T.C.
NECMETTİN ERBAKAN NİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KRİPTOGRAFİDE KULLANILAN ASAL SAYI TEST YÖNTEMLERİ ÜZERİNE BİR ÇALIŞMA

Fatma ÇETİN

YÜKSEK LİSANS TEZİ

Matematik Anabilim Dalı

Aralık-2021
KONYA

TEZ KABUL VE ONAYI

Fatma ÇETİN tarafından hazırlanan "Kriptografide Kullanılan Asal Sayı Test Yöntemleri Üzerine Bir Çalışma" adlı tez çalışması …/…/… tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı'nda YÜKSEK LİSANS olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Başkan Doç. Dr. Oğuz YAYLA	……..
Danışman Dr. Öğr. Üyesi Ahmet SINAK	……..
Üye Doç. Dr. Yasin ASAR	……..

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …./…/20.. gün ve …….. sayılı kararıyla onaylanmıştır.

Prof. Dr. İbrahim Kalaycı
FBE Müdürü

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Fatma ÇETİN
Tarih: 10/12/2021

ÖZET

YÜKSEK LİSANS TEZİ

KRİPTOGRAFİDE KULLANILAN ASAL SAYI TEST YÖNTEMLERİ ÜZERİNE BİR ÇALIŞMA

Fatma ÇETİN

Necmettin Erbakan Üniversitesi Fen Bilimleri Enstitüsü
Matematik Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ahmet SINAK

2021, 83 Sayfa

Jüri
Doç. Dr. Yasin ASAR
Doç. Dr. Oğuz YAYLA
Dr. Öğr. Üyesi Ahmet SINAK

Matematikte zor problem olarak kabul edilen problemlerden Çarpanlara Ayırma Probleminin zorluğu verilen bileşik sayının asal çarpanlarının büyüklüğüne bağlıdır. Daha açık bir ifadeyle, bu problemin zorluğu üzerine dayanan kriptosistemin güvenilir olabilmesi için çarpan olarak kullanılan asal sayıların yeteri kadar büyük olması gerekmektedir. Bu durumda yeteri kadar büyüklükte asal sayı üretme problemi karşımıza çıkmaktadır. Literatürde, büyük sayıların asal olup olmadığını belirlemek için çeşitli asallık testleri önerilmiştir ve bazıları günümüzde pratik olarak kullanılmaktadır. Bu tez çalışmasında literatürde yer alan olası asallık testleri ve kesin asallık testleri ayrıntılı olarak incelenmiş ve örneklerle desteklenmiştir. Olası asallık testlerinin çalışma zamanları esas alınarak performans analizleri yapılmış ve karşılaştırmaları sayısal verilerle sunulmuştur. Olası asallık test algoritmalarının hata oranları ve karmaşıklıkları verilerek bir karşılaştırma sunulmuştur. Ek olarak bu tez çalışmasında, güvenliği çarpanlara ayırma probleminin zorluğuna dayanan ve günümüzde pratik kullanımda çok önemli bir yere sahip olan RSA şifreleme algoritması ayrıntılı olarak incelenmiştir. Bu tezin ekler bölümünde, tezde verilen bazı algoritmaların ve asallık testlerinin C++ programlama dilindeki kodları sunulmuştur.

Anahtar Kelimeler: Asal sayılar, Asallık testleri, Kriptoloji, Şifreleme, Şifre çözme, RSA algoritması

ABSTRACT

MS THESIS

A STUDY ON PRIME NUMBER TEST METHODS USED IN CRYPTOGRAPHY

Fatma ÇETİN

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCE OF NECMETTİN ERBAKAN UNIVERSITY
THE DEGREE OF MASTER OF SCIENCE
IN MATHEMATICS

Advisor: Asst. Prof. Dr. Ahmet SINAK

2021, 83 Pages

Jury
Assoc. Prof. Dr. Yasin ASAR
Assoc. Prof. Dr. Oğuz YAYLA
Asst. Prof. Dr. Ahmet SINAK

The hardness of the Integer Factorization Problem that is considered as one of the hard problems in mathematics depends on the sizes of the prime factors of the given odd composite number. More precisely, the prime factors must be large enough so that cryptosystem whose security is based on the difficulty of this problem can be reliable. In this case, we encounter the problem of finding sufficiently large prime numbers. In the literature, to determine whether large odd numbers are prime numbers, primality test algorithms have been proposed, and some of them have been applied in practical life. In this thesis, the probabilistic and deterministic primality test algorithms are studied in detail, and several concrete examples are presented. The performance analyses of the probabilistic primality tests are performed by considering the running times. Moreover, a comparison of these tests is presented according to their error probability and time complexities. In addition, we investigate the RSA encryption algorithm whose reliability is based on the hardness of the integer factorization problem and which has a significant role in practical use today. The appendix provides the implementation codes of some algorithms and primality tests in the C++ programming language.

Keywords: Prime numbers, Primality tests, Cryptology, Encryption, Decryption, RSA algorithm

İÇİNDEKİLER

ÖNSÖZ

Tez çalışmam sırasında değerli bilgi, birikim ve görüşlerinden faydalandığım, bana yol gösteren, geliştiren ve her zaman destek olan değerli danışman hocam sayın Dr. Öğr. Üyesi Ahmet SINAK'a sonsuz teşekkür ve saygılarımı sunarım.

Lisans ve yüksek lisans eğitimlerim boyunca bilgi, birikim ve deneyimlerini her zaman benimle paylaşan, çalışmanın gerçekleştirilmesindeki destek ve katkılarından dolayı Necmettin Erbakan Üniversitesi Fen Fakültesi Matematik-Bilgisayar Bölümü hocalarıma ayrıca teşekkürü bir borç bilirim.

Son olarak çalışmalarım boyunca maddi, manevi destekleriyle beni yalnız bırakmayan, daima destek olup varlıklarıyla bana güç veren anneme ve babama sonsuz saygı ve minnetlerimi sunarım.

Fatma ÇETİN
KONYA-2021

SİMGELER VE KISALTMALAR

Simgeler

$\mathbb{R}$: Reel Sayılar Kümesi
$\mathbb{Z}$: Tam Sayılar Kümesi
$\mathbb{Z}^+$: Pozitif Tam Sayılar Kümesi
$\mathbb{Z}_n$: $\mathrm{mod}\, n$ bağıntısına göre denklik sınıfları kümesi
$a|b$: a sayısı b sayısını böler
obeb (a, b) : a ve b sayılarının en büyük ortak böleni
$\varphi\ (n)$: Euler Totient Fonksiyonu
$\left(\frac{a}{n}\right)$: a ve n sayıları için Jacobi sembolü
F_n : n-inci Fermat sayısı
M_p : asal p sayısı için Mersenne asal sayısı
$\cdot$: Standart çarpma işlemi
$E_n(t)$: n sayısı için t parametresine bağlı hata oranı

Kısaltmalar

AKS : Agrawal-Kayal-Saxena
CRT : Chinese Remainder Theorem
ÇKT : Çin Kalan Teoremi
DES : Data-Encryption-Standart
M&R : Miller-Rabin
OAEP : Optimal Asymmetric Encryption Padding
OAT : Olasılık Asallık Testleri
PKCS : Public-Key Cryptography Standards
RSA : Rivest-Shamir-Adleman
SSL : Secure Socket Layer
S-HTTP : Secure Hyper Text Transfer Protocol
S-MIME : Secure/Multipurpose Internet Mail Extensions
S/WAN : Secure/Wide Area Network

1. GİRİŞ

Günümüzün önemli gereksinimlerinden birisi bilgilerin (verilerin) güvenli olarak taşınması ve saklanmasıdır. İletişimde verilerin güvenli bir şekilde alıcıya gönderilmesi sadece kriptografi (şifreleme) bilimi sayesinde mümkündür.

Kriptografi bilimi bilginin güvenliğini sağlamak için tasarlanan belirli güvenlik kriterlerini sağlayan birçok şifreleme algoritmasından oluşmaktadır. Bu algoritmaların tasarımı bazı matematiksel yöntemlerden oluşmaktadır. Özellikle açık anahtarlı şifreleme algoritmalarının güvenliği bazı matematiksel problemlerin zorluğuna dayanmaktadır.

Matematikte zor problem olarak kabul edilen problemlerden "Çarpanlara Ayırma Problemi (Integer Factorization Problem)" nin güvenliği asal çarpanlarının büyüklüklerine bağlıdır. Diğer bir ifadeyle, çarpanlara ayırma probleminin zor olabilmesi için çarpan olarak kullanılan asal sayıların yeteri kadar büyük olması gerekmektedir. Bu durumda başka bir önemli problem karşımıza çıkmaktadır. Bu problem, çarpanlara ayırma probleminin zorluğuna dayanan RSA algoritmasının güvenilir olması için yeteri kadar büyüklükte asal sayı üretme problemidir. Bu tez çalışmasında temel olarak bu problem ele alınacaktır.

Şifreleme algoritmalarında güvenirliği arttırmak için yeterince büyük asal sayılara ihtiyaç duyulmaktadır. Büyük sayıların asal olup olmadığını belirleyebilmek için önerilen asal sayı test yöntemleri ile bir sayının asal olup olmadığı belirlenebilmektedir. Asal sayı test yöntemleri kesin ve olası asallık testleri olarak ikiye ayrılmaktadır. Olası asallık testleri ihmal edilebilecek düzeyde bir hata oranı ile test edilecek sayı hakkında "asal değil" ya da "olası asal" sonucunu vermektedir. Bu tez çalışmasında en çok bilinen ve kullanılan olası asallık test yöntemleri incelenmiştir. Test algoritmalarının en kısa sürede doğru sonucu vermesi şifreleme algoritmalarının güvenilir olması ile de doğru orantılıdır. Test algoritmaları birbirinden farklı sürelerde ve farklı hata oranları ile sonuca ulaşmaktadır. Çalışma zamanlarındaki bu farklılık en kısa sürede sonuca ulaştıran algoritmanın tespit edilmesini sağlamaktadır. Testin hızlı sonuç vermesinin yanı sıra hata oranının da ihmal edilebilecek düzeyde küçük olması istenilen bir durum olmaktadır. Sonuç olarak büyük basamak değerine sahip bir sayı için en kısa sürede ve en düşük hata oranı ile asal olup olmadığını belirlemek, kriptografi için büyük önem arz etmektedir.

