DOI: http://dx.doi.org/10.18185/eufbed.89538

EÜFBED - Fen Bilimleri Enstitüsü Dergisi Cilt-Sayı: 7-1 Yıl: 2014 *105-132*

RSA ŞİFRELEME SİSTEMİNE KARŞI YENİ BİR ÇARPANLARA AYIRMA SALDIRISI

A NOVEL FACTORIZATION ATTACK FOR RSA CRYPTO SYSTEM

Cihan MERT[1*] ve Şadi Evren ŞEKER[2]

[1]*Uluslararasi Karadeniz Üniversitesi, Bilgisayar Teknolojileri ve Mühendislik Fakültesi, Bilişim Bölümü, 0131, Tiflis, Gürcistan*

[2]*İstanbul Üniversitesi, Bilgisayar Mühendisliği Bölümü, Avcılar, İstanbul, Türkiye*

Geliş Tarihi: 13 Mayıs 2013 **Kabul Tarihi:** 14 Mayıs 2014

ÖZET

Bu çalışmanın amacı, öncelikli olarak RSA şifreleme yönteminde kullanılan ve iki asal sayının çarpımından oluşan yarı-asal sayıları, çarpanlara ayırmaya yöneliktir. Bu makale kapsamında sık kullanılan ve öne çıkan çarpanlara ayırma yöntemlerinin açıklanması ve performanslarının karşılaştırılması yapılmıştır. Çalışma kapsamında yeni bir çarpanlara ayırma yöntemi önerilmiştir. Ayrıca çarpanlara ayırma yöntemleri, rastgele üretilen asal sayılar üzerinde denenerek yeni önerilen yöntemin başarısı sınanmıştır. Yapılan çalışmalar, RSA yönteminde kullanılan yarı-asal sayılara saldırmak için, önerilen yeni yöntemin, mevcut yöntemlere göre daha avantajlı olduğunu ortaya koymaktadır.

Günümüz şifreleme teknolojilerinin tamamı, matematiksel bir zorluğa dayalı olarak geliştirilmiştir. Örneğin iki sayının çarpılması kolay ancak bir sayının çarpanlarına ayrılması zordur. Benzer şekilde a^b şeklinde üst almak kolay ancak tersi olan logaritma hesaplanması zor işlemdir. Bu zorluk, bilgisayar bilimleri açısından işlem karmaşıklığı veya daha açık bir ifadeyle zaman karmaşıklığı olarak ortaya çıkmaktadır. Günümüz şifreleme sistemlerine yapılan saldırıların tamamının bilgisayar tabanlı olduğu düşünülürse, bir sistemin güvenliği ne kadar uzun süre saldırıya dayanabileceği ile ölçülmektedir. En yaygın kullanılan şifreleme algoritmalarından birisi olan RSA, hem logaritma hem de çarpanlara ayırma zorluğu üzerine inşa edilmiştir. Örneğin RSA sistemine yapılacak bir saldırı sırasında kullanılan anahtarın, asal çarpanlarına ayrılması gerekmektedir. Çarpanlara ayırma için ise yıllar boyunca çeşitli yöntemler geliştirilmiştir. Bu yazı kapsamında mevcut çarpanlara ayırma yöntemleri incelenmiş ve bilgisayar üzerinde Java dili ile kodlanarak üretilen 1,000 adet 15 haneli rast gele sayı üzerinde performansları karşılaştırılmıştır. Bu çalışmada kullanılan çarpanlara ayırma yöntemleri, önerilen yeni yönteme ilave olarak, Deneme, Fermat, Eliptik Eğri (elliptic curve method), ikinci dereceden kalbur

***Sorumlu Yazar: mertcihan1997@gmail.com**

(quadratic sieve), Eratosthene Sieve (Atkin Kalburu) ve Polllar-Rho yöntemleridir.

Anahtar Kelimeler: RSA, Eliptik Eğri, ECC, Ayrık Logaritma, Şifreleme, Kalburlama

ABSTRACT

The aim of this study is to factorize semi-prime numbers which are multiplication of two primes and primarily used in RSA encryption method. In this article, commonly used factorization methods are introduced and their performance are compared. A new method of factorization is proposed. Also the success of the new proposed method is tested by testing these factorization methods on randomly generated prime numbers. Studies reveals that the proposed new method has more advantages compared to existing methods in attacking semi-primes numbers used in the RSA method.

All of today's encryption technology has been developed on the basis of a mathematical difficulty. For example, multiplying two numbers is easy but factoring a number is difficult. Similarly, taking bth power of a is an easy operation while its inverse operation calculating logarithms is relatively difficult. This difficulty in terms of computer science means complexity of the process or the time complexity. All of the attacks in today's sencryption systems are computer-based so the security of the system is measured by how long it can withstand the attack. RSA which is one of the most widely used cryptographic algorithms has been built on the difficulty of factorization and logarithm. For example, the key used during an attack on the RSA system must be factorized into its prime factors. Over the years several methods have been developed for the factorization. In this article some popular current factorization methods were examined and their performances were compared on 1000 numbers which are 15-digit randomly generated and coded by using the Java language. Factorization methods used in this study, in addition to the proposed new method, are Trial Division , Fermat, Elliptic Curve, Quadratic Sieve, Atkin Sieve and Polllar-Rho methods.

Keywords: RSA, Elliptic Curve, ECC, Discrete Logarithm, Cryptography, Sieving

1. GİRİŞ

Şifreleme algoritmaları genel olarak tek yönlü kolay ve tersi zor fonksiyonlar üzerinde çalışmaktadır. Burada dikkat edilmesi gereken önemli bir nokta, bu fonksiyonların tek yönlü olmamasıdır. Örneğin bir özetleme fonksiyonu (hashing function) tek yönde çalışması mümkünken geri dönüşü imkansızdır (Seker vd., 2014). Buna karşılık şifreleme algoritmaları, geri dönüşü mümkün ancak matematiksel olarak zor fonksiyonlar üzerine kuruludur. WPE, SSL gibi teknolojilerin (Brumley and Boneh, 2003) temelini oluşturan ve asimetrik şifreleme algoritmalarından birisi olan RSA algoritması da matematiksel güçlük olarak çarpanlara ayırma yöntemine dayanmaktadır (Seker ve Mert, 2013) . Bu yaklaşımda, iki sayının çarpılması kolay ancak işlemin tersten işlemesi yani çarpımın, çarpanlarına geri ayrılması güçtür. Örneğin saniyeler seviyesinde yapılan ve yüksek haneli iki sayının çarpımından çıkan sonucun, çarpanlarına ayrılması işlemi aylar mertebesinde vakit alabilmektedir. RSA algoritması, çarpma işlemi için iki adet büyük haneli asal sayı kullanmakta ve bu asal sayıların çarpımını herkese açık olan bir anahtar olarak yayınlamaktadır. Yani RSA algoritmasına saldırmak isteyen saldırganın elinde bu anahtar açıkça bulunmaktadır.

Bu makale kapsamında, çarpanlara ayırma zorluğuna bir saldırı olarak yeni bir yöntem önerisinde bulunulmaktadır. Ayrıca önerilen yeni yönteme ilave olarak, kullanılan 6 farklı çarpanlara ayırma yöntemine giriş mahiyetinde bilgi verilecek, birer örnek üzerinden çalışmaları gösterilecek ve kodlamaların rast gele üretilen sayılar kümesi üzerindeki yapılan performans ölçümleri karşılaştırılacaktır.

2. RSA Algoritması ve Çalışması

Bir açık anahtarlı şifreleme yöntemi olan RSA, 1977 yılında Ron Rives, Adi Shamir ve Leonard Aldeman tarafından bulunmuştur. Şifreleme yönteminin adı da bu üç kişinin soy isimlerinin baş harflerinden oluşur.

