ÖNSÖZ

Tezimde yapmış oldukları yardımlardan ve desteklerinden dolayı değerleri hocalarım Prof. Dr. A. Emre Harmancı ve Yrd. Doç. Dr. Berna Örs Yalçın'a teşekkürü bir borç bilirim.

Ocak 2008

Ferhat Karakoç

Bilgisayar Mühendisi

İÇİNDEKİLER Sayfa

TABLO LİSTESİ

Sayfa No:

ŞEKİL LİSTESİ

Sayfa No:

SEMBOL LİSTESİ

3DES **:** Üçlü DES (Triple DES)
DES **:** Veri Şifreleme Standardı (Data Encryption Standard)
SPN **:** Yer Alma- Yer Değiştirme Ağı (Substution-Permutation Network)

KRİPTO ANALİZDE MELEZ BİR YÖNTEM: ÇAKIŞMA SALDIRISI

ÖZET

Şifre bilimi olan kriptoloji kendi içerisinde iki branşa ayrılır. Bu branşlardan biri olan kriptografi şifreleme sistemlerinin oluşturulmasını içerirken kriptoanaliz şifreleme sistemlerinin analizi ile uğraşır. Kriptoanalizin zamanla gelişmesi ile şifreleme sistemleri kırılabildiği için bu gelişme kriptografinin de gelişmesine neden olmaktadır. Sistemin kırılması, sistemde kullanılan gizli anahtarın ele geçirilmesi ve/veya şifrelenmiş (kapalı) metinden şifrelenmemiş (açık) metnin bulunabilmesi demektir. Bir şifreleme sisteminin saldırılara dayanıklı olması ise yapılan saldırı ile sistemin kolay bir şekilde kırılamaması demektir. Kolaylıktan kasıt sistemin kırılabilmesi için zaman, kaynak ve eldeki verilerin makul olmasıdır. Kriptografinin gelişmesi ile de kriptoanaliz gelişmesini devam ettirmek zorunda kalmaktadır. Dolayısıyla bu iki branş birbirini tetikleyerek ilerlemektedir.

Sistemi analiz ederek sistemin kırılması için gerekli süre, kaynak ve veri ihtiyacını ortaya koyan kriptoanaliz içinde temelde iki farklı yaklaşım bulunmaktadır. Klasik kriptoanaliz bu yaklaşımlardan bir tanesi iken yan kanal saldırıları da diğeridir. Klasik kriptoanalize matematiksel veya teorik kriptoanaliz de denmektedir.

Klasik kriptoanalizde saldırı direk olarak şifreleme algoritmasına yapılmaktadır. Algoritmanda bulunan zayıflıklar ortaya çıkarılarak bu zayıflıklar ile sistemde kullanılan anahtar elde edilmeye çalışılır. Bu zayıflıklara sistemin bir olasılık ile doğrusal (*linear*) modelinin çıkarılması, açık metin faklarının algoritma çıkışında bir olasılık ile başka bir farka yaklaşması, açık metin toplamlarının algoritma içerisinde bazı özellikleri sağlayarak ilerlemesi örnek olarak verilebilir. Klasik kriptoanalizin tipik özelliği elde sadece açık ve/veya kapalı metinlerin olmasıdır.

Yan kanal saldırılarında ise saldırı şifreleme sisteminin uygulanışına yapılmaktadır. Yan kanal saldırılarında açık ve/veya kapalı metin çiftlerinin yanı sıra fiziksel bilgiler de elde edilebilir. Bu fiziksel bilgiler çekilen akım, ortama yayılan manyetik alan, oluşan ısı gibi büyüklükler olabilir. Bu değerleri elde edebilmek için de şifreleme yapan sisteme sahip olmak gerekir. Mesela bir kart üzerindeki anahtarı bulabilmek için kart elde bulunmalı.

Son zamanlarda, klasik ve yan kanal saldırılarının bir arada kullanılması üzerine çalışmalar yapılmaktadır. Bu yöntemlerin herhangi birinden elde edilen bilgi diğer yöntemde kullanılabilir. Böylece, ortaya daha güçlü bir saldırı çıkmaktadır. İki yöntemin beraber kullanıldığı saldırılardan bir tanesi çakışma saldırısıdır. Bu saldırıda farksal kriptoanaliz ile yan kanal atağı beraber kullanılmaktadır. Şifreleme algoritmasına giren iki farklı girişin aynı çıkışa neden olması ile saldırı gerçekleştirilebilmektedir. Bu saldırı algoritma içerisinde olduğundan ve algoritma çıkışına yansımadığından yan kanal bilgileri kullanılır. Yan kanal bilgileri ile çakışma olduğu anlaşılır ve bu çakışmaya neden olan girişler kullanılarak anahtar hesaplanır.

Yapılan çalışmada DES algoritmasına uygulanan çakışma saldırısının uygulanması yapılıp saldırının geniş bir tanımı oluşturulmuş ve saldırı 3DES algoritmasına uygulanmıştır. 3DES'e yapılan diğer saldırılarla çakışma saldırısı karşılaştırıldığında çakışma saldırısının daha etkili olduğu görülmüştür.

A NEW COMBINED CRYPTANALYSIS METHOD: COLLISION ATTACK

SUMMARY

Cryptology is the practice and study of hiding information. It has two main areas. One of them is cryptography that is the use and practice of cryptographic techniques. The other one is cryptanalysis. The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme to get the secret. Because cryptanalysis is being more powerful, cryptography also getting stronger. They challenge each other.

There are two main approach in cryptanalysis. Classical cryptanalysis is one of them. The other one is side channel attacks. Classical cryptanalysis is called as mathematical or theorical cryptanalysis also. Other name of side channel attacks is electronically cryptanalysis.

In classical cryptanalysis, weakness is searched in the algorithm itself while in side channel attack the weakness is search in the usage of the algorithm.

In recent years, there are some studies on combined cryptanalysis. Combined cryptanalysis is the cryptanalysis in which classical and electronic cryptanalysis is used.

In this thesis, we searched a new method, which was applied on DES algorithm. We give a general definition for the method whose name is collision attack and apply the method to the 3DES algorithm. We show that, the best attack to the 3DES is the collision attack.

1. GİRİŞ

1.1 Tezin Amacı

Bu çalışma kapsamında farksal kripto analiz [1] ile güç tüketimi analizinin [2] kullanıldığı çakışma saldırı [3] yönteminin DES algoritması [1] üzerindeki uygulaması incelendi ve saldırı simülasyon kullanılarak gerçeklendi. Saldırının genel bir tanımı yapıldı ve 3DES [4] algoritması üzerinde uygulaması gerçeklendi.

1.2 Çakışma Saldırısının DES Algoritması Üzerinde Uygulanması

[3]'te anlatılan çakışma saldırısı incelendi ve saldırının uygulaması gerçeklendi. Uygulama C programlama dili kullanılarak gerçeklendi. Güç tüketimi için ise akım ölçme fiziksel olarak yapılmadı bunun yerine simülasyon kullanıldı.

1.3 Saldırının Genel Tanımı

Çakışma saldırısı ile ilgili yapılan çalışmalar genelde algoritmaya özel anlatılmaktadır [3,5]. Çakışma saldırısının genel tanımı bu tez kapsamında yapılmıştır. Saldırı genel bir ifade ile Bölüm 5'de anlatılmıştır.

1.4 Saldırının 3DES Algoritması Üzerinde Uygulaması

3 adet DES algoritmasının art arda kullanıldığı 3DES algoritmasına çakışma saldırısının nasıl uygulanabileceği gösterildi ve uygulaması yapıldı. Bölüm 6'da 3DES algoritmasına saldırının uygulanması anlatılmıştır.

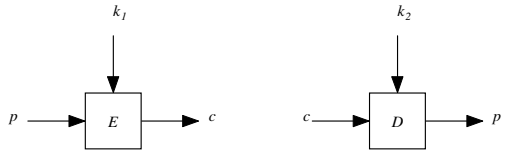
2. BLOK ŞİFRELEME ALGORİTMALARI

2.1 Giriş

Bir şifreleme sistemi (*P,C,K,E,D*) beşlisinden oluşur [6]. *P* açık metin (şifrelenecek metin) kümesi, *C* kapalı metin (şifreli metin) kümesi, *K* anahtar kümesi, *E* şifreleme kuralı (algoritması) ve *D* şifre çözme kuralıdır. Açık metinden kapalı metin elde etme işlemine şifreleme, kapalı metinden açık metin elde etme işlemine de şifre çözme işlemi denir. $p \in P, c \in C, k_1 \in K, k_2 \in K$ olmak üzere şifreleme ve şifre çözme işlemleri sırayla eşitlik (2.1) ve (2.2)'deki gibidir.

$$c = E(k_1, p) \tag{2.1}$$

$$p = D(k_2, c) \tag{2.2}$$



Şekil 2.1 : Şifreleme ve şifre çözme

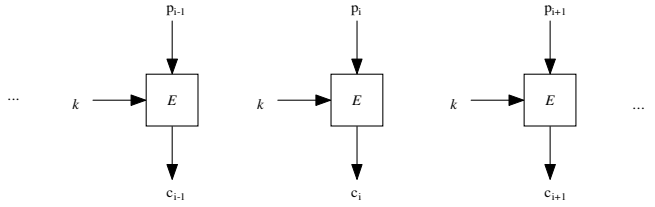
$k_1 = k_2$ ise veya birinden diğeri kolayca elde edilebiliyor ise şifreleme sistemine simetrik anahtarlı şifreleme sistemi denir [7]. k_1 anahtarı açık (herkes tarafından bilinir) ve k_2 anahtarı gizli ise ve k_1'den k_2'yi bulmak pratik olarak imkansız ise bu tip şifreleme sistemlerine de açık anahtarlı şifreleme sistemleri denir [7].

Simetrik anahtarlı şifreleme olan blok şifreleme sistemlerinde açık metin bloklara ayrılır ve her blok bir bütün olarak şifrelenir [7]. ECB modunda şifrelemede bloklar birbirlerinden bağımsız şifrelendiği için şifrelenmiş verinin iletimi sırasında meydana gelen hata sadece hatanın olduğu bloğun bozulmasına neden olur, diğer bloklar

bundan etkilenmez [7]. Metin içinde aynı bloklara karşılık gelen şifrelenmiş bloklar aynı olur. Bu durum sistemin güvenilirliği için dezavantaj oluşturur. Blok şifrelemedeki bu dezavantajı ortadan kaldırmak için bazı farklı modları geliştirilmiştir. Bu modlar Bölüm 2.2 de yer almaktadır.

Blok şifrelemede kullanılan algoritmalara blok şifreleyiciler denir [7]. Modern blok şifreleyiciler birim şifreleyicilerin çarpımından oluşur [6]. Birim şifreleyiciler yerini alma ve yer değiştirme işlemlerine dayanan şifreleyicilerdir [8]. Çarpım şifreleyiciler birim şifreleme algoritmalarının bir arada kullanılması ile oluşmuş şifreleme yapılarıdır. Çarpım şifreler [9]'da Shannon tarafından yayımlanan makaleye dayanmaktadır.

Blok şifreleyicilerde kullanılan parametreler blok uzunluğu ve anahtar uzunluğudur. Şifrelenecek olan veri blok uzunluğunda bloklara bölünerek şifrelenir.

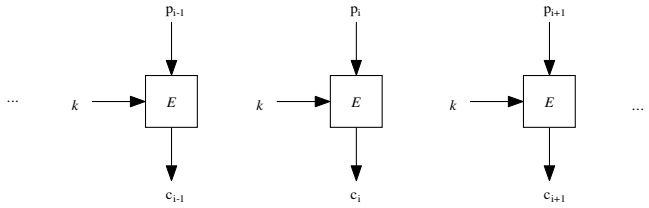


Şekil 2.2 : Blok şifreleyici

2.2 Şifreleme Modları

2.2.1 Elektronik Kod Modu

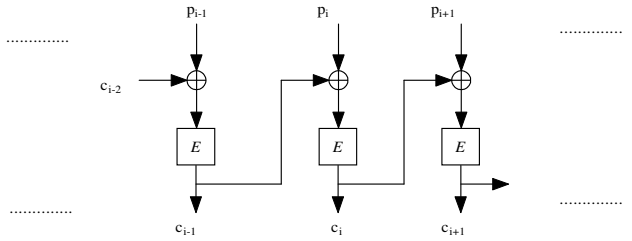
Elektronik kod modunda şifrelenecek her bir blok birbirinden bağımsız olarak şifrelenir [7].



