

1 VERİ VE AĞ GÜVENLİĞİNE GİRİŞ

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

Bilgisayarlaşmanın artmasıyla birlikte, dosyaları ve bilgisayarda saklanan diğer bilgileri korumak gereği 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 güvenlige 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, ağdaki bütün birimleri kapsar.

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 kaybına neden olunur ve müşterinin mesaj göndermesi engellenebilir.

2. Saldırılar servisler ve Mekanizmalar.

Güvenlik saldırısı: Bir kuruluşun bilgi güvenliği saygınlığını azaltır. Engellemeye, Dinleme, Değiştirme ve yeniden oluşturma olarak 4 sınıf saldırısı vardır.

Güvenlik Mekanizması: Bir güvenlik atağının anlaşılması, korunma veya onarımdır.

Güvenlik Servisi: Veri işleme sistemi ve kuruluşun bilgi iletim sisteminin güvenliğini artırma servisidir. Servis güvenlik saldırılardan engeller ve servis sağlamak için çeşitli güvenlik mekanizması kullanır.

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 servisinin fonksiyonu, alıcıya mesajın

kaynağı konusunda güven vermektedir.

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 sınırlarıdır. Bu denetimi sağlamak için, her bir kişiye erişim hakkı verilmelidir.

Kullanıma hazırlık: Saldırıların bir kısmı kullanılabilirliğin azalması veya kaybolmasına neden olabilir. Saldırıların bir kısmı iyi niyetli olabilir, oysa bir kısmı sistemin kullanılabilirliğini engeller. Bu servis kullanılabilirliğin sürekli olmasını sağlamaya yönelikdir.

Güvenlik mekanizmaları

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

Saldırılar

Bilgi sistemini saldırılardan korumak için saldırıları tanımak gereklidir. Bu kapsamda tehdit(threat) ve saldırı(attack) terminlerini kısaca açıklaymak gereklidir. Tehdit, belirli durum, yetenek, veya olay olduğu anlarda güvenlik foksiyonunun yerine getirilmesini engelleyen potansiyel bir güvenlik bozucusu olduğu halde; saldırısı, 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 yerlestirmesi
- Haberleşme hattının dinlenilmesi
- Haberleşmenin engellenmesi
- Saldırgan tarafından oluşturulan diğer bir kullanıcıya ait bilgilerin alındığını açıklamak İletilen bilgilerin içeriğinin değiştirilmesi.

OSI Güvenlik Mimarisi

Bilgi güvenliğinde sistematik bir yaklaşım olarak X.800 OSI güvenlik mimarisi, yöneticilerin güvenlik organizasyonlarını düzenlemeleri için önemli bir yaklaşımdır. OSI yaklaşımı güvenlik servisleri, mekanizmalar ve saldırırlara yoğunlaşmıştır.

Güvenlik Servisleri

- Kimlik Doğrulama(Authentication)
- Erişim Denetimi(Access Control)
- Veri Gizliliği(Data Confidentiality)
- Veri Bütünlüğü(Data Integrity)
- İnkar edememe(Nonrepudiation)

Güvenlik Mekanizmaları

X.800 OSI güvenlik mimarisinde mekanizmalar iki grupta toplanmıştır., Kendine özgü güvenlik mekanizmaları

Şifreleme, Sayısal imzalar, Erişim denetimi, 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 türüdür.

Pasif saldırılar, mesaj içeriğinin ifşa edilmesi ve trafik analizidir. Veri içeriği değiştirilmediği için pasif saldırıları ortaya çartmak çok güçtür. Bu saldırılarından korunmak, anlamaktan daha uygun çözümlerdir.

Aktif Saldırılar, saldırganın kimliğini gizlemesi(masquerade), geri gönderme(replay), Mesajın değiştirilmesi(modification of message) ve servis durdurma(denial of service) dir.

Aktif saldırılar pasiflere göre zıt özelliklerdir. Aktif saldırılar tespit edilebilirler ve karşı önlem alınabilirler. Buna karşı aktif saldırıları tamamen önlemek çok zordur.

Ağ Güvenliği için bir model

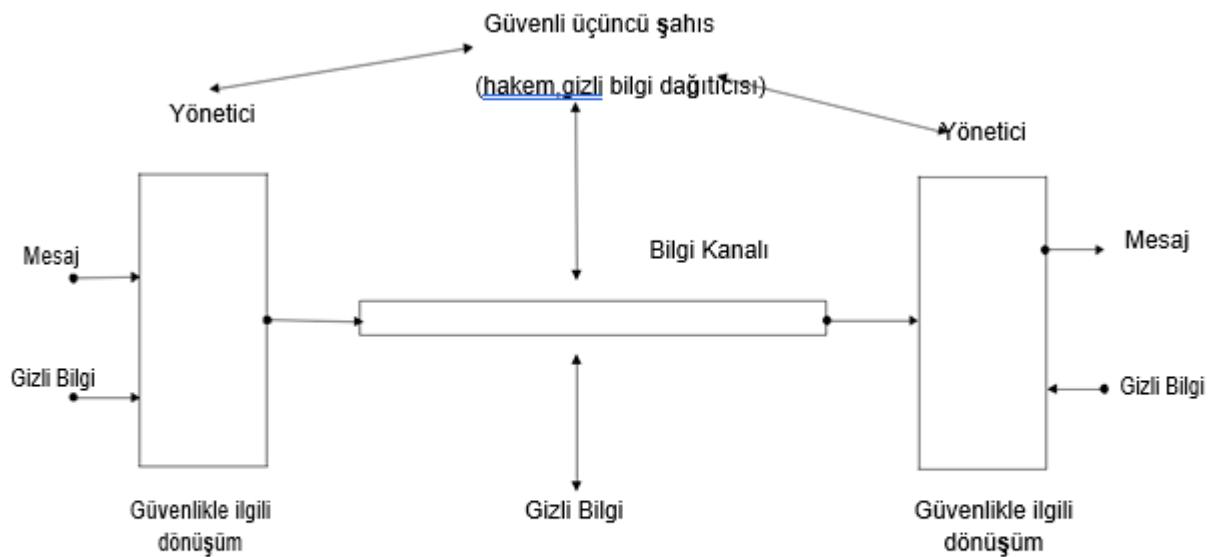
Ağ güvenliğinde genel bir model şekil 1.1'de gösterilmiştir. Gönderici ve alıcı mesajları gizli olarak ileterken, güvenli bir üçüncü şahıs gizli bilgilerin dağıticısı olarak hizmet vermektedir, her iki taraf arasında noter görevi de görmektedir.

Bu genel güvenlik mimarisi, güvenlik servislerinin tasarılarında dört temel işi gösterir.

Güvenlik ilişkili dönüşümler için bir algoritma tasarımlı

Algoritma ile kullanılacak gizli bilginin üretimi
Gizli bilginin dağıtımını ve paylaşımı için yöntem geliştirme

Güvenlik algoritmasını ve güvenlik servisini sağlayacak gizli bilginin kullanımını sağlayacak bir protokol belirleme.



Şekil-1.1. Ağ Güvenliği için Model

2 GÜVENLİK GEREKLİRİ VE KORUNACAK VARLIKLAR

28 Kasım 2020 Cumartesi 00:42

GÜVENLİK GEREKLİRİ VE KORUNACAK VARLIKLAR (Why Secure Your Network)

1. Giriş

Bilgisayar ağları, insanların bilgiye kolay ulaşımı, dolayısıyla çalışmalarındaki verimin artmasını sağlayan büyük bilgi ağlarıdır. Bilgiye kolay ulaşım için sunulan hizmetler (servisler, http, ftp, vs) aynı zamanda zarar verilebilme riski de taşımaktadır. Bilgisayar ağlarının sunduğu imkanlardan faydalananmak, fakat gelebilecek zararları en aza indirmek gereklidir. Fakat bu tedbir birtakım şeylerden ödün vermemizi gerektirir. Güvenliği ön plana almak, hızı da aynı oranda azaltmak anlamına gelmektedir.

Alınabilecek güvenlik önlemlerini tartışmadan önce güvenlik konusunun neden gerekliliğinin olduğunu, nelerin korunması gerektiğini anlaşılması daha faydalı olacaktır.

2. Korunacak varlıklar

Bir ağa güvenlik ile ilgili bir çalışma yapılmaya başlandığında ilk karar verilmesi gereken nelerin korunması gerektidir. Korunması gereken varlıklar üç ayrı ana başlıkta toplanabilir.

- Veriler
- Kaynaklar
- Saygınlık

Bu varlıklar ayrı ayrı inceleneciktir.

2.1 Veriler

Veriler, güvenlikle ilgili olarak üç özelliğe sahip olmalıdır;

Gizlilik: Verilerin, başkaları tarafından öğrenilmesi istenmeyebilir.

Bütünlük : Sahip olunan verilerin başkaları tarafından değiştirilmesi istenmeyebilir

Kullanıma hazırlık: Verilerin istediği zaman ulaşılabilir olup kullanıma hazır olması istenir.

Daha çok gizlilikle ilgili güvenlik üzerinde durulur. Gerçekten de bu konuda risk çoktur. Bir çok kişi ya da kuruluş için gizli bilgiler bilgisayar üzerinde tutulur. Bu bilgisayarların güvenliği de internet bağlantısı kopartılarak sağlanmaktadır. Bu şekilde bilginin gizliliği sağlanmış olabilir ama kolay ulaşılabilirlik ortadan kalkmış olur. Yani bir şekilde ağa bağlanılmalıdır. Bu durumda güvenlik politikaları belirlenerek, bilgilerin güvenliğinin sağlanması gerekmektedir.

2.2 Kaynaklar

Halka açık olan ağlara(İnternet'e) bağlanmakla riske atılacak ikinci şey, bilgisayar kaynaklarıdır. Başka insanların bir kuruluşu ait bilgisayardaki sabit diskte yer alan boş alanları kendi amacıyla kullanmak istemesi her ne kadar mevcut verilere zarar vermeyecek bir şey olsa da istenecek bir durum değildir. Bunun gibi diğer kaynakların da (işlemci,bellek, ...) başkaları tarafından kullanılması, kabul edilebilir bir şey olamaz

2.3 Saygılılık

Her kişi ya da kurumun saygınlığının ağ üzerinde de korunması önemlidir. Meydana gelebilecek güvenlik problemleri kişi ve kurumların doğrudan aleyhine olup kötü reklamdır. ağ üzerinde işlemler yapan bir kişinin, başka bir kişinin adını kullandığı düşünülürse, zarar verme durumunda doğrudan muhatap alınacak kişi saygınlığını kaybetme durumuyla karşı karşıya kalacaktır.

Genelde başka birinin hesabından girip sahte elektronik postalar atarak zarar verilir. Bunun sahte olduğunu kanıtlanması neredeyse imkansızdır. Böyle durumlarda, sahteciliği yapan kişinin kullandığı hesaba sahip kişi kadar kurum da zarar görür.

Halka açık ağlara(örn.internet) açılmayı düşünen kurumların eğitim ya da güvenlik politikası içinde, saygınlığın korunması için kişilere düşen güvenlik tedbirlerinin anlatılması gereklidir. Ayrıca periyodik olarak takibinin yapılması şarttır.

3. Bilgisayar Ağına Saldırı

2.3.1 Giriş

Internet'in doğuşu ve gelişimi arasında aslında çok kısa bir zaman aralığı vardır. Bu kadar hızlı bir büyümeyen olabileceği Internet'in doğuş yıllarda beklenmiyordu. Özellikle 1985 yılından sonra büyük yatırımlar yapıldı ve hızla yaygınlaştı. Ama bu hızlı gelişim birtakım konuların standartlarının tam olarak oluşturulmadan kullanıma geçirilmesinden dolayı bazı sorunları beraberinde getirdi. Özellikle de güvenlik sorunlarını ortaya çıkardı.

Güvenlik hemen her bilgisayar ağında öncelikli olarak düşünülmlesi gereken bir konudur. Halka açık ağ(Internet) ortamında ise çok daha önemlidir. Birçok ticari firma ya da başka kuruluşlar ürünlerini ve hizmetlerini Internet ortamına aktarmak istemektedirler. Ancak bu birtakım risklerin de alınması gerektiği anlamına da gelir. Değişik güvenlik mekanizmalarının bir arada kullanılmasıyla bu riski azaltmak mümkündür.

2.3.2 Saldırganlar

Saldırgan (Hacker), ağ üzerindeki genelde bazı servisler veren makinalara hiçbir hakkı olmadan erişip zarar veren kişidir. Bilgi hırsızı olarak da tarif edilebilir. Fakat ev ya da banka soyguncularından çok farklıdırlar. İyi görünümlü, sistemler hakkında çok bilgisi olan insanlardır. Genellikle sistemin bilinen açıklıklarından ve sistem yöneticisinin bilgisizliğinden faydalananırlar.

İstatistikler raporlarına göre saldırıların çoğunuun firma içerisinde yapıldığı tespit edilmiştir. İçeriden gelen saldırı, sistem sadece dışarıya karşı korunmalı durumda ise çok zarar verebilir.

Saldırganların büyük firmaların ağına girdikleri ve büyük ölçekte sistemlere zarar verdikleri bilinmektedir. Bunu genellikle eğlence, kendini göstermek ya da sisteme zarar vermek için yaparlar.

Saldırganlar iki genel türde toplanabilir:

Kötü niyetli saldırılar

Kötü niyetli olmayan saldırılar

2.3.2.1 Kötü niyetli saldırılar (Malicious hacker)

Sisteme gerçekten zarar vermek amacıyla girerler. Açığını buldukları sistemlere verebilecekleri en büyük zararı vermeyi amaçlarlar. Bu tür saldırılar genelde ekip halinde

çalışırlar. Kredi kartları kullanan sitelerden kart numaralarını ve parolalarını alıp kişi ve şirketlere büyük zararlar verebilirler.

Tablo2-1'de belirtilen bilgisayar korsanları, casuslar, teröristler ve profesyonel suçlular bu gruba girmektedir.

2.3.2.2 **Kötü niyetli olmayan saldırıcılar**

Sistemlere genellikle eğlenmek için saldırıda bulunurlar. Çok fazla zararlı olmayan tiplerdir. Hatta sistem yöneticiyi eksikliklerini ve sistemin zayıf noktalarını bu sayede görebilir. Bu tür saldırıcılar, bir saldırıcı grubuna üye değildir. Genellikle yaptıkları, kendilerine daha sonra kullanmak için hesap açmak ve sistemin zayıf olduğu yerleri belirten notlar koymaktır. Bu işi zevk için yapan insanları bu kategoriye sokabiliriz. Bu gruba girenler aşağıdaki tabloda **meraklılar** olarak belirtilmiştir.

Saldırıcı, kullandığı araçlar, sisteme erişim yolları ve amaçlarının ne olabilecekleri tablo 2-1'de özetlenmiştir.

Saldırıcılar	Araçlar	Erişim	Sonuç	Amaç	
Bilgisayar korsanları	Kullanıcı komutları	Gerçekleme zayıflıkları	Bilgi bozma	Finansal kazanç	
Casuslar	Komut dosyası veya Program	Tasarım zayıflıkları	Bilgi çalma ya da açığa çıkartma	Politik kazanç	
Teröristler	Araç takımı	Yapılardırma zayıflıkları	Hizmet çalma	Sosyal statüye	
Meraklılar	Dağıtık araçlar	İzinsiz erişim	Hizmet önleme	Zevk için	
Profesyonel suçlular	Veri dinleyici sistemler				

Tablo 2-1 Saldırıcılar ve amaçları

2.3.3 **Saldırı Türleri**

Bu bölümde son yıllarda internette kullanılan saldırısı yöntemlerine değinilecek ve sınıflandırılmaya çalışılacaktır. Saldırıların tanımları yapılacak ve sisteme verebileceği zararlar üzerinde durulacaktır. Saldırıya karşı alınabilecek önlemler güvenlik duvarı olmaksızın anlatılacak ve güvenlik duvarı düzeyinde yapılabilecekler açıklanacaktır.

Saldırıcılar sisteme ağ üzerinden ulaşabilecekleri için, ağa bağlı cihazlar her zaman saldırıyla açık durumdadır. Burada yapacakları, hedef makinaya ulaşmak, yazılım ve donanıma zarar vermek şeklinde olabilir. Şirkete ait veritabanına ulaşıp verilere erişebilir, değiştirebilir ya da silebilirler. Burada asıl olan, saldırının ne yapmak istediği. İşine yarayan kayıtları, dosyaları alabilir ve sisteme (yazılım, donanım) zarar verebilir. Verilen hizmetleri servis dışı bırakabilir. Sadece Internet bağlantısına zarar verebilir. Truva atı türünde programları bir şekilde hedef makinaya yükleyerek kullanıcıyı takip edebilir.

2.3.3.1 **Saldırıların Sınıflandırılması**

Bilgisayar ve ağ saldıruları için değişik sınıflandırmalar yapılmıştır. Aşağıda süreçsel ve

işlemsel sınıflandırmalar anlatılacaktır.

2.3.3.1.1 Süreçsel Sınıflandırma

Internet'te gerçekleştirilen veri transferiyle ilgili güvenlik sorunları dört kategoriye sokulabilir.

Engelleme: Sistemin bir kaynağı yok edilir veya kullanılamaz hale getirilir. Donanımın bir kısmının bozulması iletişim hattının kesilmesi, veya dosya yönetim sisteminin kapatılması gibi.

Kriptografi ve Bilgisayar Güvenliği Ders Notları(içindekiler)

1	VERİ VE AĞ GÜVENLİĞİNE GİRİŞ(IntroductIon to data and network securIty).....	2
1.1	Bazı Güvenlik Tecavüzleri.....	2
1.2	Saldırılar servisler ve Mekanizmalar.....	2
1.3	Güvenlik Servis özellikleri aşağıda açıklanmıştır.	2
1.4	Güvenlik mekanizmaları	3
1.5	Saldırılar	3
1.6	OSI Güvenlik Mimarisi.....	3
1.7	Güvenlik Mekanizmaları.....	3
1.8	Ağ Güvenliği için bir model	4
2	KRIPTOSİSTEMLER VE SİMETRİK ŞİFRELEME/ DEŞİFRELEME(Cryptosystems and Symmetric Encryption/Decryption)	5
2.1	Güvenliğin geliştirilmesi ihtiyacı.....	5
2.2	Ağ Üzerinde Yapılan Saldırı Türleri.....	5
2.3	İyi Doğrulama Gereklidir.....	5
2.4	Kriptolama	6
2.5	Temel Kavramlar	7
2.6	Kripto sistemler.....	8
2.7	Kriptografinin kısa Tarihçesi	10
3	Sayı Teorisine Giriş.....	12
3.1	Modüler Aritmetik	13
3.2	GF(p) (Galois Field) şeklindeki sonlu alanlar.....	15
3.3	Euler Totient fonksiyonu	16
3.4	GF(p) 'de üstel işlem	17
3.5	GF(p) 'de ayrik Logaritma Problemi	18
3.6	En Büyük ortak Bölüm(Greatest Common Divisor)	18
3.7	Teorem (Chinese Remainder Teoremi).....	19
3.8	Karmaşıklık Teorisi (saksı benzeri bakış).....	20
4	Gizli anahtarlı (simetrik) kriptosistemler:.....	22
4.1	Simetrik Şifreleme Algoritmaları.....	23
4.2	DES.....	25
4.3	DES' in Güvenliği :.....	31
4.4	Diferansiyel ve Doğrusal(Lineer) Kiriptoanaliz.....	31
4.5	Zayıf Anahtarlar (Weak Keys):.....	33
4.6	DES'in Farklı Şekilleri :	33
4.7	Blok Şifreleme Çalışma modları	36
4.8	AES (Advanced Encryption Standard)	37
4.9	Gizli anahtarlı (simetrik) kriptosistemlerin Güvenliği :	38
4.10	Anahtar Dağıtımı.....	40
5	AÇIK ANAHTARLI KRIPTOSİSTEMLER VE SAYISAL İMZALAR (Public Key Cryptosystems and Digital Signatures)	42
5.1	Açık anahtarlı (asimetrik) kriptosistemler:.....	42
5.2	Açık anahtarlı Şifreleme sistemlerinde Anahtar Yönetimi	47
5.3	Eliptik Eğri Kriptografi.....	50
5.4	Mesaj Doğrulama ve Özetteleme Fonksiyonları (Hashing Functions).....	54
Mesaj	56	
5.5	Kimlik Doğrulama ve Sayısal İmzalar	58

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

Bilgisayarlaşmanın artmasıyla birlikte, dosyaları ve bilgisayarda saklanan diğer bilgileri korumak gereğiği açıktır. Özellikle, zaman-paylaşımı 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 güvenlige 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 gelmiş gibi E' ye gönderir. E' de bu şekliyle kullanıcıların yetkilendirilmelerini günceller.
- Kullanıcı F, aldığı bir mesajı değiştirmek yerine kendi mesajını hazırlayarak sanki D'den gelmiş gibi E' ye gönderir. E aldığı bu mesaja göre yetkilendirme dosyasını günceller.
- Farklı işlemler için ,müsteriden gelmiş gibi borsa aracısına gönderilen bir mesaj ile para kaybı'na neden olunur ve müsterinin mesaj göndermesi engellenebilir.

1.2 Saldırılar servisler ve Mekanizmalar.

1. **Güvenlik saldırısı:** Bir kuruluşun bilgi güvenliği saygılığını 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 iletişim 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 saldırılarından 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 servisinin fonksiyonu, alıcıya mesajın kaynağı konusunda güven vermektedir.
- **Bütünlük:** Mesajın bütünlüğünü sağlar. Mesajın tamamının değişmemesini temin eder.
- **İnkar edilemem:** Gönderici veya alıcının iletilen bir mesajı inkar etmemesini sağlar.

- **Erişim Denetimi:** Erişim denetimi ağ güvenliğinde, host sistemlere ve uygulamalara haberleşme bağlantıları ile erişimi sınırlıdır. Bu denetimi sağlamak için, her bir kişiye erişim hakkı verilmelidir.
- **Kullanıma hazırlık:** Saldırıların bir kısmı kullanılabilirliğin azalması veya kaybolmasına neden olabilir. Saldırıların bir kısmı 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 gereklidir. Bu kapsamda tehdit(threat) ve saldırı(attack) termilerini kısaca açıklamak gereklidir. **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 yerlestirmesi

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 organizasyonlarını düzenlemeleri için önemli bir yaklaşımdır. OSI yaklaşımı güvenlik servisleri, mekanizmalar ve saldırırlara yoğunlaşmıştır.

Güvenlik Servisleri

Kimlik Doğrulama(Authentication)

Erişim Denetimi(Access Control)

Veri Gizliliği(Data Confidentiality)

Veri Bütünlüğü(Data Integrity)

İnkar edememe(Nonrepudiation)

1.7 Güvenlik Mekanizmaları

X.800 OSI güvenlik mimarısında 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 saldırırıları

X.800 mimarisinde güvenlik saldırırıları pasif ve aktif saldırırılar olmak üzere iki türüdür.

Pasif saldırırılar, mesaj içeriğinin ifşa edilmesi ve trafik analizidir. Veri içeriği değiştirilmemiş için pasif saldırırıları ortaya çkartmak çok güçtür. Bu saldırırılardan korunmak, anlamaktan daha uygun çözümlerdir.

Aktif Saldırırılar, saldırganın kimliğini gizlemesi(masquerade), geri gönderme(replay), Mesajın değiştirilmesi(modification of message) ve servis durdurma(denial of service) dir.

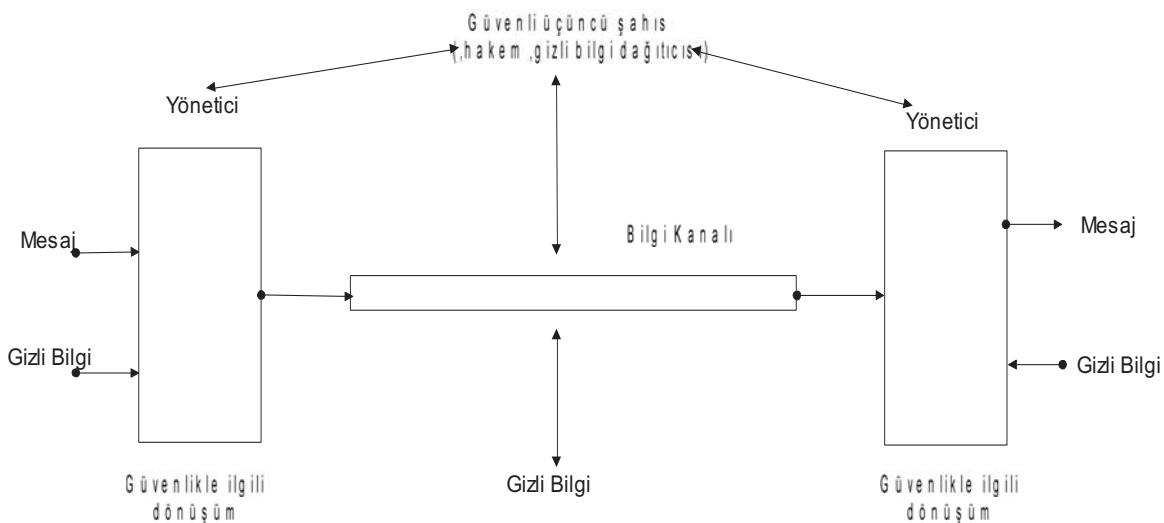
Aktif saldırırılar pasiflere göre zıt özelliktedirler. Aktif saldırırılar tespit edilebilirler ve karşı önlem alınabilirler. Buna karşı aktif saldırırıları tamamen önlemek çok zordur.

1.8 Ağ Güvenliği için bir model

Ağ güvenliğinde genel bir model şekil 1.1'de gösterilmiştir. Gönderici ve alıcı mesajları gizli olarak ileterken, güvenli bir üçüncü şahıs gizli bilgilerin dağıtıcısı olarak hizmet vermektedir, her iki taraf arasında noter görevi de görmektedir.

Bu genel güvenlik mimarisi, güvenli servislerinin tasarımda dört temel işi gösterir.

1. Güvenlik ilişkili dönüşümler için bir algoritma tasarımlı
2. Algoritma ile kullanılacak gizli bilginin üretimi
3. Gizli bilginin dağıtımını 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 SYMMETRIC ENCRYPTION/DECRYPTION)

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 üç 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 dağıtilır.
- 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. İfşaatt(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 gelmiş 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.

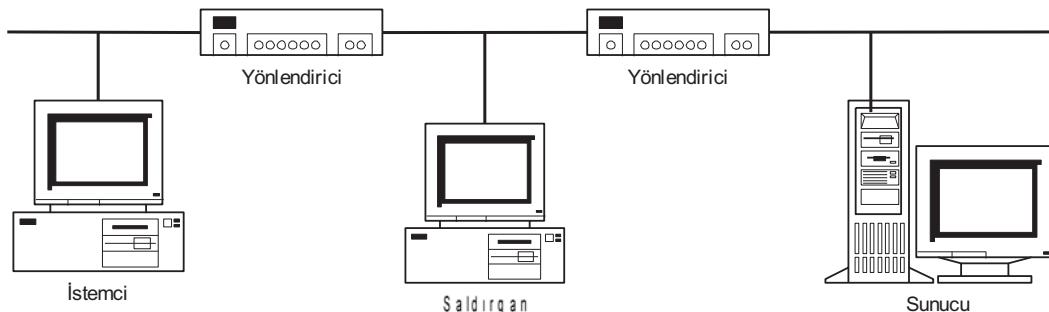
6.Zamanlamayı Değiştirme(Timing Modification): Mesajları geciktirme veya yeniden yollama. Bir bağlantı orijinal uygulamada bütün oturum veya mesajların bir kısmı ya önceki geçerli bir oturumun bir tekrarlanan sırası veya sıradaki kısmı mesajlar olarak geciktirilebilir veya tekrar gönderilir.

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

2.3 İyi Doğrulama Gereklidir.

İyi doğrulama gerektiği açıklıktır. Açık metin olarak logon bilgisini iletken 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 ataka 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 gözlemlerebilir. 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 iletişiminden önce ve sonra doğrulanması gerektiği açıklıdır. 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 önemnen en emin yolu verileri şifreleyerek iletmemektir.

2.4 Kriptolama

Bilgisayar ağlarının ve haberleşme sistemlerinin güvenliğinin sağlanması için kullanılan en önemli işlem, verilerin şifrelerek 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 bölüm ise kodlanmış veya şifrelenmiş olan gizli bilgilerin bulunmasına yönelik çalışmaların yapılması demek olan **kriptanaliz** 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 şeke 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) : Dönüşürülen metin

Şifreleyici(cipher) : Anlaşılır bir metni, yerlerini değiştirme ve/veya yerine koyma yöntemlerini kullanarak anlaşılır bir metni anlaşılmaz şeke dönüştürmek için kullanılan bir algoritma.

Anahtar(key) : Sadece gönderici ve alıcının bildiği şifreleyici tarafından kullanılan kritik bilgiler

Şifreleme(encrypt (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 şeke dönüştürme için bir algoritma

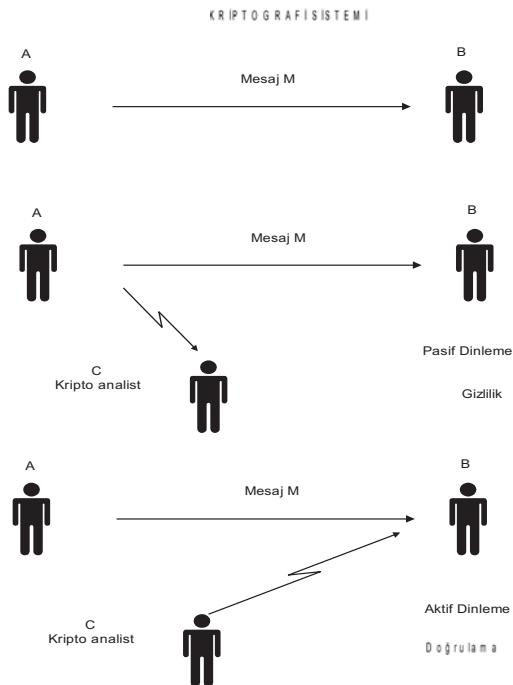
Şifreleme(Encryption) $c = E_K(m)$

Deşifreleme(Decryption) $m = D_K(c)$

E_K , kriptografik sistem olarak bilinen transformasyon ailesinden seçilir.

