

# MUH441 Bilişimde Güvenlik – 1

Prof. Dr. Hasan Hüseyin BALIK  
(2. Hafta)

# İçerik

- 2.Bilgisayar Güvenliği Teknolojisi ve İlkeleri
  - 2.1 Şifreleme Araçları
  - 2.2.Kullanıcı doğrulama
  - 2.3 Giriş kontrolü
  - 2.4 Veritabanı ve Veri Merkezi Güvenliği
  - 2.5 Kötü amaçlı yazılımlar
  - 2.6 Hizmet Reddi Saldırıları
  - 2.7 İzinsiz giriş tespiti
  - 2.8 Güvenlik Duvarları ve Saldırı Önleme Sistemleri

## 2.1 Şifreleme Araçları

# 2.1.İçerik

- Simetrik Şifreleme ile Gizlilik
- Mesaj Doğrulama ve Hash Fonksiyonları
- Açık Anahtar Şifreleme
- Dijital İmzalar ve Anahtar Yönetimi
- Rastgele ve Sözde Rastgele Sayılar

# Simetrik Şifreleme

- İletilen veya depolanan veriler için gizlilik sağlama amacıyla kullanılan evrensel bir tekniktir
- Geleneksel şifreleme veya tek anahtarlı şifreleme olarak da adlandırılır
- Güvenli kullanım için iki gereksinim:
  - Güçlü bir şifreleme algoritmasına ihtiyacınız vardır.
  - Gönderici ve alıcı gizli anahtarın kopyalarını güvenli bir şekilde almış olmalı ve anahtarı güvende tutmalıdır.

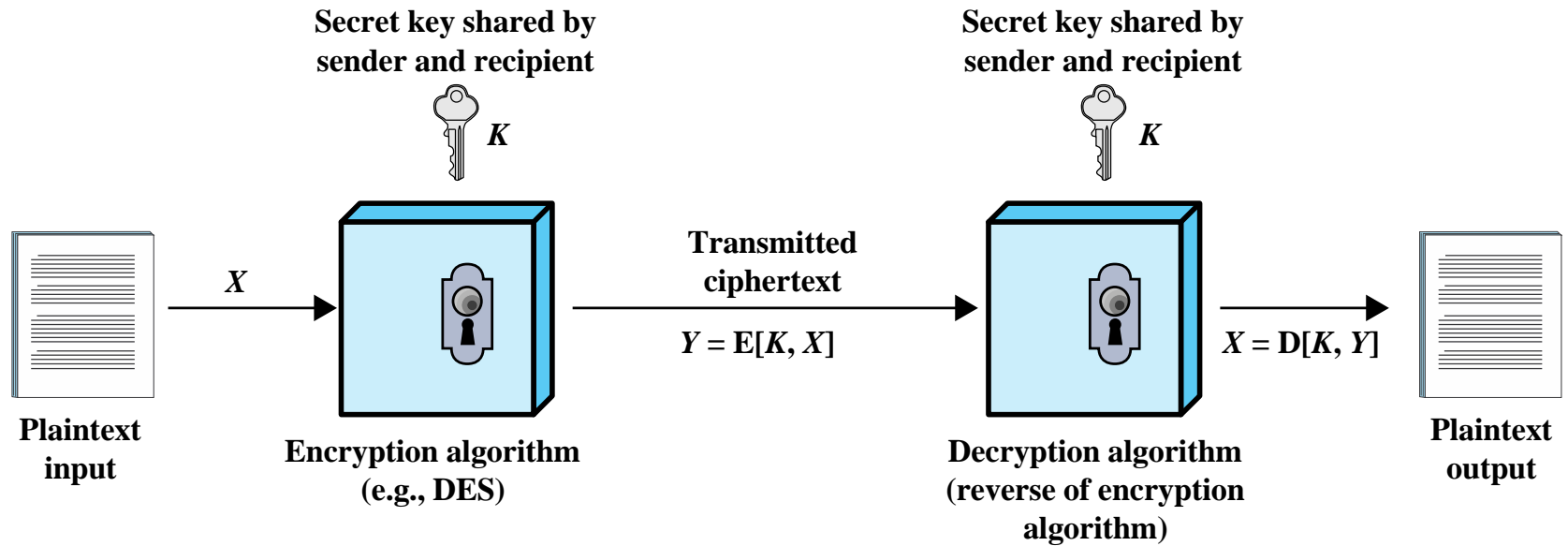


Figure 2.1 Simplified Model of Symmetric Encryption

# Simetrik Şifrelemeye Saldırı

## Kriptanalitik Saldırıları

- .... dayanır:
  - Algoritmanın doğası
  - Düz metnin genel özellikleri hakkında biraz bilgi
  - Bazı örnek düz metin-şifreli metin çiftleri
- Belirli bir düz metin veya kullanılan anahtarı çıkarmaya çalışmak için algoritmanın özelliklerinden yararlanır
  - Başarılı olursa, bu anahtarla şifrelenmiş tüm gelecekteki ve geçmiş iletiler tehlikeye girer

## Kaba Kuvvet Saldırıları

- şifreli metin üzerinden düz metne anlaşılır bir çeviri elde edilene kadar tüm ihtimaller denenir
  - Başarıya ulaşmak için tüm olası anahtarların ortalama yarısı denenmelidir

# Veri Şifreleme Standardı (DES)



- Yakın zamana kadar en yaygın kullanılan şifreleme şemasıydı
  - NIST tarafından 1977 FIPS PUB 46'da uyarlanmıştır
  - Veri Şifreleme Algoritması (DEA) olarak anılır
  - 64 bit şifreli metin bloğu oluşturmak için 64 bit düz metin bloğu ve 56 bit anahtar kullanır
- Güç Endişeleri:
  - Algoritmanın kendisiyle ilgili endişeler.  
DES, var olan en çok çalışılan şifreleme algoritmasıdır  
Şimdiye kadar ölümcül bir hata bildirilmedi.
  - 56 bitlik anahtarın kullanımıyla ilgili endişeler  
Ticari kullanıma hazır işlemcilerin hızı, bu anahtar uzunluğunu ne yazık ki yetersiz kılıyor.



