiştirme-permütasyon ağları olmak üzere iki ana blok şifreleme mimarisi vardır. Her ikisi de yer değiştirme ve lineer transformasyonu kullanır. Ayrıca her iki mimari ürün şifrelerinin örneklerindendir. Yani birden fazla şifreleme işleminin birleşmesi ile oluşturulurlar. Tekrarlanan şifreler yine ürün şifreleridir ve aynı şifreleme adımının tekrarlanan uygulamasını içerir ve her şifreleme adımına döngü denir. Bir döngü birden fazla şifreleme adımı içerebilir. Genellikle her döngüde farklı anahtar materyali kullanılır. Şekil 3.1.'de blok şifrelemeye örnek gösterilmiştir [3].



Şekil 3.1. Blok şifreleme

Blok Şifreleme Algoritmalarının Önemli Özellikleri

- *Anahtar*

Blok şifreleme algoritmalarında anahtarın uzunluğu ya da bit sayısı en temel saldırı olan geniş anahtar arama saldırısına karşın güçlü olmalıdır. Örneğin DES algoritması 56 bit anahtar kullanırken AES algoritması DES'in bu zaafını örter niteliktedir ve 128, 192, 256 bit anahtar seçenekleri mevcuttur. Ayrıca anahtarın rastlantısal olması gerekir.

- *Döngü sayısı*

Blok şifreleme algoritmalarında döngü sayısı iyi seçilmek zorundadır. Çünkü lineer transformasyon ve yer değiştirmelerin bu seçilen değerle algoritmaya yeterli gücü vermesi gerekmektedir. Ayrıca yapılan saldırıların başarısız olması için en önemli şartlardan biridir. Bu sayı için herhangi bir teorik hesaplama olmamasına rağmen Lars Knudsen'e göre kabaca döngü sayısı

r >= dn/w

gibi olmalıdır. Burada r döngü sayısını, d yer değiştirme durumuna bir kelimeyi almak için gerekli maksimum döngü sayısını, n blok genişliğini, w ise tüm şifrede yer değiştirme durumuna giriş olan minimum kelime genişliğini temsil eder.

- *S kutuları*

S kutuları bir blok şifreleme algoritmasının en önemli ana elemanıdır. Çünkü algoritmadaki tek non-lineer yapıdır ve dolayısıyla algoritmaya gücünü verir. S kutuları için üç önemli nokta vardır. Bunların belirlenmesinde lineer kripto analiz, diferansiyel kripto analiz, Davies saldırıları etkili olur. Bunlar;

SAC (Strict Avalance Criteria); 1 bit giriş değişimi sonucunda her çıkış bitinin değişme olasılığı 1/2 olur.

S kutularının genişliği; Kripto analiz saldırıları düşünüldüğünde büyük bir kutu küçüğüne oranla daha iyi olacaktır. Ayrıca diferansiyel saldırılardan korunmak için büyük sayıda çıkış bitleri ve lineer saldırılardan korunmak için büyük sayıda giriş bitleri gereklidir.

S kutusu gereksinimleri; çıkışların dağılımları Davies saldırısına karşın kontrol edilmeli, çıkışlar girişe göre lineer olmamalı, S kutusunun her sırasındaki değerler tek olmalıdır. Daha güçlü S kutuları yaratmak için çeşitli çalışmalar da yapılmıştır [3,18].

Blok şifreleme algoritmalarına yönelik saldırı tipleri

- *Temel saldırılar*

Blok uzunluğu n bit olan ve k bit anahtar uzunluğuna sahip bir blok şifresi için en temel saldırılardan biri sözlük saldırısıdır. Bu saldırıda k bitlik anahtarı kullanan saldırgan bir açık metni mümkün 2^k anahtarla şifreler ve şifreli metinleri sıralı bir sözlükte tutar. Daha sonra gizli anahtarla şifrelenmiş seçilmiş bir açık metni elde eder ve uygun bir eşleşmeyi sözlükten kontrol eder. Sözlükte arama ihmal edilebilir, fakat saldırı için 2^k tane n-bit bellek kelime gerekir. Bu yüzden bu saldırı pahalı bir saldırı olarak nitelenebilir.

Diğer bir saldırı türü olan kod kitabı saldırısında, saldırgan 2^n mümkün açık metnin gizli bir anahtar ile şifrelenmiş şifreli metinlerini elde eder ve bunları bir tabloya depolar. Bir şifreli metin için bekler ve elde ettiği gibi bu şifreli metni tablodakiler ile karşılaştırarak açık metni elde eder. Saldırı 2^n açık metin ve 2^n bellek kelime gerektirir. Bu saldırı türü de pahalı bir saldırı türüdür.

Geniş anahtar arama saldırısında ise 2^k olası anahtar denenerek şifreli metinden anlamlı bir açık metin elde edilince saldırı tamamlanır. 2^k olası deneme her ne kadar mümkün görünmese de zaman karmaşıklığı veri ve alan karmaşıklığından daha ucuz olarak düşünülür. Eğer bir kripto analiz saldırısı geniş anahtar aramadan daha az efor ile blok şifreyi kırarsa bu saldırı başarılı bir saldırı olarak görülebilir.

- *Gelişmiş saldırılar*

Bu tür saldırılar matematiksel bir fikrin kullanılması ile geliştirilmişlerdir. Bu saldırılara örnek olarak lineer kripto analiz, diferansiyel kripto analiz, kesik diferansiyel kripto analiz, imkansız diferansiyel kripto analiz verilebilir [3].

<u>En çok kullanılan blok şifreleme çeşitleri</u>

- *AES*

AES, uluslararası olarak kullanılan bir blok şifreleme algoritmasıdır. Vincent Rijmen ve Joan Daemen tarafından geliştirilmiş olup DES şifreleme algoritmasının zayıf noktalarını karşı oluşturulmuştur.

- *Blowfish*

Eskiyen DES şifreleme algoritmasının yerini alması ve diğer şifreleme algoritmalarının yerine alması için Bruce Schneier tarafından 1993 yılında tasarlanmıştır.

- *Camellia cipher*

Bilinen tüm şifre çözme saldırılarına karşı koyabilmek için yüksek güvenliği göz önünde bulundurularak tasarlanmıştır.

- *DES*

DES dünyanın en çok kullanılan şifreleme algoritmalardandır. 1970'li yıllarda geliştirilmiştir. Günümüzde DES şifreleme algoritması güvenini kaybetmiştir.

- *3DES (Üçlü Veri Şifreleme Standardı - Triple Data Encryption Standard)*

DES şifreleme algoritmasının güvenini kaybetmesinden dolayı IBM tarafından 1987 yılında geliştirilmiştir.

- *Feistel cipher*

Bu simetrik algoritma bilinen şifreleme blok şifreleme algoritmalarının temelini teşkil etmektedir.

- *IDEA*

IDEA, 1992 yılında geliştirilmiştir. Kaynak kodu ücretsiz olmasına rağmen ticari amaçlı kullanımlar için lisans gerektirmektedir.

- *Playfair cipher*

Temelini matematiksel matrislerden alan ve bilinen en basit algoritmadır.

- *Skipjack*

NSA tarafından tasarlanmış ve tasarımı ile ilgili bilgiler 1993'den 1998'e kadar gizli tutulmuştur [7].

3.2.2. Akan şifreleme algoritması

Kriptolojide açık veriyi rasgele bir şifreleme verisiyle karıştıran, simetrik anahtar şifreleyicisine akan veri şifreleyicisi denir. Buna karşın blok

şifreleyiciler büyük bloklar üzerinde sabit ve değişmeyen dönüşümler yapar. Akan veri şifreleyicisi, blok şifreleyicisinden daha hızlıdır ve daha düşük donanıma ihtiyaç duyar. Öte yandan yanlış kullanıldığında, özellikle aynı başlangıç durumu 2. kez kullanıldığında, büyük güvenlik açıkları verebilir. Şekil 3.2.'de akan şifrelemeye örnek verilmiştir.

Şekil 3.2. Akan şifreleme

En çok kullanılan akan şifreleme algoritmaları şunlardır;

RC4 (Ron's Code 4)

RC4 başlangıçta ticari bir sırdı ancak 1994 Eylül 'ünde Cypherpunks e-posta listesine isimsiz bir kişi tarafından detaylı açıklaması ve kaynak kodu yazıldı. Çok kısa bir süre sonra da sci.crypt haber grubunda da açıklandı. Buna rağmen RC4 ticari bir marka haline getirildi ve marka sorunlarıyla uğraşmamak için RC4 'e Arcfour ve Arc4 gibi isimler takıldı. Alleged RC4, tabirinin nedeni ise RSA firmasının şimdiye dek RC4 algoritmasının ne olduğunu resmi olarak açıklamamış olmasıdır. Kullanımı lisans gerektirir.

A5/1

İletişimde gizlilik dolayısıyla cep telefonları için kullanılmıştır. Güvensizdir.

A5/2

A5/1 in güvensizliğinden dolayı hazırlanmıştır.

Panama

Joan Daemen, Craig Clapp tarafından hazırlanmıştır.

Sober – 128 (SOBER Ailesi Şifreleme)

2003 yılında tasarlanmıştır. Amaç MAC (Ortam Erişim Yönetimi - Media Access Control) işlevselliği sağlamaktır [7,19].

3.3. Hibrit Sistemler

Saldırganlara karşı mesajların gizliliğini korumak için simetrik ve asimetrik şifreleme algoritmaları kullanılır. Hibrit şifreleme algoritmaları, simetrik ve asimetrik şifreleme algoritmalarının avantajlarını birleştirir. Hibrit şifreleme sisteminde mesajı şifrelemek için simetrik şifreleme algoritması, mesajı şifrelemek için kullanılan simetrik anahtarı şifrelemek için de asimetrik şifreleme algoritması kullanılır.

Son zamanlarda hibrit şifreleme üzerinde yapılan araştırma ve çalışmalar, anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması üzerine yoğunlaşmaktadır. Bu modelde, hibrit şifreleme iki ayrı parçadan oluşur. Tag anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması metodu ve Fujisaki-Okamoto anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması metodu hibrit şifreleme yöntemi için gösterilebilecek bilinen örneklerdir. Tag anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması metodunda anahtar kapsülleme mekanizmasının hesaplanması iki fonksiyona paylaştırılmıştır. Birinci fonksiyon rasgele bir anahtar seçerken ikinci fonksiyon verilen bir tag ile bu anahtarı şifreler. Fujisaki-Okamoto anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması metodu

ise zayıf simetrik ve asimetrik şifreleme algoritmalarını güçlü bir asimetrik şifreleme algoritmasına dönüştüren bir yapıya sahiptir.

Veri bütünlüğü, kimlik kontrolü, anahtar güvenliği ve bunları sağlarken harcanan süre kriptolojide önemli parametrelerdir. Hem Tag anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması metodu hem de Fujisaki-Okamoto anahtar kapsülleme mekanizması ve veri kapsülleme mekanizması metodu bu kıstasları sağlamaktadır [20].

4. KRİPTOLOJİDE KULLANILAN MATEMATİKSEL İŞLEMLER

4.1. Sonlu Uzay Aritmetiği

4.1.1. Matematiksel kavramlar

Aşağıda AES algoritmasında kullanılan Galois Alanlarını anlamak için bazı tanımlar verilmektedir.

Abelian Grubu

Abelian Grubu, bir G kümesi ve bu kümenin elemanları üzerinde tanımlanmış olan bir + işleminden oluşur. Bir grubun Abelian Grubu olması için aşağıdaki özellikleri sağlaması gerekir.

Kapalılık özelliği

a ve b ikisi de G kümesinin elemanı olmak üzere kapalılık özelliği gereğince $(a + b)$ ' de G kümesinin elemanı olmalıdır.

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

Değişme özelliği

a ve b ikisi de G kümesinin elamanı olmak üzere değişme özelliği gereğince $(a + b) = (b + a)$ eşitliği sağlanmalıdır.

$\forall a,b \in G: a + b = b + a$

Birleşme özelliği

a ve b ikisi de G kümesinin elamanı olmak üzere birleşme özelliği gereğince $(a + b) + c = a + (b + c)$ eşitliği sağlanmalıdır.

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

Etkisiz eleman özelliği

a G kümesinin elemanı olmak üzere G kümesinde a + 0 = a eşitliğini sağlayan bir tane etkisiz eleman vardır.

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

Ters eleman özelliği

a ve b G kümesinin elemanı olmak üzere öyle bir b elemanı vardır ki a + b = 0 olsun.

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

Halka

R boş kümeden farklı bir küme olsun. Bu küme üzerinde + ve ı ikili işlemleri tanımlı olsun. Eğer; (R,+) kümesi değişmeli bir küme, (R,ı) kümesi sadece birleşme özelliğini sağlayan bir küme ve ı işlemi + işlemi üzerine sağdan ve soldan dağılmalı ise (R,+,ı) kümesine halka denir. Eğer ı işlemi değişme özelliğine sahipse, (R,+,ı) halkası değişmeli halka olarak adlandırılır. + işleminin birim elemanı 0, ı işleminin birim elemanı ise 1'dir.