Anahtar denilen K parametresi anahtar uzayından seçilir

Diger 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 alfabetidir. **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 alfabetesinden alınmış fakat P den farklı bir diziliş gösteren elemanlardan oluşur. **K** ise daha önce bahsettiğimiz **Anahtar Uzayını** (Key Space) ifade eder. Anahtar yine A alfabetesindeki 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.

$$\begin{aligned} & \forall k \in K, D_k \in D \text{ fonksiyonuna uyan bir } E_k \in E \text{ fonksiyonu vardır. Öyle ki;} \\ & \forall E_k : P \rightarrow C \text{ ve } \forall D_k : C \rightarrow P \text{ ve her } x \in P \text{ için } D_k(E_k(x)) = x \end{aligned}$$

Kriptosistemler genel olarak aşağıdaki üç bağımsız özelliği göre sınıflandırılırlar.

- Şifresiz metinden şifreli metne dönüşüm için kullanılan işlemlerin tipi:** Bütün şifreleme algoritmaları yerine koyma(substitution) 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.
- 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.
- 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 karakteristiğilarındaki 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) saldırısı: 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 Metin (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
Seçilen 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.
Seçilen 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) dir.

- 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 gereklidir gibi), içinde şifre kırlamaz.

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	$2^{24} = 1.6 \times 10^7$	$2^{23} \mu\text{s} = 8.4$ saniye	$8.4 \mu\text{saniye}$
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8$ dakika	2.15 milisaniye
48	$2^{48} = 2.8 \times 10^{14}$	$2^{47} \mu\text{s} = 4.46$ yıl	2.35 dakika
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142$ yıl	10 saat
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24}$ yıl	$5.4 \times 10^{18} \text{ yıl}$
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36}$ yıl	$5.9 \times 10^{30} \text{ yıl}$
26 karakter permutasyonu	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12}$ yıl	$6.4 \times 10^6 \text{ yıl}$

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 misirlilar 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 İbraniler 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 tekerliği kullandı ve frekans analizinin prensiplerini açıkladı.

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

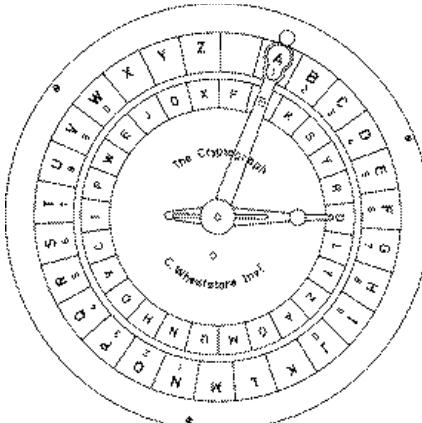
2.7.2 Makina Şifreleri

- 1790 larda geliştirilen **Jefferson cylinder**, herbiri rastgele alfabeli 36 adet diskten oluşmaktadır, disklerin sırası anahtarı oluşturmaktadır, mesaj ayarlanınca diğer satır şifreyi oluşturmaktadır.(Şekil 2.4)



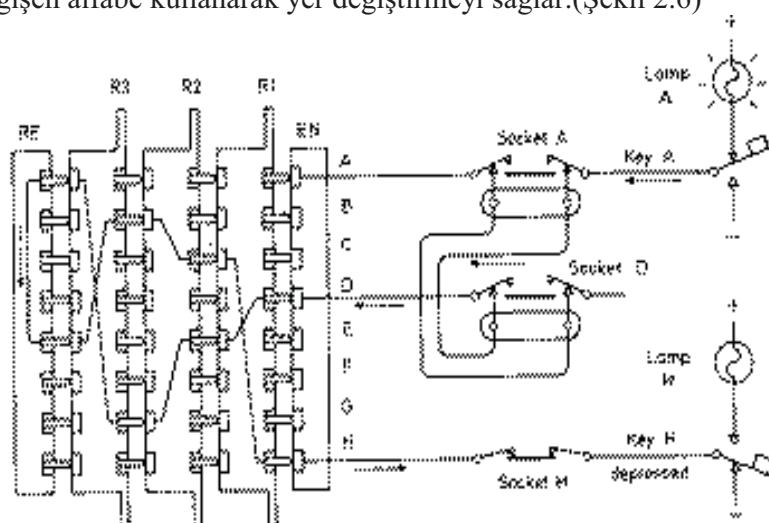
Şekil 2.4. Jefferson Cylinder

- **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)



Sekil 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)



Sekil 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, $a*e = e*a = a$ dir.

(G_4) G 'nin her bir elemanın tersi mevcuttur. $\forall a \in G$ için, G 'de bir a' vardır ve $a*a' = a'*a = e$ dir.

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 yazacağımız (ancak çarpma işlemi ile karıştırmamalıyız). Ayrıca aşağıdaki gibi x 'in üslerini tanımlayacağız.

$$n \in \mathbb{Z}^+ \text{ olmak üzere } x^n = x*x*\dots*x \text{ (n tane)}$$

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

Ayrıca etkisiz elemanı da şu şekilde tanımlarız: $x^0 = e$.

Herhangi bir $(G, *)$ grubun en belirgin özelliği büyülüğu yani grubun temelini oluşturan G kümesinin eleman sayısıdır. Buna $(G, *)$ grubunun order'i denir.

Tanım: $(G, *)$ grubunun order'i 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, $a*b = b*a$ dir.

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

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

$$g^j = g^l \Leftrightarrow j \equiv l \pmod{\text{ord}(g)}$$

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. yazabiliz. Aslında kümenin her bir elemanını a' nin 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 elemanın bir üssüdür ve $\langle g \rangle$ ifadesiyle gösterilirler. Eğer $\langle g \rangle = G$ 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 $\langle g \rangle$ nin düzeni 1 veya p sayısı olur.

Doğal olarak, diğer başka elemanlar da grubun üretecidir? sorusu aklımıza gelir. c elemanın 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, $ab = ba$ dir.

Eğer bir komutatif halka aşağıdaki aksiyomları sağlıyor ise integral domain dir.

(H_5) Çarpımsal etkisiz eleman. $\forall a \in R$ için, $a1 = 1a = a$ dir.

(H_6) Sıfır bölen olmaması $\forall a, b \in R$ ve $ab = 0$ ise ya $a = 0$ veya $b = 0$ dir.

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_1 den 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 $aa^{-1} = (a^{-1})a = 1$ dir.

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 aritmetiği”dir”

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:

$$a = k \cdot n + r$$

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, $a \bmod n$, a, n ile bölündüğünde kalan olarak tanımlanır. Böylece herhangi a tamsayı için,

$$a = [a/n]n + a \bmod n \text{ her zaman yazılabilir. (Örn: } 11 \bmod 7 = 4)$$

a ve b iki tamsayısi eğer $a \bmod n = b \bmod n$ iseler benzer modulo n olarak tanımlanır ve $a \equiv b \pmod{n}$ olarak yazılabilir.

Bölenler: Eğer sıfır olmayan bir b ve m tamsayısi için $a=mb$ şeklinde yazılabilse $b, a'yı böler$ 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 $a = \pm 1$ dir.
- Eğer $a|b$ ve $b|a$ ise $a = \pm b$ 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,

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

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

- Bir $a = b \bmod n$ eşitliği, a ve b aynı n ile bölündüğünde aynı kalanı verdiklerini ifade eder.
Örnek,
 - $100 = 34 \bmod 11$
 - Genellikle $0 \leq b < n-1$ dir.
 - $2 \bmod 7 = 9 \bmod 7$
 - b 'ye $a \bmod n$ 'nin kalanı denir.
- Tamsayı modulo n ile yapılan bütün aritmetikte bütün sonuçlar 0 ve n arasında olur.

3.1.1 Modül işleminin özelliklerı

Modül işlemi aşağıdaki özelliklere sahiptir.

Eğer, $n|(a-b)$ ise $a \equiv b \pmod{n}$ dir.

$a \equiv b \pmod{n}$, $b \equiv a \pmod{n}$ anlamına gelir.

$a \equiv b \pmod{n}$ ve $b \equiv c \pmod{n}$, $a \equiv c \pmod{n}$ anlamına gelir.

3.1.2 Modüler Aritmetik işlemleri

Toplama

$$(a+b) \bmod n = [(a \bmod n) + (b \bmod n)] \bmod n$$

Çıkartma

$$(a-b) \bmod n = [(a \bmod n) - (b \bmod n)] \bmod n$$

Çarpma

$$axb \bmod n = [(a \bmod n) \times (b \bmod n)] \bmod n$$

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

Bölme

$$a/b \bmod n$$

- 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 $2.3=1 \bmod 5$ bu nedenle $4/2=4.3=2 \bmod 5$ dir.

Özellikler :

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

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

Z_n kalanlar sınıfı olarak adlandırılır. Daha doğrusu, Z_n de her bir tamsayı bir kalan sınıfını temsil eder. $[r] = \{ a : a \text{ bir tamsayı; öyleki } ; a = r \bmod n \text{ dir.} \}$

Z_n içerisinde yapılacak modüler aritmetik işlemleri Tablo 3.2'deki özellikleri Z_n deki tamsayılar ile sağlanır. Z_n çarpımsal etkisiz eleman ile birlikte bir değiştirilebilir halka oluşturur.

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

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.
 - $a+/-b \bmod n = [a \bmod n +/- b \bmod n] \bmod n$
 - $(a.b) \bmod n = ((a \bmod n).(b \bmod n)) \bmod n$
- 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) 1 bir p asal sayısının n . kuvveti(p^n) olarak gösterilmelidir. Burada n pozitif bir tamsayıdır. Düzeni p^n olan bir sonlu alan, genellikle $GF(p^n)$ olarak yazılır. GF sonlu alanı ilk defa çalışan matematikçi olan Galois'ten 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 $GF(p)$, $\{0, 1, \dots, p-1\}$ 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 $w \times z = 1 \bmod p$ 'dir.

Cünkü, w , p ye göre asaldır. Eğer, Z_p nin elemanlarını w ile çarparsa, 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 $\pm p$ ise asal sayıdır. Asal sayılar, Açık-anahtarlı kripto sistemlerinde büyük rol oynarlar. Asal sayıarda karşımıza çıkan önemli problemler, asal bir sayının oluşturulması ve bir sayının asal olup olmadığını 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.

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

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

Tanım: $a^{s-1} \equiv 1 \pmod{s}$ 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,

$$a^p \equiv a \pmod{p} \quad \text{denkliği;}$$

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

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

$$\begin{aligned} ax \times 2ax \times \dots \times ((p-1)a) &= [(a \pmod{p}) \times (2a \pmod{p}) \times \dots \times ((p-1)a \pmod{p})] \pmod{p} \\ &= [1 \times 2 \times \dots \times (p-1)] \pmod{p} \\ &= (p-1)! \pmod{p} \end{aligned}$$

Fakat, $a \times 2a \times \dots \times ((p-1)a) = (p-1)!a^{p-1}$ dir

Bu yüzden, $(p-1)!a^{p-1} = (p-1)! \pmod{p}$ dir. Burada $(p-1)!$ 'i atabiliyoruz. Sonuçta: $a^{p-1} = 1 \pmod{p}$ olduğu gösterilmiş olur.

Örn: $a=7$, $p = 19$ verilsin.

$$7^2 = 49 = 11 \pmod{19}$$

$$7^4 = 121 = 7 \pmod{19}$$

$$7^8 = 49 = 11 \pmod{19}$$

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

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

alternatif olarak $a^p \equiv a \pmod{p}$ 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.

$n=pq$ 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 $[0, 1, \dots, (pq-1)]$. Olduğunu düşünelim. Kalanlar kümesindeki $\{p, 2p, \dots, (q-1)p\}$, $\{q, 2q, \dots, (p-1)q\}$ ve $0, n$ 'e göre asal değildirler. Buna uygun olarak,

$$\begin{aligned}\phi(n) &= pq - [(q-1) + (p-1) + 1] \\ &= pq - (p+q) + 1 \\ &= (p-1) \times (q-1) \\ &= \phi(p) \cdot \phi(q)\end{aligned}$$

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

$$\begin{array}{ccc} F: & X & \longrightarrow Y \\ f: & x & \longrightarrow f(x)=y \end{array}$$

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 ayrik 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 sayılarla 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 büyük 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 iyİ 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 sayısı 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 sağlaması. 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 ne kadar ayrık logareitma problemi herhangi bir grup üzerinde isede kriptografik amaçla genellikle \mathbb{Z}_n grubu kullanılır.

Bir başka ifade ile ayrık logaritma :

- Üstelleştirmede ters problem, bir modulo p sayısının ayrık logaritmasının bulunmasıdır.
 - $\alpha^x = b \pmod{p}$ '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 $b \neq 0$ için her zaman bir ayrık logaritması olan bir α olduğu gösterilebilir.
 - α 'nın ardışıl kuvvetleri mod p ile **grup** oluşturur
 $\alpha \pmod{p}, \alpha^2 \pmod{p}, \dots, \alpha^{p-1} \pmod{p}$ 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;

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

Üs x ayrık logaritma veya indis olarak gösterilir.

3.6 En Büyük ortak Bölüm(Greatest Common Divisor)

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

$$OBEB(a,n) = 1 \Leftrightarrow \exists b \in [a,n-1], 1 = a \cdot b \pmod{n}, \text{ yani } b = a^{-1} \text{ dir.}$$

- 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 $g0=n$

$g1=a$

$gi+1 = gi-1 \pmod{gi}$

when $gi=0$ then $(a,n) = gi-1$

örn. $(56,98)$ 'i bulalım.

$g0=98$

$g1 = 56$
 $g2 = 98 \text{ mod } 56 = 42$
 $g3 = 56 \text{ mod } 42 = 14$
 $g4 = 42 \text{ mod } 14 = 0$
 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.

$$x \equiv 13 \pmod{27}$$

$$x \equiv 7 \pmod{16}$$

Bu uygulamada iki modülo birbire 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 ;

$$x \equiv a \pmod{m}$$

$$x \equiv b \pmod{n}$$

x için modulo mn şeklinde tekbir çözümü vardır. Örnek problemde $\text{mod } 16 \cdot 27 = 432$ tek bir çözümü olacaktır.

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

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

İlk benzerliği sağlayan sekans $2, 5, 8, 11, 14, 17, \dots$ dir. Bu sekans tarandığında 5^2 2 bölündüğü zaman 3 kalan terim 8 olduğu için cevap 8'dir. Bunun daha kolay bulılması için Öklid'in enbüyük ortak bölen algoritmasından faydalanjılır.

Bütün işlemi genelleştirirsek ;

$$x \equiv a \pmod{m}$$

$$x \equiv b \pmod{n}$$

Önce $mu+nv=1$ denklemini sağlayan, u ve v tamsayıları bulunmalıdır. Sonra bütün çözümler $x = (mu)b + (nv)a \pmod{mn}$ ni sağlamalıdır.

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

Önce; $100u + 49v = 1$ çözülmelidir.

Euclid's algoritması aşağıdaki şekilde kullanılır.

Bölünen	=	Bölüm	.	Bölen	+	Kalan	0	1
							1	0

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

Buradan $49 \cdot 49 - 24 \cdot 100 = 1$ dir. Çözüm $49 \cdot 49 \cdot 23 - 24 \cdot 200 \cdot 31 = -19177 \equiv 423 \pmod{4900}$ dir.

Genel hali;

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\dots \\ x &\equiv a_r \pmod{m_r} \quad \text{ve } \text{OBEB}(m_i, m_j) = 1, i \neq j \end{aligned}$$

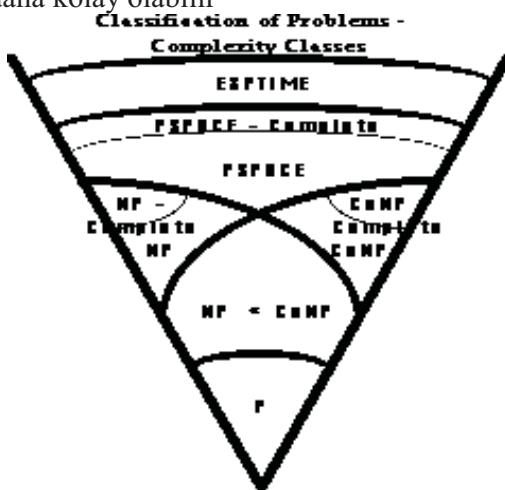
benzerlik sistemleri için, x 'in en az bir çözümü vardır:

$$\begin{aligned} x &= \sum a_i \cdot M_i \cdot N_i \\ M &= m_1 \cdot m_2 \cdot m_3 \dots m_r \quad \text{ve } M_i = M / m_i, \quad N_i = M_i^{-1} \pmod{m_i}. \end{aligned}$$

En önemli uygulama RSA algoritmasındaki çok büyük olan p ve q asal sayılarının çarpımında çok zaman alan işlemleri azaltmak için kullanılır. Hesaplamlar Z_n 'den $Z_p \times Z_q$ '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 denir.
 - Ortalamada daha kolay olabilir



Some Unknowns in Complexity Theory :

- $NP = P$
- $NP = \text{coNP}$
- $P = \text{coNP} \leftarrow NP$

3.8.1 Karmaşıklık Teorisi- Bazı Terminoloji

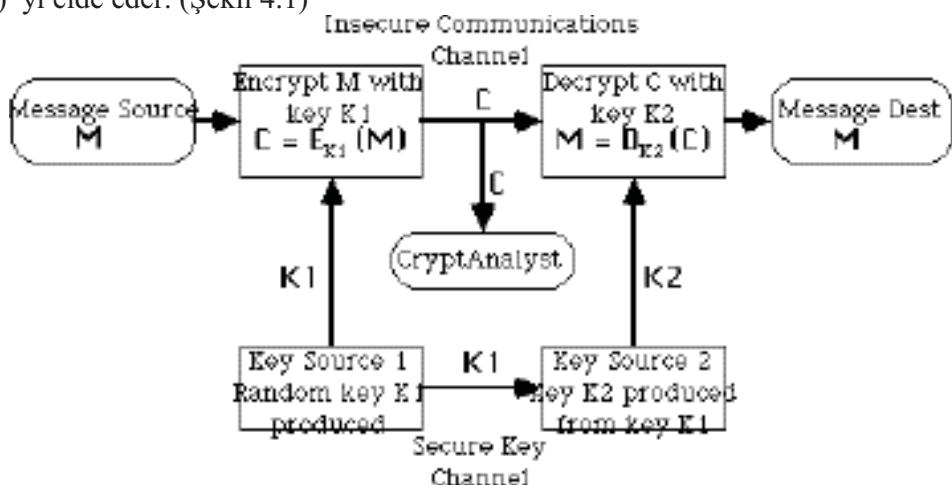
- Bir problemin anlık durumu genel bir problemin kısmi örneğidir



- 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.
 - $f(n) \leq c|g(n)|$, bütün, $n \geq 0$, bazı c için
- **(P)** polinomsal zaman algoritması $O(p(n))$ zaman karmaşıkçı kısmı 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 eşleniğiidir, 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 kullanılmış suregelen 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)



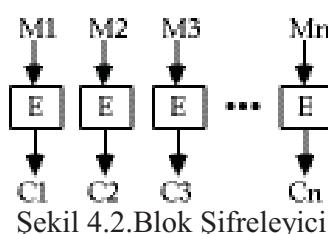
Symmetric (Private-Key) Encryption System

Şekil 4.1 Gizli-anahtarlı kriptosistem ile haberleşme

Gizli-anahtarlı kripto sistemleri uygulama sahalarında ikiye ayrılır;



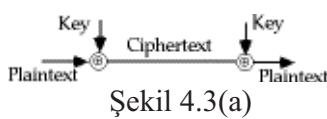
i. Blok Şifreleme: Şifreleme ve deşifreleme işleminde metinler sabit uzunluklu dizilere bölünüp blok blok işleme tabi tutulur (örneğin 8, 16, 32 bit veya bayt). Anahtar uzunluğu ise yine sabittir. Blok şifrelemeye örnek olarak IBM tarafından 1976 yılında tasarlanan ve A.B.D Teknoloji Standartları Enstitüsü 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 ayıklanması durumunda anahtar boyunun 56 bite indiğini hatırlatmak gerekmektedir. Diğer bilinen blok şifrelemeli algoritmalar ise FEAL, IDEA ve RC5 örnek olarak gösterilebilir. Çalışacağımız çoğu modern şifreleyici bu formdadır. (Şek. 4.2)



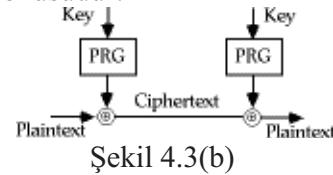
Şekil 4.2.Blok Şifreleyici

ii. Dizi Şifreleme: Bu çeşit şifrelemede algoritmanın girdisi yalnızca anahtardır. Algoritma anahtardan rastgele bir diziye çok benzeyen karanlık anahtar dizisi üretir. Daha sonra karanlık 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. Pratikte zordur.(örn. Pratikte mag tespit veya CDROM da dağıtılmış)
- anahtar tamamen rastgele olduğu için koşulsuz güvenlik sağlanır.
- böyle büyük bir anahtar dağıtımları 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 Horst 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 ayırp 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 Shannon tarafından önerilen confusion ve diffusion fonksiyonlarını şifreleme algoritmasında önerdi.

Diffusion da, şifresiz metnin istatistiksel yapısı, şifreli metnin istatistiğine dağılırlar. Bu, şifresiz metnin her bir díjítinin, şifreli metnin etkilediği díjítlerinin bulunmasıyla sağlanır., başka bir ifade ile, her bir şifreli metin díjít'i birçok şifresiz metin díjít'i tarafından etkilenir. Örnek olarak; Bir $M = m_1, m_2, m_3, \dots$ karakterlerinden oluşan bir şifresiz metni ortalama işlemi ile k ardışılık karakteri ekleyerek şifrelemek;

$y_n = \sum_{i=1}^k m_{n+i} \pmod{26}$ 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 saldırılmasına karşı, şifreli metnin istatistiği ile şifreleme anahtarlarının olabildiğince karmaşık olmasını araştırır. Böylece bir saldırgan şifreli metnin istatistiğini hesapla 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 sonra 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 kriptoanalizi zorlaştırır.

Tur Fonksiyonu : Fazla karmaşık olan tur fonksiyonu kriptanalizi zorlaştırır.

Feistel şifreleyici için diğer özellikler ,

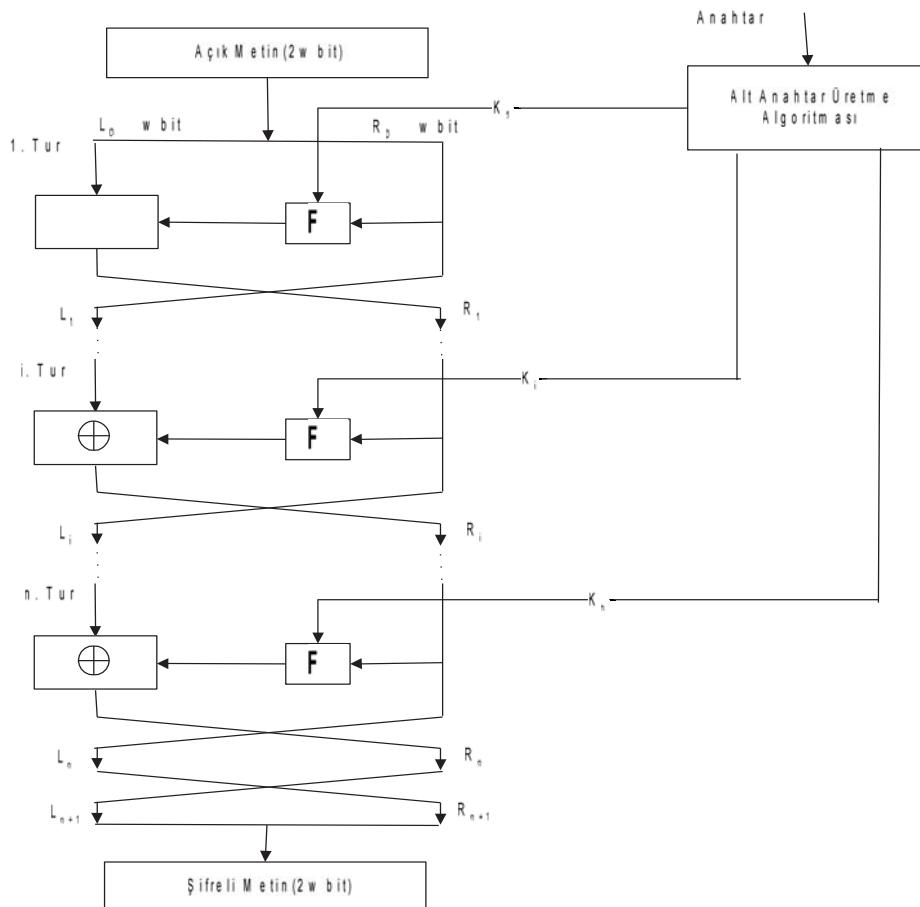
Hızlı yazılım şifreleme/deşifreleme: Çoğu uygulamada, şifreleme uygulamaları veya donanım gerçeklemesi şeklinde kullanım fonksiyonlarının içine koymak. Dolayısı ile algoritmanın icra hızının düşünülmeli gerekir.

Analiz Kolaylığı : Her ne kadar algoritmanın olası kriptanaliz saldırılara 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 gerekmekz.

Algoritmanın genel matematiksel hesaplanması; LE_i : Sol şifrelenmiş blok, RE_i : Sağ şifrelenmiş blok, olmak üzere,

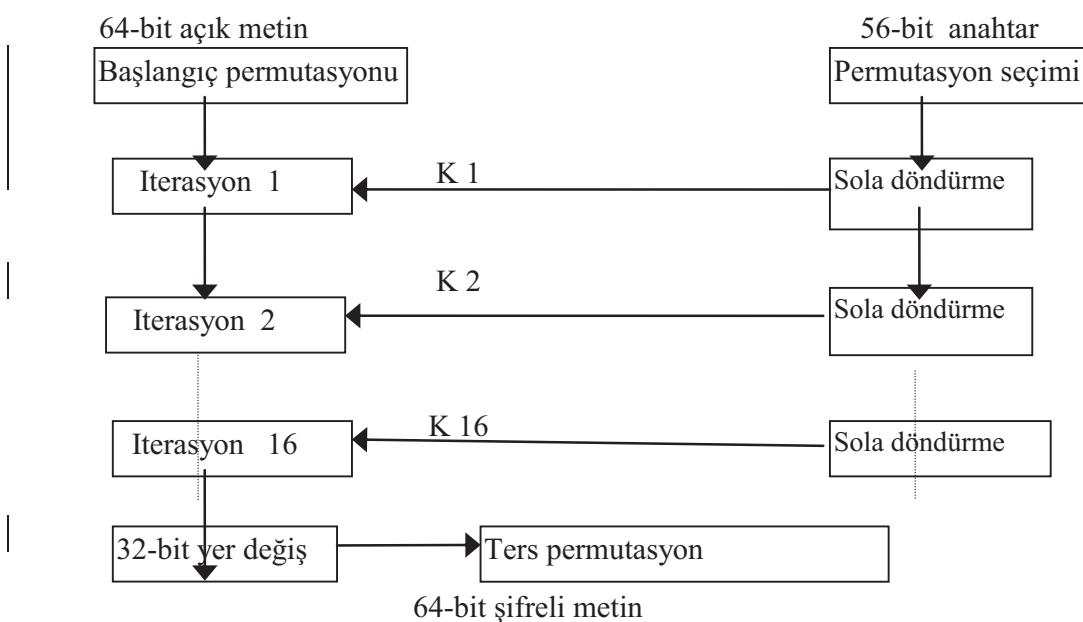
$$\begin{aligned} LE_i &= RE_{i-1} \\ RE_i &= LE_{i-1} \oplus F(RE_{i-1}, K_i) \end{aligned}$$



Şekil 4.4. Klasik Feistel Network

4.2 DES

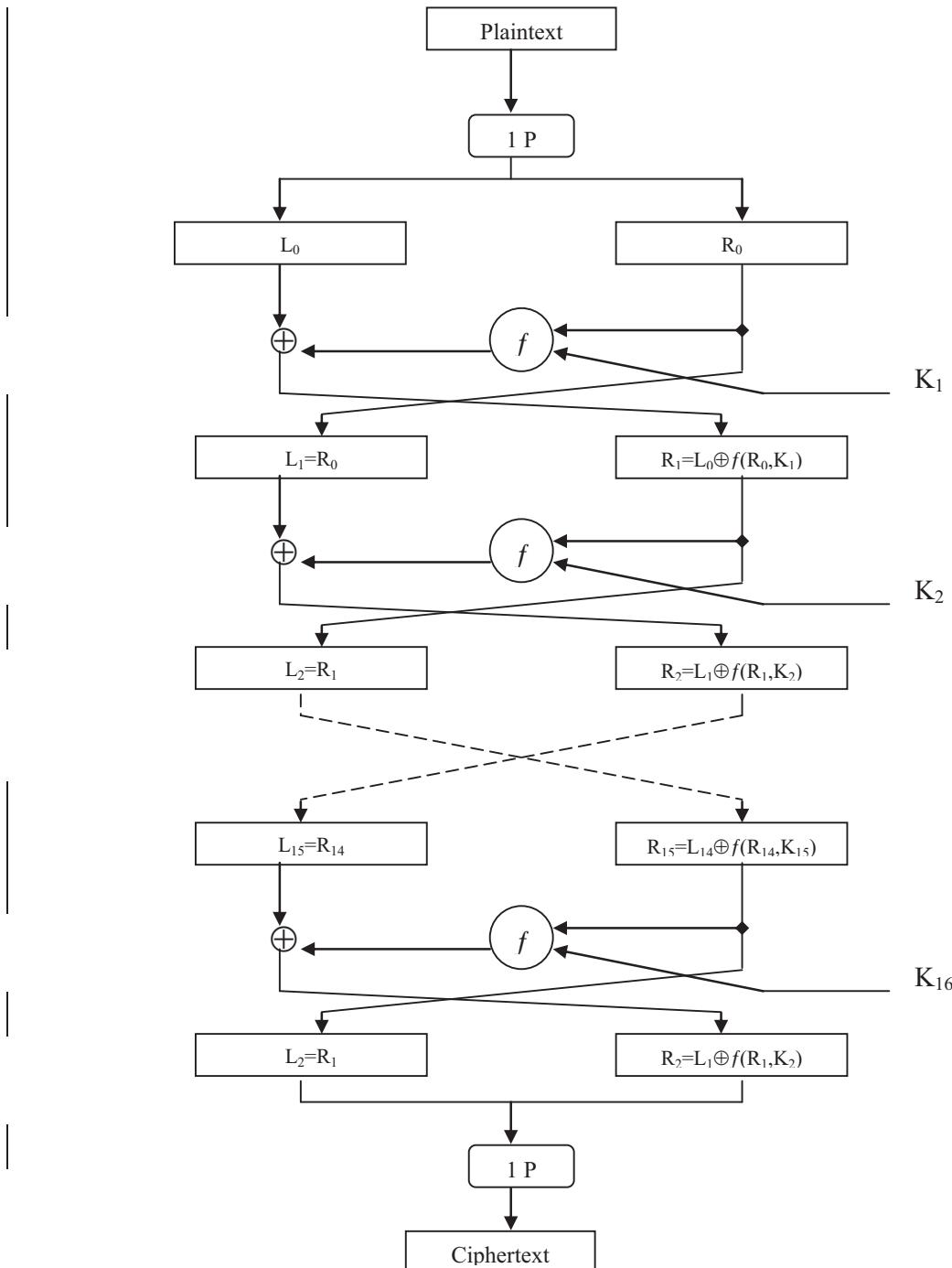
Data Encryption Standard (DES) 1974 yılında IBM tarafından geliştirilmiş ve 1977 yılında yasal olarak atanmıştır. Basit blok şema Şekil 4.5'de gösterilmiştir. Temeli Feistel networküne dayanır.