Çalışması:

1. Yeterince büyük iki adet asal sayı seçilir: Bu sayılar örneğimizde p ve q olsunlar.

2. n=pq hesaplanır. Buradaki n sayısı iki asal sayının çarpımıdır ve hem umumî hem de hususî şifreler için taban (modulus) olarak kabul eder.
3. Totient fonksiyonu hesaplanır. Bu örnek için çarpanların ikisi de asal sayı olduğu için φ(n) = (p-1)(q-1) olarak bulunur.
4. Hesaplanan totient fonksiyonu değeri (φ(n)) ile aralarında asal olan öyle bir e sayısı alınır ki 1 < e < φ(n) olmalıdır. Bu seçilen e sayısı umumî anahtar olarak ilan edilebilir.
5. d gibi bir sayı hesaplanır ki bu sayı için şu denklik geçerli olmalıdır: de ≡ 1 mod (φ(n)). Bu d değeri hususî şifre olarak saklanır. Bu sayının hesaplanması sırasında uzatılmış Öklid (Extended Euclid) algoritmasından faydalanılır.

Yukarıdaki şifreleme yönteminin en önemli dezavantajlarından birisi büyük asal sayılar bulmak aşamasında ortaya çıkar. Bilindiği üzere ele alınan bir sayının asal olup olmadığını bulmak kolay bir işlem değildir. Bunun için Fermat teoreminden yararlanılabilir.

Şifreleme işlemi:

Şifreleme işlemi için Alice kendi umumî şifresi olan (n,e) ikilisini yayınlar. Bu şifreyi alan Bob aşağıdaki şekilde mesajını şifreler:

c = me mod n

Burada m, şifrelenecek olan açık metin, e ve n ise Alice tarafından yayınlanan umumî şifredir.

Şifrenin Açılması:

Alice, Bob tarafından yollanmış olan mesajın açılması sırasında aşağıdaki formülü kullanır:

m = cd mod n

Burada açılacak olan şifrelenmiş metin c, Alice'in hususî şifresi ise d ile gösterilmiştir. n ise taban değeri olan modulus'tur.

Örnek:

- İki asal sayı seçilir

p = 61 ve q = 53

- n değeri hesaplanır n = pq şeklinde

n = 61 * 53 = 3233

- Totient fonksiyonu hesaplanır

φ(n) = (p-1)(q-1)

φ(n) = (61-1)(53-1) = 3120

- Totient fonksiyon sonucu ile aralarında asal olan ve 1 den büyük bir sayı seçilir

e > 1 => e = 17 (3120 ile aralarında asal) , bu sayı aynı zamanda umumî şifredir.

- Hususî şifre olması için bir d sayısı seçilir:

de ≡ 1 mod(n) olacak şekilde d sayısı bulunur, d = 2753 (çünkü 17 * 2753 = 46801 = 1 + 15 * 3120) Bu sayının hesaplanması sırasında uzatılmış Öklid (Extended Euclid) yöntemi kullanılmıştır.

- Örneğin mesaj olarak 123 gönderilecek olsun:

12317 mod 3233 = 855 olarak şifreli metin bulunur.

- açacak taraf için tersi işlem uygulanır:

8552753 mod 3233 = 123 şeklinde orijinal mesaj geri elde edilir.

3. MATERYAL VE METOT

2. Bölümde anlatılan RSA algoritmasına saldırı için çeşitli yöntemler geliştirilmiştir (Dubey, Ratan, Verma, & Saxena, 2014). Bu çalışma kapsamında, RSA algoritmasına karşı çarpanlara ayırma yöntemi şeklinde bilinen yöntem üzerinde durulacaktır(Lu, Zhang, & Lin, 2013).

3.1. Deneme Yöntemi

Deneme yöntemi ile çarpanlara ayırma direk metotlardan biridir ve çarpanlara ayırmak istenen *n* sayısının sıra ile kendisine kadar olan tüm asal sayılara bölünüp bölünmediğinin tek tek kontrol edilmesi esasına göre çalışır. Deneme yöntemi herhangi bir *n* sayısını kısmen veya tam olarak çarpanlarına ayırmak için kullanılan etkili ve basit bir metottur. Çarpanlarına ayrılmak istenen *n* sayısının büyük olmadığı durumlarda bu metodu kullanmak daha iyidir.

3.2. Fermat Yöntemi

Fermat'ın çarpanlara ayırma (Fermat factorisation) (McKee, 1999) yöntemi iki kare farkı elde etmeye dayanır. Basitçe, herhangi bir n sayısı şayet iki kare farkı şeklinde yazılabilirse $n = a^2 - b^2$,

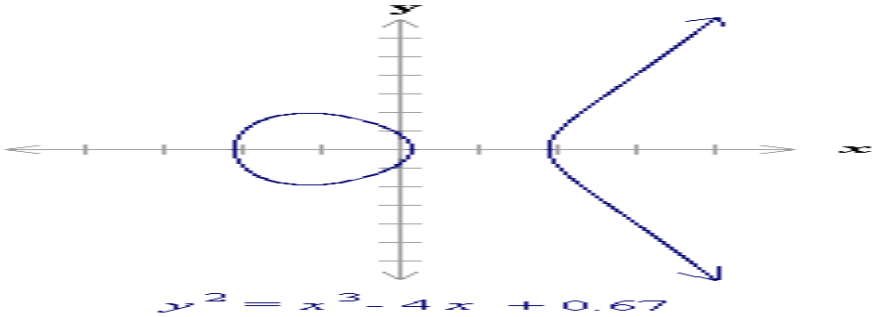
Bu durumda n sayısını veren çarpanlar $(a + b)(a - b)$ şeklinde bulunmuş olur.

Bu teoriyi ilerletirsek, n sayısının, $n = [(c + d) / 2]^2 - [(c - d) / 2]^2$ şeklinde yazılmasını sağlayan bir n=cd elde edilmiş olur. Yani ilk denklemde c = a + b ve d = a - b yazılacak olursa yukarıdaki bölümlerle elde edilen yeni sayılarda çarpan olarak elde edilebilir. Ancak çarpanlara ayırmak için genellikle ilk denklemden elde edilen (a + b) ve (a - b) çifti yeterli kabul edilebilir. Fermat yöntemi eğer çarpanlar birbirine yakınsa çok iyi çalışır ve tersi durumda da çok zayıf kalır.

Fermat teoremi özellikle şifreleme işlemleri sırasında çarpanlara ayırmaya dayalı zorluğa sahip RSA (Rivest, Shamir, & Adleman, 1978) gibi yöntemlere saldırı için kullanışlıdır.

3.3. Eliptik Eğri Yöntemi

Eliptik eğriler (Lenstra, 1987), şifreleme sistemlerinde oldukça yoğun ve güncel kullanım alanına sahiptir. Örneğin yapılan yeni çalışmalar klasik şifreleme algoritmalarına göre daha hızlı ve verimli olduğunu göstermiştir (Akben & Subaşı, 2005). Eliptik eğri (Eliptic curve), gerçek sayılar kümesi (real numbers) üzerinde tanımlanan ve $y^2 = x^3 + ax + b$, genel denklemini x ve y gerçek sayıları için sağlayan eğrinin ismidir. Bu genel denklem için her a ve b değeri farklı bir eğri verir. Örneğin a = -4 ve b = 0.67 değerleri için $y^2 = x^3 - 4x + 0.67$ denklemi elde edilir. Bu denklemin grafiği aşağıda verilmiştir:

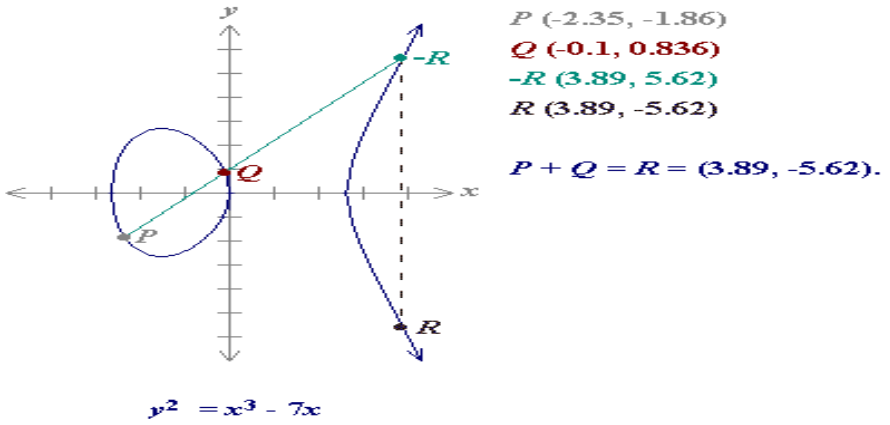


Şekil 1. *Örnek Eliptik Eğri*

Şayet $x^3 + ax + b$ genel denkleminin tekrarlı kökü yoksa veya diğer bir ifadeyle $4a^3 + 27b^2$ değeri 0 değilse, $y^2 = x^3 + ax + b$ genel denklemi için bir grup oluşturacağı söylenebilir. Eliptik bir grup ile kast edilen, eliptik eğrinin üzerinde tanımlı olan noktalardır ve bu noktalar öyle bir O noktasında sonsuza gider.