Alan

Bir F kümesi, üzerinde tanımlanmış + ve ı işlemleriyle birlikte aşağıdaki koşulları sağlıyorsa, bir alan oluşturulur.

(F,+) bir Abelian Grubu olmalıdır. (F,ı) da bir Abelian Grubu olmalıdır, ancak sadece 0 elemanı için ters eleman olmayabilir. ı işleminin + işlemi üzerinde dağılma özelliği olmalıdır.

Sonlu alan

Sonlu alan ise yukarıdaki alan tanımına uyan ancak adından da anlaşılacağı gibi sonlu sayıda elemanı olan alanlardır.

Galois Alanı

p' yi asal sayı olarak düşünelim. GF(p) üzerinde + ve ı işlemleri tanımlanmış p sayıda elemandan oluşan bir sonlu alanı temsil eder. P sonlu alanın karakteristiği olarak adlandırır. GF(p^n) ise GF(p)'nin genişletilmiş sonlu alanını ifade eder, GF(p^n)'nin p karakteristiğini gösterir. Eleman sayısı ise GF(p) sonlu alanının elaman sayısının n katıdır. GF(p^n)'nin yapılan + ve ı işlemlerinin alan içerisinde kalmasını sağlamak için n. dereceden bir indirgeme polinomuna ihtiyacı vardır [21].

GF(2n) Galois Alanı

Karakteristiği 2 olan GF(2^n) sonlu alanının kriptografide kullanımı yaygındır. 2^n adet eleman içerir. Elemanları yan yana yazılmış bitler {0,1} olmasından dolayı kullanımı oldukça rahattır. 2 karakteristikli Galois Alanlarında gösterim olarak polinomsal gösterim yaygın olarak kullanılır. 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 elemanının GF(2^n) 'ye ait bir elemanla çarpılması ile elde edilir. Örneğin, GF(2^8) 'in bir elemanı olan {10100111} polinomsal olarak şu şekilde gösterilir;

$a(x)=x^7+x^5+x^2+x+1$

4.1.2. Bayt gösterilim şekilleri

Aşağıdaki dört bölümde sonlu uzaydaki bir elemana ait farklı gösterilim şekilleri verilmiştir. Ayrıca birer örnekte her gösterilim için eklenmiştir.

İkilik gösterilim

Bir bayt, 8 bit içerir. Aşağıda ikilik gösterilime ait örnek vardır.

11001100_2

Onluk gösterilim

Onluk gösterilim, ikilik sayının tüm bitlerinin karşı düşen ikinin üs değeri ile çarpılıp toplanması ile elde edilen sayıdır. Ekteki örnekte ayrıntılı gösterilim mevcuttur:

$1.2^7+0.2^6+1.2^5+0.2^4+0.2^3+0.2^2+1.2^1+1.2^0=2^7+2^5+2^1+2^0$

$=128+32+2+1$

$=163_{10}$

Onaltılık gösterilim

0 ile 15 arası rakamlar dört bit ile gösterilebilirler. Ancak 10 ile 15 arası rakamlar (0 . . . 9) arası rakamlarla gösterilemezler ve bu nedenle A ile F arası harfler onaltılık düzende bu rakamları göstermek için kullanılırlar.

$0000_2 =$ 0_{10} 0_{16}
$0001_2 =$ 1_{10} 1_{16}
$0010_2 =$ 2_{10} 2_{16}
$0011_2 =$ 3_{10} 3_{16}
$0100_2 =$ 4_{10} 4_{16}
$0101_2 =$ 5_{10} 5_{16}
$0110_2 =$ 6_{10} 6_{16}
$0111_2 =$ 7_{10} 7_{16}
$1000_2 =$ 8_{10} 8_{16}
$1001_2 =$ 9_{10} 9_{16}
$1010_2 =$ 10_{10} A_{16}

$1011_2 = 11_{10}$ B_{16}
$1100_2 = 12_{10}$ C_{16}
$1101_2 = 13_{10}$ D_{16}
$1110_2 = 14_{10}$ E_{16}
$1111_2 = 15_{10}$ F_{16}

İkilik gösterilimden onaltılık gösterilime geçmek için bir bayt iki adet dörtlüye bölünür ve her iki dörtlü, karşı düşen onaltılık karşılığı ile gösterilir. Aşağıdaki örnekte ayrıntılı gösterilmiştir:

$10100011 = 1010\ 0011 = A3_{16}$
$A_{16}\ \ 3_{16}$

Onaltılık düzenden onluk düzene geçmek için sol hane 16 sayısı ile çarpılır ve sağ hane 1 ile çarpılır ve her iki çarpım sonucu toplanır.

$A3_{16} = A.2^4 + 3.2^0 = 10.16+3.1=163_{10}$

Polinom gösterilim

Bir baytın polinom gösterilimi ikilik sistemden onluk sisteme geçmek için kullanılan denkleme çok benzer. Bu denklemde 2 sayısı yerine x konularak polinom gösterilim elde edilir:

$1.x^7+0.x^6+1.x^5+0x^4+0x^3+0x^2+1x^1+1x^0= x^7+x^5+x^1+1$

Unutulmaması gereken en önemli nokta polinom gösterilimde Galois Alanı sonlu uzayında katsayıların 1 değerini alabileceğidir.

İndirgenemez polinom

$F(x)=x^m+\Sigma^{m-1}(f_i*x_i)$ i=0 fi $\in$ {0,1} olmaz üzere GF(2)'de tanımlı m. dereceden f(x) polinomu aynı alan üzerinde tanımlı diğer polinomların çarpımı şeklinde

yazılamıyorsa f(x) polinomu indirgenemez polinomdur ve indirgeme polinomu olarak adlandırılır.

$\forall a \in GF(2^m)$ için a elemanının polinomsal baz gösterilimi $a_i \in \{0,1\}$ olmak üzere aşağıdaki denklem verilmektedir.

$a=(a_{m-1}a_{m-2}a_{m-3}....a_1a_0) = a_{m-1}x^{m-1}+a_{m-2}x^{m-2}+....+a_1x+a_0$

Polinomsal gösterilim Galois Alanları üzerinde çok yaygın bir kullanıma sahiptir [22].

4.1.3. GF(2^n)'de matematiksel işlemler

Toplama işlemi

İki adet polinomun toplaması işlemi aynı üstel değere sahip x değerlerinin katsayıların toplanması şeklindedir.

$$(x^6 + x^4 + x^2 + x +1) + (x^7 + x^5 + x + 1)$$

$$x^6 + \quad x^4 \quad +x^2 + \ x + 1$$

$$x^7 \quad + \ x^5 + \quad\quad x \ + 1$$

$$x^7 \ + \ x^6 + x^5 + \ x^4 \ + x^2 + 2x \ +2$$

Bu işlem sonucunda sonuç polinomun bazı katsayıları 0 veya 1 olmayabilir. Bu nedenle sonuç polinomu bir bayt gösterilimi olamayacaktır.

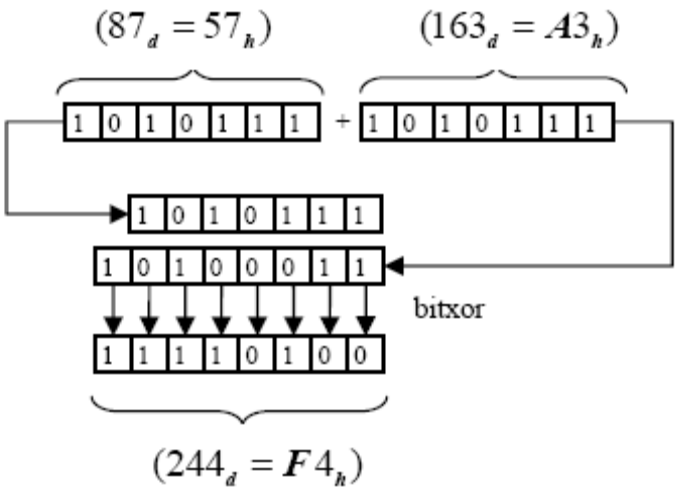
Toplama sonuç polinomunda sadece ikilik düzende katsayıların olmasını sağlamak için aşağıdaki sonuç kümesi gösterilmiş olan XOR işlemi kullanılır. İki adet 1 rakamının XOR işlemi 2 olmadığından sonuç polinomda 2

değerinde katsayı olmayacaktır. Çizelge 4.1.'de XOR işlemi tablosu verilmiştir.

Çizelge 4.1. XOR işlemi tablosu

x	y	x XOR y
0	0	0
0	1	1
1	0	1
1	1	1

Şekil 4.1.'de 2 adet ikilik düzende sayının toplama işlemi ve sonuç bayt değeri gösterilmiştir:



Şekil 4.1. Toplama işlemi

Sonuç polinomu;

11110100_b = 244_b = $F4_h$ = $x^7 + x^6 + x^5 + x^4 + x^2$

Çarpma işlemi

İki adet polinomun çarpımı, aynı üstel değere sahip x'lerin katsayılarının toplanması ile gerçeklenir. Bu klasik polinom çarpımıdır.

$$(x^6 + x^4 + x^2 + x + 1) * (x^7 + x^5 + x + 1)$$

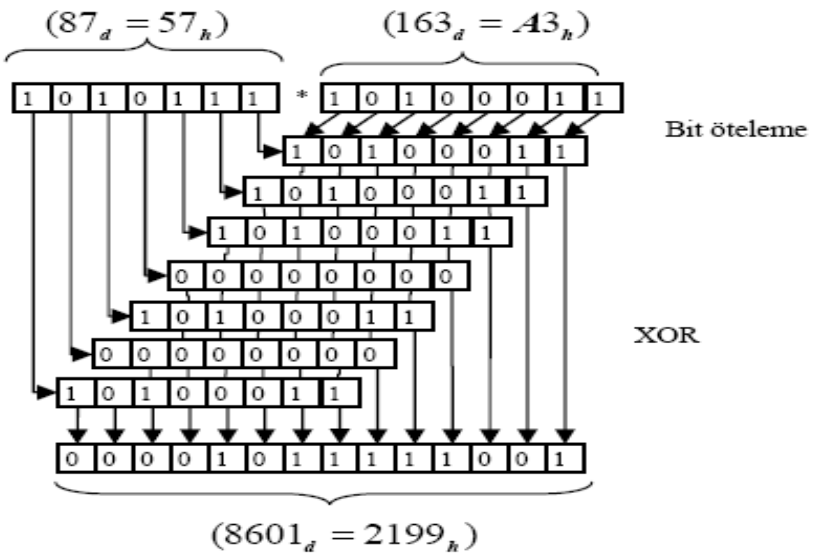
$$x^7 + x^5 + x + 1$$
$$x^8 + x^6 + x^2 + x$$
$$x^9 + x^7 + x^3 + x^2$$
$$x^{11} + x^9 + x^5 + x^4$$
$$x^{13} + x^{11} + x^7 + x^6$$

$$x^{13} + 2x^{11} + 2x^9 + x^8 + 3x^7 + 2x^6 + 2x^5 + x^4 + x^3 + 2x^2 + 2x + 1$$

Tekrar klasik polinom çarpımında bazı katsayılar 1 veya 0 değerinden farklı değerler almıştır. Bu katsayılar sonlu uzay işleminde bu değerleri almamalıdırlar ve bu nedenle düzenlenmesi gereklidir. Bu nedenle genelleştirilmiş XOR işlemi her katsayı için göz önüne alınmalıdır ve her tek sayı 1 değerine çevrilmelidir.

$$x^{13} + x^8 + x^7 + x^4 + x^3 + 1$$

Şekil 4.2.'de bit seviyesinde çarpma işlemi verilmiştir. İkilik sayılardan bir tanesi diğerinin bit değerine göre her bit için ötelenir. Eğer sıradaki sayı 1 değerinde ise diğer baytın hepsi, 0 ise sadece 0 değerleri konulur. Sırayla her bit için öteleme işlemi bittikten sonra aynı sıradaki tüm bitler XOR işlemine tabi tutulurlar ve sonuç bit serisi elde edilir.



