

# Kriptoloji

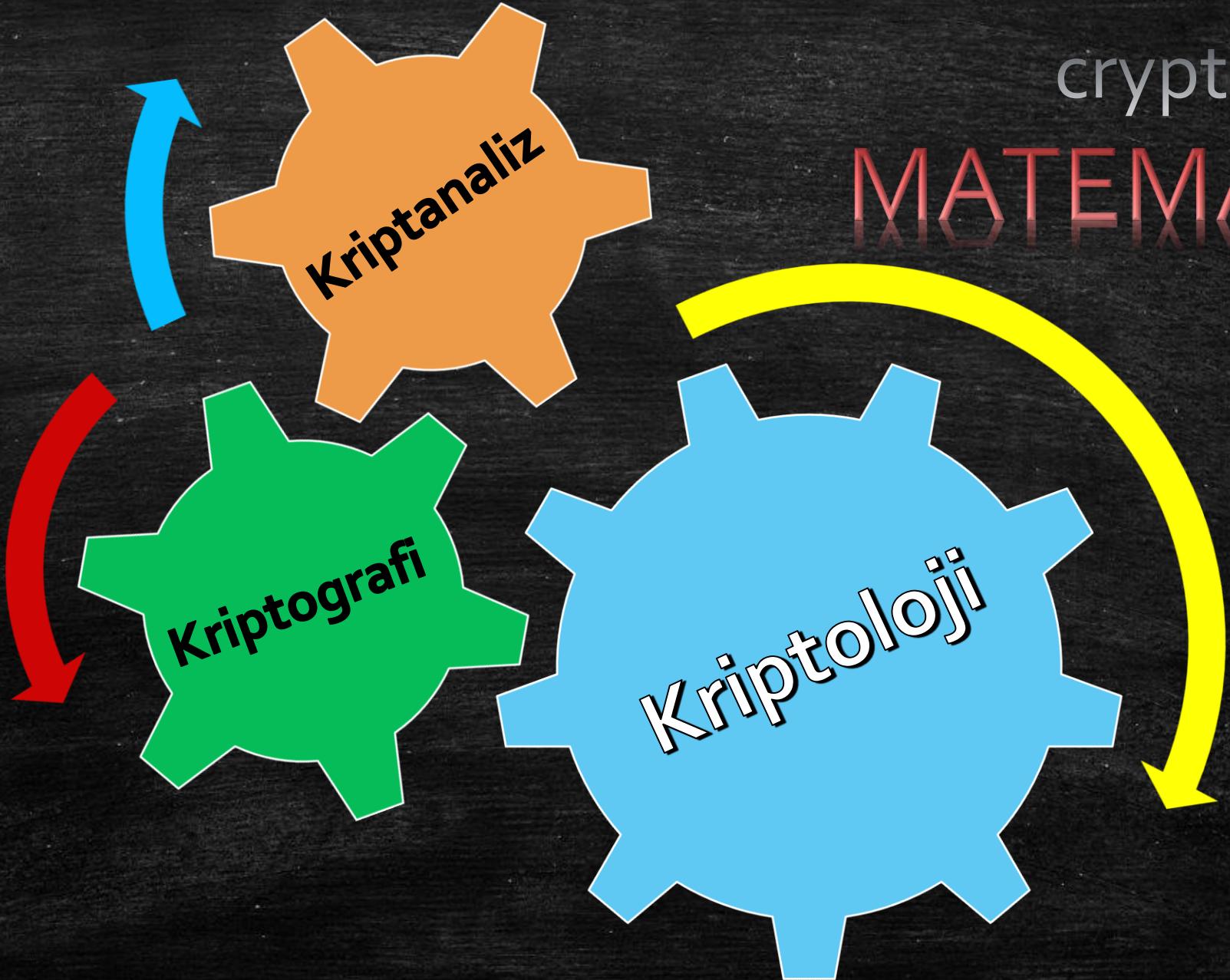
---

Ağ ve Bilgi Güvenliği Dersi – Necmettin ÇARKACI

# Kriptos

Logos

# Graphi



cryptologie  
MATEMATİK

Sayı Teorisi

# Gizli Bilgi

iglib ilzig

..... . . . . . / - - - - - . . . . .



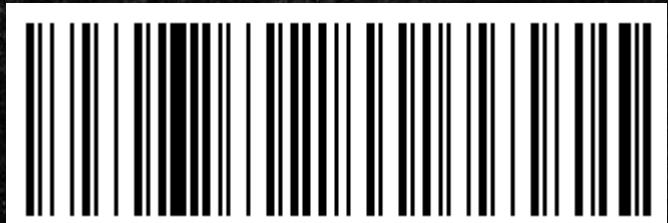
غىزلى بىلغى

. كۆنەتىرى



Gizli Bilgi

عِزْلِ بِلْعِ



ئەخەن ئەنەن

tvmyv ovytv

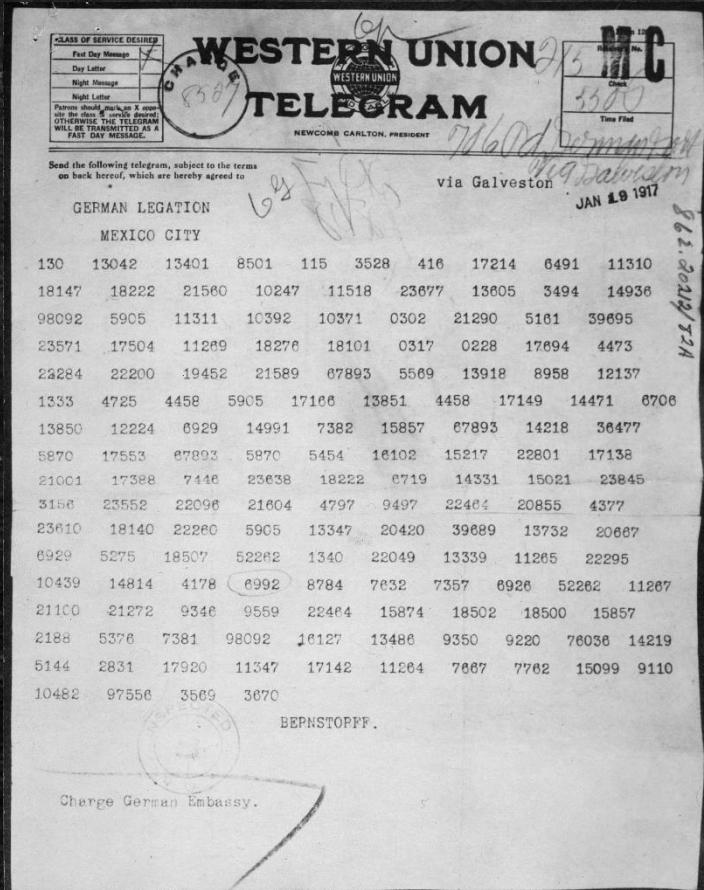


# Tarihçe

---

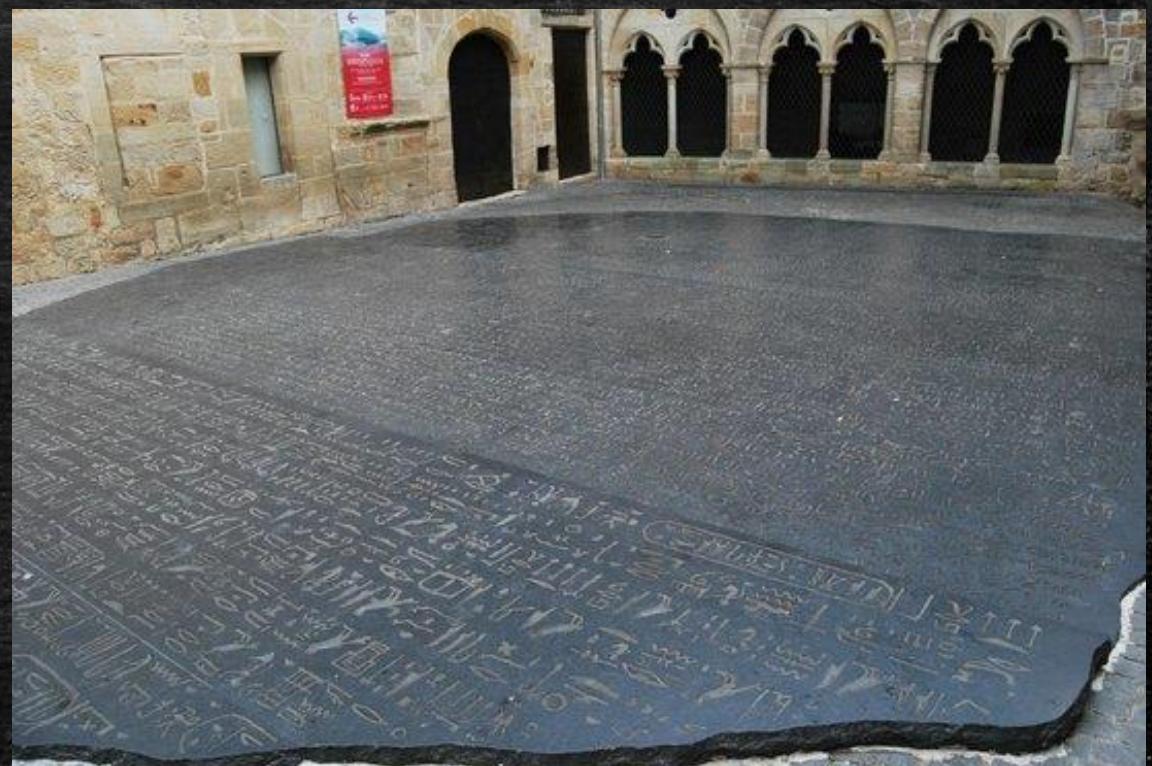
- ⌚ En az 4000 yıl öncesine dayanır.
- ⌚ Eski misirliler anıtlara yazdıkları resimli yazılarını şifrelemişlerdir.
- ⌚ Eski İbraniler kutsal kitaplarındaki belirli kelimeleri şifrelemişlerdir.
- ⌚ 2000 sene önce Jul Sezar, basit bir yerine koyma şifresi kullandı.
- ⌚ Leon Alberti 1460 larda bir şifre tekerleği kullandı ve frekans analizinin prensiplerini açıkladı.
- ⌚ Blaise de Vigenère 1855 de kriptoloji üzerine bir kitap yayınladı ve çoklu alfabe değiştirme şifresini açıkladı.

# Zimmerman Telgrafı



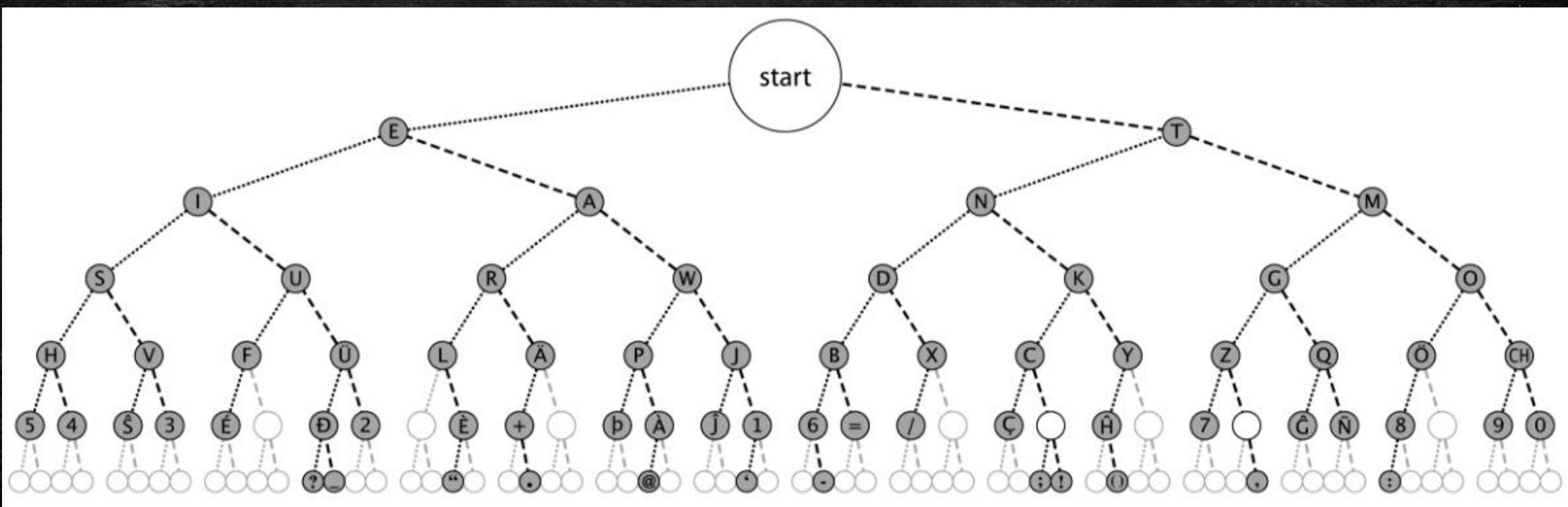
# Rosetta Tableti

---



- ⌚ II. Dünya Savaşı esnasında Almanlar da bu teknikten yararlandı. Fakat Yunanlıların yaptığı gibi mesajı fiziksel olarak saklamadılar. “Null Ciphering” terimiyle açıklanan bir metot kullandılar. Örneğin aşağıdaki mesaj II. Dünya Savaşı’nda Alman bir casus tarafından gönderildi.
- ⌚ Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit. Blockade issue affects pretext for embargo on by-product, ejecting suets and vegetable oils.
- ⌚ Bu mesajdaki her kelimenin 2. harfini alırsak karşımıza Pershing sails from NY June 1 şeklinde bir gizli mesaj ortaya çıkacaktır.

