Doktora Tezi
Trakya Üniversitesi Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Bölümü

ÖZET

Bu tezin amacı, günümüzde yaygın olarak kullanılan simetrik ve asimetrik şifreleme algoritmalarının yapılarının incelenmesi, yakın zamanda oluşturulmuş algoritmaların incelenerek karşılaştırılması, uygulamalarının gerçekleştirilmesi ve bu algoritmalara karşı yapılan güncel saldırıları tekniklerinin araştırılmasıdır.

Tezin giriş kısmı olan birinci bölümünde şifreleme algoritmalarının temel yapıları ve sınıflandırılması yapılmıştır. Kriptoloji bilimin önemi üstünde durulmuş ve simetrik ve asimetrik şifreleme algoritmalarının temel yapısı verilmiştir.

Tezin ikinci bölümünde asimetrik şifreleme algoritmalarının yapısın daha anlaşılır olması için matematiksel teoremler verilmiştir. Asimetrik şifreleme sistemleri için çok önemli olan asal sayı tanımları verilmiştir.

RSA şifreleme algoritması günümüzde yaygın olarak kullanılan en önemli asimetrik şifreleme algoritmalarından biridir. Üçüncü bölümde RSA şifreleme algoritmasının yapısı incelenmiş ve Bu algoritmaya karşı yapılan saldırı tekniği incelenmiştir.

Tezin dördüncü bölümünde eliptik eğri şifreleme algoritması (ECC) incelenmiştir. Eliptik eğri şifreleme algoritması asimetrik şifreleme algoritmalarının en büyük dez avantajı olan çok büyük asal sayılar kullanmadan daha düşük anahtar değerleriyle aynı güvenliği sağlayan bir algoritmadır. Bu bölümde eliptik eğri şifleme algoritması ve bu algoritmaya karşı yapılan saldırı teknikleri incelenmiştir. Pollard Rho saldırı tekniği ayrıntılı olarak incelenmiştir.

Simetrik şifreleme algoritmalarından DES ve AES tezin beşinci bölümünde incelenmiştir. Bu algoritmaların şifreleme ve deşifreleme işlemleri nasıl gerçekleştirdiği araştırılmıştır ve AES şifreleme algoritması seçilirken, finalist olan diğer dört algoritmalar belirli krıterlere göre karşılaştırılmıştır.

Tezin altıncı bölümünde, DES ve AES şifreleme algoritmaları kullanılarak resmin içine gömülen şifrelenmiş metinlerin, açık metine göre örtü nesnesi üzerindeki etkileri gösterilmiştir.

Son bölümde tezde verilen teorik açıklamaların uygulamaları yapılmış ve hazırlanmış olan yazılım programları, performans analizleri ve sonuçlar verilmiştir. Öncelikle RSA şifreleme algoritmasının programı gerçekleştirilmiş ve sonuçları verilmiştir. RSA şifreleme algoritmasının kriptanalizi gerçekleştirilmiş ve sayısal örneği verilmiştir. Eliptik eğri şifreleme algoritmasının C++ ta uygulaması gerçekleştirilmiştir. Pollard Rho algoritmasının akış şeması verilmiş ve eliptik eğri şifreleme algoritmasının kriptanaliz uygulaması gösterilmiştir. AES şifreleme algoritmasının performans analizi 128, 192, ve 256 bit anahtar kullanılarak sonuçları verilmiştir. Son olarak DES ve AES şifreleme algoritmalarının Steganografi üzerinde uygulaması ve sonuçları verilmiştir.

Anahtar Sözcükler: Asimetrik Şifreleme Algoritmaları, Simetrik Şifreleme Algoritmaları, RSA, Eliptik Eğri Şifreleme, DES(Data Encryption Satandard), AES (Advanced Encryption Algorithm), Kriptanaliz, Şifreleme Algoritmalarına Saldırılar

Yıl: 2006
Sayfa: 139

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

ABSTRACT

The purpose of this thesis is to examine Symmetric and Asymmetric crypto algorithms that are widely used today, examine and compare recent future developed algorithms, realize applications and study attack methods done/performed to this algorithms.

In first section which is entry of thesis, crypto algorithms' basic structures and classification are studied. Importance of cryptology science is mentioned and symmetric and asymmetric crypto algorithms' basic structure is given.

In thesis second section, for improving comprehension of asymmetric crypto algorithms, mathematical theorems are given. Prime numbers' definition which are most important thing for asymmetric cryptology systems are given.

Widely used today RSA crypto algorithm is one of most important asymmetric algorithm. In third section RSA crypto algorithms structure is examined. Attack method performed to this algorithm is studied

In thesis forth section ECC crypto algorithm is examined. ECC is an algorithm which performs same security level using small key definitions despite asymmetric algorithms' obligatory for bigger prime number usage. in this section, attack methods performed to this algorithm are studied. Pollard rho attack method is examined in detail.

In section fifth, DES (Data Encryption Standard) and AES (Advanced Encryption Standard) algorithms are examined. In the last part of this section, the AES finalists are compared with certain properties.

In the sixth section, a selected sample text was encrypted by using symmetric encryption algorithms AES and DES. The sample text and the two encrypted text using DES and AES are concealed into 24 bit bmp selected image. Selected image was designed using LSB insertion method. The changes on the cover objects are examined.

In the last section, Firstly, application program of RSA crypto algorithms is given in Delphi 6.0. and shown the performance analysis. Attack method on RSA algorithm is shown in program application. Secondly, the program of elliptic curve cryptography and numerical example are given and example of Pollard Rho algorithm is shown in detail. The software of AES encryption algorithms is written in C++ and given the performance analysis in the last section of the thesis. In the end of the this section, Selected image was designed using LSB insertion method. The changes on the cover objects are examined while encrypt the plan text with AES and DES. The application program and the results are given.

Keywords: Asymmetric Encryption Algorithms, Symmetric Encryption Algorithms, RSA, Elliptic Curve Cryptography, DES (Data Encryption Standard), AES (Advanced Encryption Algorithm), Cryptanalysis, Cryptanalytic Attacks against Encryption Algorithms

Year: 2006
Page: 139

TEŞEKKÜR

Bu önemli ve güncel konuda çalışmamı sağlayan, bana yol gösteren, destek ve yardımlarını benden esirgemeyen, konu hakkındaki bilgisinden yararlandığım ve çalışma hayatımda bana her konuda destek olan danışman hocam Sayın Yrd. Doç. Dr. Ercan BULUŞ'a sonsuz teşekkür ederim.

Bana olan destekleri ve katkıları için Maltepe Üniversitesi Rektörü Sayın Prof. Dr. Mesut RAZBONYALI ve Ege Üniversitesi Bilgisayar Mühendisliği Öğretim Üyesi Sayın Prof. Dr. Şaban EREN'e sonsuz teşekkürlerimi sunarım.

Tez izleme komitesinde ve tez jürisinde yer alan sayın hocalarım Yrd. Doç. Dr. Aydın CARUS, Yrd. Doç. Dr. Rembiye KANDEMİR ve Yrd. Doç. Dr Şaban AKTAŞ'a tez çalışmama yaptıkları olumlu eleştiriler ile katkıda bulundukları için sonsuz teşekkürlerimi sunarım.

Tez çalışmam sırasında bana her zaman destek olan ve sabır gösteren babam Mehmet YERLİKAYA ve annem Hafiz YERLİKAYA'ya çok teşekkür ederim.

Ayrıca çalışma hayatım da bana destek olan ve yardımlarını esirgemeyen Bilgisayar mühendisliği bölüm hocalarıma ve çalışma arkadaşlarıma çok teşekkür ederim.

Temmuz 2006 **Tarık YERLİKAYA**

1. GİRİŞ

Kriptoloji, Yunanca krypto's (saklı) ve lo'gos (kelime) kelimelerinin birleştirilmesinden oluşturulmuştur ve iletişimde gizlilik bilimi olarak değerlendirilmektedir[1].

Ticari ilişkilerde, devlet işlerinde, askeri işlerde ve personel ilişkilerinde güvenli iş çalışması yapmak büyük bir sorundur.

Sistemler arası bağlantılarda ya da herhangi iki nokta arasındaki haberleşmede verinin güvenli bir şekilde gittiğinden emin olmak gerekir. Bunun sağlanması ise gönderilen verinin şifrelenmesi ile olur. Böylece açık haberleşme kanalları kullanılarak verinin güvenli bir şekilde ulaştırılması sağlanır. İletişimde, açık bir haberleşme kanalı kullanılıyorsa gizli tutulmak istenen bilginin yetkisiz bir kişi tarafından dinlenebileceği veya haberleşme kanalına girip (araya girme) veriyi bozabileceği ya da değiştirebileceği (yanlış verinin gönderilmesi) düşüncesi her zaman için önemli bir problem oluşturur.[2]

Kriptoloji esas olarak iki bölüme ayrılır, Kriptografi (şifreleme) ve kriptanaliz (şifre çözme). Gönderilmek istenen orijinal mesaj açık mesaj (plain text) ve bu mesajın şifrelenmiş hali şifreli mesaj (cipher text-cryptograph) olarak adlandırılır.[3] Şifreleme, askeri ve diplomatik iletişimde (haberleşmede) güvenliği sağlamak için bin yıldır kullanılmaktadır. Ancak bugün artık özel sektörde de gereksinim duyulmaktadır. Sağlık hizmetleri, finansal işler (örneğin: kredi oranları) gibi konularda bilgisayarlar arasındaki haberleşmede açık kanallar kullanılarak yapılmaktadır. Bu açık kanalların kullanılması sırasında yukarıda sayılan işlerin güvenli ve gizli bir şekilde yapılabilmesi için şifrelemeye gerek duyulmaktadır.[3]

1.2. Kripto Algoritmalarına Giriş

Şifreleme işlevinin güvenli bir şekilde gerçekleştirilmesi şifreleme sırasında kullanılan tüm yöntem ve bilgilerin gizliliğine dayanır. Ancak herhangi bir nedenle şifreleme işlevlerinin açığa çıkabileceği düşünülerek iletişim güvenliği, şifreleme anahtarı denen ek bilgi ile arttırılmıştır. Bu durumda şifreleme işlemi sırasında açık mesaj, şifre anahtarı aracılığı ile şifrelenir. Bir başka deyişle açık mesaj ile şifrelenmiş mesaj arasındaki geçişler şifreleme algoritmasına bağlı olduğu kadar kullanılan anahtar bilgisine de bağlıdır.

Günümüzde kullanılan kriptografik algoritmalar ikiye ayrılır. Bunlar, kullandıkları anahtar biçimine göre simetrik veya asimetrik olarak adlandırılırlar.[4]

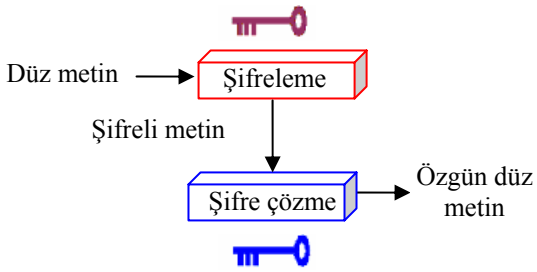
1.2.1. Simetrik şifreleme algoritmaları

Simetrik şifreleme algoritmaları şifreleme ve deşifreleme işlemleri için tek bir gizli anahtar kullanmaktadır. Şifreleme işlemlerini gerçekleştirdikten sonra şifreli metni alıcıya gönderirken şifreli metinle birlikte gizli anahtarı da alıcıya güvenli bir şekilde göndermesi gerekmektedir. Simetrik şifreleme algoritmaları çok hızlı şifreleme ve deşifreleme işlemleri gerçekleştirebildiğinden dolayı günümüzde çok yaygın olarak kullanılmaktadır Bu algoritmalara örnek olarak DES ve AES şifreleme algoritmalarını örnek verebiliriz. [5],[6], [11]

1.2.2. Asimetrik şifreleme algoritmaları

Simetrik şifreleme tekniğinde bulunan anahtar dağıtım problemini çözmek için şifreleme ve çözme işlemlerinin her birisi için ayrı ayrı anahtar kullanma prensibine dayanan bir şifreleme sistemi geliştirilmiştir.

Bu sistemde şifreleme işlemi herkes tarafından bilinen açık anahtarla yapılır. Şifreleme ve çözme işlemi birbirinin simetriği olmayan (yani aynısı olan) algoritmalarla gerçekleştirildiğinden dolayı da asimetrik şifreleme sistemi olarak bilinir. [7],[8],[24]



Şekil 1. Asimetrik Şifreleme Algoritmalarının Yapıları[24]

1.2.3. Anahtar yönetimi

Asimetrik algoritmalarda ise her iki kullanıcı veriyi farklı anahtar bilgisi ile kriptolar ve çözerler. Bu amaçla her bir kullanıcı biri gizli diğeri açık iki anahtar kullanır. Açık anahtar bilgisi alıcı tarafa herhangi bir koruma yapılmadan iletilir. Açık anahtarı alan taraf bu bilgiden kriptoyu çözmek amacı ile kullanacağı gizli anahtarı üretir. Açık anahtarın üçüncü bir kişinin eline geçmesi tek başına hiç bir şey ifade etmez. Her ne kadar bu yöntemin, anahtarların alıcı tarafa iletilmesi işlemini basitleştirdiği düşünülse de asimetrik algoritmaların işlem süresinin yüksek oluşu, veri kriptolamada yaygın olarak kullanılmalarını engellemektedir. Hem simetrik hem de asimetrik algoritmaların temel özelliği yapılarının açık olarak bilinmesidir. Böylece kripto algoritmalarının tasarımında, şifrelenmiş verinin anahtar bilgisi olmadan şifrenin

çözülememesi ilkesi göz önüne alındığından, iletişim güvenliği, kripto anahtarlarının güvenliği problemine indirgenmiştir.[9]

Kriptografik anahtarların üretilmelerinden dağıtılmalarına ve kullanılmalarına kadar olan sürecin sıkı bir denetim altına alınması kriptografik işlemlerin güvenilirliği açısından büyük öneme sahiptir. Bu işlemlerin tümüne anahtar yönetim denir. Anahtar yönetiminin en zayıf halkası, dış müdahalelere tümüyle açık olan anahtarların güvenli bir şekilde dağıtılması işlevidir. Geleneksel olarak kriptografiden yararlanan, askeri ve diplomatik servisler, bu problemi, anahtarları, güvenilir kuryeler aracılığı ile ileterek çözmüşlerdir. Günümüzde bu yöntem hala kullanılmaktadır. Ancak bu oldukça pahalı ve yavaş bir yöntemdir.

Günümüzde kişilerin veya kuruluşların gereksindiği bilgi miktarı geçmiş on yılla karşılaştırıldığında hızla yükselmiştir. Bunun sonucu olarak iletişim olanakları ile birlikte iletişim ortamları da değişmiştir. Bugün artık sistem tasarımcıları çağın gerektirdiği bilgiyi işlemek üzere bilgisayarları geliştirmişler ve verilerin hızlı bir şekilde iletilebilmesini sağlamak amacı ile veri ağlarını tasarlamışlardır. Veri işlenmesinde kullanılan bilgisayar ağları, veri işleyen ana bilgisayar birimlerinden, kontrol birimlerinden ve terminallerden oluşur. Tüm bu yapılar birbirlerine veri bağları ile bağlıdırlar. Veri bağları uydu ve mikrodalga bağlantıları olabileceği gibi anahtarlanmış veya anahtarlanmamış iletişim hatları da olabilirler. Veri ağı birbirine veri bağları ile bağlı düğümlerden (ana bilgisayarlar) ve veri ağının yalnızca ana bilgisayarlar üzerinden kullanabilen terminallerden oluşmaktadır. Veri ağının temel işlevi bir kullanıcının (kişi veya programın) diğer kullanıcı ile haberleşmesi için gerekli erişim yolunu sağlamaktır. Veri ağlarının bu yapısı isteyen kullanıcının bu sistemleri kullanmasını sağlamakla birlikte ortaya iletişim sırasında kullanılan gizli mesajların korunması problemini de çıkarmaktadır. Her ne kadar mesaj güvenliği kriptografik işlemlerle sağlansa da kripto anahtarlarının güvenli bir biçimde üretilmesi ve kullanıcılara dağıtılması problem olmaya devam etmektedir.[10]

Kriptografik anahtar yönetimini ana bilgisayar ve terminallerden oluşan veri ağlarında gerçekleştirebilmek için anahtarların üretilmesinden saklanmasına ve

kullanım amacı i!e dağıtılmasına kadar olan sürecin denetlenmesi gerekmektedir. Güvenli iletişimin gerçekleştirileceği ortamın askeri, diplomatik veya sivil olması, kullanılacak anahtar yönetim sistemini de belirlemektedir. Askeri uygulamalarda anahtar üretimi genellikle, Merkezi Denetim Sistemi (MDS) denilen tek bir merkezde yapılmakta ve çağrı kurmak isteyen kullanıcılar bağ!ı bulundukları düğüm aracılığı ile bu merkeze çağrı isteklerini iletmektedirler. Bu istek üzerine üretilen anahtarlar şifrelenmiş biçimde oturum anahtarı bilgisini çağrıyı kurmak isteyen terminale iletirler. Bu yöntemin avantajı, anahtar yönetimini tek bir merkezden belirlediği için güvenlik denetiminin tek bir ana bilgisayar üzerinde olmasının yeterli oluşudur. Askeri uygulamalar için oldukça uygun olan bu yapının birinci dezavantajı, güvenli iletişim başlamadan önce tüm kullanıcıların tek bir merkezden oturum anahtarı almak zorunluluğunda oluşlarından dolayı MDS üzerindeki işlem yükünün veri ağına bağlı kullanıcı sayısına ve çağrı istek yoğunluğuna bağlı olarak artması, dolayısıyla çağrı kurulma süresinin uzamasıdır. İkinci dezavantajı ise MDS ile oluşabilecek herhangi bir iletişim kopukluğunun güvenli iletişimi tamamen durdurma riskidir.

Bir diğer yöntem MDS'nin işlevlerinin ana bilgisayarlara yüklendiği, tek bir merkez yerine veri ağını oluşturan tüm ana bilgisayarlara dağıtılmış anahtar yönetim sistemidir. Bu yöntemin ana ilkeleri IBM tarafından konmuştur. Bu sistemin amacı özellikle sivil uygulamalarda kullanılabilecek, tek bir MDS'nin getirdiği risklerden arınmış, yeni bir anahtar yönetimi ortaya koymaktır.

Bu ortamda herhangi bir A terminaline bağlı kullanıcı, güvenli bir haberleşme gerçekleştirmek istediği zaman çağrı isteğini bağlı bulunduğu ana bilgisayar A'ya iletir. Ana bilgisayar, gelen oturum başlatma isteği mesajını aldığında iletişim sırasında kullanılacak olan KS oturum anahtarını üretir ve bunu A terminaline, ilgili terminalin K_{MTa} master anahtarını kullanarak kriptolayıp $K_{KMTa}(KS)$ biçiminde iletir. Aynı anahtar ve oturum çağrı isteği, karşı terminalin bağlı bulunduğu ana bilgisayar B'ye, ana bilgisayarlar arasında kutlanılan tek yönlü bir anahtar olan K_{NCab} eşliğinde, $E_{KNCab}(KS)$ biçiminde iletilir. Çağrı isteğini alan ana bilgisayar, gerekli oturum anahtarını, çağrılan terminalin master anahtarını kullanarak ilgili terminale $E_{KMTb}(KS)$ biçiminde iletir. Bu işlemler sırasında her iki terminal birbirlerinin kimlik verilerini karşılıklı olarak

sorgularlar. Bu sorgulamanın amacı, iletişimi gerçekleştirecek olan her iki kullanıcının, karşı tarafın, A veya B terminali yerine, bir düşman terminal olmasından şüphe etmesidir. Bu işlemin tamamlanmasından sonra kullanıcılardan her biri, M açık mesajını $E_{KS}(M)$ biçiminde kriptolayıp diğer kullanıcı kriptolanmış mesajı $D_{KS}(E_{KS}(M))$ biçiminde çözerek haberleşebileceklerdir.

Oturum anahtarının bu şekilde dağıtılması, haberleşme sırasında kullanıcının dışarıdan bir anahtar yükleme gereksinimini ortadan kaldıracaktır. Her terminal için özel olan ve terminallere manüel olarak yüklenen terminal master anahtarlarının değişme gereksinimi sık olmadığından güvenli haberleşme, ana bilgisayarlar tarafından üretilen oturum anahtarları ile gerçekleşir. Bunun sonucunda, veri anahtarlarının haberleşme ortamının dışında ayrı bir güvenli kanaldan iletim gereksinimi ortadan kalkmaktadır. Bu çalışmada IBM PC ortamında çalışabilecek simetrik bir kripto algoritması örneği olarak DES (Veri Şifreleme standardı) kullanılmıştır. Ayrıca DES algoritmasını kullanan, bir yan rassal anahtar üretim fonksiyonu, tek yönlü hash fonksiyonu standardı olan MD4 kullanılarak gerçekleştirilmiştir. Anahtar yönetim fonksiyonları, terminal ve ana bilgisayar uygulamaları olarak ayrı ayrı tasarlanmıştır. Bu fonksiyonlar ana bilgisayar ve terminallerden oluşan bir ağ topolojisinde kullanılacak örnek bir anahtar yönetim protokolünde kullanılmışlardır.

1.2.4. Kriptoloji Temelleri

Kriptoloji gizli sistem oluşturma veya çözme bilimidir. Bunun tasarım veya gizliliği oluşturma kısmına kriptografi(şifre bilimi), çözme veya gizliliğin ortadan kaldırılma kısmına ise kriptoanaliz denir. .

Bir mesajın anlaşılabilirliğini gizlemek amacı ile iki yöntem uygulanabilir. Bunlardan birincisi mesajın belirli bir yöntem uyarınca kısaltılması, diğeri ise belirli bir teknikle mesajın anlaşılamaz bir biçime dönüştürülmesidir. Bunlardan ilkine steganografi, diğerine ise kriptografı denir.[1]

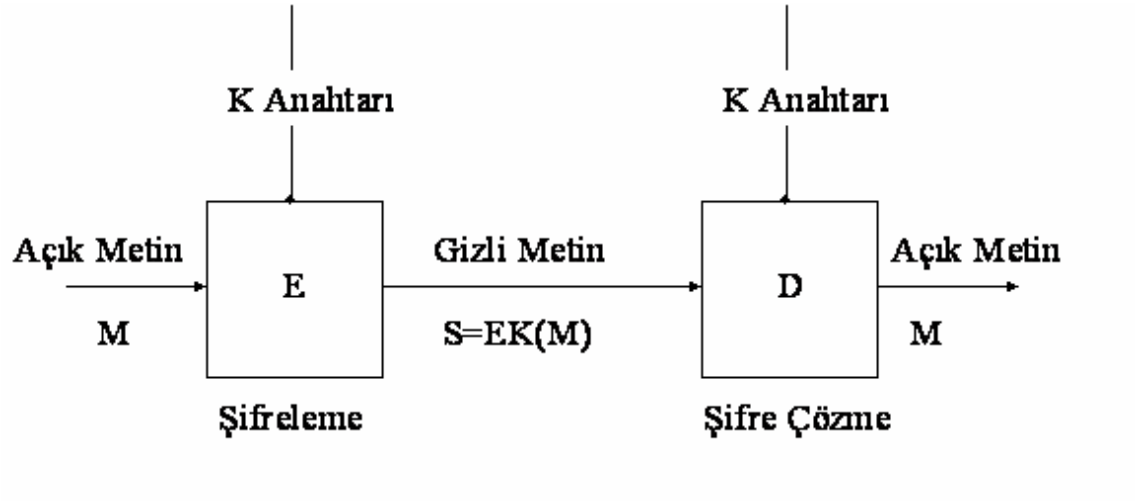
Verinin orijinal biçimindeki haline açık metin denir. Dönüştürülmüş biçimine ise şifrelenmiş metin veya gizli metin denir. Dönüştürme işlemine şifreleme veya kriptolama diyoruz. Dönüştürülmüş metnin ters işlem sonucu açık metin halinin elde edilmesine ise şifre veya kripto çözme denir. M açık metni, E ve D simgeleri de kriptolama ve kripto çözme işlemlerini temsil ettiğinde, gizli metin S,

$$S = E_K(M)$$

olarak gösterilir. Gizli metinden açık metnin elde edilmesi amacı ile kripto çözme işlevinde kullanılması,

$$M = E_K(S)$$

olarak gösterilir. Her iki ifadede kullanılan K şifreleme anahtarını temsil etmektedir. Şekil 1.2 bu iki işlemi göstermektedir.



Şekil 1.2: Şifreleme ve şifre çözme işlemleri[12]

Burada K anahtarını kullanmanın amacı, eğer M açık metni sadece E kripto işlevi ile şifrelenirse gizliliği sağlayacak olan tek parametrenin kripto algoritması olması ve iletişim güvenliğini arttırmak için alıcı tarafın her yeni kripto işlevi için farklı bir kripto algoritmasını bilmesinin gerekmesidir. Eğer mesajın şifrelendiği E işlevi kaybolacak olursa yeni bir algoritma tasarlanmak durumundadır. Bu durumda zaman kaybı da göz önüne alınacak olursa bu oldukça pahalı bir yöntemdir. Bunun yanı sıra K

anahtarı mesajın kripto algoritması bilinse de ikinci bir güvenlik parametresi olduğundan iletişim güvenliğini arttırmaktadır

Şifrenin üçüncü kişiler tarafından çözülebilmesine kripto analitik saldırı denir. Kripto analitik saldırı üç biçimde gerçekleşebilir, bunlar gizli metin saldırısı, bilinen açık metin saldırısı ve seçili açık metin saldırısıdır. Adından da anlaşılabileceği gibi gizli metin saldırısı kripto analizcinin yalnızca gizli metni elde etmesi ile gerçekleşir. Sadece şifrelenmiş metinden hareket edilerek çözülebilen kripto işlevine sahip sistemler kullanılması uygun değildir. Bilinen açık metin saldırısında, kripto analizci, elinde bulunan açık metin ile gizli metin çifti arasında ilişkiler bularak sistemi çözer. En güçlü saldırı yöntemi seçili açık metin saldırısıdır. Bu yöntemde kripto analizci kendisinin belirlediği bir açık metine karşılık düşen gizli metin yardımı ile kullanılan anahtarı belirlemeye çalışır. Bu yöntemde eğer algoritma bazı açık metinler karşısında anahtar bilgisini dışarıya sızdırıyor ise istenen amaç elde edilmiş olur.[12]

Kriptografik sistemlerin güvenli olabilmesi iki şekilde mümkündür. Birinci durumda kripto analizi gerçekleştirmeye çalışan kişi, elinde tüm olasılıkları denemek için ne kadar işlem gücü olursa olsun, hiç bir zaman, gizli mesajdan açık mesaja dönüşümü sağlayacak yeterli veriyi elde edemez. Bu durumda sisteme koşulsuz olarak güvenli denir. İkinci durumda ise kripto analizi gerçekleştiren kişinin elinde yeterli veri olsa da kripto analitik problemin çözüme ulaşabilmesini sağlayacak yeterli işlem gücüne sahip değildir. Bu tür sistemlere işlemsel olarak güvenli sistemler denir

1.2.5. Şifreleme Algoritmalarının Performans Kriterleri

Bir şifreleme algoritmasının performansını şu kriterlere göre belirleyebiliriz.[12]

- Kırılabilme süresinin uzunluğu.
- Şifreleme ve çözme işlemlerine harcanan zaman (Zaman Karmaşıklığı).

- Şifreleme ve çözme işleminde ihtiyaç duyulan bellek miktarı (Bellek Karmaşıklığı).
- Bu algoritmaya dayalı şifreleme uygulamalarının esnekliği.
- Bu uygulamaların dağıtımındaki kolaylık ya da algoritmaların standart hale getirilebilmesi.
- Algoritmanın kurulacak sisteme uygunluğu.

2. ASİMETRİK ŞİFRELME ALGORİTMALARINDA KULLANILAN MATERYAL VE METOTLAR

İlk bölümde şifreleme algoritmalarının yapılarını ve bu algoritmaların sınıflandırılmasını inceledik. Günümüzde birçok haberleşme ortamında kullanılan asimetrik şifreleme algoritmalarının alt yapısını matematiksel teoremler oluşturmaktadır. Asimetrik şifreleme algoritmaları zor matematiksel problemler üzerine oturtulmuş şifreleme algoritmalardır. Bu algoritmaların daha iyi anlaşılması ve analizlerinin gerçekleştirilebilmesi için öncelikle teoremler üzerinde durulması gerekmektedir.

2.1. Teoremler

2.1.1. Bölme Algoritması ve Bölünebilirlik

Sayılar teorisinin temel teoremlerinden biri olan bölme algoritması başlangıç düzeyindeki her cebir ve sayılar teorisi kitabında ifade ve ispat edilir[13]

Teorem 1 (Bölme Algoritması) . a, b Є Z ve a ≠ 0 ise

$$b= aq+r,\ 0 < r < |a|$$

olacak şekilde tek türlü belirli q , r Є Z vardır. Teorem 1 de a ve b tarafından tek türlü belirlenen q ve r tamsayılarına, sırasıyla b nin a ile bölünmesinden elde edilen bölüm ve b nin a ile bölünmesinden elde edilen kalan denir.[13]

a ve b herhangi iki tamsayı ve b= ad olacak biçimde bir d Є Z varsa, bu taktirde a,b yi böler veya b,a nın bir katıdır veya a, b nin bir bölenidir denir ve $a|b$ yazılır. Böylece

ortaya çıkan bağıntıya bölünebilirlik bağıntısı denir. Bölünebilirlik bağıntısı tamsayılar kümesi Z içinde bir denklik bağıntısıdır. Bu bağıntının aşağıdaki özellikleri de tanım kullanarak kolayca görülebilir: a, b, c Є Z olsun. Bu takdirde

$$a|b \Rightarrow a|bc\,;\ a|b\,,\ a|c \Rightarrow a|b \pm c$$

Eğer p Є Z , p> 1 ve p nin 1 ve kendisinden başka pozitif böleni yoksa p ye asal sayıların diğer sayılar için temel yapı taşları olduğunu ifade eder.Başlangıç düzeyindeki her cebir ve sayılar teorisi kitabında bulanabilecek bu teoremi de aşağıdaki gibi ifade edilir.[13], [14]

Teorem 2 (Aritmetiğin Temel Teoremi). n Є Z , n> 1 ise $n = p^{\alpha_1}{}_1 \ldots p^{\alpha_r}{}_r$ olacak biçimde tek türlü belirli $p_1 < \ldots p_r$ asal sayıları ve $\alpha_1, \alpha_2, \ldots, \alpha_r$ doğal sayıları vardır.

Teorem 2 deki (*) ifadesine n sayısının asal çarpanlarına ayrılışı denir. Eğer n>1 sayısının asal çarpanlarına ayrılışı (*) da olduğu gibi ise n nin $(a_1 +1)\ldots.(a_r +1)$ tane farklı böleni vardır. Eğer p bir asal sayı ve a bir pozitif tam sayı ise p nin n yi bölen en büyük kuvvetinin p^{α} olduğunu göstermek için $pa \| n$ yazılır.

Başka bir deyimle $p^{\alpha} \| n \Leftrightarrow p^{\alpha} | n$ ve $p^{\alpha+1} | n$ dir. a ,b Є Z , ab $\neq$ 0 olsun a ve b nin en büyük ortak böleni a ve b nin her ikisini de bölen en büyük pozitif tamsayıdır ve OBEB (b,a) ile gösterilir.

OBEB (a,b) = OBEB (b,a) = OBEB (-a,b)

= OBEB (a,b+ka), k Є Z olduğu açıktır.

Eğer a= 0 , b$\neq$ 0 ise OBEB (a,b) = $|b|$ dir.

a ,b Є Z a$\neq$ 0 , b$\neq$ 0 olsun. a ve b nin her ikisinin de katı olan en küçük pozitif tam sayıdır ve OKEK (a, b) ile gösterilir. a ,b Є Z a$\neq$ 0 ve b$\neq$ 0 ise a ve b nin asal çarpanlarına ayrılışında, bazı asal sayıların üslerini 0 almak sureti ile, $a = \pm p^{\alpha_1}{}_1 p^{\alpha_2}{}_2 \ldots p^{\alpha_r}{}_r, b = \pm p^{\beta_1}{}_1 p^{\beta_2}{}_2 \ldots p^{\beta_r}{}_r$ kabul edilebilir.

Örneğin ;

$$36 = 2^2 .3^{\,2} .7^0, \; -63 = -\,2^0 . 3^2 .7^1$$

a ve b nin çarpanlarına ayrılışı yukarıdaki gibi ise ve her i= 1 ….r için $s_i=\min\{\alpha_i,\beta_i\}, t_i=\text{maks}\{\alpha_i,\beta_i\}$ tanımlanırsa,

$$OBEB(a,b) = p_1^{\,s_1} p_2^{\,s_2} \ldots p_r^{\,s_r}$$

$$OBEB(a,b) = p_1^{\,t_1} p_2^{\,t_2} \ldots p_r^{\,t_r}$$

olduğu görülür. Aşağıdaki sonuç bu ifadelerden hemen elde edilir.

Sonuç: a, b Є Z a> 0 , b> 0 ise
OBEB (a b). OKEK (a ,b) = ab.

Çok büyük tam sayılarla çalışıldığı zaman, asal çarpanların bulunması kolay değildir. a veya b nin asal çarpanları bilinmese de; Öklit Algoritması yardımıyla OBEB (a,b) yi buluabilir.

Öklit Algoritması aşağıdaki gibi çalışır:

a>b olmak üzere OBEB (a,b) yi bulmak için: ilk önce a, b ye bölünür ve bölüm q_1, kalan r_1 olarak yazılır: $a = q_1 b + r_1$

İkinci bölmede a yerine b, b yerine de r_1 alınır: $b = q_2 r_1 + r_2$. Daha sonra r_1, r_2 ye bölünür: $r_1 = q_3 r_2 + r_3$.

Bu şekilde devam edilirse, $r_1 > r_2 > \ldots.$ olduğundan sonlu adımda sıfır kalanı elde edilir. Bu işlemde sıfırdan farklı son kalan a ve b'nin en büyük ortak bölenidir. Bu düşünce biçimiyle aşağıdaki önermeyi kolayca ispatlanabilir.[15]

Önerme 1. $a,b \in Z,\ ab \neq 0$ ise

OBEB(a,b)=ua+vb olacak şekilde u ve v tamsayıları mevcuttur. OBEB(a,b)=1 ise a ve b tamsayıları aralarında asaldır denir.

Önerme 2. Eğer $a,b,c \in Z$, a ve b aralarında asal $a \mid bc$ ise ya $a \mid b$ ya da $a \mid c$ dir.
İspat: a ve b aralarında asal olduğundan

$$1 = OBEB(a,b) = ua + vb$$

olacak biçimde $u, v \in Z$ vardır. Eğer $a|b$ ise; yukarıdaki eşitliği c ile çarpalım:

$$c = uac + vbc$$

Sağ taraf a ile bölündüğünden, $a|c$ dir.

2.1.2. Kongruanslar-Çin Kalan Teoremi

Bir m pozitif tamsayısı verilmiş olsun. $a,b \in Z$ için $m|(a-b)$ ise a ve b m modülüne göre kongruenttirler denir ve $a \equiv b(\mathrm{mod}\, m)$ yazılır. Böylece ortaya çıkan bağıntıya m modülüne göre kongruans bağıntısı denir.

Bu bağıntı bir denklik bağıntısı olup aşağıdaki ek özelliklere sahiptir:

1. $a \equiv b \bmod m$ ve $c \equiv d \bmod m \Rightarrow a \pm c \equiv b \pm d \bmod m$
   $ac \equiv bd \bmod m$ dir.
2. $a \equiv b \bmod m$ ise $d|m$ için $a \equiv b \bmod d$ dir.
3. $a \equiv b \bmod m$, $a \equiv b \bmod n$ ve $(m,n) = 1$ ise $a \equiv b \bmod mn$ dir.

m modülüne göre kongruans bağıntısı ile ortaya çıkan denklik sınıflarına, m modülüne göre kalan sınıfları denir. m modülüne göre kalan sınıflarından oluşan küme Z/mZ ile gösterilir. Z/mZ içinde tam m tane kalan sınıfı vardır:

$$0 + mZ = \{mk : k \in Z\}$$
$$1 + mZ = \{1 + mk : k \in Z\}$$
$$\cdots$$
$$(m-1) + mZ = \{(m-1) + mk : k \in Z\}$$

Herhangi bir $a \in Z$ için bölme algoritması uygulandığında;

$$a = mq + r,\ 0 \le r < m$$

ise, Z/mZ içinde a nın temsil ettiği kalan sınıfı $r + mZ$ dir.

$Z/_{mZ}$ içinde;

$$(r + mZ) + (s + mZ) = (r + s) + mZ$$

$$(r + mZ)(s + mZ) = rs + mZ$$

ile tanımlanan toplama ve çarpma işlemleri ile birlikte $Z/_{mZ}$; birimli, değişmeli bir halka olur. $Z/_{mZ}$ toplamsal grup olarak bir devirli gruptur. $Z/_{mZ}$ nin toplamsal birim elemanı $0 + mZ$, çarpımsal birim elemanı: $1 + mZ$ dir.[16]

Önerme 2.1. $Z/_{mZ}$ nin bir $x + mZ$ elemanının çarpımsal tersinir olması için gerek ve yeter koşul m ile x in aralarında asal olmasıdır.

İspat: $x + mZ$ tersinir ise

$$1 + mZ = (x + mZ)(y + mZ) = xy + mZ$$

ya da

$$1 = xy + mk$$

olacak biçimde $y, k \in Z$ vardır. Buradan, OBEB(x,m)=1 olduğu görülür.