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

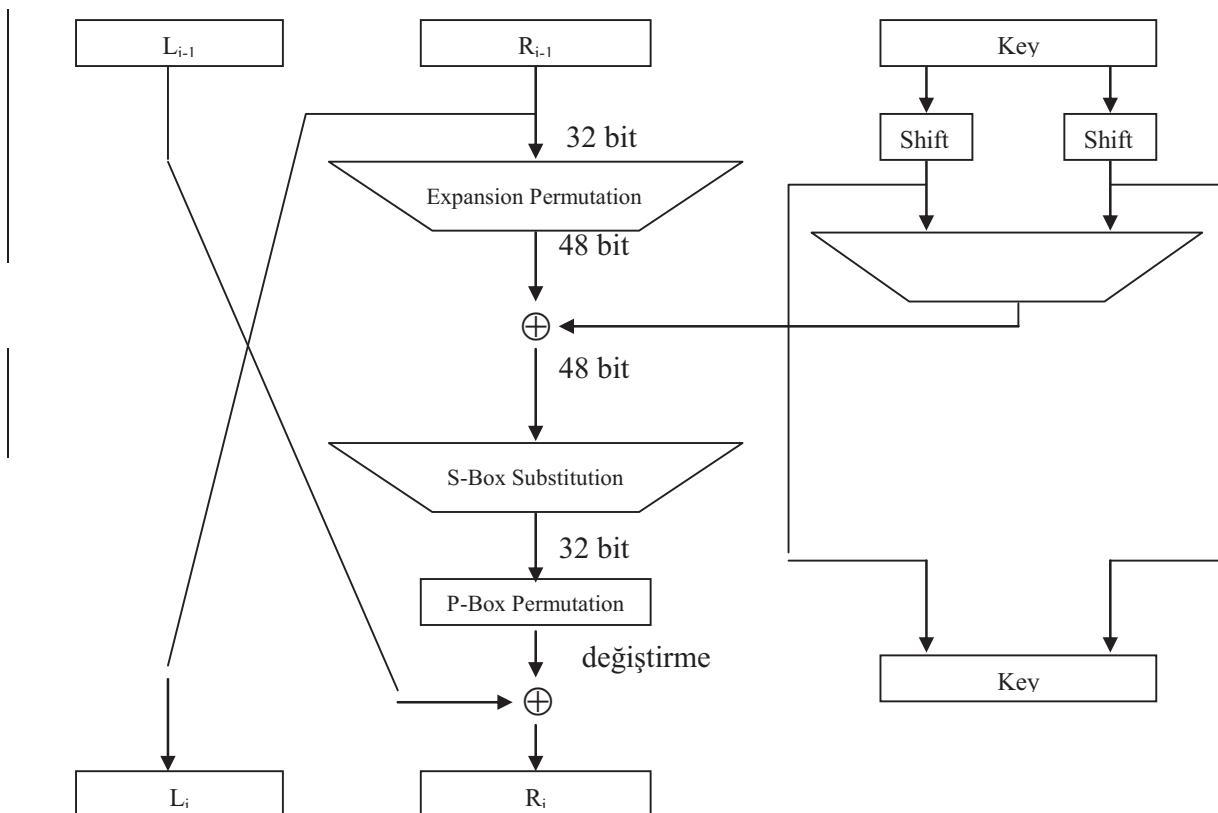
4.2.1 Algoritmanın Özeti :

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 yarımla olmakta ve sol yarımla ise sağ yarımlının eski hali olmaktadır. (4.1) de gösterilen bu işlem 16 kez tekrar eder.

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

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



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 tanımladığı 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
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8

57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
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 ihmali edildiği için 56 bite düşürülür. Bu tablo 6.8' de tanımlanmıştır. İhmali 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-anahtarlar (K_i) şu şekilde belirlenir.

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
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	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
23	19	12	4	26	8
41	52	31	37	47	55
44	49	39	56	34	53

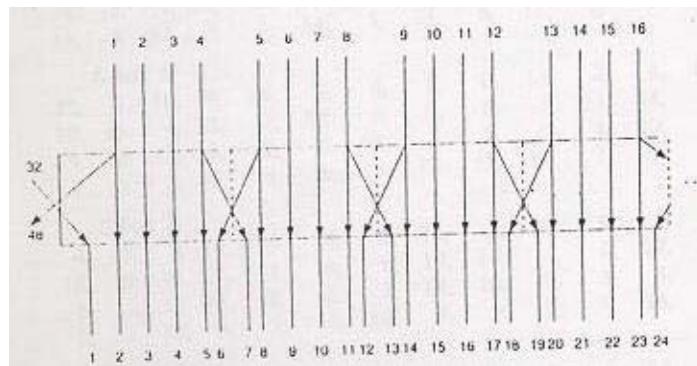
3	28	15	6	21	10
16	7	27	20	13	2
30	40	51	45	33	48
46	42	50	36	29	32

Tablo 4.4 Sıkıştırma Permutasyonu

4.2.4 Genişleme permütasyonu :

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 yarımi anahtar ile aynı uzunlukta yapmak ve yerine koyma (substitution) işlemi sırasında sıkıştırılabilen daha uzun sonuç sağlamak.

Şekil 4.8' de genişleme permütasyonu 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

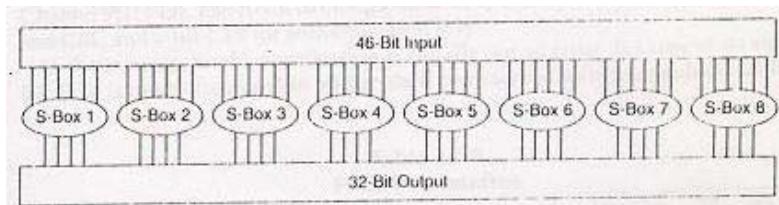
Tablo 4.5’de çıktı pozisyonlarının hangi girdi pozisyonlarına göre nasıl yerleştirildiği görülmektedir. Örneğin; girdi bloğunun 3. pozisyonu çıktı bloğunun 4. pozisyonuna karşılık gelmektedir ve girdi bloğunun 21. pozisyonu çıktı bloğunun 32. pozisyonuna karşılık gelmektedir.

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

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

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 işlenilir. 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’da 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 çağrırlır. 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 şifrelemeye 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 metindekı küçük değişikliklerin şifreli metrin üzerinde büyük değişiklige neden olmasına çığ(avalance) etkisi denir.

4.3 DES' in Güvenliği :

Anahtar Uzunluğu :

Bilindiği gibi DES'in anahtar uzunluğu 56 bittir. Bu ise brute-force atakları için $2^{56} = 7.2 \times 10^{16}$ anahtar sayısı demektir. Tablo 6.2 gözönüne alırsa, mikrosaniye başına bir çözümleme yapan bir makinenin 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 kırılmıştır. Bu nedenle anahtar sayısının ortalama yarısı kadar deneme yapılabileceği varsayımlı ile DES'in brute-force saldırılara karşı zayıf olduğu söylenebilir. DES'in alternatifleri olab 3DES ve AES geliştirilmiştir.

DES zamanlama saldırılarına karşı oldukça güçlündür.

4.4 Diferansiyel ve Doğrusal(Lineer) Kriptoanaliz.

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

Diferansiyel 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 saldırısı 2^{47} adet seçilen şifresiz metin, veya 2^{55} bilinen şifreli metin ve 2^{47} DES işlemi gerektir.

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

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

Diff. Kriptanaliz

$$\Delta m = m \oplus m' \quad (\text{Mesaj yarları})$$

$$\Delta m_i = m_i \oplus m'_i$$

$$\begin{aligned} \Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= m_{i-1} \oplus f(m_i, K_i) \oplus m'_{i-1} \oplus f(m'_i, K'_i) \\ &= \Delta m_{i-1} \oplus [f(m_i, K_i) \oplus f(m'_i, K'_i)] \end{aligned}$$

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 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 olasılığı 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.

$$P[1], P[2], \dots, P[n], \text{ ve } C[1], C[2], \dots, C[n]$$

$K[1], K[2], \dots, K[m]$ olsun ve;

$$A[i,j,\dots,k] = A[i] \oplus A[j] \oplus \dots \oplus A[k] \text{ tanımlansın. (bitler bir biri ile XOR'lanır)}$$

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

$$P(\alpha_1, \alpha_2, \dots, \alpha_a) \oplus C(\beta_1, \beta_2, \dots, \beta_b) = K(\gamma_1, \gamma_2, \dots, \gamma_c)$$

Burada $x=0,1$; $1 \leq a, b \leq n$, $1 \leq c \leq m$ 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 $K(\gamma_1, \gamma_2, \dots, \gamma_c) = 0$ dır. Eğer çoğunda 1 ise $K(\gamma_1, \gamma_2, \dots, \gamma_c) = 1$. 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 yarımda 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	OE0E	OE0E	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.

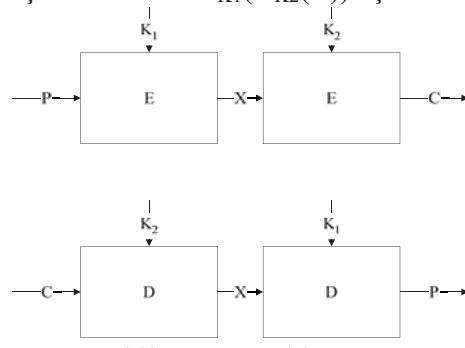
DES 16' dan daha az turda gerçekleştiği zaman, brute force saldıruları olarak bilinen saldırularla daha kolay ve verimli bir şekilde kırılabilir.

4.6 DES'in Farklı Şekilleri :

4.6.1 Double DES :

DES'in iki ayrı anahtar ile arad arda şifrelemede kullanılmıştır. Bu durumda anahtar uzunluğu 112 bit olacaktır. Brute-Force saldırularına 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) saldırularına zayıflığı vardır.

Ortada Karşılaşma(Meet in the middle attack) saldırısı

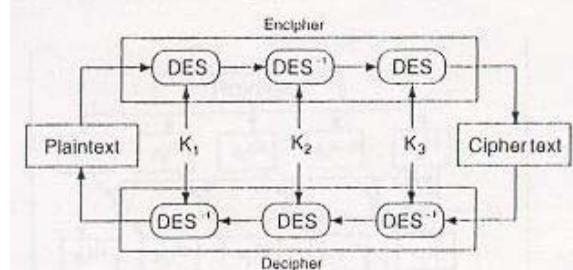
Şekil 6.16.'dan görüldüğü gibi X değerinin hesabı aşağıdaki şekilde yapılabilir.

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

Verilen bir (P, C) çifti ile P, K_1 'in bütün anahtar kombinasyonları(2^{56}) ile şifrelenerek X 'n değerine göre sıralanır. C yine K_2 'nin bütün anahtar kombinasyonları(2^{56}) ile deşifrelenerek X 'n değerine göre sıralanır. Herikisinde 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.

$$C = E_{K_3}(E_{K_2}(E_{K_1}(P))) ; \quad P = D_{K_1}(D_{K_2}(D_{K_3}(C)))$$



Şekil 4.11 Triple DES

4.6.2 CRYPT(3) :

UNIX sistemler üzerinde bulunan DES tabanlı algoritmadır. Aslında passwordlar için bir yolu 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 artırılmıştır.

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. Criptanalizi 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 çalışıkları 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 gereklidir. 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 biri 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 bittten 128 bit'e kadar değişen anahtar uzunlukları kullanır. CAST, DES'te kullanılanlardan daha uzun olan sabit S-boxlar kullanır. Bu S-boxların tasarımları 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 modları 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 işlemeye 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:

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

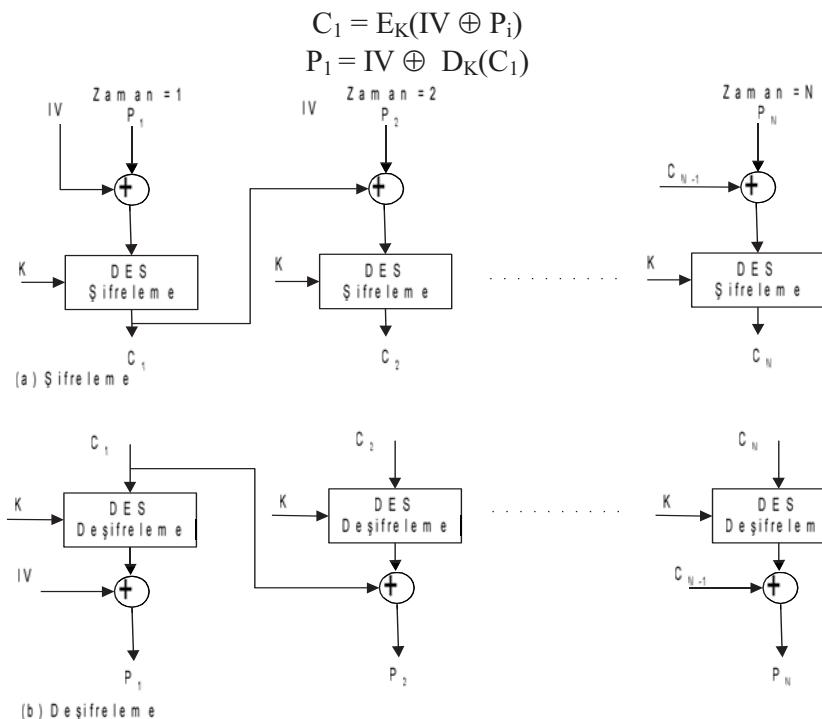
Burada $E_K[X]$, X'in K anahtarı kullanılarak şifrelenmiş şekli ve \oplus ise XOR işlemidir. Sonra,

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

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

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

Şekil 4.12 'de görüldüğü gibi, ilk şifreli bloğu elde etmek için başlatma vektörü(IV) ilk açık metin bloğu ile XOR işlemeye 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 gereklidir. İlk bloğun şifrelenmesi aşağıdaki ifadede gösterilmiştir.



Şekil 4.12. CBC (Cipher Block Chaining Mode)

4.7.2 CFB(Cipher Feedback Mode)

DES tasarımlı 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 varsayımlı 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 çıkıştı 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 ne kadar 168 bitlik anahtar kullanıyor ve brute-force saldırılara 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 verilmiştir.

- 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)

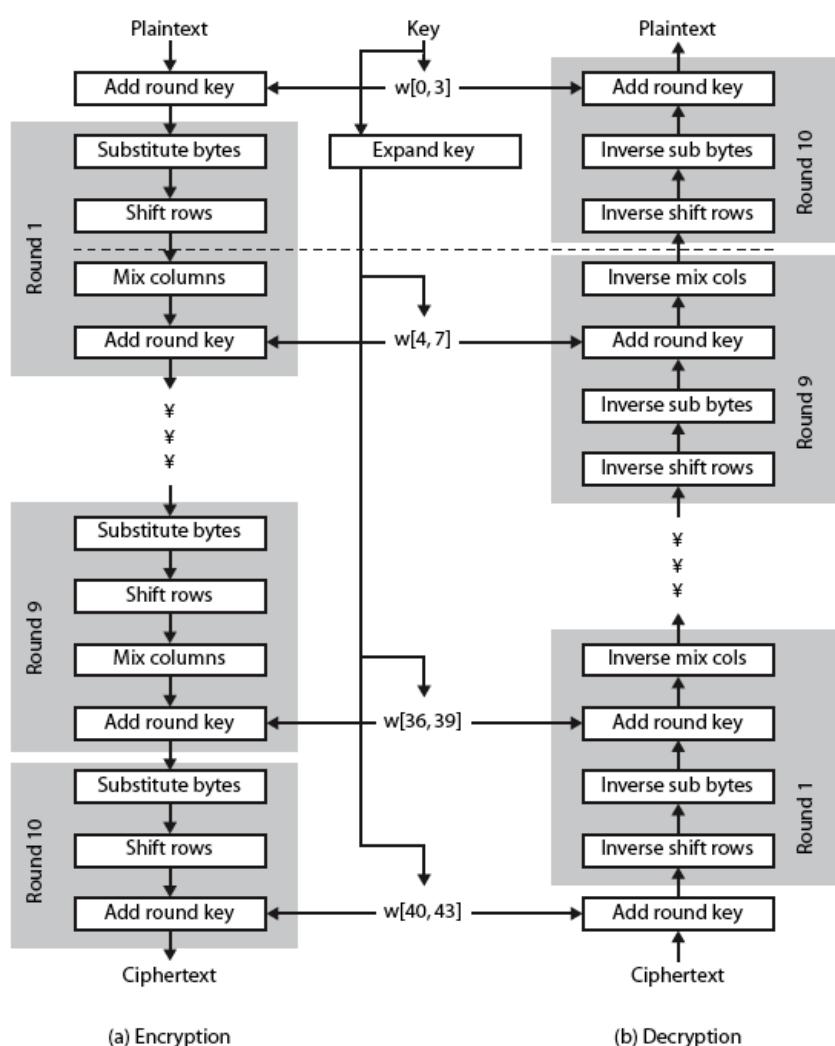
- 3 Yapı çok basit: Şifreleme ve deşifreleme için şifreleyici add round key evresi ile başlar. Her biri 4 evre olan 9 tur ile devam eder.

- 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 deşfreleme algoritması anahtarı ters yönde genişletir. Bununla birlikte deşfreleme algoritması, şifrelemeye benzemez. Bu AES'in parçalı yapısının sonucudur.
- 8 Dört evre ters çevrilebilir şekilde kurulduğunda deşfrelemenin plaintext'i bulması sağlanır.
- 9 Son turda , şifreleme ve deşfrelemenin 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 Deşfreleme evreleri:
- Inverse-Shift-Rows:
 - Inverse Sub bytes:
 - Inverse Mix columns:



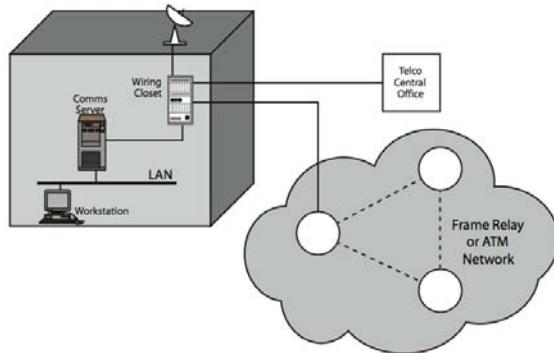
Şekil 4.13 : AES Şifreleme ve Deşfreleme adımları

4.9 Gizli anahtarlı (simetrik) kriptosistemlerin Güvenliği :

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

- a. Link Şifreleme : Şifreleme her bir iletişim bağlantısı üzerinde bağımsız olarak yapılır. Bağlantılar arasındaki trafiğin deşifrelenmesi gereklidir. Birçok cihaz ve birçift anahtar gereklidir.
- b. Uçtan uca şifreleme : Şifreleme orijinal kaynak ve son varış noktası arasında yapılır. Her iki ucsta paylaşılmış anahtarlar ve cihazlar gereklidir.

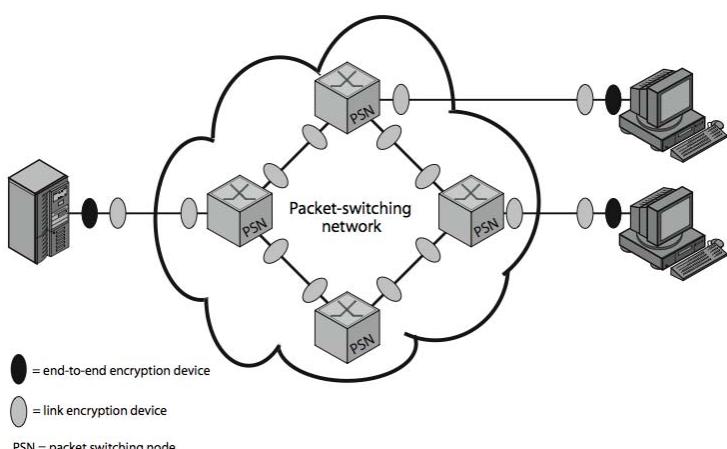
Şekil 4.14'de gösterilen haberleşme ağı'nda açıklık noktaları belirtilmiştir. YAŞ'ne bağlı olan bir iş istasyonunun gönderdiği mesajlar YAŞ'ın özelliği itibarı ile dinlenmeye müsaittir. Haberleşme sunucusuna erişim hakkı elde eden bir saldırgan ağ trafigini dinleyip analiz edebilir. YAŞ 'nin dışında bir yönlendirici veya çevirmeli modem ile dış ağa bağlantı olabilecektir. Bunların bağlantı noktaları zayıf noktalardır. Dış ağdaki herhangi bir haberleşme bağlantısı saldırıyla açık yerlerdir. Böylece saldırıyla 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 bağlantıyı ayrı ayrı şifrelemek ve uçtan uca haberleşmeyi şifrelemek Şekil 4.15'de bir paket anahtarlamalı ağı'da bağlantıların ve uçtan uca haberleşmenin şifrelenmesi gösterilmiştir



Şekil 4.15. Paket anahtarlamalı ağı'da şifreleme

Uçtan uca haberleşme kullanıldığı zaman başlık şifresiz olarak bırakılmalıdır. Böylece ağı yönlendirme bilgisini doğru olarak sağlayabilir.

Bu nedenle her ne kadar, içerik şifrelensede, trafik izi akışını anlamak mümkündür. Idealde heriki şifrelemede

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

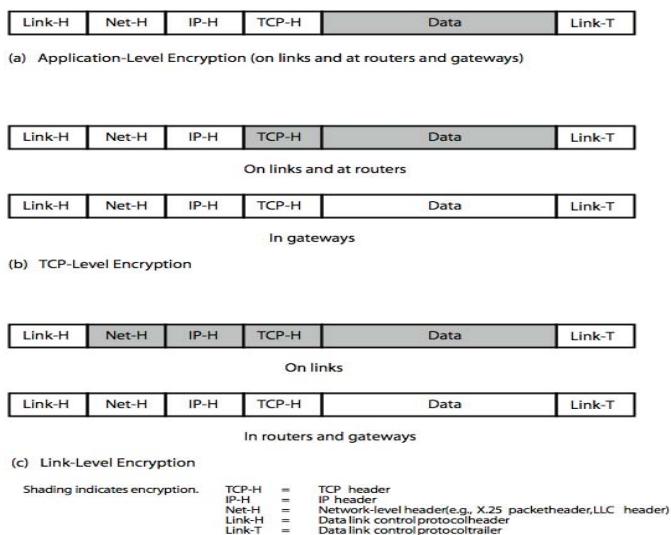
Bağlantı şifreleme ise trafik akışının gözlenmesini engeller. OSI referans modelinin değişik katmanlarında şifreleme fonksiyonu sağlanabilir

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 anahtarların güvenli olarak dağıtılmasıdır. Güvenli bir sistem sık sık anahtar dağıtım yönteminin kırılmasıyla etkisiz hale gelebilir

Verilen A ve B grupları için değişik anahtar dağıtım alternatifleri olabilir

A anahtarını secer ve fiziksel olarak B'ye iletir.

Üçüncü şahıs anahtarını secer, A ve B'ye dağıtır

Eğer A ve B önceden haberleşiyorsa, önceki anahtarını kullanarak yeni anahtarını şifreler

Eğer A ve B, C ile birlikte güvenli bir iletişim kanalına sahipse, C anahtarını 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.

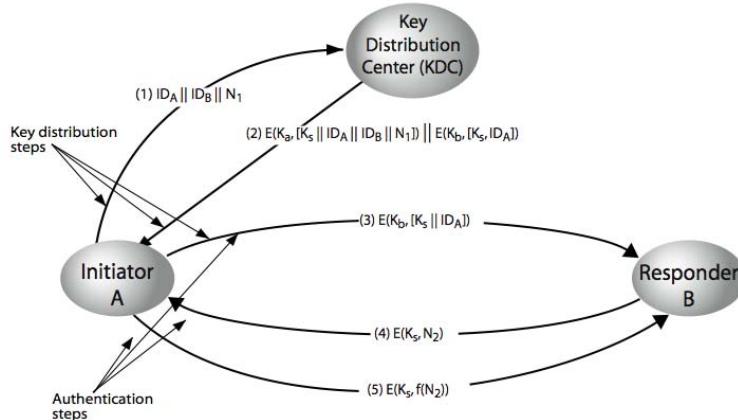
Ana Anahtar

Anahtar dağıtım merkezi ile kullanıcılar arasında N adet tır.

Ana anahtar;

Oturum anahtarlarını şifrelemek için kullanılır

Kullanıcı ile anahtar dağıtım merkezi arasında paylaşılır



Şekil 4.17: Anahtar dağıtımlı senaryosu

Merkezi olmayan anahtar dağıtımlı

Merkezi olmayan anahtar dağıtımda, her bir üç sistem, oturum anahtarını dağıtmak için güvenli bir şekilde haberleşmesi gereklidir. Böylece N adet üç sistemin konfigürasyonu için $N(N-1)/2$ adet anahtar gerekebilir.

Oturum anahtarları aşağıdaki adımlar ile sağlanır

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

Böylece her bir düğüm en çok ($N-1$) ana anahtar saklamak zorunda kalır veya gerekiğinde üretilip kullanılır

1)

a) Simetrik Sifreleme (symmetric encryption)

Aynı algoritma, aynı anahtar hem sifreleme hem de desifreleme için kullanılabilir. Simetrik sifrelemedir.

Göndərci ve alıcı aynı anahtarı paylaşır.
private key / single key

Simetrik sifrelemede bu anahtar iki taraf tarafından daha önceden bilinir.

Avantajları

Sifreleme ve şifreyi çözme hızıdır. Anahtarlar kısadır.

Tarafalar arasındaki iletişim gizliliği sağlanır.

Desavantajları

Anahtar saklamak zordur. (Key Storage Problem)

Güvenilir anahtar dağıtmı zordur. (Key Distribution Problem)

Kimlik doğrulama sağlanamaz. Aynı anahtara sahip olan herhangi biri veriyi şifreleyebilir.

Asimetrik Sifreleme (asymmetric encryption)

Sifrelemede ve şifre çözmede farklı anahtarlar kullanılır. Bu anahtarlar public key ve private key olarak geçmektedir. Sifreleme ve doğrulamada public key kullanılır. Şifre çözme için de private key kullanılır. Ayrıca gönderen taraf private key ile veriyi imzalar. Anahtarlar daha uzundur. Daha yavaştır.

Özelliklerin Karaktertilmeleri

	Simetrik Sifreleme	Aritmetik Sifreleme
Gizlilik	saglar	saglar
Bütünlük	-	saglar
Kimlik Dogrulama	-	saglar
İnkar edilememelik	-	saglar
Performans	hızlı	yavaş
Güvenlik	anatlar bağlı	uzunluğuna → 4.

mutlak Güvenlik

Bilgisayar gücü ne kadar fazla olursa olsun

Sifre hiçbir eklede kırılabilir.

Hesaplamağa bağlı Güvenlik

Bir sifreleme algoritması aşağıdaki kriterlerini sağlıyor ise hesaplamağa bağlı güvenlidir.

- Sifrenin kırılmasını maalesef sifrelenmiş bilginin değerinden fazla ve

- Sifreyi kırmak için gereken zaman, bilginin garanti ömründen fazla ise sifre kırılma riski düşer.

b) Euler Totient
p and irreducible
Fonksiyonu ($\Phi(n)$)
 $\Phi(p) = p-1$ dir.

\forall prime = aral

$n = p \cdot q$ p, q aral sayılar irreducible

$$\Phi(n) = \Phi(p \cdot q) = \Phi(p) \cdot \Phi(q) = (p-1)(q-1)$$

$$\Phi(p) = (p-1)$$

$$\Phi(pq) = (p-1)(q-1)$$

2)

RSA

En çok bilinen ve en pratik açık anahtarlı taranndır.

Güvenliği, büyük sayıların çarpanlarının hesaplanmasının zorluğuna bağlıdır.

RSA modüler aritmetiği kullanır.

Açık anahtarlı şifreleme algoritmasıdır.

Yöntemin uygulanması için önce anahtarların üretilmesi gereklidir.