# Morse Alfabesi



## Uygulama : Morse Alfabesi

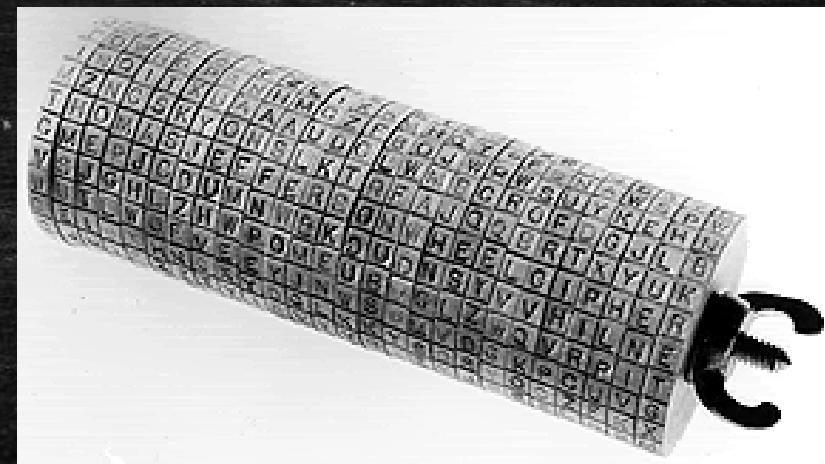
---

- ⌚ Veriel ses dosyasını online araçlar veya doğrudan herhangi bir ses işleme yazılımı ile çözmek mümkündü. Örneğin audacity aracı ile açıldığında, ince-kalın ikilileri rahat bir şekilde gözükyordu.

# Jefferson Silindir

---

- ⌚ 1790 'larda geliştirilen Jefferson cylinder, her biri rastgele alfabeli 36 adet diskten oluşmaktadır, disklerin sırası anahtarı oluşturmaktaydı, mesaj ayarlanınca diğer satır şifreyi oluşturmaktaydı



# Wheatsone Disk

---

- ⌚ 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.



# Enigma Makinesi

---

- ⌚ 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

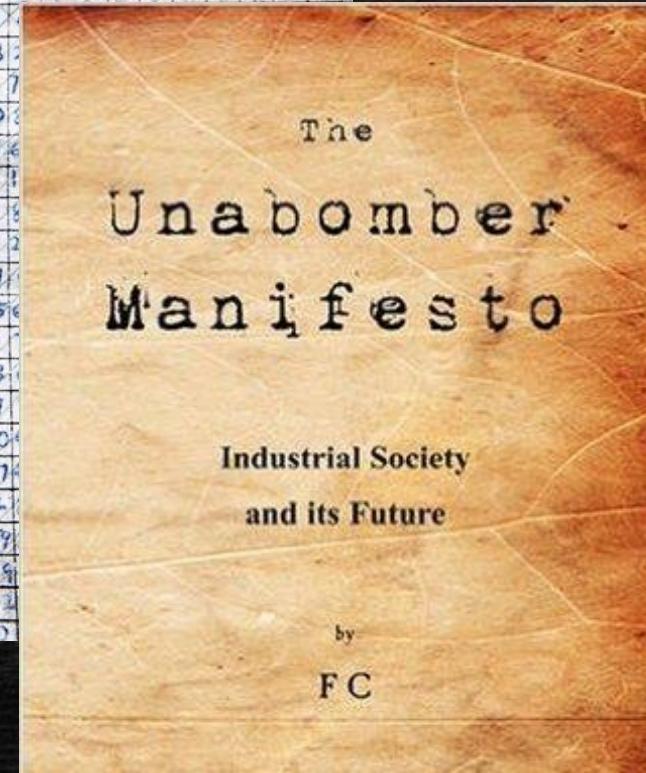
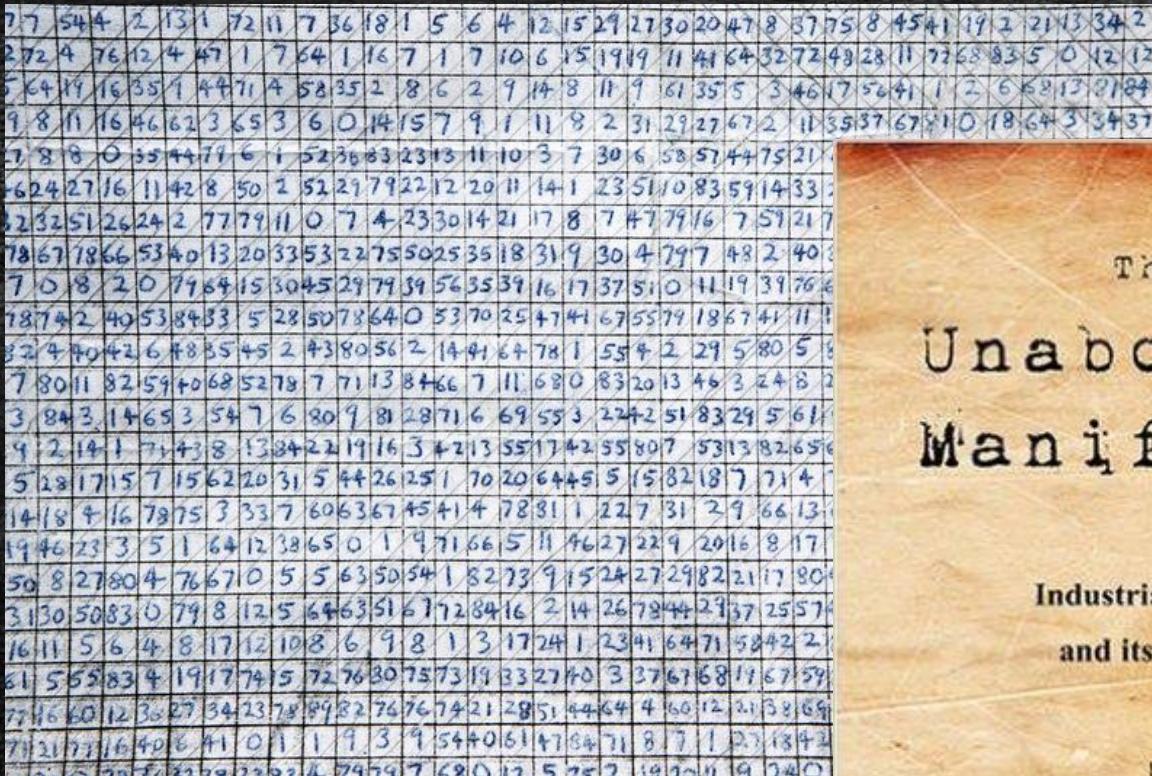


# Matematikçi

# Kriptografi

# Anarşist

# Terörist



# Şifrelemenin ardındaki itici güç

---

- ⌚ Şavaş (Dünya savaşları)
- ⌚ Ticaret (15. yy. İtalyası) İtalya Avrupa Şifre Biliminin Merkezi
- ⌚ Taht kavgaları (İngiltere)
- ⌚ İstihbarat servisleri
- ⌚ Entrika (Sezar)
- ⌚ Aşk (Kamasutra)

**Kaos yaratıcılığı artırır.**

# Şifrelemede Amaç

---

- ⌚ **Yetkinlik (Confidentiality)** : Sadece yetkili yetkisi olan bilgiye ulaşabilisin.
- ⌚ **Doğrulama (Authentication)** : Bilginin kaynağını doğrulasın.
- ⌚ **Bütünlük (Integrity)** : Bilginin bütünlüğü korunsun.

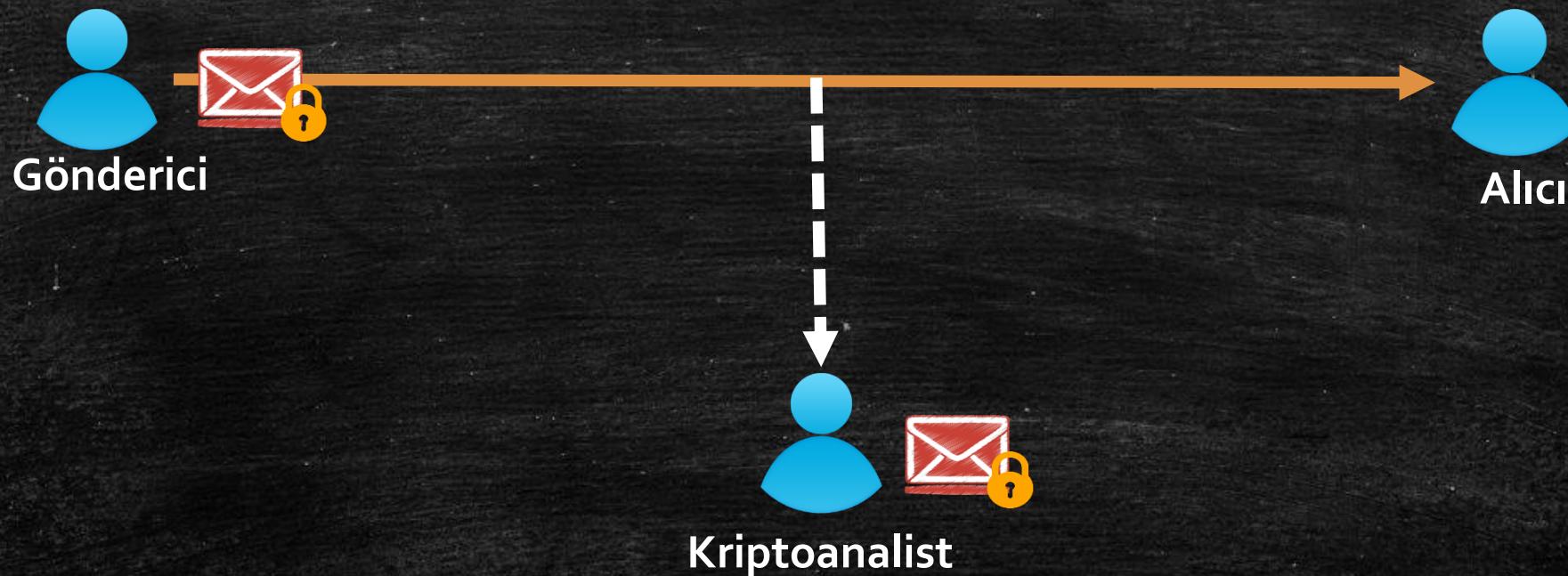
# Temel Kavramlar

- ⌚ **Kriptografi (cryptography)**: Anlaşılır bir mesajı anlaşılmaz şekle dönüştürme ve mesajı tekrar eski orijinal haline geri dönüştürme prensipleri ve yöntemlerini içeren sanat veya bilimdir.
- ⌚ **Kriptoanaliz (cryptanalysis)** : Bilgi ve anahtar olmaksızın anlaşılmaz mesajı anlaşılır mesaj olarak geri dönüştürme prensipleri ve yöntemleridir. Aynı zamanda kod kırma(code breaking) olarak da adlandırılır.
- ⌚ **Kriptoloji (cryptology)** : Kriptografi ve kriptoanalizin her ikisi kriptoloji bilimini oluşturur.

## KRIPTOGRAFİ



# Kriptanaliz - Pasif Dinleme



**KRIPTANALİZ : Pasif Dinleme → Gizlilik İhlali**

# Kriptanaliz - Aktif Dinleme



**KRIPTANALİZ : Aktif Dinleme → Doğrulama İhlali**

# Kriptosistem

---

- ⌚ Sonlu sayıda elemanlar kümesinden oluşan **A alfabesi**
- ⌚ A alfabeden alınmış sonlu sayıda elemanlar dizisinden oluşan **Açık Metin Uzayı P**
- ⌚ A alfabetesinden alınmış, P'de farklı diziliş gösteren elemanlardan oluşan **Şifreli Metin Uzayı C**
- ⌚ A alfabetesinden alınmış sonlu sayıda elemanlardan oluşan **K Anahtar Uzayı**.
- ⌚ **E şifreleme, D ise deşifreleme** fonksiyonu veya algoritmasını olmak üzere;
- ⌚ **Tanım :** Bir kriptosistem aşağıdaki şartları sağlayan **(P,C,K,E,D)** beşlisinden oluşur.
- ⌚  $\forall k \in K, D_k \in D$  fonksiyonuna uyan bir  $E_k \in E$  fonksiyonu vardır. Öyle ki;
- ⌚  $\forall E_k: P \rightarrow C$  ve  $\forall D_k: C \rightarrow P$  ve her metin  $\in P$  için  $D_k(E_k(\text{metin})) = \text{metin}$

# Temel Kavramlar

---

- ⌚ **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 (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

# Saldırı Yöntemleri

---

- ⌚ **Kriptoanaliz :** Kriptoanalitik saldırılar, algoritmanın özelliği, şifresiz metnin genel karakteristiği hakkında 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.