3.3.1. Eliptik Eğrilerde Toplama

Eliptik gruplar toplanabilir gruplardır. Bu grupların en temel fonksiyonu toplamadır. Eliptik bir eğri üzerinde iki nokta geometrik olarak tanımlanabilir. Örneğin P=(xP,yP) noktasını tanımlamak aşağıdaki şekilde olduğu üzere mümkündür. Eliptik eğrilerin bir özelliği de x eksenine göre simetrik eğriler oluşudur. Örneğin P noktasının simetriği ve dolayısıyla eksi değeri -P = (xP,-yP) olarak tanımlanır.

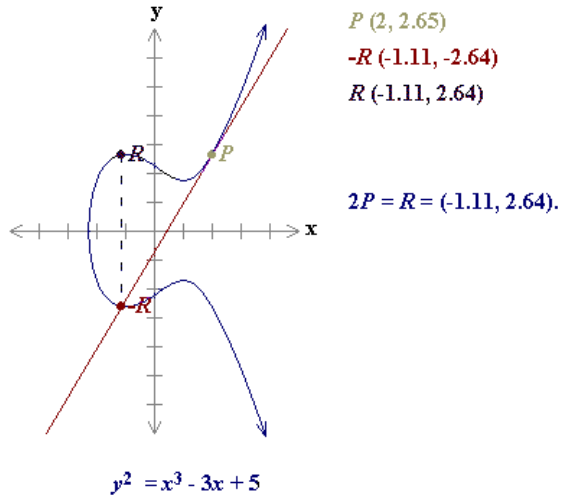


Şekil 2. *Örnek Eliptik Eğri üzerinde toplama işlemi*

Örneğin yukarıdaki şekilde gösterilen iki nokta olan P ve Q noktalarının toplamı şu şekilde alınır. P ve Q noktalarının ikisinden de geçen bir doğru çizilir (uzayda iki nokta bir doğru belirtir ve burada $Q \neq -P$ olmalıdır çünkü simetrik bir nokta alınması durumunda çizilen doğru y eksenine paralel olur).

Bu iki noktadan çizilen doğru, eliptik eğriyi 3. bir noktada kesmek zorundadır ve bu nokta -R olarak ifade edilirse P + Q = R ifadesi doğrudur.

Bir noktanın kendisi ile toplanması da eliptik eğrilerde mümkündür. Örneğin aşağıdaki eğride P noktasının değeri kendisi ile toplanmıştır.



Şekil 3. *Örnek Eliptik Eğri üzerinde toplama*

Yukarıdaki şekilde çizilen doğrunun yönü tek noktadan çıktığı için belirsizdir. Bu durum o noktadaki tanjant değeri alınarak çözülür. Başka bir değişle bir noktanın kendisi ile toplamı o noktadaki eğimin yönünde bir doğrunun yine eliptik eğri üzerindeki kestiği noktanın x eksenine göre tersini alarak bulunur.

Yukarıdaki P noktasının kendisi ile toplanması sırasında P noktasının y değeri 0′ dan farklı kabul edilmiştir. Ancak bir noktanın y değeri 0 olsa bile kendisi ile toplanması mümkündür. Bu özel durumda noktanın eğimi y eksenine paralel olacağı için eliptik eğriyi ikinci bir noktadan kesemeyecektir. Bu durumda P noktasının kendisi ile toplamı O olacaktır.

Şayet O değerine P noktası eklenecek olursa bu durumda sonuç yine P noktasına eşit olur. Bu durumda 3P = P, 4P = O, 5P = P olduğunu söylemek doğrudur.

3.3.2. Toplama İşleminin Cebirsel Gösterimi

Eliptik Eğrilerde Toplama işlemleri cebirsel olarak gösterilebilinir. Örneğin P = (xP,yP) ve Q = (xQ , yQ) noktaları birbirinin x eksenine göre tersi olmayan iki nokta ve P + Q = R

olsun. Bu R noktasının x ve y koordinatları $x_R = s^2 – x_P – x_Q$ ve $y_R = -y_P + s(x_P – x_R)$ şeklinde bulunur. Bu hesaplamada s değeri için $s = (y_P – y_Q) / (x_P – x_Q)$ işlemi yapılır. Eğer P = Q ve $y_P = 0$ ise, $s = (3x_P^2 + a) / (2y_P)$ denklemi için R noktasının koordinatları $x_R = s^2 – 2x_P$ ve $y_R = -y_P + s(x_P – x_R)$ olarak hesaplanır.

3.3.3. Eliptik Eğrilerin Şifrelemede Kullanımı

Eliptik eğriler şifreleme ve veri güvenliğinde tam değer vermeleri özelliği ile kullanışlıdırlar. Genelde gerçek sayı kümesinde çalışan fonksiyonlar yuvarlama veya belirsizlik durumlarından dolayı şifreleme sistemlerinde tercih edilmemektedir. Ancak eliptik eğriler burada bir alternatif olarak kullanılabilir (Certicom Corp. Elliptic Curves). Bu kullanım aşağıda anlatılmıştır:

Bir F_p grubu tanımlanırken 0 ile p-1 arasındaki tam sayılar kastedilir. Buradaki kasıt modulo p ile de ifade edilebilir. Örneğin F_{23} ifadesi ile 0 ile 22 arasındaki sayılar kastedilmiştir. Ayrıca bu kümede tanımlı olan herhangi bir işlem yine 0 ile 22 arasında bir sonuç üretebilir.

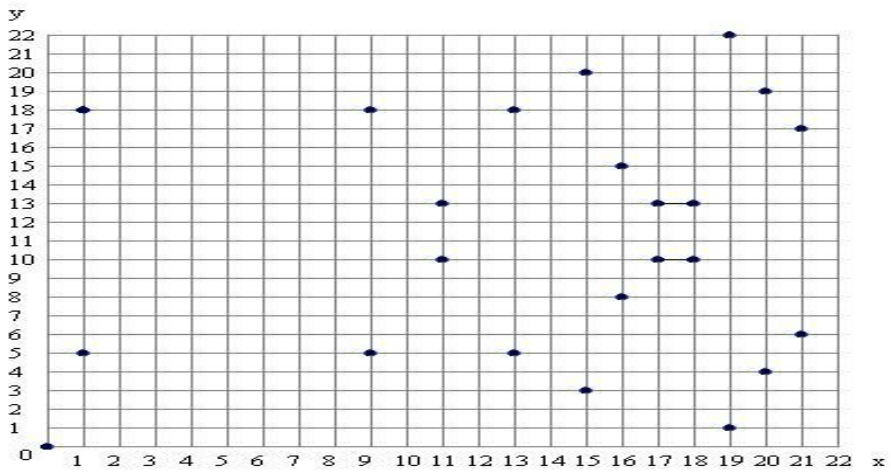
Bu durumda F_p grubunun üyesi olan her (x,y) ikilisi için yine F_p grubunda karşılık gelen bir sayı eliptik eğri üzerinde bulunabilir. Örneğin, F_{23} grubu üzerinde tanımlı olan sayılar, a=1 ve b=0 durumu için $y^2 = x^3 + x$ denklemi elde edilir. Burada (9,5) ikilisi denklemi aşağıdaki şekilde sağlar:

$y^2 \bmod p = x^3 + x \bmod p$
$25 \bmod 23 = 729 + 9 \bmod 23$
$25 \bmod 23 = 738 \bmod 23$
$2 = 2$

Denklemi sağlayan 23 nokta da aşağıda verilmiştir:

(0,0) (1,5) (1,18) (9,5) (9,18) (11,10) (11,13) (13,5)
(13,18) (15,3) (15,20) (16,8) (16,15) (17,10) (17,13) (18,10)
(18,13) (19,1) (19,22) (20,4) (20,19) (21,6) (21,17).