Diğer yandan, x ve m aralarında asal ise

$$OBEB(x, m) = 1 = ux + vm$$

olacak biçimde $u, v \in Z$ vardır ve bu u için

$$ux \equiv 1 (\mathrm{mod}\, m),$$

$$1 + mZ = ux + mZ = (u + mZ)(x + mZ)$$

ve sonuç olarak $x + mZ$, $Z/_{mZ}$ içinde çarpımsal tersidir.

Sonuç: p bir asal sayı ise $Z/_{pZ}$ nin sıfırdan farklı her elemanı (çarpımsal) tersinirdir ve bu nedenle $Z/_{pZ}$ bir cisimdir. [15], [16]

Bu sonuç ve Grup kuramındaki Langrange teoremi kullanılarak, Fermat'ın küçük teoremi olarak bilinen teoremin ispatı elde edilebilir. Aşağıda bu teoremin değişik bir ispatını sunulmaktadır.[17]

Önerme 2.2 (Fermat'ın Küçük Teoremi). p bir asal sayı olsun. $\forall a \in Z$ için $a^p \equiv a \bmod p$ ve $\forall a \in Z$, $p|a$ için $a^{p-1} \equiv 1 \bmod p$ dir.

İspat: $p|a$ olduğunu kabul edelim. $0.a, 1.a, 2.a, ..., (p-1)a$ tam sayılarının p modülüne göre kalanların bir tam kümesini oluşturduğunu gösterelim. Bunu görmek için, bu sayılardan ikisinin, ia ve ja nın, aynı kalan sınıfında, yani, $ia \equiv ja \bmod p$ olduğunu farz edelim. Bu durumda $p|(i-j)a$ ve a,p ile bölünmediği için $p|(i-j)$ dir. i ve j lerin her ikisi de p den küçk olduğundan, $p|(i-j)$; ancak i=j olması durumunda mümkündür. Modulo p de $a, 2a, ..., (p-1)a$ tamsayılarının 1,2,….,p-1 e denk olduğu sonucuna ulaşılır. Böylece $a^{p-1}(p-1)! \equiv (p-1)! \bmod p$ ve $p|((p-1)!(a^{p-1}-1))$ dir.

(p-1)!, p ile bölünmediği için $p|(a^{p-1}-1)$, $a^{p-1} \equiv 1 \bmod p$ dir.

$a^{p-1} \equiv 1 \bmod p$ kongruansının her iki tarafını a ile çarparsak $a^p \equiv a \bmod p$ yi elde edilir. Sonuç:

$$p|a \text{ ve } n \equiv m \bmod (p-1) \text{ ise } a^n \equiv a^m \bmod p \text{ dir.}$$

İspat: n>m olsun. $p-1|n-m$, $n \equiv m + c(p-1), Z^+$.

$a^{p-1} \equiv 1 \bmod p$ kongruansını kendisiyle c defa ve daha sonra $a^m \equiv a^m \bmod p$ ile çarpılırsa istenen sonuç:

$$a^n \equiv a^m \bmod p$$

elde edilir. Sayılar kuramının ilginç problemlerinden biri a ve b tamsayıları verildiğinde

$$ax \equiv b \bmod m \quad (\#)$$

lineer kongruansının çözümünü bulmaktır. Bu problem Z/mZ içinde

$$(a+mZ)(x+mZ) = b+mZ \qquad (\#')$$

olarak da ifade edilebilir. Kolayca görüleceği üzere, (#) veya (#') probleminin çözümü tartışılırken, genelliği bozmaksızın $0 \le a, b < m$ kabul edilebilir.

Eğer OBEB(a,m)=d ise, (#) denkleminin bir çözümü bulunması için gerek ve yeter koşul $d|b$ olmasıdır. Dolayısıyla, $a' = a|d$, $b' = b|d$ ve $m' = m|d$ tanımlanırsa,

çözümü bulunması durumunda, (#) denklemi aşağıdaki denklemle aynı çözüm kümesine sahiptir:

$$a^{'}x \equiv b^{'}(\mathrm{mod}\, m^{'})$$

Daha genel olarak (#) türünde lineer kongruansların bir sistemi düşünülebilir. Bu tür sistemlerin çözümü, ünlü bir teorem ile belirlenir [15].

Teorem 2.3. (Çin Kalan Teoremi). $m_1, m_2, ..., m_r$ pozitif tamsayılar ver her $i \neq j$ için $(m_i, m_j) = 1$ olsun. Rasgele $a_1, a_2, ..., a_r$ tamsayıları verildiğinde

$$x \equiv a_1 \bmod m_1$$
$$x \equiv a_2 \bmod m_2$$
$$\cdots\cdots$$
$$x \equiv a_r \bmod m_r$$

kongruans sisteminin Z de bir x çözümü vardır ve $M = m_1 m_2 ... m_r$ modülüne göre $x + mZ$ kalan sınıfı, sistemin çözüm kümesi olur.

İspat: $m_k^{'} = m_1 m_2 .... m_{k-1} m_{k+1} .... m_r$ olsun. Her k=1,2,….,r için

$$(m^{'}_k, m_k) = 1$$

olduğundan

$$1 = m_k u_k + m_k^{'} u_k^{'}$$

olacak şekilde $u_k, u^{'}_k \in Z$ vardır.

$$n_1 = m_1^{'} u_1^{'} a_1$$
$$n_2 = m_2^{'} u_2^{'} a_2$$
$$\vdots$$
$$n_k = m_k^{'} u_k^{'} a_k$$
$$\vdots$$
$$n_r = m_r^{'} u_r^{'} a_r$$

sayılarını alalım. $n_1, n_2, ..., n_{k-1}, n_{k+1}, ..., n_r \in m_k Z$ dir.

$$x = n_1 + n_2 + ... + n_k + ...n_r$$

alırsak,

$$n_k - a_k = m_k' u_k' a_k - a_k = (m_k' u_k' - 1)a_k = m_k(-u_k)a_k$$

ve dolayısıyla

$$x - a_k = (n_1 + n_2 + ... + n_k + ...n_r) - a_k$$

$$= (n_k - a_k) + n_1 + n_2 + ... + n_{k-1} + n_{k+1} + ... + n_r$$

$$= m_k(-u_k)a_k + n_1 + n_2 + ... + n_{k-1} + n_{k+1} + ... + n_r$$

olur. Demek ki k=1,2,…,r için

$$x \equiv a_k \bmod m_r$$

olur. O halde x aranan çözümdür. Özelliğin ikinci kısmına gelince, x_1 başka bir çözüm ise, her i=1,2,…,r için $x \equiv a_i \bmod m_i$ ve dolayısıyla $x \equiv x_1 \bmod m_i$ olur. Her $i \neq j$ için $OBEB(m_i, m_j) = 1$ olduğundan $x \equiv x_1 (\bmod m_1 m_2 ... m_r)$ olduğu görülür. Sonuç olarak, $M = m_1 ... m_r$ modülüne göre $\bar{x} = x + MZ$ kalan sınıfı verilen sistemin çözüm kümesidir.

2.2. Asal Sayılar

Asal sayılar hakkında şu ana kadar belirli bir seri keşfedilmemiştir. Bir sonraki asal sayının ne olduğu belli değildir ve bunu bulmak birçok insanı kışkırtmaktadır. Binyıllardır birçok matematikçi çeşitli fonksiyonlar bularak asal sayıları kendi nezdinde modellemişlerdir. Fakat genel bir seri henüz mevcut değildir. Günümüzde daha çok asal sayı üreteçleri adı verilen paralel işleme aracılığıyla büyük asal sayılar bulunmaya çalışılmaktadır. Tabi bu sayıları bulmak için sıralı bir arama algoritması yoktur belirli kıstaslara göre (fonksiyonlar, Mersenne, N-1) bazı sayılar seçiliyor sonar asallık testlerine tabi tutuluyorlar. Şu anda önemli birkaç fonksiyondan ve asallık testlerinden bahsedilecektir.

2.2.1. Euclid kuramı

1'den n'e kadar olan tüm asal sayıların çarpımı olan bir P sayısı düşünelim (P=1x2x3x5x7x.....xn) Şimdi N=P+1 ele alalım

- N eğer asalsa n'den büyük bir asal sayı olduğunu gösterir. Bir başkası, Bir başkası diye gider.(Tümevarım)

- N eğer asal değilse N'in 1 ile n arasında hiçbirine bölünemeyeceğini ispatlamıştır.(Tümdengelim)[18],[19]

2.2.2. Mersenne Sayıları

Eski çağlardan beri matematikçiler 2^{n-1}(n asal olmak şartıyla) asal sayıları modellemeye çalıştılar.[20]

- 1644 yılında Mersenne n'in 257'e kadar sadece 11 sayıda diğerleri için olmadığını gösterilmiştir.
    - 2^3 021 377 -1 (909526) 1998 (GIMPS)
    - 2^2 976 221 -1 (895932) 1997 Spence

2.2.3. Fermat teoremi

Varsayım:Üçüncü dereden bir ifadeyi üçüncü dereceden bir başka terime ayırmak olanaksızdır. Eğer n asal değilse bir p.q seçilir (p asal bir sayı yada asal sayıların çarpımı,q ise asal). Böylece:

a(s)n+b(s)n=c(s)n;

a(s) pxq+ b(s)pxq= c(s)pxq

(a(s)p)xq + (b(s)p)xq = (c(s)p)xq yada

a (s) q + b (s) q = c (s) q

Asal olmayan bir n'i çözmek için asal olan bir q seçilir. Sophie Germain yorumu N ve 2n+1 asal olmaları durumunda x(n)+y (n)=z (n) ; (x,y,z tamsayıları n'e bölünebilir olduğunu gösteriyor)

Fermatın son kuramına göre x,y,z 'den hiçbiri n'e bölünemez. x,y,z'den yalnızca biri n'e bölünebilir

Germain 1 numaralı kuralı 100'den küçük bütün sayılar için, Legendre'de bu sayıyı 197'e kadar kanıtlamıştır. 1955 yılında Yutaka Taniyama Eliptik eğrilerle ilgilenirken bunun Fermat'ın kuramıyla ilişkilendirmiştir.[20]

2.3. Asal Sayı Üretimi

Açık anahtarlı sistemlerde büyük asal sayıların hızla üretilmesi önemli bir rol oynar. Hemen hemen bütün sayısal imza algoritmalarında en az bir asal sayı kullanılır. Burada sistemin güvenliği asal sayının büyüklüğüne ve "rasgele" olmasına dayanır. Buradaki rasgele'nin manası başka biri tarafından seçilme ihtimalinin düşük olmasıdır.

Ancak sayıların çarpanlara ayrılması oldukça uzun zaman aldığı için çoğunlukla kullanılan bir yöntem değildir. Bir sayının asal olup olmadığını anlamak için kullanılan

yöntem şu şekildedir. Uygun büyüklükte bir N sayısı üretilir ve asal olup olmadığı test edilir.

Bu test sayının asal olduğunu kanıtlayan(Bu durumda kanıtlanmış asal) ya da büyük olasılıkla asal denir. Sayı muhtemel asal çıkıyorsa bu tür testlere olasılıklı asallık testleri; eğer matematiksel olarak ispatlıyorsa buna da Gerçek asallık testleri denir.

2.3.1. Fermat'ın Asallık Testi

Fermat'ın teoremine göre n asalsa $n-1 \geq a \geq 1$ olacak şekilde herhangi bir a sayısı için $a^{n-1} = 1 \pmod n$ 'dir. Bu yüzden n sayısının asal olmadığını ispatlamak için bu aralıkta bu eşitliği sağlamayan bir a sayısı bulunması yeterli olacaktır. n asal olmayan bir tek sayı olsun, $n-1 \geq a \geq 1$ ve $a^{n-1} \neq 1$ şeklindeki bir a sayısına n için bir Fermat tanığı denir.[20]

Algoritma

Giriş: n>=3 ve t>=1

For i =1 to t do

$n-2 \geq a \geq 2$ olacak şekilde bir rasgele a sayısı seç .

$r = a^{n-1}$ 'i hesapla

If (r=\1) return ("Asal değil")

Return("Asal")

2.3.2. Miller-Rabin Asallık testi

Pratikte en çok kullanılan asallık testi Michael Rabin tarafından Gary Miller'ın fikirlerine dayanılarak geliştirilmiştir. Bu teste gören, asal tek sayı ve $n-1 = 2^s r$ olsun (r tek).[21]

a,gcd(a,n)=1

bu durumda $s-1 \geq j \geq 0$ sağlayan bir j değeri için aşağıdakilerden biri doğrudur.

$$U=2^{s} r$$

$$a^{r} = 1 \ (\mathrm{mod}\ n)$$

$$a^{U} = -1 \ (\mathrm{mod}\ n)$$

Algoritma

Giriş n>=3 ve t>=1

R tek olacak şekilde $n-1 = 2^{s} r$ 'i yaz

$n-2 \geq a \geq 2$ olacak şekilde bir rasgele a sayısı seç .

For i = 1 to t do

y=a^{r} mod n

If y=/1 and y =/ n-1 Then

J ←1

While j =< s-1 AND y =/ n-1 DO

y ← y^{2} mod n

IF (y=1) THEN RETURN ("Asal Değil")

j ← j + 1

IF (y=/ n-1) THEN RETURN ("Asal Değil")

RETURN ("Asal")

3. RIVEST-SHAMIR-ADLEMAN -RSA- KRİPTOSİSTEMİ

RSA (RIVEST-SHAMIR-ADLEMAN) asimetrik şifreleme algoritması 1977 yılında R.Rivest, A.Shamir ve L.Adleman tarafından bulunmuş ve daha sonra asimetrik şifreleme algoritmalarına (genel anahtar şifrelemesi) uygun biçimde geliştirilmiştir. Bu algoritma, açık anahtarlı şifreleme sistemlerini ve sayısal imza işlemlerini işlemlerinde güvenli bir şekilde kullanılır.[1],[2]

S/MIME,PEM,MOSS ve PGP gibi gizli haberleşme protokolleri temel olarak RSA kullanır, ve bazı SSL,PCT gibi protokollerde kullanılır.

RSA Algoritması public-key şifreleme yönteminin temel bir uygulamasıdır. Bu şifreleme yönteminin önemli bir özelliği ise önceden aralarında hiçbir görüşme yapmamış olan alıcı ve vericinin kendi aralarındaki iletişimin güvenli bir ortamda(güvenli gönderim ve mesajların doğrulanması) yapılmasıdır.

Görünüşte son derece basit matematiksel ilişkilerle çalışan bu yöntem de iki ayrı anahtar bulunmaktadır. Anahtarlardan birisi kamuya açık, birisi de gizlidir. Herkes açık anahtarını yayınlar ve kendisine şifreli bir mesaj göndermek isteyen birisi bu anahtarı kullanarak mesajı şifreler ve gönderir. Ancak mesajı sadece gizli anahtar kimde ise o çözebilir. Gizli anahtar da sadece sahibinde bulunur. Böylece, herkes çözüm için gerekli anahtarı bilmeden, güçlü bir şifreyle mesajları gizleyebilir.

Daha önce hiç karşılaşmamış, birbirini tanımayan kişiler bile birbirlerine gizli mesajlar gönderebilir. Örneğin Internet'ten alışveriş yapan birisi, kendisini hiçbir şekilde tanımayan bir web sitesine giderek, sitenin kamuya açık anahtarını alır, kart numarasını bu anahtarla şifreleyerek gönderir. Şifreli bilgiyi gönderen dahil hiç kimse çözemez, sadece web sitesinde bulunan gizli anahtarla gelen kart numarasını web sitesi çözebilir. Böylece kart hamili kart numarasının başkası tarafından okunmayacağından emin olacaktır. Ama, acaba Web sitesi gerçekten dürüst bir satıcı mı, yoksa sahte bir site mi? Bundan emin olamayacaktır, ancak bunun da çözümü SERTİFİKA yöntemiyle sağlanmaktadır.[22]

İnternetin şu anki durumu güvenliği tam olmayan kanallar üzerinden işler. Bu durumda asimetrik şifreleme tekniğiyle internette güvenlik sağlanmıştır.

RSA şifreleme algoritmasının çalışması sekil 3.1'da gösterilmiştir. Şekilde gösterildiği gibi 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.[25]

3.1. RSA Algoritmasının Yapısı

RSA şifreleme algoritmasında şifrelenecek olan açık metni öncelikle [0, n-1] arasındaki pozitif tamsayı bloklar haline dönüştürülür. Şekil 3.1 de ayrıntılı olarak matematiksel işlemler gösterilmektedir.[23]

Bundan sonraki işlem gizli anahtar ve açık anahtar çiftlerini elde etmektir. Bunun için p ve q şeklinde çok büyük iki tane birbirinden farklı iki asal sayı bulunur.

$$n = p.q \text{ ve } Z=(p-1).(q-1)$$

hesaplanır. *Z* ile ortak böleni 1 olacak şekilde bir *E* sayısı bulunur. Açık anahtar (Public key) {E,n} olarak belirlenir.

$$D=E^{-1} \bmod Z$$

olacak şekilde bir D sayısı bulunur. Gizli anahtar (Private key){D,n} olarak belirlenir.

Şifrelenecek mesajı *m* kabul edersek bu mesaj ikilik olarak $2^k<N$ olacak şekilde k bitlik kısımlara ayrılır.

$$m=m(1)+m(2)+m(3)+...+m(n)$$

Daha sonra şifreleme için her bir kısma $C(i)=m(i)^E \bmod N$ işlemi uygulanır.

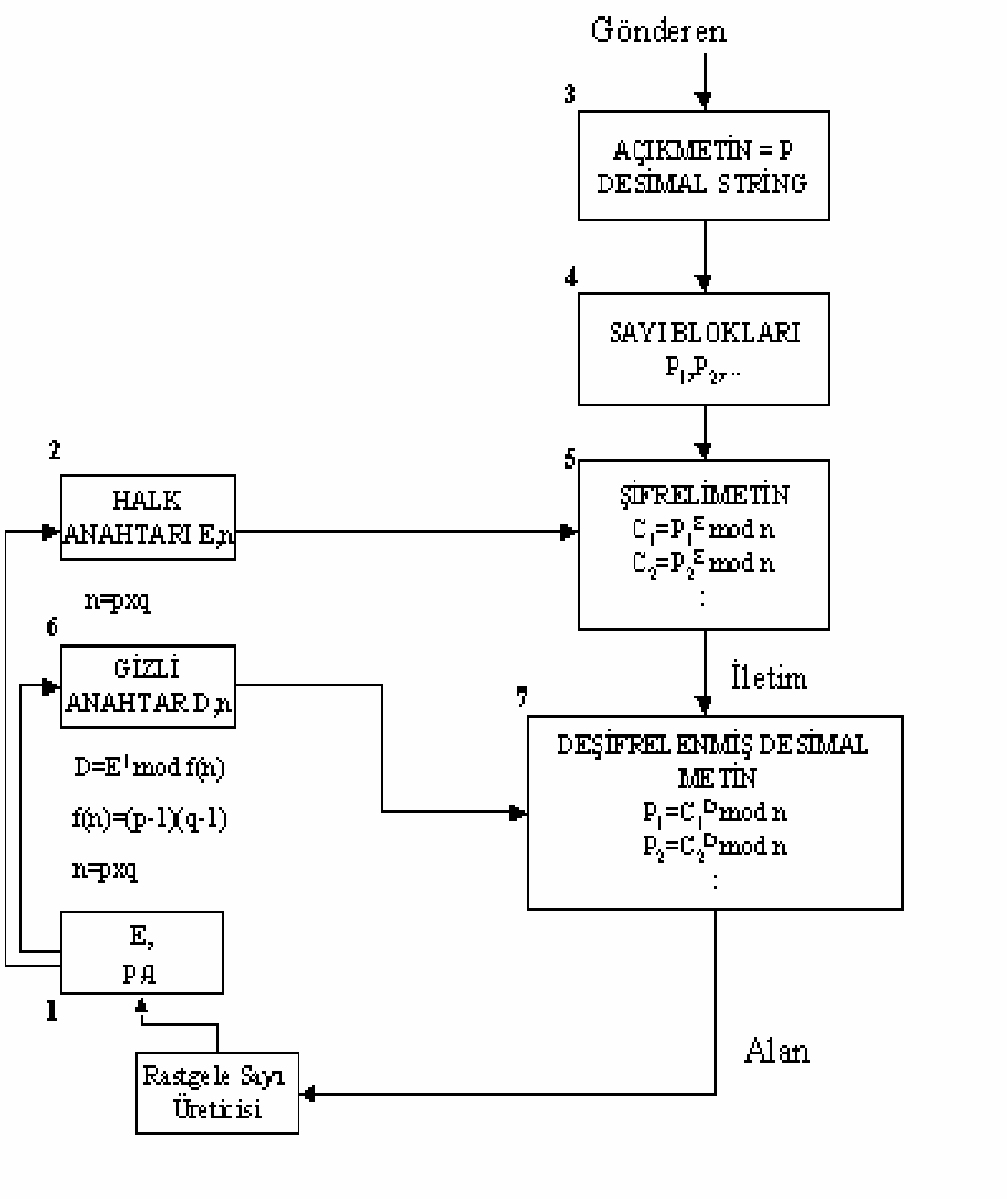
Böylece şifreleme işlemi bitmiş olur. Girişte kullanılan açık metin **m** şifrelenmiş olarak C şeklinde elde edilir.

Deşifreleme

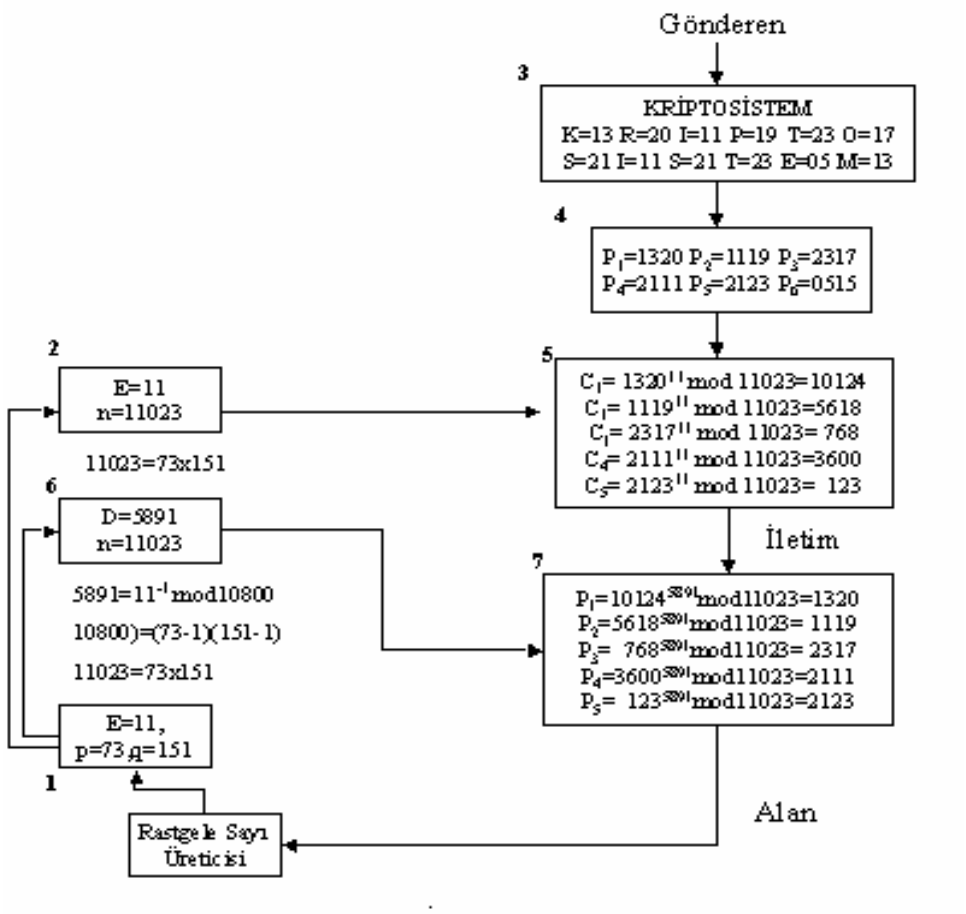
Belirlenen D gizli anahtarı ile elimizde bulunan şifrelenmiş C metnini çözülmesi gerekmektedir. Bunun içinde şifrelemek için kullanılan bir matematiksel işlem kullanılır. [25]

Gizli anahtar {D,n} kullanılarak şifre çözümü:

$$m(i)=C(i)^D \bmod N$$



Şekil 3.1 RSA Algoritması[23]



Şekil 3.2 RSA Algoritması Sayısal Örneği

3.2. RSA Sisteminin Güvenirliği

RSA sisteminin "kırılması" birkaç değişik şekilde yorumlanabilir. Sisteme en çok zarar verecek saldırı bir kriptanalistin belli bir açık anahtara karşı gelen gizli anahtarı bulmasıdır. Bunu başarabilen bir "hasım" hem şifrelenen bütün masajları 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.[21]

Ancak n yeterince büyük seçilirse günümüzün teknolojisiyle n'nin çarpanlarına ayrılması "yeterince" uzun süreceği için bu yöntemle n'nin hesaplanması hesaplama açısından verimsiz olacaktır.

RSA'yı kırmanın bir başka yolu mod n'e göre e'inci köklerin hesaplanmasıdır. $c=m^e$ olduğuna göre c'nin e'inci kökü m'dir. Böyle bir saldırı, gizli anahtar bilinmese dahi şifrelenmiş mesajların deşifre edilmesini ya da imzaların taklit edilmesini sağlayabilir. Böyle bir saldırının n'nin çarpanlarına ayrılmasına eşdeğer olup olmadığı bilinmemektedir. Şu ana kadar RSA yöntemini bu yolla kırmaya çalışan bir metoda rastlanmamıştır.

Tabii bir kriptanalistin doğru olanı bulana kadar mümkün olan tüm d'leri denemesi mümkündür. Ancak böyle bir brute-force saldırı n'in çarpanlarına ayrılmasından daha verimsizdir, bir başka yöntem ise (p -1)(q -1)'in değerinin tahmin edilmesi olabilir. Bu da aynı şekilde n'in çarpanlarına ayrılmasından daha kolay değildir.

Yayınlanan makalede bahsedilen bir başka saldırı da n'i çarpanlarına ayırmadan ϕ>(n)'nin hesaplanmasıdır. Eğer bir kriptanalist ϕ >(n)'i hesaplayabilirse e'nin ϕ >(n) modülüne göre çarpımsal tersini hesaplayarak d'yi bulabilir. Ancak bu n'nin çarpanlarına ayrılmasından daha kolay değildir çünkü bu yolla kriptanalist ϕ >(n)'i kullanarak n'i kolaylıkla çarpanlarına ayırabilir. Bunun için ϕ >(n) = n -(p + q) + 1 eşitliğinden n bilindiği için. (p + q) hesaplanır.

$$(p-q) = \sqrt{[(p+q)^2 - 4n]}$$

olduğundan (p -q) bulunur. Bu iki eşitlikten p ve q hesaplanabilir. Yine orijinal makalede d'nin hesaplanmasının n'in çarpanlarına ayrılmasından daha kolay

olamayacağı iddia edilmektedir. Çünkü eğer d bilinirse n kolaylıkla çarpanlara ayrılabilir.[26],[30],[33]

3.3. RSA Şifrele Algoritmasının Kriptanalizi

RSA şifreleme algoritmasına karşı yapılan kriptanaliz çalışmaları iki asal (p,q)çarpandan oluşan çok büyük sayıyı (n) asal çarpanlarına ayırma metotlarıyla gerçekleştirilmektedir. RSA şifreleme algoritmasından da anlaşılacağı gibi, algoritma içindeki n sayısını p ve q asal çarpanlara ayırdığı durumda deşifreleme anahtarı olan d sayısına rahatlıkla ulaşılabilir. Şunu da biliiyorki ki RSA'nın güvenilirliğini arttırmak için çok büyük n sayısı kullanılmaktadır**.** [26]

Bu bölümde Michel J. Wiener tarafından önerilen sürekli kesirlerin kullanılarak n sayısını asal çarpanlara ayırma metodunu kullanılacaktır. [27], [28]

3.3.1. Sürekli Kesirler

Genel olarak real bir sayının sürekli kesirler şeklinde ifadesi aşağıda ki gibidir. Burada a ve b tam sayılardır.[31]

$$x = a_0 + \cfrac{b_1}{a_1 + \cfrac{b_2}{a_2 + \cfrac{b_3}{a_3 + \dots}}},$$

Bu ifade aşağıdaki form şeklinde de gösterilebilir.

$$x = a_0 + \frac{b_1}{a_1 +} \frac{b_2}{a_2 +} \cdots$$

Örnek olarak:

$$3796/1387 = 2 + 1022/1387$$
$$= 2+1/1+ 1/(1022/365)$$
$$= 2+1/1+1/2+ 1/(365/292)$$
$$= 2+1/1+1/2+1/1+1/4$$
$$3796/1387 = [2;1;2;1;4]$$

Bu örnekte olduğu gibi kesirli bir ifadeyi tam sayılarla gösterebilmektedir. [2;1;2;1;4] gösterimi, 3796/1387 değerinin genişletilmiş sürekli kesir ifadesi olarak kullanılmaktadır. f değeri, literatürde kesirli sayının genişletilmiş sürekli kesir olarak geçmekte ve her seferinde kendin den çıkartılarak sıfıra ulaşana kadar bu işlem tekrarlanarak sonuca ulaşılmaktadır.[32]

3.3.2. Sürekli Kesir Algoritması

f ve f' birer kesir olsun, eğer;

$$f' = f(1-\delta), \qquad 0 \geq \delta$$

Eğer yeterince ufak δ değeri bulabiliyorsa,

$$\delta < \frac{1}{\frac{3}{2} n_m d_m}$$

n_m ve d_m f fonksiyonun pay ve paydası olduğundan dolayı, f' değeri hesaplanabilir.[28] RSA algoritmasının kriptanalizini yaparken açık anahtar olan (e,n) değerlerini kullanarak, f' değerini oluşturarak f değerinin pay ve paydasına ulaşmaya çalışılır.[29],[33]

3.3.3. RSA Kriptanalizi

RSA şifreleme algoritmasının yapısındaki açık anahtar(e) ve gizli anahtar (d) arasındaki matematiksel bağıntıyı kullanarak,

$$d \,.\, e \bmod \Phi(n) = 1 \,,\ \Phi(n) = (p-1).(q-1)$$

Bu formülü şu şekilde yazılabilir,

$$e \,.\, d = 1 \pmod{\text{LCM}(p-1,q-1)}$$

Bu formülde e.d nin bir K sabitine bölündüğünde 1 kalanını verdiğini görülür.

$$e \,.\, d = K \,.\, \text{LCM}((p-1)\,,(q-1)+1$$

K sabitini kullanarak formülü bu şekilde değiştirebilir. G gibi bir sabiti G = GCD (p-1,q-1) şeklinde ifade edilerek ve

$$\text{LCM}((p-1)\,.\,(q-1)) = ((p-1)\,.\,(q-1)) \,/\, \text{GCD}(p-1,q-1)$$

Bu sabitleri göz önüne alınıp 2 nolu denklemimizi şu şekilde yazabilir.

$$e.d = \frac{K}{G}(p-1).(q-1)+1$$

k = K/GCD(K,G) ve g = G/ GCD(K,G) şeklinde kısaltma yaparak denklemimizin son hali aşağıdaki şekilde olur.

$$e.d = \frac{k}{g}(p-1).(q-1)+1$$

Denklemin her iki tarafın dpq ya bölerek ve sürekli fonksiyon algoritmasını kullanarak

$$\frac{e}{pq} = \frac{k}{dg}(1-\delta) \text{ ve } \delta = \frac{p+q-1-\frac{g}{k}}{pq}$$

Denklemine ulaşılır. RSA şifreleme algoritmasının yapısından e ve p.q=n değeri biliniyor. Bu bilinen değerleri ve sürekli fonksiyon algoritmasını kullanarak,

$$kdg < \frac{pq}{\frac{3}{2}(p+q)}$$

denklemine ulaşılır.

Şimdi *k* ve *dg* değerlerini doğru tahmin etmek için bir test uygulanacaktır. *edg* değerini k ile bölünerek, (*p*-1)(*q*-1) ve g'nin kalanı arasında bir bölüm değeri bulunur. (*p*-1)(*q*-1) ve g değeri doğruluğunu belirmek en önemli nokta olmaktadır. (*p*-1)(*q*-1) değeri eğer sıfır ise tahmin değeri yanlış, eğer sıfırdan farklı bir değer olursa bulunması gereken sonuçlardan biri olabilir.

$$\frac{pq-(p-1)(q-1)+1}{2} = \frac{p+q}{2}$$

(*p*-1)(*q*-1) değeri ve yukarıda ki denklem kullanılarak $\frac{p+q}{2}$ değeri bulunabilir. Eğer (*p*-1)(*q*-1) doğru ise $\frac{p+q}{2}$ değeri bir tam sayı çıkmaktadır. Tam sayı çıkmadığı takdir de k ve dg değerleri yanlıştır.

Eğer $\frac{p+q}{2}$ değeri tam sayı çıktıysa aşağıdaki denklem kullanılarak, $\left(\frac{p-q}{2}\right)^2$ değeri bulunabilir.

$$\left(\frac{p+q}{2}\right)^2 - pq = \left(\frac{p-q}{2}\right)^2$$

$\left(\frac{p-q}{2}\right)^2$ değeri bir tam sayının karesi olarak çıkarsa, k ve dg değerleri doğrudur. $\frac{p+q}{2}$ ve $\left(\frac{p-q}{2}\right)^2$ değerleri bulunduktan sonra bu iki denklem kullanılarak p ve q değerleri bulunur.

$d = \frac{dg}{g}$ denklemi ile d (gizli anahtar) değeri bulur.

4. ELİPTİK EĞRİ ŞİFRELEME ALGORİTMASI (ECC)

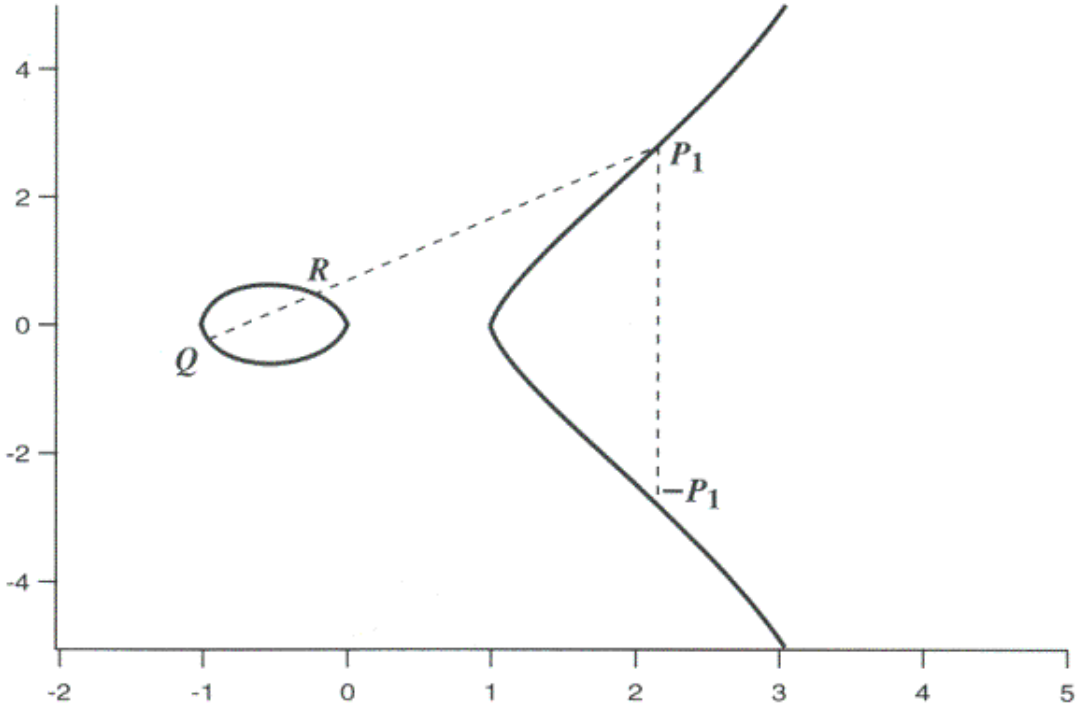
Günümüzde açık anahtarlı kriptografik uygulamalar başlıca üç ana matematiksel probleme dayandırılarak geliştirilmektedir.[34] Bu alanlar,

- Bir tam sayının çarpanlarına ayrılması problemi
- Ayrık logaritma problemi
- Eliptik eğri ayrık logaritma problemi

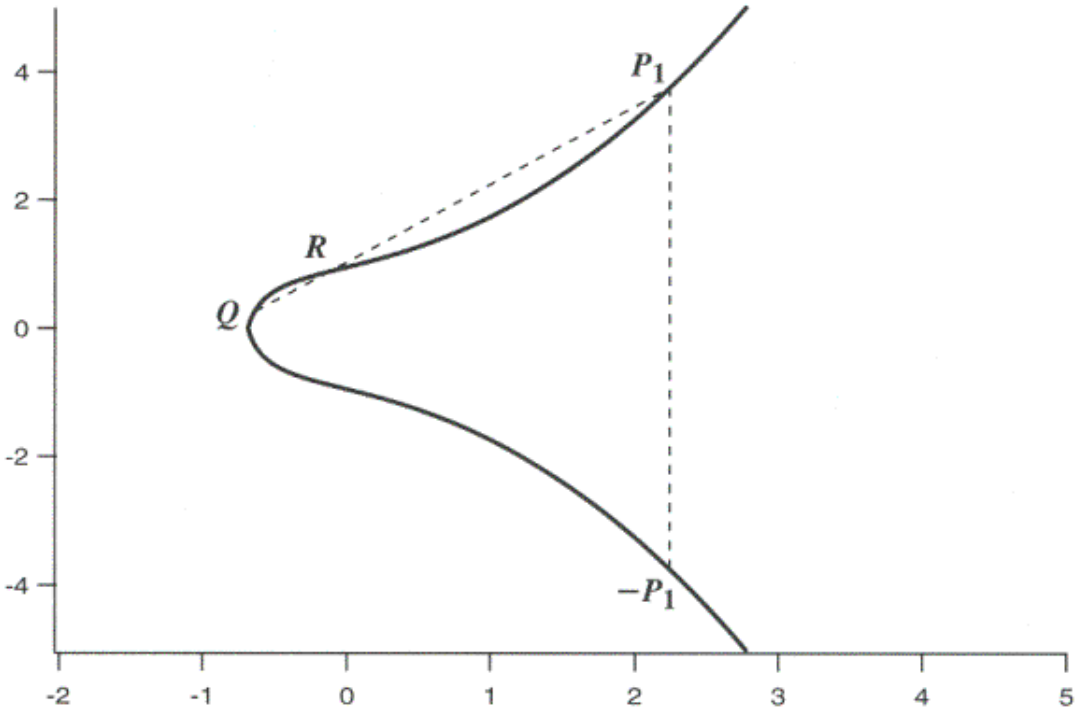
Son dönemlerde organizasyonlar, güvenlik gereksinimlerini karşılamak üzere, daha yüksek boyutlu anahtarlara ihtiyaç duymuşlar, bu gereksinim ise gerek hafıza ihtiyacı, gerekse işlem yükü açılarından maliyet ile doğru orantılı olarak organizasyon sistemlerine büyük yük getirmiştir. Eliptik eğri grupları temeline dayanan şifreleme, anahtar boyutları ve transmisyon hızlarında büyük gelişmelere olanak sağlamaktadır.