# Şifrelerin Güvenliği

---

- ⌚ **Mutlak Güvenlik** : Bilgisayar gücü ne kadar fazla olursa olsun şifre hiçbir şekilde kırılamaz.
- ⌚ **Hesaplamaya Bağlı Güvenlik** : Bir şifreleme algoritması aşağıdaki kriterleri sağlıyor ise hesaplamaya bağlı güvenli (computationally secure) dır.
  - Şifrenin kırılmasının maliyeti şifrelenmiş bilginin değerinden fazla ise
  - Şifreyi kırmak için gereken zaman, bilginin yararlı ömründen fazla ise

Hesaplamaya bağlı güvenlik için şifreleme algoritması ve kullanılan anahtar uzunluğu önemlidir.

# Hesaplamaya Bağlı Güvenlik

Anahtar Uzunluğu(bit)	Alternatif Anahtar Sayısı	1 çözümleme/ $\mu$ s hızında gereken zaman	$10^6$ çözümleme/ $\mu$ s hızında gereken zaman
24	$2^{24} = 1.6 \times 10^7$	$2^{23} \mu\text{s} = 8.4$ saniye	8.4 $\mu$ 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	$5.4 \times 10^{18}$ yıl
168	$2^{168} = 3.7 \times 10^{50}$	$2^{167} \mu\text{s} = 5.9 \times 10^{36}$ yıl	$5.9 \times 10^{30}$ yıl
26 karakter permutasyonu	$26! = 4 \times 10^{26}$	$2 \times 10^{26} \mu\text{s} = 6.4 \times 10^{12}$ yıl	$6.4 \times 10^6$ yıl

Anahtar uzunluklarına göre hesaplamaya bağlı güvenlik tablosu

# Karmaşıklık Teorisi

---

- ⌚ Karmaşıklık teorisi, bir problemin çözümünün genelde ne kadar zor olduğu ile ilgilenir.
- ⌚ Problem çeşitlerinin sınıflandırılmasını sağlar
- ⌚ Bazı problemler esastan diğerlerinden daha zordur

# ŞİFRELEME ALGORİTMALARI

## Klasik Şifreleme Algoritmaları

Yerine Koyma

Yer Değiştirme

## Modern Şifreleme Algoritmaları

Anahtar  
Kullanımına Dayalı

Simetrik Şifreleme

Asimetrik Şifreleme

Metin İşleme  
Yöntemine Dayalı

Blok Şifreleme

Dizi Şifreleme

- ⌚ **Yerine Koyma (Substitution)** : Şifresiz metindeki her harfi şifreleme alfabetesindeki bir harfle değiştirme.
- ⌚ **Yer Değiştirme (Transposition)** : Şifresiz etindeki harflerin yeri değiştirilir; öteleme (shifting). Çözüm basit alfabe miktarı kadar brute-force atak.
- ⌚ **Blok Şifreleme (Block cipher)** : Şifresiz metnin her bir adımda blok olarak işlenerek çıkış blok olarak elde edilirse blok şifreleme. Elimizdeki anahtar blok uzunluğu kadar.
- ⌚ **Dizi Şifreleme (Stream cipher)** : Şifresiz metin dizi olarak sürekli şekilde işlenirse dizi şifreleme adı verilir. Anahtar uzunluğu metin uzunluğu kadar. Anahtar dağıtım problemi.
- ⌚ **Simetrik Şifreleme (Symetric encrypton)** : Gönderici ve alıcı aynı anahtarı kullanırsa buna simetrik şifreleme.
- ⌚ **Asimetrik Şifreleme (Asymmetric encryption)** : Açık ve gizli olmak üzere iki anahtarın varlığına dayalı şifreleme yöntemidir. Açık anahtar herkes tarafından bilinirken gizli anahtar sadece anahtar sahibi tarafından bilinir.

## Çıg Etkisi

---

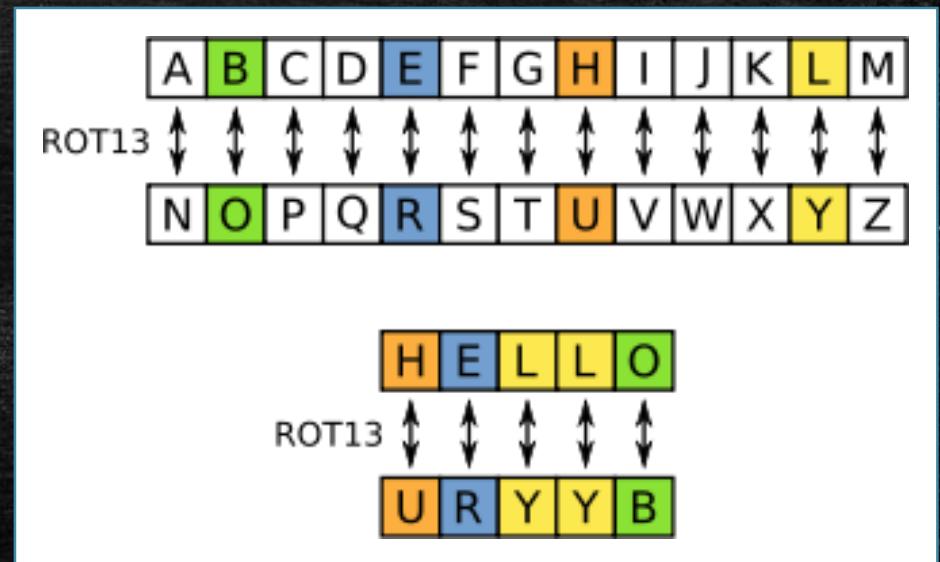
Bir şifreleme algoritmasında anahtar veya şifresiz metindeki küçük değişiklıkların şifreli metin üzerinde büyük değişikliğe neden olmasına çıg (avalanche) etkisi denir.

# Yer Değiştirme



# Yerine Koyma

- ⌚ Yerine koyma şifrelemesinde amaç bir alfabede bulunan karakterlerin her birisinin yerine aynı veya farklı bir alfabeden farklı bir karakter koyarak şifreleme yapmaktadır.
- ⌚ Kamasutra



Sezar Şifreleme (ROT 13)

# Çok Alfabeli Şifreler

---

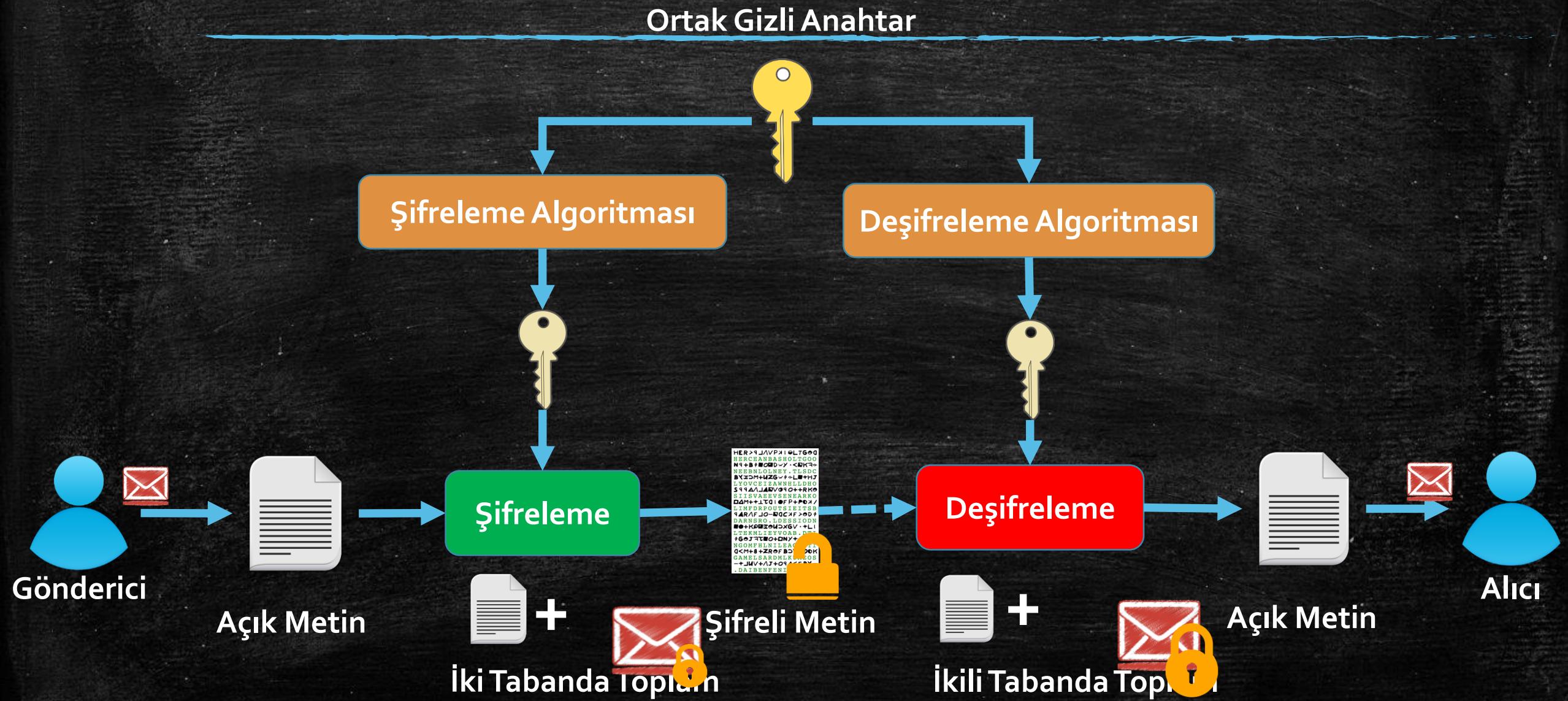
- ⌚ Vigenere şifreleme
- ⌚ 15 yy. dan 19 yy. sonuna kadar kullanıldı.
- ⌚ Kırılmaz şifre olarak anılıyordu.
- ⌚ Babbage tarafından fark motoru ve n-gram analizle çözülmüştür.
- ⌚ Her harf için birden fazla alfabe kullanılır, harfin kelimedeki sırasına göre şifreleme alfabesi de değişir.
- ⌚ Çoklu alfabe şifrenin frekans analizi yöntemiyle çözülmesi zorlaştırır.

# One Time Pads

---

- ⌚ 1918 yılında Gilbert Vernam tarafından bulundu.
- ⌚ Sadece **bir defa kullanılacak** rastgele bitler oluşturulacak bir anahtar önerir.
- ⌚ Kırılamaz bir yöntem önerir.
- ⌚ XOR işleminin aynı anahtar kullanılması durumunda oluşacak zafiyetin giderilmesini amaçlamaktadır.

# Dizi Şifreleme



# 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.
- ⌚ Mesajı bit bit işler.
- ⌚ En meşhur olanı Vernam cipher şifreleyicisidir (aynı zamanda one-time pad denir)

# Dizi Şifreleme ve Anahtar Dağıtımı

---

- ⌚ Mesaj biti kadar anahtar biti gereklidir. Pratikte zordur.
- ⌚ Anahtar dağıtım problemi : Pratikte mag teyp veya CDROM da dağıtılır
- ⌚ Anahtar tamamen rastgele olduğu için koşulsuz güvenlik sağlanır.
- ⌚ Böyle büyük bir anahtar dağıtımı güç olduğu için anahtar dizisi daha küçük(taban) bir anahtardan üretilebilir. Bunun için rasgele sembol fonksiyonları kullanılır.
- ⌚ Her ne kadar bu çok çekici gözükse de pratikte iyi bir kriptografik güçlü rasgele fonksiyon bulmak çok güçtür. Bu hala birçok araştırmacının konusudur.

# Blok Şifreleme

---

- ⌚ Şifreleme ve deşifreleme işleminde metinler sabit uzunluklu dizilere bölünüp blok blok işleme tabi tutulur (örneğin 8, 16, 32 bit veya bayt)
- ⌚ Blok şifreleme algoritmaları FEAL, IDEA, RC5, DES algoritmaları. Çoğu modern şifeleyici blok şifreleme yapar.
- ⌚ Anahtar uzunluğu ise sabittir.
- ⌚ Örneğin DES Şifrelenecek metni 64 bitlik bloklar halinde şifreler, kullandığı anahtar boyu ise yine 64 bittir.