Bu noktaların koordinat sistemi üzerinde çizilmiş halleri Şekil 4 de verilmiştir. Her ne kadar rasgele dağılmış bir grafik gibi görülse de y=11,5 doğrusundan bir simetri vardır.



Şekil 4. *Ayrık Eliptik Eğri kullanımı*

F_p grubu üzerinde tanımlı eliptik eğriler ile gerçek sayılar kümesi üzerinde tanımlı eğriler arasındaki en önemli fark F_p grubundaki sayıların sonlu sayıda olmasıdır. Bu durum şifreleme için de tercih edilen bir özelliktir. Tam olarak bir eğri çizmenin ayrık noktalar için mümkün olmadığı dezavantajının yanında eliptik eğriler için geçerli olan bütün cebirsel kurallar F_p grubunda tanımlı eliptik eğriler için de geçerlidir. Ayrıca F_p grubundaki bütün sayılar tam sayı olduğu için gerçek sayılar kümesindeki yuvarlama ve belirsizlik durumu da söz konusu değildir. F_p ayrık grubu üzerinde iki noktanın toplanması işlemi yukarıda anlatılan işlemin birebir aynısıdır:

P + Q = R olarak ifade edilecek olursa, $s = (y_P - y_Q) / (x_P - x_Q) \bmod p$, değeri için

$x_R = s^2 - x_P - x_Q \bmod p$ ve $y_R = -y_P + s(x_P - x_R) \bmod p$ eşitlikleri yazılabilir.

Buradaki s değeri yukarıdaki durumun benzeri olarak P ve Q noktalarından geçen doğrunun eğimidir. Bir noktanın kendisi ile toplanması durumu da yukarıdaki gerçek sayılara benzer şekilde: 2P = R olarak gösterilsin. $s = (3x_P^2 + a) / (2y_P) \bmod p$, $x_R = s^2 - 2x_P \bmod p$ ve $y_R = -y_P + s(x_P - x_R) \bmod p$ dir.

Eliptik eğrilerin şifreleme sistemlerinde kullanılması aynı zamanda ayrık logaritma (discrete logarithm) hesaplamalarını da beraberinde getirmektedir. Buna göre bir sayının kendisi ile defalarca

kere toplanması bir ahenk sınıfı için (congruence class) dairesel bir grup (Cyclic group) oluşturur ve bu dairesel grubu kullanan algoritmalarda kullanılabilir (Trappe & Washington, 2006).

3.3.4. Eliptik Eğriler Üzerinde Ayrık Logaritma Kullanımı

Örneğin $y^2 = x^3 + 9x + 17$ şeklinde F_{23} için tanımlanan bir eliptik eğride, Q= (4,5) noktası için P = (16,5) noktasına göre ayrık logaritma bulmak için çözümlerinden birisi Q değerine ulaşıncaya kadar P noktasının kendisi ile toplanmasıdır. Bu işlem yapılırsa:

P = (16,5), 2P = (20,20), 3P = (14,14), 4P = (19,20) ,5P = (13,10), 6P = (7,3), 7P = (8,7), 8P = (12,17) ,9P = (4,5).

Dolayısıyla P tabanında Q noktasının logaritma değeri $\log_P(Q) = 9$ olarak bulunur.

Öncelikle sistemin çalışacağı bir eliptik eğri belirlenir. Bu eğrinin karakteristik denklemi aşağıdaki şekildedir:

$y^2 = x^3 + ax + b \pmod{n}$.

Ardından P = (x, y) şeklinde bir nokta belirlenir. Toplama işlemi kP = P + P + … + P şeklinde ilerletilerek ve k adet toplama işlemi yapılarak sonuçlarının ürettiği grup üzerinde, çarpanlarına ayrılmak istenen n sayısını tam bölen bir değer olup olmadığı aranır.

Bu ilerleme sırasında herhangi bir eP değeri (e kadar toplanmış P noktası) sonsuz çıkabilir.

Bu durumda yeni bir P noktası seçilerek ilerleme işleminin bu yeni P noktası üzerinden tekrarlanması gerekir.

Örnek

Çarpanlarına ayırmak istediğimiz sayı 455839 olsun. Eliptik eğri olarak aşağıdaki denklemi seçiyoruz:

$y^2 = x^3 + 5x - 5$

Ayrıca başlangıç noktası olarak bu eğri üzerinde bulunan P(1,1) değeri seçilir. ($1^2 = 1^3 + 5.1 - 5$ denklemi çözülerek 1 =1 olduğu ve eğri üzerine bulunan bir nokta olduğu sınanabilir)

Öncelikle P+ P = 2P değerini hesaplanır.

$s = (3x^2+5)/(2y) = (3 + 5) / 2 = 4$ (x ve y değerleri P(1,1) noktası için 1'dir)

$x' = s^2-2x = 14$ ve $y' = s(x-x')-y = 4(1-14)-1 = -53$ değerleri bulunarak 2P = (x', y') yani 2P = (14, -53) değeri bulunur.

Sonra 2P değeri için eğim (Slope veya s olarak geçen değer) bulunur.

$s=(3x^2+5)/(2y)=(3\ 14^2 + 5) / 2(-53) = -593 / 106$ olarak bulunur. Bulunan bu eğim değerinin paydası çarpanlardan birisini önermektedir. Buna göre bulunan eğim değerinin paydasının çarpanları aranan n değerini tam bölüp bölmediği kontrol edilir: gcd (455839, 106) = 1 olarak bulunur. Görüldüğü üzere sayı bölmediğinden ilerlemeye devam edilir ve bir sonraki nokta bulmaya çalışılır. Algoritma bundan sonraki adımlarda 3P, 4P, … şeklinde noktaları bulup bu noktaların eğimlerinin, çarpanları aranan n sayısını tam bölüp bölmediğine bakar.

Bu işlemlere devam edilirse, 8P değeri için s değerinin paydasının 599 olduğu görülür. Bu sayı 455839 sayısının çarpanlarından birisidir ve 455839 = 599 x 761 olarak bulunur.

3.4. İkinci Dereceden Kalbur Yöntemi (Quadratic Sieve)

İkinci derecenden kalbur (quadratic sieve) (Gerver, 1983) yöntemi özellikle veri güvenliği konusunda, çarpanlara ayırmaya dayanan zorluk üzerine inşa edilmiş olan şifreleme algoritmalarına bir saldırı için kullanılır.

Sistem kabaca bir sayıyı çarpanlarına ayırır. Bu ayırma işlemi aşağıdaki adımlardan oluşur.

Öncelikle n asal çarpanlarına ayırmak istediğimiz, bir asal olmayan sayı (bileşik sayı, composite number) olsun. İlk olarak asal çarpanlar için bir üst limit belirlenir.

Algoritma bu limitten küçük çarpanlar üzerinde çalışır. Bu çarpanlar kalbur (sieve) üzerinde eleme yapmak için kullanılır. En büyük asal sayı limiti b olsun (b sayısı asal sayıdır). Bu sayı P = { 2,3,5,7,11,13 … } asal sayılar kümesinde en büyük asal sayıyı belirtir.

Bu asal sayılar kümesindeki sayıların n sayısı ile legendre sembolü bulunur ve bu değer sayılar asal sayı olduğu zaman -1 veya 1 çıkar. Bu sonuçlardan 1 çıkaranları alınır ve -1 sonucu verenler elenir. Bu yeni küme B olsun.

Ardından kalbur (eleme, sieving) işlemi ile hedef aralık belirleyerek bu aralıktaki sayılar elenir. Hedef aralık olarak n sayısının karekköüne yakın değerler hesaplanır.

