BİLGİ GÜVENLİĞİ

Kriptoloji Nedir?

Kriptoloji kelimesi Eski Yunanca'da yer alan "kryptos logos" kelimelerinden gelmektedir. "kryptos" kelimesi "gizli dünya" anlamını, "logos" ise "sebep-sonuç ilişkisi kurma, mantıksal çözümleme alanı" anlamını taşımaktadır.

Kriptoloji, haberleşen iki veya daha fazla tarafın bilgi alışverişini emniyetli olarak yapmasını sağlayan, temeli matematiksel zor problemlere dayanan tekniklerin ve uygulamaların bütünüdür.

Kriptolojinin iki temel alt dalı vardır. Bunlar Kriptografi ve Kriptoanaliz.

Kriptografi, belgelerin şifrelenmesi ve şifresinin çözülmesi için kullanılan yöntemlere verilen addır.

Kriptoanaliz, kriptografik sistemlerin kurduğu mekanizmaları inceler ve çözmeye çalışır. Kriptoanalizin kriptoloji içindeki önemi çok büyüktür çünkü ortaya konan bir şifreleme sistemini inceleyerek, zayıf ve kuvvetli yönlerini ortaya koymak için kriptoanaliz kullanılır.

Kriptolojinin başlıca kullanım alanı hareket halindeki veya depolanmış bilginin şifrelenmesi ve istendiğinde bu şifrenin çözülmesidir.

Kriptoloji'nin Tarihçesi (derste bahsedilen kısımları)

- M.Ö. 1900 dolaylarında Mısırlı katip yazdığı kitabelerde standart dışı hiyeroglif işaretleri kullanmıştır.
- M.Ö. 60-50 Julius Caesar, normal alfabedeki harflerin yerini değiştirerek oluşturduğu şifreleme yöntemini devlet haberleşmesinde kullandı. Bu yöntem, açık metindeki her harfin alfabede kendisinden 3 harf sonraki harfle değiştirilmesine dayanıyordu.

...

- II. Dünya Savaşı'nda Almanlar Arthur Scherbius tarafından icat edilmiş olan Enigma makinesini kullandılar. Bu makine Alan Turing ve ekibi tarafından çözüldü.
- 1976'da DES, ABD'nin FIPS 46 standardı olarak açıklandı.
- 1978'de RSA bulundu.
- 1985'de ECC "
- 1990'da IDEA "
- 1997'de ABD'nin NIST kurumu DES'in yerini alacak bir simetrik algoritma bulabilmek için bir yarışma açtı. 2001'de bu yarışmayı kazanan Rijndael Algoritması, AES adıyla standart haline getirildi.

Haberleşmede Emniyet

Haberleşmedeki emniyet öğeleri aşağıdaki gibidir.

- **Gizlilik** : taşınan bilginin içeriğinin gizli kalmasıdır.
- **Bütünlük** : taşınan bilginin içeriğinin yolda değiştirilememesidir.
- **Kimlik doğrulama** : bilgiyi gönderen kişinin kimliğinin doğruluğundan emin olmaktır
- **İnkar edememezlik**: bilgiyi gönderen veya işleyen kişinin yaptığı işi sonradan inkar edememesidir.
- **Haberleşmenin sürekliliği**: haberleşmenin kesintiye uğramadan yapılmasıdır.

Günlük hayatta bu emniyet gereksinimlerini karşılamak için aşağıdaki yöntemler kullanılmaktadır:

- Gizlilik sağlamak için mühürlü zarf,
- Bütünlük " imza, barkod, damgalama,
- Kimlik doğrulaması " noter, kimlik kartı, trafik ehliyeti, kişinin şahsen başvuru yapması,
- İnkar edememezlik " imza, alındı, onay
- Haberleşmenin sürekliliği " farklı, birbirine alternatif iletişim yolları.

Elektronik Tehditler

Haberleşen iki taraf, bilgisayar ağları, kablolu veya kablosuz ağlar kullanarak bir bilgiyi, mesajı bir taraftan diğerine iletirler. Elektronik ortamlarda haberleşen taraflar çeşitli tehditlerle karşı karşıya kalırlar. Bunlar;

- Gizlilik ihlali
- Bütünlük ihlali
- Kimlik doğrulama ihlali
- İnkar edememezlik ihlali
- Süreklilik ihlali.

Gizlilik ihlali; 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.

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

Kimlik doğrulama ihlali; Saldırgan, alıcıya göndericinin kimliğini taklid ederek bir mesaj gönderebilir. Bu durumda eğer alıcı güvenilir bir kimlik doğrulaması yapmıyorsa yanlış mesajlarla kandırılabilir. Bu tehdit **oluşturma tehdididir.**

İnkar edememezlik ihlali; Mesajı gönderen veya alan tarafın inkar etmesi sözkonusu olabilir.

Süreklilik ihlali; Saldırgan, haberleşen iki taraf arasındaki hattı kullanılamaz hale getirir.

Elektronik Tedbirler

Elektronik tehditlere karşı alınabilecek elektronik tedbirler, aşağıdaki gibidir.

- **Gizlilik** sağlamak için veri **şifreleme** yöntemleri kullanılır.
- **Bütünlük** sağlamak için **özetleme** algoritmaları, mesaj özetleri, sayısal (elektronik) imzalar kullanılır.
- **Kimlik doğrulaması** için özetleme algoritmaları, mesaj özetleri, **sayısal (elektronik) imza**, sertifikalar kullanılır.
- **İnkar edememezlik** için **sayısal (elektronik) imzalar**, işlem kayıtları kullanılır.
- **Süreklilik** için yedek sistemler, bakım, yedekleme, alternatif haberleşme kanalları kullanılır.

Elektronik Emniyet Yöntemlerinin Karşılaştırılması

	Kimlik Doğrulama	Gizlilik	Bütünlük	İnkar edememe
Anti-Virüs			✓	
Güvenlik duvarları	✓	✓		
Erişim denetimi	✓	✓		
Şifreleme		✓		
Açık Anahtar Alt Yapısı	✓	✓	✓	✓

- Anti-Virüs programları, CRC32 gibi "checksum" (bir çeşit özet) kullanarak bilgisayarda ki programların kontrol dışı değiştirilip değiştirilmediğini kontrol ederler. Bu nedenle sadece bütünlük hizmetini verebilirler.
- Güvenlik duvarları (firewall), kimlik doğrulama yaparak belirli kaynaklara erişimi sınırlarlar. Bu nedenle sadece kimlik doğrulama ve gizlilik hizmetlerini sağlarlar.
- Şifreleme programları veya yöntemleri tek başlarına kullanıldığında sadece gizlilik hizmetini verebilirler.
- Açık Anahtar Altyapısı kimlik doğrulama, gizlilik, bütünlük, inkar edememe hizmetlerini sağlayarak çok daha kapsamlı çözüm sunmaktadır.

Şifreleme

Şifreleme, bir bilginin özel bir yöntemle değiştirilerek farklı bir şekle sokulması olarak tanımlanabilir. Şifreleme işlemi sonucunda ortaya çıkan yeni biçimdeki bilgi, şifre çözme işlemine tabi tutularak ilk haline dönüştürülebilir.

Şifreleme yönteminde aranan bir takım özellikler vardır. Bunlar;

1 → Şifreleme ve şifre çözme işleminin zorluğu ihtiyaç duyulan güvenlikle doğru orantılı olmalıdır. Çok önemli olmayan bir bilginin şifrelenmesi için, bilginin kendisinden daha fazla iş gücü ve zaman harcanması verimli olmayacaktır.

2 → Anahtar seçimi ve şifreleme algoritması özel koşullara bağlı olmamalıdır. Şifreleme yöntemi her türlü bilgi için aynı şekilde çalışmalıdır.

3 → Sürecin gerçekleşmesi mümkün olduğunca basit olmalıdır. Çok karışık bir sistemin gerçekleşmesi hem hatalara sebep olabilir hem de performans açısından tatmin edici olmayabilir.

4 → Şifrelemede yapılan hatalar sonraki adımlara yansımamalı ve mesajın tamamını bozmamalıdır. Saldırılara karşı bu özellik koruyucu olacaktır. Ayrıca haberleşme hattında meydana gelen bir hata bütün mesajın bozulmasına neden olmayacağı için bu özellik tercih edilmektedir.

5 → Kullanılan algoritmanın karıştırma özelliği olmalıdır. Mesajın şifrelenmiş hali ile açık hali arasında ilişki kurulması çok zor olmalıdır.

6 → Kullanılan algoritmanın dağılma özelliği olmalıdır. Mesajın açık hali şifreli hale gelirken içerdiği kelime ve harf grupları şifreli mesajın içinde olabildiğince dağıtmalıdır.

Basit Şifreleme Yöntemleri

Basit şifreleme yöntemleri genellikle kağıt kalem kullanılarak gerçekleştirilebilen, çok karışık matematik temellere dayanmayan sistemlerdir. En gelişmiş örnekleri mekanik cihazlar olan basit şifreleme yöntemleri, elektronik cihazların kullanılmaya başlanmasıyla beraber ortadan kalkmıştır.

Basit şifreleme yöntemleri;

- Mono Alfabetik Şifreleme (Sezar yöntemi) — Örnek
- Poli Alfabetik Şifreleme (Vigenere yöntemi) — Örnek
- Tek Kullanımlık karakter dizisi (one-time-pad) (Vernam Şifreleme Yöntemi) — Örnek

şeklindedir.

Mono Alfabetik Şifreleme

• En eski ve basit şifreleme yöntemlerinden birisidir. Sezar yöntemi mono alfabetik şifrelemenin tipik bir örneğidir. Sezar döneminde kullanılan bu yöntemde harflerin yeri değiştirilir. Şifrelenecek metindeki harfler, alfabede 3 harf kaydırılarak değiştirilir.

• Sezar şifresi: $c_i = E(p_i) = p_i + 3 \mod 29$

Açık mesaj: Gizli Bilgi
Şifreli mesaj: Ilcol Dlorl

• Bu yöntemin biraz daha gelişmişi olan **tablo yönteminde** ise alfabedeki her harf başka bir harfle yer değiştirir ama bu bir kurala bağlı olmadan yapılır.

A B C Ç D E F G Ğ H I İ J K L M N O Ö P R S Ş T U Ü V Y Z
C Ç A V Y J Ş Ü Z K Ö T U E N O İ P F P G I L Ğ H R M B D S

• Monoalfabetik şifreleme yöntemleri bilgisayar yardımıyla çok kısa sürede kırılabilir. Bu yöntemler kullanılan dildeki harflerin yerini değiştirir ama harflerin kullanım sıklığını (frekansını) değiştirmez. Türkçe'de A yerine C kullanılması, frekans analizi ile C nin A olduğu bulunabilir.

Poli Alfabetik Şifreleme

• Bu tip şifrelemede mono alfabetik şifrelemeden farklı olarak, bir harf değiştirilince her seferinde aynı harfle dönüşmez. Bu yöntemlere güzel bir örnek Vigenere tablosudur.

	A	B	C	Ç	D	E	F	G	Ğ	H	I	İ	J	K	...
A	A	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k	...
B	b	c	ç	d	e	f	g	ğ	h	ı	i	j	k	l	...
C	c	ç	d	e	f	g	ğ	h	ı	i	j	k	l	m	...
Ç	ç	d	e	f	g	ğ	h	ı	i	j	k	l	m	n	...
D	d	e	f	g	ğ	h	ı	i	j	k	l	m	n	o	...
E	e	f	g	ğ	h	ı	i	j	k	l	m	n	o	ö	...
F	f	g	ğ	h	ı	i	j	k	l	m	n	o	ö	p	...
G	g	ğ	h	ı	i	j	k	l	m	n	o	ö	p	r	...
Ğ	ğ	h	ı	i	j	k	l	m	n	o	ö	p	r	s	...
H	h	ı	i	j	k	l	m	n	o	ö	p	r	s	ş	...
I	ı	i	j	k	l	m	n	o	ö	p	r	s	ş	t	...
İ	i	j	k	l	m	n	o	ö	p	r	s	ş	t	u	...
J	j	k	l	m	n	o	ö	p	r	s	ş	t	u	ü	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

(1) → kesişim
(2) → kesişim

• Bu yöntemde oluşturulan tablo ve bir anahtar kelime ile şifreleme yapılır.

• Şifreleme
Açık M.: Buluşma yeri Ankara
Anahtar: Kalem

	B	U	L	U	Ş	M	A	Y	E	R	İ	A	N	K	A	R	A
	K	A	L	E	M	K	A	L	E	M	K	A	L	E	M	K	A
	L	U	Z	A	Ğ	Z	A	J	I	F	U	A	B	Ö	M	D	A

(1) → satır
(2) → sütun

• Şifre çözme: Aynı mantık (şifrelemetin ile anahtar)

→ Tek Kullanımlık Karakter Dizisi (One-time pad)

Bu basit şifreleme yönteminde rastgele üretilen bir karakter (harf veya rakam) dizisi kullanılarak şifreleme yapılır. Açık mesaj içinde yer alan her karakter, üretilen dizide karşısına denk gelen karakterle işleme sokularak (ör: mod toplama işlemi) şifreli mesaj elde edilir. Mesajı çözmek için rastgele dizinin bilinmesi gerekliidir. Bu yönteme **Vernam şifreleme yöntemi** denir.

Açık Mesaj : BULUŞMAYERİANKARA
Rastgele dizi : SEFRYPLCNMLJKHFGH
Şifreli mesaj : RLBYBOL...

Bu yöntemin güvenliği rastgele üretilen diziye bağlıdır. Bu dizi gerçekten rastgele üretilmelidir, eğer bir kurala bağlı olarak üretilirse ve bu kural saldırgan tarafından bilinirse sistem kırılabilir. Bu tehdit dışında sistem mükemmel bir sistemdir, ve ilk olarak 1917'de bulunup **"teletype"** makinelerinde kullanılmıştır.

Kriptoanaliz Yöntemleri

Kriptoanaliz bir şifreleme sistemini veya sadece şifreli mesajı inceleyerek, şifreli mesajın açık halini elde etmeye çalışan kriptoloji disiplinidir. Kriptoanaliz çalışması sırasında kriptoanaliz yapan kişinin elinde çoğu zaman çok az bilgi vardır. Değişik durumlar aşağıda listelenmiştir.

- **Şifrelenmiş mesaj analizi**; Kriptoanaliz yapan kişinin elinde sadece şifreli bir mesaj vardır. Mesajın açık hali ile ilgili hiç bir ipucu yoktur.
- **Tam bir açık mesajın analizi**; Kriptoanaliz yapan kişinin elinde bütün bir mesajın hem açık hali hem de şifreli hali vardır.
- **Yarım olarak elde edilmiş açık mesajın analizi**; Kriptoanaliz yapan bir kişinin bir mesajın açık halinin belirli bir kısmına ve şifreli halinin tamamına sahiptir.
- **İstenen açık mesajın şifrelenmiş halinin analizi**; Kriptoanaliz yapan kişi istediği açık mesajın şifreli halini elde edebilmektedir. Bu şifrelemeyi yapan cihazın veya yazılımın çalışan bir kopyasına sahip olarak veya şifrelemeyi yapan sistemi fark edilmeden kullanmakla mümkün olur.
- **Şifreli mesajın şifreleme algoritması bilinerek analizi**; Kriptoanaliz yapan kişi elindeki şifreli mesajın hangi yöntemle şifrelendiğini bilmektedir.

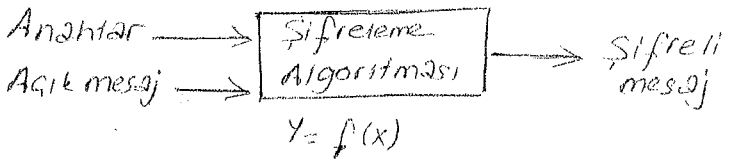
Kullanılan kriptoanaliz yöntemleri ise aşağıdaki gibidir.

→ **Kaba kuvvet yöntemi**: Bir şifreleme alg.'nın kullanabileceği tüm anahtarları tek tek dener.

→ **Diferansiyel kriptoanaliz**: Bilinen açık-şifreli mesaj çiftleri arasındaki farkların hesaplanması temeline dayanır.

Güvenli Şifreleme Yöntemleri

Güvenli şifreleme yöntemleri, klâsik şifreleme yöntemlerinin zayıf yönlerini ortadan kaldıran ve kriptoanalize karşı dirençli olan algoritmalarla gerçekleşir. Bu yöntemler elektronik sistemlerde kullanılır ve binary düzende saklanan ve taşınan bilgi üzerinde uygulanır. Bu nedenle anahtar olarak **bit dizileri** kullanılır.



Bir şifreleme algoritmasının güvenliğini belirleyen en önemli değişkenlerden birisi anahtar uzunluğudur. Örneğin 64 bitlik bir anahtar kullanan şifreleme algoritması için toplam anahtar sayısı $2^{64} = 10^{19}$ adettir. Şifrelemede bu anahtarlardan herhangi birisi kullanılabileceği için, bu anahtarı tahmin yoluyla elde etme olasılığı çok düşüktür.

Güvenli şifreleme temel olarak iki çeşittir:

- Simetrik kriptografi
- Asimetrik kriptografi.