Şekil 2.3 : Elektronik kod modu

2.2.2 Kapalı Metin Zincirleme Modu

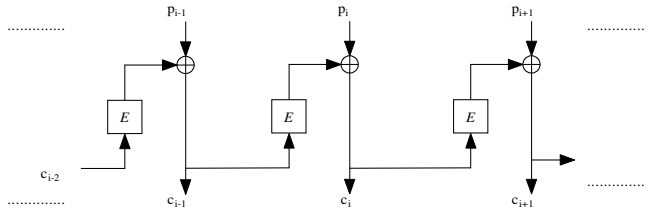
Kapalı metin zincirleme modunda bir bloğun şifrelenmiş hali bir sonraki bloğun şifrelenmesi sırasında kullanılır [7].



Şekil 2.4 : Kapalı metin zincirleme modu

2.2.3 Çıktıyı Geri Besleme Modu

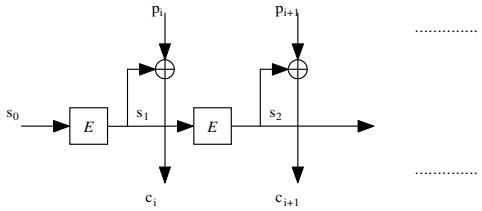
Çıktıyı geri besleme modunda şifrelenmiş veri tekrar şifrelenmektedir [7].



Şekil 2.5 : Çıktıyı geri besleme modu

2.2.4 Girdiyi Geri Besleme Modeli

Girdiyi geri besleme modunda başlangıçta seçilen bir değer sürekli şifrelenir ve açık metin ile şifreleme sonucu elde edilen değer *yada* (*xor*) işlemine sokularak kapalı metin elde edilir [7].



Şekil 2.6 : Girdiyi geri besleme modu

2.3 Blok Şifreleyici Tasarım Kriterleri

Şifreleme ve şifre çözme işlemlerinin kolay yapılabilmesi yanında $c = E(k, p)$ ve $p = D(k, c)$ eşitliklerinden k anahtarı kolaylıkla bulunamamalıdır. İlk defa Shannon tarafından önerilen tasarım ölçütleri yayılma ve nüfuz etmedir [9].

2.3.1 Yayılma

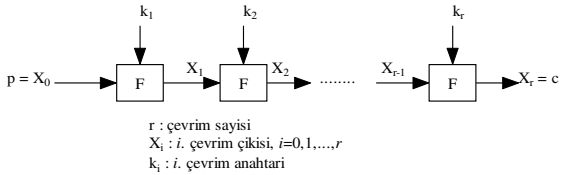
Yayılma, şifreli metin ile anahtar arasındaki ilişkinin olabildiğince karışık olması demektir. Bu ilişki doğrusal olmamakla beraber anahtarın tüm bitlerinin çıkışta eşit şekilde etkili olması sağlanır [9].

2.3.2 Nüfuz Etme

Nüfuz etme ile anahtarın ve açık metnin her bitinin kapalı metni etkilemesi sağlanır. Böylece şifreli metin ile açık metin arasında istatistiksel bağlılık kurulamaz hale getirilir [9].

2.4 İteratif Blok Şifreleyiciler

Blok şifrelerin büyük bir kısmı iteratif yapıdadır [7]. Çevrim sayısı arttıkça yayılma ve nüfuz etme artmaktadır. Çevrim adı verilen fonksiyon çevrim sayısı kadar çalıştırılır. Bir çevrimin çıkışı bir sonraki çevrimin girişi olur. İlk çevrim girişi açık metin, son çevrim çıkışı da kapalı metindir. Çevrim fonksiyonu çevrim girişini ve çevrim anahtarını alır ve bir çevrim çıkışı oluşturur. Çevrimlerde genelde farklı çevrim anahtarı kullanılır. Çevrim anahtarı bir anahtar üretme algoritması kullanılarak ana bir anahtardan üretilir.



Şekil 2.7 : İteratif blok şifreleme

2.4.1 Yerine Koyma- Yer Değiştirme Ağı

Yerine koyma- yer değiştirme ağı döngülü blok şifreleme yapısındadır [6]. İçerisinde temel iki işlem vardır. Blok uzunluğu $n = l.m$ olsun.

$$\pi_s : \{0,1\}^l \rightarrow \{0,1\}^l \qquad (2.3)$$

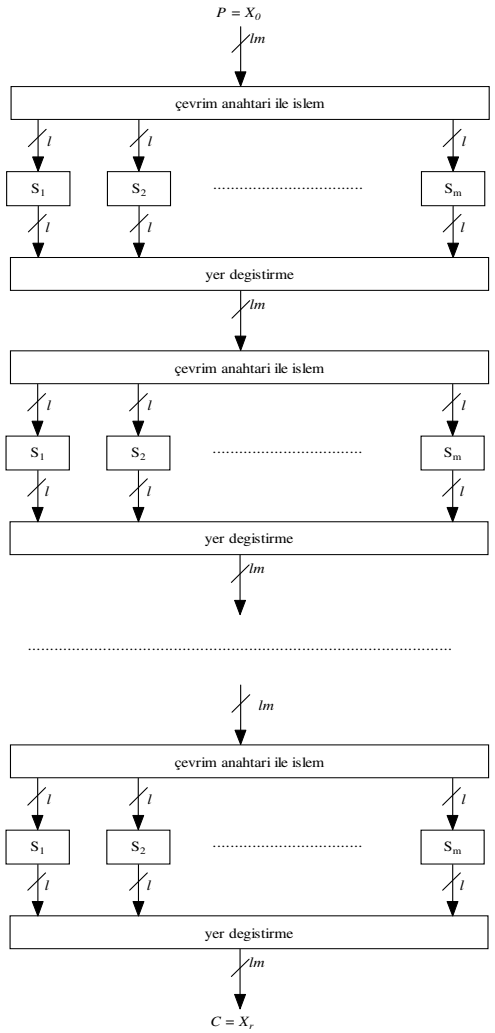
$$\pi_p = \{1,...,lm\} \rightarrow \{1,...,lm\} \qquad (2.4)$$

(2.3) numaralı denklem yerine koyma işlemi olup doğrusal olmayan bir fonksiyondur ve s-kutusu olarak adlandırılır. Bu fonksiyon ile yayılma yani anahtar bitleri ile açık

metnin bitlerinin birbirine karıştırılması sağlanır. Ağ içindeki her bir çevrimde *m* adet s-kutusu bulunur.

(2.4) numaralı denklem yer değiştirme fonksiyonudur. Bu işlem de anahtar bitlerinin kapalı metin içerisinde dağılmasını sağlar. Her çevrimde bir adet yer değiştirme işlemi bulunur.

Yerine koyma- yer değiştirme ağının yapısı Şekil 2.8'de gösterilmiştir.



Şekil 2.8 : Yerine koyma – yer değiştirme ağı

2.4.2 Feistel Yapıları

Feistel yapılarında çevrim bloğu iki eşit parçaya ayrılır ve her çevrimde sadece bir parça değiştirilir [7]. Diğer parça çevrim çıkışına bir değişikliğe uğramadan yansır. Feistel yapılarının temel özelliği şifreleme işlemi ile şifre çözme işleminin aynı olmasıdır. Şifreleme ile şifre çözme arasındaki tek fark çevrim anahtarlarının ters sırada olmasıdır. Şifreleme yapılırken çevrim anahtarları sırasıyla $\{k_0, k_1, \ldots, k_r\}$ iken şifre çözme işleminde çevrim anahtarları sırası ile $\{k_r, k_{r-1}, \ldots, k_0\}$'dır.

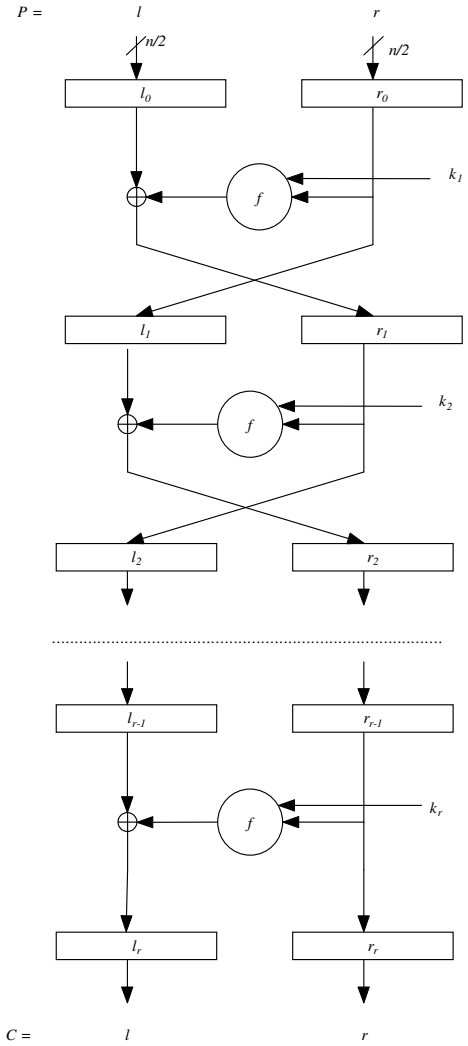
F çevrim fonksiyonu olmak üzere (2.5) eşitliği ile ifade edilir.

$$F(l,r) = (r, l \oplus f(r,k)) \tag{2.5}$$

F fonksiyonu iki parça giriş alır (çevrim girişinin sağ ve sol yarımları) ve iki parça çıkış üretir. *f* çevrim içerisinde kullanılan bir fonksiyondur ve anahtarı da giriş olarak alarak üzerinde içlem yapar. Yayılma ve nüfuz etme özelliklerinin sağlanması için içerisinde s-kutuları ve yer değiştirme işlemleri vardır.

Algoritmanın tersinin kendisine eşit olması için son çevrimde sağ ve sol yarımların yer değiştirme işlemi bulunmaz.

Yapı grafiksel olarak Şekil 2.9'da yer almaktadır.



Şekil 2.9 : Feistel yapısı

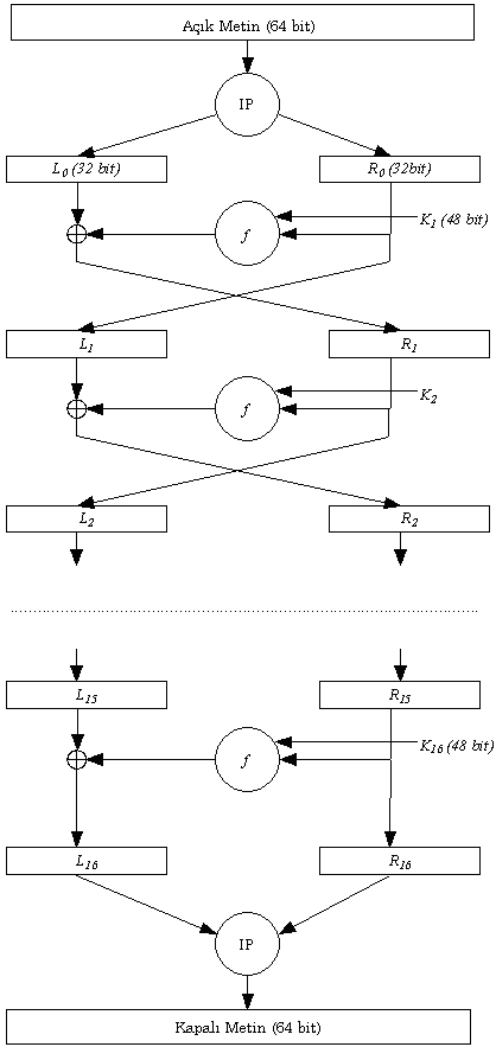
2.5 DES

DES (Data Encryption Statndard) algoritması, 1970 yılında IBM tarafından geliştirilen Lucifer algoritmasının biraz daha geliştirilmiş halidir [7]. DES 1974 yılında IBM ve NSA'nın birlikte çalışması ile geliştirilmiştir. Algoritmanın anahtar uzayının günün teknolojisi karşısında küçük olması nedeniyle algoritma artık saf hali ile kullanılmamaktadır. Değiştirilmiş halleri kullanılmaya devam etmektedir.