Bu değerlerden B kümesindeki sayılar ile çarpan şeklinde yazılabilenler alınır, geri kalanlar elenir. Yeni küme H olsun. Sonuçta elde edilen sayılardan iki sonuç elde edilir.

x = H kümesindeki sayıların çarpımı

y = h, H kümesinin her bir elemanı olmak üzere h^2-n şeklinde yazılan sayıların çarpımının karekökü

Sonuçta $x^2 \equiv y^2 \pmod{n}$ şeklinde bir denklem elde edilir ve bu denklem Fermat'ın çarpanlara ayırma yöntemine göre çözülmüş sayılır ve son adım olarak iki çarpan:

gcd(x-y, n) ve gcd (x+y, n) olarak elde edilir.

Örnek

Çarpanlarına ayırmak istediğimiz sayı n = 87463 ve b = 37 olsun.

Buna göre P = {2,3,5,7,11,13,17,19,23,29,31,37} şeklinde asal sayılar kümesini elde edilir.

Bu asal sayılar kümesinin her elemanı için legendre sembolü bulunur. Tablo 1 bu işlemin sonuçlarını göstermektedir.

Tablo 1. P asal sayılar kümesinin elemanlarının legendre sembolleri

P	2	3	5	7	11	13	17	19	23	29	31	37
(n/p)	1	1	-1	-1	-1	1	1	1	-1	1	-1	-1

Tablo 1' deki sayılardan 1 sonucunu verenler bırakılır ve -1 sonucunu verenler elenir. Bu durumda yeni küme B= {2,3,13,17,19,29} olur.

Eleme adımına başlamak için bir aralık belirlenir. Bunun için n=87463 sayısının karekökü alınır. Bu değer yaklaşık olarak 295 sonucunu verdiğinden aralık olarak 260 ile 330 sayı aralığını alabiliriz. Aranan sayılar B kümesindeki sayılar cinsinden çarpanlara ayrılabilen sayılardır. Bu sayılar: H = {265, 278, 296, 299, 307, 316} sayılarıdır. Bu kümedeki her bir h sayısı, h2-n şekline dönüştürülür ve B kümesindeki sayılar cinsinden çarpanlarına ayrılır. Tablo 2'de bu sonuçları göstermektedir.

Tablo 2. H kümesinin elemanlarının çarpan analizi

H	-1	2	3	13	17	19	29
265	1	1	1	0	1	0	0
278	1	0	1	1	0	0	1
269	0	0	0	0	1	0	0
299	0	1	1	0	1	1	0
307	0	1	0	1	0	0	1
316	0	0	0	0	1	0	0

Tablo 2' de, her satır ilgili kümedeki sayının çarpanlarıdır.

Örneğin 265 sayısı:
$265^2 = 70225$
$70225 - 87463 = -17238 = -1 \times 2 \times 3 \times 13^2 \times 17$

Bu durumda 265 sayısının çarpanları, Tablo 2'de görüldüğü gibi işaretlenir.

Yukarıda elde ettiğimiz Tablo 2'deki matrisin tersyüzünü alıp (transpose), sonucu 0 yapan vektör aranır (yöney, vector) .

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \end{pmatrix} \cdot \underline{v} = \underline{0}$$

Bu sonuca ulaşan birden fazla vektör bulunabilmektedir. Bunlardan bir tanesi aşağıda verilmiştir:

v = (1,1,1,0,1,0).

Ardından bulunan V vektöründe 1 değerinin bulunduğu sıradaki H kümesinin elemanları alınır ve 0'a tekabül eden sayılar elenir.

H = {265, 278, 296, 299, 307 , 316} kümesindeki 4. ve 6. sayılar, v vektöründe 0'a denk geldiklerinden elenirler ve yeni küme H = {265, 278, 296, 307 } olur.

Son olarak x ve y değerleri bu yeni küme üzerinden üretilir.

x sayısı, H kümesindeki elemanların çarpımından oluşur:

x = 265.278.296.307 = 6694540240 □ 34757 (mod n) ve y sayısı ise $y = \sqrt{\left(265^2 - n\right)\left(278^2 - n\right)\left(296^2 - n\right)\left(307^2 - n\right)}$ denklemi çözülerek y = 13497354 □ 28052 (mod n) elde edilir.

Bulunan x ve y değerleri gcd (x-y, n) ve gcd (x+y, n) ifadelerinde yerlerine yerleştirilerek n sayısının çarpanları bulunmuş olur .

İlk çarpan gcd(34757 – 28052, 87463) = gcd (6705, 87463) =149 ve ikinci çarpan gcd (34757 + 28052, 87463) = gcd (62809, 87463) =587 dir.

3.5. Atkin Kalburu (Eratosthene Sieve)

Belirli bir aralıkta verilen bütün asal sayıları bulmaya yarayan algoritmadır (Atkin & Bernstein, 2004). Bu algoritmada bir kalbur problemi olarak görülebilir ve daha önceden problemle uğraşmış olan Eratosten tarafından geliştirilen çözümün gelişmiş halidir.

Algoritmanın ismi, 2004 yılında bu yöntemi geliştiren kişiden gelmektedir.

Algoritma adımları aşağıda açıklanmıştır:

- Öncelikle bütün sayılar mod 60 ta çalışır. Yani sonuçlar 60'a bölümden kalan olarak değerlendirilir.
- Sistemdeki bütün sayılar (x ve y dahil olmak üzere) pozitif tam sayılardır.
- Sistemdeki sayılar asal veya değil (bu yazıda a ve d harfleri kullanılacaktır) olarak işaretlenebilir.
- Kalbur listesindeki bir işaretin tersinin alınması a->d veya d->a dönüşümünün yapılması demektir.
- Listenin ilk 3 elemanı, 2,3 ve 5 sayılarıdır.

Asal sayıların bulunması istenen aralıktaki bütün sayıların bulunduğu bir liste oluşturulur ve ilk başta bütün sayılar, asal değil anlamında d olarak işaretlenir.

Listedeki her sayının sırayla mod 60 sonucu bulunur. Sırayla bulunan değere n denir,

- Bu değerin {1,13,17,29,37,41,49,53} küme elemanlarından birisi olması halinde $4x^2+y^2 = n$ sonucunu veren bütün çözümler ters çevrilir.
- Kalan değeri {7,19,31,43} kümesinde ise $3x^2 + y^2 = n$ sonucu veren bütün çözümler ters çevrilir.
- Kalan değeri {11,23,47,59} kümesinde ise $3x^2 – y^2 = n$ sonucu veren bütün çözümler ters çevrilir.

Yukarıdaki algoritmada bulunan kümelerin bir özelliği vardır. İlk kümedeki değerler, mod 12 için 1 veya 5 veren değerlerdir. İkinci küme mod 12 için 7 ve üçüncü küme mod 12 için 11 sonucunu veren değerlerdir.

Örnek

40'a kadar olan asal sayıların yukarıdaki yöntemle bulunuşu:

x = 1 ve y = 1 durumu ile başlanır ve bu değerler denklemde yerlerine yazılır.

$4x^2+y^2 = n$ denklemi için n = 5 bulunur. Bu değer mod 12'de 5'tir ve ilk kümeye girer. Dolayısıyla 5 için asal işaretlemesi yapılır.

$3x^2 + y^2 = n$ denklemi için n = 4 bulunur. Bu değer mod 12'de 7 olmadığı için bir işlem yapılmaz.

$3x^2 – y^2 = n$ denkleminde n = 2 bulunur ve mod 12'de 11 olmadığı için bir işlem yapılmaz.

Yukarıdaki sorgulama işlemi x ve y değerleri arttırılarak tekrarlanır.

x=1 ve y = 2 için işlemler yapıldığında sadece ikinci denklem sağlanır:

$3x^2 + y^2 = n$ denklemi için n = 7 bulunur. Bu değer de mod 12'de 7 olduğu için 7 sayısı asal olarak işaretlenir.

Geri kalan sayılar için algoritmanın çalışması aşağıda Tablo 3'de verilmiştir.