# Üçlü DES (3DES)

- DES'in ömrü, üçlü DES kullanımıyla uzatıldı.
- İki veya üç benzersiz anahtar kullanarak temel DES algoritmasını üç kez tekrarlar
- İlk olarak 1985'te ANSI standardı X9.17'de finansal uygulamalarda kullanım için standartlaştırılmıştır.
- Avantajları:
  - 168-bit anahtar uzunluğu, DES'nin kaba kuvvet saldırısına karşı güvenlik açığının üstesinden gelir
  - Temel şifreleme algoritması DES ile aynıdır
- Dezavantajları:
  - Algoritma yazılımda yavaş
  - 64 bit blok boyutu kullanır

# Gelişmiş Şifreleme Standardı (AES)

3DES için bir yedek  
gerekiyordu

3DES, uzun süreli  
kullanım için  
makul değildi

NIST, 1997'de yeni  
bir AES için teklif  
çağrısında bulundu

3DES'e eşit veya daha iyi  
bir güvenlik gücüne  
sahip olmalıdır

Önemli ölçüde  
geliştirilmiş verimlilik

Simetrik blok şifreleme

128 bit veri ve  
128/192/256 bit  
anahtarlar

Kasım 2001'de  
seçilen Rijndael

İlk değerlendirme  
turunda önerilen 15  
algoritma kabul edildi

İkinci tur, alanı 5  
algoritmaya daralttı

FIPS 197 olarak  
yayınlandı

# Üç Popüler Simetrik Şifreleme Algoritmasının Karşılaştırılması

	DES	Üçlü DES	AES
Düz metin blok boyutu (bit)	64	64	128
Şifreli metin blok boyutu (bit)	64	64	128
Anahtar boyutu (bit)	56	112 veya 168	128, 192 veya 256

DES = Veri Şifreleme Standardı

AES = Gelişmiş Şifreleme Standardı

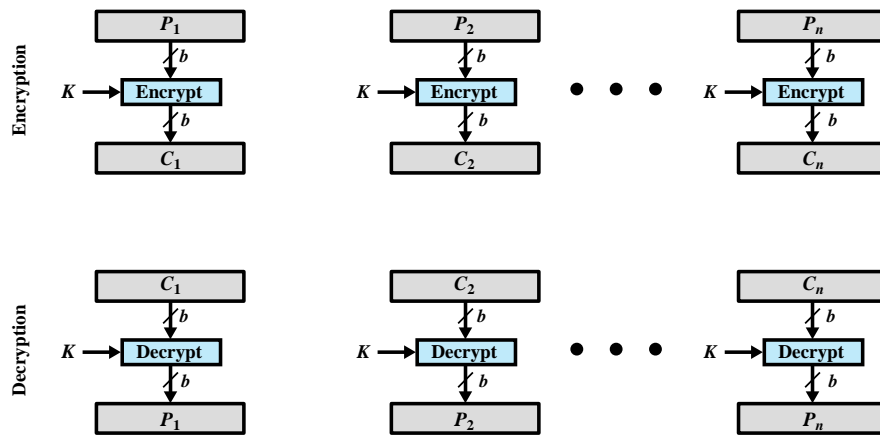
# Kapsamlı Anahtar Arama için Gerekli Ortalama Süre

Anahtar boyutu (bit)	şifre	Alternatif Anahtar Sayısı	$10^9$ şifre çözme/sn'de Gerekli Süre	$10^{13}$ şifre çözme/sn'de Gerekli Süre
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	255 ns = 1.125 yıl	1 saat
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2127 ns = $5,3 \times 10^{21}$ yıl	$5.3 \times 10^{17}$ yıl
168	Üçlü DES	$2^{168} \approx 3.7 \times 10^{50}$	2167 ns = $5,8 \times 10^{33}$ yıl	$5.8 \times 10^{29}$ yıl
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2191 ns = $9.8 \times 10^{40}$ yıl	$9.8 \times 10^{36}$ yıl
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2255 ns = $1.8 \times 10^{60}$ yıl	$1.8 \times 10^{56}$ yıl

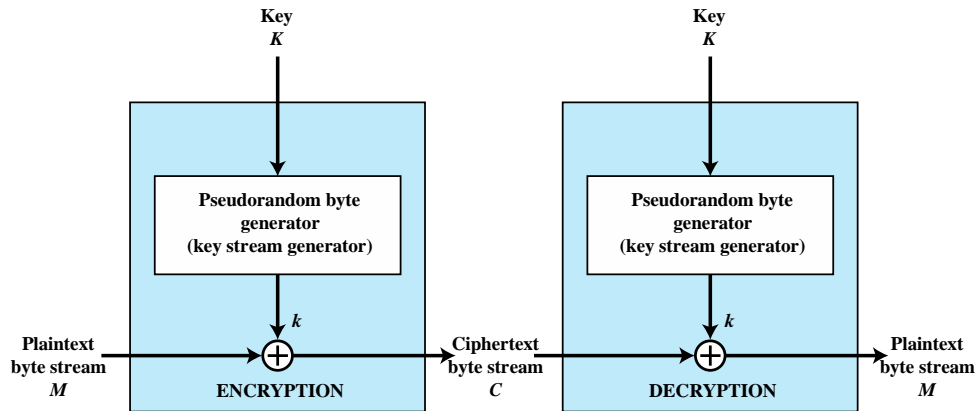
Bu tablo, çeşitli anahtar boyutları için kaba kuvvet saldırısı için ne kadar zaman gerektiğini gösterir.