1.1. Motivasyon ve Tezin Önemi

Açık anahtarlı kriptografi'de birçok kriptosistemin tasarımı için çok büyük asal sayılara ihtiyaç vardır. Bu büyük asal sayıları üretmek için kriptografi'de polinom zamanda çalışan asal sayı test yöntemleri kullanılmaktadır. Bu tez çalışmasının amacı kriptografi'de kullanılan büyük asal sayıların nasıl üretildiği sorusuna cevap vermektir. Bu sebeple olasılıksal (probabilistic) asallık testleri ve kesin (deterministic) asallık testleri üzerinde ayrıntılı bir kaynak taraması yapılarak elde edilen veriler sistematik bir şekilde bir arada sunulmuştur. Özel olarak, olasılıksal asallık testlerinin çalışma zamanlarına ve hata oranlarına bakılarak performans analizlerini yapmak ve elde edilen verilerle en etkili yöntemin hangisi olduğunu gözlemlemek çalışmanın temel amacını oluşturmaktadır. Ayrıca, büyük asal sayılar kullanılarak tasarlanan açık anahtarlı RSA kriptosistemini ayrıntılı olarak incelemek amaçlanmıştır.

Bu çalışmada kriptolojinin tarihini, RSA kriptosistemini, asal sayıları, büyük sayıların asal olup olmadığını belirleyebildiğimiz asallık testlerini ayrıntılı olarak inceleyebileceğimiz şekilde geniş bir kaynak taraması yapılmıştır. İncelenen olası asallık testlerinin çalışma zamanları ve hata oranları esas alınarak performans analizi yapılmış ve C++ programlama dilinde kodlaması yapılmıştır. Elde edilen sonuçlar neticesinde alanında geniş çaplı araştırılan Türkçe kaynak olması ve kriptolojide bu alandaki Türkçe kaynak ihtiyacını karşılaması açısından bu çalışmanın başta lisans, yüksek lisans ve doktora çalışmaları yapan öğrenciler ve kriptoloji konusuna ilgi duyan herkes için anlaşılabilir nitelikte bir kaynak olması çalışmanın önemini artırmaktadır.

1.2. Organizasyon

Bu tezin sunumu aşağıda verildiği gibi organize edilmiştir. Bölüm 1 de çalışmada odaklanılacak probleme yer verilecektir. Bölüm 1.3.'de öncelikle kriptoloji terminolojisine yer verilecek ve çalışma da kullanılacak ifadeler tanımlanacaktır. Daha sonra çalışmanın konusu için kısaca kaynak araştırması verilecektir. Bölüm 2'de kriptolojinin tarihsel gelişim aşamaları ve Türkiye'de kriptoloji çalışmaları verilecektir. Bölüm 3'de çalışma boyunca incelenecek ve kullanılacak olan tam sayılarla ilgili işlemler ve özellikler, temel matematik tanım ve teoremlerine yer verilecektir. Bölüm 3.2.'de asal sayıların tarihinden başlayarak, özellikleri ve çeşitleri verilerek önemi vurgulanacaktır. Bölüm 4'te RSA algoritması genel olarak incelenerek, algoritmanın şifreleme ve şifre çözme hızını etkileyen faktörler, çarpanlara ayırma metotları ve RSA standartlarına yer verilecektir. Bölüm 5.1'de verilen bir tek sayının asal olup olmadığını belirleyebildiğimiz kesin asallık testlerinden olan Lucas-Lehmer ve AKS testleri, Bölüm 5.2.'de olası asallık testlerinden olan Fermat, Solovay-Strassen, Miller-Rabin, Lehmann ve Frobenius asallık testleri ayrıntılı olarak incelenecek, hata oranları ve karmaşıklıkları verilerektir. Bölüm 6'da olası asallık testlerinden, Fermat, Solovay-Strassen ve Miller-Rabin testlerinin çalışma zamanları ile bir karşılaştırma sunulacaktır.

1.3. Kriptoloji Terminolojisi

Bu bölümde kriptoloji biliminde yer alan temel tanımlardan bahsedilmiştir. Tanımlar ve terminoloji oluşturulurken (Yeşilbaş, 2016) ve (Kodaz, 2003) çalışmalarından faydalanılmıştır.

Kriptografi (Cryptography): Verilerin istenmeyen taraflar (kişiler) tarafından anlaşılamayacak şekle dönüştürülmesinde (şifrelenmesinde) kullanılan tekniklerin/yöntemlerin bütünüdür.

Kriptanaliz (Cryptanalysis): Şifrelenmiş verileri anahtar olmadan çözebilmek için kriptografik sistemlerin güvenliklerini inceleyen ve zayıf yönlerini bulmaya çalışan bilim dalıdır.

Kriptoloji (Cryptology): Kriptografi ve kriptanaliz bilimlerinden oluşan bilimdir.

Steganografi (Steganography): Eski Yunanca da saklanmış yazı anlamına gelen ve bilgiyi saklama bilimine verilen addır. Kriptografi biliminden farklıdır.

Düz Metin (Plaintext): Şifrelenmemiş anlamlı açık metindir.

Şifreli Metin (Ciphertext): Açık metinin kriptografik yöntemlerle şifrelenmiş (anlamsız) halidir. Ayrıca kapalı metin olarakta adlandırılır.

Şifreleme (Encryption): Açık metinden kapalı metine dönüştürme işlemidir.

Şifre Çözme (Decryption): Şifreleme işleminin tersidir, diğer bir ifade ile, kapalı metini açık metine dönüştürme işlemidir.

Anahtar (Key): Şifreleme ve şifre çözme işlemlerinde kullanılan en kritik bilgidir.

Açık Anahtar (Public Key): Herkes tarafından görülebilen anahtardır.

Gizli anahtar (Secret Key): Şifreli veriyi çözmek için kullanılan anahtardır.

Anahtar Üretimi (Key Generation): Şifreleme ve şifre çözme işlemlerinde kullanılacak olan anahtarları üretme işlemidir.

Gizlilik (Confidentiality): Verinin istenmeyen üçüncü kişilerden uzak tutulması, içeriğinin gizli kalmasıdır.

Bütünlük (Integritiy): Verinin değiştirilmesi veya yok edilmesine karşı korunmasıdır.

Süreklilik (Continuity): Verinin zamanında ve güvenilir bir şekilde kullanımının sağlanması.

Saldırı (Attack): Bir kriptosistemi kırmak için yapılan teşebbüstür.

1.4. Kaynak Araştırması

Gizli haberleşme isteğinin ortaya çıkmasından bu yana teknolojinin de gelişimiyle şifreleme sistemleri ve şifreleme cihazları da değişmiştir.

(Hill, 1929), "Cryptography in an Algebraic Alphabet" adlı makalesinde geliştirdiği şifreleme sistemini cebirsel ifadelerle anlatmış ve matematiğin şifrelemede ne kadar etkin kullanılabileceğini göstermiştir. İkinci Dünya savaşı sırasında askeri alanda kullanılan şifreleme sistemleri ve bunların çözülmesinde kullanılan algoritmaların çeşitliliği de artmış, böylece bu alandaki çalışmalar hız kazanmıştır.

(Schneier, 1996) çalışmasında modern kriptografinin kapsamlı bir incelemesini sunmaktadır. Şifreleme algoritmalarında verilerin gizliliğini korumak için kullanılacak kriterleri vermektedir. Ayrıca olası asallık testleri ile ilgili çalışma prensipleri verilmiştir.

Koca (2020) "Asal Sayıların Tespiti İçin Farklı Metot ve Uygulamaları" isimli çalışmasında, yeni bir asal sayı bulma yöntemi vererek bir asal sayı dizisi tanımlamıştır.

(Koblitz, 1994) "A Course in Number Theory and Cryptography" isimli çalışmasında sayılar teorisi uygulamalarında ve özellikle kriptografide yer alan aritmetik konuları ele almaktadır. Bu konulara dayanan kriptografik şifreleme yöntemlerini vermektedir. Bunun yanında asallık testlerinin verimliliğine ilişkin uygulamaları vermektedir.

(Wong, 2021) “Real-World Cryptography” isimli çalışmasında günlük hayatta hayatta karşımıza çıkan pratik ve kullanışlı gerçek dünya kriptografisinin durumunu ele almaktadır.

(Agrawal, Kayal, & Saxena, 2004) çalışmasında kesin asallık test yöntemlerinden biri olan AKS testinin adımlarını açıklamaktadırlar.

(Rabin, 1977) “ Probabilistic Algorithm for Testing Primality” isimli çalışmasında olası asallık testlerinden hata oranı en düşük ve en pratik test olan Miller-Rabin olası asallık testini vermektedir.

(Solovay & Strassen, 1977), “A Fast Monte-Carlo Test for Primality” isimli çalışmada jacobi sembolüne dayanan olası asallık testlerinden biri olan Solovay-Strassen olası asallık test yöntemi önerilmiştir.

(Grantham, 1998) çalışmasında Frobenius olası asallık testi algoritması önerilmekte ve hata oranı verilmektedir.

(Yıltaş, 2003) “Kriptolojide Kullanılan Asal Sayı Test Algoritmalarının Performans Açısından Karşılaştırılması” isimli çalışmasında olası asallık test yöntemleri algoritmalarının çalışma zamanları için bir karşılaştırma yapmaktadır.

(Akyıldız, Cenk, & Sınak, 2021), çalışmasında karmaşıklık teorisinin temel kavramları ve kriptografide kullanılan bazı algoritmalar verilmektedir. Herhangi bir şifreleme sisteminin uygulanmasında ihtiyaç duyulan algoritmalar ve karmaşıklıklarına da yer verilmektedir.

(Rivest, Shamir, & Adleman, 1978) çalışmasında yeteri kadar büyük sayılar için çarpanlara ayırma probleminin zorluğuna dayanan açık anahtarlı şifreleme sistemi olan RSA kriptosistemi önerilmiştir.

(Smart, 2016) “Cryptography Made Simple” isimli çalışmasında modern kriptografinin temellerini ve güvenliklerini vermektedir.

(Menezes & Oorschot, 1997) çalışmasında kriptolojinin tarihsel gelişiminden başlayarak açık anahtarlı şifreleme sistemi olan RSA kriptosistemini ve olası asallık testlerini incelenmekte ve bu testlerin karşılaştırması verilmektedir.

(Koç, 1994) **“**High-Speed RSA Implementation” isimli, RSA Laboratories de yayınlanan raporunda modüler üs alma işlemine dayanan kriptosistemler için algoritmaları ve çalışma sürelerini vermektedir.

(Akyıldız, Çalık, Özarar, Tok, & Yayla, 2013) “RSA Kriptosistemi Parametreleri için Güvenlik Testi Yazılımı” isimli ISC Turkey 6. Uluslararası Bilgi Güvenliği ve Kriptoloji Konferansında yayınlanan çalışmalarında RSA kriptosisteminde üretilen açık ve gizli anahtar parametrelerinin güvenilir olması için gereken kriterleri açıklamaktadırlar.