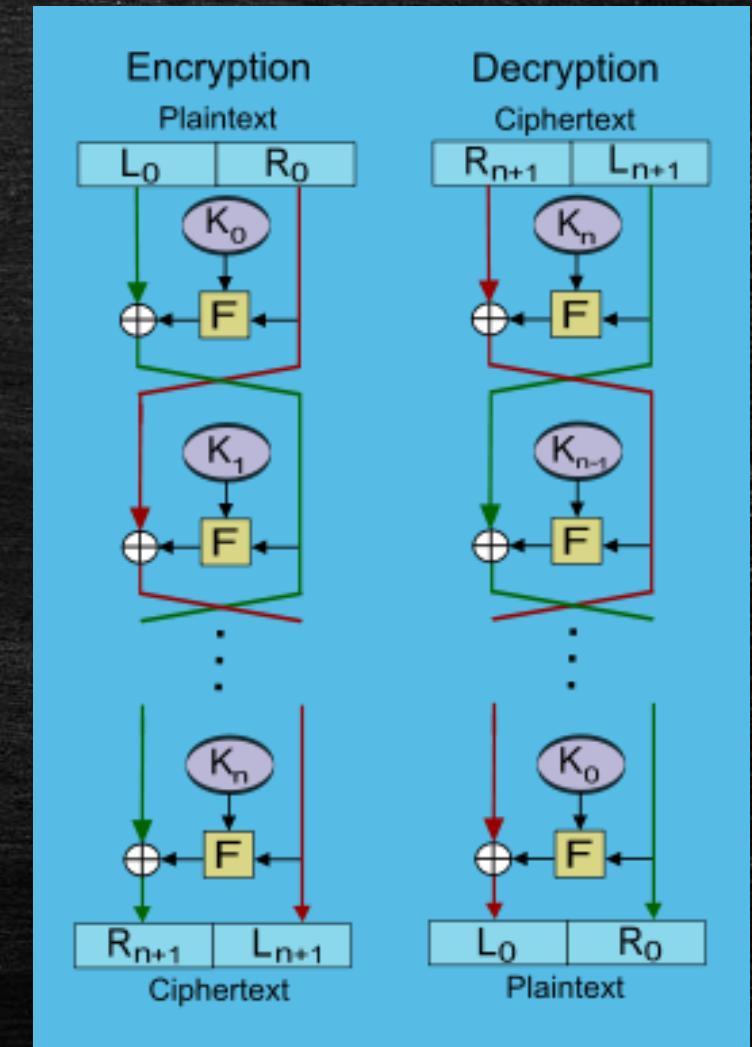
# Feistel Şifreleyici

---

- ⌚ Geleneksel simetrik blok şifreleme algoritmaları 1973'de IBM'de çalışan Horst Feistel tarafından geliştirilen Feistel networkküne dayanır.
- ⌚ Feistel, pratikte yerine koyma ve yer değiştirme işlemlerine alternatif olan ve Shannon tarafından önerilen confusion ve diffusion fonksiyonlarını şifreleme algoritmasında önerdi.
- ⌚ Feistel şifreleyicinin deşifreleme algoritması da aynıdır. Tersinirdir. Şifreli metin giriş olarak kullanılırken alt anahtar tersinden kullanılır.

# Feistel Şifreleyici Yapısı

- ⌚ Karıştırma (Confusion) : Yerine koyma işlemi Feistel şifreleyiciler için karıştırma işlemi olarak kabul edilir.
- ⌚ 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.
- ⌚ Dağıtma (Diffusion) : Veri bitlerinin yerlerinin değiştirilmesi permutasyon şifreleme Feistel şifreleyiciler için dağıtma işlemi olarak kabul edilir.
- ⌚ Diffusion da, şifresiz metnin istatistiksel yapısı, şifreli metnin istatistiğine dağıtılr. Bu, şifresiz metnin her bir dijitinin, şifreli metnin etkilediği dijitlerinin bulunmasıyla sağlanır.
- ⌚ Doğrusal karıştırma (XOR) :

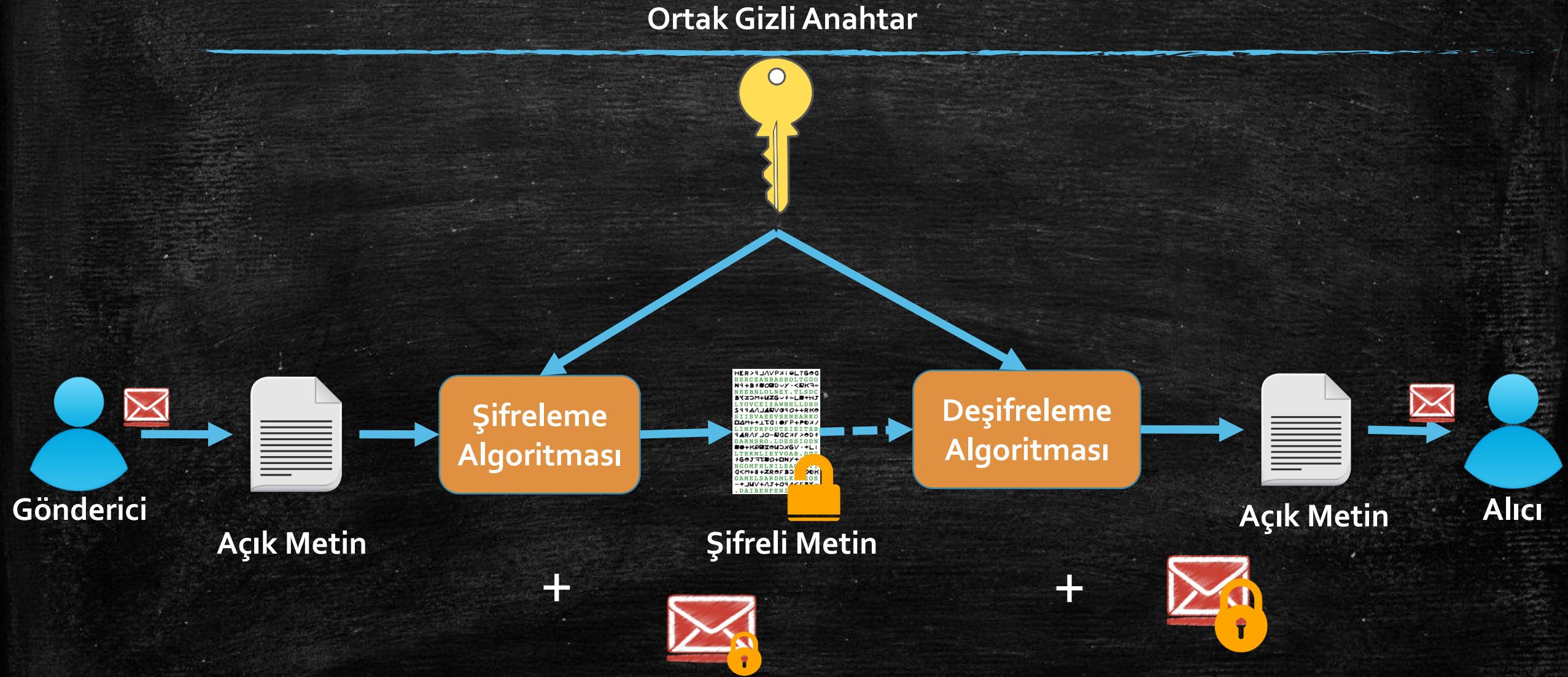


# Feistel algoritmasının önemli parametreleri

---

- ⌚ 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 zorlaştırmır.
- ⌚ Tur Fonksiyonu :Fazla karmaşık olan tur fonksiyonu kriptoanalizi zorlaştırmır.

# Simetrik Şifreleme



# Simetrik Şifreleme

---

- ⌚ Gizli anahtar ile şifreleme ve deşifreleme yapılır.
- ⌚ Anahtar gizliliği önemli. İşlemler tersinirdir ve aynı anahtar kullanılır.
- ⌚ Anahtar dağıtım problemini beraberinde getiriyor; güvenli bir haberleşme kanalı veya güvenilir bir kurye yoluyla anahtar ulaşımı sağlanmalı.
- ⌚ Anahtar gizliliğine dayandığından sıkılıkla yeni anahtar üretimi gereklidir.
- ⌚ DES, 3DES, BLOWFISH, IDEA, CAST128, AES, RC5 yaygın algoritmalar.

# Simetrik Şifreleme

---

- ⌚ Dönüşüm işlemi tersine çevrilebilirdir. Bu nedenle; Algoritmanın gizlenmesi gereklidir. Ancak; Sadece şifreleyici ve şifre çözümünün bileyce bir anahtarın olması algoritma gizliliğini ortadan kaldırır.
- ⌚ Simetrik Şifreleme İçin İki Şeye İhtiyacımız var.
  - Güçlü şifreleme algoritması
  - Sadece gönderici ve alıcı tarafından bilinen gizli anahtar

Şifreleme algoritmasının bilindiğini var saydığımızda. İki tarafa arasında gizli anahtarın paylaşılması problemidir.

Anahtar dağıtımı için güvenli bir yapı oluşturmalıyız.

# DES Simetrik Blok Şifreleme Algoritması

---

- ⌚ Data Encryption Standart (DES) 1974 yılında IBM tarafından geliştirilmiştir.
- ⌚ Temeli Feistel networküne dayanır.
- ⌚ DES bir blok şifrelemedir, 64 bit bloklardaki veriyi şifreler.
- ⌚ 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 sayılabılır ve her zaman değiştirilebilir.

# DES Güvenliği

---

- ⌚ DES'in anahtar uzunluğu 56 bittir ve brute-force atakları için  $2^{56} = 7.2 \times 10^{16}$  anahtar gereklidir. **Yaklaşık 72 quadrillion** ihtimal anlamına gelir.
- ⌚ Mikrosaniye başına bir çözümleme yapan bir makinenin bin yıl gibi bir sürede DES'i kırabileceğini söylemek mümkündür.
- ⌚ Ancak, DES 1970'lerde geliştirilmiş bir algoritmadır. O yıllarda donanım yazılıma oranla daha ön planda olduğu için donanımsal olarak uygulanabilirliği yüksek bir algoritma olarak geliştirildi. Bununla birlikte 1998 yılında donanımsal olarak özel amaçlı olarak tasarlanan bir "DES kırcı" bilgisayar(\$250.000) ile üç günden daha kısa sürede kırılmıştır.
- ⌚ DES güvenliğini artırmak için DES üç kere çağrılmaması yöntemine dayalı 3 DES geliştirildi.

# AES Simetrik Şifreleme Algoritması

---

- ⌚ 128 bit veri, 128/192/256 bitlik anahtar uzunluğuna sahiptir.
- ⌚ Feistel networkü yerine iteratif olarak çalışır.
- ⌚ Veriyi dört bayt lık dört sütunluk bloklar halinde işler.
- ⌚ Her bir tur'da veri bloğunun tamamı üzerinde işlem yapar.
- ⌚ Basit, bilinen saldırırlara karşı dirençli, birçok işlemcide hızlı kod basitliği sağlayacak şekilde tasarlanmıştır.

# Simetrik Şifreleme Algoritmaları

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

Değişik Simetrik Kriptolama algoritmalarının özellikleri

- DES bitti.
- AES ömrünün sonuna yaklaştı.
- Algoritma seçiminde sadece güçlü olmasına bakılmıyor, uygulanabilirlikte önemli.

# Simetrik Şifrelemede Anahtar Dağıtımı

---

- ⌚ A anahtarını seçer ve fiziksel olarak B'ye iletir.
- ⌚ Üçüncü şahıs anahtarını seçer, 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

# İyi Bir Şifreleme Sistemi

---

- ⌚ Hesaplamaya bağlı güvenlik değeri büyük olmalıdır.
- ⌚ Şifreleme, deşifreleme algoritmaları sistem performansını düşürmeyecek kadar hızlı olmalıdır.
- ⌚ Kolay uygulanabilir yapıya sahip olmalıdır.
- ⌚ Geniş kullanım alanına sahip, esnek bir yapıya sahip olmalıdır.
- ⌚ Şifrelenmiş resim, ses verilerinde fazla büyümeye ve veri kaybı olmamalıdır.

# Uygulama – Simetrik Şifreleme

---

- ① **openssl enc -aes-256-cbc -in metin.txt -out sifreliMetin.dat** komutunu kullanarak metin.txt dosyasını AES şifreleme algoritması kullanarak şifreleyin.
  - ① -enc : Şifrele
  - ① -aes-256-cbc : Kullanılacak şifreleme algoritması
  - ① -in : Şifrelenecek dosya
  - ① -out : Dosyanın şifrelendikten sonraki adı.
- ① **cat sifreliMetin.dat** ve **file sifreliMetin.dat** komutları ile şifreli metin içeriğini görüntüleyin.
- ① **openssl enc -aes-256-cbc -d -in sifreliMetin.dat –out metin2.txt** komutunu kullanarak şifreli metni deşifre edin.
  - ① -d : Deşifre et
- ① **diff metin.txt metin2.txt** komutlarıyla iki dosya arasında fark olup olmadığını kontrol edin.
- ① aes-256-cbc parametresindeki 256 ve cbc kavramlarını açıklayın.
- ① Şifreleme ve deşifreleme işlemlerini DES algoritmasını kullanarak gerçekleştirin.

