**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**(YÜKSEK LİSANS TEZİ)**

# KRİPTOLOJİ SİSTEMLERİ VE UYGULAMALARI ÜZERİNE

**Shahin NASIBOV**

**Tez Danışmanı: Yard. Doç. Dr. Arif GÜRSOY**

**Matematik Anabilim Dalı**

**Bilim Dalı Kodu : 619.03.03.**
**Sunuş Tarihi : 17.06.2015**

**Bornova-İZMİR**

**2015**

Shahin NASIBOV tarafından Yüksek Lisans tezi olarak sunulan “Kriptoloji Sistemleri ve Uygulamaları Üzerine” başlıklı bu çalışma E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliği ile E.Ü. Fen Bilimleri Enstitüsü Eğitim ve Öğretim Yönergesi’nin ilgili hükümleri uyarınca tarafımızdan değerlendirilerek savunmaya değer bulunmuş ve 17.06.2015 tarihinde yapılan tez savunma sınavında aday oybirliği/oyçokluğu ile başarılı bulunmuştur.

| | **Jüri Üyeleri:** | **İmza** |
|---|---|---|
| **Jüri Başkanı** | **: .....** | **.....** |
| **Raportör Üye** | **: .....** | **.....** |
| **Üye** | **: .....** | **.....** |

**EGE ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ**

**ETİK KURALLARA UYGUNLUK BEYANI**

E.Ü. Lisansüstü Eğitim ve Öğretim Yönetmeliğinin ilgili hükümleri uyarınca Yüksek Lisans Tezi olarak sunduğum "Kriptoloji Sistemleri ve Uygulamaları Üzerine" başlıklı bu tezin kendi çalışmam olduğunu, sunduğum tüm sonuç, doküman, bilgi ve belgeleri bizzat ve bu tez çalışması kapsamında elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara atıf yaptığımı ve bunları kaynaklar listesinde usulüne uygun olarak verdiğimi, tez çalışması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını, bu tezin herhangi bir bölümünü bu üniversite veya diğer bir üniversitede başka bir tez çalışması içinde sunmadığımı, bu tezin planlanmasından yazımına kadar bütün safhalarda bilimsel etik kurallarına uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul edeceğimi beyan ederim.

17 / 06 / 2015

İmzası

Shahin NASIBOV

# ÖZET

## KRİPTOLOJİ SİSTEMLERİ VE UYGULAMALARI ÜZERİNE

NASIBOV, Shahin

Yüksek Lisans Tezi, Matematik Anabilim Dalı
Tez Danışmanı: Yard. Doç. Dr. Arif GÜRSOY
Haziran 2015, 37 sayfa

Tez konusunun amacı, günlük hayatta en çok tercih edilen açık anahtarlı kriptosistem olan RSA'nın parametre seçiminde dikkat edilecek konular hakkında detaylı bir araştırma yapmaktır. RSA sisteminin güvenliği, iki büyük asal sayının çarpımıyla oluşturulan yeni sayının çarpanlarına ayrılmasının zorluğuna bağlıdır. Başka bir ifadeyle, yeni oluşan sayı, oluşabilecek matematiksel çözüm yöntemlerine karşı direnç gösterebilmelidir. Temel olarak gösterilebilecek etkin matematiksel çözüm yöntemlerinden biri Fermat Çarpanlara Ayırma Yöntemi'dir. Varolan diğer çarpanlara ayırma yöntemleri (örneğin number field sieve) makul bir süre içinde verimli bir şekilde çalışmamaktadır. Fermat Çarpanlara Ayırma Yöntemi'ne karşı dirençli asal sınıfları veya ikilileri oldukça büyük bir küme oluşturmaktadır. Bu küme içerisinde yer alan ve bilinen bazı ataklara da dirençli bir bileşik sayıyı oluşturmak için belli araştırmalar yapılmıştır. Bu tez kapsamında, p ve q asalları seçilirken bu asal sayılar arasındaki uzaklık ölçümü tanımlanarak, bunun için uygulamalı olarak çeşitli sınırlar belirlenmiştir.

**Anahtar sözcükler:** RSA Kriptosistemi, Kriptoanaliz, Çarpanlara ayırma, Fermat Yöntemi,

# ABSTRACT

## ABOUT CRYPTOGRAPHY SYSTEMS AND APPLICATIONS

NASIBOV, Shahin

MSc in Mathematics
Supervisor: Assist. Prof. Dr. Arif GÜRSOY
June 2015, 37 pages

The aim of this thesis is to define new metrics for the parameter selection of RSA the most used public key cryptosystem in daily life. The security of RSA depends on the factorization of multiplication of specially selected two large primes. Moreover, this composite number shouldn't be factorized in a reasonable time. Fermat factorization idea is known as one of the most efficient method to factorize a nonsecure composite number in terms of RSA. Other successfull factorization methods such as number field sieve cannot result in a reasonable time due to the exponential complexity. The number of prime pairs or sets for RSA is very large and all of them cannot be computed for relatively large sizes. There have been so many studies on the selection of primes for RSA resistant to well-known attacks. In this thesis, while the selection process of primes p and q, the distance between these primes is defined and some bounds are provided with experimental results This distance criterion helps us to generate secure RSA prime pairs in terms of Fermat factorization method.

**Keywords:** RSA cryptosystem, cryptanalysis, integer factorization, Fermat's factorization method

## TEŞEKKÜR

Bu tezin gerçekleştirilmesinde, çalışmam boyunca benden yardımını esirgemeyen Danışman Hocam Sayın Yrd. Doç. Dr.Arif Gürsoy‘a ve bugüne kadar hep yanımda olan ve beni destekleyen Prof. Dr. Urfat NURİYEV`e teşekkürü bir borç bilirim.

# İÇİNDEKİLER

Sayfa

# İÇİNDEKİLER (Devam)

Sayfa

# ŞEKİLLER DİZİNİ

Şekil Sayfa

# ÇİZELGELER DİZİNİ

# 1 GİRİŞ

Yazının icadından günümüze, insanlar iletişimde gizliliği her zaman ön planda tutmuşlardır. Gizli haberleşmenin ortaya çıktığı ilk yıllarda şifreleme kâğıt kalemle yapılan tekniklere dayanmaktadır. Şifreleme teknikleri, özellikle günümüz insanının kişisel bilgilerini saklaması, depolaması, koruması ve iletmesi için kullandığı en önemli araçtır (Çimen C., Akleylek S. ve Akyıldız E., 2002).

Bu tezin ikinci bölümünde, kriptoloji, kriptoğrafi ve kriptanaliz kavramları açıklanmış, bir şifreleme yönteminin sağlaması gereken özelliklere ve bazı güvenlik kavramlarına değinilmiştir.

Üçüncü bölümde, basit şifreleme yöntemlerinden bahsedilmiş, sırasıyla mono alfabetik şifreleme, poli alfabetik şifreleme ve tek kullanımlık karakter dizisi şifreleme yöntemleri incelenmiştir.

Dördüncü bölümde, ilk olarak güvenli şifreleme yöntemlerinden simetrik şifreleme ele alınmıştır. Simetrik şifrelemede, anahtar yönetimi, blok şifreleme algoritmaları ve DES algoritması incelenmiştir. Ardından, asimetrik şifreleme yöntemlerinden RSA şifreleme sistemi, RSA sisteminin anahtar üretimi, RSA sisteminde şifreleme ve deşifreleme işlemi, RSA'nın güvenliği ve Fermat Çarpanlara Ayırma Yöntemi anlatılmıştır.

Beşinci bölümde, güvenli şifreleme sistemleri olan simetrik şifreleme ve asimetrik şifreleme yöntemlerinin avantajları, dezavantajları, güçlü ve zayıf yönleri karşılaştırılmıştır.

Altıncı bölümde, RSA'nın Fermat Çarpanlara Ayırma Yöntemi ve buna benzer diğer yöntemler karşısında güvenliğini artırmak için Dirençli Asal Sayı Seçim Yöntemi sunulmuş ve bu yönteme dayanan ve C++ programlama diliyle kodlanmış bir algoritma verilmiştir.

