

# Bilgi Güvenliđi ve Kriptoloji

Dr. Hamdi Murat Yıldırım

Bilgisayar Teknolojisi ve Bilişim Sistemleri Bölümü  
Bilkent Üniversitesi

<http://hmurat.bilkent.edu.tr>

hmurat@bilkent.edu.tr

@h\_muratyildirim

Uluslararası Adli Bilişim Sempozyum 2014  
31 Mayıs – 01 Haziran

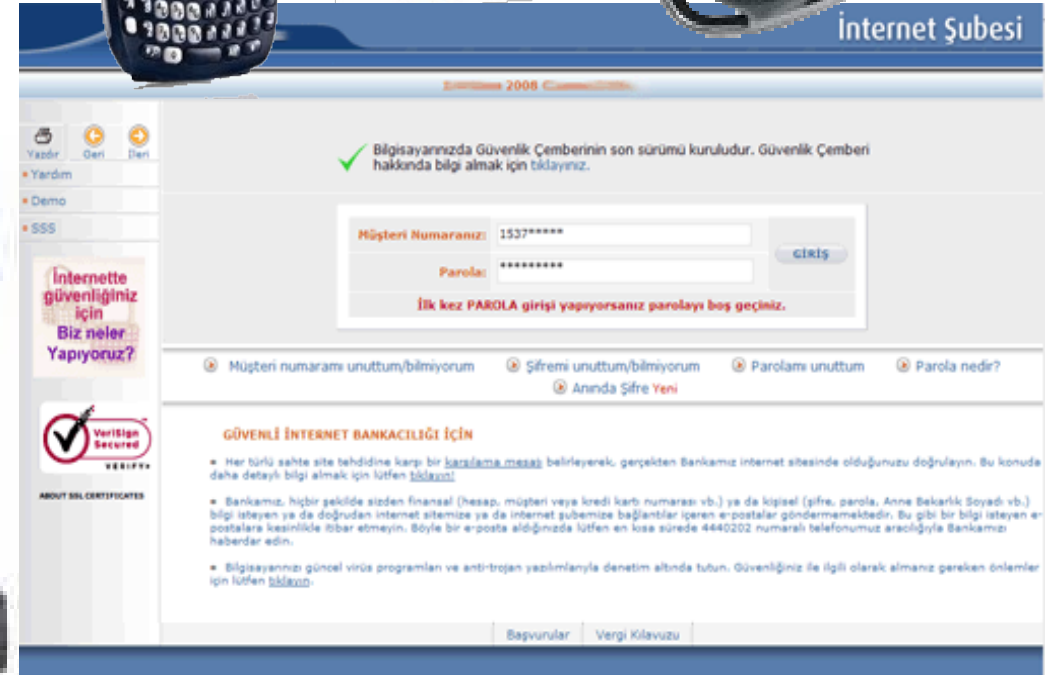
# Giriş: Tanımlar

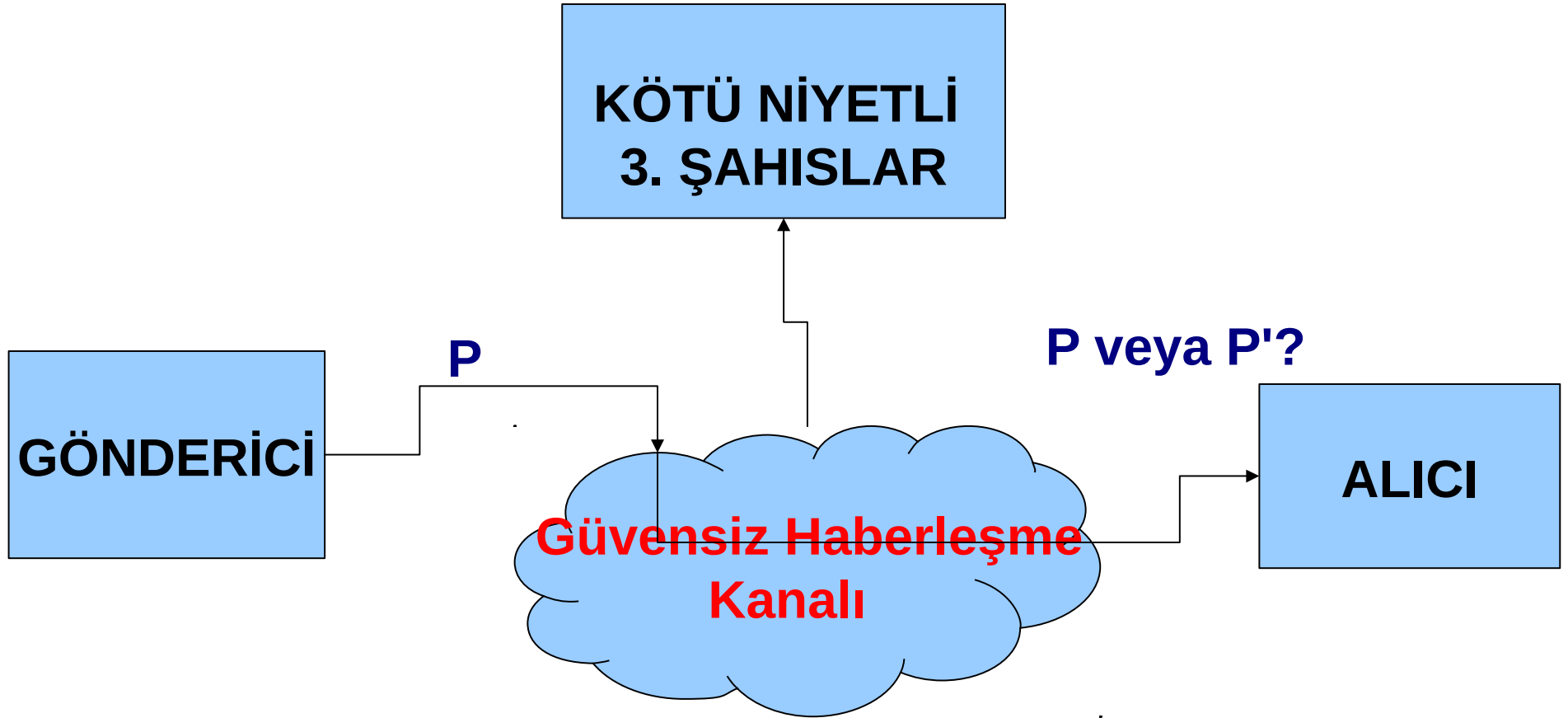
- **Kriptografi**, haberleşmenin güvenliğini üçüncü, kötü niyetli taraflara karşı sağlamak için uygulama ve teknikleri içeren çalışmaların bütünüdür.
- **Kriptoloji** ise kriptografi ve kriptanaliz -kriptografik algoritmalarını analizleri- ile ilgili bir bilim dalıdır.
- **Bilgi güvenliği** için verinin yetkisiz değiştirilmeye, kullanılmaya, ifşa edilmesine, incelenmesine, kaydedilmesine, hasar verilmesine karşı koruma yöntemleri (güvenlik hedeflerine ulaşılarak) çalışılır.
- **Ağ güvenliğinde**, ağ erişimli sistemlere ve kaynaklara yetkisiz erişimleri izlemek ve önlemek için uygulanan yöntemler (kriptografik algoritmaların ağ protokollerinde kullanımı vb.) incelenir.

# Günümüzde kriptografik sistemler

Bugün, kriptografi çok geniş uygulama alanlarına dahil olarak günlük hayatın önemli bir parçası olmuştur:

- sim kartlar,
- cep telefonları,
- uzaktan kumandalar,
- online bankacılık,
- online alışveriş,
- uydu alıcıları,
- VS.

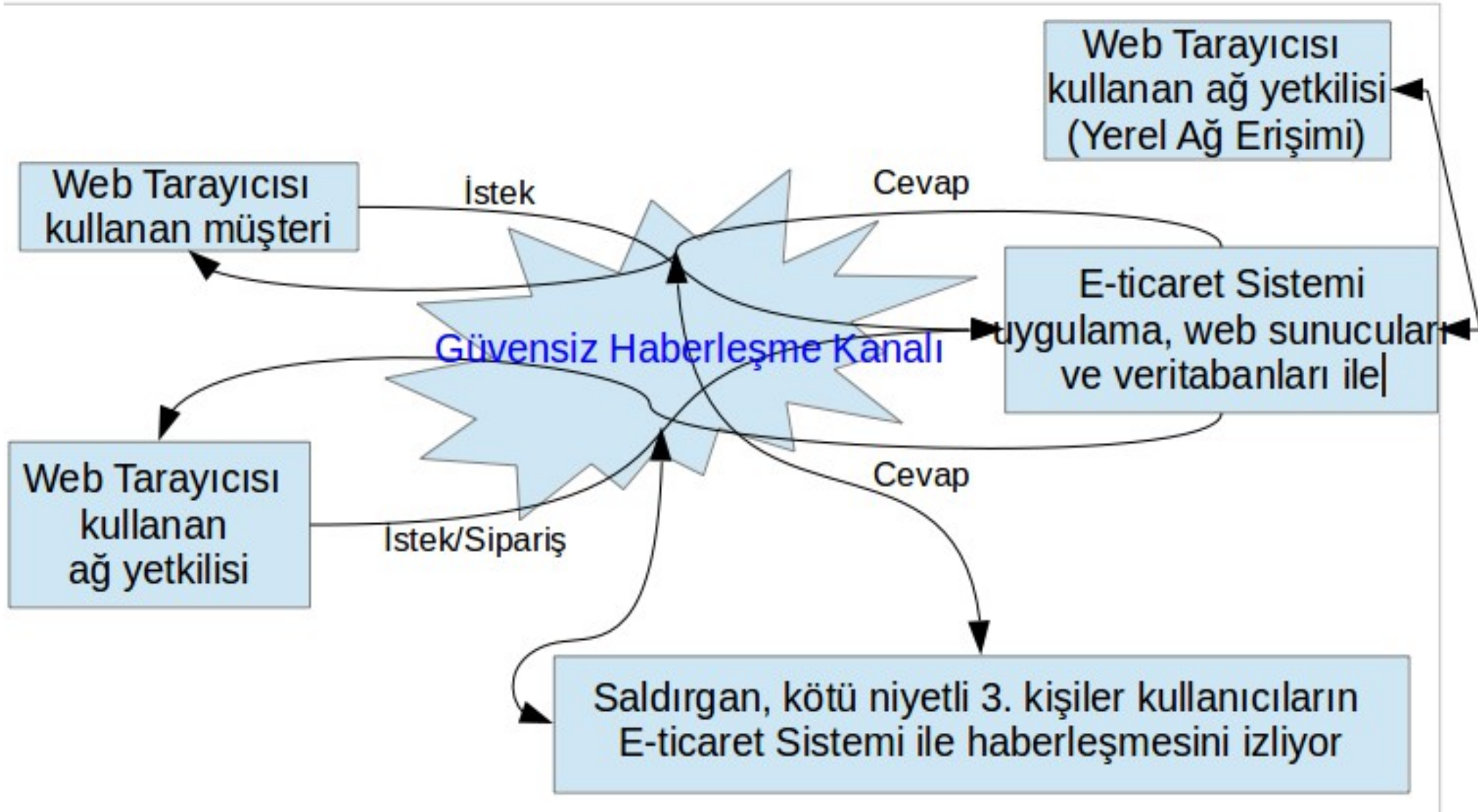




**Güvenli Olmaya Kanal Üzerinden Haberleşme**

**P: Açık Metin**

**P': Açık Metin Değişmiş hali**



# Giriş: Tanımlar

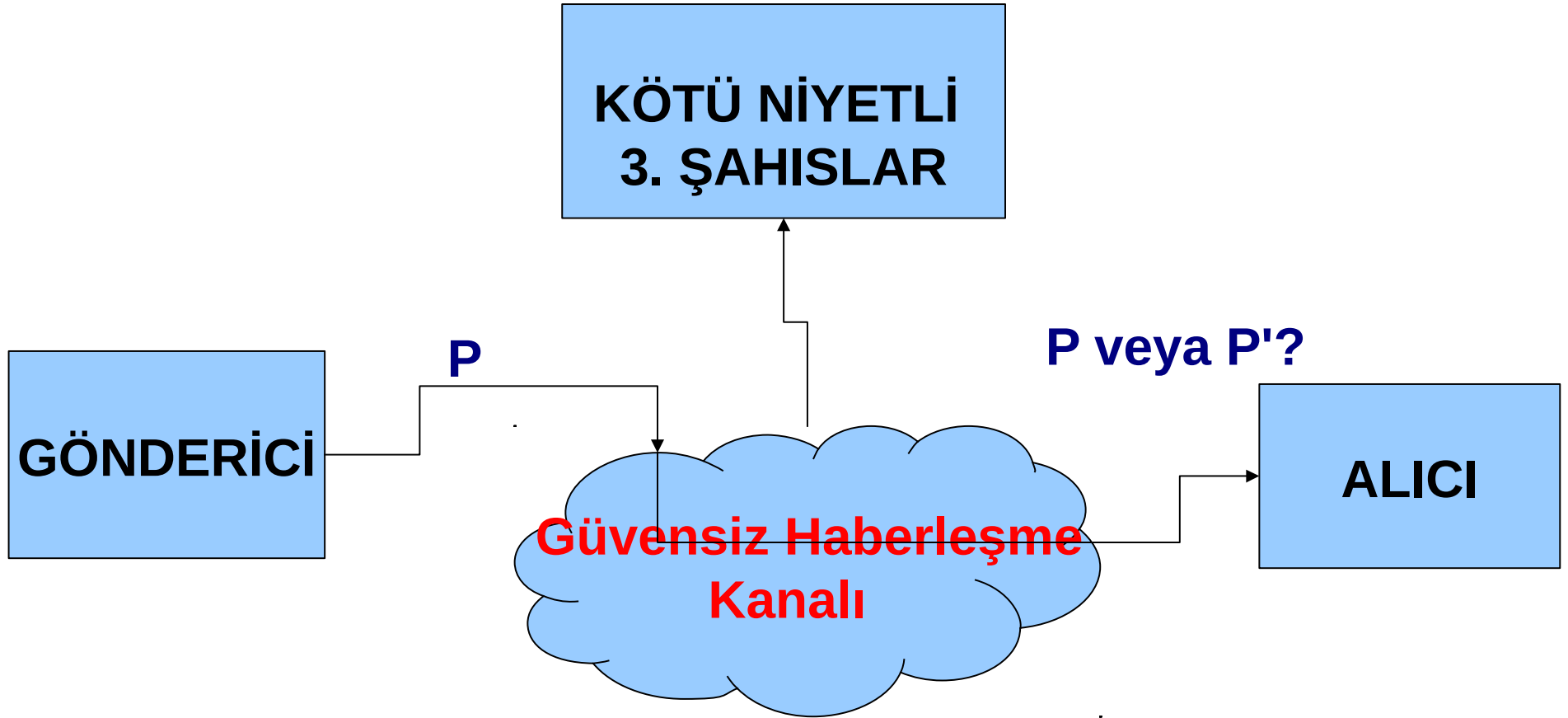
- **Güvenlik hedeflerine**

- Gizlilik (mahremiyet) -confidentiality-,
- bütünlük -integrity-,
- süreklilik -availability,
- kimlik denetimi (doğrulama) -authentication-,
- inkar edilememelik -non-repudiation-,
- izlenebilirlik -accountability-

ulaşılmasına sağlayan **kriptografik algoritmalar**, güvenlik protokollerin bileşenleri olarak haberleşmenin mahruz kalabileceği kötü niyetli etkileri önlemek için kullanılırlar.

# Bilgi Güvenliđi Hedefleri

- **Gizlilik:** bilgiye erişim yetkisine sahiplerin erişim sağlamak
- **Bütünlük:** bilginin deđiştirilme veya yok edilmeye karşı korunması (bilgini kaynađını dođrularak vb.)
- **Süreklilik:** Zamanında ve güvenilir şekilde bilgi kullanımının sağlanması
- **Kimlik denetimi:** kullanıcıların iddia ettikleri kullanıcılar olduđunun ve bir sisteme erişen verinin güvenilir kaynaktan geldiđinin kontrolü.
- **İnkâr edilemezlik:** oluşturulan kanıtlar yardımıyla veri gönderen veya alanın veri aktarımı veya alımı yaptıđını inkâr etmesini engellemek.
- **İzlenebilirlik:** bir güvenlik ihlali yapıldıđında sorumlusunu ortaya çıkarmak için kullanıcı işlemlerinin takibinin yapılmasının sağlanması  
- İnkâr edilemezliđi, caydırmayı, ihlal tespitini ve önlenmesini destekler-



**Güvenli Olmaya Kanal Üzerinden Haberleşme**

**P: Açık Metin**

**P': Açık Metin Değişmiş hali**

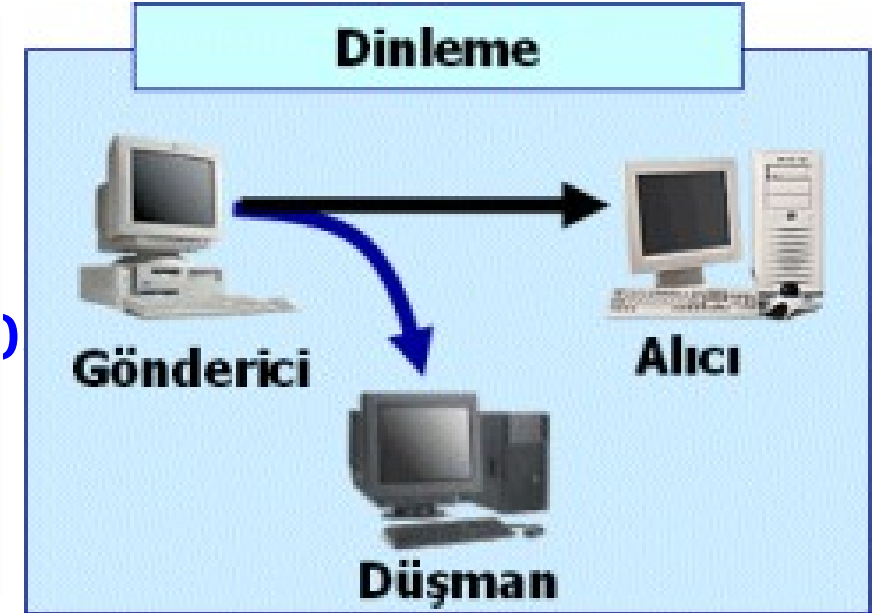
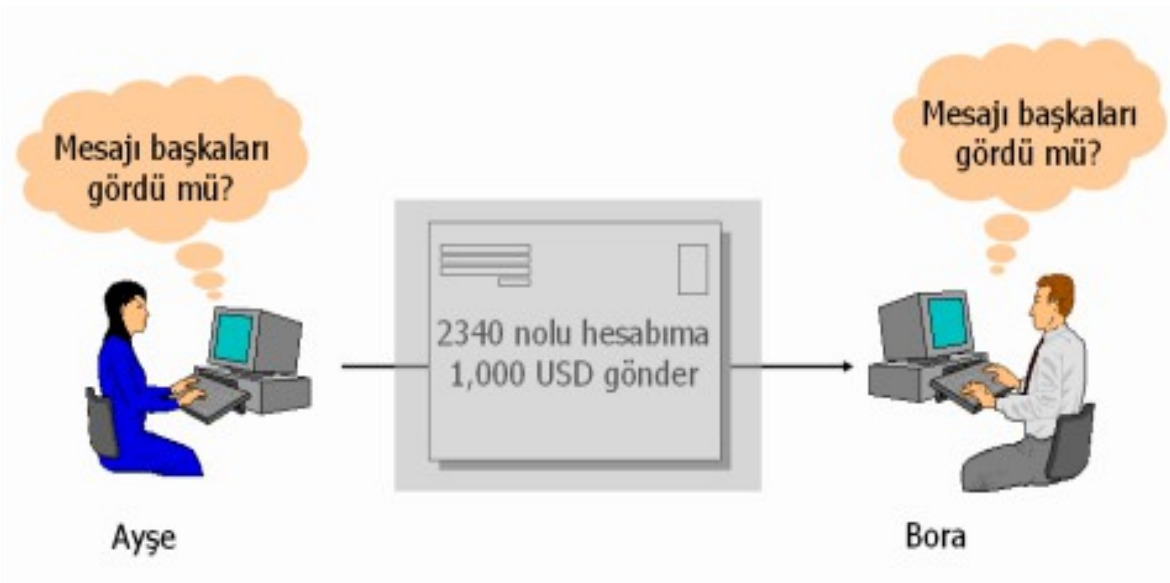


# Güvenlik ihlallerine örnekler

- **Örnek 1 :** Kullanıcı A, e-posta mesajı ekinde açığa çıkması / başka birinin eline geçmesi istenmeyen bir dosyayı kullanıcı B ye gönderir. Bu dosyayı okuma yetkisi olmayan Kullanıcı C, e-posta iletimini izleyebilir ve dosyanın transferi sırasında tarafların haberi olmadan bir kopyasını alabilir (**Gizlilik ihlali**)  
-Pasif Saldırı -

# Gizlilik İhlali

Haberleşme kanalını dinleyen saldırgan gönderici ile alıcı arasındaki mesaj trafiğini dinleyebilir ve elde ettiği mesajları okuyarak bu haberleşmenin gizliliğini bozar. Bu tehdit dinleme tehdidi olarak bilinir.

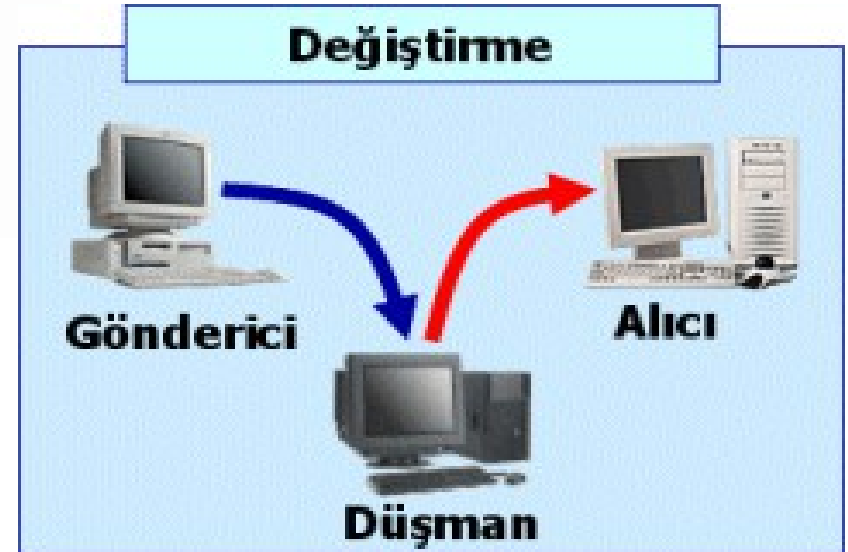
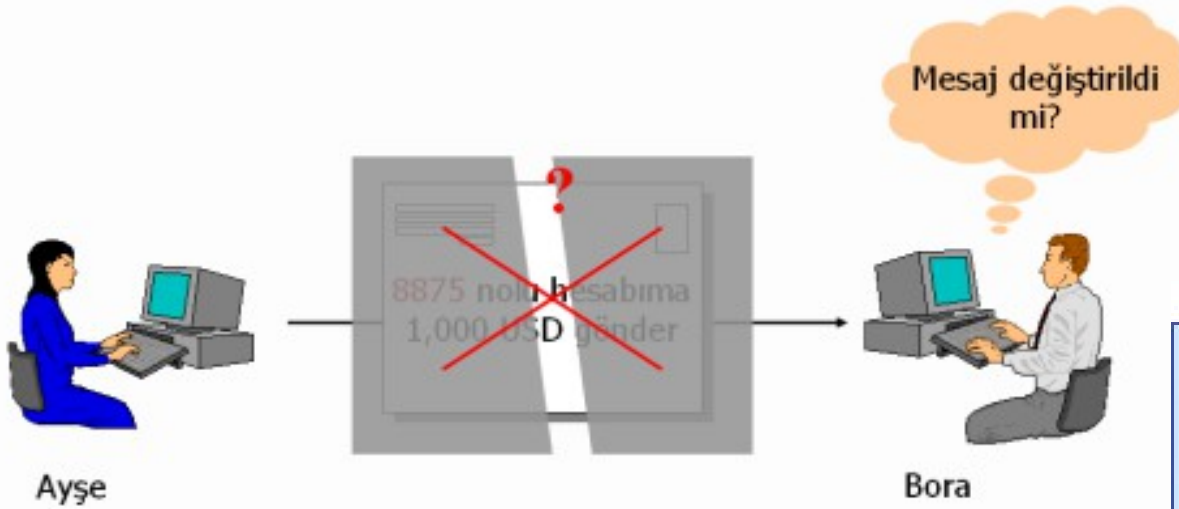


# Güvenlik ihlallerine örnekler

**Örnek 2 :** Kullanıcı C, örnek 1 de yapılanların yerine e-posta iletimine müdahale edebilir ve bu dosyayı başka bir dosya ile değiştirip, sanki mesaj Kullanıcı A dan geliyormuşcasına tekrar Kullanıcı B ye gönderebilir (**Bütünlük ihlali, mesajın sahibi veya mesajı/dosyayı oluşturanın kontrol edilmemesi durumu**) -**Aktif atak: araya girme ve tekrar gönderme**

# Bütünlük İhlali

Haberleşmeye müdahale edip göndericinin mesajlarını değiştiren saldırgan, alıcıya giden mesajı istediği şekle sokabilir. Bu tehdit mesajın bütünlüğünü bozan değiştirme tehdididir.