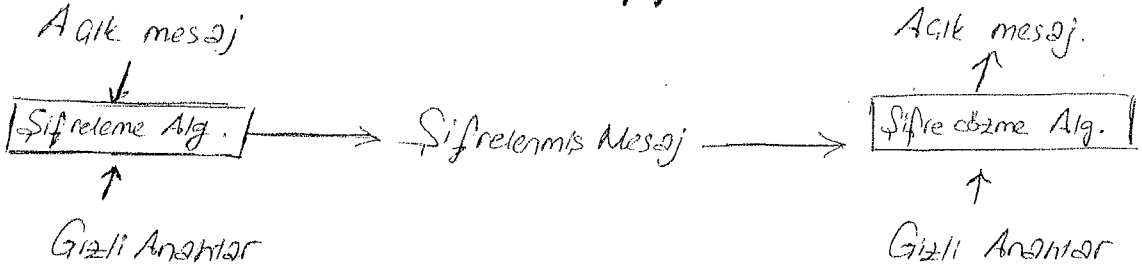
Simetrik Kriptografi :

- Simetrik kriptografide şifreleme ve şifre açma işlemi **aynı anahtar** ile yapılır.
- Simetrik kriptografide bu anahtar **gizli** tutulmalıdır. Bu nedenle, bu tip sistemlere **gizli anahtarlı kriptografi sistemi** adı da verilmektedir.

Bu sistemle haberleşen taraflar;

✓ Aynı şifreleme algoritmasını kullanırlar.
✓ Birbirine uyumlu gerçeklemeler kullanırlar.
✓ Aynı anahtarı kullanırlar.

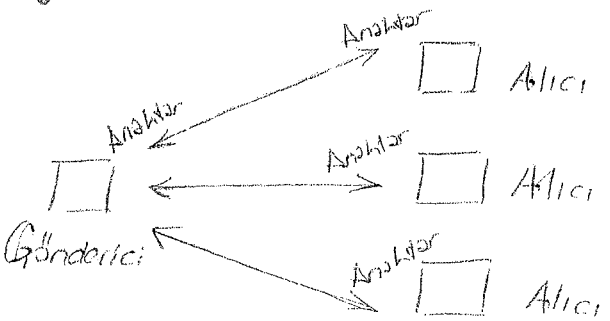
Gizli Anahtarlı Şifreleme



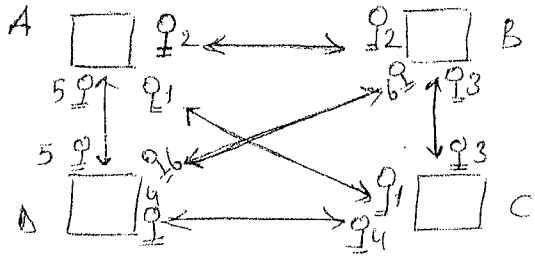
- Simetrik kriptografinin en önemli öğesi anahtar gizliliği olduğun için, birden fazla kişinin haberleştiği bir ortamda anahtar yönetimi büyük dikkat gerektirmektedir.

Simetrik Kriptografi Anahtar Yönetimi

Birden Çoğa : (One-to-Many) Bu yöntemle haberleşen tüm taraflar, aynı gizli anahtarı kullanırlar. Bu nedenle herkes birbirinin şifreli mesajlarını açıp okuyabilir.



Çoktan Çoğa : (Many-to-many) Bu yöntemle haberleşen tüm taraflar kendi aralarında bir gizli anahtar kullanmak üzere anlaşırlar. Bu nedenle herkes, şifreli haberleşeceği her kişi için bir anahtar tutar.



Bu yöntem, sistemdeki kişi sayısına bağlı olarak çok fazla anahtar üretimini gerektirdiği için çok kullanışlı değildir.

Anahtar sayısının, kullanıcı sayısına bağlı artışı aşağıda gösterilebilir:

Kullanıcı Sayısı	Anahtar Sayısı
3	3
4	6
10	45
100	4.950
n	n.(n-1)/2

Simetrik Kriptografi Artılar Eksiler

Kuvvetli Yönler

- Algoritmalar hızlıdır.
- Algoritmaların donanımla gerçeklenmesi kolaydır.
- "Gizlilik" güvenlik hizmetini yerine getirir.

Zayıf Yönleri

- Ölçeklenebilir değildir.
- Emniyetli anahtar dağıtımı zordur.
- "Bütünlük" ve "Kimlik doğrulama" güvenlik hizmetlerini gerçeklemek zordur.

Simetrik Kriptografi Algoritmaları

1. Blok Şifreleme Algoritmaları: Bu tip algoritmalar, şifrelenecek veriyi sabit uzunlukta bloklar olarak şifreleme fonksiyonuna alırlar ve aynı uzunlukta şifrelenmiş veri blokları üretirler. Bu algoritmalara örnek olarak AES, DES, IDEA, Skipjack, RC5 vb. verilebilir. Bu algoritmalar aşağıdaki özellikleri gerçekleştirmeye çalışırlar.

Karıştırma: Anahtar ve şifrelenmiş mesaj arasındaki ilişki olabildiğince karışık olmalıdır.

Dağıtma: Tek bir açık mesaj karakterinin etkisi olabildiğince fazla şifrelenmiş karaktere yansıtılmalıdır.

Transpoze İşlemi: Şifrelemeye başlamadan önce açık mesajın içeriği değişik bir sıraya konur.

Yer Değiştirme İşlemi: Tekrar edilen kalıplar, başka kalıplarla değiştirilir.

2. Bit Katarı (dizi) Şifreleme Algoritmaları; Bu tip algoritmalar, veriyi akan bir bit dizisi olarak alırlar. Vernam tipindeki bu algoritmalarda, rastgele bit dizisi üretiminin kendini tekrarlamayan bir yapıda olması gerekir. RC2, RC4 Algoritmaları örnek gösterilebilir.

AYRINTILI BİLGİ

Blok Şifreleme Algoritmaları: Blok şifreleme algoritmaları veriyi bloklar halinde işler. Bu işleme yöntemi bazen blokları birbirinden ayrı olarak bazen de birbirine bağlı olarak kullanır. Bu nedenle blok şifrelemede değişik kullanımlar ortaya çıkmıştır. Bunlardan iki tanesi ECB ve CBC'dir.

ECB; (Electronik Codebook = Elektronik kod kitabı)

- Her açık mesaj bloğu ayrı ayrı şifrelenir.
- Biçimi belli olan veri için güvenli değildir.
- Şifrelenmiş mesaj blokları birbirinden bağımsızdır.

CBC; (Cipher Block Chaining = Şifre bloğu zincirleme)

- Bir şifreleme adımının çıktısı, diğer şifreleme adımının girdisini etkiler.
- Kendi kendini işlemci saatine uygun olarak senkronize eder.
- Şifrelenmiş bloklardan biri hatalıysa en fazla iki bloğun şifresiz hali hatalı olur.

En fazla kullanılan blok şifreleme algoritmaları;

DES (Data Encryption Standard) Algoritması; Bankacılık ve finans sektöründe ağırlıklı olarak kullanılan bu algoritma IBM firması tarafından 1974'te bulunmuş 1977'de ABD standardı olmuştur. Üzerinde en çok çalışılmış olan algoritmadır. Günümüzde bu algoritma 3DES şeklinde, üç farklı anahtarla aynı bloğa üç defa DES uygulanarak da kullanılmaktadır.

AES (Advanced Encryption Standard) Algoritması; ABD'de NIST tarafından yapılan bir yarışma sonucunda yeni ABD standardı olmuştur. Algoritmanın orijinal adı Rijndael'dir. Bu algoritma seçime göre 128 bit, 192 bit ve 256 bit uzunluğunda anahtarlar kullanmaktadır. Algoritmanın blok boyu 128 bit olarak standartlaştırılmıştır.

Bit Katarı (dizi) Şifreleme Algoritmaları: Bu tip algoritmalar, veriyi akan bir bit dizisi olarak algılar. Başlıca iki grup altında toplanırlar;

1-) Senkron Algoritmalar;

- Anahtarı oluşturan bit katarı açık mesajdan bağımsız olarak üretilir.
- Hata izole olarak kalır, katarı etkilemez.
- Açık mesaj ile anahtar arasında mükemmel senkronizasyon gerektirir.

2-) Kendi Kendine Senkronize Algoritmalar;

- Anahtarı oluşturan bit katarı önceden üretilmiş şifreli mesaj bloklarıyla ilişkilidir.
- Hata izole olarak kalır.
- Kendi kendine senkronize olabilir.

Bit katarı şifreleme algoritması genellikle hız gerektiren uygulamalarda kullanılırlar. Bunlar arasında SSL protokolü tarafından da kullanılan RC4 algoritması yer almaktadır.

RC4 ALGORİTMASI

- RSA tarafından bulunmuştur. Meçhul kişiler tarafından kaynak kodu internette yayınlanmıştır.
- Değişken anahtar uzunluğuna sahiptir.
- Güvenliği, rastgele bir permütasyon kullanımına bağlıdır.
- Tekrarlama periyodu 10100'den daha büyüktür.
- Bilinen kötü anahtarı yoktur.
- Şifreleme hızı yaklaşık olarak Megabyte/sn seviyesindedir.

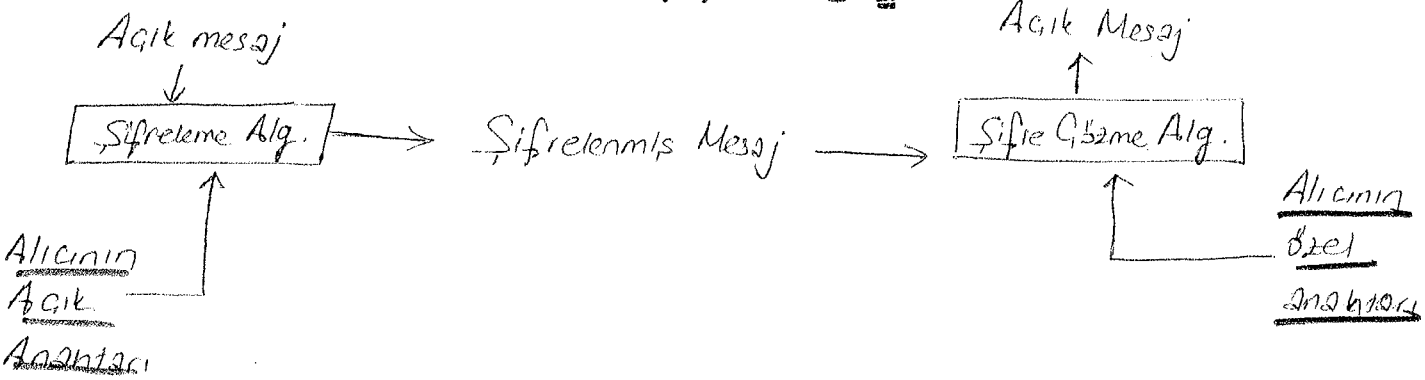
Asimetrik Kriptografi

Asimetrik kriptografide şifreleme ve şifre çözme işlemi farklı anahtarlar ile yapılır. Bu anahtar çiftini oluşturan anahtarlara **açık** ve **özel anahtar** adı verilir. Özel anahtar gizli tutulmalıdır fakat açık anahtar gerekli kişilere verilebilir ve başka kişilerle paylaşılabilir. Bu özelliğinden dolayı asimetrik kriptografi, **açık anahtarlı şifreleme** adıyla da anılır.

Bu sistemi kullanarak haberleşen taraflar,

- Aynı şifreleme algoritmasını kullanırlar.
- Birbirleriyle uyumlu gerçeklemeler kullanırlar.
- Gerekli anahtarlara erişebilirler.

= Açık Anahtarlı Şifreleme =



Asimetrik Kriptografi'de Anahtar Yönetimi

- Anahtar yönetimi için dikkat edilmesi gereken noktalar;
  - ✓ Açık anahtarlar kontrollü olarak bir otorite tarafından yayınlanmalı ve değiştirilmeleri önlenmelidir.
  - ✓ Anahtar çiftleri merkezi bir otorite tarafından üretilebilir veya her kullanıcı kendi anahtar çiftini üretebilir.
  - ✓ Şifreleme ve imza için ayrı ayrı anahtar çiftleri olmalıdır. Çok özel durumlar için imzalama ve şifreleme anahtar çiftlerinin aynı olmasına izin verilebilir.
  - ✓ Anahtar iptalleri kontrollü bir şekilde yapılmalı ve duyurulmalıdır.

- Asimetrik kriptografi için anahtar yönetimi, simetrik kriptografiye göre daha kolaydır. Çünkü bir kullanıcıyla şifreli haberleşmek isteyen bir kişi, karşı tarafın açık anahtarına ihtiyaç duyar. Bu açık anahtar kamuya açık olarak yayınlandığı için sisteme giren bir kişi için sadece bir anahtar çifti üretmek yeterli olmaktadır.

Kullanıcı Sayısı	Anahtar Çifti Sayısı
3	3
10	10
n	n

Asimetrik Kriptografi Artıları Eksileri

Kuvvetli Yönler

- Anahtar yönetimi ölçeklenebilir.
- Kriptoanalize karşı dirençlidir. (Kırılması zor)
- Bütünlük, kimlik doğrulama ve inkar edememezlik güvenlik hizmetleri sağlanabilir.

Zayıf Yönler

- Algoritmalar genel olarak yavaş çalışırlar. Simetrik kriptografi algoritmalarına göre yaklaşık 1500 kat daha yavaştırlar.
- Anahtar uzunluğu bazı durumlar için kullanışlı değildir. Mobil cihazlar için klasik algoritma anahtar uzunlukları sorunlu olabilir.

Asimetrik Kriptografi Algoritmaları

Başlıca asimetrik kriptografi algoritmaları RSA, ECC, El-Gamal ve Diffie-Hellman anahtar belirleme olarak sıralanabilir.

RSA ALGORITMASI

En yaygın olarak kullanılan asimetrik algoritmadır. Özellikleri;

Açık anahtar kriptografik sistemi ve sayısal imzalama yöntemi olarak kullanılır.

Çarpanlarına ayırma problemi üzerine inşa edilmiştir.

Bileşik tamsayı olan n'i oluşturan, asal sayılar p ve q bulunur. Öyleki $n = p.q$ olur.

Yeterince büyük bir n için kırılması çok zordur.

Ayrıca kök bulma problemine de dayanır.

Çok güvenlidir; fakat fazla hızlı değildir.

Algoritmanın kullandığı parametreler
Algoritmanın kullanımı
Algoritmanın incelenmesi ile ilgili ayrıntılı bilgi için;
kamusm.gov.tr/dosyalar/kitaplar/...

Kripto Sistemlerin Karşılaştırılması

Konu	Simetrik Kriptografi	Asimetrik Kriptografi
Gizlilik	✓	✓
Bütünlük	—	✓
Kimlik doğrulama	—	✓
İnkar Edememezlik	—	✓
Performans	HIZLI	YAVAŞ
Güvenlik	Anahtar uzunluğuna bağlı	Anahtar uzunluğuna bağlı

ELEKTRONİK İMZA

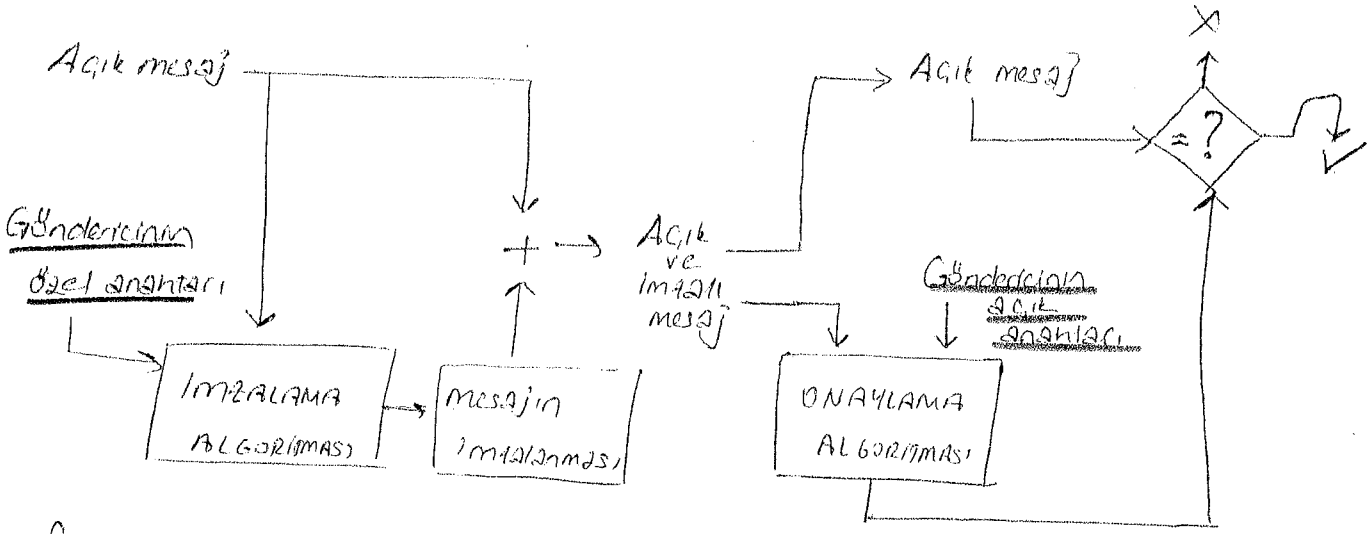
- Elektronik imza kriptografik bir algoritma olarak tanımlanabilir. Elektronik imza, mesajın içeriği ile mesajı imzalayan kişinin asimetrik özel anahtarının beraber kullanılması ile elde edilir. Sayısal (elektronik) imza aşağıdaki özelliklere sahiptir.

✓ Mesajın sonuna eklenir.

✓ Mesaj alıcısının, mesajın göndericisinin kimliğini doğrulamasını ve mesajın bütünlüğünü kontrolünü sağlar.