(Montgomery, 1985) çalışmasında RSA kriptosisteminin şifreleme hızını arttıran Montgomery Modüler Çarpım Algoritması’ önerilmiştir.

2. KRİPTOLOJİ TARİHİ

Bu bölümde tarih boyunca kullanılan şifreleme yöntemleri tarih sırasına göre ele alınacak ve 29 harfli Türk alfabesi kullanılarak örnekler verilecektir. Ayrıca, Türkiye'de kriptoloji tarihi incelenerek, şifreleme cihazlarına ve şifreleme sistemlerine yer verilecektir.

2.1. Kriptolojinin Tarihsel Gelişimi

Bu bölümde kriptolojinin tarihsel gelişiminden bahsedilecek ve önemli kısımlar verilecektir. Bu bilgilerin derlenmesinde genel olarak (Yeşilbaş, 2016), (Çimen, Akleylek, & Akyıldız, 2007) ve (Topaloğlu, Calp, & Türk, 2016) çalışmalarından faydalanılmıştır.

Kriptografi ve Kriptanaliz, Kriptoloji bilimini oluşturan iki bilim dalıdır. Kriptografi Yunanca gizli anlamına gelen "kriptos" ve yazı anlamına gelen "graphi" sözcüklerinden türetilmiştir. Verilerin güvenli olarak saklanmasını ve iletilmesini sağlar. Kriptanaliz ise şifrelenmiş verileri anahtar olmadan çözebilmek için kriptografik sistemlerin güvenliklerini inceleyen ve zayıf yönlerini bulmaya çalışan bilim dalıdır.

İlk kriptolog, 4000 yıl önce yaşamış Mısırlı bir kâtiptir. Kâtip daha önce hiç kullanılmamış hiyeroglifleri şifrelenmiş bir şekilde oluşturmuştur.

M.Ö. 1900: İlk kriptografik belge (Şekil 2.1.1.), yaklaşık olarak M.Ö. 1900 yılında yazıldığı tahmin edilen Rosetta tabletidir.

Şekil 2.1.1. Rosetta Tableti (URL-1, 2021)

Napolyon'un M.S. 1798'de Mısır seferi sırasında Fransız askeri Pierre-François Bouchard'ın kale yapımında bulduğu Rosetta Taşının yardımıyla çözülmüştür. Rosetta Taşı mısır halkının kullandığı dil olan Demotik, Hiyeroglif ve Antik Yunanca olmak üzere üç farklı dilde yazılmıştır.

Hiyeroglif yazısı Jean-Francois Champollion tarafından M.S. 1822 yılında çözülmüştür. Yazının çözülmesi ile eski mısır bilimi olan "egyptology" oluşturulmuş ve geçmişte bulunan Hiyeroglif eserlerine açıklık kazandırılmıştır. Rosetta taşı, M.S. 1802 yılından bu yana Londra'da British Museum'da sergilenmektedir (URL-1, 2021).

M.Ö. 1500: Mezopotamya tabletlerinde çömleklerin cilalanması hakkındaki bilgilerin şifrelenmiş olarak bulunması kriptografinin ilk kullanımı olarak kayıtlara geçmiştir (Çimen, Akleylek, & Akyıldız, 2007).

M.Ö. 600-500: Eski Ahit de yer alan ve İbrani alfabesine dayanan ilk yer değiştirme şifresi ATBASH şifrelemesidir. Bu şifreleme sisteminde alfabedeki, ilk harf son harfle, ikinci harf sondan ikinci harfle yer değiştirir, diğer harfler de de sırasıyla bu uygulanır. (Yeşilbaş, 2016).

Örnek 2.1.1. "PRESTİJ" kelimesini şifreleyelim. ATBASH şifresini Türkçe alfabeye göre uyarlayacak olursak "P", "H" ile "R", "Ğ" "ile "E", "T" ile "S", "G" ile "T", "E" ile "İ". "O" ile "J", "N" ile yer değiştirir. Böylece şifreli sözcük "HĞTGEON" olur.

Kelime şifreleme işlemlerinde, şifreleme yapılırken alfabedeki harflerin yerine sayılar da kullanılabilir. Arapça için de kullanılan bu sistem, alfabedeki harflerin kolay öğrenilmesi için düzenlenmiş sözcükler olan "EBCET" hesabıdır.

ARAP ALFÂBESİNİN SIRA VE SAYISAL DEĞERLERİ

Sıra değeri:	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Arapça harfler:	ا	ب	ج	د	ه	و	ز	ح	ط	ي	ك	ل	م	ن
Türkçe okunuşu:	elif	be	cim	dal	he	vav	ze	ha	tı	ye	kef	lam	mim	nun
Sayısal değer:	1	2	3	4	5	6	7	8	9	10	20	30	40	50
Sıra değeri:	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Arapça harfler:	س	ع	ف	ص	ق	ر	ش	ت	ث	خ	ذ	ض	ظ	غ
Türkçe okunuşu:	sin	ayn	fe	sad	kaf	re	şin	te	se	hı	zel	dad	zı	ğayn
Sayısal değer:	60	70	80	90	100	200	300	400	500	600	700	800	900	1000

Şekil 2.1.2. Arap Alfabesinin Sıra ve Sayısal Değerleri (URL-2, 2021)

Ebcet hesabında alfabedeki her harfe bir sayı değeri verilerek, bir kelimeyi oluşturan harflerin toplam sayı değerini, anlatılmak istenen bir olayın tarihine denk düşürmeye çalışılır.

M.Ö. 480: Eski Yunancada "saklanmış yazı" anlamına gelen, esas amacı mesajın saklanması olan Steganografi tekniği Yunanlılar ve Persler arasındaki savaşta kullanılmıştır. Pers Kralı Daryus'un elinde tutsak olan Yunanlı komutan Histiaeus, Milet nehrinde ki damadı Aristagoras'a göndermek üzere kölesinin saçını kazıtıp üzerine mesajını yazmış, kölenin saçı uzayınca onu Milet'e göndermiş ve kölenin saçı kazıtılınca mesaj okunmuştur. Bu sayede Yunanlar Pers istilasına karşı savaşı kazanmışlardır.

Steganografiden bir örnekte kendi tarihimizden verilebilir. 26 Ağustos 1922 - 18 Eylül 1922 tarihleri arasında Büyük Taarruz emrinin verildiği Afyonkarahisar'daki Türk istihbarat timleri halk ile haberleşmede steganografi kullanıyordu. Düşman askerlerinden edinilen bilgileri Fahrettin Alay Paşa'ya istihbarat görevlileri ulaştırıyordu. Toplanan bilgiler limon suyuyla kağıt üzerine yazılıyordu. Beyaz kağıt üzerine limon suyuyla yazılan bilgiler görünmüyordu. Görünmeyen bilgiler ateşe tutulduğunda görülür hale geliyor ve yetkili kişilerce okunabiliyordu.

M.Ö. 205-123: Yunanlılar tarafından tasarlanan Polybius'un dama tahtası şifreleme sisteminde kullanılan şifreleme alfabesi Yunan ve Roma alfabesidir. Kural şöyledir: Şifreleme için kullanılacak olan alfabedeki harfler sırayla 5×6 lık bir matrisin satırlarına sırasıyla yazılır. Her harfi temsil eden iki rakam vardır; ilk rakam bulunduğu satırı, ikinci rakam bulunduğu sütunu temsil eder (Çimen, Akleylek, Akyıldız, 2007).

	1	2	3	4	5	6
1	A	B	C	Ç	D	E
2	F	G	Ğ	H	I	İ
3	J	K	L	M	N	O
4	Ö	P	R	S	Ş	T
5	U	Ü	V	Y	Z	

Şekil 2.1.3. Polybius'un Dama Tahtası

Örnek 2.1.2. "PRESTİJ" kelimesini Polybius'un dama tahtasına göre şifreleyelim ve şifreyi çözelim.

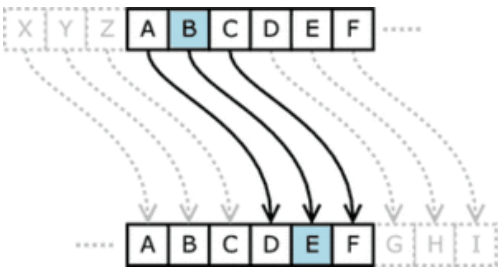
Düz Metin: PRESTİJ

Polybius'un Dama Tahtası'na göre "P" harfini şifrelemek için harfin bulunduğu satır ve sütun numarasına bakılır. İlk sayı satır ikinci sayı sütun bilgisini verecek şekilde sayı oluşturulur.

Şifreli Metin : 42-43-16-44-46-26-31

Şimdi "42-43-16-44-46-26-31" şifreli metni çözelim. Verilen kurala göre sayı çiftlerindeki rakamlardan ilki bulunduğu satırı, ikincisi bulunduğu sütunu temsil eder. Buna göre 42 sayısında 4. satır ve 2. sütunun kesişimin deki harf olan "P" alınır. Bu şekilde şifreli verilen sayı çiftlerinin hepsi çözülerek düz metne ulaşılır.

M.Ö. 60-50: Romalı lider Julius Caesar (Jül Sezar) tarafından devlet haberleşmesinde kullanılan Sezar Şifreleme, alfabedeki harflerin yerini değiştirerek oluşturulan şifreleme yöntemidir. Bu yöntem şifrelenecek metindeki her harf alfabede kendisinden 3 harf sonraki harfle değiştirilmesine dayanmaktadır.



Şekil 2.1.4. Üç harf atlamalı Sezar Şifreleme (URL-3, 2021)

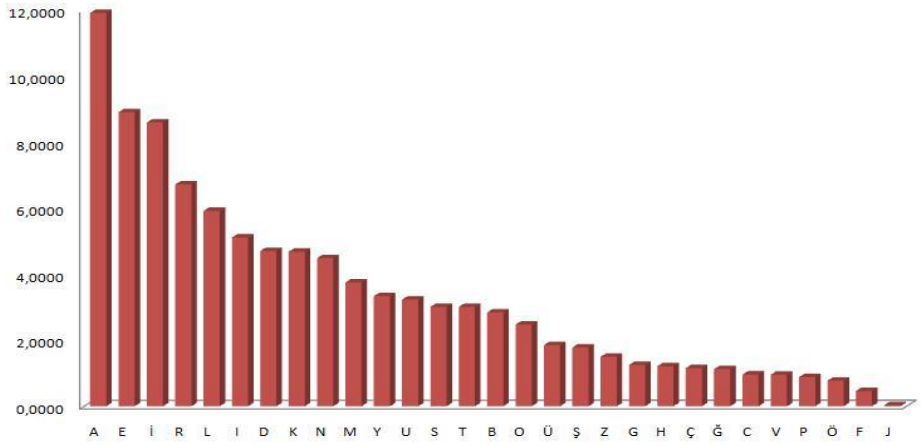
"PRESTİJ" kelimesini bu kez de 3 harf atlamalı Sezar şifreleme tekniği ile şifreleyelim.