Açık anahtarlı kriptografi kullanan standartların ve ürünlerin hemen hemen hepsi, şifreleme ve dijital imza için RSA kullanmaktadır. Daha önceki bölümlerde de gördüğümüz gibi RSA`nın güvenli kullanımı için, çalışılan bit uzunlukları zaman içerisinde büyümüş ve dolayısıyla RSA kullanan uygulamalar üzerine büyük bir hesapsal ağırlık getirmiştir. Özellikle büyük sayıların transferini güvenlik içinde gerçekleştirmesi gereken ticari siteler bundan çok fazla etkilenmişlerdir. Son zamanlarda, RSA`ya karşı rakip olarak geliştirilmiş bir sistem ortaya atıldı. Eliptik Eğri Kriptografisi (ECC). ECC, şimdiden açık anahtarlı kriptografi için öngörülmüş IEEE P1363 standartlarını yerine getirmektedir. [35]

ECC`nin RSA`ya karşı en büyük avantajı, daha küçük bitler kullanılarak yapılan işlemlerin RSA gibi yüksek güvenlik sağlayabilmesi, bunun sayesinde kullanıcıların şifreleme ve deşifreleme esnasında hesaplamalar için harcadıkları efor azalıyor. Diğer bir taraftan da, her ne kadar ECC`nin teorisi kısa bir süre önce ortaya atıldıysa da, ECC kullanan ürünler kısa süre içerisinde kendisini göstermeye başladı ve bu ürünler üzerinde yapılan güvenlik testleri ECC`nin henüz RSA`nın sağladığı kadar yüksek bir güvenlik sağlayamadığını göstermektedir.[37]



(a) $y^2 = x^3 - x$



(b) $y^2 = x^3 + x + 1$

Şekil 4.1 Eliptik Eğri Örneği [37]

4.1. Eliptik Eğriler

Eliptik eğriler elips değildirler. Bu şekilde adlandırılmalarının sebebi, bir elipsin çemberinin hesaplanması için kullanılan kübik denkliklere benzer ifadeler ile gösterilmeleridir.[36] Genel olarak, eliptik eğriler için kübik denklemler aşağıdaki formdadır:

$$y^2 + axy + by = x^3 + cx^2 + dx + e$$

Bu denklemdeki a,b,c,d ve e sayıları reel sayılardır ve bazı basit koşulları sağlarlar. Ayrıca, eliptik eğrinin tanımlamasında, daha sonra daha ayrıntılı şekilde inceleyeceğimiz, sonsuzluk yada sıfır nokta adı verilen bir O notasyonu vardır. En büyük dereceli üs 3 olduğundan dolayı bu tip denklemler kübik olarak adlandırılırlar. Şekil 4.1 iki eliptik eğri örneği göstermektedir. Gördüğünüz gibi formüller bazen garip görünüşlü eliptik eğriler doğurmaktadırlar.[4],[37],[47]

Eğer bir eliptik eğrinin 3 noktası düz bir çizgi üzerinde bulunuyorsa, bunlar O olarak özetlenir. Bu açıklamadan yola çıkarak, bir eliptik eğri için şu kuralları tanımlanabilir:[38]

1. Eliptik eğri üzerindeki herhangi bir P noktası için, $P + O = P$ olur.
2. Bir dikey çizgi, aynı x değeri için eliptik eğriyi $P_1 = (x, y)$ ve $P_2 = (x, -y)$ gibi iki noktasında kesiyorsa, bu çizgi aynı zamanda eliptik eğriyi sonsuzluk noktasında da kesiyordur. Bu yüzden, $P_1 + P_2 + O = O$ ve $P_1 = -P_2$ olur. Böylece bir noktanın negatifi, x ekseni üzerinde aynı değeri alacak şekilde bir noktadır ve bu noktanın y ekseni üzerindeki değeri ilk noktanın negatiflisidir. Bunu Şekil 4.1`de de görülebilmektedir.
3. x koodrinatı farklı olan Q ve R noktası seçip bu iki noktadan geçen düz bir çizgi çizildiğinde kesişimin üçüncü noktası olan P_1`i buluruz. Ve çok kolay bir

şekilde görülebilir ki, P_1 noktası sadece bir tanedir (eğer çizilen doğru, Q veya R noktalarından birisinden teğet geçiyorsa bu durumda $P_1 = Q$ veya $P_2 = R$ alırız). Bu durumda $Q + R + P_1 = O$ ve dolayısıyla $Q + R = -P_1$ olacaktır. 4.1 numaralı şekilde incelenebilir.

4. Bir Q noktasını çift katlı yapmak için, bir teğet çizgisi çizip eğriyi kestiği diğer noktayı buluruz. Eğer bu noktaya S diyecek olursak $Q + Q = 2Q = -S$ eşitliği sağlanır.

4.2. Sonlu Alanlardaki Eliptik Eğriler

ECC için, eliptik eğrilerin, "sonlu alanlardaki eliptik eğriler" olarak tanımlanan bir formu ile ilgilenebilmektedir.[40] Bu şu şekilde gösterilir: p bir asal sayı olsun ve a ve b, p `den küçük, negatif olmayan iki tam sayı olsun:

$$4a^3 + 27b^2 \pmod{p} \neq 0$$

Bu durumda, $E_p(a,b)$, (x,y) `nin $p$ `den küçük negatif olmayan tam sayılar olduğu durum için, O sonsuz noktası ile beraber şu eşitliği ifade eder:

$$y^2 \equiv x^3 + ax + b \pmod{p} \quad (1)$$

Örneğin, $p = 23$ ve eliptik eğrimiz de $y^2 = x^3 + x + 1$ olsun. Bu durumda, $a = b = 1$ olur. Bu durumda, 4 x 1^3 + 27 x 1^2 (mod 23) = 8 $\neq$ 0, bizim eliptik grubumuzun mod 23`e göre durumunu gösterir.

Önceki paragrafta gösterilen eşitlik, 6 şeklindeki ikinci grafik ile aynıdır. Eliptik grup için sadece (mod p`den dolayı), (0, 0)-(p, p) aralığında olan pozitif tamsayılar ile denklem oluştururuz. Tablo 4.1`te, $E_{23}(1, 1)$ için O dışındaki noktaları listelenmiştir. Genel olarak liste aşağıdaki yolla oluşturulmuştur:[39],[40]

1. $0 \le x < p$ koşulunu sağlayan her x değeri için $x^3 + ax + b \pmod p$ denklemi hesaplanmıştır.
2. Önceki adımın her sonucu için, sonucun $\bmod p$ `ye göre çift katlı kökü olup olmadığına bakılır, eğer yoksa bu $x$ değeri için, $E_p(a,b)$ `nin bir değeri yoktur. Aksi taktirde, çift katlı kök koşulunu sağlayan iki adet y vardır (y `nin 0 olduğu durum haricinde). Bu (x,y) değerleri $E_p(a,b)$ `nin noktalarıdır.

Tablo 4 .1 $E_{23}(1, 1)$ Eliptik eğrisi için noktalar.[41]

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

$E_p(a,b)$ için bahsedilmiş kurallar, Şekil 4.1`deki geometrik şekil ile uyuşmaktadır. Kurallar, her $P,Q \in E_p(a,b)$ olacak şekilde alınan noktalar için aşağıdaki gibi gösterilebilir:

1. $P+O=P$.
2. Eğer $P=(x,y)$ ise, $P+(x,-y)=O$ olur. $(x,-y)$ noktası, P`nin negatifidir ve $-P$ olarak gösterilir. Diyelim ki, $(x,-y)$ Şekil 4.1 şeklinde gösterilen $E_p(a,b)$ eliptik eğrisi üzerinde bir nokta. Örneğin, $E_{23}(1, 1)$` de $P=(13,7)$ alalım. Bu durumda $-P=(13,-7)$ olacaktır. Fakat –7 mod 23 e göre 16 ettiğinden dolayı, bizim $-P$ noktamız aslında, yine $E_{23}(1, 1)$`de yer alan (13,16) noktasıdır.
3. Eğer $P=(x_1,y_1)$ ve $Q=(x_2,y_2)$ ise ve $P \neq -Q$ ise, bu durumda, $P+Q=(x_3,y_3)$ şu kuralla hesaplanır:

$$x_3 \equiv \lambda^2 - x_1 - x_2 (\mathrm{mod}\ p)$$
$$y_3 = \lambda(x_1 - x_3) - y_1 (\mathrm{mod}\ p)$$

λ için koşul,

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & eger \quad P \neq Q \quad ise \\ \dfrac{3x_1^2 + a}{2y_1} & eger \quad P = Q \quad ise \end{cases}$$

4.3. Eliptik Eğri Aritmetiği

Eliptik eğrilerin şifrelemede kullanılmasının en büyük avantajı sahip oldukları matematiksel işlemleridir. Bu bölümde eliptik eğriler üstünde iki noktanın toplamı ve aynı noktanın aynılanmasını inceleyeceğiz.[41],[42]

Eliptik eğri E(Fp) üzerindeki iki noktanın toplanarak üçüncü noktanın eldesi için kullanılan yöntem, “chordand-tangent rule” olarak bilinir. Bu toplama işlemiyle E(Fp) üzerindeki noktalar kümesi, bir grup oluşturur.[43]

4.3.1. İki Farklı P ve Q Noktasının Eklenmesi

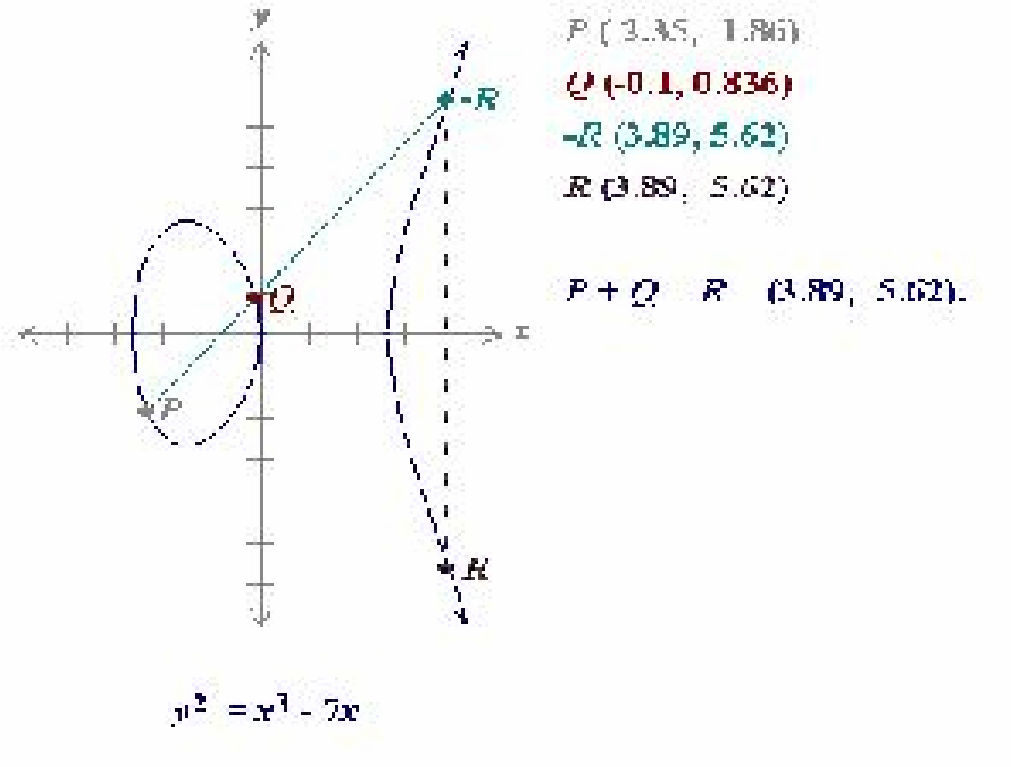
İki noktanın toplamı olan P + Q = R hesaplamak için aşağıdaki denklem çifti kullanılarak eğrideki yeni noktanın koordinatları hesaplanır.[42],[44]

$$P= (x_p , y_p),\ \ Q=(x_q , y_q) \text{ ve } R= (x_r , y_r) \text{ olsun}$$

$$x_r = [\lambda^2 – x_p – x_q] \text{ modp}$$

$$yr = [- y_p + \lambda (x_p – x_r)] \text{ modp}$$

Burada $\lambda = (y_p – y_q)/ (x_r – x_q)$ ise R ve Q noktalarından geçen doğrunun eğimidir.



Şekil 4.2 İki Farklı P ve Q Noktasının Eklenmesi Eliptik Eğri Grafiği[42]

4.3.1.1. Toplama Aritmetiği Sayısal Örneği

F_{23} üzerinde tanımlı eliptik eğri, E: $y^2 = x^3 + x + 4$, ve P = (7,3) ve Q = (8,8) ise; P+Q=R=(x_3, y_3) şöyle bulunabilir:

- x_3= [(8-3)/(8-7)]2 –8–7=52 –15=10 ≡ 10 (mod 23)
- y_3=5(7-10)–3=-18 ≡ 5 (mod 23)

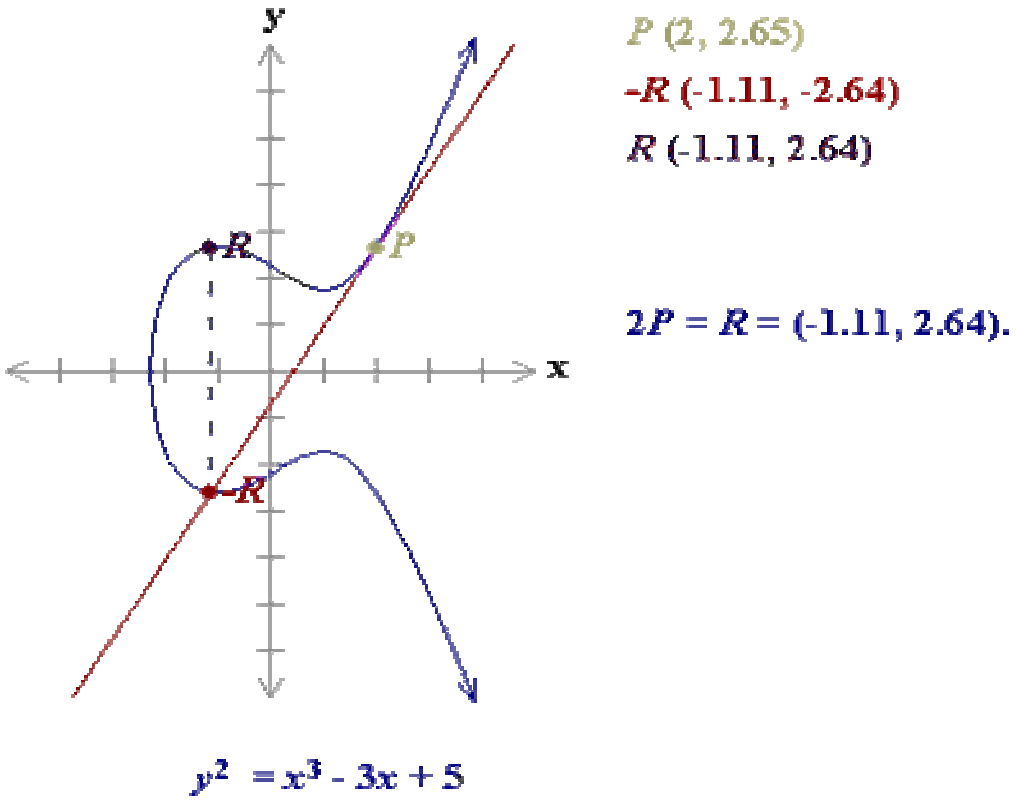
O halde P+Q=R=(10,5).

4.3.2. Bir Noktanın Aynılanması

P = (x_p , y_p) noktasının aynılanması için P +P = R işlemi aşağıdaki denklem çifti kullanılarak gerçekleştirilir.[41],[42]

- $x_r = [\lambda^2 - 2xp] \bmod p$
- $y_r = [- y_p + \lambda (x_p – xr)] \bmod p$

bura da $\lambda = (xp^2 – a) / (2yp)$ eğimdir ve a ‘da eğri parametresidir.



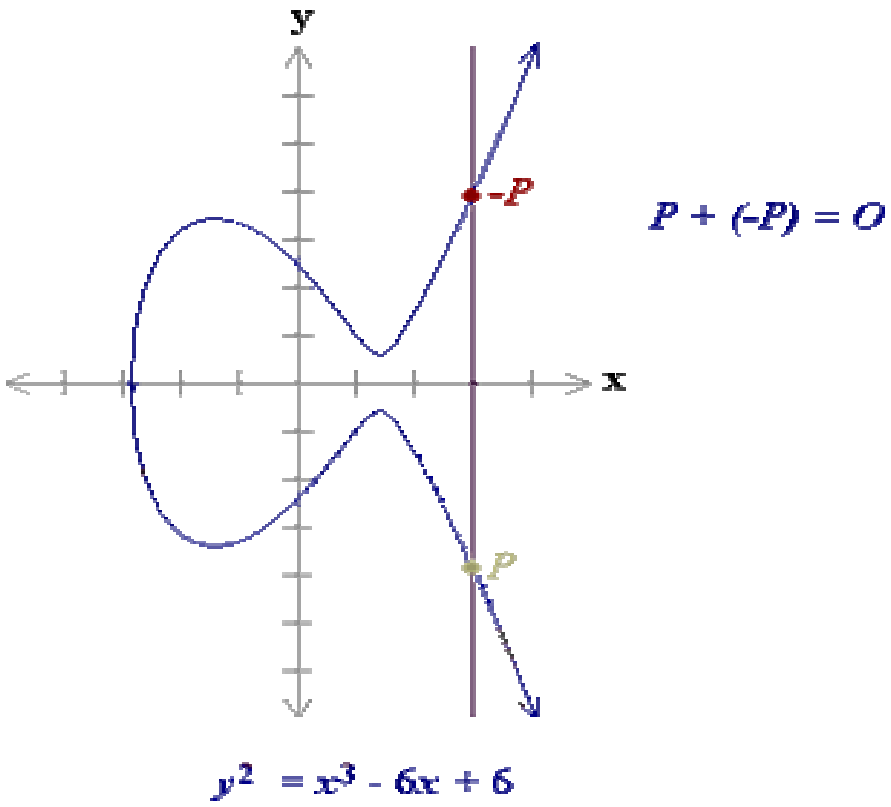
Şekil 4.3 Bir Noktanın Aynılanması[45]

Eğri üzerindeki bir noktayı bulmak ve bu noktayı aynılayarak neticede sonsuzdaki (O) noktasının bulmak için, nokta sonsuzdaki noktaya ulaşıncaya kadar

kendisine eklenir, eklenme sayısına noktanın derecesi denir. Kriptografik uygulamalarda, şifreleme süreci için global açık parametre olan temel nokta yüksek dereceden bir nokta olarak seçilir.[45]

4.3.3. Aynı noktanın tersi ile toplanması

Bir P noktasının tersi olan –P noktası ile toplanması sonsuza gitmektedir**.**



Şekil 4.4 Aynı noktanın tersi ile toplanması[45]

4.4. Şifreleme Algoritmaları İçin Eliptik Eğrilerin Seçimi

Şifreleme işleminde kullanılacak eliptik eğriyi seçerken bilinen tüm ataklara karşı dirençli olması sağlanmalıdır.

- Pollard-*p* atağına karşı koymak için E(Fq)'nun yeterince büyük bir n asal sayı çarpanı olmalıdır.($n > 2^{160}$)
- Weil ve Tate ikili atakları için n'nin (q^k-1) sayısını $1 \leq k \leq C$ için bölmememsi gerekir.[46]

4.5. Eliptik Eğri Şifrelemesi/Deşifrelemesi

Literatürde, eliptik eğriler yardımıyla şifreleme/deşifreleme yapan bir çok yöntem bulunmaktadır. Biz burada elbette en basitini inceleyeceğiz. Bu sistem içerisindeki ilk görev, plaintext mesaj olan m`yi, bir x-y koordını ile belirlenmiş P_m noktası şeklinde göndermek üzere encode etmektir. Bu $P_m$noktası bir chipertext gibi şifrelenecek daha sonrasında da deşifre edilecektir. Bir mesajı x ya da sadece y koordinat noktası olarak basitçe encode edemeyiz çünkü, tüm olası noktalar $E_p(a,b)$ içinde bulunmayabilir. Elbette bu encode işlemi içinde bir çok yöntem mevcut fakat burada bu yöntemlerden bahsetmeyeceğiz.[47]

Anahtar değişimi sisteminde olduğu gibi şifreleme / deşifreleme sistemi de parametre olarak bir Gnoktası ve bir $E_p(a,b)$ eliptik grubuna ihtiyaç duyar. Her bir kullanıcı, bir n_A özel anahtarı seçer ve $P_A = n_A G$ ile bir açık anahtar üretir.

P_m gibi bir mesajı şifrelemek ve bir B kullanıcısına göndermek isteyen bir A kullanıcısı, rastgele pozitif bir k tam sayısı seçer ve C_m şifreli metnin noktalarını aşağıdaki şekilde elde eder:

$$C_m = \{kG, P_m + kP_b\}$$

A kullanıcısının şifreleme esnasında B`nin açık anahtarı olan $P_b$`den yararlandığına dikkat ediniz. B aşağıdaki şekilde mesajı deşifre eder:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

A, P_m mesajını, ona kP_B ekleyerek maskeledi. k değerini bilmeyen kimse A`nın uyguladığı maskeyi kaldıramaz. Bunun yanında A, n_B özel anahtarını bilen bir kişinin maskeyi kaldırabilmesi için bir ipucu bırakmaktadır. [48]

4.6. Eliptik Eğri Kriptografisinin Güvenliği

ECC`nin güvenliği, $kP$ ve $P$ verildiğinde $k$ değerini elde etmenin zorluğuna bağlıdır. Bu durum, eliptik eğri ayrık logaritması problemi olarak adlandırılır. Eliptik eğri logaritması alan bilinen en hızlı teknik Pollard rho (rho=eşkenar dörtgen) yöntemidir. Tablo 4.2`te, eliptik eğri metodu ile RSA`daki tamsayının iki asal çarpanına generalized number field sieve yöntemi ile ayrılması esnasında gereken hesapsal karşılaştırılmıştır.

Tablodan anlaşılacağı üzere, RSA`nın sağladığı direnci ECC, çok daha düşük anahtar boyutları ile sağlamaktadır. Bu yüzden ECC, düşük anahtar boyutu ile sağladığı

yüksek güvenlik sayesinde RSA`ya karşı büyük bir hesapsal üstünlük sağlamaktadır.[48],[49]

Tablo 4.2 Anahtar Uzunluk Karşılaştırılması[48]

Anahtar Uzunluğu	MIPS
150	$3.8 * 10^{10}$
205	$7.1 * 10^{18}$
234	$1.6 * 10^{28}$

a) Pollard Rho Method

Anahtar Uzunluğu	MIPS
512	$3 * 10^{4}$
768	$2 * 10^{8}$
1024	$3 * 10^{11}$
1280	$1 * 10^{14}$
1536	$3 * 10^{16}$
2048	$3 * 10^{20}$

b)Tam Sayı Faktörler Ayırma (General Number Field Sieve)

4.7. Gelecek Neden ECC?

Çoğu kuruluş, çalışanların verimliliğini arttırmak ve ağ üzerinde uygun bir işbirliği sağlamak için kablosuz ağ sistemini yaygınlaştırıyor. Bu ağ sistemlerinin korunması büyük bir önem teşkil eder. Çünkü kablosuz ağ trafiği kolayca saldırılara açık olabilir. [50]

Kablosuz ağ kullanıcılarının isteklerine bakıldığında güvenliğin en önemli koşul olduğu gözlenir. Bununla birlikte kablosuz donanımlara güvenlik düzeyinin kurulması kaynakların sınırlarına karşı bir meydan okuma, kaynakların sınırlarını zorlama anlamına gelebilir (depolama,batarya ömrü,güç oluşumu gibi). Çözüm ise bu kaynaklardan bazılarının küçük pratik değerlerle tüketilmesiyle elde edilebilir.

Sonuçta, araştırmacılar bazen kendilerini zorlanmış kaynakların ortamının güvenliğine yer bulma uyuşması içinde bulurlar. Olumlu gelişme ise araştırmacıların güvenlik sistemlerini ECC ile sağlanmış performans konularına uyuşturma zorunluluklarının olmaması olmustur.[50]

Bazı güvenlik sistemleri 1024-bit RSA genel anahtarlama planının uygulamasını yaymaya çalışmaktadır. Çünkü kuruluşlar bunun yeterince iyi olduğunu düşünmektedir.

Bununla birlikte bu tehlikeli bir yaklaşımdır. Çünkü genel anahtarlama sisteminin güvenliği kullanılan simetrik şifrelemeyle birebir eşleşmiş olmalıdır. Tablo 4.3 de görüldüğü gibi 1024-bit RSA, simetrik şifrelemede kullanılan 128-bit güvenlik seviyesiyle uyuşmuyor.[50] Bu gereksinimi karşılamak yani genel anahtarlama planını eşleştirmek için istenen 3072-bit RSA yada 256-bit ECC kullanılmasıdır.[42],[50]

Tablo 4.3. ECC ile Diğer Algoritmaların Anahtar Uzunluklarının Karşılaştırılması

Güvenlik (Bits)	**Simetrik Şifreleme Algoritmaları**	**DSA / DH**	**RSA**	**ECC**
80	Skipjack	1024	1024	160
112	3DES	2048	2048	224
128	AES-128	3072	3072	256
192	AES-192	7680	7680	384
256	AES-256	15360	1530	512

4.8. Eliptik Eğri Ayrık Logaritma Problemi (ECDLP) ve Kriptanalatik Yaklaşımlar

ECC`deki toplama işlemi ile RSA`daki modüler çarpım işlemi ve çoklu toplama işlemi ile de, RSA`daki modüler üs alma işlemi birbirlerine çok benzemektedir. Eliptik eğriler kullanan bir kriptografik sistem oluşturabilmek için, bir sayıyı iki asal çarpanına ayırmak ya da ayrık logaritma almak gibi zor bir problem bulmamız gerekmektedir.

1985 ‘de Neal Koblitz ve Victor Miller tarafından bulunan sonlu alanlar üzerinde eliptik eğriler deki ayrık logaritma denetlenemez görünümümdedir. Eliptik eğri ayrık logaritma problemi(ECDLP) aşağıdaki şekilde açıklanabilir:

- P, büyük asaldır ve büyük dereceli P, E eğrisi üzerindedir.
- x.P , P’nin skaler çarpımı olarak verilsin, x kere (aynı zamanda P’nin x kere kendi üzerinde toplanmasıdır.)
- Q, x.P = Q ‘i sağlayan eğri üzerindeki diğer bir noktadır.
- Eliptik eğri ayrık logaritma problemi, verilen P ve Q ile x değerinin bulunmasıdır.

ECC şifreleme algoritması ECDLP (Eliptik Eğri Ayrık Logaritma Problemi) problemi üstüne kurulmuş bir şifreleme sistemi olduğunu söylemiştik. ECC algoritması üstüne yapılan kriptanalizler bu problemin çözümü üzerine yapılan çalışmalarla gerçekleştirilebilir. [44], [51]Bu çalışmaların en önemlileri:

1. Pollard Rho Algoritması
2. Gaudry – Hess – Smart Attack (Ghs)
3. Weil Descent

ECDLP’yi çözmek için kullanılan algoritmalardan en bilineni ve en genel-amaçlı olan algoritma Pollard Rho algoritmasıdır. [44]

Sonlu bir A alanı eliptik eğri kriptografisinde eğer k üzerindeki tüm eliptik eğriler ayrık logaritma problemi için örnekleniyor ise örneklerin zayıf olduğu söylenmektedir. Bunlar Pollard Rho metodu ile daha zor örneklerin çözümünden daha kısa sürede çözülebilmektedir.

4.9 Pollard Rho Algoritması

Bu algoritma için öncelikle modüler aritmetik ve Modül işlemleri tanımlanmalıdır[52]

4.9.1 Modüler Aritmetik

x ve y birer tam sayı olsun. Eğer (x-y), n' nin bir çarpanı ise x ve y' nin n 'ne bölümü aynıdır

$$x = q\ x*n + rx$$

$$y = q\ y*n + ry$$

Eğer $rx = ry$

$$q\ x*n - q\ y*n + r\ x - r\ y = (q\ x - q\ y) * n$$

Örnek:

$$x = 37,\ y = -14 \text{ and } n = 17.$$

$$(x-y) = 37 + 14 = 51 = 3*17$$

$$x = 2*17 + 3 \text{ and } y = (-1)*17 + 3$$

Modül işlemleri

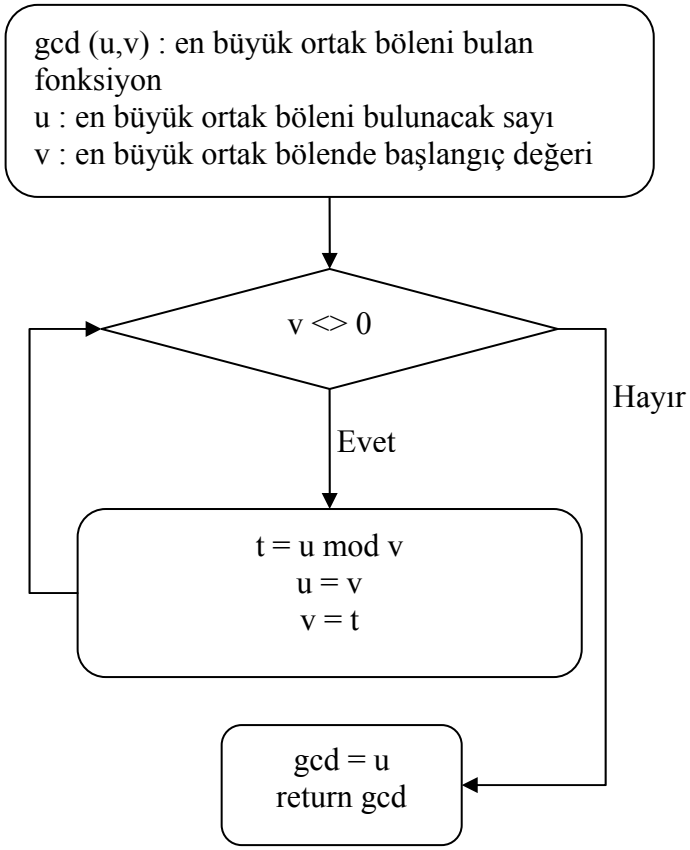
$$37 + 15 = 3 + 15 = 18 = 1 \pmod{17}$$

$$374 = 34 = (\ 33\ )*3 = 27 * 3 = 10 * 3 = 30 = 13 \pmod{17}$$

4.9.2 En Büyük Ortak Bölen

658 ve 154 en büyük ortak bölenini bulalım

658 = 4 * 154 + 42
154 = 3 * 42 + 28
42 = 1 * 28 + 14
28 = 2 * 14 + 0



Şekil 4.5 En Büyük Ortak Böleni Bulma Algoritması

4.9.3 Pollard Rho Algoritması

Örneğin çok büyük bir n sayısı alalım. L=16843009. Bu sayının asal olmadığını biliniyor ve iki asal çarpandan oluştuğunu bilinmektedir. d bu n sayısının bir böleni olsun
d sayısının asal olduğunu, n'den ufak olduğunu ve kök n den küçük olduğunu biliyoruz.

$$a = b \bmod n$$

Bu noktada a ve b hemen hemen aynı sayılara eşit olduğu zaman

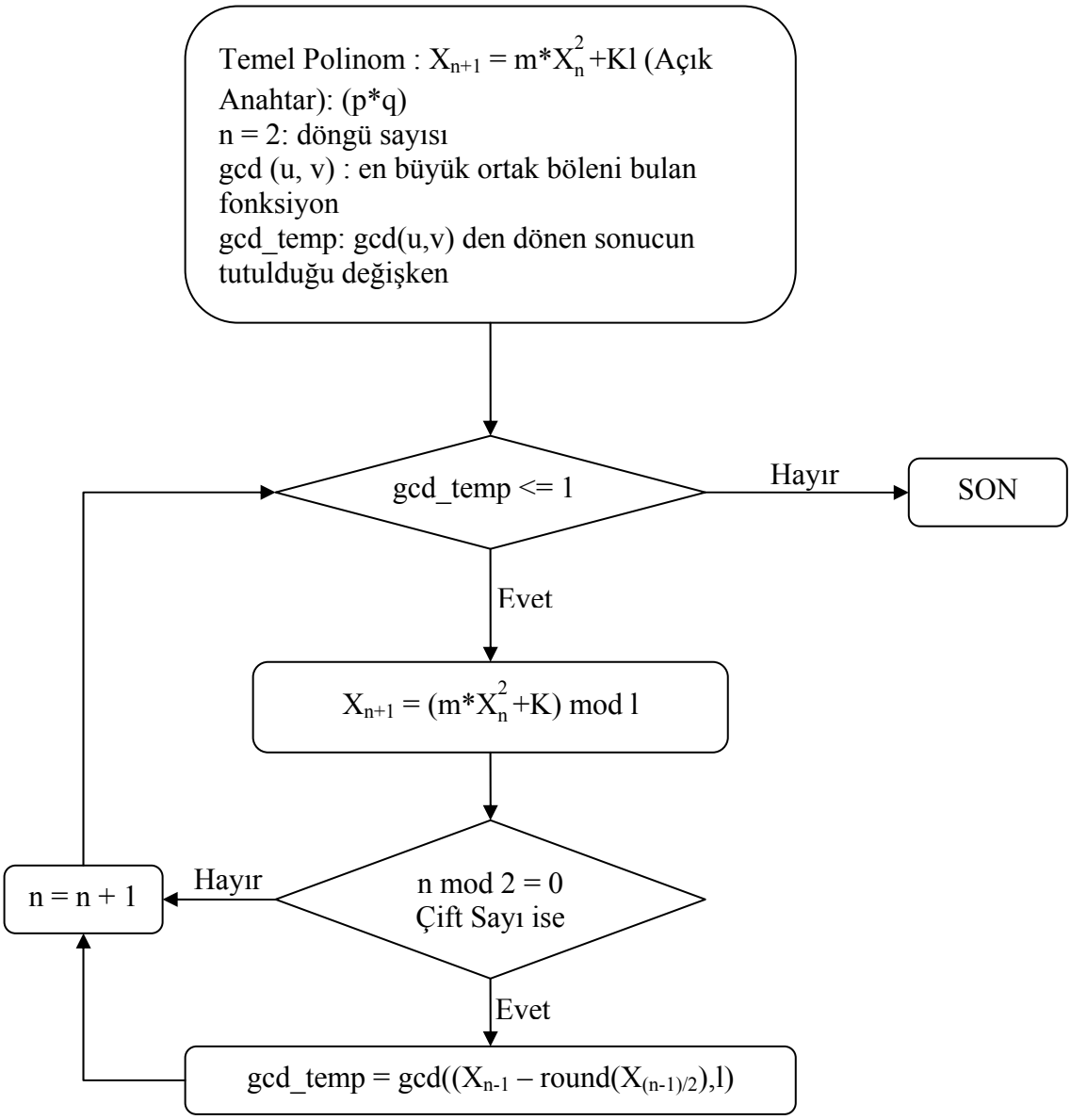
$$a = b \bmod d$$

denkleminin de doğru olduğunu söyleyenebilir. a=b mod d denkleminden yukarıda gösterdiğimiz teoremlerden yola çıkarak (a-b) değeri de d sayısının bir çarpanıdır. d de n'nin bir çarpanı olduğuna göre (a-b) n'nin bir çarpanıdır.

Bu algoritmaya göre randam a ve b sayısı alıyoruz. (a-b) ve n sayısının en büyük ortak bölenin 1 den birden büyük değere sahip noktayı bulmaya çalışılır. Bu olmazsa bir c sayısı alınır. (c-b) yada (c-a) sayılarının n ile en büyük ortak bölenini bulmaya çalışılmaktadır.