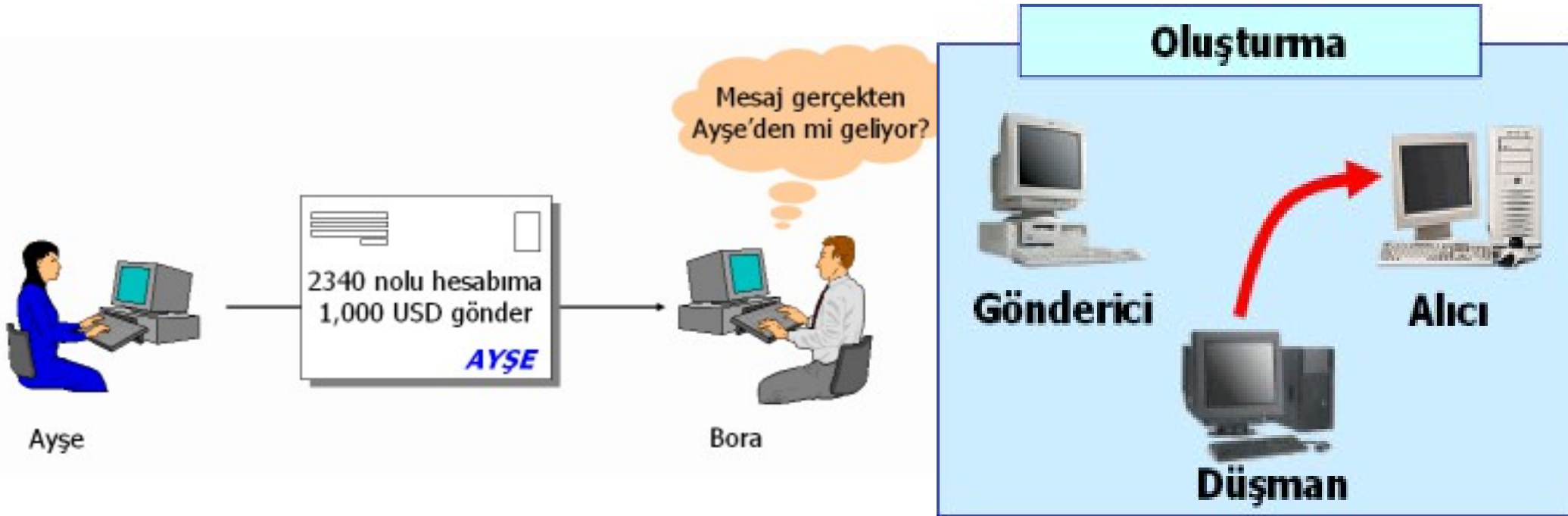


# Güvenlik ihlallerine örnekler

- **Örnek 3 :** Kullanıcı C nin, Kullanıcı A yerine (Kullanıcı A böylesine bir mesajı Kullanıcı B ye göndermemektedir), Kullanıcı B ye e-posta mesaj göndermesi (**Kimlik denetimi ihlali, mesajın sahibi veya mesajı/dosyayı oluşturanın kontrol edilmemesi durumu**) – **Aktif Saldırı: başkasını taklit etme**

# Kimlik Doğrulama İhlali

Saldırgan, alıcıya göndericinin kimliğini taklit ederek bir mesaj gönderebilir. Bu durumda eğer alıcı güvenilir bir kimlik doğrulaması yapmıyorsa yanlış mesajlarla kandırılabilir.



# Güvenlik ihlallerine örnekler

- **Örnek 4 :** Müşteri B nin, çağrı merkezi üzerinden müşteri temsilcisi G ye bir talimat vermesi ve G nin bu talimatı yerine getirmesi. Fakat Müşteri B nin bu talimatı verdiğini inkar etmesi (**İnkâr edilememelik ihlali**, **talimatı verenin inkar etmemesini için kanıtın oluşturulmaması**)

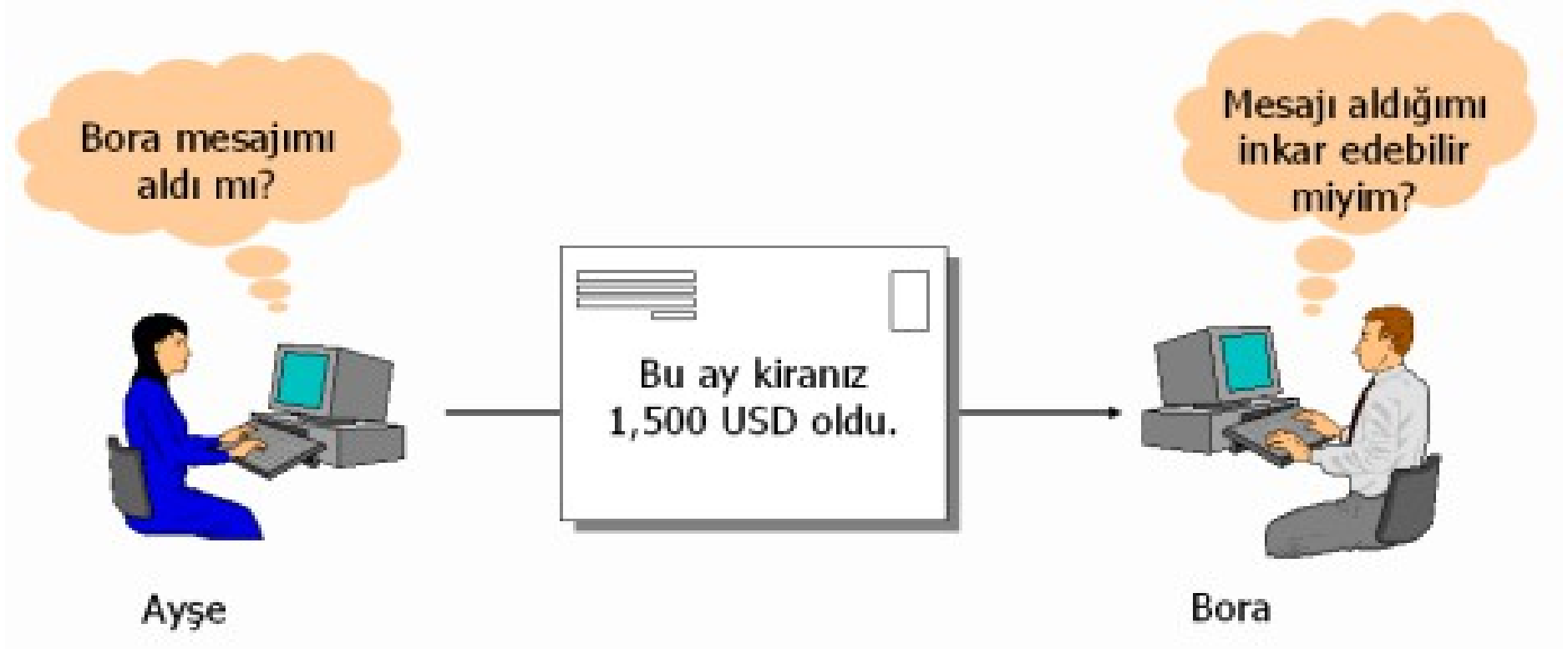
# Güvenlik ihlallerine örnekler

- **Örnek 5 :** Müşteri B nin, çağrı merkezi üzerinden müşteri temsilcisi G ye bir talimat vermesi ve G nin bu talimatı yerine getirmemesi. G nin bu talimatı aldığını inkar etmesi (**İnkâr edilememezlik ihlali**, talimatı alanın, aldığını inkar etmemesini için kanıtın oluşturulmaması)



# İnkâr Edememezlik İhlali

Mesajı gönderen veya alan tarafın bu işi yaptığını inkâr etmesi söz konusu olabilir. Bu kötü niyetli girişimi boşa çıkaracak mekanizmalara ihtiyaç vardır.



# Güvenlik ihlallerine örnekler

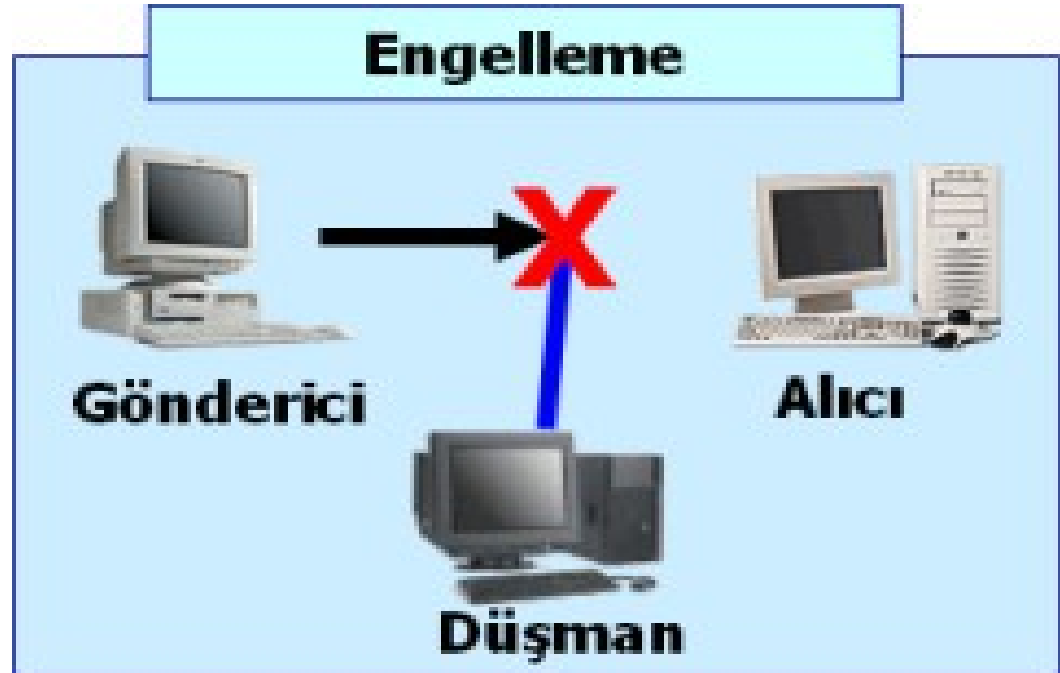
- **Örnek 6** : Yönetici D, çalışanı E yi işten çıkarır ve E nin firma sistemlerine ulaşma yetkisinin kaldırılma emrini içeren e-posta mesajını gönderir fakat bunun yerine ulaşması E tarafı bir süreliğine engeller. Bu süre boyunca çalışan E nin erişim yetkisinin olduğu firma kaynaklarına zarar verir ve bu zarar sonrası, yönetici D nin e-posta mesajının yerine ulaşmasına izin verir (**Süreklilik ihlali**)  
**Aktif saldırı**

Örnekler: Kitap “Cryptography and Network Security” 5. baskı, Yazar: W. Stallings

Slayt 10,12,14, 17 ve 18 şekilleri: <http://www.kamusm.gov.tr/dosyalar/kitaplar/aaa/>

# Sürekli İhlali

Saldırgan, haberleşen iki taraf arasındaki hattı veya haberleşme araçlarını kullanılmaz hale getirerek haberleşmenin sürekliliğini engellemeye çalışır.



# Güvenlik Hedeflerine Ulaşmanı Yolu: Kriptografi

- **Gizlilik hedefi:** Simetrik şifreleme (büyük dosyalar) ve Asimetrik Şifreleme (küçük mesajlar)
- **Bütünlük hedefi:** Mesaj Kimlik doğrulama (MAC) algoritmaları, Özet fonksiyonları
- **İnkâr edilememelik hedefi:** Dijital imza ve X.509 sertifikaları (dolayısıyla Asimetrik şifreleme, özet fonksiyonları ve Açık anahtarlı Altyapısı)
  - **Kimlik Denetimi hedefi:** Mesaj Kimlik doğrulama (MAC) algoritmaları, Özet fonksiyonları ve dijital imza
  - **Süreklilik hedefi:** Erişim ve Yetkilendirme kontrolü, dolaylı yoldan: Mesaj Kimlik doğrulama (MAC) algoritmaları, Özet fonksiyonları ve dijital imza
  - **İzlenebilirlik hedefi:** İnkâr edilememelik kanıtı için, dijital imza

# Güvenlik hedeflerinin gözetilen Standart: OpenPGP

- **Pretty Good Privacy (PGP)** : veri şifreleme ve deşifreleme yazılımı.
- Haberleşme için veri gizliliği ve kimlik denetimi sağlar.
- PGP dosya, e-posta, klasör ve disk bölümlerinin dijital imzalama, şifreleme ve deşifrelemesinde kullanılır.
- 1991 de Phil Zimmermann tarafından yaratıldı.
- PGP ve benzerleri **OpenPGP** standartını (RFC 4880) izler.
- OpenPGP nin özgür gerçekleştirilmesi: **GNU PG**
- **Tavsiye: güvenlik hedeflerinin nasıl gerçekleştirildiğini görmek için PGP işlemlerini incelemek**

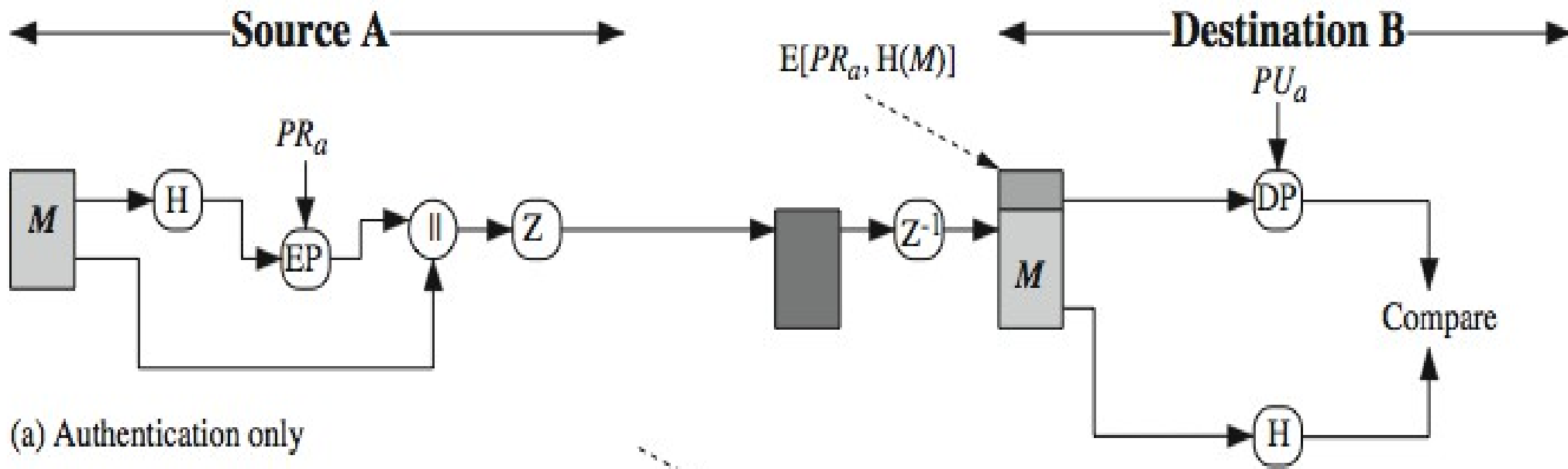
# PGP Servisleri

- Kimlik Denetimi (Authentication)
- Gizlilik (Confidentiality)
- Sıkıştırma (Compression)
- E-posta uyumluluğu (E-mail compatibility)
- Parçalama ve birleştirme  
(Segmentation and Reassembly)

Kullanıcı sadece ilk iki işlemi gözlemler.

# PGP: Kimlik Doğrulama basamakları

- Dijital imza oluşturmakla aynıdır:
- **Gönderen**
  - Mesaj  $M$  yi yaratır.
  - Özet fonksiyonu  $H$  ile mesajın özetini  $H(M)$  oluşturur
  - $H(M)$ , **Göndericinin Kapalı anahtarı (GK)** ile şifrelenir ve imza oluşur.
  - Mesaj  $M$  nin sonuna bu imza eklenir.



(a) Authentication only

## Kimlik Denetim Aşamaları

Stallings kitabından

**M = Mesaj**  
**H = Özet fonksiyonu**  
**|| = birleştirme**  
**Z = sıkıştırma**  
 **$Z^{-1}$  = Sıkıştırılmışı açmak**

**EP = asymmetric şifreleme**  
**DP = asymmetric şifreleme**  
 **$KR_a$  = A'nın kapalı anahtarı**  
 **$KU_a$  = A'nın açık anahtarı**



# PGP: Gizlilik

- **Gönderen**

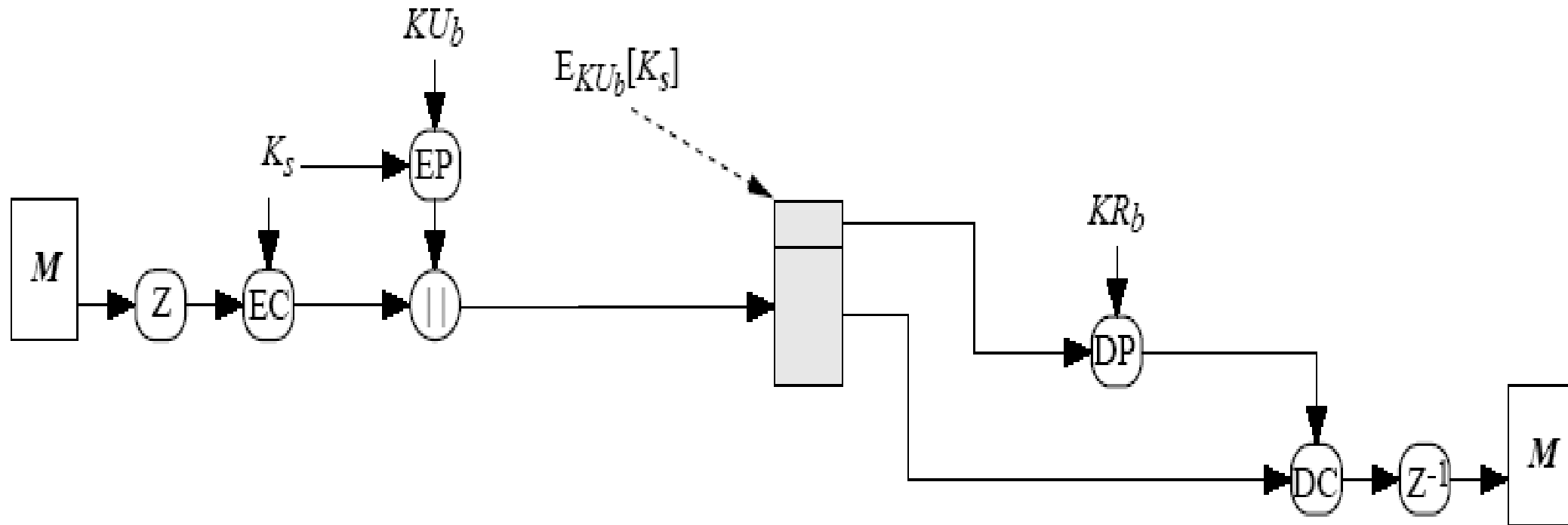
- Mesajı ve mesaja özgü gizli simetrik şifreleme anahtarını (session key) oluşturur.
- Mesajı blok şifreleme (EC) algoritmalarından biri ile şifreler.
- Gizli anahtarı, Alıcının açık anahtarı ile şifreler  
-anahtar paylaşımı-
- Şifrelenen mesaj ve gizli anahtar iletilir.

# PGP: Gizlilik

- **Alıcı**
  - Gizli anahtarı, açık anahtarı ile deşifreleme yaparak kurtarır.
  - Gizli anahtar ile şifreli mesajı deşifreler.