Şekil 4.2. Çarpma işlemi

Bölme işlemi

$$(x^{13} + x^8 + x^7 + x^4 + x^3 + 1) : (x^8 + x^4 + x^3 + x + 1) = x^5 - x$$

$$-\ (x^{13} + x^9 + x^8 + x^6 + x^5)$$

$$-x^9 + x^7 - x^6 - x^5 + x^4 + x^3 + 1$$

$$(-x^9 - x^5 - x^4 - x^2 - x)$$

$$x^7 - x^6 + 2x^4 + x^3 + x^2 + x + 1$$

Bölünen polinomun en büyük üstel değeri x^{13}, bölenin en yüksek üstel değeri x^8 ile bölünür ve sonuç x^5 olarak elde edilir. Daha sonra x^5 bölenin tüm değerleri ile çarpılarak bölünen polinomunda çıkarılır ve yeni bir bölünen polinomu elde edilir;

$(-x^9 + x^7 - x^6 - x^5 + x^4 + x^3 + 1)$

Daha sonra aynı işlemler tekrar yapılır ve bu işlem bölünen polinomunun en yüksek üstel değeri bölenin en yüksek üstel değerinden küçük değerde

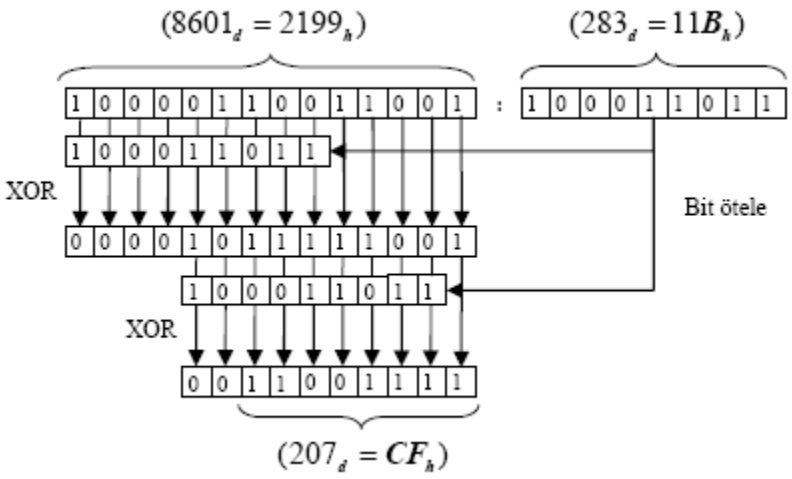
oluncaya kadar devam eder. En sonda kalan bölünen, işlem sonucunda kalan polinomunu oluşturur;

$(x^7 + x^6 + 2x^4 + x^3 + x^2 + x + 1)$

Sonuç polinomuna genel XOR işlemi uygulanarak bayt gösteriliminde sonuç elde edilir;

$(x^7 + x^6 + x^3 + x^2 + x + 1)$

Bit seviyesinde işlemler Şekil 4.3.'te gösterilmiştir [16].



Şekil 4.3. Bölme işlemi

4.2. AES Şifreleme Algoritmasında Kullanılan Matematiksel İşlemler

4.2.1. Toplama

Toplama işlemi, Bölüm 4.1.3' de anlatıldığı gibi, iki bayt'ın karşılıklı bitlerinin XOR işlemine tabi tutulmasıyla gerçekleşir.

$a,b,c \in GF(2^m)$

$a = (11001111) = \{x^7 + x^6 + x^3 + x^2 + x + 1\}$, $b=(10001000) = \{x^7 + x^3\}$

$c = a+b = x^7 + x^6 + x^3 + x^2 + x + 1 + x^7 + x^3 = 2x^7 + x^6 + 2x^3 + x^2 + x + 1$

$c = x^6 + x^2 + x + 1 = (01000111)$

Yapılan işlemler incelendiğinde toplama işleminin bit bit XOR'lama işleminden ibaret olduğu görülebilir. Bu özellik donanım gerçeklemelerinde büyük kolaylık sağlar.

$GF(2^m)$ sonlu alanı üzerinde çıkarma işlemi toplama işlemiyle aynıdır. Her iki işlem de sonlu alan üzerinde tanımlı elemanların bit bit XOR'lanması ile gerçekleşir.

4.2.2. Çarpma

Çarpma işlemi, Bölüm 4.1.3' de anlatıldığı gibi gerçekleşir.

$GF(2^m)$ alanı üzerinde tanımlı çarpma işlemi mod P(x) polinomu uyarınca gerçekleştirilir. Çarpılacak olan elemanların polinomsal gösterilimi birbirleri ile çarpılır ve çıkan sonucun P(x) polinomuna bölümünden kalan $GF(2^m)$ sonlu alanı üzerinde tanımlı çarpma işleminin sonucuna eşittir.

$P(x) = x^8 + x^4 + x^3 + x + 1$ $a,b,c \in GF(2^m)$

$a = (01010111) = \{ x^6 + x^4 + x^2 + x + 1 \}$, $b = (10000011) = \{x^{7+} x + 1 \}$

$c = a * b = \{ x^6 + x^4 + x^2 + x + 1 \} * \{ x^7 + x + 1 \}$

$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1$

$= x^{13} + x^{11} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + 1 \quad (\text{mod } x^8 + x^4 + x^3 + x + 1)$

$= x^7 + x^6 + 1$

P(x) indirgeme polinomu çarpma işleminin sonucunun aynı alan üzerindeki bir elemana denk düşmesini garanti eder.

$GF(2^m)$ alanı üzerinde tanımlı sıfırdan farklı her elemanın çarpmaya göre tersi mevcuttur. Sıfır elemanının tersi ise yine sıfırdır. Alan üzerinde tanımlı herhangi bir b(x) polinomunu tersi $b(x)^{-1}$ olarak gösterilir.

$a(x) * b(x) = 1 \pmod{P(x)}$ şartını sağlaması gerekir [2].

4.2.3. x ile çarpma

$a(x) = a_7x^7 + a_6x^6 + a_5x^5 + a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$

ikili polinomunun, x polinomuyla çarpımı neticesinde;

$a_7x^8 + a_6x^7 + a_5x^6 + a_4x^5 + a_3x^4 + a_2x^3 + a_1x^2 + a_0x$

polinomu elde edilir.

$x*a(x)$ ifadesinin sonucu, yukarıdaki polinomun mod P(x)'de indirgenmesiyle bulunur. a_7 0'a eşitse, sonuç zaten indirgenmiş durumdadır. a_7'nin 1'e eşit olması durumunda, x^8'li terimin açığa çıkması nedeniyle, P(x) (1{1a}) polinomu ile indirgeme işlemine tabi tutulur. Dolayısıyla x ({00000010} ya da {02}) ile çarpma; sola kaydırma ve sonrasında uygulanan, {1a} ile şartlı XOR işlemi şeklinde, bayt düzeyinde gerçekleştirilir [23].

4.2.4. Tablo kullanarak çarpma

Çizelge 4.2. Matris çarpım tablosu

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

Gelişmiş Şifreleme Standardında Çizelge 4.2.'deki S Kutuları kullanılır. Bu S Kutularını kullanabilmek için tablo kullanarak çarpma işlemini bilmemiz gerekir. Yukarıdaki tabloda üreteç {03}'dür. Hangi ikilinin karşılığına bakacaksak o ikilinin tabloda kesiştiği noktadaki değeri buluruz.

Örneğin 72 'nin tablodaki karşılığına bakacak olursak x değerimiz 7, y değerimiz 2'dir. Tablodan 7 ve 2 nin kesiştiği noktaya bakarsak 0a olduğunu görürüz ve aşağıdaki gibi ifade ederiz;

$$\{72\} = \{03\}^{(0a)}$$

5. GELİŞMİŞ ŞİFRELEME STANDARDI

Ocak 1997'de NIST, yeni bir şifreleme standardının geliştirilmesi için bir çalışma başlatmıştır. Geliştirilecek yeni şifreleme standardının mevcut standart olan DES'in yerini alması düşünülmüştür. Çünkü DES'in 64 bitlik anahtar uzayı, gelişen teknoloji ve artan işlemci hızları karşısında güvenilirliğini yitirmeye başlamıştı.

NIST tarafından, yeni şifreleme standardını belirlemek amacıyla bir yarışma düzenlenmiş ve Eylül 1997'de algoritmalar için resmi çağrıda bulunulmuştur. Ağustos 1998'de 15 aday algoritmanın değerlendirmeye alındığı duyurulmuş ve Nisan 1999'da gerçekleşen 2. Tur seçimlerinde algoritma sayısı 5'e indirilmiştir. Seçilen bu 5 finalist aşağıda sıralanmıştır:

IBM'in geliştirdiği MARS (Multiplication, Addition, Rotation and Substitution),
Ronald Rivest'in geliştirdiği RC6 (Ron's Code 6),
Belçikalı takım tarafından geliştirilen Rijndael,
İngiltere, İsrail ve Norveç'ten üyeleri olan bir takımın ortak olarak geliştirdiği Serpent,
Bruce Schneier tarafından geliştirilen Twofish.

Dört yıl boyunca süren değerlendirme ve eleme süreci sonrasında, Ekim 2000'de sonuç açıklanmış ve NIST, Joan Daemen ve Vincent Rijmen tarafından tasarlanan, Rijndael algoritmasının Gelişmiş Şifreleme Standardı olarak kullanılacağını ilan etmiştir [24, 25, 26, 27].

5.1. Gelişmiş Şifreleme Sistemi Hakkında Genel Bilgi

AES algoritmasında simetrik blok şifreler kullanılarak şifreleme ve şifre çözme işlemleri yapılır. Şifreyle şifrelenmiş metin anlamsız bir ifadedir ve şifre çözme işleminden sonra orijinal metine geri dönülür. AES algoritmasında 128 bitlik bloklar şeklindeki veri 128,192 ve 256 bitlik

anahtarlar oluşturularak şifrelenir ve şifre çözülür. Bu anahtarlar AES128, AES192 ve AES 256 adıyla geçer. Diğer uzunluktaki anahtarlara izin verilmez. Bu standart yazılım donanım veya bu ikisinin kombinasyonu şeklinde kullanılabilir [23].

5.1.1. Algoritma parametreleri

Tur anahtarını ekleme

Şifreli metinden şifresiz metine veya şifresiz metinden şifreli metine çevirirken duruma tur anahtarı XOR ile ekler. Durum ile tur anahtarın boyut uzunlukları eşit olmak zorundadır.

Sütunları karıştırma

Her bir satırdaki durum ve karışık olan datayı anahtarla şifreleyip yeni bir satıra yazar.

Satırları öteleme

En alttaki 3 satırı özel bir şekilde kaydırır.

Bayt yer değiştirme

S kutusuna göre çıkarma işlemi yapar.

Ters sütunları karıştırma

Şifreli metinden orijinal metine dönerken sütunları karıştırmanın yaptığının tersini yapar.

Ters satırları öteleme

Şifreli metinden orijinal metine dönerken satırları ötelemenin yaptığının tersini yapar.

Ters bayt yer değiştirme

Şifreli metinden orijinal metine dönerken bayt yer değiştirmenin yaptığının tersini yapar.

Rcon

Sabit kelimedir.

5.1.2 Notasyon

Girişler ve çıkışlar

AES algoritmasında girişler ve çıkışlar 128 bit içerir. Genelde blok şeklinde oluşumu ister. Şifre anahtarı 128,192 veya 256 bit olur.

Bayt

Giriş, çıkış ve şifre anahtarı a ile gösterilirse, bytelar a_n veya a[n] form çeşitlerinden biriyle gösterilir. Burada n şöyle gösterilebilir:

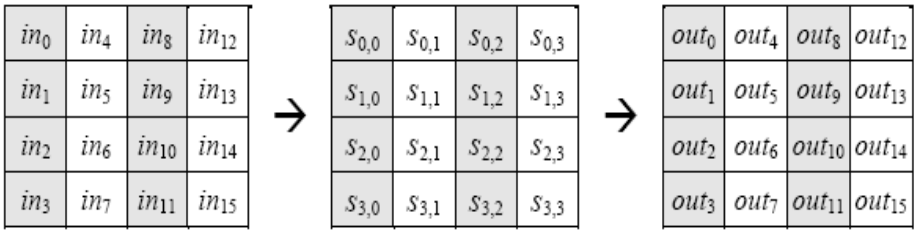
Anahtar uzunluğu = 128 bits, $0 \leq n < 16$;
Anahtar uzunluğu = 192 bits, $0 \leq n < 24$;
Anahtar uzunluğu = 256 bits, $0 \leq n < 32$;
Blok uzunluğu = 128 bits, $0 \leq n < 16$;

Bir byte şu şekilde gösterilebilir;

$b_7x^7 + b_6x^6 + b_5x^5 + b_4x^4 + b_3x^3 + b_2x^2 + b_1x + b_0 = \Sigma^7_{i=0} (b_ix^i)$

Örneğin {01100011} $x_6 + x_5 + x + 1$ şeklinde gösterilir.

Bitler Şekil 5.1.'deki gibi state durumunda da gösterilebilir.



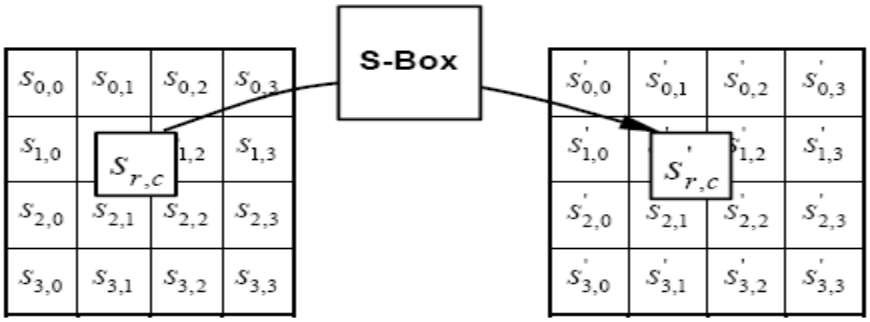
Şekil 5.1. Bitlerin state durumu

5.2. Şifreleme

Gelişmiş Şifreleme Standardında bilgiyi şifrelemek için 4 aşama vardır.