$P \rightarrow Ş,\ R \rightarrow T,\ E \rightarrow Ğ,\ S \rightarrow U,\ T \rightarrow V,\ İ \rightarrow L,\ J \rightarrow M$ şeklinde şifrelenir ve şifreli metin "ŞTĞUVLM" olur.

M.Ö. 5: İlk kriptografik cihaz olarak kabul edilen Scytale, Yunanlılar tarafından kullanılan şifreleme cihazıdır. Gönderilecek mesaj bir şerit üzerine yazılır, daha sonra kalın bir sopaya sarılırdı. Şerit açık olarak karşı tarafa yollanırdı. Karşı taraf, şeridi aynı kalınlıkta bir sopaya sararak mesajı okuyabilirdi.

Şekil 2.1.5. Scytale (URL-4, 2021)

M.S. 873: Al-Kindi kriptanaliz üzerine ilk makale olan "Kriptografik Mesajların Deşifresi" isimli makaleyi yazmıştır. Makalede bir kriptanaliz yöntemi olan frekans analizi kavramı ortaya atılmıştır. Yapılan araştırmalarda Türk alfabe sisteminde en çok kullanılan sesli harfler "A, E" , en çok kullanılan sessiz harfler ise "N, R, L, K, D" harfleri olarak bulunmuştur. (Gazete köşe yazıları ve 9 yazara ait 37 kitaptan elde edilmiş, 11 milyon karakterden oluşan 13,4 MB boyutundaki metin seti üzerinden elde edilmiştir).



Şekil 2.1.6. Türkçe Harf Frekansları (URL-7, 2021)

1450: Voynich Yazmaları, 200 sayfalık kısmı bilinmeyen bir dilde yazılmış, farklı bitkiler, astronomik resimler ile resmedilmiş bir el yazmasıdır. 2018 yılına kadar deşifre edilemeyeceği düşünülmekteydi.

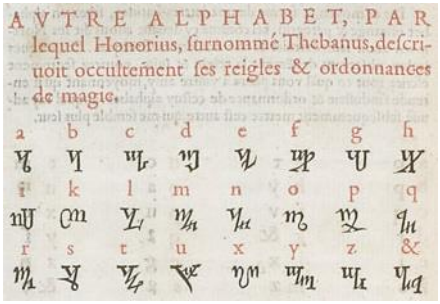
Şekil 2.1.7. Voynic El Yazması (URL-6, 2021)

Kanadalı bilgisayar bilimcilerinin geliştirdikleri yapay zeka, bu kitabın 600 yıllık gizemini çözmeyi başarmıştır. 2018 yılında Alberta Üniversitesi'nden araştırmacılar, garip kitabın kelimelerinin arkasındaki gizli ve şifreli dili anlaşılır hale getirmek için *algoritmik deşifre* tekniği kullanarak el yazmasındaki bölümleri çözmüşlerdir. Kelimelerin %80'inden fazlasının İbranice sözcüklerden oluştuğu ortaya çıkarılmıştır. Çeşitli bitki türlerini içeren yazmanın 'Bitkiler' bölümünün ilk kısmında, çiftçi, ışık, hava ve ateş gibi botanikle ilgili birçok terim bulunmaktadır (URL-7, 2021).

1460: Leone Battista Alberti tarafından ilk kez çoklu alfabe kullanılarak kriptosistem tasarlanmıştır. Alberti eş merkezli iki diskten oluşan bir alet geliştirdi. İçteki çemberin sabit, olduğu, dıştaki çemberin onun etrafında dönebildiği bu disk yardımıyla, her harfin istenilen miktarda ötelenmiş hali kolaylıkla görülebilmektedir. Disklerde kaydırma miktarı sabit değildir (Topaloğlu, Calp, & Türk, 2016).

Şekil 2.1.8. Alberti Diski (URL-8, 2021)

1518: Johannes Trithemus (Alman rahip) tarafından yazılan "Polygraphie" adlı kitapta çoklu alfabe kullanılan şifreleme sisteminden söz edilmiştir.



Şekil 2.1.9. Örnek Alfabe (URL-9, 2021)

1586: Blaise De Vigenère (Fransız diplomat) adına adanan Vigenère şifreleme yönteminde şifrelenecek metinde bulunan her bir harf farklı bir alfabeyle şifrelenmektedir. Hangi alfabenin belirleneceği anahtar sözcüğe bakılarak karar verilmektedir. Vigenère Şifresi uzun yıllar boyunca "le chiffre indechiffrable" diğer bir ifade ile kırılamayan şifre olarak adlandırılmıştır. 1854-1863 yılları arasında İngiliz matematikçi Charles Baggage ve Avusturya Ordusunda görevli kriptograf Friedrich Kasishi tarafından kırılmıştır (Bruen & Forcinito, 2011).

Çalışmamızda vereceğimiz örneklerde kullanılacak olan Türkçe alfabedeki harflerin sıra sayılarını veren tabloyu verelim.

Tablo 2.1.1. Türk alfabesindeki harflerin karşılığı

A	B	C	Ç	D	E	F	G	Ğ	H	İ	I	J	K	L
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
M	N	O	Ö	P	R	S	Ş	T	U	Ü	V	Y	Z	
15	16	17	18	19	20	21	22	23	24	25	26	27	28	

Örnek 2.1.3. 'PRESTİJ ' kelimesini 'AY' anahtar sözcüğü kullanılarak Vigenère algoritması ile şifreleyelim.

- Şifrelenecek metin: P R E S T İ J
- Şifrelenecek metnin harf karşılığı: 19 20 05 21 23 10 12
- Anahtar: AY (00 27)
- Tekrarlı anahtar: AYAYAYA
- Anahtarın harf karşılığı: 00 27 00 27 00 27 00
- Şifreleme işlemi;

$$\begin{array}{rr} & 19\ 20\ 05\ 21\ 23\ 10\ 12 \\ + & 00\ 27\ 00\ 27\ 00\ 27\ 00 \\ \hline & 19\ 47\ 05\ 48\ 23\ 37\ 12 \end{array}$$

Şifreleme işleminin sonucunun mod 29 da karşılıkları 19 18 05 19 23 08 12 şeklinde bulunur. Dolayısıyla, şifreli metin P Ö E P T Ğ J şeklinde elde edilir.

Şimdi, "P Ö E P T Ğ J" şifreli metnini "AY" anahtar sözcüğü ile çözelim.

- Şifreli metin: P Ö E P T Ğ J
- Şifreli metnin harf karşılığı: 19 18 05 19 23 8 12
- Anahtar: AY
- Tekrarlı anahtar: AYAYAYA
- Anahtarın harf karşılığı: 00 27 00 27 00 27 00
- Şifre çözme işlemi;

$$\begin{array}{rr} & 19\ 18\ 05\ 19\ 23\ 08\ 12 \\ - & 00\ 27\ 00\ 27\ 00\ 27\ 00 \\ \hline & 19\ 20\ 05\ 21\ 23\ 10\ 12 \bmod 29 \end{array}$$

Dolayısıyla, 'PÖEPTĞJ' şifreli mesajını 'AY' anahtar sözcüğü ile çözme işlemi yapınca "PRESTİJ" orijinal metnine ulaşılır.

Türk Alfabesine Göre Vigenère Tablosu

Vigenère şifreleme algoritması 26 x 26 bir tablo oluşturularak kullanılmaktadır. Bu tabloda ilk satır ve sütunda alfabe yer almaktadır. Daha sonraki satırlar oluşturulurken alfabe bir adım sola kaydırılarak oluşturulur.

Biz bu çalışmada Türkçe alfabe kullanacağımız için 29×29 lık bir tablo kullanacağız. Aşağı sıralı sütunlarda her satırdaki harfler 1 sola kaydırılarak yeni alfabeler elde edilir.

Tablo 2.1.2. Türk alfabesine göre Vigenère tablosu

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

Şifreleme işlemi yapılırken anahtar olarak seçilen kelime şifrelenecek metnin uzunluğu kadar tekrarlı olarak yazılır. Metnin harfleri satır, anahtar kelimenin harfleri sütun olmak kaydıyla kesişim noktasındaki harfler alınarak şifreleme işlemi tamamlanır.

Örnek 2.1.4. "FEN BİLİMLERİ ENSTİTÜSÜ" metnini Vigenère tablosuna göre şifreleyip çözelim.

- Şifrelenecek metin : F E N B İ L İ M L E R İ E N S T İ T Ü S Ü
- Anahtar : MATEMATİK

Şifrelenecek metin 21 karakter uzunluğunda, anahtarımız ise 9 karakter uzunluğundadır. Öncelikle anahtar metni şifrelenecek mesaj uzunluğu kadar tekrarlı yazılır.

F	E	N	B	İ	L	İ	M	L	E	R	İ	E	N	S	T	İ	T	Ü	S	Ü
M	A	T	E	M	A	T	İ	K	M	A	T	E	M	A	T	İ	K	M	A	T

Vigenère tablosundan ilk harf satır ikinci harf sütun eşleştirmesi yapacak olursak, F-satır, M-sütun kesişimindeki harf olan "S" alınır. Aynı şekilde diğer harfler için de bu işlem yapılarak şifreli metin

- S E İ F Ü L D Ü Y R R D İ C S O R G I S P

şeklinde elde edilir. Şimdi şifreli metni Vigenère tablosu yardımıyla çözelim. Şifre çözmek için anahtar şifresi çözülecek metin uzunluğu kadar tekrarlı yazılır.

S	E	İ	F	Ü	L	D	Ü	Y	R	R	D	İ	C	S	O	R	G	I	S	P
M	A	T	E	M	A	T	İ	K	M	A	T	E	M	A	T	İ	K	M	A	T

Şifreli metni çözmek için verilen metindeki ilk harf olan S harfinin karşılığını bulmak için anahtar olan M sütunundaki S harfi bulunup buna karşılık gelen satıra bakılır. O halde S harfine karşılık gelen harf "F" harfidir. Aynı şekilde anahtar olan A sütunundaki E harfi bulunup buna karşılık gelen satıra bakılır. O halde E harfine karşılık gelen harf "E" harfidir. Bu şekilde devam edilerek, İ → N, F → B, Ü → İ, L → L, D → İ, Ü → M, Y → L, R → E, R → R, D → İ, İ → E, C → N, S → S, O → T, R → İ, G → T, I → Ü, S → S, P →Ü karşılıkları bulunur. Açık metin olan "FEN BİLİMLERİ ENSTİTÜSÜ" metnine ulaşılır.