Source A

Destination B



**EC = simetrik  
şifreleme**

**DC = simetrik  
deşifreleme**

**$K_s$  = oturum gizli  
anahtarı**

**Stallings  
kitabından ...**

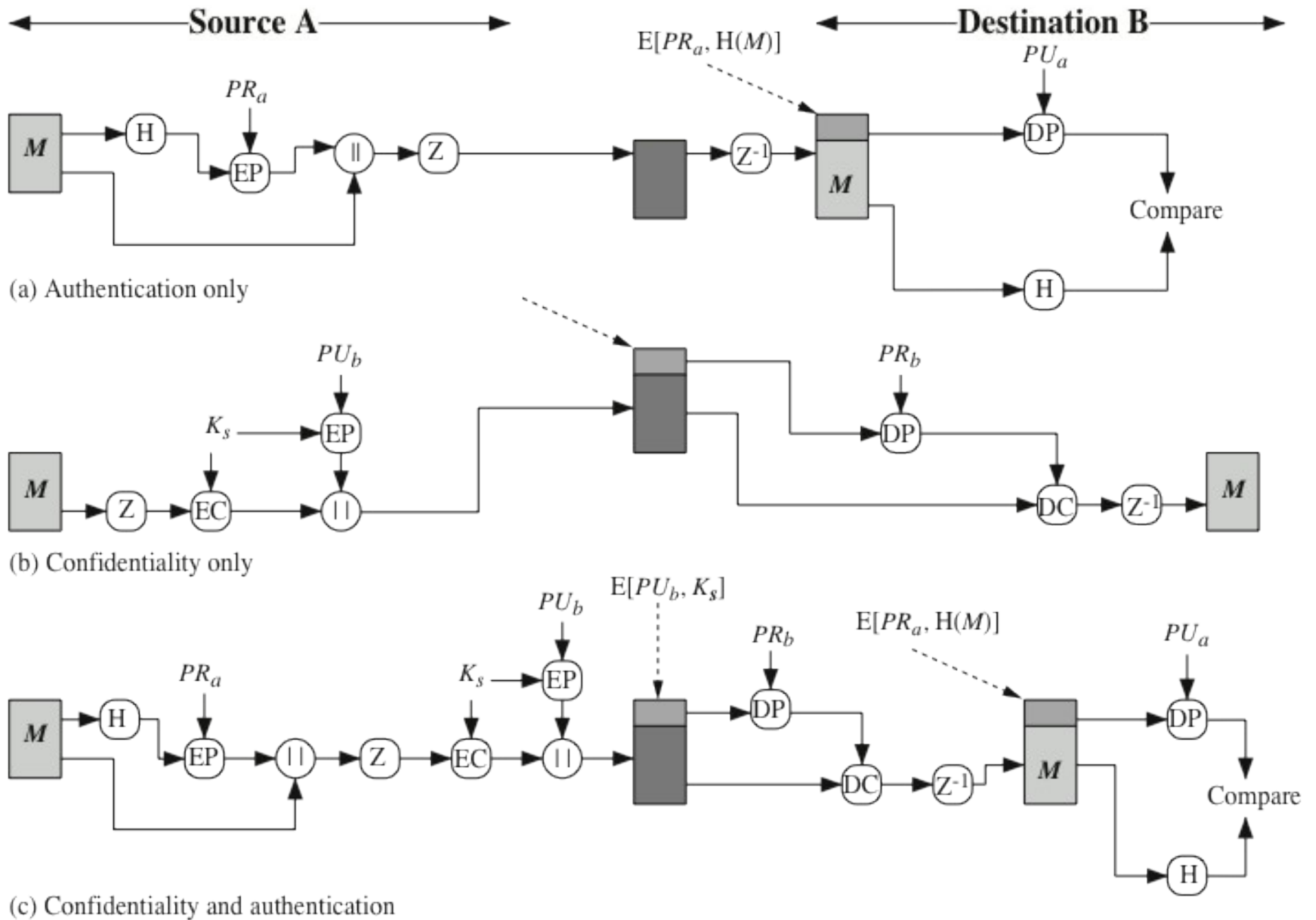


Figure 19.1 PGP Cryptographic Functions

# Kaynak Tavsiyesi

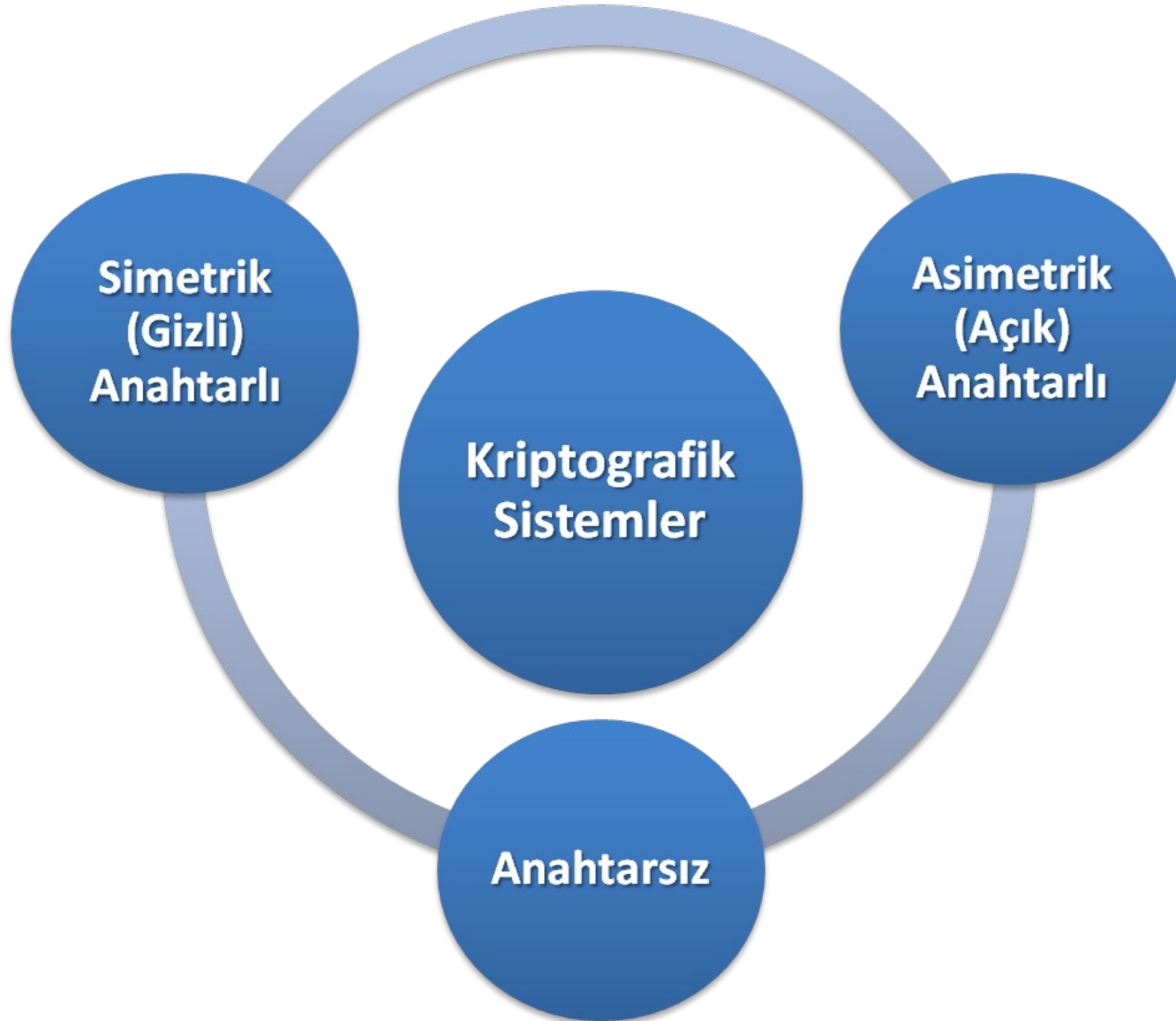
- Sistem tasarımıda güvenlik hedeflerini düşünürken ortak dil için RFC 2828 (Internet Güvenliği Terimler Sözlüğü)

<http://www.ietf.org/rfc/rfc2828.txt>

- Akademik Bilişim, [Kriptoloji Kurs Notları](#)
- TÜBİTAK UEKAE, [Temel Kavramlar](#)
- TÜBİTAK UEKAE, [Açık Anahtar Altyapısı Eğitim Kitabı](#)
- Cryptology, Encyclopædia Britannica
- [Bilgi Güvenliği \(İngilizce, Wikipedia\)](#)
- <https://wiki.internet2.edu/confluence/display/2014infosecurityguide/Cryptography>
- [NIST, Information Security Guide For Government Executives](#)

# Sınıflandırma

---



# KRİPTOLOJİ

## KRİPTOGRAFI

## KRİPTOANALİZ

### -anahtarlı-

## Simetrik Şifreleme

## Asimetrik Şifreleme

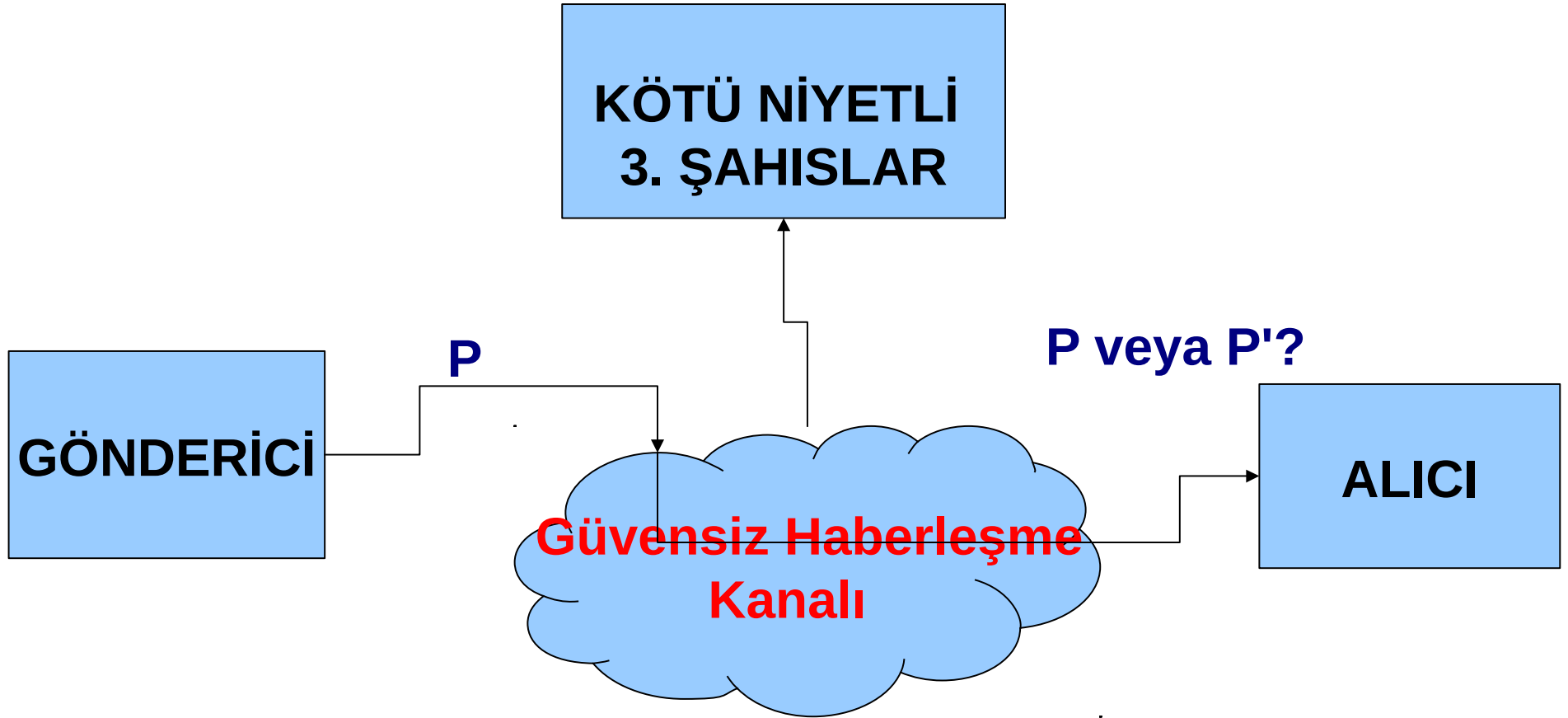
## MAC -anahtarlı-/ Özet Fonksiyonlar-anahtarsız

“Blok Şifre Sistemleri”  
(AES, 3DES, Camellia,vb.)

“Akan Şifre”  
Sistemleri  
(RC4, A5/1,  
A5/2 vb.)

İmzalama  
Algoritmaları  
DSA, ECDSA,  
RSA vb.

Anahtar Paylaşımı  
Algoritmaları  
RSA, DH, Eliptik  
Eğri tabanlılar vb.

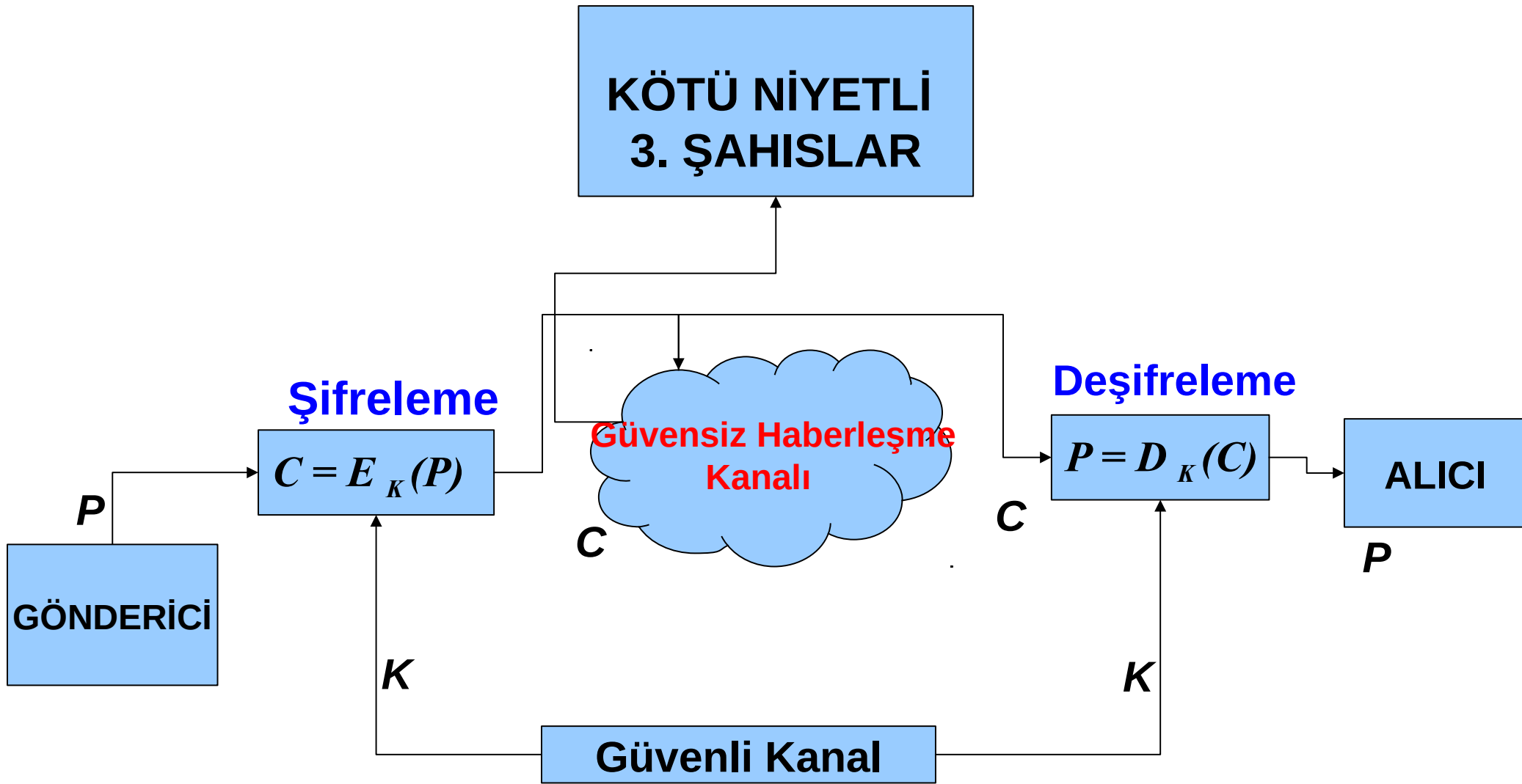


**Güvenli Olmaya Kanal Üzerinden Haberleşme**

**P: Açık Metin**

**P': Açık Metin Değişmiş hali**





## Simetrik Şifreleme

**K:** Gizli Anahtar

$C = E_K(P) = E(K, P)$ ;  $E$ : Şifreleme

**C:** Kapalı (Şifrelenmiş) Metin /  $P = D_K(C) = D(K, C)$ ;  $D$ : Deşifreleme

# Simetrik Şifreleme

Algoritmaların beş bileşeni:

- $\mathcal{P}$ : sonlu sayıda açık metin içeren küme.
- $\mathcal{C}$ : sonlu sayıda kapalı metin içeren küme.
- $\mathcal{K}$ : sonlu sayıda anahtar içeren küme.
- $\mathcal{E}$ : sonlu sayıda  $\mathcal{P} \rightarrow \mathcal{C}$  şifreleme fonksiyonlarının kümesi
- $\mathcal{D}$ : sonlu sayıda  $\mathcal{C} \rightarrow \mathcal{P}$  deşifreleme fonksiyonlarının kümesi
- Her anahtar  $K \in \mathcal{K}$  bir  $E_K \in \mathcal{E}$  ve  $D_K \in \mathcal{D}$  belirler.

**Tanım:** Her bir  $K \in \mathcal{K}$  için, n-bit blok şifre  $E : \mathcal{P} \times \mathcal{K} \rightarrow \mathcal{C}$ .

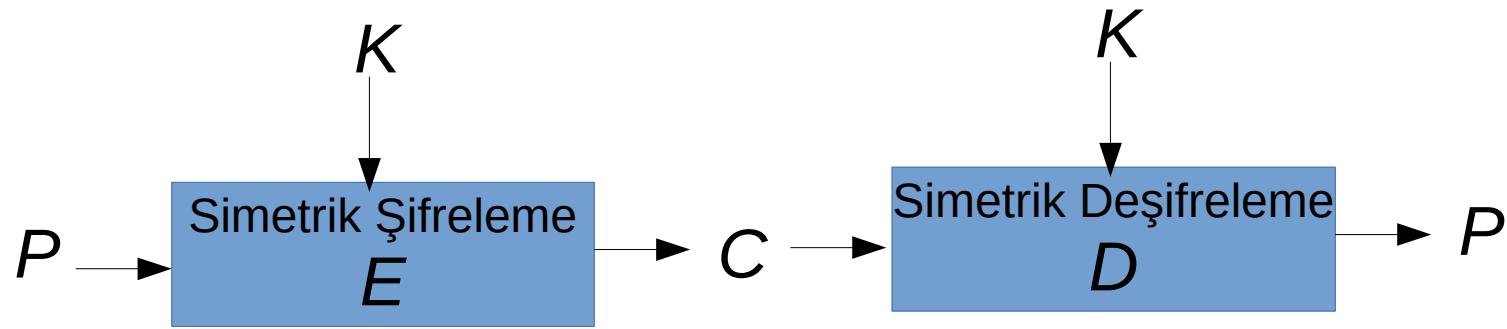
$E(P, K) = E_K(P) = C$  tersi alınabilir bir fonksiyon (Burada  $P \in \mathcal{P}$  ve  $E_K : \mathcal{P} \rightarrow \mathcal{C}$ ).

$E_K$  nin tersi, deşifreleme fonksiyonu  $D_K : \mathcal{C} \rightarrow \mathcal{P}$ , her bir  $C \in \mathcal{C}$  için  $D(C, K) = D_K(C) = P$  şeklinde tanımlanır.

Verilen her  $K$  anahtarı için,

$$D_K(E_K(P)) = P \text{ her açık metin } P \in \mathcal{P} \text{ ve}$$

$$E_K(D_K(C)) = C \text{ her kapalı metin } C \in \mathcal{C} \text{ için.}$$



**Blok şifre sistemi/algorithm**,  $n$ -bit açık metin bloğunu  $n$ -bit kapalı metin bloğuna taşır. Burada  $n$ , bloğun uzunluğudur.

Bu algoritma (fonksiyon),  $k$ -bit vektörler anahtar uzayı  $\mathcal{K}$  dan seçilen gizli anahtar  $K$  ile parametrize edilir.

GENEL VARSAYIM: Gizli anahtar rastgele seçilir.

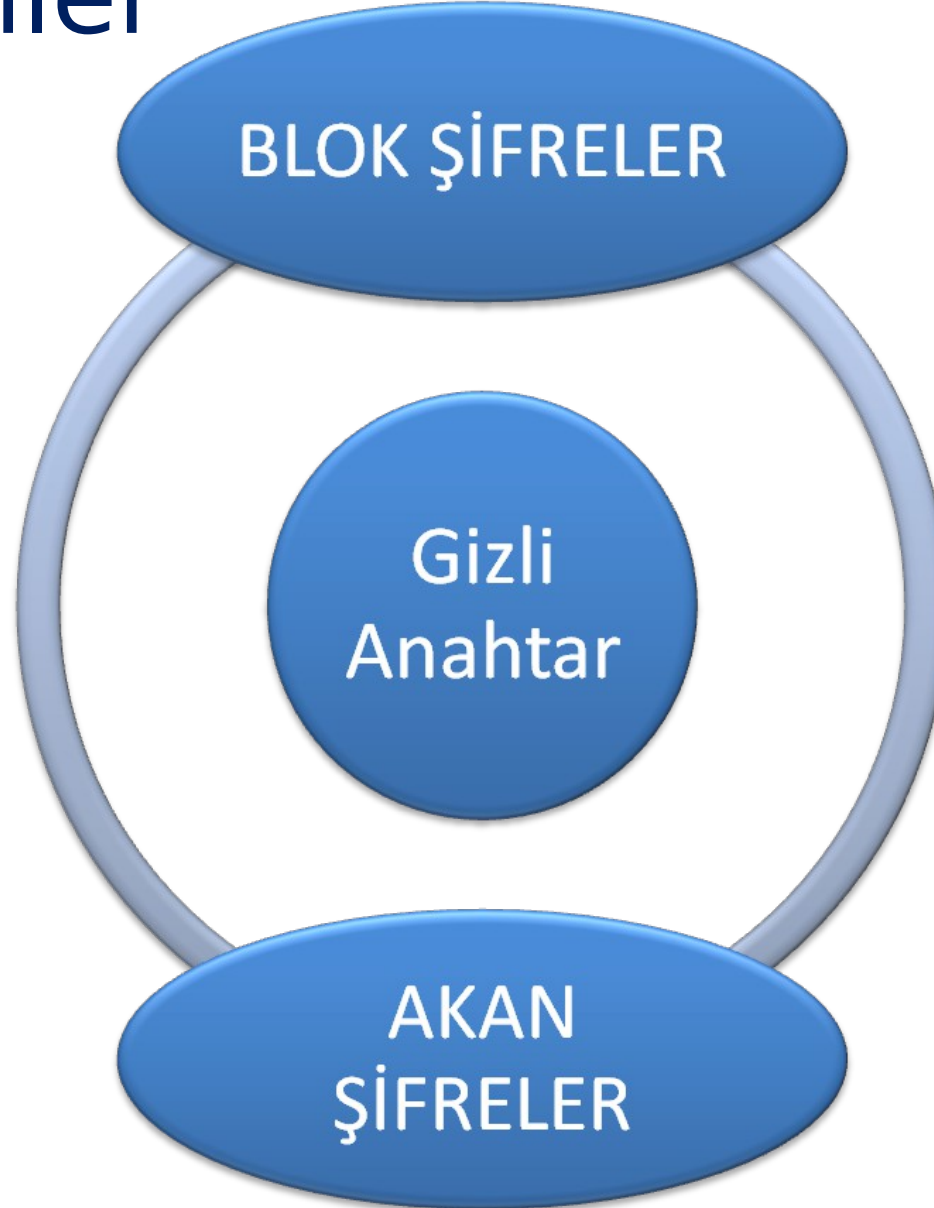
$\mathcal{P}$  ve  $\mathcal{C}$  kümelerinin her birinin eleman sayısı  $2^n$ .  
Toplamda  $\mathcal{P} \rightarrow \mathcal{C}$  ye  $|\mathcal{C}|^{|\mathcal{P}|} = 2^{n2^n}$  fonksiyon vardır. Fakat 1-1 fonksiyon sayısı  $2^n!$ .

Her bir gizli anahtar  $K \in \mathcal{K}$ , bu bütün 1-1 fonksiyonların  $|\mathcal{K}| = 2^k$  tane arasından birini seçmek için kullanılır.