5.2.1. Bayt yer değiştirme

Bayt yer değiştirme yaparken Çizelge 5.1.'deki S kutusu devreye giriyor. Örneğin 2 baytlık 63'ü ele alalım. S kutusuna bakıyoruz ve 63'e denk gelen değeri bulup onun yerindeki satıra yazıyoruz.



Şekil 5.2. Bayt yer değiştirme

Çizelge 5.1.'e bakarsak 63'e karşılık gelen değer fb'dir. Bunun gibi birçok örnek yapabiliriz. Örneğin a4'e S kutusunda denk gelen değer 49'dur. 5d'ye denk gelen değer 58'dir.Bu şekilde örnekleri arttırabiliriz. Şekil 5.2.'de bayt yer değiştirme gösterilmiştir.

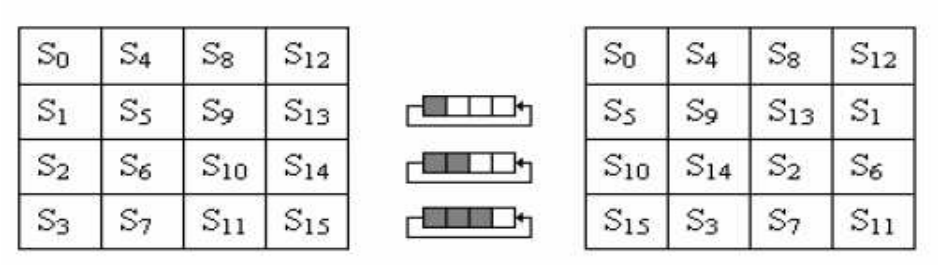
Çizelge 5.1. S kutusu

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

5.2.2. Satırları öteleme

Daha önce de belirtilmiş olduğu gibi, 16 baytlık durum verisi, elemanları baytlar olan 4∗4 boyutunda bir matris gibi ele alınır. Satırları öteleme katmanı, adından da anlaşıldığı üzere, bu matrisin son üç satırı üzerinde işlem yapar.

Şifreleme için, ilk satır aynen bırakılır. İkinci satır sağdan sola doğru bir pozisyon değiştirecek şekilde, dairesel olarak, ötelenir. Üçüncü satır benzer şekilde iki pozisyon, dördüncü satır da üç pozisyon sola doğru, dairesel olarak ötelenir. Şekil 5.3.'te satırları öteleme gösterilmiştir.



Şekil 5.3. Satırları öteleme

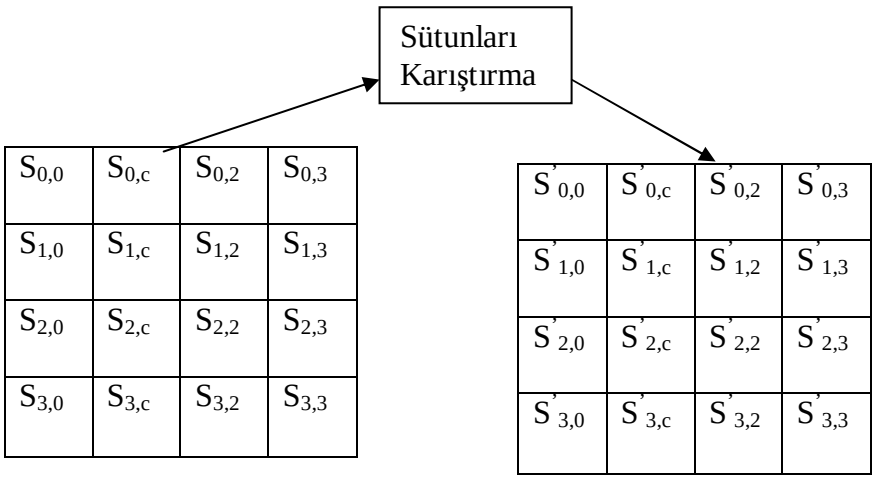
5.2.3. Sütunları karıştırma

Sütunları karıştırma katmanı, 4∗4'lük durum matrisinin sütunları üzerinde işlem yapar. Giriş durum matrisinin her sütunu, sütunları karıştırma dönüşümünden geçirilerek, çıkış durum matrisi elde edilir. GF(2) için doğrusal bir işlem olan sütunları karıştırma, algoritmaya, nüfuz etme özelliği kazandırır.

Durum matrisinin her bir sütunu, katsayıları GF(2^8)'in birer elemanı olan üçüncü dereceden birer polinom olarak ele alınır. Elde edilen bu polinomlar sabit bir polinomla, $x^4 + 1$'e göre mod alınarak, çarpılarak sütunları karıştırma dönüşümü gerçekleştirilir. Şekil 5.4.'te sütunları karıştırma gösterilmiştir.

$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$

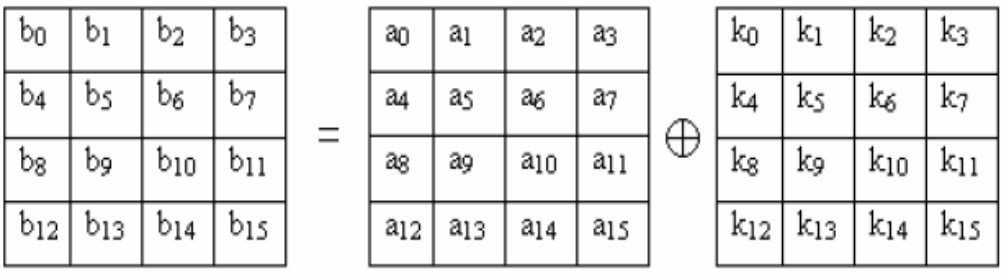
$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb.$$



Şekil 5.4. Sütunları karıştırma

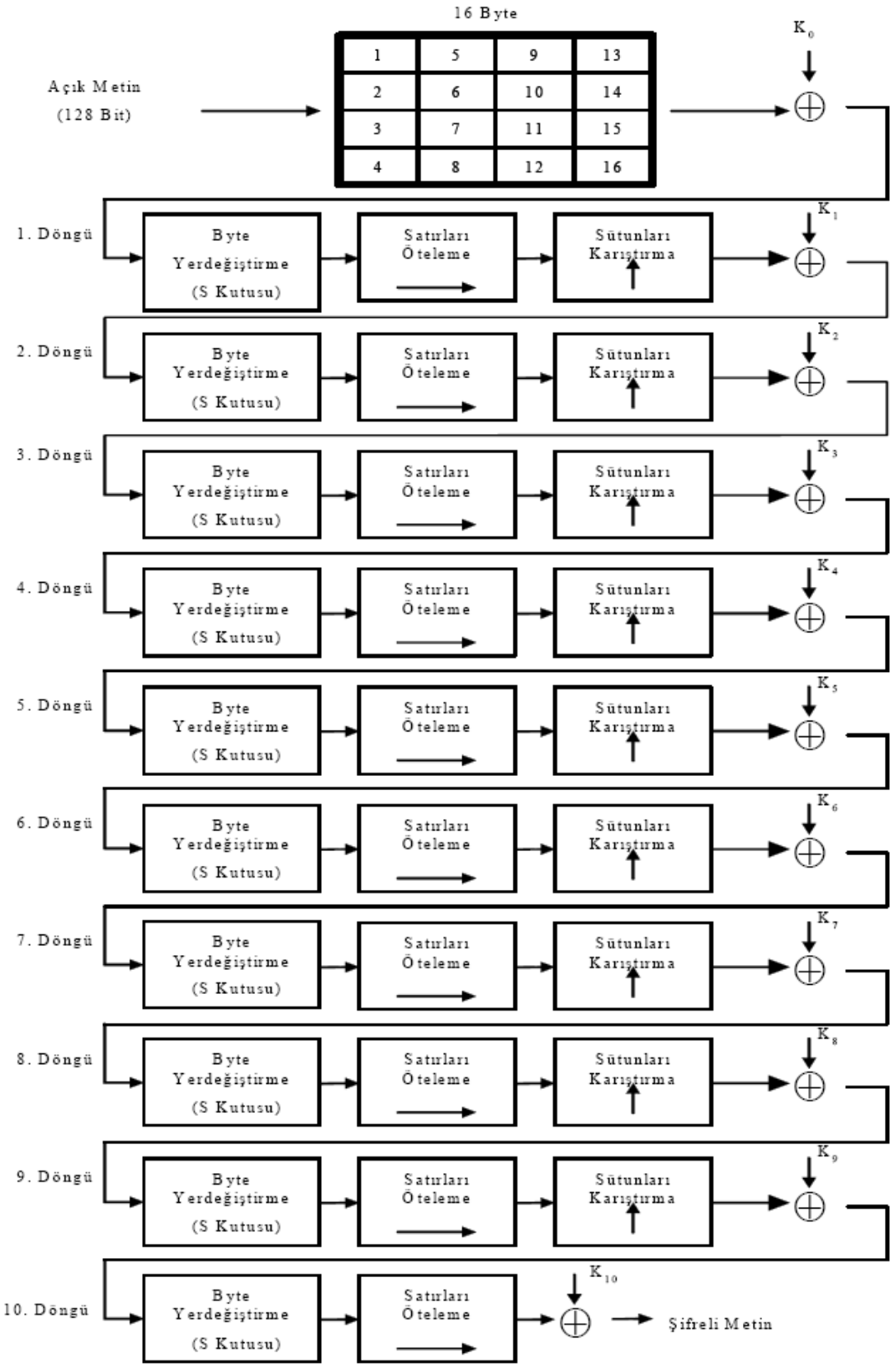
5.2.4. Tur anahtarının toplamı

AES algoritmasında, her bir tur sonunda elde edilen matris, anahtar üreteci tarafından ilgili tur için üretilmiş, tur anahtarı ile karşılıklı XOR işlemine tabi tutulur. Turun son katmanı olan tur anahtarını toplama, GF(2) üzerinde doğrusal bir dönüşümdür. Şifreleme ve şifre çözme işlemleri için, bu katman tamamen aynı yapıdadır. Bir başka ifadeyle, söz konusu katmanın tersi kendisine eşittir. Şekil 5.5.'te tur anahtarının toplamı gösterilmiştir.



Şekil 5.5. Tur anahtarının toplamı

Bu işlemlerden geçtikten sonra verimiz Şekil 5.6.'daki gibi şifrelenmiş olacaktır.



Şekil 5.6. AES şifreleme örneği

5.3. Anahtar Üreteci

AES anahtar üreteci, farklı uzunluklarda anahtar bloklarıyla (128, 192 veya 256 bit) çalışabilir. Ancak veri bloklarının uzunluğu sabit 128 bit olduğu için, ürettiği tur anahtarlarının uzunluğu da sabit olup, 128 bittir.

Anahtar üreteci, her turun son adımında, üretilen ara değerle toplanacak olan tur anahtarlarını üretir. Bu nedenle anahtar üretecinin şifreleme anahtarından tur sayısı kadar, 128 bitlik tur anahtarı üretmesi gerekir.

Farklı uzunluktaki anahtar blokları (128,192 veya 256 bit) $4*N_k$ boyutlarına ayırarak matrislere yerleştirilir. Matristeki her bir sütun satırları öteleme ve bayt yer değiştirme işlemleri gerçekleştirdikten sonra Rcon ile XOR işlemi yapılır. Rcon kaçıncı döngü ise onun değerini alır. Daha sonra bir önceki döngü matrisinin aynı sütunundaki değerle XOR işlemi yapılır. Bu işlemler yapıldıktan sonra sütunlar birleştirerek tur anahtarımızı matrise yazabiliriz. Çizelge 5.2.'de hangi anahtar bloklarının kaç tur atacağının tablosu yer almaktadır.

Çizelge 5.2. Anahtar uzunluğuna göre tur sayısı

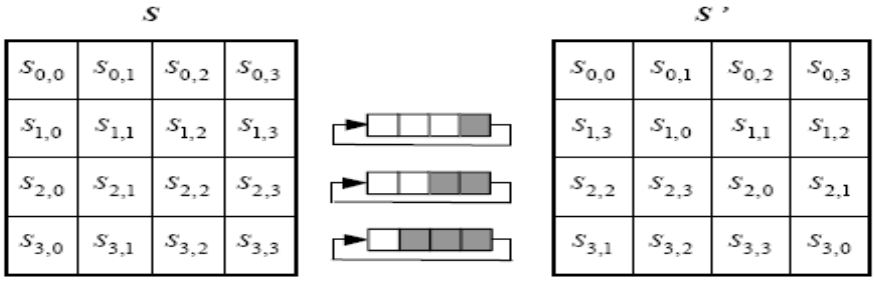
	Anahtar Uzunluğu (N_k kelime)	Blok Uzunluğu (N_b kelime)	Tur Sayısı (N_r)
AES–128	4	4	10
AES–192	6	4	12
AES–256	8	4	14

5.4. Deşifreleme

Gelişmiş Şifreleme Standardında şifrelemede olduğu gibi deşifrelemede de 4 aşama vardır [23].

5.4.1. Ters satırları öteleme

Burada şifrelemede yaptığımızın tam tersini yapıyoruz. Deşifreleme için, ilk satır aynen bırakılır. İkinci satır soldan sağa doğru bir pozisyon değiştirecek şekilde, dairesel olarak, ötelenir. Üçüncü satır benzer şekilde iki pozisyon, dördüncü satır da üç pozisyon sağa doğru, dairesel olarak ötelenir. Şekil 5.7.'de ters satırları öteleme gösterilmiştir.