Tablo 3. Atkin Kalburu ile 40 kadar olan asal sayıların buluşu

x	y	$4x^2+y^2$	$4x^2+y^2$ mod 12	$3x^2+y^2$	$3x^2+y^2$ mod 12	$3x^2-y^2$	$3x^2-y^2$ mod 12
1	1	5	5	4	4	2	2
1	2	8	8	7	7	-1	-1
1	3	13	1	12	0	-6	-6
1	4	20	8	19	7	-13	-1
1	5	29	5	28	4	-22	-10
1	6	40	4	39	3	-33	-9
2	1	17	5	13	1	11	11
2	2	20	8	16	4	8	8
2	3	25	1	21	9	3	3
2	4	32	8	28	4	-4	-4
2	5	41	5	37	1	-13	-1
2	6	52	4	48	0	-24	0
3	1	37	1	28	4	26	2
3	2	40	4	31	7	23	11
3	3	45	9	36	0	18	6
3	4	52	4	43	7	11	11
3	5	61	1	52	4	2	2
3	6	72	0	63	3	-9	-9
4	1	65	5	49	1	47	11
4	2	68	8	52	4	44	8
4	3	73	1	57	9	39	3
4	4	80	8	64	4	32	8
4	5	89	5	73	1	23	11
4	6	100	4	84	0	12	0
5	1	101	5	76	4	74	2
5	2	104	8	79	7	71	11
5	3	109	1	84	0	66	6
5	4	116	8	91	7	59	11
5	5	125	5	100	4	50	2
5	6	136	4	111	3	39	3
6	1	145	1	109	1	107	11
6	2	148	4	112	4	104	8
6	3	153	9	117	9	99	3
6	4	160	4	124	4	92	8
6	5	169	1	133	1	83	11
6	6	180	0	144	0	72	0

Tablo 3'de renklendirilen değerler asal sayılardır ve bu değerlerin oluşturduğu küme aşağıdaki şekildedir:

{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37}.

Yukarıdaki kümede görülen bu değerler, aynı zamanda 2-40 arasındaki asal sayılar kümesidir.

3.6. Pollard Rho Yöntemi

Pollard Rho çarpanlara ayırma yöntemi (factorization) (Pollard, 1975), büyük asal sayıların hızlı bir şekilde çarpanlara ayırılmasını amaçlamaktadır. Veri güvenliği (kriptoloji) açısından oldukça önemli olan bu yöntemin çalışması aşağıdaki adımlardan oluşur:

- Çarpanlarına ayrılmak istenen sayın n olsun.
- Bulanacak çarpan d olarak isimlendirilir ve d sayısı d | n şartını sağlayacaktır (tam bölecektir)
- Algoritma sırasında kullanılacak iki değişken olan a ve b değerlerine 2 atayarak başlanır. $a \leftarrow 2$, $b \leftarrow 2$
- Bir döngü içerisinde, sonucu bulana kadar aşağıdaki adımlar takip edilir:
- $a \leftarrow a^2+1 \bmod n$, $b \leftarrow b^2+1 \bmod n$, $b \leftarrow b^2+1 \bmod n$
- Yukarıdaki satırdan görüldüğü üzere a sayısının 1 kere karesi alınıp arttırılırken, b sayısı iki kere aynı fonksiyona tabi tutulmaktadır. Dolayısıyla b sayısı, a'ya göre daha hızlı ilerlemektedir.
- $d = \gcd(a-b, n)$ değeri hesaplanır.
- Şayet $1<d<n$ şartını sağlayan bir d değeri bulunuyorsa başarıyla tamamlanmıştır ve işlem bitirilir.
- Şayet d=n durumu oluşursa algoritmayı başarısız bir şekilde sonlandırılır ve daha fazla devam edilmez.
- Yukarıdaki iki durum dışında döngüye devam edilir.

Örnek

Çarpanlarına ayırmak istediğimiz sayı 187 olsun.
n ← 187
$a \leftarrow 2$, $b \leftarrow 2$

$a \leftarrow 2^2+1 \bmod 187$, $b \leftarrow 2^2+1 \bmod 187$, $b \leftarrow 5^2+1 \bmod 187$ *(bu satırdan sonra a = 5, b = 26 olur.)*

$d = \gcd(a-b, n)$, d= gcd (5-25, 187) =1, dolayısıyla henüz sonuç bulunmadığından işleme devam edilir. Bu adım ve sonraki adımlar aşağıdaki Tablo 4'de gösterilmiştir.

Tablo 4. *Pollard Rho Yöntemi ile 187 sayısının çarpanlarına ayrılması*

a	b	a-b	gcd(a-b,n)
$2^2+1 \bmod 187$	$(2^2+1)^2+1 \bmod 187=26$	-21	1
$5^2+1 \bmod 187$=26	$(26^2+1)^2+1 \bmod 187=180$	-154	11

Tablo 4'de görüldüğü üzere sayının çarpanlarından birisinin 11 olduğu bulunmuştur.

Örnek

Çarpanlara ayırmak istediğimiz sayı n = 87463 olsun.

Tablo 5. Pollard Rho Yöntemi ile 87463 sayısının çarpanlarına ayrılması.

A	b	a-b	d
5	26	-21	1
26	21015	-20989	-1
677	21634	-20957	1
21015	5536	15479	1
29539	26558	2981	1
21634	4917	16717	1
15444	65711	-50267	-1
5536	-48077	53613	1
35247	-23506	58753	1
26558	79715	-53157	-1
25733	38437	-12704	-1
4917	-21802	26719	1
37102	-11916	49018	1
65711	33219	32492	1
52920	78001	-25081	1
-48077	20372	-68449	1
-83452	57576	-141028	-1
-23506	78754	-102260	-1
28266	12329	15937	1
79715	-22519	102234	1
22669	-37328	59997	1
38437	72544	-34107	1
65437	-41348	106785	1
-21802	43017	-64819	-1
53263	19222	34041	1
-11916	45450	-57366	1
38608	31967	6641	1
33219	-63715	96934	1
68754	44914	23840	149

Sonuç 149 olarak bulunmuştur.

3.7. Ters Kalbur Çarpanlara Ayırma Ağacı

Bu çalışma kapsamında önerilen yeni bir çarpanlara ayırma yöntemidir. Basitçe kalbur (sieving) yöntemlerinin bir ağaç yapısı üzerinde inşa edilmesi ile sağlanır.

Eratosten kalburu, oransal elek (rational sieving) veya Atin Kalburu (Sieve of Atking) gibi klasik kalbur ile çarpanlara ayırma yöntemleri, düşük sayılardan başlayarak işlem yapmaktadır. Bu durumun genel olarak sayıların çarpanlara ayrılmasında avantaj sağladığı kesindir. Çünkü büyük ve asal olmayan sayıların test edilerek başlanması hem bölme işleminin karmaşıklığı artacağı hem de zaten daha küçük asal sayılara bölünmesi halinde testini gereksiz kılacağı için anlamsızdır. Daha basit bir ifadeyle, örneğin bir sayının 6'ya bölündüğünü test etmek, hem 2 hem de 3'e bölündüğünü test etmek demektir. Bir sayı 6'ya bölünebiliyor mu diye test etmek yerine, 2'ye bölündüğünü test etmek, 2 ve 2'nin bütün katlarını test etmek anlamına geleceği için performans açısından daha avantajlıdır.

Bu avantaj p ve p'nin bir katı olan k gibi iki sayı arasında karşılaştırılacak olursa, basitçe n adet sayıdan oluşan N kümesi için kaç farklı bölen olduğunu bulmaya çalışalım. Öncelikle m+1 adet çarpanı olan k değerini yeniden yazacak olursak:

$$k = p \prod_{i=1}^{m} c_i$$

Bu değere göre n adet farklı sayı için aşağıdaki durum söylenebilir:

$$(n_i \in N \ \wedge \ n_i | k) \Leftrightarrow n_i | \left\{ \mathbb{Z}_{|p} \ \cap \ \left(\bigcap_{i=1}^{m} \mathbb{Z}_{|c_i} \right) \right\}$$

Yani, N kümesinin herhangi bir elemanı için $n_i | k$ sağlanması, aslında p sayısı tarafından bölünebilen tam sayılar $(\mathbb{Z}_{|p})$ ile diğer m adet farklı çarpanın tam bölebildiği sayılar kümesinin kesişimi olarak düşünülebilir.