# Simetrik (Gizli) Anahtarlı Sistemler

---

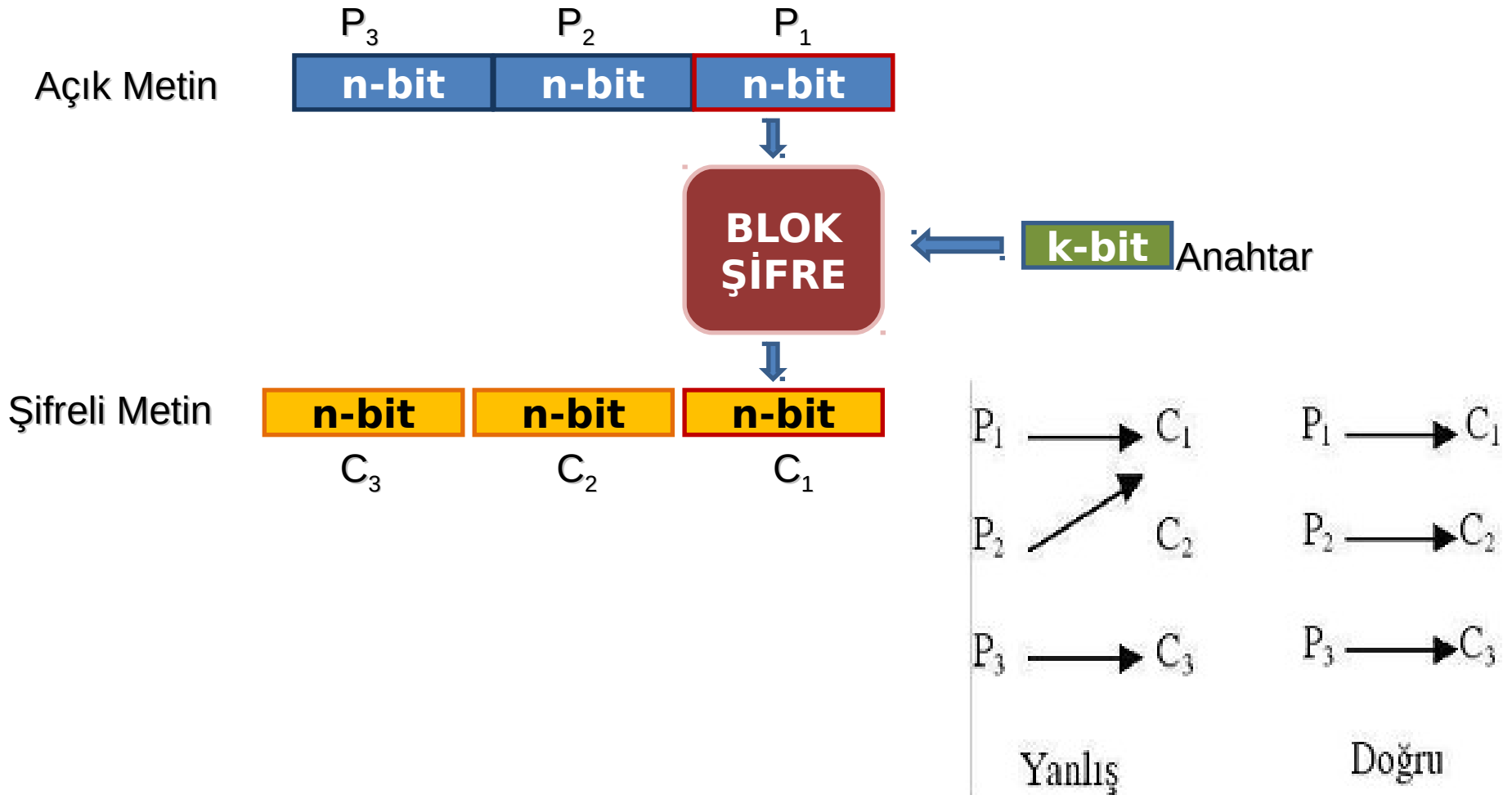
DES, AES, IDEA,  
KASUMI, SAFER



A5/1 (GSM),  
RC4 (WEP),  
E0(Bluetooth)

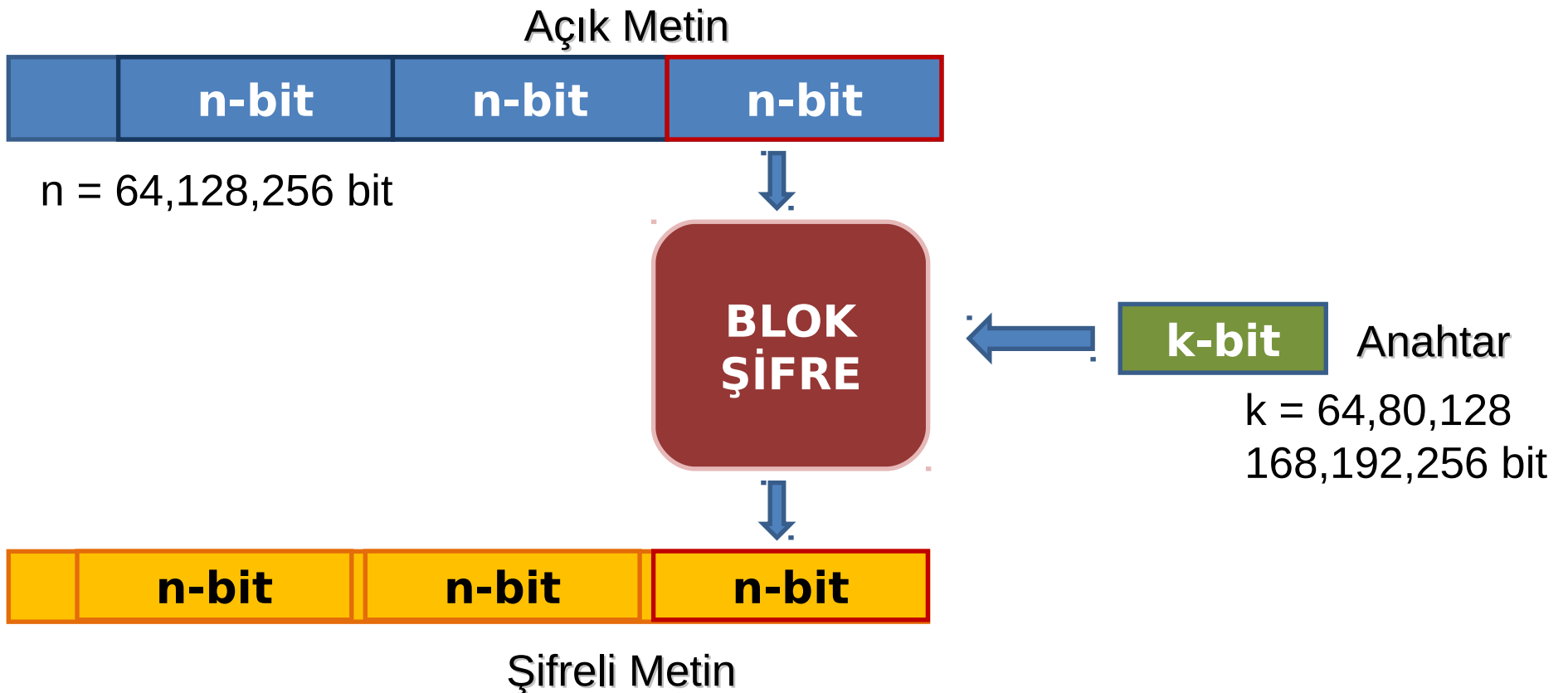
# Blok Şifreler

➤ Aynı anahtarı kullanarak her farklı girdi bloğu için farklı bir çıktı oluşturulmaktadır.



# Blok Şifreler

- Blok şifreler, açık metni eşit uzunluktaki bloklara ayırıp, her bir bloğu bir fonksiyon yardımı ile tek tek şifreleyerek şifreli metni oluşturur.



# Kriptoloji Eğitimi

- Modern blok şifre sistemlerinin nasıl inşa edildiğini ve güvenlik özelliklerini anlamak için klasik şifre sistemleri ve analizleri çalışmak gerekir.
- Kitaplar:
  - Kriptografiye Giriş Notları, ODTÜ Uygulamalı Matematik Enstitüsü
  - **Kitap:** Şifrelerin Matematiği: Kriptografi
  - **Kitap:** Cryptography, An Introduction : Third Edition
  - **Kitap:** Cryptography: Theory and Practice, Third Edition
  - **Kitap:** Handbook of Applied Cryptography
  - **Kitap** “Cryptography and Network Security” 5. baskı, Yazar: W. Stallings

Eğitim Yazılımları: Cryptool Projesi Cryptool1, Cryptool2, Jcryptool

Daha fazlası: Seminer:"Kriptoloji Eğitimi için Yazılımlar ve Projeler"

- **Y. Lisans ve Doktora seviyesinde Eğitim için:**  
**ODTÜ Uygulamalı Matematik Enstitüsü Kriptografi Programı**

# Klasik Kriptografik Sistemler

---

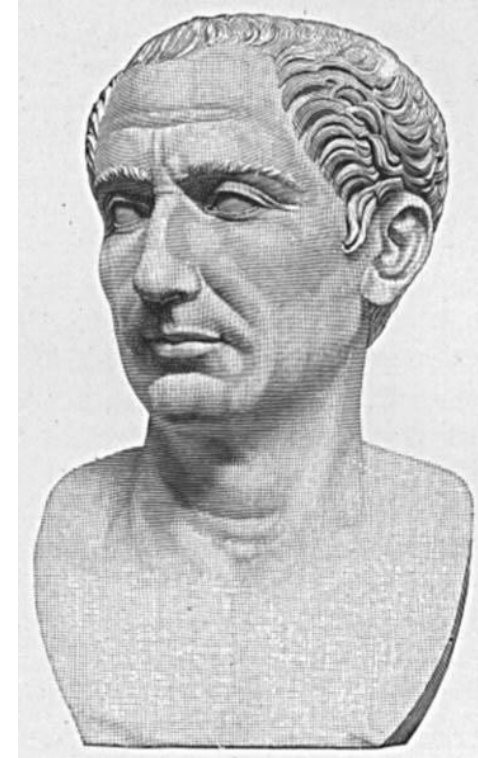
## ➤ Sezar Şifresi (M.Ö. 100-44)

- Harflerin alfabede  $K = 3$  konum sonrasındaki karşılığı ile değiştirir

- $C = P + K \pmod{26}$

- MERHABA → PHUKDED

- Kriptanalizi istatistiksel analizle kolayca yapılmaktadır



a b c d e f g h i j k l m n o p q r s t u v w x y z

D E F G H I J K L M N O P Q R S T U V W X Y Z A B C



# Klasik Kriptografik Sistemler

- Enigma 1923 (Arthur Scherbius, 1878-1929)
  - Dönen silindir ile açık metnin her harfi yeni bir permütasyonla şifrelenir.
  - 2.Dünya Savaşında 200.000'den fazla sayıda Enigma makinesi üretilmiştir.
  - Kriptanalizine Polonyalılar önderlik etmiştir
  - Daha sonra, İngiltere'de yaklaşık 7000 kişinin çabaları sonucu, Enigma ile şifrelenen metinler çözülmüştür.
  - Çözülmesi savaşın bitmesinde önemli rol oynamıştır



# Örnek Temel/Klasik Blok Şifre

---

$$E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

- $k = 2$ -bit ve  $n = 3$ -bit diyelim,
- $(2^n)! = 8! = 40320$  olası permütasyon'dan sadece  $2^k = 2^2 = 4$  tanesi kullanılıyor
- Bu blok şifrenin dönüşümleri aşağıdaki gibi olsun :

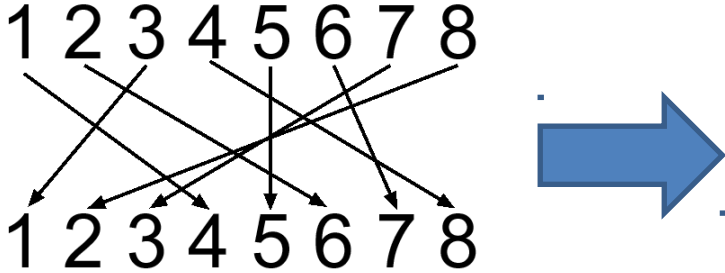
$$k=00, p_0=(4,6,1,8,5,7,3,2)$$

$$k=01, p_1=(5,2,7,1,8,6,4,3)$$

$$k=10, p_2=(8,6,2,1,3,4,5,7)$$

$$k=11, p_3=(3,8,6,2,4,7,5,1)$$

# Örnek Temel/Klasik Blok Şifre

	Adres	Açık metin	Şifre metin
<u>ŞİFRELEME</u>	1	001	100
E: $\{0,1\}^2 \times \{0,1\}^3 \rightarrow \{0,1\}^3$	2	010	110
$k=00, p_0=(4,6,1,8,5,7,3,2)$	3	011	001
	4	100	000
X=011'i şifreleyelim.	5	101	101
X'in adresi 3	6	110	111
$Y=E_k(X)=E_k(3)=001$	7	111	011
	8	000	010

# Örnek Temel/Klasik Blok Şifre

Adres    Şifre metin    Açık metin

1    001    011

2    010    000

3    011    111

4    100    001

5    101    101

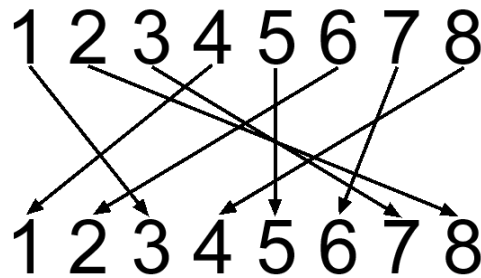
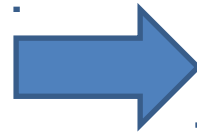
6    110    010

7    111    110

8    000    100

## ŞİFRE ÇÖZME

$$P_0' = \begin{pmatrix} 12345678 \\ 38715264 \end{pmatrix}$$



Y=001' i çözelim

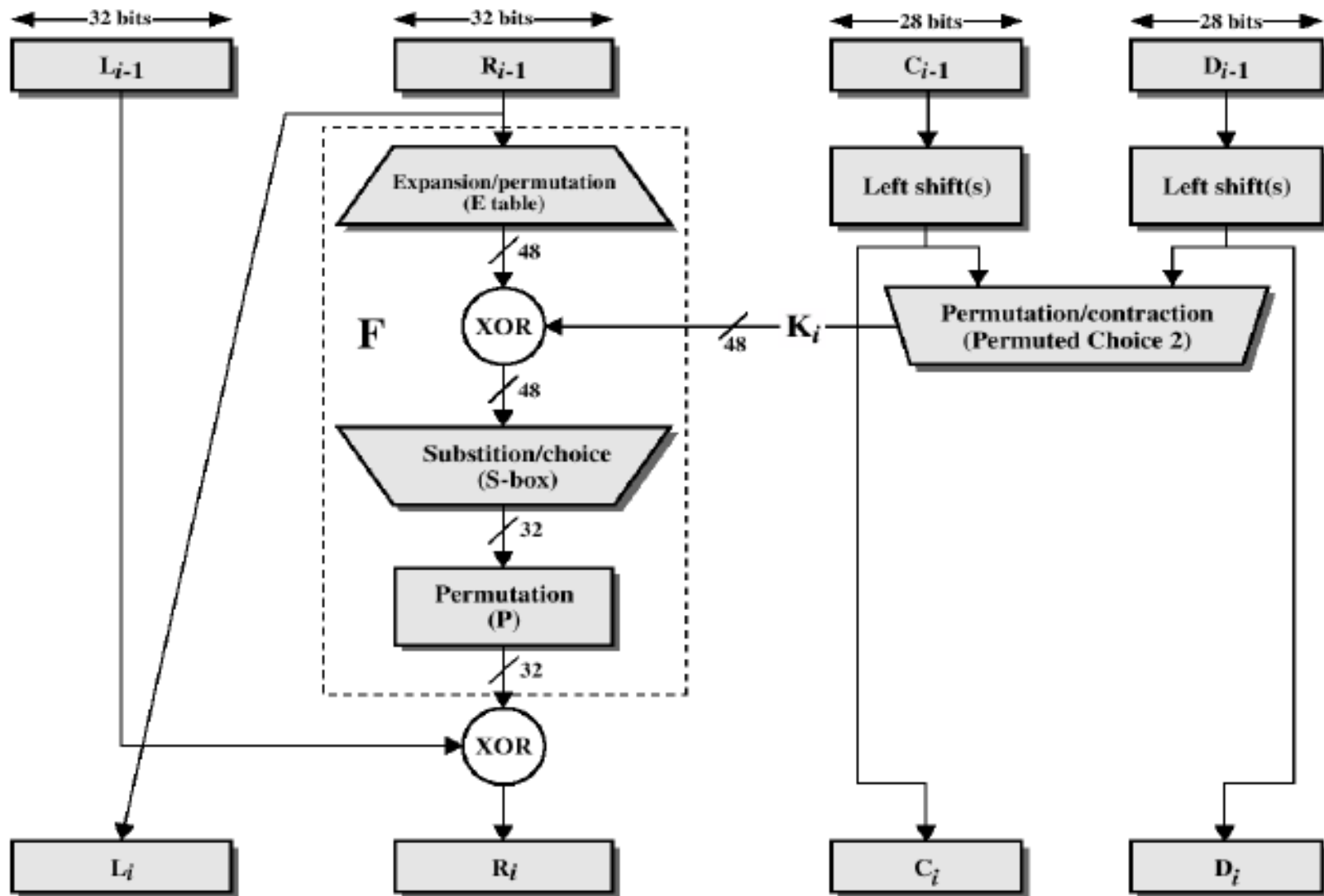
Y'nin adresi 1.

$$E_k^{-1}(Y) = E_k^{-1}(1) = 011 = X.$$

# Blok Şifreler - Standartlaştırma

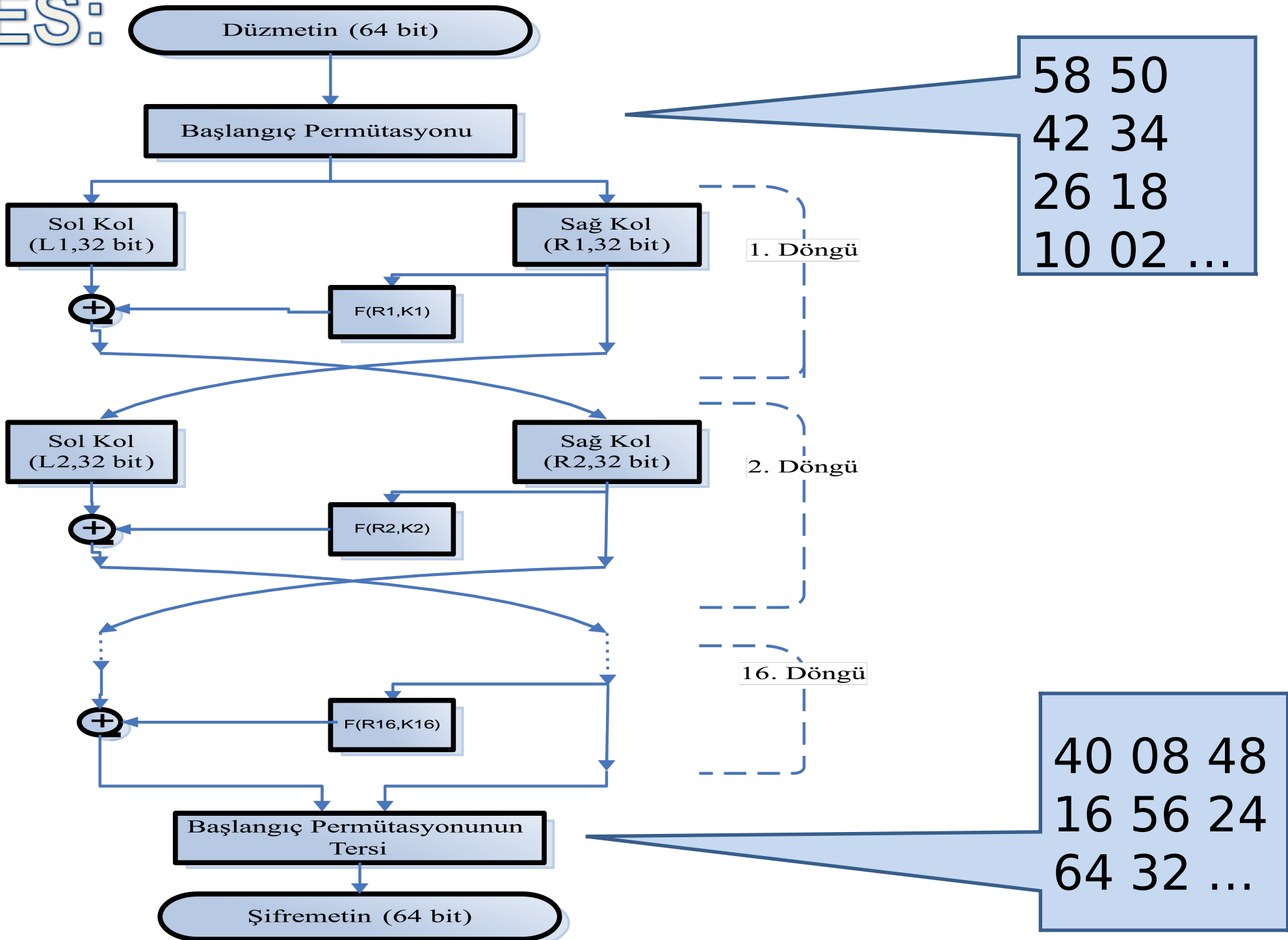
---

- Amerikan Milli Standartlar Bürosu (NBS) bilgi güvenliğini sağlamak için bir şifreleme algoritması geliştirmek amacıyla 1973'de proje başlattı.
- 1974'te IBM tarafından finansal uygulamalar için geliştirmiş bir şifreleme ailesi (LUCIFER) duyuruldu.
- NBS tarafından geliştirildi ve 1977'de NBS ilk standart Data Encryption Standard (DES)"i Federal Information Processing Standard (FIPS 46) olarak duyurdu.



# DES Döngüsü

# DES:



# DES Güvenliği

- DES gizli anahtarı 56-bit
- Kaba kuvvet saldırısına (**brute force attack**) karşı dayanıksız

*$D(K\text{-aday}, C)$  = anlamlı açık metin mi?*

*D: deşifreleme, C: kapalı/şifreli metin*

- Ortalama *K-aday*  $2^{55}$  DES gizli anahtarlarını deneme yolu ile şifrelenmiş bir metni deşifre ederek kullanılan gizli anahtarı (anlamlı bir açık metin veren) bulunma çalışması
- 1998 yılında Electronic Frontier Foundation (EFF), 250,000 Dolar maliyet ile
  - 2006 yılında yapılan COPACOBANA (Paralel çalışan 120 FPGA) ile bu işlemi ortalamada dokuz günde yapmak mümkün oldu.

[http://en.wikipedia.org/wiki/EFF\\_DES\\_cracker](http://en.wikipedia.org/wiki/EFF_DES_cracker)

[http://en.wikipedia.org/wiki/Data\\_Encryption\\_Standard#Security\\_and\\_cryptanalysis](http://en.wikipedia.org/wiki/Data_Encryption_Standard#Security_and_cryptanalysis)



# DES Güvenliği

- Şifre sisteminin “Hesaplamaya karşı güvenli” (computationally secure) olması demek
  - bu sistemi kırmak maliyetinin şifrelenen verinin değerini aşması veya
  - bu sistemi kırmak için harcanan zamanın şifrelenen verinin değerli kullanım ömrünü aşmasını
- DES (56-bit anahtarlı) “hesaplamaya karşı güvenli” değil çünkü anahtar uzunluğu kısa.
- Bu yüzden modern simetrik şifre algoritmaları 128- den 256-bit anahtarlar kullanırlar.