## Uygulama – Simetrik Şifreleme

---

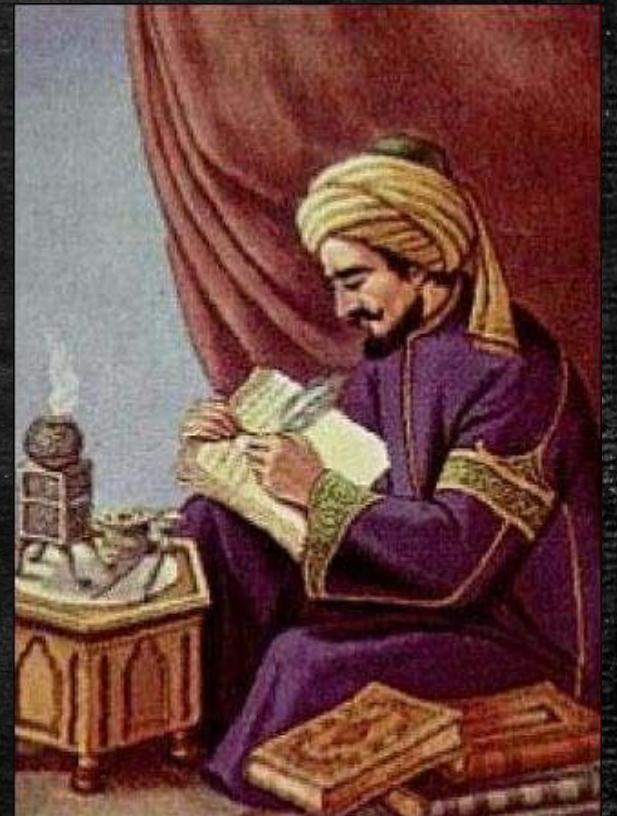
- ⌚ Büyük boyutlu müzik (mp3) dosyasını şifreleyin, deşifreleyin ve de/shifreleme zamanlarını ölçün.
- ⌚ Büyük boyutlu resim dosyasını şifreleyin.
- ⌚ Şifrelenmiş resim görüntüleyin.
- ⌚ Şifrelenmiş dosyayı deşifreleyin ve diff komutuyla orijinal resim ve deşifrelenmiş resim dosyası arasında fark olup olmadığını kontrol edin.

**Resim şifrelemede veri kaybı var. Bu nedenle resim büyük olduğunda direk şifreleme verimli değil. Sıkıştırma işlemi sonrası şifreleme daha verimli.**

# Frekans Analizi

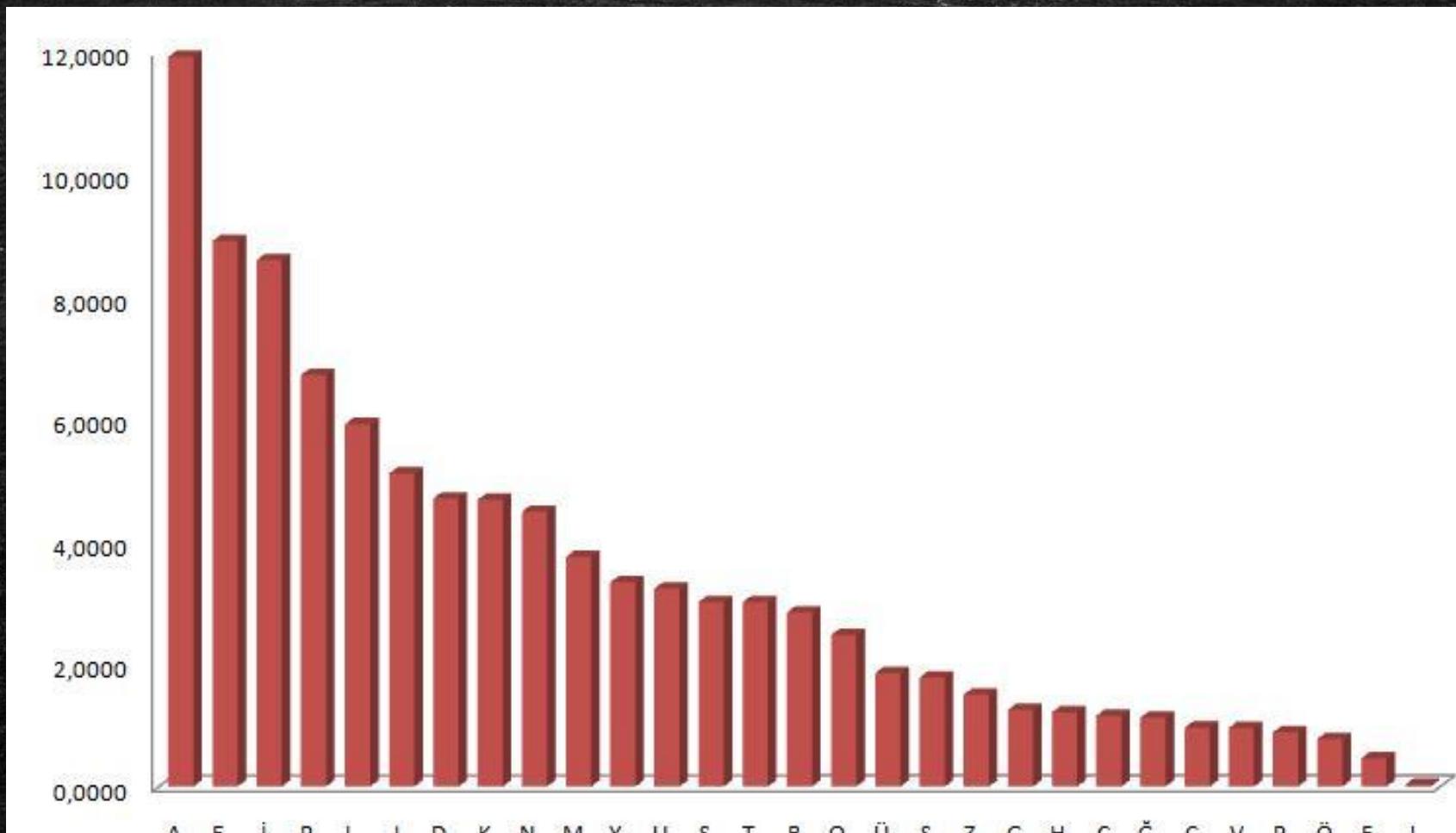
---

- ⌚ Dil içerisindeki harf kombinasyonlarının sıklığına bakar.
- ⌚ İlk olarak 9. yy ortaya atıldı.
- ⌚ Criptografik Mesajların Deşifresi Üzerine isimli kitap.
- ⌚ Orjinali 1987 İstanbul'da Osmanlı arşivlerinde bulundu.



Al - Kindi

# Türkçe Harf Frekans Tablosu



Türkçe Harflerin Kullanım Sıklıkları Tablosu[9]

# Türkçede En Sık Kullanılan 12 Sözcük

S.No	Sözcük	Tekrar Sayısı	Tekrar Oranı (253 milyon sözcük içinde)
1	bir	7588925	0,029995751
2	ve	5564600	0,021994466
3	bu	3213146	0,012700182
4	de	2042980	0,00807502
5	da	2017210	0,007973162
6	için	1475364	0,005831478
7	o	1184617	0,004682281
8	gibi	1163764	0,004599858
9	daha	1136033	0,004490249
10	ama	1040456	0,004112474
11	çok	962295	0,003803538
12	sonra	936503	0,003701593

Türkçede en sık kullanılan sözcüklerin tekrar sayısı ve oranı [10]

# N-Gram Analiz

unigram

C | O | L | D

C | O | L | D

C | O | L | D

C | O | L | D

bigram

C | O | L | D

C | O | L | D

C | O | L | D

trigram

C | O | L | D

C | O | L | D

n-gram (n = 4)

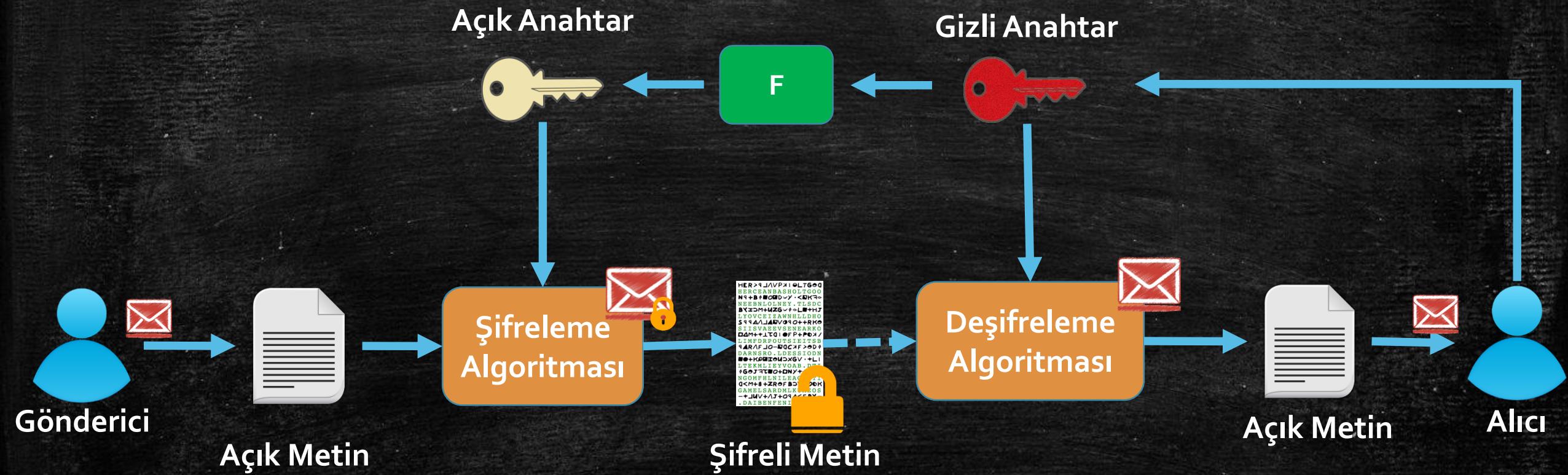
C | O | L | D

## Uygulama : n-gram

---

- ⌚ Türkçe sözlük içerisindeki harflerin n-gram değerlerini bulun.

# Asimetrik Şifreleme



# Asimetrik Şifreleme

---

- ⌚ Açık anahtarlı kripto sistemlerin amacı anahtar dağıtım problemini çözmektir.
- ⌚ Açık-anahtarlı kripto sistemleri üzerine ilk öneri, 1976 yılında Diffie ve Hellman tarafından yapılmıştır.
- ⌚ 1977 yılında Rivest, Shamir ve Adleman **RSA** geliştirdi.
- ⌚ El-Gamal tarafından eliptik eğri Açık-anahtarlı kripto sistem tasarlandı.

# Asimetrik Şifreleme

---

- ① Açık anahtarlı kripto sistemlerin amacı açık ve gizli anahtar mantığına üzerinden anahtar dağıtım problemini çözmektir.
- ① Bilgi gizliliği (şifrelenme) ve bütünlüğü (e-imza) sağlamaası için kullanılır.
- ① Tek yönlü bir mesajlaşma söz konusudur.
- ① Mesaj alıcısı sadece kendisinin bileceği “**Gizli-anahtar**” ve diğer kişilere dağıtabileceği bir “**Açık-anahtar**” dan oluşan anahtar çifti belirler.
- ① Kullanılan anahtar üretim algoritmasına göre bu iki anahtar arasında matematiksel bir bağlantı mutlaka olabilecektir
- ① Asıl amaç, bilinen açık anahtardan gizli anahtarın hesaplanmasıının polinomsal zamanda imkansız olabilmesini sağlayacak bir algoritma olmasıdır.