# Pratik Güvenlik Sorunları

- Tipik olarak simetrik şifreleme, tek bir 64-bit veya 128-bit bloktan daha büyük bir veri birimine uygulanır.
- Elektronik kod kitabı (ECB) modu, çok bloklu şifrelemeye en basit yaklaşımdır
  - Her düz metin bloğu aynı anahtar kullanılarak şifrelenir
  - Kriptanalistler, düz metindeki düzenliliklerden yararlanabilir
- Operasyon modları
  - Büyük diziler için simetrik blok şifrelemenin güvenliğini artırmak için geliştirilmiş alternatif teknikler
  - ECB'nin zayıflıklarının üstesinden gelir



(a) Block cipher encryption (electronic codebook mode)



(b) Stream encryption

Figure 2.2 Types of Symmetric Encryption

# Blok ve Kesintisiz Şifreleme

## Blok Şifreleme

- Girdiyi bir seferde bir eleman bloęu işler
- Her giriş bloęu için bir çıkış bloęu üretir
- Anahtarları yeniden kullanabilir
- Daha yaygın

## Kesintisiz şifreleme

- Giriş öğelerini sürekli olarak işler
- Her seferinde bir eleman çıktı üretir
- Birincil avantaj, neredeyse her zaman daha hızlı olmaları ve çok daha az kod kullanmalarıdır.
- Düz metni her seferinde bir bayt şifreler
- Sözde akış, giriş anahtarı bilgisi olmadan öngörülemeyen akıştır.

# Mesaj Kimliđi Doğrulama

Aktif saldırılara  
karşı korur

Alınan mesajın  
gerçek olduğunu  
doğrular

Geleneksel  
şifrelemeyi  
kullanabilir

- İçerik deđiştirilmedi
- Orijinal kaynaktan
- Zamanında ve doğru sırayla

- Yalnızca gönderen ve alıcı bir anahtarı paylaşır



# Gizlilik Olmadan Mesaj Doğrulama

- İleti şifreleme tek başına güvenli bir kimlik doğrulama biçimi sağlamaz
- Bir mesajı ve onun kimlik doğrulama etiketini şifreleyerek kimlik doğrulama ve gizliliği tek bir algorithmada birleştirmek mümkündür.
- Tipik olarak mesaj doğrulama, mesaj şifrelemeden ayrı bir fonksiyon olarak sağlanır.
- Gizlilik olmadan mesaj doğrulamanın tercih edilebileceği durumlar şunlardır:
  - Aynı mesajın birkaç hedefe yayınlandığı birkaç uygulama vardır.
  - Bir tarafın ağır bir yüke sahip olduğu ve gelen tüm mesajların şifresini çözecek zamanı karşılayamadığı bir değişim.
  - Düz metin olarak bir bilgisayar programının kimlik doğrulaması çekici bir hizmettir
- Böylece güvenlik gereksinimlerini karşılamada hem kimlik doğrulama hem de şifreleme için yer vardır.

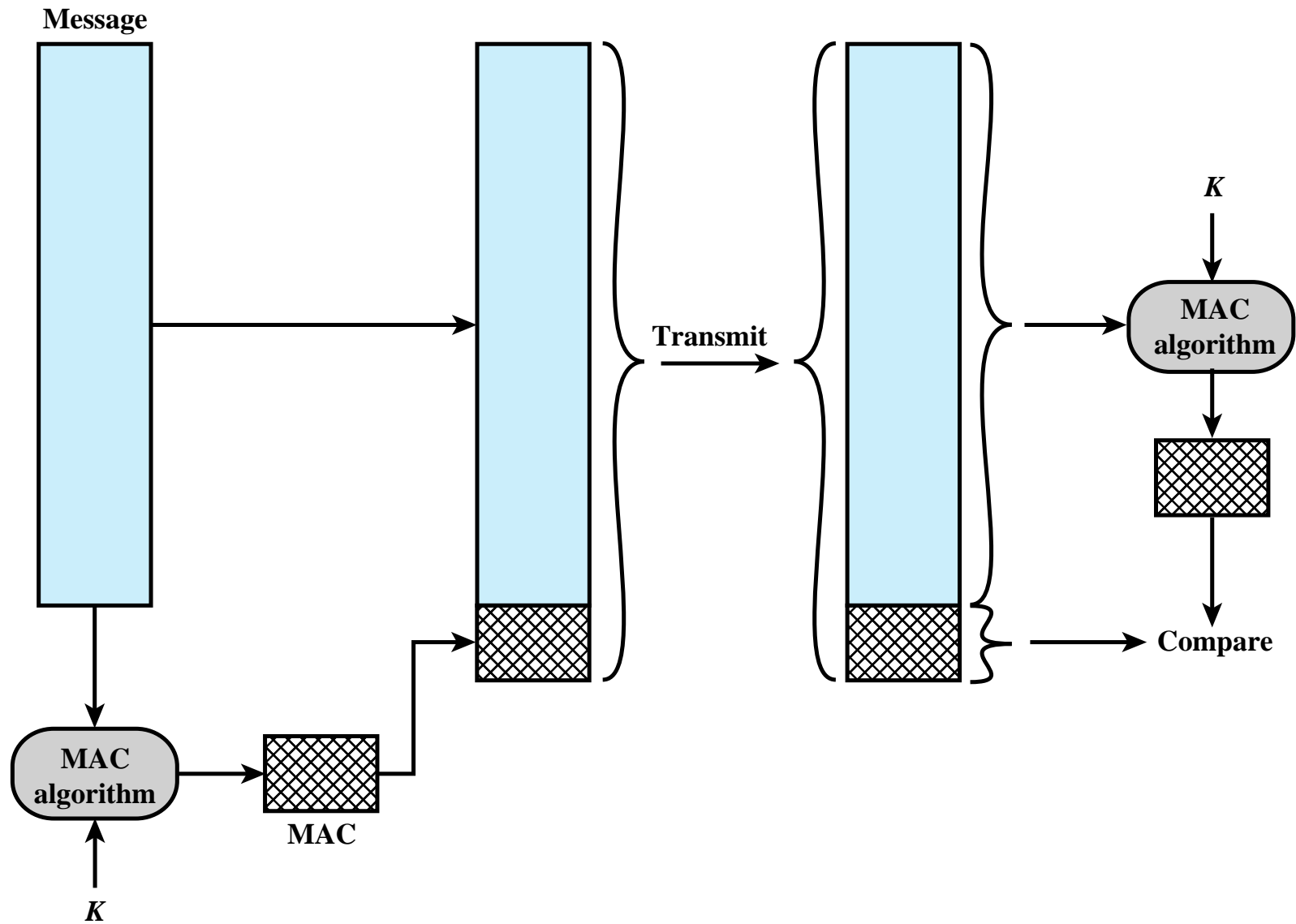
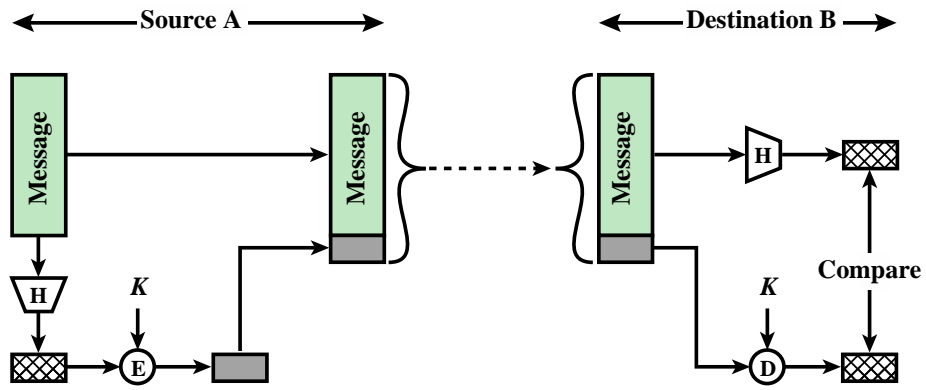
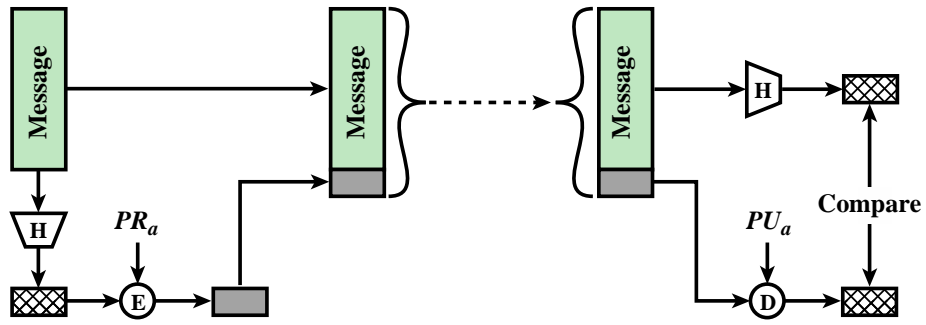