Görüldüğü gibi bu biraz hantal bir işlem Bundan dolayı Pollard Rho algoritması bunu gidermek için k iterasyonunu ve uygun bir polinom kullanılmaktadır.

$$X_1=1 \text{ with } x_{n+1}= 1024*x^2_n + 32767$$



Şekil 4.6 Pollard Rho Algoritması

5. SİMETRİK ŞİFRELEME ALGORİTMALARI

5.1. Günümüzde Kullanılan Simetrik Algoritmalar

Günümüzde kullanılan birçok şifreleme yöntemi esas olarak yer değiştirme ve dönüştürme yöntemlerinin bir arada kullanıldığı yapılardan oluşur. Bu algoritmaların güvenilirliği, çözülmeleri için gerekli işlem miktarının yüksek oluşuna bağlıdır. Burada tanıtılan tüm yöntemlerin güvenilirliği en doğru biçimde, seçilmiş açık bir metin ile ondan elde edilmiş gizli metin arasındaki ilişkinin, istatistiksel olarak değerlendirilmesiyle mümkündür. Açık metin dizisi üzerinde her bir karakter yerine kullanılan bit dizisi üzerinde yapılan kriptolama işine dizi kriptolama denir. Diğer bir yöntem ise metnin bloklara ayrılması ve bunlar üzerinde şifrelemenin gerçekleştirildiği blok kriptolamadır.[53]

5.1.1. Dizi Kriptolama (STREAM CHIPHERS)

Geçmişte kullanılan yer değiştirme algoritmalarının günümüzde kullanılan biçimidir. Dizi kriptolama yönteminde, tek kullanımlı şerit yönteminde kullanıldığı gibi, uzun anahtar bilgisine ihtiyaç vardır. Bu sebepten, yarı rassal nitelikte bir anahtar üretmek amacıyla, geri beslemeli öteleme kütüklerinden yararlanılır. Üretilen anahtar ile açık metin dış veya'lanarak gizli metin elde edilir. Şifrenin çözülebilmesi için üretilen anahtarın alıcı tarafta da üretilmesi gerekir. Bunun sonucunda gizli metin ile dış veya'lanan anahtar bilgisi açık metni verecektir. Ancak anahtar üretecinde doğrusal olmayan bir yöntem kullanılmayacak olursa bu şifreleme yöntemi bilinen metin saldırısına açık olacaktır.[1],[3]

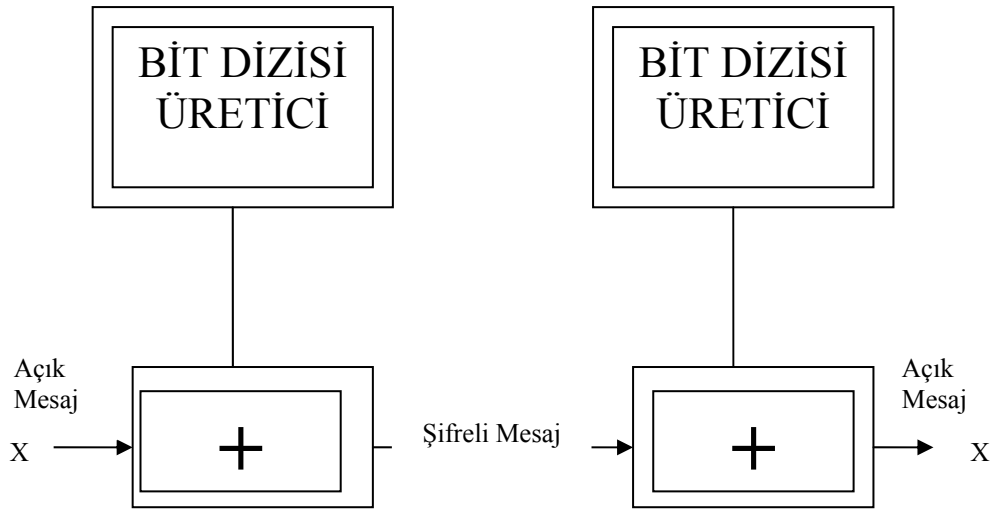
Dizi şifreleyiciler, gizli anahtarlı şifreleme sistemler içinde yer alırlar. Bugün en çok kullanılan dizi şifreleyiciler, iki toplayıcı dizi şifreleyicileridir. Böyle bir şifreleyicide K bitlik gizli anahtar (Z) bir kayan anahtar üretici kontrol etmek için kullanılır. Kayan Anahtar Üretici Z_1, Z_2,.....,Zn ikili dizisini üretir. Bu dizi kayan anahtar olarak adlandırılır. Genellikle N>>K dir.Şifreli mesaj dizileri basit olarak söylenirse açık mesajın modulo 2'ye göre toplanmasından oluşur.[43]

$$Y_n=X_n+Zn \qquad n=1,2,.....,N \text{ dir.}$$

Şifreleme ve şifre çözme aynı araçlarla yapılır. Açık mesajın bir tek bitin, şifreli mesajın bir tek bitini etkilerse bu, kötü yayılma özelliğidir. Fakat anahtar biti şifreli mesajın bir çok bitini etkilerse bu da, iyi yayılma özelliğidir.

İkili toplayıcılı dizi şifreleyicilerle bir tek ikili one-time-pad (tek kullanımlı anahtar) arasında kesin bir benzerlik vardır. Eğer $Z=Z_n$ ise yani gizli anahtar kayan anahtar olarak kullanılırsa toplayıcılı dizi şifreleyici one-time-pad'in aynısı olur.

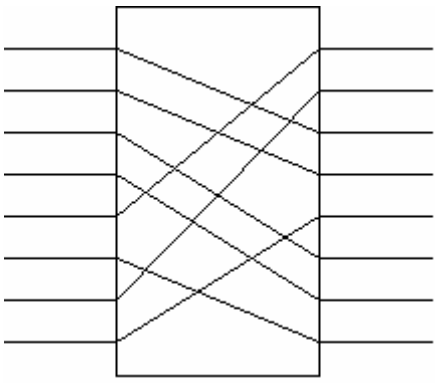
Dizi şifreleyiciler, blok şifreleyicilere göre daha güvenilirdirler fakat burada senkronizasyon problemi vardır. Çünkü açık mesajın şifrelendiği anahtarın şifre çözümü sırasında oluşması için kesin senkronizasyon yapılmış olması gerekir. Burada anahtar zamanla değişen bir fonksiyondur.



Şekil 5.1. Dizi Şifreleyici Blok Diyagramı[43]

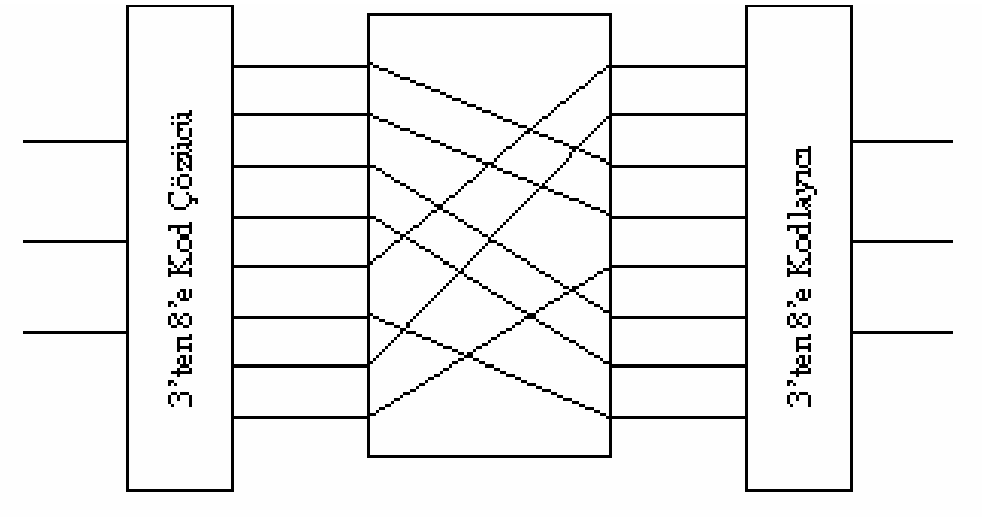
5.1.2. Blok Kriptolama (BLOCK CHIPHERS)

Günümüzde kullanılan blok şifreleme metodu temel olarak basit yer değiştirme ve dönüşüm şifreleme yöntemlerine dayanır. Her bir blok 32'den 128'e kadar değişen oktet uzunluklarından oluşur. Anahtar uzunluğu blok uzunluğuna eşittir.[52],[54]



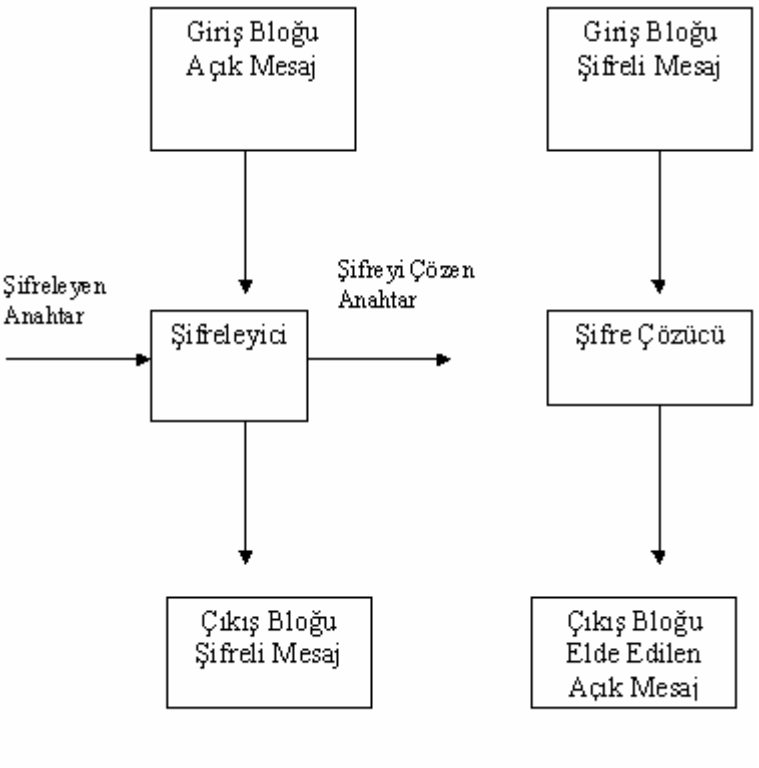
Şekil 5.2 Permitasyon Kutusu[52]

Blok şifrelemeyi oluşturan yapılardan biri P (Permütasyon), diğeri S (Yer Değiştirme) kutusudur. Şekil 5.2.`de bir P kutusu gösterilmektedir. P kutusunun amacı, girişinde kullanılan bit dizisinin sıralamasını belirli bir permütasyon uyarınca değiştirmektir. S kutusu Şekil 5.3.`de gösterildiği gibi üç aşamadan oluşmaktadır. Bunlardan birincisi kod çözücü, ikincisi bir P kutusu, üçüncüsü ise bir kodlayıcıdır. P ve S kutularının bir araya getirilmesi ile güçlü şifre algoritmaları oluşturulabilir. Yaygın bir şekilde kullanılan Veri Kriptolama Standardı (DES) bu şekilde tasarlanmıştır.



Şekil 5.3 Yerdeğiştirme Kutusu[52]

Bir blok şifreleyici sabit uzunluktaki bir dizi giriş bitini sabit uzunluktaki bir dizi çıkış bitine dönüştürür. Şifreleme ve şifre çözme fonksiyonları öyle tasarlanmışlardır ki çıkış bloğundaki her bir bit giriş bloğundaki ve anahtardaki her bir bite bağımlıdır. Bir şifrenin blok uzunluğu (blok içindeki bitlerin sayısı) kriptografik olarak sağlamlık düşüncesi ile belirlenir, dolayısı ile basit, şifreli mesaj çözme saldırılarını azaltabilmek için yeterli miktarda büyük olmalıdırlar.



Şekil 5.4 Blok Şifreleme[52]

5.1.3. Blok Şifreleme ile Dizi Şifreleme Arasındaki Temel Farklar

Blok şifreler, bir anda bir tek veri bloğunu şifreler. Kriptografik sağlamlık düşüncesi ile belirlenen minimum blok büyüklüğü blok şifreler için yeterlidir. Dizi şifrelerin minimum blok büyüklükleri yoktur. (Dizi şifreler minimum blok büyüklüğüne gerek duymazlar) En uç noktada 1 bit'e 1 bit temeline dayalı şifreleme kullanırlar.[55]

Blok şifrelemede, her şifreli bit (şifrelenmiş bit) uygun giriş bloğundaki her açık mesaj bitinin karmaşık bir fonksiyonudur. Dizi şifrelemede, her şifreli bit y(i), onun açık mesaj biti x(i)'ye y(i)=x(i)+r(i) ilişkisi ile bağlıdır.

Blok şifreleyiciler bir başlangıç vektörüne ihtiyaç duyabilirler yada duymayabilirler. Açık mesaj bilgisi ve uygun mesaj dizi şifrelemede olduğu gibi aynı yolla bilgiyi açığa çıkarmaz. Kriptografik olarak güçlü dizi şifreleyici tekrar üretilemez

(reorginate) ve bu yüzden bir başlatma vektörü gerektirir. Bir başlangıç vektörü blok şifreleyici için her zaman bir ihtiyaç olmamasına rağmen, blok zincirlemede kullanılırlar. Yeniden üretilen (reoriginate) blok şifreleyicilerde, başlangıç vektörü (initializing vector) sınırlı zaman periyodunda kullanılabilir.

Blok ve Dizi şifreleyicilerin her ikisi de iletişimde ve veri işleme sistemlerinde (Data Processing System) kullanılabilir. Bir blok şifreleyci ile, veri şifrelenir ve bloklardaki şifre çözülür. Bu işlemlere tabi tutulan uzunluk önceden algoritma tasarımcısı tarafından belirlenmiştir. Bir dizi şifreleyicisinde ise şifrelenen ve şifresi çözülen verinin uzunluğu kullanıcı tarafından belirlenir. Bu durum dizi şifrelemeye esneklik getirir. Dizi şifreleme, algoritma ve anahtara ek olarak başlama vektörü (initializing vector) olarak tanımlanan diğer bir parametreyi de kullanır. Blok ve Dizi Şifreleme'de geri besleme modları elde edilebilir. Zincirleme, şifrelemeyi yalnızca güçlendirmez aynı zamanda öncelik gerektirmediği durumlarda datayı doğrulamak içinde kullanılabilir.

Bir kriptografik sistemin güçlülüğü, zincirleme metodu kullanılarak zenginleştirilebilir.

Zincirleme (chaining) şifreleme işlemi sırasında kullanılabilen bir prosedürdür. Buna göre bir çıkış bloğu sadece o andaki giriş bloğu ve anahtara değil fakat aynı zamanda daha önceki giriş yada çıkışa da bağlıdır.

Kriptografide (şifrelemede) temel problem, açık mesajı şifreli mesaja dönüştüren kriptoanalize dayanıklı prosedürler oluşturmaktır. Çünkü kriptoanaliz, şifreli iletişimin içine nüfuz etmek (girmek) ve kaynak bilgiyi elde etmek için değişik teknikler kullanır.

Böyle sistemleri kullanan prosedürler ya kod sistemi yada şifreli sistem içerirler. Kod sistemler bir kod kitaba yada sözlüğe ihtiyaç duyarlar. Bu kitap yada sözlükler kelimeleri, deyimleri ve açık mesaj cümlelerini buna eşdeğer şifreli kod grubuna dönüştürürler. Bununla birlikte dönüştürülebilen açık mesaj gruplarının sayısı kod

kitabının büyüklüğüne bağlıdır. Ayrıca her mesaj kodlanamayabilir ve bu kod sisteminin yeteneği sınırlıdır.

Diğer taraftan, şifreleme sistemleri yeteneklidir. İki temel elemana ihtiyaç duyarlar: Bir kriptografik Algoritma, bir prosedür yada doğada değişmez olan bir kurallar yada adımlar kümesi; ve kriptografik anahtar değişkenlerinin bir kümesidir. Anahtar; kullanıcı tarafından seçilen, nispeten kısa gizli karakter yada sayı dizisidir.

Simetrik Algoritmalar (DES gibi) ve Asimetrik Algoritmaları (RSA algoritması) blok şifreleyici algoritmalar arasında gösterilebilirler.[56]

5.2. Veri Şifreleme Standardı – Data Encryption Standard (DES)

En çok kullanılan şifreleme tekniği 1977'de şimdiki adı Ulusal Standart ve Teknolojiler Enstitüsü olan Ulusal Standartlar Bürosunda ortaya atılan Veri Şifreleme Standardıdır (DES). DES' de veri, 56-bitlik bir anahtar kullanılarak 64-bitlik bloklar halinde şifrelenir. Algoritma, 64-bitlik bir girişi bazı aşamalar sonucu 64-bitlik bir çıktı oluşturacak şekilde dönüştürür. Şifrelemeyi geri almak için aynı adımlar, aynı anahtar kullanılarak işlenir.[54]

DES' in çok geniş bir kullanım sahası vardır. Güvenilirlik derecesi ise tartışıla gelen bir konu olmuştur. DES' in tarihçesi şu şekildedir:

1960'ların sonunda IBM , Horst Feistel'in liderliğinde bilgisayar şifrelemeleri için bir proje araştırması başlatılmıştır. Proje, 1971'de LUCIFER adı verilen bir algoritmanın geliştirilmesiyle sonuçlandı. LUCIFER, 64-bitlik blokları,128-bitlik bir anahtarla şifreleyen bir yapıdaydır. LUCIFER'i kazanç haline dönüştürmek isteyen IBM, tek bir çip halinde bir şifreleyici ürün geliştirme peşine düştü. Bu görevi yürütülmesi Walter Tuchman Carl Meyer başkanlığında, sadece IBM araştırmacılarına

değil ayrıca diğer bilirkişi ve teknik danışmanlarca yapıldı. Bu çalışmanın ürünü LUCIFER'in tasfiye edilmiş ve anahtar büyüklüğünün azaltılarak 56-bite düşürülmüş tek bir çipe sığdırılabilen bir versiyonudur.[53]

Bu arada, Ulusal Standartlar Bürosu 1973'te ulusal bir şifreleme standardı arayışı içindeydi. IBM; Tuchman-Meyer projesinin sonuçlarını bildirdi. Bu, şimdiye kadar oluşturulmuş en iyi algoritmaydı ve 1977'de Data Encryption Standard (DES) olarak kabul edildi.

5.2.1. DES Şifreleme Algoritması

DES şifrelemesinin akışı Şekil 5.5'de gösterilmiştir. Diğer tüm şifreleme yöntemleri gibi bunda da iki girdi vardır: şifrelenecek düzyazı ve anahtar. Burada düzyazı 64-bit, anahtar ise 56-bit uzunluğunda olmalıdır.

Şeklin sol yanına bakarsak, düzyazının üç safhada işlendiğini görürüz. Önce, 64-bitlik düzyazı bir başlangıç permütasyonundan geçerek, bitlerinin sırası değiştirilmiş bir hali elde edilir. Bunu hem permütasyon hem de yerinde koyma fonksiyonlarını içeren bu fonksiyonunun 16 kez tekrar edilmesi izler. Sonuncu (16.) tekrar, giriş değerleri olan düzyazı ve anahtarın 64-bitlik bir sentezidir. Bu çıktı sol ve sağ olmak üzere iki parçaya ayrılır ve ön çıktı adını alır. Son olarak, ön çıktı başlangıç permütasyon fonksiyonunun tersi olan (IP^{-1}) işlemine tabi tutulur ve 64-bitlik şifreli metin oluşur.

Şeklin sağ bölümünde ise 56-bitlik anahtarın işlenmesi gösteriliyor. Başlangıç olarak anahtar, bir permütasyon fonksiyonuna gönderilir. Sonra her bir 16 tekrarı için, sol kaydırma döngüsü ve permütasyonu içeren işlem sonucu bir alt anahtar (K_i) üretilir. Permütasyon fonksiyonu her tekrar için aynıdır ancak anahtar bitlerinin tekrarlı kayması sonucu her seferinde farklı bir alt anahtar üretilir.

5.2.2. Başlangıç Permütasyonu

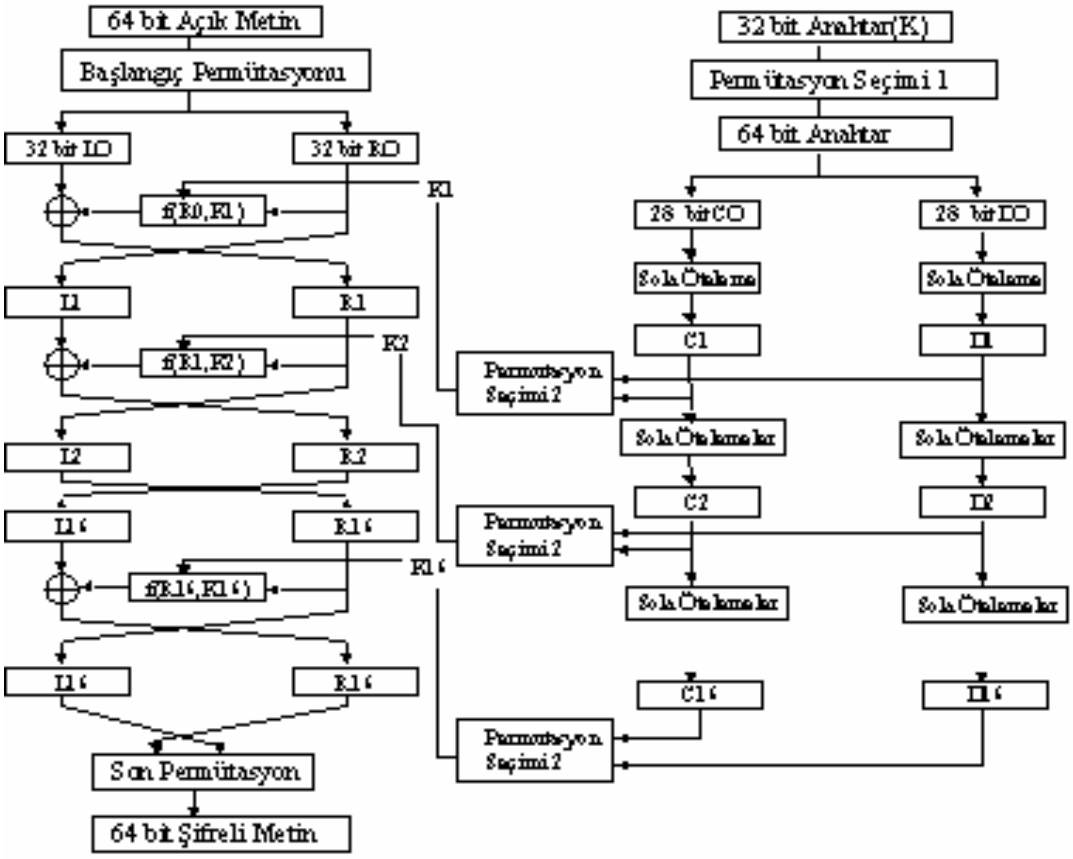
Başlangıç permütasyonu ve tersi aşağıda sırasıyla verilmiştir. Bu iki permütasyon fonksiyonunun birbirinin tersi olduğunu görmek açısından şöyle bir 64-bitlik M'yi giriş olarak alınır:

M_1 M_2 M_3 M_4 M_5 M_6 M_7 M_8 M_9 M_{10} M_{11} M_{12} M_{13} M_{14} M_{15} M_{16}
M_{17} M_{18} M_{19} M_{20} M_{21} M_{22} M_{23} M_{24} M_{25} M_{26} M_{27} M_{28} M_{29} M_{30} M_{31} M_{32}
M_{33} M_{34} M_{35} M_{36} M_{37} M_{38} M_{39} M_{40} M_{41} M_{42} M_{43} M_{44} M_{45} M_{46} M_{47} M_{48}
M_{49} M_{50} M_{51} M_{52} M_{53} M_{54} M_{55} M_{56} M_{57} M_{58} M_{59} M_{60} M_{61} M_{62} M_{63} M_{64}

Burada M_i'ler ikili basamaklardır. O halde permütasyon X = IP(M) şu şekilde olur:

M_{58} M_{50} M_{42} M_{34} M_{26} M_{18} M_{10} M_2 M_{60} M_{52} M_{44} M_{36} M_{28} M_{20} M_{12} M_4
M_{62} M_{54} M_{46} M_{38} M_{30} M_{22} M_{14} M_6 M_{64} M_{56} M_{48} M_{40} M_{32} M_{24} M_{16} M_8
M_{57} M_{49} M_{41} M_{33} M_{25} M_{17} M_9 M_1 M_{59} M_{51} M_{43} M_{35} M_{27} M_{19} M_{11} M_3
M_{61} M_{53} M_{45} M_{37} M_{29} M_{21} M_{13} M_5 M_{63} M_{55} M_{47} M_{39} M_{31} M_{23} M_{15} M_7

Bu permütasyonun tersini alınırsa yani $Y = IP^{-1}(IP(M))$ işlemi sonucu orijinal sıralamanın elde edildiği gösterilebilir.



Şekil 5.5. DES Algoritmasının Genel Bir Gösterimi[54]

Tablo 5.1(a) DES için Kullanılan Permütasyon Tabloları[54]

(a) Başlangıç Permütasyonu (IP)																
Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Girdi bitinden	58	50	42	34	26	18	10	2	60	52	44	36	28	20	12	4
Çıktı biti	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Girdi bitinden	62	54	46	38	30	22	14	6	64	56	48	40	32	24	16	8
Çıktı biti	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Girdi bitinden	57	49	41	33	25	17	9	1	59	51	43	35	27	19	11	3
Çıktı biti	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
Girdi bitinden	61	53	45	37	29	21	13	5	63	55	47	39	31	23	15	7

Tablo 5.2 Permitasyon Kutuları[54]

(a) Başlangıç Permütasyonunun Tersi (IP^{-1})

Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Girdi bitinden	40	8	48	16	56	24	64	32	39	7	47	15	55	23	63	31
Çıktı biti	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Girdi bitinden	38	6	46	14	54	22	62	30	37	5	45	13	53	21	61	29
Çıktı biti	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Girdi bitinden	36	4	44	12	52	20	60	28	35	3	43	11	51	19	59	27
Çıktı biti	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
Girdi bitinden	34	2	42	10	50	18	58	26	33	1	41	9	49	17	57	25

(b) Genişleme Permütasyonu (E)

Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Girdi bitinden	32	1	2	3	4	5	4	5	6	7	8	9	8	9	10	11
Çıktı biti	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Girdi bitinden	12	13	12	13	14	15	16	17	16	17	18	19	20	21	20	21
Çıktı biti	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
Girdi bitinden	22	23	24	25	24	25	26	27	28	29	28	29	30	31	32	1

(c)Permütasyon Fonksiyonu (P)

Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Girdi bitinden	16	7	20	21	29	12	28	17	1	15	23	26	5	18	31	10
Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Girdi bitinden	2	8	24	14	32	27	3	9	19	13	30	6	22	11	4	25

5.2.3. Bir Tek Döngünün Ayrıntıları

Şimdi, Şekil 5.5'da gösterilen bir döngünün daha yakından incelemesini yapalım. Yine şeklin sol tarafını ele alarak gösterilir. [1],[3],[53],[56]

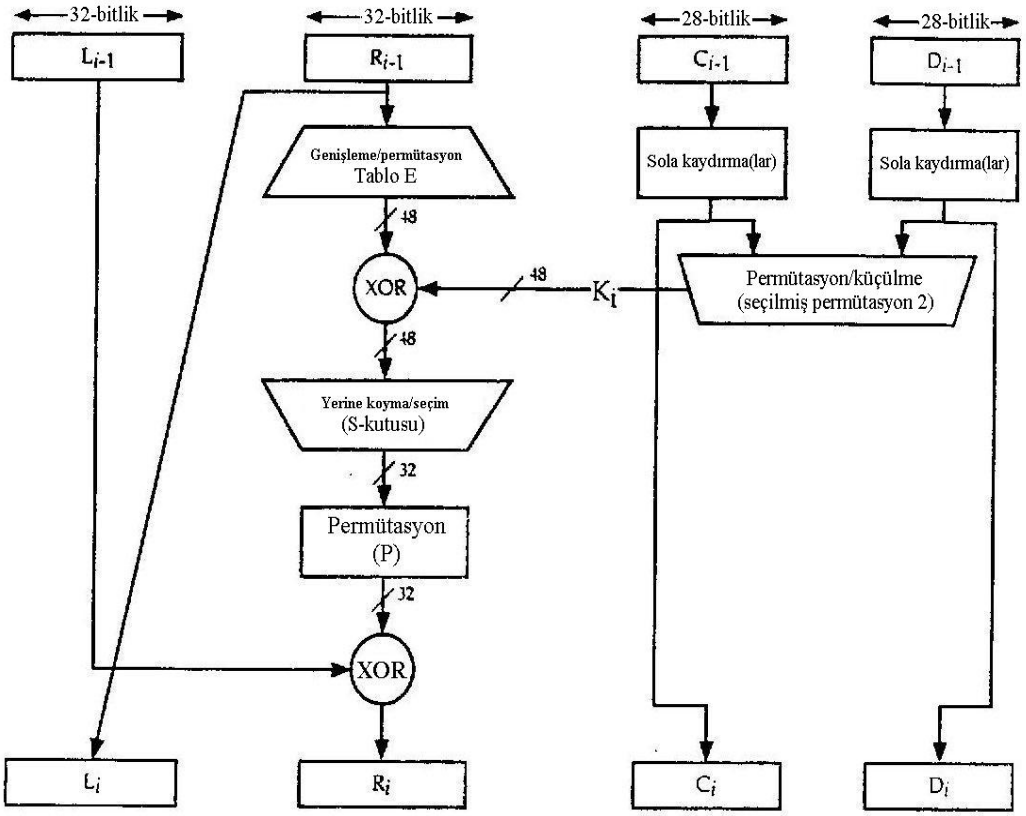
Esas olarak, yer değiştirmiş 64-bitlik giriş, 16 tekrardan geçerek her tekrar sonucu 64-bitlik orta seviyeye sahip bir değer üretir. Her 64-bitlik orta seviyeli değer, sol ve sağ olarak ayrılır ve 32-bitlik bu parçalar L(Left) ve R(Right) olarak adlandırılır.Her bir tekrarın tüm işlemleri şu şekilde şu formüllerle özetlenebilir:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

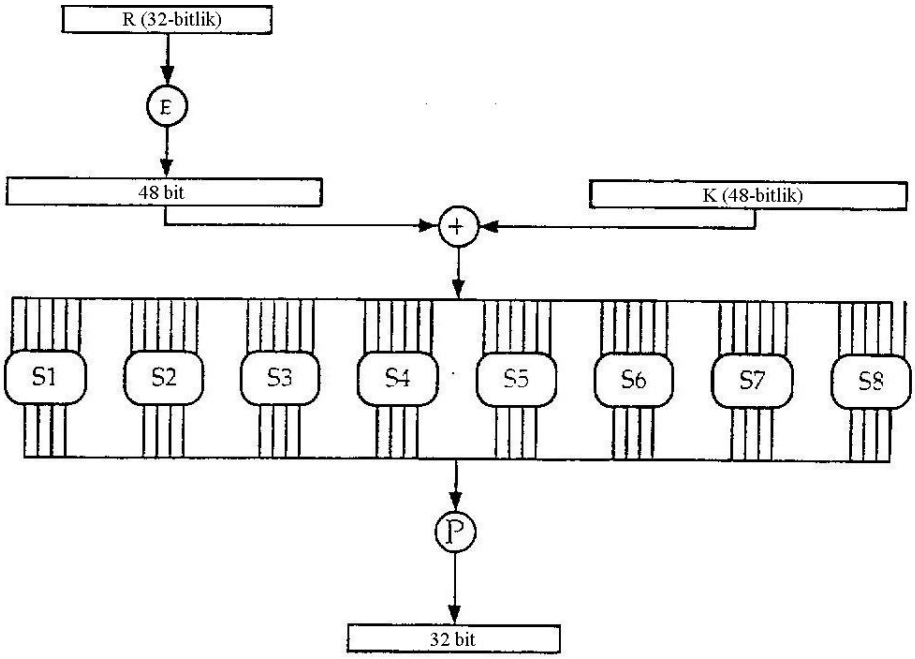
Burada $\oplus$, bit işlemi XOR dur.

Buna bakarak, bir tekrarın sol çıktısının (L_i), basitçe sağ girdi (R_{i-1}) olduğu söylenebilir. Sağ çıktı (R_i) ise, L_{i-1} ve karmaşık f fonksiyonunun özel veyalanması sonucu elde edilir. f fonksiyonu Şekil 5.6'da canlandırılmıştır. Tekrar anahtarı K_i ,48 bittir. R girişi ise 32 bittir. Bu R girişi önce yukarıda verilen işlem sonucu 48 bite genişletilir. Bu genişletme işlemi, bir permütasyon ve R bitlerinin 16 tanesinin tekrarı ile genişletme işlemlerini kapsar. Sonuçta oluşan 48 bit, R_i ile özel veyalanır. Bu 48-bitlik sonuç, 32-bitlik bir çıkış üreten bir yerine koyma fonksiyonuna iletilir. Bu fonksiyon Tablo 5.1(a)'de verilmiştir.



Şekil 5.6 DES Algoritmasının Bir Tur Döngüsü[54],[56]

Yerine koyma işlemi, giriş olarak 6 bit alan ve çıkışta 4 bit üreten 8 adet S-kutusunun bir kümesini içerir. Bu dönüşümler Tablo 5.2(b)'da gösterilmiştir ve şu şekilde ifade edilebilir: S_i kutusuna girdi olan değerin ilk ve son basamakları 2-bitlik ikili bir sayı oluşturur ve bu sayı S_i kutusunun içindeki tabloda hangi satırın alınacağını belirler.



Şekil 5.7 *f*(R,K)'nın Hesaplanması[54]

Geriye kalan ortadaki 4 bit ise sütun numarasını oluşturur. Bu satır ve sütunun kesişimin deki hücrede yer alan onluk değer, çıkış oluşturmak için 4-bitlik gösterimine dönüştürülür. Örneğin S_1 için girdi 011011 olduğu taktirde, satır 01 (yani 1.satır)'dır. Sütun ise 1101 (yani 13. sütun)'dur. 1. satır ve 13. sütunun kesişimin deki değer 5'tir ve bunun dönüştürülmüş hali olan çıktı 0101 olur.

Şekil 5.7, S-kutusu işleminin ayrıntılarını gösteriyor. Birinci ve sonuncu bitlerin oluşturduğu sayı satırlarca tanımlanmış dört permütasyondan birini seçiyor. Şekil 5.8, S_1 kutusundaki sıfırıncı satırın permütasyonunu göstermektedir.

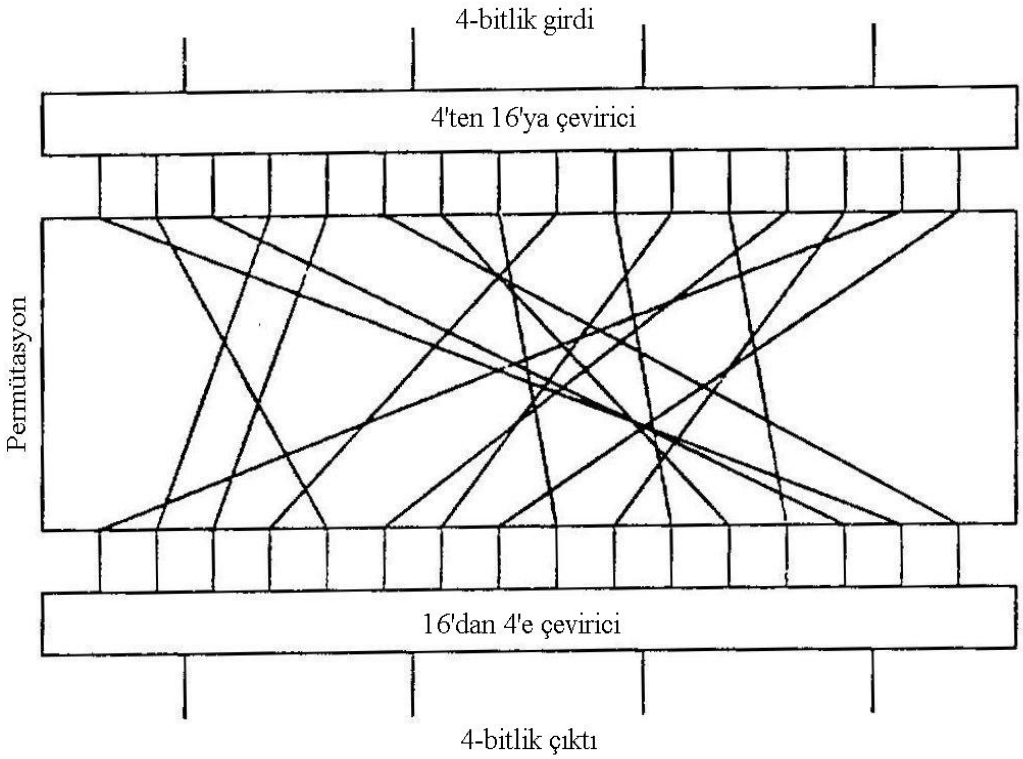
S-kutularının yapısı daha dikkatle incelenmeye değer. K_i'nin elde edilmesini bir yana bırakalım. Şimdi genişleme tablosu incelenirse 32-bitlik girdinin 4-bitlik gruplara bölündüğü ve sonra iki ardışık gruptan dıştaki bitleri alarak 6 bitlik gruplara dönüştüğü görülür. Örneğin giriş kelimesinin bir bölümü şuysa:

.... e f g h i j k l m n o p .....

Dönüşümden sonra şöyle olur:

... d e f g h i h i j k l m l m n p o q .....