Son olarak, yedinci bölümde, tezde elde edilmiş sonuçlar tartışılmıştır.

# 2 KRİPTOLOJİ

*Kriptoloji* (*şifreleme bilimi*), iletişim kurmak isteyen iki veya daha fazla tarafın veri alışverişini güvenli bir şekilde yapmasını sağlayan ve matematiksel temellere dayanan tekniklerin ve uygulamaların bütünüdür. Kriptoloji, günümüzde Bilgisayar Bilimleri, Elektronik, Matematik, Optik gibi disiplinleri kullanan özelleşmiş, disiplinler arası bir bilim dalıdır. Kriptolojinin, kriptografi ve kriptoanaliz olarak temel iki alt dalı vardır (Stinson D.R., 2002).

*Kriptografi*, açık metin/belgeleri şifrelemek ve şifrelenmiş metin/belgelerin şifresini çözmek (deşifre) için kullanılan yöntemlerin bütünüdür (Stinson D.R., 2002).

*Kriptoanaliz*, kriptografik sistemlerle uygulanan mekanizmaları inceler ve bu mekanizmaların şifresini çözmeye çalışır. Mevcut bir şifreleme sistemini inceleyerek, zayıf ve kuvvetli yönlerini ortaya koymak için kullanıldığından dolayı çok önem taşımaktadır (Wagstaff S.S., 2002).

Bir şifreleme yönteminin sağlaması beklenen özellikler (Schneier B., 1996):

- Şifreleme ve şifre çözme işleminin zorluğu güvenlikle aynı oranda yüksek olmalıdır.
- Çok önemli olmayan bir bilginin şifrelenmesi için bilginin kendisinden daha fazla işgücü ve zaman harcanması verimli olmayacaktır.
- Anahtar seçimi ve şifreleme algoritmasında istisnai durumlar olmamalıdır. Şifreleme yöntemi her türlü bilgi için aynı şekilde çalışmalıdır.
- Şifrelemede oluşabilecek hatalar/kayıplar mesajın genelini bozmamalıdır.
- Şifreleme algoritmasının ürettiği şifreli mesaj ile açık mesaj arasında ilişki kurulması çok zor olmalıdır. Açık mesajdaki karakterler şifreli mesajın içinde olabildiğince dağıtılmalıdır.

Bir bilginin güvenli bir şekilde iletilebileceğinden ya da güvenli bir şekilde elde edildiğinden bahsedilebilmek için kullanılan iletişim sistemlerinin sahip olması beklenen bazı güvenlik kavramları aşağıdaki gibi sıralanabilir (Pomerance, C., Goldwasser, S., Lagarias, J.C., Arjen K.L., McCurley, K.S., Odlyzko A.M. 1989).

- Gizlilik: Bilgiye sadece yetkisi olanların erişebilmesidir.
- Kimlik denetimi: Gönderilen bir mesajın göndericisinin doğru kişi olduğunun garantisidir.
- Bütünlük: Gönderilen mesajın gönderildiği gibi olduğu, hiçbir değişiklik, ekleme, düzenleme yapılmadığı garantisidir.
- Reddedilmezlik: Göndericinin gönderdiği mesajı inkar edememesidir.
- Erişim kontrolü: İzinsiz kişi veya uygulamaların erişmemesi gereken kaynaklara erişemeyeceklerinin garantisidir.

# 3 BASİT ŞİFRELEME YÖNTEMLERİ

Basit şifreleme yöntemleri, günümüzün bilgisayarlarıyla kolayca çözülebilen şifreleme yöntemleri olarak da tanımlanabilir. Basit şifreleme yöntemlerini Mono Alfabetik Şifreleme, Poli Alfabetik Şifreleme ve Tek Kullanımlık Karakter Dizisi olarak üç ana başlık altına toplayabiliriz.

## 3.1 Mono Alfabetik Şifreleme

Harflerin konumlarının alfabede kaydırılarak değiştirilmesi sonucu oluşan şifreleme yöntemlerine mono alfabetik şifreleme denir. Mono alfabetik şifreleme yöntemleri bilgisayar vasıtasıyla kısa sürede kırılabilir. Bu yöntemde, şifrelemede kullanılan dildeki harflerin yeri değiştirilir fakat harflerin frekansı değişmez. Örnek olarak, Türkçe dilinde en çok kullanılan "a" harfi tablo yöntemi kullanılarak "c" harfi ile yer değiştirildiğinde şifreli metinde en çok tekrar eden harf "c" olacaktır. Buna göre diller arasında denemeler yapılarak, şifrelemenin hangi dilde yapıldığı tespit edilebilir ve mesaj deşifre edilebilir. (Arjen K. L., 2000)

### 3.1.1 Sezar Şifresi

Sezar şifresi yöntemi, bir mono alfabetik şifrelemedir. Bu yöntemde harflerin yeri değiştirilir. Örneğin, şifrelenecek metindeki harfler kullanılan alfabeye bağlı olarak 3 harf kaydırılarak yer değiştirildiğinde aşağıdaki şekilde şifreleme ve deşifreleme yapılabilir. (Çimen C., Akleylek S. and Akyıldız E., 2002)

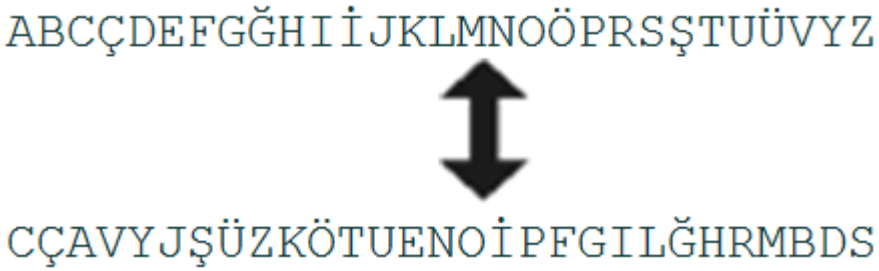
Sezar Şifresi, $c_i = E(p_i) = p_i + 3 \ (mod\ 29)$ olsun. Buna göre verilen bir açık mesaja göre şifrelenmiş mesaj şu şekildedir:

Açık Mesaj : Gizli Bilgi

Şifreli Mesaj : Ilcol Dloıl

### 3.1.2 Tablo Yöntemi

Tablo yöntemi, Sezar şifresi yönteminin gelişmiş versiyonudur. Bu yöntemde, alfabedeki her harf rastgele başka bir harfle yer değiştirilir.




**Şekil 3.1** Tablo şifreleme yöntemi için örnek bir dizi

### 3.2 Poli Alfabetik Şifreleme

Bir harfin birden çok (POLİ) harfle gösterilebildiği şifreleme sistemleridir Bu şifreleme yönteminde, mono alfabetik yöntemlerden farklı olarak bir harf değiştirildiğinde her seferinde aynı harfe dönüşmez. Poli alfabetik şifreleme yöntemleri bilgisayar kullanılarak frekans sayımı ile kolay ve hızlı bir şekilde çözülebilir. (Stinson D.R., 2002)

### 3.2.1 Vigenere tablosu

Vigenere tablosu, poli alfabetik şifreleme yöntemlerine güzel bir örnektir. Bu yöntemde bir anahtar kelime ve oluşturulan tablo kullanılarak şifreleme yapılır. (Stinson D.R., 2002)