Algoritma şu şekildedir:

- $n = p \cdot q$ hesaplanır. $\rightarrow p$ ve q sonda yerlidir.
- $\phi(n) = (p-1)(q-1)$ \rightarrow Euler Totient fonksiyonu
- Rastgele şifreleme anahtarı seçilir.
 $1 < e < \phi(n)$ \rightarrow bu anahtar (e) $\phi(n)$ ile aralarında aralı olmalıdır.
- Deşifreleme anahtarı (d) hesaplanır.
 $d, e \equiv 1 \pmod{\phi(n)}$ \rightarrow d yi deneyerek bul
 $d \leq n$
- Açık anahtar (public key)
 $PU = \{ e, n \}$
- Gizli anahtar (private key)
 $PR = \{ d, n \}$

- encrypt a message M \rightarrow m mesajını şifreleme
public key $PU = \{e, n\}$

$$C = M^e \text{ mod } n$$

- decrypt the ciphertext C

private key $PR = \{d, n\}$

$$M = C^d \text{ mod } n$$

Örnek:

Sonuda $p=3$, $q=11$, $e=7$ ve $M=5$ için ~~iki asal sayı~~ şekilde parçalanır.

acıklayınız denmiş.

① $n = p \cdot q = 3 \cdot 11 = 33$ n = 33

② $\Phi(n) = (p-1)(q-1) = 2 \cdot 10 = 20$ Φ(n) = 20

③ e yi bul. $\rightarrow e$ sonda verilmiş, e = 7

④ d yi bul.

$$d, e \equiv 1 \pmod{\Phi(n)} \rightarrow 1'i sol tarafta at.$$

$$d \cdot 7 - 1 \equiv 0 \pmod{20} \quad \boxed{d = 3} \text{ olabilir.}$$

⑤ public key $PU = \{e, n\}$

$$PU = \{7, 33\}$$

⑥ private key $PR = \{d, n\}$

$$PR = \{3, 33\}$$

(7)

m sonda verilmis, $M=5$

$$C = M^e \text{ mod } n \quad \text{public key} = \{7, 33\}$$

$$C = 5^7 \text{ mod } 33$$

$$5^7 = 78125$$

$$78125 \mod 33 = 2$$

2 mesajının şifrelenmiş hali

$$2^7 = 128$$

mesajın şifrelenmesi

$$128 \mod 33 = 22$$

mesajın şifrelenmesi

$$22^7 = 201326592$$

mesajın şifrelenmesi

$$201326592 \mod 33 = 2$$

mesajın şifrelenmesi

$$2^7 = 128$$

mesajın şifrelenmesi

$$128 \mod 33 = 22$$

mesajın şifrelenmesi

$$22^7 = 201326592$$

mesajın şifrelenmesi

$$201326592 \mod 33 = 2$$

mesajın şifrelenmesi

$$2^7 = 128$$

mesajın şifrelenmesi

$$128 \mod 33 = 22$$

mesajın şifrelenmesi

$$22^7 = 201326592$$

mesajın şifrelenmesi

$$201326592 \mod 33 = 2$$

mesajın şifrelenmesi

4)

a) Simetrik şifrelerin çalışma modları nelerdir?

Elektronik Codebook Book (ECB)

Cipher Block Chaining (CBC)

Cipher Feedback (CFB)

Output Feedback (OFB)

Counter (CTR)

Output feedback ve counter modu karşılaştırınız?

Output feedback ;

Mesaj bir bit akışı olarak değerlendirilir.

metin bitleri mesaja şifre çıkışının eklendi.

Çıktı sonra genel birleme

Gen bilindiğim (feedback) mesajdan bağımsızdır.

(*) bit hataları yayılmaz.

(*) değişikliklere karşı daha ravinmasız

(*) göndençi ve alıcı senkronize kalmalıdır.

(*) Gürültülü kanallarda akış şifrelemesi

Counter ;

OFB ye benzer ancak herhangi bir genel birleme degeni yerine sayısal değerini şifreler.

Her düz metin bloğu için farklı bir anahtar ve sayısal değer olmalıdır.

(*) yüksek hızlı ağ şifrelemesi

(*) avantaj → efficiency (verimlilik)

(*) yüksek hızlı bağlantılar için uygun.

(*) Kamuflanabilir güvenlik

(*) Sıfırı verilen bloklarına rastgele erişim

T Ancak anahtar sayacı değerini tekrar

- kullanılmamalıdır

b)

MAC fonksiyonu ne demekti?

(Mesaj doğrulama kodu)

Mesajın sonuna eklenen bir veri bloğu ile yapılır

Bu teknikte, A ve B olarak adlandırılan iki haberleşme grubu ortak bir K anahtarını paylaşır.

Alici mesaj üzerindeki ayaş hesaplamayı yapar ve sonucu MAC ile karşılaştırır. Böylece göndərciden gelen mesajın değişikliliklerini garanti eder.

Sadece doğrulama gerekliliğinin MAC kullanması

MAC mesajın sonuna eklenir ve varmca yeniden hesaplanarak doğrulama yapılır.

MAC fonksiyonu, farklı uzantıktaki mesajları aynı uzunluğa dönüştürdüğü için sadece bir-bir fonksiyondur.

(*) MAC (mesaj Authentication Code)

mesaj doğrulama kodu

MAC kodu sayesinde mesajı alan, mesajın değişikliğine ikna olur. Mesajı alan mesajın doğru kısımdan geldiğine de ikna olur.

HASH fonksiyonu ne demektr?

Kriptolojik özetleme fonksiyonu

Sayısal imza ve veri bütünlüğünün korunması
alanlarında yaygındır.

Bilginin bütünlüğünü sağlamak için de kullanılır.

(*) Değişik uzunlukta bit dizilerini sabit uzunluklu bit
dizilene taşıyan polinomral zamanda kolay hesaplanabilen
fonksiyonlardır.

Sabit uzunluklu bu dizide özet-değer denir
(Hash-value)

Mesajı alan kişi özetleme fonksiyonunu hesaplar.

Elde ettiği özet-değer ile kendisine gelen özet-değeri
karşılaştırır. Mesajın bütünlüğünün ispatı için bu iki
değeri aynı olması gereklidir.

Collision resistance nedir?

(Faklımaya dayanıklılık)

Aynı özet veren iki mesaj bulmak hesapsal
olarak imkansızdır.

1)

a) Kriptolama nedir?

Kriptolama, açık metnin anahtar yardımıyla şifreli metne dönüştürülmesidir.

Ayrık logaritma problemi nedir?

Üstelleştirmede ters problem, bir modulo p

sayısının ayrık logaritmasının bulunmasıdır.

$a^x \equiv b \pmod{p}$ x bul.

$$3^x \equiv 13 \pmod{17} \quad x = ? \text{ tür.}$$

Ana tek çözüm bu değildir.

$$3^{20} \equiv 13 \pmod{17} \quad \text{denkleğinde doğrudur.}$$

2)

Yerine koyma ve yer değiştirme tabanlı şifreleme nedir?

- yerine koyma (substitution)

- yerini değiştirme (transposition)

Yerine koyma da, şifresiz metindeki her bir eleman diğer elemana dönüştürülür.

Yerini değiştirme de, şifresiz metindeki elemanların yerini değiştirilir.

Yerine koyma frekans analizi ile çözülebilir.

(4)

a) DES yönteminin bir turunda yapılan işlemlerinizi yazınız.

DES bir blok şifrelemeydir, - 64 bit bloklardaki veriyi şifreler. Plain text'in 64 bitlik bloğu bir algoritma sokulur ve 64 bitlik şifrelenmiş bir ifade elde edilir. Şifrelenmede ve şifreyi çözüken aynı algoritma ve anahtar kullanılır. Anahtarın uzunluğu 56 bittir.

Algoritmanın Özeti

DES 64 bit blok plaintext de işlem görür. Plaintext, ilk permutasyondan sonra yarı sağ yarı solda her bin 32 bit uzunlığında 2 parçaya bölünür. Daha sonra f fonksiyonu ve anahtar ile birləştiştirerek sonraki adıma geçilir. Aynı işlem 16 kez tekrarlanır ve 16. turun sonunda, sağ ve sol parçalar birləştilir. Son permutasyondan sonra algoritma tamamlanarak biter.

Her bir turda anahtar bitleri değişir ve anahtarın 56 bitinden 48 biti seqilir. Verinin sağ yarımı genişleme permutasyonu yoluyla 32 bitten 48 bit'e genişletilir. Genişletilen kisim 48 bit anahtarıla XOR işlemine sokulur. Daha sonra 32 yeni bit ureten 8 S-box içine göndertir ve tekrar değişir. f fonksiyonunun eksiği verinin sol yarımı ile XOR lar. Bu işlem 16 kez tekrarlanır.

b)

AES yöntemi için (a da) yapılan işlemlerin yazınız.

AES (Advanced Encryption Standard)

128 bitlik ve 128 / 192 / 256 bitlik anahtar

Uzunlukuna sahiptir.

Iteratif olarak çalışır.

Veriyi 4 bytelik dört sütunlu bloklar halinde işler.

Her bir turda ve bloğunun tamamı üzerinde işlem yapar.

Büyük saldırılara karşı dirençlidir.

Hızlıdır, kod稳定性 sağlar.

Algoritması;

AES mimarisinde her ikisi yan da paralel şekilde işlenir.

Sağlanan gizli anahtarı 40 adet dörtlü 32 bitli wordler şeklinde genişletilir. Dört farklı 128 bitlik kelime her bir turda tur anahtarı olarak kullanılır.

Her bir turdaki 4 farklı evrede, bir permutasyon ve üç yer değiştirme kullanılır.

- Substitute byteları; bloğun byte byte yer değiştirmesi için S-boxlar kullanılır.

- Shift Rows; bir permutasyon (byteları grup ve sütunlar arasında değiştirme)

- Mix columns; GF(2⁸) üzerinde yapılan antisimetrik kullanarak yer değiştirme

- Add-Rand key; bir bit bir XOR işlemi

- Yapı çok basit. Şifreleme ve desifreleme için şifreleyici add round key evesi ile başlar. Her blok 4 evesi olan 9 tur ile devam eder.
- Sadece add-round key evesi anahtar kullanır. Bu nedenle şifreleyici add-round key evesi ile başlar ve biter.
- Diğer üç evede confusion, diffusion ve diagonal olmamayı sağlar. Fakat anahtar kullanmadıkları için güvenlik sağlanır.

5)

b) D-H (Diffie-Hellman) yönteminin çalışma yazınız.

Açık anahtar dağıtım şemasıdır.

Keyfi mesajı değiştirmek için kullanılır.

Çalışması:

- Güvenli bir iletişim kanalı üzerinden bazı anahtarları değiştirmek isteyen iki A & B olsun. Bunlar;
- Büyük bir asal sayı seçecekler.
- a bir mod p primitive elemandır.
- A nin x_A gibi bir gizli sayı varsa ($x_A < p$)
- B nin x_B " " " " ". ($x_B < p$)

★ y_A, y_B sırasıyla hesaplanır.

$$y_A = a^{x_A} \bmod p \quad y_B = a^{x_B} \bmod p$$

★ Sonra anahtar hesaplanır.

- $K_{AB} = a^{x_A \cdot x_B} \bmod p$ (ortak gizli anahtar)
- $y_A^{x_B} \bmod p$ (B hesaplayabilir.)
- $y_B^{x_A} \bmod p$ (A hesaplayabilir.)

6 KRİPTOSİSTEMLER VE SİMETRİK ŞİFRELEME/DEŞİFRELEME(Cryptosystems and Symmetric Encryption/Decryption)

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.

6.1 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 dağıtilır.
- http Sayfa içeriği ve formlardaki bilgilerin içeriği açık metin olarak gönderilir.
- IMAP Doğrulama açık metindir
- SNMPv1 Doğrulama açık metindir

6.2 Ağ Üzerinde Yapılan Saldırı Türleri

1. İfşaatt(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. Gerçegi gizleme (Masquerade) Hileli bir kaynaktan ağ'a mesaj ekleme. Bu işlem muhalif tarafından yetkili bir kullanıcından gelmiş gibi görünen mesajların oluşturulmasını içerir.

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

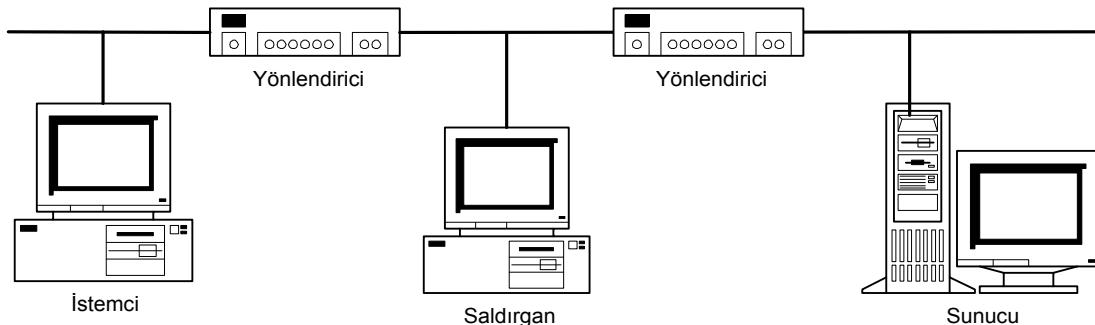
5.Sıra Değiştirme(Sequence Modification): Ekleme silme ve yeniden sıralama ile mesajın sırasında değişiklik yapma.

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

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

6.3 İyi Doğrulama Gereklidir.

İyi doğrulama gerekligi 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 6.1. Oturum Korsanlığı

6.3.1 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 gözlemlayabilir. 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)

6.3.2 Varışın Doğrulanması

Kaynağın iletişiminden 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 önemnen en emin yolu verileri şifreleyerek iletmemektir.

6.4 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.

6.4.1 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

kısımında 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 **kriptanaliz** 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.

6.5 Temel Kavramlar

Kriptografi(cryptography) : Anlaşılır bir mesajı anlaşılmaz şeke 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 şeke 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 6.2)

Kod(code) : Anlaşılır bir mesajı bir kod kitabı kullanarak anlaşılmaz şeke dönüştürme için bir algoritma

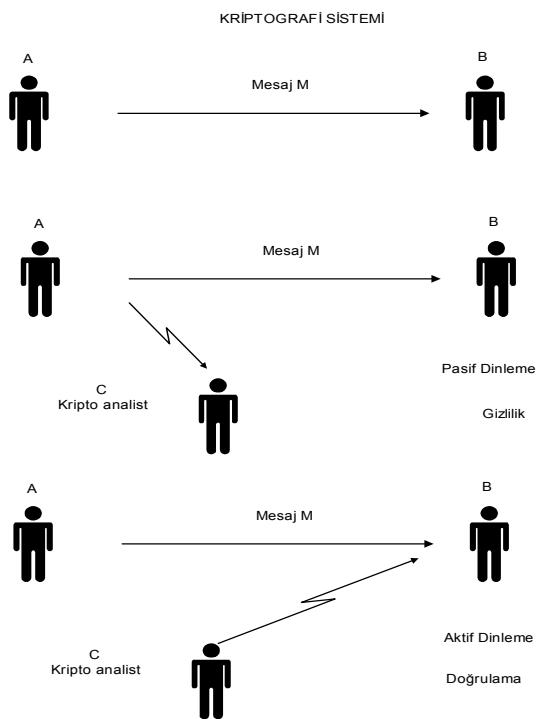
Şifreleme(Encryption) $c = E_K(m)$

Deşifreleme(Decryption) $m = D_K(c)$

E_K , kriptografik sistem olarak bilinen transformasyon ailesinden seçilir.

Anahtar denilen K parametresi anahtar uzayından seçilir

Diger 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.



Şekil6.2. Kriptografi Sistemi

6.6 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 alfabetdir. **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 alfabetesinden alınmış fakat P den farklı bir diziliş gösteren elemanlardan oluşur. **K** ise daha önce bahsettiğimiz **Anahtar Uzayını** (Key Space) ifade eder. Anahtar yine A alfabetesindeki 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.

$$\begin{aligned} & \forall k \in K, D_k \in D \text{ fonksiyonuna uyan bir } E_k \in E \text{ fonksiyonu vardır. Öyle ki;} \\ & \forall E_k : P \rightarrow C \text{ ve } \forall D_k : C \rightarrow P \text{ ve her } x \in P \text{ için } D_k(E_k(x)) = x \end{aligned}$$

Kriptosistemler genel olarak aşağıdaki üç bağımsız özelliği göre sınıflandırılırlar.

- Şifresiz metinden şifreli metne dönüşüm için kullanılan işlemlerin tipi:** Bütün şifreleme algoritmaları yerine koyma(substitution) 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.
- 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.
- 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.

6.6.1 Kriptolama güvenliği ve Kriptanaliz.

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

Kriptanaliz: Kriptanalitik saldırular, algoritmanın **özellikleri**, 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) saldırısı: 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 6.1'de ise Şifrelenen mesajı çözmek için yapılan saldırısı 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
Seçilen 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.
Seçilen 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 6.1: Şifrelenen mesaja karşı yapılan saldırısı Tipleri

6.6.2 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) dir.

- 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 gereklidir gibi), içinde şifre kıramaz.

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 hesabını ortalamaya 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/ μ s hızında gereken zaman	10^6 çözümleme/ μ s hızında gereken zaman
24	$2^{24} = 1.6 \times 10^7$	$2^{23} \mu\text{s} = 8.4$ saniye	8.4 μ saniye
32	$2^{32} = 4.3 \times 10^9$	$2^{31} \mu\text{s} = 35.8$ dakika	2.15 milisaniye
48	$2^{48} = 2.8 \times 10^{14}$	$2^{47} \mu\text{s} = 4.46$ yıl	2.35 dakika
56	$2^{56} = 7.2 \times 10^{16}$	$2^{55} \mu\text{s} = 1142$ yıl	10 saat
128	$2^{128} = 3.4 \times 10^{38}$	$2^{127} \mu\text{s} = 5.4 \times 10^{24}$ yıl	5.4×10^{18} yıl
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36}$ yıl	5.9×10^{30} yıl
26 karakter permutasyonu	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12}$ yıl	6.4×10^6 yıl

Tablo 6.2. : Anahtar uzunluklarına göre hesaplamaya bağlı güvenlik

6.7 Kriptografinin kısa Tarihçesi

6.7.1 Çok Eski(Ancient) şifreleyiciler

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



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

Şekil 6.3.

- Eski İbraniler kutsal kitaplarındaki belirli kelimeleri şifrelemiştir.
- 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 şifresini açıkladı.
- Kullanımı ülkelerde özellikle diplomasi ve savaşlarda artmaktadır.

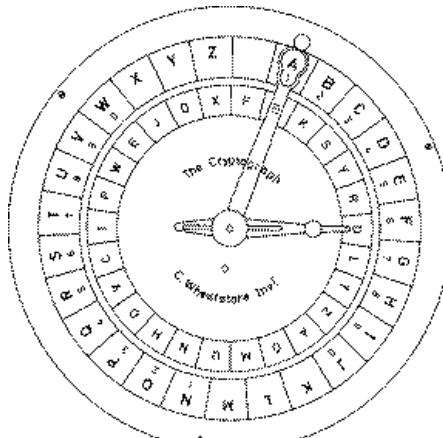
6.7.2 Makina Şifreleri

- 1790 larda geliştirilen **Jefferson cylinder**, herbiri rastgele alfabeli 36 adet diskten oluşmaktadır, disklerin sırası anahtarı oluşturmaktaydı, mesaj ayarlanınca diğer satır şifreyi oluşturmaktaydı.(Şekil 6.4)



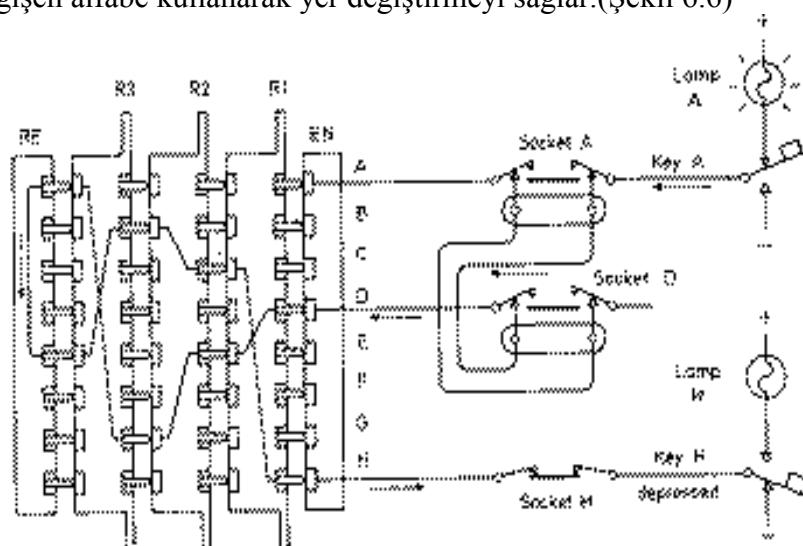
Şekil 6.4. Jefferson Cylinder

- **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 6.5)



Şekil 6.5. Wheatsone Disk

- **Enigma Rotor makinası**, ikinci dünya savaşı sırasında çok kullanılan şifre makinalarının önemli bir sınıfını teşkil eder, içinde çapraz bağlılı, bir seri rotordan meydana gelir, sürekli değişen alfabe kullanarak yer değiştirmeyi sağlar.(Şekil 6.6)



Şekil 6.6. Enigma Rotor Makinası

6.8 Sayı Teorisine Giriş

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, $a*e = e*a = a$ dir.

(G_4) G 'nin her bir elemanın tersi mevcuttur. $\forall a \in G$ için, G 'de bir a' vardır ve $a*a' = a'*a = e$ dir.

Bu bölümde ve bundan sonraki bölümlerde belirtmemiş 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.

$$n \in \mathbb{Z}^+ \text{ olmak üzere } x^n = x * x * \dots * x \text{ (n tane)}$$

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

Ayrıca etkisiz elemanı da şu şekilde tanımlarız: $x^0 = e$.

Herhangi bir $(G, *)$ grubun en belirgin özelliği büyülüklüğü 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, $a*b = b*a$ dir.

Eğer H grubu G grubunun bir alt grubu ise $|H|$ değeri $|G|$ değerini böler. Böylece eğer G grubunun *düzeni* bir asal sayısıysa 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 $i \in \mathbb{N} \cup \{\infty\}$ ve $g^i = 1$ şartını sağlayan en küçük i değeridir. Burada $\forall j, l \in \mathbb{Z}$:

$$g^j = g^l \Leftrightarrow j \equiv l \pmod{\text{ord}(g)}$$

Tablo 6.3'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 6.3

Her bir elemanın bir tamsayı olmak üzere a^n biçiminde yazabileceğimizden bu grub 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^n nin 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 elemanın bir üssüdür ve $\langle g \rangle$ ifadesiyle gösterilirler. Eğer $\langle g \rangle = G$ 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 $\langle g \rangle$ nin düzeni 1 veya p sayısı olur.

Doğal olarak, diğer başka elemanlar da grubun üretecidir? sorusu aklımıza gelir. c elemanın 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, $ab = ba$ dir.

Eğer bir komutatif halka aşağıdaki aksiyomları sağlıyor ise integral domain dir.

(H_5) Çarpımsal etkisiz eleman. $\forall a \in R$ için, $a1 = 1a = a$ dir.

(H_6) Sıfır bölen olmaması $\forall a, b \in R$ ve $ab=0$ ise ya $a=0$ veya $b=0$ dir.

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_1 den 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 $aa^{-1} = (a^{-1})a = 1$ dir.

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

Modüler Aritmetik

Modüler aritmetik “saat aritmetiği”dir

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:

$$a = k \cdot n + r$$

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 , $a \mod n$ ile bölündüğünde kalan olarak tanımlanır. Böylece herhangi a tamsayı için,

$$a = [a/n]x n + a \text{ mod } n \text{ her zaman yazılabilir. (Örn: } 11 \text{ mod } 7 = 4\text{)}$$

a ve b iki tamsayısı eğer $a \equiv b \pmod{n}$ iseler benzer modulo n olarak tanımlanır ve $a \equiv b \pmod{n}$ olarak yazılabilir.

Bölenler: Eğer sıfır olmayan bir b ve m tamsayısı için $a \equiv mb$ şeklinde yazılabilirse 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 $a = \pm 1$ dir.
- Eğer $a|b$ ve $b|a$ ise $a = \pm b$ 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,

$$(a_1 \text{ op } a_2) \pmod{n} \equiv [(a_1 \pmod{n}) \text{ op } (a_2 \pmod{n})] \pmod{n}$$

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

- Bir $a \equiv b \pmod{n}$ eşitliği, a ve b aynı n ile bölündüğünde aynı kalanı verdiklerini ifade eder.
- Örnek,
 - $100 \equiv 34 \pmod{11}$
 - Genellikle $0 \leq b < n-1$ dir.
 - $2 \pmod{7} = 9 \pmod{7}$
 - b 'ye $a \pmod{n}$ 'nin kalanı denir.
- Tamsayı modulo n ile yapılan bütün aritmetikte bütün sonuçlar 0 ve n arasında olur.

Modül işleminin özellikleri

Modül işlemi aşağıdaki özelliklere sahiptir.

Eğer, $n|(a-b)$ ise $a \equiv b \pmod{n}$ dir.

$a \equiv b \pmod{n}$, $b \equiv c \pmod{n}$ anlamına gelir.

$a \equiv b \pmod{n}$ ve $b \equiv c \pmod{n}$, $a \equiv c \pmod{n}$ anlamına gelir.

Modüler Aritmetik işlemleri

Toplama

$$(a+b) \pmod{n} = [(a \pmod{n}) + (b \pmod{n})] \pmod{n}$$

Çıkartma

$$(a-b) \pmod{n} = [(a \pmod{n}) - (b \pmod{n})] \pmod{n}$$

Çarpma

$$axb \pmod{n} = [(a \pmod{n}) \times (b \pmod{n})] \pmod{n}$$

- Tekrarlanan toplamdan türetilir
- Ne a ne de b sıfır değil iken $a.b=0$ olabilir
 - örnek $2.5 \pmod{10}$

Bölme

$$a/b \pmod{n}$$

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

Özellikler :

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

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

Z_n kalanlar sınıfı olarak adlandırılır. Daha doğrusu, Z_n de her bir tamsayı bir kalan sınıfını temsil eder. $[r] = \{ a : a \text{ bir tamsayı; öyleki } a = r \text{ mod } n \text{ dir.}\}$

Z_n içerisinde yapılacak modüler aritmetik işlemleri Tablo 6.4'deki özelliklerini Z_n deki tamsayılar ile sağlar. Z_n çarpımsal etkisiz eleman ile birlikte bir değiştirilebilen bir halka oluşturur.

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

Tablo 6.4.

- Aynı zamanda, indirgeme tamsayılar halkasından tamsayı modulo n 'lerin halkasına bir homomorfizm olduğu için, bir işlem ve sonra modulo n i indirgeyip indirgemeyeceği veya indirdikten sonra yapacağı işlem seçilebilir.
 - $a+/-b \text{ mod } n = [a \text{ mod } n +/- b \text{ mod } n] \text{ mod } n$
 - $(a.b) \text{ mod } n = ((a \text{ mod } n).(b \text{ mod } n)) \text{ mod } n$
- 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.

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

Birçok kriptografik algoritmada sonlu alanlar önemli bir rol oynarlar. Bir sonlu alanın düzen(order) 1 bir p asal sayısının n . kuvveti(p^n) olarak gösterilmelidir. Burada n pozitif bir tamsayıdır. Düzeni p^n olan bir sonlu alan, genellikle $GF(p^n)$ olarak yazılırlar. GF sonlu alanı ilk defa çalışan matematikçi olan Galois'ten 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 $GF(p)$, $\{0,1,\dots,p-1\}$ 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 $w \times z = 1 \text{ mod } p$ 'dir.

Çünkü, w , p ye göre asaldır. Eğer, Z_p nin elemanlarını w ile çarparsa, 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 6.5'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 6.5 Modulo 7 için toplamsal ve çarpımsal tersler

Asal Sayılar

Bir $p > 1$ sayısı ancak bölenleri ± 1 ve $\pm p$ ise asal sayıdır. Asal sayılar, Açık-anahtarlı kripto sistemlerinde büyük rol oynarlar. Asal sayıarda karşımıza çıkan önemli problemler, asal bir sayının oluşturulması ve bir sayının asal olup olmadığını 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.

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

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

Tanım: $a^{s-1} \equiv 1 \pmod{s}$ 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,

$$a^p \equiv a \pmod{p} \quad \text{denkliği;}$$

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

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

$$\begin{aligned} ax \cdot 2ax \cdots ((p-1)a) &= [(a \pmod{p}) \times (2a \pmod{p}) \times \cdots \times ((p-1)a \pmod{p})] \pmod{p} \\ &= [1 \times 2 \times \cdots \times (p-1)] \pmod{p} \\ &= (p-1)! \pmod{p} \end{aligned}$$

Fakat, $a \times 2a \times \cdots \times ((p-1)a) = (p-1)!a^{p-1}$ dir

Bu yüzden, $(p-1)!a^{p-1} = (p-1)! \pmod{p}$ dir. Burada $(p-1)!$ 'i atabiliyoruz. Sonuçta:

$$a^{p-1} = 1 \pmod{p}$$

Örn: $a=7$, $p = 19$ verilsin.

$$7^2 = 49 = 11 \pmod{19}$$

$$7^4 = 121 = 7 \pmod{19}$$

$$7^8 = 49 = 11 \pmod{19}$$

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

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

alternatif olarak $a^p \equiv a \pmod{p}$ olarak da yazılabilir.

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.

$n=pq$ 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, \mathbb{Z}_n 'deki kalanlar kümesinin $[0, 1, \dots, (pq-1)]$. Olduğunu düşünelim. Kalanlar kümesindeki $\{p, 2p, \dots, (q-1)p\}$, $\{q, 2q, \dots, (p-1)q\}$ ve 0 , n 'e göre asal değildirler. Buna uygun olarak,

$$\begin{aligned} \phi(n) &= pq - [(q-1) + (p-1) + 1] \\ &= pq - (p+q) + 1 \\ &= (p-1) \times (q-1) \\ &= \phi(p).\phi(q) \end{aligned}$$

elde edilir. Tablo 6.6'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 6.6. 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

$$F: X \longrightarrow Y$$

$$f: x \longrightarrow f(x)=y \quad \text{yalnız ve yalnız aşağıdaki şartları taşıdığı takdirde}$$

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.