Diğer bir deyişle, bölüm sınamasında bu sayılardan birisi tarafından tam bölünebilmesi n_i sayısının asal olmadığını göstermek için yeterlidir.

Ancak, genelde üretilen çok büyük iki asal sayının çarpımından oluşan anahtar niteliğindeki sayının çarpanları, tanımı itibariyle bu elemeler sırasında en son bulunacak değerlerdir. Ayrıca istatistiksel olarak iki asal sayının çarpımından üretilen anahtarın genelde çarpanı olan iki asal sayı birbirine yakın ve hatta çoğu zaman aynı hane sayısına sahiptir. Bunun sebebi, güvenliğin aslında küçük sayı kadar olduğudur. Yani saldırgan sayıyı çarpanlarına ayırmak istediği sırada, çarpanlardan herhangi birisini bulması yeterlidir. Doğal olarak sayılardan birisi küçük olursa bulunması da daha kolay olacaktır.

Ayrıca, Fermat teorisinin ikinci kuralına göre, bir sayının çarpanı en fazla karekökü kadar olabilir diyebiliriz. Bu durumda bizim önerimiz sayıların çarpanlarının aranmasına kareköründen başlanarak daha düşük sayılara doğru ilerlenmesidir.

Ayrıca asallığı test edilen sayıların elenmesi ve bir kalbur oluşturulması sayesinde asal sayının aranması sırasında hız kazanılabilir.

Durumu daha basit bir şekilde ifade edecek olursak, 3 haneli bir sayının çarpanları, 1 ve 3 haneli iki sayı olabileceği gibi 2 ve 2 haneli iki sayı da olabilir. Bizim faydalandığımız durum, şifreleme sistemlerinde kullanılan ve iki asal sayının çarpımından üretilen anahtarların 1 ve 3 haneli sayılar yerine 2 ve 2 haneli sayılar olduğudur. Aksi durumda bir güvenlik zafiyeti oluşacağı kesindir. O halde bizim iddiamız, önceliği bu 2 ve 2 haneli sayılara vermek yönündedir.

Örneğin 47 x 53 = 2491 sayısını ele alalım. Bu sayının çarpanlarını aramaya en küçük asal sayı olan 2 sayısından başlanması halinde gerçek çarpanı olan sayılara ulaşılana kadar neredeyse bütün ihtimaller denenecektir. Bunun yerine bizim önerimiz, $\sqrt[3]{2491} = 49$ sayısından başlanmasıdır. Ve sayıların azaltılarak asallığının denenmesidir. Bu sayede örnekte de görüleceği üzere ikinci denemede asal çarpanı bulunacaktır.

Ayrıca bu yönteme ilave olarak kalbur elemesi sayesinde çarpanların çarpanlarının da test edilerek listeden elenmesi sağlanabilir. Örneğin ilk denenen sayı olan 49 için $49 = 7^2$ olarak yazılabileceği için aslında 49 yerine 7'nin katlarının test edilmesi ve ihtimal listesinden 7 ve

49'un kaldırılması önerilmektedir. Bu durumda ihtimal kümesi Tablo 6'daki gibi olacaktır:

Tablo 6. *İhtimal Kümesi*

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

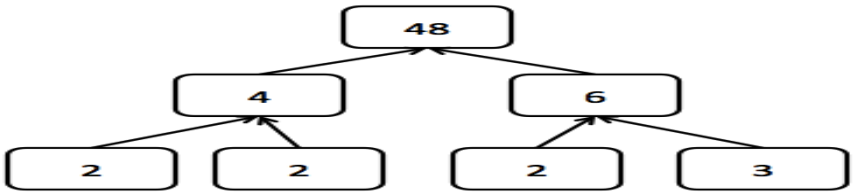
Tablo 6'da, elenen 7'nin katları işaretlenmiştir. Benzer şekilde, bir önceki sayı için de $48 = 2^3 \times 3$ yazılabileceği için ihtimal kümesinden 2 ve 3'ün bütün katlarının silinmesi önerilmektedir.

Tablo 7. *İlerletilmiş İhtimal Kümesi*

1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

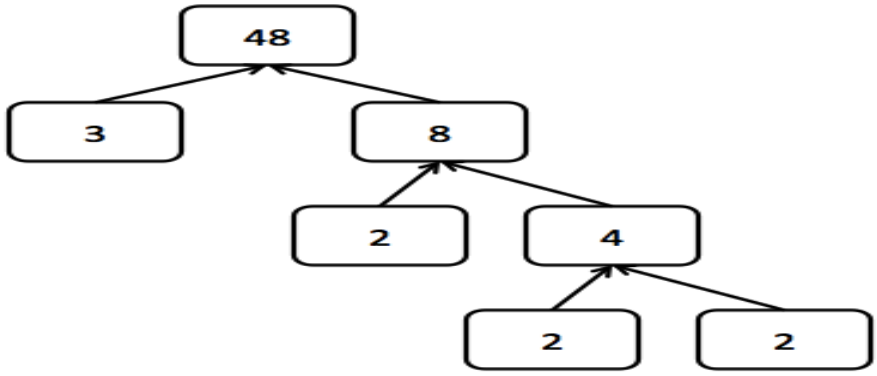
Tablo 7'deki ihtimal kümesinden görüleceği üzere, 3. adımda ihtimal kümesinde, 49 sayıdan sadece 14 ihtimal kalmıştır. Şayet 47 sayısının bir çarpan olduğunu tespit edemeyerek denemeye devam edecek olsaydık bir sonraki denenen sayı 43 olacaktı, yani daha önceki denemelerimiz sırasında 46, 45 ve 44 sayılarını ihtimal kümesinden çıkartmak burada bir avantaj sağlayacaktı.

Bu sayıların çıkarılması sırasında, bir çarpanlara ayırma ağacından faydalanılabilir. Örneğin 48 sayısının çarpanlara ayırma ağacı Şekil 5'de verilmiştir:



Şekil 5. *Örnek Çarpanlara Ayırma Ağacı*

Çarpanlara ayırma ağacı (factorization tree), tanımı itibariyle muğlaktır (ambigious) ve örneğin, Şekil 6'da olduğu gibi aynı sayı için farklı çarpanlara ayırma ağaçlarının çizilmesi mümkündür.



Şekil 6. *Örnek Çarpanlara Ayırma Ağacı*

Ancak, ağacın derinliği, en küçük asal sayı 2 olduğu için en fazla $\log_2 n$ boyutunda olacağından, ağacın hafıza ve işlem hızının yaprak düğümler çıktığında $\log_2 \frac{n}{2} - 1$ olacağını söyleyebiliriz.

Lütfen, en uzun ikili ağacın (binary tree) iç düğüm sayısının (internal node) toplam düğüm sayısının yarısından bir eksik olacağını hatırlayınız.

Bu durum yukarıdaki, özel olarak seçilmiş örnekte 3. adımda sonucu bulmaktadır ancak her sayı için bu kadar şanslı olamayacağımıza göre gerçekte sağladığı avantajı, nümerik olarak üretilen bir veri kümesi üzerinde test ederek göstermeye çalışalım. Öncelikle, eşit hane sayısına sahip asal sayılar üretiyor, ardından da bu asal sayıları ikili olarak çarparak bir deneme kümesi oluşturuyoruz. Ardından, bu küme üzerinden yeni yöntemimizi ve mevcut çarpanlara ayırma yöntemlerini deneyerek süre karşılaştırması yapıyoruz.