**Çizelge 3.1** Örnek bir Vigenere tablosu

| | 0 | | | | | 5 | | | | | 10 | | | | | 15 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **A** | **B** | **C** | **Ç** | **D** | **E** | **F** | **G** | **Ğ** | **H** | **I** | **İ** | **J** | **K** | **L** | **M** | **N** | **O** |
| **A** | a | b | c | ç | d | e | f | g | ğ | h | ı | i | j | k | l | m | n | o |
| **B** | b | c | ç | d | e | f | g | ğ | h | ı | i | j | k | l | m | n | o | ö |
| **C** | c | ç | d | e | f | g | ğ | h | ı | i | j | k | l | m | n | o | ö | p |
| **Ç** | ç | d | e | f | g | ğ | h | ı | i | j | k | l | m | n | o | ö | p | r |
| **D** | d | e | f | g | ğ | h | ı | i | j | k | l | m | n | o | ö | p | r | s |
| **E** | e | f | g | ğ | h | ı | i | j | k | l | m | n | o | ö | p | r | s | ş |
| **F** | f | g | ğ | h | ı | i | j | k | l | m | n | o | ö | p | r | s | ş | t |
| **G** | g | ğ | h | ı | i | j | k | l | m | n | o | ö | p | r | s | ş | t | u |
| **Ğ** | ğ | h | ı | i | j | k | l | m | n | o | ö | p | r | s | ş | t | u | ü |
| **H** | h | ı | i | j | k | l | m | n | o | ö | p | r | s | ş | t | u | ü | v |
| **I** | ı | i | j | k | l | m | n | o | ö | p | r | s | ş | t | u | ü | v | y |
| **İ** | i | j | k | l | m | n | o | ö | p | r | s | ş | t | u | ü | v | y | z |
| **J** | j | k | l | m | n | o | ö | p | r | s | ş | t | u | ü | v | y | z | a |
| **K** | k | l | m | n | o | ö | p | r | s | ş | t | u | ü | v | y | z | a | b |
| **L** | l | m | n | o | ö | p | r | s | ş | t | u | ü | v | y | z | a | b | c |
| **M** | m | n | o | ö | p | r | s | ş | t | u | ü | v | y | z | a | b | c | ç |

Çizelge 3.1'deki Vigenere tablosu dikkate alınarak "KOORDİNAT BELİRLENDİ" metni üzerinde yapılacak bir şifreleme ve şifre çözme işlemi aşağıdaki gibidir.

**Şifreleme**

Açık Mesaj (sütun) : KOORD İNATB ELİRL ENDİ

Anahtar Kelime (satır) : ÇAKIL ÇAKIL ÇAKIL ÇAKIL

Şifreli Mesaj : NOBBÖ LNKDM ĞLUBZ ĞNOS

**Deşifreleme**

Şifreli Mesaj (tablo) : NOBBÖ LNKDM ĞLUBZ ĞNOS

Anahtar Kelime (satır) : ÇAKIL ÇAKIL ÇAKIL ÇAKIL

Açık Mesaj (sütun) : KOORD İNATB ELİRL ENDİ

### 3.3 Tek Kullanımlık Karakter Dizisi

Tek Kullanımlık Karakter Dizisi (One-Time Pad) şifreleme yönteminde, rastgele üretilen bir karakter (harf veya rakam) dizisi kullanılarak şifreleme yapılır. Açık mesajdaki her karakter, üretilmiş olan karakter dizisinde karşılık gelen karakterle işleme sokulur ve şifreli mesaj elde edilir. Şifreli mesajı çözmek için ise rastgele karakter dizisini bilmek gereklidir. Bu yönteme Vernam şifreleme yöntemi denir. (Stinson D.R., 2002)

Açık Mesaj : BULUSMAYERIANKARA

Rastgele Dizi : DEFRYPLCNMLJKHFGH

Şifreli Mesaj : RLDYDOY....

Bu yöntemin güvenliği rastgele üretilen karakter dizisine bağlıdır. Eğer bu karakter dizis bir kurala bağlı olarak üretilir ve bu kural tespit edilirse sistem kırılabilir. Bu tehdit dışında mükemmel çalışan bir şifreleme sistemidir. Bu sistem ilk olarak 1917'de bulunmuş ve "teletype" makinelerinde kullanılmıştır.

**Şekil 3.2.** Teletype cihazı

# 4 GÜVENLİ ŞİFRELEME YÖNTEMLERİ

Güvenli şifreleme yöntemleri olarak bilinen Simetrik şifreleme ve Asimetrik şifreleme yöntemleri, Mono alfabetik ve Poli alfabetik şifreleme yöntemleriyle kıyasladığında daha güvenilirdir. (Koblitz N., 1994)

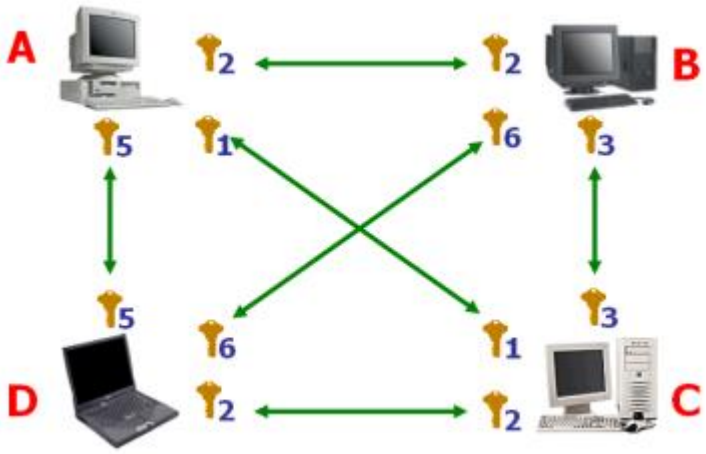
## 4.1 Simetrik Şifreleme

Simetrik şifreleme yönteminde, şifreleme ve deşifreleme işlemi aynı anahtar kullanılarak yapılır ve bu anahtar gizli tutulmalıdır. Bu nedenle, bu sistemlere aynı zamanda Gizli Anahtarlı Kriptografi Sistemi denir.

Simetrik şifreleme sistemlerinde, anahtar yönetimi Birden-Çoğa (One-to-Many) ve Çoktan-Çoğa (Many-to-Many) Anahtar Yönetimi olmak üzere iki kısma ayrılabilir.


**Şekil 4.1** Birden-Çoğa Anahtar Yönetimi

Birden-Çoğa Anahtar Yönetimi yönteminde (Şekil 4.1) haberleşen gönderici ve alıcılar aynı gizli anahtarı kullanmaktadır. Bu nedenle alıcılar başkasına gönderilen şifreli mesajları açabilir ve okuyabilirler.




**Şekil 4.2** Çoktan-Çoğa Anahtar Yönetimi

Çoktan-Çoğa Anahtar Yönetimi yöntemiyle (Şekil 4.2) haberleşen taraflarda, her ikili için farklı gizli anahtar kullanılmaktadır. Anahtar sayısı kullanıcı sayısına bağlı olarak değişmektedir ve anahtar sayısı kullanıcı sayısının 2'li kombinasyon sayısına eşittir.

| Kullanıcı Sayısı | Anahtar Sayısı |
|---|---|
| 3 | 3 |
| 4 | 6 |
| 10 | 45 |
| 100 | 4,950 |
| 1,000 | 499,500 |
| 10,000 | 49,995,000 |
| n | n*(n-1) /2 |

**Şekil 4.3** Çoktan-Çoğa Anahtar Yönetiminde kullanıcı ve anahtar sayıları

Simetrik şifreleme sistemlerine örnek olarak, blok şifreleme algoritmaları, bit katarı (dizi) şifreleme algoritmaları verilebilir.

