Kriptografi ve Bilgisayar Güvenliği Ders Notları(içindekiler)

1 VERİ VE AĞ GÜVENLİĞİNE GİRİŞ(INTRODUCTION TO DATA AND NETWORK SECURITY)

Bilgisayarlaşmanın artmasıyla birlikte, dosyaları ve bilgisayarda saklanan diğer bilgileri korumak gerektiği açıktır. Özellikle, zaman-paylaşımlı ve halka açık iletişim sistemleri gibi paylaşılmış sistemlerde veri güvenliği daha da önemlidir. Veriyi korumak ve saldırganları engellemek için tasarlanmış olan sistem ve araçların genel adı Bilgisayar Güvenlik Sistemidir.

İkinci ana konu, dağıtık sistemler ve son kullanıcının terminali ile bilgisayar arasındaki veri taşıyan haberleşme olanaklarının güvenliğe etkileridir. Ağ güvenliği tedbirleri verinin iletimi sırasında onun korunmasını esas alır. Gerçekte ağ güvenliği kavramı, bütün iş yerleri, devlet ve akademik kuruluşlar veri işleme birimlerini birbirlerine iletişim ağı ile bağladıkları için ortak bir ağ ortaya çıkar ki bunda birbirine bağlı ağlar adı verilir. Bu durumda koruma, ağ'daki bütün birimleri kapsar.

1.1 Bazı Güvenlik Tecavüzleri

• Kullanıcı A , Kullanıcı B' ye bir dosyayı transfer eder. Dosya, bozulmadan korumayı gerektiren hassas bilgileri(Ödeme bordrosu gibi) içermektedir. Dosyayı okumaya yetkili olmayan kullanıcı C, iletimi gözleyebilir ve iletim sırasında, dosyanın bir kopyasını alabilir.

• Bir ağ yöneticisi olan D, kendi yönetimindeki bilgisayar E' ye bir mesaj gönderir.Gönderilen mesaj, E' de bir grup kullanıcının bilgisayar erişim yetkilerinin güncellemesini içerir. Kullanıcı F, mesajı alıp,içeriğini değiştirerek, D'den geliyormuş gibi E' ye gönderir. E' de bu şekliyle kullanıcıların yetkilendirilmelerini günceller.

• Kullanıcı F, aldığı bir mesajı değiştirmek yerine kendi mesajını hazırlayarak sanki D'den geliyormuş gibi E' ye gönderir. E aldığı bu mesaja göre yetkilendirme dosyasını günceller.

• Farklı işlemler için ,müşteriden geliyormuş gibi borsa aracısına gönderilen bir mesaj ile para kaybı'na neden olunur ve müşterinin mesaj göndermesi engellenebilir.

1.2 Saldırılar servisler ve Mekanizmalar.

1. **Güvenlik saldırısı:** Bir kuruluşun bilgi güvenliği saygınlığını azaltır. Engelleme, Dinleme, Değiştirme ve yeniden oluşturma olarak 4 sınıf saldırı vardır.
2. **Güvenlik Mekanizması:** Bir güvenlik atağının anlaşılması, korunma veya onarımdır.
3. **Güvenlik Servisi:** Veri işleme sistemi ve kuruluşun bilgi iletim sisteminin güvenliğini artırma servisidir. Servis güvenlik saldırılarını engeller ve servis sağlamak için çeşitli güvenlik mekanizması kullanır.

1.3 Güvenlik Servis özellikleri aşağıda açıklanmıştır.

- **Gizlilik:** İletilen verinin pasif saldırılardan korunması. Diğer bir konu trafik akışının analiz edilmekten korunması. Bir saldırganın kaynak ve hedef arasında trafiği izlemesi önlenir.
- **Yetkilendirme**: Bu servis, haberleşmenin yetkili kişilerce yapılmasını sağlar. İkaz veya alarm gibi tek bir mesaj durumunda, yetkilendirme servisininin fonksiyonu, alıcıya mesajın kaynağı konusunda güven vermektir.
- **Bütünlük:** Mesajın bütünlüğünü sağlar. Mesajın tamamının değişmemesini temin eder.
- **İnkar edilememe**: Gönderici veya alıcının iletilen bir mesajı inkar etmemesini sağlar.

- **Erişim Denetimi:** Erişim denetimi ağ güvenliğinde, host sistemlere ve uygulamalara haberleşme bağlantıları ile erişimi sınırlandırır. Bu denetimi sağlamak için, her bir kişiye erişim hakkı verilmelidir.
- **Kullanıma hazırlık:** Saldırıların bir kısmı kullanılabilirliğin azalması veya kaybolmasına neden olabilir. Saldırıların bir kısamı iyi niyetli olabilir, oysa bir kısmı sistemin kullanılabilirliğini engeller. Bu servis kullanılabilirliğin sürekli olmasını sağlamaya yöneliktir.

1.4 Güvenlik mekanizmaları

Bilgi ve ağ güvenliğini sağlamak için birçok mekanizma mevcuttur. Bunlar kriptografik teknikler, şifreleme benzeri transformasyonlar sıkça kullanılan tekniklerdir.

1.5 Saldırılar

Bilgi sistemini saldırılardan korumak için saldırıları tanımak gerekir. Bu kapsamda tehdit(threat) ve saldırı(attack) termilerini kısaca açıklamak gerekir. **Tehdit**, belirli durum, yetenek, veya olay olduğu anlarda güvenlik foksiyonunun yerine getirilmesini engelleyen potansiyel bir güvenlik bozucusu olduğu halde; **saldırı**, sistemin güvenlik servislerini etkisiz hale getirmeyi amaçlayan akıllı bir tehditten üretilen ani bir hücumdur.
Bazı örnek saldırılar aşağıda verilmiştir.

Bilgilere yetkisiz erişimin elde edilmesi
Başka bir kullanıcının yetkilerini alarak onun yerine geçme
Saldırganın yasal lisansını genişletme
Saldırganın kendisini haberleşme yapan kullanıcıların arasına yerleştirmesi
Haberleşme hattının dinlenilmesi
Haberleşmenin engellenmesi
Saldırgan tarafından oluşturulan diğer bir kullanıcıya ait bilgilerin alındığını açıklamak
İletilen bilgilerin içeriğinin değiştirilmesi.

1.6 OSI Güvenlik Mimarisi

Bilgi güvenliğinde sistematik bir yaklaşım olarak X.800 OSI güvenlik mimarisi, yöneticilerin güvenlik oraganizasyonlarını düzenlemeleri için önemli bir yaklaşımdır. OSI yaklaşımı güvenlik servisleri, mekanizmalar ve saldırılara yoğunlaşmıştır.
Güvenlik Servisleri
Kimlik Doğrulama(Authentication)
Erişim Denetimi(Access Control)
Veri Gizliliği(Data Confideality)
Veri Bütünlüğü(Data Integrity)
İnkar edememe(Nonrepudation)

1.7 Güvenlik Mekanizmaları

X.800 OSI güvenlik mimarisinde mekanizmalar iki grupta toplanmıştır.,
Kendine özgü güvenlik mekanizmaları
Şifreleme, Sayısal imzalar, Erişim dentimi, Veri bütünlüğü, Kimlik doğrulama, Trafik analizini önleme, Yönlendirme denetimi ve noter makamı kullanılması

Kendine özgü olmayan güvenlik mekanizmaları
Güvenli fonksiyonellik, Güvenlik etiketi, Olay ortaya çıkartma, Güvenlik denetleme izleme, Güvenlik geri kazanımı

Güvenlik saldırıları

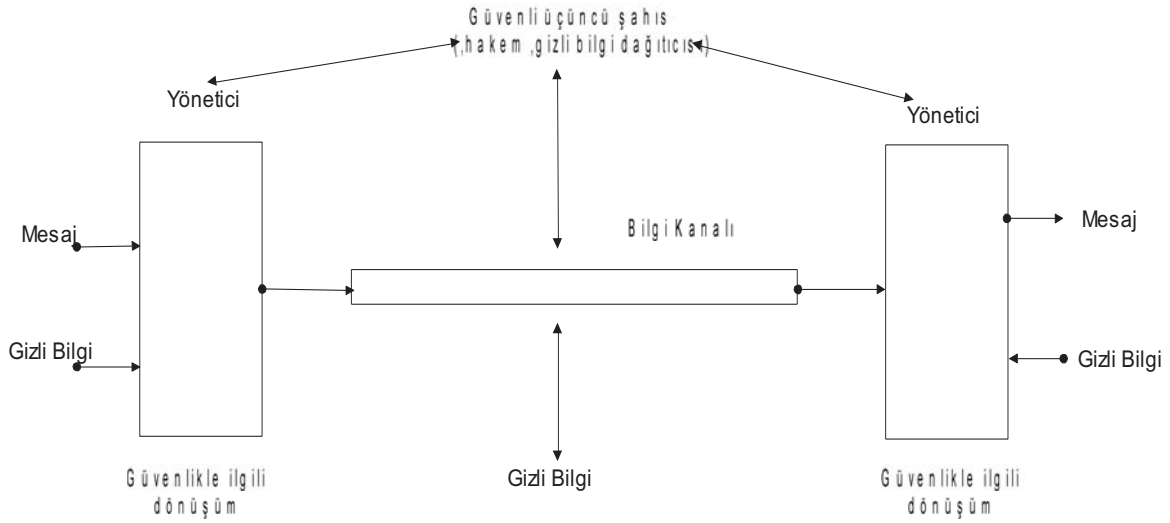
X.800 mimarisinde güvenlik saldırıları pasif ve aktif saldırılar olmak üzere iki türlüdür.
Pasif saldırılar, mesaj içeriğinin ifşa edilmesi ve trafik analizidir. Veri içeriği değiştirilmediği için pasif saldırıları ortaya çkartmak çok güçtür. Bu saldırılardan korunmak , anlamaktan daha uygun çözümlerdir.

Aktif Saldırılar, saldırganın kimliğini gizlemesi(masquarade), geri gönderme(replay), Mesajın değiştirilmesi(modification of message) ve servis durdurma(denial of service) dir.
Aktif saldırılar pasiflere göre zıt özelliktedirler. Aktif saldırılar tespit edilebilirler ve karşı önlem alınabilirler. Buna karşı aktif saldırıları tamamen önlemek çok zordur.

1.8 Ağ Güvenliği için bir model

Ağ güvenliğinde genel bir model şekil 1.1'de gösterilmiştir. Gönderici ve alıcı mesajları gizli olarak iletirken ,güvenli bir üçüncü şahıs gizli bilgilerin dağıtıcısı olarak hizmet vermekte, her iki taraf arasında noter görevi de görmektedir.
Bu genel güvenlik mimarisi, güvenlki servislerinin tasarımında dört temel işi gösterir.

1. Güvenlik ilişkili dönüşümler için bir algoritma tasarımı
2. Algoritma ile kullanılacak gizli bilginin üretimi
3. Gizli bilginin dağıtımı ve paylaşımı için yöntem geliştirme
4. Güvenlik algoritmasını ve güvenlik servisini sağlayacak gizli bilginin kullanımını sağlayacak bir protokol belirleme.



Şekil-1.1. Ağ Güvenliği için Model

2 KRİPTOSİSTEMLER VE SİMETRİK ŞİFRELEME/ DEŞİFRELEME(CRYPTOSYSTEMS AND SYMMETRİC ENCRYPTİON/DECRYPTİON)

Kimlik doğrulama ve şifreleme, verinin emniyetini sağlamaya yarayan birbiriyle bağlantılı iki teknolojidir. Kimlik doğrulama, haberleşmede her iki tarafta bulunanların ne söylüyorlar ise onun doğru olmasını sağlama sürecidir. Şifreleme ise iletişim sırasında verinin hem güvenliğini sağlamak hem de değiştirilmesini önlemeye yönelik işlemlerdir.

2.1 Güvenliğin geliştirilmesi ihtiyacı.

1970'li yıllarda IP version4 Internet'te kullanılmaya başlanınca ağ güvenliği ön planda bir konu değildi. Bu nedenle IP, bütün veriyi açık metin şeklinde gönderir. Bunun anlamı, eğer gönderilen paketler dinlenirse hem içeriği öğrenilebilir hem de değiştirilebilir. Ağ analizi yapan bir uç noktadaki saldırgan bu analizler sonucunda, hem oturumları öğrenebilir, hemde veri paketlerinin içeriklerini değiştirebilir. Aşağıdaki protokoller açık metin(Clear text) ileten protokollerdir.

- FTP Doğrulama açık metindir.
- Telnet Doğrulama açık metindir
- SMTP posta mesajlarının içeriği açık metin olarak dağıtılır.
- http Sayfa içeriği ve formlardaki bilgilerin içeriği açık metin olarak gönderilir.
- IMAP Doğrulama açık metindir
- SNMPv1 Doğrulama açık metindir

2.2 Ağ Üzerinde Yapılan Saldırı Türleri

1. İfşaat(Disclosure) Mesaj içeriğinin herhangi birisine verilmesi veya Uygun kriptografik anahtara sahip olmama

2.Trafik Analizi: Ağdaki trafik akışının analiz edilmesi.Bağlantı esaslı uygulamalarda, bağlantının sıklığı ve süresi. belirlenebilir. Bağlantı esaslı veya bağlantısız ortamda, bağlantılardaki mesajların sayısı ve uzunluğu belirlenebilir.

3. Gerçeği gizleme (Masquerade) Hileli bir kaynaktan ağ'a mesaj ekleme. Bu işlem muhalif tarafından yetkili bir kullanıcıdan geliyormuş gibi görünen mesajların oluşturulmasını içerir.

4.İçerik Değiştirme(Content Modification): Ekleme, silme, sırasını değiştirme veya içeriğini değiştirme yöntemleriyle mesajın değiştirilmesi.

5.Sıra Değiştirme(Sequence Modification): Ekleme silme ve yeniden sıralama ile mesajın sırasında değişiklik yapma.

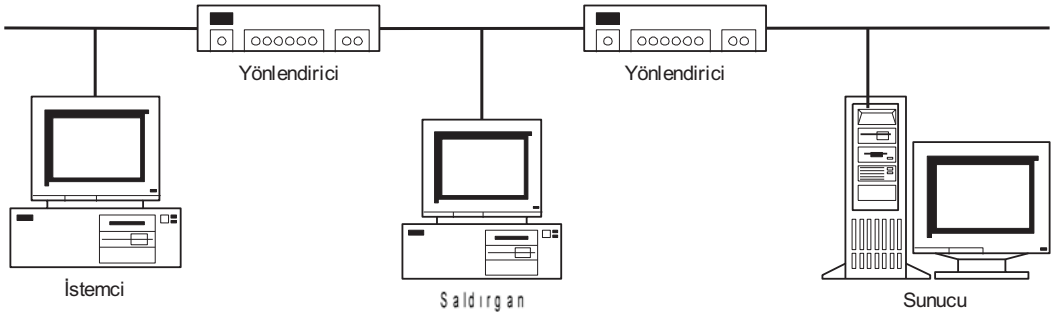
6.Zamanlamayı Değiştirme(Timing Modification): Mesajları geciktirme veya yeniden yollama. Bir bağlantı orijinli uygulamada bütün oturum veya mesajların bir kısmı ya önceki geçerli bir oturumun bir tekrarlanan sırası veya sıradaki kısmi mesajlar olarak geciktirilebilir veya tekrar gönderilir.

7.İnkarcılık (Repudiation): Alınan mesajın varış tarafından inkarı veya gönderilen mesajın kaynak tarafından inkarı.

2.3 İyi Doğrulama Gereklidir.

İyi doğrulama gerektiği açıktır. Açık metin olarak logon bilgisini ileten bir protokol ile sunucuya erişen bir istemcinin logon ve password bilgisini bir saldırgan elde edebilir. Bu ise saldırganın o birim yerine geçmesi demektir. İyi doğrulamanın bir başka sebebi bir servise erişen kaynak

istemcinin veya sunumcunun doğrulanmasıdır. Aynı zamanda hostun iletişim oturumu sırasında değişmediğinden emin olunması gereklidir. Bu tip bir atağa oturum korsanlığı adı verilir.



Şekil 2.1. Oturum Korsanlığı

2.3.1 Oturum Korsanlığı

Şekil 6.1'deki ağ üzerinde bir istemci, sunumcu ile haberleşme yapmaktadır. İstemci sunumcu tarafından doğrulanmış ve erişimi yönetici seviyesinde sağlanmıştır. Kendini istemci ile sunumcu arasındaki ağ segmentinde gizlemiş bir saldırgan oturumları gözlemleyebilir. Bu saldırgana haberleşme yapan uçların port numaraları ve sıra numaralarını öğrenme imkanı verir. Bunları öğrenen saldırgan yöneticinin oturumunu kullanarak yönetici seviyesinde yeni hesap açmayı gerçekleştirebilir.(man in the middle attack)

2.3.2 Varışın Doğrulanması

Kaynağın iletişimden önce ve sonra doğrulanması gerektiği açıktır. Ancak varışın(sunucu) doğrulanması da gereklidir.

C2MYAZZ, Sunucu aldatması için kullanılan iyi bir yardımcıdır. Windows95'in kullanıcı doğrulanması sırasında pasif olarak bekler. Bir logon işlemi olduğunda,istemciye LANMAN doğrulama bilgisi gönderir. İstemci ise bilginin sunucudan geldiğini sanarak logon ve şifre bilgisini gönderir. Böylece kullanıcı şifresi öğrenilmiş olur.

DNS Poisoning

DNS te bir hostun adresi yerine rastgele başka adres bilgisinin yayınlanması işlemidir. Saldırgan trafiği böylece başka sunumcuya yönlendirir.Sayısal sertifikalar kullanılmadığı sürece istemci ve sunumcuların yerine bir saldırganın geçebilmesi mümkün olabilmektedir. Bunu önemenin en emin yolu verileri şifreleyerek iletmektir.

2.4 Kriptolama

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.

Bir şifreli haberleşme için;
1. Şifreleme algoritması (E)
2. Deşifreleme algoritması (D)
3. Bir anahtar bilgisine(K),

ihtiyaç vardır.