Şekil 5.7. Ters satırları öteleme

5.4.2. Ters bayt yer değiştirme

Bayt yer değiştirmedeki gibi, fakat bu sefer ters S kutusundaki değerine bakılıyor ve bu değer matristeki yerinde yazılıyor.

Çizelge 5.3. Ters S kutusu

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

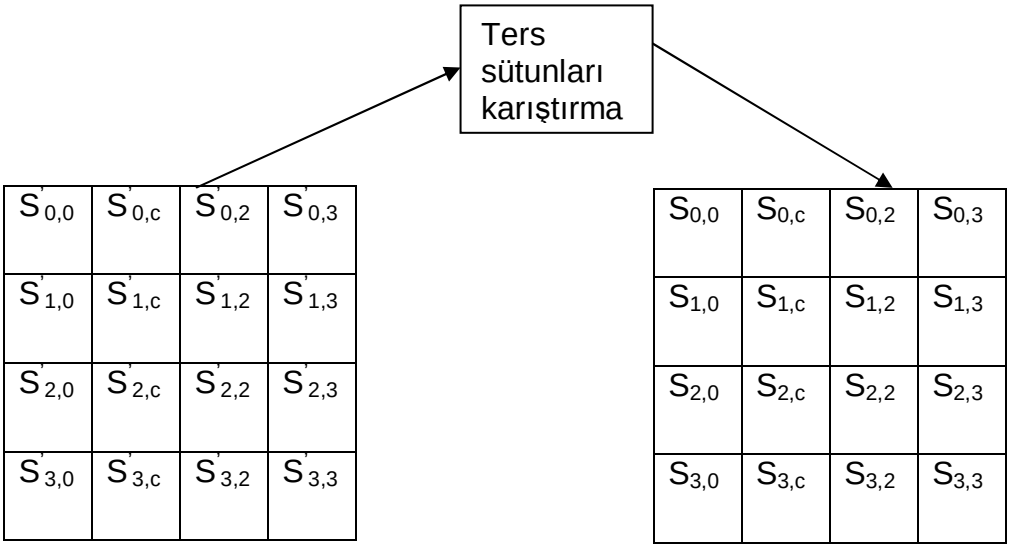
Örneğin fb'ye karşılık gelen değer 63'tür. Çizelge 5.3.'te ters S kutusu verilmiştir.

5.4.3. Ters sütunları karıştırma

Sütunları karıştırmadaki gibidir fakat aşağıdaki polinom uygulanır. Şekil 5.8.'de ters sütunları karıştırma gösterilmiştir.

$a{-}1(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}.$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < \boldsymbol{Nb}.$$



Şekil 5.8. Ters sütunları karıştırma

5.4.4. Ters tur anahtarının toplamı

Tur anahtarını toplamında yaptığımız gibi elde edilen matrisle tur anahtarı XOR işlemine tabi tutulur.

5.5. Gelişmiş Şifreleme Standardına Örnek

5.5.1. 128 bitlik anahtar genişletme

Çizelge 5.4.'te 128 bitlik anahtar genişletme adımları gösterilmiştir.

Şifre anahtarı: 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c

w0 = 2b7e1516 w1 = 28aed2a6 w2 = abf71588 w3 = 09cf4f3c

Çizelge 5.4. 128 bitlik anahtar genişletme

i	Kelime	Satır Öteleme Sonrası	Bayt Yer Değiştirme Sonrası	Rcon[i\N$_k$]	Rcon ile XOR	W[i-N$_k$]	W[i]= Kelime XOR W[i-N$_k$]
4	09cf4f3c	cf4f3c09	8a84eb01	01000000	8b84eb01	2b7e1516	a0fafe17
5	a0fafe17					28aed2a6	88542cb1
6	88542cb1					abf71588	23a33939
7	23a33939					09cf4f3c	2a6c7605
8	2a6c7605	6c7605a	50386be5	02000000	52386be5	a0fafe17	f2c295f2
9	f2c295f2					88542cb1	7a96b943
10	7a96b943					23a33939	5935807a
11	5935807a					2a6c7605	7359f67f
12	7359f67f	59f67f73	cb42d28f	04000000	cf42d28f	f2c295f2	3d80477d
13	3d80477d					7a96b943	4716fe3e
14	4716fe3e					5935807a	1e237e44
15	1e237e44					7359f67f	6d7a883b
16	6d7a883b	7a883bd	dac4e23c	08000000	d2c4e23c	3d80477d	ef44a541
17	ef44a541					4716fe3e	a8525b7f
18	a8525b7f					1e237e44	b671253b
19	b671253b					6d7a883b	db0bad00
20	db0bad00	0bad00b	2b9563b9	10000000	3b9563b9	ef44a541	d4d1c6f8
21	d4d1c6f8					a8525b7f	7c839d87
22	7c839d87					b671253b	caf2b8bc
23	caf2b8bc					db0bad00	11f915bc

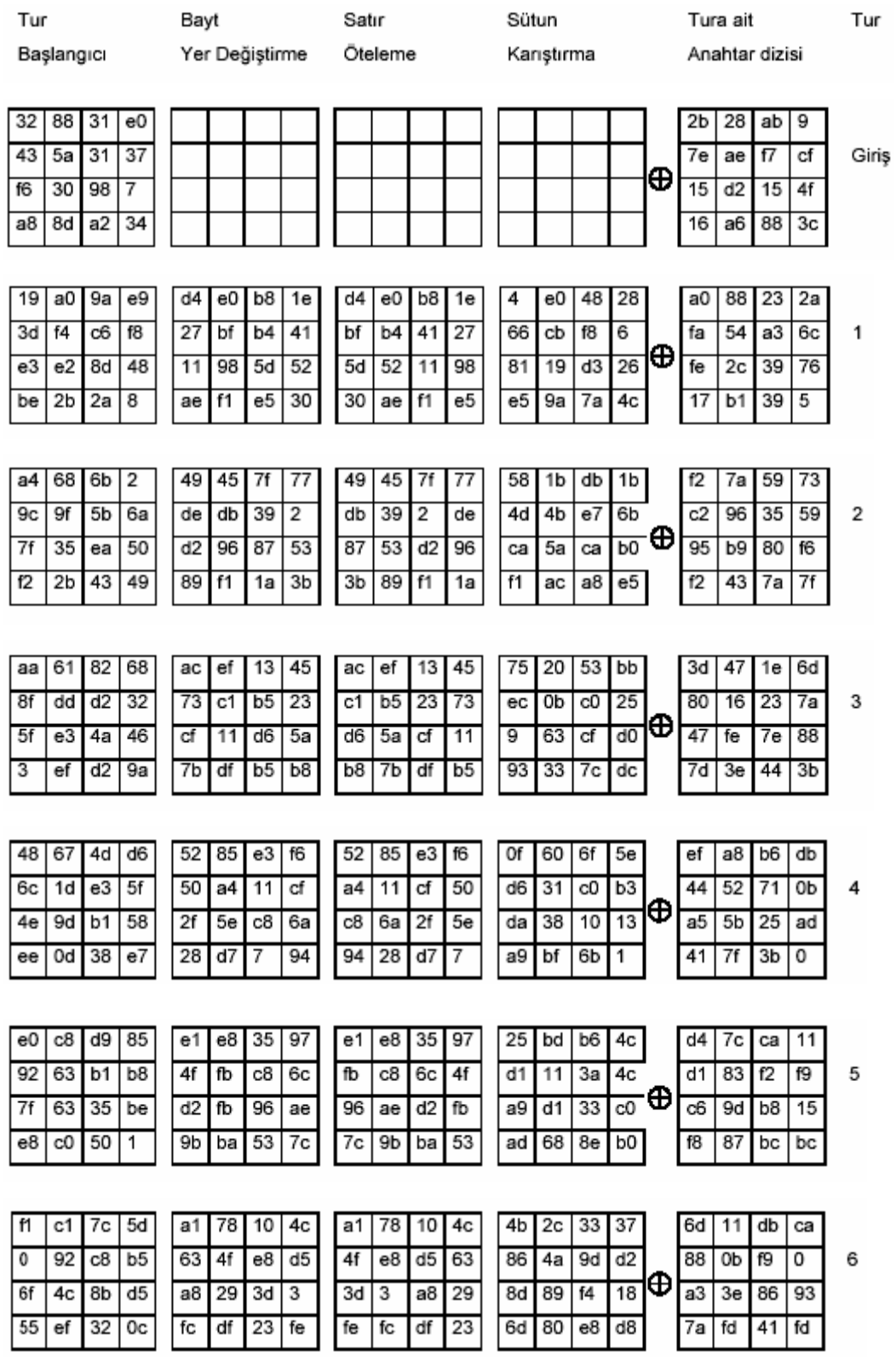
24	11f915bc	f915bc11	99596582	20000000	b9596582	d4d1c6f8	6d88a37a
25	6d88a37a					7c839d87	110b3efd
26	110b3efd					caf2b8bc	dbf98641
27	dbf98641					11f915bc	ca0093fd
28	ca0093fd	0093fdca	63dc5474	40000000	23dc5474	6d88a37a	4e54f70e
29	4e54f70e					110b3efd	5f5fc9f3
30	5f5fc9f3					dbf98641	84a64fb2
31	84a64fb2					ca0093fd	4ea6dc4f
32	4ea6dc4f	a6dc4f4e	2486842f	80000000	a486842f	4e54f70e	ead27321
33	ead27321					5f5fc9f3	b58dbad2
34	b58dbad2					84a64fb2	312bf560
35	312bf560					4ea6dc4f	7f8d292f
36	7f8d292f	8d292f7f	5da515d2	1b000000	46a515d2	ead27321	ac7766f3
37	ac7766f3					b58dbad2	19fadc21
38	19fadc21					312bf560	28d12941
39	28d12941					7f8d292f	575c006e
40	575c006e	5c006e7	4a639f5b	36000000	7c639f5b	ac7766f3	d014f9a8
41	d014f9a8					19fadc21	c9ee2589
42	c9ee2589					28d12941	e13f0cc8
43	e13f0cc8					575c006e	b6630ca6

5.5.2. 128 bitlik anahtar ile şifreleme

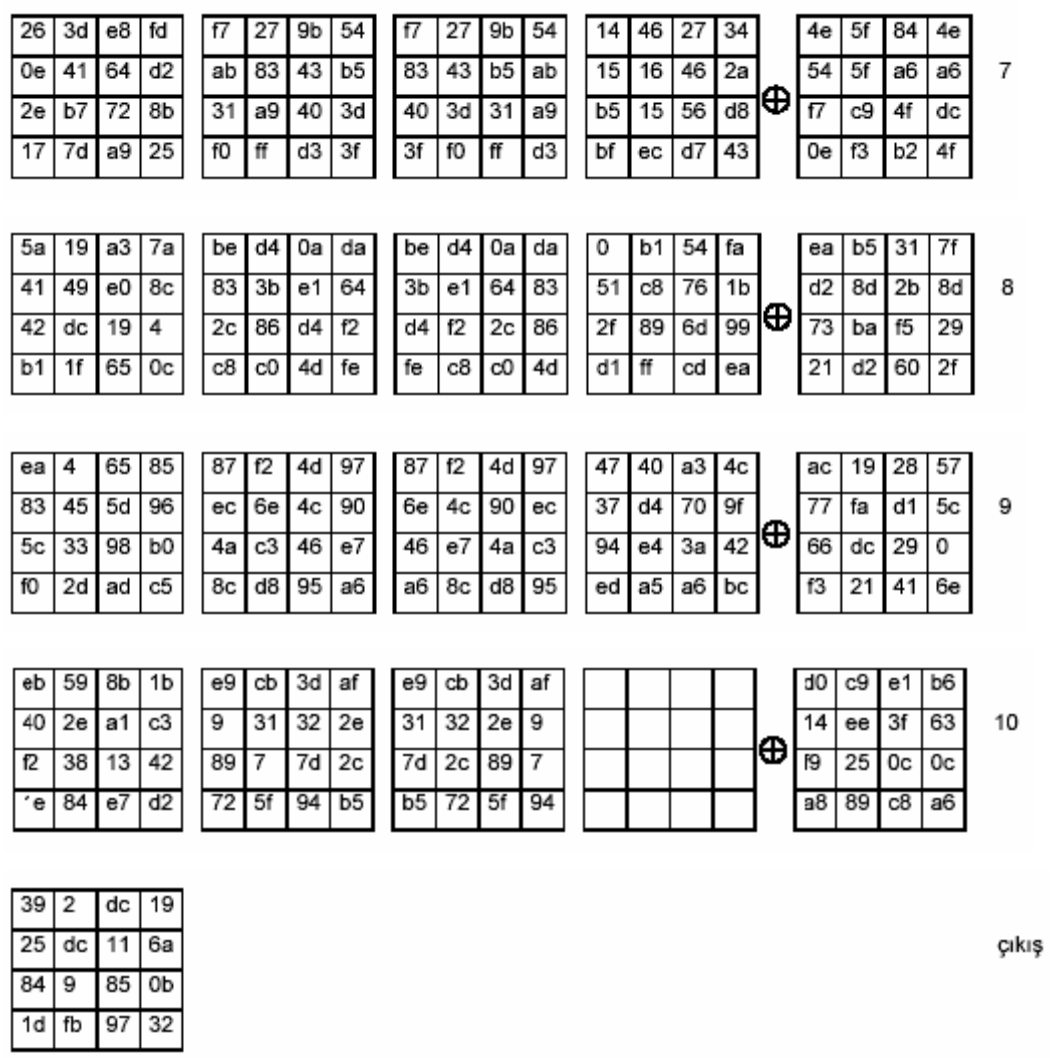
Şekil 5.9.'da 128 bit uzunluğa sahip bir blok ve anahtar için şifreleme adımları gösterilmiştir.