1790: ABD Başkanı Thomas Jefferson, matematikçi Dr. Robert Patterson ile birlikte günümüzde Jefferson Diski adıyla bilinen sistemi geliştirmiştir. Mucit olan Thomas Jefferson Amerikan kriptolojisinin atası olarak adlandırılmaktadır (Yeşilbaş, 2016).

Şekil 2.1.12. Jefferson Diski (URL-10, Jefferson Disk, 2021)

1854: Charles Wheatstone, Playfair şifreleme yöntemini tasarlamıştır. Playfair yöntemi için 5 x 5 bir matris kullanılmaktadır. Kullanılan alfabe her kareye bir harf gelecek şekilde tüm matris hücreleri doldurulur. 26 harften oluşan İngiliz alfabesi için tasarlanmış olan bu yöntem de "I" ve "J" harfleri aynı kareye gelecek şekilde birlikte düşünülmüştür.

Tablo 2.1.4. Playfair tablosu

A	B	C	D	E
F	G	H	I/J	K
L	M	N	O	P
Q	R	S	T	U
V	W	X	Y	Z

Yöntem şu şekilde çalışır;

- 5 x 5 boyutlu matris oluşturulur.
- Anahtar kelimede ki birden fazla yer alan harfler atılır ve matriste soldan başlayarak yazmaya başlanır. Boş kalan karelere alfabede kalan harfler yazılır.
- Şifrelenecek metin iki harften oluşan gruplara ayrılır. Metin tek sayıda harften oluşuyorsa, metnin sonuna istenilen bir harf eklenir.
- İkili gruplardaki harfler aynı satırda ise sağlarındaki harflerle şifrelenir.
- İkili gruplardaki harfler aynı sütunda ise bir alt satırdaki harflerle şifrelenir.
- İkili harfler aynı satırda veya sütunda değil ise, ilk harfi şifrelemek için bu harfin bulunduğu satır ve ikinci harfin bulunduğu sütunun kesimindeki harf alınır.
- İkinci harfi şifrelemek için bulunduğu sütun ile ikinci harfin bulunduğu satırın kesişimin deki harf alınır.

Örnek 2.1.5. Türk alfabe sistemini kullanarak "PRESTİJ" anahtarı ile "FEN BİLİMLERİ" metnini Playfair algoritması ile şifreleyelim ve çözelim.

- Anahtar : "PRESTİJ"
- Şifrelenecek metin : "FEN BİLİMLERİ"

Yöntem de verildiği gibi anahtar kelimesinde birden fazla kullanılan harfler atılır ve matris oluşturulur. PRESTİJ anahtar kelimesi ile oluşturulan matris aşağıdaki gibidir.

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

Türkçe de Şekil 2.1.6. da verilen harf frekansları da dikkate alınarak, az sıklıkla kullanılan harfleri iki harf bir kareye gelecek şekilde matris oluşturulmuştur. Açık metin,

- FE NB İL İM LE Rİ

şeklinde ikili gruplara dönüştürülür. Şifreleme için "FE" harf çiftine bakılır. Aynı satır veya sütunda olmadıkları için ilk harfi şifrelemek için bu harfin bulunduğu satır ve ikinci harfin bulunduğu sütunun kesimindeki harf olan "G" harfi alınır. İkinci harfi şifrelemek için bulunduğu sütun ile ikinci harfin bulunduğu satırın kesişimin deki harf olan "R" harfi alınır. Bu şekilde diğer tüm harf çiftleri için şifreleme yapılırsa;

- Açık Metin : FE NB İL İM LE Rİ
- Şifreli Metin : GR YH JK AK MR PJ

"FENBİLİMLERİ" metni "PRESTİJ" anahtarı yardımı ile "GRYHJKAKMRPJ" olarak şifrelendi. Şimdi şifrelenmiş olan "GR YH JK AK MR PJ" mesajını çözelim.

- GR çifti aynı satır veya sütunda olmadıkları için G → F olarak R → E olarak çözülür ve FE çifti elde edilir.
- YH çifti aynı sütunda oldukları için üstteki harfler olarak NB çifti elde edilir.
- JK çifti aynı satır veya sütunda olmadıkları için J→İ olarak K-L olarak çözülür ve İL çifti elde edilir.
- AK, MR, PJ çiftleri de aynı satır ve sütunda olmadıkları için aynı yöntem kullanılarak AK → İM, MR → LE, PJ → Rİ olarak çözülür.

Sonuç olarak "GRYHJKAKMRPJ" metni "FENBİLİMLERİ" olarak çözülmüş olur.

1883: Auguste Kerckhoffs (Hollandalı dilbilimci ve kriptograf) "La Cryptographie Militarie" isimli makalesinde bir şifreleme sisteminde sistem ile ilgili Kerckhoff Prensipleri'ni vermiştir. Kerckhoffs'un ortaya koyduğu altı prensip aşağıda verilmiştir.

- Kriptosistem, matematiksel olarak olmasa da, uygulamada çözülemez olmalıdır.
- Kriptosistem gizli olmamalı ve düşmanın eline geçmesi sorun yaratmamalıdır.
- Anahtar yazılı notlara gerek duyulmadan iletilebilmeli ve elde tutulabilmelidir, tarafların isteğiyle değiştirilebilmelidir.
- Kriptosistem telegraf yazışmalarına uygulanabilir olmalıdır.
- Taşınabilir olmalıdır ve kullanımı ve işlevi birçok insanın toplanmasını gerektirmemelidir.
- Kriptosistemin kullanımı kolay olmalı, zihinsel zorlamayı veya uzun kural serilerinin bilinmesini gerektirmemelidir.

1917: Zimmermann Note/Telegram (Zimmermann Telgrafı) olarak bilinen olayda ilk büyük etkili şifre kırma olayı gerçekleşmiştir. Alman İmparatorluğu'nun Dışişleri Bakanı Arthur Zimmermann tarafından Alman Büyükelçiliklerine gönderilen telgraf Alman Dışişleri şifreleme standartlarına göre kodlanmıştır. İki İngiliz'in şifre çözücünün mesaj içeriğini çözmeleri ve bunu ABD Başkanı Wilson'a okutmaları sonucunda 2 Nisan 1917'de ABD Birinci Dünya Savaşı'na katılmıştır.

1917: Joseph Mauborgne ve Gilbert Vernam koşulsuz güvenilir olan Vernam şifreleme yöntemini önermişlerdir (Stallings , 2003).

1929: Leste S. Hill'in yayınladığı "Cryptography In An Algebraic Alphabet" adlı eserinde, çoklu alfabe şifreleme sistemi olan Hill şifresini önermiştir. (Hill, 1929).

m pozitif bir tam sayı ve n şifreleme yapılacak dilin karakter sayısı olmak üzere, açık mesaj ve kapalı mesaj uzayları $\mathbb{Z}_n^m$ olsun. Şifreli metindeki her eleman, düz metnin n tane elemanının lineer kombinasyonu şeklinde ifade edilir. m tane karakterden oluşan düz metin $X = (x_1, x_2, \dots, x_m)$ ve bu metnin şifrelenmiş hali $Y = (y_1, y_2, \dots, y_m)$ olsun. Buradan,

$$y_1 \equiv k_{11}x_1 + k_{21}x_2 + \cdots + k_{m1}x_m \pmod{n}$$
$$y_2 \equiv k_{12}x_1 + k_{22}x_2 + \cdots + k_{m2}x_m \pmod{n}$$

$$\vdots$$
$$y_m \equiv k_{1m}x_1 + k_{2m}x_2 + \cdots + k_{mm}x_m \pmod{n}$$

m bilinmeyenli m denklemden oluşan sistem oluşur. Bu sistem matrislerle;

$$Y = [y_1 \ y_2 \ldots y_m] \text{ ve } X = [x_1 \ x_2 \ldots x_m]$$

$$K = \begin{bmatrix} k_{11} & \ldots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{1m} & \ldots & k_{mm} \end{bmatrix} \pmod{n}$$

şeklinde gösterilir ve $\boldsymbol{Y = X \cdot K} \pmod{n}$ olarak ifade edilir. Burada şifreleme ve deşifreleme fonksiyonları,

$$E_K(X) = \boldsymbol{X \cdot K} \bmod n \text{ ve } D_K(Y) = \boldsymbol{Y \cdot K^{-1}} \bmod n$$

olur. K anahtar matrisi tersinir bir matris olmalıdır. $\mathbb{Z}_n$ halkası üzerinde matrisin tersinin olması için $\det(K) \not\equiv 0$ ve $\text{obeb}(\det(K), n) = 1$ olmalıdır.

Örnek 2.1.6. "*MATEMATİK*" düz metnini

$$K = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 8 & 7 \end{bmatrix}$$

anahtar matrisini Türkçe alfabeyi kullanarak $\mathbb{Z}_{29}$ üzerinde Hill şifreleme sistemi ile şifreleyelim. Öncelikle matrisin determinantını bulalım. $\det(K) = 27$ olarak bulunur ve bu determinant değeri 29 ile aralarında asal olduğu için K tersi alınabilen bir matristir. Şifreleme yaparken daha önce Tablo 2.1.2. de verilen Türkçe alfabedeki harf karşılıkları kullanılmıştır.

Anahtar 3×3 matris olduğundan, $m = 3$ ve düz metin Türkçe olduğundan, $n = 29$ olur. Şifrelenecek metin $m = 3$ olduğundan 3 uzunluklu bloklara ayrılır ve sayı değerleri bulunur.

(M, A ,T) = (15,0,23), (E,M,A) = (5,15,0), (T,$İ$,K) = (23,10,13)

$$Y = E_K(X) = (x_1, x_1, x_1) \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 8 & 7 \end{bmatrix} \bmod 29$$

$$(y_1, y_2, y_3) \equiv (15, 0, 23) \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 8 & 7 \end{bmatrix} \bmod 29 \equiv (15, 11, 3)$$

$$(y_1, y_2, y_3) \equiv (5, 15, 0) \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 8 & 7 \end{bmatrix} \bmod 29 \equiv (7, 27, 18)$$

$$(y_1, y_2, y_3) \equiv (23, 10, 13) \cdot \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 8 & 7 \end{bmatrix} \bmod 29 \equiv (5, 26, 17)$$

(15, 11, 3), (7, 27, 18), (5, 26, 17) değerleri elde edilir. Bu değerlere karşılık gelen harfler bulunur. Böylece, şifreli metin "*MIÇ GYÖ EVO*" olarak bulunur.