2.4.1 Terminoloji ve Notasyon

Kriptoloji, latince gizli anlamına gelen *kryptos* ve yine latince sözcük anlamına gelen *logos* kelimelerinin birleşiminden oluşan gizli ve güvenli haberleşme bilimidir. Kriptoloji temelde iki kısımda incelenir; bunların birincisi kritik bilgilerin yetkisiz kişi ve/veya kurumlardan korunması amacıyla geri dönüşümü ümkün olarak anlaşılmaz hale getirilmesi yani şifrelenmesi için kripto sistemlerinin tasarlanması demek olan **kriptografi** bilimidir. İkinci kısım ise kodlanmış veya şifrelenmiş olan gizli bilgilerin bulunmasına yönelik çalışmaların yapılması demek olan **kript analiz** bilimidir.
Kriptolojide daha çok bilginin güvenliği ve gizliliği üzerinde durulacaktır. Bunun yolu genellikle bilgilerin veya mesajların bir takım transformasyonlara tabi tutulmasıyla olur. Daha sonra bu bilgi topluluğunun tekrar elde edilebilmesi için şifreli metne aynı transformasyonların tersi uygulanır. Orijinal mesaj burada kısaca **m** harfiyle, mesajı transformasyona tabi tutma işlemi **şifreleme** adıyla, ortaya çıkan anlaşılmaz metin ise kısaca **c** harfi ile gösterilecektir**.** Ters transformasyon işleminin şifreli metne uygulanıp tekrar orijinal mesajı elde etmeye yönelik yapılan işleme ise **deşifreleme** adı verilir.

2.5 Temel Kavramlar

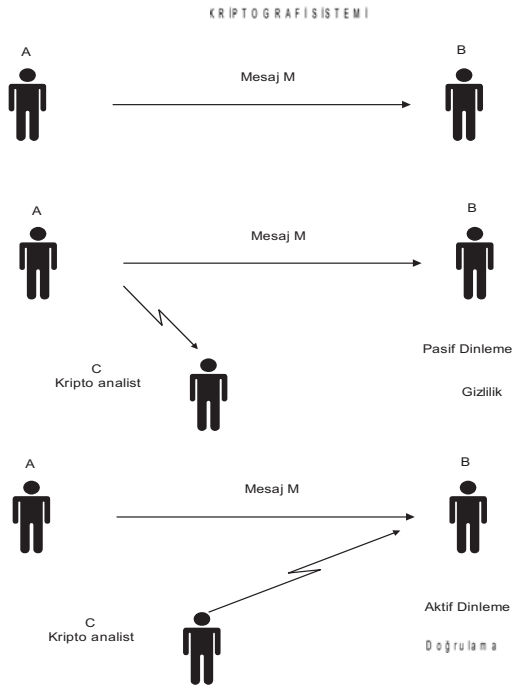
Kriptografi(cryptography): Anlaşılır bir mesajı anlaşılmaz şekle dönüştürme ve mesajı tekrar eski orijinal haline geri dönüştürme prensipleri ve yöntemlerini içeren sanat veya bilimdir.
Açık metin(plaintext): Anlaşılır orijinal metin
Şifreli metin(ciphertext) : Dönüştürülen metin
Şifreleyici(cipher) : Anlaşılır bir metni, yerlerini değiştirme ve/veya yerine koyma yöntemlerini kullanarak anlaşılır bir metni anlaşılmaz şekle dönüştürmek için kullanılan bir algoritma.
Anahtar(key) : Sadece gönderici ve alıcının bildiği şifreleyici tarafından kullanılan kritik bilgiler
Şifreleme(encipher (encode)) : Açık metni bir şifreleyici ve bir anahtar kullanarak şifreli metne dönüştürme süreci
Deşifreleme(decipher (decode)): Şifreli metni bir şifreleyici ve bir anahtar kullanarak açık metne dönüştürme süreci
Kriptanaliz(cryptanalysis) : Bilgi ve anahtar olmaksızın anlaşılmaz mesajı anlaşılır mesaj olarak geri dönüştürme prensipleri ve yöntemleridir. Aynı zamanda kod kırma(**codebreaking**) olarak da adlandırılır.
Kriptoloji(cryptology) :Kriptografi ve kriptanalizin her ikisi(şekil 2.2)
Kod(code): Anlaşılır bir mesajı bir kod kitabı kullanarak anlaşılmaz şekle dönüştürme için bir algoritma

Şifreleme(Encryption) $c = E_K(m)$

Deşifreleme(Decryption) $m = D_K(c)$

E_K, kriptografik sistem olarak bilinen transformasyon ailesinden seçilir.
Anahtar denilen K parametresi anahtar uzayından seçilir

Diğer bir deyişle, şifreleme işlemi $E_K(m)=c$ fonksiyonunu sağlayan bire-bir, bir fonksiyondur. E_K fonksiyonunun tersi olan D_K fonksiyonu ise, $D_K(c)=m$ şartını sağlayan deşifreleme işlemini gerçekleştirir. Burada yer alan bütün transformasyon işlemleri tersinir olduğundan dolayı açık bilginin şifreli bilgiden direkt olarak elde edilmesini önlemek için *E* ve *D* algoritmalarının gizli tutulması düşünülebilir. Şifreleme ve deşifreleme algoritmalarının herhangi bir şekilde yetkisiz kişilerin eline geçmesine karşı yalnızca mesajlaşacak kişilerin bilebileceği bir **anahtar** bilgisi, *K*, kullanılmalıdır. Dolayısıyla, mesajlaşmada önemli olan kriter kullanılan anahtarın gizliliği olacaktır. Sonuçta anahtar gizli tutulduğu halde algoritmalar açık olabilir.



Şekil2.2. Kriptografi Sistemi

2.6 Kripto sistemler

Kripto sistemlerinde kullanılan başlıca terimler kısaca şunlardır; ***A*** ile gösterilen **Alfabe** kavramı sonlu sayıda elemanlar kümesidir. Örneğin $A=\{0,1\}$ sık kullanılan ikili (binary) bir alfabedir. ***P*** ile gösterilen **Açık Metin Uzayı** (Plaintext Space) ise alfabeden alınmış sonlu sayıda eleman dizilerinden oluşur. Örneğin *P*, 0 ve 1 ler den meydana gelen bit dizilerini içerebilir. ***C*** ile gösterilen **Şifreli Metin Uzayı** (Ciphertext Space) ise yine *A* alfabesinden alınmış fakat *P* den farklı bir diziliş gösteren elemanlardan oluşur. ***K*** ise daha önce bahsettiğimiz **Anahtar Uzayını** (Key Space) ifade eder. Anahtar yine ***A*** alfabesindeki elemanların belli uzunluklarda bir araya gelmiş elemanlarından oluşur.

Tanım : Bir kriptosistem aşağıdaki şartları sağlayan (P,C,K,E,D) beşlisinden oluşur. Burada *E* şifreleme, *D* ise deşifreleme fonksiyonu veya algoritmasını gösterir.

$\forall\ k \in K$, $D_k \in D$ fonksiyonuna uyan bir $E_k \in E$ fonksiyonu vardır. Öyle ki;

$\forall\ E_k : P \to C$ ve $\forall\ D_k: C \to P$ ve her $x \in P$ için $D_k(E_k(x)) = x$

Kriptosistemler genel olarak aşağıdaki üç bağımsız özelliğe göre sınıflandırılırlar.

1. **Şifresiz metinden şifreli metne dönüşüm için kullanılan işlemlerin tipi:** Bütün şifreleme algoritmaları yerine koyma(substutition) ve yerini değiştirme(transposition) olmak üzere iki genel prensibe dayanır. Yerine koymada, şifresiz metindeki her bir eleman diğer bir elemana dönüştürülür, yerini değiştirme de ise, şifresiz metindeki elemanların yerleri değiştirilir.

2. **Kullanılan anahtarın sayısı:** Gönderici ve alıcı aynı anahtarı kullanırsa buna simetrik (tek anahtarlı, gizli anahtarlı, veya geleneksel) şifreleme, eğer gönderici ve alıcının her biri farklı anahtar kullanırsa buna asimetrik(iki anahtarlı, veya açık anahtarlı) şifreleme denir.

3. **Şifresiz metni işleme yöntemi:** Eğer giriş verisi, herbir adımda blok olarak işlenerek çıkış blok olarak elde edilirse blok şifreleme, giriş verisi dizi olarak sürekli şekilde işlenirse dizi şifreleme adı verilir.

2.6.1 Kriptolama güvenliği ve Kriptanaliz.

Şifrelenen metnin ne kadar güvenli olduğu ve çözümlenmesi için yapılacak saldırı tiplerinin neler olduğunun bilinmesi önemlidir. Geleneksel şifreleme yöntemlerine saldırı için iki adet genel yaklaşım mevcuttur.

Kriptanaliz: Kriptanalitik saldırılar, algoritmanın özelliği, şifresiz metnin genel karakteristiği hakkındaki bilgilere ve şifresiz metin–şifreli metin çiftinin bazı örneklerine dayanır. Bu saldırı sonucunda kullanılan anahtar veya şifresiz metin, algoritmanın eksikliklerine dayanılarak elde edilmeye çalışılır.
Deneme-Yanılma(Brute-Force Attack) saldırsı: Saldırgan mümkün olan bütün anahtar kombinasyonlarını, şifresiz metin elde edilene kadar şifreli metni çözmek için dener. Ortalama olarak bütün anahtar kombinasyonlarının yarısı başarılı bir saldırı için denenmelidir.

Şifreli metin için güvenlik bir sonraki paragrafta açıklanmıştır. Tablo 2.1'de ise Şifrelenen mesajı çözmek için yapılan saldırı tipleri ve kripto analistin neler bildiği gösterilmiştir.

Saldırı Tipi	Kriptoanalist'in bildiği
Sadece Şifreli Metne (ciphertext only)	Kriptolama algoritması Kodu çözülecek şifreli metin (istatistiksel Saldırı, brute force)
Bilinen Düz metin (known plaintext)	Kriptolama algoritması Kodu çözülecek şifreli metin Gizli anahtar ile şifrelenen bir veya daha fazla düz-şifreli metin çifti(Şifreye saldırı için kullanılır.)
Seçilen Düz metin (chosen plaintext)	Kriptolama algoritması Kodu çözülecek şifreli metin Kriptanalist tarafından seçilen açık metin, bununla birlikte açık metnin gizli anahtar ile üretilen şifreli hali
Şeçilen Şifreli metin (chosen ciphertext)	Kriptolama algoritması Kodu çözülecek şifreli metin Kriptanalist tarafından seçilen kuvvetle muhtemel şifreli metin ve karşılığı olan, gizli anahtar ile üretilen çözümlenmiş açık metin.
Şeçilen metin (chosen text)	Kriptolama algoritması Kodu çözülecek şifreli metin Kriptanalist tarafından seçilen açık metin, bununla birlikte açık metnin gizli anahtar ile üretilen şifreli hali Kriptanalist tarafından seçilen kuvvetle muhtemel şifreli metin ve karşılığı olan, gizli anahtar ile üretilen çözümlenmiş açık metin.

Tablo 2.1: Şifrelenen mesaja karşı yapılan saldırı Tipleri

2.6.2 Mutlak ve hesaplama güvenliği

İki farklı temel yöntem ile şifreler güvenli olabilir.

Mutlak güvenlik

- Bilgisayar gücü ne kadar fazla olursa olsun şifre hiçbir şekilde kırılamaz.

Hesaplamaya bağlı güvenlik

Bir şifreleme algoritması aşağıdaki kriterleri sağlıyor ise hesaplamaya bağlı güvenli(computationally secure) dır.

- Şifrenin kırılmasının maliyeti şifrelenmiş bilginin değerinden fazla ise
- Şifreyi kırmak için gereken zaman, bilginin yaralı ömründen fazla ise.

Hesaplamaya bağlı güvenlikte verilen bilgisayar gücü sınırları(örn. Evrenin yaşından daha fazla hesaplama zamanı gerekir gibi), içinde şifre kırılamaz.

Hesaplamaya bağlı güvenlik için şifreleme algoritması ve kullanılan anahtar uzunluğu önemlidir. Şifreleme algoritmasının kriptoanalist tarafından bilindiği kabul edilerek şifre uzunluğu ve bilgisayarın hesaplama gücüne bağlı olarak şifrelerin çözümleme süreleri Tablo 6.2'de gösterilmiştir. Çözümleme süresi için gerekli olacak zaman hesabı ortalama olarak alternatif şifre sayısının yarısı kadardır. Bilgisayar hesaplama gücünü ise paralel mimarili tasarım ile artırmak mümkün olmaktadır.

Anahtar Uzunluğu(bit)	Alternatif Anahtar Sayısı	1 çözümleme/μs hızında gereken zaman	10^6 çözümleme/μs hızında gereken zaman
24	$2^{24} = 1.6 \times 10^7$	2^{23} μs = 8.4 saniye	8.4 μsaniye
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 dakika	2.15 milisaniye
48	$2^{48} = 2.8 \times 10^{14}$	2^{47} μs = 4.46 yıl	2.35 dakika
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 yıl	10 saat
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} yıl	5.4×10^{18} yıl
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} yıl	5.9×10^{30} yıl
26 karakter permutasyonu	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} yıl	6.4×10^6 yıl

Tablo 6.2. : Anahtar uzunluklarına göre hesaplamaya bağlı güvenlik

2.7 Kriptografinin kısa Tarihçesi

2.7.1 Çok Eski(Ancient) şifreleyiciler

- En az 4000 yıl öncesine dayanır.
- Eski mısırlılar anıtlara yazdıkları resimli yazılarını şifrelemişlerdir.(Şekil 6.3)

Hieroglyphic encipherments of proper names and titles, with cipher hieroglyphs at left, plain equivalents at right

Şekil 2.3.

- Eski ibraniler kutsal kitaplarındaki belirli kelimeleri şifrelemişlerdir.
- 2000 sene önce Jul Sezar, şimdi Sezar şifresi olarak bilinen basit bir yerine koyma şifresi kullandı
- Roger Bacon 1200 lerde birkaç yöntem açıkladı.
- Geoffrey Chaucer çalışmalarında birkaç adet şifre kullandı
- Leon Alberti 1460 larda bir şifre tekerleği kullandı ve frekans analizinin prensiplerini açıkladı.

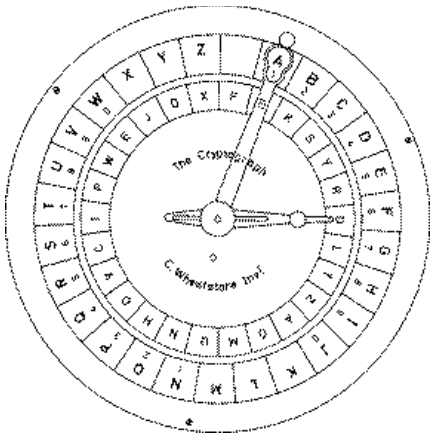
- Blaise de Vigenère 1855 de kriptoloji üzerine bir kitap yayınladı ve çoklu alfabe değiştirme sifresini açıkladı.
- Kullanımı ülkelerde özellikle diplomasi ve savaşlarda artmaktadır.

2.7.2 Makina Şifreleri

- 1790 larda geliştirilen **Jefferson cylinder**, herbiri rastgele alfabeli 36 adet disk ten oluşmaktaydı, disklerin sırası anahtarı oluşturmaktaydı, mesaj ayarlanınca diğer satır şifreyi oluşturmaktaydı.(Şekil 2.4)

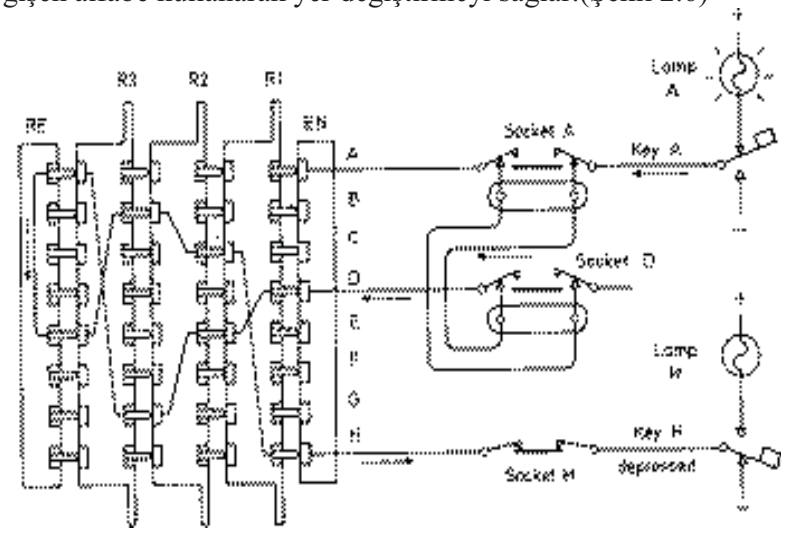
Şekil 2.4. Jefferson Cylinder

- **Wheatstone disc**, orijinal olarak 1817'de Wadsworth tarafından icat edildi, fakat 1860 da Wheatstone tarafından geliştirildi. Çoklu alfabeli şifreyi oluşturmak için merkezi olarak kullanılan tekerleklerden meydana gelmekteydi. (Şekil 2.5)



Şekil 2.5. Wheatsone Disk

- **Enigma Rotor makinası**, ikinci dünya savaşı sırasında çok kullanılan şifre makinalarının önemli bir sınıfını teşkil eder, içinde çapraz bağlantılı, bir seri rotordan meydana gelir, sürekli değişen alfabe kullanarak yer değiştirmeyi sağlar.(Şekil 2.6)



Şekil 2.6. Enigma Rotor Makinası

3 SAYI TEORİSİNE GİRİŞ

Bu bölümde kriptolama algoritmalarının matematik modellemesinde kullanılan modüler aritmetik kavramları üzerinde kısaca durulacaktır.