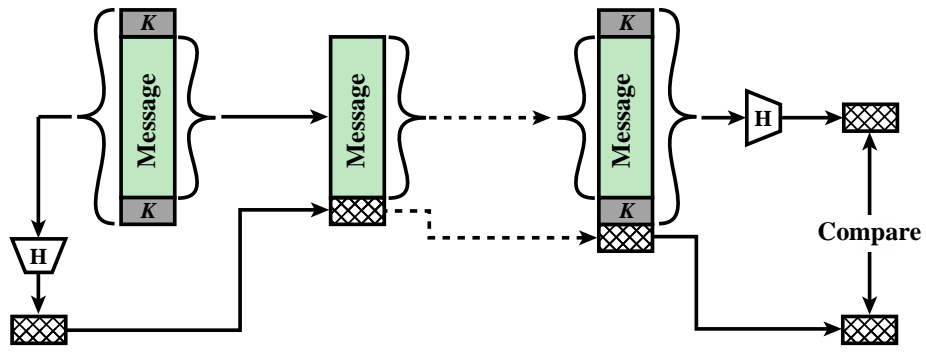
Figure 2.3 Message Authentication Using a Message Authentication Code (MAC).



(a) Using symmetric encryption



(b) Using public-key encryption



(c) Using secret value

Figure 2.5 Message Authentication Using a One-Way Hash Function.

# Mesaj doğrulamada kullanılabilmesi için, bir hash fonksiyonu $H$ aşağıdaki özelliklere sahip olmalıdır:

Herhangi bir boyuttaki bir veri bloğuna uygulanabilir

Sabit uzunlukta bir çıktı üretir

$H(x)$  herhangi bir  $x$  için hesaplanması nispeten kolaydır

Tek yönlü.  $H(x) = h$  olacak şekilde  $x$ 'i bulmak hesaplama açısından mümkün değil

$H(y) = H(x)$  olacak şekilde  $y \neq x$ 'i bulmak hesaplama açısından mümkün değil

Çarpışmaya dayanıklı veya güçlü çarpışma direnci.  $H(x) = H(y)$  olacak şekilde herhangi bir  $(x,y)$  çifti bulmak hesaplama açısından mümkün değil

# Hash Fonksiyonlarının Güvenliđi

Güvenli bir H fonksiyonuna saldırmak için iki yaklaşım vardır.:

## Kriptoanaliz

- Algoritmadaki mantıksal zayıflıklardan yararlanır

## Kaba kuvvet saldırısı

- Hash fonksiyonunun gücü sadece algoritma tarafından üretilen hash kodunun uzunluđuna bađlıdır.