# Anahtar Uzunlukları

- Kriptografik protokollerde bir takım kriptografik algoritmalar beraber kullanılmaktadır.
- Kaba-kuvvet saldırılarına dayanıklılık için günümüz hesaplama yetenekleri düşünülerek

bazı standartlar, organizasyonların ve projelerin anahtar uzunlukları için tavsiyede bulunurlar:

“Algorithms, Key Sizes and Parameters Report2013 recommendations”, ENISA  
European Network of Excellence in Cryptology II (AB projesi) Tavsiyeleri

NIST

Level	Protection	Symmetric	Asymmetric	Discrete Key	Logarithm Group	Elliptic Curve	Hash
1	Attacks in "real-time" by individuals <i>Only acceptable for authentication tag size</i>	32	-	-	-	-	-
2	Very short-term protection against small organizations <i>Should not be used for confidentiality in new systems</i>	64	816	128	816	128	128
3	Short-term protection against medium organizations, medium-term protection against small organizations	72	1008	144	1008	144	144
4	Very short-term protection against agencies, long-term protection against small organizations <i>Smallest general-purpose level, 2-key 3DES restricted to <math>2^{40}</math> plaintext/ciphertexts, protection from 2014 to 2015</i>	80	1248	160	1248	160	160
5	Legacy standard level <i>2-key 3DES restricted to <math>10^6</math> plaintext/ciphertexts, protection from 2014 to 2020</i>	96	1776	192	1776	192	192
6	Medium-term protection <i>3-key 3DES, protection from 2014 to 2030</i>	112	2432	224	2432	224	224
7	Long-term protection <i>Generic application-independent recommendation, protection from 2014 to 2040</i>	128	3248	256	3248	256	256
8	"Foreseeable future" <i>Good protection against quantum computers, unless Shor's algorithm applies</i>	256	15424	512	15424	512	512

Date	Minimum of Strength	Symmetric Algorithms	Asymmetric	Discrete Logarithm		Elliptic Curve	Hash (A)	Hash (B)
				Key	Group			
2010 (Legacy)	80	2TDEA*	1024	160	1024	160	SHA-1** SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
2011 - 2030	112	3TDEA	2048	224	2048	224	SHA-224 SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
> 2030	128	AES-128	3072	256	3072	256	SHA-256 SHA-384 SHA-512	SHA-1 SHA-224 SHA-256 SHA-384 SHA-512
>> 2030	192	AES-192	7680	384	7680	384	SHA-384 SHA-512	SHA-224 SHA-256 SHA-384 SHA-512
>>> 2030	256	AES-256	15360	512	15360	512	SHA-512	SHA-256 SHA-384 SHA-512

# Parola Güvenliği

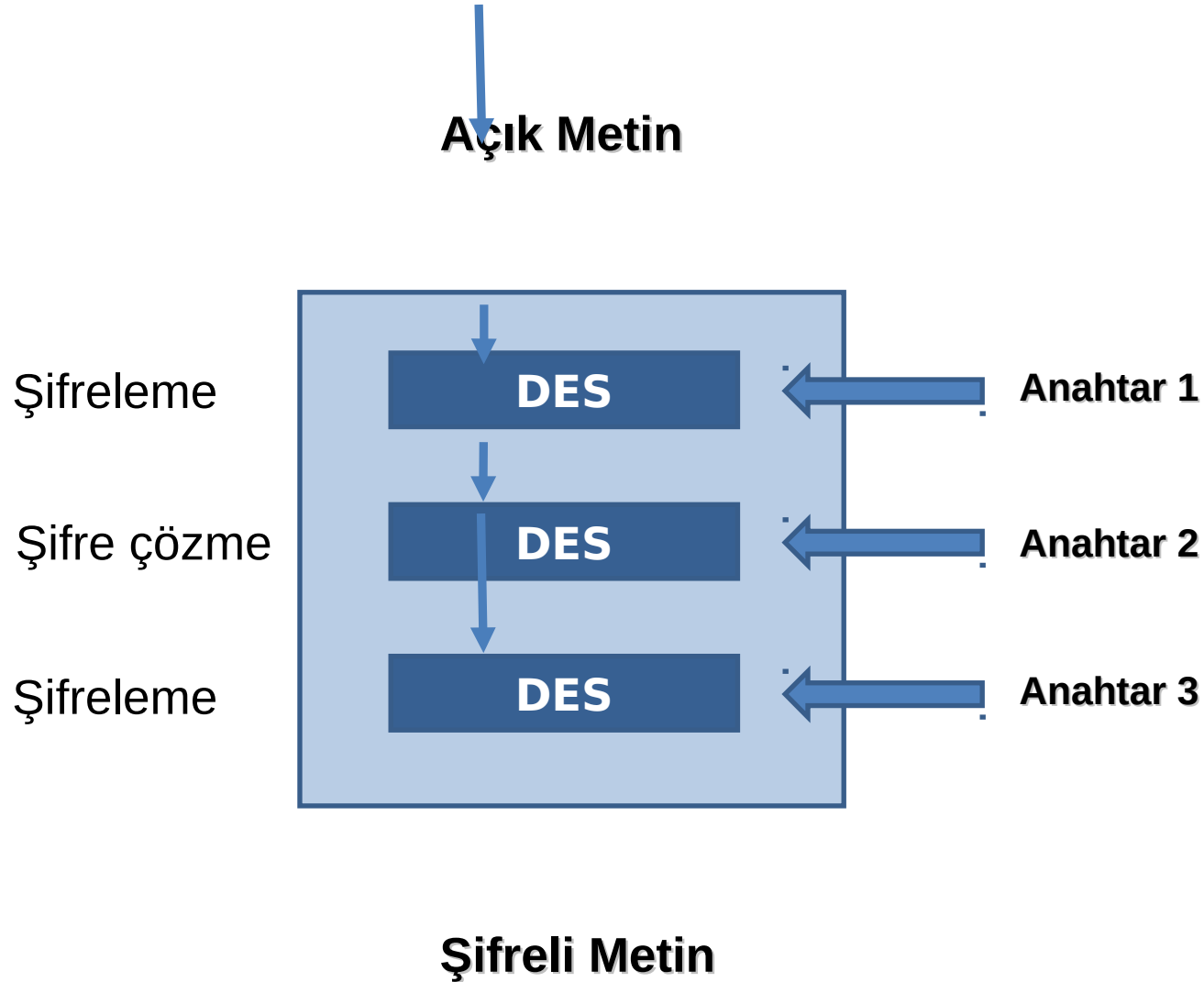
- Parolalar oluşturulurken özet algoritmaları kullanılıyor.
- Parola için tavsiye edilen uzunluk en az 16 karakter. Her bir karakter 8-bit. Toplamda en az 128-bit. Karakterler için harfler, sayılar, semboller rastgele seçilmeli.
- **Cryptool parola üretici** (Eğitim amaçlı)
- **Parolaları test** yazılımları

John the Ripper

ophcrack

Benzerleri..

# Bilgi Şifreleme Standardı(3DES)



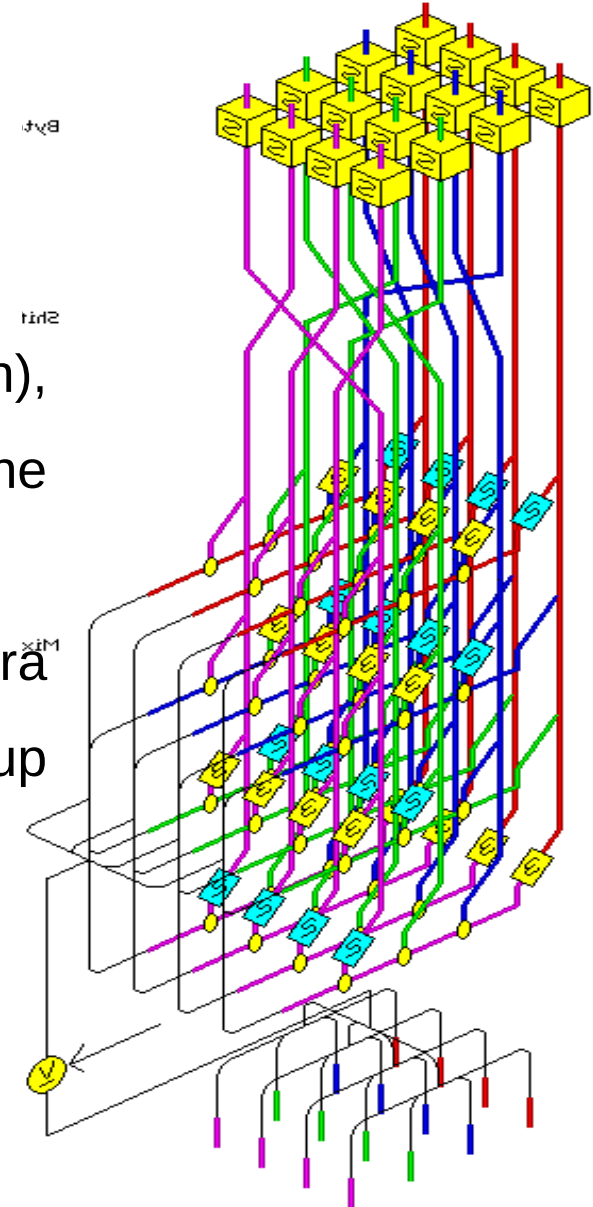
3DES anahtar uzunluğu  $56 \cdot 2 = 112$ -bit veya  $56 \cdot 3 = 168$ -bit

# Blok Şifreler – Standartlaştırma

## Gelişmiş Şifreleme Standardı

AES (Advanced Encryption Standard )

- NIST 1997'de yarışma açar
- Rijndael Algoritması (Joan Daemen, Vincent Rijmen), 2001 yılında 5 finalist algoritma arasından DES'in yerine standart olarak atanmıştır.
- DES'in zayıf yönlerinden hareketle bilinen tüm ataklara karşı önlemler alınmış, kolayca anlaşılabilir yapıda olup birçok ortamda çalışabilecek düzeydedir.



# AES (Advanced Encryption Standard )

- Üç tane AES blok şifre sistemi: AES-128, AES-192 and AES-256
- Her bir AES blok şifre sistemi 128-bit blok uzunluğu ve 128, 192 ve 256 bit anahtar uzunluklarına sahiptir.
- DES şifre sisteminin tersine, AES sistemleri kriptografi akademik topluluk tarafından incelendi.
- AES-128 blok şifre sistemi animasyonunu:

Video <http://www.youtube.com/watch?v=mlzxpkdXP58>

Flash

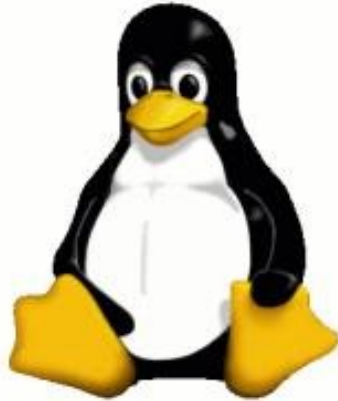
[http://www.cs.bc.edu/~straubin/cs381-05/blockciphers/rijndael\\_ingles2004.swf](http://www.cs.bc.edu/~straubin/cs381-05/blockciphers/rijndael_ingles2004.swf)



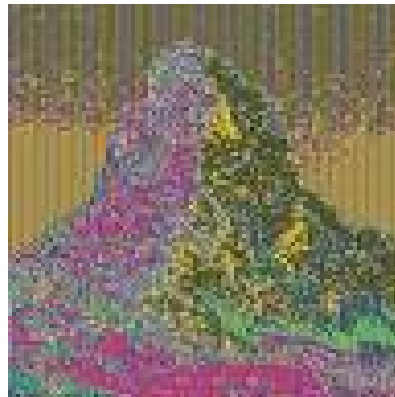
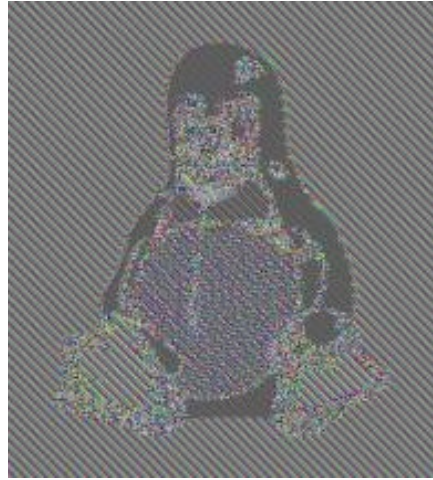
# Blok Şifreler - Çalışma

## Modları

Orjinal resim



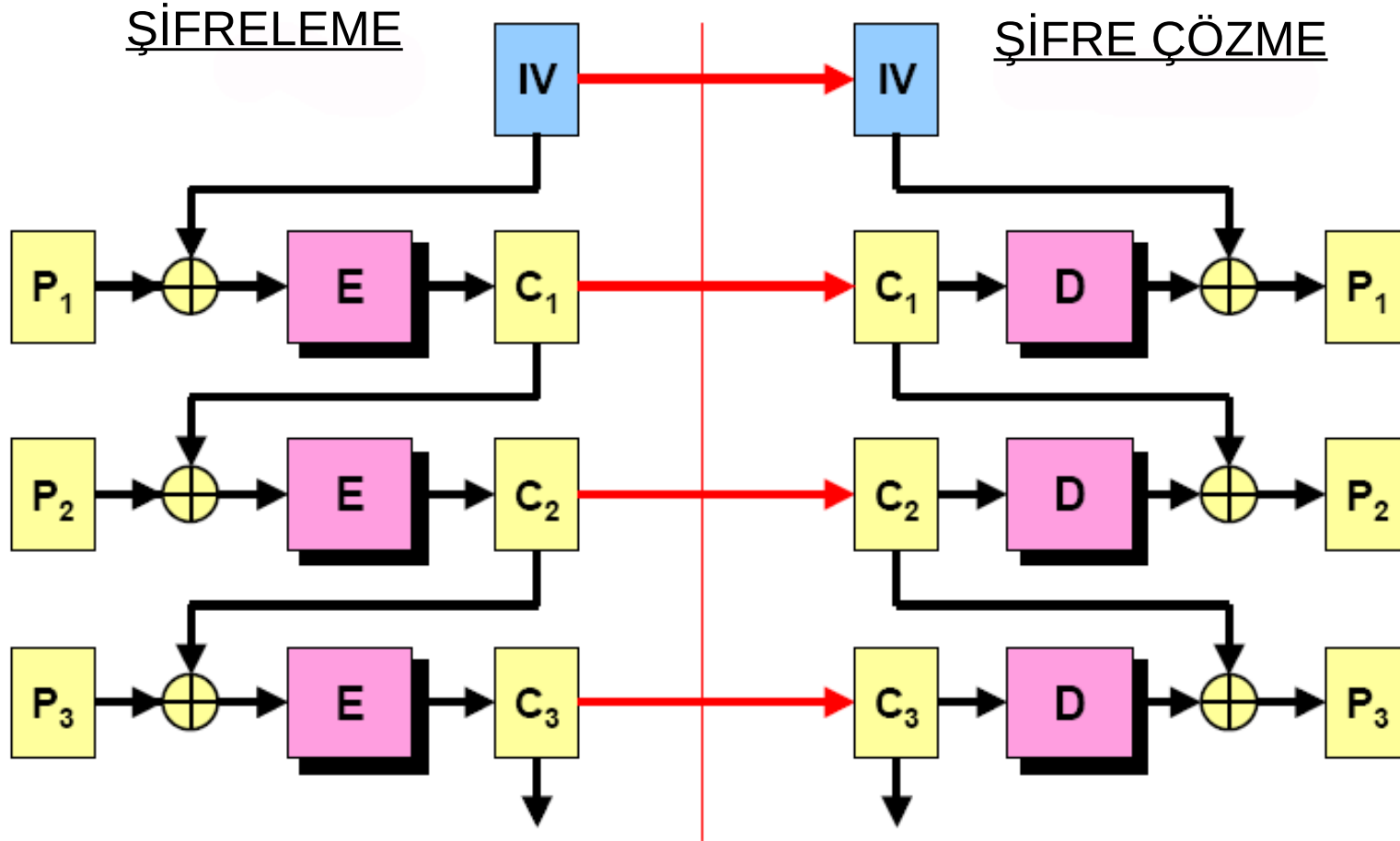
Elektronik Kod Modu \*



➤ Açık metinde aynı olan bloklar aynı anahtar kullanılarak şifrelendiğinde aynı şifreli metni verir.

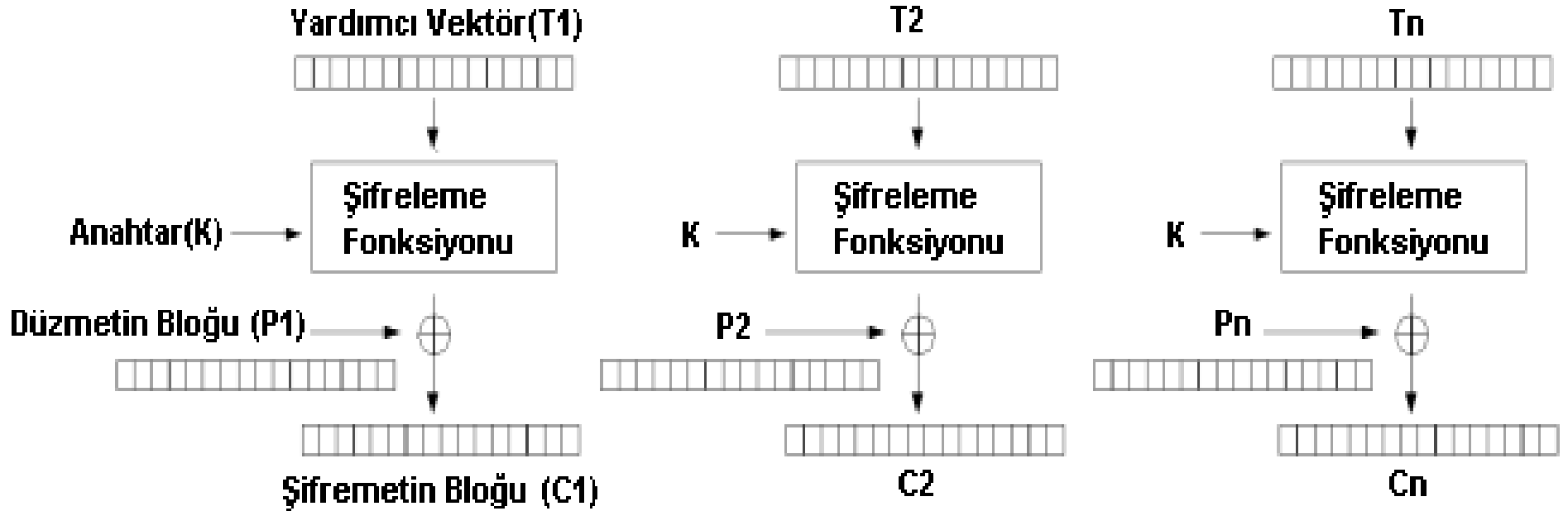
➤ İstatistiksel olarak açık metin hakkında bilgi veren bu durumu ortadan kaldırmak için çeşitli çalışma modları önerilmiştir.

# Blok Şifreler - Çalışma Modları



Kapalı Metin Zincirleme Modu  
CBC (Cipher-Block Chaining )

# Blok Şifreler - Çalışma Modları



**Karşılık Modu**  
CTR (Counter)

# Blok Şifreler - Çalışma Modları

---

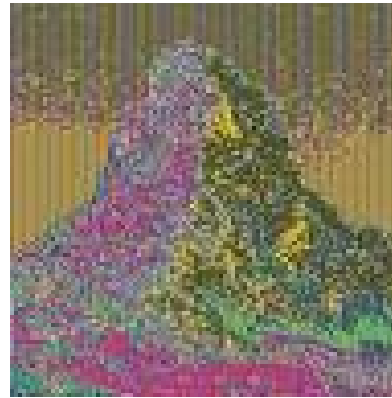
Orjinal resim



ECB modu



CTR modu



# Blok Şifreler - Çalışma Modları

---

- Daha fazlası için
  - [http://en.wikipedia.org/wiki/Block\\_cipher\\_modes\\_of\\_operation](http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation)
  - [http://en.wikipedia.org/wiki/ISO/IEC\\_10116](http://en.wikipedia.org/wiki/ISO/IEC_10116)
- Blok şifre sistemleri diğer kriptografik algoritmaları tasarlamak için kullanabiliriz: Özet fonksiyonları, Akan Şifre sistemleri, MAC algoritmaları vb.

## AES için NIST nin tavsiye ettiği modlar :

- Kimlik Denetimli Şifreleme,
- Kimlik Denetimi,
- Şifreleme için.

## Disk Şifreleme için özel modlar mevcut:

Tweakable narrow-block encryption modes (LRW, XEX ve XTS) ve de Güvenli şekilde disk sektörlerini şifrelemek için CMC ve EME modları

# Simetrik (Gizli) Anahtarlı Sistemler

---

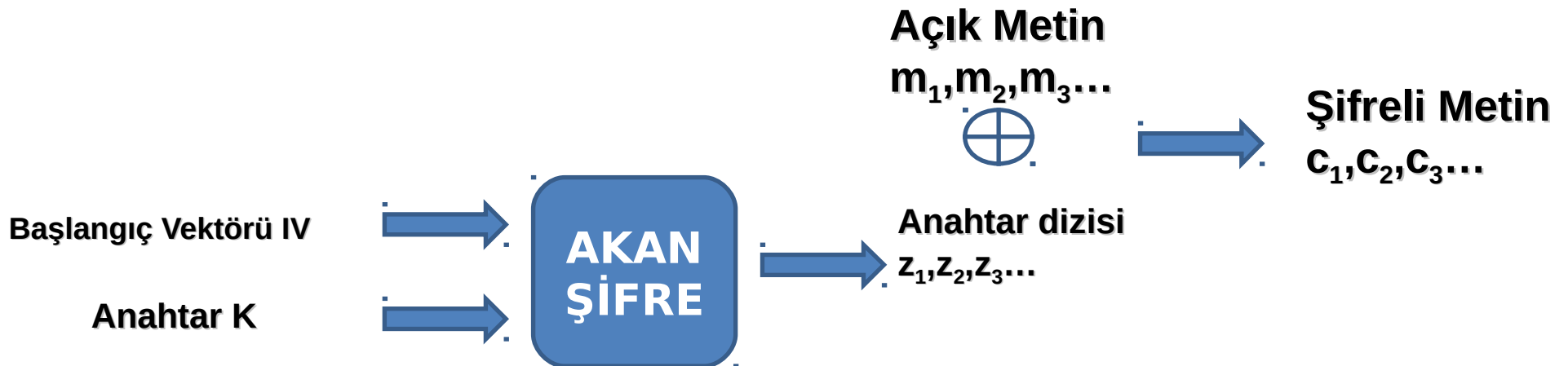
DES, AES, IDEA,  
KASUMI, SAFER



A5/1 (GSM),  
RC4 (WEP),  
E0(Bluetooth)

# Akan Şifreler

- Akan Şifreler işlevini açık metnin her bir karakterini zamanla değişen bir fonksiyona sokarak yerine getirir .
- Girdi olarak alınan bir anahtar (K) ve başlangıç vektörü (IV) ile üreteç mümkün olduğu kadar uzun periyotlu ve rastgele gözüken anahtar dizilerini ( $z_1, z_2, z_3, \dots$ ) üretir ve elde ettiği anahtarı açık metinle şifreleme fonksiyonuna (h) sokarak şifreli metni elde eder.