Her grubun dıştaki iki biti, dört olası yerine koyma tablosundan birini seçer. Sonra 4-bitlik giriş için (ortada yer alan 4 giriş biti) , 4-bitlik bir çıktı değeri oluşturulur. Sekiz S-kutusundan çıkan 32-bit, permütasyona tabi tutulur öyle ki, bir sonraki tekrarda S-kutularının çıktısı mümkün olan en çok etkiyi sağlamaktadır.[57]



Şekil 5.8. S-kutusunun Ayrıntıları (S_1'in 0. satırı)

5.2.4. Anahtarın Oluşturulması

Şekil 5.3'e tekrar bakarsak, algoritmaya girdi olarak kullanılacak 56-bitlik anahtar öncelikle "Seçilmiş Permütasyon Bir" isimli Tablo 5.2'de tanımlanmış permütasyona tabi tutulur. Buradan elde edilen 56-bitlik anahtar iki tane 28-bitlik parça olarak ayrılır ve C_0, D_0 olarak adlandırılır. Her bir fonksiyon tekrarında, C ve D ayrık olarak Tablo 5.2(c)'den faydalanılarak döngülü sola kaydırma veya çevrim işlemine tabi tutulur ve 1 veya 2 bit kaydırılır.

Sütun Numarası

Satır	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Kutu
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7	
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8	S_1
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0	
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10	
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5	S_2
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15	
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9	
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8	
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1	S_3
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7	
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12	
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15	
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9	S_4
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4	
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14	
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9	
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6	S_5
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14	
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3	
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11	
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8	S_6
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6	
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13	
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1	
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6	S_7
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2	
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12	
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7	
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2	S_8
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8	
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11	

Şekil 5.9 DES'deki S Kutularının Tanımları[53],[54]

Bu kaydırılmış değerler bir sonraki tur için girdi oluşturur. Bunlar ayrıca, "Seçilmiş Permütasyon İki" (Tablo 5.2) için de girdi olurlar. Bu ikinci permütasyon sonucu $f(R_{i-1}, Ki)$ Fonksiyonuna girdi olarak alınarak 48-bitlik bir çıktı oluşturulur.

5.2.5. DES Deşifrelemesi

DES' in deşifre edilme işlemi, özünde şifreleme işlemi ile aynıdır. Kural şöyle ifade edilebilir: Şifreli metin, DES algoritmasına girdi olarak kullanılır, ancak K_i anahtarları ters sırada kullanılır. Yani ilk turda K_{16} , ikinci turda K_{15} kullanılır. Son tura gelindiğinde ise kullanılacak olan anahtar, K_1 olur.[53]

Anahtarların ters sırada kullanılmasıyla aynı algoritmanın doğru sonucu verdiğini göstermek açısından; solda yukarıdan-aşağıya şifreleme işleminin, sağda ise aşağıdan-yukarıya doğru deşifreleme işleminin yer aldığı Şekil 5.5'i inceleyelim.

Tablo 5.3 DES Anahtarı Hesaplanmasında Kullanılan Tablolar[54]

Seçilmiş Permütasyon Bir (PC-1)

Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Girdi bitinden	57	49	41	33	25	17	9	1	58	50	42	34	26	18
Çıktı biti	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Girdi bitinden	10	2	59	51	43	35	27	19	11	3	60	52	44	36
Çıktı biti	29	30	31	32	33	34	35	36	37	38	39	40	41	42
Girdi bitinden	63	55	47	39	31	23	15	7	62	54	46	38	30	22
Çıktı biti	43	44	45	46	47	48	49	50	51	52	53	54	55	56
Girdi bitinden	14	6	61	53	45	37	29	21	13	5	28	20	21	4

Seçilmiş Permütasyon İki (PC-2)

Çıktı biti	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Girdi bitinden	14	17	11	24	1	5	3	28	15	6	21	10	23	19	12	4
Çıktı biti	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Girdi bitinden	26	8	16	7	27	20	13	2	41	52	31	37	47	55	30	40
Çıktı biti	33	34	35	36	37	8	39	40	41	42	43	44	45	46	47	48
Girdi bitinden	51	45	33	48	44	49	39	56	34	53	46	42	50	36	29	32

Sola Kaydırma Tablosu

Döngü numarası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Çevrilen bitler	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tablo 5.3 gösteriyor ki, her bir seviyede deşifrelemenin orta seviye değeri, şifrelemenin ilgili değerinin iki yarısının yer değiştirilmiş haline eşittir. Bunu bir başka şekilde göstermek istersek, i . şifrelemenin sonucunun $L_i \,||\, R_i$ (L_i ve R_i'nin oluşturduğu değer) olduğunu düşünelim. O zaman, (16 – i) . deşifreleme seviyesinin ilgili girdi değeri $R_i \,||\, L_i$ olur.

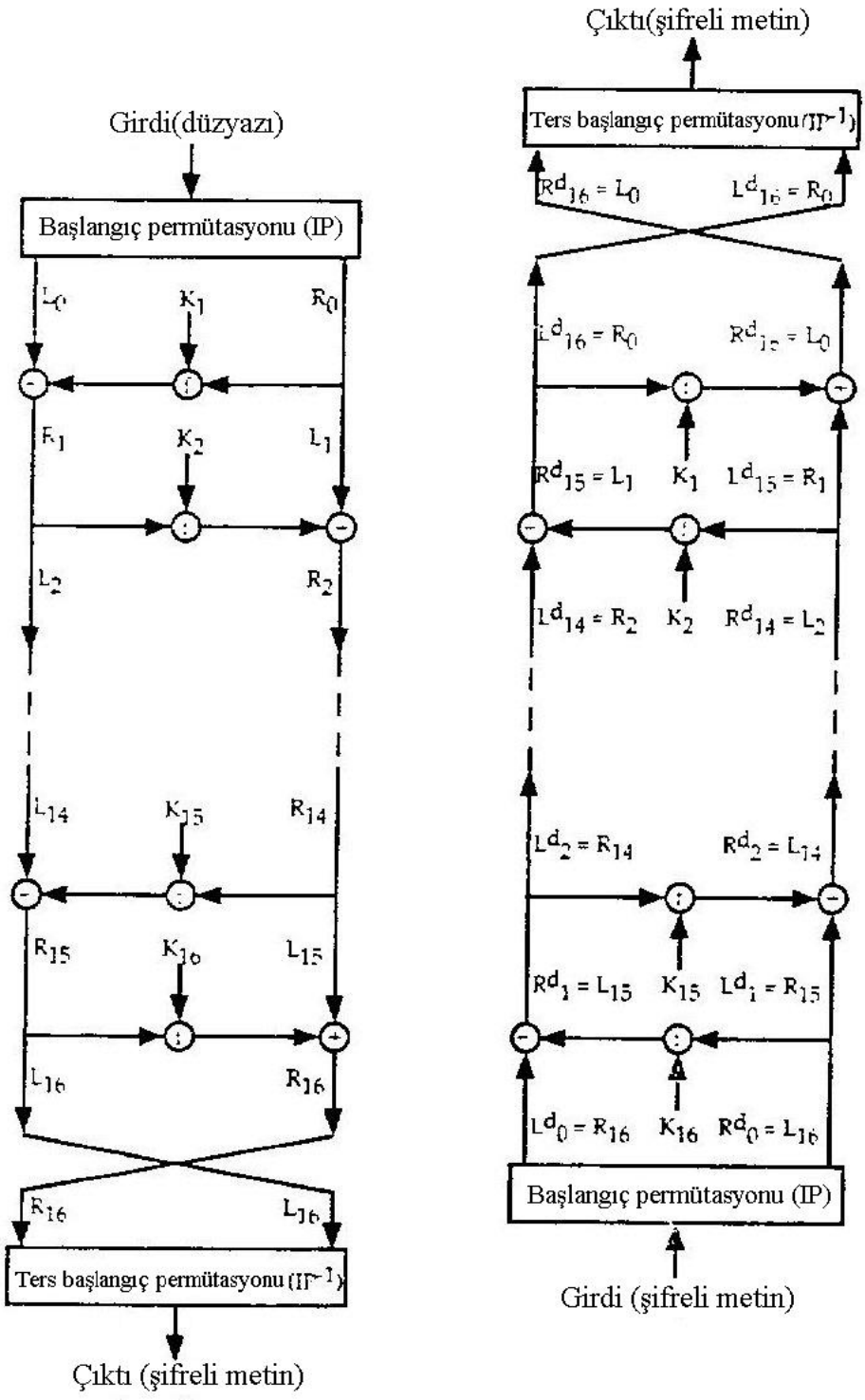
Bu saptamaların doğruluğunu göstermek için Şekil 5.9'yi incelendiğinde. şifreleme işleminin son basamağından sonra oluşan değerin iki yarısı yer değiştirilir öyle ki son IP^{-1} seviyesine girdi olarak $R_{16} \,||\, L_{16}$ alınır. Bu adımın sonucu şifreli metindir. Şimdi şifreli metni DES algoritmasına girdi olarak alalım. İlk adım 64-bitlik bir değer olan $L^d_0 \,||\, R^d_0$ 'ı üretmek için kullanılan IP'den (initial permütation) geçirmektir. Fakat biz zaten biliyoruz ki IP,IP^{-1}'in ters işlemidir. Bundan dolayı:

$$L^d_0 \,||\, R^d_0 = IP \text{ (şifreli metin)}$$

$$\text{Şifreli metin} = IP^{-1} (R_{16} \,||\, L_{16})$$

$$L^d_0 \,||\, R^d_0 = IP\,(IP^{-1} (R_{16} \,||\, L_{16})) = R_{16} \,||\, L_{16}$$

Yani deşifreleme işleminin ilk adımına girdi olarak kullanılan değer, şifreleme işleminin 16 . adımında üretilen 32-bitlik yer değiştirmiş halidir.



Şekil 5.9 DES ile Şifreleme ve Deşifreleme[54]

Şimdi de deşifreleme işleminin ilk adımının çıktısının, şifreleme işleminin 16. adımına girdi olan değerin 32-bitlik yer değiştirmiş (takas edilmiş) hali olduğunu gösterelim. İlk olarak şifreleme işlemi düşünülürse:

$$L_{16} = R_{15}$$
$$R_{16} = L_{15} \oplus f(R_{15} , K_{16})$$

olduğunu görülür.

Diğer taraftan deşifreleme işleminde:

$$L^d_1 = R^d_0 = L_{16} = R_{15}$$
$$R^d_1 = L^d_0 \oplus f(R^d_0 , K_{16})$$
$$= R_{16} \oplus f(R_{15} , K_{16})$$
$$= [L_{15} \oplus f(R_{15} , K_{16})] \oplus f(R_{15} , K_{16})$$

eşitlikleri yazılabilir.

XOR, şu özelliklere sahiptir:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

Yani, $L^d_1 = R_{15}$ ve $R^d_1 = L_{15}$ eşitlikleri geçerlidir. Bundan dolayı, deşifreleme işleminin ilk adımının çıktısı, şifreleme işleminde 32-bit takas edilerek, 16 . adımda girdi olarak alınan $L_{15} \| R_{15}$ 'dir. Bu benzerlik, her 16 adım için de geçerlidir ve şekilde gösterilmiştir. Bu işlemi genel bir eşitlik olarak yazarsak, şifreleme algoritmasının i . adımı için :

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \oplus f(R_{i-1} , K_i)$$

Eşitliği düzenlenirse:

$$R_{i-1} = L_i$$
$$L_{i-1} = R_i \oplus f(R_{i-1} , K_i) = R_i \oplus f(L_i , K_i)$$

olur.

Bununla i . adımın girdilerini, çıktıların bir fonksiyonu olarak tanımlanır ve bu eşitlikler Şekil 5.9'un sağ tarafındaki atamaların doğruluğunu göstermektedir..

Son olarak, deşifreleme işleminin son adımının çıktısının $R_0 \parallel L_0$ olduğu görülmektedir. Bu, 32-bitlik bir takasa tabi tutulur (her iki yarısı yer değiştirir) ve IP^{-1} adımına $L_0 \parallel R_0$ olarak girer. Ancak,

$$IP^{-1}(L_0 \parallel R_0) = IP^{-1}(IP(\text{düzyazı})) = \text{düzyazı}$$

olduğundan orijinal düzyazı elde edilir ve DES deşifreleme işleminin geçerliliği ispatlanmış olur.

5.2.6. 56-Bitlik Anahtarları Kullanılması

56-bit uzunluğunda bir anahtar kullanılarak 2^{56} olası anahtar (yaklaşık 7.2×10^{16}) ortaya çıkabilir. Bu sebepten, bir kaba-kuvvet saldırısı pratikte kullanışlı değildir. Anahtarların yarısının deneneceği düşünülürse, bir DES şifrelemesini bir mikrosaniye de yapan tek bir makinenin şifreyi çözmesi 1,000 yıldan fazla sürer (Tablo 5.2).

Bununla birlikte, bir mikrosaniye de bir şifreleme yapılması kabulü pek doğru değildir. 1977 yılı itibariyle, Diffie ve Hellman teknolojinin gelişeceğini ve her birinin bir mikrosaniye de bir şifreleme yapabilen ve 1 milyon şifreleme cihazı içeren bir paralel makinenin inşa edilebileceğini düşünmüştür. Maliyetinin ise 1977'de, 20 milyon dolar civarında olacağını tahmin etmişlerdir.[53][54]

Problemin en güncel analizi Wiener tarafından "bilinen düzyazı saldırısı" temel alınarak yapılmıştır. Bu analizde, saldırganın en azından birer tane (düzyazı,şifreli metin) çiftine sahip olduğu kabul edilir. Wiener, tasarımının ayrıntılarını şöyle ifade etmiştir:

Şimdiye kadar DES anahtarlarının kırılmasının hızlı bir yöntemini bulmak için doğrulanamayan bir çok iddia ortaya atılmıştır. Böylesi iddialara bir yenisini eklememek için, bir anahtar arama makinesinin ayrıntıları eklerde verilmiştir. Bu ayrıntılı araştırma daha çok, böyle bir makinenin maliyeti ve DES anahtarını kırmak için gereken süre hakkında bir yaklaşımda bulunmaktadır. Böyle bir makinenin inşası için, hiçbir plan mevcut değildir."[54],[57]

Wiener, saniyede 50 milyon anahtar deneyebilen bir çipin tasarımındaki hesaplamaları yapmıştır. 1993 fiyatlarına göre 5,760 çipten oluşan ve maliyeti 100,000$ olan bir modül tasarlanmıştır. Bu tasarımın sonuçları tablo 5.4 te verilmektedir.[57]

Bununla birlikte bir anahtar taramak için bilinen tek saldırı tüm olası anahtarları denemekle bitmez. Bilinen düzyazı mevcut değilse, analiz eden kişi düzyazıyı düzyazı olarak ayırt edebilmelidir. Eğer mesaj sadece düzyazı halinde İngilizce bir metin ise, İngilizce'yi tanıyabilen bir otomasyon yapılması gerekliliği ile birlikte sonuçlar oldukça olumlu bir yönde değişir. Eğer metin, şifrelenmeden önce sıkıştırılmışsa, tanınması çok daha zor olur. Hatta mesaj, bir metin değil de, nümerik bir dosya veya genel türde bir veri dosyası ise ve üstüne üstlük sıkıştırılmışsa otomasyon iyice zorlaşır. Bu sebeplerden, kaba-kuvvet saldırısı yaklaşımının gerçeklenmesi için, beklenen düzyazı hakkında bir parça ön bilgiye ihtiyaç duyulacaktır. Wiener'in tasarımı, DES' in güvenilirliği konusundaki yıllar boyu sürecek olan tartışmaların bir dönüş noktası ve belki de erken bir sonucudur. Bu yazı yazıldığı zamanlarda bile, kişisel ve ticari alandaki uygulamalarda DES'e güvenmek hala mantıklı gözükmektedir. Ancak, geleneksel şifrelemeye alternatif olacak çözümler arama zamanı gelmiştir. DES'in yerini alabilecek iki ümit verici aday, Üçlü(triple) DES ve IDEA olarak gösterilebilir.

Tablo 5.4 Eksiksiz Anahtar Tarama İçin Gereken Zaman **[53]**

Anahtar Büyüklüğü	Alternatif Anahtar sayısı	mikrosaniye de bir şifreleme	mikrosaniye de 10^6 şifreleme
32 bit	$2^{32} = 4.3 \times 10^9$	$2^{31}\mu s = 35.8$ dakika	2.15 ms
56 bit	$2^{56} = 7.2 \times 10^{16}$	$2^{55}\mu s = 1142$ yıl	10.01 saat
128 bit	$2^{128} = 3.4 \times 10^{38}$	$2^{127}\mu s = 5.4 \times 10^{24}$ yıl	5.4×10^{18} yıl
26 karakter	$26! = 4.03 \times 10^{26}$	$2 \times 10^{26}\ \mu s = 6.4 \times 10^{12}$ yıl	6.4×10^6 yıl

5.2.7. DES Operasyon Modları

DES algoritması, veri güvenliği sağlamada kullanılan temel bir yapıdır. Ancak DES'i çeşitli uygulamalarda kullanmak üzere dört farklı işlem modu tanımlanmıştır Bu dört mod DES'in kullanılabileceği her türlü uygulamayı sanal olarak içermektedir. Bu modlar Tablo 5.5'da özetlenmiştir. [1], [3], [53]

Tablo 5.5 DES İşlem Modları[53]

Mod	Açıklama	Tipik Uygulamaları
Elektronik Kod Kitabı(EBC)	64-bitlik düzyazı blokları ayn anahtar kullanılarak bağımsız olarak şifrelenir	Tek değerin güvenli olarak iletilmesi(örneğin şifreleme anahtarı)
Şifreli Blok Zincirleme(CBC)	Şifreleme algoritmasının girdis sonraki 64-bitlik düzyazı ile önceki 64-bitlik şifreli metnin XOR lanması ile elde edilen değerdir	Genel amaçlı blok odaklı iletim Doğrulama
Şifre Geri Besleme(CFB)	Girdi,bir seferde J tane bit alınarak işlenir.Önceki şifreli metin,girdi olarak alınır ve geçici rasgele çık-tı üretilir. Bu da düzyazıyla XOR lanarak sonraki şifreli metin birimi üretilir.	Genel amaçlı akış odaklı iletim Doğrulama
Çıktı Geri Besleme(OFB)	Şifreleme algoritmasına girdi olarak önceki DES çıktısını alması dışında CFB ile aynı.	Gürültülü kanal boyunca akış odaklı iletim(ör neğin,uydu haberleşmesi)

5.2.7.1. Elektronik Kod Kitabı Modu (ECB)

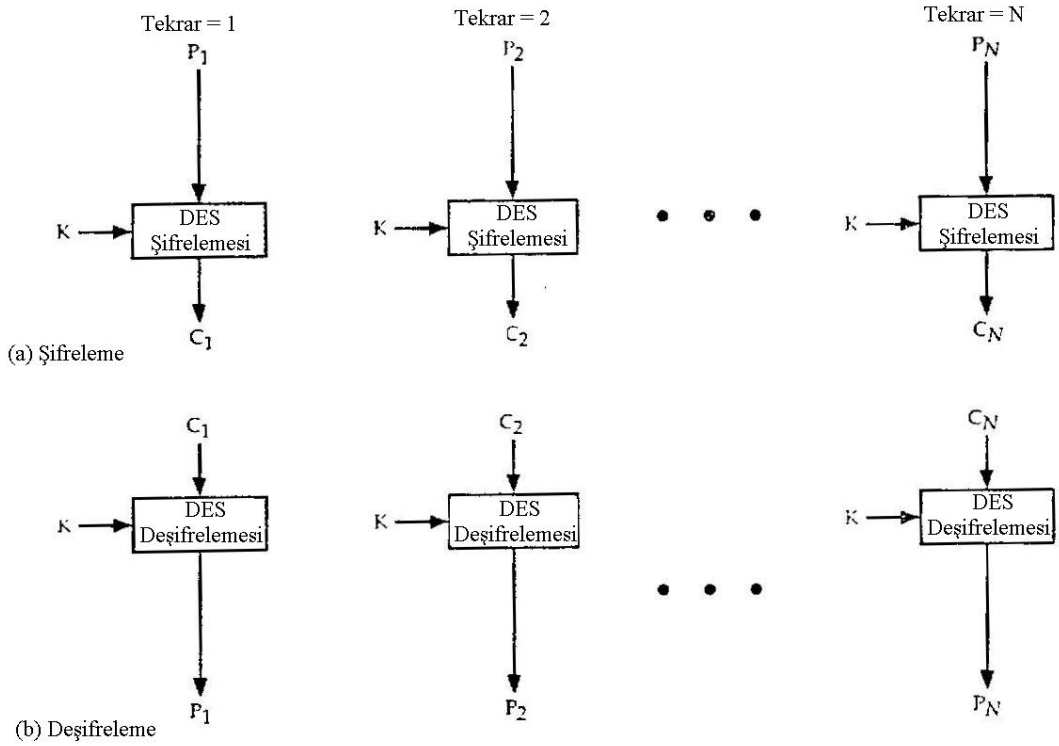
Düzyazının 64 bit parçalar halinde işleme alındığı ve her birinin aynı anahtarla şifrelendiği bu mod, en basit DES modudur. "Kod Kitabı" ifadesinin kullanılmasının nedeni, verilen bir anahtara karşılık her bir 64-bitlik düzyazı bloğu için tek bir şifreli metnin var olmasıdır. Yani her olası 64-bitlik düzyazı kalıbına karşılık bir kayıt barındıran devasa bir kod kitabı hayal edilebilir.

64 bitten uzun bir mesaj için yapılan işlem mesajı 64-bitlik bloklara ayırmak ve gerekirse son bloğu bölmektir. Deşifreleme işlemi, bir seferde bir blok alınarak ve hep aynı anahtar kullanılarak yapılır. Şekil5.10'da düzyazı 64-bitlik blokların dizilişinden (P_1, P_2,..... P_n) oluşur. İlgili şifreli metin blokları da (C_1, C_2,....., C_n) aynı şekilde dizilmiştir.

ECB metodu, az sayıdaki veriler için idealdir. Örneğin, şifreleme anahtarı gibi. Demek ki bir DES anahtarını güvenli bir şekilde iletmek istersek, ECB kullanılacak en iyi moddur.

ECB modunun en göze çarpan özelliği aynı 64-bitlik düzyazı bloğunun birden fazla defa mesajda yer almasıyla, şifreli metinde de aynı şekilde ortaya çıkacağıdır.

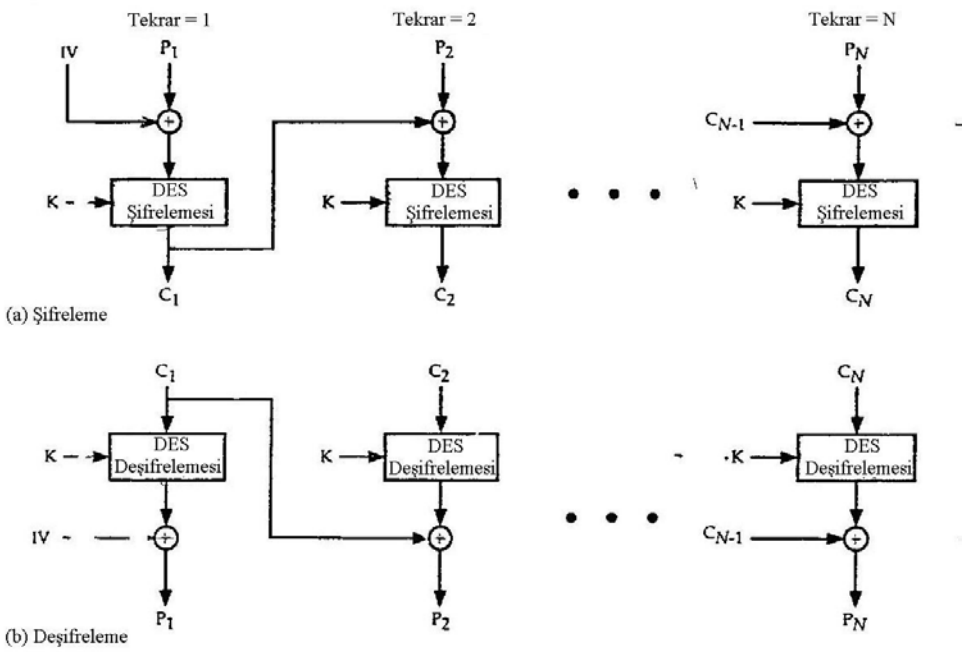
Uzun mesajlar için ECB modu güvenli olmayabilir. Eğer mesaj yüksek derecede yapısalsa, bir şifre analizcinin bu düzenlilikleri ortaya çıkarması olasıdır. Örneğin, eğer mesajın hep aynı önceden belirlenmiş kalıplarla başladığı biliniyorsa o zaman şifre analizci üzerinde çalışabileceği bir çok düzyazı/şifreli metin çiftine sahip olabilir. Bu da ECB modunu, uzun mesajlar için güvensiz yapar.



Şekil 5.10 Elektronik Kod Kitabı (ECB) Modu[53]

5.2.7.2. Şifreli Blok Zincirleme Modu (CBC)

ECB'deki güvenlik eksiklerini giderebilmek açısından, aynı düzyazı bloklarına karşılık farklı şifreli metin blokları üretebilen bir teknik gerekmektedir. Bunu sağlayan en basit yollardan biri CBC modudur (Şekil 5.11). Bu teknikte, şifreleme algoritmasının girdisini,o an ki düzyazı bloğu ile önceki şifreli metin bloğunun özelveya'lanması oluşturur. Her bir blok için aynı anahtar kullanılır. Ardışık düzyazı blokları sanki birbirine zincirlenmiş gibidir. Şifreleme fonksiyonunun girdisi her bir düzyazı bloğu için, düzyazı bloğuyla ilişkili değildir. Bu sayede, 64 bitlik tekrarlı kalıplar kendini ele vermez.



Şekil 5.11 Şifreli Blok Zincirleme (CBC) Modu [53]

Deşifre etmek için, her bir şifreli blok deşifreleme algoritmasından geçirilir. Sonuç, önceki şifreli metin bloğu ile XOR lanır ve düzyazı bloğunu oluşturur. Bunun doğruluğunu görmek için şunu yazabiliriz:

$$C_n = E_k[C_{n-1} \oplus P_n]$$

O zaman;

$$D_k[C_n] = D_k[E_k(C_{n-1} \oplus P_n)]$$

$$D_k[C_n] = C_{n-1} \oplus P_n$$

$$C_{n-1} \oplus D_k[C_n] = C_{n-1} \oplus C_{n-1} \oplus P_n = P_n$$

İlk şifreli metin bloğunu üretmesi için, bir başlama vektörü (IV) , düzyazının ilk bloğu ile XORlanır. Deşifreleme de; IV, deşifreleme algoritmasının çıktısı ile XORlanır ve düzyazının ilk bloğu geri alınır.

IV vektörü, hem gönderen hem de alıcı tarafından bilinmelidir. Üst düzey güvenlik için, IV vektörü en az anahtar kadar iyi korunmalıdır. Bu, IV vektörünü ECB şifrelemesini kullanarak göndermekle sağlanabilir. IV vektörünün korunmasının bir nedeni şudur: Eğer bir rakip, alıcıyı kandırarak farklı bir IV değeri kullandırırsa o zaman bu rakip düzyazının ilk bloğundaki seçtiği bitleri çevirebilir. Bunu görebilmek için, şu düşünülebilir:

$$C_1 = E_k(IV \oplus P_1)$$
$$P_1 = IV \oplus Dk(C_1)$$

Şimdi X[i] gösterilişinin 64-bitlik X'in i . bitini gösterdiğini düşünelim. O halde,

$$P_1[i] = IV[i] \oplus Dk(C_1)[i]$$

O zaman, XORun özelliklerini kullanarak denilebilir ki;

$$P_1[i]' = IV[i]' \oplus Dk(C_1)[i]$$

Bu demektir ki eğer bir rakip IV içindeki bitleri bilerek değiştirebilirse alınan P_1 değerinin ilgili bitleri değiştirilebilir.

Sonuç olarak CBC'nin zincirleme mekanizmasından ötürü, 64 bitten uzun olan mesajları şifrelemede kullanılabilecek bir moddur.

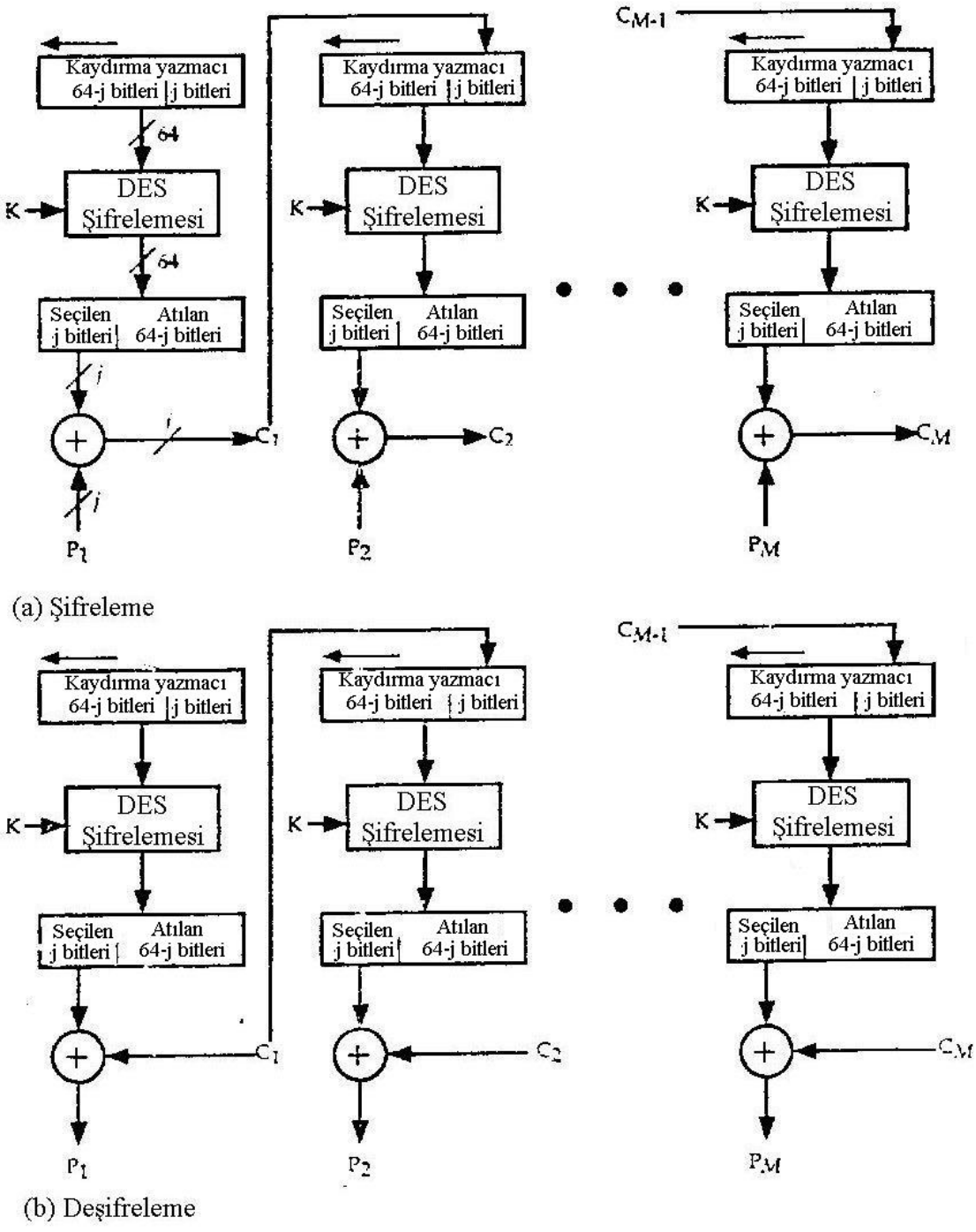
5.2.7.3. Şifre Geri Besleme Modu (CFB)

DES şifrelemesi özünde 64-bitlik blokları kullanan bir blok şifreleme tekniğidir. Bununla birlikte, DES'i bir sürekli şifrelemeye dönüştürebiliriz. Bu iki şekilde mümkündür: şifreli geri besleme ve çıktı geri beslemeli modlar. Bir sürekli şifreleme,

mesajı belli bir sayıda bloğun katları uzunlukta olması için düzenleme gerekliliğini ortadan kaldırır. Ayrıca gerçek zamanlı olarak işlem yapılabilir. Yani, eğer bir karakter şeridi yollanıyorsa, her bir karakter anında karakter temelli bir şifreleme ile şifrelenerek yollanabilir.[1],[3],[53]

Bir süreğen şifrelemenin ihtiyacı olan bir özellik şifreli metnin, düzyazıyla aynı uzunlukta olmasıdır. O zaman 8-bitlik karakterler gönderiliyorsa her bir karakter 8 bit kullanılarak şifrelenmelidir. Eğer 8 bitten fazlası kullanılırsa, gönderme kapasitesi ziyan edilmiş olur. Şekil 5.12, CFB tekniğini şemasal olarak göstermektedir. Şekilde, gönderilen birim verinin j = 1 olduğu kabul edilmiştir; ortak bir değer olarak j = 8 dir. CBC'deki gibi düzyazının birim parçaları birbirine zincirlenir öyle ki herhangi bir düzyazı biriminin şifreli metni o ana kadar olan tüm düzyazının bir fonksiyonudur.

Öncelikle şifreleme işlemini düşünelim. Şifreleme fonksiyonunun girdisi, bir başlama vektörüne(IV) atanan 64-bitlik kaydırılmış bir yazmaçtır. Çıktının en değerli j bitleri, düzyazı olan P_1'in ilk birimiyle XORlanır ve şifreli metin C_1'in ilk birimini üretir. Bu da, iletilir. Buna ek olarak, kaydırma yazmacının içeriği j bitleri ile sola kaydırılır ve C_1, kaydırma yazmacının en değersiz j bitlerine yerleştirilir. Bu işlem, tüm düzyazı birimleri şifreleninceye kadar sürdürülür.



Şekil 5.12 J-Bitlik Şifre Geri Besleme (CFB) Modu[53]

Deşifre etmek için aynı teknik kullanılır ancak farklı olarak; alınan şifreli metin birimi, şifreleme fonksiyonunun çıktısı ile XORlanarak düzyazı birimi üretilir. İlginç bir olay, deşifreleme değil de şifreleme fonksiyonunun kullanılmasıdır. Bu, kolayca açıklanabilir. X'in en değerli j bitlerini $S_j(X)$ olarak gösterirsek:

$$C_1 = P_1 \oplus S_j E(IV)$$

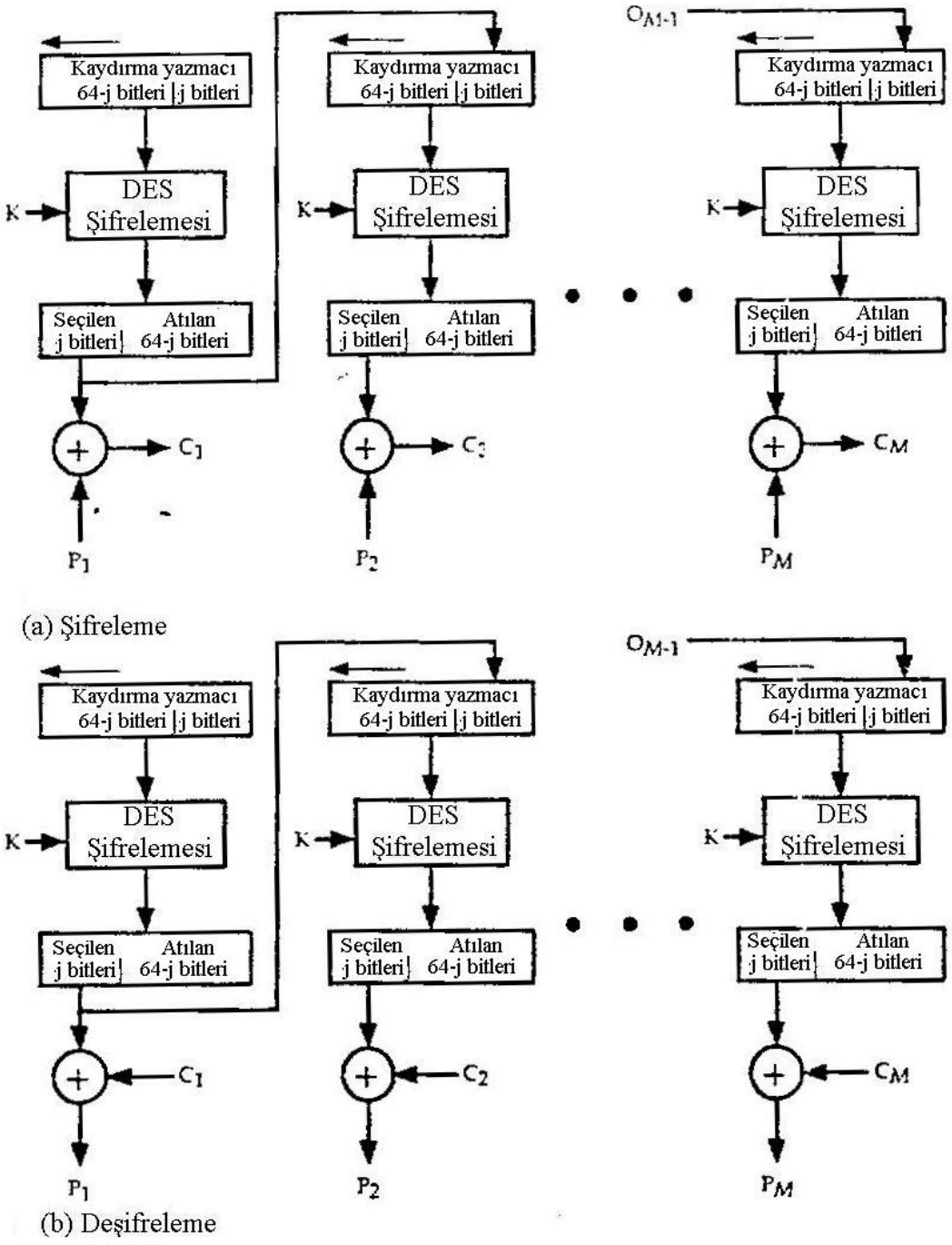
O halde,

$$P_1 = C_1 \oplus S_j E(IV)$$

Aynı mantık işlemin takip eden adımları için de geçerlidir.