SHA en yaygın kullanılan hash algoritması

SHA, NIST tarafından geliştirildi ve 1993'te yayınlandı

Ek güvenli hash fonksiyonu uygulamaları:

## Parolalar

- Bir parolanın hash'i bir işletim sistemi tarafından saklanır

## İzinsiz giriş tespiti

- Her dosya için  $H(F)$ 'yi bir sistemde saklayın ve hash değerlerini güvenceye alın

# Açık Anahtar Şifreleme Yapısı

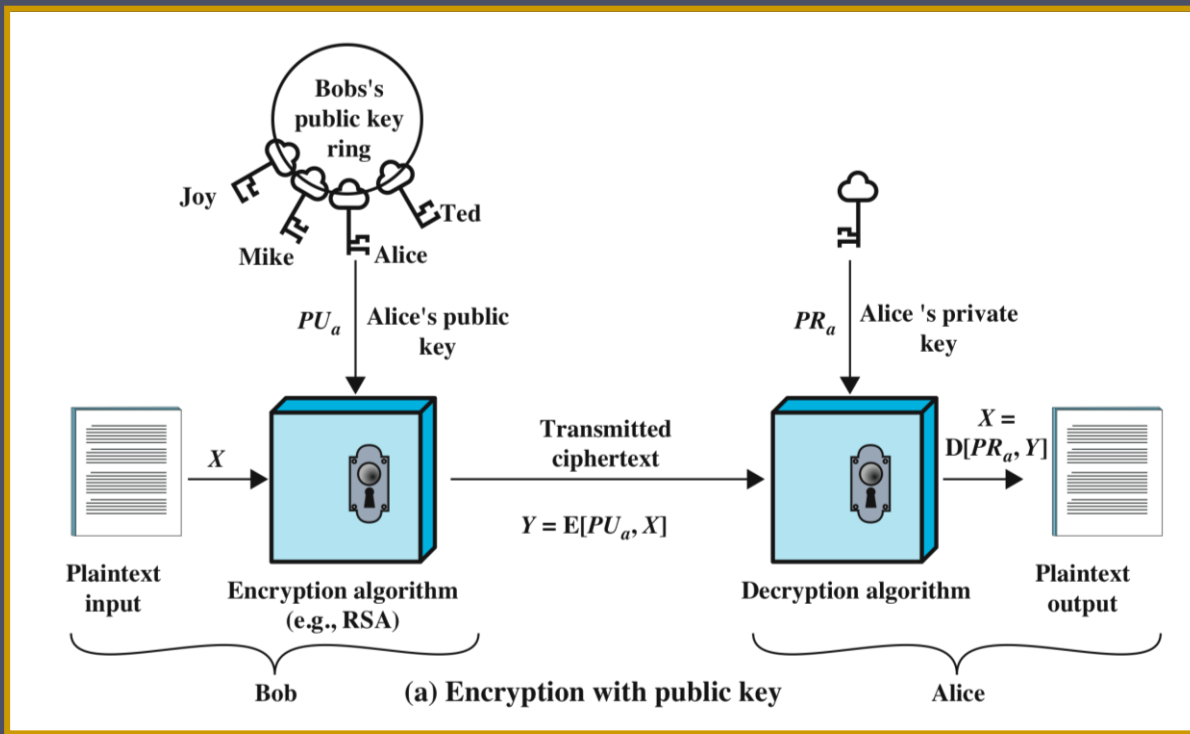
Diffie ve Hellman tarafından 1976'da halka açık olarak önerilmiştir.

Matematiksel fonksiyonlara dayalı

Asymmetric

- İki ayrı anahtar kullanır
- Genel anahtar ve özel anahtar
- Genel anahtar, başkalarının kullanması için herkese açık hale getirilir

Dağıtım için bir tür protokol gereklidir



- **düz metin**

- Girdi olarak algoritmaya beslenen okunabilir mesaj veya veriler

- **Şifreleme algoritması**

- Düz metin üzerinde dönüşümler gerçekleştirir

- **Genel ve özel anahtar**

- Bir çift anahtar, biri şifreleme için, biri şifre çözme için

- **şifreli metin**

- Çıktı olarak üretilen şifreli mesaj

- **şifre çözme algoritması**

- Orijinal düz metni üretir

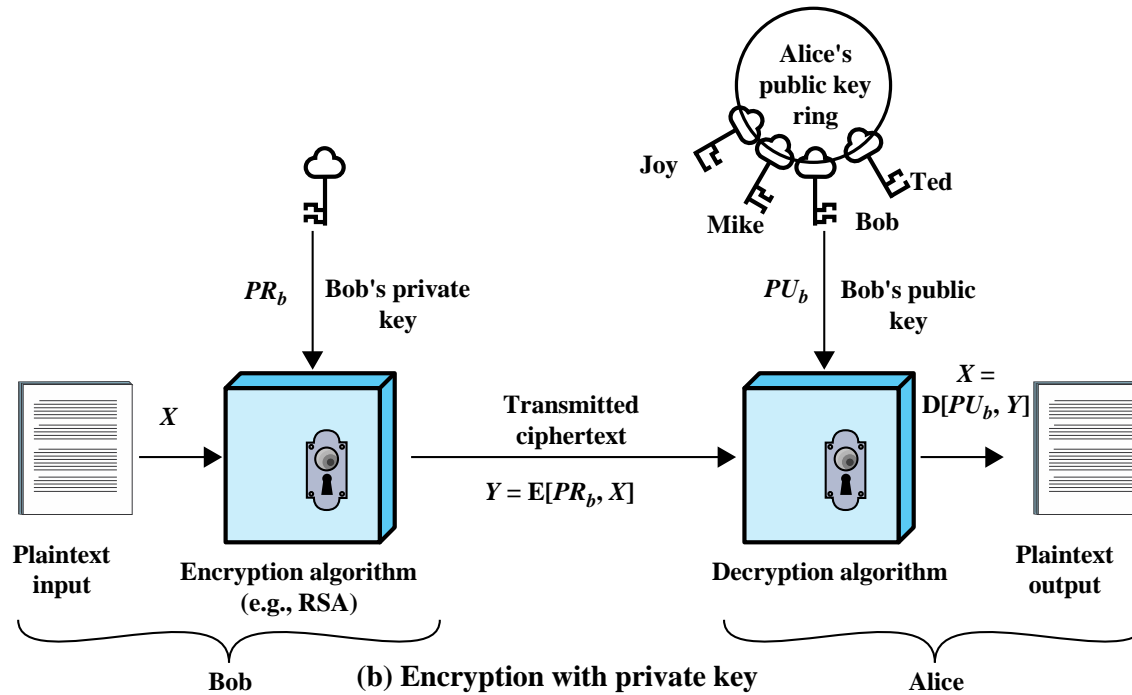


Figure 2.6 Public-Key Cryptography

- Kullanıcı, kendi özel anahtarını kullanarak verileri şifreler
- Karşılık gelen ortak anahtarı bilen herkes mesajın şifresini çözebilir.