✓ İnkar edememezlik hizmetini sağlar.

✓ Asimetrik kriptografi kullanır.



- Sayısal imzanın bu şekilde kullanılması bir problemi beraberinde getirir. Bu kullanım şeklinde sayısal imza mesaj uzunluğunu iki katına çıkarır. Bu sorunu çözmek için **özetleme fonksiyonu** kullanılarak bir "Mesaj Özeti" çıkarılır.

Mesaj Özeti

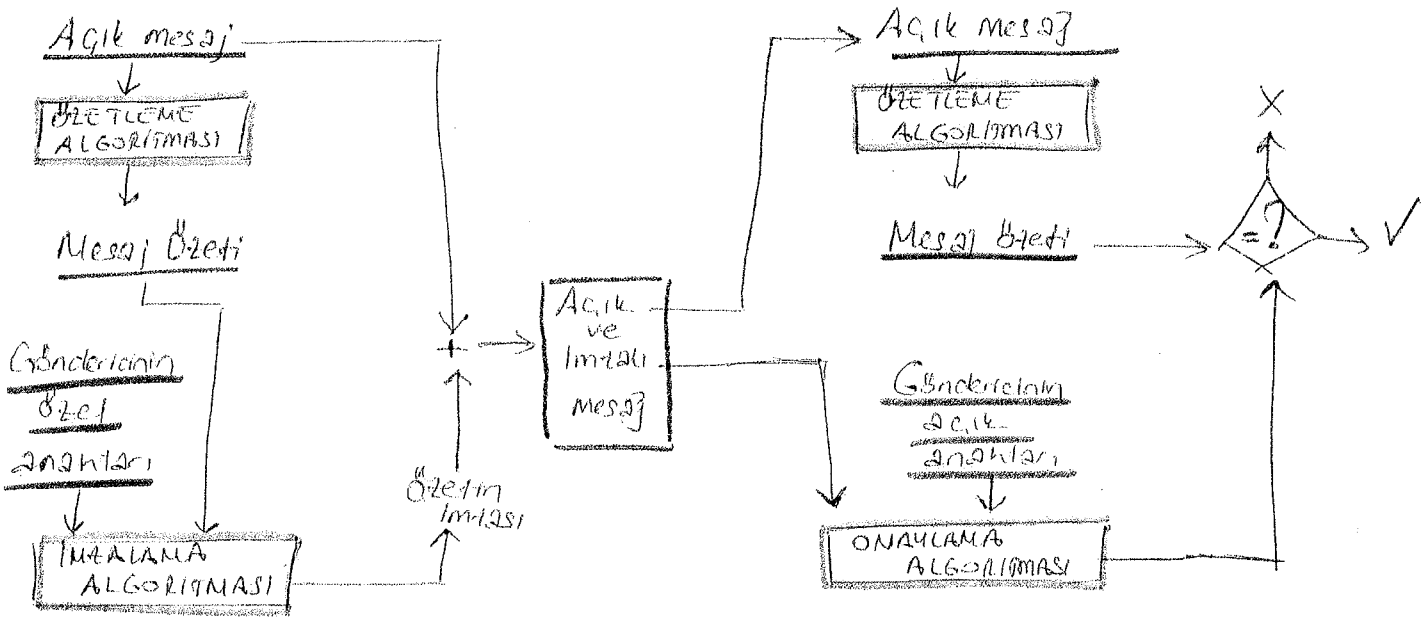
→ Herhangi bir uzunluktaki veriyi alıp işleyen ve bu veriye özgü olan, sabit uzunlukta bir değer çıkaran algoritmalara **mesaj özeti algoritması** denir. Bu algoritmaların çıktısı olan değer, mesaj özetidir. En çok bilinen özet algoritmaları **MD5** ve **SHA ve Aileleri**'dir. Mesaj özeti elde etmek için kullanılan fonksiyonların özellikleri şunlardır:

✓ Özet fonksiyonları, sabit çıkış uzunluğu üretirler. (mesajdan çok kısa). Mesaj hangi uzunlukta olursa olsun MD5 fonksiyonu 128 bit uzunluğunda, SHA-1 fonksiyonu 160 bit uzunluğunda özet değeri üretir.

✓ Mesajdaki küçük değişiklikler bile özette büyük değişikliklere yol açabilir.

✓ Özet fonksiyonları kriptografik tek yönlü fonksiyonlardır. Bir mesajın özetini elde etmek çok kolaydır, bir özetten asıl mesajı çıkarmak ise çok zordur.

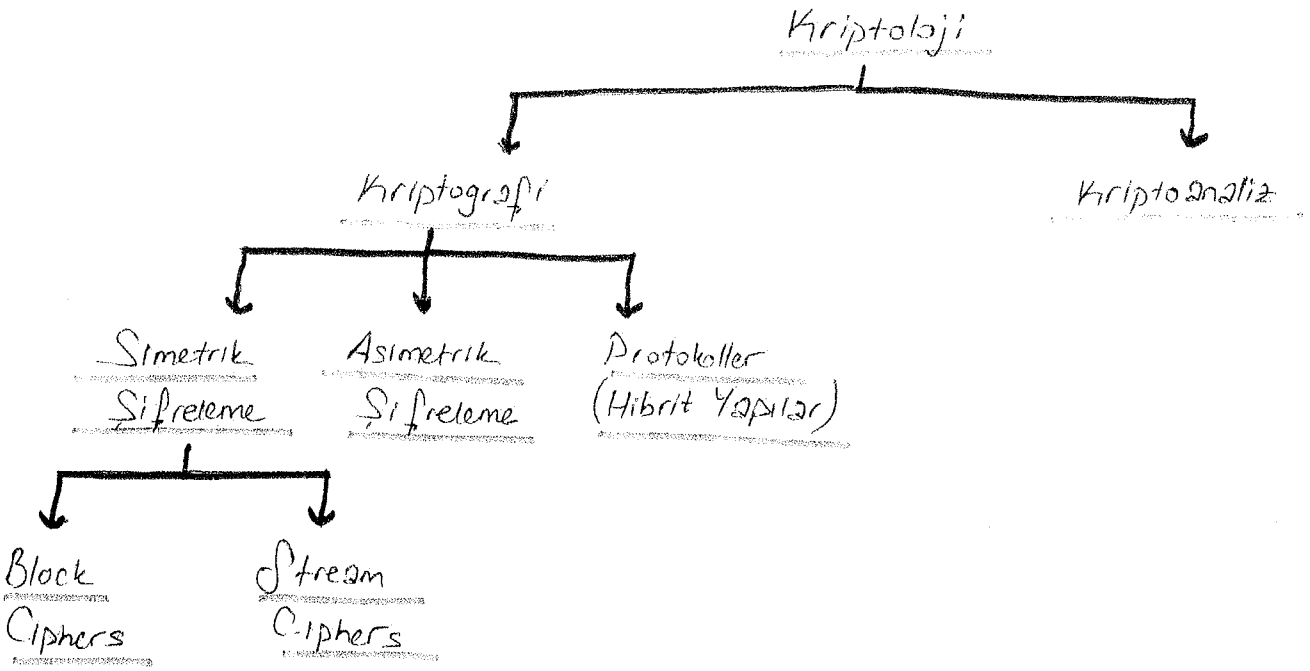
→ Mesaj özeti kullanılarak sayısal imzalama şu şekilde yapılır:


Açık mesaj
ÖZETLEME ALGORİTMASI
Mesaj Özeti
Gönderenin Özel Anahtarı
İMZALAMA ALGORİTMASI
Özetin İmzası
Açık ve İmzalı mesaj
Açık Mesaj
ÖZETLEME ALGORİTMASI
Mesaj Özeti
=?
X
✓
Gönderenin Açık Anahtarı
DOĞRULAMA ALGORİTMASI

BİLGİ GÜVENLİĞİ UNITE-1

- ŞİFRELEMEYE GİRİŞ -

Şifreleme Alanlarının Sınıflandırılması :



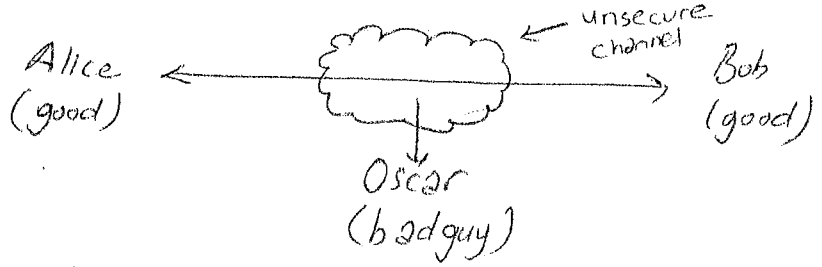
Bazı temel gerçekler :

İlk çağ şifreleme : M.Ö. 2000 Mısır'da, Sezar daha sonra popüler oldu.
Simetrik şifreleme : Tarih öncesinden 1976'ya kadar.
Asimetrik şifreleme : 1976'da Diffie Hellman ve Merkle tarafından
Karışık Yapılar : Simetrik + Asimetrik = Günümüz protokolleri.

(şifreleme ve mesaj doğrulama) + (mesaj değişimi ve sayısal imza)

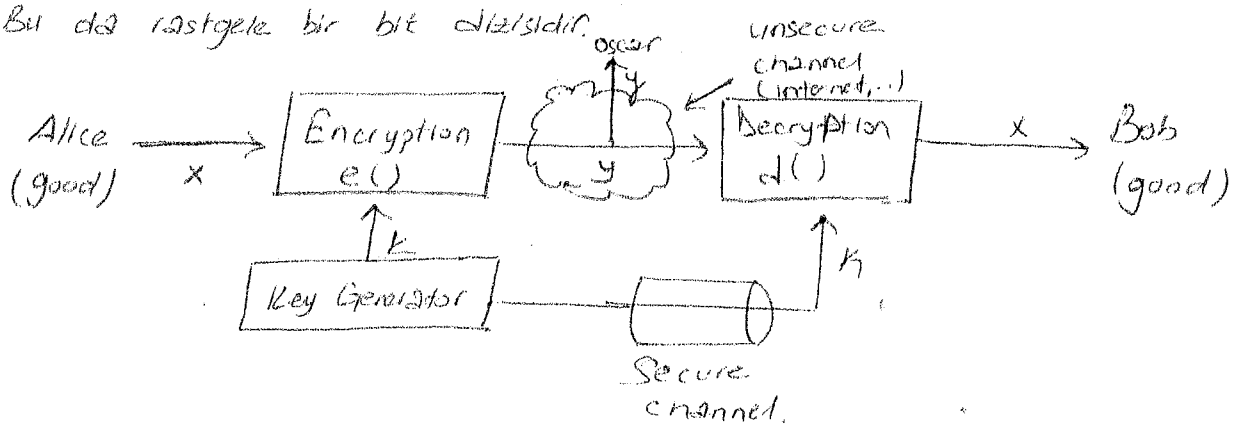
Simetrik Şifrelemenin Temelleri :

Özel anahtarlı, tek anahtarlı veya gizli anahtarlı şifreleme de denir.



Alice ve Bob güvenli olmayan kanal üzerinden haberleşmek istiyorlar. (Internet'ten) Oscar, üçüncü kişi (kötü), bu kanala ulaşabilir fakat haberleşmeyi anlayamamalı Bu nasıl sağlanır?

Çözüm : Simetrik şifre ile şifreleme tir. Oscar sadece şifreli metni elde edebilir. Bu da rastgele bir bit dizisidir.



x : düz metin plaintext
y : şifreli metin ciphertext
K : anahtar key ($K_1, K_2, \ldots K_n$ → Anahtar uzayı)

Simetrik Şifreleme :

Şifreleme denklemi : $y = e_k(x)$
Şifre çözme denklemi : $x = d_k(y)$

İki tarafta da aynı anahtar değeri kullanılıyorsa şifreleme ve şifre çözme işlemleri birbirinin terstir.

$$d_k(y) = d_k(e_k(x)) = x$$

burada önemli olan :

- Anahtar değeri Bob ve Alice arasında, güvenli bir kanalda iletilmesidir. Güvenli kanal kurye veya benzeri yöntemlerle oluşturulabilir.
- Bununla birlikte saldırgan K anahtar değerini bilmediği sürece güvenlidir.
- Güvenli haberleşme problemini **güvenli iletim** ve **K anahtar değerinin saklanması** azaltılmaktadır.

Kriptoanaliz

Neden kriptoanalize ihtiyaç duyarız?

Herhangi pratik bir şifre için güvenliğin matematiksel bir kanıtı yoktur. Sistemin güvenli olduğunu garanti etmenin tek yolu sistemi kırmaya çalışmaktır.

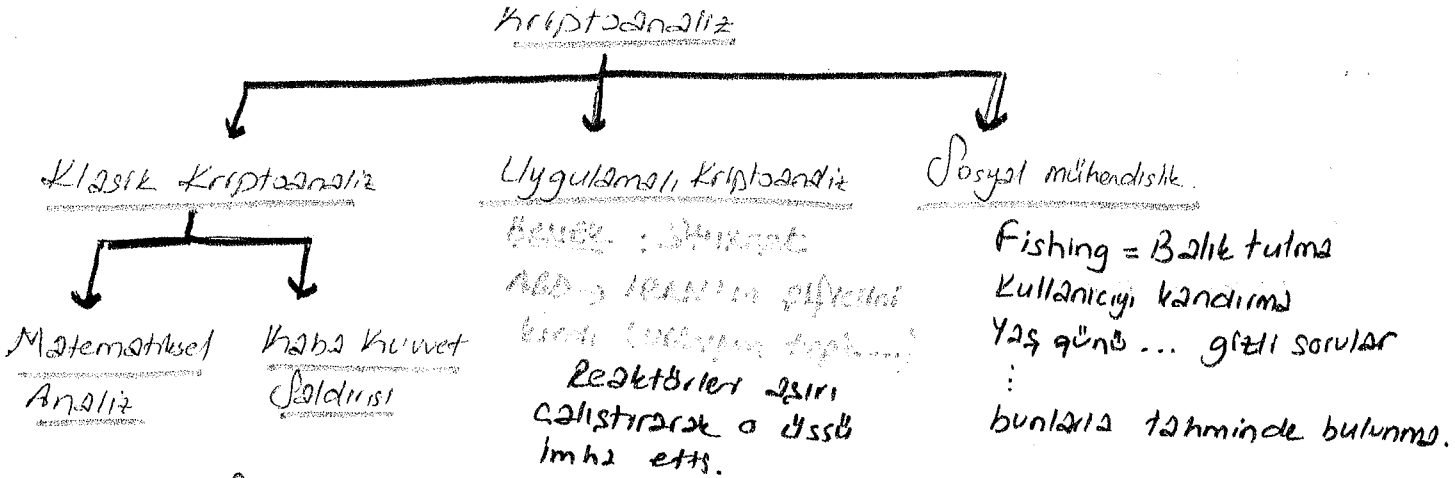
Kerckhoff prensibi modern şifrelemede çok iyidir.

> Saldırgan gizli anahtar dışında her şeyi biliyor olsa bile şifreleme sistemi güvenli olmalıdır.

Kırışof prensiplerini pratikte uygulayabilmek için: Sadece iyi şifreleyiciler tarafından birkaç yıl kriptoanaliz edilmiş geniş bir ölçekte kullanılan şifreler kullanılmalıdır.

Şifrelemenin daha güvenli olması için şifreleme yapısının detaylarını gizlemek daha cazip gelebilir, fakat bu yapılar mühendisler tarafından incelendiğinde her zaman kırılmıştır. (Örnek: DVD içeriğini korumak için kullanılan program Content Scrambling System (CSS))

Şifre Analizi: Şifreleme Sistemlerine Saldırı:



Klasik Saldırılar:

- Matematiksel analiz
- Kaba kuvvet (Brute-force) saldırısı

Uygulama Saldırıları:

Ters mühendislikle veya güç ölçümüyle anahtar değerini çıkarmaya çalışırız, örneğin bankalar için kullanılan banking smart card.

Sosyal Mühendislik:

Örneğin, kullanıcının şifresini vermesi için kandırmak.

Simetrik şifrelemeye karşı kaba kuvvet saldırısı

- Şifreye bir kutuları gibi davranır.
- En az bir şifreli metin-düz metin çifti gerekir. (X_0, Y_0)
- Şart gerçekleşinceye kadar bütün mümkün anahtarları kontrol eder.

$$d_k(y_0) \stackrel{?}{=} x_0$$

Kaç farklı anahtara ihtiyaç duyarız?

Bitsel olarak anahtar uzunluğu	Anahtar uzayı	Security life time
64	2^{64}	Short-term (birkaç gün veya dakikaya)
128	2^{128}	Long-term (several decades)
256	2^{256}	Long-term

⚠ Karşı taraf başarmak için sadece bir saldırıya ihtiyaç duyar. Böylece sosyal mühendislik gibi diğer saldırılar yapılabiliyorsa uzun anahtar uzayı pek de faydalı olmayacaktır.

YER DEĞİŞTİRMELİ ŞİFRELEME

- Tarihsel bir şifrelemedir.
- Kaba kuvvet, ... gibi analitik saldırılar, anlamak için bir araçtır.
- II. Dünya Savaşı'na kadar bu vardır (bitlerden ziyade harfler kullanılmıştır)
- **Çalışma mantığı**: Düz metindeki her bir harfe şifreli metinde sabit bir harfe denk gelsin.

Plain text	Cipher text
A	K
B	...
C	J

Öz: ABBA, kddk gibi şifrelensin.