6.8.1 $GF(p)$ 'de üstel işlem

- Birçok kriptolama algoritması üstelleştirmeyi kullanır, b üssü ne göre büyüyen bir a sayısı(taban) $\bmod p$
 - $b = a^e \bmod p$
- 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.

6.8.2 $GF(p)$ ‘de ayrik Logaritma Problemi

Ayrik 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 sayısı için; α^n , α nin n kere kendisi ile çarpımından elde edilisin; $\alpha^2 = \alpha * \alpha$, $\alpha^3 = \alpha * \alpha * \alpha$,

Ayrik 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 sağlamasın. Örneğin, $3^x \equiv 13 \pmod{17}$ probleminin çözümü 4 ‘tür. Çünkü $3^4 = 81 \equiv 13 \pmod{17}$ dir.

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

Bir başka ifade ile ayrik logaritma :

- Üstelleştirmede ters problem, bir modulo p sayısının ayrik logaritmasının bulunmasıdır.
 - $\alpha^x = b \pmod{p}$ ‘de x ’i bul
- üstelleştirme nispeten kolay iken, ayrik logaritmanın bulunması genellikle kolay yolu olmayan zor bir problemdir.
- Bu problemde, eğer p asal ise , herhangi bir $b \neq 0$ için her zaman bir ayrik logaritması olan bir α olduğu gösterilebilir.
 - α ’nın ardışıl kuvvetleri mod p ile **grup** oluşturur
 - $\alpha \pmod{p}$, $\alpha^2 \pmod{p}$, ..., $\alpha^{p-1} \pmod{p}$ 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;

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

Üs x ayrik logaritma veya indis olarak gösterilir.

6.8.3 En Büyük ortak Bölen(Greatest Common Divisor)

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

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

- 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 $g0=n$

$g1=a$

$gi+1 = gi-1 \pmod{gi}$

when $gi=0$ then $(a,n) = gi-1$

örn. $(56,98)$ ‘i bulalım.

$g0=98$

$g1=56$

$g2 = 98 \pmod{56} = 42$

$g3 = 56 \pmod{42} = 14$

$g4 = 42 \pmod{14} = 0$

sonuçta EBOB $(56,98)=14$

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 parçalanabileceğinin olacaktır. Bu Chinese kalan teoremi ile yapılabilecektir.

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

$$x \equiv 13 \pmod{27}$$

$$x \equiv 7 \pmod{16}$$

Bu uygulamada iki modülo birbire 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 ;

$$x \equiv a \pmod{m}$$

$$x \equiv b \pmod{n}$$

x için modulo mn şeklinde tekbir çözümü vardır. Örnek problemde $\text{mod } 16 \cdot 27 = 432$ tek bir çözümü olacaktır.

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

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

İlk benzerliği sağlayan sekans $2, 5, 8, 11, 14, 17, \dots$ dir. Bu sekans tarandığında 5' 2 bölündüğü zaman 3 kalan terim 8 olduğu için cevap 8'dir. Bunun daha kolay bulılması için Öklid'in en büyük ortak bölen algoritmasından faydalанılır.

Bütün işlemi genelleştirirsek ;

$$x \equiv a \pmod{m}$$

$$x \equiv b \pmod{n}$$

Önce $mu+nv=1$ denklemini sağlayan, u ve v tamsayıları bulunmalıdır. Sonra bütün çözümler $x = (mu)b + (nv)a \pmod{mn}$ ni sağlamalıdır.

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

Önce; $100u + 49v = 1$ çözülmelidir.

Euclid's algoritması aşağıdaki şekilde kullanılır.

Bölünen	=	Bölüm	.	Bölen	+	Kalan	0	1
100	=	2	.	49	+	2	2	1
49	=	24	.	2	+	1	49	24
2	=	2	.	1	+	0	100	49

Buradan $49 \cdot 49 - 24 \cdot 100 = 1$ dir. Çözüm $49 \cdot 49 \cdot 23 - 24 \cdot 200 \cdot 31 = -19177 \equiv 423 \pmod{4900}$ dir.

Genel hali;

$$\begin{aligned}x &\equiv a_1 \pmod{m_1} \\x &\equiv a_2 \pmod{m_2} \\&\dots \\x &\equiv a_r \pmod{m_r} \text{ ve } OBEB(m_i, m_j) = 1, i \neq j\end{aligned}$$

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

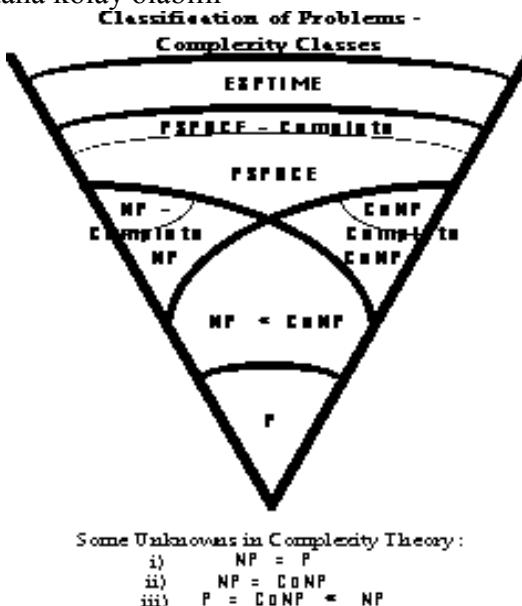
$$x = \sum a_i \cdot M_i \cdot N_i$$

$$M = m_1 \cdot m_2 \cdot m_3 \dots m_r \text{ ve } M_i = M / m_i, \quad N_i = M_i^{-1} \pmod{m_i}.$$

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 $Z_p \times Z_q$ 'ya taşınarak daha küçük bit uzunluklu verilerle işlemler basitleştirilir.

6.9 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



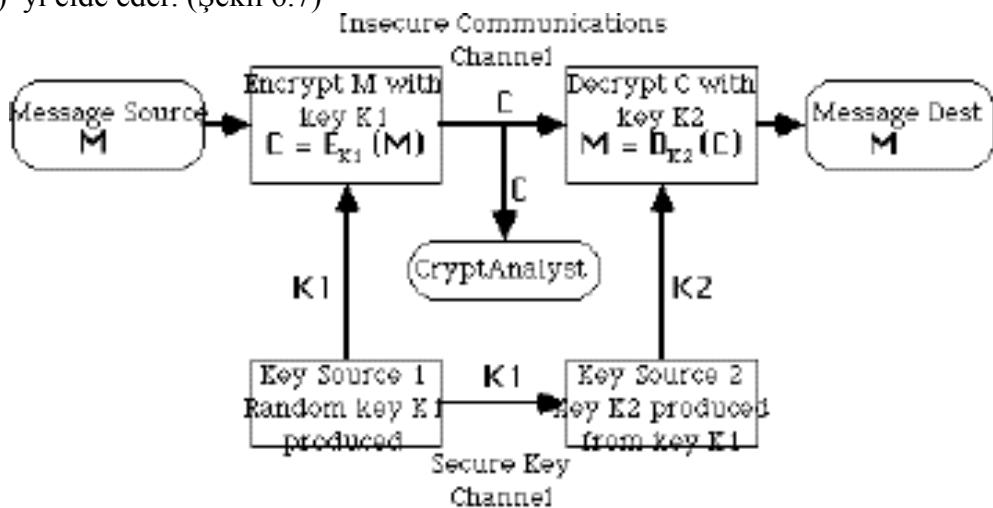
6.9.1 Karmaşıklık Teorisi- Bazı Terminoloji

- Bir problemin anlık durumu genel bir problemin kısmi örneğidir
- Bir problemin giriş uzunluğu, onun kısmi örneğini karakterize etmek için kullanılan n simbol sayısıdır.
- Bir fonksiyonun derecesi, $f(n)$ bazı $g(n)$ in $O(g(n))$ idir.
 - $f(n) \leq c|g(n)|$, bütün, $n \geq 0$, 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 eşlenigidir, Co-NP problemlerinin bir çözümünü tahmin etmek çözüm uzayının detaylı araştırılmasını gerektirir

6.10 Gizli anahtarlı (simetrik) kriptosistemler :

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 arasında 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 6.7)

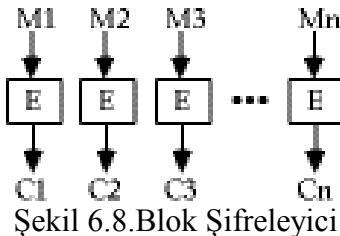


Symmetric (Private-Key) Encryption System

Şekil 6.7 Gizli-anahtarlı kriptosistem ile haberleşme

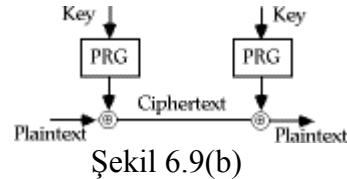
Gizli-anahtarlı kripto sistemleri uygulama sahalarında ikiye ayrılır;

- 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 ayıklanması durumunda anahtar boyunun 56 bite indiğini hatırlatmak gereklidir. Diğer bilinen blok şifrelemeli algoritmalar ise FEAL, IDEA ve RC5 örnek olarak gösterilebilir. Çalışacağımız çoğu modern şifreleyici bu formdadır. (Şek. 6.8)



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. 6.9(a))
- mesaj biti kadar anahtar biti gerekir. Pratikte zordur.(örn. Pratikte mag teyp veya CDROM da dağıtırlır)
- anahtar tamamen rastgele olduğu için koşulsuz güvenlik sağlanır.
- böyle büyük bir anahtar dağıtımları güç olduğu için anahtar dizisi daha küçük(taban) bir anahtardan üretilebilir. Bunun için rasgele simbol fonksiyonları kullanılır.(şek 6.9(b))
- Her ne kadar bu çok çekici gözükse de pratikte iyi bir kriptografik güçlü rasgele fonksiyon bulmak çok güçtür. Bu hala birçok araştırmacının konusudur.



6.11 Simetrik Şifreleme Algoritmaları

Geleneksel simetrik blok şifreleme algoritmaları(örn. DES) 1973'de IBM'de çalışan Horst 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 Shannon tarafından önerilen confusion ve diffusion fonksiyonlarını şifreleme algoritmasında önerdi.

Diffusion da, şifresiz metnin istatistiksel yapısı, şifreli metnin istatistiğine dağıtır. Bu, şifresiz metnin her birijinin, şifreli metnin etkilediği dijitalerinin bulunmasıyla sağlanır., başka bir ifade ile, her bir şifreli metin dijiti' i birçok şifresiz metin dijiti tarafından etkilenir. Örnek olarak; Bir $M = m_1, m_2, m_3, \dots$ karakterlerinden oluşan bir şifresiz metni ortalama işlemi ile k ardışılık karakteri ekleyerek şifrelemek;

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

63 Dr.İ.SOĞUKPINAR G.Y.T.E. Bil.Müh.Böl.

Confusion'da ise, anahtarın keşfedilmesi saldırılara karşı, şifreli metnin istatistiği ile şifreleme anahtarının olabildiğince karmaşık olmasını araştırır. Böylece bir saldırgan şifreli metnin istatistiğini hesapla 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 işlemeye tabi tutulur. Daha sonra 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 ,

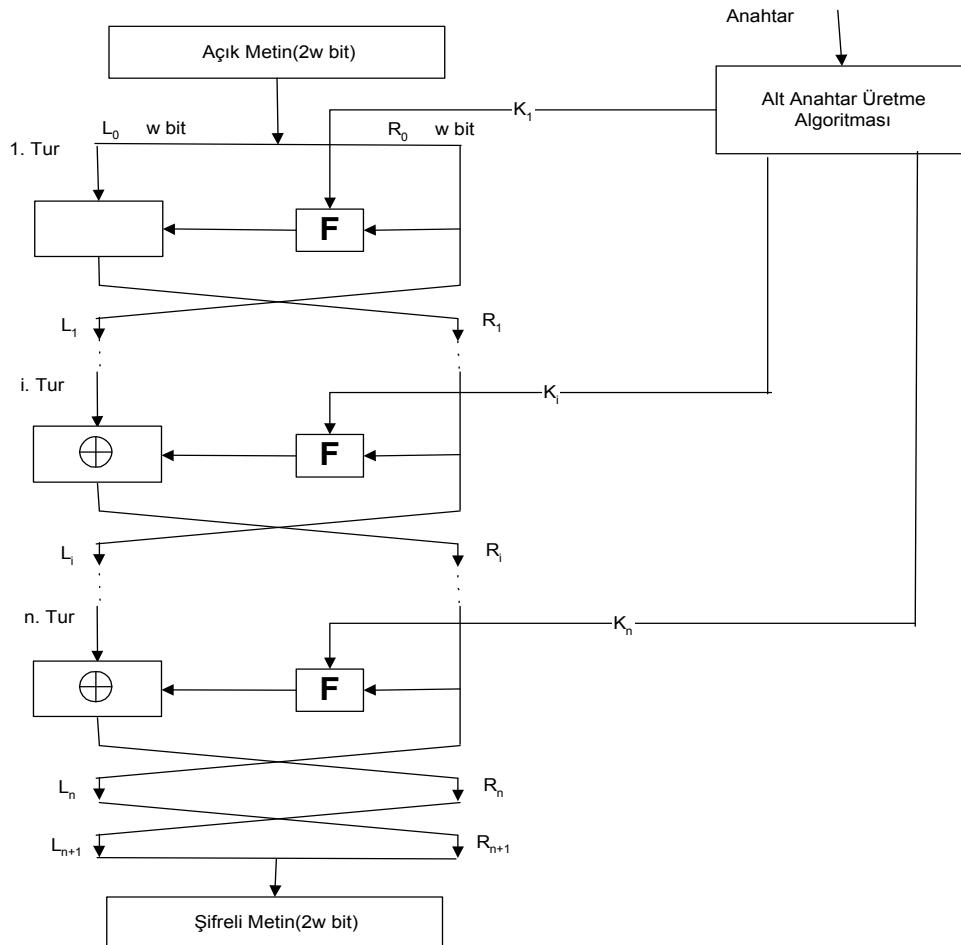
Hızlı yazılım şifreleme/deşifreleme: Çoğu uygulamada, şifreleme uygulamaları veya donanım gerçeklemesi şeklinde kullanım fonksiyonlarının içine koymak. Dolayısı ile algoritmanın icra hızının düşünülmeli gerekir.

Analiz Kolaylığı : Her ne kadar algoritmanın olası kriptanaliz saldırılara 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 gerekmek.

Algoritmanın genel matematiksel hesaplanması; LE_i : Sol şifrelenmiş blok, RE_i : Sağ şifrelenmiş blok, olmak üzere,

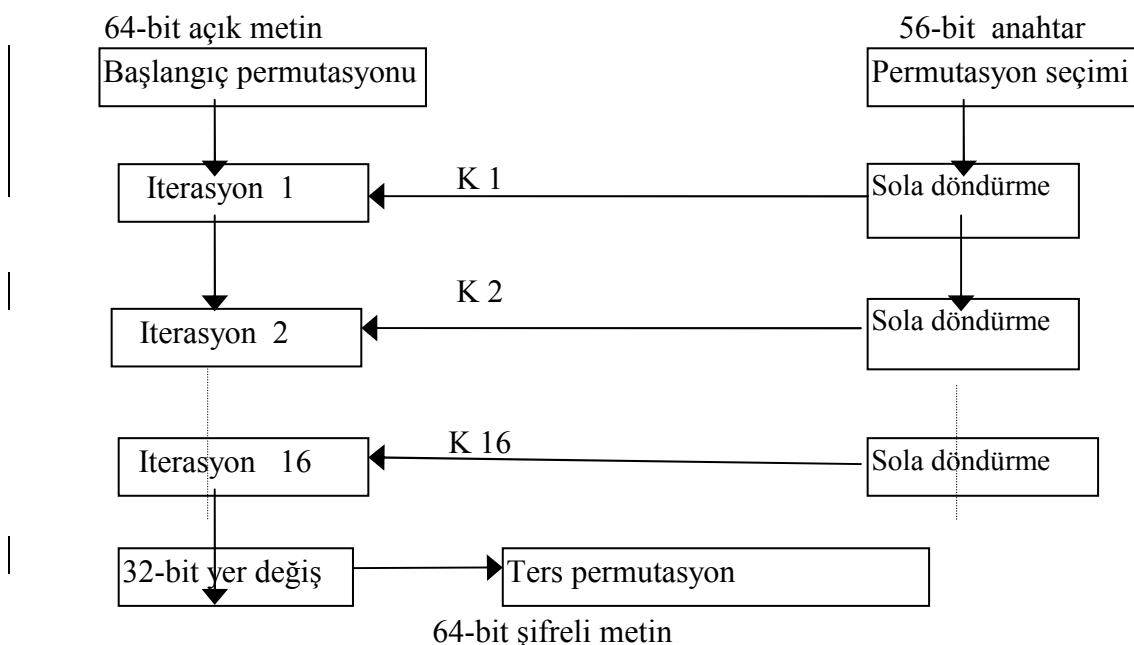
$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$



Şekil 6.10. Klasik Feistel Network

6.11.1 DES

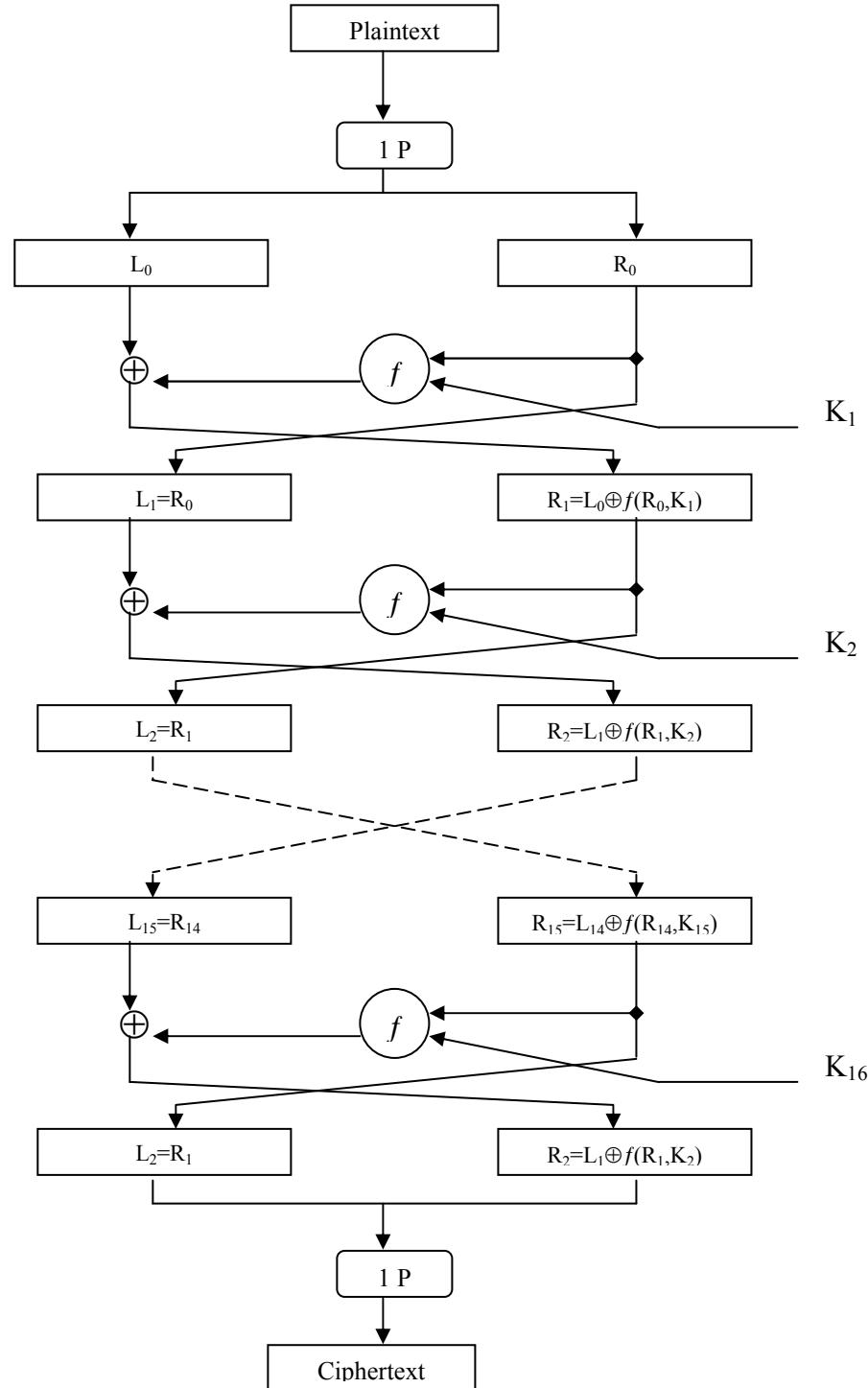
Data Encryption Standard (DES) 1974 yılında IBM tarafından geliştirilmiş ve 1977 yılında yasal olarak atanmıştır. Basit blok şema Şekil 6.11'de gösterilmiştir. Temeli Feistel networküne dayanır.



Şekil 6.11. 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 6.12 DES Algoritması

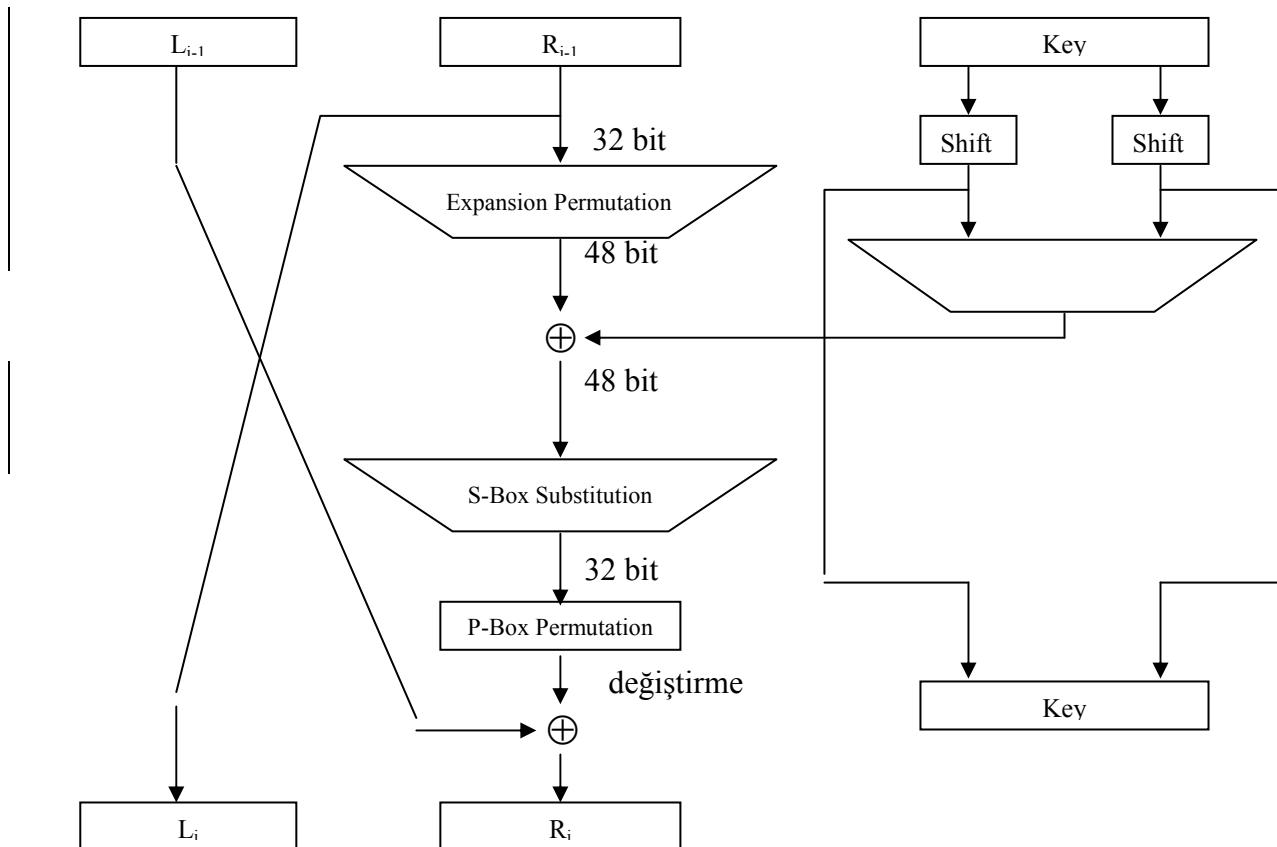
Algoritmanın Özeti :

DES 64 bit blok plaintext de işlem görür. Plaintext, ilk permutasyondan sonra yarısı sağda yarısı solda her biri 32 bit uzunluğunda iki parçağa 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 içeresine 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 yarımlı olmakta ve sol yarımlı ise sağ yarımlının eski hali olmaktadır. (6.1) de gösterilen bu işlem 16 kez tekrar eder.

$$L_i = R_{i-1} \quad R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \quad (6.1)$$

$$\text{Genel} \quad m_{i+1} = m_{i-1} \oplus f(m_i, K_i) \quad (6.2)$$



Şekil 6.13 DES' in bir turu

Başlangıç Permutasyonu :

Başlangıç permutasyonu tur 1' den önce meydana gelir. Şifrelemeden önce 64 bitlik plain text 32 bitlik iki parçağa bölünür. Tüm çift bitler sol tarafta ve tek pozisyondaki bitler de sağ tarafta yer alır. Tablo 9.3' de tanımladığı 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 6.7 Başlangıç Permutasyonu

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

Anahtar Dönüşümü :

Başlangıçta, 64 bitlik DES anahtarı her sekiz bit ihmali edildiği için 56 bite düşürülür. Bu tablo 6.8' de tanımlanmıştır. İhmali 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 6.8 Anahtar Permutasyonu

İlk olarak 56 bitlik anahtar 28 bitlik iki parçağa bölünür. Turun ihtiyacına göre parçaların bir veya iki biti değiştirilir. Değiştirilecek bit sayıları tablo 6.9' 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 1 2 2 2 2 2 2 1

Tablo 6.9 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 6.10' da compression permutation tanımlanmıştır.

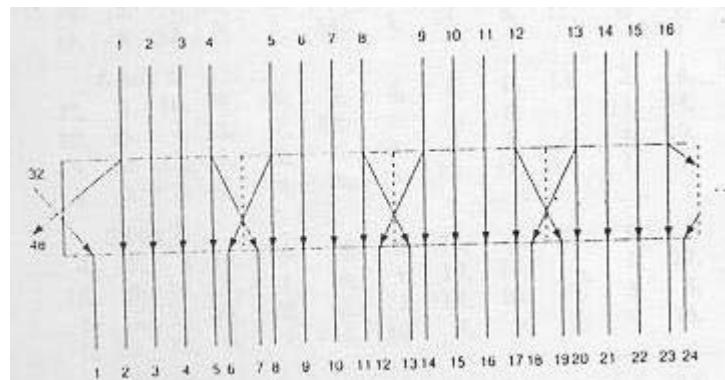
14 17 11 24 1 5	3 28 15 6 21 10
23 19 12 4 26 8	16 7 27 20 13 2
41 52 31 37 47 55	30 40 51 45 33 48
44 49 39 56 34 53	46 42 50 36 29 32

Tablo 6.10 Sıkıştırma Permutasyonu

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 sağ yarımi anahtar ile aynı uzunlukta yapmak ve yerine koyma (substitution) işlemi sırasında sıkıştırılabilen daha uzun sonuç sağlamak.

Şekil 9.14' 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.



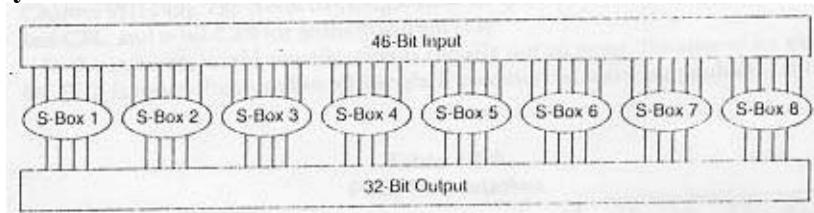
Şekil 6.14 Genişleme Permutasyonu

Tablo 6.11'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 6.11 Genişleme Permutasyonu

S-Box Yerine Koyma :



Şekil 6.15 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 işlenir. 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 6.12'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 6.12 S-Box lar

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 çağrırlar. Tablo 6.13'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 6.13 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.

Sonuç Permutasyonu :

Sonuç permutasyonu başlangıç permutasyonunun tersi şekilde çalışır ve tablo 9.10' 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 6.14 Sonuç Permutasyonu

Çığ Etkisi :

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

DES' in Güvenliği :

Anahtar Uzunluğu ;

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

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

DES'in alternatifleri olab 3DES ve AES geliştirilmiştir.

DES zamanlama saldırılarına karşı oldukça güçlündür.

Diferansiyel ve Doğrusal(Lineer) Kriptoanaliz.

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

Diferansiyel 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 şifrelenen metin bloğu iki eşit parçaya ayrılır ($m = m_0 + m_1$) Her bir çevrimde $2 \leq i \leq 17$ olmak üzere m_i yeni blok blok elde edilir.

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

Diff. Criptanaliz

$$\Delta m = m \oplus m' \quad (\text{Mesaj yarları})$$

$$\Delta m_i = m_i \oplus m'_i$$

$$\begin{aligned} \Delta m_{i+1} &= m_{i+1} \oplus m'_{i+1} \\ &= m_{i-1} \oplus f(m_i, K_i) \oplus m'_{i-1} \oplus f(m'_i, K'_i) \\ &= \Delta m_{i-1} \oplus [f(m_i, K_i) \oplus f(m'_i, K'_i)] \end{aligned}$$

Eğer biz, Δm_{i-1} ve Δm_i 'yi yüksek bir olasılık ile bilirse Δm_{i+1} 'i de yüksek olasılık ile bileyebiliriz. 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 Criptanalizin 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) Criptanaliz

Diğer bir yöntem ise doğrusal criptanalizdir. Doğrusal criptanalizde DES için 2^{47} bilinen şifresiz metin ile 2^{47} seçilen şifresiz metin karşılaştırılarak anahtar bulunabilir. Her ne kadar bu küçük bir iyileştirme olsada doğrusal criptanaliz 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 yaklaşım ve bir p olasılığı ile sağlanır. Eğer bu olasılık $p \neq 0,5$ ise, bu işlem anahtarın bulunması için kullanılabilir. Toplanan şifresiz metinler ve karşılığında atanmış şifreli metinler anahtar bitlerinin tahmin edilmesi için kullanılabilir. İşlemler aşağıda matematiksel olarak açıklanmıştır.