Grup Teorisi

Tanım(Grup): Her bir elemanın tersinin olduğu monoide (G, *) **grup** denir. Yani (G, *) çifti şu dört şartı sağlar:
(G_1) *, Kapalılık, Eğer a ve b $\in$ G ise a*b $\in$ G dir.
(G_2) *, G üzerinde birleşme özelliğine sahiptir. $\forall$a,b,c $\in$ G için, a*(b*c) = (a*b)*c dir.
(G_3) bir etkisiz eleman mevcuttur. $\forall$a $\in$ G için, a*e = e*a =a dır.
(G_4) G' nin her bir elemanının tersi mevcuttur.$\forall$a$\in$G için,G'de bir a' vardır ve a*a'= a'*a =e dır.

Bu bölümde ve bundan sonraki bölümlerde belirtilmemiş ikili işlemler içeren ifadeler yazarken * simgesini göz ardı edeceğiz. Sadece yanlış anlamalara imkan verecek iki ikili işlemi birbirinden ayırt etmek için kullanacağız. Örneğin x*y yerine xy yazacağız (ancak çarpma işlemi ile karıştırmamalıyız). Ayrıca aşağıdaki gibi x' in üslerini tanımlayacağız.

n$\in Z^+$ olmak üzere $x^n=x*x*\ldots*x$ (n tane)

ve x$\in Z^-$ olmak üzere $x^n=(x^{-1})^{|n|}=x^{-1}*x^{-1}*x^{-1}*\ldots*x^{-1}$.(n tane)

Ayrıca etkisiz elemanı da şu şekilde tanımlarız: x^0=e.
Herhangi bir (G,*) grubun en belirgin özelliği büyüklüğü yani grubun temelini oluşturan G kümesinin eleman sayısıdır. Buna (G,*) grubunun order'ı denir.
Tanım: (G,*) grubunun order'ı G kümesinin kardinalitesidir ve |G| şeklinde gösterilir.

Eğer bir grup, sonlu sayıda elemana sahipse sonlu grup, ve grubun order'i gruptaki eleman sayısıdır. Diğer durumda grup sonsuz gruptur.
Eğer bir grup aşağıdaki ilave koşulu sağlıyor ise **abealian** grup adı verilir.

(G_5) Komutatiflik. $\forall$a,b $\in$ G için, a*b = b*a dır.

Eğer H grubu G grubunun bir alt grubu ise $|H|$ değeri $|G|$ değerini böler. Böylece eğer G grubunun *düzeni* bir asal sayıysa G' nin tek alt grubu kendisidir. Bu durumda G grubu çarpmalı olarak yazılabilir.

Eğer G grubu çarpmalı olarak yazılabilirse ve $g \in G$ olmak üzere g sayısı G grubunun düzeni ise bu g sayısı $i \in \mathrm{N} \cup \{\infty\}$ ve $g^i = 1$ şartını sağlayan en küçük i değeridir. Burada $\forall\, j, l \in Z$:

$$g^j = g^l \quad \Leftrightarrow \quad j \equiv l \mod ord(g)\text{' dir.}$$

Tablo 3.1'deki Cayley tablosu ile tanımlanmış grubu ele alalım:

*	e	a	b	c
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

Tablo 3.1

Her bir elemanı n bir tamsayı olmak üzere a^n biçiminde yazabileceğimizden bu grup için a^1=a, a^2=b, a^3=c ve a^4=e 'dir. Verilen herhangi bir eleman için bu gösterim aynı değildir. Örneğin, $b=a^2=a^6=a^{-2}$ vs. yazabiliriz. Aslında kümenin her bir elemanını a' nın kuvvetleri biçiminde göstermek için sonsuz sayıda yol vardır. {e,a,b,c} 'nin her elemanı a^n biçiminde yazılabilir ve bu duruma a grubun bir üretecidir (generator) denir.

G grubunun altgrubu olan tüm gruplar *g* elemanının bir üssüdür ve <*g*> ifadesiyle gösterilirler. Eğer <*g*> = *G* ise *g* sayısı *G* grubunun **üreteci** (jeneratörü) olur. Bir üreteci olan tüm gruplara **devirli grup** (cyclic group) adı verilir.

G grubunun düzeni *p* asal sayısı ise grup içerisinde yer alan 1 dışındaki tüm sayılar *G* grubunun üreteci olur. Diğer bir deyişle <*g*> nin düzeni 1 veya *p* sayısı olur.

Doğal olarak, diğer başka elemanlar da grubun üretecimidir? sorusu aklımıza gelir. c elemanının da bir üreteç olduğunu fakat n çift ise b^n=e ve b tek ise b^n=b olduğundan b 'nin bir üreteç olmadığını söyleyebiliriz. En az bir tane üretece sahip gruplara halka denir.

Halkalar: {R,+,X) ile gösterilen bir *R* halkası, $\forall a,b,c \in R$ için aşağıdaki aksiyomları toplama ve çarpma ikili işlemleriyle sağlayan bir elemanlar kümesidir.
(G_1-G_5) R, toplama altında bir abelian grup tur.

(H_1) Çarpma altında kapalılık, Eğer a ve $b \in R$ ise $ab \in R$ dir.
(H_2) Çarpma ile birleşme özelliğine sahiptir. $\forall a,b,c \in R$ için, a(bc) = (ab)c dir.
(H_3) Dağılma kuralı, $\forall a,b,c \in R$ için, a(b+c) = ab + ac , (a+b)c = ac +bc dir.

Eğer bir halka aşağıdaki koşulu sağlıyor ise komutatif halkadır.
(G_4) Çarpmada Komutatiflik. $\forall a,b \in R$ için, ab = ba dır.

Eğer bir komutatif halka aşağıdaki aksiyomları sağlıyor ise integral domain dir.
(H_5) Çarpımsal etkisiz eleman. $\forall a \in R$ için, a1 = 1a =a dır.
(H_6) Sıfır bölen olmaması $\forall a,b \in R$ ve ab=0 ise ya a=0 veya b=0 dır.

Alanlar(Field) : {F,+,X) ile gösterilen bir *F* alanı, $\forall a,b,c \in F$ için aşağıdaki aksiyomları toplama ve çarpma ikili işlemleriyle sağlayan bir elemanlar kümesidir.
(G_1-H_6) F, G_1 den G_5 'e ve H_1den H_6 ya aksiyomları sağlayan bir integral domain dir.
(H_7) Çarpımsal invers . $\forall a \in F$ için (sıfır hariç) F'de bir a^{-1} vardır ve $aa^{-1}=(a^{-1})a$ =1dir.

Esasında bir **alan**, kümenin dışına çıkmaksızın, toplama çıkartma çarpma ve bölme yapılabilen bir kümedir. Bölme $a/b = a(b^{-1})$ kuralı ile tanımlanır.

3.1 Modüler Aritmetik

Modüler aritmetik "saat aritmetiğidir"

Tanım *a, r* ve *n* tam sayıları ve $n \neq 0$ şartı için, eğer *a* ve *b* nin farkı *n* 'in *k* katı kadarsa bu şu şekilde gösterilebilir:

$$a = k \,.\, n + r$$

burada; a ve n pozitif tamsayılardır. Bu bağıntıyı sağlayan k ve r değerlerini her zaman bulmak mümkündür. kn'den a ya olan uzaklık r'dir ve kalan(residue) olarak adlandırılır. Veya eğer a ve

n pozitif tamsayı iseler, a mod n , a ,n ile bölündüğünde kalan olarak tanımlanır. Böylece herhangi a tamsaysı için,

a= [a/n]x n + a ***mod*** n her zaman yazılabilir. (Örn: 11 mod 7 = 4)

a ve b iki tamsayısı eğer a ***mod*** n =b ***mod*** n iseler benzer modulo n olarak tanımlanır ve $a \equiv b \bmod n$ olarak yazılabilir.

Bölenler: Eğer sıfır olmayan bir b ve m tamsayısı için a=mb şeklinde yazılabiliyorsa b, a'yı böler denir. Böyle bir bölünebilirlik var ise kalan sıfırdır. b|a notasyonu b'nin a'yı kalansız bölebildiğini belirtmek için sıkça kullanılır. Aşağıdaki bağıntılar vardır.

- Eğer a|1 ise a= ±1 dir.
- Eğer a|b ve b|a ise a= ±b dir.
- Herhangibir $b \neq 0$ sıfırı böler.
- Eğer, b|g ve b|h ise , b|(mg +nh) herhangi m ve n tamsayıları için vardır.

Teorem a_1, a_2 ve n tam sayıları ve $n \neq 0$ şartı için,

$$(a_1 \ op \ a_2) \bmod n \equiv [(a_1 \bmod n) \ op \ (a_2 \bmod n)] \bmod n$$

denkliği gösterilebilir, burada *op*, " + " veya " * " şeklinde bir operatör olabilir.

- Bir a = b mod n eşitliği, a ve b aynı *n* ile bölündüğünde aynı kalanı verdiklerini ifade eder. Örnek,
  - 100 = 34 mod 11
  - Genellikle 0<=b<=n-1 dir.
  - 2mod7 = 9mod7
  - b 'ye a mod n 'nin kalanı denir.
- Tamsayı modulo n ile yapılan bütün aritmetikte bütün sonuçlar 0 ve n arasında olur.

3.1.1 Modül işleminin özellikleri

Modül işlemi aşağıdaki özelliklere sahiptir.

Eğer, n|(a-b) ise a ≡ b ***mod*** n dir.

a ≡ b ***mod*** n , b ≡ a ***mod*** n anlamına gelir.

a ≡ b ***mod*** n ve b ≡ c ***mod*** n , a ≡ c ***mod*** n anlamına gelir.

3.1.2 Modüler Aritmetik işlemleri

Toplama

(a+b) mod n = [(a mod n) + (b mod n)] mod n

Çıkartma

(a-b) mod n = [(a mod n) - (b mod n)] mod n

Çarpma

axb mod n = [(a mod n) x (b mod n)] mod n

- Tekrarlanan toplamdan türetilir
- Ne a ne de b sıfır değil iken a.b=0 olabilir
  - örnek 2.5 mod 10

Bölme

a/b mod n

- b nin tersi ile çarpmak gibidir: $a/b = a.b^{-1} \bmod n$
- eğer n asal ise $b^{-1} \bmod n$ vardır. $b.b^{-1} = 1 \bmod n$
  - örnek 2.3=1 mod 5 bu nedenle 4/2=4.3=2 mod 5 dir.

Özellikler :
n'den küçük olan pozitif tamsayıların kümesi Z_n aşağıdaki gibi tanımlansın.

$$Z_n = \{ 0,1,\ldots\ldots\ldots\ldots,(n-1)\}$$

Z_n kalanlar sınıfı olarak adlandırılır. Daha doğrusu, Z_n de her bir tamsayı bir kalan sınıfını temsil eder. [r] = { a : a bir tamsayı; öyleki ; a = r mod n dir.]
Z_n içersinde yapılacak modüler artimetik işlemleri Tablo 3.2'deki özellikleri Z_n deki tamsayılar ile sağlar. Z_n çarpımsal etkisiz eleman ile birlikte bir değiştirilebilen bir halka oluşturur.

Özellik	Açıklama
Değişme Kuralı (Commutative)	(a + b) mod n = (b + a) mod n (a x b) mod n = (b x a) mod n
Birleşme Kuralı (Associative)	[(a+b) +c] mod n = [a +(b + c)] mod n [(axb) x c] mod n = [a x(b x c)] mod n
Dağılma Kuralı (Distributive)	[ax(b + c)] mod n = [(axb) + (axc)] mod n
Etkisiz eleman (Identity element)	(0 + a) mod n = a mod n (1xa) mod n = a mod n
Toplamsal invers(-a)	$\forall a \in Z_n$ için ; bir b vardır öyleki ; a + b = 0 mod n dir.

Tablo 3.2.

- Aynı zamanda, indirgeme tamsayılar halkasından tamsayı modulo n 'lerin halkasına bir homomorfizm olduğu için, bir işlem ve sonra modulo n i indirgeyip indirgemeyeceği veya indirgedikten sonra yapacağı işlem seçilebilir.
  - a+/-b mod n = [a mod n +/- b mod n] mod n
  - (a.b) mod n =((a mod n).(b mod n)) mod n
- eğer n, p doğal sayısı olmaya zorlanırsa bu form bir **Galois Field modulo p** ve **GF(p)** ile gösterilir ve bütün tamsayı aritmetiğindeki normal kurallar geçerlidir.

3.2 GF(p) (Galois Field) şeklindeki sonlu alanlar.

Birçok kriptografik algoritmada sonlu alanlar önemli bir rol oynarlar. Bir sonlu alanın düzen(order) ı bir p asal sayısının n. kuvveti(p^n) olarak gösterilmelidir. Burada n pozitif bir tamsayıdır. Düzeni p^nolan bir sonlu alan, genellikle GF(p^n) olarak yazılılr. GF sonlu alanı ilk defa çalışan matematikçi olan Galoi'den gelmektedir. Özel durum olan n=1 için, sonlu alan GF(p) olarak yazılır.
Özel durum olarak GF(2^n) ve GF(3^n) verilebilir.
Düzeni p olan bir sonlu alan GF(p), {0,1,.....,p-1} Z_p tamsayılar kümesinin modulo p aritmetik işlemleri ile birlikte tanımlanmasıdır.
Burada herbir elemanın bir çarpımsal tersi vardır ve çarpımsal invers olarak (w^{-1}) Çarpımsal invers . $\forall w \in Z_p$ için (sıfır hariç) Z_p'de bir z vardır ve w***x***z= 1mod p 'dir.
Çünkü , w, p ye göre asaldır. Eğer, Z_p nin elemanlarını w ile çarparsak, sonuçtaki kalanlar Z_p nin elemanlarının tamamının tekrarıdır. Böylece en az bir kalanın değeri 1'dir.Bu yüzden Z_p 'de en az bir eleman vardır öyleki, w ile çarpıldığında kalan 1'dir. Bu tamsayı w'nın çarpımsal tersi(w^{-1}) dir.
Tablo 3.3'de GF(7) sonlu alanında Modulo 7 nin toplamsal ve çarpımsal tersleri gösterilmiştir.

w	-w	w^{-1}
0	0	-
1	6	1
2	5	4
3	4	5

4	3	2
5	2	3
6	1	6

Tablo 3.3 Modulo 7 için toplamsal ve çarpımsal tersler

Asal Sayılar

Bir p >1 sayısı ancak ve ancak bölenleri ±1 ve ±p ise asal sayıdır. Asal sayılar, Açık-anahtarlı kripto sistemlerinde büyük rol oynarlar. Asal sayılarda karşımıza çıkan önemli problemler, asal bir sayının oluşturulması ve bir sayının asal olup olmadığının test edilmesidir. Asal sayı oluşturma, verilmiş bir [r_1,r_2] tam sayılar aralığında asal sayı bulma işlemidir.

Herhangi bir a>1 tamsayısı tek bir şekilde aşağıdaki gibi ifade edilebilir.

$$a= p_1^{a_1} p_2^{a_2} .....p_t^{a_t}$$

burada $p_1,p_2,.....,p_t$ asal sayılardır ve a_i tamsayıdır. (örn: $3600 = 2^4 \times 3^2 \times 5^2$)

Tanım: $a^{s-1} \equiv 1 \bmod s$ şartını ve $1<a<s$ şartını sağlayan s tam sayısına a tabanına göre **sanki asal** (pseudoprime) sayı denir**.**

Teorem (Fermat teoremi) p bir asal sayı olsun. Her p ile bölünemeyen a pozitif tam sayısı için,

$$a^p \equiv a \bmod p \qquad \text{denkliği;}$$

ve p ile bölünmeyen her a tam sayısı için ise $a^{p-1} \equiv 1 \bmod p$. denkliği her zaman doğrudur:

İsp: Önceki bölümlerde açıklandığı üzere, eğer,Z_p nin elemanlarını{0,1,....,(p-1)} a , modulo p ile çarparsak, sonuçtaki kalanlar Z_p nin elemanlarının tamamının sekansıdır. Bundan başka, a x 0 = 0mod p dir. Bu yüzden (p-1) sayı, { $a$mod p, 2$a$modp, ...,(p-1)$a$mod p } ,dizisi {0,1,....,(p-1)} sayısı ile aynı düzendedir. Her iki kümenin sayılarını çarpıp mod p'sini alarak aşağıdaki bağıntı yazılabilir.

ax2ax ..... x ((p-1)a) = [(a modp) x (2$a$modp) x .....x((p-1)a modp]modp
= [1 x 2 x ..... x (p-1)] modp
= (p-1)! modp

Fakat, a x 2a x .....x((p-1)a) = (p-1)!a^{p-1} dir

Bu yüzden, (p-1)!a^{p-1} = (p-1)! modp . dir. Burada (p-1)! 'i atabiliriz. Sonuçta:

a^{p-1} = 1 modp olduğu gösterilmişolur.

Örn: a=7, p = 19 verilsin.

$7^2 = 49 = 11 \bmod 19$

$7^4 = 121 = 7 \bmod 19$

$7^8 = 49 = 11 \bmod 19$

$7^{16} = 121 = 7 \bmod 19$

$a^{p-1} = 7^{18} = 7^{16} \times 7^2 = 7 \times 11 = 1 \bmod 19$

alternatif olarak $a^p \equiv a \bmod p$ olarak da yazılabilir.

3.3 Euler Totient fonksiyonu

n tam sayısı için Euler Totient fonksiyonu $\phi(n)$, n den daha küçük olan ve n ile aralarında asal olan bütün pozitif tam sayıların sayısını verir.

p asal ise $\phi(p) = p-1$ dir.

$n=p.q$ ve p, q asal sayılar ise $\phi(n) = \phi(.pq). = \phi(p).\phi(q) = (p-1).(q-1)$ dir.