### 4.1.1 Blok şifreleme algoritmaları

Blok şifreleme algoritmalarında, şifrelenecek veri sabit uzunluklu bloklar halinde şifreleme fonksiyonuna alınır ve aynı uzunluklu şifrelenmiş veri blokları üretilir. Daha açık olarak, şifrelenmemiş açık metin bitişik bloklara bölünür, her blok şifrelenerek şifreli metin bloklarına dönüştürülür ve son olarak bu şifreli

bloklar şifreli metin çıkışı olarak gruplandırılır. Blok şifreleme sistemlerinin en büyük avantajı, şifreli metin bloklarının herhangi birinin kaybı durumunda deşifre işleminde bir hata oluşmamasıdır. Blok şifreleme sistemlerinin en büyük dezavantajı ise, açık metindeki benzer bloklar şifrelenmiş metinde de birbirlerine benzemektedirler. Blok şifreleme algoritmalarına, AES, DES, IDEA, Skipjack, RC5 örnek olarak verilebilir. (Schroeder M.R., 1997)

#### 4.1.1.1 DES Algoritması

DES (Data Encryption Standart) algoritması, 1974 yılında IBM'in NSA (National Security Agency) ile geliştirdiği bir blok şifreleme algoritmasıdır. DES algoritması, şifrelenecek metni bloklar halinde şifreleme işleminden geçirir ve simetrik şifreleme prensibine dayanmaktadır. DES, 64 bitlik metin bloklarını, 56 bitlik bir anahtar ile şifreleyerek yine 64 bit genişlikteki şifrelenmiş metin bloklarına dönüştürür. Şifre çözme işleminde de 64 bitlik şifreli metin blokları, 56 bitlik anahtar yardımıyla yine 64 bitlik deşifrelenmiş metinlere (açık metine) dönüştürülür. (Stinson D.R., 2002)

DES algoritmasında, şifreleme ve deşifreleme işlemleri, yer değiştirme, permütasyon gibi bir dizi işlem çalıştırılarak gerçekleştirilmektedir. Bu işlemlerin sırası şifreleme ve deşifreleme için birbirlerinin tam tersi şeklindedir. DES algoritması basit şekilde aşağıdaki gibi ifade edilebilir.

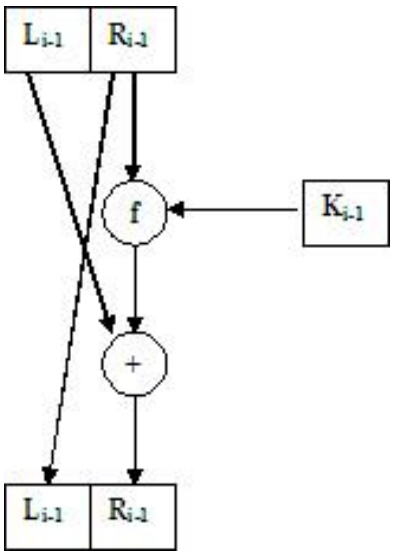
**A1.** *BP* başlangıç permütasyonu olsun. Verilen bir açık metin (*x*) başlangıç permütasyonuna tabi tutularak $x_0$ elde edilir.

$$x_0 \leftarrow BP(x)$$
$$L_0 = x' in ilk\ 32 biti$$
$$R_0 = x' in son\ 32 biti$$

**A2.** 16 defa tekrarlanan bir fonksiyon yardımıyla yeni değerler hesaplanır, $L_i, R_i\ (1 \le i \le 16)$ olmak üzere

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} XOR\ f\left(R_{i-1}, K_i\right)$$



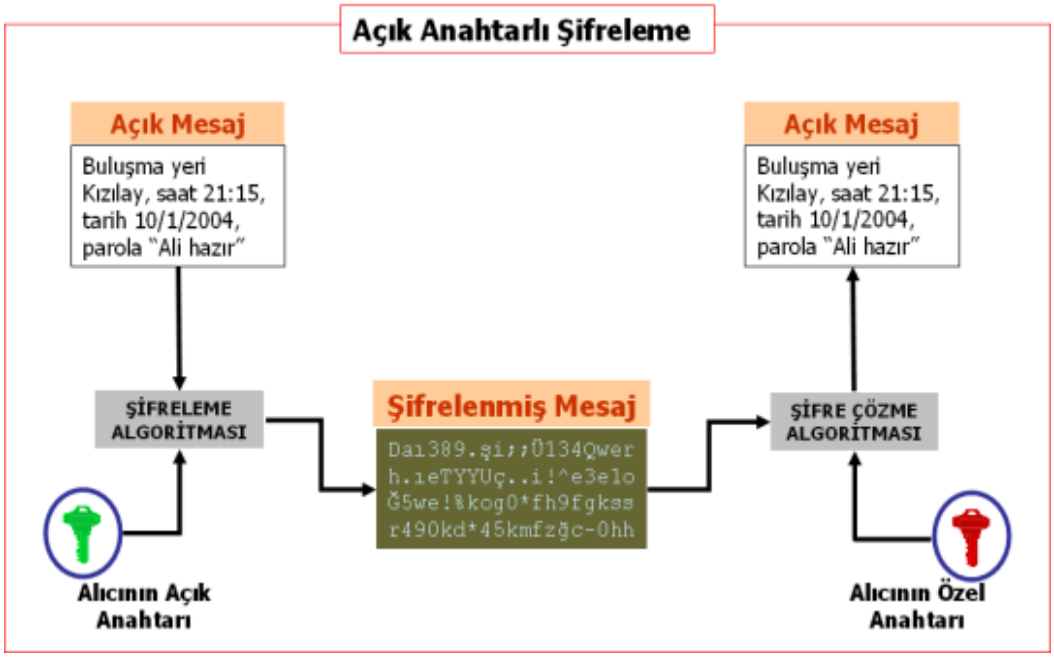

**A3.** Son olarak $L_{16}R_{16}$'ya $BP^{-1}$ (başlangıç permütasyonunun tersi) uygulanarak şifreli metin elde edilir.

### 4.2 Asimetrik Şifreleme

Asimetrik kriptografide, şifreleme ve deşifreleme işlemi farklı anahtarlar kullanılarak yapılır ve bu anahtarlar açık ve özel anahtar olarak adlandırılırlar. Asimetrik kriptografide, özel anahtar gizli tutulmalıdır, açık anahtar ise gerekli kişilerle paylaşılabilir. Bu nedenle, asimetrik kriptografi, açık anahtarlı şifreleme olarak da bilinir. (Nabiyev V.V., 2007)

Asimetrik kriptografi kullanarak haberleşen taraflar:

- Aynı şifreleme algoritmasını kullanırlar
- Birbiriyle uyumlu gerçeklemeler kullanırlar
- Gerekli anahtarlara erişebilirler




**Şekil 4.4.** Açık Anahtarlı Şifreleme

Asimetrik kriptografi yöntemlerinde anahtar yönetimi simetrik kriptografi yöntemlerine oranla daha kolaydır çünkü bir kullanıcıyla şifreli haberleşmek isteyen kişi karşı tarafın açık anahtarına ihtiyaç duyar. Bu açık anahtar halka açık olarak yayınlandığı için sisteme giren bir kişi için sadece bir anahtar çifti üretmek yeterli olmaktadır. Asimetrik kriptografi algoritmaları, simetrik kriptografi algoritmalarına göre çözülmesi zor matematiksel problemlere dayanmaktadır. Bazı asimetrik kriptografi algoritmalarına örnek olarak RSA, El Gamal, Eliptik Eğri Sistemleri, Diffie-Hellman verilebilir.

| Kullanıcı Sayısı | Anahtar Çifti Sayısı |
|---|---|
| 3 | 3 |
| 10 | 10 |
| 100 | 100 |
| 1,000 | 1,000 |
| 10,000 | 10,000 |
| n | n |

**Şekil 4.5** Açık anahtarlı şifrelemede kullanıcı sayısı ve anahtar cifti sayısı

### 4.2.1 RSA

1977 yılında R. Rivest, A. Shamir ve L. Adleman tarafından geliştirilmiş 4405829 Nolu Amerikan Patenti ile korunan ve kısaca ismi ile RSA en çok kullanılan açık anahtarlı şifreleme sistemidir. Bu yöntemin temel tekniği ise 1973 yılında C. Cocks tarafından önerilmiştir. RSA hem veri güvenliği, hem de sayısal imza için kullanılabilir. RSA'nın güvenliği çarpanlara ayırma probleminin zorluğuna dayanmaktadır. RSA ile yapılacak şifrelemede öncelikle anahtar üretimi yapılmalıdır. İletişimde yer alacak kişiler bir RSA açık anahtarı ve buna karşılık gelen özel anahtarı oluşturur. Günümüzde RSA'nın, SSL, S-HTTP, S-MIME, S/WAN, STT ve PCT gibi bir çok uygulama alanı bulunmaktadır. (Schneier B., 1996)

#### 4.2.1.1 RSA'da Anahtar Üretimi

RSA algoritmasında anahtar üretimi aşağıdaki işlem adımlarını içermektedir.