Yer değiştirme şifrelemeye karşı saldırılar

1. Sonsuz anahtar arama (kaba kuvvet saldırısı)
2. Harf frekanslarının analizi

1. Sonsuz Anahtar Arama (Kaba Kuvvet Saldırısı)

- Mantıklı bir düz metin elde edilene kadar bütün mümkün yer değiştirmelerin yapılması.
- $26 \times 25 \times \dots \times 2 \times 1 = 26! \approx 2^{88}$, 2^{88} anahtar arasından değerleri araştırmak günümüz bilgisayarları için mümkün değildir.

2. Harf Frekanslarının Analizi

- İngilizcede harflerin farklı kullanım sıklığı vardır. Düz metindeki bu sıklık, şifreli metinde de korunur. Örneğin, ingilizcede en çok "e" kullanılır. e yerine kullanılan bir değer, şifreli metinde de çok yer alacaktır.
- En çok kullanılan harf %13 ile e, sonra %9 ile t dir.
- Çözüm adımları;
  - ✔ Şifreli metinde en çok geçen harfi bul.
  - ✔ Yerine e yaz.
  - ✔ Kalan harflerin sıklığına bak ve sırasıyla t, a, o, ... yaz.
- Sadece harflerin değil kelime ikilileri ve üçlüleri de şifreli metin çözmek için kullanılır.

Sonuç Olarak;

Yer değiştirmeli şifre 2^{88} gibi yeterli büyüklüklükte bir anahtar uzayına sahip olmasına rağmen, sayısal yöntemlerle kolaylıkla kırılabilir. Bu, şifreleme yapılarının bütün saldırılara karşı dayanıklı olması gerektiğini gösteren güzel bir örnektir.

MODÜLER ARİTMETİK

Neden Modüler Aritmetik?

Asimetrik şifreleme için çok önemlidir (RSA, ECC, ...)

Bazı tarihsel şifreler, modüler aritmetikle ifade edilir (Sezar, Affine)

Modüler Aritmetik

Birçok şifreleme sistemi sayı kümelerine dayanır. Şöyle ki;

1. Ayrı (sayılı kümeler olduğu için faydalıdır)
2. Sonlu (sadece sonlu sayılarla hesaplanır)

Saat örneği; Saatin sürekli artması ama 12 yi geçmemesi. Çarpma ve toplama işlemlerinden sonra elde edilen sayıların yine küme içerisinde kalması.

$a \equiv r \bmod m$

Ör: $12 \equiv ? \bmod 9 \rightarrow ? = 3$

Ör: $34 \equiv ? \bmod 9 \rightarrow ? = 7$

Ör: $-7 \equiv ? \bmod 9 \rightarrow ? = 2$.

İlginç olan mod m'de kalan değere karşılık sonsuz değer denk gelmesidir. Örnek,
- $12 \equiv 3 \mod 9$
- $12 \equiv 21 \mod 9$
- $12 \equiv -6 \mod 9$

} Ancak biz en küçük pozitif tamsayıyı seçeriz.

Modüler bölme işlemi: $a/b \equiv r \mod m \Rightarrow a.b^{-1} \equiv r \mod m$ yani sayının tersini alarak çarpma işlemi yapılır.

Ör: $5/7 \mod 9 = ?$

$5/7 \equiv \mod 9 \rightarrow 5.7^{-1} \equiv ? \mod 9$ → 7 nin mod 9'daki tersini bulmalıyız.

$7.7^{-1} = 1 \mod 9$

$? = 4$

$28 \equiv 1 \mod 9$

O halde

$5.4 = 20 \equiv ? \mod 9$

$? = 2$ //

Ters Hesaplama;

- $\gcd(a, m) = 1$ oluyorsa a nın mod m'de tersi vardır.

Modüler aritmetik'e cebirsel bakış.

Halka($\mathbb{Z}_m$)

- Kapalılık; işlemlerin sonucu halka içinde olmalı. (ikili işlem = binary)
- Toplamada değişme; $a+b = b+a$ Sonuç $\in \mathbb{Z}_m$, (ikinci işlemde varsa Abelyen Halka)
- Dağılma ve Birleşme
- Ters; Toplamaya göre: $-a$ Çarpmaya göre: a^{-1}
- Etkisiz eleman; Toplamada: 0, Çarpmada: 1

*$\mathbb{Z}_m$'de elemanların toplamaya göre tersi vardır. Çarpmaya göre olmayabilir.

* Halkalarda toplama, çıkarma, çarpma yapabiliriz; bölme, tersi olan elemanlar için yapabiliriz

Ör: $\mathbb{Z}_9 = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$

$1^{-1} \equiv 1 \mod 9$ $\quad 4^{-1} \equiv 7 \mod 9$ $\quad 7^{-1} \equiv 4 \mod 9$

$2^{-1} \equiv 5 \mod 9$ $\quad 5^{-1} \equiv 2 \mod 9$ $\quad 8^{-1} \equiv 8 \mod 9$

0, 3 ve 6 9 ile aralarında asal old. tersleri yoktur.

NOT: monary-unary işlem $f(x) = \frac{1}{x}$ $\quad$ binary işlem $+, -, *$, $\mathbb{Z}/\mathbb{Z} \rightarrow$ binary değil, $\mathbb{R}/\mathbb{R} \rightarrow$ binary

$\mathbb{R} \rightarrow \mathbb{R}$? $\Rightarrow$ monary

$\mathbb{Z} \rightarrow \mathbb{Z}$? $\Rightarrow$ monary değil.

Kaymalı Şifreleme (Sezar) ve Affine Şifreleme :

Kaymalı (veya Sezar) Şifreleme :

- Düz metindeki her harfin yerine bir harf yazılır.
- Değişim kuralı : belirlenen bir k anahtar değeriyle her harften sonraki harf alınır.

"de

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

k=7 için ; Düz metin : ATTACK

Şifreli metin : HAAHJR = 7, 0, 0, 7, 9, 17

- Burada 26 yı aşan değerler için mod26 alınarak başa dönülür.
- Şifrelemenin matematiksel tanımı;

Encryption : $y = e_k(x) = x + k \pmod{26}$	$k, x, y \in \{0, 1, \ldots, 25\}$
Decryption $x = d_k(y) = y - k \pmod{26}$	

*** Sonsuz anahtar tarama ve harf sıklığı analizi saldırılarında kırılması mümkün

Affine Şifreleme :

- Düz metne sadece anahtar değerini eklemeyip, ek olarak başka bir anahtar değeri ile de topluyoruz.
- Burada iki parçadan oluşan bir anahtar değeri kullanılır $[(a, b) = k]$
- Şifrelemenin matematiksel tanımı;

Encryption : $y = e_k(x) = a \cdot x + b \pmod{26}$	$k, x, y \in \{0, 1, \ldots, 25\}$
Decryption : $x = d_k(y) = a^{-1}(y - b) \pmod{26}$	

- Şifreleme işleminde tersini almaya ihtiyaç duyduğumuz için, a ifadesine göre sadece ters olan sayılar kullanılır.

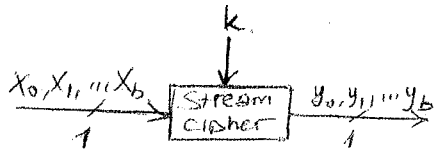
$\gcd(a, 26) = 1$ bu şartı sağlayan 12 değer vardır.

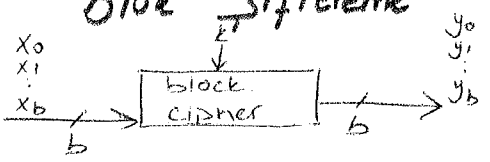
- Bundan dolayı anahtar uzayı $12 \times 26 = 312$ dir.

*** Detaylı anahtar uzayı tarama ve harf sıklığı analizi saldırılarında kırılması mümkün

STREAM ŞİFRELEME

Stream Şifreleme ve Blok Şifreleme



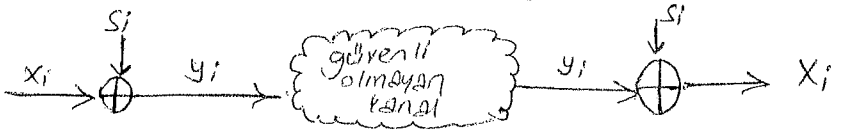


- Bitler tek tek şifrelenir.
- Gömülü sistemlerde yaygındır.
- Genellikle küçük ve hızlıdır.

- Her zaman blokun hepsini şifreler (birkaç bit)
- İnternet uygulamaları için yaygındır.

Stream şifreleme ile şifreleme ve şifre çözme

→ Düz metin x_i, şifreli metin y_i ve stream anahtarı s_i tek tek bitlerden oluşur.



→ Şifreleme ve şifre çözme mod 2 de toplama işlemi gibidir (XOR)

→ Şifreleme ve şifre çözme aynı fonksiyondur.

→ **şifreleme** = $y_i = e_{s_i}(x_i) = x_i + s_i \pmod 2$

→ **deşifreleme** = $x_i = e_{s_i}(y_i) = y_i + s_i \pmod 2$ $\quad x_i, y_i, s_i \in \{0,1\}$

Senkron ve Asenkron Stream Şifreleme

s_i anahtar üretiminin güvenliği tamamiyle anahtar değerine bağlıdır. Rastgele olmalı, gönderici ve alıcı tarafından yeniden üretilmelidir. ($Pr(s_i=0) = 0.5 = Pr(s_i=1) = 0.5$)

Senkron stream şifrelerde anahtar üretimi sadece anahtara bağlıdır. (ve IV başlangıç değeri olabilir) **Asenkron stream şifrelerde** anahtar üretimi şifreli metne de bağlıdır.

Neden mod 2 'de toplamak iyi bir şifreleme fonksiyonudur? Çünkü mod 2 toplama işlemi XOR işlemine eşittir. Mükemmel bir stream s_i anahtarı için her şifreli çıkış bitinin 0 veya 1 olma ihtimali % 50 olmalıdır. ve ters çevrilmiş XOR basittir, çünkü XOR işleminin aynısıdır.

x_i	s_i	y_i
0	0	0
0	1	1
1	0	1
1	1	0

Düz p: 10111 $\quad$ c: 11001
k: 01110 $\quad$ k: 01110
c: 11001 $\quad$ p: 10111

Simetrik şifrelerin performans karşılaştırması;

CIPHER	Key length	Mbit/s
DES	56	36.95
3DES	112	13.32
AES	128	51.19
RC4 (Stream cipher)	(choosable)	211.34

RASTGELE SAYI ÜRETECİ (Random Number Generators (RNGs))

RNG

- True RNG
- Pseudorandom NG
- Cryptographically Secure PRNG

→ Gerçek Rastgele Sayı Üreteçleri (TRNG)

- Fiziksel rastgele işlemlere dayanır: yarı iletken gürültüsü, radyoaktif kalıntılar, fare hareketleri.
- Fiziksel bilgisayarları seçemeyiz. Örneğin; Elazığ'ın sıcaklığına göre şifrelenen metin, ABD'de çözülemez.
- Gerçek Rastgele Sayı Üreteçlerinin Özellikleri;
  - ✓ Çıkış stream s_i değeri iyi istatiksel özelliklere sahip olmalı: $Pr(s_i=0) = Pr(s_i=1) = \%50$
  - ✓ Çıkış değeri hem tahmin edilememeli hem de yeniden üretilememelidir.
- Kullanımı: Anahtar üretimi (sadece tek kullanımlık değerler)

→ Yalancı Rastgele Sayı Üreteçleri (PRNG)

- Başlangıç gerçekçi değerinden bir zincir üretir. Yani tohum değerinden başlayarak adım adım, iteratif olarak, üretilir.
- Çıkış stream, iyi istatiksel özelliklere sahiptir.
- Özyinelemeli yollarla çıkış değeri yeniden üretilebilir ve tahmin edilebilir:

$$S_0 = \text{seed}$$
$$S_{i+1} = f(S_i, S_{i-1}, \ldots, S_{i-t})$$

- **kuvvetli yanı;** karşı taraf aynı anahtarı üretebilir.
- **zayıf yanı;** denkleme bağlı olduğundan kolaylıkla çözülebilir.

ÖRNEK: C'deki rand() function:

$$S_0 = 12345$$
$$S_{i+1} = 1103515245 \cdot S_i + 12345 \mod 2^{31}$$

→ CSPRNG : Yalancı Rastgele Sayı Üreteci Şifresel Güvenliği;

* Ek özellikleri ile PRNG olduğu söylenebilir.
* Çıkış kesinlikle tahmin edilememelidir. Daha doğrusu S_i değerinin çıkışından elde edilen n bit, bir sonraki S_{i+1} çıkışında tahmin edilmesi polinomsal zamanda mümkün değildir.
* Stream şifrelemede ihtiyaç duyulur.

TEK KULLANIMLIK BLOK NOTLAR (OTP) = One Time Pad :

* Güvenli sistem; sonsuz hesaplama kaynağı ile çözülemeyen sistemdir.
* OTP sistemi, Vernam stream şifre yapısına dayanır. Vernam yapısı;

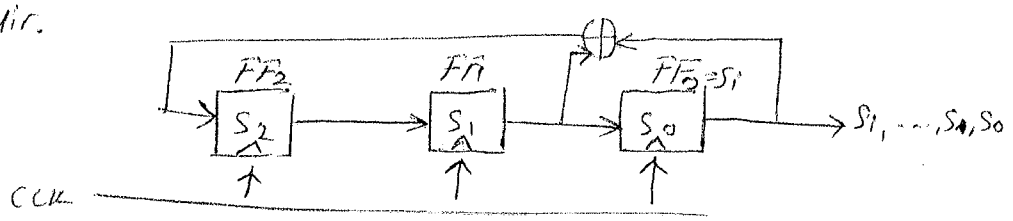
$$e(x) = x \oplus k$$
$$d(y) = y \oplus k \qquad x, y, k \in \{0,1\}$$

* Eğer k anahtar değeri sadece bir defa kullanılırsa OTP alışılmadık ölçüde güvenlidir.
* OTP her seferinde anahtarın farklı üretilmesini gerektirir. Bu nedenle çok güvenli yapılardır. **dezavantajı** ise; anahtarın mesajla aynı boyutta olması gerekliliğidir. (Mesaj = 1 GB ise Anahtar = 1 GB olmalı) Çok büyük boyutlu anahtar üretiminde sıkıntılıdır.

LINEAR FEEDBACK SHIFT REGISTER (LFSR)

* Geri besleme durum bitlerini XOR'layarak yeni giriş değerini üretmeye çalışır.
* m derecesi depolanan elementlerin sayısını verir.
* Eğer $P_i = 1$ ise geri besleme bağlantısı vardır. (kapalı anahtar).
  $P_i = 0$ ise geri besleme yoktur (açık anahtar)
* Çıkış zinciri, periyodik olarak tekrarlar. Maximum çıkış uzunluğu: $2^m - 1$

NOT | FF'lar ile bağlı olan bu sistemlerde, FF'lar farklı davranabilir. Bu nedenle aynı FF'lar farklı değerler üretebilir. Deterministik olduğundan gerçek üreteçlerdir.



* LFSR çıkışı rekürsif denklemlerle tanımlanır.
  $S_{i+3} = S_{i+1} + S_i \bmod 2$

Burada max çıkış uzunluğu $2^3 - 1 = 7$ dir.
FF_0 çıkışı anahtarlardır. Tablosunu çizelim;

NOTLAR

1-) IV Başlangıç Vektörü (Initialization Vector)
Senkronda IV olabilir (Olmak zorunda değil)

2-) Neden böyle bir anahtar üretme yöntemine ihtiyaç duyulmuş?
Çünkü anahtar uzunluğu bilinmiyor.

3-) 3DES, 3 tane DES'in arka arkaya bağlanması.

RC4	Anahtar	Şifreleme	
Stream →	Uzunluğu →	hızlı	Önemli!
Cipher	değişken	2li	

4-) Stream özellikleri
→ Hızlı
→ CPU'yu daha az yorar.
→ Anahtar üretme mekanizması sıkıntılı.

Basit Bir PRNG Kriptoanalizi

ÖRNEK PRNG: Linear Congruential Generator ⟶ Sistem saatine göre üretim.
Kendini tekrar etmez 2^{31} farklı değer üretir.

S_0 = seed
$S_{i+1} = A \cdot S_i + B \bmod m$

Farz edelim ki;
A, B ve S_0 anahtar gibi bilinmiyor.
A, B ve S_i boyutu 100 bit
Çıkış 300 bit (Çıkış = S_1, S_2 ve S_3 gibi)

Çözüm

$S_2 = A \cdot S_1 + B \bmod m$
$S_3 = A \cdot S_2 + B \bmod m$
...

} A ve B buradan direkt olarak bulunur ve bütün S_i'ler kolaylıkla hesaplanır.

ÖRNEK $f(x) = 3x + 5 \bmod 9$, $X_0 = 4$. İki tarafın bilmesi gereken değerler 3, 5 ve 9 dur. (A, B ve m). Bu değerler sabit olduğundan bu sistemin gizli anahtarı olan $X_0 = 4$ olduğu bulunabilir.

* Birçok PRNG lineer yapıda olduğundan şifreleme özellikleri kötüdür.
↳ Şifrelemede RNG kullanılır.
↳ Stream

clk	FF_2	FF_1	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

Security of LFSR

- (Aşağıdaki polinomsal hesaplar)
- LFSR polinomlarla şöyle gösterilir:

$$P(X) = X^m + P_{k-1}X^{m-1} + \dots + P_1 X + P_0$$

- Tek LFSR, tahmin edilebilir bir sonuç üretir.
- Eğer $2m$ çıkış biti varsa, LFSR nin derecesi m olmalıdır, geri besleme sabitleri P_i ler lineer sistemlerdeki çözümlerle bulunabilir.
- Bir case stream şifreleme LFSR nin birleşimi şeklindedir.