$\phi(n) = \phi(.pq)$.olduğunu görmek için, Z_n 'deki kalanlar kümesinin [0, 1, ..., (pq-1)]. Olduğunu düşünelim. Kalanlar kümesindeki {p, 2p,.....,(q-1)p}, {q, 2q, ..., (p-1)q} ve 0 , n'e göre asal değildirler. Buna uygun olarak,

$\phi(n) = pq - [(q-1) + (p-1) + 1]$
$= pq - (p+q) + 1$
$= (p-1) \times (q-1)$
$= \phi(p).\phi(q)$

elde edilir. Tablo 3.4'da n = 30 'a kadar olan sayıların $\phi(n)$ değerleri gösterilmiştir

n	$\phi(n)$	**n**	$\phi(n)$	**n**	$\phi(n)$
1	1	11	10	21	12
2	1	12	4	22	10
3	2	13	12	23	22
4	2	14	6	24	8
5	4	15	8	25	20
6	2	16	8	26	12
7	6	17	16	27	18
8	4	18	6	28	12
9	6	19	18	29	28
10	4	20	8	30	8

Tablo 3.4. 1-30 arası sayılar için $\phi(n)$ değerleri

Teorem (Fermat teoremi) Eğer ***s*** bir asal sayı ve ***OBEB*(*a,s*)=1** ise ***s***, ***a*** tabanına göre bir sanki asal (pseudo prime) sayıdır.

Tek Yönlü Fonksiyon

F: $X \longrightarrow Y$

f: $x \longrightarrow f(x)=y$ yalnız ve yalnız aşağıdaki şartları taşıdığı takdirde tek yönlü bir fonksiyondur:

- $f(x)$ bütün x değerleri için polinomsal zamanda çözümlenebilir olmalıdır.
- Verilen bir y değeri için x değeri polinomsal zamanda bulunamamalıdır.

Örnek olarak verilirse $a^m \bmod n \equiv x$ bir modüler üs alma işlemidir ve kolaylıkla yapılabilir, fakat var olan x değerinden m değerini bulmak ayrık logaritma problemine girer ve bunun da hesaplanma süresi polinomsal çözümleme süresinden çok daha uzundur.

Kapaklı Tek Yönlü Fonksiyonlar (Trapdoor One-Way Functions)

Kapaklı tek yönlü fonksiyonlarda ise tek yönlü fonksiyonlara ek olarak analizciye başka bilgiler verilirse fonksiyon daha kolay tersinir hale getirilebilir.
Örneğin yalnız $a^m \bmod n$ değerini bilmekten öte buradaki n değerinin iki asal sayının çarpımı olduğunu ve anahtarların bu sayılara bağlı olduğunu bilmek buradan m değerini bulma aşamasında analizciye ipucu vermiş olur.

3.4 GF(p) 'de üstel işlem

- Birçok kriptolama algoritması üstelleştirmeyi kullanır, b üssü ne göre büyüyen bir a sayısı(taban) mod p
  - $b = a^e \bmod p$
- üstelleştirme basit olarak bir n sayısı için O(n) çarpma olan tekrarlanan çarpmalardır.
- Daha iyi bir yöntem kare ve çarpma algoritmasıdır.

let base = a, result =1
for each bit ei (LSB to MSB) of exponent
if ei=0 then

square base mod p
if ei=1 then
multiply result by base mod p
square base mod p (except for MSB)
required ae is result

- Bir n sayısı için sadece $O(\log_2 n)$ çarpma yapılır.

3.5 GF(p) 'de ayrık Logaritma Problemi

Ayrık logaritma problemi, grup olarak tanımlanan matematiksel yapılara uygulanır. Daha önce de açıklandığı gibi, bir grup çarpımı dediğimiz bir ikili işlem ile elemanların birlikte toplanmasıdır. Bir grup elemanı α ve bir n sayısısı için; α^n , α nin n kere kendisi ile çarpımından elde edilisin; $\alpha^2 = \alpha * \alpha$, $\alpha^3 = \alpha * \alpha * \alpha$, ….

Ayrık logaritma problemi, aşağıdaki gibidir. Bir sonlu grup G'de verilen bir α elemanı ve diğer eleman $b \in G$ için ; Öyle bir x tamsayısı bulunsun ki $\alpha^x = b$ eşitliğini sağlasın. Örneğin, $3^x \equiv 13 \pmod{17}$ probleminin çözümü 4 'tür. Çünkü $3^4 = 81 \equiv 13 \pmod{17}$ dir.

Çarpanlara ayırma problemi gibi, ayrık logaritma probleminin de zor olduğu kabul edilir ve bir tek yönlü fonksiyonun sert yönü gibidir. Her nekadar ayrık logareitma problemi herhangi bir grup üzerinde isede kriptografik amaçla genellikle $\mathbf{Z}_n$ grubu kullanılır.

Bir başka ifade ile ayrık logaritma :

- Üstelleştirmede ters problem, bir modulo p sayısının ayrık logaritmasının bulunmasıdır.
  - $\alpha^x = b \bmod p$'de *x* 'i bul
- üstelleştirme nispeten kolay iken, ayrık logaritmanın bulunması genellikle kolay yolu olmayan zor bir problemdir.
- Bu problemde, eğer p asal ise , herhangi bir b! = 0 için her zaman bir ayrık logaritması olan bir α olduğu gösterilebilir.
  - α'nın ardışıl kuvvetleri mod p ile **grup** oluşturur

  α mod p, α^2 mod p,....., α^{p-1} mod p 1 farklıdır ve 1 ila p-1 arasında değer alır.
- Öyle ki α ya **primitif kök** denir ve aynı zamanda bulmak nispeten zordur.

α'nın ardışıl kuvvetlerinin mod p ile oluşturduğu **grup'**ta, herhangi bir *b* tamsayısı ve p'nin primitif kökü olan α için bir *x* üssü bulunabilir ki;

$$b = \alpha^x \bmod p \quad 0 \le x \le (p-1) \text{ dir.}$$

Üs *x* ayrık logaritma veya indis olarak gösterilir.

3.6 En Büyük ortak Bölen(Greatest Common Divisor)

Teorem *a* ve *n* tam sayıları için, ($a \in \{0,1,...n-1\}$); eğer *a* ve *n* aralarında asal iki sayıysa *a* nın modül *n'* e göre yalnız bir tane tersi vardır ve a^{-1} sembolüyle gösterilir.

$OBEB(a,n) = 1 \Leftrightarrow \exists\ b \in [a,n-1],\ 1 = a.b \bmod n$, yani $b = a^{-1}$ dır.

- A ve b'nin en büyük ortak böleni(a,b) a ve b'nin her ikisini de bölen en büyük sayıdır.
- **Euclid's Algoritması** iki a ve n(a<n) sayısının en büyük ortak bölenini bulmak için kullanılır,
  - Eğer a ve b nin böleni d ise, a-b ve a-2b yi bulur

GCD (a,n) is given by:
let g0=n
g1=a
gi+1 = gi-1 mod gi
when gi=0 then (a,n) = gi-1

örn. (56,98) 'i bulalım.
g0=98

g1=56
g2 = 98 mod 56 = 42
g3 = 56 mod 42 = 14
g4 = 42 mod 14 = 0
sonuçta EBOB (56,98)=14

3.7 Teorem (Chinese Remainder Teoremi)

Modüler karekök bulunması problemlerini göz önüne alırsak, asal üs modülo için indirgenebilen genel bir mod m problemi buluruz. Bir sonraki problem, orijinal benzerliği çözmek için, asal üslerin çözümün nasıl parçalanabileceği olacaktır. Bu Chinese kalan teoremi ile yapılabilecektir.

Tipik bir problem eşzamanlı olarak çözülen tamsayı x 'leri bulmaktır.

x **≡ 13 (mod 27)**

x **≡ 7 (mod 16)**

Bu uygulamada iki modülo birbine göre asal olması önemlidir. Diğer durumda iki benzerliğin uygunluğu test edilmelidir. Chinese kalan teoreminin çok basit bir cevabı vardır.

Chinese Kalan teoremi: Birbirine göre asal olan modul *m* ve *n*,için benzerlik ;

x **≡ a (mod m)**

x **≡ b (mod n)**

x için modulo *mn* şeklinde tekbir çözümü vardır.Örnek problemde mod 16.27 = 432 tek bir çözümü olacaktır.

Problemi çözmek için daha basit bir yöntem vardır. Daha basit bir örnek üzerinde düşünelim. Bütün *x* lerin sağladığı

x **≡ 2 (mod 3)**

x **≡ 3 (mod 5)**

İlk benzerliği sağlayan sekans 2,5,8,11,14,17 ,…… dir. Bu sekans tarandığında 5' 2 bölündüğü zaman 3 kalan terim 8 olduğu için cevap 8'dir. Bunun daha kolay bulnması için Öklid'in enbüyük ortak bölen algoritmasından faydalanılır.

Bütün işlemi genelleştirirsek ;

x **≡ a (mod m)**

x **≡ b (mod n)**

Önce *mu*+*nv*=1 denklemini sağlayan, u ve v tamsayıları bulunmalıdır. Sonra bütün çözümler x= (mu)b + (nv) a (mod mn) nı sağlamalıdır.

Bir diğer örnek $x \equiv 23 \pmod{100}$ $x \equiv 31 \pmod{49}$ verilsin.

Önce; 100u + 49 v = 1 çözülmelidir.

Euclid's algoritması aşağıdaki şekilde kullanılır.

							0	1
Bölünen	=	Bölüm	.	Bölen	+	Kalan	1	0

100	=	2	.	49	+	2	2	1
49	=	24	.	2	+	1	49	24
2	=	2	.	1	+	0	100	49

Buradan 49.49 -24.100 = 1 dir. Çözüm 49.49..23- 24.200.31 = -19177 ≡ 423 (mod 4900) dir.

Genel hali;

$x \equiv a_1 \mod m_1$

$x \equiv a_2 \mod m_2$ .....

$x \equiv a_r \mod m_r$ ve $OBEB(m_i, m_j) = 1,\ i \neq j$

benzerlik sistemleri için, *x* ' in en az bir çözümü vardır:

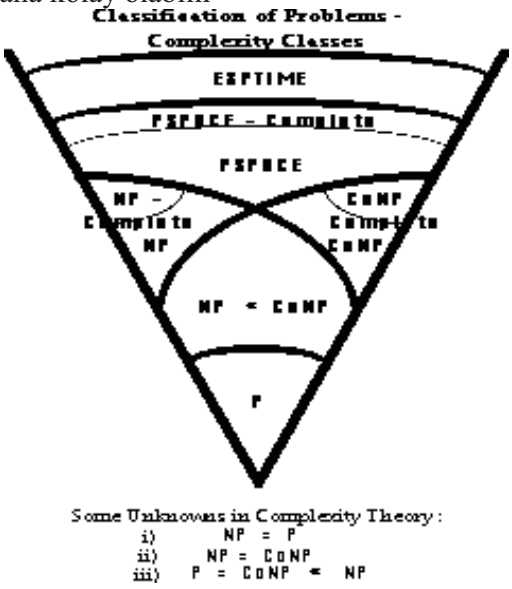
$$x = \sum a_i \,.\, M_i \,.\, N_i$$

$$M = m_1 . m_2 . m_3 .. \ldots . m_r \quad ve \quad M_i = M / m_i, \quad N_i = M_i^{-1} \bmod m_i .$$

En önemli uygulama RSA algoritmasındaki çok büyük olan p ve q asal sayılarının çarpımında çok zaman alan işlemleri azaltmak için kullanılır. Hesaplamalar Z_n 'den Z_p x Z_q 'ya taşınarak daha küçük bit uzunluklu verilerle işlemler basitleştirilir.

3.8 Karmaşıklık Teorisi (saksı benzeri bakış)

- Karmaşıklık teorisi, bir problemin çözümünün genelde ne kadar zor olduğu ile ilgilenir.
- Problem çeşitlerinin sınıflandırılmasını sağlar
- Bazı problemler esastan diğerlerinden daha zordur.,örneğin
  - Sayıların çarpımı $O(n^2)$
  - Matrislerin çarpımı $O(n^{(2)(2n-1)})$
  - Çapraz kelime çözümleri $O(26^n)$
  - Asal sayıların tanınması $O(n^{\log \log n})$
- En kötü durum karmaşıklığına değinir.
  - Ortalamada daha kolay olabilir



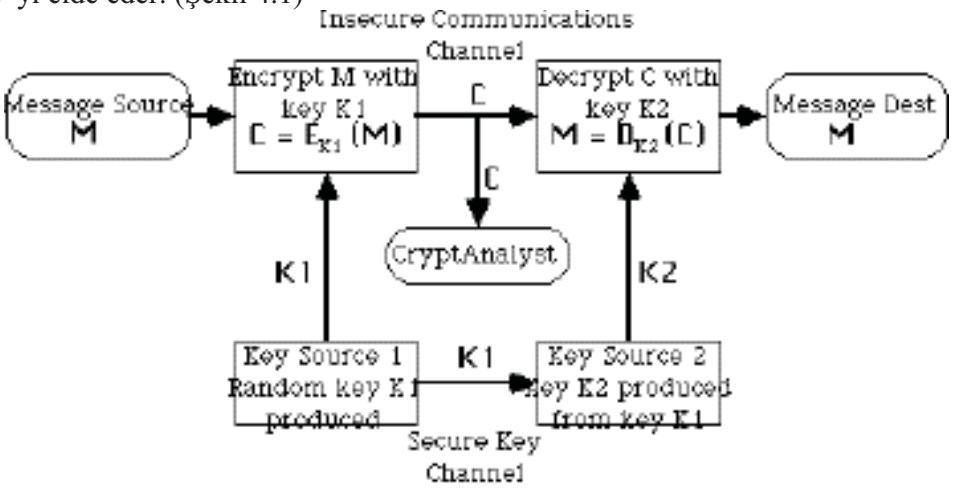
3.8.1 Karmaşıklık Teorisi- Bazı Terminoloji

- Bir problemin anlık durumu genel bir problemin kısmi örneğidir

- Bir problemin giriş uzunluğu, onun kısmi örneğini karakterize etmek için kullanılan n sembol sayısıdır.
- Bir fonksiyonun derecesi, **f(n)** bazı **g(n)** in **O(g(n))** idir.
    - f(n)<=c.|g(n)|, bütün, n>=0, bazı c için
- (**P**) polinomsal zaman algoritması O(p(n)) zaman karmaşıklı kısmi bir problemin herhangi bir anını çözer,burada p giriş uzunluğu üzerine bazı polinomlardır
- çözüm zamanı olan (**E**) üstel zaman algoritması sınırlanmamıştır.
- Problemin ani çözümünün bir tahmini için polinomsal zamanda doğruluk testi yapılabilen (**NP**) **non-deterministic polinomsal zaman** algoritmasıdır.
- **NP-complete** problemleri polinomsal çözüme sahip olan bir problem olarak bilinen NP problemlerin alt sınırıdır. Burada bütün NP problemleri polinomsal çözüme sahiptir. Bunlar en zor NP problemleridir
- **Co-NP** problemleri NP problemlerinin eşleniğidir, Co-NP problemlerinin bir çözümünü tahmin etmek çözüm uzayınının detaylı araştırılmasını gerektirir .

4 GİZLİ ANAHTARLI (SİMETRİK) KRİPTOSİSTEMLER:

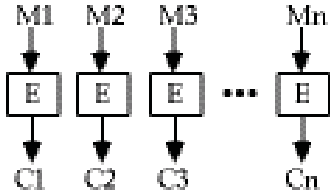
Gizli anahtarlı kriptografik sistemler tarihin ilk devirlerinden beri dünyada kullanımı süregelen kriptografik sistemlerdir. Bu sistemlerde şifreleme algoritması ve deşifreleme algoritması birbirinin tersi şeklindedir. Öncelikle haberleşecek iki grup aralarında gizli bir anahtar tespit ederler. Eğer bu iki grup birbirlerine yakın yerlerde yer almıyorlarsa güvenli bir haberleşme kanalı veya güvenilir bir kurye yoluyla anahtarları birbirlerine ulaştırabilirler. Bir taraf şifreleme algoritmasında girdi olarak açık metin (P) ve anahtarı (K) uygular, ardından şifreli metin (C) yi elde eder ve mesajın alıcısına gönderir. Mesaj alıcısı ise deşifreleme algoritmasının girdileri olarak şifreli metin (C) yi ve aynı (K) anahtarını kullanır ve ardından çıktı olarak açık metin (P) yi elde eder. (Şekil 4.1)



Şekil 4.1 Gizli-anahtarlı kriptosistem ile haberleşme

Gizli-anahtarlı kripto sistemleri uygulama sahalarında ikiye ayrılır;

i. Blok Şifreleme: Şifreleme ve deşifreleme işleminde metinler sabit uzunluklu dizilere bölünüp blok blok işleme tabi tutulur (örneğin 8, 16, 32 bit veya bayt). Anahtar uzunluğu ise yine sabittir. Blok şifrelemeye örnek olarak IBM tarafından 1976 yılında tasarlanan ve A.B.D Teknoloji Standartları Enstitüsü NIST tarafından her dört yılda bir güvenliği onaylanan DES (Data Encryption Standard) algoritması verilebilir. DES algoritması şifrelenecek metni 64 bitlik bloklar halinde şifreler, kullandığı anahtar boyu ise yine 64 bittir. Yalnız burada anahtarın işaret bitlerinin ayıklanmaları durumunda anahtar boyunun 56 bite indiğini hatırlatmak gerekir. Diğer bilinen blok şifrelemeli algoritmalara ise FEAL, IDEA ve RC5 örnek olarak gösterilebilir. Çalışacağımız çoğu modern şifreleyici bu formdadır. (Şek. 4.2)

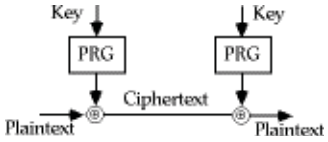


Şekil 4.2.Blok Şifreleyici

ii. Dizi Şifreleme: Bu çeşit şifrelemede algoritmanın girdisi yalnızca anahtardır. Algoritma anahtardan rastgele bir diziye çok benzeyen kayan anahtar dizisi üretir. Daha sonra kayan anahtar dizisinin elemanları ile açık metin veya kapalı metin dizisinin elemanları ikili tabanda toplanarak şifreleme veya deşifreleme işlemi tamamlanır. Dizi şifreleme algoritmalarına örnek olarak **RC4** algoritması gösterilebilir.