5.2.7.4. Çıktı Geri Beslemeli Mod (OFB)

Çıktı geri beslemeli mod yapı olarak CFB'ye benzerdir ve şekil 5.13 'de gösterilmiştir. Görüldüğü üzere, OFB'deki kaydırma yazmacına geri beslenen değer şifreleme fonksiyonunun çıktısı iken, CFB'de şifreli metin birimi, kaydırma yazmacına geri besleme yapıyordu.



Şekil 5.13 J-bitlik Çıktı Geri Beslemeli (OFB) Mod[53]

CFB metodunun bir avantajı, iletilmiş bit hatalarının büyümemesidir. Örneğin eğer C_1'de bir bit hatası olursa sadece geri kazanılan P_1 değeri bundan etkilenecek ve

geri kalan düzyazı birimleri bozulmayacaktır. CFB ile C_1 aynı zamanda kaydırma yazmacına girdi olarak da hizmet vererek ek bozulmaları da önlemiş olur.

OFB'nin dezavantajı ise bir mesaj-akış modifiyesi saldırısına CFB'den daha az dayanıklı olmasıdır. Düşünün ki şifreli metindeki bir biti değiştirmek geri kazanılan düzyazıdaki ilgili biti de değiştirmektedir. O halde, geri kazanılan düzyazı içerisinde kontrollü değişiklikler yapılabilir. Bu bir rakibin mesajdaki önemli bir kısımda gerekli değişiklikleri yaparak, şifreli metinde herhangi bir hata düzeltici kodun algılayamayacağı değişiklikleri yapmasına izin verir.

5.2.8. DES'in S-kutularına yaklaşımları

S-kutularının sahip olduğu gösterilen karakteristiklerinin rastgele (random) seçilmediği artık kabul edilmektedir. Helman ve arkadaşları DES'in S-kutuları üzerine araştırmalar yapmışlardır ve bazı şaşırtıcı sonuçlara varmışlardır. Bu sonuçlar ayrıca Standford Üniversitesinde bir rapor olarak yayınlanmıştır. Bulunan çoğu tuzaklar S4 kutusunu ilgilendirmektedir. S4 kutusunun %75'inin işe yaramaz (redundant) olduğu konusunda bazı duyumlar vardır. DES'in S-kutularındaki eylem 4 permütasyonla ifade edilir.; bir permütasyon S-kutusunun iki giriş bitinin değerine göre seçilir. S-kutusunun 4 çıkış biti, S-kutusunun iki giriş bitinin değerine göre seçilir. S-kutusunun 4 çıkış biti, S-kutusunun geri kalan girişlerinin üzerinde bu özel permütasyon hareketi ile seçilir.

Hellman ve arkadaşları, S4 kutusunun tanımında kullanılan permütasyonlardan üçünün, ilki tarafından kolaylıkla ifade edilebileceğini ortaya çıkarmışlardır. Diğer bir ilginç sonuç S-kutularının affine yaklaşıklarını içermektedir.[59]

Bir affine dönüşüm bir lineer dönüşümden daha karmaşık olacaktır. Fakat bu karmaşıklık, bir kriptoanalistin ilgisini çekmeye yetecek kadar basittir. S4'ün affine yaklaşığı kötü değildir.

Keşfedilen diğer özellikler, S-kutusunun giriş bitlerinden biri ters çevrildiğinde en az iki çıkış bitinin ters çevrilmesidir. Bu özellik diferansiyel kriptoanalize karşı alınmış bir önlem olarak bilinmektedir.

Diğer bir aktif araştırma alanı ise (bugün de devam eden) S-kutularını tasarım kriterlerine erişmekle kolayca şifre çözmeye izin veren ve oldukça büyük bir ustalıkla gizlenmiş bazı tuzakların DES'de olma olasılıklarının araştırılmasıdır.

Shamir , S-kutularını az çok dengesiz olduğunu göstermiştir. Fakat aynı zamanda da böyle bir özelliğin kripto analitik saldırıya yada bir tuzağın gerçeklenmesine nasıl önayak olabileceğinin açık olmadığını da belirtmiştir.[1], [59]

Yıllardır biriken olaylardan anlaşılmaktadır ki çeşitli kritik kararlar ya S-kutularının tasarımı sırasında yada NSA tarafından gözden geçirilirken alınmıştır.

Hellman ve arkadaşlarının ulaştığı sonuç aşağıdaki paragrafla başlar;
"DES'te bilinen tipte saldırılara karşı sistemi güçlendirmek için yerleştirilmiş yapılar bulundu. Ayrıca sistemin zayıflığını gösteren yapılar da bulundu."[58]

Brickell, Moore ve Purtill daha sonra açıkça bilinen tasarım kriterlerine göre S-kutularını üreterek, DES içindeki S-kutularının görünen karakteristiklerinin, tasarım kriterlerinin seçimine bağlı olabileceğini belirttiler.[59]

DES için yapılan çalışmakar 1980'lere kadar devam etti. Davio ve arkadaşları DES'in kriptoanalizini deneyen farklı yaklaşımları gözden geçiren bir makale yayınladılar. Bu metodlar S-kutusu zayıflıklarını ele almaktadır. DES için eşdeğer formüller bularak ve hatta S-kutularının eylemi için resmi bir ifade bularak, 16 çevrimli DES boolean ifadelerin bir kümesinin analizine indirgemiştir. Bu ilk önce Hellman ve arkadaşları tarafından desteklenmiş sonra Schaumller-Bichl tarafından da praik olmadığı gerekesi ile reddedilmiştir.[58]

DES bütün bu ilk saldırılara dayanabilmiştir. Bütün bunların sonucu olarak ilginçtir ki bir çok araştırıcı Lineer Kriptoanaliz ortaya atıldıktan sonra Lineer Kriptoanalizin DES için en iyi saldırı yöntemi olduğuna karar vermişlerdir. Coppersmith'in raporunda DES'in tasarımcılarının diferansiyel kriptoanalize karşı uyanık olduklarını belirtmiştir. DES tasarımcılarının bu adımları diferansiyel bir saldırıyı geciktirmiştir. Bu adımlardan bazıları S-kutularını daha önce dikkate alınmayan özelliklerinden ve f fonksiyon çevriminde kullanılan permütasyondan kaynaklanmaktadır.[1],[3]

Lineer Kriptoanaliz yöntemi yaklaşık 2 yıl önce ilk 2^{43} açık mesaj –şifereli mesaj çifti biliniyorsa (aynı anahtar kullanılarak) 2^{30} anahtar uzayı taranarak DES algoritması kırılmıştır. Burada problem 2^{43} açık mesaj –şifreli mesaj çiftini (aynı anahtarla şifrelenmiş) elde edecek bilgisayarın bulunmasıdır. Bu yöntem den daha pratik bir yöntem ayrıntılı tarama (exhaustive search) dır. Çünkü bu yöntemde bir veya iki açık mesaj –şifreli mesaj çifti için 2^{56}'lık anahtar uzayını taramak yeterli olacaktır.

1976'da, National Security Agency (NSA) S-kutularının aşağıdaki özelliklerini tasarım ölçütleri olarak kabul etmiştir [58] ,[59]

- Her S kutusunun her bir satırı 0'dan 15'e kadar tamsayıların bir permütasyonudur.
- Hiçbir S-kutusu girişlerinin lineer yada affine bir fonksiyonu değildir.
- Bir S – kutusunun bir giriş bitinin değişmesi en az iki çıkış bitinin değişmesine neden olur.
- Herhangi bir S-kutusu için herhangi bir x girişi S(x) ve S(x + 001100) en az iki bitte farklıdır. (Burada x, 6 bit uzunluğunda bir bit dizisidir). S kutularının aşağıdaki iki özelliği (tasarım ölçütlerini yaratan) NSA tarafından tasarlanmıştır.
- Herhangi bir S-kutusu, herhangi bir x girişi ve e, f Є (0,1) için, $S(x) \neq S(x +$ 11ef00).
- Herhangi bir S kutusu için eğer herhangi bir giriş biti sabit tutulursa, 0'a eşit olan çıkış bitlerinin sayısı, 1'e eşit olan çıkış bitlerinin sayısına yakındır. Eğer birinci veya altıncı giriş biti sabit tutulursa 16 çıkış biti sıfır ve 16 tane çıkış biti bir olur. İkinci bitten beşinci bite kadar olan giriş bitleri için bu doğru değildir

fakat dağılım eşit olmaya (uniform) yakındır. Daha net söylemek gerekirse herhangi bir S-kutusu için eğer herhangi bir giriş biti sabit tutulursa, 13 ile 19 arasında çıkış sıfır /bir değerini alır. [57]

DES'in anahtar uzayı büyüklüğü 2^{56} dır. Bu güvenilirlik için çok çok küçüktür. Çeşitli özel makineler, bir bilinen açık mesaj (known plaintext) saldırısı için önerilmişlerdir. Bu makineler anahtar uzayında ayrıntılı tarama (exhustive search) gerçekleştirmek için yapılmışlardır. Yani verilen bir 64 bit açık mesaj X ve ona uygun şifreli mesaj Y için, her olası aday anahtar bir K anahtarı bulunana kadar test edilir. $E_k(X)=Y$ koşulunu sağlayan belki birden fazla K anahtarı bulunabilir.

1977'lerin başlarında Diffie ve Helmann, saniyede 10^6 anahtarı test edebilen bir VLSI chip yapılabilirse bütün anahtar uzayının 1 günde girilebileceğini ve böyle bir makinannın o zamanlar 20 milyon $'a inşaa edilebileceini öngörmüşlerdir.

CRYPTO'93 Ramp session'ında Micheal Wiener, bir anahtar tarama (key search) makinesinin ayrıntılı tasarımını vermiştir. İş hattı yapısına sahip makine, 16 şifrelemeyi eş zamanlı olarak yapabilmektedir. Bu chip saniyede $5*10^7$ anahtarı test edebilmekte ve bugünkü teknoloji ile chip başına maliyet 10.50$ olmaktadır. 5760 chip'i içeren bir çerçeve 100.000$'a inşaa edilebilir. Bu bir DES anahtarının ortalama 1.5 günde bulunmasını sağlar. 10 çerçeve kullanan bir makine 1 milyon $'a malolur fakat ortalama tarama zamanını üç buçuk saate indirmektedir.

Bu gün donanım (hardware) gerçeklemeleri oldukça hızlı şifreleme yapabilmektedirler. Digital Equipment Corporation, CRYPTO 92'de, 250 Mhz saat hızı (clock rate) ile saniyede 1 Gbit hızında 50K transistörle şifreleme yapılabildiğini belirtmştir. Bu chip'in maliyeti 300$'dır. 1991'de DES'in 45 tane donanım ve firmware gerçeklemesi yapılmıştır ve bütün bunlar National Bureau of Standards tarafından geçerli sayılmıştır.

Bankalar DES'in önemli uygulama alanlarından biridir (Amerikan Bankacılar Birliği tarafından geliştirilen standardlar kullanılarak). DES personel kimlik numarasını şifrelemek (PIN) için kullanılır ve ATM (Automated Teller Machines) de kullanılan

hesap işlerinde, hafta başına 1.5 $*10^{12}$ $'ın üzerinde olan doğrulama işlemleri (authenticate transactions) için Clearing House Interbank Payment System (CHIP) de ve aynı zamanda oldukça geniş bir şekilde devlet organizasyonlarında kullanılmaktadır (Amerika'da Enerji Departmanı, Adliye Departmanı ve Federal Depolama Sistemi gibi).[53], [58],[59]

5.3. AES - İLERİ ŞİFRELEME STANDARDI

5.3.1. AES'in Gelişim Süreci

DES blok şifreleme algoritması 1970'li yılların sonuna doğru çıkmış bir algoritmadır. Anahtar uzunluğu 56, blok uzunluğu 64 bittir. Günümüz teknolojisi ile 2^{56} (tüm olası anahtarlar) adet anahtarın denenmesi çok kısa bir zaman almaktadır. Bu nedenden dolayı, DES algoritmasının arka arkaya çalıştırılması anlamına gelen 3DES algoritması kullanılmaktadır. Bu yöntem ile algoritmanın anahtar uzunluğu 112 bite çıkmakla beraber blok uzunluğu 64 bitte kalmaktadır. Anahtar uzayının iyileştirilmesine karşın 3DES algoritması çok yavaş bir algoritmadır [54],[55]. Bu nedenlerden dolayı NIST (Ulusal Teknoloji ve Standartlar Enstitüsü) tarafından anahtar ve blok uzunluğu günümüz şartlarına uygun, değişken bir algoritma oluşturmak için 2 Ocak 1997'de 3 yıllık bir maraton olacak yarışmanın startını vermiştir. Bu yarışmaya dünyadaki kriptografi topluluğu üyelerinden 15 algoritma kabul edilmiş ve 1998 Ağustos ayında yapılan AES1 konferansında duyurulmuştur. 1999 yılında belirlenen, ana temaları güvenlik, maliyet, algoritma ve uygulama karakteristikleri olan kriterlere göre rastgelelik ve ansi c uyarlamalarının etkinlik testleri ve analizleri yapılmıştır. Analizlerin sonuçları AES2 konferansında değerlendirilerek 5 algoritma (MARS, RC6, Rijndael, Serpent, Twofish) finalist olarak seçilmiştir.[60] Bu 5 algoritma üzerinde yine belirlenen kriterlere bağlı olarak testler yapılmış, bunların sonuçları 2000 Nisan ayında yapılan AES3 konferansında tartışılmıştır. NIST tüm bu değerlendirmeler ve sonuçlar üzerinde çalışarak 2 Kasım 2000 tarihinde Rijndael'in AES için seçildiğini duyurmuştur. 26 Aralık 2001 tarihinde FIBS PUB 197 numarasıyla Rijndael algoritmasını temel alan İleri Şifreleme Standardı (Advanced Encryption Standard) yayınlanmıştır.[55],[61]

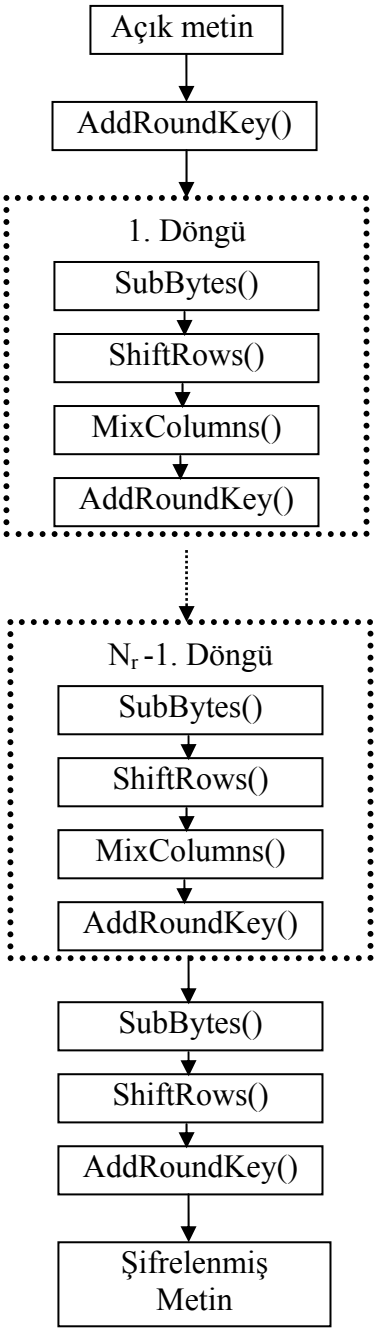
5.3.2. AES Algoritması

AES (Rijndael) algoritması değişik blok boyunu ve değişik anahtar büyüklüğünü destekleyen bir simetrik blok şifreleme algoritmasıdır. AES 128, 192 ve 256 bit uzunlukta anahtar boyutunu ve blok uzunluğunu desteklemektedir. Ancak kriptanaliz çalışmalarının bir alanda yoğunlaşmasını sağlayabilmek amacı ile 128 bit dışındaki blok boyları AES standardına dahil edilmemiştir.[61] Standart 128 bit blok boyunu ve 128, 192, 256 bit anahtar uzunluğunu içermektedir.

AES'te döngü sayısı anahtar uzunluğuna göre değişmektedir. 128 bit anahtar için 10 döngüde şifreleme yaparken 192 ve 256 bit anahtar için sırasıyla 12 ve 14 döngüde şifreleme yapmaktadır. AES algoritmasında her döngü dört katmandan oluşur.

Bu katmanlar sırasıyla Subbytes(), Shiftrows(), MixColumns() ve AddRoundKey() işlemleridir. Her döngünün çıkışı bir sonrakinin giriş değeridir.

Şekil 5.14'deki N_r değeri döngü sayısını ifade eder. Buda 10,12,14 (128,192,256 bit anahtar için) değerlerinden biridir.



Şekil 5.14 AES Algoritması Ana Akış Şeması[62]

AES en küçük işlem birimi olarak baytları kabul eder. Şifrelenecek metin, şifrelenmiş metin ve anahtar bilgileri bayt dizileri olarak kabul edilirler. Bu diziler 4 satır ve N_b adet sütündan oluşur ve her bir hücre baytlık bilgi tutar. Metin 4 baytlık sütün vektörleri şeklinde, yani 128 bit için 4x4 (N_b= 4), 192 bit için 4x6 (N_b= 6) ve 256

bit içinde 4x8'lik (N_b= 8) matrislerle ifade edilir. 128 bitlik bir metin için aşağıdaki gibidir.[62]

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix}$$

5.3.2.1 Subbytes Fonksiyonu (Baytların yerdeğiştirmesi)

S kutusunun olduğu katmandır. Giriş matris bilgisini alıp, her bir baytı tanımlanmış bir S kutusundan geçirerek, bu baytın karşılığı olan S kutusu değeri ile değiştirir. Baytların yerdeğiştirilmesinde 16 bayt değerinin her biri 8 bit girişli ve 8 bit çıkışlı S kutusuna sokulur. S kutusu değerleri, Galois alanı'nda (Galoıs Field - GF) GF(2^8), 8 bitlik polinom ($m(x) = x^8 + x^4 + x^3 + x + 1$) için ters alındıktan sonra lineer bir dönüşüme sokularak elde edilmiştir. Matristeki her baytın tersi bulunur.

5.3.2.2 ShiftRows Fonksiyonu (Satırların ötelenmesi)

Bu fonksiyon matrisi alır ve son üç satırını belli değerlere göre dairesel olarak sola öteler. N_b = 4 olmak üzere öteleme miktarı 2. satır için1, 3. satır için 2, 4. satır için 3'tür.

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

a.

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

b.

Şekil 5.15 ShiftRows Örneği a. Öteleme Öncesi, b. Öteleme Sonrası[62]

Tablo 5.6 AES S Kutusu (Hexadecimal notasyonda xy byte için)[61]

		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

5.3.2.3 MixColumns Fonksiyonu (Sütunların karıştırılması)

Bu fonksiyon matris üzerinde sütün bazında çalışır. Her sütun $GF(2^8)$'de 4 terimli bir polinom olarak kabul edilerek sabit bir $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ polinomu ile çarpılır ve elde edilen polinom başlangıç sütununun yerine geçer. b(x) matristen bir sütun olmak üzere, $b'(x) = a(x) \oplus b(x)$ eşitliğinden b'(x) hesaplanarak b(x) sütununun yerini alır.

5.3.2.4 AddRoundKey Fonksiyonu (Anahtar ekleme)

Bu fonksiyonda, bir döngü anahtarı matrise XOR işlemiyle eklenir. Her döngü anahtarı N_b kelimeden oluşmaktadır. $N_b = 4$ için döngü anahtarının boyu 128 bittir.

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \oplus \begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} \\ k_{10} & k_{11} & k_{12} & k_{13} \\ k_{20} & k_{21} & k_{22} & k_{23} \\ k_{30} & k_{31} & k_{32} & k_{33} \end{bmatrix}$$

$$= \begin{bmatrix} a_{00} \oplus k_{00} & a_{01} \oplus k_{01} & a_{02} \oplus k_{02} & a_{03} \oplus k_{03} \\ a_{10} \oplus k_{10} & a_{11} \oplus k_{11} & a_{12} \oplus k_{12} & a_{13} \oplus k_{13} \\ a_{20} \oplus k_{20} & a_{21} \oplus k_{21} & a_{22} \oplus k_{22} & a_{23} \oplus k_{23} \\ a_{30} \oplus k_{30} & a_{31} \oplus k_{31} & a_{32} \oplus k_{32} & a_{33} \oplus k_{33} \end{bmatrix}$$

5.3.3. Şifre Çözme

Algoritmayı oluşturan fonksiyonların tersleri alınarak ve ters sırada işleme sokularak şifre çözme işlemi gerçekleşebilir. [61,[63].

5.3.4. Güvenlik

AES şifreleme algoritması, bilinen tüm saldırı tiplerine karşı güvenlidir. Tasarımında kullanılan S kutusu sonlu alanda ters alma işleminin kullanılması ile gerçekleştirilmiştir. Bu algoritmayı lineer ve diferansiyel saldırılara karşı güvenli kılarken, algoritmanın tasarımında kullanılan, MixColums (sütünları karıştırma) ikinci lineer dönüşüm, lineer ve diferansiyel saldırılarda az sayıda aktif S kutusunun işin içine girmesine mani olmaktadır. [62]

Algoritmada herhangi bir zayıf anahtar şimdiye kadar tespit edilememiştir. Bu yüzden anahtar seçimine dönük herhangi bir kısıtlama standartta tanımlanmamıştır.

AES (Rijndael) algoritması hem donanım hemde yazılım uygulamalarında iyi performans vermiştir. Anahtar hazırlama süresi yeterince kısadır. Düşük bellek gereksinimi sayesinde, kısıtlı belleğe sahip ortamlarda da uygulanabilir [64]

5.4. AES Finalistlerinin Karşılaştırılması

5.4.1. Beş AES Finalisti

5.4.1.1 Mars Şifreleme Algoritması

MARS , IBM tarafından geliştirilmiştir ve algoritma yapısı olarak DES'e hiçbir özelliği benzememektedir. MARS donanımla birleştiği noktada son derece akla uygun bir şekilde dizayn edilmiş ve de son derece hızlıdır. Bunun anlamı şudur ki: SMART kartlara uygulamak için MARS'ın 70 bin civarında gate'e ihtiyacı vardır. 200 Mhz İntel Pentium da, MARS'ın 65 Mbit/s tutacağını ve normal donanım ciplerini kapsayacağını,100 saniyede 8Gigabitle ile 393 bin gate tutacağını iddia edilir. MARS anahtarlarının 128-448 bit olması taraftarıdır.[1],[3],[65]

5.4.1.2 Serpent Şifreleme Algoritması

Cambridge Üniversitesi, Halfa-İsrail ve Bergen Üniversitesi-Norveç tarafından geliştirilen bir şifreleme algoritmasıdır. SERPENT temel olarak DES'e benzer, diğer algoritmalarla aynı tipte ve değişkenler temelde son derce iyi anlaşılır prensiplere sahiptir. 20 yıldan fazla bir çok analizlere direndi. SERPENT anahtar alanı 40-256 bit

olanağı sağlar. Şifreleme deşifreleme hızı MARS şifreleme algoritmasına göre daha yavaştır.[65]

5.4.1.3 RC6 Şifreleme Algoritması

RC6, RSA laboratuarlarında geliştirildi, onun ve diğer AES adaylarının arasındaki en büyük bir fark ki tablo arası onda kullanılmaz, önemlidir ki bu genişliği azaltır(SMART kartları için önemli bir faktör). Maalesef RC6, diğer algoritmalar ile karşılaştırıldığında en yavaş şifreleme algoritmasıdır. Çünkü bu algoritma temelde uygulanan basit donanım operasyonlarını kullanır, 100 saniyede 1Gigabit civarında yetenekli olması gerekir. Bu daha sonra yeterli iken çoğu güncel uygulamaların sorunları 10 yıl içinde ortaya çıkabilir. RC6 daima anahtar genişliğinin 256 bitten daha geniş olmasını istenir.[65]

5.4.1.4 Rijndael Şifreleme Algoritması

RİJNDAEL(AES), bir Alman bankası tarafından geliştirildi ve bir çok ATM'lerde kullanıldı, ve bu gösteriyor ki bir çok önyargıları bu önledi. RİJNDAEL'in anahtar genişliği 8 bit işlemciler için oldukça etkileyicidir, kod uzunluğu 1K'nın üzerindedir ve 256 bitlik anahtar kullanıldığında RAM'in gereksinimi 52 bytedir(daha küçük anahtarlar olmadıkça). Diğer özeliği Pentium 200 de hızlanmak için numaraları kendisi sağlar, ANSI C de 27Mbit/s ile 128bitlik anahtarı ve 19.8Mbit/s ile 256 bit anahtarı çalıştırır. C++'ı kullanmak numaraları %250 oranında arttırır, 70.5 Mbit/s ile 128 bit anahtarı ve 51.2 Mbit/s ile 256 bit anahtarı çalıştırır.[65], [66]

5.4.1.5 Twofısh Şifreleme Algoritması

TWOFISH, bir US Counterpane Systems şirketi tarafından geliştirilmiştir. 256 bit üzerindeki anahtar boyutlarını destekler, çok aşırı hızlıdır.[67]

5.4.2. Beş AES Finalistinin Performansının Karşılaştırılması

Şifreleme algoritmalarının dizayn edilmesin deki ana amaç güvenlik olmalıdır. Gerçek dünyada performansla uygulama maliyetleri birbiriyle alakalıdır. En önemli özellikleri algoritmaların, performanslar ve maliyetlerdir.[67]

Bu çalışmada güvenliği yok sayıp onun yerine finalistlerin çeşitli platformlardaki performanslarını ele alınacaktır. Bu platformlar; 32 bit işlemci,64 bit işlemci,8 bit akıllı kart işlemci ve donanım olacaktır.[65]

5.4.2.1. 32 Bit İşlemcide Performans Karşılaştırılması

Mars Şifreleme Algoritması

Mars geniş operatör sistemli, yönlendirme ve çarpma yapan 82'e 32 bitlik S-kutusu içeren hızlı bir yazılım sistemidir. Mars blok başına 390 saatle AES finalistleri içinde en hızlısı olarak düşünülür. Bu hızının sebebi değişik derleme sistemi kullanılmasıdır. Mars şifrelemesinin 3 basamağı vardır; giriş ileri karıştırma, Cryptographic öz ve son karıştırma.[65]

RC6 Şifreleme Algoritması

RC6 en iyi taarlanmış ve AES finalistlerinin kolayca anlaşılanıdır. AES'in hedef platformunda, Assembly dilinde blok başına 250 saatle en hızlı algoritmadır. RC6 birçok platformda istenilen sonucu vermemektedir. Genellikle parça uyuşmazlığı bulunmaktadır.

Rijndael Şifreleme Algoritması

Rijndael bir diğer kare varvasyonudur. Bütün platformlarda çok iyi çalışan hızlı bir yazılımdır(cipher). Açık matematiksel yapısı en büyük özelliğidir. Şifreleme ve şifre çözme algoritmaları tam olarak mantıklı olmasa da genel yapıları ve performansları tartışılmazdır.[65],[67]

Rijndael'in Assembly dili ile derlenmiş ve Pentium ve pentium Pro işlemcilerde blok başına 300 saat civarındadır. [67]

Serpent Şifreleme Algoritması

Serpent 32 bit işlemcilerde "bit-slice" uygulamalarına imkan sağlamak için dizayn edilmiştir. Assembly dilinde SERPENT, DES uygulamasından daha yavaştır(45 saat/byte).

Twofish Şifreleme Algoritması

Twofish Pentium da AES parçalarının en hızlısı ve pentium Pro 2 de en iyi ikinci hızlısıdır. Sadece basit RISK işlemlerinde kullanılır ve performansı diğer 32 bit platformlarla karşılaştırılabilir.

5.4.2.2. Smart Kart Performasının Algoritmalar Üzerindeki Yorumu

Mars Şifreleme Algoritması

Mars 2 KB ROM gerektirdiği için , bu durum S-kutusunda bazı sorunlara yol açar. Bu büyüklük tek başına diğer AES finalistlerinin toplamından büyüktür. Kod büyüklüğü hakkında bir tahmin yapılamaz, fakat genellikle 3 KB'nin altındadır.

Mars'ın üzerindeki fly-subkey jenerasyonu bulunmaz ve aynı zamanda RAM'de 160 Byte subkey gerektirir. 16 byte Plaintext ve 16 Byte anahtar eklendiğinde diğer değişkenlerde arttırılır.

Marsta bu kullanımların çarpımların kullanımı değişkenlerin rotasyona uğraması ve artışlar zamanlaşmış artıkları mümkün kılmaktadır.[67],[69],[70]

RC6 Şifreleme Algoritması

RC6'nın küçük kod boyutu SMART CARD kelimelerini taşımalıdır. 8 bit CPU'lardaki zayıf performansı küçük bir sorundur. Fakat CPU'ların çoğunda bir problem olmaz.

RC6 da fly-subkey jenerasyonu bulunmaz ve RAM de 176 Byte subkey gerektirir. 16 Byte Plaintext ve anahtar eklendiğinde diğer değişkenleri artırır.

Rıjndael Şifreleme Algoritması

Rijndael SMART CARD'ın içinde DR98b dikkate alınarak dizayn edilmiştir. Üzerindeki fly-subkey RAM'i minimum kullanılmasında izin verir. Eğer subkey'ın hesaplanması için 160 Byte bulunursa rijndael daha hızlı olabilmesi düşünülmektedir.

Serpent Şifreleme Algoritması

Serpnet açıkçası, SMART CARD'daki ABK98b dikkate alınarak dizayn edilmiştir. ROM ayak izi çok küçüktür.(1 KB'nin altında). Serpent üzerindeki fly-key programı çok küçük RAM kullanımına izin vermektedir.[70]

Twofısh Şifreleme Algoritması

TWOFISH SMART card'lar dikkate alınarak dizayn edilmiştir. Subkey'i fly da hesaplanabilirdi. Şifreleme ve deşifreleme RAM'in 60 Byte'ı kullanılarak yapılabilirdi. Şifreleme ve deşifreleme aynı hızdadır. Biraz daha RAM kullanılabilirse TWOFISH'in şifrelemesi ve deşifrelemesi daha hızlı olabilirdi.[65],[70]

5.4.2.3. Donanım Performanslarının Karşılaştırılması

AES finalistlerinin şifreleme ve deşifreleme işlemleri genel olarak hızlı görünür. AES finalistlerinin karşılaştırılması ise oldukça zordur. Birçok tahmin edici karışık farklı işlem teknolojisi, farklı ölçüm büyüklükleri ve dizayn metodları kullanılır.[65],[67] [69]

Mars Şifreleme Algoritması

MARS donanım uygulamasını kullanışsız yapar. Bir uygulama tam bir çarpıcı tam bir yönlendirici ve geniş bir S-kutusu ROM'u gerektirir. Blokların her biri makul performanslı, küçük donanım modülü oluşturmakla geniş ve sınırlıdır. Mars kağıdı diğer finalistlerden daha geniş olan 70000 "cell" büyüklüğünü kabul eder. MARS'ın donanımdaki anahtar çevikliği düşüktür.[65]

RC6 Şifreleme Algoritması

Bu algoritmanın basitliği donanım için çok uygundur. Donanım uygulamalarında diğer algoritmalara göre daha hızlıdır. Zaman çarpıcı ve yönlendirici kullanması veya tekli çarpıcı ve yönlendirici kullanması bu algoritmanın en büyük özelliğidir.[65],[67]

Rijndael Şifreleme Algoritması

RIJNDAEL donanımı çok iyi tamamlar ve işletir. RIJNDAEL'in donanımdaki en büyük ilişkisi tam paralel versiyonu 16-256 Byte ROM gerektirmesidir.. Bu yüksek performans dizaynı için makul değildir. Ama kapı aşama S-kutusu özellikle donanım büyüklüğünü kesecektir. RIJNDAEL'in düşük performanslı versiyonunu bir kaç tane S-kutusu ROM'la oluşturmak oldukça kolaydır. [65]

Donanımla deşifrelemeyi oluşturmak çip büyüklüğünü 2 katına çıkarmakla mümkündür.

Serpent Şifreleme Algoritması

SERPENT donanımı çok makul şekilde tamamlar. Değiş tokuş uygulamasını çok iyi oluşturur. Tam paralel versiyonu toplam 256 S-kutusu gerektirir. Devir fonksiyonu çok hızlı çalışır ve devir performans başına çok iyi sonuç verir. Deşifreleme çarpık permütasyon ve lineer değiştirme gerektirir. Donanımdaki anahtar çevikliği çok iyidir.

Twofish Şifreleme Algoritması

TWOFISH donanım performans uygulamalarının başından ilan edilmiştir. Algoritmayı yüksek hız ve alçak kapı sayma uygulamasındaki birçok uygun boşluk-zaman değiş tokuşu vardır. Anahtar çevikliği şifreleme ve deşifreleme için yüksektir. Çünkü anahtar programlaması fly'ın her yönünde çalışabilir. Bu özellik hiçbir finalistte yoktur.

5.4.2.4. Yazılım Performansı

32 bitlik CPU'daki verimlilik NİST'in ortaya koyduğu performans kritelerinden biridir. Modern CPU'nun mimarisi çok karmaşıktır. Bu yüzden karşılaştırma için kullanılır. Bugün en yüksek mikroişlemciler 32 bit mimari kullanmaktadır. Bu mikroişlemciler Intel-Pentium ailesinden , ARM ailesinin 32 bit SMART CARD'larına benzeyen CPU'lara kadar değişir. Bunların mimaride en etkili olması bizim için şaşırtıcı değildir. Çünkü bütün AES finalistleri 32 bitlik sistemi kullanır. 2 bit CPU ile kaplanan performans boşluğu gerçekten oldukça büyüktür. Bu 386'dan 6800'e kadar değişir. Yada Pentium ailesinin en eskilerinde en yeni CPU'larına kadar değişir.

Her bir CPU çipi, saatin vuruş sıklığını, cache-size mimariyi içeren güçlü ya da zayıf standartlara sahiptir. Her bir özellik algoritmalarının çalıştırılma hızında oldukça önemli bir etkiye sahiptir. Bu noktada; Pentium , CPU, Pentium Pro-CPU ile karşılaştırıldığında çarpım performansında oldukça düşük bir hıza sahiptir.

AES adayları bu tür işlemlere dayalıdır. Bizi cesaretlendiren şudur ki; trend, sonraki jenerasyondaki işlem kodlarında daha iyi performans gösterir. Ama gelecek jenerasyon işlemcileri bu çalıştırma sistemleri ile oldukça yavaş performans gösterecektir. Burada önemli olan nokta şudur ki; AES algoritmalarının performansları değişik CPU'larda kötü bir şekilde değişim gösterebilir. Örnek olarak RC6 küçük marjiniyle Pentium II/III ailesinin en hızlı algoritmasıdır. Ama bunun hızı, Pentiumdaki ve PA,RISC'deki en hızlı adayların hızının yarısından daha azdır. Çünkü bütün adaylar oldukça iyi bir hıza sahiptir.[70]

128 anahtarları için RIJNDAEL ve TWOFISH en hızlı algoritmadır. MARS ve RC6 orta hızlı ve SERPENT ise en yavaş olanıdır. Daha büyük anahtarlar uzunluklar için RIJNDAEL yavaş ilerler. Bazı uygulamalarda da MARS'tan daha yavaş çalışır. Bu eğilim, C'de ve Assembly dilde doğrudur. Buna rağmen SERPENT diğerlerinden Assembly dili performansına yakın olan C kodu üretmeye daha yakındır.[65],[67]

5.4.2.5. En Yüksek Düzeydeki Güvensiz Değişkenlerin Yazılım Performansı

Biham algoritmalarının en düşük düzeydeki güven değişkenlerine göre kıyaslanması fikrini ortaya koymuştur. Farklı dizayn grupları, birbirinden daha az yada daha çok tutucudur. Biham güvenli bir yol olan döngülerin, en düşük numaralarının belirlemek suretiyle algoritmaları normalleştirmiştir. Bu Biham' ın en iyi tahminidir.[65]

Sonra iki standart devre ekledi. NIST üzerindeki yorumlarında Lars Knudsen, farklı algoritmalardaki döngülerin numaralarını değiştirmek için başka bir başparmak kuralı getirdi. " "r"' nin, döngülerin en yüksek numarası olmasına izin ver ki; tüketici anahtar araştırmasından daha atak bir çalışma olsun. " bu kural bize yeni karşılaştırma ölçüsünü vermiştir. [65]

Bu kıyaslama için, ölçme faktörünü farklı bir çalışma için burada bırakıyoruz ve döngülerin en yüksek numaralarının en iyi şifreleme atağı için kıyaslanmaktadır (Bu 256 bit brüt zorlayarak araştırmadan daha kolaydır.). buna da şöyle bir isim verilmektedir (" En yüksek güvensiz değişkenler").