Giriş bloğu = 32 43 f6 a8 88 5a 30 8d 31 31 98 a2 e0 37 07 34

Şifre anahtarı = 2b 7e 15 16 28 ae d2 a6 ab f7 15 88 09 cf 4f 3c



Şekil 5.9. AES şifreleme örneği



Şekil 5.9. (devam) AES şifreleme örneği

6. GELİŞMİŞ ŞİFRELEME STANDARDININ MATLAB UYGULAMASI

Bu tez çalışmasında gelişmiş şifreleme standardına ait bir simulasyon geliştirilmiştir. Bu simulasyon özellikle elektronik mühendisliğinde çok kullanılan MATLAB uygulaması kullanılmıştır.

Gelişmiş şifreleme standardına ait alt yapıyı 4. ünite ve 5. ünitede detaylı olarak anlattık.

Bu ünitede yapılan simulasyonun çalışma şekli adım adım anlatılacaktır. Simulasyonun akış diyagramı Şekil 5.6.' da verilmiştir.

6.1. Şifreleme

Şifrelemeye başlarken ilk önce S kutusu, ters S kutusu ve Rcon hesaplanır. S kutusu ve ters S kutusu hesaplanırken GF(2^8) de ters alma işlemi yapılır. Çizelge 6.1.'de S kutusu, Çizelge 6.2.'de ters S kutusu, Çizelge 6.3.'te Rcon gösterilmiştir.

Çizelge 6.1. Oluşturulan S kutusu

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

Çizelge 6.2. Oluşturulan ters 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	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

128 bitlik gelişmiş şifreleme sisteminde 10 tur döneceği için 10 adet Rcon üreteceğiz.

Çizelge 6.3. Rcon

01 00 00 00
02 00 00 00
04 00 00 00
08 00 00 00
10 00 00 00
20 00 00 00
40 00 00 00
80 00 00 00
1b 00 00 00
36 00 00 00

6.1.1. Anahtar genişletme

AES başlangıç kısmında girmek istediğimiz şifreyi 16'lık sistemde gireriz. Eğer 16 adet anahtar girmezsek 'Anahtar vektör olmalı' uyarısı, ff'den büyük girersek 'vektör elemanları byte olmalı (0 <= anahtar(i) <= 255)' uyarısı çıkar.

Örnek anahtar olarak Çizelge 6.4.'teki anahtarı girelim.

Çizelge 6.4. Başlangıç tur anahtarı

00	04	08	0c
01	05	09	0d
02	06	0a	0e
03	07	0b	0f

Her bir satırı 1 sola öteleyip S kutusundaki karşılığıyla bayt yer değiştirip Rcon ile XOR işlemine sokarız. 4. satırda yapılan işlemleri gösterecek olursak;

Satır öteleme sonrası: 0d 0e 0f 0c (sola doğru 1 satır kaydırdık).
Bayt yer değiştirme sonrası: d7 ab 76 fe (S kutusunda kendine karşılık gelen değeri aldık).
rcon(05, :) : 01 00 00 00 (ilk Rcon değeri).
rconla XOR sonrası: d6 ab 76 fe (Rcon ile XOR işleminden sonra).

5. sütunu oluştururken 1. sütundaki değerlerle rconla XOR sonrasını XOR işlemine koyarız. Daha sonraki sütunları oluştururken, bir üstündeki sütunla 4 satır önceki sütunu XOR işlemine koyarak buluruz. Bu işlemleri tüm satırlarda uyguladıktan sonra;

w(05, :) : d6 aa 74 fd
w(06, :) : d2 af 72 fa
w(07, :) : da a6 78 f1
w(08, :) : d6 ab 76 fe

değerlerini elde ederiz. Çizelge 6.5.'te 4 anahtar matris şeklinde gösterilmiştir. Fakat 10 tane döngü yapacağımız için 10 tane daha anahtar oluşturmamız gerekir.

Çizelge 6.5. 1. Tur anahtarı

d6	d2	da	d6
aa	af	a6	ab
74	72	78	76
fd	fa	f1	fe

Satır öteleme sonrası: ab 76 fe d6
Bayt yer değiştirme sonrası: 62 38 bb f6
rcon(09, :) : 02 00 00 00
rconla XOR sonrası: 60 38 bb f6
w(09, :) : b6 92 cf 0b
w(10, :) : 64 3d bd f1
w(11, :) : be 9b c5 00
w(12, :) : 68 30 b3 fe

Çizelge 6.6.'da 2. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.6. 2. Tur anahtarı

b6	64	be	68
92	3d	9b	30
cf	bd	c5	b3
0b	f1	00	fe

Satır öteleme sonrası: 30 b3 fe 68
Bayt yer değiştirme sonrası: 04 6d bb 45
rcon(13, :) : 04 00 00 00
rconla XOR sonrası: 00 6d bb 45
w(13, :) : b6 ff 74 4e
w(14, :) : d2 c2 c9 bf
w(15, :) : 6c 59 0c bf
w(16, :) : 04 69 bf 41

Çizelge 6.7.'de 3. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.7. 3. Tur anahtarı

b6	d2	6c	04
ff	c2	59	69
74	c9	0c	bf
4e	bf	bf	41

Satır öteleme sonrası: 69 bf 41 04
Bayt yer değiştirme sonrası: f9 08 83 f2
rcon(17, :) : 08 00 00 00
rconla XOR sonrası: f1 08 83 f2
w(17, :) : 47 f7 f7 bc
w(18, :) : 95 35 3e 03
w(19, :) : f9 6c 32 bc
w(20, :) : fd 05 8d fd

Çizelge 6.8.'de 4. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.8. 4. Tur anahtarı

47	95	f9	fd
f7	35	6c	05
f7	3e	32	8d
bc	03	bc	fd

Satır öteleme sonrası: 05 8d fd fd

Bayt yer değiştirme sonrası: 6b 5d 54 54

rcon(21, :) : 10 00 00 00

rconla XOR sonrası: 7b 5d 54 54

w(21, :) : 3c aa a3 e8

w(22, :) : a9 9f 9d eb

w(23, :) : 50 f3 af 57

w(24, :) : ad f6 22 aa

Çizelge 6.9.'da 5. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.9. 5. Tur anahtarı

3c	a9	50	ad
aa	9f	f3	f6
a3	9d	af	22
e8	eb	57	aa

Satır öteleme sonrası: f6 22 aa ad

Bayt yer değiştirme sonrası: 42 93 ac 95

rcon(25, :) : 20 00 00 00

rconla XOR sonrası:62 93 ac 95

w(25, :) : 5e 39 0f 7d

w(26, :) : f7 a6 92 96

w(27, :) : a7 55 3d c1

w(28, :) : 0a a3 1f 6b

Çizelge 6.10.'da 6. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.10. 6. Tur anahtarı

5e	f7	a7	0a
39	a6	55	a3
0f	92	3d	1f
7d	96	c1	6b

Satır öteleme sonrası: a3 1f 6b 0a

Bayt yer değiştirme sonrası: 0a c0 7f 67

rcon(29, :) : 40 00 00 00

rconla XOR sonrası: 4a c0 7f 67

w(29, :) : 14 f9 70 1a

w(30, :) : e3 5f e2 8c

w(31, :) : 44 0a df 4d

w(32, :) : 4e a9 c0 26

Çizelge 6.11.'de 7. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.11. 7. Tur anahtarı

14	e3	44	4e
f9	5f	0a	a9
70	e2	df	c0
1a	8c	4d	26

Satır öteleme sonrası: a9 c0 26 4e

Bayt yer değiştirme sonrası: d3 ba f7 2f

rcon(33, :) : 80 00 00 00

rconla XOR sonrası: 53 ba f7 2f

w(33, :) : 47 43 87 35

w(34, :) : a4 1c 65 b9

w(35, :) : e0 16 ba f4

w(36, :) : ae bf 7a d2

Çizelge 6.12.'de 8. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.12. 8. Tur anahtarı

47	a4	e0	ae
43	1c	16	bf
87	65	ba	7a
35	b9	f4	d2

Satır öteleme sonrası: bf 7a d2 ae

Bayt yer değiştirme sonrası: 08 da b5 e4

rcon(37, :) : 1b 00 00 00

rconla XOR sonrası: 13 da b5 e4

w(37, :) : 54 99 32 d1

w(38, :) : f0 85 57 68

w(39, :) : 10 93 ed 9c

w(40, :) : be 2c 97 4e

Çizelge 6.13.'de 9. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.13. 9. Tur anahtarı

54	f0	10	be
99	85	93	2c
32	57	ed	97
d1	68	9c	4e

Satır öteleme sonrası: 2c 97 4e be

Bayt yer değiştirme sonrası: 71 88 2f ae

rcon(41, :) : 36 00 00 00

rconla XOR sonrası: 47 88 2f ae

w(41, :) : 13 11 1d 7f

w(42, :) : e3 94 4a 17

w(43, :) : f3 07 a7 8b

w(44, :) : 4d 2b 30 c5

Çizelge 6.14.'de 10. tur anahtarı matris şeklinde gösterilmiştir.

Çizelge 6.14. 10. Tur anahtarı

13	e3	f3	4d
11	94	07	2b
1d	4a	a7	30
7f	17	8b	c5

6.1.2. Metin şifreleme

acik_metin. txt dosyasına şifrelemek istediğimiz rakamları 10'luk sistemde alt alta yazarız. Örnek olarak açık metini;

22
21
32
34
54
65
43
3
5
55
87
98
6
7
54
0

girelim.10'luk sistemde girdiğimiz bu rakamlar ilk önce 16'lık sisteme çevrilir. Çizelge 6.15.'de başlangıç durumu 1 gösterilmiştir.

Çizelge 6.15. Başlangıç durumu 1

16	36	05	06
15	41	37	07
20	2b	57	36
22	03	62	00

Başlangıç durumu ile Çizelge 6.4. teki başlangıç tur anahtarı XOR işlemine koyulur. Çizelge 6.16.'da XOR'lama işlemi sonucu gösterilmiştir.

Çizelge 6.16. Durum baslangic turu 1

16	32	0d	0a
14	44	3e	0a
22	2d	5d	38
21	04	69	0f

Sonra herbir değer yerine S Kutusundaki karşılığı yazılır. Örneğin 16 değerinin S kutusundaki karşılığı 47'dir. Ve bu değer 16 yerine yazılır. Çizelge 6.17.'de bayt yer değiştirme sonrası gösterilmiştir.

Çizelge 6.17. Bayt yer değiştirme sonrası 1

47	23	d7	67
fa	1b	b2	67
93	d8	4c	07
fd	f2	f9	76

Daha sonra satır öteleme işlemi yapılır. 1. satır aynı kalır, 2. satır 1 kademe sola, 3. satır 2 kademe sola ve 4. satır 3 kademe sola kaydırılır. Çizelge 6.18.'de satır öteleme sonrası 1 gösterilmiştir.

Çizelge 6.18. Satır öteleme sonrası 1

47	23	d7	67
1b	b2	67	fa
4c	07	93	d8
76	fd	f2	f9

Daha sonra herbir sütun Çizelge 6.19.'daki sabit matrisle tek tek çarpılır.

Çizelge 6.19. Sabit matris

02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

Bu matrisle çarpıldıktan sonra Çizelge 6.20.'deki matris oluşur.

Çizelge 6.20. Sütun karıştırma sonrası 1

99	71	7d	fa
d3	a8	45	02
5e	83	80	26
72	31	69	62

Bu işlemler geliştirdiğimiz 10 tur anahtarıyla beraber arka arkaya uygulanır.

Çizelge 6.5.'teki 1. tur anahtarı eklenir. Çizelge 6.21.-6.24. arasında 2. tur döngüsü gösterilmiştir.

Çizelge 6.21. Durum başlangıç turu 2

4f	a3	a7	2c
79	07	a3	a9
2a	f1	f8	50
8f	cb	98	9c

Çizelge 6.22. Bayt yer değiştirme sonrası 2

84	0a	5c	71
b6	c5	11	d3
e5	a1	41	53
73	1f	46	de

Çizelge 6.23. Satır öteleme sonrası 2

84	0a	5c	71
c5	11	d3	b6
41	53	e5	a1
de	73	1f	46

Çizelge 6.24. Sütun karıştırma sonrası 2

d8	07	2c	c4
08	ae	ca	b8
ba	28	7f	54
b4	ba	ec	08

Çizelge 6.6.'daki 2. tur anahtarı eklenir. Çizelge 6.25.-6.28. arasında 3. tur döngüsü gösterilmiştir.