2.5.1 Algoritma

DES algoritması Feistel yapısında olup 16 döngüden oluşur. İlk döngü girişinden önce başlangıç permütasyonu, son döngüden sonra da başlangıç permütasyonunun tersi vardır. Başlangıç permütasyonunun güvenlik açısından bir etkisi bulunmaz. Blok uzunluğu 64 bittir. 64 bitlik anahtarın 56 biti kullanılır, diğer 8 bit eşlik biti olarak kullanılabilir.

Algoritma Şekil 2.10'da bulunmaktadır.



Şekil 2.10 : DES algoritması

Başlangıç permütasyonu olan IP Tablo 2.1'de bulunmaktadır. *i*. satırın *j*. sütununda bulunan değer (*i* ve *j* 0'dan başlıyor) permütasyon çıkışının (8*j*+*i*). bitinin girişin hangi bitine karşılık geldiğini ifade eder. Giriş ve çıkış bitlerinin indeksleri de 0'dan başlamaktadır. Bu bölümde anlatılan tabloların okunuşları Tablo 2.1'deki gibidir.

Tablo 2.1 : DES algoritması başlangıç permütasyonu

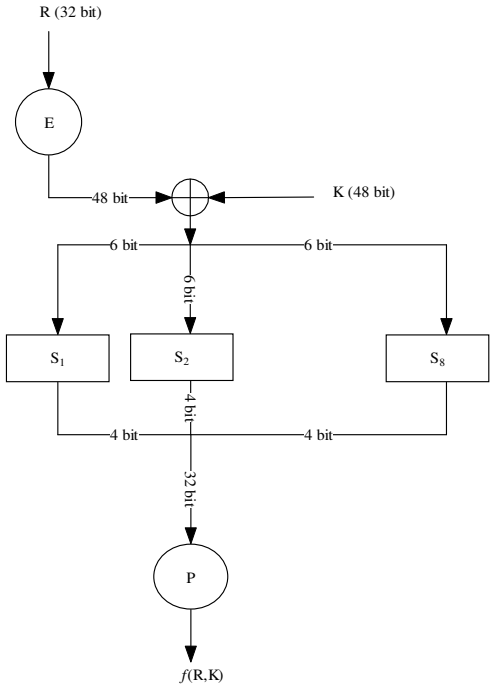
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Başlangıç permütasyonunun tersi (IP^{-1}) Tablo 2.2'de bulunmaktadır.

Tablo 2.2 : DES algoritması çıkış permütasyonu

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

f fonksiyonu içerisinde 32 bitlik giriş 48 bite genişletilir ve çevrim anahtarı ile bit bazında *yada* işlemine sokulur. Bit bazında işlem demek her iki girişteki *i*. bitlerin kendi aralarında *yada* işlemine girmesi demektir. *Yada* işlemi sonucunda oluşan 48 bitlik veri 6'şar bitler bazında 8 S-kutusuna girer. Her bir S-kutusu 6 bite karşılık 4 bitlik çıkış üretir. S-kutuları çıkışı oluşan 4x8=32 bitlik veri çevrim permütasyonundan geçirilir ve *f* fonksiyonunun çıkışı elde edilir. *f* fonksiyonu grafiksel olarak Şekil 2.11'de bulunmaktadır.



Şekil 2.11 : DES algoritması *f* fonksiyonu

Genişletme fonksiyonu (E) Tablo 2.3'te bulunmaktadır.

Tablo 2.3 : DES çevrim fonksiyonu içerisindeki genişletme fonksiyonu

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

S-kutuları Tablo 2.4'te bulunmaktadır.

Tablo 2.4 : DES algoritması s-kutuları

S_1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S_2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

S_4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

S_5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

S_6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S_7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

S_8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

S-kutuları 4x16 boyutunda matristen oluşur. Satırları, girişin ilk ve son bitleri indeksler, sütunları da girişlerin ortada kalan 4 biti indeksler. Örneğin S_1'e 001011 girişi geldiğinde çıkış 01 = 1. satırın 0101 = 5. sütunudur. Değerler 10'luk tabanda yazılmıştır.

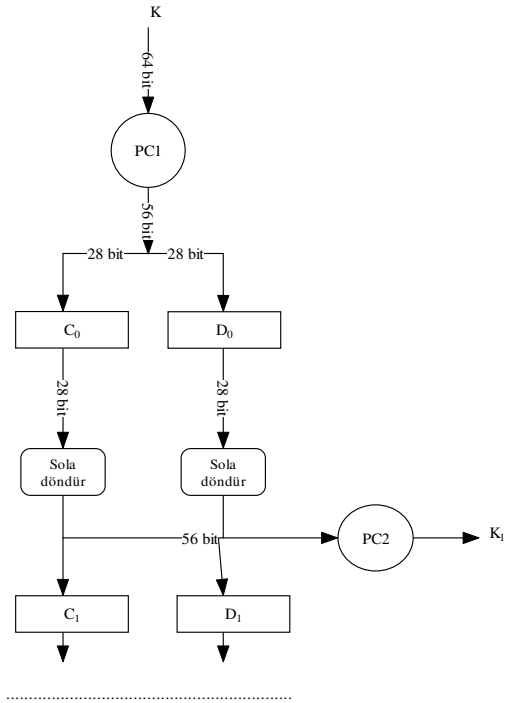
Çevrim sonundaki permütasyon Tablo 2.5'te bulunuyor.

Tablo 2.5 : DES algoritması çevrim permütasyonu

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

2.5.2 Anahtar Üreteci

64 bitlik anahtardan 16 adet 48 bitlik anahtar üretilir. Grafiksel olarak Şekil 2.12'deki gibidir.



Şekil 2.12 : DES çevrim anahtarı üreteci

PC1 ile 64 bitlik anahtardan 56 bit seçilir. Yapısı Tablo 2.6'daki gibidir.

Tablo 2.6 : DES anahtar üreteci PC1 permütasyonu

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

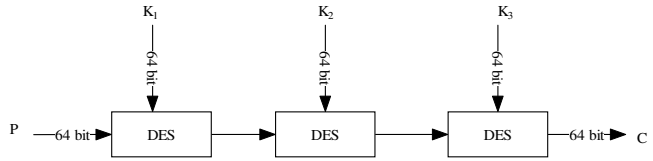
Çıkan 56 bit iki eşit parçaya ayrılır ve sola döndürme işlemi yapılır. Sola döndürme miktarları sırayla 1, 1, 2, 2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 1 şeklindedir. Sola döndürme işleminden sonra PC2 ile 56 bitten 48 bit seçilir. PC2 Tablo 2.7'de bulunmaktadır.

Tablo 2.7: DES anahtar üreteci PC2 fonksiyonu

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

2.6 3DES

DES algoritmasında 56 bit anahtar kullanılması bilgisayar teknolojisinin gelişmesiyle birlikte DES'e karşı deneme-yanılma saldırısını güçlendirmiştir [7]. Bu nedende art arda 3 DES şifrelemesini içeren 3DES [4] kullanılmaktadır. Anahtar uzunluğu 56 x 3 = 168 bit olduğu için de deneme yanılma ile anahtar bulma günün teknolojisi ile imkansız hale gelmiştir. Yapısı Şekil 2.13'te görülmektedir.



Şekil 2.13 : 3DES

3. FARKSAL KRİPTOANALİZ

3.1 Giriş

Farksal kripto analiz DES, GDES, Lucifer, FEAL dahil olmak üzere bir çok sayıda blok şifre sistemine uygulanmış bir seçilmiş açık metin saldırısıdır [7,10]. Seçilmiş açık metin saldırısı, istenilen açık-kapalı metin çiftlerinin elde edilip kullanılabildiği saldırı demektir [7]. İlk olarak Biham ve Shamir tarafından geliştirildi ve DES'in indirgenmiş döngü çeşitlerine uygulandıktan sonra 16 döngülü DES algoritmasına uygulandı [10]. Farksal kriptoanaliz açık metin farklarının algoritma içerisinde ilerlemesine dayanmaktadır. Farkın algoritmada ilerlemesi istatistiksel olarak hesaplanır ve çıkışta bu istatistiksel özellik gözlenir. İstatistiksel olmasının nedeni algoritma içerisinde yer alan ve doğrusal olmayan S-kutularıdır. Fark iki farklı girişin *yada* işlemi sonucu olarak tanımlanır.

3.2 Yerine Koyma- Yer Değiştirme Ağı Üzerinde Farksal Kripto Analiz

Yerine koyma- yer değiştirme ağı üzerinde farksal kripto analiz basit bir ağ üzerinde anlatılmıştır [6]. Ağ Şekil 3.1'de bulunmaktadır. Blok uzunluğu 16 bit, her döngüde 4x4 boyutunda 4 S-kutusu var ve permütasyon da şekilde görüldüğü gibidir. Örnekte ağ 5 çevrim ve son çevrimde permütasyon işlemi yapılmaktadır. Ayrıca çıkışta bir anahtar ile bit bazında *yada* işlemi yapılmaktadır. Anlatılacak yöntem ile K^5 yani çıkışta kullanılan anahtar bulunacaktır. Algoritmada ayrıca s-kutuları da aynıdır. S-kutusunun fark tablosu Tablo 3.1'de bulunmaktadır. Fark tablosu bir giriş farkına karşı bir çıkış farkının kaç farklı giriş ile sağlandığını gösterir. 16 farklı giriş olduğundan bu 16 farklı girişi bir farka götüren tek bir giriş bulunur. Bu bir fark için 16 farklı giriş çifti demektir. Bu nedenle s-kutusunun fark tablosunda bir satırda bulunan sayıların toplamı 16'dır.

Tablo 3.1 : S-kutusu fark tablosu

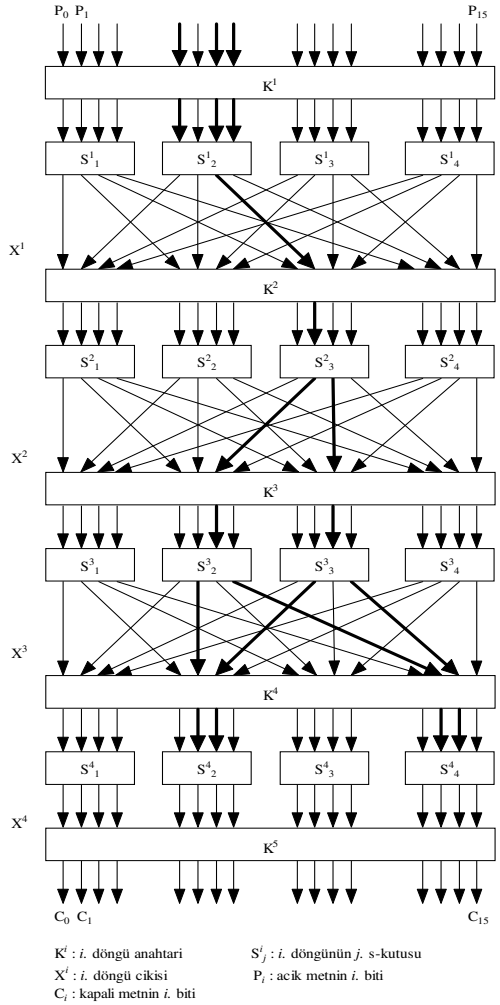
giriş farkı \ çıkış farkı	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Tabloyu yorumlamada kolaylık açısından bir örnek vermek gerekirse, giriş farkı 5 (0101) ve çıkış farkı 1 (0001) olan 4 farklı giriş çifti bulunduğu tablodan görülmektedir.

Şekil 3.1 iki farklı girişin paralel olarak aktığı düşünülerek yorumlanmalıdır. İki verinin algoritma içerisinde akışında koyu oklara karşılık gelen bitler farklı, diğer bitler aynıdır.