TRIVIUM: A MODERN STREAM ŞİFRELEME

- Shannon'ın Konfüzyon ve Difüzyon kuralını uygular.
- Üç nonlineer LFSR (NLFSR) nin uzunlukları 93, 84, 111 (Seçilebilen sayılar)
- XOR toplama : Üç NLFSR çıkışları stream anahtarı üretir.
- Donanımda küçüklük :
  - Toplam kaydedici sayısı : 288
  - Lineer olmayan 3 AND kapısı
  - 7 XOR kapısı.

- Donanımda gecikmeler,
- Donanımın ideal olmaması,
- Çevre şartları (olumsuz)

} için kullanılıyor ve sayı üretiyor.

VERİ ŞİFRELEME STANDARDI (DES)

- DES, Block Cipher'da yer alır.
- 64 bit uzunluktaki veri bloklarını şifreler.
- Lucifer şifreleme yapısını temel alır.
- IBM tarafından geliştirilmiştir.
- NIST olarak ifade edilen NBS tarafından 1997'de standartlaştırıldı.
- Son 30 yılın en popüler blok şifreleme yapısıdır.
- Bugüne kadar en çok çalışılan simetrik algoritmadır.
- Günümüzde 56 bitlik anahtar uzunluğu güvenli değildir. Fakat **3DES**, güvenli bir şifreleme yapısı oluşturmuştur. Halen kullanılmaktadır.
- 2000 yılında yerini **AES** almıştır.

Block Şifreleme Temelleri : Confusion and Diffusion

Shannon'a göre, güçlü şifreleme algoritmaları inşa edebilmek için iki temel işlem vardır:

1. Confusion (karıştırma): Bir şifreleme işleminde şifreli metin ile anahtar arasında bir ilişki olmamalıdır. Ek olarak;

* İstatistiksel yöntemlere karşı dirençli olmalıdır.

Konfüzyon işlemini sağlayan en yaygın yapı AES ve DES'te yer alan **substitution (yer değiştirme)** işlemidir.

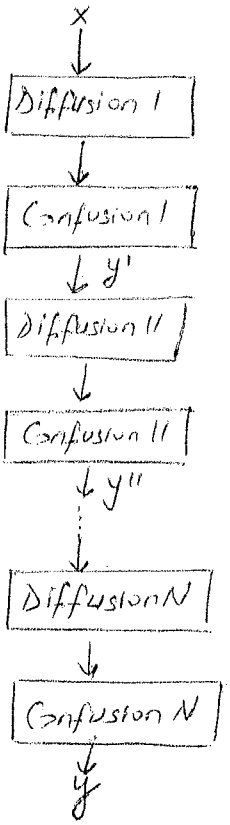
2. Diffusion (yayma): Sabit yaklaşımları engellemek amacıyla düz metindeki bir sembolün, şifreli metindeki birçok sembolü etkilemesi işlemidir. Amaç; şifreli metin ile şifresiz metin arasındaki ilişkiyi engellemektir.

Basit bir difüzyon elemanı DES içinde sıklıkla kullanılan **bit permutasyonu**.

NOT

İki işlem de kendi başlarına güvenliği sağlayamazlar. Buradaki amaç, bu iki işlemi ard arda kullanarak güvenli şifreler oluşturmaktır. Yani bu iki koşul gerekli ama yeterli değildir.

Ürün Şifreler :



Bugün kullanılan block şifrelerin çoğu bu yapıları tekrar tekrar giriş verisine uygulayan round denilen yapılardan oluşur. (yani her tura round denir)

Her round'da farklı anahtar kullanılır.

Sadece ilk anahtar bilinir, diğerleri ondan oluşturulur.

Eğer düz metindeki bir biti değiştirdiğinizde, ortalama olarak çıkış bitlerinin yarısı değişiyorsa mükemmel difüzyona ulaşabilirsiniz.

ÖRNEK :

$x_1 = 0010\ 1011$
$x_2 = 0000\ 1011$

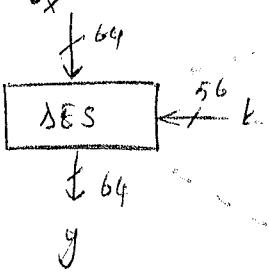
→ Block Cipher →

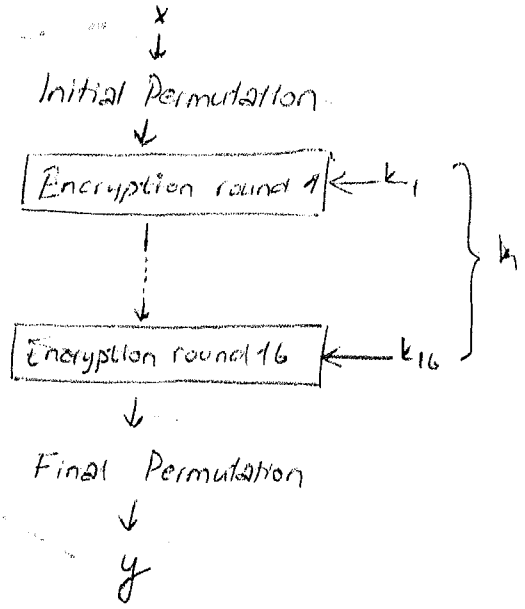
$y_1 = 1011\ 1001$
$y_2 = 0110\ 1100$

tek bit değişti ⋯⋯→ birçok bit değişti

ve buna Çığ Etkisi denir.

DES Algoritmasına Genel Bakış :





1. 64 bit uzunluktaki blokları şifreler.
2. 56 bitlik anahtar değeri kullanır.
3. Simetrik şifreleme dir ; şifreleme ve deşifreleme için **aynı anahtar** kullanılır.
4. Aynı işlemleri yapan **16 round** vardır.
5. Her round'da kullanılan **farklı anahtarlar** başlangıç anahtarından türetilir.

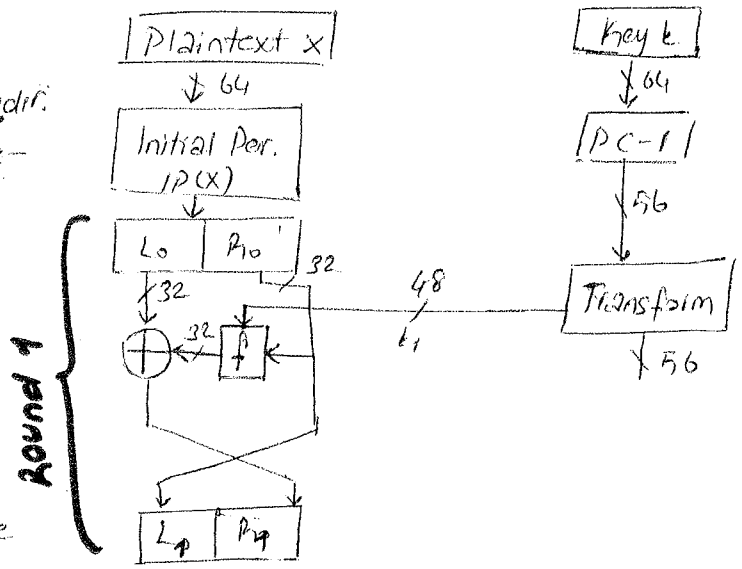
DES FIESTEL YAPISI (1)

→ DES'in yapısı Fiestel yapısıdır.

→ Avantajı: şifreleme ve deşifre-lemede sadece anahtar tarifesi farklıdır.

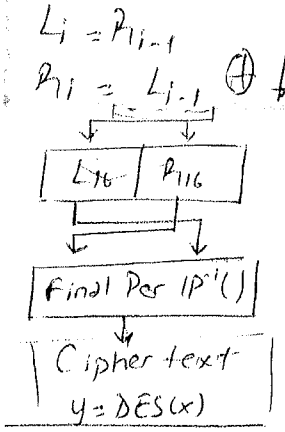
→ Bit düzeyinde ilk permutasyon, ardından 16 round gerçekleştirilir.

1. Girdi metnin 32 bitlik L_i ve R_i alt parçalarına ayrılır.
2. R_i f fonksiyonuna sokulur ve bunun çıkışı L_i ile XOR'lanır.
3. Sağ ve sol yarımlar yer değiştirir.



→ Roundlar şöyle ifade edilebilir: →

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



→ 16. round sonunda L ve R tekrar yer değiştirir. Böylece geri dönüşüm sağlanır.

→ Initial Permutation IP(X) ve Final Permutation $IP^{-1}(X)$, birbirinin tersi alınabilir iki permütasyondur.

DES'in İç Yapısı:

→ **Başlangıç ve Son permütasyon**

Bit düzeyindedirler.

Birbirinin tersi şeklindedir.

IP ve IP^{-1} tablolarında gösterilmiştir.

Başlangıç Permütasyon

IP							
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	...		25	17	...	
59	51	...					
61	53	...					
63	55	...					

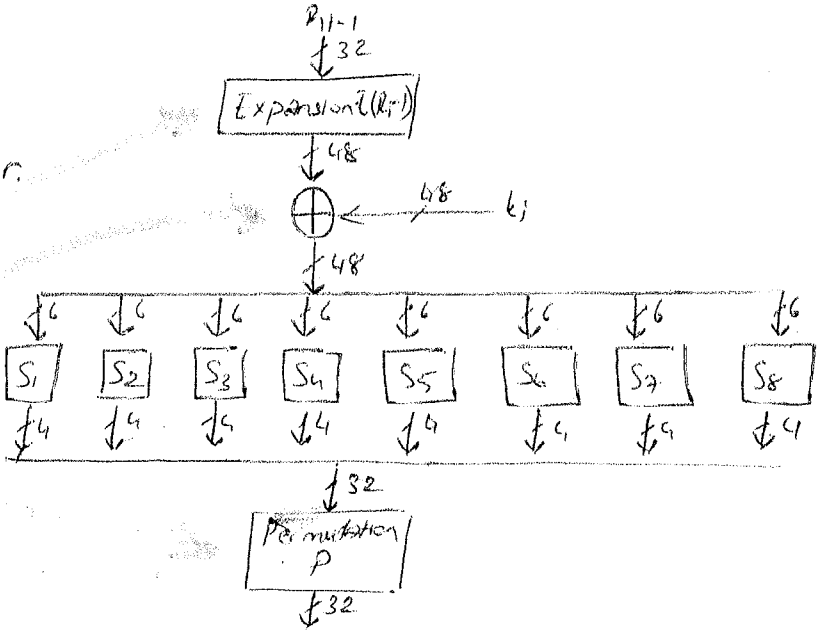
Son Permütasyon

IP^{-1}							
40	8	48	16	56	24	64	32
39	7	...		55	...	...	...
38	6	...		...	...	...	...
37	5	...		...	...	...	...
36	4	...		...	...	...	...
35	3	...		...	...	...	...
34	2	...		...	...	...	...
33	1	41	9	49	17	57	25

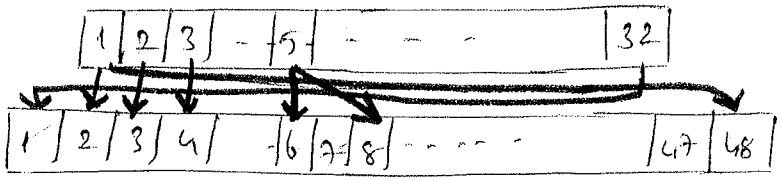
→ f fonksiyonu

f Fonksiyonu girişleri; R_{i-1} ve round anahtarı k_i'dir.

1. Genişletilmiş E
2. Round anahtarı ile XOR
3. S box substitution
4. Permütasyon



1 → Genişletilmiş E fonksiyonu: Amaç difüzyonu artırmak.

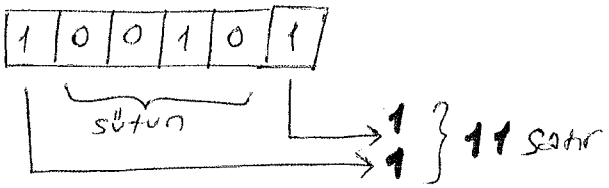


2 → Round Anahtarı Ekleme:

✓ Round anahtarı ile genişletilmiş E'nin çıkışı bit düzeyinde XOR'lanır. Round anahtarı DES anahtar tarifesi ile başlangıç anahtarından üretilir.

3 → DES S-Boxları:

✓ 8 tane substitution tablosu. 6 bit giriş 4 bit çıkışa sahiplerdir.
✓ Linear değillerdir.
✓ Diferansiyel kriptoanalize karşı dayanıklıdır.
✓ DES'in güvenliği için önemli yapılardır.



S_1	0	1	2	3	4	5	...	14	15
0	14	04	13	01					
1	00	15	07	04					
2	04	01	14	08					
3	15	12	08	02					

4 → Permütasyon

✓ Bit düzeyinde permütasyon işlemi.
✓ Amaç difüzyonu sağlamak.
✓ S-box bitlerinin çıkışı, bir sonraki roundda birçok S-box'ı etkiler.
✓ E difüzyonu, S-boxlar ve P 5 roundden sonra her bit düz metnin ve anahtarın fonksiyonu olduğunu garanti eder.

Key Schedule (1)

- 48 bitlik her k_i alt anahtarı, orijinal 56 bitlik anahtardan oluşur. DES'te anahtar boyutu 64 tür. Burada 56 bit **anahtar** ve 8 bit **parity** bitidir.
- İlk permütasyon seçiminde parity bitler kaldırılır. (PC-1). Bu kullanılmayan bitler; 8, 16, 24, 32, 40, 48, 56 tür. (Her 8 'inin 8. biti Parity)
- Anahtar C_0 ve D_0 denilen iki parçaya ayrılır. (28 bitlik)
- $k_i = 1, 2, 9, 16$ numaralı roundlarda **1 bit sola** kaydırılır.
- Diğer bütün roundlarda her iki parça **2 bit sola** kaydırılır.
- Her roundaki anahtar değeri o roundaki her iki anahtar yarısını kullanarak elde edilir. Her k_i alt anahtarı kendim bir permütasyonudur.

NOT Döndürmelerin toplam sayısı $(4 \times 1) + (12 \times 2) = 28$ $\left[D_0 = D_{16} \text{ ve } C_0 = C_{16}\right]$

DEŞİFRELEME

Feistel şifrelemede, deşifreleme için sadece anahtar tarifesi değiştirilir. Aynı 16 anahtar sırası ile ters çevrilerek üretilir.

ters anahtar üretme:

$D_0 = D_{16}$ ve $C_0 = C_{16}$ yapılarak ilk anahtar üretilebilir.
Her roundda anahtar üretim işlemlerinin tersi yapılır.

1. roundda döndürmek yok.
2., 9., 16. roundda 1 bit sağa kay.
Diğer roundlarda iki bit sağa kay.

DES'İN GÜVENLİĞİ

→ Anahtar uzayı çok küçük (2^{56} anahtar) } **iki eleştiri**
S-box tasarlama kriterleri gizli tutuldu. }

→ Sayısal saldırı: DES lineer ve diferansiyel kriptoanalize çok duyarlıydı. Bu, IBM ve NASA'nın bu saldırılardan haberdar olması anlamına gelir. Şimdiye kadar DES'i kırabilen bilinen analitik bir saldırı yoktur.

→ Ayrıntılı anahtar arama: Şifreli-şifresiz her (x,y) çifti için deneme. 2^{56} anahtar vardır. $DES_k(x) = y$ sağlayana kadar denemek. Bugünün bilgisayar teknolojisi ile kolay.

3lü DES - 3DES

Anahtar uzunluğunu 112 ye çıkarmak için DES algoritmasının üç. defa kullanılmasına dayanır.

$$y = DES_{k_3}(DES_{k_2}(DES_{k_1}(x)))$$

Avantajı ; $k_1 = k_2 = k_3$ seçilirse performansı tek DES gibidir.
Banka sistemlerinde kullanılır.

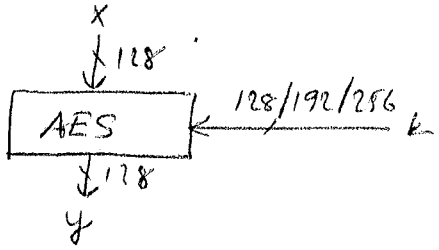
DES'e Alternatifler;

Algoritma	I/O Bit	Anahtar uzunluğu
AES (Rijndael	128	128/192/256
Triple DES	64	112 (effective)
Mars	128	128/192/256
RC6	128	"
Serpent	128	"
Twofish	128	"
IDEA	64	128

THE ADVANCED ENCRYPTION STANDARD (AES)

- Günümüzde en yaygın kullanılan simetrik şifrelemedir.
- Bütün AES adaylarının için gerekli özellikler;
  - 128 bit block uzunluğunda bloc şifreleme
  - 128, 192 ve 256 bit anahtar uzunluklarıyla desteklenmelidir.
  - Yayınlanan diğer algoritmalara göre daha güvenli olmalıdır.
  - Yazılım ve donanımda etkinlik
- Son 5 algoritma Ağustos 1999 da ilan edildi.
  Mars, RC6, Rijndael, Serpent, Twofish.
- Ekim 2000 'de AES olarak bilinen Rijndael Algoritması seçildi.