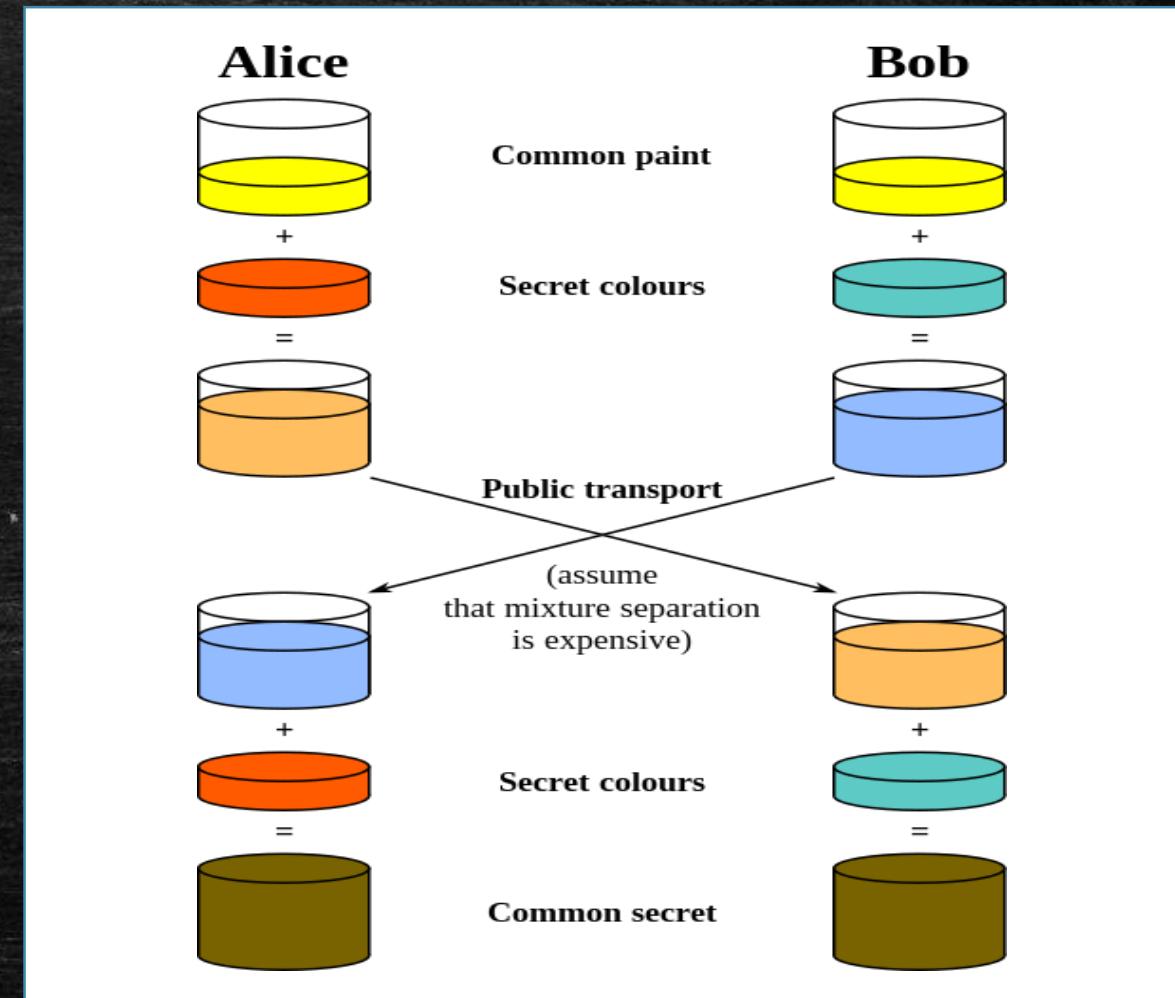
# Asimetrik Şifreleme

---

- ⌚ Mesaj gönderici herkes tarafından bilinen açık anahtarları kullanarak mesajı şifreleyerek alıcıya gönderir.
- ⌚ Mesaj sadece mesaj alıcısı tarafından bilinen gizli anahtar kullanılarak açılabilecektir.
- ⌚ Mesajlaşma tek yönlüdür; göndericiden alıcıya doğru.
- ⌚ 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.
- ⌚ Açık anahtar özel(gizli) anahtardan ve şifreleme hakkındaki diğer bilgilerinden kolaylıkla hesaplanır.

# Diffie Hellman Anahtar Değişim Protokolü

- ⌚ Büyük bir asal sayı seçerler p.
- ⌚ a bir mod primitif elamanıdır.
- ⌚ A'nın f gibi bir gizli sayısı vardır. ( $f < p$ )
- ⌚ B'nin g gibi bir gizli sayısı vardır. ( $g < p$ )
- ⌚ A açıklayacağı x'i hesaplar.  $X = a \cdot \text{power}(f) \bmod p$
- ⌚ B açıklayacağı Y'yi hesaplar.  $Y = a \cdot \text{power}(g) \bmod p$
- ⌚ Sonra anahtarlar aşağıdaki şekilde hesaplanır.
- ⌚ Ortak gizli anahtar;  $K = a \cdot \text{power}(f * g) \bmod p$
- ⌚ B'nin hesaplayabildiği  $K = X \cdot \text{power}(f) \bmod p$
- ⌚ A'nın hesaplayabildiği  $K = Y \cdot \text{power}(g) \bmod p$



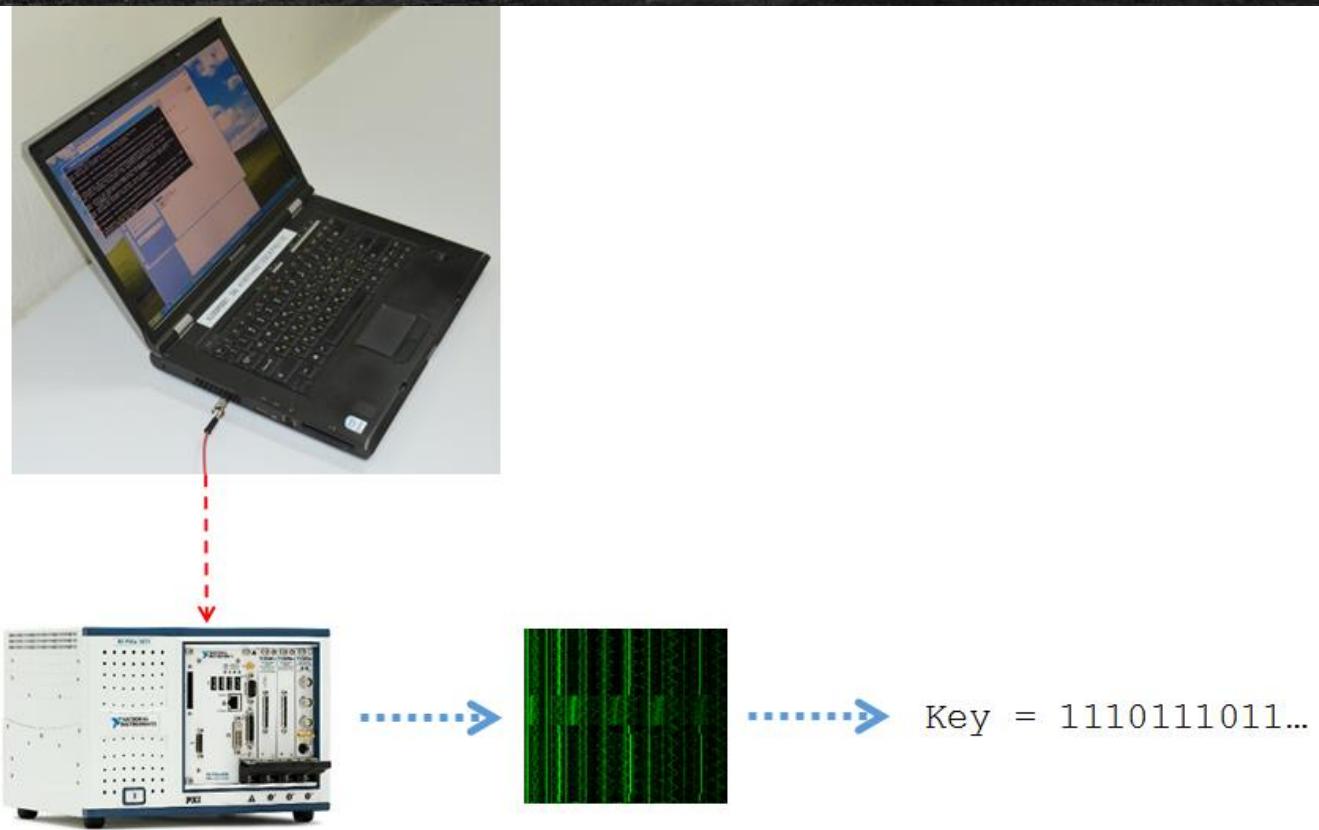
# RSA Açık Anahtarlı Kriptosistem

---

- 🕒 Mesajları şifrelemek, Anahtar değiştirmek ve sayısal imza oluşturmak için kullanılan bir açık anahtarlı tasarımdır.
- 🕒 Güvenliği, büyük sayıların çarpanlarının hesaplanmasıının zorluğuna bağlıdır.

# RSA Kırılması

- ⌚ Shamir akustik kriptanaliz. <https://www.tau.ac.il/~tromer/acoustic/>



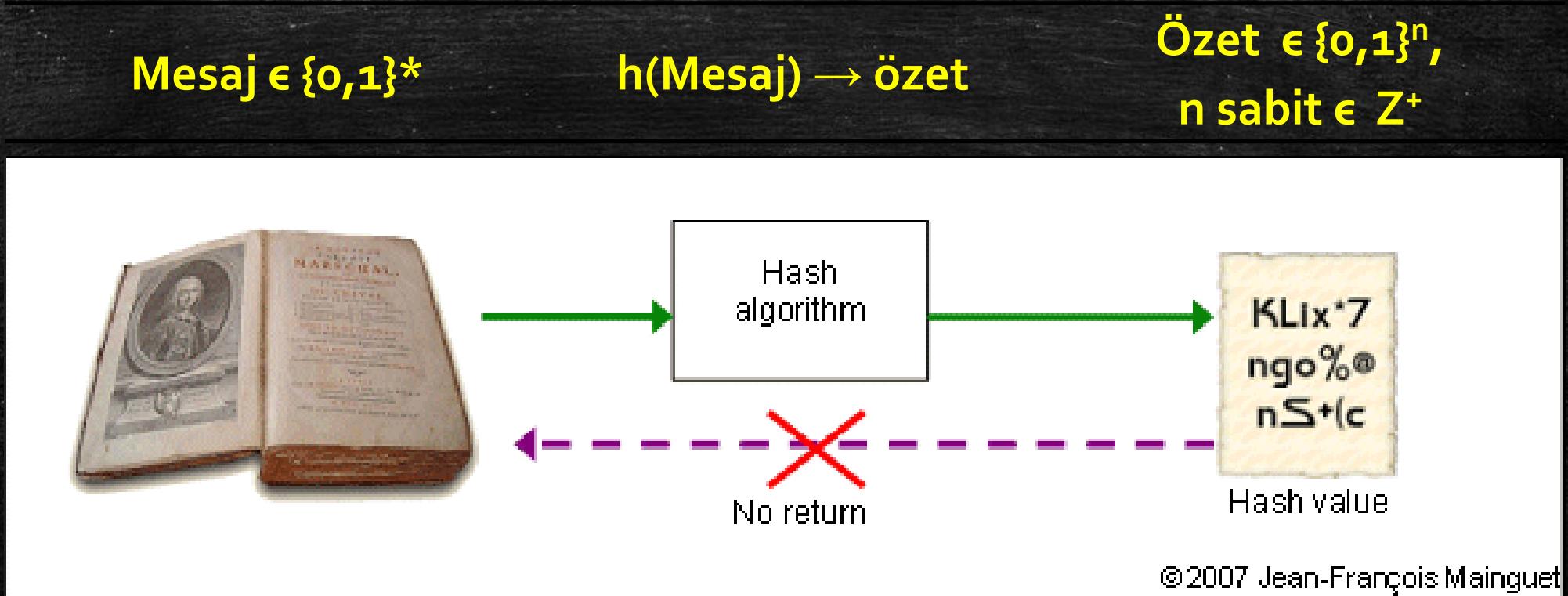
# Uygulama – Asimetrik Şifreleme

---

- ⌚ ***openssl genrsa -out private\_key.pem 1024*** komutunu kullanarak 1024 uzunluğunda özel anahtar oluşturun.
- ⌚ ***cat private\_key.pem*** komutuyla anahtarın içeriğini gözatın ve 1024 değerini değiştirerek yeni anahtarlar oluşturup farklarını gözlemleyin.
- ⌚ ***openssl rsa -in private\_key.pem -out public\_key.pem -outform PEM -pubout*** komutunu kullanarak oluşturduğunuz özel anahtardan açık anahtar oluşturun.
- ⌚ ***cat public\_key.pem*** komutuyla açık anahtarın içeriğini gözatın.
- ⌚ ***openssl rsautl -encrypt -inkey public\_key.pem -pubin -in metin.txt -out sifreliMetin.dat*** komutunu kullanarak açık anahtarla metin.txt dosyasını şifreleyin.
- ⌚ ***openssl rsautl -decrypt -inkey private\_key.pem -in sifreliMetin.dat -out metin3.txt*** komutunu kullanarak şifreli metni deşifre edin.