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

$$P[1], P[2], \dots, P[n] \text{ ve } C[1], C[2], \dots, C[n]$$

$K[1], K[2], \dots, K[m]$ olsun ve;

$$A[i, j, \dots, k] = A[i] \oplus A[j] \oplus \dots \oplus A[k] \text{ tanımlansın. (bitler bir biri ile XOR'lanır)}$$

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

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

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

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

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 yarımda 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 diğeri yarısı 0' lardan oluşuyorsa meydana gelir.

Tablo 9.11' de 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 6.15 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.

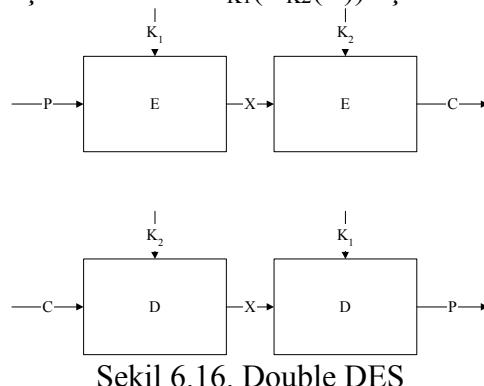
DES 16' dan daha az turda gerçekleştiği zaman, brute force saldırıları olarak bilinen saldırılarla daha kolay ve verimli bir şekilde kırılabilir.

DES'in Farklı Şekilleri :

Double DES :

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

Şifreleme $C = E_{K_2}(E_{K_1}(P))$ Deşifreleme $P = D_{K_1}(D_{K_2}(C))$ şeklinde olacaktır.



Şekil 6.16. Double DES

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

Ortada Karşılaşma(Meet in the middle attack) saldırısı

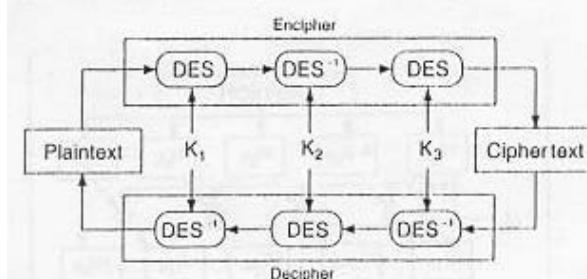
Şekil 6.16.'dan görüldüğü gibi X değerinin hesabı aşağıdaki şekilde yapılabilir.

$$X = E_{K_1}(P) = D_{K_2}(C)$$

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. Herikisinde 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 6.17) 3DES iki veya üç ayrı anahtar kullanılarak yapılabilir.

$$C = E_{K_3}(E_{K_2}(E_{K_1}(P))) ; \quad P = D_{K_1}(D_{K_2}(D_{K_3}(C)))$$



Şekil 6.17 Triple DES

CRYPT(3) :

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

Generalized DES :

Generalized DES (GDES), algoritmayı kuvvetlendirmek ve DES'i hızlandırmak amacıyla tasarlanmıştır. Hesap miktarı sabit iken blok boyutu artırılmıştır.

DES varyanslarına ek olarak DESX, RDES, s^n DES de verilebilir.

6.11.2 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. Criptanalizi 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 çalışıkları bir algoritmadır.

6.11.3 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 kullanan algoritma olduğu için kriptoanalistlerin üzerinde çok çalışıkları bir algoritmadır.

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 gereklidir. Dolayısı ile çok sık gizli anahtar değişimi gerektiren uygulamalarda blowfish kullanılması uygun değildir.

6.11.4 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.

6.11.5 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 britten 128 bit'e kadar değişen anahtar uzunlukları kullanır. CAST, DES'te kullanılanlardan daha uzun olan sabit S-boxlar kullanır. Bu S-boxların tasarımları 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 6.16. 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

6.12 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 modları ile gelinebilir.

6.12.1.1 CBC(Cipher Block Chaining Mode)

Bu modda (CBC) o andaki şifresiz metin bloğu ile bir önceki şifreli metin bloğu, XOR mantıksal işlemeye 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:

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

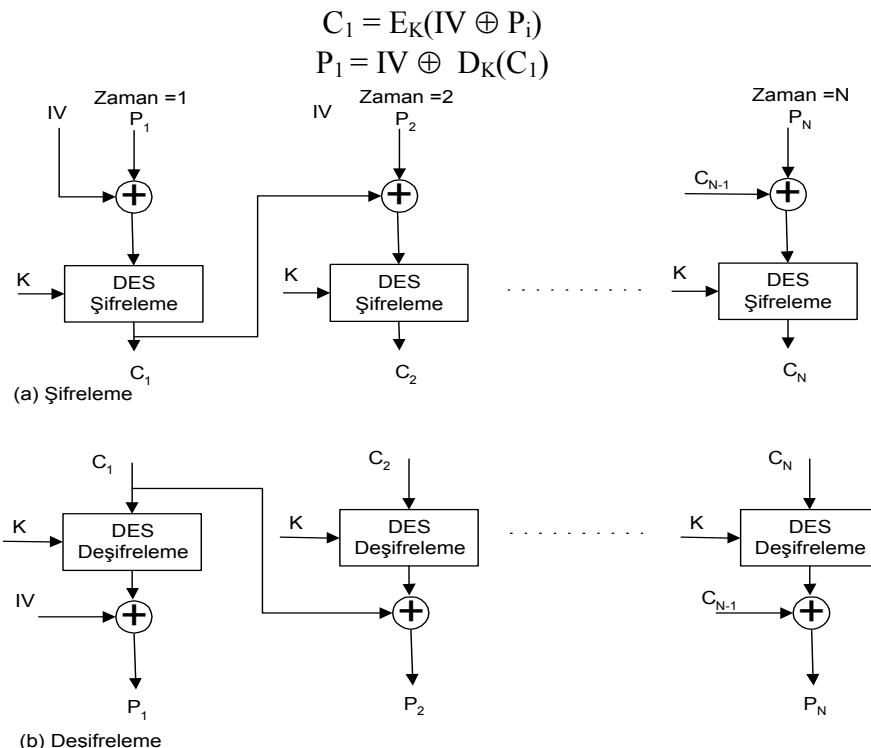
Burada $E_K[X]$, X'in K anahtarı kullanılarak şifrelenmiş şekli ve \oplus ise XOR işlemidir. Sonra,

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

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

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

Şekil 6.18 'de görüldüğü gibi, ilk şifreli bloğu elde etmek için başlatma vektörü(IV) ilk açık metin bloğu ile XOR işlemeye 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 gereklidir. İlk bloğun şifrelenmesi aşağıdaki ifadede gösterilmiştir.



Şekil 6.18. CBC (Cipher Block Chaining Mode)

6.12.1.2 CFB(Cipher Feedback Mode)

DES tasarımlı 64 bitlik blok şifrelemeyi kullanır. Bununla birlikte CFB modu ile DES'i dizi şifreleyici haline dönüştürmek mümkün olmaktadır. Bu yapıda her bir karakterin 8 bit olduğu

varsayımlı ile 8 bitlik alt bloklar ile yapılan şifrelemede karakter bazında dizi şifrelemesi gerçekleştirilmiş olmaktadır.

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

Deşifreleme için düz metin birimini elde etmek için alınan şifreli metin biriminin şifreleme fonksiyonunun çıkışları 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.

6.12.2 AES (Advanced Encryption Standard)

3DES algoritması her ne kadar 168 bitlik anahtar kullanıyor ve brute-force saldırılara karşı yeterli güvenlik sağlıyor ise de üç adet DES’ın ard arda çalışması nedeniyle yavaş bir algoritmadır. Bu nedenle NIST 1997’de 3DES’ın 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 özelliklerini aşağıda verilmiştir.

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

Şekil 6.19 ‘da blok diyagramı gösterilen AES’in çalışması aşağıda özetlenmiştir.

- 1 DES(Feistel) mimarisinde veri bloğunun yarısı diğer yarısını modifiye etmeye çalıştırılır, 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.

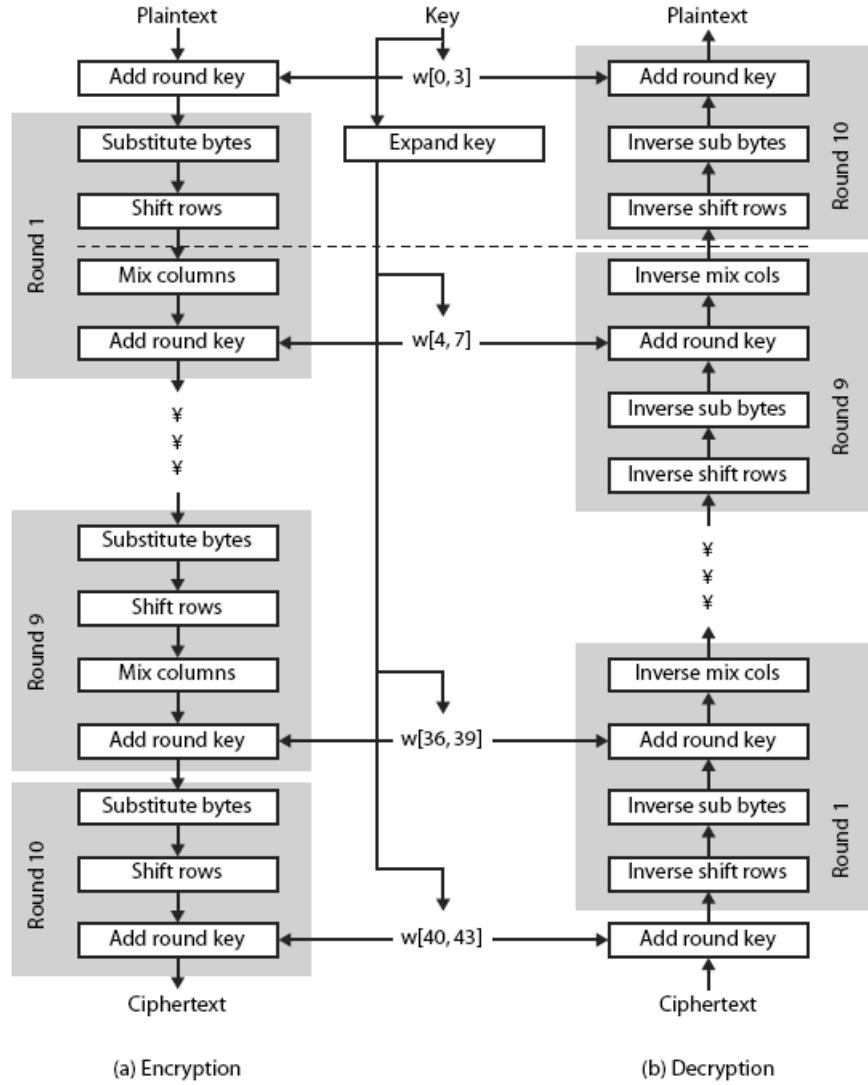
Herbir turdaki dört farklı evrede, bir permutasyon ve üç yer değiştirme kullanılır.

- 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. Her biri 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 deşifreleme algoritması anahtarları 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 Desifreleme evreleri:

10 Deşifreleme evreleri:

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



Şekil 6.19 : AES Şifreleme ve Deşifreleme adımları

Gizli anahtarlı (simetrik) kriptosistemlerin Güvenliği :

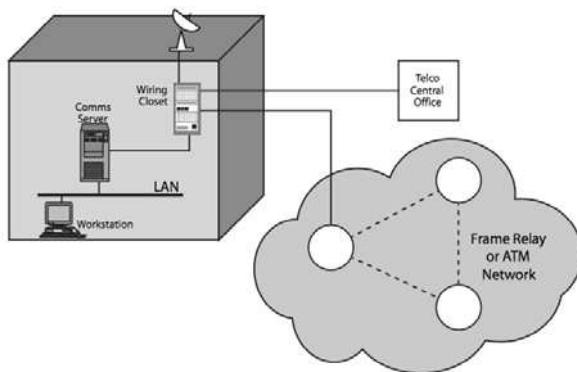
Geleneksel olarak simetrik şifreleme mesaj gizliliğini sağlamak için kullanılır.

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 birçofta anahtar gerektir
 - b. Uçtan uca şifreleme : Şifreleme orijinal kaynak ve son varış noktası arasında yapılır. Her iki ucda paylaşılmış anahtarlar ve cihazlar gerekir.

Şekil 6.20'de gösterilen haberleşme ağı'nda açıklık noktaları belirtilmiştir. YAS'ne bağlı olan bir iş istasyonunun gönderdiği mesajlar YAŞ'ın özelliği itibarı ile dinlenmeye müsaittir. Haberleşme sunucusuna erişim hakkı elde eden bir saldırgan ağ trafiğini dinleyip analiz edebilir. YAS'ının

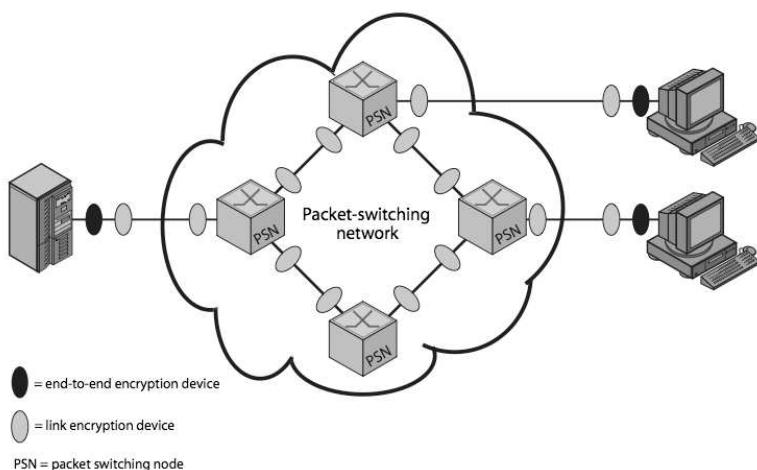
dışında bir yönlendirici veya çevirmeli modem ile dış ağa bağlantı olabilecektir. Bunların bağlantı noktaları zayıf noktalardır. Dış ağdaki herhangi bir haberleşme bağlantısı saldırıyla açık yerlerdir. Böylece saldırıyla açık birçok nokta bulunduğu görülmektedir.



Şekil 6.20. Açıklık noktaları

Bağlantılara karşı uçtan uca Şifreleme

İletişimde şifreleme için iki yöntem düşünülebilir. Herbir bağlantı yi ayrı ayrı şifrelemek ve uçtan uca haberleşmeyi şifrelemek Şekil 6.21'de bir paket anahtarlama ağ'da bağlantılarının ve uçtan uca haberleşmenin şifrelenmesi gösterilmiştir



Şekil 6.21. Paket anahtarlama ağ'da şifreleme

Uçtan uca haberleşme kullanıldığı zaman başlık şifresiz olarak bırakılmalıdır. Böylece ağ yönlendirme bilgisini doğru olarak sağlayabilir.

Bu nedenle her ne kadar, içerik şifrelensesde, trafik izi akışını anlamak mümkündür. Idealde heriki şifrelemede

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

Bağlantı şifreleme ise trafik akışının gözlenmesini engeller

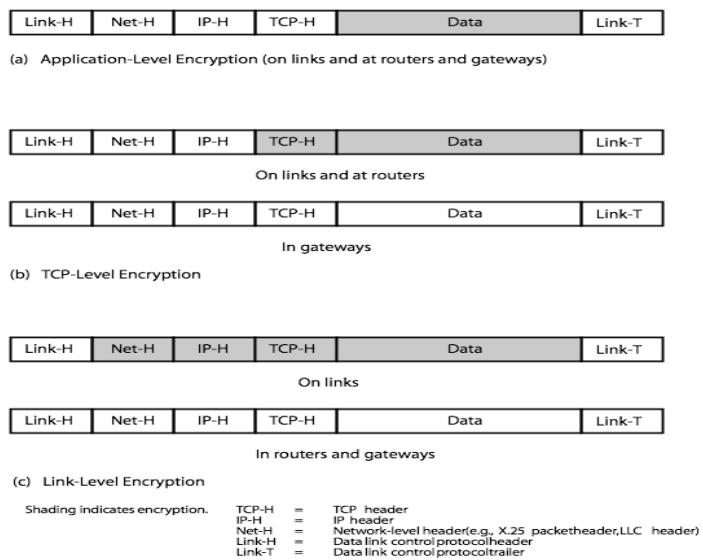
OSI referans modelinin değişik katmanlarında şifreleme fonksiyonu sağlanabilir

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 6.22'de gösterilen protokol seviyelerindeki şifrelemelerde üst katmanlarda daha az verinin şifrelendiği, alt katmanlarda ise daha fazla verinin şifrelendiği görülmektedir.



Şekil 6.22: Şifreleme ve protokol seviyeleri arasındaki bağlantı.

Trafik Analizi, iletişim grupları arasındaki haberleşme akışını gözlemektedir.

Askeri ve ticari alanda faydalı olabilir

Gizli bir kanal oluştumakta kullanılabilir

Bağlantı şifreleme başlık detaylarını gizler fakat, ağ parçalarında ve üç noktalarda hala gözlenebilir

Trafik padding akışı anlaşılmaması güç haler getirir fakat, sürekli trafigin maliyeti artar

Anahtar Dağıtımı

Şimetrik şifreleme yöntemlerinde ortak bir anahtar her iki grub 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 dağıtım alternatifleri olabilir

A anahtarını secer ve fiziksel olarak B'ye iletir.

Üçüncü şahıs anahtarını secer, A ve B'ye dağıtır

Eğer A ve B önceden haberleşiyorsa, önceki anahtarları kullanarak yeni anahtarını şifreler

Eğer A ve B, C ile birlikte güvenli bir iletişim kanalına sahipse, C anahtarını 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 kullanılabilir.

Oturum anahtarı;

Geçici anahtardır

Verinin kullanıcılar arasında şifrelenmesi için kullanılır.

Tek bir oturumda kullanılır ve sonra atılır.

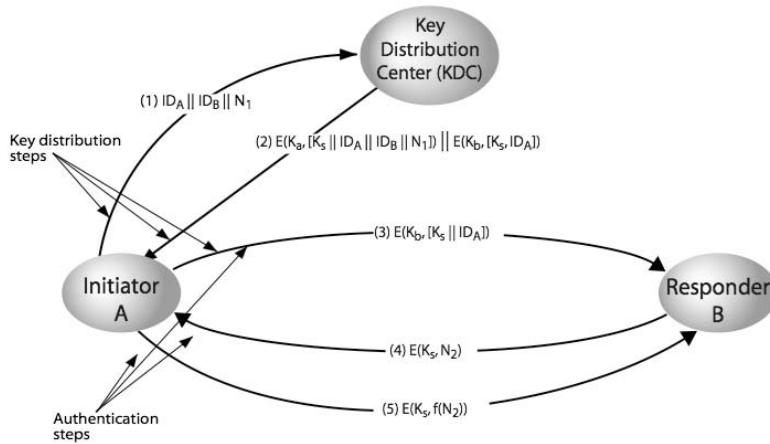
Ana Anahtar

Anahtar dağıtım merkezi ile kullanıcılar arasında N adet tır.

Ana anahtar;

Oturum anahtarlarını şifrelemek için kullanılır

Kullanıcı ile anahtar dağıtım merkezi arasında paylaşılır



Şekil 6.23: Anahtar dağıtımlı senaryosu

Merkezi olmayan anahtar dağıtımlı

Merkezi olmayan anahtar dağıtımda, herbir üç sistem, oturum anahtarı dağıtımlı için güvenli bir şekilde haberleşmesi gereklidir. Böylece N adet üç sistemin konfigürasyonu için $N(N-1)/2$ adet anahtar gerekebilir.

Oturum anahtarı aşağıdaki adımlar ile sağlanır

- A,B 'den N_1 içeren bir mesaj ile oturum anahtarı ister
- B, ortak olan 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.

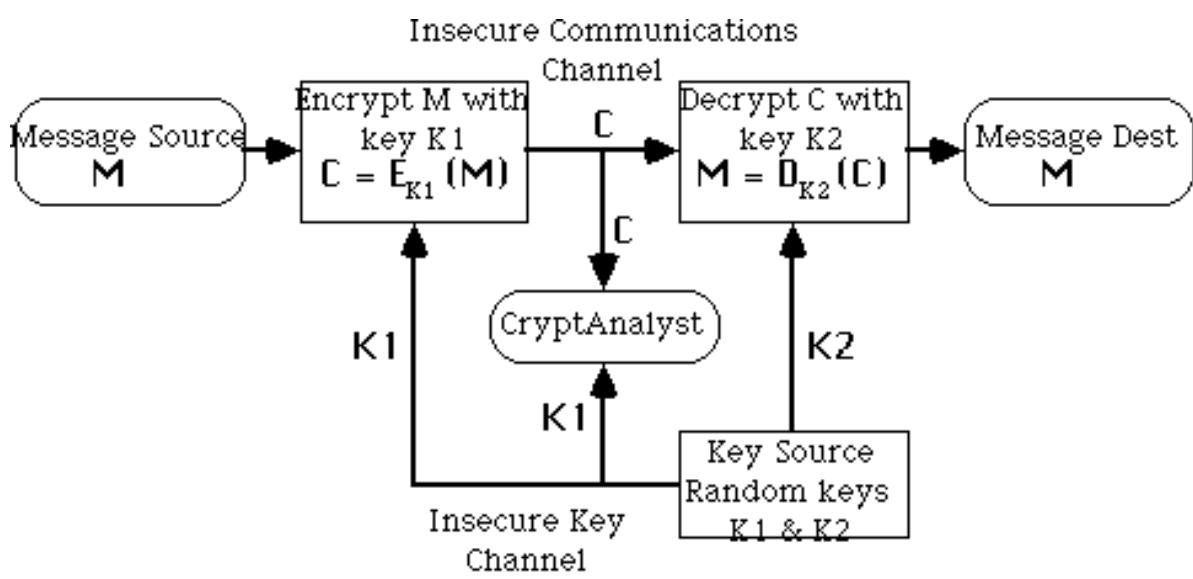
Böylece herbir düğüm en çok $(N-1)$ anahtar saklamak zorunda kalır veya gerektiğinde üretilip kullanılır

7 AÇIK ANAHTARLI KRİPTOSİSTEMLER VE SAYISAL İMZALAR (Public Key Cryptosystems and Digital Signatures)

7.1 Açık anahtarlı (asimetrik) kriptosistemler:

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

Mesaj göndericisi alıcıya ait herkesçe bilinen açık anahtarı kullanarak Açık-anahtarlı kripto algoritmasıyla göndereceği mesajı kapatır ve alıcıya gönderir, mesajın alıcısı ise yalnız kendisinin bildiği gizli anahtar ile deşifreleme algoritmasını kullanarak mesajı açabilir. Gizli anahtar yalnız alıcı tarafından bilindiği için başka birinin bu mesajı açması mümkün olamayacaktır. Gizli anahtarın açık anahtardan polinomsal zamanda türetilmesini imkansız kılmak için Diffie ve Hellman’ın “**tek-yönlü fonksiyon**” mantığı üzerine kurulu **Anahtar değişim protokolü** (Key Exchange Method) vardır .



Asymmetric (Public-Key) Encryption System

Şekil 7.1 Açık-anahtarlı kripto sistemi

Özellikler

- Geleneksel Gizli anahtarlı kriptografi gönderici ve alıcının birlikte paylaştığı tek bir anahtar kullanır.
- Eğer bu anahtar açıklanırsa haberleşme tehlikeye düşer

- **Açık anahtarlı**(veya çift anahtarlı, asimetrik) kriptografi iki anahtar kullanmayı gerektirir:
 - Mesajları şifrelemekte ve imzaları doğrulamakta kullanılan herkes tarafından bilinebilen bir **açık anahtar**
 - Mesajları deşifrelemekte ve imza oluşturmakta kullanılan sadece alıcı tarafından bilinen bir **gizli anahtar**
- Açık anahtar özel(gizli) anahtardan ve şifreleme hakkında diğer bilgilerinden kolaylıkla hesaplanır (Bu bir polinomsal zaman problemidir (P-time))
- Bununla beraber açık anahtarın ve şifrelemenin bilinmesi, gizli anahtar hesaplamak için hala hesaplama bakımından verimsizdir (NP-time problem)
- Böylece açık anahtar, kendisi ile güvenli haberleşmek isteyen herhangi birisine dağıtılabılır. (her ne kadar açık anahtarın güvenli dağıtımını önemsiz olmayan bir anahtar dağıtımını problemidir)
- Açık anahtarlı algoritmaların üç önemli sınıfı vardır.
 - **Açık anahtar Dağıtım Şeması (Public-Key Distribution Schemes PKDS)** burada şema bilginin bir kısmının güvenli olarak değiştirilmesi için kullanılır (değer iki tarafa bağlıdır).
 - **İmza Şeması(Signature Schemes)** Sadece sayısal imza üretmek için kullanılır, burada gizli anahtar imzayı üretmekte , açık anahtar ise doğrulamakta kullanılır
 - **Açık anahtar Şeması(Public Key Schemes PKS)** –şifrelemek için kullanılır, burada açık anahtar mesajları şifreler, gizli anahtar mesajları deşifreler
 - Herhangi bir açık anahtar şeması, gerekli olan oturum anahtarlı mesajı seçmek suretiyle PKDS olarak kullanılabilir.,
 - Çoğu açık anahtar şeması aynı zamanda imza şemasıdır(sağlanan şifreleme&deşifreleme her iki sırada yapılabilir.)

7.1.1 **Diffie-Hellman Açık anahtar Dağıtım Şeması**

- İlk açık anahtar tipi şema PKDS idi ve 1976 da Diffie & Hellman tarafından yayınlandı:
- Bu zamanda mükemmel bir kriptografi üst bakıştır:
- Açık anahtar dağıtım şemasıdır
 - Herhangi bir keyfi mesajı değiştirmek için kullanılmaz
 - Değeri üyelere bağlı olan bir anahtاردır(ve onların açık ve gizli anahtar bilgisi)
- Sonlu bir alanda(Galois) ya bir asal sayının tamsayı modulu veya bir polinomsal alan üzerinde üstelleştirilmesine dayanır.
 - nb üstelleştirme $O((\log n)^3)$ işlem mertebesindedir.
- Güvenliği bu alanlardaki logaritmik hesaplanmanın güçlüğüne dayanır
 - nb ayrik logaritma $O(e^{\log n \log \log n})$ işlem mertebesindedir.
- Diffie-Hellman PKDS aşağıdaki şekilde çalışır.
 - Güvensiz bir iletişim kanalı üzerinden bazı anahtarları değiştirmek isteyen iki A& B olsun, Bunlar;;
 - Büyük bir asal sayı seçerler. p (~200 digit), ve
 - α bir mod p pirimitif elemandır
 - A nin x_A gibi bir gizli sayısı vardır ($x_A < p$)
 - B nin x_B gibi bir gizli sayısı vardır($x_B < p$)
 - A ve B açıklayacakları y_A ve y_B yi sırasıyla hesaplarlar
 - $y_A = \alpha^{x_A} \text{ mod } p$ $y_B = \alpha^{x_B} \text{ mod } p$
 - Sonra anahtar aşağıdaki şekilde hesaplanır
 - $K_{AB} = \alpha^{x_A \cdot x_B} \text{ mod } p$ (ortak Gizli Anahtar)
 - $= y_A^{x_B} \text{ mod } p$ (**B** hesaplayabilir)
 - $= y_B^{x_A} \text{ mod } p$ (**A** hesaplayabilir)

- A ve B arasında güvenli haberleşme için bir gizli anahtarlı şifreleyicide kullanılabilir.
- nb: Eğer iki kişi sonradan haberleşmek isterse kendi açık anahtarlarını değiştirmedikçe aynı gizli anahtara sahip olacaklardır.(genellikle sık olmaz)

7.1.2 RSA Açık anahtarlı Kriptosistem

- En çok bilinen ve en pratik açık anahtarlı tasarım olarak kabul edilen algoritma 1977'de Rivest, Shamir & Adleman tarafından önerildi:
- Mesajları şifrelemek, Anahtar değiştirmek ve sayısal imza oluşturmak için kullanılan bir açık anahtarlı tasarımdır.
- Tamsayı modulo bir sonlu alan(Galois) içinde tamsayı modulo üzerinde üstelleştirmeye dayanır
 - nb üstelleştirme işlemleri $O((\log n)^3)$ mertebesindedir.
- güvenliği,büyük sayıların çarpanlarının hesaplanması zorluğuna bağlıdır.
 - nb faktörizasyon işlemleri $O(e^{\log n \log \log n})$ mertebesindedir.
 - (Ayrık logaritma ile benzerdir.)
- Algoritma Kuzey Amerikaya patentlidir. (Bu nedenle dünyanın başka bir yerinde patentlenemez)
 - Bu yöntemin uygulanmasında yasal zorlukların kaynağıdır
- RSA, modüler aritmetiği kullanarak üstelleştirmeye dayanan bir açık anahtarlı şifreleme algoritmasıdır.
- Yöntemin uygulanması için önce anahtarların üretilmesi gereklidir.
- Her bir kullanıcı tarafından anahtar üretimi aşağıdakileri içerir:
 - Rasgele çok büyük iki asal sayı seçilir(~100 digit), p, q
 - $n = p \cdot q$ hesaplanır
 - $\phi(n) = (p-1) \cdot (q-1)$ hesaplanır.
 - rasgele bir şifreleme anahtarı seçilir öyle ki : $\text{ebob}(\phi(n), e) = 1$; $e < \phi(n)$,
 - deşifreleme anahtarı d hesaplanır: $d = e^{-1} \bmod \phi(n)$, $0 \leq d \leq n$
 - Açık Anahtar: $KA = \{e, n\}$
 - Gizli Anahtar : $KG = \{d, n\}$
- Şifreli metin C'yi elde etmek için M mesajının şifrelenmesi: $C = M^e \bmod n$ $0 \leq d \leq n$
- M Mesajını elde etmek için C şifreli metnin deşifre edilmesi: $M = C^d \bmod n$ dir.
- RSA sistemi aşağıdaki sonuca dayanır:

Eğer $n = pq$ burada p, q farklı büyülükteki asal sayılardır. Buradan,

$$x \phi(n) = 1 \bmod n$$

Bütün x'ler p veya q tarafından bölünemezler

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

7.1.2.1 RSA Örneği

- $p=7$ ve $q= 17$ olan iki asal sayı seçilir.
- $n = p \cdot q = 119$ değeri hesaplanır.
- $\phi(n) = (p-1)(q-1) = 96$ hesaplanır.
- Bir e sayısı seçilir , öyle ki $\phi(n) = 96$ ve $\text{ebob}(\phi(n), e) = 1$; $e < \phi(n)$, buradan $e= 5$ seçilir,
- Öyle bir d sayısı belirlenir ki, $d = 1 \bmod 96$ ve $d < 96$ d için doğru değer $d= 77$ dir.
- Çünkü $77 \cdot 5 = 385 = 4 \cdot 96 + 1$
- Sonuçta anahtarlar ; açık anahtar $KA = \{ 5, 119 \}$; gizli anahtar ; $KG = \{ 77, 119 \}$ olacaktır.
- Şifreleme için ise;
- Açık metin olarak $M= 19$ seçilsin;
- $C = M^e \bmod n$, $19^5 \equiv 66 \bmod 119$ elde edilir. Şifreli metin 66'dır.