- Mesajı bit bit işler. (dizi olarak)
- En meşhur olanı **Vernam cipher** şifreleyicisidir(aynı zamanda **one-time pad denir**)
- 1917'de AT&T de çalışan Vernam tarafından geliştirildi
- basit olarak mesaj bitlerini rastgele anahtar bitlerine ekler.(şek. 4.3(a))
- mesaj biti kadar anahtar biti gerekir. Prtatikte zordur.(örn. Pratikte mag teyp veya CDROM da dağıtılır)
- anahtar tamamen rastgele olduğu için koşulsuz güvenlik sağlanır.
- böyle büyük bir anahtar dağıtımı güç olduğu için anahtar dizisi daha küçük(taban) bir anahtardan üretilebilir. Bunun için rasgele sembol fonksiyonları kullanılır.(şek 4.1(b))
- Her ne kadar bu çok çekici gözükse de pratikte iyi bir kriptografik güçlü rasgele fonksiyon bulmak çok güçtür. Bu hala birçok araştırmacının konusudur.

Şekil 4.3(a)



Şekil 4.3(b)

4.1 Simetrik Şifreleme Algoritmaları

Geleneksel simetrik blok şifreleme algoritmaları(örn. DES) 1973'de IBM'de çalışan Hort Feistel Tarafından geliştirilen Feistel networküne dayanır. Bu nedenle Feistel blok şifreleyicinin anlaşılması önemlidir.

Bir dizi şifreleyici sayısal bir veriyi bit bit veya bayt bayt şifreleme yapar.(Örnek vernem şifreleyici) Blok şifreleyici ise veryi sabit uzunluklu bloklara ayırıp bu blokları şifrelereyerek aynı uzunluklu şifreli bloklar elde eder. Tipik blok uzunlukları 64 veya 128 bit olabilir.

Feistel Şifreleyicinin yapısı

Feistel, pratikte yerine koyma ve yer değiştirme işlemlerine alternatif olan ve shanon tarafından önerilen confusion ve diffusion fonksiyonlarını şifreleme algoritmasında önerdi.

Diffusion da, şifresiz metnin istatistiksel yapısı, şifreli metnin istatistiğine dağıtılır. Bu, şifresiz metnin herbir dijitinin, şifreli metnin etkilediği dijitlerinin bulunmasıyla sağlanır., başka bir ifade ile,her bir şifreli metin dijit'i birçok şifresiz metin dijiti tarafından etkilenir. Örnek olarak; Bir M = m_1,m_2, m_3,…….. karakterlerinden oluşan bir şifresiz metni ortalama işlemi ile k ardışıl karakteri ekleyerek şifrelemek;

$y_n = \sum_{i=1}^{k} m_{n+i} (\mathrm{mod}\, 26)$ ile yapılmış olsun. Şifresiz metnin istatistiksel yapısının dağılmış olduğu gösterilebilir. Böylece şifreli metindeki karakter dağılımı şifresiz metindeki karakter dağılımının yakınında olacaktır.

Confusion'da ise, anahtarın keşfedilmesi saldırılarına karşı, şifreli metnin istatistiği ile şifreleme anahtarının olabildiğince karmaşık olmasını araştırır. Böylece bir saldırgan şifreli metnin istatisiğini hesaplasa bile hangi anahtar ile şifrelendiğini anlaması çok zorlaşır.

Şekil 6.10'da gösterilen bu algoritmada 2w bit uzunluğun da olan şifresiz metin iki eşit sol ve sağ parçaya ayrılır. Her bir turda ana şifreden üretilen alt şifre ile sağ tarafa F fonksiyonu uygulanır. Bunun sonucu ise sol taraf ile EXOR mantıksal işlemine tabi tutulur. Daha sonrada elde edilen

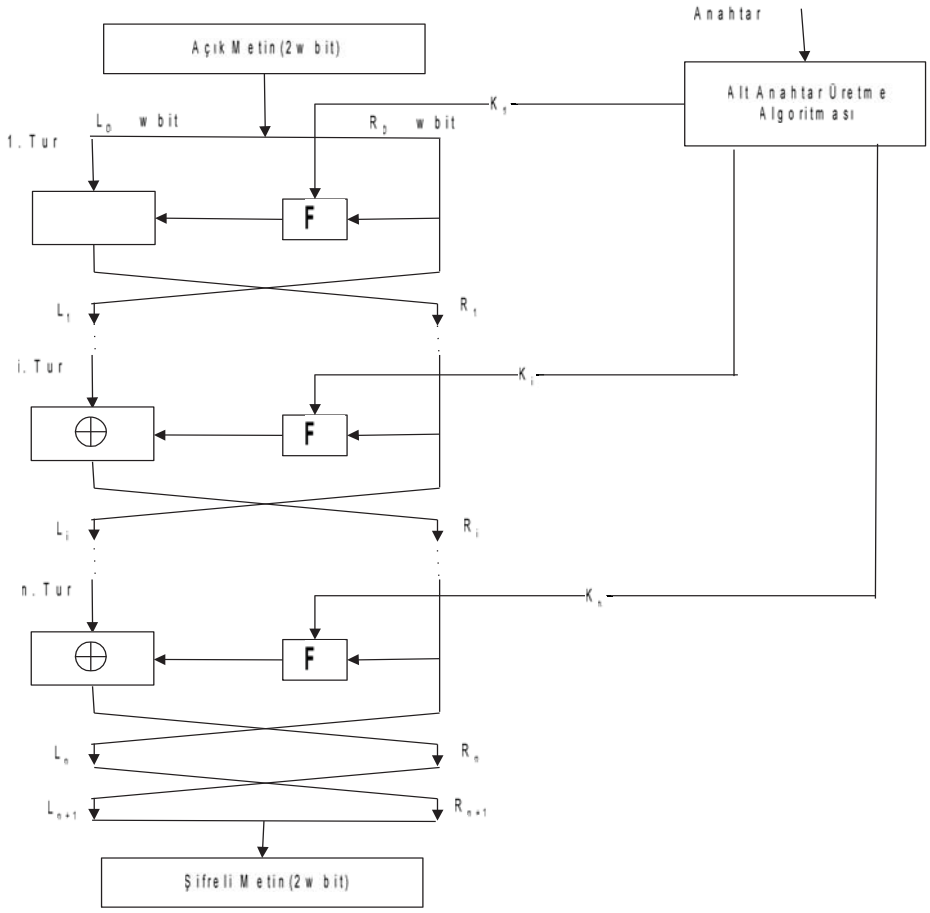
sonuçlar çaprazlanır. Yani sağ taraf sola sol taraf sağa geçer. Böylece turlar devam eder. Asıl anahtardan alt anahtarlar her turda üretilerek F fonksiyonuna girdi olarak kullanılır.
Feistel algoritmasının önemli parametreleri aşağıda açıklanmıştır.

Blok uzunluğu: Büyük blok uzunluğu daha fazla güvenlik anlamındadır. Fakat şifreleme/deşifreleme hızını azaltır. Genel olarak 64 bitlik blok genişliği kullanılır.
Anahtar Uzunluğu: Büyük anahtar genişliği daha fazla güvenlik anlamındadır. Fakat şifreleme/deşifreleme hızını azaltır. Çok kullanılan anahtar uzunluğu 128 bittir.
Tur Sayısı: Fazla tur sayısı şifreleme güvenliğini artırır .Genel olarak 16 Tur kullanılır.
Alt Anahtar Üretme Algoritması : Karmaşıklığı fazla olan bir alt anahtar üretimi kirptoanalizi zorlaştırır.
Tur Fonksiyonu : Fazla karmaşık olan tur fonksiyonu kriptanalizi zorlaştırır.

Feistel şifreleyici için diğer özellikler ,
Hızlı yazılım şifreleme/deşifreleme: Çoğu uygulamada, şifreleme uygulamaları veya donanım gerçeklemesi şeklinde kullanım fonksiyonlarının içine koyulur. Dolayısı ile algoritmanın icra hızının düşünülmesi gerekir.
Analiz Kolaylığı : Her ne kadar algoritmanın olası kriptanaliz saldırılarına karşı olabildiğince karmaşık olması istenirse, bu özellik algoritmanın anlaşılabilirliğini de azaltır. Örneğin DES kolay analiz edilen bir algoritma değildir.

Feistel şifreleyicinin deşifreleme algoritması da aynıdır. Şifreli metin giriş olarak kullanılırken alt anahtar tersinden kullanılır. Yani önce K_n , en son olarak da K_1 kullanılır. Bu özellik nedeniyle Şifreleme ve deşifrelemede farklı algoritma kullanılması gerekmez.
Algoritmanın genel matematiksel hesaplanması; LE_i : Sol şifrelenmiş blok, RE_i : Sağ şifrelenmiş blok, olmak üzere,

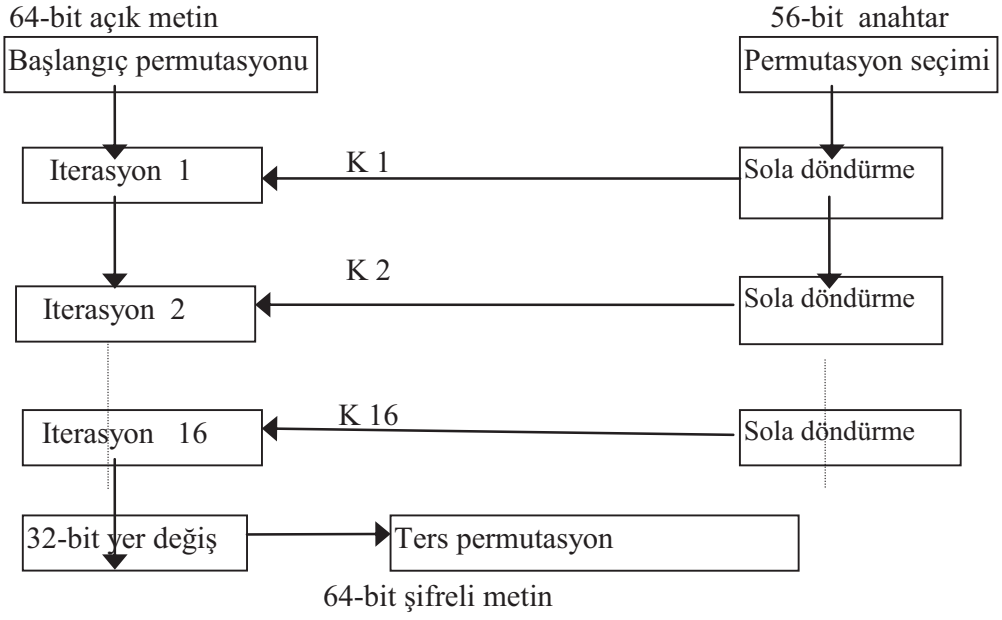
$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F(RE_{i-1} , K_i)$$



Şekil 4.4. Klasik Feistel Network

4.2 DES

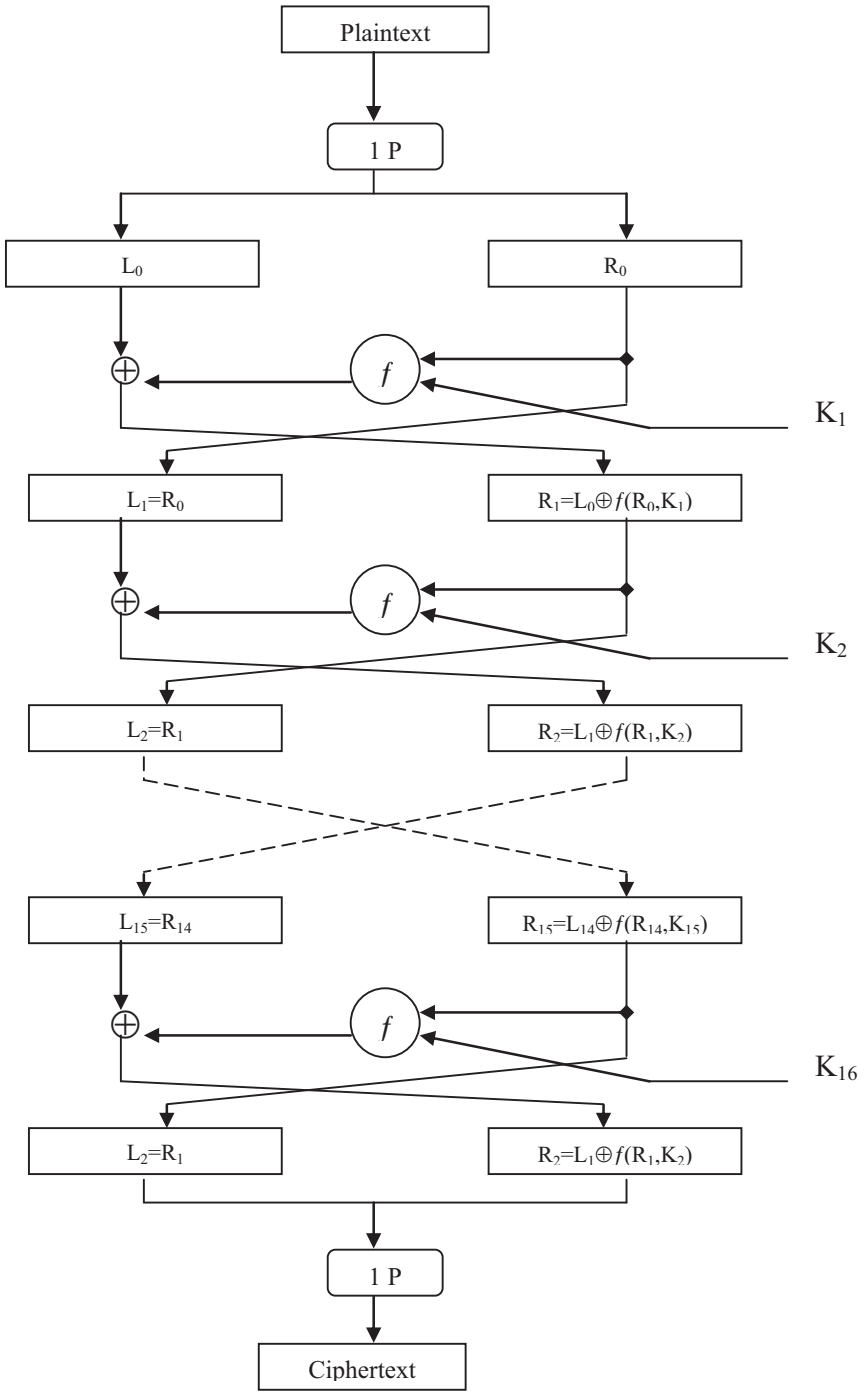
Data Encryption Standart (DES) 1974 yılında IBM tarafından geliştirilmiş ve 1977 yılında yasal olarak atanmıştır. Basit blok şema Şekil 4.5'de gösterilmiştir. Temeli Feistel networküne dayanır.



Şekil 4.5. DES Algoritmasının genel yapısı

DES bir blok şifrelemedir, 64 bit bloklardaki veriyi şifreler. Plain textin 64 bitlik bloğu bir algoritmaya sokulur ve 64 bitlik şifrelenmiş bir ifade elde edilir. Şifrelemede ve şifreyi çözerken her ikisinde de aynı algoritma ve anahtarlar(key) kullanılır.

Anahtar uzunluğu 56 bittir. (Anahtar genellikle 64 bit olarak ifade edilir, fakat her sekizinci bit parity biti olarak kullanılır ve ihmal edilir.) Anahtar herhangi bir 56 bit sayı olabilir ve her zaman değiştirilebilir.



Şekil 4.6 DES Algoritması

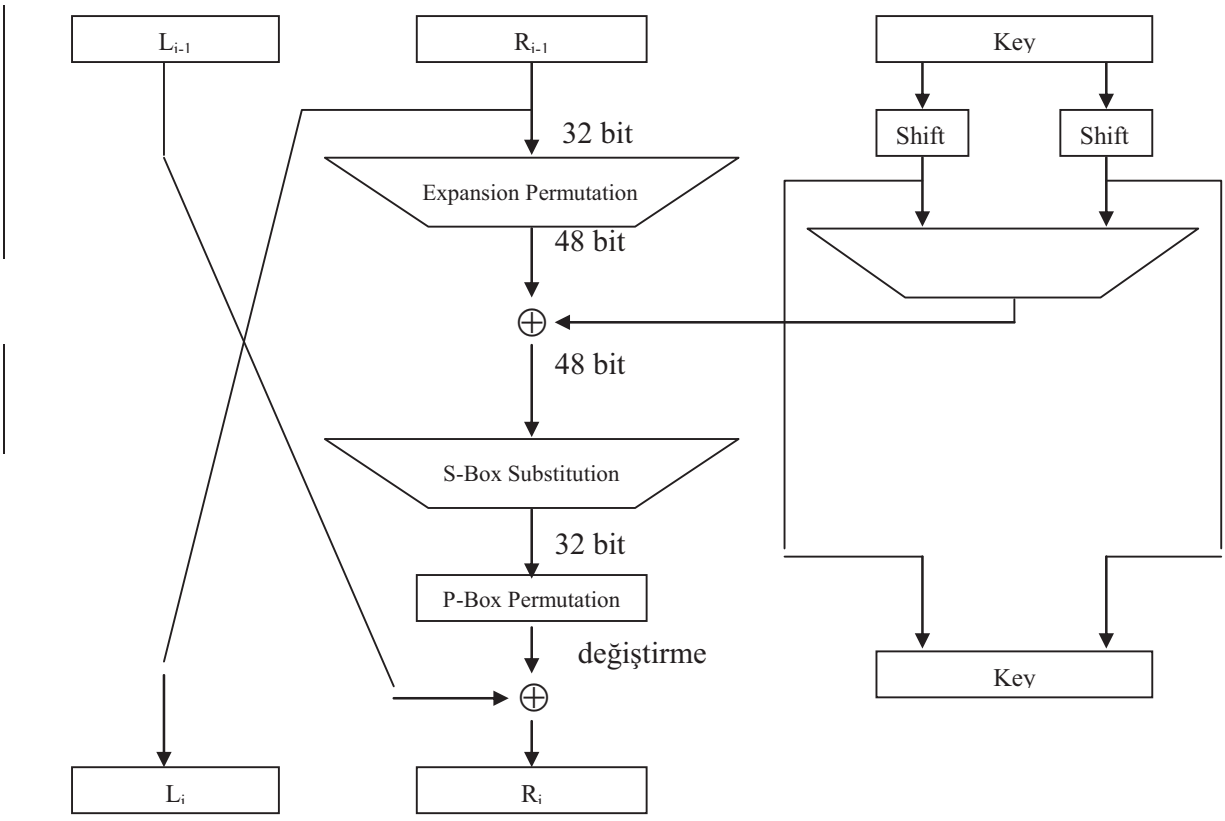
4.2.1 Algoritmanın Özeti :

DES 64 bit blok plaintext de işlem görür. Plaintext, ilk permutasyondan sonra yarısı sağda yarısı solda her biri 32 bit uzunluğunda iki parçaya bölünür. Daha sonra *f* fonksiyonu ve anahtar ile birleştirilerek sonraki adıma geçilir. Aynı işlem 16 kez tekrarlanır ve 16. turun sonunda, sağ ve sol parçalar birleştirilir. Son permutasyondan sonra (başlangıçtaki permutasyonun tersi) algoritma tamamlanarak biter.