Farkı (0000 1011 0000 0000) olan iki farklı açık metin seçilsin. Bu fark sadece S^1_2 kutusuna girer. Kutu giriş farkı 1011 = B olan giriş çiftlerinden 8 tanesinde çıkış farkını 0010 = 2 vermektedir. Girişe uygulanan fark s-kutusundan 8/16 = 1/2 olasılıkla şekil 3.1'de görüldüğü gibi çıkışa yansımaktadır. İlk döngü sonucunda fark X^1_{10} bitine yansır. Fark ikinci döngüde S^2_3 kutusuna (0100) olarak yansır. S-kutusu giriş farkı 4 olan 6 farklı giriş çifti için çıkış farkını 6 (0110) vermektedir. İkinci çevrim sonucunda iki açık metin farkı 1/2 x 6/16 = 3 / 16 olasılıkla (0000 0010 0010 0000) olmaktadır. Bu fark 3. çevrimde S^3_2 ve S^3_3 kutularına yansır. S-kutusu giriş farkı 2 olan 6 giriş çifti için çıkış farkını 5 vermektedir. 3. çevrimde iki s-kutusu da birlikte çıkıştaki farkı sağladığı için çıkışın Şekil 3.1'de görüldüğü gibi olması olasılığı 6/16 x 6/16 = 9/64'tür. Açık metin farkının 3 döngü sonunda (0000 0110 0110 0000) olmasının olasılığı 3/16 (ilk iki döngü olasılığı) x 9/64 (üçüncü döngü olasılığı) = 27/1024'tür. Bir giriş farkı için 3. çevrim çıkışının farkının Şekil 3.1'de

gösterildiği gibi olma olasılığı 27/1024 olduğundan, üçüncü döngü sonucu Şekil 3.1'de gösterildiği gibi fark oluştuğunun görülmesi için yaklaşık 1024/27 giriş çifti seçilmelidir.



Şekil 3.1 : Farksal kripto analizi

Giriş farkları (0000 1011 0000 0000) olan açık-kapalı metin çiftleri seçilir. 16 farklı $K^5_4K^5_5K^5_6K^5_7$ değeri için S^4_2 nin girişine tersten gidilir ve 0110 farklını sağlayan çiftler sayılır. Farkı en fazla sağlayan aday anahtar doğru kabul edilir. Benzeri işlem 16 farklı $K^5_8K^5_9K^5_{10}K^5_{11}$ değeri içinde S^4_2 girişine tersten gidilerek yapılır ve doğru anahtar parçası bulunur. Çıkışta kullanılan anahtarın 8 biti bulunmuş olunur. Başka fark akışı seçilerek diğer 8 bit de bulunur. Eğer anahtar üretme algoritması tersi alınabilir ise şifreleme algoritması çıkışında bulunan anahtar ile ana anahtar bulunur. Tersi alınabilir değilse şifreleme algoritmasının döngü sayısı bir azaltılmış olduğundan aynı işlemler K^4 ü bulmak için yapılır.

4. YAN KANAL SALDIRILARI

Yan kanal saldırısı ile anahtar, algoritmanın zayıflığından yararlanma yerine algoritmanın elektroniksel uygulanması sonucu çalıştığı elektriksel ortamın sağladığı bilgilerden yararlanarak bulunur. Bu bilgiler zamanlama bilgisi, güç tüketimi olabilmektedir [11,12].

Yan kanal saldırısı klasik kripto analizden daha güçlü olabilirken şifreleme yapılan sisteme fiziksel erişim gereklidir. Yan kanal saldırılarının uygulandığı cihazlar genelde ağ üzerindeki bir bilgisayar yerine basit elektronik cihazlardır.

4.1 Saldırı Çeşitleri

4.1.1 Zamanlama Saldırısı

Algoritmanın uygulanmasından doğan zayıflıktan faydalanılır. Algoritma içerisindeki parçaların çalışma süresinin analizine ve sürenin veriye bağlılığının incelenmesine dayanır [11]. Veriye bağımlılık genelde koşullu dallanmalardan kaynaklanmaktadır. Elektronik ortamda bulunan algoritmanın bilinmesi saldırıyı güçlendirmektedir.

Zamanlama bilgileri elde edilirken oluşan gürültüleri yok etmek için yeterli sayıda ölçüm yapmak ve istatistiksel filtreler kullanmak gerekir.

Zamanlama saldırısına örnek olarak *mod* alma işlemi verilebilir. *x* mod *n* algoritma içerisinde bir işlem olsun ve *n* gizli bilgi olsun. *x* *n*'den küçükken bir işlem yapılmazken *x* *n*'den büyükken çıkarma işlemi yapılır. *x*'e 0'dan başlayıp büyüterek değerler verilir, zaman açısından ani bir büyüme görüldüğünde büyümeye neden olan *x* değeri *n*'dir denir.

4.1.2 Güç Analizi Saldırısı

Şifreleme algoritması elektronik sistem üzerinde çalışır ve elektronik sistem transistörlerden oluşur. Transistörler anahtar gibi davranmaktadır. Eğer anahtar 0 ise akım bloke olur, 1 ise akım oluşur. 0'dan 1'e geçişler akım çekilmesine neden olur.

Güç analizi ile akım çekme farklarından yararlanılarak bilgi elde edilmeye çalışılır. Genelde bilinen bir algoritmanın hangi safhasının işlediğini anlamak için yardımcı bir yöntem olarak kullanılır.

Güç analizi 2'ye ayrılır:

- Basit Güç Analizi (*SPA, Simple Power Analysis*)

  Sadece işlemcinin harcadığı enerji analiz edilir.

- Farksal Güç Analizi (*DPA, Differential Power Analysis*)

  Basit güç analizine ek olarak istatistiksel analizlerden yararlanarak ve hata düzeltme algoritmaları çalıştırarak gerekli örnek sayısını azaltılır.

4.1.3 Hata Analizi Saldırısı

Algoritma işletilirken sıcaklık, elektrik gerilimi gibi çevresel bazı parametreler değiştirilerek işlemcinin hata yapması sağlanır. Yanlış sonuçlar elde edilip doğru sonuçlarla karşılaştırılarak bilgi elde edilmeye çalışılır.

4.1.4 Akustik Analiz

Elektronik devreden çıkan ve duyulabilen veya duyulamayan seslerin hassas cihazlarla kaydedilip incelenmesi sonucu bilgi elde etmeye çalışılır. Shamir tarafından sadece işlemciden çıkan ses ile zamanlama saldırısı yapılabileceği ispatlanmıştır.

4.1.5 Elektromanyetik Analiz

Elektronik devrede oluşan elektromanyetik alanın izlenip değerlendirilmesi ile yapılan yan kanal saldırısı yöntemidir.

4.2 Saldırılara Karşı Önlemler

Elektronik ve fiziksel verilerden bilgi edilebilmesini imkansız hale getirmek için bazı önlemler geliştirilmiştir.

4.2.1 Zamanlama Saldırısına Karşı Önlemler

Gecikme eklemek zamanlama saldırısına karşı bir önlem olabilir. Tüm işlemlerin aynı sürede bitmesi sağlanarak saldırganların işlem süresini hesaplayarak bilgi elde

etmesi engellenebilmektir. Bu yöntem ile tüm işlemler süresi en uzun işlem süresine eşitleneceği için algoritma üzerinde süre iyileştirmesi yapılamaz duruma gelir.

Çarpma ve üs alma sürelerinin eşitlemek de zamanlama saldırılarına karşı etkili olmaktadır.

4.2.2 Güç Analizini Saldırısına Karşı Önlemler

Güç analizi saldırısına karşı geliştirilen önlemlerden bir tanesi harcanan gücün dengelenmesidir. Dengeleme için gereksiz saklayıcılar ve kapılar kullanılır. Dengelemenin sağlanması için diğer bir yöntem de başka bir donanımın enerji harcamayı bir seviyede tutma çalışmasıdır.

Güç analizini imkansız hale getirmek için geliştirilen diğer bir yöntem de uygulama içerisine gereksiz işlemler konulmasıdır. Gereksiz işlemlerden dolayı güç tüketiminde meydana gelen gürültüler analizi zorlaştırmaktadır.

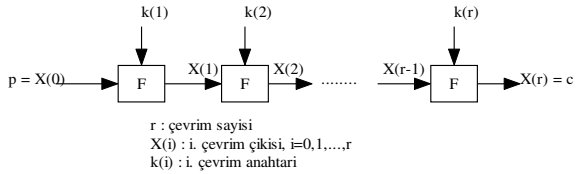
5. ÇAKIŞMA SALDIRISI

Bu bölümde, tezde tanımı yapılan çakışma saldırısı anlatılmaktadır.

5.1 Saldırı Tanımı

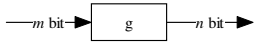
Çakışma saldırısı şifreleme algoritması içerisinde meydana gelen çakışmalara dayanır. Çakışma bir fonksiyonun farklı iki giriş için aynı çıkışı üretmesidir.

Şekil 5.1'de yapısı gösterilen iteratif blok şifreleyicilerde çevrim içerisinde çakışma meydana gelebilir. *F,* çevrim fonksiyonu olmak üzere birkaç parçadan (fonksiyondan) oluşur. Parçalardan bir veya birkaç tanesi doğrusal olmayan fonksiyonlardır.



Şekil 5.1 : İteratif blok şifreleme

Şekil 5.2'de gösterilen *g* fonksiyonu, *F* fonksiyonu içerisinde olsun ve *m* bitlik giriş, *n* bitlik çıkış üretsin ve ayrıca *m>n* olsun.



Şekil 5.2 : Çakışma oluşturan fonksiyon

g fonksiyonunun giriş uzayı kümesi *A* ve çıkış uzayı kümesi *B* olsun. $|A| = 2^m$ ve $|B| \le 2^n$ olur. *m>n* olduğundan |A|>|B| dir. Giriş uzayının eleman sayısı çıkış uzayının eleman sayısından büyük olması bazı farklı giriş çiftleri için aynı çıkışın oluşması sonucunu doğurur. Δ giriş farkları kümesi olsun. $|\Delta| = 2^m$'dir. α çakışma bağıntısı $\alpha = \{(x, y) \mid x \in A, y \in \Delta, g(x) = g(x \oplus y)\} \subset A x \Delta$ şeklinde tanımlansın.

Bağıntılar kümesi olan γ şu şekilde tanımlansın;

$$\gamma = \{\alpha_i \mid \alpha_i \subset \alpha \wedge i \in \{0,1,...,2^m - 1\} \wedge [(x, y) \in \alpha_i \wedge (u, v) \in \alpha_i \Rightarrow y = u] \wedge [(x, y) \in \alpha_i \wedge (u, v) \in \alpha_j \wedge i \neq j \Rightarrow y \neq u] \wedge [\alpha_0 \cup \alpha_1 \cup ... \cup \alpha_{i-1} = \alpha]\}$$

Eşitlik 5.1'de tanımlanan *h,* girişi α_i kümesi olan ve çıkışı *m* bitlik vektör olan bir fonksiyon olsun.

$$h(\alpha_i)[j] = \begin{cases} x[j] & ,\forall (x, y), (u, v) \in \alpha_i, x[j] = u[j] \\ -1 & ,\text{diğer} \end{cases} \qquad (5.1)$$

[*i*] vektörün *i.* biti anlamında kullanılmıştır. *h* fonksiyonu bir giriş farkı ile girişteki hangi bitlerin bulunabileceğini verir. *h* fonksiyonu ile bir çevrim içerisinde çakışma durumunda *g* fonksiyonunun girişinde bulunan bazı bitler öğrenilir. *g* fonksiyonu girişine gelen bitlerden bazıları bilinerek de gizli olan anahtar bitlerinden bazıları bulunabilir.

Çevrim fonksiyonu içerisinde çakışma olması ise yan kanal bilgileri ile elde edilir. *g* fonksiyonu farklı iki giriş için aynı çıkış vereceğinden *g* fonksiyonu çıkışında belli bir komut dizisi boyunca aynı veri işlenir. Bir komut dizisi boyunca aynı verinin işlenmesi kaynaktan çekilen akımın o komut dizisi boyunca aynı olmasına neden olur. Yan kanal bilgisi olarak giriş farkları belli şifrelemeler sırasında çekilen akım incelenerek çakışma durumu sezilir. Fark ve çakışma belli olduğundan dolayı da döngü içerisinde bulunan bazı bitler bulunur ve bu bitler ile de döngü anahtarının bazı bitleri bulunur.

Çoğu blok şifreleme işleminde giriş kümesinin eleman sayısı çıkış kümesinin eleman sayısından fazla olan fonksiyonlar bulunmaz. Çakışma saldırısı bu tip algoritmalara parçalı çakışma kavramı ile uygulanabilir. Parçalı çakışma, iki farklı giriş için fonksiyonun çıkışının bir kısmının aynı olması demektir.