- Deşifreleme için;
 $M = C^d \text{ Mod } n$; $66^{77} \equiv 19 \text{ mod } 119$ elde edilir ki açık metinin 19 olduğu sonucuna ulaşılır.

7.1.2.2 RSA'nın Güvenliği

- RSA algoritmasının güvenliği, n 'nin modülünün çarpanlarına ayrılmasının zorluğuna dayanır,
- En iyi bilinen çarpanlarına ayırma algoritması olan (Brent-Pollard) n sayıs üzerindeki en büyük çarpan p ise

$$O\left(\frac{e^{\sqrt{2 \ln p \ln \ln p}}}{\ln p}\right)$$

işlem mertebesindedir. (Talo 9 .13)

Tablo 7.1

n 'deki onlu digit sayısı	n 'nin çarpanlarına ayrılmasındaki işlem(bit) sayısı
20	7200
40	3.11e+06
60	4.63e+08
80	3.72e+10
100	1.97e+12
120	7.69e+13
140	2.35e+15
160	5.92e+16
180	1.26e+18
200	2.36e+19

- Bu 200 digit uzunlığında olan n için 1-100 MIPS lik modern bilgisayar için sayı 10^6 ya bölünerek saniye cinsinden zaman hesaplanır.
 - nb: halen $1e+14$ işlem hesaplama için elverişlilik limiti olarak kabul edilir ve $3e+13$ usec/yıl alır.
- Fakat çoğu bilgisayarlar 32/64 bitten daha büyük sayılar üzerinde doğrudan işlem yapamazlar.
- Bu nedenle büyük sayılar üzerinde işlem yapmak için kütüphaneleri kullanılır.

7.1.2.3 Multi-Precision Arithmetic

- Çoklu kelime(multiple precision) sayılar üzerinde çalışan fonksiyon kütüphanelerini kapsar.
- Klasik referanslar “yarı nümerik algoritmalar” olarak bilinir
 - Dijit dijit çarpma yapılır
 - Kare alma ve çarpma ile üs alma işlemi yapılır
- Bilinen kütüphaneler kullanılıp tekerlek yeniden keşfedilmeye uğraşılmamalıdır.
- Modülo aritmetiği özellikle modülo indirmeler ile özel hünerler kullanabilir.

7.1.2.4 Daha Hızlı modülo İndirgeme

* Chivers (1984), multi-precision aritmetik işlemleri yaparken modülo indirmeleri yapmanın hızlı bir yolunu gösterdi

Bir b tabanlı n karakterli tamsayı $A(a_0, \dots, a_{n-1})$ verilsin tamsayı A aşağıdaki gibi gösterilebilir

$$A = \sum_{i=0}^{n-1} a_i b^i$$

buradan

$$A \equiv \left\{ \sum_{i=0}^{n-2} a_i b^i + a_{n-1} b^{n-1} \pmod{jm} \right\} \pmod{m}$$

yukarıdaki ifade, bir sayının En yüksek Anlamlı Dijitinin çıkartılabilceğini ve kalan dijitelere eklenebilen mod m kalanının asıl sayıya mod m uyumlu olan bir sayıda sonuçlanacağını gösterir.

bir sayıyı indirmek için Chivers algoritması aşağıdaki gibidir:

1. $R = (b^d, 2.b^d, \dots, (b-1).b^d) \pmod{m}$ şeklinde Bir dizi düzene

2. $FOR i = n-1 \text{ to } d \text{ do}$

$WHILE A[i] \neq 0 \text{ do}$

$j = A[i];$

$A[i] = 0;$

$A = A + b^{i-d}.R[j];$

$END WHILE$

$END FOR$

Burada; $A[i]$ A sayısını i inci karakteridir, $R[j]$ R dizisinden j . tamsayı kalandır.

A daki simbol sayısı n , Modüldeki simbol sayısı d 'dir.

7.1.2.5 RSA 'in Hızlandırılması – Değişik Çarpma Teknikleri

- Geleneksel çarpma $O(n^2)$ mertebesinde bit işlemi yapar, daha hızlı teknikler aşağıdakileri içerir:
- Schonhage-Strassen Tamsayı Çarpma Algoritması:
 - Herbir tamsayı bloklara bölünür, ve bir polinomun katsayıları olarak kullanılır.
 - Bu polinomların uygun noktalarda değeri hesaplanır, sonuç değerler çarpılır
 - Çarpım polinomun katsayılarını oluşturmak için bu değerlerin interpolasyonu alınır.
 - Orijinal tamsayının çarpımını bulmak için katsayılar birleştirilir
 - Enterpolasyon fazını hızlandırmak için Ayrık Fourier dönüşümü ve Katlama (konvolüsyon) dönüşümü kullanılır.
 - $O(n \log n)$ bit işleminde çarpma yapılabılır .
- Özel donanım kullanılabilir Çünkü:
 - Elde propagasyon gecikmesi nedeniyle geleneksel aritmetik birimler büyütülemez.
 - Böylece $O(n)$ bit işlemde çarpma yapmak için $O(n)$ kapılı ya paralel elde saklama veya gecikmeli elde-saklama teknikleri kullanılır.
 - Veya, $O(\log n)$ bit işlemde çarpma yapmak için $O(n^2)$ kapılı, paralel-paralel teknikler kullanılır

7.1.2.6 RSA ve Chinese Kalan Teoremi

- RSA için deşifreleme hızında anlamlı bir iyileştirme, sırasıyla modulo p ve modulo q çalıştırılmak için Chinese kalan teoremini kullanarak yapılır.
 - P ve q yarı büyükükte olduğu için, $n = p.q$ nin büyülüüğü yarıdır ve böylece aritmetik çok daha hızlıdır.
- Deşifreleme hesabından, iki denklem üretmek suretiyle RSA'de Chinese kalan teoremi kullanılır

$$M = C^d \pmod{R}$$

Aşağıdaki gibidir:

$$M_1 = M \pmod{p} = (C \pmod{p})^d \pmod{(p-1)}$$

$$M_2 = M \pmod{q} = (C \pmod{q})^d \pmod{(q-1)}$$

Buradan denklem çiftinin

$$M = M_1 \pmod{p} \quad M = M_2 \pmod{q}$$

CRT ile aşağıda verilen tek bir çözümü vardır:

$$M = [(M_2 + q - M_1)u \pmod{q}]^p + M_1$$

Burada $p \cdot u \bmod q = 1$ dir.

7.1.2.7 Pratikte RSA Gerçeklenmesi

- Yazılım ile gerçeklemeler
 - Genellikle 256-512 bit blok uzunlığında 1-10 bits/saniye de icra edilir
 - Gerçeklemenin iki ana şekli:
 - Mikrobilgisayarlarla, hibrid bir algoritmada anahtar değiştirme mekanizmasının parçası olarak.
 - daha büyük makinalarda güvenli bir posta sisteminin elemanları olarak
- Donanım Gerçeklemeleri
 - Genellikle 256-512 bit blok uzunlığında 100-10000 bits/saniye de icra edilir
 - Bütün bilinen gerçeklemeler büyük bit uzunluklu geleneksel Aritmetik Mantık Birimdir

7.1.3 El Gamal

- Diffie-Hellman anahtar dağıtım şemasının mesajlar güvenli değiştirmeyi
- 1985 de ElGamal tarafından geliştirildi
- Diffie-Hellman gibi güvenliği faktör işlemeli logaritmaların zorluğuna dayanır.
- **Anahtar Üretimi**
 - Büyük bir asal sayı seçerler. p (~200 digit), ve
 - α bir mod p primitif elemandır
 - A'nın x_A gibi bir gizli sayısı vardır ($x_A < p$)
 - B'nin x_B gibi bir gizli sayısı vardır ($x_B < p$)
 - A ve B açıklayacakları y_A ve y_B yi sırasıyla hesaplarlar
 - $y_A = \alpha^{x_A} \bmod p$ $y_B = \alpha^{x_B} \bmod p$
- M mesajını C şifreli metni kriptolamak için,
 - Rasgele bir k sayısı seçilir, $0 \leq k \leq p-1$
 - K mesaj anahtarı aşağıdaki şekilde hesaplanır
 - $K = y_B^k \bmod p$
 - Şifreli metin çifti : $C = \{c_1, c_2\}$ aşağıdaki gibi hesaplanır
 - $c_1 = [\alpha]^k \bmod p$ $c_2 = K \cdot M \bmod p$
- Mesajı deşifrelemek için
 - K mesaj anahtarı çıkartılır
 - $K = c_1^{x_B} \bmod p = [\alpha]^{k \cdot x_B} \bmod p$
 - M için aşağıdaki denklem çözülerek M elde edilir:
 - $c_2 = K \cdot M \bmod p$

7.2 Açık anahtarlı Şifreleme sistemlerinde Anahtar Yönetimi

Açık anahtarlı şifrelemenin en önemli özelliklerinden birisi anahtar dağıtımını probleme getirdiği çözümüdür. Bu çözümler;

Açık anahtarların dağıtımını ve gizli anahtarların dağıtımında açık anahtarlı şifrelemedir.

Açık anahtar dağıtımında aşağıda gruplandırılan teknikler önerilmiştir.

Açık duyuru yapılması

Açıkça erişilebilen katalog

Açık-anahtar otoritesi

Açık-anahtar sertifikaları

Açık duyuru

Açık anahtarlı şifrelemenin önemli noktası, herhangi bir iştirakçının kendi açık anahtarını diğer bir iştirakçiye gönderebilmesi veya daha geniş bir gruba yaymayıabilmesidir. Bunun ana zayıf

noktası, herhangi birisinin başkasına ait anahtarını üretip yayımılaması, ve bu kişinin tespit edilinceye kadar kendisini gizleyebildiği sahtekarlıktır.

Açık erişilebilir Katalog

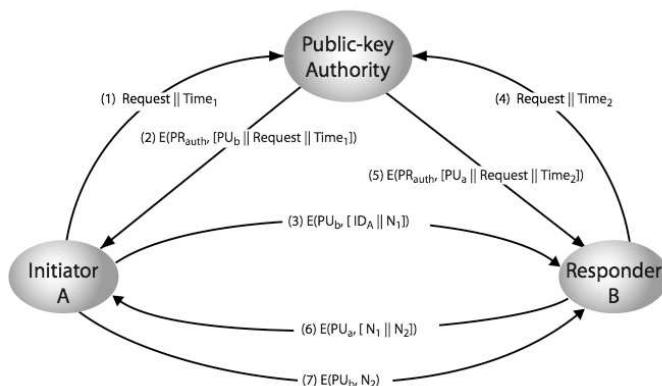
Güvenilgi daha yüksek olan diğer bir yöntem ise, açık anahtarların serbestçe erişilebilen bir katalogda duyurulmasıyla sağlanır. Bu katalogların bakım ve dağıtımları, diğer güvenilir bir kuruluşun sorumluluğunda olabilir. Katalog aşağıdaki özelliklerle güvenlidir.

- {Ad, açık-anahtar} girişlerini içerir
- Üyeler, kataloga güvenli olarak kayıt olurlar
- Üyeler anahtarlarını herhangi bir zamanda değiştirebilirler
- Katalog periyodik olarak yayımlanır
- Kataloga elektronik olarak erişilebilir

Bu yöntem, açık duyuruya göre daha güvenilirdir ancak hala karıştırma veya sahtekarlığa karşı zayıflıkları vardır.

Açık Anahtar Otoritesi

Daha güvenli olarak açık anahtar dağıtımının yapılması, açık anahtarların kataloglardan dağıtımının daha sıkı denetimi ile sağlanabilir. Kullanıcının catalog için açık anahtarını bilmesini ve istediği bir açık anahtarını güvenli olarak alabilmesi için catalog ile gerçek zamanda etkileşime girmesi gereklidir. Bunun için Şekil 7.2'de gösterilen 7 adet mesajlaşma gereklidir



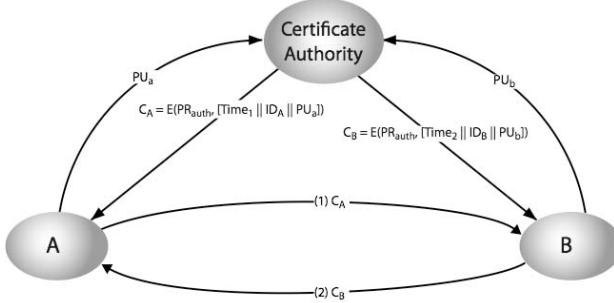
Şekil 7.2. Açık anahtar otoritesi ile anahtar dağıtımını için mesajlaşma

Açık-Anahtar Sertifikaları

Diğer bir geliştirme, anahtarlar bir açık anahtar otoritesinden alınmış olsa da, bir açık anahtar otoritesi ile bağlantı kurmadan anahtar değişimi yapmak için sertifikaların kullanılmasıdır. Bir sertifika, bütün içeriği güvenli açık anahtar veya sertifika otoriteisi tarafından imzalanmış olan açık anahtarın kimlik bilgilerini içeren halidir. Bu, açık anahtar otoritesinin açık anahtarını bilen birisi tarafından doğrulanabilir.

Açık anahtar sertifikaları tarafından onaylanmış olan X.509 standarı uluslararası kabul görmüş bir yöntem. X.509 sertifikaları, IPSEC, SSL, SET, ve S/MIME gibi güvenlik uygulamalarında sıkça kullanılır.

Bu yöntemle anahtar dağıtımının akışı Şekil 7.3'te gösterilmiştir.



Şekil 7.3 : Sertifikalı anahtar dağıtıımı

Gizli Anahtarların açık anahtarlı Dağıtıımı

Açık anahtarlar bir kere dağıtıldığı veya erişilebilir hale geldiği zaman, güvenli haberleşme, gizlice dinleme ve/veya karıştırma saldırısını önlemeyi mümkün kılar. Bununla birlikte birkaç kullanıcı, başarılı nispeten yavaş veri hızları nedeniyle açık-anahtarlı şifreleme sistemlerinin kullanımını haberleşme için pahalı bulabilecektir. Buna bağlı olarak, açık-anahtarlı şifreleme geleneksek şifreleme için gizli anahtar dağıtımı için kullanılabilir. Açık anahtarlı sistemlerin özellikleri aşağıdadır.

- Açık anahtarları elde etmek için önceki yöntemler kullanılabilir
 - Gizlilik ve kimlik doğrulama için kullanılabilir.
 - Fakat açık anahtar algoritmaları yavaştır
 - Bu nedenle mesaj içeriğini şifrelemek için genellikle gizli anahtarlı şifreleme kullanılır
 - Burada oturum anahtarına ihtiyaç vardır
 - Uygun bir oturum sağlamak için birkaç alternatifte sahiptir

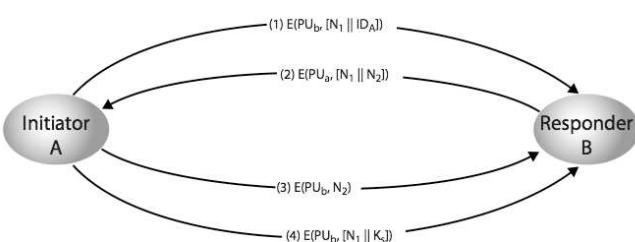
Basit Gizli Anahtar Dağıtımı

Son derece basit bir yöntem 1979 yılında Merkle tarafından önerildi. Fakat bu yöntem, mesajı alıp bir başka mesaj ile değiştirme yapan ortadaki adam saldırılariana karşı güvensiz idi.

Merkle'nin 1979'daki önerisinde;

- Kullanıcı A yeni bir geçici anahtar çifti üretir
 - A kullanıcısı, B'ye, açık anahtarları ve kimlik bilgilerini gönderir
 - B , bir K oturum anahtarları üretir ve sağlanan açık anahtar ile şifreleyerek A'ya gönderir
 - A, oturum anahtarlarını açar ve her ikisi kullanır

Burada problem, bir başkasının araya girmesi ve protokoloün her iki yarısında rol yapmasıdır. Aktif ve pasif saldırılara karşı güvenli bir anahtar değişim yöntemi Şekil 7.4'de gösterilmiştir. Burada, açık anahtarların güvenli olarak değiştirildiği varsayılmıştır. Mesajlar da heri ki taraf birbirlerinin açık anahtarlarını kullanarak haberleşirler.



Sekil 7.4 Gizli anahtarların açık anahtarlar ile güvenli iletişim

7.3 Eliptik Eğri Kriptografi

Günümüzdeki açık anahtarlı kriptografik uygulamalar başlıca 3 ana matematiksel probleme dayandırılarak geliştirilmektedir. Bu alanlar sırasıyla; bir **tamsayının çarpanlarına ayrılmaması problemi, ayrik logaritmik problemi ve eliptik eğrilerde ayrik logaritma problemi** olarak sınıflandırılmaktadır. Son dönemlerde organizasyonlar, güvenlik gereksinimlerini karşılamak üzere, daha yüksek boyutlu anahtarlarla 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 temelde şifreleme, anahtar boyutları ve transmisyon hızlarında büyük gelişmelere olanak sağlamaktadır.

Eliptik Eğri Aritmetiği

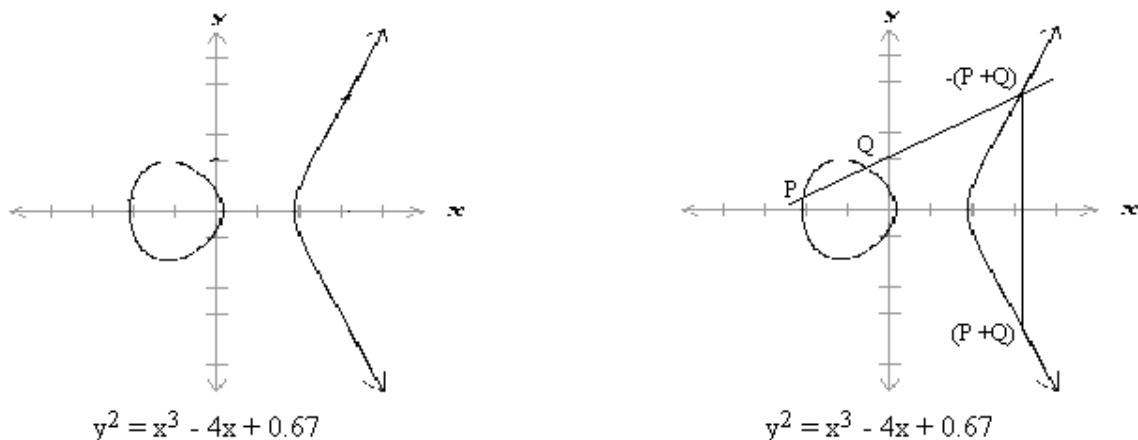
Eliptik Eğri Nedir?

Eliptik eğri çalışmaları matematiğin önemli bir dalıdır. **Eliptik eğriler (x,y) düzleminde yavaşça büükülerek çizilebilen basit fonksiyonlardır.** Fakat eğrinin (x,y) koordinatlarını kestiği noktaları matematikçiler çalışmaya başlayınca ilginç sonuçlar ortaya çıktı. Eliptik eğri kriptografi eliptik eğri kuramının önemli bir uygulamasıdır.

Bir eliptik eğri seçilen belirli a ve b sayıları ile aşağıdaki eşitliği sağlayan (x,y) noktalarının kümesidir.

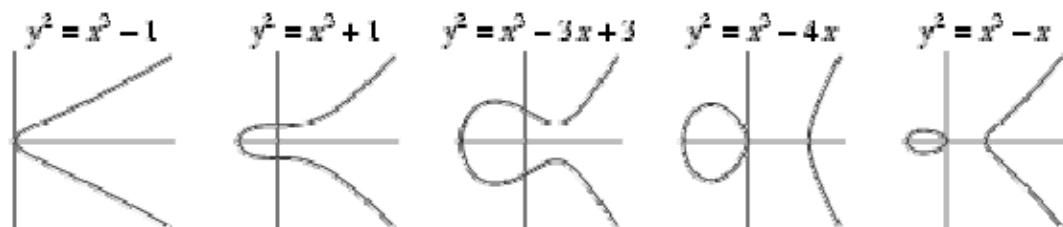
$$y^2 = x^3 + ax + b \quad x, y, a, b \in \mathbb{R}$$

a ve b tipik olarak tamsayıdır, fakat sistem gerçek sayılar ilede çalışabilir. Eliptik ismine rağmen eğri elips şeklinde değildir. Örneğin $a = -4$ ve $b = 0.67$ için eğri denklemi $y^2 = x^3 - 4x + 0.67$. şeklindedir Denklemin sağladığı eğri şekil 7.5'de gösterilebilir.



Şekil 7.5. Eliptik eğri

Değişik a ve b değerleri için elde edilen eliptik eğri şekil 7.6'da gösterilmiştir.



Şekil 7.6. Farklı Eliptik eğriler

Gerçek sayılar üzerinde bir ECC grubu, sonsuzda özel bir nokta(O) ile birlikte eğri üzerindeki noktaları içerir. Eğer $x^3 + ax + b$ ifadesi tekrarlanan faktör içermiyorsa veya eşdeğer olarak eğer, $4a^3 + 27b^2 \neq 0$ ise eliptik eğri bir grup oluşturmak için kullanılabilir. Bir grup basitçe eğri üzerindeki noktaların kümesidir denilebilir. Grup olması nedeniyle, eğri üzerindeki diğer bir noktayı veren noktalar eklemek mümkündür. Graf üzerinde eğriyi P ve Q

noktalarında kesen bir doğru çizerek iki nokta eklenebilir. Yukarıdaki ifadeler basitçe eliptik eğriyi tanımlar. Elijptik eğrinin diğer özellikleri olan **toplama** ve **eğri üzerindeki bir noktanın aynılanması** sonraki bölümde anlatılacaktır.

Sonlu alan (F_p) üzerinde Eliptik Eğriler

Güvenli verinin önemi nedeniyle şifreleme uygulamaları hızlı hesaplama ve tam çözüm gerektirir. Kriptografide eliptik eğrilerin gerçek sayılar üzerinde kullanılması **daha fazla yuvarlatma hatası**, **yavaş hesaplama** ve **hesaplamada artış** getirir. Bu nedenle şifreleme uygulamalarında sonlu alanlar $F(p)$ ve $F(2^m)$ sıkça kullanılır.

Sonlu alan $F(p)$ 'nin kısa açıklaması, kendisinin 0 ve $p-1$ arasında değer alması ile yapılır. Gerçek sayılar kısımındaki kurallar ile aynı şekilde, eğrinin elemanları olan $x, y, a, b, F(p)$ 'nin de elemanları olmalıdır. Eğer $x^3 + ax + b$ nin F_p içinde indirgenemeyen polinom olduğu hatırlanırsa, (eğer $4a^3 + 27b^2 \bmod p \neq 0$ ise) eğrimiz bir grup olarak kullanılabilir. Bu tanımlamalar ile F_p üzerindeki bir eliptik eğri grubu eğri üzerindeki noktalara ilaveten sonsuzdaki noktadan meydana geldiği söylenebilir. Kriptografik hesaplmalarda bazı cebrik kurallar eliptik eğriler için uyarlanabilir.

İki farklı P ve Q noktasının eklenmesi:

$P = (x_p, y_p)$ noktasının negatifi $-P = (x_p, -y_p)$ 'ye eşitti.

İki noktanın toplamı olan $P + Q = R$ hesaplamak için aşağıdaki denklem çifti kullanılarak eğrideki yeni noktanın koordinatları hesaplanır.

$$X_r = [\lambda^2 - x_p - x_q] \bmod p$$

$$y_r = [-y_p + \lambda(x_p - x_r)] \bmod p$$

burada $\lambda = (y_p - y_q) / (x_p - x_q)$ iki noktanın eğimidir.

Bir noktanın Aynılanması

$P = (x_p, y_p)$ noktasının aynılanması için aşağıdaki denklem çifti kullanılır.

$$X_r = [\lambda^2 - 2x_p] \bmod p$$

$$y_r = [-y_p + \lambda(x_p - x_r)] \bmod p$$

burada $\lambda = (x_p^2 - a) / (2y_p)$ eğimdir ve a 'da eğri parametresidir.

Eğri üzerindeki bir noktayı bulmak ve bu noktayı aynılayarak neticede sonsuzdaki (O) noktasının bulmak için, nokta sonsuzdaki noktaya ulaşınca kadar kendisine eklenir, eklenme sayısına noktanın derecesi denir. Kriptografik uygulamalarda, şifreleme süreci için global açık parameter olan temel nokta yüksek dereceden bir nokta olarak seçilir.

Elijptik Eğri Kriptografi (ECC)

1985 'de Neal Koblitz ve Victor Miller tarafından bulunan sonlu alanlar üzerinde eliptik eğrilerdeki ayrik logaritma denetlenemez görünümündedir. Elijptik eğri ayrik 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.
 - Elijptik eğri ayrik logaritma problemi, verilen P ve Q için x değerinin bulunmasıdır.

Eliptik eğri Anahtar Değişimi:

Eliptik eğri kullanarak anahtar değişimi aşağıdaki şekilde gerçekleşir.

Öncelikle, asal sayı olacak şekilde bir $p \approx 2^m$ (kriptografi pratiği için $m > 150$, $m = 180$) (Sonlu alan $F(2^m)$ için) ve eliptik eğri denklemi için a , b 'yi seçmek gereklidir. Bu $E_p(a,b)$ noktalarının bir eliptik grubunu tanımlar. Sonra, $E_p(a,b)$ içinde bir üretici noktası olan $G = (x_1, y_1)$ seçilir. G 'nin seçiminde önemli noktası, en küçük n değerinde $nG = O$ çok büyük asal sayı olmalıdır. $E_p(a,b)$ ve G parametreleri bütün üyeleri tarafından bilinirler. Seçilen bu parametreler ile anahtar değişimi aşağıdaki şekilde yapılır.;

$A, n_A < n$. olacak şekilde bir tamsayı seçer. Sonra $P_A = n_A \times G$ 'yi hesaplar

$$n_A = A \text{ 'nın gizli anahtarı}$$

$$P_A = A \text{ 'nın açık anahtarı}$$

Aynı yöntem ile, $B, n_B < n$. olacak şekilde bir tamsayı seçer. Sonra $P_B = n_B \times G$ 'yi hesaplar

$$n_B = B \text{ 'nın gizli anahtarı}$$

$$P_B = B \text{ 'nın açık anahtarı}$$

Sonra her kullanıcı için sistemin ana gizli anahtarı üretilir.

$$K = n_A \times P_B \Rightarrow \text{kullanıcı } A \text{ için}$$

$$K = n_B \times P_A \Rightarrow \text{kullanıcı } B \text{ için}$$

Basitçe gösterilebilir ki;

$$K = n_A \times P_B = n_A \times (n_B \times G) = n_B \times (n_A \times G) = n_B \times P_A \text{ dir}$$

Bu algoritmada ayrık logatırma problemi(kriptanaliz) verilen G ve $K \cdot G$ ile K değerini çözmektir ki oldukça zordur.

Örnek olarak $p=211$, $E_p(0,-4)$ ile $y^2 = x^3 - 4$ denklemine eşittir. ve $G=(2,2)$ dir. Birisi $241G = O$ hesaplayabilir. A 'nın özel anahtarı $n_A = 121$, böylece A 'nın açık anahtarı $P_A = 121(2,2) = (115,48)$ dir. B 'nın özel anahtarı $n_B = 203$, böylece B 'nın açık anahtarı $P_B = 203(2,2) = (130,203)$ dir. Paylaşılan anahtar ise $121(130,203) = 203(115,48) = (161,169)$ dir.

Ana gizli anahtarın x,y koordinatlarından ibaret olan iki değere sahip olduğu görülür. Geleneksel kriptografide tek bir K değeri(ya x yada y değeri) kullanılabaktır.

Eliptik Eğri ile Şifreleme/Deşifreleme:

Literatürde eliptik eğrileri kullanarak değişik mesaj şifreleme/deşifreleme yaklaşımı mevcuttur. Burada en basit olan yaklaşım ele alınacaktır. Yaklaşımından diğer ElGamal yöntemidir.