# Akan Şifreler - Çeşitleri

---

## Kullan-At (One Time Pad)

- 1917'de Gilbert Vernam ve Major Joseph Mauborgne tarafından bulunmuştur.
- 1948'de Shannon, eğer anahtar rastlantısal ve bir kereliğine kullanılırsa bu şifrenin kırılmadığını kanıtladı
- Açık metin, sahip olduğu uzunluğa eşit ve rastgele oluşturulan bir anahtar ile şifreli metine dönüştürülür.
- Mükemmel gizlilik sağlar ama anahtar uzunluğu açık metin ile aynı uzunlukta olduğundan pratik değildir.



# Örnek Kullan-At Şifreleme

## ŞİFRELEME



Anahtar  $K = 010011010001$  (rastgele ve tek kullanımlık)

Açık metin  $P = 101010111010$

Şifreli metin  $C = 111001101011$

$\oplus$	0	1
0	0	1
1	1	0

## ŞİFRE ÇÖZME

Şifreli metin  $C = 111001101011$

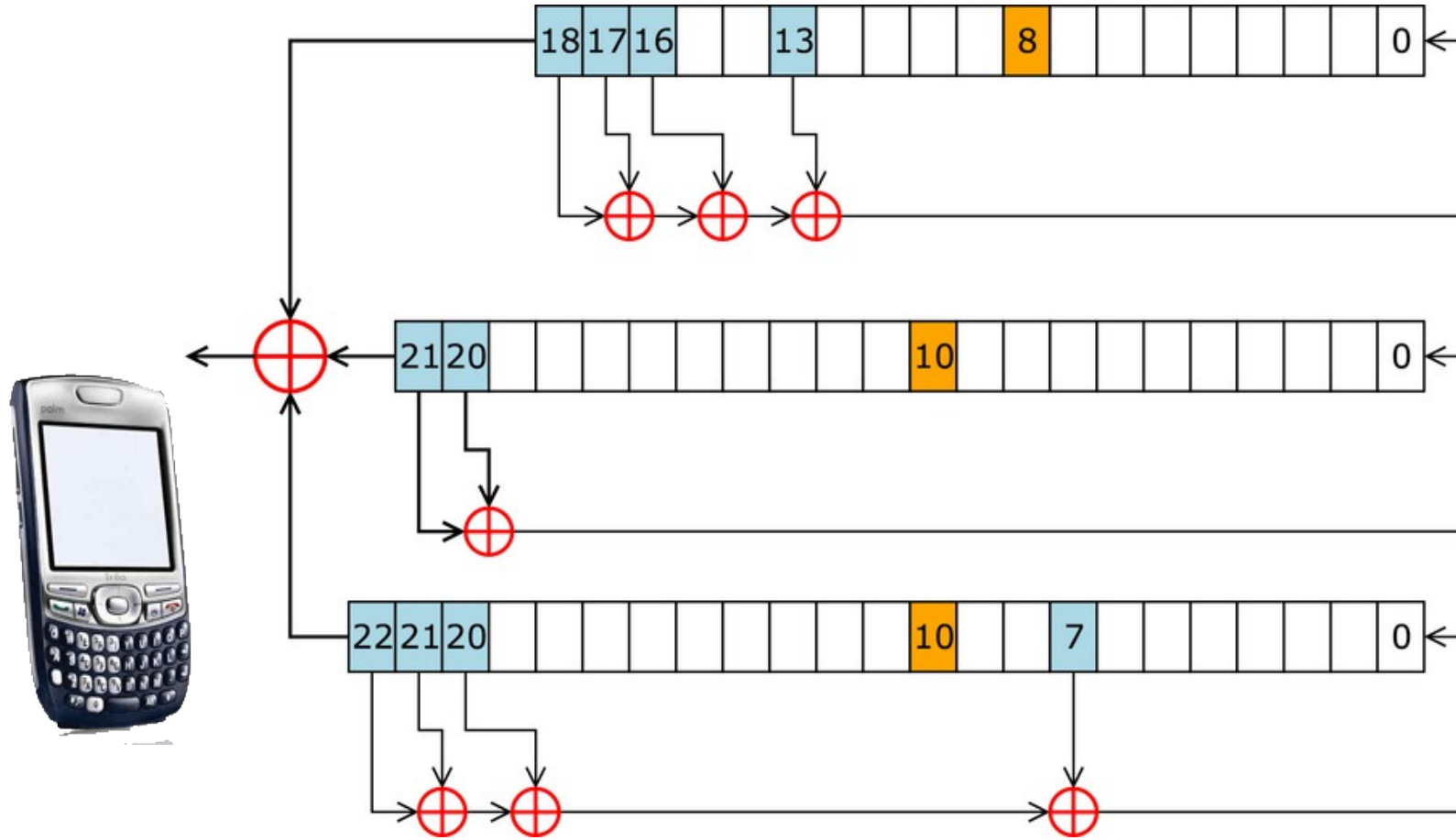
Anahtar  $K = 010011010001$  ( $2^{12}$  olası anahtar)

Açık metin  $P = 101010111010$

# Akan Şifreler - Çeşitleri



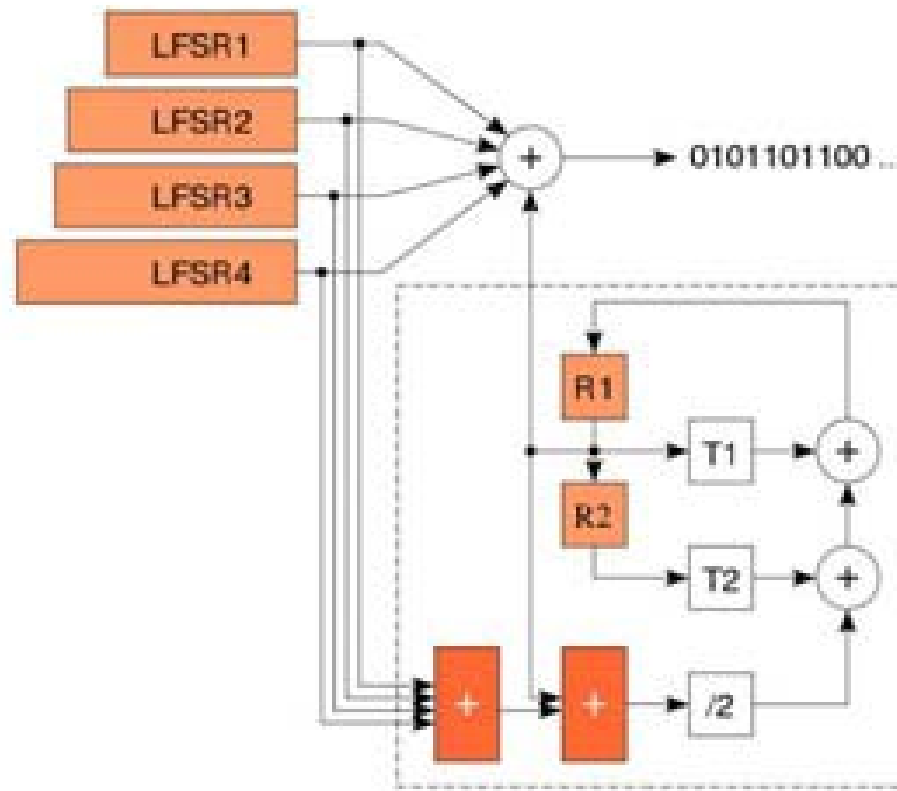
## A5/1 (GSM)



A5/1 hava kanalı üzerinden ses şifrelemesinde kullanılan bir şifreleme algoritmasıdır

# Akan Şifreler - Çeşitleri

E0 (  Bluetooth® )



# Akan Şifrelerin Gelişimi

---

- Son 5 yıl içerisinde büyük bir gelişim gösterdi
- Donanım için uygun ve hızlı algoritmalar mevcut
- Standart ve açık çözümler konusunda eksikler var
- ECRYPT'in 2004'te açtığı yarışma eStream:
  - 34 aday algoritma'dan Eylül 2008 itibariyle 7 tanesi kullanılabilir olarak seçildi ancak bunların standart olmaları için henüz erken olduğu belirtilmektedir.

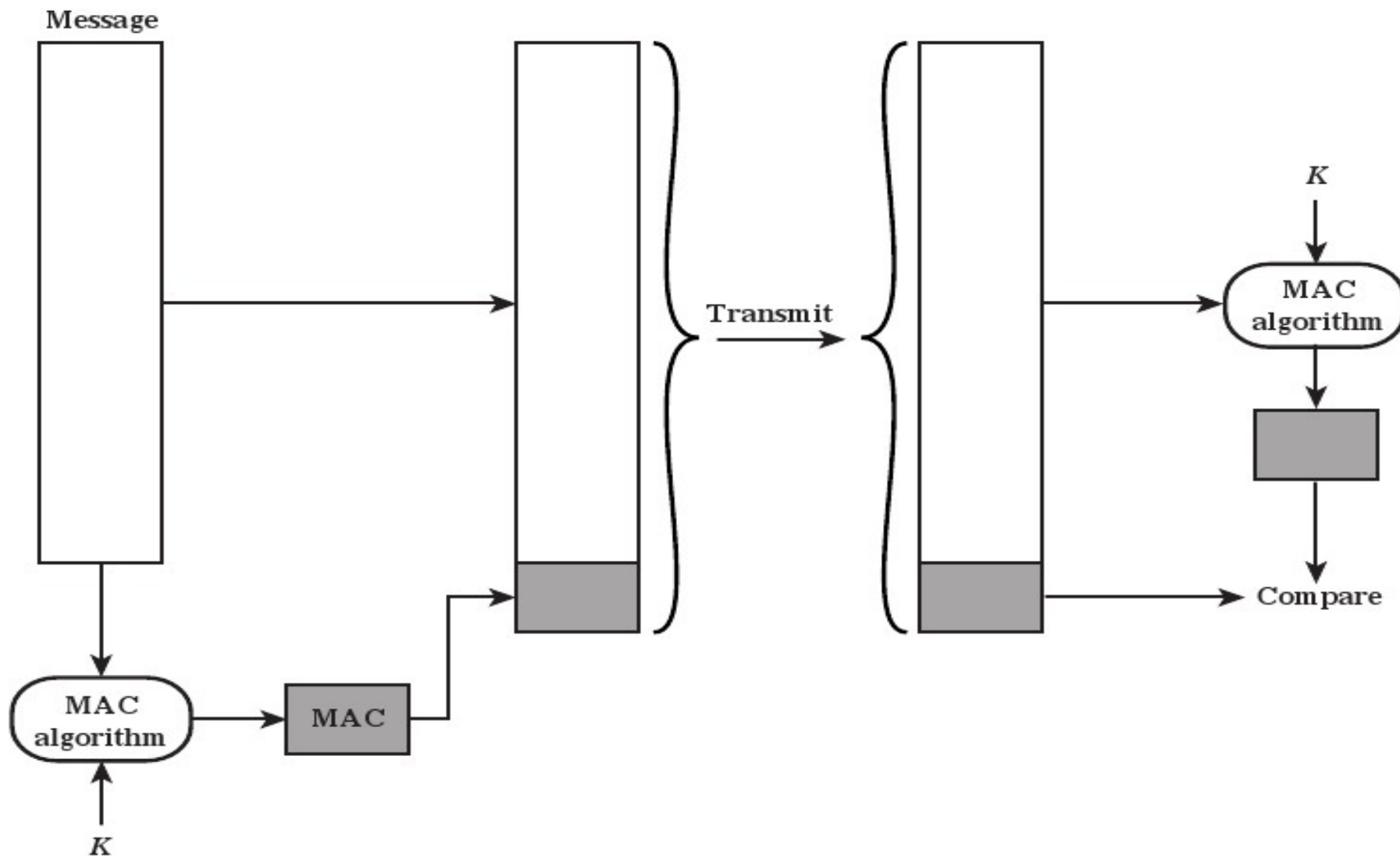
# Mesaj Kimlik Doğrulama Kodu (MAC)

Mesaj Kimlik Doğrulama Kodu (Message Authentication Code (MAC) ) algoritması, bir gizli anahtar kullanır ve verilen mesaj için küçük bir veri oluşturur. Bu veri, mesajın sonuna eklenir.

Varsayım: A ve B kullanıcıları ortak ve gizli anahtar  $K_{AB}$  üzerinde anlaşsınlar.

$$MAC_M = F(K_{AB}, M)$$

# Mesaj Kimlik Doğrulama Kodu (MAC)



06/ Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)

# Mesaj Kimlik Doğrulama Kodu (MAC)

- Mesaj Kimlik Doğrulama Kodu sayesinde mesajı alan, mesajın değiştirilmediğine ikna olur.
- Mesajı alan, mesajın beklenen kişiden geldiğine de ikna olur (yerine geçme / taklit etme saldırısına önlem)

# Mesaj Kimlik Doğrulama Kodu (MAC)

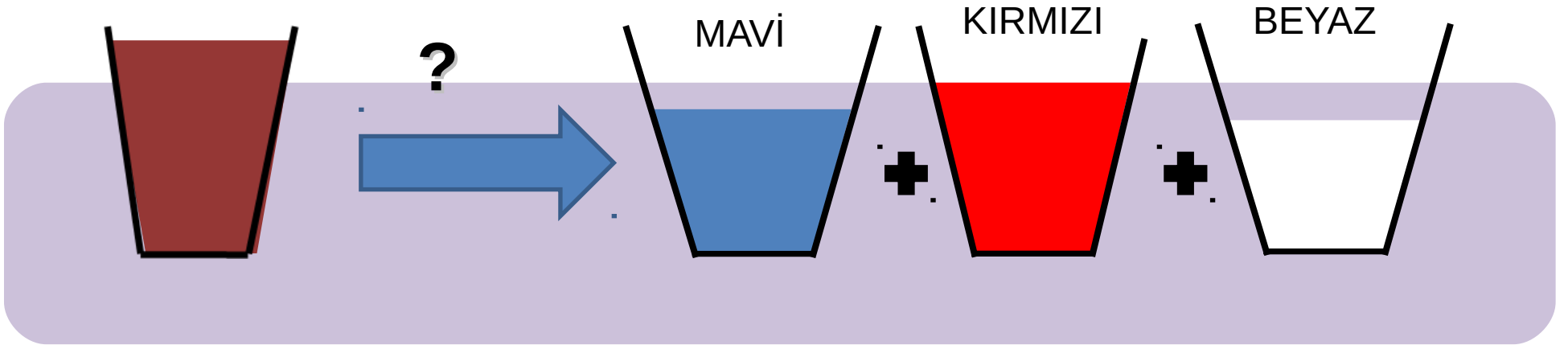
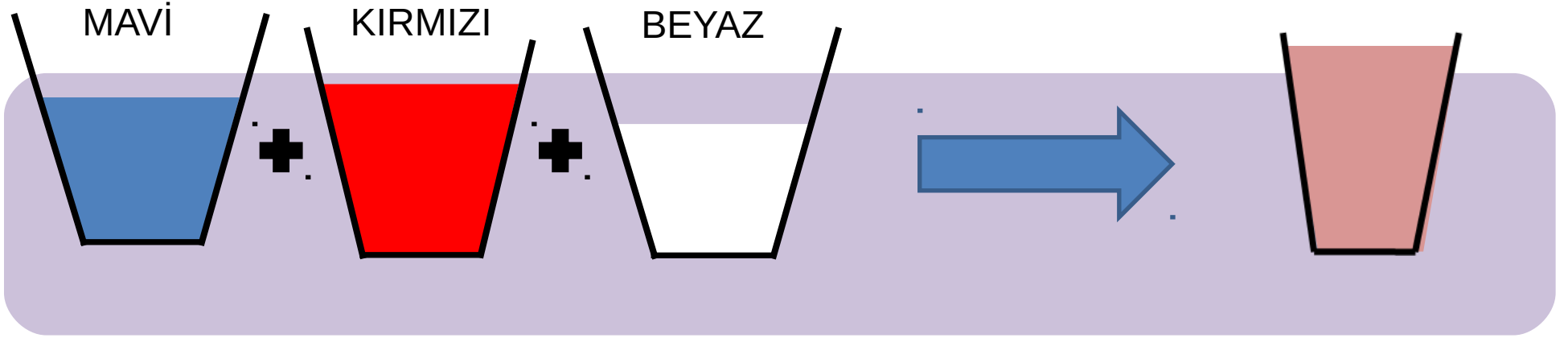
- Blok Şifre Sistemleri kullanılarak MAC algoritmaları oluşturulmaktadır.
- MAC algoritmasının tersi yoktur.
- Bir çeşit girdi mesajına ve ayrıca gizli anahtara özgü özet çıktı olarak verir.
- Alternatifi, kriptografik özet fonksiyonlarıdır.



# Tek yönlü Özet Fonksiyonu

- **Özet Fonksiyonu Hash function** değişken uzunlukta bir mesajı girdi olarak alıp, çıktı olarak sabit uzunlukta  $H(M)$  (parmak izi, özet) yi üretir.
- **MAC algoritmaları** gibi gizli bir anahtar girdilerden biri olmaz
- Mesajın parmak izi, mesaj ile birlikte kimlik (mesajın kaynağını) doğrulama için gönderilir

# Özet Fonksiyonlar



# Özet Fonksiyonlar

---

$$h: \{0,1\}^* \rightarrow \{0,1\}^n$$

❖ h fonksiyonu, herhangi bir uzunluktaki açık metni alıp sabit uzunlukta bir çıktı verir.

❖ Büyük bir tanım kümesinden sabit görüntü kümesine çoktan-bire eşlemedir.

h: Açık Metin  $\rightarrow$  Özet

❖ Bu nedenle aynı özete sahip metinler bulunabilir.

Temel Özellikleri:

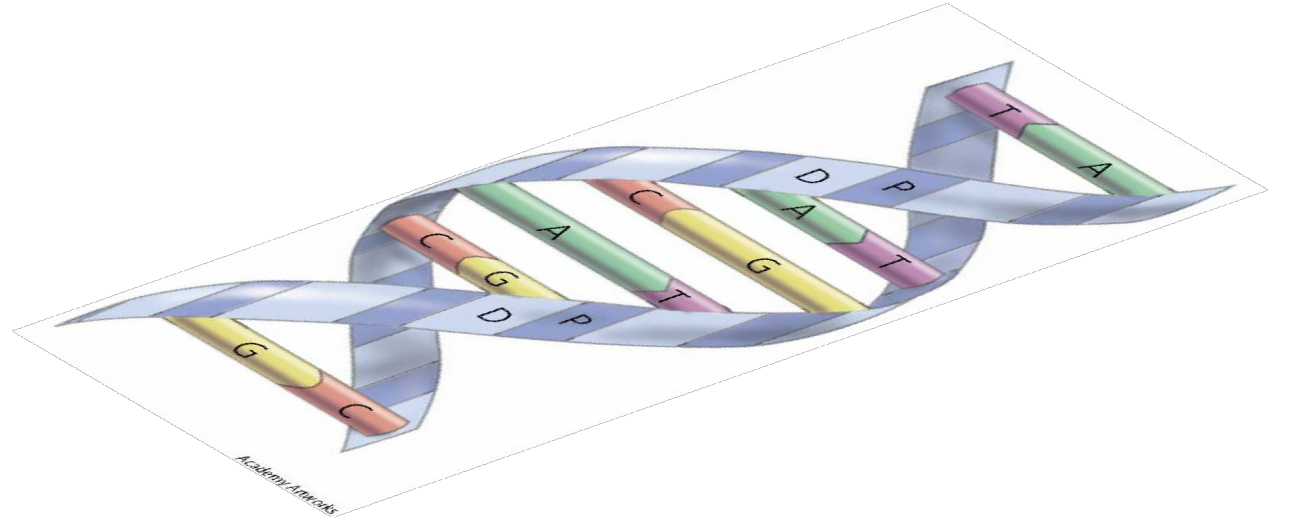
Sıkıştırma

Hesaplama kolaylığı

# Özet Fonksiyonlar - Güvenlik Kriterleri

---

- Bir açık metnin özet fonksiyon değeri o metnin parmak izi veya DNA'sı gibi olmalıdır.



# Kriptografik Özet Fonksiyonu

Bu sunu dosyasında “özet fonksiyonu” ile “kriptografik özet fonksiyonu” kastedilmektedir.

“İdeal bir kriptografik özet fonksiyonu şu dört özelliği sağlamalıdır:

- + Herhangi bir mesaj için özet hesaplamak kolay olmalıdır.
- + Bir özete karşılık gelecek mesajı oluşturmak (hesaplama karşı imkansız) zor olmalıdır.
- + Özeti değiştirmeyecek şekilde mesajı değiştirmek (hesaplama karşı imkansız) zor olmalıdır.
- + Aynı özete sahip iki farklı mesaj bulmak zor olmalıdır.”

Kriptografik özet fonksiyonun sağlaması gereken özellikler fazlası için:

[http://tr.wikipedia.org/wiki/Kriptografik\\_%C3%B6zet\\_fonksiyonu](http://tr.wikipedia.org/wiki/Kriptografik_%C3%B6zet_fonksiyonu)

[http://en.wikipedia.org/wiki/Cryptographic\\_hash\\_function](http://en.wikipedia.org/wiki/Cryptographic_hash_function)

(İngilizce)

“ Kriptografik özet fonksiyonları, bilgi güvenliği konuları olan sayısal imza, mesaj doğrulama kodu ve diğer doğrulama yöntemlerinde yaygın olarak kullanılmaktadır. “

# Kriptografik Özet Fonksiyonu

## Çakışma Saldırısı (Collision attack)

özet(m1) = özet(m2) durumunu sağlayan İki farklı mesaj m1 and m2 bulunmasıdır.

Bu durumda, bu tür bir özet fonksiyonu güvenilirliği kaybeder.