6. ÇAKIŞMA SALDIRISI UYGULAMALARI

Bu bölümde, tezde tanımlanan çakışma saldırısının önceden yapılmış uygulamaları anlatılmaktadır. Tezde yapılan saldırı uygulamaları da bu bölümde yer almaktadır. Bu uygulamalardan bir tanesi DES algoritmasının 1. ve 2. çevrimlerine saldırıdır. Diğer bir uygulama da 3DES algoritmasına saldırının nasıl uygulanabileceği ile ilgilidir.

6.1 Önceden Yapılan Bazı Çalışmalar

Son zamanlarda çakışma saldırısı ile ilgili yapılmış çalışmalar bulunmaktadır.

2003 yılında Schramm ve diğerleri tarafından DES algoritması üzerinde saldırı uygulanmıştır [3]. Saldırı, DES algoritmasındaki s-kutularının girişlerinin 6 bit çıkışlarının ise 4 bit olmasına dayanmaktadır. Tezde DES üzerindeki saldırı detaylı incelenilmiş ve s-kutuları analiz edilmiştir.

Saldırı Schramm ve diğerleri tarafından 2004'te AES [13] algoritmasına uygulanmıştır [5]. AES algoritmasında DES algoritmasında olduğu gibi çakışmaya neden olan fonksiyonlar bulunmamaktadır yani fonksiyonların giriş uzay kümesi ile çıkış uzay kümesi aynıdır. AES için kısmi çakışma uygulanmıştır. Kısmi çakışma çıkışın bir bölümünde meydana gelen çakışmadır.

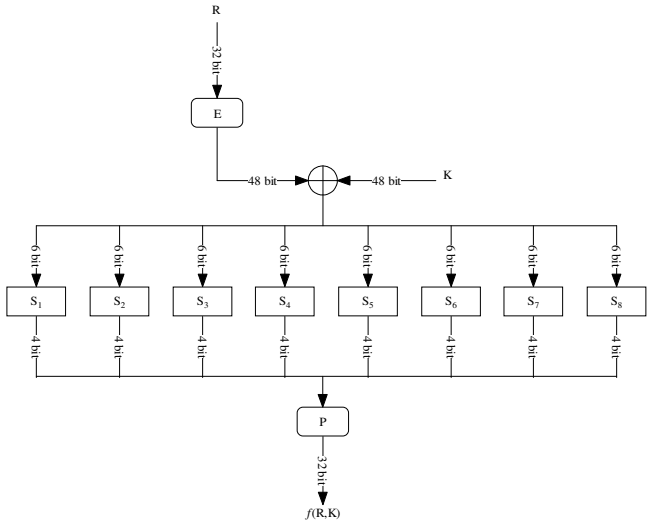
6.2 DES Algoritmasına Saldırı

DES algoritmasının başında ve sonunda bulunan permütasyon işlemleri (IP, IP^{-1}) çakışma saldırısına etki etmemektedir. Saldırının kolay anlaşılabilir olması için IP ve IP^{-1} işlemleri algoritmada yokmuş gibi davranılacaktır.

6.2.1 Çevrim Fonksiyonu

Fesitel yapısında olan DES algoritmasının çevrim fonksiyonu $F(L,R) = (R, L \oplus f(R,K))$ dir. L ve R çevrim girişinin 32'şer bitlik sol ve sağ yarımlarını, K ise 48 bitlik çevrim anahtarını ifade etmektedir. f ise çevrim içerisinde

bulunan fonksiyondur. f fonksiyonu 32 bitlik girdiyi 48 bite genişletir ve çevrim anahtarı ile *yada* işlemine tabi tutar. Çıkan sonuç s-kutularından geçirilir ve sonuca permütasyon uygulanır. Permütasyon çıkışı f fonksiyonunun çıkışıdır. Grafiksel olarak Şekil 6.1'de görülmektedir.

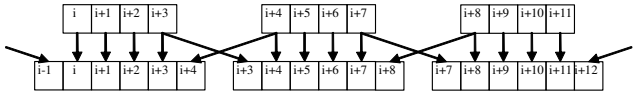


Şekil 6.1: DES f fonksiyonu

f fonksiyonu içerisindeki s-kutularının giriş uzayı çıkış uzayından büyük olduğundan bazı girişler için çakışma meydana gelmektedir. Giriş uzayı 2^6 elemandan oluşurken çıkışı uzayı 2^4 elemandan oluşmaktadır. Burada dengeli bir dağılıp oluşur yani çıkıştaki her bir değere girişten 4 değer karşılık gelir. Böylece bir s-kutusu için $C(4,2) \cdot 2^4 = 96$ farklı giriş çiftleri aynı çıkışa neden olmaktadır.

Çakışma oluşturmak için olabildiğince az s-kutusu aktif olarak kullanılmalı. Aktif olarak kullanılmak, farklı iki girişin değişen bitlerinin s-kutusuna yansıması demektir. Az sayıda aktif s-kutusu kullanılması, çakışma oluşturmak için denenmesi gereken açık metin çiftinin az olmasını sağlar. Çakışma oluşturmak için tek bir s-kutusu kullanılmış olsa, bir s-kutusuna açık metnin 6 biti yansıdığından denenmesi gereken açık metin sayısı en fazla $2^6/2=32$ olur.

DES algoritmasında s-kutularına gelen giriş önce genişletme fonksiyonundan geçtiği için girişte meydana gelen bir değişiklik s-kutu girişlerine aynen yansımayabilir. 32 bitlik $X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{18}, X_{19}, X_{20}, X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}, X_{29}, X_{30}, X_{31}, X_{32}$ girişi, genişletme fonksiyonu sonucunda s-kutularına $X_{32}, X_1, X_2, X_3, X_4, X_5, X_4, X_5, X_6, X_7, X_8, X_9, X_8, X_9, X_{10}, X_{11}, X_{12}, X_{13}, X_{12}, X_{13}, X_{14}, X_{15}, X_{16}, X_{17}, X_{16}, X_{17}, X_{18}, X_{19}, X_{20}, X_{21}, X_{20}, X_{21}, X_{22}, X_{23}, X_{24}, X_{25}, X_{24}, X_{25}, X_{26}, X_{27}, X_{28}, X_{29}, X_{28}, X_{29}, X_{30}, X_{31}, X_{32}, X_1$ olarak yansır. f fonksiyonu girişinin s-kutuları girişlerine yansıması Şekil 6.2 ile görülmektedir.



Şekil 6.2 : f girişinin s-kutularına dağılımı

Bir adet s-kutusu üzerinde çakışma oluşturmak için s-kutusu girişindeki ilk ve son bitlerin değişmemesi gerekir, aksi halde değişiklik diğer s-kutularına yansır. Fakat ilk ve son bitler sabit iken s-kutusu bire-bir fonksiyon gibi davranmaktadır. Bu durum saldırılara karşı s-kutusunun sağlaması gereken ölçütlerden bir tanesidir [1]. Bir s-kutusu Şekil 6.3'te bulunmaktadır.

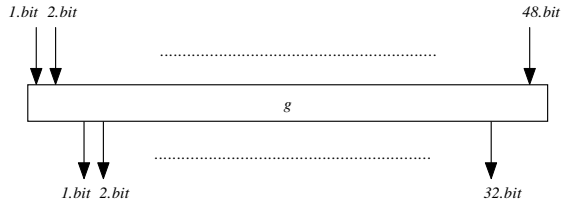
giriş = $g_0g_1g_2g_3g_4g_5$

		$g_1g_2g_3g_4$																
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
g_0g_5	00	0010	1100	0100	0001	0111	1100	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1100	0011	1001	1000	0110
	10	0100	0010	0001	1011	1100	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1100	0100	0101	0011

Şekil 6.3 : DES S_5-kutusu

En az peş peşe 3 aktif s-kutusu kullanılarak çakışma oluşturulabilmektedir [14].

g fonksiyonu s-kutuları olsun. 48 bit giriş almakta ve 32 bit çıkış üretmektedir. g giriş bitlerinin indeksleri soldan 1 ile başlamaktadır. g fonksiyonu Şekil 6.4 de bulunmaktadır.



Şekil 6.4 : DES içerisinde tanımlanan g fonksiyonu

Çevrimdeki f fonksiyonu $f(R,K) = P(g(E(R) \oplus K))$ şeklinde yazılabilir. g fonksiyonu, giriş farklarının 3 s-kutusuna yansıdığı 258 fark için çakışma üretmektedir. 258 farktan 249 tanesi çakışma saldırısında kullanılabilirdir. Kullanılabilir olması, fark kullanılarak g fonksiyonu girişindeki bazı bitlerin bulunabilmesi demektir. Daha genel açıklama Bölüm 5'te yer almaktadır. 249 fark, bu farklar ile g girişinin hangi bitlerinin bulunabileceği ve çakışmanın yaklaşık kaç açık metin çifti denenerek oluşturulabileceği tablo A.1'de bulunmaktadır. Tablo A.1'den fark seçimi iki ölçüte göre yapılır; farkı sağlayan giriş sayısı da fazla olsun ve olabildiğince çok anahtar biti bulunabilir olsun. Farkı sağlayan giriş sayısının fazla olması denenmesi gereken açık metin çiftinin sayısını azaltmaktadır. Fark 3 s-kutusuna uygulanacağı için, bu s-kutularına giren bitlerin açık metindeki karşılıkları olan 14 bit değiştirilir. Eğer bir farkı sağlayan giriş sayısı m ise bir tane çakışma oluşturmak için denenmesi gereken açık metin sayısı yaklaşık $2^{14}/m$'dir.

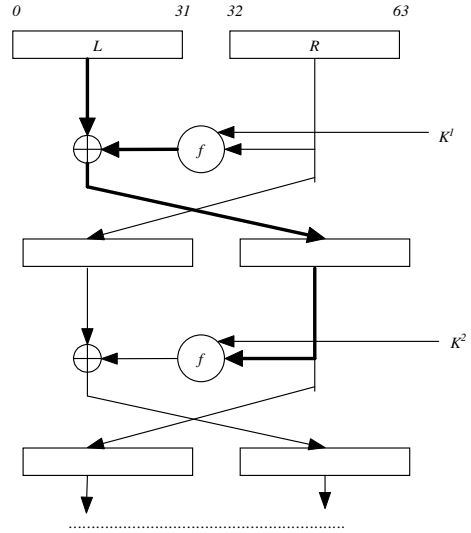
Tablo A.1 deki farklar kullanılarak çakışma ile g girişindeki bitlerden en fazla 39 bit bulunabilmektedir. Bulunabilen bitler 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 25, 26, 27, 28, 29, 30, 31, 32, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 46, 47 ve 48. bitlerdir. Bu bitler Tablo 6.1 deki farklar kullanılarak yaklaşık 1154 açık metin çiftlerinin şifrelenmesi ile bulunabilmektedir.

Tablo 6.1 : Çevrim İçerisindeki 39 Biti Veren Farklar

aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
123	0000111100101001000	5	10,12,14,17,18	448	37
123	0010111100101011000	5	1,2,4,10,12	160	103
345	0000111100101010000	4	26,28,29,30	256	64
567	0000111110101001000	4	36,38,41,42	256	64
567	0010111100101010000	4	28,34,36,25	224	74
678	0001111100101010000	4	42,32,39,40	120	137
678	0001111110101001000	4	44,47,48,32	384	43
812	0011111100101001000	3	6,11,3	320	52
345	0010111100101011000	3	13,14,16	256	64
456	0000111100101001000	3	20,27,28	320	52
781	0011111110101001000	3	37,38,48	192	86
781	0001111110101010000	3	38,39,48	384	43
234	0010111101101010000	2	15,18	384	43
567	0011111101010100000	2	36,26	448	37
234	0000111111101010000	2	8,10	512	32
678	0011111110101010000	2	31,32	320	52
781	0000111101010110000	2	48,2	256	64
234	0001111111101010000	2	7,9	512	32
812	0001111101010010000	2	11,43	360	46
781	0000111100101001000	1	46	576	29

6.2.2 1. Çevrime Saldırı

1. çevrimdeki s-kutularının yani *g* fonksiyonunun girişleri açık metnin sağ yarımı ile çevrim anahtarından oluşmaktadır. Açık metnini girişleri ve çakışma ile de *g* girişlerinin bazı bitleri bilindiğinden dolayı çevrim anahtarının bazı bitleri bulunabilir. Çakışma oluşması ise 2.çevrimdeki çekilen akımın aynı olmasından anlaşılır. Buradaki yaklaşım görsel olarak Şekil 6.5'de bulunmaktadır. Şekil, farklı iki girişin paralel olarak algoritma içerisinde akması şeklinde düşünülmelidir. Koyu oklarla gösterilen kısımlardaki veriler iki farklı giriş için eşit anlamına geliyor. Şifrelenecek olan metin çiftleri (L,R) ve $(L,R\oplus\Delta)$ olsun. K^1 1. döngü anahtarı olmak üzere $f(R,K^1)=f(R\oplus\Delta,K^1)$ olursa ikinci çevrimdeki f fonksiyonuna giren girişler $L\oplus f(R,K^1)$ ve $L\oplus f(R\oplus\Delta,K^1)$ eşit olur. Bu durumda ikinci çevrimdeki f fonksiyonu aynı akımı çeker. Böylece çekilen akımın aynı olmasından çakışma olduğu anlaşılır.