Şimdi şifreli "*MIÇGYÖEVO*" metnini çözmeye çalışalım. Şifrelemede kullanılan K anahtarı 3×3 matris olduğundan $m = 3$. Verilen metin üçer parçaya ayrılıp sayı değerleri bulunur,

$$(M, I, Ç) = (15, 11, 3),\ (G, Y, Ö) = (7, 27, 18),\ (E, V, O) = (5, 26, 17)$$

Daha sonra anahtar olarak verilen

$$K = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 0 & 8 & 7 \end{bmatrix}$$

matrisinin tersi bulunur. Bu matrisin tersini kofaktör yöntemi kullanarak bulabiliriz. Bunun için

$$M_{11} = \begin{bmatrix} 5 & 6 \\ 8 & 7 \end{bmatrix} \quad M_{21} = \begin{bmatrix} 2 & 3 \\ 8 & 7 \end{bmatrix} \quad M_{31} = \begin{bmatrix} 2 & 3 \\ 5 & 6 \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} 4 & 6 \\ 0 & 7 \end{bmatrix} \quad M_{22} = \begin{bmatrix} 1 & 3 \\ 0 & 7 \end{bmatrix} \quad M_{32} = \begin{bmatrix} 1 & 3 \\ 4 & 6 \end{bmatrix}$$

$$M_{13} = \begin{bmatrix} 4 & 5 \\ 0 & 8 \end{bmatrix} \quad M_{23} = \begin{bmatrix} 1 & 2 \\ 0 & 8 \end{bmatrix} \quad M_{33} = \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$$

matrislerinin determinantlarını bulalım.

$|M_{11}| = 35 - 48 = -13,\quad |M_{21}| = 14 - 24 = -10,\quad |M_{31}| = 12 - 15 = -3,$
$|M_{12}| = 28 - 0 = 28,\quad |M_{22}| = 0 - 7 = -7,\quad |M_{32}| = 6 - 12 = -6,$
$|M_{13}| = 32 - 0 = 32,\quad |M_{23}| = 8 - 0 = 8,\quad |M_{33}| = 5 - 8 = -3.$

Kofaktörlerini bulalım, diğer bir ifade ile $C_{ij} = (-1)^{i+j} \cdot M_{ij}$ değerlerini hesaplayalım.

$C_{11} = -13 \quad C_{21} = 10 \quad C_{31} = -3$
$C_{12} = -28 \quad C_{22} = -7 \quad C_{32} = 6$
$C_{13} = 32 \quad C_{23} = -8 \quad C_{33} = -3$

Buradan $C = \begin{vmatrix} -13 & -28 & 32 \\ 10 & -7 & -8 \\ -3 & 6 & -3 \end{vmatrix}$ matrisi ve $C^T = \begin{vmatrix} -13 & 10 & -3 \\ -28 & -7 & 6 \\ 32 & -8 & -3 \end{vmatrix}$ matrisi elde edilir.

Sonuç olarak, K matrisinin tersi $K^{-1} = \frac{1}{|K|} \cdot C^T$ şeklindedir.

$$K^{-1} = \frac{1}{27} \cdot \begin{bmatrix} -13 & 10 & -3 \\ -28 & -7 & 6 \\ 32 & -8 & -3 \end{bmatrix} = 14 \cdot \begin{bmatrix} -13 & 10 & -3 \\ -28 & -7 & 6 \\ 32 & -8 & -3 \end{bmatrix} = \begin{bmatrix} 21 & 24 & 16 \\ 14 & 11 & 26 \\ 13 & 4 & 16 \end{bmatrix} \bmod 29.$$

Burada 27'nin $\bmod 29$ daki tersi 14 tür, $27 \cdot 14 \equiv 1 \bmod 29$.

Şifreli halde verilen "*MIÇGYÖEVO*" kelimesini $m = 3$ parçaya ayırarak alfabedeki karşılıklarını yazalım.

M	*I*	*Ç*	*G*	*Y*	*Ö*	*E*	*V*	*O*
15	11	3	7	27	18	5	26	17

Şifreli metin $X = D_K(Y) = Y \cdot K^{-1}$ şeklinde çözülür.

$$\left([15 \quad 11 \quad 3] \cdot \begin{bmatrix} 21 & 24 & 16 \\ 14 & 11 & 26 \\ 13 & 4 & 16 \end{bmatrix} \right) \bmod 29 = [15 \quad 0 \quad 23]$$

$$15 \rightarrow M \quad 0 \rightarrow A \quad 23 \rightarrow T$$

$$\left([7 \quad 27 \quad 18] \cdot \begin{bmatrix} 21 & 24 & 16 \\ 14 & 11 & 26 \\ 13 & 4 & 16 \end{bmatrix}\right) \bmod 29 = [5 \quad 15 \quad 0]$$

$$5 \rightarrow E \quad 15 \rightarrow M \quad 0 \rightarrow A$$

$$\left([5 \quad 26 \quad 17] \cdot \begin{bmatrix} 21 & 24 & 16 \\ 14 & 11 & 26 \\ 13 & 4 & 16 \end{bmatrix}\right) \bmod 29 = [23 \quad 10 \quad 13]$$

$$23 \rightarrow T \quad 10 \rightarrow \text{İ} \quad 13 \rightarrow K$$

Sonuç olarak; (15, 0, 23), (5, 15, 0), (23, 10, 13) sayılarına karşılık gelen harfler (*M, A, T*), (*E, M, A*), (*T, İ, K*) olduğu için "*MATEMATİK*" düz metnine ulaşılır.

1940-1944: İkinci Dünya Savaşı sırasında şifreleme sistemlerine olan ilgi artmış ve literatürde birçok çalışma yapılmıştır. Alman elektrik mühendisi Arthur Scherbius tarafından mekanik şifreleme cihazı ENIGMA tasarlanmıştır.

Şekil 2.1.13. Enigma (URL-11, 2021)

Almanya savaş yıllarında haberleşmenin diğer ülkeler tarafında çözülmemesi amacıyla şifreli mesajlar kullanmaktaydı. Bu şifreli mesajları çözmekle görevli ekibin 24 saat süresi vardı. Enigma şifreyi her gece yarısı 00:00 da değiştirmekteydi. Bu yüzden şifre çözme adına yapılan işlemler boşa gitmekteydi. Bunun için şifre çözme işleminde hızlı olmak ve zamandan tasarruf etmek önemliydi. İngiliz matematikçi ve kriptolog olan Alan Mathison Turing tarafından tasarlanan Turing makineleri ile geliştirilen Colossus bilgisayarı sayesinde Enigma'nın şifrelerinin kırılması sağlanmıştır.

1970: Ulusal Standartlar Dairesi (NBS - National Bureau of Standards); 1973'te "ulusal bir standart olabilecek kriptografik bir algoritma" talebinde bulunmuştu. Bu talebe yönelik Dr. Horst Feistel öncülüğünde IBM'nin 1974'te sunduğu LUCIFER algoritması, Ulusal Güvenlik Ajansı'nın (NSA – National Security Agency) uyguladığı değişikliklerin ardından 1977'de ilk modern şifreleme sistemi olan DES (Data Encryption Standard) adıyla resmi bilgi şifreleme standardı olarak kabul edildi.
DES şifreleme algoritması üzerine detaylı bilgi için (IBM, 1999) kaynağından faydalanılabilir.

1976: Simetrik şifreleme sistemlerinde göndericinin şifreleme anahtarını alıcı ile paylaşması gerekmektedir. İletişim kurmak isteyen ve iki farklı konumda bulunan kişilerin bu gizli anahtarı paylaşmaları büyük bir problem oluşturmaktaydı. 1976 yılında Whitfield Diffie ve Martin Hellman bir protokol geliştirerek bu paylaşım problemini ortadan kaldırmıştır.

Alice				Bob		
Gizli	Açık	Hesaplar	Gönderir	Hesaplar	Açık	Gizli
a	p, g		p,g→			b
a	p, g, A	$g^a \bmod p = A$	A→		p, g	b
a	p, g, A		← B	$g^b \bmod p = B$	p, g, A, B	b
a, **s**	p, g, A, B	$B^a \bmod p = s$		$A^b \bmod p = s$	p, g, A, B	b, **s**

Şekil 2.1.19. Diffie-Hellman Protokolü (URL-14, 2021)

Önerilen bu protokolü bir örnekle açıklamaya çalışalım.

Örnek 2.1.7.

1. Alice ve Bob aralarında asal sayı olarak $p = 11$ seçsinler ve $\mathbb{Z}_{11}^*$ çarpımsal devirli grubunun üreteci $g = 7$ olsun.
2. Alice gizli $\boldsymbol{a = 3}$ sayısını seçsin, $A = g^a \bmod p$ değerini hesaplasın ve Bob'a göndersin.
$$A = 7^3 \bmod 11 = 2$$
3. Bob gizli $\boldsymbol{b = 6}$ sayısını seçer, $B = g^b \bmod p$ değerini hesaplar ve Alice'e gönderir.
$$B = 7^6 \bmod 11 = 4$$
4. Alice $\boldsymbol{s} = B^a \bmod p = 4^3 \bmod 11 = \mathbf{9}$ değerini hesaplar.
5. Bob $\mathbf{s} = A^b \bmod p = 2^6 \bmod 11 = \mathbf{9}$ değerini hesaplar.
6. Sonuç olarak Alice ve Bob $\boldsymbol{s} = \mathbf{9}$ ortak gizli anahtara sahip olurlar.

1977: Ronald Rivest, Adi Shamir ve Leonard Adleman çarpanlara ayırma problemine dayanan RSA şifreleme sistemini tasarladılar (Rivest, Shamir, & Adleman, 1978). RSA şifreleme sistemi Bölüm 5'te detaylı incelenecektir.

1981: Kriptoloji üzerine ilk konferans olan CRYPTO'81 California Santa Barbara Üniversite'sinde gerçekleştirildi.

1990: Xuejia Lai ve James Massey, IDEA blok şifre algoritmasını tasarladılar.

1991: Phil Zimmerman, veri şifreleme ve şifre çözme işlemleri yapan iletişim güvenliğini artırma yöntemlerinden olan PGP (Pretty Good Privacy) sistemini geliştirdi.

1995: SHA-1 (Secure Hash Algorithm 1) kriptografik özet fonksiyonu NIST (National Institute of Standarts and Technology) tarafından yayınlanmıştır (Eastlake & Jones, 2001).