Çizelge 6.25. Durum başlangıç turu 3

6e	63	92	ac
9a	93	51	88
75	95	ba	e7
bf	4b	ec	f6

Çizelge 6.26. Bayt yer değiştirme sonrası 3

9f	fb	4f	91
b8	dc	d1	c4
9d	2a	f4	94
08	b3	ce	42

Çizelge 6.27. Satır öteleme sonrası 3

9f	fb	4f	91
dc	d1	c4	b8
f4	94	9d	2a
42	08	b3	ce

Çizelge 6.28. Sütun karıştırma sonrası 3

ec	19	e7	0e
79	ed	d3	4a
76	01	64	34
16	43	f5	bd

Çizelge 6.7.'deki 3. tur anahtarı eklenir. Çizelge 6.29.-6.32. arasında 4. tur döngüsü gösterilmiştir.

Çizelge 6.29. Durum başlangıç turu 4

5a	cb	8b	0a
86	2f	8a	23
02	c8	68	8b
58	fc	4a	fc

Çizelge 6.30. Bayt yer değiştirme sonrası 4

be	1f	3d	67
44	15	7e	26
77	e8	45	3d
6a	b0	d6	b0

Çizelge 6.31. Satır öteleme sonrası 4

be	1f	3d	67
15	7e	26	44
45	3d	77	e8
b0	6a	b0	d6

Çizelge 6.32. Sütun karıştırma sonrası 4

ad	eb	d7	3c
eb	ce	58	1a
ea	a5	3e	89
f2	b6	6d	b2

Çizelge 6.8.'deki 4. tur anahtarı eklenir. Çizelge 6.33.-6.36. arasında 5. tur döngüsü gösterilmiştir.

Çizelge 6.33. Durum başlangıç turu 5

ea	7e	2e	c1
1c	fb	34	1f
1d	9b	0c	04
4e	b5	d1	4f

Çizelge 6.34. Bayt yer değiştirme sonrası 5

87	f3	31	78
9c	0f	18	c0
a4	14	fe	f2
2f	d5	3e	84

Çizelge 6.35. Satır öteleme sonrası 5

87	f3	31	78
0f	18	c0	9c
fe	f2	a4	14
84	2f	d5	3e

Çizelge 6.36. Sütun karıştırma sonrası 5

7e	08	48	65
04	e1	88	59
f8	65	c6	8e
70	ba	86	7c

Çizelge 6.9.'daki 5. tur anahtarı eklenir. Çizelge 6.37.-6.40. arasında 6. tur döngüsü gösterilmiştir.

Çizelge 6.37. Durum başlangıç turu 6

42	a1	18	c8
ae	7e	7b	af
5b	f8	69	ac
98	51	d1	d6

Çizelge 6.38. Bayt yer değiştirme sonrası 6

2c	32	ad	e8
e4	f3	21	79
39	41	f9	91
46	d1	3e	f6

Çizelge 6.39. Satır öteleme sonrası 6

2c	32	ad	e8
f3	21	79	e4
f9	91	39	41
f6	46	d1	3e

Çizelge 6.40. Sütun karıştırma sonrası 6

59	d0	22	83
37	9e	c5	c6
37	e0	ce	cc
89	6a	15	fa

Çizelge 6.10.'daki 6. tur anahtarı eklenir. Çizelge 6.41.-6.44. arasında 7. tur döngüsü gösterilmiştir.

Çizelge 6.41. Durum başlangıç turu 7

07	27	85	89
0e	38	90	65
38	72	f3	d3
f4	fc	d4	91

Çizelge 6.42. Bayt yer değiştirme sonrası 7

c5	cc	97	a7
ab	07	60	4d
07	40	0d	66
bf	b0	48	81

Çizelge 6.43. Satır öteleme sonrası 7

c5	cc	97	a7
07	60	4d	ab
0d	66	07	40
81	bf	b0	48

Çizelge 6.44. Sütun karıştırma sonrası 7

14	fa	55	bb
5d	19	b4	62
40	ba	1f	54
47	2c	93	89

Çizelge 6.11.'deki 7. tur anahtarı eklenir. Çizelge 6.45.-6.48. arasında 8. tur döngüsü gösterilmiştir.

Çizelge 6.45. Durum başlangıç turu 8

00	19	11	f5
a4	46	be	cb
30	58	c0	94
5d	a0	de	af

Çizelge 6.46. Bayt yer değiştirme sonrası 8

63	d4	82	e6
49	5a	ae	1f
04	6a	ba	22
4c	e0	1d	79

Çizelge 6.47. Satır öteleme sonrası 8

63	d4	82	e6
5a	ae	1f	49
ba	22	04	6a
79	4c	e0	1d

Çizelge 6.48. Sütun karıştırma sonrası 8

eb	34	da	7b
7b	b9	50	d7
dd	ea	ae	5c
b7	73	5d	28

Çizelge 6.12.'deki 8. tur anahtarı eklenir. Çizelge 6.49.-6.52. arasında 9. tur döngüsü gösterilmiştir.

Çizelge 6.49. Durum başlangıç turu 9

ac	90	3a	d5
38	a5	46	68
5a	8f	14	26
82	ca	a9	fa

Çizelge 6.50. Bayt yer değiştirme sonrası 9

91	60	80	03
07	06	5a	45
be	73	fa	f7
13	74	d3	2d

Çizelge 6.51. Satır öteleme sonrası 9

91	60	80	03
06	5a	45	07
fa	f7	be	73
2d	13	74	d3

Çizelge 6.52. Sütun karıştırma sonrası 9

e4	ca	1e	af
a5	c5	a7	4b
0f	fa	3e	8c
0e	2b	88	cc

Çizelge 6.13.'deki 9. tur anahtarı eklenir. Çizelge 6.53.-6.55. arasında 10. tur döngüsü gösterilmiştir.

Çizelge 6.53. Son tur başlangıcındaki durum

b0	3a	0e	11
3c	40	34	67
3d	ad	d3	1b
df	43	14	82

Çizelge 6.54. Son tur bayt yer değiştirme sonrası

e7	80	ab	82
eb	09	18	85
27	95	66	af
9e	1a	fa	13

Çizelge 6.55. Son tur satır öteleme sonrası

e7	80	ab	82
09	18	85	eb
66	af	27	95
13	9e	1a	fa

Çizelge 6.14.'deki 10. tur anahtarı eklenir. 10. tur anahtarıyla Çizelge 6.55.'teki son tur satır öteleme sonrası XOR işlemine koyulur ve şifrelenmiş metin elde edilir.

Çizelge 6.56. Şifrelenmiş metin

f4	63	58	cf
18	8c	82	c0
7b	e5	80	a5
6c	89	91	3f

6.2. Şifre Çözme

Şifre çözme işlemini yaparken şifrelemede yaptığımız adımların benzerlerini ters olarak yaparız. Yine başlangıçta S kutusu, ters S kutusu ve Rcon üreterek başlarız.

6.2.1. Anahtar genişletme

Şifrelenmiş bir metinin şifresini çözebilmemiz için şifrelenen metinle aynı anahtara sahip olmamız lazım. Örnek olarak Bölüm 6.1.1 ‘deki genişlettiğimiz anahtarı kullanacağız.

6.2.2. Şifrelenmiş metini açık metine çevirme

Burada şifrelemede yaptığımız işlemlerin tersini uygulayacağız. Koyacağımız anahtarı da sondan başlayarak uygulayacağız. Çizelge 6.57.’de şifre çözme başlangıç durumu gösterilmiştir.

Çizelge 6.57. Şifre çözme başlangıç durumu

f4	63	58	cf
18	8c	82	c0
7b	e5	80	a5
6c	89	91	3f

Çizelge 6.14.’deki 10. tur anahtarı eklenir. Çizelge 6.58.-6.60. arasında 10. tur döngüsü gösterilmiştir.

Çizelge 6.58. Şifre çözme durum Başlangıç Turu 9

e7	80	ab	82
09	18	85	eb
66	af	27	95
13	9e	1a	fa

Çizelge 6.59. Ters Satır Öteleme Sonrası 9

e7	80	ab	82
eb	09	18	85
27	95	66	af
9e	1a	fa	13

Çizelge 6.60. Ters Bayt Yer Değiştirme Sonrası 9

b0	3a	0e	11
3c	40	34	67
3d	ad	d3	1b
df	43	14	82

Çizelge 6.13.'deki 9. tur anahtarı eklenir. Çizelge 6.61.-6.64. arasında 9. tur döngüsü gösterilmiştir.

Çizelge 6.61. Tur anahtarı ekledikten sonra 9

e4	ca	1e	af
a5	c5	a7	4b
0f	fa	3e	8c
0e	2b	88	cc

Çizelge 6.62. Şifre çözme durum Başlangıç Turu 8

91	60	80	03
06	5a	45	07
fa	d7	be	73
2d	13	74	d3

Çizelge 6.63. Ters Satır Öteleme Sonrası 8

91	60	80	03
07	06	5a	45
be	73	fa	f7
13	74	d3	2d

Çizelge 6.64. Ters Bayt Yer Değiştirme Sonrası 8

ac	90	3a	d5
38	a5	46	68
5a	8f	14	26
82	ca	a9	fa

Çizelge 6.12.'deki 8. tur anahtarı eklenir. Çizelge 6.65.-6.68. arasında 8. tur döngüsü gösterilmiştir.

Çizelge 6.65. Tur anahtarı ekledikten sonra 8

eb	34	da	7b
7b	b9	50	d7
dd	ea	ae	5c
b7	73	5d	28

Çizelge 6.66. Şifre çözme durum Başlangıç Turu 7

63	d4	82	e6
5a	ae	1f	49
ba	22	04	6a
79	4c	e0	1d

Çizelge 6.67. Ters Satır Öteleme Sonrası 7

63	d4	82	e6
49	5a	ae	1f
04	6a	ba	22
4c	e0	1d	79

Çizelge 6.68. Ters Bayt Yer Değiştirme Sonrası 7

00	19	11	f5
a4	46	be	cb
30	58	c0	94
5d	a0	de	af

Çizelge 6.11.'deki 7. tur anahtarı eklenir. Çizelge 6.69.-6.72. arasında 7. tur döngüsü gösterilmiştir.

Çizelge 6.69. Tur anahtarı ekledikten sonra 7

14	fa	55	bb
5d	19	b4	62
40	ba	1f	54
47	2c	93	89

Çizelge 6.70. Şifre çözme durum Başlangıç Turu 6

c5	cc	97	a7
07	60	4d	ab
0d	66	07	40
81	bf	b0	48

Çizelge 6.71. Ters Satır Öteleme Sonrası 6

c5	cc	97	a7
ab	07	60	4d
07	40	0d	66
bf	b0	48	81

Çizelge 6.72. Ters Bayt Yer Değiştirme Sonrası 6

07	27	85	89
0e	38	90	65
38	72	f3	d3
f4	fc	d4	91

Çizelge 6.10.'daki 6. tur anahtarı eklenir. Çizelge 6.73.-6.76. arasında 6. tur döngüsü gösterilmiştir.

Çizelge 6.73. Tur anahtarı ekledikten sonra 6

59	d0	22	83
37	9e	c5	c6
37	e0	ce	cc
89	6a	15	fa

Çizelge 6.74. Şifre çözme durum Başlangıç Turu 5

2c	32	ad	e8
f3	21	79	e4
f9	91	39	41
f6	46	d1	3e

Çizelge 6.75. Ters Satır Öteleme Sonrası 5

2c	32	ad	e8
e4	f3	21	79
39	41	f9	91
46	d1	3e	f6

Çizelge 6.76. Ters Bayt Yer Değiştirme Sonrası 5

42	a1	18	c8
ae	7e	7b	af
5b	f8	69	ac
98	51	d1	d6

Çizelge 6.9.'daki 5. tur anahtarı eklenir. Çizelge 6.77.-6.80. arasında 5. tur döngüsü gösterilmiştir.

Çizelge 6.77. Tur anahtarı ekledikten sonra 5

7e	08	48	65
04	e1	88	59
f8	65	c6	8e
70	ba	86	7c

Çizelge 6.78. Şifre çözme durum Başlangıç Turu 4

87	f3	31	78
0f	18	c0	9c
fe	f2	a4	14
84	2f	d5	3e

Çizelge 6.79. Ters Satır Öteleme Sonrası 4

87	e3	31	78
9c	0f	18	c0
a4	14	fe	f2
2f	d5	3e	84

Çizelge 6.80. Ters Bayt Yer Değiştirme Sonrası 4

ea	7e	2e	c1
1c	fb	34	1f
1d	9b	0c	04
4e	b5	d1	4f

Çizelge 6.8.'deki 4. tur anahtarı eklenir. Çizelge 6.81.-6.84. arasında 4. tur döngüsü gösterilmiştir.