Şekil 6.5 : DES üzerinde çakışma atağı

1. çevrim anahtarı oluşturan ana anahtar bitleri Tablo 6.2'de gösterilmiştir. İlk satır çevrim anahtar bitlerini, ikinci satır ise ana anahtar bitlerini göstermektedir. İlk satırda koyu olarak yazılan bitler, çakışma ile bulunabilen bitleridir.

Tablo 6.2 : 1. Çevrimde Çakışma ile Bulunabilen Anahtar Bitleri

1	**2**	**3**	**4**	5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
10	51	34	60	49	17	33	57	2	9	19	42	3	35	26	25

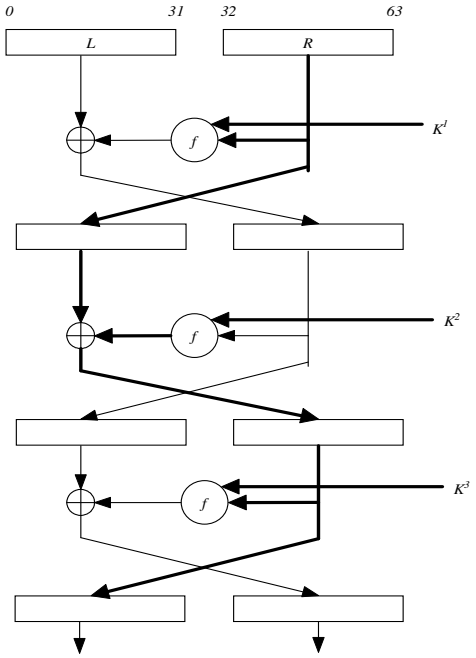
17	**18**	19	**20**	21	22	23	24	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**
44	58	59	1	36	27	18	41	22	28	39	54	37	4	47	30

33	**34**	35	**36**	**37**	**38**	**39**	**40**	**41**	**42**	**43**	**44**	45	**46**	**47**	**48**
5	53	23	29	61	21	38	63	15	20	45	14	13	62	55	31

1. çevrimde oluşturulan çakışma ile anahtarın 39 biti bulunabilmektedir. 56 bitlik anahtarın diğer bitleri de çok kısa bir sürede deneme yanılma ile bulunabilir. Tüm anahtar, 1154 metin çiftinin algoritmanın bulunduğu kart üzerinde şifrelenmesi ve herhangi bir bilgisayar üzerinde 2^{17} şifreleme işlemi ile bulunmuş olunur. Gerekli olan veri ise sadece bir adet bilinen açık-kapalı metin çiftidir.

6.2.3 2. Çevrime Saldırı

Saldırı 1.çevrimdekine benzer olarak ikinci çevrime genişletilebilir. 3. çevrimdeki *f* fonksiyonu nedeniyle çekilen akımların eşit olması ile çakışma olduğu anlaşılır. Saldırıdaki farkların akışı Şekil 6.6'da bulunmaktadır.

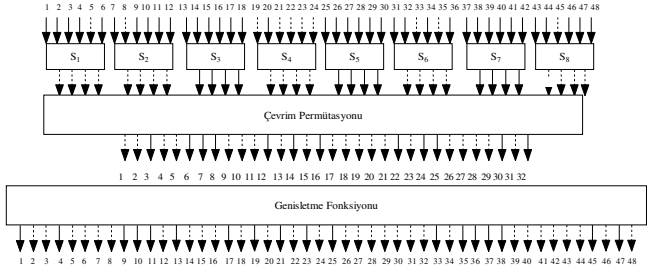


.....

Şekil 6.6 : 2. Çevrime Saldırı

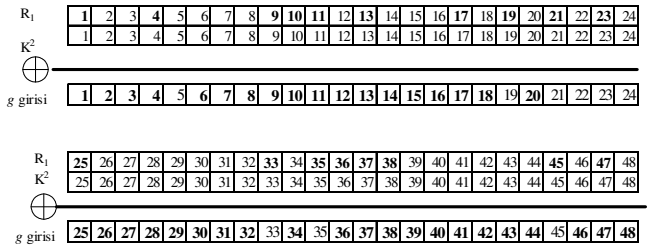
İkinci çevrimde *g* fonksiyonunun girişinde 2. çevrim anahtarı ve açık metin bitleri ile beraber 1. çevrim anahtar bitleri de bulunmaktadır. R_1'in (ikinci çevrim girişinin sağ yarımı) bilinebilen bitleri, 2. çevrimde bulunabilecek anahtar bitlerini etkilemektedir.

1. çevrime yapılan saldırı sonucu elde edilen anahtar bitleri ve bu bitlerin s-kutularına girişi ile 2. çevrime yansıyışı Şekil 6.7'de bulunmaktadır. Kesikli oklar bilinmeyen bitleri ifade etmektedir.



Şekil 6.7 : 1. çevrim sonucu bilenebilen bitler

2. çevrimdeki *g* girişini R_1 ve K^2 (2. çevrim anahtarı) oluşturduğu için 2. çevrim anahtarından R_1 ve *g* girişinin bilinen bitlerine karşılık gelen bitleri bulunabilir. Şekil 6.8 de görüldüğü gibi 2. çevrim anahtarının 1, 4, 9, 10, 11,13, 17, 25, 36, 37, 38 ve 47. bitleri bulunabilir.



Şekil 6.8 : 2. Çevrim Anahtarının Bulunabilen Bitleri

Tablo 6.3'de 2. çevrim anahtarını oluşturan ana anahtar bitleri bulunmaktadır. ilk satır çevrim anahtarı, ikinci satır ise ana anahtar bitleridir. Koyu olanlar bulunabilen çevrim anahtarlarıdır.

Tablo 6.3 : 2. Çevrim Anahtarını oluşturan Anahtar Bitleri

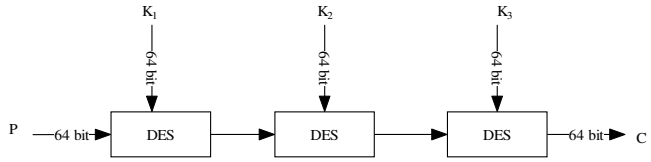
1	2	3	**4**	5	6	7	8	**9**	**10**	**11**	12	**13**	14	15	16	**17**	18	19	20	21	22	23	24
2	43	26	52	41	9	25	49	59	1	11	34	60	27	18	17	36	50	51	58	57	19	10	33

25	26	27	28	29	30	31	32	33	34	35	**36**	**37**	**38**	39	40	41	42	43	44	45	46	**47**	48
14	20	31	46	29	63	39	22	28	45	15	21	53	13	30	55	7	12	37	6	5	54	47	23

2. çevrime saldırı ile de ana anahtarın 2, 52, 59, 1, 11, 60, 36, 14, 21, 53, 13 ve 47. bitleri bulunabilir. Böylece 1. çevrim ile bulunamayan 52, 59, 11, 36, 3. anahtar bitleri de bulunmuş olunur. Çakışma saldırısı ile 56 bitlik anahtarın 44 biti bulunmuş olunur.

6.3 3DES Algoritmasına Saldırı

Şekil 6.9'da yapısı görülen 3DES algoritmasında 3 DES şifreleme işlemi peş peşe uygulanır.



Şekil 6.9 : 3DES

DES algoritmasında çakışma ile bulunamayan anahtar bitleri deneme yanılma ile bulunabiliyordu. 3DES algoritmasında ise K_1 anahtarının tamamı çakışma sağlanarak bulunamamaktadır. Bunun nedeni, ilk DES işlemi çıkışının bilinememesidir. Sadece çakışma K_1 anahtarının en fazla 44 biti bulunabilmektedir. Şifreleme ve şifre çözme işlemi DES algoritmasında aynı olduğundan çakışma saldırısı ile K_3 anahtarının da 44 biti bulunabilir. K_2 anahtarını bulabilmek için K_1 anahtarının 12 biti tahmin edilir ve her tahmin için K_2 anahtarının 44 biti bulunur. K_2 anahtarının da 12 biti tahmin edilir. K_3 anahtarı için de 12 bit tahmin edilir ve 2^{36} tahmin yapılarak deneme yanılma ile tüm anahtarlar bulunmuş olunur.

7. ÇAKIŞMA SALDIRISININ DES ÜZERİNDE GERÇEKLENMESİ

Çakışma saldırısının gerçeklenmesi için güç harcama ölçümü yerine simülasyon kullanılması tercih edilmiştir. Bu nedenle güç harcama simülasyonu hazırlanılmış ve saldırı simülasyon kullanılarak gerçeklenmiştir.

7.1 Güç Harcama Simülasyonu

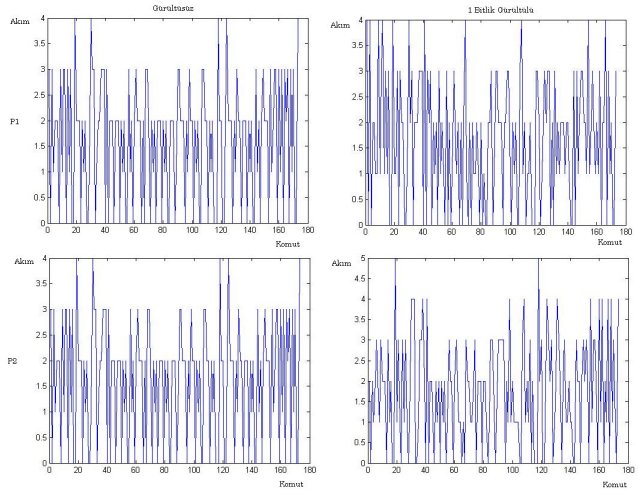
Algoritma içerisinde çakışma olduğu, seçilen farklı iki açık metin için şifreleme yapılırken aynı gücün harcanması ile anlaşılmaktadır. DES algoritmasında 1. çevrimde çakışma olması 2. çevrimde genişletme fonksiyonuna aynı girişlerin gelmesine neden olur. 2. çevrim genişletme fonksiyonu sırasında harcanan güç karşılaştırılarak çakışma olup olmadığı anlaşılabilmektedir. Simülasyonda genişletme fonksiyonunun makine kodu yazılmış ve yazılan makine kodunu yürütecek program hazırlanmıştır. Program ayrıca her makine komutu sırasında çekilen akımı dosyaya kaydetmiştir. Komutun akım çekme miktarı için aşağıdaki model kullanılmıştır.

7.1.1 Güç Harcama Modeli

CMOS transistorlerde 0'dan 1'de geçişler güç harcanmasına neden olmaktadır. Her bir komutta bir saklayıcı değeri değişmektedir. Bir komut sırasında harcanan güç, değeri değişen saklayıcıdaki 0'dan 1'e geçişlerin sayısı olarak alınmıştır. Saklayıcı 8 bitlik olduğundan harcanan güç 0-8 arası bir ayrık bir değer almaktadır. Harcanan güce 1 birimlik gürültü eklenmiştir. 1 birimlik gürültü, çekilen akıma -1, 0 veya 1 eklenmesidir.