2000: Belçikalı Joan Daemen ve Vincent Rijmen tarafından AES (Advance Encryption Standard- Gelişmiş Şifreleme Standardı) blok şifre algoritması tasarlanmıştır (Daemen & Rijmen, 2002). AES günümüzde veri güvenliğini sağlamak için birçok alanda kullanılan en önemli blok şifre algoritmasıdır. Uluslararası bir şifreleme standardı olması nedeni ile tüm dünyada standart olarak kullanılmaktadır. Örneğin, WhatsApp iletişim uygulamasında uçtan uca şifreleme işlemi AES algoritması ile yapılmaktadır.

2.2. Türkiye'de Kriptoloji Tarihi

Türkiye'de şifreleme cihazları üzerine ilk çalışmalar 1970'li yıllarda TÜBİTAK'a bağlı çalışan Gebze'deki Elektronik Araştırma Ünitesi'nde (EAÜ) Türk Silahlı Kuvvetleri (TSK) için yerli şifre cihazı üretme çalışmaları ile başlamıştır. Bu bölümdeki tarihsel gelişim aşamaları verilirken (URL-20) kaynağından yararlanılmıştır.

1972: Bugünkü adıyla Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü'nün (UEKAE) temeli; 1968 yılında Orta Doğu Teknik Üniversitesi Mühendislik Fakültesi'nde kurulan Elektronik Araştırma Ünitesi'nin (EAÜ) ile atılmıştır.

1975: 1974 yılında Kıbrıs Barış Harekatı sonrasında EAÜ milli çevrimiçi kripto cihazının geliştirilebileceğini raporlamış ve Türk Silahlı Kuvvetleri (TSK) bu doğrultuda yapılacak çalışmaları destekleme kararı almıştır.

1978-1983: MİLON-1 adı verilen ilk milli kripto cihazı üretilmiştir.

1997: UEKAE Kriptanaliz Merkezi faaliyete geçmiştir. Kripto cihazı MİLON-5 ve bu cihazlar için kripto anahtarları üreterek elektronik olarak dağıtan TAFICS Elektronik Anahtar Yönetim Sistemi geliştirilmiştir. Bu yıl aynı zamanda güvenli ses haberleşmesi sağlayan kriptolu telefon MİLSEC-1'in geliştirilmesi tamamlanarak ilk üretimi yapılmıştır.

1999: UEKAE Yarıiletken Teknolojileri Araştırma Laboratuvarında (YİTAL) ilk milli kripto tümdevresi üretilmiştir.

2002: IP Kripto Cihazı'nın (IPKC-E) geliştirilmesi tamamlanarak üretimine başlanmıştır.

2004: Kripto cihazlarında anahtar üreten, saklayan ve yöneten Elektronik Anahtar Dağıtım Sistemi (EKADAS-I) geliştirilmiştir.

2007: TAFICS Elektronik Anahtar Yönetim Sistemi (TELAYS) geliştirilmiştir. TAFICS kripto cihazlarının ihtiyaç duyduğu kripto anahtarlarını üreten, çevrim-içi ve çevrim-dışı güvenli yollardan uç noktadaki kripto cihazlarına dağıtan ve muhasebe bilgilerini saklayan yönetim sistemidir. TELAYS, TAFICS'in güvenliğini sağlayan cihazların tüm anahtar ihtiyaçlarına çözüm sunar.

2008: Milli KAYC-S cihazı ve NATO SECAN onaylı KAYC-S/N anahtar taşıma ve yükleme cihazları geliştirilmiştir. SIR kriptolu USB bellek cihazı geliştirildi ve NATO SECAN onayı almıştır.

2009: Kriptolu Cep Telefonu MİLCEP-K1'in geliştirilmesi tamamlanarak üretimine başlanmıştır. Milli Mesajlaşma Sistemi MEDAS-2 geliştirmesi tamamlanmıştır.

2011: 1 Gbit/s hızında güvenli iletişim sağlayan IP Kripto Cihazının (IPKC-G) geliştirilmesi tamamlanarak üretimine başlanmıştır.

2012: Kriptolu Cep Telefonu MİLCEP-K2'nin geliştirilmesi tamamlanarak üretimine başlanmıştır.

2015: SIR-S Kriptolu Taşınabilir Sabit Disk geliştirilmesi tamamlanmıştır.

2016: Ülkemizde İlk Kuantum Tabanlı Rastgele Sayı Üretim Cihazının geliştirmesi tamamlandı. İlk Milli özet algoritması geliştirmesi tamamlanmıştır.

2017: GÖKTÜRK-1 uydusu ve yer sistemleri için kriptolu haberleşme sistemi geliştirilmiştir.

2018: 10 GBit veri işleme kapasitesine sahip IP Kripto cihazının geliştirilmesi tamamlanmıştır. SIR-II kriptolu USB bellek cihazı geliştirilmiştir.

Şekil 2.2.1. Mılon-4a: İlk Türk kripto cihazı (URL-20, 2021)

Ülkemizde 2005 yılından itibaren Orta Doğu Teknik Üniversitesi'nde Ulusal Kriptoloji Sempozyumu (ISCTURKEY) düzenlenmektedir. Bu alanda üniversitelerde birçok tez ve makaleler yazılmıştır. (Erhan, 1993) tarafından yapılan çalışmada, RSA algoritması kullanarak kişisel bilgisayarlarda dosya güvenliğini sağlamak amacıyla bir açık anahtar şifreleme yazılımı tasarlanmıştır. (Gül, 1997) tarafından yapılan çalışmada, RSA tabanlı açık anahtarlı şifreleme sistemini kullanarak ortak anahtarlı bir kripto sistem önerilmiştir. (Hassanpour, 2015) tarafından asal sayıların şifreleme üzerindeki uygulamaları incelenmiştir.

3. TEMEL KAVRAMLAR

Bu bölümde, çalışma boyunca cebirsel yapılarla ilgili kullanılan temel işlemler ve özellikler, bazı matematiksel kavramlara, tanım ve teoremlere yer verilecektir.

3.1. Sayılar Teorisi

Bu bölümde pozitif tam sayıların özellikleri ve bunlarla ilgili işlemler verilecektir. Çalışma boyunca kullanılacak olan algoritmalar tanımlanacak ayrıca özel örneklerde ve algoritmalarda kullanılacak olan modüler aritmetik ayrıntılı bir şekilde verilecektir. Algoritmalar, aksi belirtilmedikçe (Erdoğan & Yılmaz, 2008) çalışmasından alınmıştır.

Bölme Algoritması: Her sıfırdan farklı a ve b tam sayıları için, $0 \leq r < |b|$ olmak üzere,

$$a = q \cdot b + r$$

olacak şekilde q ve r tam sayıları belirlenebilir. r sayısı a sayısının b ile bölümünden elde edilen kalandır. $r = 0$ durumunda b, a yı böler denir ve $b|a$ ile gösterilir.

Bölünebilirlik: a ve b tam sayılar olmak üzere eğer, $a = q \cdot b$ eşitliğini sağlayan bir q sayısı varsa b, a 'yi böler denir ve $b|a$ şeklinde gösterilir.

Örnek 3.1.1. 16 sayısı, 48 sayısını böler. 16 | 48 ile gösterilir. Çünkü $48 = 16 \cdot 3$ tür.

Bölünebilirlik Özellikleri: $a, b, c \in \mathbb{Z}$ tam sayıları için bölünebilirlik özellikleri aşağıda verilmiştir.

1. $a|a$
2. Eğer, $a|b$ ve $b|c$ ise $a|c$ dir.
3. Eğer, $a|b$ ve $a|c$ ise bütün $x, y \in \mathbb{Z}$ için $a \mid bx + cy$ ifadesi doğrudur.
4. Eğer, $a|b$ ve $b|a$ ise $a = \pm b$ dir.

Logaritma Fonksiyonu: $a \in \mathbb{R}^+ - \{1\}$ olmak üzere, $f: \mathbb{R} \to \mathbb{R}^+, f(x) = a^x$ biçiminde tanımlanan üstel fonksiyonun ters fonksiyonuna logaritma fonksiyonu denir.
$f^{-1}: \mathbb{R}^+ \to \mathbb{R}$, $f^{-1}(x) = \log_a x$ şeklinde gösterilir. $f^{-1}(x) = \log_a x$ fonksiyonunda taban $a = 10$ alınırsa,

$$f^{-1}: \mathbb{R}^+ \to \mathbb{R}, f^{-1}(x) = \log_{10} x = \log x$$

$f^{-1}(x)$ fonksiyonuna onluk logaritma fonksiyonu denir ve kısaca $\log x$ biçiminde gösterilir.

Algoritmaların Zaman Karmaşıklığı (Time Complexity of Algorithms)

Belirli bir problemi çözmek için birden fazla algoritma verilebilmektedir. Problemi çözmek için hangi algoritmanın daha yararlı olduğuna karar vermek için algoritmaların zaman karmaşıklığının (verimliliğinin) hesaplanması gerekmektedir. Bir algoritmaların zaman karmaşıklığı algoritmanın maliyeti, zaman tahmini ve zaman hesabı olarak da adlandırılmaktadır. Bir algoritmanın zaman hesabı algoritma için gereken işlem sayısı ile orantılıdır. Zaman hesabı için girdinin boyutu önemli bir parametredir. Algoritmanın karmaşıklığı girdi değeri olan n sayısı ve n'nin bit sayısı olan k sayısına bağladır. Bir n sayısı ikilik tabanda $k = \lfloor \log_2 n \rfloor + 1$ bitlik bir sayıdır.

Bir algoritmanın zaman tahmini, büyük O-gösterimi ile "asimptotik olarak" ölçülmektedir. f ve $g: \mathbb{N} \to \mathbb{R}^+$ iki fonksiyon olmak üzere, eğer pozitif bir c sabiti ve pozitif bir n_0 tam sayısı varsa öyle ki tüm $n \geq n_0$ için $f(n) \leq c\, g(n)$ olmak üzere $f(n) = O(g(n))$ olarak gösterilir ve buna "f, g'nin büyük-O'su" denir.

Büyük-O gösterimini kullandığımızda, yapılacak işlem sayısı işlemcinin kelime uzunluğundan bağımsızdır. k, W tabanındaki n'nin basamak sayısı olmak üzere $n = (d_{k-1} \dots d_1 d_0)_W$ dir. Dolayısıyla, $k = \lfloor \log_W n \rfloor + 1 = \left\lfloor \frac{\log n}{\log W} \right\rfloor + 1 \leq c \log n$ dir. Özel olarak, $W = 2$ için geleneksel bilgisayar mimarisinde n sayısının bit sayısı $k = O(\log n)$ şeklinde ifade edilebilir (Akyıldız, Cenk, & Sınak, 2021).