Aşağıdaki tabloda en iyi şekilde yayınlanmış olan şifreleme sonuçlarını verilmektedir.[67]

Tablo 5.7 Şifreleme algoritmalarının döngü sayısı[67]

Algoritma Adı	Döngü
MARS	9-16
RC6	15-20
RİJNDAEL	8-14
SERPENT	9-32
TWOFİSH	6-16

Eğer çok zayıf bir anahtar sınıfı varsa ve eğer atağın zorluk derecesi 2’ den yüksek değilse ve zayıf anahtar zamanını bulma olasılığı varsa atakları sayılır. Bağlantılı bir anahtar atağı varsa yine aynı hesaplama yapılmaktadır. [70]

MARS çok karmaşık bir algoritmadır. Çünkü 4 farklı döngü fonksiyonu vardır. MARS’ ı dizayn eden grup, algoritmanın şifreleme gücünün “core”’ de olduğuna inanmaktadır. Çünkü bu döngüler üzerine yoğunlaşılmıştır. MARS çekirdeğinde(core) döngü atağı vardır. Simetrik olarak 8’ den 3’ e kadar inen 4 farklı döngü fonksiyonu olan “chipper”’ a karşı bir atak vardır.[67],[68]

RC6, 15 atağa karşılık bir döngüye sahiptir. Bu atak ayanı zamanda zayıf anahtar sınıfına da uygulanabilir. Bu atak 2 anahtarda bir için uygulanabilir ve atağın zorluk dercesi 2 dir. RC6 dizaynları, 16 döngünün atak olabileceğini öne sürmektedir. Çünkü somut bir atağı yoktur.[70]

Rijndael, 8 döngüye karşılık farklı bir atağa sahiptir.
Serpent , 9 döngüye karşılık bir atağa sahiptir.
Twofish 6 döngüye karşılık bir atağa sahiptir.

6. DES ve AES Şifreleme Algoritmalarının Steganografi Uygulamarındaki Etkileri

Bu bölümde DES ve AES şifreleme algoritmalarının steganografi uygulamalarında ki örtü nesnesinin(cover-data) üzerindeki etkisini incelenecektir.

6.1. Steganografi

Bu yaklaşım, bir nesnenin içerisine bir verinin gizlenmesi olarak tanımlanabilir. Bu yaklaşımla ses, sayısal resim, video görüntüleri üzerine veri saklanabilir. Görüntü dosyaları içerisine saklanacak veriler metin dosyası olabileceği gibi, herhangi bir görüntü içerisine gizlenmiş başka bir görüntü dosyası da olabilir. Bu yaklaşımda içine bilgi gizlenen ortama örtü verisi (cover-data), oluşan ortama da stego-metin (stego-text) veya stego-nesnesi (stego-object) denmektedir.[71],[72]

Steganografi kelimesi Yunanca “steganos: gizli, saklı” ve “grafi: çizim yada yazım” kelimelerinden gelmektedir. Günümüzde ise sayısal (dijital) nesneler üzerinde steganografi uygulamaları yapılmaktadır ve gelişen teknoloji nedeniyle, verilerimizi korumak amacıyla son yıllarda sıklıkla kullanılmaya başlanmıştır. Steganografi, Dilbilim Steganografi ve Teknik Steganografi olmak üzere kendi içerisinde ikiye ayrılmaktadır. Dilbilim steganografi, taşıyıcı verinin metin (text) olduğu steganografi koludur. Teknik Steganografi ise birçok konuyu içine almaktadır. Bunlar; görünmez mürekkep, gizli yerler, microdot’lar, ve bilgisayar tabanlı yöntemler gibi başlıklar altında toplanabilmektedir.[77],[78]

Steganografi kullanım alanları açısından üçe ayrılmaktadır. Bunlar aşağıdaki gibidir:

- Metin (text) steganografi
- Görüntü (image) steganografi
- Ses (audio) steganografi.

Görüntü steganografisinde bilgiyi resmin içine gizlemek için çeşitli yöntemler vardır. Bunlar şu şekilde sınıflandırılabilir.[71],[73],[79]

- En önemsiz bite ekleme
- Maskeleme ve filtreleme
- Algoritmalar ve dönüşümler

6.1.1 Sayısal Resmin Yapısı

Sayısal (dijital) resim N satır ve M sütunluk bir dizi ile temsil edilir. Genellikle satır ve sütun indeksleri y ve x veya r ve c olarak gösterilebilir. Bir resim dizisinin elemanlarına piksel denir. En basit durumda pikseller 0 veya 1 değerini alırlar. Bu piksellerden oluşan resimlere ikili (binary) resim denir. 1 ve 0 değerleri sırasıyla aydınlık ve karanlık bölgeleri veya nesne ve zemini (nesnenin önünde veya üzerinde bulunduğu çevre zemini) temsil ederler [71],[74]. Sayısal (dijital) görüntü dosyaları renkli olarak genellikle 24 yada 8 bit; gri-seviye görüntüler 1-2-4-6 yada 8 bit olabilirler.

6.2. LSB (Least Significant Bit - En az öneme sahip bit) ekleme yöntemi

"LSB (Least Significant Bit) Insertion" yöntemi, örtü verisine ait segmentlerde her byte'ın en az anlamlı biti yerine gizlenecek verinin bitleri sırasıyla verinin başlangıcından itibaren birer birer yerleştirilir. Burada her sekiz bitin en fazla bir biti değişikliğe uğratıldığından ve eğer değişiklik olmuşsa da değişiklik yapılan bitin byte'ın en az anlamlı biti olmasından dolayı, ortaya çıkan stego verisindeki (= örtü verisi + gömülü veri) modifikasyonlar insan tarafından algılanamaz boyuttadır. Steganaliz yapıldığında, stego verinin her byte'ının son bitlerinin birleştirilmesiyle gizli mesajı

oluşturmak kolaydır; dolayısıyla pratikte diğer yöntemlere göre daha az güvenilirdir ve daha az kullanılır. [76]

Sondaki bitin 1 veya sıfır olması sayının değerini çok fazla değiştirmeyecektir. Sondaki bit değeri 1 yerine 0 olsaydı bu, renk üzerinde gözle görülecek büyük bir değişikliğe neden olmayacaktı. İşte bu sondaki biti LSB olarak adlandırılır. Bu bitler yerine gömülecek veriler girilirse veri gizlenmiş olabilir. Orijinal resimle içerisine data sakladığımız resim arasında gözle görülür bir fark yoktur.

Günlük yaşantıdaki işlemlerin dijitalleşme yüzdesinin hızlıca artması bunun bir göstergesidir. Bu oran Amerika'da %60 seviyelerinde iken ülkemizde %10'un altında olup, yapılan işlemlerin büyük çoğunluğu bankacılık işlemlerini kapsamaktadır. Ülkemizde bilgi güvenliği için yeni yaklaşımlara duyulan ihtiyacın bu yüzdenin artmasıyla daha da yükseleceği ortadadır.

2. dünya savaşından 35 yıl sonra açıklanan bir raporda, İngiltere ile Almanya arasında olan savaşın seyrinin, savaş taktiklerini içeren mesajların çözülmesiyle değiştiği bildirilmiştir. Bu şifre çözme olayı, savaşın müttefikler tarafından kazanılmasında büyük rol oynamıştır. Boston Globe gazetesi, bu olayın 2. dünya savaşı hakkında daha önce yazılmış tarih kitaplarında bir çok değişiklikler yapılmasını gerektirecek kadar önemli olduğunu vurgulamıştır. Tarihi değiştirecek kadar önemli olan şifre çözme olayı bu konunun önemini vurgulamada verilebilecek en iyi örneklerden birisidir. [76]

6.3. Şifreleme algoritmalarının Örtü Nesnesi Üzerindeki Etkisi

Steganografi yukarıda da bahsedildiği gibi verileri resmin için gömme işlemini gerçekleştirmektedir. Steganografi ile şifreleme algoritmalarını birleştirirsek yani resmin içine şifreli metinleri gömersek, üçüncü kişiler resmin içine bir veri

gömüldüğünü fark etseler bile ikinci bir zorlukla karşılaşacaklardır. Böylelikle daha güçlü bir algoritmaya sahip olunmaktadır. Ama burada ki önemli nokta kullanılan şifreleme algoritması resmin örtü nesnesini çok fazla bozmamalıdır. Bu bölümde DES ve AES ile şifrelenmiş metni ve açık metin olarak resmin içine gömülen açık ve şifrelenmiş metinlerin örtü nesnesi üzerindeki etkisi incelenecektir.

Bir resmin içine steganografi ile bir veri gömüldüğünde, örtü nesnesin üzerindeki değişim çok önemlidir. Bundan dolayı, örtü nesnesin üzerindeki değişim oranını belirlemek için metotlar bulunmaktadır.[80] Bunlar;

- Ortalama Hata Kareleri (MSE)

$$\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(x_n - y_n)^2$$

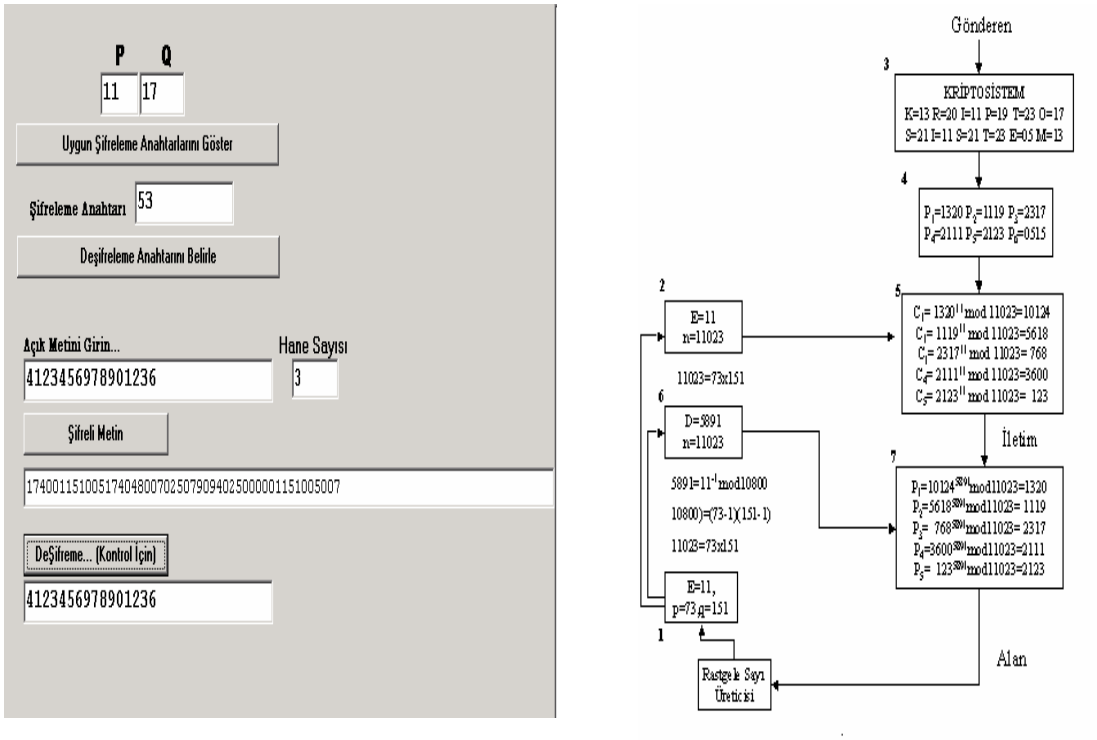
- En büyük işaret gürültü oranı (PSNR)

$$PSNR(dB) = 10\log_{10}\frac{x_{peak}^2}{\sigma_d^2}$$

7. TEZ ÇALIŞMASINDA YAPILAN UYGULAMALAR

7.1 RSA Uygulamamsı

RSA algoritmasının programını Delphi 6.0 da gerçekleştirdik ve Tablo 7.1 de şifreleme ve deşifreleme hızlarına ayrıntılı olarak gösterilmiştir. Bu çalışmada Pentium III 2.6 Ghz işlmenci ve 512 RAM kullanılmıştır.[25] Şekil 7.1'de RSA şifreleme algoritmasının akış diagramı ile uygulama programı görüntüsü ile bir sayısal örnek verilmiştir.



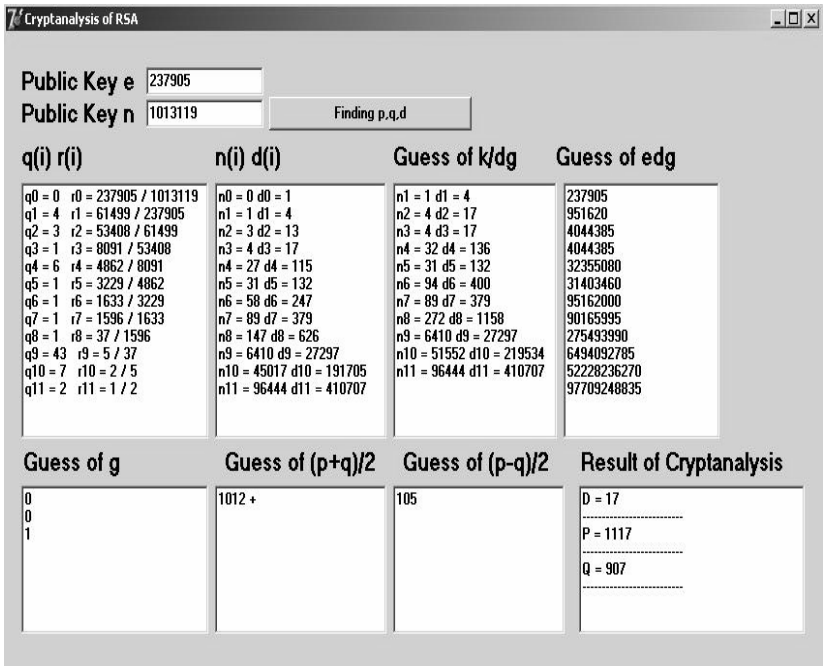
Şekil 7.1 RSA Uygulama Programının Görünüşü

Tablo 7.1 RSA Şifreleme ve Deşifreleme Süreleri

P	Q	E	D	Deşifreleme anahtarının bulunma süresi	Şifreleme süresi (4 digit)	Deşifreleme süresi
137	149	127	6815	1 sn.'nin altında.	1 sn.'nin altında	1 sn.'nin altında.
503	509	523	212107	8 sn	1 sn.'nin altında	1 sn.'nin altında.
887	907	911	423827	14 sn	1 sn.'nin altında	1 sn.'nin altında.
1559	1567	1549	2312245	1 dk	1 sn.'nin altında	4 sn
2767	2777	2789	3055949	1.30 dk	1 sn.'nin altında	10 sn
3533	3457	3511	9818119	6.48 dk	1 sn.'nin altında	34 sn
4483	4597	4649	20368885	15.52 dk	1 sn.'nin altında	1.12 dk
5221	5197	5153	25965137	19.55 dk	1 sn.'nin altında	2.50 dk
6389	6359	6421	27411221	22.38 dk	1 sn.'nin altında	2.85 dk
7451	7457	7393	41594657	27.53 dk	1 sn.'nin altında	3.15 dk
8117	8221	8147	47933207	35 dk	1 sn.'nin altında	3.90 dk
9851	9923	9859	54640139	41.46 dk	1 sn.'nin altında	4.21 dk

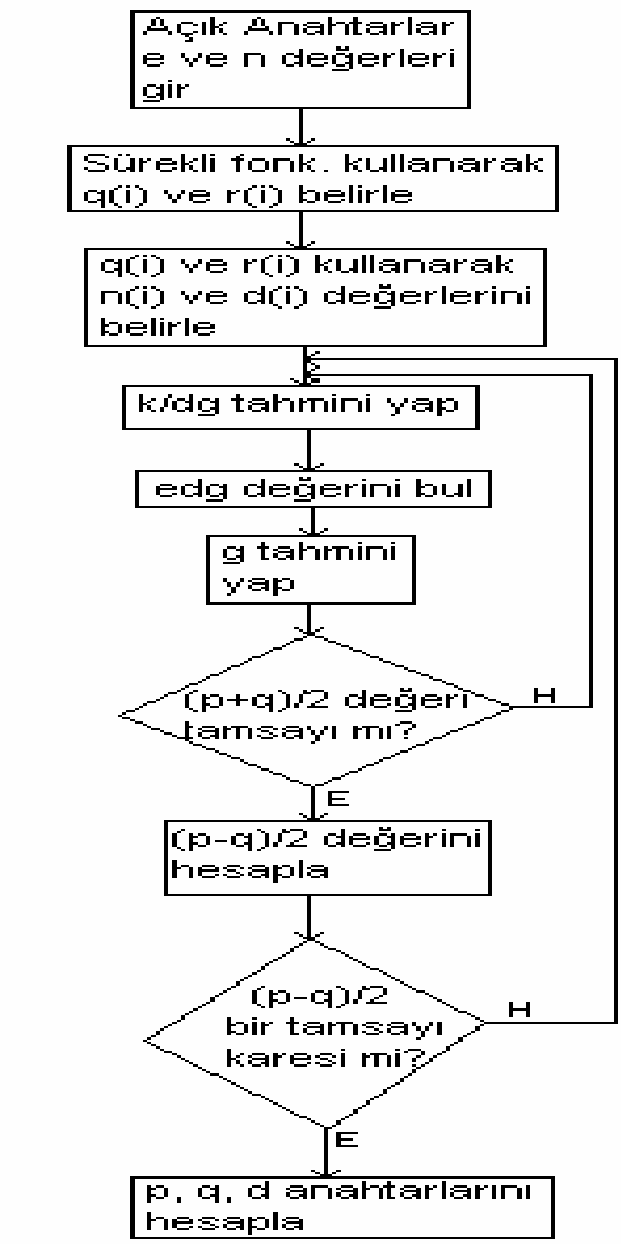
7.2 RSA Kriptanalizi Uygulama Programı

Tezin 3.4 bölümünde teorik olarak anlatılan RSA şifreleme algoritmasının kriptanalizi Delphi 7.0 da uygulama programı gerçekleştirildi. Şekil 7.2'de uygulama programının bir görünüşü ve bir sayısal örneği verilmiştir**.**



Şekil 7.2 RSA Kriptanaliz Uygulama Programı Görünüşü

Şekil 7.3 ‘te RSA şifreleme algoritması üzerine uygulanan kriptanaliz yönteminin akış şeması verilmiştir.



Şekil 7.3 RSA Kriptanalizi Akış Diagramı

7.3 Eliptik Eğri Şifreleme Algoritmasının Uygulaması

Bu çalışmada C++ programlama dili ile ECC kullanan bir dijital imza uygulaması gerçekleştirildi. Bu çalışmada Pentium III 2.6 Ghz işlmenci ve 512 Mb RAM kullanılmıştır. Örnek 1 ve örnek 2 de uygulama programının görüntüsü verilmiştir.

Programda öncelikle uygun eliptik eğri seçilmekte ve hatalı bir eliptik eğri seçildiği takdirde program hata mesajı vermektedir. [41],[42]

Örnek 1:

Öncelikle uygun eliptik eğri seçiliyor.

$$y^2 = x^3 + x + 19 , \quad p=59; \text{ Ep } (1, 19)$$

Daha sonra Tablo 7.2'de verildiği gibi bu eliptik eğri üstünde kullanabilecek bütün noktalar belirleniyor. Bu noktalar arasından rasgele bir G=(12,15) noktası seçiliyor.

Tablo 7.2. Eliptik Eğri Üstündeki Noktalar

Eğri üzerindeki noktalar						
(0,14)		(11,2)		(28,13)		(44,46)
(0,45)		(11,57)		(28,46)		(46,13)
(1,27)		(12,15)		(30,26)		(46,46)
(1,32)		(12,44)		(30,33)		(47,7)
(2,18)		(13,20)		(31,20)		(47,52)
(2,41)		(13,39)		(31,39)		(49,22)
(3,7)		(14,2)		(32,29)		(49,37)
(3,52)		(14,57)		(32,30)		(50,15)
(4,21)		(15,20)		(34,2)		(50,44)
(4,38)		(15,39)		(34,57)		(54,19)
(6,8)		(16,1)		(35,6)		(54,40)
(6,51)		(16,58)		(35,53)		(56,15)
(7,29)		(18,21)		(37,321)		(56,44)
(7,30)		(18,38)		(37,38)		(57,3)
(9,29)		(19,17)		(38,0)		(57,56)
(9,30)		(19,42)		(42,24)		(58,28)
(10,12)		(20,29)		(42,35)		(58,31)
(10,47)		(20,30)		(44,13)		

Son olarak dijital imza gerçekleştirecek kişiler kendilerinin bileceği gizli anahtarları belirler. Burada K_A ve K_B kişilerin gizli anahtarlarını temsil etmektedir. Seçtikleri ortak noktayı (G) kendi gizli anahtarları ile çarparak açık anahtarlarını elde etmektedirler.(P_A, P_B)

$K_A = 31$, $K_A * G = P_A\,(31,39)$

$K_B = 41$, $K_B * G = P_B\,(31,21)$

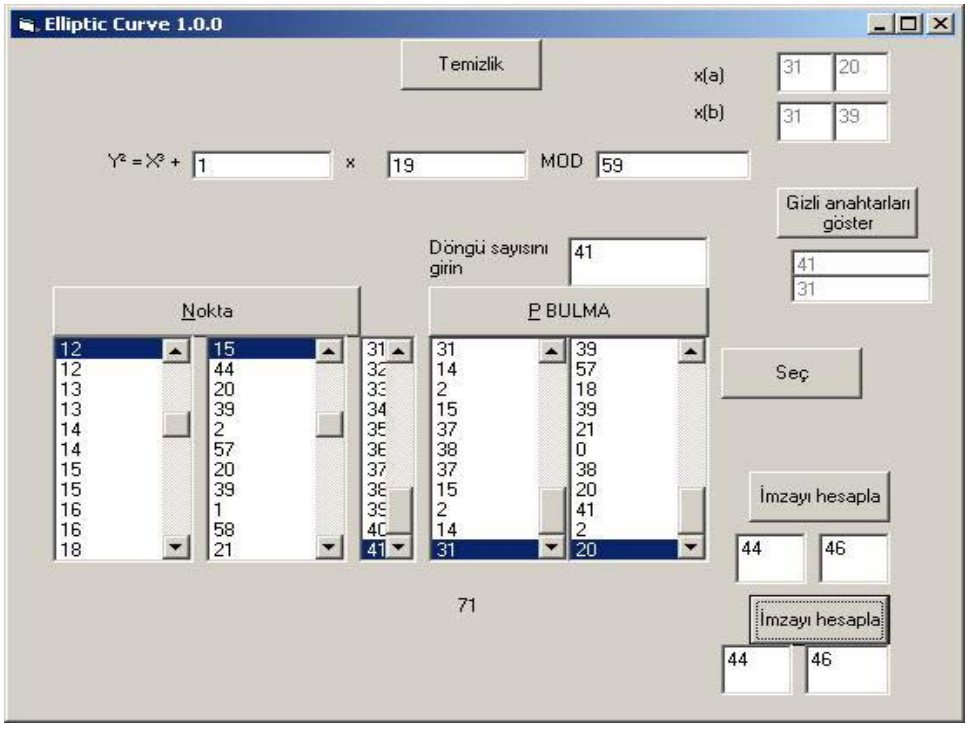
Birbirlerinin açık anahtarlarını ve kendilerinin gizli anahtarlarını kullanarak ,

$C = K_A * P_B$ ve $C = K_B * P_A$ işlemlerini gerçekleştirerek;

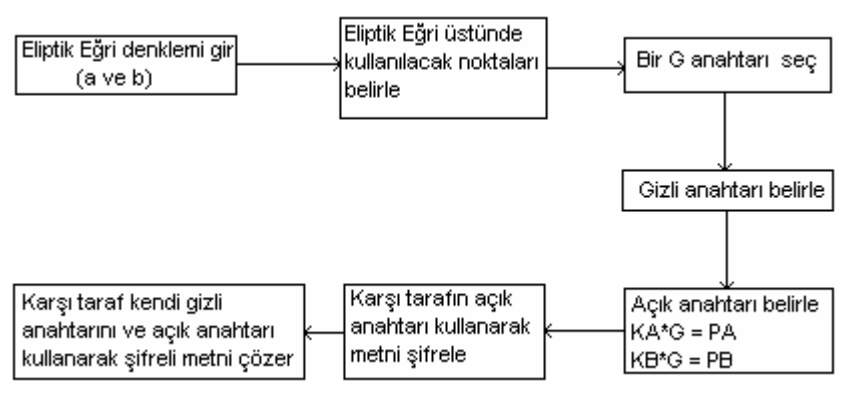
$C = K_A * P_B = (44,46)$ ve $C = K_B * P_A = (44,46)$

noktalarını bulurlar.

Sonuçtan da anlaşılacağı gibi, birbirlerinin gizli anahtarlarını bilmemelerine rağmen kendi gizli anahtarlarını ve karşısındaki kişinin açık anahtarını kullanarak aynı sonuca ulaşmışlardır.[41],[42]



Şekil 7.4 ECC Program Görüntüsü



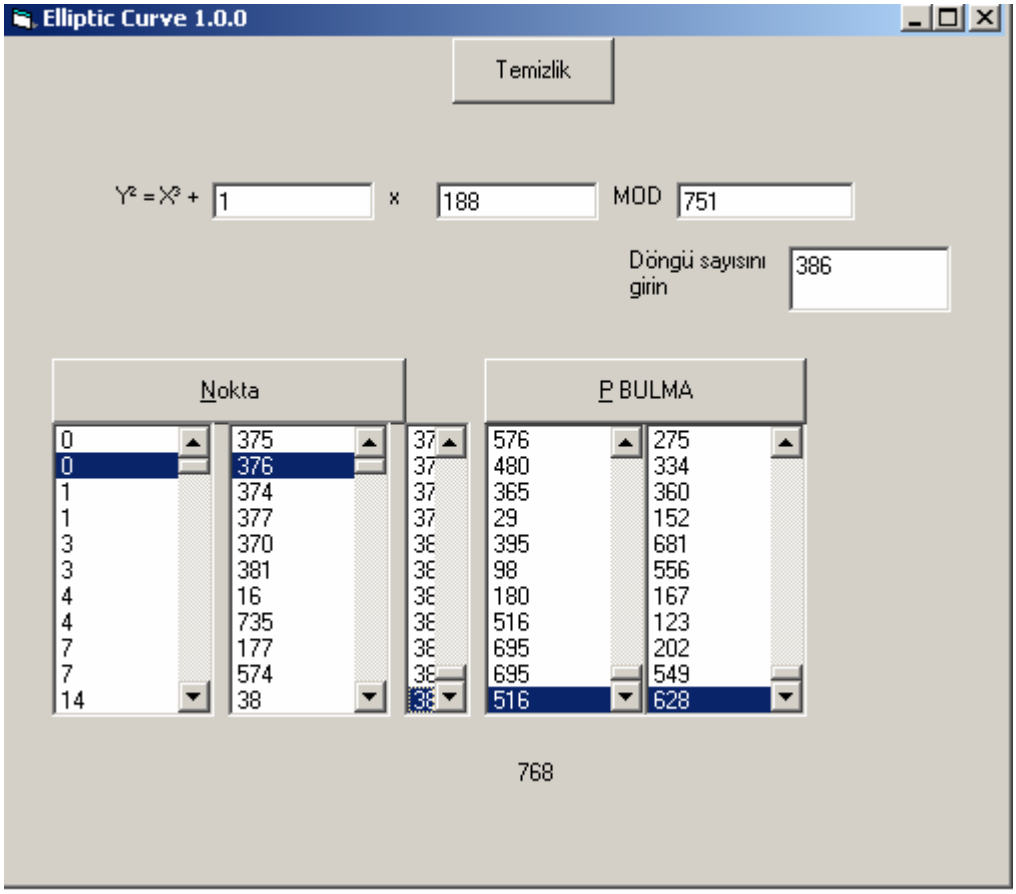
Şekil 7.5 ECC Akış Diagramı

Örnek 2:

Örnek olarak: p=751; Ep (-1, 188), alınırsa eğri

$y^2= x^3 –x+188$ olur ve kullanılabilecek 726 eliptik eğri noktası bulunmaktadır. Daha büyük asal sayılar seçildiğinde, kullanılabilecek nokta sayısı artmaktadır. Uygulama programımız 100.000 seviyesinde nokta bulabilme kapasitesine sahiptir.
G=(0,376) seçilirse;

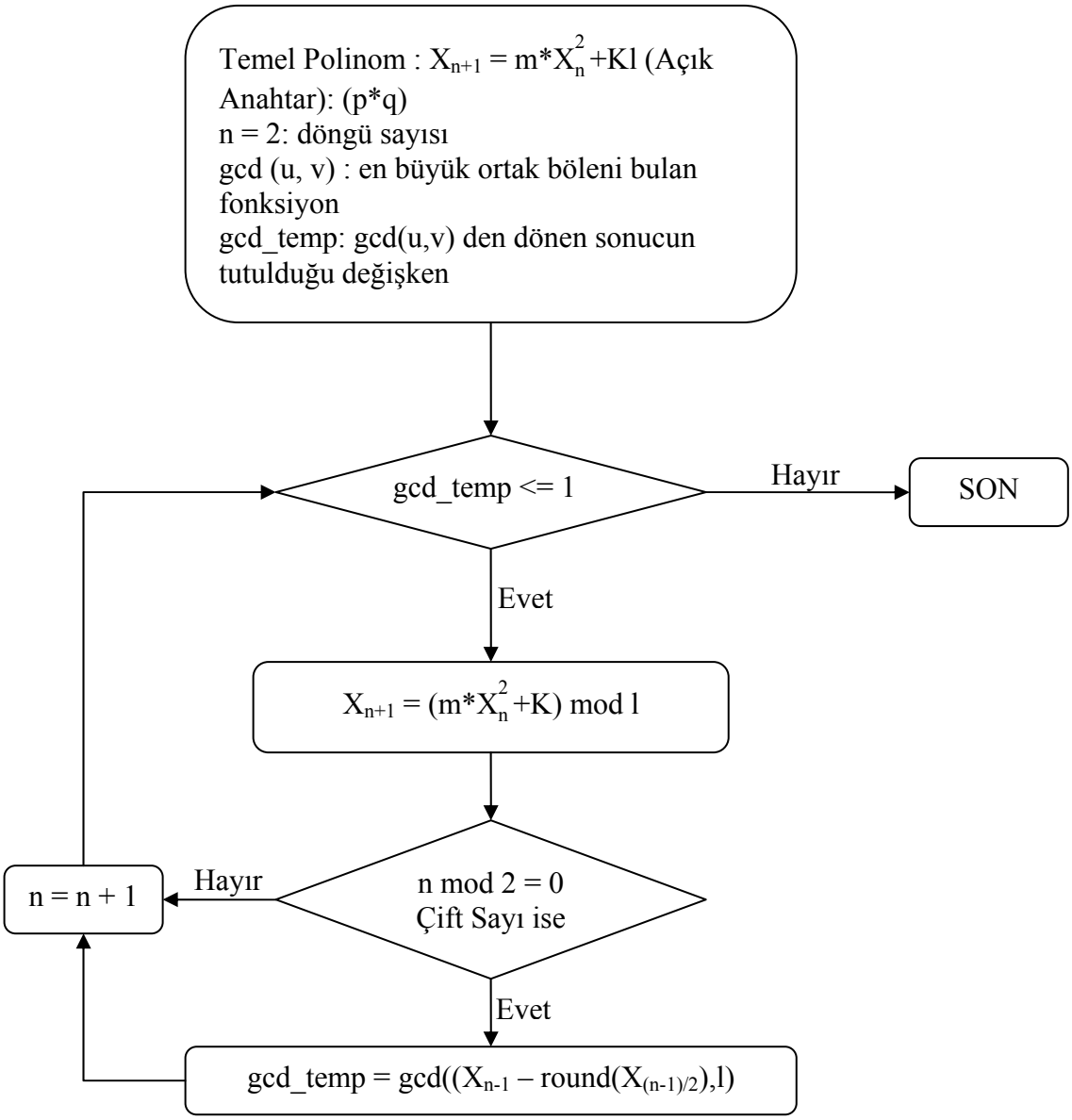
- A'nın B'ye , Pm (562,201) eliptik noktasında şifreli bir mesaj göndereceğini ve A'nın k=386 seçtiğini farzedelim.
- B'nin açık anahtarı PB=(201,5) dir. Buradan 386(0,376) = (676,558)
- (562,201) + 386(201,5) = (385,328) elde edilir. Böylece A şifreli metin olarak
- {(676,558),(385,328)} gönderir.



Şekil 7.6 Örnek Program Görüntüsü

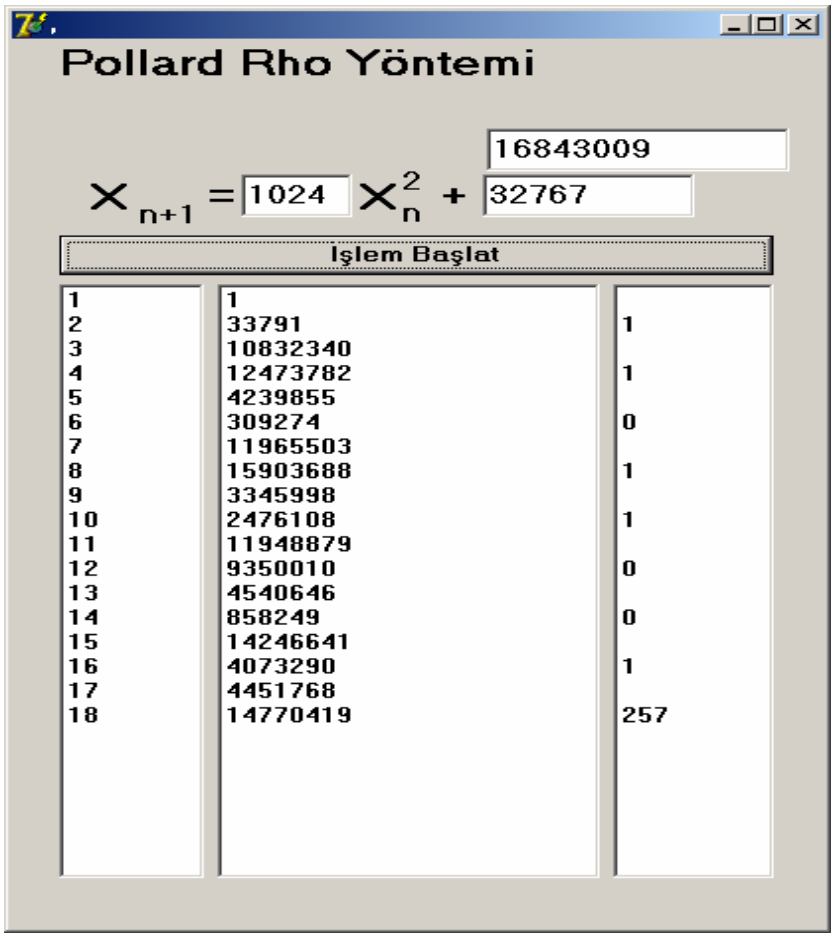
7.4 Pollard Rho Algoritması ve Sayısal Örnek uygulaması

Bölüm 4.9 anlatılan teorik olarak anlatılan Pollard Rho algoritmasının şekil 7.7 akış diyagramı ve sayısal örneği verilmiştir.

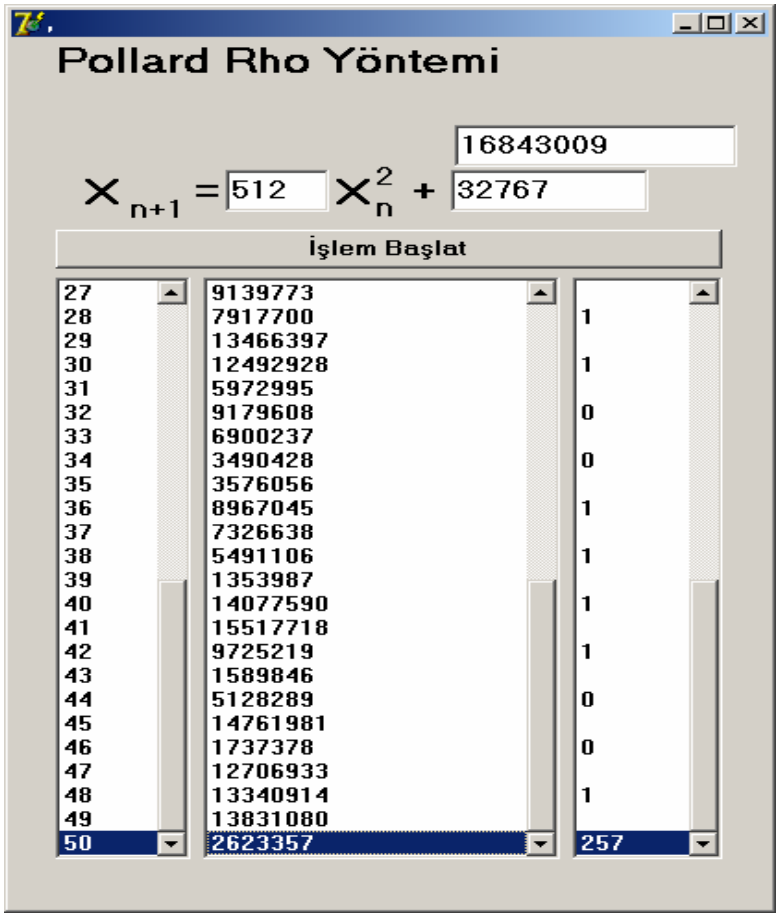


Şekil 7.7 Pollard Rho Algoritması

Delphi 7.0 da uygulaması gerçekleştirlen yazılım programı şekil 7.8 ve şekil 7.9 ile gösterilmiştir. Bu çalışmada Pentium III 2.6 Ghz işlmenci ve 512 RAM kullanılmıştır. Bu örnekte, iki asal çarpandan oluşan 16843009 sayısı asal çarpanlarına ayrılmıştır.