7.1.2 Modelin Testi

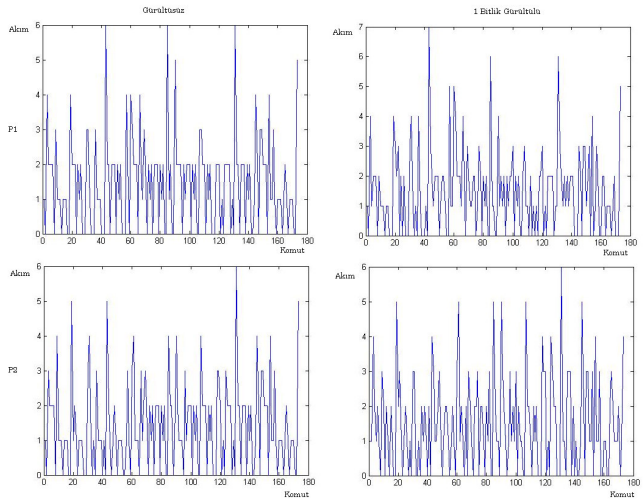
Genişletme fonksiyonuna aynı girişler verildiğinde gürültüsüz ve 1 bitlik gürültü ile harcanan gücün zamana göre değişimi Şekil 7.1'de bulunmaktadır.



Şekil 7.1: Aynı iki giriş için genişletme fonksiyonu güç harcaması

Gürültüsüz durumda ilinti katsayısı 1, 1 birimlik gürültülü durumda ise ilinti katsayısı 0.734 olarak hesaplanmıştır.

Farklı giriş çiftleri için simülasyon çalıştırılmış ve farklı güç harcamanın oluştuğu gözlenmiştir. 1 farklı giriş çifti için çekilen akım Şekil 7.2'de görülebilmektedir. Gürültüsüz durumda ilinti katsayısı 0.926, 1 birimlik gürültülü durumda ise ilinti katsayısı 0.712 olarak hesaplanmıştır.



Şekil 7.2: Farkı iki giriş için genişletme fonksiyonu güç harcaması

7.2 Saldırının Gerçeklenmesi

Tablo 6.1'deki bir fark alınır ve 1. çevrimdeki s-kutularına bu fark gelecek şekilde iki giriş çifti belirlenir ve şifreleme işlemi yapılır, çekilen akım kaydedilir. Bu işlem tabloda belirtilen sayı kadar devam edilir. Çekilen akımlar karşılaştırılır ve çakışmaya neden olan çift belirlenir. Çakışma olduğunda s-kutularının bazı girişlerine gelen bit değerleri belli olduğundan ve açık metinden gelen bitler de belli olduğundan ilgili anahtar bitleri bulunur.

8. SONUÇ

Tezde, klasik kripto analizin ve yan kanal saldırısının bir arada kullanıldığı çakışma saldırısının bazı algoritmalara uygulaması incelenilmiştir. [3]'te anlatılan saldırı detaylı incelendikten sonra aşağıda anlatılan çalışmalar yapılmıştır.

Algoritma bazında anlatılan çakışma saldırısının geniş bir tanımı çıkartılmıştır.

DES algoritmasındaki çevrim fonksiyonu içerisinde bulunan s-kutuları analiz edildikten sonra çakışmaya neden olan girişler ve giriş farkları bulunmuş ve bulunan farklar ile anahtarın hangi bitlerinin kaç deneme sonra bulunacağı hesaplanmış ve çalışmada belirtilmiştir.

DES algoritmasında 1. çevrime saldırının nasıl uygulanacağı, hangi giriş farklarının kullanılarak anahtarın hangi bitlerinin bulunabileceği üzerinde çalışma yapılmış ve tez içerisinde anlatılmıştır.

3DES algoritmasına saldırının nasıl uygulanabileceği analiz edilmiştir. 3DES algoritmasında saldırının daha kolay uygulanabilmesi için çakışma saldırısı ile DES anahtarının daha fazla bitlerinin bulunması ihtiyacı çıkmıştır. Bu ihtiyacı karşılamak için de DES üzerindeki saldırı 2. çevrime genişletilmiş ve hangi farklar kullanılarak anahtarın hangi bitlerinin nasıl bulunabileceği çıkartılmıştır.

8.1 Devamında Yapılabilecek Çalışmalar

Tezde çakışma saldırısı deneyinde gerçek güç harcama ölçümü yapılmamış, bunun yerine güç harcama simülasyonu kullanılmıştır. Saldırı deneyi güç harcama ölçümleri yapılarak gerçeklenebilir.

Güç harcama ölçümlerini yapmak ve analiz etmek kolay gerçekleştirilemediği için çakışma saldırısının yan kanal bilgileri kullanılmadan uygulanabilirliği araştırılabilir.

Çakışma saldırısının başka şifreleme algoritmalarına uygulanması şeklinde de çalışmalar yapılabilir.

KAYNAKLAR

[1] **Coppersmith, D.,** 1994. The Data Encryption Standard (DES) and its Strength Against Attacks. Technical report rc 186131994, IBM Thomas J. Watson Research Center.

[2] **Kocher, P., Jaffe, J., Jun, B.,** 1999. Differential Power Analysis. In Advanges in Cryptology – CRYPTO '99. Springer-Verlag.

[3] **Schramm, K., Wollinger, T., and Paar, C.,** 2003. A New Class of Collision Attacks and Its Application to DES. In: Johansson, T. (ed.) FSE 2003. LNCS, vol 2887, pp. 206-222. Springer, Heidelberg.

[4] **NIST,** 2004. Recommendation fort the Triple Data Encryption Algorithm (TDEA) Block Cipher. Special Publication 800-67.

[5] **Schramm, K., Leander, G., Fekle, P., and Paar, C.,** 2004. A Collision-Attact on AES: Combining Side Channel- and Differential-Attack. In: Joye, M., Quisquarter, J.-J. (eds.) CHES 2004. LNCS, vol. 3156, pp. 163-175. Springer, Heidelberg.

[6] **Stinson, D., R.,** 2002. Cryptography Theory and Practice. CRC Press. New York.

[7] **Menezes, A., J., Oorschot, P., C., Vanstone, S., A.,** 1996. Handbook of Applied Cryptography. CRC Press.

[8] **Sinkov, A.,** 1966. Elementary Cryptanalysis: A Mathematical Approach. Mathematical Association of America.

[9] **Shannon, C., E.,** 1949. Communication theory and secrecy systems. Bell Systems Technical Journal, 28, 656-715.

[10] **Biham, E., Shamir, A.,** 1990. Differential Cryptanalysis of the Data Encryption Standard. Advances in Cryptology – CRYPTO'90. Springer-Verlag 2-21.

[11] **Acıiçmez, O., Schindler, W., Koç, Ç., K.,** 2007. Cache Based Remote Timing Attack ont the AES.

[12] **Kocher, P., Jaffe, J., Jun, B.,** 1998. Introduction to Differential Power Analysis and Related Attacks.

[13] **Daemen, J., Rijmen, V., 1998**. AES proposal: Rijndael. In AES Round 1 Technical Evaluation CD-1: Documentation. NIST.

[14] **Davio, M., Desmedt, Y., and Quisquarter, J.,** 1984. Propagation Characteristics of the DES. In Advances in Cryptology – CRYPTO '84, pages 62-74. Springer-Verlag.

EKLER

EK A.1 : DES 3lü s-kutusu fark tablosu

EK A.1

Tablo A.1 : DES 3lü s-kutusu fark tablosu

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
1	123	000111 110010 100100	8	1,2,3,10,12,14,17,18	64	256
2	123	000011 111110 101100	1	12	560	30
3	123	001011 110110 100100	7	1,2,4,12,14,17,18	64	256
4	123	001011 111010 100100	7	18,1,2,4,12,14,17	32	512
5	123	001011 110010 101000	7	1,2,4,10,12,14,18	96	171
6	123	000111 111110 100100	7	18,1,2,3,12,14,17	32	512
7	123	000111 110010 101000	7	18,1,2,3,10,12,14	96	171
8	123	000111 110110 100100	7	1,2,3,12,14,17,18	64	256
9	123	000111 111010 100100	7	18,1,2,3,12,14,17	32	512
10	123	001111 110010 100100	7	18,1,2,10,12,14,17	64	256
11	123	001011 111110 100100	7	1,2,4,12,14,17,18	32	512
12	123	001011 110110 101000	6	1,2,4,12,14,18	96	171
13	123	001111 110110 100100	6	18,1,2,12,14,17	64	256
14	123	001111 111110 100100	6	1,2,12,14,17,18	32	512
15	123	000111 111110 101000	6	1,2,3,12,14,18	48	342
16	123	001111 110010 101000	6	1,2,10,12,14,18	96	171
17	123	001011 111010 101000	6	18,1,2,4,12,14	48	342
18	123	001011 111110 101000	6	18,1,2,4,12,14	48	342
19	123	000111 111010 101000	6	1,2,3,12,14,18	48	342
20	123	001111 111010 100100	6	1,2,12,14,17,18	32	512
21	123	000111 110110 101000	6	18,1,2,3,12,14	96	171
22	123	000111 110010 101100	5	1,2,3,10,12	160	103
23	123	001111 111010 101000	5	1,2,12,14,18	48	342
24	123	000011 110010 100100	5	10,12,14,17,18	448	37
25	123	001011 110010 101100	5	1,2,4,10,12	160	103
26	123	001111 110110 101000	5	1,2,12,14,18	96	171
27	123	001111 111110 101000	5	1,2,12,14,18	48	342
28	123	000011 111010 100100	4	18,12,14,17	224	74
29	123	000111 110110 101100	4	1,2,3,12	160	103
30	123	000111 111110 101100	4	1,2,3,12	80	205
31	123	000011 110110 100100	4	18,12,14,17	448	37
32	123	001011 110110 101100	4	1,2,4,12	160	103
33	123	001111 110010 101100	4	1,2,10,12	160	103
34	123	001011 111110 101100	4	1,2,4,12	80	205
35	123	000011 110010 101000	4	10,12,14,18	672	25
36	123	001011 111010 101100	4	1,2,4,12	80	205
37	123	000111 111010 101100	4	1,2,3,12	80	205
38	123	000011 111110 100100	4	18,12,14,17	224	74
39	123	000011 110110 101000	3	18,12,14	672	25
40	123	001111 111010 101100	3	1,2,12	80	205

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
41	123	001111 110110 101100	3	1,2,12	160	103
42	123	001111 111110 101100	3	1,2,12	80	205
43	123	000011 111010 101000	3	18,12,14	336	49
44	123	000011 111110 101000	3	18,12,14	336	49
45	123	000011 110010 101100	2	10,12	1120	15
46	123	000011 110110 101100	1	12	1120	15
47	123	000011 111010 101100	1	12	560	30
48	123	001011 110010 100100	8	18,1,2,4,10,12,14,17	64	256
49	234	000011 110010 101100	6	8,10,15,16,18,20	64	256
50	234	000111 110010 101100	6	7,9,15,16,18,20	64	256
51	234	000111 110110 101100	5	7,9,15,18,20	128	128
52	234	000011 110110 101100	5	8,10,15,18,20	128	128
53	234	000111 110010 101000	5	7,9,15,16,18	128	128
54	234	000011 110010 101000	5	8,10,15,16,18	128	128
55	234	000011 110110 101000	4	8,10,15,18	256	64
56	234	001011 110010 101100	4	15,16,18,20	96	171
57	234	000111 110110 101000	4	7,9,15,18	256	64
58	234	000011 111010 101100	3	8,10,20	192	86
59	234	001011 110010 101000	3	15,16,18	192	86
60	234	001011 110110 101100	3	15,18,20	192	86
61	234	000111 111110 101100	3	7,9,20	256	64
62	234	000011 111110 101100	3	8,10,20	256	64
63	234	000111 111010 101100	3	7,9,20	192	86
64	234	001011 110110 101000	2	15,18	384	43
65	234	000011 111110 101000	2	8,10	512	32
66	234	000011 111010 101000	2	8,10	384	43
67	234	000111 111010 101000	2	7,9	384	43
68	234	000111 111110 101000	2	7,9	512	32
69	234	001011 111110 101100	1	20	384	43
70	234	001011 111010 101100	1	20	288	57
71	345	001011 110010 101000	7	13,14,16,26,28,29,30	64	256
72	345	000111 110010 101000	7	13,14,15,26,28,29,30	64	256
73	345	001111 110010 101000	5	13,26,28,29,30	128	128
74	345	000011 110010 101000	4	26,28,29,30	256	64
75	345	000111 110010 100100	4	13,14,15,30	192	86
76	345	001111 110010 101100	1	13	512	32
77	345	001011 110010 101100	3	13,14,16	256	64
78	345	000111 110010 101100	3	13,14,15	256	64
79	345	001111 110010 100100	2	13,30	384	43
80	345	000011 110010 100100	1	30	768	22