1. $p$ ve $q$ olarak iki büyük asal sayı üretilir. Buarada, büyüklük bit genişliğini ifade etmektedir.

2. $n = p \cdot q$ ve $\phi(n) = (p-1)\cdot(q-1)$ hesaplanır.

3. $1 < e < \phi(n)$ ve $ebob\left(e, \phi\left(n\right)\right) = 1$ olacak şekilde rasgele bir $e$ sayısı seçilir.

4. $1 < d < \phi(\mathrm{n})$ ve $e \cdot d = 1\left(\mathrm{mod}\ \phi(n)\right)$ şartını sağlayan $d$ sayısı hesaplanır.

5. Açık anahtar ($n$,$e$), özel anahtar ise $d$'dir.

Herhangi bir mesajı şifrelemek ve deşifrelemek için aşağıdaki yöntem izlenir.

#### 4.2.1.2 RSA'da şifreleme ve deşifreleme:

RSA sisteminde şifreleme işlemini gerçekleştirmek için aşağıdaki algoritma kullanılır.

**A1.** Mesajın gönderileceği açık anahtar $(n,e)$ elde edilir.

**A2.** Şifrelenecek mesaj, $[0,n-1]$ aralığındaki bir $M$ sayısına dönüştürülür.

**A3.** $C = M^e \pmod{n}$ hesaplanır.

**A4.** Elde edilen C şifreli mesajı alıcıya gönderilir.

Şifreli mesajı deşifre etmek için ise $d$ özel anahtarı yardımıyla $M = C^d \pmod{n}$ hesaplanır ve orijinal mesaj elde edilir.

$e$ ve $d$ aşağıdaki denkliği sağlayacak şekilde seçilmiştir.

$$d = e^{-1} \bmod \phi(n)$$

Bunun sonucunda,

$$e \cdot d = 1 \pmod{\phi(n)}.$$

Bu yüzden $d \cdot e = k \cdot \phi(n) + 1$ dir. Bu denkliğin Euler teoreminin bir sonucu olduğu, iki asal sayı olan $p$ ve $q$ ile birer tamsayı olan $n\ (n = p \cdot q)$ ve $M\ (0 < M < n)$ alınarak ispatlanabilir.

$$M^{k\cdot\phi(n)+1} = M^{k\cdot(p-1)\cdot(q-1)+1} \equiv M \pmod{n}$$

Dolayısıyla, $M^{ed} \equiv 1 \pmod{n}$

$$C = M^e \ mod \ n$$

$$M = \ C^d \ mod \ (n) \equiv \left(M^e\right)^d \ mod \left(n\right) \equiv M^{ed} \ mod \left(n\right) \equiv M \ mod \ n$$

Hem gönderen hem de alıcı *n*'in değerini bilmelidir. Gönderen *e*'nin değerini bilir ve sadece alıcı d'nin değerini bilir. Böylece; (*e*,*n*) açık anahtar ikilisi, *d* gizli anahtar olur ve bu bir açık anahtarlı şifreleme algoritmasıdır. Açık anahtarlı şifreleme bir tatmin edici olması için, bu algoritmada aşağıdaki gereklilikler yerine getirilmelidir:

1. $M < n$ koşulunu sağlayan tüm $M$ değerleri için $M^e$ ve $C^d$ hesaplanabilmelidir.

2. Sadece *e* ve *n* verildiğinde, *d*'nin hesaplanması imkansız olmalıdır.

**4.2.2 RSA'nın Güvenliği** (Riesel H., 1985)

RSA sistemi *n*'nin çarpanlarına ayrılması ile çözülebilir, Yani gizli anahtar bulunabilir. *n,* uzunluğu 512 ile 2048 bit arasında değişen bir sayı olduğundan mevcut teknoloji ile gereken hesaplama gücü ve zamanı oldukça fazladır. 100 haneli bir sayıyı kaba kuvvet yöntemi (brute force attack) ile çarpanlarına ayırmak için en fazla $10^{50}$ işlem gerekir. Günümüzdeki en güçlü bilgisayarlar bile standartlara uygun seçilmiş 512 bitlik bir sayıyı makul bir zamanda çarpanlarına ayırmada güçlük çekmektedir. Aynı şekilde 1024 bitlik bir sayının çarpanlarına ayrılması zaman açısından pratik değildir. Yine de *n*'nin çarpanlarına ayrılması için kaba kuvvet yönteminden (brute force attack) daha verimli ataklar bulunmaktadır. Aşağıda RSA şifreleme yöntemine ilişkin bir örnek verilmiştir.

Şifreleme yapılmadan önce anahtar üretimi gerçekleştirilmelidir. *p*=13, *q*=23 olarak seçilirse, *n*=13x23=299 olur. $\phi(n) = 12\text{x}22 = 264$ olarak hesaplanır. *e*=35 olarak seçildiğinde, *d*=83 olur; 35x83≡1 (mod 264)'dir. Anahtar üretimi yapıldıktan sonra şifreleme işlemine geçilebilir.

Şifrelenecek mesaj, yani düz metin **"zeka"** kelimesi olsun. RSA'da işlemler sayılar üzerinden yapıldığından, mesaj sayısal bir ifadeye dönüştürülmelidir. Bu amaçla her karakterin ASCII kodu kullanılabilir. Bu durumda açık metin aşağıdaki şekli alır.

$z \rightarrow 122$, $e \rightarrow 101$, $k \rightarrow 107$, $a \rightarrow 097 \Rightarrow$ "122101107097"

Şifrelenecek sayılar *n*'den küçük olmak zorundadır. Bu nedenle sayısal metin *n*'nin basamak sayısının bir eksiği uzunluktaki parçalara ayrılır. Bu sayı $L_{clear}$ olarak adlandırılır. Örnekte *n* sayısı 3 basamaklı olduğundan, $L_{clear}=2$ olur ve mesaj ikili bloklara ayrılır.

"12 21 01 10 70 97"

Her blok $L_{clear}$ basamaklı olmak zorundadır. Bu nedenle eksik kalan blokların sonuna gereken sayı kadar 0 eklenir.

Şimdiki aşama bu sayıların kurala uygun bir şekilde şifrelenmesidir. Şifreleme işlemi aşağıdaki şekilde yapılır.

$12^{35} \bmod 299=259$

$21^{35} \bmod 299=226$

$01^{35} \bmod 299=1$

$10^{35} \bmod 299=119$

$70^{35} \bmod 299=47$

$97^{35} \bmod 299=297$

Bu aşamada oluşan sayılar *n* ile aynı basamak sayısına sahip olmalıdır. Bu sayı $L_{cipher}$ ile gösterilir. Örnek için $L_{cipher}$ 3'e eşittir. Bu nedenle gerekli bloklar

için sayıların sol tarafına gerekli sayıda 0 eklenmelidir. Bu durumda şifreli metin aşağıdaki hale gelir ve gönderilir.

"259 226 001 199 047 297" → **Şifreli Metin: "**2592260011990472 97"

Alınan "2592260011990472 97" şifreli metni öncelikle $L_{cipher}$ uzunluklu bloklara bölünür ve hemen ardından deşifreleme kuralı uygulanır.

"259 226 001 199 047 297"

$259^{83} \bmod 299=12$

$226^{83} \bmod 299=21$

$001^{83} \bmod 299=1$

$199^{83} \bmod 299=10$

$047^{83} \bmod 299=70$

$297^{83} \bmod 299=97$

Sonuçların $L_{clear}$ uzunlukta olması gerektiğinden gerekli bloklarda sol tarafa gerekli miktarda 0 eklenir.