Şekil 7.8 Pollard Rho Örnek-1

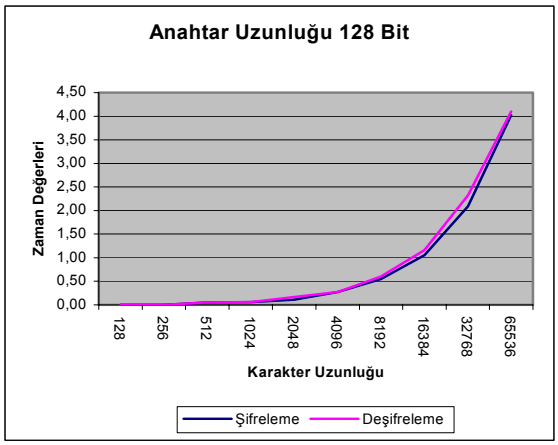


Şekil 7.9 Pollard Rho Örnek-2

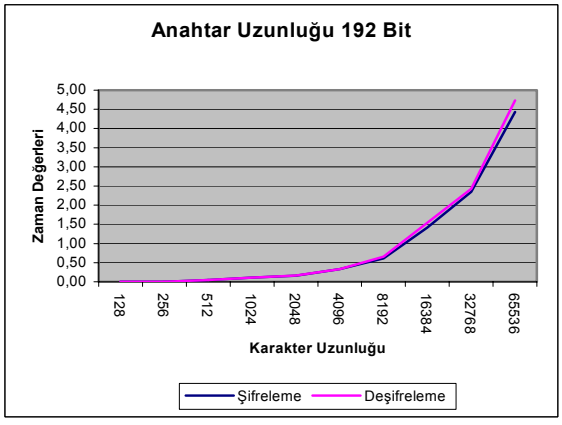
7.5 AES Performans analizi

AES finalistlerinin arasında yazılım ve donanım performansı olarak en iyi algoritma seçilen RIJNDAEL şifreleme algoritmasını Visual C++'ta şifreleme ve deşifreleme yapan bir uygulama programı gerçekleştirildi. . Bu çalışmada Pentium III 2.6 Ghz işlmenci ve 512 RAM Mb kullanılmıştır RIJNDAEL şifreleme algoritmasının belirli uzunluktaki mesaj verilerinin şifreleme ve deşifreleme süreleri Şekil 7.10, Şekil 7.11 ve Şekil 7.12 ile verilmiştir. [67],[70]

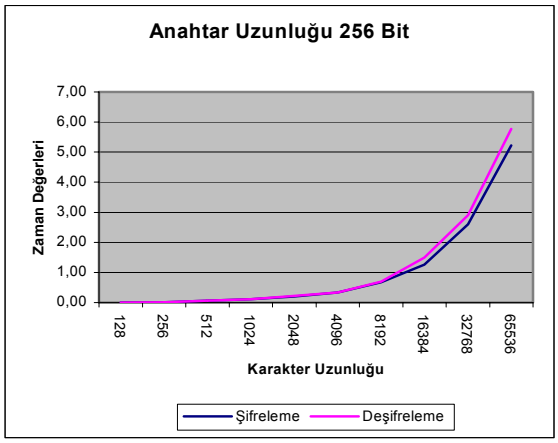
Şekil 7.10. 128 Bit Anahtar Uzunluğuna Sahip RIJNDAEL Algoritmasının Şifreleme Ve Deşifreleme Süreleri



Şekil 7.11. 192 Bit Anahtar Uzunluğuna Sahip RIJNDAEL Algoritmasının Şifreleme Ve Deşifreleme Süreleri



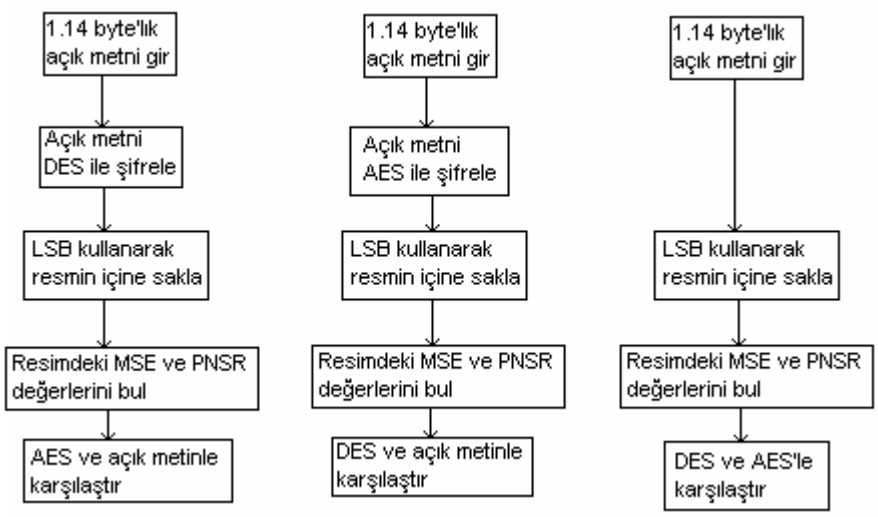
Şekil 7.12. 256 Bit Anahtar Uzunluğuna Sahip RIJNDAEL Algoritmasının Şifreleme Ve Deşifreleme Süreleri



7.6 DES ve AES Şifreleme Algoritmalarının Steganografi (Örtü Nesnesi) Üzerindeki Uygulamamsı

Bu uygulama da 1.14 KB'lık bir açık metin seçildi ve bu açık metin 56-bit lik anahtar değerine sahip DES algoritmasıyla ve 128-bitlik anahtar kullan AES şifreleme algoritmalarıyla ayrı ayrı şifrelendi. Resmin içine gömme işlemi LSB yöntemi kullanılarak gerçekleştirildi.

Şekil 7.13 Örnek Resim (150*130 Piksel)



Şekil 7.14 Uygulama Akış Diagramı

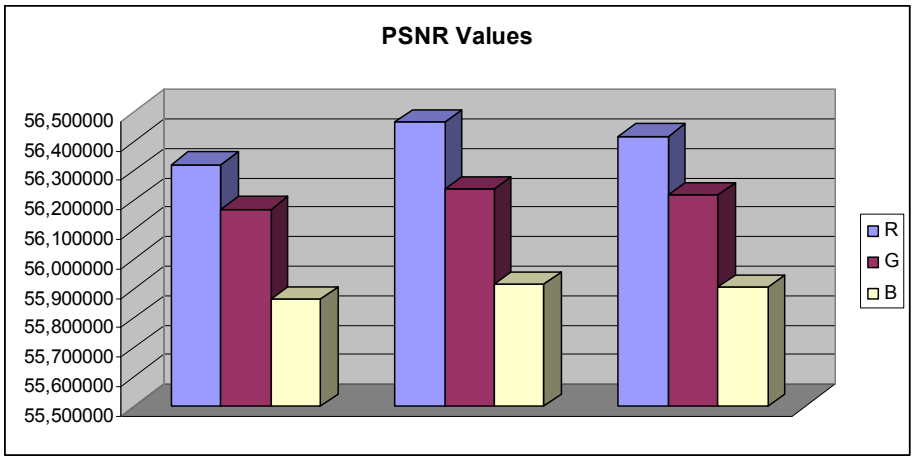
(a) (b) (c)

Şekil 7.15 Mesaj Gömülmüş Resimler

Şekil 7.15'teki resmin içine sadece açık metin gömülmüştür. Şekil 7.15 b ve Şekil 7.15 c ye sırasıyla DES ile şifrelenmiş metin ve AES ile şifrelenmiş metin gömülmüştür.

Tablo 7.3 MSE ve PNSR Değerlerinin Karşılaştırılması

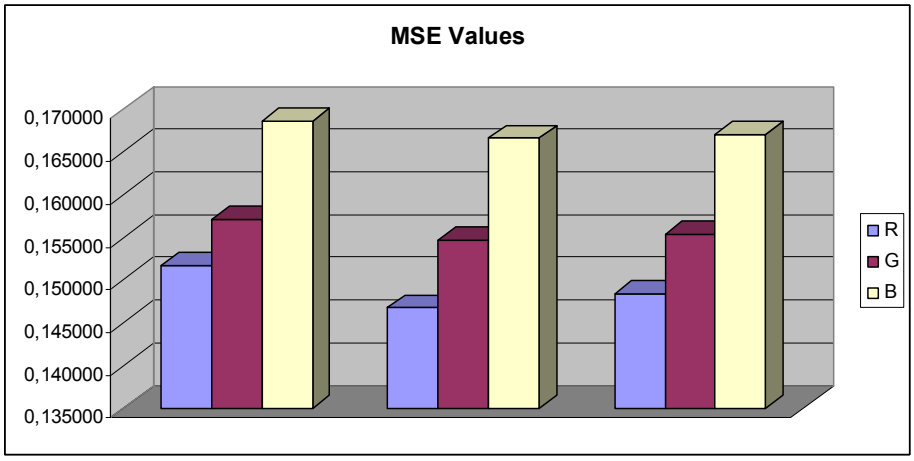
Karşılaştırılan Resimler		PSNR Değerleri	MSE Değerleri
	R	56.321168	0.151692
girl.bmp – (a)	G	56.169680	0.157077
	B	55.863153	0.168564
	R	56.464453	0.146769
girl.bmp – (b)	G	56.236836	0.154667
	B	55.913653	0.166615
	R	56.417665	0.148359
girl.bmp – (c)	G	56.216723	0.155385
	B	55.904306	0.166974



Şekil 7.16 PSNR Değerlerinin Karşılaştırılması

Şekil 7.16 ve şekil 7.17 den de anlaşılacağı gibi DES ile şifrelenmiş veri resme gömüldüğünde AES ile şifrelenmiş metni gömülen resimden daha iyi sonuçlar

vermektedir. Sonuç olarak DES şifreleme algoritmasını kullanarak Steganografi işlemimi üçüncü kişilere karşı daha güçlü hale getirilebilir.



Şekil 7.17 MSE Değerlerinin Karşılaştırılması

8. SONUÇ

Günümüzde bilgisayar ağlarının ve haberleşme sistemlerinin güvenliğinin sağlanması için kullanılan en önemli işlem, verilerin şifrelenerek anlamsız hale getirilip hedefe gönderilmesi ve hedefte tersi işlem yapılarak tekrar eski hale getirilmesidir. Kriptografi bilimi aracılığıyla verilerin güvenli bir şekilde şifrelenip gönderilmesi ve tekrar deşifre edilebilmesi için şifreleme algoritmaları oluşturulmaktadır. Kriptografi biliminin başlangıcı Sezar dönemine kadar uzanmakta ve günümüze gelene kadar her dez avantajını ortadan kaldırabilecek algoritmalar oluşturarak gelişimini sürdürmüştür.

Bu çalışmada günümüzde yaygın olarak kullanılan simetrik ve asimetrik şifreleme algoritmaları incelendi ve uygulamalarını gerçekleştirildi.Simetrik şifreleme algoritmalarından DES ve AES şifreleme algoritmaları ve asimetrik şifreleme algoritmalarından RSA ve Eliptik eğri şifreleme algoritması incelendi.

Öncelikle asimetrik şifreleme algoritmalarından günümüzde yaygın olarak kullanılan RSA ve yakın zamanda oluşturulan Eliptik eğri şifreleme algoritmaları incelenildi. Asimetrik şifreleme algoritmaları çözülmesi zor matematiksel problemler üzerine kurulmuşlardır. Matematik teoremleri şifreleme algoritmalarının alt yapısını oluşturmaktadır. Asimetrik şifreleme algoritmalarının gelişimine bakarsak her bir algoritma yeni bir teorem üstüne kurularak bir önceki algoritmanın dezavantajlarını ortadan kaldırmayı amaçlamıştır. 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. Ancak n (asal çarpanları olan çok büyük sayı) yeterince büyük seçilirse günümüzün teknolojisiyle n'nin çarpanlarına ayrılması "yeterince" uzun süreceği için bu yöntemle n'nin hesaplanması hesaplama açısından verimsiz olacaktır.

Günümüzde RSA algoritmasının daha güvenli hale getirebilmek için çok büyük asal sayılar kullanılmaktadır. Bu algoritmaların daha güvenli hale getirmek için büyük anahtar değerleri kullanılmaktadır. Büyük anahtar değerlerini kullanmak birçok

uygulamada şifreleme ve deşifreleme sürelerini uzatmaktadır. Şifreleme algoritmalarının güçlü güvenliğe sahip olmaları yanında yeni donanımlarla gerçekleştirilebilmeleri, kolaylık ve performansı yüksek olması göz önünde bulundurulmalıdır Aynı şekilde yazılım olarakta kolaylığı ve Cpu'u fazla meşgul etmemesi gerekmektedir. Çoğu kuruluş çalışanların verimliliğini arttırmak ve ağ üzerinde uygun bir işbirliği sağlamak için kablosuz ağ sistemini yaygınlaştırıyor. Bu ağ sistemlerinin korunması büyük bir önem teşkil eder. Çünkü kablosuz ağ trafiği kolayca engellenmeye açık olabilir. Kablosuz ağlarda bant genişliğinin verimli bir şekilde kullanılması için uygulanacak şifreleme algoritması bant genişliği ve hız bakımından bu sisteme uyumlu olması gerekmektedir. Bu özellik Asimetrik şifreleme algoritmalarının en büyük dezavantajıdır. Son yıllarda yapılan çalışmalarla, asimetrik şifreleme algoritmalarının bu dez avantajını ortadan kaldırmak veya daha düşük anahtar değerleriyle aynı güvenliği sağlayabilme çalışmaları yapılmıştır. Bu çalışmalar sonucun da Eliptik eğri şifreleme algoritması (ECC) geliştirilmiştir.

1985 yılnda Eliptik eğri gruplarının sonlu alanlar üzerinde tanımlanmasının kripto sistem için temel alınması ilk olarak Neal Koblitz ve Victor Miller tarafından önerilmiştir. Eliptik eğri yaklaşımı standart RSA sisteminden daha zengin matematiksel işlemler içermektedir. Eliptik eğri şifreleme algoritması Eliptik Eğri Ayrık Logaritma Problemi (ECDLP) üzerine kurulmuş bir algoritmadır.

Eliptik eğri şifreleme algoritması, RSA şifreleme algoritmasına göre daha düşük anahtar seviyelerinde aynı güvenliği sağlamıştır. Ama günümüzde daha çok fazla yaygın olarak kullanılmaya başlanmamıştır.

Asimetrik şifreleme algoritmalarına karşı yapılan saldırılar çok büyük sayıları asal çarpanlara ayırma metotlarıyla gerçekleştirilmektedir. Bu çalışmada, Michel J. Wiener tarafından önerilen sürekli kesirlerin kullanılarak n sayısını asal çarpanlara ayırma metodunu incelenildi. RSA şifreleme algoritmasına farklı saldırı teknikleri bulunmaktadır ve belli sayı değerlerine kadar bu saldırılar başarılı olmuştur. Fakat RSA şifreleme algoritması yüksek anahtar değerlerini kullanarak bu saldırılara karşı hala güvenilirliğini korumaktadır.

Eliptik eğri şifreleme algoritmasına karşı yapılan saldırılar ECDLP problemini çözmeye yönelik saldırılardır ECDLP'yi çözmek için kullanılan algoritmalardan en bilineni ve en genel-amaçlı olan algoritma Pollard'ın rho algoritmasıdır. Tüm-üssel olarak çalışma zamanı (pr)1/2/2 nokta toplamıdır. Sabitlenmiş bir Fq alanı, Pollard'ın rho metoduna maksimum rezistansı ile r asal olmak ve olabileceği kadar büyük bir sayı olması koşulu ile E de bir eliptik eğri seçimi yapılmaktadır. Örneğin r≈q. kriptanalistlerin karşılaştığı zorluklar, bu eğrileri hızlı çözebilecek eliptik eğri ayrık logaritmik problemler keşfetmek olmuştur. Sonlu bir A alanı eliptik eğri kriptografisinde eğer k üzerindeki tüm eliptik eğriler ayrık logaritma problemi için örnekleniyor ise örneklerin zayıf olduğu söylenmektedir. Bunlar Pollard'ın rho metodu ile daha zor örneklerin çözümünden daha kısa sürede çözülebilmektedir.

Asimetrik şifreleme algoritmalarından incelediğim RSA ve ECC şifreleme algoritmaları günümüzde hala kriptanalatik saldırılara hala karşı koymaktadır. Asimetrik şifreleme algoritmalarının hala en büyük dezavantajı olan büyük anahtar değerlerinin kullanımı bu algoritmaların güvenliği açısından hala çok önemlidir. Asimetrik şifreleme algoritmaları üstünde çalışmalar devam etmekte ve bu dezavantajı ortadan kaldırabilecek algoritmalar oluşturulmaya çalışılmaktadır. ECC algoritmasını bu dezavantajı ortadan kaldırmak için günümüzde kullanılan en iyi asimetrik şifreleme algoritmasıdır.

Ayrıca bu çalışmada, simetrik şifreleme algoritmalarından en önemlilerinden DES ve AES şifreleme algoritmaları incelenilmiştir. DES şifreleme algoritması yapılan saldırılara karşı zayıf kalmasından sonra daha güçlü bir algoritma olan AES şifreleme algoritması oluşturuldu. Asimetrik şifreleme algoritmalarında olduğu gibi simetrik şifreleme algoritmalarında da yeni oluşturulan şifreleme algoritmaları bir öncekinin dezavantajlarını ortadan kaldırılması için oluşturulmuştur.

Asimetrik 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 (DES, AES) 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 edilmektedir.

Son bölümde kısaca steganografi bilimi incelenilmiştir. Simetrik şifreleme algoritmalarından DES ve AES şifreleme algoritmalarını steganografi sistemiyle birleştirilerek daha güvenli bir sistem elde edilmesi amaçlanmıştır.. Veriyi resmin içine açık metin olarak değil de şifrelenmiş bir şekilde gömerek örtü nesnesi üstündeki değişim gözlenilmiştir. Bu çalışma sonucunda; yüksek anahtar değerleri kullanılırsa örtü nesnesi üstündeki değişimin artmakta olduğu sonucu elde edilmiştir.

Sonuç olarak kriptografi bilimi kendini geliştirmekte ve oluşturulan her yeni algoritma bir önceki şifreleme algoritmasının dezavantajlarını ortadan kaldırmayı amaçlamıştır. Asimetrik şifreleme algoritmaları üzerine yapılan çalışmalar doğru yönde ilerleyip daha düşük anahtar değerleriyle aynı güvenlik seviyeleri yakalanırsa, asimetrik şifreleme algoritmalarının en büyük avantajlarında biri olan hem şifreleme-deşifreleme işlemlerini gerçekleştirmek hem de sayısal imza ve sertifika işlemlerini birlikte gerçekleştirerek uygulama olanakları daha ileri seviye çıkacaktır. Asimetrik şifreleme algoritmaları matematik teoremleri üzerine kurulmuş algoritmalardır. Matematiksel teoremler incelenip, bu teoremler üzerine yerleştirilecek yeni şifreleme algoritmaları asimetrik şifreleme algoritmalarını daha ileriye götürebilir. Farklı gruplardaki şifreleme algoritmaları birleştirilerek daha güçlü şifreleme algoritmalar gerçekleştirilebilir.

TEZ SIRASINDA YAPILAN ÇALIŞMALAR

Uluslararası Kongre ve Sempozyum Bildirileri

1- T.YERLİKAYA, E.BULUŞ, "A digital signature application with elliptic curve cryptography", 1st International Symposium on Information Technologies, ISIT2005, Girne-KIBRIS, Nisan-2005

2- T.YERLİKAYA, E.BULUŞ, "New Generation Public Key Crypto Elliptic Curve Cryptology", International Scientific Conference-UNITECH'04, Gabrovo-BULGARIA, Kasım-2004

Ulusal Kongre ve Sempozyum Bildirileri

1- T.YERLİKAYA, E.BULUŞ, H.N.BULUŞ, "Eliptik Eğri Aritmetiği ve Şifrelemede Kullanılması-Arithmatic of elliptic Curves and use in cryptography", Sinyal İşleme ve İletişim Uygulamaları Kurultayı-SİU 2006, Antalya-TÜRKİYE, Nisan-2006

2- T.YERLİKAYA, E.BULUŞ, H.N.BULUŞ, "ASİMETRİK ŞİFRELEME ALGORİT-MALARINDA ANAHTAR DEĞİŞİM SİSTEMLERİ", Akademik Bilişim Konferansları 2006-AB2006, Denizli-TÜRKİYE, Şubat-2006.

3- T.YERLİKAYA, E.BULUŞ, H.N.BULUŞ, "KRİPTO ALGORİTMALARININ GELİŞİMİ VE ÖNEMİ", Akademik Bilişim Konferansları 2006-AB2006, Denizli-TÜRKİYE, Şubat-2006.

4- T.YERLİKAYA, E.BULUŞ,, D.ARDA, "ASİMETRİK KRİPTO SİSTEMLER VE UYGULAMALARI", II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi-MBGAK'2005, İstanbul-TÜRKİYE, Kasım-2005.

5- D.ARDA, E.BULUŞ, T.YERLİKAYA, "Simetrik Kriptosistemlerden Cok Alfabeli Yerine Koyma Metodunun Türkiye Türkcesinin Yapısal Özelliklerini Kullanarak Kriptanalitik İncelenmesi", Ağ ve Bilgi Güvenliği Ulusal Sempozyumu-ABG2005, İstanbul-TÜRKİYE, Haziran-2005.

6- T.YERLİKAYA, E.BULUŞ, D.ARDA, "ELİPTİK EĞRİ ŞİFRELEME ALGORİTMASI KULLANAN DİJİTAL İMZA UYGULAMASI", Ağ ve Bilgi Güvenliği Ulusal Sempozyumu-ABG2005, İstanbul-TÜRKİYE, Haziran-2005.

7- D.ARDA, E.BULUŞ, T.YERLİKAYA, "TÜRKİYE TÜRKÇESİNİN BAZI DİL KARAKTERİSTİK ÖLÇÜTLERİNİ KULLANARAK VİGENERE ŞİFRESİ İLE ŞİFRELEME VE KRİPTANALİZ", Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu (ELECO 2004), Bursa-TÜRKİYE, Aralık-2004.

8- T.YERLİKAYA, E.BULUŞ, D.ARDA , "AES ADAY ŞİFRELEME ALGORİTMALARININ YAZILIM VE DONANIM PERFORMANS KARŞILAŞTIRILMASI VE UYGULAMALARI", Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu (ELECO 2004), Bursa-TÜRKİYE, Aralık-2004

9- T.YERLİKAYA, E.BULUŞ, "AES Finalistlerinin Karşılaştırılması - RSA Şifreleme Algoritmasının İncelenmesi ve Kriptanalizi", 20. Türkiye Bilişim Kurultayı, İstanbul-TÜRKİYE, 2003.

KISALTMALAR LİSTESİ

RSA	RIVEST-SHAMIR-ADLEMAN CRYPTO SYSTEM
ECC	ELLIPTIC CURVE CRYPTOGRAPHY
DES	DATA ENCRYPTION ALGORITHM
AES	ADVANCED ENCRYPTION ALGORITHM
LSB	LEAST SIGNIFICANT BIT
MDS	MICROWAVE DATA SYSTEMS
IBM	INTERNATIONAL BUSINESS MACHINES
K	ENCRYPTION KEY
PGP	PRETTY GOOD PRİVACY
PEM	PRIVACY ENHANCED MAIL
MOSS	OBJECT SECURITY SERVICES
PCT	PRIVATE COMMUNICATONS TECHNOLOGY
SSL	SECURE SOCKET LAYER
E	PUBLIC KEY
D	PRIVATE KEY
F	FINITE FIELD
LCM	LEASTT COMMON MULTIPLE
GCD	GREATEST COMMON DIVISOR
ECDLP	ELLIPTIC CURVE DISCRETE LOGARITHM PROBLEM
DLP	DISCRETE LOGARITHM PROBLEM
DPS	DATA PROCESSING SYSTEM
NIST	NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
ECB	ELECTRONIC CODEBOOK MODE
CBC	CIPHER BLOCK CHAINING MODE
CFB	CIPHER FEEDBACK MODE
OFB	OUTPUT FEEDBACK MODE
CPU	CENTRAL PROCESSING UNİT
MSE	MEAN SQUARED ERROR
PSNR	PEAK SIGNAL TO NOISE RATIO

KAYNAKLAR

[1] Schneider B. ”*Applied Cryptography Second Edition*”, John Wiley & Sons, Inc., New York, 1996

[2] Koltuksuz A., “*Elektronik Ticarette Güvenlik, Özgürlük Denetimi, Doğruluk-Bütünlük ve Sayısal İmza*“, 4.Türkiye İnternet Konferansı, 1998, İstanbul, Türkiye

[3] Stallings W., “*Cryptography and Network Security: Principles and Practice*”, ISBN 0-13-869017-0, Prentice Hall, 1998

[4] Koblitz, N., “*A course in Number theory and Cryptography*”, Springer Verlag, 1994

[5] Tsunoo Y., Tsujihara E., Minematsu K., Miyauchi H.,”*Cryptanalysis of Block Ciphers Implemented on Computers with Cache*”, ISITA, 2002

[6] NBS FIPS PUB 46, ”*Data Encryption Standard*” , National Bureau of Standarts, U.S. Departman of Commerce, January 1977

[7] Rivest R.,. Shamir A., Adleman L., “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems,” Communications of the ACM, v. 21, n. 2 (Feb 1978), pp. 120-126.

[8] Ellison C., Schneier B., “*Ten Risks of PKI: What You're Not Being Told About Public Key Infrastructure*”, Computer Security Journal, vol. 16, no. 1, pp. 1-7, 2000.

[9] Diffie W., Hellman M.E., “*New Directions in Cryptography*”, IEEE Trans. IT-22 1976, no. 6, 644–654.

[10] Levi A., Özcan M., “*Açık Anahtar Tabanlı Şifreleme Neden Zordur?*” Türkiye Bilişim Konferansı, 2002

[11] Diffie, W., Hellman, M., ”New directions in cryptography.” IEEE Trans. Inform. Theory IT-22, (Nov. 1976), 644-654.

[12] Atay S.,”*Eliptik Eğri Tabanlı Kriptografik Protokol ve Akıllı Kart Üzerinde Bir Uygulama*”, Ağ ve Bilgi Güvenliği Sempozyumu, 2005

[13] Lang S., “*Algebra 2nd Edition*”, Addision-Wesley, 1984

[14] Adleman L., Manders K., Miller G., “*On Takingroots İn Finite Fields*” Proceedings of the 20 th Annual Symposium on the foundations of computer Science, pp.175-178

[15] Rademacher H., “*Lectures On Elementary Number Theory*”, Kriger, 1977

[16] Schroeder M., "*Number Theory İn Science And Communication*", 1984, Springer-Verlag

[17] Herstein I. N., "*Topics in Algebra 2nd Edition*", Wiley, 1975

[18] Artman B., "*Euclid—The Creation of Mathematics",* Springer, New York, Second Edition 2001. (Contains more specific information about the material in this note. The material on the icosahedron is not in the first ed., 1999.) MR1695164 (2000d:01006)

[19] Mueller, *On The Notion Of A Mathematical Starting Point İn Plato, Aristotle, AndEuclid",* A. C. Bowen, ed.: Science And Philsophy İn Classical Greece. New York: Garland, 1991, pp. 59–97. MR1157400

[20] Berlekamp E., "*Algebraic Coding Theory*", Aegean Park Press, 1984.

[21] P1398, "*Standard Specifications For Public−Key Cryptography*", IEEE, October 1998.

[22] Montgomery P., "*Modular Multiplication Without Trial Division*,", Mathematics of Computation, 44, 1985.

[23] El Gamal T., "*A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Advances in Cryptology: Proceedings of CRYPTO 84*", Springer Verlag, pp. 10-18, 1988

[24] Yerlikaya T., Buluş E., Arda D., "*Asimetrik Kripto Sistemler Ve Uygulamaları*", II. Mühendislik Bilimleri Genç Araştırmacılar Kongresi-MBGAK'2005, İstanbul-TÜRKİYE

[25] Tektaş M. ,Baba F. ,Çalışkan M., "*Şifreleme Algoritmalarının Sınıflandırılması Ve Bir Kredi Kartı Uygulaması*", 3RD Internatıonal Advanced Technologıes Symposıum, August 18-20, 2003, Ankara

[26] Quisquater, J.J. and C. Couvreur, "Fast Decipherment Algorithm for RSA Public-Key Cryptosystem", *Electronics Letters*, vol. 18, no. 21, October 1982, pp. 905-907.

[27] Williams H.C., "*A p+1 Method of Factoring*", Mathematics of Computation,vol. 39, no. 159, July 1982, pp. 225-234.

[28] Wiener M., "*Cryptanalysis Of Short RSA Secret Exponents*", IEEE Transactions on Information Theory 36, 1990, 553-558

[29] Wang A., "*Cryptanalysis of RSA With Small Private Key*", IMA Summer Program University of Notre Dame, June 8 – June 27, 2004

[30] Kaliski B., "*The Mathematics of the RSA Public-Key Cryptosystem*", RSA Laboratories NY, 2001

[31] Shiu P., "*Computation of Continued Fractions Without İnput Values*", Math.Comp. 64, 1995, 1307–1317

[32] Aissi C., "I*mplementation of The RSA Algorithm and its Cryptanalysis*", ASEE Gulf-Southwest Conference, March 2002

[33] Knuth, D.E., "*Art of Computer Programming Volume 2 / Seminumerical Algorithms"*, Addison Wesley, 1969

[34] Vanstone, S., Menezes A., "*Introduction to Elliptic Curve Cryptograph Technology",* New York *:* Springer-Verlag, 2002.

[35] V. Gupta, S. Gupta, and S. Chang, "Performance Analysis of Elliptic Curve Cryptography for SSL," *ACM Wksp. Wireless Security*, *Mobicom 2002*, Atlanta, GA, Sept. 2002,

[36] Stalings W., "*Cryptography and Network Security Second Edition*", Prentice Hall, 1997

[37] Silverman, J.H., "Advanced Topics in the Arithmetic of Elliptic Curves", Springer, 1994.

[38] Baki A., "*The Use of Elliptic Curves in Cryptography",* The Graduate Scool of Natural and Applied Sciences of Middle East Technical University in Partial Fulfillment For The Degree of Master of Science in Mathematics, 1994.

[39] Soğukpınar İ., "*Eliptik Eğri Kriptografi*", www. gyte.edu.tr/bilgmuh

[40] Dalkılıç G., Öztaşır G., "*Eliptik eğri Şifrelemesine Karşı Ataklar*" Ağ ve Bilgi Güvenliği Ulusal Sempozyumu-ABG2005, İstanbul-TÜRKİYE, Haziran-2005

[41] Yerlikaya T. and Buluş, E. "*A Digital Signature Application With Elliptic Curve Cryptography*", 1st International Symposium on Information ISIT2005, Girne-KIBRIS

[42] Yerlikaya T., Buluş, E., Arda D., "*Eliptik Eğri Şifreleme Algoritması Kullanan Dijital İmza Uygulaması*", Ağ Ve Bilgi Güvenliği Ulusal Sempozyumu-Abg2005, İstanbul-Türkiye, 2005

[43] Koltuksuz A., "*Cryptography in Action*", ISCIS'99, 1999

[44] Koblitz N., "*Elliptic Curve Cryptosystems*", Math Comp Vol. 48, 203-209,1987.

[45] Koblitz N., Vanstone, S. and Menezes A., "*The State of Elliptic Curve Cryptography*", Designs, Codes and Cryptography, 19, 173–193, 2000

[46] Elliptic Curve Cryptography http://en.wikipedia.org/wiki/Public-key_cryptography

[47] Başaran A.S., "*Public-Key Cryptosystems Using Elliptic Curves"*, The Middle East Technical University In Partial Fulfillment of the Requirements for the Degree of Master of Science in The Department of Electrical and Electronics Engineering, 1999.

[48] Okhovat N. , "*Cryptanalysis of Elliptic Curve Cryptosystems"*, In Partial Fulfillment of the Requirements for the Degree of Master of Science in The Department of Electrical and Electronics Engineering, 2001

[49] Saldamlı G., "*Discrete Logarithm Problem on Elliptic Curves"*, The Middle East Technical University In Partial Fulfillment for the Degree of Master of Science In Mathematics, 2000

[50] Scutt A.Vanstone., "*Next Generation Security For Wireless: Elliptic Curve Cryptography*". Certicom

[51] J. Pollard, "*Monte Carlo Methods For İndex Computation Mod P*", Mathematics of Computation, 32 , 918-924, 1978.

[52] Pollard Rho Algorithmwww. Csh.rit. edu. Pat/math/rho

[53] Yerlikaya T., "*Şifreleme Teknikleri ve Güncel Uygulama Olanakları*", Yüksek Lisans Tezi Trakya Üniversitesi, 2002

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

[55] FIPS 46-3, "*Data Encryption Standard*", Federal Information Processing Standart (FIPS), Publication 46-3, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., October 25, 1999

[56] Matsuı M., 1993, "*Linear Cryptanalysis Method for DES Cipher*", Advances in Cryptology –EUROCRYPT'93, 386-397.

[57] Stinson D. R., "*Cryptography: Theory and Practice Second Edition*", CRC Press, 2002.

[58] Trappe W., Washington L., "*Introduction to Cryptography with Coding Theory*", 0-13-061814-4, 2002

[59] Matsui M., "*Linear Cryptanalysis Method for DES Cipher"*, Advances in Cryptology - Eurocrypt '93, Springer-Verlag, pp. 386-397, 1994.

[60] BİRİNCİ F., "*Güvenlik Protokollerinin Tasarımı ve Analizi*", Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü, 2002

[61] GÜNCAN M., "*Kimlik Tabanlı Kriptosistemler ile Güvenli Veri Aktarımı*", Yüksek Lisans Tezi, İTÜ., 2002

[62] Şen Ş. "*İndirgenmiş Spn (Substıtutıon Permutatıon Network) Algoritması İçin Lineer Kriptanaliz Uygulaması*" , Yüksek Lisans Tezi Trakya Üniversitesi, 2004

[63] Daemen J., Rijmen V., "*The Block Cipher Rijndael*, *Smart Card Research and Applications*", LNCS 1820, Springer-Verlag, pp. 288-296, 2000

[64] Daemen J., V. Rijmen, "*AES Proposal: Rijndael"*, First Advanced Encryption Conference, California, 1998.

[65] Schneıer B., Kelsey J., Whıtng D., Wagner D., Hall C., Ferguson H., "*Performance Comparison Of AES Submission",* 1999.

[66] "*Nist Institude of Standards on Technology", www.nist.gov*

[67] Yerlikaya T., Buluş E., "*AES Finalistlerinin Karşılaştırılması - RSA Şifreleme Algoritmasının İncelenmesi ve Kriptanalizi*", 20. Türkiye Bilişim Kurultayı, İstanbul-TÜRKİYE, Eylül-2003

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

[69] Patel D.,"*The AES Winner*",S V Rgional Engineering Collage, Indiana,2000

[70] Yerlikaya T., Buluş E., Arda D., *"AES Aday Şifreleme Algoritmalarının Yazılım Ve Donanım Performans Karşılaştırılması Ve Uygulamaları*", Elektrik Elektronik Bilgisayar Mühendisliği Sempozyumu (ELECO 2004), Bursa-TÜRKİYE, Aralık-2004.

[71] Şahin A., Buluş E., Sakallı M.T., "*Gri Seviye Resimler Üzerinde Rasgele Lsb Yöntemini Ve Sayı Teorisini Kullanarak Bilgi Gizleme Ve Steganaliz*", Akademik Bilişim Konferansları 2006-Ab2006, Denizli-Türkiye, Şubat-2006.

[72] Petitcolas F.A.P., Anderson R.J., Kuhn M.G., "*Information Hiding–A Survey*", Proceedings of the IEEE, Special Issue on Protection of Multimedia Content, 87(7):1062-1078, July 1999.

[73] Sellars D., "*An Introduction to Steganography*", Student Papers, 1999. http://www.cs.uct.ac.za/courses/CS400W/NIS04/papers99/dsellars/index.html

[74] Sağıroğlu Ş., Tunçkanat M., "*Güvenli İnternet Haberleşmesi İçin Bir Yazılım: Türksteg*", http://mf.erciyes.edu.tr/turksteg/, 2002.

[75] Şahin A., Buluş E., Sakallı M.T., "*24-Bit Renkli Resimler Üzerinde En Önemsiz Bite Ekleme Yöntemini Kullanarak Bilgi Gizleme*", Trakya Üniversitesi Fen Bilimleri Dergisi, Edirne-TÜRKİYE, 2005

[76] Farid H., "*Steganography: The Art And Mathematics Of Hiding İnformation*", Teaching Notes, Summer 2003.

[77] Elke F., "*Steganography Preserving Statistical Properties*", F.A.P. Petitcolas (Ed.): IH 2002, LNCS 2578, pp. 278–294, Springer-Verlag Berlin Heidelberg, 2003

[78] Westfeld A., Pfitzmann A., "*Attacks on Steganographic Systems*", Information Hiding. Third International Workshop, IH'99, Dresden, Germany, September/October, 1999, Proceedings, LNCS 1768, Springer-Verlag Berlin Heidelberg, 2000

[79] Johnson N. F., Jajodia S., "*Exploring Steganography*", Seeing the Unseen, February 1998, http://www.jjtc.com/pub/r2026.pdf

[80] Sayood K., "*Introduction to Data Compression*", Morgan Kauffman Publishers, Inc. 340 Pine Street, Sixth Floor, San Francisco, CA 94104-3205, USA, 1996

[81] Grgic S., Grgic M., Mrak M., "*Reliability of Objective Picture Quality Measures*", Journal of Electrıcal Engineering, Vol. 55, No. 1-2, 3-10, 2004

ÖZGEÇMİŞ

19.12.1977 tarihinde Edirne'de doğdu. İlk öğrenimini Keşan/Edirne'de, orta ve lise eğitimimi Edirne Anadolu Lisesi'nde tamamladı. 1995 -1999 yılları arasında Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü'nde öğrenim gördü. 1999 yılında Trakya Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans'a başladı.. Aynı yıl Trakya Üniversitesi Mühendislik- Mimarlık Fakültesi Bilgisayar Mühendisliği Bölümü'nde Araştırma Görevlisi olarak çalışmaya başladı. 2002'de Yüksek Lisan eğitimini tamamlayıp, aynı yıl Doktora eğitimine başladı. Halen Bilgisayar Mühendisliği Bölümlünde araştırma görevlisi olarak çalışmaktadır.