Her bir turda anahtar bitleri değiştirilir ve anahtarın 56 bitinden 48 biti seçilir. Verinin sağ yarımı genişleme permutasyonu (expansion permutation) yoluyla 32 bitten 48 bite genişletilir. Genişletilen kısım seçilen 48 bit anahtarla XOR işlemine sokulur. Daha sonra 32 yeni bit üreten 8 S-box içerisine gönderilir ve tekrar değiştirilir. Bu dört işlem *f* fonksiyonunu oluşturur. *f* fonksiyonunun çıktısı verinin sol yarımı ile XOR işlemine tabi tutulur. Sonuçta elde edilen değer yeni sağ yarım olmakta ve sol yarım ise sağ yarımın eski hali olmaktadır. (4.1) de gösterilen bu işlem 16 kez tekrar eder.

$$L_i=R_{i-1} \qquad R_i=L_{i-1}\oplus f(R_{i-1},K_i) \qquad (4.1)$$

$$\text{Genel} \quad m_{i+1} = m_{i-1} \oplus f(m_i,K_i) \qquad (4.2)$$



Şekil 4.7. DES' in bir turu

4.2.2 Başlangıç Permutasyonu :

Başlangıç permutasyonu tur 1' den önce meydana gelir. Şifrelemeden önce 64 bitlik plain text 32 bitlik iki parçaya bölünür. Tüm çift bitler sol tarafta ve tek pozisyondaki bitler de sağ tarafta yer alır. Tablo 9.3' de tanımlandığı gibi giriş bloklarının yerleri değiştirilir. Tabloda görüldüğü gibi örneğin; başlangıç değişiminde plaintext in 1. pozisyonundaki bite 58 nolu bit taşınmış, 2. pozisyonuna 50 nolu bit atanmış vb...

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

Tablo 4.1 Başlangıç Permutasyonu

Başlangıç permutasyonu ve benzer şekilde sonuç permutasyonu DES' in güvenliğine etki etmez.

4.2.3 Anahtar Dönüşümü :

Başlangıçta, 64 bitlik DES anahtarı her sekiz bit ihmal edildiği için 56 bite düşürülür. Bu tablo 6.8' de tanımlanmıştır. İhmal edilen bu bitler anahtarı kontrol etmek için parity kontrolünde kullanılır. 56 bitlik anahtar elde edildikten sonra DES' in 16 turunun her biri için farklı 48 bit alt-anahtar üretilir. Bu alt-anahtar ler(K_i) şu şekilde belirlenir.

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

Tablo 4.2 Anahtar Permutasyonu

İlk olarak 56 bitlik anahtar 28 bitlik iki parçaya bölünür. Turun ihtiyacına göre parçaların bir veya iki biti değiştirilir. Değiştirilecek bit sayıları tablo 4.3' de belirtilmiştir.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Tablo 4.3 Turların her biri için değiştirilen anahtar bitlerinin sayısı

Değiştirmeden sonra, 56 bitten 48 biti seçilir. Bu işlemde bitlerin altkümesi seçildiği için, bitlerin düzeni değişir. Bu işlem *compression permutation* olarak adlandırılır. Tablo 4.4' da compression permutation tanımlanmıştır.

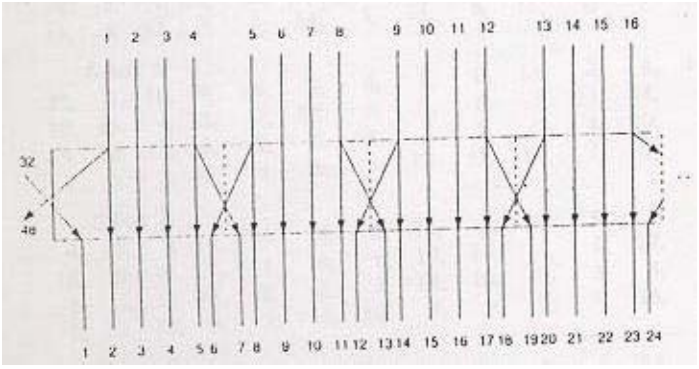
14 17 11 24 1 5	3 28 15 6 21 10
23 19 12 4 26 8	16 7 27 20 13 2
41 52 31 37 47 55	30 40 51 45 33 48
44 49 39 56 34 53	46 42 50 36 29 32

Tablo 4.4 Sıkıştırma Permutasyonu

4.2.4 Genişleme permutasyonu :

Bu işlemde verinin sağ yarısı (R_i) 32 bitten 48 bite genişletilir. Çünkü bu işlem tekrar eden belirli bitleri en uygun şekilde değiştirir. Bu işlem iki amaç için yapılır. XOR işlemi için sağ yarımı anahtar ile aynı uzunlukta yapmak ve yerine koyma (substitution) işlemi sırasında sıkıştırılabilen daha uzun sonuç sağlamak.

Şekil 4.8' de genişleme permutasyonu tanımlanmıştır. Her 4 bit giriş bloğu için, birinci ve dördüncü bitlerin her biri çıkış bloğundan iki biti gösterir, ikinci ve üçüncü bitler ise çıkış bloğundan birer bit gösterir.



Şekil 4.8 Genişleme Permutasyonu

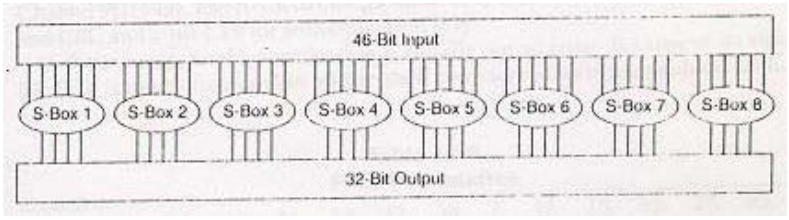
Tablo 4.5'de çıktı pozisyonlarının hangi girdi pozisyonlarına göre nasıl yerleştirildiği görülmektedir. Örneğin; girdi bloğunun 3. pozisyonu çıktı bloğunun 4. pozisyonuna karşılık gelmektedir ve girdi bloğunun 21. pozisyonu çıktı bloğunun 32. pozisyonuna karşılık gelmektedir.

	1	2	3	4		5	6	7	8		9	10	11	12		13	14	15	16		17	18	19	20	
32	1	2	3	4	5	4	5	6	7	8	9	8	9	10	11	12	13	12	13	14	15	16	17	16	17	18	19	20	21

	21	22	23	24		25	26	27	28		29	30	31	32			
20	21	22	23	24	25	24	25	26	27	28	29	28	29	30	31	32	1

Tablo 4.5 Genişleme Permutasyonu

4.2.5 S-Box Yerine Koyma :



Şekil 4.9 S-Box Yerine Koyma

Sıkıştırılmış anahtar genişletilmiş blok ile XOR edildikten sonra, 48 bit yerine koyma işlemine taşınır. Yerine koymalar sekiz tane substitution boxes veya S-boxes tarafından icra edilir. Her bir S-box da 6 bit giriş ve 4 bit çıkış vardır ve sekiz farklı S-box mevcuttur. 48 bit sekiz tane 6 bitlik alt bloğa bölünür. Her bir ayrılan blok, ayrılmış S-box tarafından işletilir. Birinci blok S-box 1, ikinci blok S-box 2 tarafından işleme sokulur.

Her bir S-box 4 satır ve 16 sütundan oluşan bir tablodur. Boxlardaki her bir giriş 6 bit, çıktı 4 bitlik sayıdır. Girişin ilk ve son biti hangi satırın seçileceğini, ortadaki 4 bit ise 16 kolondan hangisinin seçileceğini belirler. Sonuçta tablonun o satır ve sütunundaki elemen çıktı değeri olarak belirlenir. Tablo 4.6'de sekiz S-box un tümü gösterilmiştir.

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

S1	0:	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7
	1:	0	F	7	4	E	2	D	1	A	6	C	B	9	5	3	8
	2:	4	1	E	8	D	6	2	B	F	C	9	7	3	A	5	0
	3:	F	C	8	2	4	9	1	7	5	B	3	E	A	0	6	D
S2	0:	F	1	8	E	6	B	3	4	9	7	2	D	C	0	5	A
	1:	3	D	4	7	F	2	8	E	C	0	1	A	6	9	B	5
	2:	0	E	7	B	A	4	D	1	5	8	C	6	9	3	2	F
	3:	D	8	A	1	3	F	4	2	B	6	7	C	0	5	E	9
S3	0:	A	0	9	E	6	3	F	5	1	D	C	7	B	4	2	8
	1:	D	7	0	9	3	4	6	A	2	8	5	E	C	B	F	1
	2:	D	6	4	9	8	F	3	0	B	1	2	C	5	A	E	7
	3:	1	A	D	0	6	9	8	7	4	F	E	3	B	5	2	C
S4	0:	7	D	E	3	0	6	9	A	1	2	8	5	B	C	4	F
	1:	D	8	B	5	6	F	0	3	4	7	2	C	1	A	E	9
	2:	A	6	9	0	C	B	7	D	F	1	3	E	5	2	8	4
	3:	3	F	0	6	A	1	D	8	9	4	5	B	C	7	2	E
S5	0:	2	C	4	1	7	A	B	6	8	5	3	F	D	0	E	9
	1:	E	B	2	C	4	7	D	1	5	0	F	A	3	9	8	6
	2:	4	2	1	B	A	D	7	8	F	9	C	5	6	3	0	E
	3:	B	8	C	7	1	E	2	D	6	F	0	9	A	4	5	3
S6	0:	C	1	A	F	9	2	6	8	0	D	3	4	E	7	5	B
	1:	A	F	4	2	7	C	9	5	6	1	D	E	0	B	3	8
	2:	9	E	F	5	2	8	C	3	7	0	4	A	1	D	B	6
	3:	4	3	2	C	9	5	F	A	B	E	1	7	6	0	8	D
S7	0:	4	B	2	E	F	0	8	D	3	C	9	7	5	A	6	1
	1:	D	0	B	7	4	9	1	A	E	3	5	C	2	F	8	6
	2:	1	4	B	D	C	3	7	E	A	F	6	8	0	5	9	2
	3:	6	B	D	8	1	4	A	7	9	5	0	F	E	2	3	C
S8	0:	D	2	8	4	6	F	B	1	A	9	3	E	5	0	C	7
	1:	1	F	D	8	A	3	7	4	C	5	6	B	0	E	9	2
	2:	7	B	4	1	9	C	E	2	0	6	A	D	F	3	5	8
	3:	2	1	E	7	4	A	8	D	F	C	9	0	3	5	6	B

Tablo 4.6 S-Box lar

4.2.6 P-Box Permutasyonu :

S-box yerine koyma işleminden sonra elde edilen 32 bitlik çıktı P-box da uygun bir şekilde değiştirilir. Bu değişiklikte girdi pozisyonuna göre çıktı pozisyonu tasarlanır. Hiçbir bit iki kez kullanılmaz ve hiçbir bit ihmal edilmez. Bu işlem *straight permutation* olarak çağrılır. Tablo 4.7'de her bir bitin taşındığı pozisyon gösterilmektedir. Örneğin, 21. bit 4. bite taşınmış ve 4. bit 31. bite taşınmıştır.

16 7 20 21 29 12 28 17

1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Tablo 4.7 P-Box Permutasyonu

En sonunda başlangıçtaki 64 bitlik verinin sol yarımı ile P-box permutasyonu sonucunda elde edilen 32 bitlik veri XOR işlemine sokulmaktadır. Sol ve sağ yarımlar değiştirilerek bir sonraki tur başlamaktadır.

4.2.7 Sonuç Permutasyonu :

Sonuç permutasyonu başlangıç permutasyonunun tersi şekilde çalışır ve tablo 4.8' da tanımlanmıştır. DES' in son turundan sonra elde edilen sağ ve sol yarımlar birleştirilerek ($R_{16}L_{16}$) sonuç permutasyonuna girdi olur. Bu algoritma şifrelemede ve şifreyi çözmede her ikisinde de kullanılır.

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

Tablo 4.8 Sonuç Permutasyonu

Çığ Etkisi :

Bir şifreleme algoritmasında anahtar veya şifresiz metindeki küçük değişikliklerin şifreli metrin üzerinde büyük değişikliğe neden olmasına çığ(avalance) etkisi denir.

4.3 DES' in Güvenliği :

Anahtar Uzunluğu ;

Bilindiği gibi DES'in anahtar uzunluğ 56 bittir. Bu ise brute-force atakları için $2^{56} = 7.2 \times 10^{16}$ anahtar sayısı demektir. Tablo 6.2 gözönüne alnırsa, mikrosaniye başına bir çözümleme yapan bir makinenın bin yıl gibi bir sürede DES'i kırabileceğini söylemek mümkündür.

Ancak,1998 yılına özel amaçlı olarak tasarlanan bir "DES kırıcı" bilgisayar($250.000) ile üç günden daha kısa sürede kırılabilmiştir. Bu nedenle anahtar sayısının ortalama yarısı kadar deneme yapılacağı varsayımı ile DES'in brute-force saldırılarına karşı zayıf olduğu söylenebilir.

DES'in alternatifleri olab 3DES ve AES geliştirilmiştir.

DES zamanlama saldırılarına karşı oldukça güçlüdür.

4.4 Diferansiyel ve Doğrusal(Lineer) Kiriptoanaliz.

DES'in anahtar uzunluğunun her ne kadar kısa olmasıyla kırılabilirliği fazla ise de daha kısa sürede kırılabilmesi için diferansiyel ve doğrusal kriptanaliz yöntemleri önerilmiştir.

Difreansiyel Kriptanaliz

Diferansiyel kriptanaliz, şifreli metin çiftleri ile onlara ait şifresiz metin çiftleri arasındaki kısmi farkları araştırır. Bu yöntem, aynı anahtar ile şifrelenen şifresiz metin, DES'in turlarında ilerlerken farkının değişimini analiz eder. Diferansiyel kriptanalizde en yi saldırı 2^{47} adet seçilen şifresiz metin, veya 2^{55} bilinen şifreli metin ve 2^{47} DES işlemi gerektirir.

DES'te şifrelenecek metin bloğu iki eşit parçaya ayrılır ($m = m_0 + m_1$) Her bir çevrimde $2 \le i \le 17$ olmak üzere m_i yeni blok blok elde edilir.

$m_{i+1} = m_{i-1} \oplus f(m_i,K_i)$ (i= 1,2, ……,16)

Diff. Kriptanaliz

$\Delta m = m \oplus m'$ (Mesaj yarıları)

$\Delta m_i = m_i \oplus m_i'$

$\Delta m_{i+1} = m_{i+1} \oplus m'_{i+1}$

$= m_{i-1} \oplus f(m_i,K_i) \oplus m'_{i-1} \oplus f(m'_i,K'_i)$

$= \Delta m_{i-1} \oplus [f(m_i,K_i) \oplus f(m'_i,K'_i)]$

Eğer biz, Δm_{i-1} ve Δm_i 'yi yüksek bir olasılık ile bilirsek Δm_{i+1} 'i de yüksek olasılık ile bilebiliriz. Eğer bu farklar belirlenebilirse f 'teki alt anahtarların da tahmin edilebilmesi mümkün olabilir.

m ve m' nün her bir çevrimdeki farkları şifreli metin için bulunur.

Diferansiyel Kriptanalizin işlemi;

İki m ve m' düz metin mesajı için verilen bir fark ile başlanır ve her bir çevrimdeki şifreli metindeki farklar izlenir. Gerçekte 32 bit yarımlık için muhtemel fark ($\Delta m_{17} \parallel \Delta m_{16}$) Sonra bilinmeyen anahtar altındaki şifreli metin arasındaki farkları belirlemek için m ve m' şifrelenir ve muhtemel fark için sonuçlar karşılaştırılır.

$E_K(m) \oplus E_K(m') = (\Delta m_{17} \parallel \Delta m_{16})$

Bütün ara turlardaki muhtemel farklar bulunarak alt anahtarların bitleri tahmin edilir.

Doğrusal(lineer) Kriptanaliz

Diğer bir yöntem ise doğrusal kriptanalizdir. Doğrusal kriptanalizde DES için 2^{47} bilinen şifresiz metin ile ile 2^{47} seçilen şifresiz metin karşılaştırılarak anahtar bulunabilir. Her ne kadar bu küçük bir iyileştirme olsada doğrusal kriptanaliz kullanılabilir.