"12 21 01 10 70 97"

Son aşama, bu sayıların birleştirilmesi ve üçlü bloklara bölünmesi ile ASCII kodların elde edilmesidir. Fazlalık olarak oluşan 0'lar göz önüne alınmaz.

"122101107097"

"122 101 107 097"

122 → z, 101→e, 107→k, 097→a

Sonuçta açık metin “zeka” olarak elde edilmiştir.

*p* ve *q* değerleri asal olmalıdır. Bu sayıların asal olmaması durumunda RSA algoritması çalışmaz. Asal sayıları $2^{-100}$ güvenlikle oluşturan algoritmalar bilinmektedir.

RSA’nın kullanımında ortaya çıkan problemler aşağıda maddeler halinde verilmiştir (Stephenson, D., 1996).

1. Güvenilir asal sayıların oluşturulması.

2. Hangi uzunlukta anahtar kullanılacağının belirlenmesi (Seçilen asalların birbirine yakın olması durumunda *n* sayısı fermat çarpanlara ayırma yöntemi ile oldukça kolay şekilde çarpanlarına ayırılabilir).

Eğer *k* bitlik bir anahtar kullanılırsa genel anahtar için $O(k^2)$, gizli anahtar için $O(k^3)$, ve yeni anahtarın oluşturulması için $O(k^4)$ işlem gerekmektedir.

Günümüzde kriptografik algoritmalar veri gizlemesine ve gömmesine dayalı stegonografik yöntemlerle beraber kullanılarak daha karmaşık hale getirilmektedir.

Hiç şüphesiz, RSA kullanımı hayatımızın her sahasında mevcuttur. Bankamatik ve kredi kartı kullanımında, cep telefonlarında ve televizyonlarda izlediğimiz şifreli kanallarda, askeri sistemler, yani kısaca hayatımızın her sahasında sıklıkla kullandığımız çok önemli bir yere sahiptir. Kısaca RSA’yı özetlersek;

- Açık anahtar kriptografi sistemi ve sayısal imzalama yöntemi olarak kullanılır.
- Çarpanlarına ayırma problemi üzerine inşa edilmiştir.
- Bileşik tam sayı olan *n*'i oluşturan, asal sayılar *p* ve *q* bulunur, öyleki *n=pq* 'dir.
- Yeterince büyük bir *n* için kırılması çok zordur.

- Ayrıca kök bulma problemine dayanır.
- Çok güvenlidir fakat fazla hızlı değildir.

**Algoritmanın kullandığı parametreler:** (Cormen T.H., Leiserson C.E., Rivest R.L. and Stein C., 2002)

- Açık anahtar : *n*, *e*
- Özel anahtar : *d*
- *n* bileşik bir tamsayıdır ("modulus")
- *e* bir tamsayıdır ("açık üs ifadesi")
- *d* bir tamsayıdır ("gizli üs ifadesi")

**Algoritmanın kullanımı:**

Merve, Cemre'ye *m* mesajını şifreli göndermek için:

- *m*'nin e'inci üssünü alır, yani *m*'yi Cemre'nin açık anahtarı ile şifreler:
- $c = m^e \bmod n$
- *c*'yi ("şifreli mesajı") Bora'ya gönderir
- Cemre *c* sayısının *d*'nci üssünü alır, yani *c*'nin şifresini kendi özel anahtarını kullanarak çözer:
- $m = c^d \bmod n$

**Algoritmanın Kriptoanalizi:** (Friese L., 2003)

- Şifreleyenin işi kolaydır, sadece iki adet modüler çarpma işlemi yapar.
- Deşifreleyenin (alıcının) işi kolaydır, $1.5 \cdot logn$ tane modüler çarpma yapar.
- Eğer asal sayılar (*p*, *q*) bilinirse işlem daha hızlı yapılabilir.
- Saldırganın işi zordur çünkü kök bulma ya da çarpanlarına ayırma problemini çözmelidir.
- Etkili bir çözümü bulunmamıştır.
  - *n* sayısı arttıkça algoritmanın güvenliği artar.

### 4.2.3 Fermat Çarpanlara Ayırma Yöntemi

RSA yı tehdit eden en önemli yöntemlerden biri Fermat çarpanlara ayırma yöntemidir.

$n$>1 ve tek sayı olmak koşulu ile ( $n$ asal değildir) ancak ve ancak $n = a^2-b^2$ olacak şekilde $a-b$>1 koşulunu sağlayacak $a$ ve $b$ pozitif tam sayıları vardır. (Giblin P., 1993)

Biraz daha açmak gerekirse $n$ pozitif tamsayısı verilip Fermat Çarpanlara Ayırma Yöntemi ile çarpanlarına ayırmamız istendiğinde, eğer $n$ asal değilse; mutlaka $n=a^2-b^2$ olacak şekilde uygun bir çözümü vardır.

n= p*q şeklinde gösterilebilmesi için $a = \frac{p+q}{2}, b = \frac{p-q}{2}$ olmalıdır.

$n= \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2$ olacaktır. Gösterelim.

$$n= \left(\frac{p+q}{2}\right)^2 - \left(\frac{p-q}{2}\right)^2 = (p+q)^2/4-(p-q)^2/4$$

$$=(p^2+2pq+q^2)/4-(p^2-2pq+q^2)/4$$

$$=((p^2+2pq+q^2)-(p^2-2pq+q^2))/4$$

$$=(p^2+2pq+q^2-p^2+2pq-q^2)/4$$

$$=(2pq+2pq)/4$$

$$=4pq/4$$

$$=p\ q$$

Yani $n$ sayısına 1 den başlayarak sırasıyla 1,2,…,($p$-$q$)/2 ye kadar tüm sayıların karesi tek tek eklenir ve $n+q^2$ nin bir tam kareye eşit olması beklenir. Bu işlem tam olarak ($p$-$q$)/2 adım sürer.

$n=\left(\frac{p+q}{2}\right)^2-\left(\frac{p-q}{2}\right)^2=p\,q$ olduğunu gösterdik.

Bu durumda çözüm için $\left(\frac{p-q}{2}\right)^2$ ifadesini karşıya atarak

$n+\left(\frac{p-q}{2}\right)^2=\left(\frac{p+q}{2}\right)^2$ bu forma dönüştürebiliriz. Bu durumda algoritmanın tek yapması gereken 1 den başlayarak birer birer $\left(\frac{p-q}{2}\right)$ kadar tek tek tüm sayıların karesini almaktır.

$n+1^2$ tam karemi ? (hayır)

$n+2^2$ tam karemi ? (hayır)

$n+\left(\frac{p-q}{2}\right)^2$ tam karemi ? (evet)

***Örneğin;***

n=527 olsun.

| q | $n+q^2$ | |
|---|---|---|
| 1 | $527+1^2=527+1=528$ | tam kare değil |
| 2 | $527+2^2=527+4=531$ | tam kare değil |

3 $527+3^2=527+9=536$ tam kare değil

4 $527+4^2=527+16=543$ tam kare değil

5 $527+5^2=527+25=552$ tam kare değil

6 $527+6^2=527+36=563$ tam kare değil

7 $527+7^2=527+49=576=(24)^2$ tam kare

$527+7^2=24^2$ olduğundan

$527 = 24^2-7^2$

$=(24+7)(24-7)$

$=31x17$

$p=31$ ve $q=17$ olur. ($p<q$)

Sonuç olarak *n* asal olmadığı sürece 1'den (*n*-1)/2 ye kadar tüm *q* lar kontrol edilerek çözüme ulaşılır. Bu durumda RSA algoritmasında aldığımız *p* ve *q* asalları ne kadar büyük olursa olsun bir birlerine ne kadar yakınlarsa *n* o kadar çabuk çarpanlarına ayrılır.