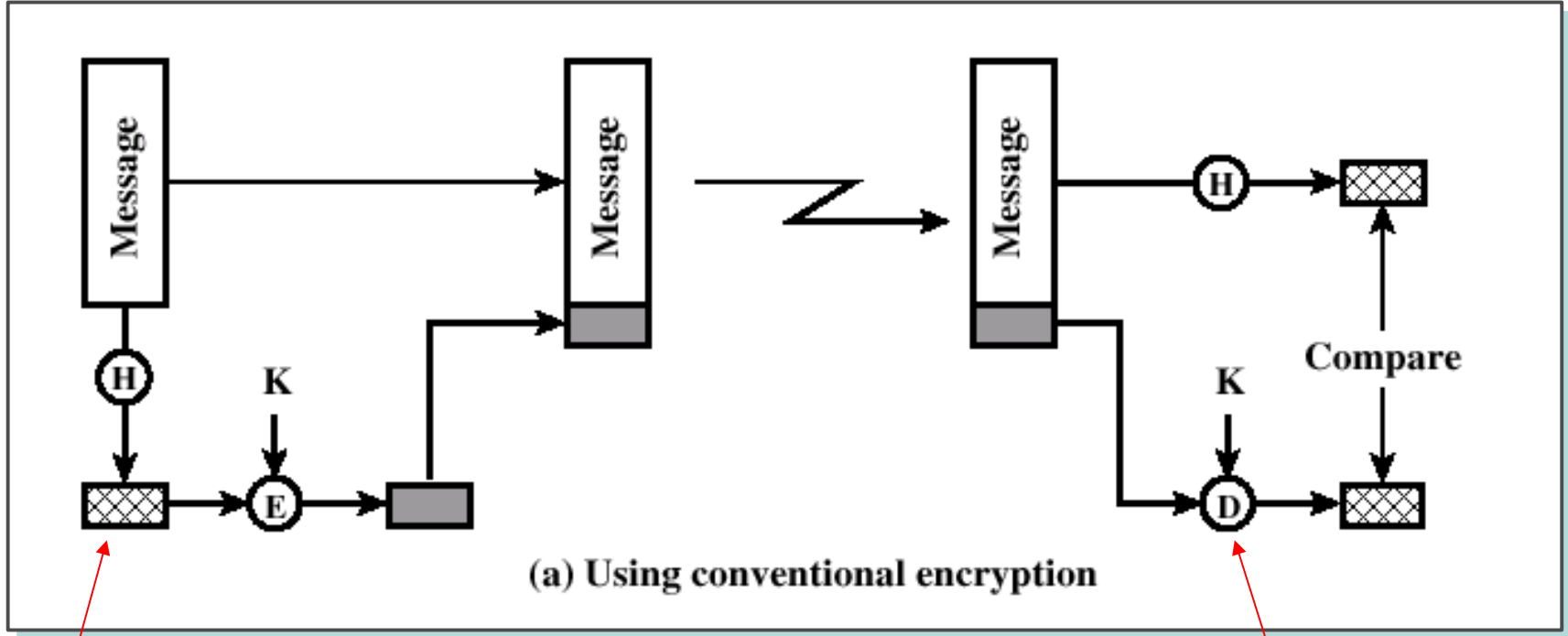
MD5 için bu tür mesajlar bulunmuştur. Yani bu saldırıya karşı MD5 dayanıklı değildir

(<http://en.wikipedia.org/wiki/MD5>)

Bakınız:

[http://en.wikipedia.org/wiki/MD5#Collision\\_vulnerabilities](http://en.wikipedia.org/wiki/MD5#Collision_vulnerabilities)

# Tek yönlü Özet Fonksiyonu

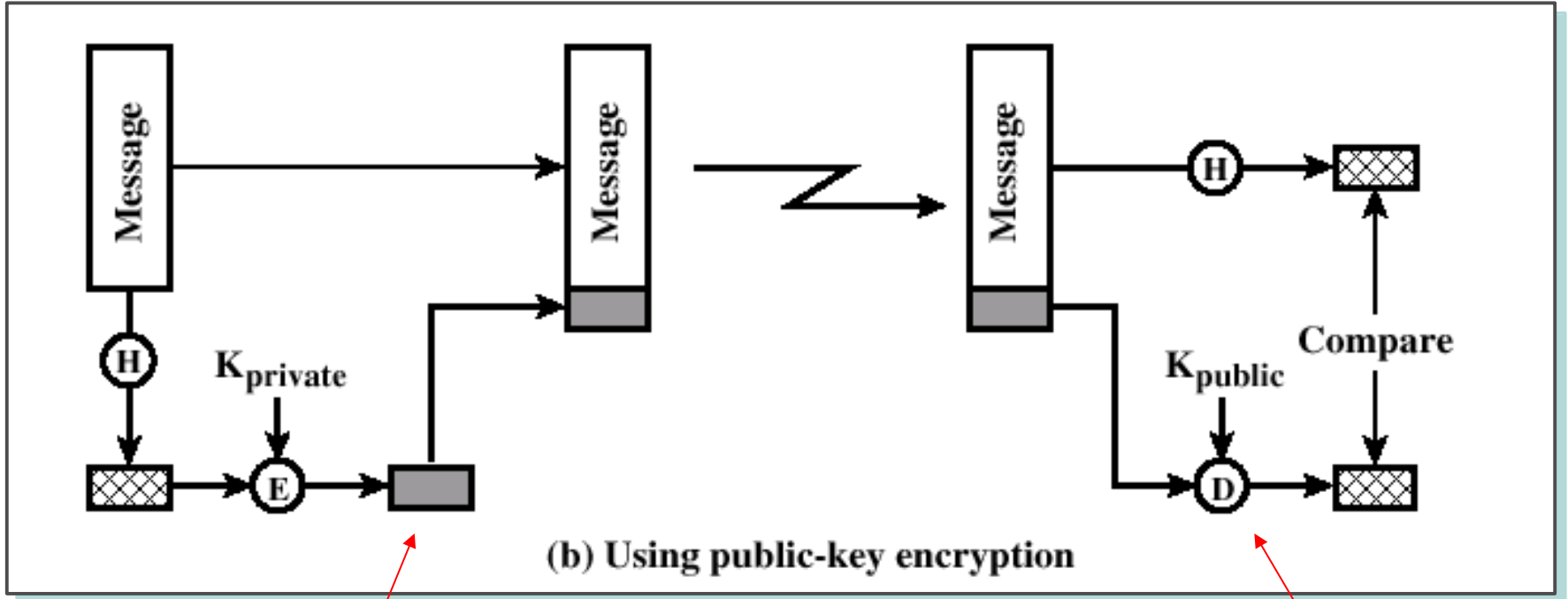


Mesaj özeti  $H(M)$

Simetrik Şifre algoritmasının  
paylaşılan gizli anahtarı  $K$

**Mesajın kaynağını, kimliği doğrulanır**

# Tek yönlü Özet Fonksiyonu



Digital İmza

Anahtar paylaşımı yok!

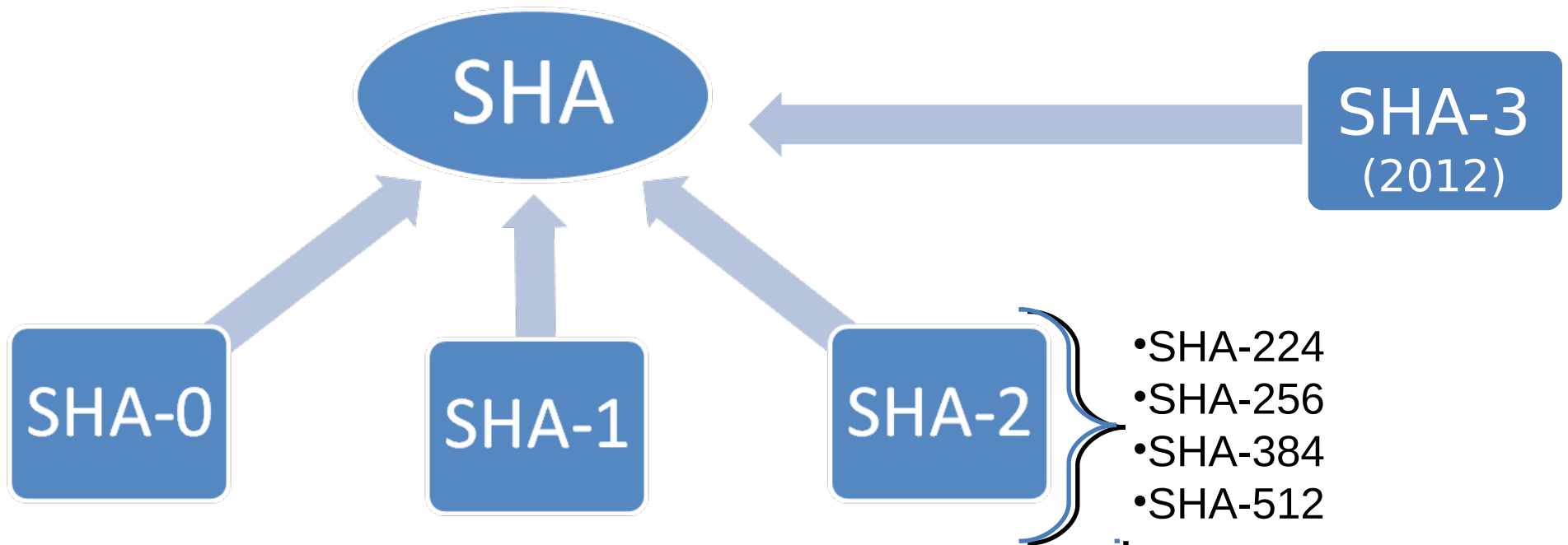
**Özet fonksiyonları ile Açık Anahtarlı Şifrelemenin birlikte kullanımı Sonuç: Mesajın kaynağını doğruma**



# SHA (Secure Hash Algorithm)

---

- 1993'te Amerikan Ulusal Güvenlik Kurumu (NSA) tasarladı ve Ulusal Teknoloji ve Standartlar Enstitüsü (NIST) yayımlandı (SHA-0)



- 1995'te SHA-0 SHA-1 olarak yeniden tasarlandı

# SHA-1 Özet Fonksiyonu

- SHA-1 birçok güvenlik uygulama ve protokollerinde yaygın olarak kullanılmaktadır:
- Güvenli olduğu düşünülüyordu fakat 2005 SHA-1 de bir güvenlik açığı bulundu. SHA-1 yapısında matematiksel bir zayıflık olma ihtimaline karşılık daha güvenli bir özet fonksiyonu arayışına girilmeye başlandı.
- Her ne kadar SHA-2 özet fonksiyonları için bir saldırı rapor edilmediyse de, SHA-2 fonksiyonları SHA-1 e algoritmik olarak benzemektedir.

# SHA-3

## **SHA-3 Seçimi:**

Yeni özet fonksiyon standardı, SHA-3, seçmek için düzenlenen açık yarışmanın (2008-2012 aralığı) ilk turu için 51 aday özet fonksiyonu vardı.

**Yarışmanın 3. turunda NIST kriptografi ile uğraşan kitleden önemli geribildirim ve kendi değerlendirmelerine dayanarak 5 tane SHA-3 finalistini seçti: -**

**BLAKE, Grøstl, JH, Keccak, ve Skein**

## **Detaylar:**

**[http://en.wikipedia.org/wiki/NIST\\_hash\\_function\\_competition](http://en.wikipedia.org/wiki/NIST_hash_function_competition)**

**<http://csrc.nist.gov/groups/ST/hash/sha-3/index.html>**

**Keccak, SHA-3 yarışmasının galibi (Ekim 2012).**

**SHA-3 detayları, FIPS 180-3 standardında yer alacak.**

# HMAC

- HMAC, bir çeşit MAC fonksiyonudur ve kriptografik bir özet fonksiyonu (gizli anahtar kullanmaz) kullanılarak tasarlanır.
- Yazılımda hızlı çalışır
- Gizli anahtar kullanımı gereklidir.
- [Tasarım ilkeleri RFC 2104](#) de listelenmiştir.
- [Ipsec](#) te kullanılır
- Aynı anda hem [bütünlük](#) hem de [kimlik doğrulama](#) için kullanılır

# Kriptografik olarak Güvenli sözde rasgele sayı üretici (pseudorandom number generator -PRNG-)

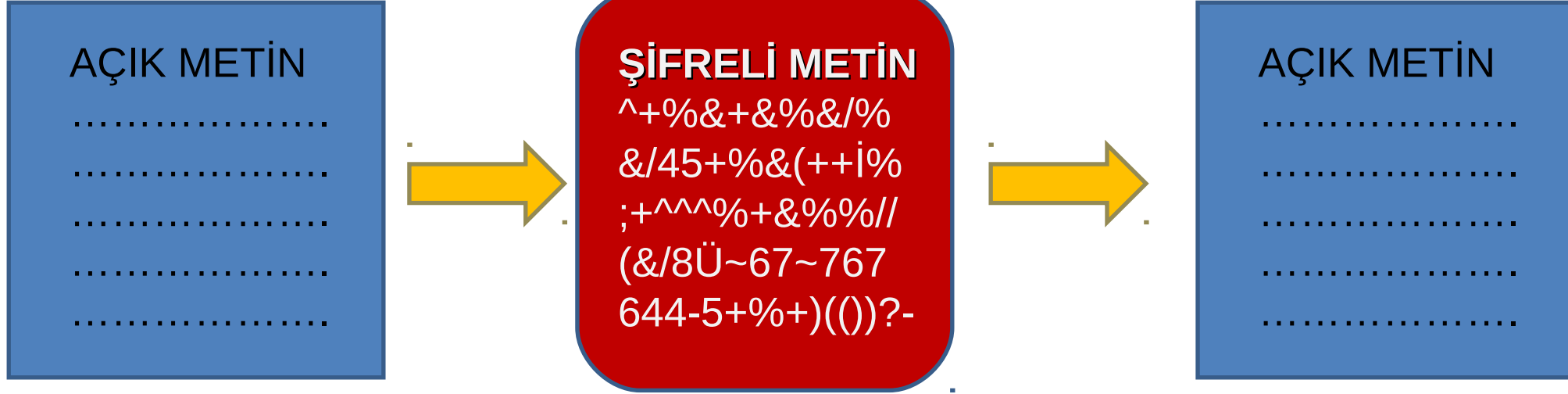
- Bir çeşit PRNG
- Kullanım alanları:
  - anahtar üretimi
  - protokollerde kullanılan rasgele sayı
  - Tek kullanımlık şerit (one time pad)
  - bazı protokollerde tuz eklemede
- Üretilen dizilere istatiksel testler uygulanmalı

# Asimetrik Kriptografi (Açık Anahtar Kriptografisi)

---

- 1976 yılında Diffie ve Hellman
- Gizliliğin yanı sıra, kimlik doğrulama ve inkar edememe
- Simetrik Kriptografi'deki anahtar dağıtımını sorununa çözüm
- Gizli ve açık, iki çeşit anahtar mevcut
- Açık anahtarlar herkes tarafından bilinir.
- Gizli anahtarlar kişiye özeldir.

# Asimetrik Kriptografi



BARIŞ



AÇIK



KAPALI

AYŞE

# Açık Anahtar Şifreleme Uygulamaları

## Şifreleme / Deşifreleme

(Encryption/Decryption) – mesaj alıcınının açık anahtarı ile şifrelenir

Dijital İmza (Digital signature) – gönderici, mesajını (genelde mesaj özetini) kapalı anahtarı ile şifreler

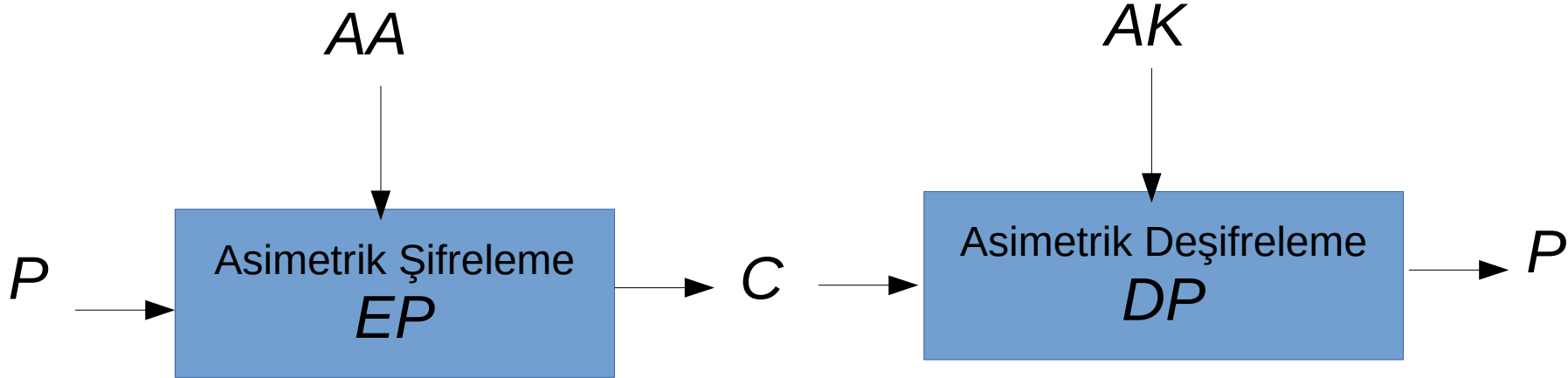
Anahtar Değişimi (Key Exchange) – Simetrik şifreleme algoritmasının gizli anahtarını, ik taraf paylaşır. Basitçe, taraflardan biri bu anahtarı belirler ve alıcınının açık anahtarı ile şifreler ve alıcıya iletir.



# Açık Anahtarlı (Asimetrik) Veri Şifreleme

$$EP(AA, P) = C$$

$$DP(AK, C) = P$$



**G**önderen (G)  
Anahtarları  
(**G**A, **G**K)

**A**lıcı (A)  
Anahtarları  
(**A**A, **A**K)

**GA**: Gönderen Açık anahtarı  
**GK**: Gönderen Kapalı anahtarı

**AA**: Alıcının Açık anahtarı  
**AK**: Alıcının Kapalı anahtarı

# Açık Anahtarlı Şifreleme

**Gönderen** kolayca, açık anahtarlı şifreleme için anahtar ikilisini yaratabilir:

Açık anahtar (public key)  $GA$  ; Kapalı anahtar (private key)  $GK$

**Alıcı** da benzer şekilde açık anahtar ikilisini yaratır:

Açık anahtar (public key)  $AA$  ; Kapalı anahtar (private key)  $AK$

**Gönderen**, izleyen mesaj  $M$  için açık anahtar şifrelemeyi,

**Alıcı** nın açık anahtarı ile yapar:  $C = E_{AA}(P)$  ve  $C$  yi **Alıcı** ya gönderir.

**Alıcı** ise gizli anahtarını ve deşifreleme algoritmasını kullanarak  $C$  den,

**Gönderenin** ilettiği  $P$  mesajını (veya dosyasını) elde eder:

$$D_{AK}(C) = D_{AK}[E_{AA}(P)] = P$$

# Açık Anahtarlı Şifreleme Gereksinimleri

$$C = E_{AA}(P)$$

Kötü niyetli veya erişim yetkisi olmayan için: açık anahtar **AA** dan kapalı anahtar **AK** elde etmesi hesaplama açısından mümkün olmamalı.

Açık anahtar **AA** ve kapalı metin  $C$  yi kullanarak orjinal metin  $P$  elde etmek hesaplama açısından mümkün olmamalı.

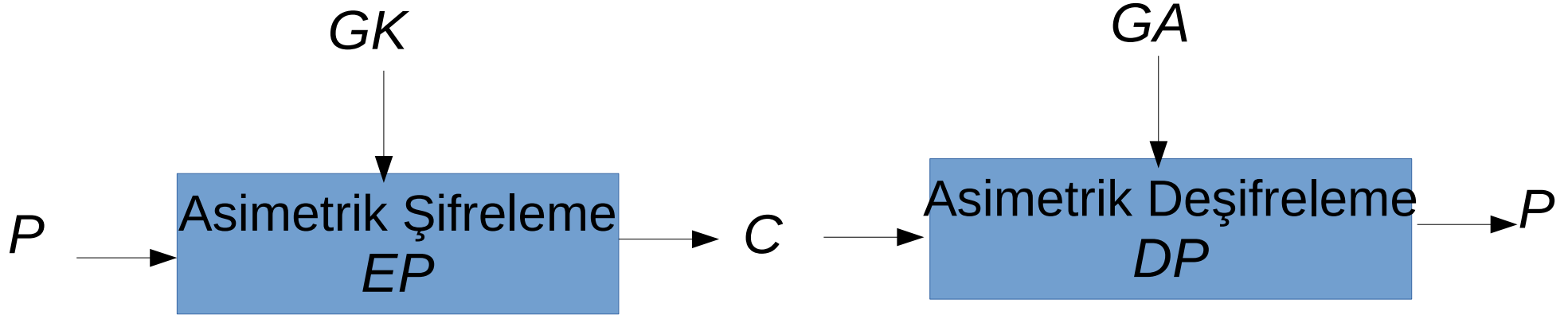
- Açık anahtar ve kapalı anahtar arasında matematiksel bir bağ mevcuttur. Birisi şifreleme için kullanılır ise bir diğeri deşifreleme için kullanılır:

$$P = D_{AK}[E_{AA}(P)] = E_{AA}[D_{AK}(P)]$$

# Asimetrik Şifreleme ile Verinin Kaynağını -Kimliğini- Denetleme

$$EP(GK, P) = C$$

$$DP(GA, C) = P$$



**G**önderen (G)  
Anahtarları  
(**GA**, **GK**)

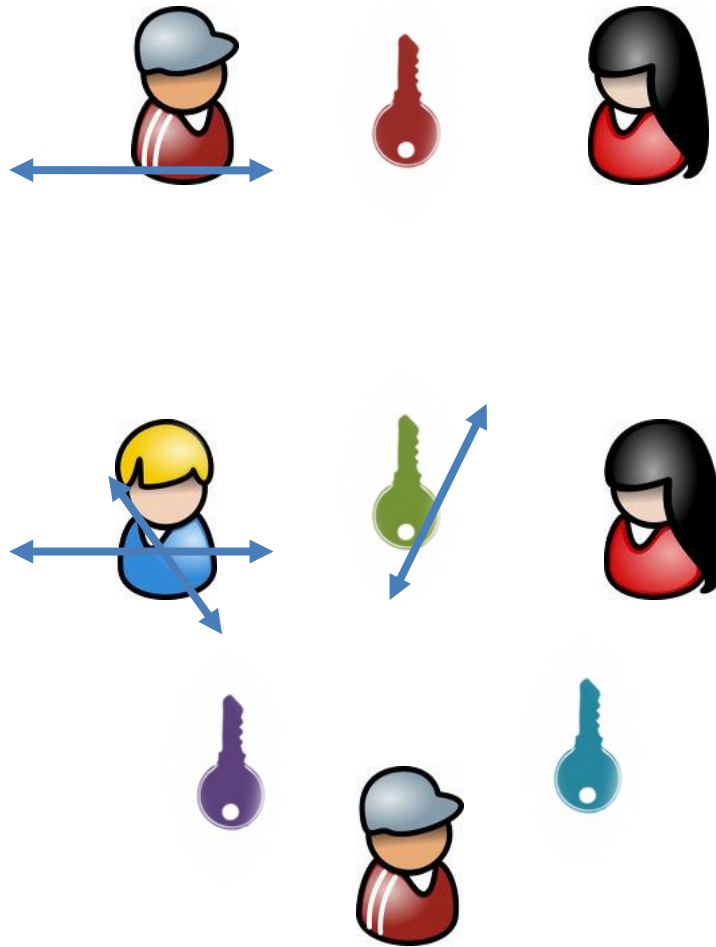
**A**lıcı (A)  
Anahtarları  
(**AA**, **AK**)

**GA**: **G**önderen Açık anahtarı  
**GK**: **G**önderen Kapalı anahtarı

**AA**: **A**lıcının Açık anahtarı  
**AK**: **A**lıcının Kapalı anahtarı

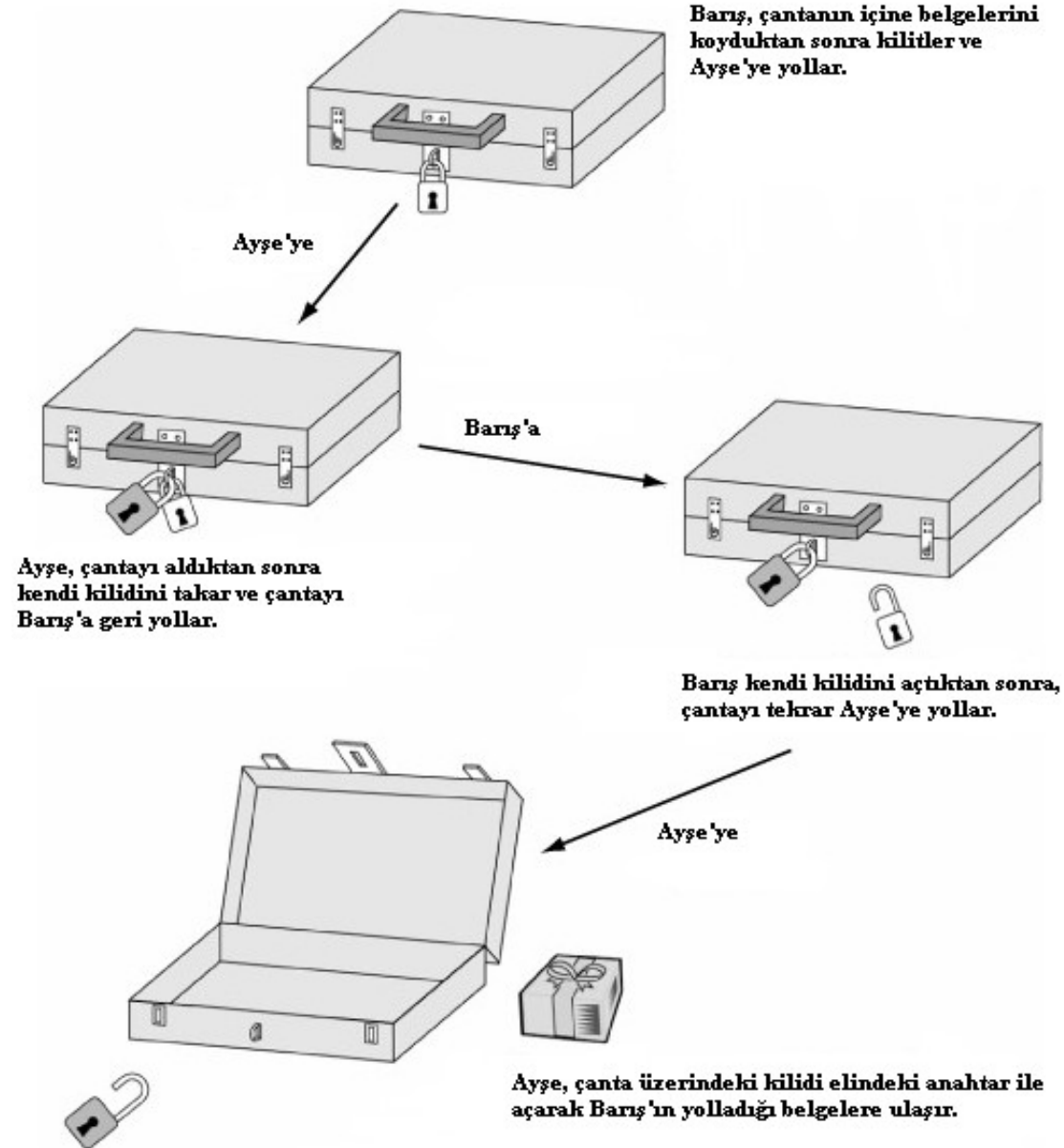
# Diffie-Hellman Anahtar Değişimi

---



# Diffie-Hellman Anahtar Değişimi

- Gizli anahtarlı sistemlerde şifreleme ve şifre çözme için ortak bir anahtar gerekmektedir.
- Diffie-Hellman anahtar değişimi bu anahtarı oluşturmak için kullanılmaktadır
- Sonlu cisimler üzerinde veya eliptik eğri aritmetiğinde bu anahtar değişiminin uygulaması yapılabilmektedir.