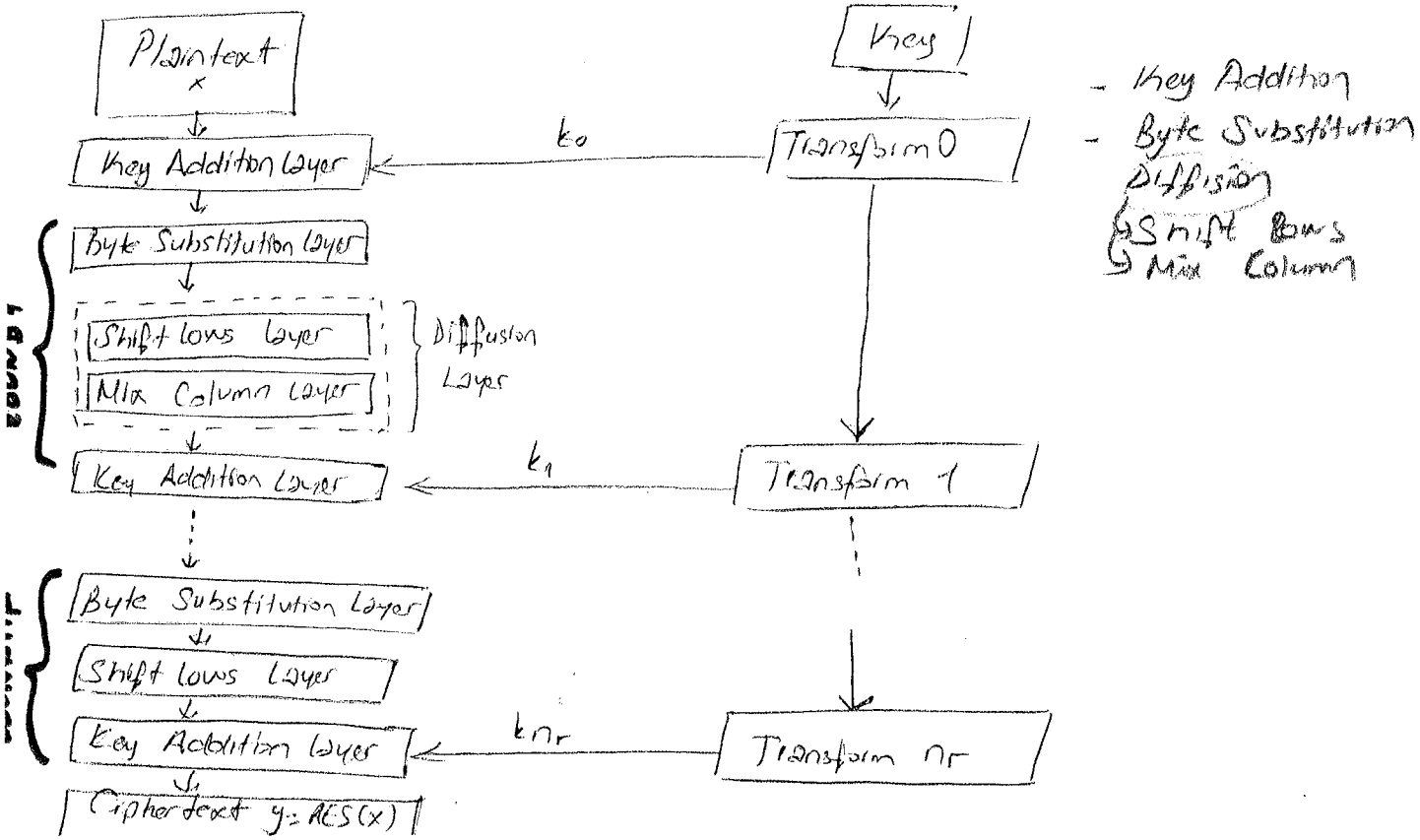
AES'e Genel Bakış:



- Round sayısı seçilen anahtar uzunluğuna bağlıdır.

Anahtar uzunluğu	Round sayısı
128	10
192	12
256	14

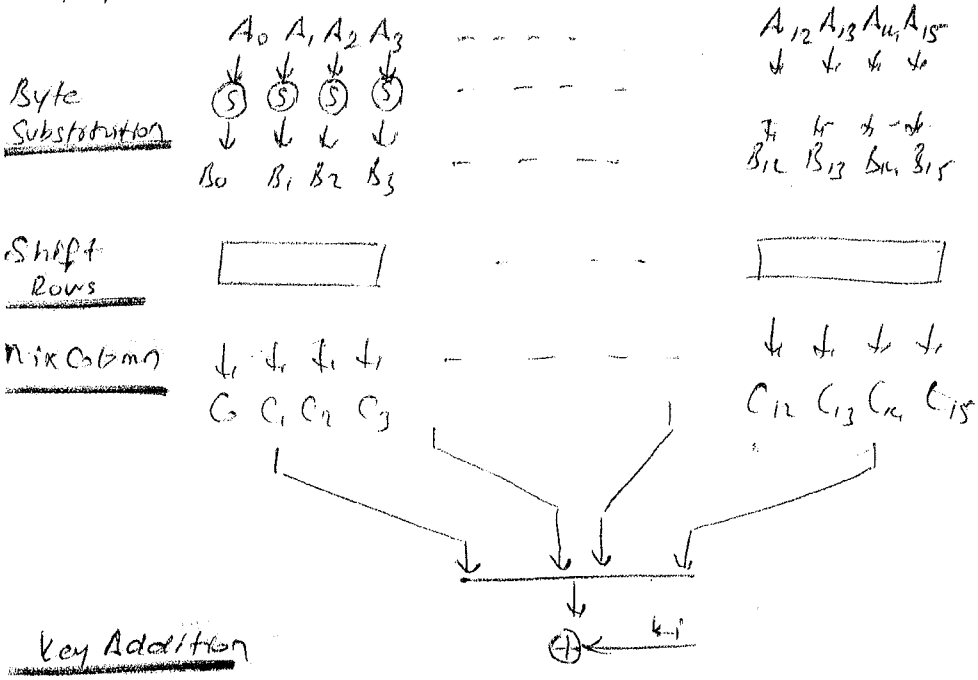
- 10/12/14 roundlu tekrarlı şifreleme.
  Her round "Layers" (Katmanlar) dan oluşur.



AES byte temelli bir şifrelemedir. 4x4 matris olarak düzenlenebilir.
$A_0, \ldots, A_{15}$ ile 16 byte'lık AES giriş verisini oluşturur. ⇒

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

1, 2, ..., nE1 round için round fonksiyonu



Son roundda MixColumn dönüşümü çıkarılmıştır.

→ Byte Değiştirme Katmanı;

16 tane S-box'tan oluşur. Bu S-box'ların özellikleri;

S-box'lar aynıdır.

AES'teki nonlinear tek elemanlardır. Yani $ByteSub(A_i)+ByteSub(A_j) \neq ByteSub(A_i+A_j)$

Giriş ve çıkış byte'ları birebir haritalanır.

S-boxlar birebir ters çevrilebilir.

→ Difüzyon Katmanı;

Bütün giriş bitleri üzerinde difüzyonu sağlar. İki alt katmandan oluşur;

Satır kaydırma Alt katmanı byte seviyesinde veri değişimi

Sütun karıştırma " " 4 byte'lık blokları içeren matris işlemleri

Matrisler üzerinde yapılan işlemler lineerdir. Yani; $DIFF(A)+DIFF(B)=DIFF(A+B)$

Satır Kaydırma Alt Katmanı;

B'lerden oluşan matrisin satırları, dairesel şekilde sıralanır.

Giriş Matrisi → Çıkış matrisi

$B_0\ B_4\ B_8\ B_{12}$ → kayma yok → $B_0 B_4 B_8 B_{12}$

$B_1\ B_5\ B_9\ B_{13}$ → sola 1 kayma → $B_5 B_9 B_{13} B_1$

$B_2\ B_6\ B_{10}\ B_{14}$ → sola 2 kayma → $B_{10} B_{14} B_2 B_6$

$B_3\ B_7\ B_{11}\ B_{15}$ → sola 3 kayma → $B_{15} B_3 B_7 B_{11}$

Sütun Karıştırma Alt Katmanı;

B lerden oluşan matristeki her bir satırı karıştıran lineer dönüşüm.

Sütundaki her 4 byte 4x4 lük bir matrisin çarpımıdır ve vektör olarak ele alınabilir. Örneğin

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 03 & 01 & 02 & 03 \\ 01 & 01 & 01 & 02 \end{pmatrix} \times \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}$$

Burada 01, 02, 03 hex gösterimde verilmiştir.

Bütün aritmetik işlemler Galois Fieldı $GF(2^8)$ 'de yapılır.

→ Anahtar Ekleme Alt Katmanı;

- Girişler; 16 byte C matrisi ve 16 byte k_i alt anahtarı.
- Çıkış; $C \oplus k_i$
- Alt anahtarlar, anahtar tarifesinde üretilir.

Anahtar Tarifesi:

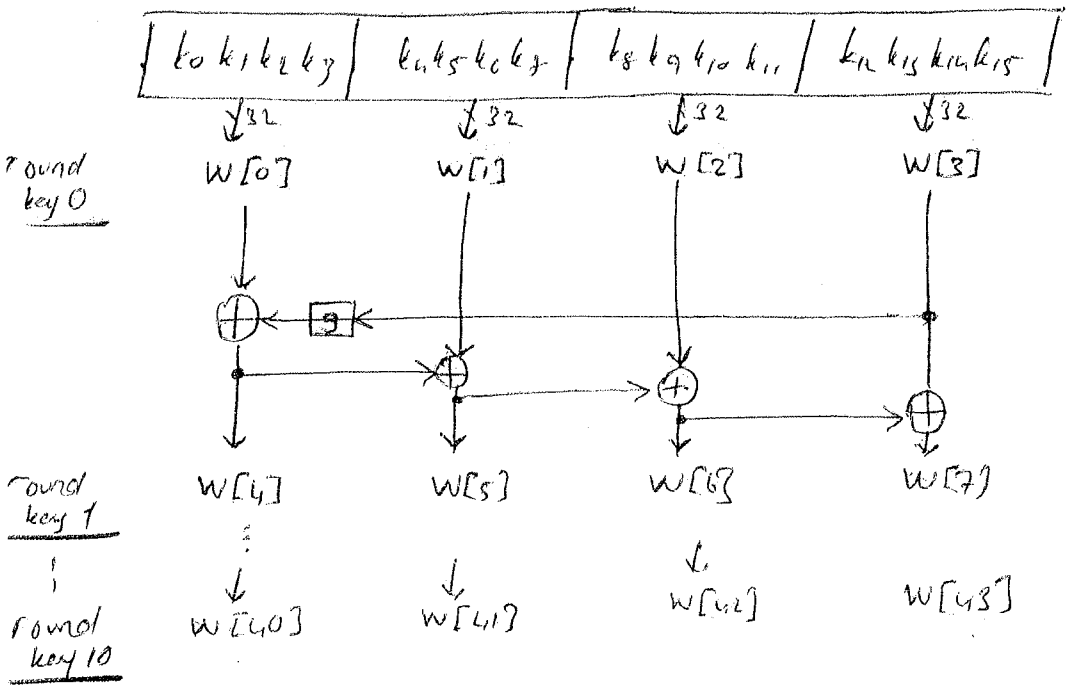
- Alt anahtarlar 128/192 veya 256 bitlik giriş anahtarlarından yinelemeli olarak üretilir.
- Her roundun bir alt anahtarı vardır. Ek olarak bir de AES'in başlangıcındaki anahtar.

Anahtar uzunluğu	Alt anahtar sayısı
128	11
192	13
256	15

- Anahtar Genişletme; Alt anahtar AES'in hem giriş hem de çıkışında kullanılır.
- Farklı anahtar boyutu için farklı anahtar tarifeleri vardır.

$$\text{Subkeys} = \text{rounds} + 1$$

ÖZNEK 128 bit anahtarlı AES için Anahtar Tarifesi = Key schedule



- Kelime tabanlıdır ; 1 kelime 32 bit.
- 11 adet anahtar şeklinde depolanır. W[0]...W[3], W[4]...W[7], ...
- İlk adet anahtar W[0]... W[3] orijinal AES anahtarıdır.

g fonksiyonu; Kendi'ye girişi byte'ını döngüsel kaydırır ve byte düzeyinde S-box yer değiştirme işlemi yapar.

Round Sabitli RC'ler sadece en sola eklenir ve her round için değişir.

$$RC[1] = x^0 = (00000001)_2$$
$$RC[2] = x^1 = (00000010)_2$$
$$\dots$$
$$RC[10] = x^9 = (00110110)_2$$

x^i Galois Field'deki elemanları gösterir.

DEŞİFRELEME

- AES Feistel yapısına dayanmaz.
- Deşifreleme için bütün katmanlar ters çevrilir.
- Sütun karıştırma katmanı → Sütun kar. katmanın tersi
- Satır karıştırma katmanı → Satır " " "
- Byte Yerdeğiştirme katmanı → Byte yerdeğiştirme " "

- Anahtar Ekleme Alt katmanı kendisi tersidir.

→ Sütun Karıştırma Katmanının Tersi;

- C matrisinin her sütunu 4x4 matrisin tersi ile çarpılmalı.

$$\begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \times \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

- 09, 0B, 0D, 0E hex. olarak gösterilmiştir.
- Bütün çarpmalar işlemler GF(2^8) de yapılır.

→ Satır kaydırma katmanının tersi;

- B matrisinin her satırı, şu şekilde kaydırılır.

Giriş matrisi → Çıkış matrisi

1	$B_0 B_4 B_8 B_{12}$	kaymayok	$B_0 B_4 B_8 B_{12}$
	$B_1 B_5 B_9 B_{13}$	sağa 1	$B_{13} B_1 B_5 B_9$
	$B_2 B_6 B_{10} B_{14}$	sağa 2	$B_{10} B_{14} B_2 B_6$
	$B_3 B_7 B_{11} B_{15}$	sağa 3	$B_7 B_{11} B_{15} B_3$

→ Byte Yerdeğiştirme Katmanının Tersi;

- $A_i = S^{-1}(B_i) = S^{-1}(S(A,0))$
- Ters S-box deşifrelemede kullanılır. Bu gerçekte bir Arama tablosudur.

Deşifreleme Anahtar Tarifesi;

- Alt anahtarlar ters sırayla kullanılır.
- Pratikte, şif. ve deşif. için aynı anahtar tarifesi kullanılır.
- İlk blok şifreleme başlamadan önce tüm alt anahtarlar hesaplanmalı.

. AES etkili bir yazılım uygulamasıdır.
. Basit uygulamalarda iyidir.
. 8 bit işlemciler için uygun (smart card) 32 veya 64 bit CPU'ya uygun değildir.

Güvenlik ;

Kaba Kuvvet Saldırısı: 128/192/256 bitlik anahtarlardan dolayı Brute-force mümkün değildir.

Analytical Attacks: Kaba kuvvetten daha iyi olan bilinen hiçbir analitik saldırı yoktur.

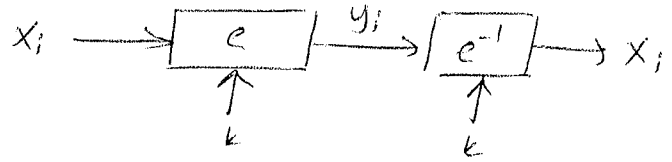
Yan Kanal Saldırıları: Algoritmanın uygulamalarına yapıldığından AES temeline yapılmaz.

MORE ABOUT BLOCK CIPHERS

- Bloğa bölme yöntemleri
  1. Electronic Code Book Mode (ECB)
  2. Cipher Block Chaining Mod (CBC)
  3. Output Feedback Mode (OFB)
  4. Cipher Feedback Mode (CFB)
  5. Counter Mode (CTR)
  6. Galois Counter Mode (GCM)
- Bu 6 modun tek bir amacı var. Doğrulama ve bütünlüğün yanısıra Gizliliği sağlamak.

ECB :

- Bildiğimiz yöntem.
- Her blok ayrı ayrı şifrelenir. Bir bloktaki hata diğerini etkilemez.



Avantajları

- Gönderici ve Alıcı arasında blok eşleşmesine gerek yoktur. (Yani senkronizasyon yok)
- Herhangi bir problemden dolayı (Gürültü gibi) bitlerde bozulma olursa, sadece ilgili blok etkilenir, diğer yapılar etkilenmez.
- Blok şifreleme paralel olarak yapılabilir.
- Yüksek hızlı uygulamalar için avantajlıdır.

Dezavantajları

- ECB yüksek seviyede deterministik bir şifreleme yapısıdır.
- Aynı düz metin aynı şifreli metni oluşturur. Aynı şifreli metin ikinci defa gönderildiğinde saldırgan bunu tanır.
- Düz metin blokları birbirinden bağımsız olarak şifrelenir. Saldırgan şifreyi çözemese bile blokların yerini değiştirebilir.

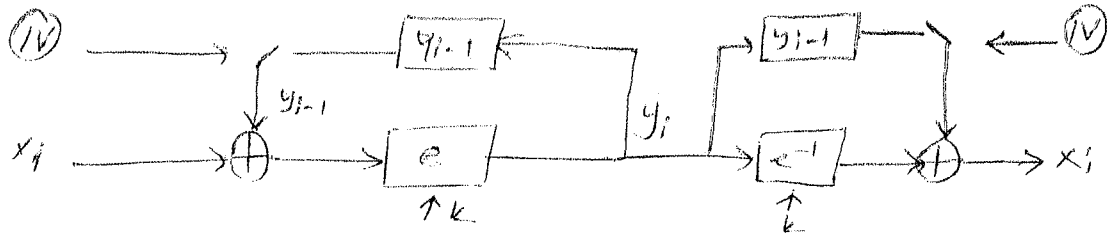
ÖRNEK : İnternet Bankacılığı

1	2	3	4	5
Sending Bank A	Sending Account	Receiving Bank B	Receiving Account	Amount

- İki Banka arasındaki anahtar çok sık değişmez
- Saldırgan sürekli A → B'ye 1 $ gönderir.
- " tekrarlayan şifreli parçaları kontrol eder.
- 1, 3 ve 4'ü saklar.
- Saldırgan 4. bloğu daha önce sakladığı 4 ile değişebilir.
- Böylece A → B transferi yanlış yönlendirilir.