Bu yöntemin esası, eğer şifresiz metin bloğunun bitlerine birbiri ile XOR işlemi uygular, şifreli metin bitlerini de birbiri ile XOR'lar ve sonra sonuçlara da XOR işlemi uygulanırsa anahtar bitlerinin bazılarının XOR'lanarak elde edildiği tekbir bir bitlik sonuç elde edilir. Bu doğrusal bir yaklaşımdır ve bir p olaslılığı ile sağlanır. Eğer bu olasılık $p \neq 0,5$ ise, bu işlem anahtarın bulunması için kullanılabilir. Toplanan şifresiz metinler ve karşılığında atanan şifreli metinler anahtar bitlerinin tahmin edilmesi için kullanılabilir. İşlemler aşağıda matematiksel olarak açıklanmıştır.

n bit şifresiz metin ,şifreli metin ve m bit anahtar alalım.

P[1], P[2], ……., P[n] , ve C[1], C[2], ……., C[n]

K[1], K[2], …….., K[m] olsun ve;

A[i,j,……..,k] = A[i] ⊕ A[j] ⊕ ……⊕ A[k] tanımlansın.(bitler bir biri ile XOR'lanır)

Doğrusal kriptanalizin amacı, aşağıdaki şekilde etkin bir lineer denklem bulmaktır.Bu denklemin sonucunun 1 olma olasalığı p'dir. Öyleki; $p \neq 0,5$ ihtimali 0,5 ten farklı olsun.

$P(\alpha_1, \alpha_2,……., \alpha_a,) \oplus C[\beta_1, \beta_2,……., \beta_b] = K[\gamma_1, \gamma_2,……., \gamma_c]$

Burada x=0,1; $1 \le a$; $b \le n$, $1 \le c \le m$ ve α,β ve γ terimleri sabit bit konumlarını belirtir.

Önce önerilen bağıntı tanımlanır(büyük miktardaki açık ve şifreli metin için) Eğer sonuç çoğunda 0 ise $K[\gamma_1, \gamma_2,……., \gamma_c] = 0$ dır. Eğer çoğunda 1 ise $K[\gamma_1, \gamma_2,……., \gamma_c] = 1$. Bu bize anahtar bitleri üzerinde doğrusal bir denklem verir. Daha fazla bağıntı bulmayı deneyerek anahtar bitleri tahmin edilebilir.

4.5 Zayıf Anahtarlar (Weak Keys):

Algoritmanın her bir turu için başlangıçtaki anahtar değiştirilerek bir alt-anahtar elde edilir. Başlangıçtaki anahtarlar zayıf anahtarlardır. Hatırlanacağı gibi başlangıç değeri iki yarım parçaya bölünmekte ve her bir yarım bağımsız olarak değiştirilmekteydi. Her bir yarımdaki tüm bitler 0 veya 1' den oluşuyorsa, o zaman algoritmanın herhangi bir dönüşümü için kullanılan anahtar, algoritmanın bütün dönüşümleri için de aynı olacaktır. Bu olay, anahtar tamamen 1' lerden, tamamen 0' lardan veya bir yarısı 1' lerden diğer yarısı 0' lardan oluşuyorsa meydana gelir.

Tablo 4.10' da hexadecimal olarak 4 zayıf anahtar örneği gösterilmiştir. (Sekizinci bitler parity biti olarak kullanılmaktadır.)

Zayıf Anahtar Değeri				Gerçek Anahtar
0101	0101	0101	0101	0000000 0000000
1F1F	1F1F	0E0E	0E0E	0000000 FFFFFFF
E0E0	E0E0	F1F1	F1F1	FFFFFFF 0000000
FEFE	FEFE	FEFE	FEFE	FFFFFFF FFFFFFF

Tablo 4.10 DES Zayıf Anahtarlar

Tur Sayısı :

Niçin 16 tur? Niçin 32 değil? Beş turdan sonra her şifrelenmiş text biti, her plaintext bitinin ve her anahtar bitinin bir fonksiyonudur. Sekiz turdan sonra şifrelenmiş text, her plaintext ve her anahtar bitinin tamamen rasgele fonksiyonudur.

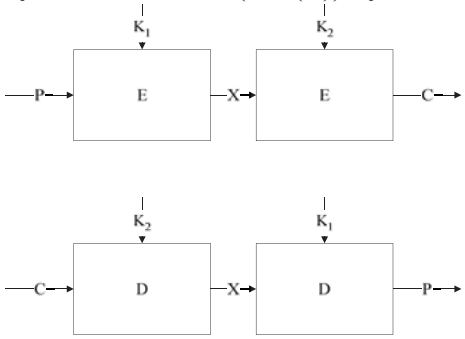
DES 16' dan daha az turda gerçekleştiği zaman, brute force saldırıları olarak bilinen saldırılarla daha kolay ve verimli bir şekilde kırılabilir.

4.6 DES'in Farklı Şekilleri :

4.6.1 Double DES :

DES'in iki ayrı anahtar ile arad arda şifrelemede kullanılmasıdır. Bu durumda anahtar uzunluğu 112 bit olacaktır. Brute-Force saldırılarına karşı 2^{112} adet anahtar kombinezonunun denenmesi gerecektir.

Şifreleme $C = E_{K2}(E_{K1}(P))$ Deşifreleme $P = D_{K1}(D_{K2}(C))$ şeklinde olacaktır.



Şekil 4.10. Double DES

Ancak bu şekilde olan şifrelemede anahtar uzunluğu artmasına karşın, Ortada karşılaşma(meet in the middle) saldırılarına zayıflığı vardır.

Ortada Karşılaşma(Meet in the middle attacak) saldırısı

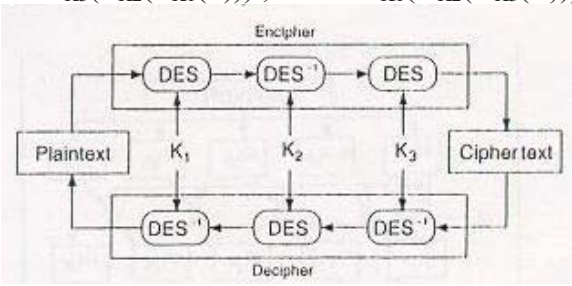
Şekil 6.16.'dan görüldüğü gibi X değerinin hesabı aşağıdaki şekilde yapılabilir.

$X = E_{K1}(P) = D_{K2}(C)$

Verilen bir (P,C çifti ile P, K_1 'in bütün anahtar kombinezonları(2^{56}) ile şifrelenerek X'n değerine göre sıralanır. C yine K_2'nin bütün anahtar kombinezonları(2^{56}) ile deşifrelenerek X'n değerine göre sıralanır. Herikisindede aynı olan X'teki K_1 ve K_2 muhtemel anahtarlardır.

Bunun önüne 3lü DES uygulaması ile geçilebilir. 3DES, bir plaintext'e üç kere DES algoritması uygulayarak şifrelenmiş text elde edilme yöntemidir. (Şekil 4.11) 3DES iki veya üç ayrı anahtar kullanılarak yapılabilir.

$$C = E_{K3}(E_{K2}(E_{K1}(P))) ; \quad P = D_{K1}(D_{K2}(D_{K3}(C)))$$



Şekil 4.11 Triple DES

4.6.2 CRYPT(3) :

UNIX sistemler üzerinde bulunan DES tabanlı algoritmadır. Aslında passwordlar için bir yollu fonksiyon gibi kullanılır, fakat bazen şifreleme için de kullanılır.

4.6.3 Generalized DES :

Generalized DES (GDES), algoritmayı kuvvetlendirmek ve DES'i hızlandırmak amacıyla tasarlanmıştır. Hesap miktarı sabit iken blok boyutu arttırılmıştır.
DES varyanslarına ek olarak DESX, RDES, s^n DES de verilebilir.

4.6.4 IDEA(International Data Encryption Algorithm)

Simetrik blok şifreleme algoritması olan IDEA 1991'de Swiss Federal Institute of Technology 'de geliştirilmiştir. 128 Bit anahtar uzunluğu kullanılır. IDEA alt anahtar üretim ve tur fonksiyonları bakımından DES'ten farklıdır. S-boxes kullanılmaz. XOR , 16 bit tam sayı toplama ve 16 bit tamsayı çarpma matematik işlemlerini kullanır. Kriptanalizi zor olan bir algoritmadır. Alt anahtar üretim algoritması sadece dairesel kaydırma üzerinedir, fakat her bir sekiz turda altı alt anahtar üreten karmaşık bir yapıya sahiptir. İlk 128 bit anahtar kullanan algoritma olduğu için kriptoanalistlerin üzerinde çok çalıştıkları bir algoritmadır.

4.6.5 BlowFish

Blowfish , bağımsız kriptocu olan Bruce Schneier tarafından 1993'te geliştirildi, kısa zamanda DES'e en popüler alternatif haline geldi. Kolay programlanabilen ve hızlı çalışan bir algoritmadır. Aynı zamanda 5K dan az bellekte çalışan çok karmaşık bir algoritmadır. Anahtar uzunluğu değişkendir ve 448 bit kadar olabilir. Pratikte 128 bit anahtar kullanılır ve 16 tur kullanır.
Blowfish DES gibi S-box ve XOR fonksiyonu kullanır fakat aynı zamanda ikili toplama da kullanır. Sabit S-boxes kullanan DES'in tersine, Blowfish anahtarın bir fonksiyonu olarak üretilen dinamik S-box kullanır.Blowfish'te alt anahtar ve S-box'lar, blowfish algoritmasının anahtar üzerinde tekrarlanarak uygulanmasıyla elde edilirler. Alt anahtar ve S-box'ların üretilmesi için Blowfish şifreleme algoritmasının toplam 512 kere icra edilmesi gerekir. Dolayısı ile çok sık gizli anahtar değişimi gerektiren uygulamalarda blowfish kullanılması uygun değildir.

4.6.6 RC5

RC5, 1994'te RSA asimetrik şifreleme algoritmasını geliştirenlerden birsi olan Ron Rivest tarafından geliştirildi. RC5 Aşağıdaki özelliklere sahiptir.

Donanım veya yazılım ile gerçeklenmeye uygundur.: Mikro işlemcilerde bulunan primitif hesaplama operatörlerine sahiptir.

Hızlılık : Basit ve kelime yönelimlidir. Temel işlemler bir anda verinin bütün kelimesi üzerinde yapılır.

Değişik kelime uzunluklu işlemcilere adapte edilebilirlik: bir kelimdeki bit sayısı RC5'te parametredir. Farklı kelime uzunluklu farklı algoritmalar oluşturur.

Değişken sayıda Tur : Değişken tur sayısı RC5'in diğer parametresidir. Bu parametre daha fazla hız ile daha fazla güvenlik arasında değişim yapar.

Değişken anahtar Uzunluğu : Anahtar uzunluğu RC5'in üçüncü parametresidir. Bu parametre de daha fazla hız ile daha fazla güvenlik arasında değişim yapar.

Basitlik : RC5 kolay programlama için basit bir yapıya sahiptir.

Düşük bellek Gereksinimi: Düşük bellek gereksinimi RC5'i smart kartlar ve sınırlı belleğe sahip diğer benzer cihazlarda kullanımını sağlar.

Yüksek Güvenlik : RC5 uygun parametreler ile yüksek güvenlik sağlar.

Veri bağımlı Döndürmeler: Verinin miktarına bağlı olarak döndürme gerçekleştirir. Bu algoritmanın kripto analistlere karşı gücünü artırır.

4.6.7 CAST-128

CAST 1997'de Entrust Teknolojiler'den Carlise Adams ve Stafford Tavares Tarafından geliştirilen bir tasarım prosedürüdür. Bir özel algoritma 8 bit artımlar ile 40 bitten 128 bit'e kadar değişen anahtar uzunlukları kullanır. CAST, DES'te kullanılanlardan daha uzun olan sabit S-boxlar kullanır. Bu S-boxların tasarımı Kriptoanaliste karşı önemlidir. CAST'taki alt anahtar üretimi diğer blok şifreleyicilerden farklıdır. Doğrusal olmayan S-boxlar kullanılarak alt anahtar üretimi yapılır. CAST-128'in diğer enteresan özelliği tur'dan tur'a değişen F tur fonksiyonudur.

Algoritma	Anahtar Uzunluğu	Tur Sayısı	Matematiksel İşlemler	Uygulamalar
DES	56 Bit	16	XOR, Sabit S-boxes	SET,Kerberos
Triple DES	112 veya 168 bit	48	XOR, Sabit S-boxes	Mali anahtar yönetimi, PGP, S/MIME
IDEA	128 Bit	8	XOR, Toplama, Çarpma	PGP
Blowfish	Değişken, 448 bit	16	XOR, Değişken S-Boxes, Toplama	
RC5	Değişken 2048 Bit	Değişken 255	Toplama, Çıkartma, XOR, Döndürme	
CAST-128	40-128 bit	16	Toplama, Çıkartma, XOR, Döndürme, Sabit S-boxes	PGP

Tablo 4.11. Değişik Simetrik Kriptolama algoritmalarının özellikleri

Gelişmiş Blok şifreleme algoritmalarının Özellikleri

- Değişken anahtar uzunluğu
- Karmaşık aritmetik işlemler
- Veriye bağlı döndürme
- Anahtar bağımlı S-box
- Çok uzunluklu anahtar düzenleme algoritmaları
- Değişken şifresiz/şifreli metin blok uzunluğu
- Değişken tur sayısı
- Her bir turda her iki yarımlık veriye işlem
- Değişken F fonksiyonu
- Anahtar bağımlı döndürme

4.7 Blok Şifreleme Çalışma modları

Simetrik blok şifreleme bir zaman diliminde bir bitlik blok veriyi işler. Veri şifreleme ve üçlü veri şifreleme algoritmalarında blok uzunluğu 64 bittir. Daha uzun veriler 64 bitlik bloklara bölünürler. ECB(Electronic codebook) modunda şifresiz metin 64 bitlik bloklar halinde işleme aynı anahtar ile girer. Codebook terimi, verilen bir anahtar için her bir 64 bitlik bloğa karşılık sadece bir şifreli metin olduğu için kullanılır.

Bu modda eğer 64 bitlik bloklar metin içerisinde tekrarlanırsa bunlar için aynı şifreli metin üretilecektir. BU ise ECB modu kriptanaliz açısından güvensiz yapar. Eğer metin her zaman önceden tanımlı alanlar ile başlarsa kriptoanalist açık ve şifreli metin çiftini elde edebilir. Eğer mesaj tekrarlanan elemanları içerirse bu tekrarlama periyodu da kripto analist tarafından tanınabilir.Bunun üstesinden iki alternatif olan CBC ve CFB modlar ile gelinebilir.

4.7.1 CBC(Cipher Block Chaining Mode)

Bu modda (CBC) o andaki şifresiz metin bloğu ile bir önceki şifreli metin bloğu, XOR mantıksal işlemine tabi tutulur. Her bir blok için aynı anahtar kullanılır. Böylece şifreli metinde tekrarlanan 64 bitler olmaz.

Deşifreleme için her bir şifreli blok deşifreleme algoritmasından geçer. Sonuç açık metin bloğunu elde etmek için önceki şifreli metin ile XOR'lanır. Bunu görmek için aşağıdaki ifadeyi yazabiliriz:

$$C_i = E_K[C_{i-1} \oplus P_i]$$

Burada $E_K[X]$, X'in K anahtarı kullanılarak şifrelenmiş şekli ve $\oplus$ ise XOR işlemidir. Sonra,

$$D_K[Ci] = D_K[E_K(C_{i-1} \oplus P_i)]$$

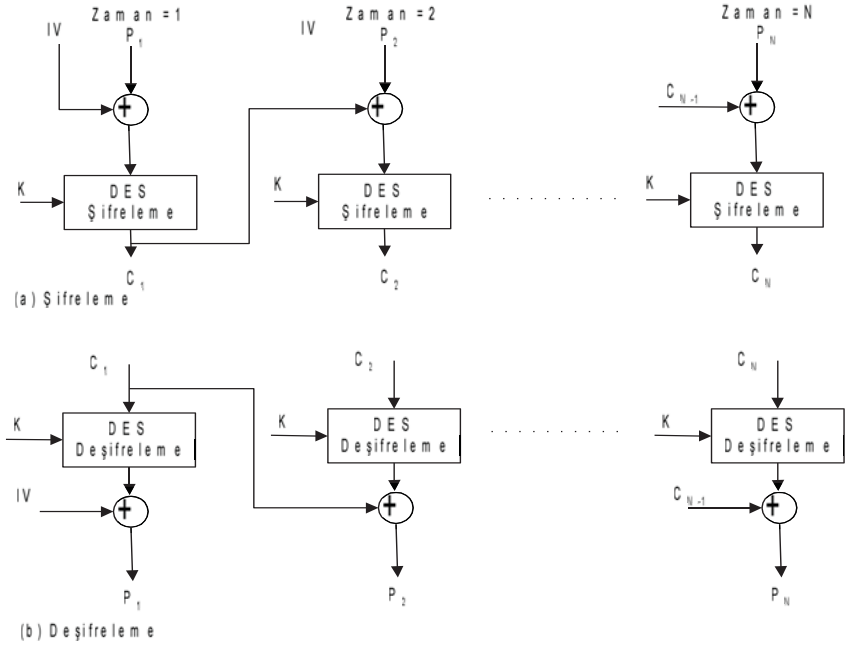
$$D_K[Ci] = (C_{i-1} \oplus P_i)$$

$$C_{i-1} \oplus D_K[Ci] = C_{i-1} \oplus C_{i-1} \oplus P_i = P_i$$

Şekil 4.12 'de görüldüğü gibi, ilk şifreli bloğu elde etmek için başlatma vektörü(IV) ilk açık metin bloğu ile XOR işlemine tabi tutulur. Deşifrelemede ise, ilk şifresiz bloğu elde etmek için IV deşifreleme algoritmasının çıkışı ile XOR'lanır. Burada başlatma vektörü (IV) güvenlik için önemlidir. Bu nedenle şifre gibi korunması gerekir. İlk bloğun şifrelenmesi aşağıdaki ifadede gösterilmiştir.

$$C_1 = E_K(IV \oplus P_i)$$

$$P_1 = IV \oplus D_K(C_1)$$



Şekil 4.12. CBC (Cipher Block Chaining Mode)

4.7.2 CFB(Cipher Feedback Mode)

DES tasarımı 64 bitlik blok şifrelemeyi kullanır. Bununla birlikte CFB modu ile DES'i dizi şifreleyici haline dönüştürmek mümkün olmaktadır. Bu yapıda herbir karakterin 8 bit olduğu varsayımı ile 8 bitlik alt bloklar ile yapılan şifrelemede karakter bazında dizi şifrelemesi gerçekleştirilmiş olmaktadır.