# Asimetrik Şifreleme Algoritmaları

**RSA (Rivest,  
Shamir, Adleman)**

1977'de geliştirildi

Açık anahtar şifrelemeye  
yönelik en yaygın kabul  
gören ve uygulanan  
yaklaşım

Bazı n için düz metin ve  
şifreli metnin 0 ile n-1  
arasında tam sayılar  
olduğu blok şifre.

**Diffie-Hellman  
anahtar değişim  
algoritması**

İki kullanıcının, mesajların  
daha sonra simetrik olarak  
şifrelenmesi için gizli  
anahtar olarak  
kullanılabilecek paylaşılan  
bir sır hakkında güvenli bir  
şekilde anlaşmaya  
varmasını sağlar

Anahtar değişimi ile  
sınırlıdır

**Dijital İmza  
Standardı(DSS)**

SHA-1 ile yalnızca dijital  
imza işlevi sağlar

Şifreleme veya anahtar  
değişimi için kullanılamaz

**Eliptik eğri  
kriptografisi(ECC)**

RSA gibi güvenlik, ancak  
çok daha küçük  
anahtarlarla

# Açık Anahtarlı Kriptosistemler için Uygulamalar

Algoritma	Elektronik imza	Simetrik Anahtar Dağıtımı	Gizli Anahtarların Şifrelenmesi
RSA	Evet	Evet	Evet
Diffie-Hellman	Hayır	Evet	Hayır
DSS	Evet	Hayır	Hayır
Eliptik Eğri	Evet	Evet	Evet

# Açık Anahtarlı Şifreleme Sistemleri için Gereksinimler

Anahtar çiftleri oluşturmak için hesaplama açısından kolay

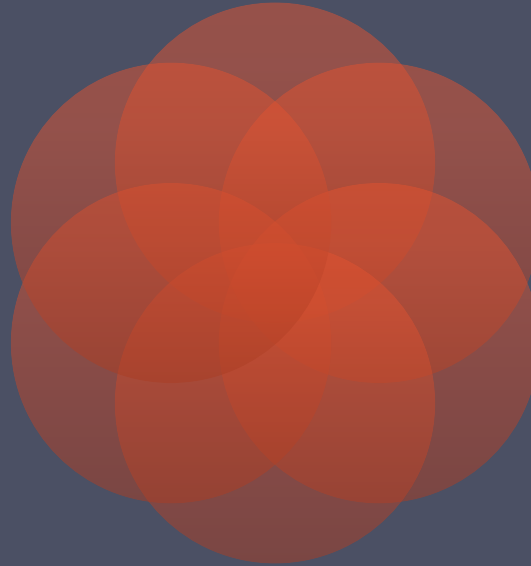
Her bir rol için anahtarlardan biri kullanılabilirse kullanışlıdır

Mesajları şifrelemek için ortak anahtarı bilen gönderici için hesaplama açısından kolay

Saldırganın orijinal mesajı kurtarması için hesaplama açısından mümkün değil

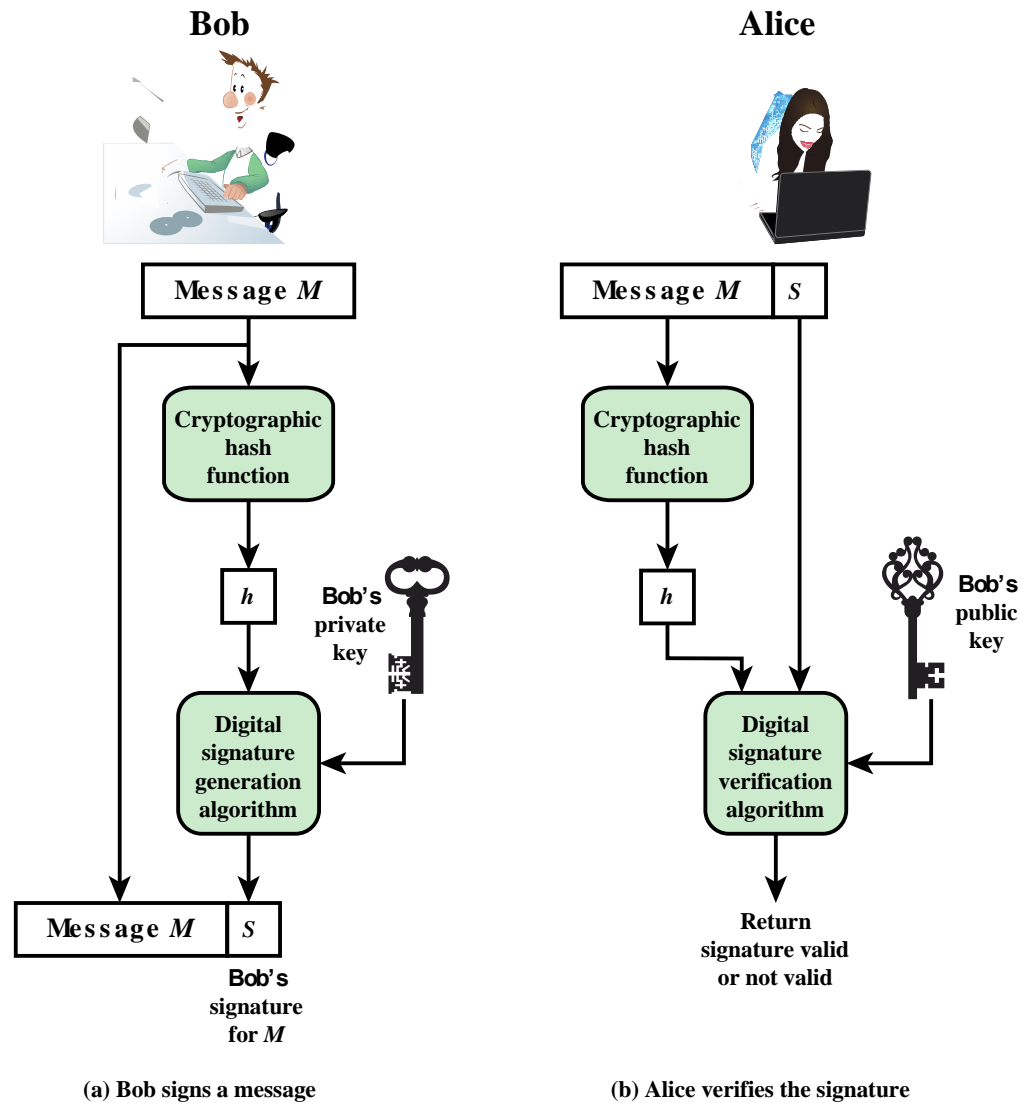
Şifreli metnin şifresini çözmek için özel anahtarı bilen alıcı için hesaplama açısından kolay