CBC

Şifrelenecek bütün bloklar birbirine zincirlenir. y_i şifreli metin bloğu sadece x_i düz metin bloğuna bağlı değildir, kendisinden önceki bütün düz metin bloklarına bağlıdır. Şifreleme bir IV kullanılarak randomize edilir.



Şifreleme **ilk blok** : $y_1 = e_k(x_1 \oplus IV)$

" **genel "** : $y_i = e_k(x_i \oplus y_{i-1})$, $i \geq 2$

Deşifreleme **ilk blok** : $x_1 = e_k^{-1}(y_1) \oplus IV$

" **genel blok** : $x_i = e_k^{-1}(y_i) \oplus y_{i-1}$, $i \geq 2$

ÖNEK : İnternet Bankacılığı

IV her havalede uygun seçilirse; bu saldırı mümkün olmaz.

IV birkaç transfer için aynı tutulursa; saldırgan transferi tanıyabilir.

✔ IV her şifrelemede yeni seçilirse (fa-ek) ; olasılıksal bir CBC modu elde edilir.

OFB

Bu yapı, bir blok şifreleme yapısından, bir senkron akış şifre oluşturmak için kullanır.

Akış anahtarı bit düzeyinde üretilmez, ama bit düzeyinde kullanılır.

Şifrelemenin çıkışı anahtar akış şifreleme anahtarını (S_i) verir. Bu değer ve düz metin şifrelenerek şifreleme işlemi gerçekleştirilir.

Şifreleme ilk blok : $S_1 = e_k(IV)$ and $y_1 = S_1 \oplus x_1$

Şifreleme genel blok : $S_i = e_k(S_{i-1})$ and $y_i = S_i \oplus x_i$, $i \geq 2$

Deşifreleme ilk blok : $S_1 = e_k(IV)$ and $x_1 = S_1 \oplus y_1$

Deşifreleme genel blok : $S_i = e_k(S_{i-1})$ and $x_i = S_i \oplus y_i$, $i \geq 2$

CFB

Bu yapılar asenkron bir akış şifreleme oluşturmak için blok şifrelemeyi kullanırlar (OFB'ye benzer)

Akış anahtarı (S_i) blok işlemlerle üretilir ve şifreli metnin bir fonksiyonudur. CFB de nondeterministik bir yapıdadır.

Şifreleme ilk blok : $y_1 = e_k(IV) \oplus x_1$

" genel : $y_i = e_k(y_{i-1}) \oplus x_i \quad i \geq 2$

Deşifreleme ilk blok : $x_1 = e_k(IV) \oplus y_1$

" genel : $x_i = e_k(y_{-1}) \oplus y_i \quad i \geq 2$

Kısa düz metin bloklarını şifrelemede kullanılabilir.

CTR

OFB ve CFB modları gibi, akış şifreleme oluşturmak için kullanılır.

Blok şifrelemenin girişi, her zaman değişen bir sayaç değeridir, böylece her zaman farklı bir akış anahtarı hesaplanır.

CFB ve OFB modlarının aksine CTR modu paralelleştirilebilir. Çünkü 1. şifreleme bitmeden 2. şifreleme gerçekleştirilebilir.

Ağ yönlendiricileri gibi **yüksek hız** gerektiren uygulamalarda kullanılır.

Şifreleme : $y_i = e_k(IV \,||\, CTR_i) \oplus x_i \quad i \geq 1$

Deşifreleme : $x_i = e_k(IV \,||\, CTR_i) \oplus y_i \quad i \geq 1$

GCM

MAC hesaplamada kullanılır.

Mesaj değiştirilme ve Mesaj bütünlüğü sağlanır.

(GF çarpma.)

AYRINTILI ANAHTAR ARAMAYA GÖZ ATMA

Daha kuvvet saldırıları mümkündür; ancak birkaç tane düz metin-şifreli metin çiftine ihtiyaç vardır.

Öz : Şifreleme bloğunun genişliği 64 bit anahtar boyutu 80 bit olsun. X_1 düz metni 2^{80} farklı anahtarla şifrelenirse 2^{80} tane şifreli metin oluşturulur. Bunlar bastırılır sadece 2^{64} tanesi ayrıdır. Verilen şifreli-şifreli metin çifti için tüm anahtarlar kullanılırsa, ortalama $2^{80}/2^{64} = 2^{16}$ anahtar $e_k(x_1) = y_1$ haritalama işlemini gerçekleştirir.

BLOK ŞİFRELEME GÜVENLİĞİNİN ARTIRILMASI

İki yaklaşım mevcut Çoklu şifreleme ve Anahtar Boyasıtma.

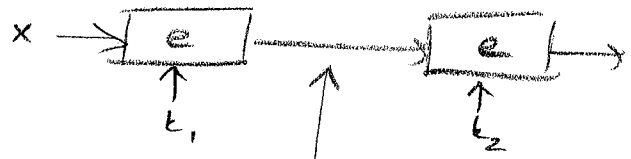
İkili Şf. Üçlü Şf.

→ Çift Şifreleme:

- x düz metni önce k_1 ile şifrelenir. Elde edilen şifreli metin de k_2 anahtarıyla tekrar şifrelenir.
- Anahtar uzunluğu k bit kabul edilirse ayrıntılı aramalar aşama $2^k \cdot 2^k = 2^{2k}$ şifreleme veya deşifreleme gerektirir.
- DES için; 2^{56} ihtimalden her biri için saldırgan $2^{56} \cdot 2^{56} = 2^{112}$ deneme yapması gerekir.

* Meet-in-the-Middle Attack

İki aşamadan oluşur:



Saldırı burada yapılır.

1. Saldırganın kaba kuvvet sonucuna göre arama tablosu hesapla. 2- Sağ tarafına göre düzenle.

- Sistemin tamamına saldırırsam 2^{2k} tarama yapmam gerekir. Ortaya saldırırsam 2^{k+1} tarama yapmam gerekir. 2^k sağdaki için, 2^k soldaki için $2 \cdot 2^k = 2^{k+1}$. Oluşturulan tabloya bakarsın. Ayrı ayrı iki şifre için değer deneyeceksin yerine bir değer deneyeceksin. Birine olmayan diğerine de olmaz. Böylece arama uzayını daraltırsın.

NOT Çift şifreleme tek bir şifrelemeden çok daha güvenli değildir.

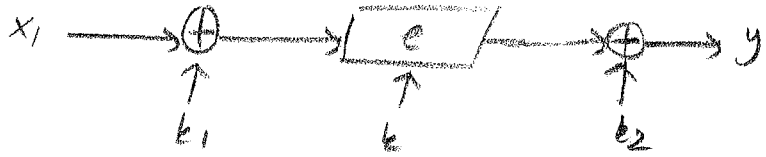
→ Üçlü Şifreleme:

- Blok 3 defa şifrelenir. $y = e_{k_3}(e_{k_2}(e_{k_1}(x)))$
- Pratikte farklı yapılar kullanılır. Şif-Des-Şif gibi $\rightarrow y = e_{k_3}(e^{-1}_{k_2}(e_{k_1}(x)))$
- $k_1 = k_2 = k_3$ seçilirse tek DES şif. yapılmıştır.
- Saldırgan 3DES için 2^{112} test yapmak zorundadır.

NOT Üçlü şifreleme etkin anahtar uzunluğunu 112 bit kabul eder.

ANAHTAR BEYAZLATMA

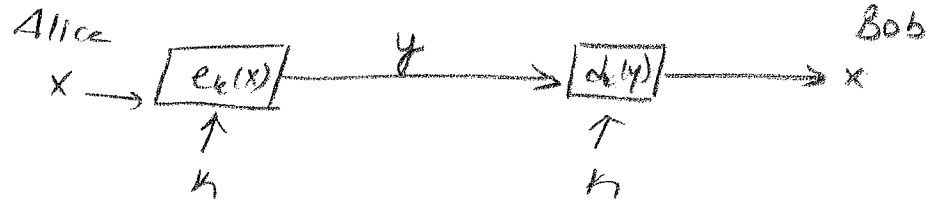
DES gibi sif. yapılarını Brute-force saldırılarına daha dayanıklı yap.
k anahtarı ile beraber iki tane beyazlatma anahtarı (k_1 ve k_2)
x_1 ve y_1 ile şifreleme için kullanılır.



Bazına uygulamaları DES gibi kısa anahtar uzayına sahip sif. yapılarını
Anahtar beyazlatmaya sahip değişik DES'ler **DESX** olarak
adlandırılır.

GENEL ANAHTARLA ŞİFRELEMEYE GİRİŞ

Simetrik Kriptografi'yi Gözden Geçirme;



- Simetrik yani gizli anahtar kripto sistemlerin en önemli iki özelliği;
  - Aynı gizli anahtar k şif. ve deşif. için kullanılır.
  - Şif. ve deşif. fonksiyonları çok benzerdir. (hatta aynıdır)
- Güçlü bir anahtarla güvenli bir sistemdir, sadece Alice ve Bob anahtarın bir kopyasına sahiptir
  - Alice kendi anahtarı ile mesajı güvenli bir şekilde şifreler. (kilitler)
  - Bob kendi anahtarı ile mesajı " " " deşifreler. (açar)

Problem:

- ✓ Anahtarları üretmek
- ✓ Anahtarı güvenli bir şekilde karşıya iletmek
- ✓ Anahtarları saklamak.

- n kişi için anahtar sayısı $\frac{n(n-1)}{2}$ dir.

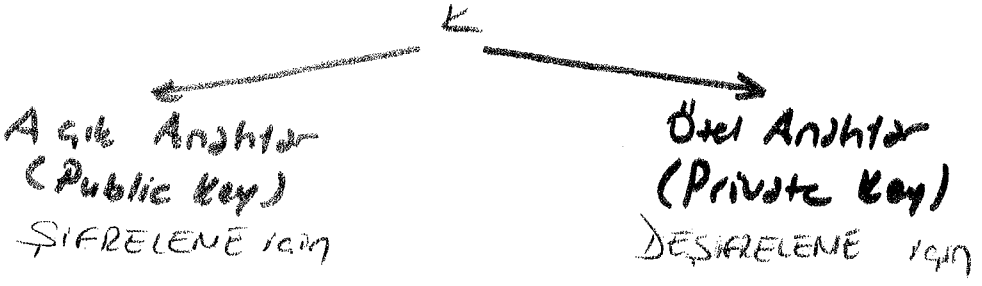
Simetrik Kriptografi: Eksiklikleri:

- AES, 3DES gibi simetrik algoritmalar oldukça güvenli hızlı ve yaygındır, fakat Anahtar değiştirim sorunu vardır; gizli anahtar güvenli taşınmalıdır. Anahtar sayısı: Bir değil kullanıcıların her çift için bir tek anahtarı olmalıdır. 6 kullanıcı için $\frac{6\times 5}{2} = 15$ anahtar.
- Alice ya da Bob birbirlerine bilgi yazabilirler. Günlük ikisinde de aynı veriler var. Birbirlerinin yerine mesaj yazabilirler.

Ör: Kasa örneği. Alice kilitler, Bob açar. Aynı anahtar.

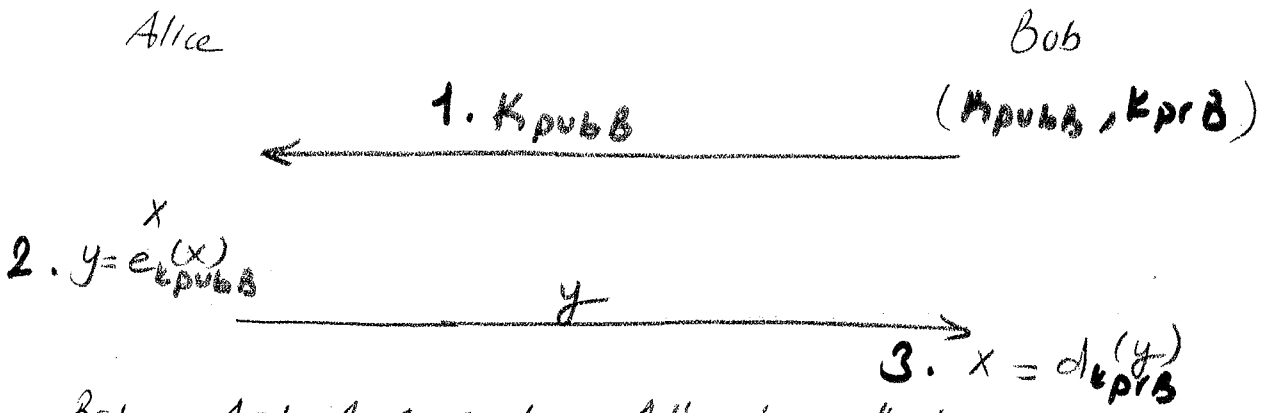
Asimetrik Kriptografi'nin Mantığı;

- Posta kutusu örneği; Herkes mektup atabilir. Sadece anahtarın sahibi onları okuyabilir. Gönderilen mesajı gönderici bile açamaz; çünkü şifreyi çözecek anahtar yok.
- Asimetrik Kriptografi'de Anahtar;



- Anahtar üretimi sırasında birbiriyle ilintili bir anahtar çifti K_{pub}-K_{pr} hesaplanır.
- Açık anahtar ve özel anahtar ile güvenlik sağlanır.

Açık Anahtarlı Şifreleme için Temel Protokol;



- Bob, Açık Anahtarı'nı Alice'e gönderir.
- Alice, bu anahtarı kullanarak şifreler ve şifreli mesajı karşıya gönderir.
- Bob gönderdiği gizli anahtarı ile mesajı açar.

- Anahtar dağıtım ve saklama problemi çözülür.
- Bütünlük ve Kimlik doğrulama sağlanır. Gizlilik sağlanmaz.

NOT Simetrik kullanılma nedeni; ✔ Hızlı
✔ Yazılım donanımsal gerçekleştirimi kolay.

Açık Anahtarlı Kriptografinin Güvenlik Mekanizmaları;

Anahtar değişimi : Diffie Hellman, RSA

Kimlik doğrulama ve Dijital imzalar : RSA, DSA --- mesaj bütünlüğü sağlamak için

Tekrarlama : Dijital imza ile protokol kullanım.

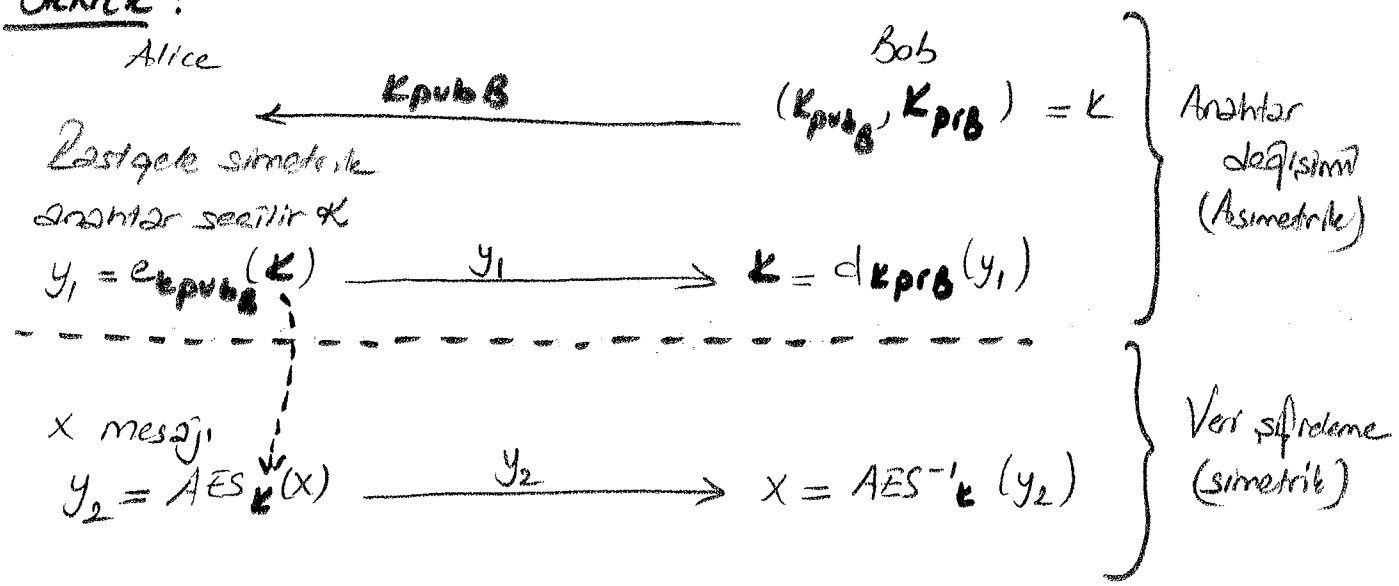
Şifreleme : Dezavantaj : Simetrik'e göre 1000 kat yavaş.