Yine ilk blok için başlangıç vektörü IV kullanılır. IV'ninde ötelenmesiyle 8 bitlik alt vektör ile ilk blok şifrelemesi gerçekleştirilir.

Deşifreleme için düz metin birimini elde etmek için alınan şifreli metin biriminin şifreleme fonksiyonunun çıkışı ile XOR'lanması dışında aynı tasarım kullanılır. Yani deşifrelemede de şifreleme fonksiyonu kullanılır. $S_j(X)$, X'in en yüksek anlamlı bitleri olarak tanımlayalım.Buradan,

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

Bu nedenle,

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

Elde edilir. Aynı şekilde sürecin alt adımlarında işlem devam eder.

4.8 AES (Advanced Encryption Standard)

3DES algoritması her nekadar 168 bitlik anahtar kullanıyor ve brute-force saldırılarına karşı yeterli güvenlik sağlıyor ise de üç adet DES'in ard arda çalışması nedeniyle yavaş bir algoritmadır. Bu nedenle NIST 1997'de 3DES'in yerini alacak daha hızlı ve güvenli bir simetrik şifreleme algoritması geliştirilmesini önerdi. Bu çağrı sonunda Belçikadan Dr. Joan Daemen ve Dr.Vincent Rijmen geliştirdiği Rijndael algoritması AES olarak kabul edildi. AES'in önemli özellikleri aşağıda verilmişitir.

1 128 bit veri, 128/192/256 bitlik anahtar uzunluğuna sahiptir.
2 Feistel networkü yerine iteratif olarak çalışır
3 Veriyi dört baytlık dört sütunluk bloklar halinde işler.
4 Herbir tur'da veri bloğunun tamamı üzerinde işlem yapar.
5 Basit, bilinen saldırılara karşı dirençli, birçok işlemcide hızlı ve kod basitliği sağlayacak şekilde tasarlanmıştır.

Şekil 4.13 'de blok diyagramı gösterilen AES'in çalışması aşağıda özetlenmiştir.

1 DES(Feistel) mimarisinde veri bloğunun yarısı diğer yarısını modifiye etmekte kullanılır, sonra yer değiştirilir. AES(Rijndael) mimarisinde her iki yarı da paralel şekilde işlenir.
2 Sağlanan giriş anahtarı 40 adet dörtlük 32 bitli wordler şeklinde genişletilir w[i]. Dört farklı 128 bitlik kelime herbir turda tur anahtarı olarak kullanılır.
   Herbir turdaki dört farklı evrede, bir permutasyon ve üç yer değiştirme kullanılır.
   - Substitute baytları: bloğun bayt bayt yer değiştirmesi için S-box'lar kullanılır(her bayt için bir S-box).
   - Shift-Rows: Basit bir permutasyon(bayt'ları grup ve sütunlar arasında değiştirme)
   - Mix columns: $GF(2^8)$ üzerinde yapılan aritmetiği kullanarak yer değiştirme
   - Add-Round key: Basit bit bit XOR işlemi(mevcut blok ve genişletilen anahtarın turdaki hali ile)
3 Yapı çok basit: Şifreleme ve deşifreleme için şifreleyici add round key evresi ile başlar. Herbiri 4 evre olan 9 tur ile devam eder.
4 Sadece add round key evresi anahtar kullanır. Bu nedenle şifreleyici add round key evresi ile başlar ve biter.
5 Etki olarak add round key evresi bir Vernam şifreleyici gibidir ve çok zor değildir.Diğer üç evre birlikte confusion, diffusion ve doğrusal olmamayı sağlar. Fakat anahtar kullanmadıkları için güvenlik sağlamazlar.
6 Herbir evre kolaylıkla evrilebilir. $A \oplus A \oplus B = B$ gibi

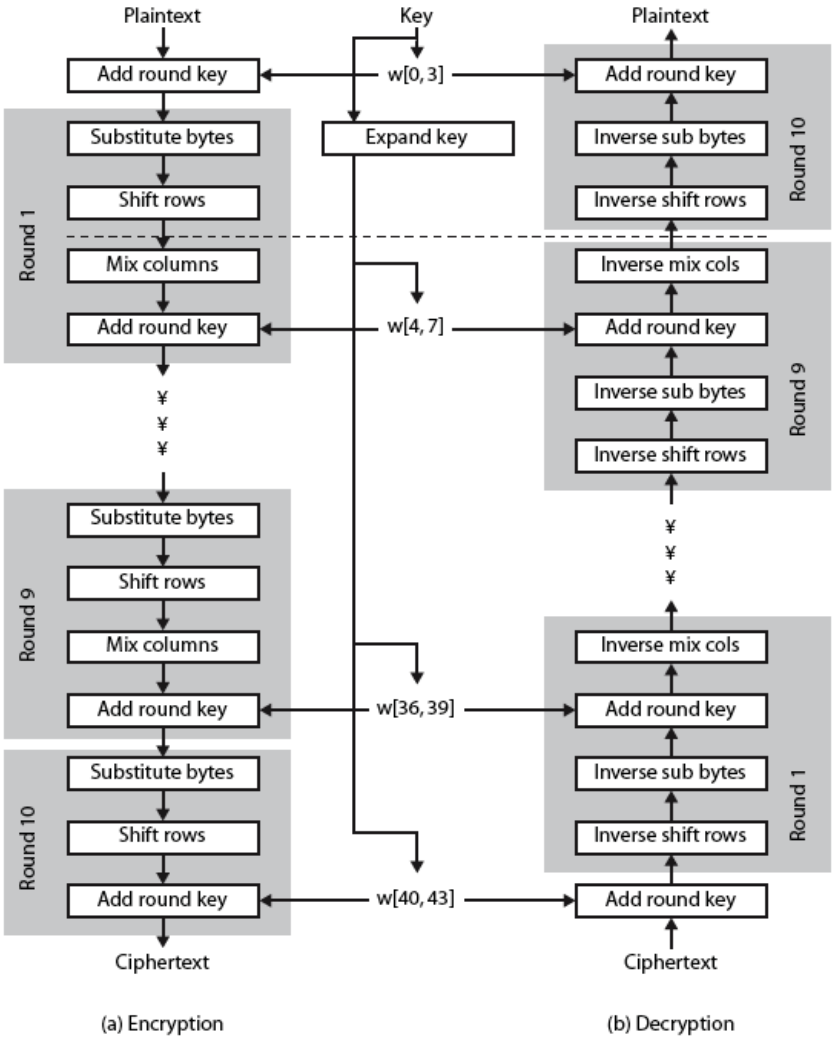
7 Çoğu blok şifreleyicide olduğu gibi deişfreleme algoritması anahtarı ters yönde genişletir. Bununla birlikte deşifreleme algoritması, şifrelemeye benzemez. Bu AES'in parçalı yapısının sonucudur.

8 Dört evre ters çevrilebilir şekilde kurulduğunda deşifrelemenin plaintext'i bulması sağlanır.

9 Son turda , şifreleme ve deşifrelemenin her ikisi de sadece üç evre içerir. Bunlar Substitute bayt, Shift columns ve add round key 'dir. Bu AES'in parçalı yapısının sonucudur ve şifreleyiciyi evrilebilir yapmayı gerektirir.

10 Deşifreleme evreleri:

- Inverse-Shift-Rows:
- Inverse Sub bayts:
- Inverse Mix columns:



Şekil 4.13 : AES Şifreleme ve Deşifreleme adımları

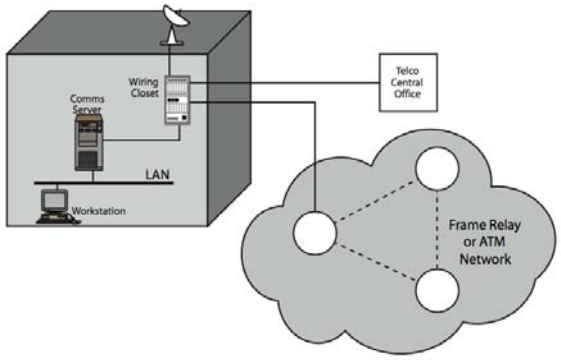
4.9 Gizli anahtarlı (simetrik) kriptosistemlerin Güvenliği :

Geleneksel olarak simetrik şifreleme mesaj gizliliğini sağlamak için kullanılır.
İki farklı şifreleme alternatifi vardır.

a. Link Şifreleme : Şifreleme her bir iletişim bağlantısı üzerinde bağımsız olarak yapılır. Bağlantılar arasındaki trafiğin deşifrelenmesi gerekir. Birçok cihaz ve birçift anahtar gerektirir

b. Uçtan uca şifreleme : Şifreleme orijinal kaynak ve son varış noktası arasında yapılır. Her iki uçta paylaşılmış anahtarlar ve cihazlar gerekir.

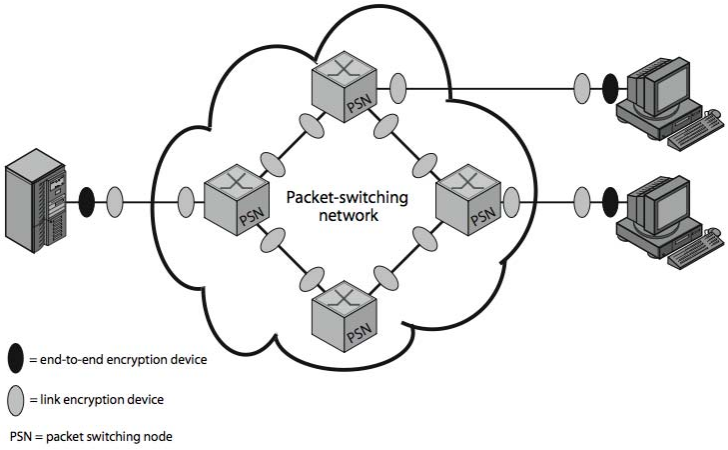
Şekil 4.14'de gösterilen haberleşme ağı'nda açıklık noktaları belirtilmiştir. YAŞ'ne bağlı olan bir iş istasyonunun gönderdiği mesajlar YAŞ'ın özelliği itibarı ile dinlenmeye müsaittir. Haberleşme sunumcusuna erişim hakkı elde eden bir saldırgan ağ trafiğini dinleyip analiz edebilir. YAŞ 'nin dışında bir yönlendirici veya çevirmeli modem ile dış ağa bağlantı olabilecektir. Bunların bağlantı noktaları zayıf noktalardır. Dış ağdaki herhangi bir haberleşme bağlantısı saldırıya açık yerlerdir. Böylece saldırıya açık birçok nokta bulunduğu görülmektedir.



Şekil 4.14. Açıklık noktaları

4.9.1 Bağlantılara karşı uçtan uca Şifreleme

İletişimde şifreleme için iki yöntem düşünülebilir. Herbir bağlantı yı ayrı ayrı şifrelemek ve uçtan uca haberleşmeyi şifrelemek Şekil 4.15'de bir paket anahtarlamalı ağ'da bağlantıların ve uçtan uca haberleşmenin şifrelenmesi gösterilmiştir



Şekil 4.15. Paket anahtarlamalı ağ'da şifreleme

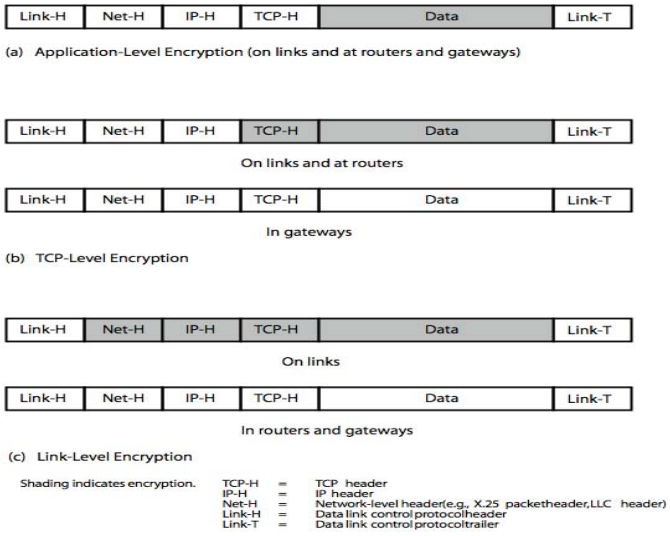
Uçtan uca haberleşme kullanıldıığ zaman başlık şifresiz olarak bırakılmalıdır. Böylece ağ yönlendirme bilgisini doğru olarak sağlayabilir.
Bu nedenle her nekadar, içerik şifrelensede, trafik izi akışını anlamak mümkündür
İdealde heriki şifrelemede

Uçtan uca şifreleme, mevcut veri hattı üzerindeki veri içeriğini şifreler ve kimlik doğrulama sağlar.

Bağlantı şifreleme ise trafik akışının gözlenmesini engeller
OSI referans modelinin değişik katmanlarında şifreleme fonksiyonu sağlanabilir

Katman 1 ve 2'de bağlantı şifreleme
Katman 3,4,6 ve 7'de uçtan uca şifreleme
Bilgi şifrelenirken anahtar ve içerik ile birlikte daha karmaşık hale gelir.

Şekil 4.16'da gösterilen protokol seviyelerindeki şifrelemelerde üst katmanlarda daha az verinin şifrelendiği, alt katmanlarda ise daha fazla verinin şifrelendiği görülmektedir.



Şekil 4.16: Şifreleme ve protokol seviyeleri arasındaki bağıntı.

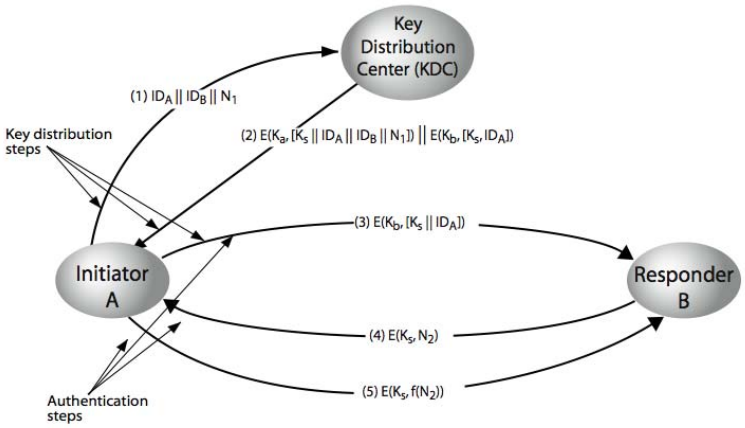
Trafik Analizi, iletişim grupları arasındaki haberleşme akışını gözlemektir.
Askeri ve ticari alanda faydalı olabilir
Gizli bir kanal oluştumakta kullanılabilir
Bağlantı şifreleme başlık detaylarını gizler fakat, ağ parçalarında ve uç noktalarda hala gözlenebilir
Trafik padding akışı anlaşılması güç haler getirir fakat, sürekli trafiğin maliyeti artar

4.10 Anahtar Dağıtımı

Şimetrik şifreleme yöntemlerinde ortak bir anahtar her iki grup tarafından paylaşılır. Problem, bu anahtarın güvenli olarak dağıtılmasıdır. Güvenli bir sistem sık sık anahtar dağıtım yönteminin kırılmasıyla etkisiz hale gelebilir
Verilen A ve B grupları için değişik anahtar dağıtımı alternatifleri olabilir
A anahtarı seçer ve fiziksel olarak B'ye iletir.
Üçüncü şahıs anahtarı seçer , A ve B'ye dağıtır
Eğer A ve B önceden haberleşiyorsa,önceki anahtarı kullanarak yeni anahtarı şifreler
Eğer A ve B , C ile birlikte güvenli bir iletişim kanalına sahipse, C anahtarı A ve B arasında iletir
Tipik olarak anahtarların bir hiyerarşisi vardır.
Oturum anahtarı, Herbir oturum için kullanılır. Ağdaki N adet hostun kurabileceği oturum sayısı N(N-1)/2 adettir. Yani N(N-1)/2 adet oturum anahtarı kullanılabilecektir.
Oturum anahtarı;
Geçici anahtardır
Verinin kullanıcılar arasında şifrelenmesi için kullanılır.
Tek bir oturumda kullanılır ve sonra atılır.

Ana Anahtar
Anahtar dağıtım merkezi ile kullanıcılar arasında N adet tir.
Ana anahtar;

Oturum anahtarlarını şifrelemek için kullanılır
Kullanıcı ile anahtar dağıtım merkezi arasında paylaşılır



Şekil 4.17: Anahtar dağıtımı senaryosu

Merkezi olmayan anahtar dağıtımı
Merkezi olmayan anahtar dağıtımında, herbir uç sistem, oturum anahtarı dağıtımı için güvenli bir şekilde haberleşmesi gerekir. Böylece N adet uç sistemin konfigürasyonu için N(N-1)/2 adet ana anahtar gerekebilir.

Oturum anahtarı aşağıdaki adımlar ile sağlanır

- A,B 'den N_1 içeren bir mesaj ile oturum anahtarı ister
- B, ortak olan ana anahtar ile şifrelenmiş şekilde A'ya cevap verir. Mesajda B'nin seçtiği oturum anahtarı ve $f(N_1)$, (N_1+1) , N_2 bulunur
- Yeni oturum anahtarı ile A, $f(N_2)$ 'yi B'ye gönderir.

Böylece herbir düğüm ençok (N-1) ana anahtar saklamak zorunda kalır veya gerektiğinde üretilip kullanılır