Saldırganın genel anahtardan özel anahtarı belirlemesi hesaplama açısından mümkün değil

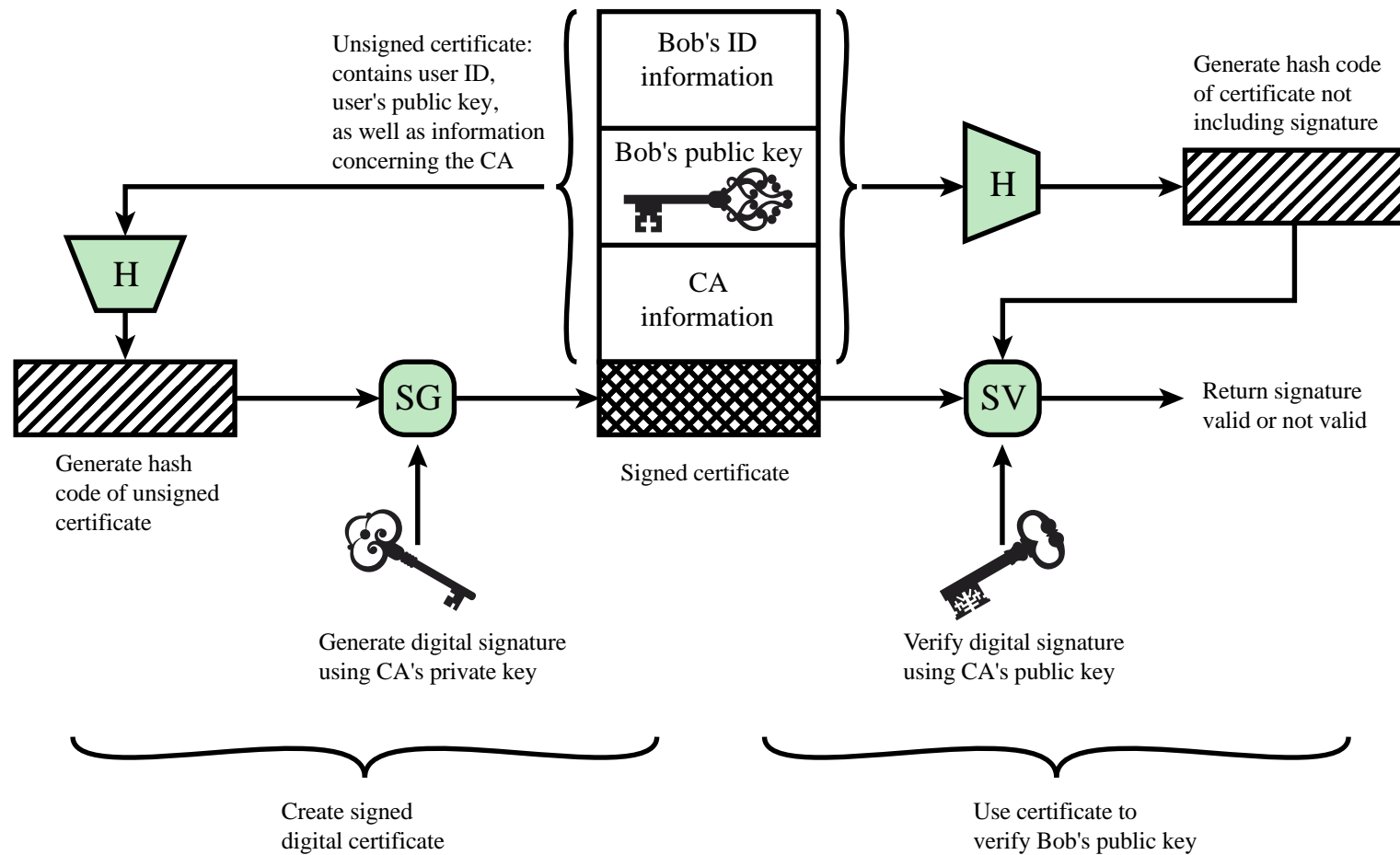


# Dijital imzalar

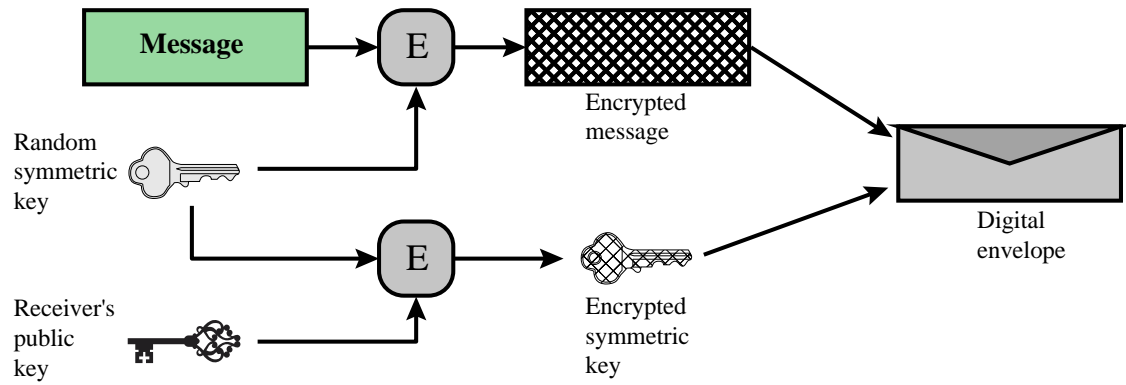
- NIST FIPS PUB 186-4 dijital imzayı şu şekilde tanımlar:  
” Doğru şekilde uygulandığında, kaynak kimlik doğrulamasını, veri bütünlüğünü ve imzanın reddedilmemesini doğrulamak için bir mekanizma sağlayan verilerin kriptografik dönüşümünün sonucudur.”
- Bu nedenle, bir dijital imza, bir dosya, mesaj veya başka bir veri bloğu biçiminin bir işlevi olarak bir aracı tarafından oluşturulan veriye bağlı bir bit modelidir.
- FIPS 186-4, üç dijital imza algoritmasından birinin kullanımını belirtir:
  - Dijital İmza Algoritması (DSA)
  - RSA Dijital İmza Algoritması
  - Eliptik Eğri Dijital İmza Algoritması (ECDSA)



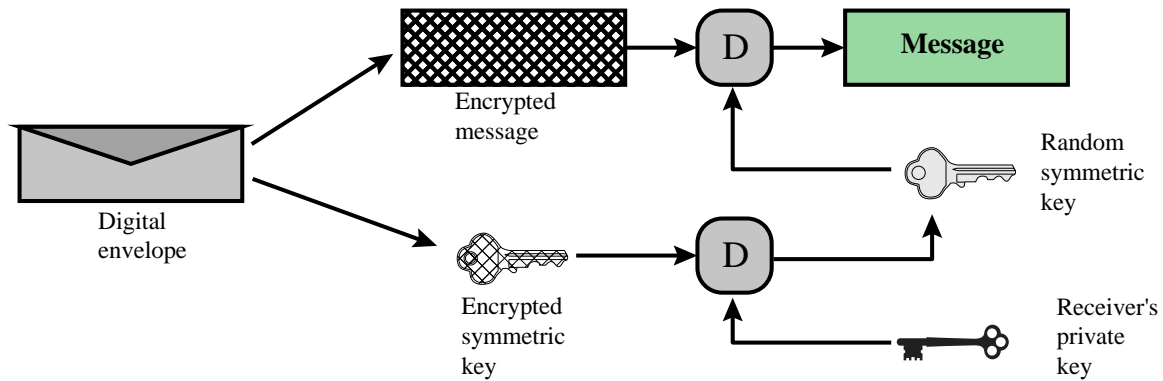
**Figure 2.7 Simplified Depiction of Essential Elements of Digital Signature Process**



**Figure 2.8 Public-Key Certificate Use**



(a) Creation of a digital envelope



(b) Opening a digital envelope

**Figure 2.9 Digital Envelopes**

# Rastgele Sayılar

## Kullanım alanları:

- Açık anahtar algoritmaları için anahtarlar
- Simetrik akış şifresi için akış anahtarı
- Geçici oturum anahtarı olarak veya dijital zarf oluşturmada kullanım için simetrik anahtar
- Tekrar saldırılarını önlemek için el sıkışma
- Oturum anahtarı



# Rastgele Sayı Gereksinimleri

## Rastgelelik

- Kriterler:
  - Üniform dağılım
    - Sayıların her birinin oluşma sıklığı yaklaşık olarak aynı olmalıdır.