** Pratikte HİBRİT sistemler kullanılır. Asimetrik ve Simetrik algoritmaların avantajları birleştirilmiştir.

1. Anahtar değişimi : Anahtarlar As. şif. yöntemi ile şifrelenerek gönderilir. Dijital imzalar Asimetrik algoritmalar ile yapılır (yavaştır)

2. Veri şifreleme (hızlı) : Simetrik şif. yöntemi ile yapılır. Örneğin; Anahtar 128 byte, Metin 1 GB olabilir. Bu nedenle metin hızlı olmalıdır Anahtarın hızlı olması gerekmez.

ÖRNEK :



Açık Anahtar Algoritmaları Nasıl Oluşturulur?

- Asimetrik yapılar tek yönlü fonksiyonlara dayanmaktadır. (BR ; NP Problem)
  - $y = f(x)$ hesaplamak hesapsal olarak kolay.
  - $x = f^{-1}(y)$ hesaplamak ise " " zordur.
- Tek yönlü fonksiyonlar, matematiksel zor problemlere dayanmaktadır.
- Üç ana bölümden oluşur;
  - Factoring integers: (RSA)
    - Birleşik bir n tamsayısının asal çarpanlarını bulmak.
    - (İki asalın çarpımını bulmak kolay)
  - Discrete Logarithm: (Diffie-Hellman, DSA)
    - Verilen a, y ve m için $a^x = y \bmod m$ olarak şekilde x değerini bulmaya çalışır. (a^x üssünü almak kolay)
  - Eliptic Eğriler (EC) : (ECDSA, ECDH)
    - Ayrık logaritma genelleme

Anahtar Uzunlukları ve Güvenlik Seviyeleri:

Symmetric	ECC	RSA, DL	Remark
64 bit	128 bit	≈ 700 bit	Sadece kısa vadeli güvenlik
80 bit	160 bit	≈ 1024 bit	Orta seviyede güvenlik
128 bit	256 bit	≈ 3072 bit	Uzun vadeli güvenlik

- RSA ve DL nin tam karmaşıklığını tahmin etmek zordur.

Açık Anahtar Algoritmaları için Temel Sayılar Teorisi:

Öklid
Genişletilmiş Öklid
Euler'in Phi Fonksiyonu
Fermat's Little Theorem
Euler's theorem

Öklid Algoritması :

- r_0 ve r_1 gibi iki sayının EBOB'u hesaplanır: $gcd(r_0, r_1)$.

ÖRNEK $r_0 = 84$, $r_1 = 30$

84	30	2 +
42	15	2
21	15	3 +
7	5	5
2	1	2
1		

$gcd(30,84) = 2 \cdot 3 = \underline{\underline{6}}$

- Büyük sayılar için çarpanlara ayırma karmaşık ve çoğu zaman olanaksızdır.
- $gcd(r_0, r_1) = gcd(r_0 - r_1, r_1)$ şeklinde ilerlediği gözlenmiştir. İşlemi sürekli olarak bu şekilde yinelemeli yaparsak; $gcd(r_i, 0) = r_i$ olana kadar devam ederiz ve problemin çözümü $= r_i$ olur.

ÖRNEK

$gcd(r_0, r_1)$, $r_0 = 27$, $r_1 = 21$ için

$$gcd(27,21) = gcd(1 \cdot 21 + \underline{6}, \underline{21}) = gcd(\underline{21}, \underline{6})$$
$$gcd(21,6) = gcd(3 \cdot 6 + \underline{3}, \underline{6}) = gcd(\underline{6}, \underline{3})$$
$$gcd(6,3) = gcd(3 \cdot 2 + \underline{0}, \underline{3}) = gcd(\underline{3}, \underline{0}) = 3$$

- Daha uzun sayılar için çok etkili bir yöntem. Karmaşıklığı bit sayısı ile doğrusal olarak büyür.

Genişletilmiş Öklid Alg.

- r_1 mod r_0 modüler tersi bulunur.
- $gcd(r_0, r_1) = s \cdot r_0 + t \cdot r_1$

$$s \cdot r_0 + t \cdot r_1 = 1$$
$$s \cdot 0 + t \cdot r_1 = 1 \mod r_0$$
$$r_1 \cdot t \equiv 1 \mod r_0$$

t, r_1 mod r_0 in tersidir.

ÖRNEK 12 mod 67 nin tersi?

Özel tablosu vardır: $-5 \cdot 67 + 28 \cdot 12 = 1$ dir. Bu durumda, 12 mod 67 nin tersi 28 dir.

Kontrol edecek olursak $12 \times 28 = 336 \equiv 1 \mod 67$

Euler's Phi Function

$m=6$, $\{0,1,2,3,4,5\}$

$gcd(0,6) = 6$
$gcd(1,6) = 1 \leftarrow$
$gcd(2,6) = 2$
$gcd(3,6) = 3$
$gcd(4,6) = 2$
$gcd(5,6) = 1 \leftarrow$

$\varphi(6) = 2$

$m=5$, $\{0,1,2,3,4\}$

$gcd(0,5) = 0$
$gcd(1,5) = 1 \leftarrow$
$gcd(2,5) = 1 \leftarrow$
$gcd(3,5) = 1 \leftarrow$
$gcd(4,5) = 1 \leftarrow$

$\varphi(5) = 4$

- m ile aralarında kalacak asal sayı olsun bulun. [$\{1, \dots, m-1\}$ aralığında]

$$m = p_1^{e_1} \cdot p_2^{e_2} \cdot \dots \cdot p_n^{e_n} \Rightarrow \boxed{\varphi(m) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1})}$$

ÖZEL DURUM

$e_i = 1$ ve $m = p \cdot q$ olmak üzere $\varphi(m) = (p-1)(q-1)$

<u>ÖRNEK</u> $m = 899 = 29 \cdot 31 \Rightarrow \varphi(899) = (29-1)\cdot(31-1) = 28 \cdot 30 = 840$

Fermat's Theorem

$$\left.\begin{aligned} a^p &\equiv a \pmod p \\ a^{p-1} &\equiv 1 \pmod p \end{aligned}\right\} \text{ } p \text{ asal ve } a \text{ tam sayısı için}$$

$$a^{-1} = a^{p-2} \pmod p$$

$a = 2$, $p = 7 \Rightarrow 2^{-1} = 2^{7-2} = 32 \pmod 7$
$= 4 \pmod 7$

Kontrol edelim. $2 \cdot 2^{-1} \equiv 1 \mod p$
$2 \cdot 4 \equiv 1 \mod 7$
$8 \equiv 1 \mod 7$

ÖDEV-2 : RC4

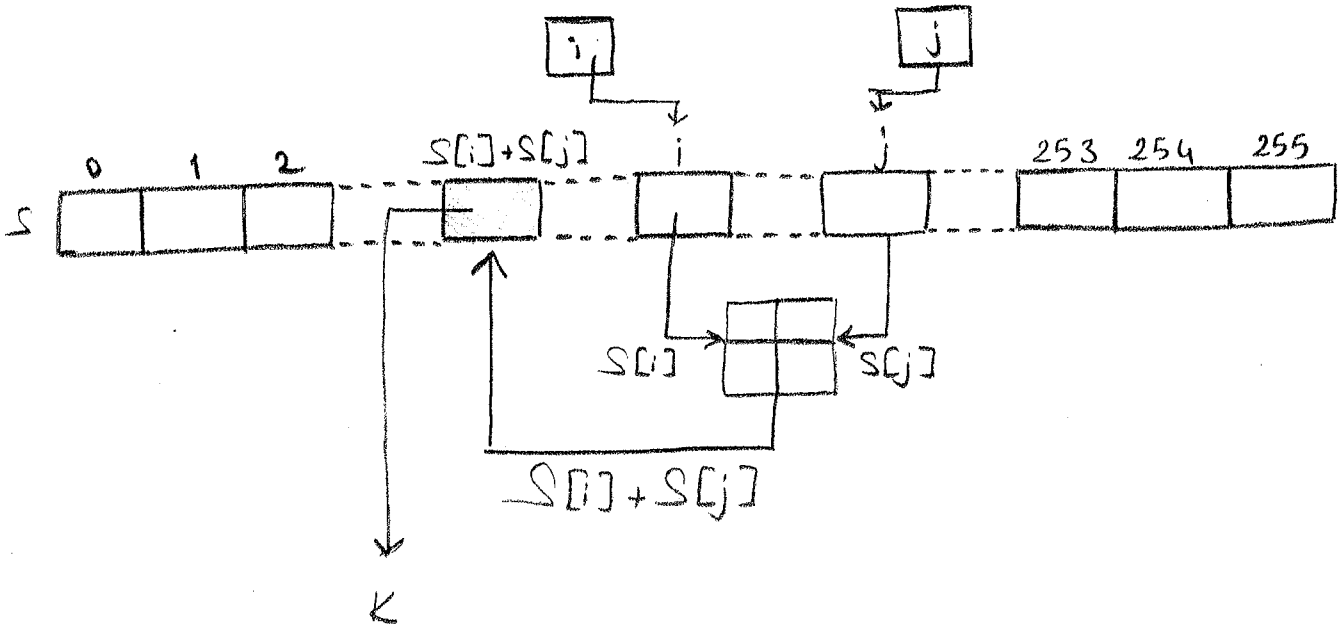
Akan Veri Şifreleme Algoritmaları ve RC4

Kriptolojide açık verileri rastgele bir şifreleme verisiyle, genelde dar veya işlemiyle, karıştıran simetrik anahtar şifreleyicisine akan veri şifreleyicisi denir. Buna karşın blok şifreleyiciler büyük bloklar üzerinde sabit ve değişmeyen dönüşümler yapar. Akan veri şifreleyicisi, blok şifreleyicisinden daha hızlıdır ve daha düşük donanıma ihtiyaç duyar. Öte yandan yanlış kullanıldığında, özellikle aynı başlangıç durumu ikinci kez kullanıldığında, büyük güvenlik açıkları verebilir.

En yaygın kullanılan akan veri şifreleme algoritması RC4 olup A5/1, A5/2, Chameleon FISH, Helix, ISAAC, MUGI, Panama, Phelix, Pike, SEAL, SOBER, SOBER-128 ve WAKE gibi çeşitli akan veri şifreleme algoritmaları mevcuttur.

Yine Ron Rivest tarafından RSA Security'de 1987 yılında geliştirilen RC4 ün, birden fazla açılımı mevcuttur. Ron's Code 4, Rivest Cipher 4. RC4 başlangıçta ticari bir sır olarak 1994 Eylül'ünde Cypherpunks e-posta listesine isimsiz bir kişi tarafından detaylı açıklaması ve kaynak kodu yazıldı. Çok kısa bir süre sonra da sci.crypt haber grubunda da açıklandı. Buna rağmen RC4 ticari bir marka haline getirildi ve marka sorunlarıyla uğraşmamak için RC4'le ARCFOUR ve ARC4 gibi isimler takıldı. Alleged RC4, yani "iddia edilen RC4" tabirinin nedeni ise RSA firmasının şimdiye dek RC4 algoritmasının ne olduğunu resmi olarak açıklamamış olmasıdır.

Şekil'de örnek bir S kutusu için RC4 algoritmasının nasıl çalıştığı gösterilmiştir.



RC4 algoritması OFB'de çalışır ve anahtar giriş verisinden bağımsızdır. 8x8 lik S kutuları ($S_0, S_1, \ldots, S_{255}$) bulunan algoritmada ilk değerleri 0 dır i ve j değişkenleri vardır.

RC4'te rastgele bir sekizli şöyle oluşturulur:

$i = i + 1 \pmod{256}$

$j = j + S_i \pmod{256}$

S_i ve S_j'yi takas et

$t = S_i + S_j \pmod{256}$

$K = S_t$

$k \oplus x = y$

$k \oplus x = y$

Son olarak oluşturulan K sekizlisi ile açık veri dâhil veyalanırsa şifrelenmiş veri, şifrelenmiş veriyle dâhil veyalanırsa da açık veri elde edilir. RC4'teki şifreleme işlemi DES'ten 10 kat daha hızlıdır.

S kutularının ilk değerlerini atamak oldukça kolaydır: $S_0 = 0, S_1 = 1, \ldots, S_{255} = 255$

Daha sonra bir 256 bitlik dizi anahtarla doldurulur. Dizinin dolması için anahtar gerektiği kadar tekrar edilir: $K_0, K_1, \ldots, K_{255}$. j değişkeni sıfırlanır ve şu işlemler yapılır:

for $i = 0$ to 255:

$j = (j + S_i + K_i) \pmod{256}$

S_i ile S_j yi takas et.

RC4 akan veri şifreleme algoritmasının bu kadar olması, ne kadar sade tasarlandığını gösterir.

RC4; BitTorrent, Kerberos, Microsoft® Point-to-Point Encryption, Oracle Secure SQL, Lotus Notes, Apple-AOCE, NDC, NDP, SASL, SSH, SSL, TLS, WPA, ve WEP gibi yerlerde kullanılan en popüler akan veri şifreleme algoritmasıdır. Sıradışı performansı ve hem donanım hem yazılım açısından çok sade gerçekleştirilebilirliği RC4'ün arkasındaki itici güçtür. Ancak; aynı anahtar veri ilkinci kez kullanıldığında veya çıkış verisinin başlangıç kısmı atılmazsa, RC4 son derece savunmasız kalır.

RC4, 1 ile 2048 bit arası uzunluklarda anahtarları destekler. DES ile karşılaştırıldığında, RC4 5 kat daha hızlıdır. Fakat RC4 kırma girişimleri başarı ile sonuçlanmıştır. RC4'ü geliştirmekte kullanılan kod DES'de kullanılanın onda biri kadardır. RC4 algoritması Çıkış Geribesleme Modunda (OFB) ve CFB'ye benzer bir şekilde çalışır. Fakat OFB'den farklı olarak önceki çıkış bloğunun bitlerini giriş bloğunun sağına gönderir. OFB işlevin düzyazı mesajı olmamışken bile yapılabilmesini sağlar. Düzyazıyı gerçekte aldığında algoritma çıktısıyla dâhil veyalanır (XOR). Bu işlem şifreli yazı bloğunu yaratır. OFB hem blok hem de akış şifrelemede kullanılabilir. Bir akış şifrelemede anahtar sonraki durum fonksiyonunu etkiler. Çıktıyı yaratan fonksiyon anahtara dayalı değildir.

RC4 128 bit uzunluğunda şifre ister.

2012 VİZE SORULARI

CEVAPLARI

1) Soyadının shift alg. ya göre şifrelenmesi (Key: Okul No son hanesi)

* Plaintext: İÇKİLİ
  Key : 3

A	B	C	Ç	D	E	F	G	Ğ	H	I	İ
0	1	2	3	4	5	6	7	8	9	10	11

J	K	L	M	N	O	Ö	P	R	S	Ş	T
12	13	14	15	16	17	18	19	20	21	22	23

U	Ü	V	Y	Z
24	25	26	27	28

Ciphertext : LFNLOOL

$I = 11 + 3 = 14 = L$
$Ç = 3 + 3 = 6 = F$
$K = 13 + 3 = 16 = N$
$L = 14 + 3 = 17 = O$

$y = e_k(x) = k + x \mod 29$
$x = d_k(y) = k + y \mod 29$

2) AES difüzyon & konfüzyon için hangi elemanlar kullanılır?

* SubByte → Byte Substitution Layer → $A_0\ A_4\ A_8\ A_{12} \Rightarrow B_0 \ldots$
  $A_1\ A_5\ A_9\ A_{13} \Rightarrow B_1 \ldots$
  $A_2\ A_6\ A_{10}\ A_{14} \Rightarrow B_2 \ldots$
  $A_3\ A_7\ A_{11}\ A_{15} \Rightarrow B_3 \ldots$
  ShiftRows & MixColumn → Diffusion Layer
  AddKey → Key Addition Layer

3)

DES	AES
→ Simetrik Algoritmalar ←	
Blok uzunluğu 64 bit	Blok uzunluğu 128 bit
Key uzunluğu 56 bit	Key uzunluğu 128/192/256 bit
Güvenlik detayında	Bilinen kırma
Bit, Linear Kriptanaliz kırıcı algoritma	bişey yok
Donanım kısmında kullanılır	Galois field (GF(N))
Şifreleme mimarisi Feistel	Rijndael Cipher

AES

```
function AES_k(M)
(k_0, ..., k_10) ← expand k
S ← M ⊕ k_0
for r ← 1 to 10 do
    S ← S(s)
    S ← shiftrows(s)
    if r ≤ 9 then S ← mixcolumn(s) fi
    S ← S ⊕ k_r
end for
returns
```

DES

```
function DES_k(M) // |k| = 56 and |M| = 64
(K_1, ..., K_16) ← keyschedule(k) // |K_i| = 48 for 1 ≤ i ≤ 16
M ← IP(M)
Parse M as L_0 || R_0 // |L_0| = |R_0| = 32
for i = 1 to do 16
    L_i ← R_{i-1}    R_i ← f(k_i, R_{i-1}) ⊕ L_{i-1}
C ← IP^{-1}(L_16 || R_16)
return C
```

Cipher 12 max width (m)

Euclid

```
Do
i = i+1
r_i = r_{i-2} mod r_{i-1}
while r_i ≠ 0
Return
gcd(r_0, r_1) = r_{i-1}
```

Extended

```
Do
i = i+1
r_i = r_{i-2} mod r_{i-1}
q_{i-1} = (r_{i-2} - r_i) / r_{i-1}
s_i = s_{i-2} - q_{i-1} * s_{i-1}
t_i = t_{i-2} - q_{i-1} * s_{i-1}
while r_i ≠ 0
gcd(r_0, r_1) = r_{i-1}
s = s_{i-1}
t = t_{i-1}
```