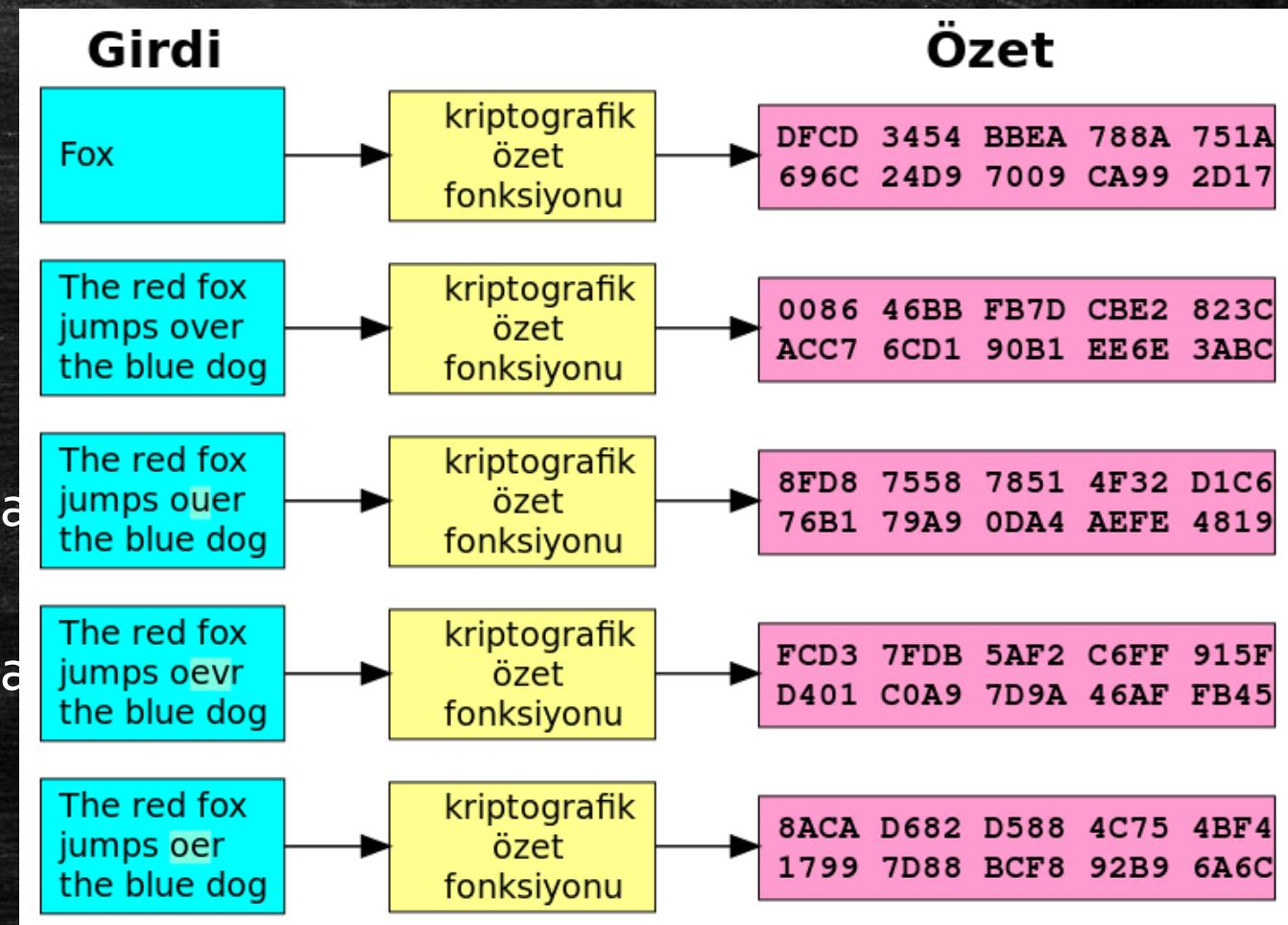
# Özet (Hash) Fonksiyonları

Değişik uzunluktaki bit dizilerini sabit uzunluklu bit dizilerine taşıyan polinomsal zamanda kolay hesaplanabilen fonksiyona “Özetleme Fonksiyonu” denir.



# Özet (Hash) Fonksiyonları

- ⌚ Hash Algoritmaları : HMAC, MD5, SHA-1, SHA-256
- ⌚ MD5 çakışma dirençli değil, aynı hash koda sahip iki farklı doküman oluşturabiliyoruz.
- ⌚  $2^{128}$  farklı md5 kodu oluşturulabilir.
- ⌚ Genelde bütünlük kontrolü, doğrulama işlemleri için kullanılır.
- ⌚ Genelde bütünlük kontrolü, doğrulama işlemleri için kullanılır.
- ⌚ Tersinir değildir.



# Uygulama – Özeti (Hash) Kod

---

- ⌚ **md5Sum \*.txt** komutunu kullanarak txt uzantılı dosyalarınız md5 özeti kodlarını bulunuz.
- ⌚ **echo -n 'Bu metnin hash değerini bul' | md5Sum** komutunu kullanarak ilgili metnin hash değerini hesaplayın. Metindeki karakterlerde küçük değişiklikler yaparak hash koddaki değişimi gözleyin.
- ⌚ Oluşturduğunuz hash değerinden metni geri dönüştürün.
- ⌚ Metnin hash değerini farklı hash hesaplama algoritmaları kullanarak hesaplayın. Sonuçları tartışın.
- ⌚ **hash-identifier** komutunu kullanarak «dac00ob9c696fb4933df91cf620f7c71ac98e481» ve «67aed6282agoacc7f1e958eagde7b53f932be2764a5dcofae1d6143bc3cf49ef» kodlarının özetleme türünü belirleyin.

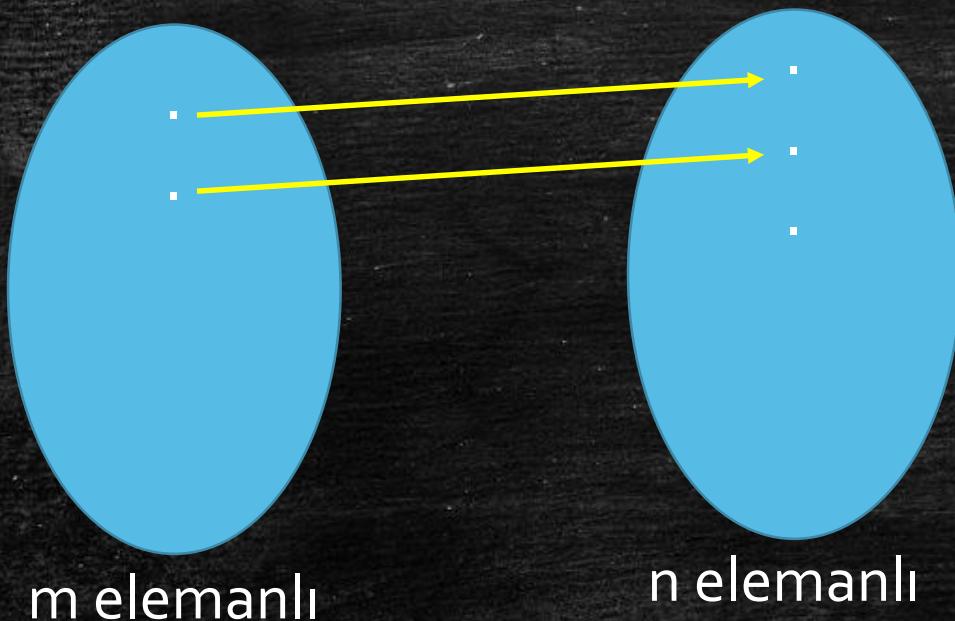
## Uygulama – Özeti (Hash) Kod Saldırısı

---

- ⌚ Özeti kodlar için daha önceden hazırlanmış data setler online olarak mevcut.
- ⌚ Bu veri kümeleri kullanılarak daha önce oluşturulmuş ise özeti kod değeri bulunabilir. (Geri döndürme işlemi değil!)
- ⌚ `findmyhash MD5 -h 098f6bcd4621d373cade4e832627b4f6`

# Hash Collision

Elimizdeki her bir metin için ayrı / tekil özet kod elde edebilir miyiz?



- ✓  $n > m$  birebir fonksiyondur. Elde edebiliriz.
- ✓  $n = m$  birebir örten fonksiyon elde edebiliriz.
- ✓  $n < m$  değer kümesindeki iki farklı eleman görüntü kümesindeki aynı elemana bağlanacak. Elde edemeyiz.

Gerçek hayatımda elimizdeki metin miktarı

# Uygulama – MD5 Hash Collision

Aynı md5 hash koda sahip fotoğrafları bulun?



# Hash Collision

hash = file1

```
if (file1 == hash)  
    run goodProgram  
else  
    run evlProgram
```

hash = file2

```
if (file1 == hash)  
    run goodProgram  
else  
    run evilProgram
```

Md5 Hash Değeri  
==

Çalışma Yapısı Farklı



## Uygulama – MD5 Hash Collision

---

- ⌚ fastcool.exe –o file1 file2 komutunu kullanarak aynı hash koda sahip iki farklı dosya oluşturun.
- ⌚ md5Sum file1 file2 komutuyla dosyaları hash kodlarını karşılaştırın.
- ⌚ diff file1 file2 komutu ile iki dosya arasında fark olup olmadığını kontrol edin.
- ⌚ Hash kod çakışması yöntemini kullanarak farklı çalışan fakat aynı hash koda sahip programlar oluşturun. ([evilize programını kullanabilirsiniz.](#))
- ⌚ Hashclash programını kullanarak hashExtend yaparak farklı iki resim için aynı hash değeri oluşturun.

# XOR (Exclusive OR)

Girdi	Çıktı	
A	B	
0	0	0
0	1	1
1	0	1
1	1	0

$$a \oplus 0 = a$$

etkisiz eleman özelliği

$$a \oplus a = 0$$

yutan eleman özelliği

$$a \oplus b = b \oplus a$$

değişme özelliği

$$a \oplus b \oplus a = b$$

olduğunu gösterelim

$$a \oplus b \oplus a = a \oplus a \oplus b$$

değişme özelliği

$$a \oplus a \oplus b = 0 \oplus b$$

yutan eleman

$$a \oplus a \oplus b = b$$

etkisiz eleman özelliği

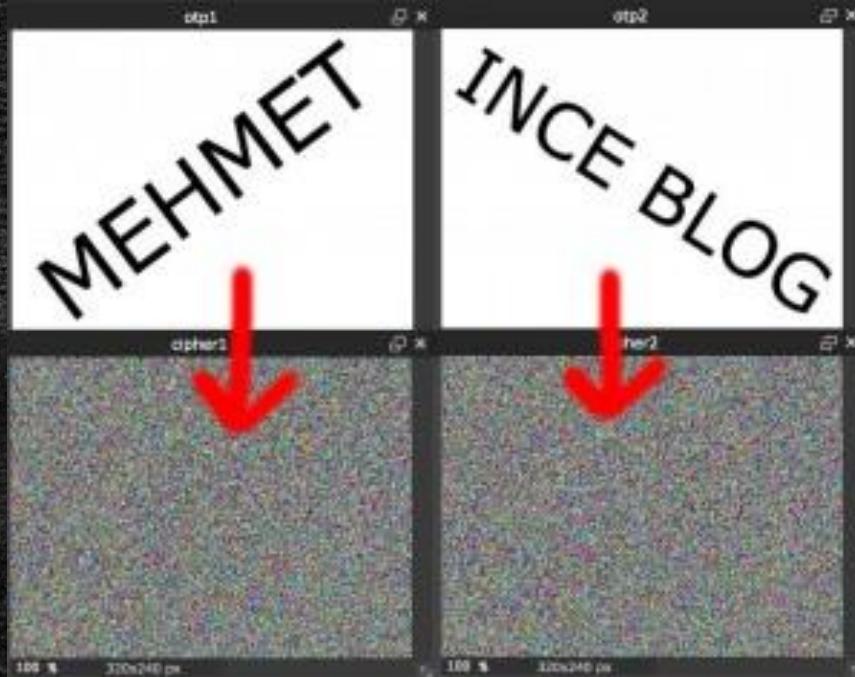
## Şifreleme

$$\begin{array}{cccc}
 01010111 & 01101001 & 01101011 & 01101001 \\
 \oplus & 11110011 & 11110011 & 11110011 & 11110011 \\
 = & 10100100 & 10011010 & 10011000 & 10011010
 \end{array}$$

## Deşifreleme

$$\begin{array}{cccc}
 10100100 & 10011010 & 10011000 & 10011010 \\
 \oplus & 11110011 & 11110011 & 11110011 & 11110011 \\
 = & 01010111 & 01101001 & 01101011 & 01101001
 \end{array}$$

# XOR (Exclusive OR)



$$a \oplus b$$

$$a \oplus c$$

$$a \oplus b \oplus a \oplus c$$

$$b \oplus c$$

A large black rectangular area contains the text "INCE MEHMET BLOG" in white, rotated diagonally. A yellow arrow points from the right side of the "otp1" window towards this result.

## Uygulama- XOR

---

- ⌚ Xor\_implement.py kodunu kullanarak herhangi bir metin ve anahtarın xor işlemi ile şifreleyip deşifreleyin.
- ⌚ İki farklı resim oluşturun ve xor\_same\_key.py kodunu kullanarak resimleri xor işlemiyle şifreleyin. Daha sonra şifreli resimleri xor'layarak sonucu gözlemleyin.
- ⌚ Aynı işlemi renkli resimleri için tekrar edin.