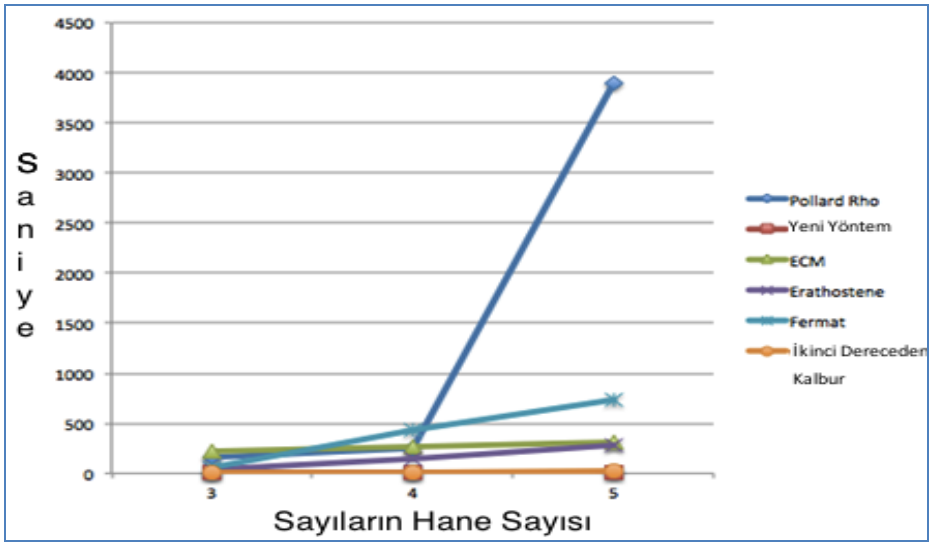
4. BULGULAR

Algoritmaların çalışma sonuçları aşağıdaki tabloda gösterilmiştir.

Tablo 8. Çarpanlara ayırma algoritmalarının performansları.

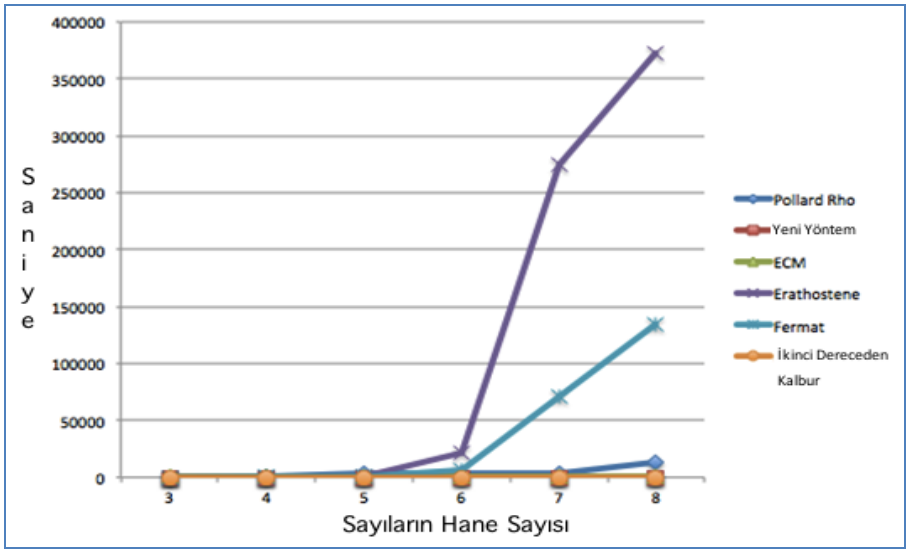
Metot	Ortalama Yürütme
Pollard Rho	398 dk
Eliptik Eğri	3443 dk
Fermat	30 dk
İkinci dereceden Kalbur	326 dk
Erathostene	1267052 dk
Deneme	5510739 dk
Yeni Yöntem	5 dk

Tablo 8'deki sonuçlar 8 hane uzunluğundaki bin adet farklı sayının çarpanlarına ayrılması sırasında elde edilmiştir.
Ayrıca, zamandaki değişimi göstermek için, algoritmaların çalışma süreleri, şekil 7'de görsel olarak sunulmuştur.



Şekil 7. *Sayı Haneleri Arttıkça Performanstaki Değişim*

Şekil 7'de gösterilen hane sayıları oldukça azdır (3-5 haneler arası), bu yüzden algoritmaların arasındaki fark net olarak görüntülenememektedir. Hane sayısının daha fazla arttırıldığı performans grafiği Şekil 8'de sunulmuştur.



Şekil 8. Hane Sayıları Daha da Arttırılmış Performans Grafiği

Tablo 9. Algoritmaların Zaman Karmaşıklıkları.

Metot	Zaman Karmaşıklığı	
Pollard Rho	O(B × log B × log2n)	B sınır ve n bileşik sayı olmak üzere.
ECM	O(L(p)M(log n))	M(log n) değeri çarpma işleminin mod n üzerindeki karmaşıklığı ve $L(p) = e^{c(\log p)^{\alpha}(\log\log p)^{1-\alpha}}$
Fermat	O(d)	d sayısı iki çarpanın birbirine olan mesafesidir.
Quadratic Sieve	O(log B loglog B)	B sınır değeridir.
Erathostene	$O(\sqrt{n} + p)$	p sayısı, $\sqrt{n}$ değerinin altındaki asal sayıların sayısıdır.
Trial Division	$O(\sqrt{n})$	
Yeni Yaklaşım	$O(d_p)$	Bileşik sayının iki çarpanı arasındaki asal sayıların sayısı d_p olarak gösterilmiştir.

Bazı algoritmaların ihtiyaç duyduğu geçiş süresi ve işlem süresi, diğer algoritmalara göre daha fazla olmaktadır. Yukarıda nümerik

sonuçları verilen algoritma karşılaştırmalarının analitik karşılaştırması ayrıca Tablo 9'da sunulmuştur.

5. SONUÇLAR VE TARTIŞMA

Karşılaştırılan yöntemler ve önerilen yeni yöntem son bölümde incelendiğinde, yeni yöntemin mevcut yöntemlere karşı bir avantajının olduğu görülmektedir. Gerek analitik gerek nümerik çalışmalar, yeni algoritmayı desteklemekte olup, yeni yöntemde kullanılan iki temel unsur, yani çarpan ağacı ve kalbur yöntemleri de ayrıca gelişmiş çoğu çarpanlara ayırma yöntemine göre daha basit bir kodlama imkanı sunmaktadır.

6. KAYNAKLAR

Akben, S. B., & Subaşı, A. 2005. RSA ve Eliptik Eğri Algoritmasının Performans Karşılaştırması, KSÜ Fen ve Mühendislik Dergisi , 8 (1), 35-40.

Atkin, A. O., & Bernstein, D. J. 2004. Prime sieves using binary quadratic forms, Math. Comp , 73, 1023-1030.

Brumley, D., & Boneh, D. 2003. Remote Timing Attacks are Practical. SSYM'03 Proceedings of the 12th conference on USENIX Security Symposium.

Certicom Corp. Elliptic Curves . (n.d.). Retrieved from http://www.certicom.com/index.php/ecc-tutorial

Dubey, M. K., Ratan, R., Verma, N., & Saxena, P. K. 2014. Cryptanalytic Attacks and Countermeasures on RSA. Proceedings of the Third International Conference on Soft Computing for Problem Solving Advances in Intelligent Systems and Computing, 258, pp. 805-819.

Gerver, J. 1983. Factoring Large Numbers with a Quadratic Sieve, Math. Comput. , 41, 287-294.

Lenstra, H. W. 1987. Factoring Integers with Elliptic Curves. Ann. of Math. , 126, 649-673.

Lu, Y., Zhang, R., & Lin, D. 2013. Factoring Multi-power RSA Modulus N = p r q with Partial Known Bits. Information

Security and Privacy Lecture Notes in Computer Science. 7959, pp. 57-71. Springer.

McKee, J. 1999. Speeding Fermat's Factoring Method, Mathematics of Computation , 1729–1738.

Pollard, J. M. 1975. A Monte Carlo method for factorization, BIT Numerical Mathematics , 15 (3), 331-334.

Rivest, R., Shamir, A., & Adleman, L. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM 21 , 2, 120–126.

Seker, S. E., & Mert, C. 2013. Reverse Factorization and Comparison of Factorization Algorithms in Attack to RSA. ISCIM'13 Proceedings of the International Conference on Scientific Computing, (pp. 881-887). Tiranna, Albania.

Seker S. E., Altun, O., Ayan, U., & Mert, C. 2014. A Novel String Distance Function based on Most Frequent K Characters. International Journal of Machine Learning and Computation (IJMLC) , 4 (2), 177-183.

Trappe, W., & Washington, L. C. 2006. Introduction to Cryptography with Coding Theory. Pearson Prentice Hall.

****