- Anahtar değişim sistemindeki gibi, bir şifreleme/deşifreleme sistemi de parametre olarak bir G noktası ve bir $E_q(a, b)$ eliptik grup gerektirir.
- Her A ve B kullanıcıı birer gizli anahtar n_A ve n_B seçer ve $P_A = n_A \times G$ ve $P_B = n_B \times G$ yi açık anahtarları olarak hesaplarlar.
- Bir P_m mesajının şifrelenmesi için, gönderici (bu örnekte kullanıcı A) rastgele bir k tamsayıyı seçer ve C_m şifreli metini iki parça olarak aşağıdaki şekilde üretir;
- $C_m = \{k \cdot G, P_m + k \cdot P_B\}$
- Burada gönderici şifrelemek için alıcının açık anahtarını kullanır.
- Deşifreleme işleminde alıcı $k \cdot G$ 'yi kendi gizli anahtarı ile çarpar ve şifreli metinden çıkartır. İşlem aşağıdaki eşitlikte gösterilmiştir;

$$P_m + k \cdot P_B - n_B \cdot (k \cdot G) = P_m + n_B \cdot k \cdot G - n_B \cdot k \cdot G = P_m$$

Anahtar değişimi örneğinde görüldüğü gibi ayrık logaritma problemi buradada k 'nın verilen G ve kG den elde edilmesi ile aynıdır.

Örnek olarak; $p=751$; $E_p(-1, 188)$, alınırsa eğri $y^2 = x^3 - x + 188$ olur; ve $G=(0,376)$. A 'nın B 'ye, $P_m(562,201)$ eliptik noktasında şifreli bir mesaj göndereceğini ve A 'nın $k=386$ seçtiğini farzedelim. B 'nin açık anahtarı $P_B=(201,5)$ dir. Buradan $386(0,376) = (676,558)$ ve $(562,201) + 386(201,5) = (385,328)$ elede edilir. Böylece A şifreli metin olarak $\{(676,558), (385,328)\}$ gönderilir.

Eliptik Eğri Şifrelemenin Güvenliği

Günümüzdeki kriptografi algoritmalarında, açık anahtarlı kriptosistemler genellikle simetrik anahtarlı kriptosistemler için anahtar iletiminde kullanılır. Son yıllarda organizasyonların 3DES'ten AES'e yöneldikleri görülür. Bu organizasyonlar da 1024 bit RSA genişçe kullanılırlar. Gelişmeler 2048 bit gibi daha uzun anahtar kullanımına yönelmeyi gerektirmektedir. Bu ise daha fazla bellek, maliyet ve hesaplama artışı demektir. Daha fazla güvenlik nedeniyle anahtar uzunluğunun artırılmasına karşı olarak eliptik eğri şifreleme kullanılarak anahtar değişimi ve şifrelemede bu problem giderilebilir. Tablo 7.2'de RSAve ECC 'nin farklı simetri şifreleme algoritmaları ile kullanıldığındaki anahtar uzunluğu bakımından karşılaştırılması görülmektedir.

Tabloyu incelediğimizde Aynı güvenlik seviyelerinde ECC'nin RSA' e göre anahtar uzunluğunun daha küçük olmasıyla daha az bellek gerektirdiği görülür.

Anahtar Uzunluğu Simetrik şifreleme Algoritması	RSA	ECC
80	1536	160
112	4096	224
128	6000	256
160	10000	320

Tablo 7.2. Şifreleme algoritmalarının karşılaştırılmaları

Geleneksel ve açık anahtarlı şifreleme yöntemlerinin karşılaştırılması tablo 7.3'te gösterilmiştir.

Geleneksel kriptolama(Gizli Anahtarlı)	Açık anahtarlı Kriptolama
Çalışma gereksinimi	Çalışma gereksinimi
Aynı algoritma aynı anahtar ile birlikte şifreleme ve deşifreleme ile birlikte kullanılır .	Bir algoritma ve iki anahtardan birisi şifreleme diğer ise deşifreleme için kullanılır.
Gönderici ve alıcı algoritma ve anahtarı paylaşır.	Gönderici ve alıcının her biri, uygun anahtara sahip olmalıdır.
Güvenlik Gereksinimi	Güvenlik Gereksinimi
Anahtarın gizliliği korunmalıdır	İki anahtardan birisi gizlidir.
Başka bir bilgi gerektirmeden mesajı çözmek mümkün veya çok kolay olmalıdır.	Başka bir bilgi olmadan mesajın çözülmesi mümkün değil veya, çok kolay olmamalıdır.
Algoritma ve şifreli metin bilgisi anahtarları tahmin etmede yeterli olamamalıdır	Algoritma, şifreli metin bilgisi ve anahtarın birisinin elde edilmesi diğer anahtarları tahmin etmek için yeterli olmamalıdır.

Algoritma	Enc/Dec	Sayısal İmza	Anahtar Değişimi
RSA	evet, Büyük bloklar için pratik değil	Evet	Evet
LUC	evet, Büyük bloklar için pratik değil	Evet	Evet
DSS	Hayır	Evet	Hayır
Diffie-Hellman	Hayır	Hayır	Evet

Tablo 7.3. Simetrik ve asimetrik şifreleme özellikleri

7.4 Mesaj Doğrulama ve Özetleme Fonksiyonları (Hashing Functions)

Mesaj Doğrulama

Buraya kadar mesaj içeriğinin şifrelenmesiyle korunması üzerinde duruldu. Bu bölümde göndericinin doğrulanması yanında mesaj içeriğinin bütünlüğünün(değiştirmelere karşı) nasıl korunacağı üzerinde durulacaktır. Genellikle mesaj bütünlüğünün korunması elektronik ticaret uygulamalarında gizlilikten daha önde gelen bir husustur. Mesaj doğrulama şu kavramları içerir. Mesaj bütünlüğünün korunması, göndericinin kimliğinin geçerliliği ve mesaj kaynağının kendisini inkar edememesi(non repudation). Bir doğrulayıcı üretmek için kullanılabilen üç adet fonksiyon vardır. Bunlar;

Mesaj şifreleme :(Şifrelenen mesaj onun doğrulanması görevini yapar)

Mesaj doğrulama kodu(MAC): (Bir fonksiyon ile bir anahtar ile sabit uzunluklu olarak üretilen değer doğrulama için kullanılır)

Hash(özet) fonksiyonu: (Herhangi uzunluktaki bir mesajdan açık bir fonksiyon ile üretilen sabit uzunluktaki özet değer doğrulama için kullanılır.)

Güvenlik gereksinimleri

Bi ağdaki haberleşmenin içeriğinde aşağıda listelenen saldırılar tanınabilir.

İlk iki gereksinim mesaj gizliliği içerisinde değerlendirilir ve açıklanan şifreleme yöntemleriyle sağlanır. Diğer gereksinimler mesaj doğrulama içerisinde kalır. Bu noktada, kaynağından gelen mesajın değiştirilmemiş olması önemlidir. Aynı zamanda adres dizisi ve zamanlamadır. Sayısal imzanın kullanımı kaynağın inkar edilmesi ile ilgilidir.

- Mesaj içeriğini açıklama (Disclosure)
- Ağ trafiginin analizi(traffic analysis)
- Gerçeği gizleme(masquerade)
- Mesaj içeriğini değiştirme(content modification)
- Mesaj sırasını değiştirme(sequence modification)
- Mesaj zamanlamasını değiştirme(timing modification)
- Kaynağın inkar edilmesi(source repudiation)
- Varışın inkar edilmesi(destination repudiation)

Mesaj şifreleme

Mesaj şifrelemenin kendisi, doğrulama işlevini sağlayabilir. Burada ,mesajın bütünlüğünün şifrelenmesi, sadece uygun anahtarları bilenlerin mesajı şifreleyebilmesi nedeniyle onun doğrulayıcısı olabilir. Böylece geçerli olan mesaj anlaşılabilir.(mesajın uygun yapıda olması veya değişikliğe karşı denetim bilgisi(checksum) bulunması).

➤ Eğer simetrik şifreleme kullanılmış ise:

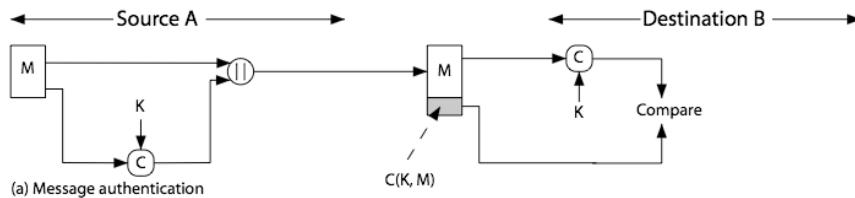
- Alıcı, mesajın gönderici tarafından oluşturulduğunu bilir.
- Kullanılan anahtarı sadece gönderici ve alıcı bilir
- Eğer mesaj, denetim bilgisi içeren uygun yapıda ise, mesajın içeriği değiştirilemez

Açık anahtar teknikleri ile, sadece anahtar sahibi tarafından üretilenbilien, sayısal imzalar kullanılabilir. Fakat mesajın sonunda iki açık anahtar işlemi gerekir.

Mesaj Doğrulama Kodu(MAC)

Diger bir doğrulama teknigi, bir gizli anahtar ile sabit uzunlukta üretilen ve kriptografik control verisi veya Mesaj doğrulama kodu olarak bilinen ve mesajın sonuna eklenen bir veri bloğu ile yapılır. Bu teknikte, A ve B olarak adlandırılan iki haberleşme grubu bir K ortak anahtarını paylaşır. Bir MAC fonksiyonu ,şifrelmeye benzeyen ve desifrelemede olduğu gibi ters çevrilme gerektirmez. Bu sonuç mesajın sonuna eklenir.

- Alıcı mesaj üzerinde aynı hesaplamayı yapar ve sonucu MAC ile karşılaştırır.
- Böylece göndericiden gelen mesajın değiştirilmediğini garanti eder. (Şekil 7. 7)



Şekil 7.7 : Mesaj doğrulama kodunun çalışması

Doğrulama ve gizliliği birlikte sağlamak için MAC şifreleme ile birleştirilebilir. Sadece doğrulama gereklisi ise MAC kullanılır.

Gönderici ve alıcının her ikisi de anahtarları paylaştığı ve üretebildiği için MAC sayısal imza değildir.

MAC özellikleri

Bir C fonksiyonu tarafından üretilen bir MAC aynı zamanda kriptografik denetim bilgisidir. MAC kaynakta mesajın sonuna eklenir ve varışta yeniden hesaplanarak doğrulama yapılır.

MAC fonksiyonu, birçok farklı uzunluktaki mesajı aynı uzunluktaki özet değere dönüştürdüğü için çoktan bire bir fonksiyondur.

- Bir MAC kriptografik denetim değeridir(checksum)

$$\text{MAC} = \text{CK}(M)$$
 - Değişken uzunluktaki M mesajını bir gizli K anahtarı kullanarak sabit uzunluktaki bir doğrulayıcıya şeklinde sıkıştırır
- CK bir çoktan bire fonksiyondur, bu fonksiyon ile potansiyel olarak birçok mesaj aynı MAC değerine sahip olabilir fakat bu sonucu elde etmek çok zordur.

MAC gereksinimleri

Bir MAC fonksiyonunun güvenliğini değerlendirmek için ona karşı olabilecek saldıruları düşünmek gereklidir. Bundan sonra listelenen gereksinimleri sağlamak gereklidir.

İlk gereksinim, saldırganın anahtarı bilmese dahi, verilen MAC ile uyusan bir başka mesajı oluşturduğu mesaj yerine koyma saldırısı ile ilgilidir.

İkinci gereksinim, seçilen bir şifresiz metin tabanlı brute-force saldırısını önlemeye ilgilidir.

Son gereksinim, doğrulama algoritmasının, mesajın belirli bir parçasının diğerlerine göre daha zayıf olmamasını diktetmektedir. Özetle;

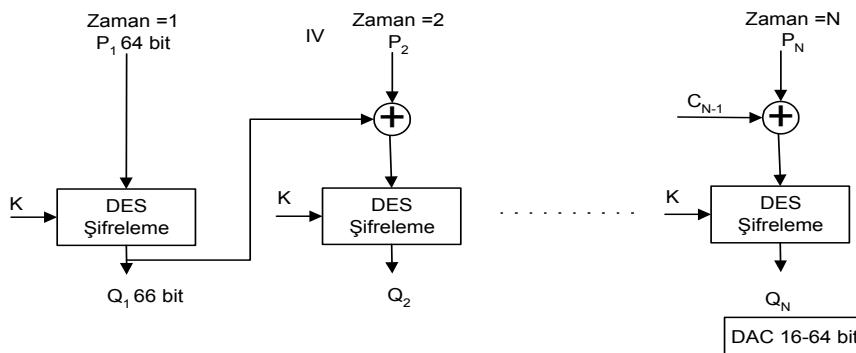
- Saldırı çeşitlerini dikkate alır
- MAC aşağıdakileri sağlamalıdır:
 1. Bilinen bir mesaj ve MAC için aynı MAC'a sahip bir başka mesaj bulunması verimsiz olmalıdır.
 2. MAC, düzenli dağıtılmış olmalıdır
 3. MAC, mesajın bütün bitlerine bağlı olmalıdır

DES'teki block cipher chaining modu, son bloğu göndermek suretiyle ayrı bir doğrulayıcı üretilmesinde kullanılabilir. Bu, DES-CBC tabanlı veri doğrulama algoritması(DAA) ile yapılır.(Şekil 7.8.)

Algoritma, DES'in şifre blok zinciri(CBC) çalışma modu olarak bir sıfır başlangıç vektörü ile tanımlanabilir. Yetkilendirilecek veri, 64 bitlik bloklar şeklinde grupperdir. D₁, D₂, D_n. Eğer gerekliyse, son blok, 64 bit elde etmek için sağına sıfır koyularak düzenlenir. DES kriptolama algoritması kullanılarak, E, ve gizli anahtar K, bir veri yetkilendirme kodu(DAC) aşağıdaki şekilde hesaplanır.

- $O_1 = Ek(D_1)$
 - $O_2 = Ek(D_2 \oplus O_1)$
 - $O_3 = Ek(D_3 \oplus O_2)$
 -
 - $O_n = Ek(D_n \oplus O_{n-1})$

DAC, ya bütün bloğu içerir yada, en soldaki M biti $16 \leq M \leq 64$. olarak içerir



Şekil 7.8 : DES CBC modunun MAC üretim için kullanılması

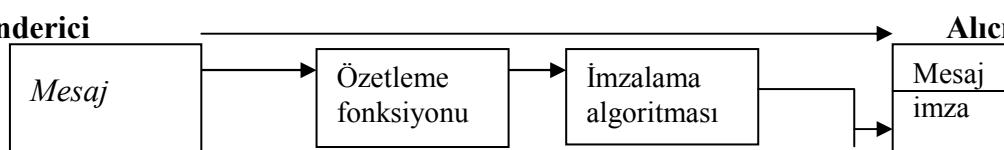
Hash Fonksiyonları

Temel kriptografik terimlerden biri “Kriptografik özetleme fonksiyonu” veya diğer adıyla tek yönlü özetleme fonksiyonu (one-way hash function) dur.

Tanım : Değişik uzunluktaki bit dizilerini sabit uzunluklu bit dizilerine taşıyan polinomsal zamanda kolay hesaplanabilen fonksiyona “Özetleme fonksiyonu” denir. Görüntü kümelerindeki sabit uzunluklu oluşan bu bit dizisine ise “Özet-değer” (Hash-value) adı verilir.

Kriptografik olarak Öztleme fonksiyonları değişken m uzunluklu mesajları sabit n uzunluklu mesajlara indirmek amacıyla kullanılır ($m > n$). Seçilen h öztleme fonksiyonu iki farklı m_1 ve m_2 mesajlarını aynı özet değerine taşımamalı ($h(m_1) \neq h(m_2)$) ve verilen bir y özet-değerinden bu değere ait m mesajı polinomsal zamanda hesaplanamamalıdır.

Özetleme fonksiyonlarının kriptografide kullanımı daha çok sayısal imza ve veri bütünlüğünün korunması alanlarında yaygındır. Sayısal imza uygulamalarında uzun mesajlar öncelikle bilinen bir Özetleme fonksiyonu ile sabit uzunluklu kısa bir diziye özetlenmeli ve bu özet-değer imzaya girmelidir.



Şekil 7. 9 Şifrelenmemiş bir mesajın bütünlüğünün ve doğruluğunun korunması için özetlenip imzalanması.

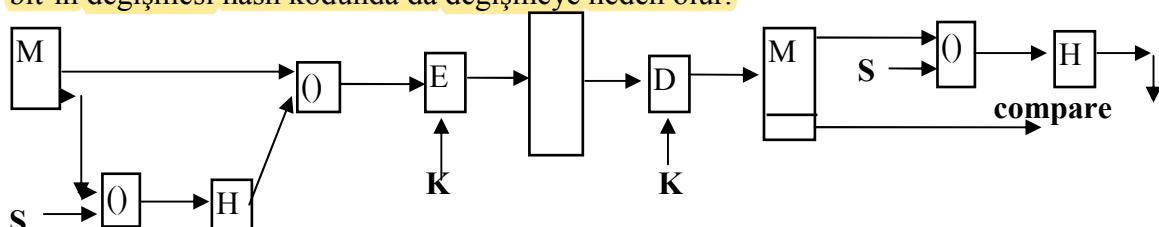
Özetleme fonksiyonu bilginin bütünlüğünü sağlamak için de kullanılabilmektedir. Fonksiyonun tanım kümesindeki her mesajı görüntü kümesinde farklı bir özet-değerine taşıması gerektiği düşünülürse içeriği değiştirilmiş bir mesajın özet-değeri de farklı olacaktır. Dolayısıyla mesajı alan kişi mesajı kendisinin de bildiği Özetleme fonksiyonundan geçirecek ve elde ettiği özet-değerle kendisine gönderilen özet-değeri karşılaştıracaktır. Mesajın bütünlüğünün korunduğunun

ispati için bu iki değerin uyuşması gerekmektedir. Virüs koruması ve yazılım korsanlığının önlenmesi Öztleme fonksiyonlarının diğer özel uygulamalarındandır.

Öztleme fonksiyonları yukarıda da belirtildiği gibi herkesçe bilinebilen ve gizli anahtar olmayan tek-yönlü fonksiyon uygulamalarıdır. Eğer belli bir mesajın değiştirilip değiştirilmemiğini tespit etmeye yönelik kullanılırlarsa “Değişiklik tespit kodları” (Modification Detection Codes) adını alırlar. Bu alanla ilgili olarak gizli bir anahtar içeren ve veri bütünlüğünün yanı sıra verinin kaynağının doğrulanması işleminde de kullanılan Öztleme fonksiyonlarına “Mesaj doğrulama kodları” (Message Authentication Codes) adı verilir.

Hash Fonksiyonu

Mesaj yetkilendirme kodunun bir çeşidi, tek yönlü hash fonksiyonu olarak çok sık kullanılır. Mesaj yetkilendirme kodu olarak, bir hash fonksiyonu, değişken uzunluklu M mesajını giriş olarak alır ve çıkış olarak sabit uzunluklu, mesaj özeti denilen $H(M)$ hash kodu üretir. Hash kodu, mesajın bütün bitlerinin bir fonksiyonudur ve hata bulma özelliği vardır. Mesajdaki bir veya birkaç bit'in değişmesi hash kodunda da değişmeye neden olur.



Şekil 7.10 Hash Fonksiyonunun temel Kullanımı: Yetkilendirme ve gizlilik sağlar.

Öztleme fonksiyonunun amacı, bir dosya, mesaj veya diğer bir veri bloğunun parmak izini üretmektir.

Güvenli öztleme(hash) fonksiyonlarının özellikleri aşağıda verilmiştir. Esas olarak iki mesaj için aynı özet değerini bulmak çok zor olmalı ve özet açık bir şekilde mesajla ilişkili olmamalıdır.(mesajın karmaşık doğrusal olmayan bir fonksiyonu olmalıdır). Öztleme fonksiyonları ve blok şifreleyicilerin tasarımları arasında birçok benzerlik bulunur.

- H herhangi boyutlu bir M mesajına uygulanabilir
- H sabit uzunlukta bir çıkış(h) üretir
- $H(M)$, verilen bir M mesajı için kolay üretilebilir
- Verilen bir h değeri için, $H(x)=h$ yi sağlayan x 'i bulmak hesaplama bakımından verimsizdir. Bu $H(x)$ 'in tekyönlü özelliği
- Verilen bir x bloğu için $H(y)=H(x)$ olan $y \neq x$ gibi bir y değerini bulmak hesaplama bakımından verimsizdir. Bu özellik “weak collision resistance” dır.
- $H(y)=H(x)$ için, bir (x,y) çifti bulmak hesaplama bakımından verimsizdir. Bu özellik “strong collision resistance” dır.

Basit Hash Fonksiyonları

Bütün hash fonksiyonları aşağıdaki genel prensipleri kullanarak çalışır. Giriş, n bitlik blok dizisi olarak görülür. Giriş her defasında bir blok olarak n-bitlik hash fonksiyonunu üretmek için işlenir. En basit hash fonksiyonu, her bloğun bit bit XOR işlemne tabi tutulmasıdır. Bu aşağıdaki gibi açıklanır:

$$C_i = b_{i1} \oplus b_{i2} \oplus \dots \oplus b_{im}$$

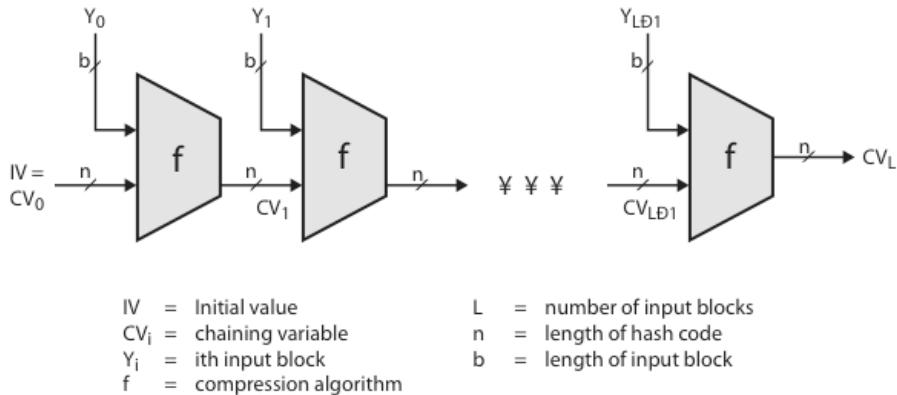
C_i : hash kodunun i. biti, $1 \leq i \leq n$; b_{ij} : j.'inci bloktaki I'inci Bit

Genel hash algoritmasının blok şeması Şekil 7.11'de verilmiştir.

$$CV_0 = IV$$

$$CV_i = f(CV_{i-1}, Y_{i-1}) \quad 1 \leq i \leq L$$

$$H(M) = CV_L \quad (CV : \text{Chaining Value})$$



Şekil 7.11. Öztleme fonksiyonu genel yapısı

Bugün sıkılıkla kullanılan Öztleme fonksiyonlarına örnek olarak 1992'de Ron Rivest tarafından geliştirilmiş **MD5** (message-digest algorithm) algoritması gösterilebilir. NIST ve NSA'nın ortaklaşa geliştirmiş olduğu **SHA** (secure hash algorithm) ise yine NIST'in Öztleme fonksiyonları için belirlediği standart olan SHS (secure hash standard) içerisinde yer almış ve kullanımı yaygın olan bir öztleme algoritmasıdır. Önemli öztleme fonksiyonlarının özellikleri Tablo 7. 3'te verilmiştir.

Adı	Blok Uz.(bit)	Max mesaj	Sonuç(bit)	Adım sayısı	Mantık F.	Ek sabit
MD5	512	∞	128	64(4x16)	4	64
SHA1	512	$2^{64} - 1$	160	80(4x20)	4	4
RipeMD-160	512	$2^{64} - 1$	160	160(5x16 çift)	5	4

Tablo 7.3 Önemli Öztleme fonksiyonlarının özellikleri

7.5 Kimlik Doğrulama ve Sayısal İmzalar

Bir belgenin elektronik ortamda imzalanmasındaki amaçlar aşağıdaki şekilde özetlenebilir.

- İmzayı atan şahsin kimliğinin imzadan anlaşılması
- Şahıs imzayı reddettiği durumda bunun ispatının yapılabilmesi
- İmzanın sahtesinin atılamaması, aksi durumda ispatının yapılabilmesi
- İmza tarihinin bilinebilmesi(imza içerişine koyulması)
- Belgenin içeriğinin değiştirilme riskine karşı imzanın metin ile ilişkilendirilmesi
- Eğer belge içeriğinin üçüncü şahıslar tarafından bilinmesi istenmiyor ise belge ayrıca şifrelerek ilettilir ve saklanır.

Bunların yanında sayısal imzanın elde edilmesi ve kime ait olduğunu anlaşılması kolay olmalıdır.

Sayısal İmza ve El ile Atılan İmzanın Karşılaştırılması

Geleneksel el ile atılan imzanın, standart yöntemi olmaması(bazları ismini yazdığı halde bazıları anlaşılmaz çizgiler çizerler) nedeniyle imzanın doğrulanması işlemi oldukça zordur. Diğer handikap, el ile atılan imzanın kolaylıkla taklit ve kopya edilebilmesidir. Bir başka dezavantaj ise her bir sayfasının imzalanması gereken çok sayfalı dökümanlarda, her sayfanın imzalandığının kontrol edilmesi gerekliliğidir. Bunlara karşı sayısal imzanın aşağıdaki avantajlarının olduğu kolaylıkla söylenebilir.

Sayısal imza uygulamasıyla;

- İmzalanan verinin bütünlüğü (değiştirilmemesi) sağlanır.
- İmza atacak şahsin bu yetkiye sahip olup olmadığı(yetkilendirme) sağlanır.
- İmza atanın bu imzayı inkar edememesi sağlanır.,
- İmzanın atıldığı tarih-saat damgasının olması(imzanın ne zaman atıldığı bilinmesi sağlanır)
- Hız ve verimlilik sağlanır.
- İstenirse gizlilik sağlanır.

Ağ üzerinde hareket eden verilerin geldikleri adresten tam olarak gönderildiği şekliyle gidecekleri yere ulaşmaları amacıyla veriler ve paketler değişik şekillerde özetlenirler. Bu şekilde kullanılan özetleme fonksiyonları (Hash Functions), verileri tek yönlü bir matematiksel fonksiyona tabi tutup özet değeri oluştururlar. Bu özet değer paket içerisinde yollanır. En çok kullanılan özetleme fonksiyonları SHA (Secure Hash Algorithm) ve MD5 algoritmalarıdır. **Başka bir yolda veri paketinin sonuna her paket için bir tane üretilen CRC (Cyclic Redundancy Check) kodu veya toplam kontrol bilgisinin (CS) eklenmesidir.**

Verilerin ağ üzerinde doğrulanmasının dışında ayrıca gerçekten üzerinde yazılı olan adresten gelip gelmediğinin kontrolü ise sayısal imza algoritmaları ile sağlanmaktadır. Sayısal imza algoritmalarında, veriyi gönderen adresin kendisine ait gizli bir anahtarla verinin kendisi, imzalama algoritmasına girer ve çıkan bilgi bize o pakete ait imzayı vermektedir. Paketin alıcısı ise, imzanın doğrulanması aşaması için, imzanın doğruluğunu kontrol eder. Sayısal imza algoritmaları olarak, aynı zamanda açık anahtarlı kripto sistem algoritması amacıyla kullanılan RSA ve ElGamal algoritmaları kullanılabilir. Bunlara ek olarak DSA (Digital Signature Algorit) imza algoritması ve pek çok özel tür algoritmada kullanılabilmektedir.

Ağ ve internet güvenliğinde bugün yaygın olarak kullanılan kimlik doğrulama uygulamaları arasında Kerberos protokolünü sayabiliriz. **Kerberos** daha çok gizli anahtarlı şifreleme üzerine kurulu bir sistem olmakla beraber istemci sunucu arasındaki diyalogların doğrulanmış bir şekilde yapılması işlemini yönetir.

X.509 dizin doğrulama servisi ise X.500 dizin servisinin kullanımını ile yaygınlaşan bir kimlik doğrulama standardıdır. X.509 dizin servisi daha çok açık anahtarlı kripto sistemler üzerine kuruludur. Bu serviste hiyerarşik bir düzende yer alan kullanıcılar arasındaki haberleşmelerde servisin kullanıcılarına sağladığı sertifikalar söz konusudur. Her sertifika, verinin göndereceği adresin kimlik numarası ve ona ait açık anahtarı içermektedir. Sertifikaların içeriğinin değiştirilmesinin engellenmesi amacıyla sertifikalar belli bir sertifikasyon otoritesi sunucu tarafından imzalanır. Dolayısıyla veri gönderecek adresteki birim alıcıya ait sertifika, sertifika otoritesinden (CA - Certification Authority) elde eder, altındaki imzayı kontrol eder. Sertifikanın doğruluğundan emin olduktan sonra içeriğinde yer alan açık anahtarı kullanarak alıcıya mesajı şifreleyerek gönderir.

Sayısal imzalar ve İmza Algoritmalarının özellikleri

- Gizli anahtar imza üretirken açık anahtar imzaları doğrulamakta kullanılır
- Sadece sahibi sayısal imza üretebilir buradan mesajı kimin ürettiğini doğrulamakta kullanılır
- Genellikle mesajın tamamı imzalanmaz(değişen bilgi iki katına çıkar), fakat mesajın özeti imzalanır,
- Bir özet fonksiyonu mesajı alır ve mesaja bağlı olan sabit uzunluklu(tipik olarak 64 to 512 bit) bir değer üretir

- Başka bir mesajın aynı özet değerini üretmesi çok zor olmalıdır(aksi halde bazı sahtecilik mümkün olur)

Sayısal imza için gereksinimler:

- İmza, imzalanacak mesaja bağlı olan bir sayısal bit paterni olmalıdır.
- İmza, sahteciliği ve inkarı önlemek için göndericiye özel bilgileri taşımalıdır.
- Sayısal imzayı üretmek kolay olmalıdır.
- Sayısal imzayı tanıtmak ve doğrulamak kolay olmalıdır.
- Verilen bir sayısal imza için bir mesaj üretmek veya, verilen bir mesaj için sayısal imza üretmek, hesaplanabilirlik açısından verimsiz olmalıdır.
- Sayısal imza bellekte kaybedilmemelidir.