  - Bağımsızlık
    - Dizideki hiçbir değer diğerlerinden çıkarılamaz

## Tahmin edilemezlik

- Her sayı dizideki diğer sayılardan istatistiksel olarak bağımsızdır
- Rakip, dizinin gelecekteki öğelerini daha önceki öğelere dayanarak tahmin edememelidir.

# Rastgele ve Sözde Rastgele

Tekrar saldırılarını önlemek için el sıkışma Kriptografik uygulamalar tipik olarak rasgele sayı üretimi için algoritmik tekniklerden yararlanır

- Algoritmalar deterministiktir ve bu nedenle istatistiksel olarak rastgele olmayan sayı dizileri üretirler.

Sözde rasgele sayılar şunlardır:

- İstatistiksel rastgelelik testlerini karşılayan diziler üretildi
- Tahmin edilebilir olması muhtemel

Gerçek rastgele sayı üretici (TRNG):

- Rastgelelik üretmek için deterministik olmayan bir kaynak kullanır
- Çoğu, öngörülemeyen doğal süreçleri ölçerek çalışır
  - Örneğin. radyasyon, gaz deşarjı, sızdıran kapasitörler
- Modern işlemcilerde giderek daha fazla destekleniyor

# Pratik uygulama: Saklanan Verilerin Şifrelenmesi

İletilen verileri şifrelemek için ortak

Depolanan veriler için çok daha az yaygın

Etki alanı kimlik doğrulaması ve işletim sistemi erişim kontrollerinin ötesinde genellikle çok az koruma vardır.

Veriler süresiz olarak arşivlenir

Silinse bile, disk sektörleri yeniden kullanılana kadar veriler kurtarılabilir

Depolanan verileri şifrelemek için yaklaşımlar:

Piyasada bulunan bir şifreleme paketini kullanın

Arka uç cihaz

Kütüphane tabanlı bant şifreleme

Arka planda dizüstü/PC veri şifrelemesi