# 5 KRİPTOSİSTEMLERİN KARŞILAŞTIRILMASI

Güvenli kriptosistemleri simetrik ve asimetrik olmak üzere iki başlık altında gruplandırdığımızda birbirlerine göre avantaj-dezavantajları, güçlü ve zayıf yönlerini aşağıdaki şekilde listeleyebiliriz (Pomerance, C., Goldwasser, S., Lagarias, J.C. and others 1989).

Simetrik kriptografinin kuvvetli yönleri:

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

Simetrik kriptografinin zayıf yönleri:

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

Asimetrik kriptografinin kuvvetli yönleri:

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

Asimetrik kriptografinin zayıf yönleri:

- 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.

**Çizelge 5.1** Simetrik ve Asimetrik kriptografinin güvenlik özellikleri

| **Konu** | **Simetrik Kriptografi** | **Asimetrik Kriptografi** |
|---|---|---|
| Gizlilik | Sağlar | Sağlar |
| Bütünlük | Sağlamaz | Sağlar |
| Kimlik Doğrulama | Sağlamaz | Sağlar |
| İnkar Edememezlik | Sağlamaz | Sağlar |
| Performans | Hızlı | Yavaş |
| Güvenlik | Anahtar uzunluğuna bağlı | Anahtar uzunluğuna bağlı |

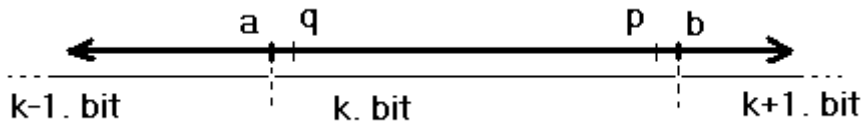
# 6 DİRENÇLİ ASAL SAYI SEÇİM YÖNTEMİ

RSA'nın güvenliğini Fermat Çarpanlara Ayırma Yöntemi ve bu yöntemden doğan diğer yöntemler karşısında artırmak için RSA sisteminde aynı bit genişliğinde olan asalları seçebileceğimiz uygun aralığı belirleyelim.

*p* ve *q* asalları aynı bit üzerinde bulunurlar. Kaba koda karşı güçlü kalması için asal sayıların ikisinin de büyük olması şarttır. En iyi durum olarak asallar aynı bit üzerinde alınmaya çalışılır.

Asallar arasında bit farkı olmadığından en uç örnek bulunulan bit değerinin en büyük ve en küçük değerlerine yaklaşmaktır.

*p* ve *q* 'yu k. bitten aldığımızı var sayalım. RSA algoritmasında seçilen asallar genelde aynı bit sayısına sahip sayılardır. Ayrıca oldukça büyük olan bu sayılar $2^{512}$, $2^{1024}$, $2^{2048}$ basamaklıdır.




$a=2^k$ ve $b=2^{k+1}$ olmak üzere ***min*(*q*-*a*)** ve ***min*(*b*-*p*)** için *p* ve *q* değerleri sırasıyla *b* ve *a*'ya çok yaklaşır. (*)

Ayrıca, $x^2 < r < (x+1)^2$ eşitliğini sağlayan en az bir r asal sayısının daima var olduğunu daha ispatlanamamış fakat kabul görmüştür. Bu önermeyi kullanarak *p* ve *q* nun analiz aşağıdaki şekilde yapılabilir.

$n = p * q$ olsun (p ve q asal sayı olmak üzere) .

$y^2 < p < (y+1)^2$ ve

$z^2 < q < (z+1)^2$ olacaktır. (**)

Buradan da

$y < \sqrt{p} < (y+1)$ *ve*

$z < \sqrt{q} < (z+1)$ *olur.* $(y, z, p, q \in \mathbb{Z}^+)$ (***)

Burada elde ettiğimiz *y* ve *z* sayıları bize *p* ve *q* ilişkisi hakkında yaklaşık bir oran verir. Bu $y/z$ oranı $\sqrt{p}/\sqrt{q}$ oranına oldukça yakın alınabilir ve $y/z$ oranını bilmek doğal olarak *p* ve *q* hakkında oldukça yakın bilgiler verecektir.

Bu yöntem ile arzu edilen güvenilir bir $y/z$ oranını belirleyip asalları bu orana göre seçerek RSA algoritmasını Fermat Çarpanlara Ayırma Yöntemi'ne karşı daha güçlü kılmaktır.

(*) da belirtildiği üzere aynı bit üzerindeki en uç örnek $p \cong 2^{k+1}$ ve $q \cong 2^k$ dır.

Bu durumda $p/q \cong 2^{k+1}/2^k = 2$ olacaktır.

$\sqrt{\frac{p}{q}} = \sqrt{p}/\sqrt{q} \cong \sqrt{2}$ ( $\sqrt{2} \cong 1.41421356$ dir.)

(***) dan $\sqrt{p}/\sqrt{q} \cong y/z \cong \sqrt{2}$ olur.

Buradan da $y/z \cong 1.41$ alınabilir.

Bu değer aynı bit üzerinde ise bize Fermat Çarpanlara Ayırma Yöntemi'ne karşı sağlanabilecek en büyük güvenliği sağlar.

$y/z \cong 1.41$ değeri en büyük değerdir. Bu durumda aynı bit üzerinde $y/z$ oranının alt sınırının nerede olucaöını bulmalıyık. Bu değeri ancak şifre kırıcının kullandığı sisteme, algoritmayı uygulayış şekline v.b. pek çok parametreye bağlı

bulmak oldukça zor olur. Bunu anahtar uzunluğu sabit ve $y/z$ oranı değişen bir tabloda görmek bize uygun fikir vericektir.

**Çizelge 6.1** $y/z$ oranı ve çözümdeki -adım sayısı

| **y/z** | **Çözümdeki adım sayısı((p-q)/2)** ($2^k$ bit anahtar genişliği için) |
|---|---|
| 1.0 | $0.000 \times 2^k$ |
| 1.1 | $0.105 \times 2^k$ |
| 1.2 | $0.220 \times 2^k$ |
| 1.3 | $0.345 \times 2^k$ |
| 1.4 | $0.480 \times 2^k$ |

Çizelge 6.1'deki değerlerin nasıl bulunduğu bir örnekle anlatalım:

Halihazırda kişisel bilgisayarlarımızda dahi p ve q asallarını $2^{512}$ bit aldığımız için $y/z$ oranındaki en küçük değişiklik bile çok büyük farklar yaratmaktadır.

$y/z$ =1.3 olsun.

$y=1.3x2^{256}$ ve $z=1x2^{256}$ olacaktır.

Yani *p* ve *q* sırasıyla,

$(y=1.3x2^{256})^2 < p < (y=1.3x2^{256}+1)^2$ ne

$(z=1.0x2^{256})^2 < q < (z=1.0x2^{256}+1)^2$ dir.

Yukarıdaki eşitsizlikten de anlaşılacağı üzere $\left(\frac{p-q}{2}\right)$ sayısı için *n* sayısı Fermat çarpanlarına ayırma yöntemine göre,

$\left(\frac{p-q}{2}\right) \cong \left(\frac{(1.69x2^{512})-2^{512}}{2}\right) = \left(\frac{0.69x2^{512}}{2}\right) = 0.345x2^{512}$ adımda çözüme kavuşur. Bu sayının büyüklüğü ise oldukça caydırıcıdır. Burada dikkat edilmesi gereken husus sayıların çok büyük olmasıdır.

$$1 < y/z < 1.4$$

Aynı bitte seçilecek asallarımızın y/z oranı yukarıdaki aralıkta olması şart olmakla beraber oranları 1.4 civarına yaklaştıkları durumda bize çok daha güvenli bir RSA sistemi sunacaktır.