Çizelge 6.81. Tur anahtarı ekledikten sonra 4

ad	eb	d7	3c
eb	ce	58	1a
ea	a5	3e	89
f2	b6	6d	b2

Çizelge 6.82. Şifre çözme durum Başlangıç Turu 3

be	1f	3d	67
15	7e	26	44
45	3d	77	e8
b0	6a	b0	d6

Çizelge 6.83. Ters Satır Öteleme Sonrası 3

be	1f	3d	67
44	15	7e	26
77	e8	45	3d
6a	b0	d6	b0

Çizelge 6.84. Ters Bayt Yer Değiştirme Sonrası 3

5a	cb	8b	0a
86	2f	8a	23
02	c8	68	8b
58	fc	4a	fc

Çizelge 6.7.'deki 3. tur anahtarı eklenir. Çizelge 6.85.-6.88. arasında 3. tur döngüsü gösterilmiştir.

Çizelge 6.85. Tur anahtarı ekledikten sonra 3

ec	19	e7	0e
79	ed	d3	4a
76	01	64	34
16	43	f5	bd

Çizelge 6.86. Şifre çözme durum Başlangıç Turu 2

9f	fb	4f	91
dc	d1	c4	b8
f4	94	9d	2a
42	08	b3	ce

Çizelge 6.87. Ters Satır Öteleme Sonrası 2

9f	fb	4f	91
b8	dc	d1	c4
9d	2a	f4	94
08	b3	ce	42

Çizelge 6.88. Ters Bayt Yer Değiştirme Sonrası 2

6e	63	92	ac
9a	93	51	88
75	95	ba	e7
bf	4b	ec	f6

Çizelge 6.6.'daki 2. tur anahtarı eklenir. Çizelge 6.89.-6.92. arasında 2. tur döngüsü gösterilmiştir.

Çizelge 6.89. Tur anahtarı ekledikten sonra 2

d8	07	2c	c4
08	ae	ca	b8
ba	28	7f	54
b4	ba	ec	08

Çizelge 6.90. Şifre çözme durum Başlangıç Turu 1

84	0a	5c	71
c5	11	d3	b6
41	53	e5	a1
de	73	1f	46

Çizelge 6.91. Ters Satır Öteleme Sonrası 1

84	0a	5c	71
b6	c5	11	d3
e5	a1	41	53
73	1f	46	de

Çizelge 6.92. Ters Bayt Yer Değiştirme Sonrası 1

4f	a3	a7	2c
79	07	e3	a9
2a	f1	f8	50
8f	cb	98	9c

Çizelge 6.5.'deki 1. tur anahtarı eklenir. Çizelge 6.93.-6.96. arasında 1. tur döngüsü gösterilmiştir.

Çizelge 6.93. Tur anahtarı ekledikten sonra 1

99	71	7d	fa
d3	a8	45	02
5e	83	80	26
72	31	69	62

Çizelge 6.94. Şifre çözme Son Tur Başlangıcındaki Durum

47	23	d7	67
1b	b2	67	fa
4c	07	93	d8
76	fd	f2	f9

Çizelge 6.95. Son tur Ters Satır Öteleme Sonrası

47	23	d7	67
fa	1b	b2	67
93	d8	4c	07
fd	f2	f9	76

Çizelge 6.96. Son tur Ters Bayt Yer Değiştirme Sonrası

16	32	0d	0a
14	44	3e	0a
22	2d	5d	38
21	04	69	0f

Çizelge 6.4.'deki başlangıç tur anahtarı ile Çizelge 6.96.'daki son tur ters bayt yer değiştirme sonrası XOR işlemine koyulur ve Çizelge 6.97.'deki açık metin 16'lık sistemde elde edilir.

Çizelge 6.97. Son durum hex

16	36	05	06
15	41	37	07
20	2b	57	36
22	03	62	00

Bu çıkan 16’lık sistemdeki rakamları 10’luk sisteme çevirirsek;

22
21
32
34
54
65
43
3
5
55
87
98
6
7
54
0

rakamlarını buluruz. Bu rakamlarında girdiğimiz rakamlarla aynı olduğunu görmüş oluruz.

7.SONUÇ

Kriptografik sistemlerin güvenliği verilerin güvenli iletilmesi için çok önemlidir. Kriptografi biliminde bir grup mevcut sistem daha güvenli olsun diye araştırmalar yapıp sistemi oluştururken, diğer bir grup güvenilirliği artırılmış olan sistemde analizler yapıp güvenlik sisteminin açıklarını aramakta ve eğer başarılı olabiliyorsa güvenlik açıklarını göstermektedir. Bu döngü içerisinde kriptografi bilimi gelişmektedir.

Şifreleme, bilgiyi matematiksel işlemleri kullanarak veya bilgiyi belli bir algoritmaya göre yer değiştirme işlemi yaparak karmaşık hale getirerek gerçekleştirir.

Günümüzde teknolojinin gelişimiyle daha hızlı ve güçlü bilgisayarlar hayatımıza girmiştir. Bu bilgisayar teknolojileri bize birçok alanda fayda sağlarken bilgi güvenliğinin korunmasını güç hale getirmektedir. Hızlı ve güçlü bilgisayarlarla şifreleme algoritmalarına karşı yapılan saldırılarda güçlenmektedir ve daha hızlı sonuç vermektedir.

Kripto sistemler şifrelemede kullanıldıkları anahtara göre açık anahtarlı sistemler ve gizli anahtarlı sistemler olmak üzere iki ana kategoriye ayrılır. Kapalı anahtarlı sistemlerde şifreleme ve deşifreleme işlemleri yapılırken aynı anahtar kullanılırken, açık anahtarlı sistemlerde bu işlemler için açık ve gizli olmak üzere iki anahtar kullanılmaktadır. Bu anahtarlar kullanılan algoritmaya göre değişik uzunlukta olabilirler.

Yapılan çalışmada günümüzün en yüksek güvenlikli gizli anahtarlı algoritmalarından olan Gelişmiş Şifreleme Standardı ile hızlı, basit ve güvenli iletişimin mümkün olduğunu gördük.

Günümüzde bilgi güvenliği çok önemli olduğu için bilginin taşınabilmesi için, Amerika'da kriptoloji enstitüleri tarafından da standart olarak kabul edilen Gelişmiş Şifreleme Standardının matematiksel özellikleri, anahtar uzunlukları ve standardı oluşturan katmanlar hakkında detaylı bilgi verilmiştir.

Yapılan simülasyonda istediğimiz anahtarı ve açık metini girmenin ardından, gizli anahtarımızdan her bir tur için anahtar genişlettik ve açık metini genişlettiğimiz anahtarla XOR işlemine tabi tutup Gelişmiş Şifreleme Standardına ait şifreleme katmanlarını uyguladık ve şifreli metini elde ettik. Daha sonra anahtarını bildiğimiz şifrelenmiş metinden, şifrelemedekine benzer şekilde anahtarı genişleterek Gelişmiş Şifreleme Standardına ait deşifreleme katmanları uygulayıp açık metini elde ettik.

Yapılan çalışma, karakterlerin ASCII kodları kullanılıp pratik uygulamalara uyarlanabilir.

Simülasyon özellikle elektronik ve bilgisayar mühendislikleri açısından çok önemli olan MATLAB ile yapılmıştır. Matematiksel işlemlerin kolay yapılabilmesi nedeniyle simülasyonda MATLAB tercih edilmiştir.

KAYNAKLAR

1. İnternet : Pro-G ve Oracle, “Bilişim Güvenliği Sürüm 1.1”, http://www.pro-g.com.tr/whitepapers/bilisim-guvenligi-v1.pdf, (2003).

2. Doğan, A. Y. , “AES Algoritmasının FPGA Üzerinde Düşük Güçlü Tasarımı”, Yüksek lisans, ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 4-10 (2008).

3. Şahin, A. , Buluş, E. , Sakallı, M. T. , “Modern Blok Şifreleme Algoritmalarının Gücünün İncelenmesi”, ***II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi***, İstanbul, 86-92 (2005).

4. Sakallı, M. T., Buluş, E., Şahin, A., Büyüksaraçoğlu, F., “AES S kutusuna benzer 4-bit girişe ve 4-bit çıkışa sahip S kutularının tasarımı”, ***II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi***, İstanbul, 16-23 (2005).

5. Aslan, B., Sakallı, M. T., “S-kutularının kriptografik özellikleri”, ***ELECO-2008***, Bursa, 24 (2008).

6. Gülaçtı, E. , “Açık Anahtar Altyapısı Eğitim Kitabı", ***TÜBİTAK UEKAE***, Ankara (2006).

7. İnternet : Cyber-Warrior “Kriptoloji-Kriptografi Seminerleri Sonuç Bildirgesi”, http://www.kriptoloji.net/sonucbildirgesi.pdf (2009).

8. Matsui, M., “Linear Cryptanalysis Method for DES Cipher”, ***Advances in Cryptology- Eurocrypt '93***, Springer-Verlag, 386-397, (1994).

9. Biham, E. , Shamir, A. , “Differential Cryptanalysis of the full 16-round Data Encryption Standard”, ***Advances in Cryptology: Proceedings of CRYPTO’92***, Springer-Verlag, Berlin, 487–496 (1993)

10. Knudsen, L. R. , “Truncated and Higher Order Differentials”, ***Fast Software Encryption***, Springer-Verlag, 196–211, (1995).

11. Daemen, J. , Knudsen, L. R. , Rijmen, V. , “The Block cipher Square”, ***Fast Software Encryption***, Springer Verlag, New York, 149–165 (1997).

12. Soğukpınar, İ. , “Veri ve Ağ Güvenliği Ders Notları”, ***GYTE Bilgisayar Mühendisliği***, 44-101 (2005).

13. Yerlikaya, T. , Buluş, E. , Arda, D. , “Asimetrik Kripto Sistemler ve Uygulamaları”, ***II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi***, İstanbul, 24-31 (2005).

14. Schneider, B., “Applied cryptography second edition”, ***John Wiley&Sons, Inc.***, New York (1996).

15. Yerlikaya, T. , Buluş, E. , Buluş, N. , “Asimetrik Şifreleme Algoritmalarında Anahtar Değişim Sistemleri”, ***Akademik Bilişim Konferansları***, Denizli, (2006).

16. Levi, A., Özcan, M., “Açık Anahtar Tabanlı Şifreleme Neden Zordur?”, ***Türkiye Bilişim Derneği 19. Bilişim Kurultayı***, İstanbul, 41-45 (2002).

17. Diffie, W., Hellman, M. E., “New directions in cryptography”, ***IEEE Transactions on Information Theory***, vol. 22, 644–654, (1976)

18. Adams, C., Tavares, S., “The Use of Bent Sequences to Achieve Higher-Order Strict Avalanche Criterion in S-Box Design”, ***Technical Report TR 90-013***, Canada (1990)

19. Kaya, M., “Kullanıcı Uzayında Dinamik Boyutlu ve Şifreli Bir Dosya Sisteminin Gerçeklenmesi”, Yüksek lisans, ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 20-28 (2008).

20. Yıldırım, K. , Demiray, H. E. , “Simetrik ve Asimetrik Şifreleme Yöntemlerine Metotlar: Çırpılmış ve Birleşik AKM-VKM”, Yüksek lisans, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, Kocaeli (2008).

21. Şahinoğlu, M., “AES Algoritmasının FPGA Üzerinde Gerçeklemesine Elektromanyetik Alan Saldırısı”, Yüksek lisans, ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 4-17 (2008).

22. Kayış, H. , “AES Uygulamasının FPGA Gerçeklemelerine Karşı Güç Analizi Saldırısı”, Yüksek lisans, ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 2-10 (2006).

23. National Institute of Standards and Technology, “Secure Hash Standard”, ***FIPS PUB 180–1***, USA, 4-8 (1995).

24. Jamil, T., “The Rijndael Algorithm, Potentials”, ***IEEE***, 36-38 (2004).

25. Sanchez-Avila, C., Sanchez-Reillo, R., “The Rijndael Block Cipher (AES Proposal): A Comparison with DES, Security Technology”, ***2001 IEEE 35th International Carnahan Conference***, London, 229-234 (2001).

26. Xiao, Y., Sun, B., Chen, H.H., Guizani, S., Wang, R., “PerformanceAnalysis of Advanced Encryption Standard (AES)”, ***IEEE GLOBECOM 2006 - 2006 Global Telecommunications Conference***, San Francisco, United States, 1-5 (2006).

27. Zhang, X., Parhi, K.K., “Implementation Approaches for the Advanced Encryption Standard Algorithm”, ***IEEE Circuits and Systems Magazine***, 24-46 (2002).

28. Başkök, M. D., “AES Şifreleme Algoritmasının Modellenmesi”, Yüksek lisans, **Gazi Üniversitesi Fen Bilimleri Enstitüsü**, Ankara, 9-13 (2007)

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : ULUTÜRK, Abdullah
Uyruğu : T.C.
Doğum tarihi ve yeri : 20.05.1983, İzmit
Medeni hali : Bekar
Telefon : 0 (505) 437 58 41
e-mail : auluturk@aselsan.com.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Lisans	Gazi Üni./ Elk-Elektronik Müh.	2005
Lise	Adana Erkek Lisesi (Y.D.A)	1996

İş Deneyimi

Yıl	Yer	Görev
2007- Devam	ASELSAN	Elektronik Müh.

Yabancı Dil

İngilizce

Hobiler

Tenis, Futbol, Voleybol, Bilgisayar teknolojileri, Basketbol