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
81	345	001011 110010 100100	4	13,14,16,30	192	86
82	456	000111 110010 101100	4	20,27,28,32	64	256
83	456	000011 111110 100100	1	20	320	52
84	456	000011 110010 101100	4	20,27,28,32	128	128
85	456	000111 110010 101000	4	20,27,28,34	96	171
86	456	000111 110110 101100	3	20,30,32	64	256
87	456	000011 110110 101000	3	20,30,34	192	86
88	456	000011 110110 101100	3	20,30,32	128	128
89	456	000011 110010 100100	3	20,27,28	320	52
90	456	000111 110010 100100	3	20,27,28	160	103
91	456	000111 110110 101000	3	20,30,34	96	171
92	456	000111 111110 101100	2	20,32	64	256
93	456	000011 111110 101000	2	20,34	192	86
94	456	000111 111110 101000	2	20,34	96	171
95	456	000111 110110 100100	2	20,30	160	103
96	456	000011 110110 100100	2	20,30	320	52
97	456	000011 111110 101100	2	20,32	128	128
98	456	000111 111110 100100	1	20	160	103
99	456	000011 110010 101000	4	20,27,28,34	192	86
100	567	000111 110010 100100	8	27,34,36,38,41,42,25,26	32	512
101	567	000011 111110 101000	1	36	448	37
102	567	000111 111110 100100	7	27,36,38,41,42,25,26	32	512
103	567	000111 111010 101100	7	27,36,38,41,42,25,26	32	512
104	567	001011 110010 101100	7	28,34,36,38,41,42,25	32	512
105	567	001011 110010 100100	7	28,34,36,38,41,42,25	64	256
106	567	000111 111110 101100	7	36,38,41,42,25,26,27	16	1024
107	567	000111 111010 100100	7	36,38,41,42,25,26,27	64	256
108	567	001011 111110 101100	6	28,36,38,41,42,25	32	512
109	567	001111 110010 101100	6	34,36,38,41,42,26	32	512
110	567	001011 111010 100100	6	28,36,38,41,42,25	128	128
111	567	001011 111010 101100	6	28,36,38,41,42,25	64	256
112	567	001111 110010 100100	6	34,36,38,41,42,26	64	256
113	567	001011 111110 100100	6	28,36,38,41,42,25	64	256
114	567	000011 110010 100100	5	34,36,38,41,42	128	128
115	567	001111 111110 101100	5	36,38,41,42,26	32	512
116	567	001111 111110 100100	5	36,38,41,42,26	64	256
117	567	000111 110010 101000	5	34,36,25,26,27	112	147
118	567	000011 110010 101100	5	34,36,38,41,42	64	256
119	567	001111 111010 100100	5	36,38,41,42,26	128	128
120	567	001111 111010 101100	5	36,38,41,42,26	64	256

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
121	567	000111 111110 101000	4	27,36,25,26	112	147
122	567	000011 111010 100100	4	36,38,41,42	256	64
123	567	000011 111110 100100	4	36,38,41,42	128	128
124	567	001011 110010 101000	4	28,34,36,25	224	74
125	567	000111 111010 101000	4	36,25,26,27	224	74
126	567	000011 111110 101100	4	36,38,41,42	64	256
127	567	000011 111010 101100	4	36,38,41,42	128	128
128	567	001011 111010 101000	3	28,36,25	448	37
129	567	001111 110010 101000	3	34,36,26	224	74
130	567	001011 111110 101000	3	28,36,25	224	74
131	567	000011 110010 101000	2	34,36	448	37
132	567	001111 111010 101000	2	36,26	448	37
133	567	001111 111110 101000	2	36,26	224	74
134	567	000011 111010 101000	1	36	896	19
135	567	000111 110010 101100	8	34,36,38,41,42,25,26,27	16	1024
136	678	001111 110010 100100	8	44,47,48,31,32,39,40,42	16	1024
137	678	000111 111010 101000	1	32	960	18
138	678	001011 110010 100100	7	44,47,48,32,39,40,42	32	512
139	678	001011 110010 101100	7	44,47,48,32,39,40,42	16	1024
140	678	001111 110110 101100	7	44,47,48,31,32,39,42	8	2048
141	678	000111 110010 100100	7	42,44,47,48,32,39,40	48	342
142	678	001111 110110 100100	7	44,47,48,31,32,39,42	16	1024
143	678	000111 110010 101100	7	44,47,48,32,39,40,42	24	683
144	678	001011 110110 100100	6	42,44,47,48,32,39	32	512
145	678	001011 110110 101100	6	42,44,47,48,32,39	16	1024
146	678	000111 110110 101100	6	44,47,48,32,39,42	24	683
147	678	000111 110110 100100	6	42,44,47,48,32,39	48	342
148	678	001111 111110 101100	5	44,47,48,31,32	16	1024
149	678	001111 110010 101000	5	31,32,39,40,42	40	410
150	678	001111 111110 100100	5	44,47,48,31,32	32	512
151	678	001111 111010 100100	5	44,47,48,31,32	128	128
152	678	001111 111010 101100	5	44,47,48,31,32	64	256
153	678	001011 111110 100100	4	44,47,48,32	64	256
154	678	001111 110110 101000	4	31,32,39,42	40	410
155	678	001011 110010 101000	4	32,39,40,42	80	205
156	678	000111 111110 100100	4	44,47,48,32	96	171
157	678	001011 111010 100100	4	44,47,48,32	256	64
158	678	000111 111110 101100	4	44,47,48,32	48	342
159	678	000111 110010 101000	4	42,32,39,40	120	137
160	678	000111 111010 100100	4	44,47,48,32	384	43

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
161	678	001011 111010 101100	4	44,47,48,32	128	128
162	678	000111 111010 101100	4	44,47,48,32	192	86
163	678	001011 111110 101100	4	44,47,48,32	32	512
164	678	000111 110110 101000	3	32,39,42	120	137
165	678	001011 110110 101000	3	42,32,39	80	205
166	678	001111 111110 101000	2	31,32	80	205
167	678	001111 111010 101000	2	31,32	320	52
168	678	000111 111110 101000	1	32	240	69
169	678	001011 111010 101000	1	32	640	26
170	678	001011 111110 101000	1	32	160	103
171	678	001111 110010 101100	8	44,47,48,31,32,39,40,42	8	2048
172	781	001111 110010 101100	4	37,38,46,2	48	342
173	781	000011 111010 101000	1	48	768	22
174	781	000111 111010 101100	4	38,39,48,2	128	128
175	781	000111 110010 101100	4	38,39,46,2	96	171
176	781	001111 110110 101100	4	37,38,48,2	32	512
177	781	001111 111010 101100	4	37,38,48,2	64	256
178	781	000111 110010 101000	3	38,39,46	288	57
179	781	000111 110110 101000	3	38,39,48	192	86
180	781	000111 110110 100100	3	38,39,48	192	86
181	781	001111 110110 101000	3	37,38,48	96	171
182	781	001111 110110 100100	3	37,38,48	96	171
183	781	000111 111110 101100	3	38,39,2	96	171
184	781	001111 111010 101000	3	37,38,48	192	86
185	781	001111 110010 100100	3	37,38,46	144	114
186	781	000111 111010 100100	3	38,39,48	384	43
187	781	001111 111110 101100	3	37,38,2	48	342
188	781	001111 110010 101000	3	37,38,46	144	114
189	781	001111 111010 100100	3	37,38,48	192	86
190	781	000111 110010 100100	3	38,39,46	288	57
191	781	000111 111010 101000	3	38,39,48	384	43
192	781	000111 111110 101000	2	38,39	288	57
193	781	000111 111110 100100	2	38,39	288	57
194	781	000011 110110 101100	2	48,2	128	128
195	781	001111 111110 100100	2	37,38	144	114
196	781	000011 110010 101100	2	46,2	192	86
197	781	001111 111110 101000	2	37,38	144	114
198	781	000011 111010 101100	2	48,2	256	64
199	781	000011 111010 100100	1	48	768	22
200	781	000011 110010 100100	1	46	576	29

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
201	781	000011 111110 101100	1	2	192	86
202	781	000011 110110 101000	1	48	384	43
203	781	000011 110010 101000	1	46	576	29
204	781	000011 110110 100100	1	48	384	43
205	781	000111 110110 101100	4	38,39,48,2	64	256
206	812	001011 110010 100100	6	3,6,11,43,44,46	80	205
207	812	000111 111010 101100	1	43	360	46
208	812	001011 110010 101000	6	3,6,11,43,44,46	64	256
209	812	001011 110110 101000	6	3,6,11,43,44,46	32	512
210	812	001011 110010 101100	5	3,6,43,44,46	80	205
211	812	001011 110110 101100	5	6,43,44,46,3	40	410
212	812	000111 110010 100100	4	6,11,43,3	240	69
213	812	000111 110010 101000	4	3,6,11,43	192	86
214	812	000111 110110 101000	4	3,6,11,43	96	171
215	812	001011 111010 101000	4	11,43,44,46	96	171
216	812	000111 110110 100100	4	3,6,11,43	120	137
217	812	001011 111110 100100	4	11,43,44,46	80	205
218	812	001011 111110 101000	4	11,43,44,46	64	256
219	812	001011 111010 100100	4	11,43,44,46	120	137
220	812	000111 110010 101100	3	3,6,43	240	69
221	812	000011 110110 101000	3	3,6,11	96	171
222	812	001111 110010 101000	3	3,6,11	256	64
223	812	001111 110010 100100	3	6,11,3	320	52
224	812	001111 110110 101000	3	3,6,11	128	128
225	812	000111 110110 101100	3	3,6,43	120	137
226	812	000011 110110 100100	3	3,6,11	120	137
227	812	000011 110010 101000	3	3,6,11	192	86
228	812	001011 111110 101100	3	43,44,46	80	205
229	812	001111 110110 100100	3	3,6,11	160	103
230	812	000011 110010 100100	3	6,11,3	240	69
231	812	001011 111010 101100	3	43,44,46	120	137
232	812	001111 110010 101100	2	3,6	320	52
233	812	000111 111110 100100	2	11,43	240	69
234	812	000011 110010 101100	2	6,3	240	69
235	812	001111 110110 101100	2	6,3	160	103
236	812	000011 110110 101100	2	3,6	120	137
237	812	000111 111010 101000	2	11,43	288	57
238	812	000111 111110 101000	2	11,43	192	86
239	812	000111 111010 100100	2	11,43	360	46
240	812	001111 111010 100100	1	11	480	35

fark numarası	aktif s-kutuları	s-kutularına giren farklar	bulunabilen *g* giriş bitleri sayısı	bulunabilen *g* giriş bitleri	farkı sağlayan giriş sayısı	bir cakışma olması için denenmesi gereken açık metin çifti sayısı
241	812	000011 111010 101000	1	11	288	57
242	812	000011 111010 100100	1	11	360	46
243	812	001111 111110 101000	1	11	256	64
244	812	001111 111110 100100	1	11	320	52
245	812	001111 111010 101000	1	11	384	43
246	812	000111 111110 101100	1	43	240	69
247	812	000011 111110 101000	1	11	192	86
248	812	000011 111110 100100	1	11	240	69
249	812	001011 110110 100100	6	6,11,43,44,46,3	40	410

ÖZGEÇMİŞ

Ad Soyad: Ferhat Karakoç

Doğum Yeri ve Tarihi: Ayancık/SİNOP , 23.07.1983

Adres: Seyrantepe M. Murat S. No:13/1 Kağıthane/İSTANBUL

Eğitim: Yüksek Lisans, İTÜ, Bilgisayar Müh., 2005 -

Lisans , İTÜ Bilgisayar Müh. (Bölüm ikincisi), 2000-2005

Lise, Samsun Karşıyaka Lisesi, 1997-2000

İş Tecrübesi: TÜBİTAK, UEKAE, Ekim 2007 -

TÜRKCELL, Temmuz 2005- Ekim 2007