### 6.1 Dirençli Asal Sayı Seçim Algoritması

Dirençli Asal Sayı Seçim algoritması iki farklı şekilde çalışmaktadır:

1. Girilen *p* ve *q* asallarını dikkate alarak y/z oranı bulunur
2. Girilen y ve z değerleri için p ve q asalları üretilir.

```
#include<stdio.h>
#include<math.h>
#include <conio.h>
int asaluret(int x2){
  int y2,i,j,say,asalsayi;
  char asal='H';
  y2=x2*x2;
  if((y2%2)==0){y2-=1;}
  for(i=y2+2;asal=='H';i+=2){
    say=0;
    for(j=2;j<sqrt(i);j++)
      if((i%j)==0) {say++;};
    if(say==0){asal='E';asalsayi=i;}
  }
  return asalsayi;
}
int altsinir(int x1){
  int t;
  for(t=1;t*t<x1;t++);
  return t-1;
}
int fermat_fac(int p1,int q1){
  int iterasyon;
  iterasyon=((p1-q1)/2);
  return iterasyon;
}
```

```
    int main(){
      FILE *dosya,*dosya2;
      char krt;
      int p,q,y,z,secim,iterasyonsay,tut;
      float oran;

      do{
        printf("\nElinizdeki asallarin y/z orani icin 1 ,
\n");
        printf("Elinizdeki y/z orani ile p,q asallarini
uretmek icin 2,\n");
        printf("Programdan cikmak icin 3 sayilarini
girin\n");
        scanf("%d",&secim);
        if(secim!=1 && secim!=2 && secim!=3){printf("\n\n
...hatali giris...\n\n");}
        if(secim==1){
          dosya=fopen("oran.txt","w");
          do{
            printf("\n P asal sayisini girin lutfen=");
            scanf("%d",&p);
            printf("\n Q asal sayisini girin lutfen=");
            scanf("%d",&q);
            if(p<q){
              tut=p;p=q;q=tut;
              printf("\nSayilarin yerleri degistirildi ");
              printf("\n p sayisi = %d ve ",p);
              printf("\n q sayisi = %d",q);
            }
            y=altsinir(p);
            printf("\n alt sinir(p)=%d",y);
            z=altsinir(q);
            printf("\n alt sinir(q)=%d",z);
            oran=(float) y/z;
            printf("\n %d",y);
            printf("\n %d",z);
            printf("\n\n y/z= %f \n",oran);
            iterasyonsay=fermat_fac(p,q);
            printf("iterasyon sayisi (fermat faktorizasyona
gore) ((p-q)/2)=%d \n",iterasyonsay);
            fprintf(dosya,"\n %d ve %d asallari için \n
%d/%d=%f",p,q,y,z,oran);
            fprintf(dosya,"\n fermat faktorizasyona göre
iterasyon sayısı=%d",iterasyonsay);
            printf("isleme devam etmek istiyormusunuz?(e,h)
\n");
            krt=getche();
          }while(krt=='e'||krt=='E');
          fclose(dosya);
        }
        if(secim==2) {
          dosya2=fopen("asal.txt","w");
          do{
            printf("\n y sayisini girin lutfen=");
            scanf("%d",&y);
            printf("\n z sayisini girin lutfen=");
```

```
            scanf("%d",&z);
            if(y<z){
            tut=y;y=z;z=tut;
            printf("\nSayiların yerleri degistirildi ");
            printf("\n y sayisi = %d ve ",y);
            printf("\n z sayisi = %d",z);
            }
            p=asaluret(y);
            printf("\n uretilen p asal sayisi= %d \n",p);
            q=asaluret(z);
            printf("\n uretilen q asal sayisi= %d \n",q);
            oran=(float) y/z;
            printf("\n\n y/z= %f \n",oran);
            iterasyonsay=fermat_fac(p,q);
            printf("iterasyon sayısı(fermat faktorizasyona
göre)((p-q)/2)=%d",iterasyonsay);
            fprintf(dosya2,"\n %d ve %d asalları için \n
%d/%d=%f",p,q,y,z,oran);
            fprintf(dosya2,"\n fermat faktorizasyona göre
iterasyon sayısı=%d",iterasyonsay);
            printf("isleme devam etmek istiyormusunuz?(e,h)
\n");
            krt=getche();
          }while(krt=='e'||krt=='E');
          fclose(dosya2);
        }
      }while(secim!=3);
      printf("\n\n\n\n Program Sonlandiriliyor. \n\n\n");
    }
```

# 7 SONUÇ

Bu tezde, RSA şifreleme sisteminin Fermat Çarpanlara Ayırma Yöntemi ve bu yöntemi temel alan pek çok yönteme karşı güvenliğini artırmak için araştırma yapılmıştır. RSA sisteminde aynı bit uzunluğunda seçilecek asallar için Fermat Çarpanlara Ayırma Yöntemi göz önüne alınarak uygun aralık belirlenmiştir. RSA'daki asal sayı seçiminde uygulanacak olan bu kriterle daha dirençli asal sayı seçiminin mümkün olduğu gösterilmiştir. Böylece, RSA sisteminin Fermat Çarpanlarına Ayırma Yöntemine karşı direncini artırmak için uygun aralık belirlenmiştir.

# KAYNAKLAR DİZİNİ

**Arjen, K.L.,** 2000, Integer Factoring, Designs, Codes and Cryptography, Kluwer Academic Publishers, 19, 101–128

**Pomerance, C., Goldwasser, S., Lagarias, J.C., Arjen K.L., McCurley, K.S., Odlyzko A.M.** 1989, Cryptology and Computational Number Theory.

**Cormen T.H., Leiserson C.E., Rivest R.L. and Stein C.,** 2002, Introduction to Algorithms, McGraw – Hill, MIT Pres.

**Çimen C., Akleylek S., Akyıldız E.**, 2002, Şifrelerin Matematiği Kriptografi, ODTÜ yayıncılık, Ankara.

**Giblin P**., 1993, Primes and Programming-An Introduction to Number Theory with Computing, Cambridge University Press, New York, Cambridge [England].

**Koblitz N.**, 1994, A Course in Number Theory and Cryptography, 2nd Edition, Springer - Verlag, New York.

**Nabiyev V. V.**, 2007, Algoritmalar. Teoriden Uygulamalara, Seçkin Yayıncılık, – 799 s. Ankara.

**Riesel H.**, 1985, Prime Numbers and Computer Methods for Factorization, Progress in Mathematics, Birkhäuser.

**Rivest R. L., Shamir A., Adleman L.** – 1977, On Digital Signatures and Public Key Cryptosystems, *MIT Laboratory for Computer Science Technical Memorandum 82*.

**Schneier B.**, 1996, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C, 2nd Edition, John Wiley & Sons, Inc.

**Schroeder, M. R.**, 1997, Number Theory in Science and Communication, Third Edition, Springer-Verlag, Berlin.

## KAYNAKLAR DİZİNİ (Devam)

**Stephenson, D.**, 1996, Factoring Very Large Numbers http://web2.clarkson.edu/projects/spr/directedStudy_fa2001/factoring/papers/stephenson96factoring.pdf

**Stinson D. R.** 2002, Cryptography: Theory and Practice, 2nd Ed., Chapman & Hall/Crc, ISBN 1-58488-206-9.

**Stinson D. R.** 2002, Cryptography: Theory and Practice, CRC Press.

**Wagstaff S. S.** – 2002, Cryptanalysis of Number Theoretic Ciphers, Computational Mathematics, Chapman & Hall/Crc, ISBN 1-58488-153-4.

# ÖZGEÇMİŞ

NASİBOV Şahin Efendi oğlu, 14 Ekim 1989 yılında Bakü'de (Azerbaycan) doğdu. 2007 yılında Azerbaycan Teknik Üniversitesini bitirdi. 2013 yılı Şubat ayından itibaren Ege Üniversitesi Fen Bilimleri Enstitüsü, Matematik Ana Bilim Dalı Bilgisayar Bilimleri Bilim Dalında Yüksek Lisans yapmaktadır.

***Yayınları:***

**Nasibov, S. E., Gürsoy A.** "Prime number selection resistant to Fermat's factorization method for RSA cryptosystem", Proceeding of the international Mathematics Symposium Karatekin Mathematics Days 2014 (KMD 2014), p. 23, Cankiri, TURKEY, June 11-13, 2014,