# Ayrık Logaritma Problemi

---

$$\triangleright Z_p^* = \{1, 2, \dots, p-1\}$$

$Z_p^*$ 'de  $g$  ve  $y$  verilmiş ve  $g^x = y \pmod{p}$  ise  $x$  kaçtır?

$x = \log_g y$  bulunması zor bir problemdir.

Örnek:  $Z_{17}$ 'de  $g=3$  ve  $y=11$ ,

$$3^x = 11 \pmod{17} \Rightarrow x = ?$$

$$3 \pmod{17} = 3$$

$$3^2 \pmod{17} = 9$$

$$3^3 \pmod{17} = 10$$

$$3^4 \pmod{17} = 13$$

$$3^5 \pmod{17} = 5$$

$$3^6 \pmod{17} = 15$$

$$3^7 \pmod{17} = 11$$

$$3^8 \pmod{17} = 16$$

$$3^9 \pmod{17} = 14$$

$$3^{10} \pmod{17} = 8$$

$$3^{11} \pmod{17} = 7$$

$$3^{12} \pmod{17} = 4$$

$$3^{13} \pmod{17} = 12$$

$$3^{14} \pmod{17} = 2$$

$$3^{15} \pmod{17} = 6$$

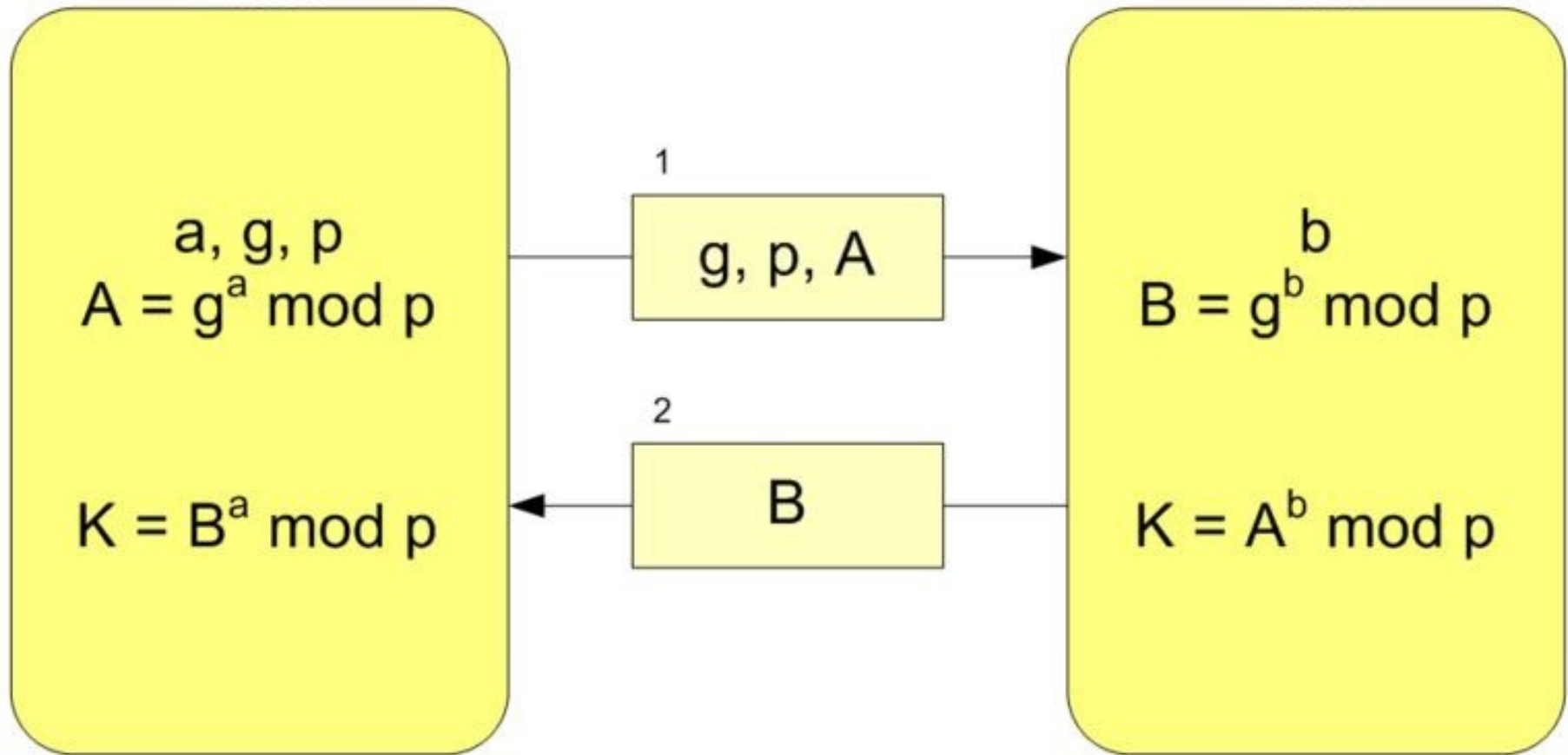
$$3^{16} \pmod{17} = 1$$

$$3^{17} \pmod{17} = 3$$

# Diffie-Hellman Anahtar Değişimi

**Ayşe**

**Barış**



$$K = A^b \text{ mod } p = (g^a \text{ mod } p)^b \text{ mod } p = g^{ab} \text{ mod } p = (g^b \text{ mod } p)^a \text{ mod } p = B^a \text{ mod } p$$



# RSA Algoritması

---

- 1977 - Ron Rivest, Adi Shamir ve Leonard Adleman
- Çarpanlara ayırmanın zorluğunu temel alır
- Şifreleme ve elektronik imza uygulamalarında kullanılmaktadır.

# RSA Algoritması

---

- Çarpanlara ayırma problemi nedir ?
- Verilen N sayısını bölen asal sayıları bulmak
- 187'in asal çarpanları nelerdir?
- 12524224246730422698081804002962677144908939996961  
633338104994162295371867324032252932820702035478088  
806772257620720696601299194344613764092266067110703  
775459945356598594258251300949290798217344667521645  
463459276100019171025163859012394863073232630792295  
249446485750541517740232249989121858230784235194221  
9477?

# RSA'da Anahtar Oluşturma

---

➤ Ayşe iki asal sayı  $p$  ve  $q$ 'yu seçer.

Örnek  $p=17$  ve  $q=11$

➤  $N = p \times q$ 'yu elde eder.

$$N = 17 \times 11 = 187$$

➤  $\varphi(N) = (q-1) \times (p-1)$ 'i hesaplar.

$$\varphi(N) = 16 * 10 = 160$$

➤  $1 < e < \varphi(N)$  ve  $\text{obeb}(e, \varphi(N)) = 1$  olan bir  $e$  seçer.

$e = 7$  diyelim;

$$1 < e=7 < 160 \text{ ve } \text{obeb}(7, 160) = 1$$



$$p = 17 \quad q = 11$$

$$N = 187 \quad e = 7$$

# RSA'da Anahtar Oluşturma

---

➤  $1 < d < \varphi(N)$  ve  $e \times d = 1 \pmod{\varphi(N)}$  olan  $d$ 'yi bulur

$$7 \times d = 1 \pmod{160} \Rightarrow d = 23$$

➤ Böylece Ayşe gizli  $p$ ,  $q$  ve  $d$  anahtarlarını, açık  $N$  ve  $e$  anahtarlarını oluşturmuş olur.

Gizli Anahtarlar:  $p = 17$   $q = 11$   $d = 23$

Açık Anahtarlar:  $N = 187$   $e = 7$

# RSA Güvenilirliği

- **Deneme-yanılma saldırısını (Brute force attack)**: tüm olası anahtarları deneyerek bulmaya- dayanıklılık için büyük  $e$  and  $d$  seçilebilir
- Bununla beraber ne kadar büyük anahtar kullanılır ise açık şifreleme algoritması o kadar yavaşlayacak
- Büyük  $n$  tam sayısı için büyük iki asal sayı çarpanını (Standartların tavsiyede ettiği şekilde, saldırılara kuvvetli asalların seçildiğini varsayarsak) bulmak adına çarpanlarına ayırma metodu zor bir problemdir.
- 1994 yılında 428 bit RSA  $n$  çarpanlarına ayrıldı. Ödül: **\$100**
- 2010 yılı başlarında 768 bit uzunluğunda  $n$  tam sayısı (232 ondalık basamaklı), **“Number Field Sieve”** algoritması ile çarpanlarına ayrılmıştır (<http://eprint.iacr.org/2010/006.pdf>).
- RSA anahtarının uzunluğu genelde 1024–2048 bit. Bazı uzmanlar, 1024-bit anahtarların yakın zamanda kırılabileceğini düşünüyorlar.
- Standartlar,  $n$  tam sayısının en az 2048 bit olmasını tavsiye ediyor.

# Eliptik Eğri Kriptografi

- Eliptik Eğri Kriptografi (Elliptic Curve Cryptography (ECC)) tabanlı şifreleme algoritmaları, RSA yerine tercih edilmeye başlandı.
- Dr. Scott Vanstone (Matematik ve Bilgisayar Bilimleri Profesörü ve Certicom şirketi kurucularından), ECC için açık anahtarlı kriptografinin gelecek nesli (özellikle kablosuz iletişim) olarak bahsediyor. Certicom firmasının ECC üzerine algoritmalar konusunda birçok patenti mevcuttur.

## **Kaynak:**

<http://www.sciencedirect.com/science/article/pii/S0167404803005078>

# Eliptik Eđri Kriptografi

## -Kullanım Yararları-

• **Eliptik Eđri Kriptografi** tabanlı şifreleme algoritmaları, DSA ve RSA parametrelerine nazaran daha küçük parametreler kullanırlar ve eşdeđer güvenlik seviyesini sađlarlar (Bakınız: “On the Security of 1024-bit RSA and 160-bit Elliptic Curve Cryptography”):

<http://lcal.epfl.ch/files/content/sites/lcal/files/papers/ecdl2.pdf>

Eđer izleyen kaynaklardan bir veya birkaçı sınırlı ise **Eliptik Eđri Kriptografi** kullanım faydası olacaktır:

- İşlemci Gücü
- Veri depolama alanı
- Band genişliđi
- Güç Tüketimi

**Eliptik Eđri Kriptografi** is özellikle sınırlı çevre birimlerine sahip akıllı kart, mobil cihazlar, el bilgisayarları ve diđer benzeri sistemler için tercih edilebilir.

# Eliptik Eğri Kriptografi

- **Standartlar:** ANS X9F1 | CRYPTREC | IEEE P1363 | NESSIE | NSA Suite B
- Netscape Security Services (NSS), yazılım kütüphanesi kümesi olup, birçok platform bağımsız istemci-server uygulamalarını desteklemektedir. NSS içinde desteği bulunanlar: SSL v2 and v3, TLS, PKCS #5, PKCS #7, PKCS #11, PKCS #12, S/MIME, X.509 v3 sertifikaları ve diğer güvenlik ile ilgili standartlar.



# Eliptik Eğri Kriptografi

- Open SSL and NSS (sürüm 3.8) Eliptik Eğri Kriptografi algoritmalarını desteklemektedir.
- Mozilla web tarayıcı Eliptik Eğri Kriptografi algoritmalarını SSL sayesinde desteklemektedir.
- E-posta imzalama/doğrulama için ECDSA kullanılabilir (Digital Signature Algorithm (DSA), [Eliptik Eğri Kriptografi sürümü](#) )

# Asimetrik Sistemlerin Karşılaştırılması

	<b>Şifreleme</b>	<b>İmzalama</b>	<b>Anahtar Paylaşımı</b>
<b>RSA</b>	Evet	Evet	Evet
<b>Diffie-Hellman</b>	Hayır	Hayır	Evet
<b>DSA</b>	Hayır	Evet	Hayır
<b>Eliptik Eğriler</b>	Evet	Evet	Evet

# Açık Anahtarlı Şifreleme ile Simetrik Şifreleme Anahtarı Paylaşımı

- Simetrik Şifreleme için, gizli anahtar (secret key) mutlaka haberleşecek iki kişi arasında güvenli bir şekilde paylaşılmalıdır (örneğin Ayşe and Bülent arasında).
- Bir yaklaşım “Diffie-Hellman anahtar değişimi algoritmasını” kullanmak. Yaygın kullanılıyor fakat basit halinde kullanımı kimlik doğrulama olmadığı için kullanılmalıdır.

Diğer bir güçlü yaklaşım açık anahtar sertifikaları (alıcının açık anahtarı edinmenin yolu) kullanmaktır:

Bülent, Ayşe ile güvenli haberleşmede bulunmak ister. Bunun için Bülent izleyen yolları takip eder:

- 1) Mesaj  $M$  yi hazırlar
- 2) Bu mesajı belirlediği gizli anahtar  $K$  (belki bir kere kullanmak üzere) ile simetrik bir şifreleme sistemi ile şifreler (AES-192 gibi):  $E_K(M)=C$
- 3) Ek olarak gizli anahtar  $K$  yı, Ayşe'nin açık anahtarı  $PU_a$  ile bir açık anahtar şifreleme algoritması ile şifreler:  $E_{PU_a}(K)=K'$
- 4) Bülent  $K'$  ve  $C$  yi Ayşe'ye gönderir.

Ayşe  $D_{PRa}[K']=D_{PRa}[E_{PU_a}(K)]=K$  elde eder ve ayrıca  $M$  yi *hesaplar*: 107

$$D_K(C)=M$$

# Kriptografik Algoritmaların Hatalı Kullanımına Örnek

- İlgili makale başlığı “An Empirical Study of Cryptographic Misuse in Android Applications”

[www.cs.ucsb.edu/~chris/research/doc/ccs13\\_cryptolint.pdf](http://www.cs.ucsb.edu/~chris/research/doc/ccs13_cryptolint.pdf)

- Google Play otomatik kontrol eden bir analiz aracı geliştirerek 11,748 uygulamadan 10,327 nin kriptografik UPA -API-“ kullandığını ve %88 nin en az bir hata yaptığını görmüşler.

## – İhlal edilen bazı kurallar

- **Kural 1:** Şifreleme için ECB mod şifrelemesinde kullanma
- **Kural 2:** Rastgele seçilmemiş başlangıç vektörünü (IV) CBC kullanma
- **Kural 3:** Sabit gizli anahtarlar kullanma

# Kriptografik Algoritmaları Gerçekleştirme

- Algoritmaları ve bunları kullanan güvenlik protokollerini gerçekleştirirken yazılım geliştirenler hatalar yapıyorlar:
  - **örneğin:** openssl Heartbleed açığı

Apple SSL/TLS hata (bug)

Kriptografik Standartlar ve geliştirilen araçlarda arka kapılar olabilir

- Algoritma ve parametreleri öneren projeler/yarışmalar mevcut:
- Algoritmalar kriptanaliz literatürü inceleyerek standartların/proje önerilerinin ilerisinde olmakta yarar var. Örneğin:

<http://safecurves.cr.yyp.to/>

# SafeCurves: choosing safe curves for elliptic-curve cryptography

- Introduction
- Curve parameters:
  - Fields
  - Equations
  - Base points
  - Prime proofs
- ECDLP security:
  - Rho
  - Transfers
  - Discriminants
  - Rigidity
- ECC security:
  - Ladders
  - Twists
  - Completeness
  - Indistinguishability
- More information:
  - References
  - Verification

## Introduction

There are several different standards covering selection of curves for use in elliptic-curve cryptography (ECC):

- [ANSI X9.62](#) (1999).
- [IEEE P1363](#) (2000).
- [SEC 2](#) (2000).
- [NIST FIPS 186-2](#) (2000).
- [ANSI X9.63](#) (2001).
- [Brainpool](#) (2005).
- [NSA Suite B](#) (2005).
- [ANSSI FRP256V1](#) (2011).

Each of these standards tries to ensure that the **elliptic-curve discrete-logarithm problem** (ECDLP) is difficult. ECDLP is the problem of finding an ECC user's secret key, given the user's public key.

Unfortunately, there is a gap between ECDLP difficulty and ECC security. None of these standards do a good job of ensuring ECC security. There are many attacks that break real-world ECC without solving ECDLP. The core problem is that **if you implement the standard curves, chances are you're doing wrong**:

- Your implementation produces incorrect results for some rare curve points.
- Your implementation leaks secret data when the input isn't a curve point.
- Your implementation leaks secret data through branch timing.
- Your implementation leaks secret data through cache timing.

These problems are exploitable by real attackers, taking advantage of the gaps between ECDLP and real-world ECC:

# Kriptografik Algoritma ve Parametreleri seçim önerileri sunan Projeler

- 4 yıllık AB projesi ECRYPT (European Network of Excellence in Cryptology) 2004-2008
  - bilgi güvenliği, özellikle kriptoloji
  - digital gizli damgalama (watermarking)
- **ECRYPT-II** 1 Ağustos 2008 - 31 Ocak 2013

## Sanal lablar

- Simetrik teknikler (SymLab), özel ilgi alanları:
  - Özet fonksiyonlar
  - Hafif siklet (Lightweight) kriptografi
- Çoklu parti ve asimetrik algoritmalar (MAYA)
- Güvenli Secure and etkili gerçekleştirmeler (VAMPIRE)

Algoritma ve Parametre önerileri

# Kriptografik Algoritma ve Parametreleri seçim önerileri sunan Projeler

- Tamamlanan AB projesi NESSIE (New European Schemes for Signatures, Integrity, and Encryption) -2004-

**Amaç:** Açık çağrı, güçlü kriptografik algoritmaları analiz etmek ve seçmek ve AES blok şifre sisteminin seçiminin son aşamasını desteklemek.

Seçilen algoritmaların son raporu

- **Japonya'ın kurduğu CRYPTREC** (Cryptography Research and Evaluation Committees) -Kriptografi Araştırma ve Değerlendirme komiteleri
  - Tavsiye edilen algoritma listesi



# Kriptografi Standartları

- [ETSI TS 102 176-1 V2.0.0](#) Electronic Signatures and Infrastructures (ESI); Algorithms and Parameters for Secure Electronic Signatures; Part 1: Hash functions and asymmetric algorithms
- [IEEE P1363 \(2000 ve 2004\)](#): "Standard Specifications for Public-Key Cryptography".
- ANSI X9.62, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA), 2005.
- FIPS PUB 140-1-2-3, Security Requirements for Cryptographic Modules (prepared by National Institute of Standards and Technology: NIST, US federal agency)
- IETF RFC 3647: Internet X.509 PKI Certification Plan (prepared by the Internet Society)
- PKCS #1: RSA Cryptography Standard
- PKCS #3: Diffie-Hellman Key Agreement Standard
- PKCS #5: Password-Based Cryptography Standard
- PKCS #7: Cryptographic Message Syntax Standard
- PKCS #10: Certification Request Syntax Standard
- PKCS #11: Cryptographic Token Interface Standard
- PKCS #12: Personal Information Exchange Syntax Standard
- PKCS #13: Elliptic Curve Cryptography Standard

# Bitirirken

- Bilgi güvenliği ile ilgili haberlerin artması sayesinde kriptografiye olan merak ve farkındalık gittikçe artmaktadır:

Akademik Bilişim 2014 öncesi [Kriptografi kursuna](#) katılım, 125 kişi (lisans/y.lisans öğrencileri, kamu kurum, firmadan) civarı

- Bilgisayar Mühendisliği, Bilişim Sistemleri vb. Bölümlerde bilgi güvenliği kavramları ve kriptolojinin temelleri işleyen zorunlu bir ders, ders programında yer almalıdır.
- Bilişim sistemlerini tasarlarken mutlaka bilgi güvenliği hedefleri dikkate alınmalıdır.

# Bitirirken

- Kriptografi ile uğraşmak isteyenlerin bilmesi gereken 52 şey:

<http://www.cs.bris.ac.uk/Research/CryptographySecurity/knowledge.html>

- Kriptografik algoritmaları anlatmak ve nerede ne şekilde kullanılacağını temellerini anlatmak kolay fakat bu algoritmaları tasarlamak ve analizini yapmak zor iş

(+Bulmaca Çözme merakı+Matematik+İstatistik+Bilgisayar Bilimleri+Mühendislik)

- <http://www.cryptool.org/en/cryptool1-en> Türkçeleştirme destek verilebilir. Çalışmayı başlattım fakat yarım kaldı.
- Ders kitabı yazılabilir...

Teşekkürler...