Endoding != Encryption

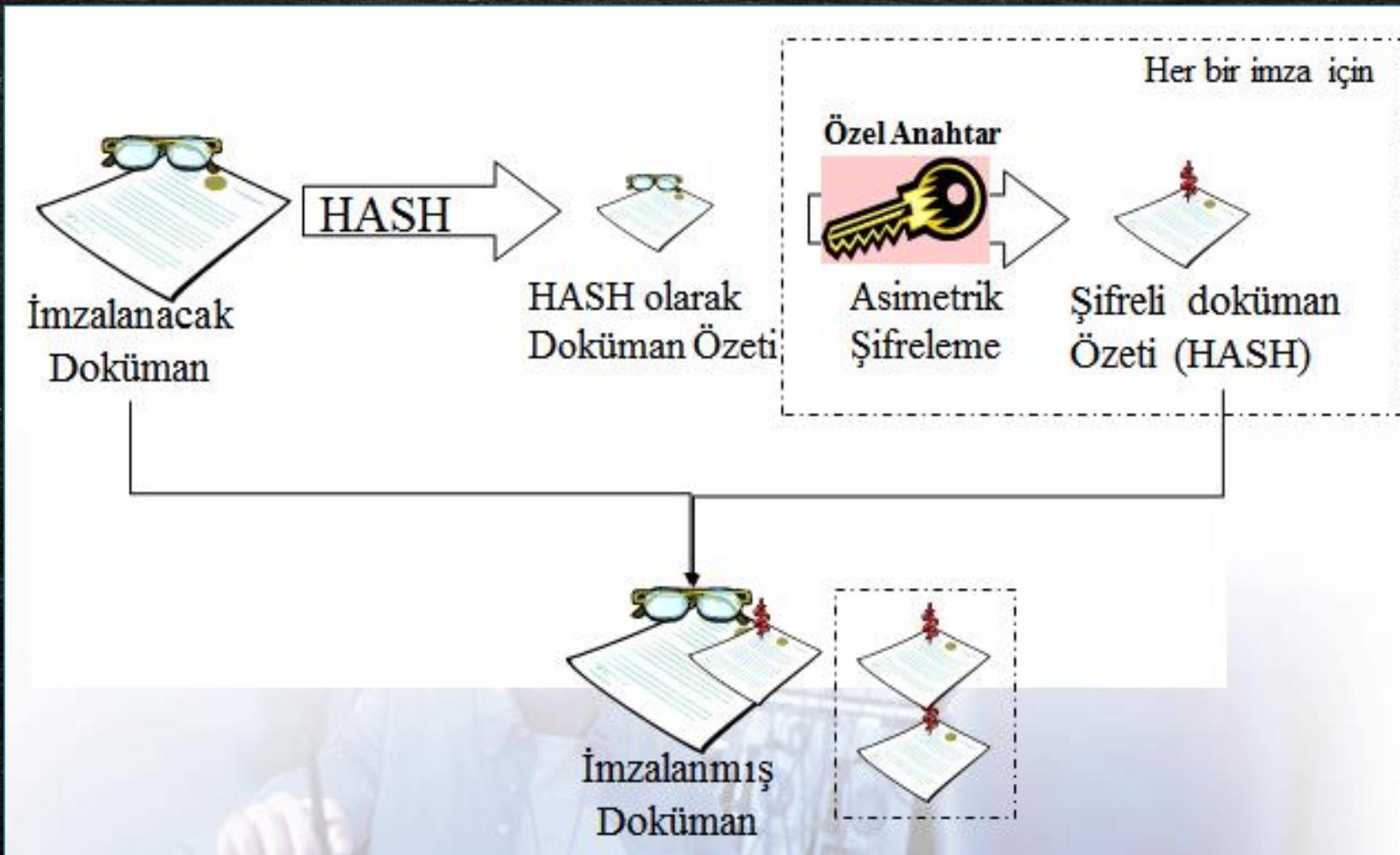
---

## Uygulama - Encoding

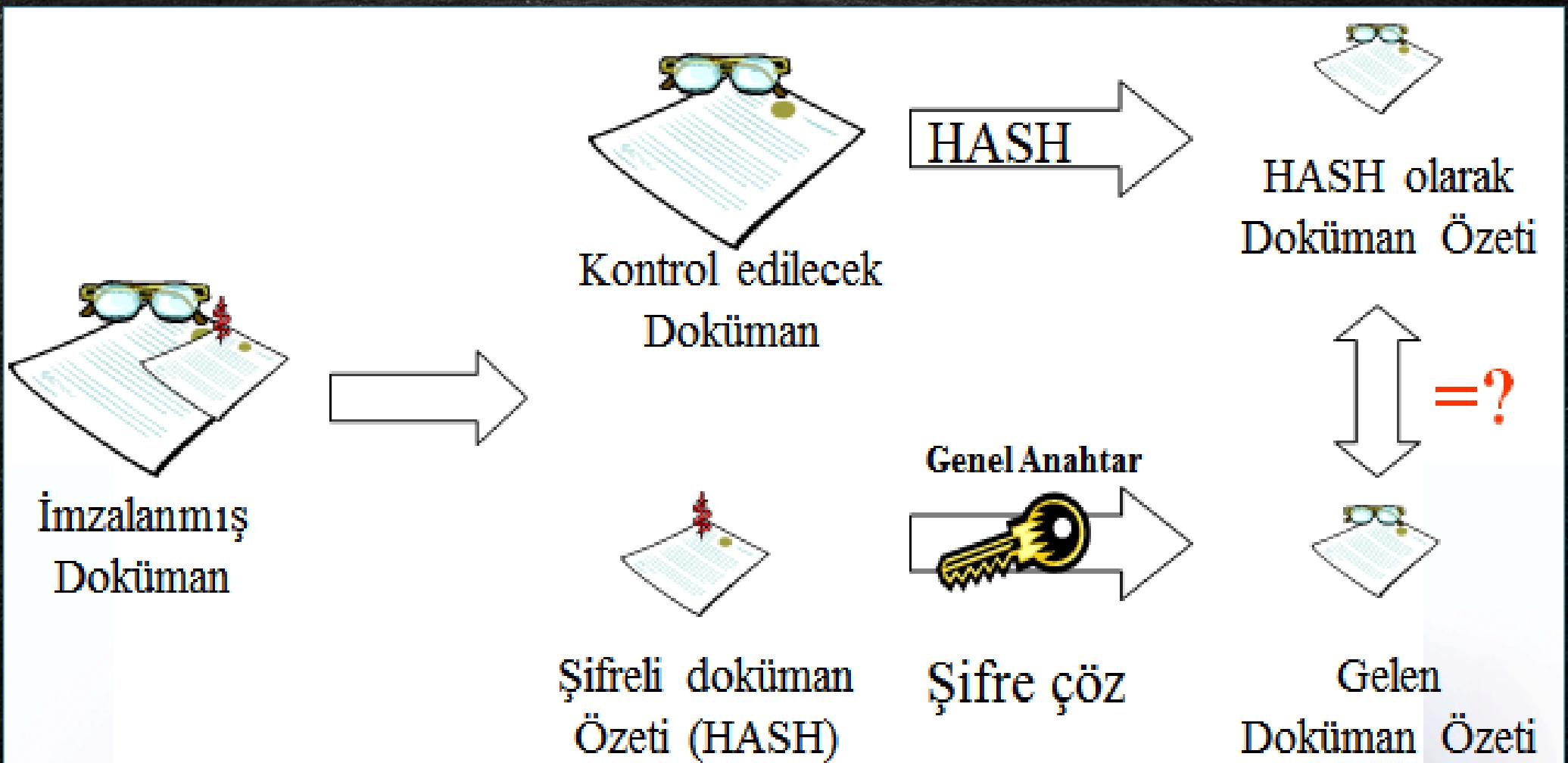
---

- ⌚ **echo -n 'Bu metni encode et' | base64** komutunu kullanarak ilgili metnin base64 encoding çıktısını oluşturun. Metinde bulunan karakterleri değiştirerek sonuçları gözlemleyin. Tamamlama (padding) olup olmadığını kontrol edin.
- ⌚ **echo -n <base64 değeri gelecek> | base64 -d** komutunu kullanarak base64 ile encode ettiğiniz değeri decode edin.

# E-imza yapısı : imzalama



# E-imza yapısı : imza kontrolü



# Pretty Good Privacy (PGB)

---

- ⌚ Açık anahtarlama yaklaşımıdır.
- ⌚ E-posta sistemlerinde sıkça kullanılır.

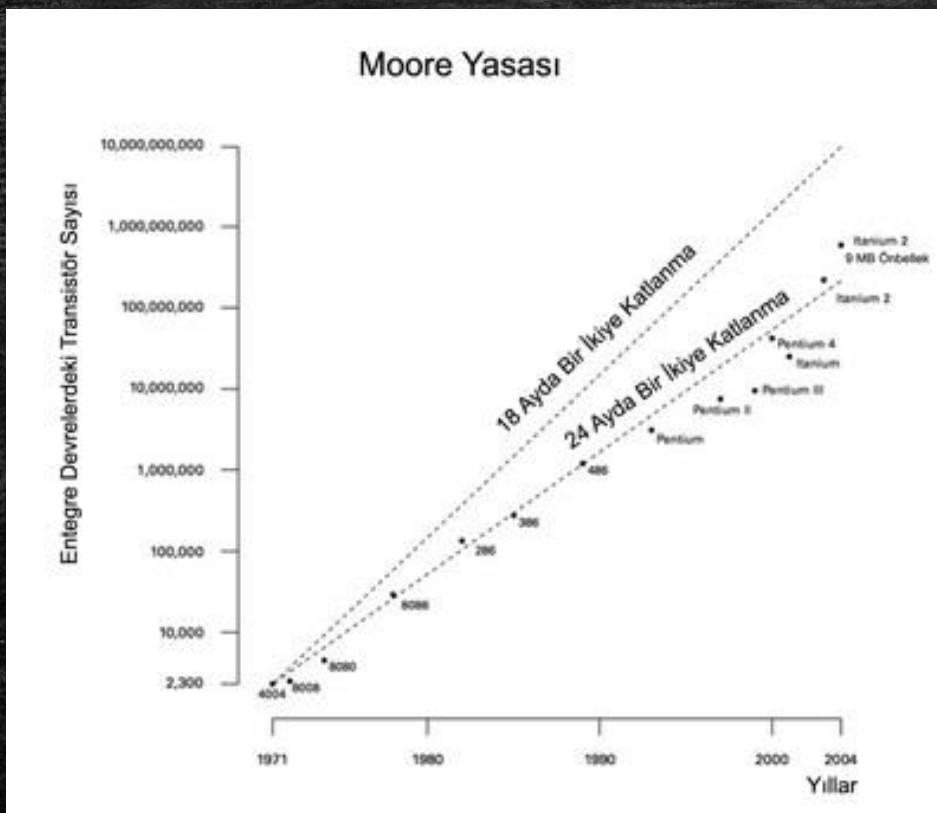
# Eliptik Eğri Kriptografi

---

# Kuantum Kriptografi

---

# Moore Yasası



- ⌚ Intel şirketinin kurucularından Gordon Moore'un 1965 yılında *Electronics Magazine* dergisinde yayınlanan makalesi
- ⌚ Mikroişlemciler içindeki transistör sayısı her yıl iki yılda bir iki katına çıkacaktır.
- ⌚ Bilgisayarların işlem kapasitelerinde büyük artışlar yaratacak
- ⌚ Üretim maliyetlerinin ise aynı kalacak
- ⌚ 1965 yılından bu yana bu yasa çoğunlukla geçerli olmuştur

Atom fiziksel boyutlarının sınırlılığı ve küçük yapıların yüksek frekanslarda çalıştırılmasında ortaya çıkan çalışma düzensizlikleri nedeniyle Moore yasasının kısa bir süre içerisinde geçerliliğini yitireceğini göstermektedir.

# Kriptografi Gelecek Çalışmalar

---

- 🕒 Kriptografik algoritmaların gücü yazılım olarak ölçülse de hardware alanında ki saldırılarından etkilenir. (Bknz: DES, RSA)

# Kaynakça

---

1. [Hobit alfabesi dönüştürücü](#)
2. [Fiorin alfabesi dönüştürücü](#)
3. [Braille alfabesi dönüştürücü](#)
4. [Babil alfabesi dönüştürücü](#)
5. [Barkod alfabesi dönüştürücü](#)
6. [Mors alfabesi dönüştürücü](#)
7. [Antik misir alfabesi dönüştürücü](#)
8. [Kiril alfabesi dönüştürücü](#)
9. [Türkçe Harf Kullanım Sıklıkları](#)

# Kaynakça

---

10. Türkçede en sık kullanılan sözcükler
11. Meaningful MD5 Collisions: Creating executables
12. Marc Steves web sayfası, Hash Collision
13. This was inspired by Richter, Wolfgang (August 3, 2012), "Unbreakable Cryptography in 5 Minutes"
14. Mehmet İnce, Crypto 101 - [1] Merhaba Exclusive OR (XOR)