Polinom Zamanlılık (Polinomial Time). Sayılar teorisinde ve kriptografide, bir problemin çözümü için kullanılan algoritmanın çalışma süresi problemin girdisine ve algoritmanın adım sayısına bağlıdır. Çalışma süresi (zaman karmaşıklığı) girdinin büyüklüğüne (bit sayısına) bir polinom cinsinden bağlı olan algoritmaya polinom zamanda çalışan algoritma adı verilir. Diğer bir ifade ile algoritmanın girdisi olan n sayısı $k = O(\log n)$ bitlik bir sayı ve c pozitif sabit olmak üzere algoritmanın problemi çözmek için yaptığı işlem sayısı $O\,(k^c)$ ise bu algoritma polinom zamanlı algoritma olarak adlandırılır. Polinom zamanda çözülebilen (yani, polinom zamanlı algoritma ile çözülebilen) problemlere P sınıfı problemler denir. Bu tez çalışmasında verilen asal sayı test yöntemleri polinom zamanlı çalışan algoritmalardır.

En Büyük Ortak Bölen (Greatest Common Divisor) Kavramı: $a, b \neq 0$ iki tam sayı olmak üzere, a ve b sayılarının en büyük ortak böleni, a ve b yi bölen en büyük d tam sayısıdır. a ve b sayılarının en büyük ortak böleni $\text{ebob}\,(a, b)$ ile gösterilir. Bu çalışma boyunca en büyük ortak bölen ifadesi $\text{obeb}\,(a, b)$ ile gösterilecektir.

Örnek 3.1.2. 24 ve 84 sayılarının en büyük ortak bölenlerini bulalım. 24 ve 84 sayılarının pozitif ortak bölenleri, 1, 2, 3, 4, 6, 12 sayılarıdır ve bunların en büyüğü 12 olduğundan, obeb (24, 84) = 12 dir.

Öklid Algoritması: Pozitif a ve b tam sayıları $a > b$ olarak verildiğinde bölme algoritması kullanılarak,

$$a = q_0 b + r_0$$

olarak yazılır.

$$\begin{aligned}
a &= q_0 b + r_0, && 0 < r_0 < b \\
b &= q_1 r_0 + r_1, && 0 < r_1 < r_0 \\
r_0 &= q_2 r_1 + r_2, && 0 < r_2 < r_1 \\
&\vdots \\
r_{k-3} &= q_{k-1} r_{k-2} + r_{k-1}, && 0 < r_{k-1} < r_{k-2} \\
r_{k-2} &= q_k r_{k-1} + r_k, && r_k = 0
\end{aligned}$$

olacak şekilde q_i ve r_i $(i = 0, 1, \dots, k)$ tam sayıları bulunur. Son bölme işleminden $r_{k-1} \mid r_{k-2}$ olduğundan, $(r_{k-2}, r_{k-1}) = r_{k-1}$ elde edilir. Buradan a ile b sayılarının obeb değeri Öklid algoritmasındaki 0 dan farklı en son elde edilen kalandır. Öklid algoritması aşağıda Algoritma 1'de verilmiştir.

Algoritma 1: $d = \text{obeb}\,(m,n)$ hesaplamak için Öklid algoritması

Girdi: $m, n \in Z^+, n \geq m$
Çıktı: $d = \text{obeb}\,(m,n)$

$k \leftarrow -1$, $r_0 \leftarrow n$ ve $r_1 \leftarrow m$
while $r_{k+2} \neq 0$
$k \leftarrow k + 1$
$q_{k+1} \leftarrow \left\lfloor \frac{r_k}{r_{k+1}} \right\rfloor$
$r_{k+2} \leftarrow r_k - q_{k+1} r_{k+1}$
end while
return $d = r_{k+1}$

Algoritma 1 de verilen Öklid algoritmasının karmaşıklığı $O(log^3 n)$ işlemdir (Akyıldız, Cenk, & Sınak, 2021).

Örnek 3.1.3. $m = 49$ ve $n = 91$ pozitif tam sayıları için $d = \text{obeb}\,(91, 49)$ değerini Öklid algoritması ile bulalım.

1. $r_0 \leftarrow 91\ ve\ r_1 \leftarrow 49$ için kalan değeri $k = 42$ tür.
2. $k = 42 \neq 0$ olduğu için,
3. $r_0 \leftarrow 49$ ve $r_1 \leftarrow 42$ yeni değerler olur.
4. $r_0 \leftarrow 49$ ve $r_1 \leftarrow 42$ için kalan değeri $k = 7$ olur.
5. $k = 7 \neq 0$ olduğu için,
6. $r_0 \leftarrow 42$ ve $r_1 \leftarrow 7$ yeni değerler olur.
7. $r_0 \leftarrow 42$ ve $r_1 \leftarrow 7$ için kalan değeri $k = 0$ olur.
5. $k = 0$ olduğu için $d = \text{obeb}\,(91, 49) = 7$ dir.

Diğer yandan Öklid algoritmasının tersi $d = \text{obeb}\,(m,n) = vm + un$ denklemindeki $v, u \in \mathbb{Z}$ sayılarını verir.

Algoritma 2: Ters Öklid Algoritması

Girdi: m, n, d, q_i ve $i = 1, \ldots, k$
Çıktı: v, u

$v_k \leftarrow q_k, u_{k-1} \leftarrow 1, q_0 \leftarrow 0$
$i = k - 1'$ i 1 yapmak için,
$v_i \leftarrow u_i - q_i v_i$+1
$u_{i-1} \leftarrow v_{i+1}$
end for
$v \leftarrow v_1$
$u \leftarrow u_0 - q_0 v_1$
return v, u

Algoritma 2 de verilen ters Öklid algoritmasının karmaşıklığı $O(log^3 n)$ işlemdir (Akyıldız, Cenk, & Sınak, 2021).

Tanım 3.1.1. $A\ x\ B$ nin boş olmayan her alt kümesine A dan B ye bir bağıntı denir.

Tanım 3.1.2. A ve B iki küme olmak üzere, A dan B ye olan bir f bağıntısı aşağıdaki özelliği sağlarsa f bağıntısına A dan B ye bir fonksiyon denir ve $f: A \rightarrow B$ şeklinde gösterilir.

- A kümesindeki her elemanın f altında bir ve yalnız bir tek görüntüsü olmalıdır.

Diğer bir ifade ile her bir $a \in A$ için $a\ f\ b$ olacak biçimde bir tek $b \in B$ var ise f bağıntısına A dan B ye bir fonksiyon denir. Burada A kümesine tanım kümesi, B kümesine de değer kümesi denir.

Tanım 3.1.3. m bir pozitif tam sayı olmak üzere eğer $m \mid (a-b)$ ise a sayısı b tam sayısına m modülüne göre denktir denir ve $a \equiv b \bmod m$ şeklinde gösterilir.

Tanım 3.1.4. $\mathbb{Z}$ deki " $\equiv$ " denklik bağıntısının belirttiği denklik sınıflarına, m modülüne göre $(\bmod m)$ kalan sınıfları denir ve tüm kalan sınıfları kümesi $\mathbb{Z}_m$ ile gösterilir.

Tanım 3.1.5. Boştan farklı bir S kümesi üstünde bir denklik bağıntısı aşağıdaki koşulları gerçekleyen $S\ x\ S$ nin alt kümesidir. $x, y, z \in S$ için,
1) $x \sim x$ (Yansıma özelliği)
2) $x \sim y \ \rightarrow\ y \sim x$ (Simetri özelliği)
3) $x \sim y\ ve\ y \sim z\ ise\ x \sim z$ (Geçişme özelliği)

Burada bir $x \in \mathbb{Z}$ nin denklik sınıfı $\bar{x} = y \in \mathbb{Z} : x \equiv y \bmod m$ şeklinde tanımlanır. Bu durumda bu denklik sınıflarından oluşan $Z_m = \{\bar{0}, \bar{1}, \bar{2}, \ \dots\ , \overline{m-1}\}$ kümesine kalan sınıfları sistemi denir (Külen, 2013).

Tanım 3.1.6. $\mathbb{Z}_n = \{1, \dots, n-1\}$ aralığında tam sayılardan oluşan bir küme olmak üzere $\mathbb{Z}_n^* = \{\bar{a} \in \mathbb{Z}_n | \text{obeb}\ (a,n) = 1\}$ şeklinde tanımlanan küme, n'den küçük ve n ile aralarında asal olan tam sayıların denklik sınıflarının oluşturduğu kümedir.

Tanım 3.1.7. Her $m > 0$ tam sayısını, m yi geçmeyen ve m ile aralarında asal olan tamsayıların sayısına eşleyen fonksiyona Euler'in φ -fonksiyonu adı verilir ve $\varphi(m)$ ile gösterilir. Sonuç olarak, $\varphi(m) = \#\mathbb{Z}_m^*$ şeklinde de tanımlanabilir.

Teorem 3.1.2. (Euler Teoremi) **:** $m \in \mathbb{Z}^+, a \in \mathbb{Z}$ ve $\text{obeb}(a,m) = 1$ olmak üzere $a^{\varphi(m)} \equiv 1 \bmod m$ dir.

Sonuç 3.1.1. (Fermat'ın Küçük Teoremi) **:** $a \in \mathbb{Z}$, p asal sayı ve $\text{obeb}(a,p) = 1$ olmak üzere $a^{p-1} \equiv 1 \bmod p$ dir.

Grup: G boş olmayan bir küme, "*" G üzerinde tanımlı bir ikili işlem olmak üzere, eğer aşağıdaki ilk dört şartı sağlanıyorsa $(G,*)$ cebirsel yapısına bir grup denir.

$G1)$ $\forall a, b \in G$ için $a * b \in G$ (kapalılık özelliği)
$G2)$ $\forall a, b, c \in G$ için $(a * b) * c = a * (b * c)$ (birleşme özelliği)
$G3)$ $\forall a \in G$ için $a * e = e * a$ olacak şekilde $e \in G$ vardır. (birim eleman özelliği)
$G4)$ $\forall a \in G$ için $a * b = b * a = e$ olacak şekilde $b \in G$ vardır. (ters eleman özelliği)
$G5)$ $\forall a, b \in G$ için $a * b = b * a$ (değişme özelliği)

Ayrıca bu grup 5. özelliği de sağlıyorsa "değişmeli grup" olarak adlandırılır.

Halka: H kümesi boştan farklı bir küme ve "+" ve "*" işlemleri H üzerinde tanımlanan ikili işlemler olmak üzere,

$H1)$ H kümesi "+" işlemine göre bir değişmeli gruptur.

$H2)$ $a, b, c \in H$ için, $a * (b * c) = (a * b) * c$ birleşme özelliği vardır.
$H3)$ $a, b, c \in H$ için,
$a * (b + c) = (a * b) + (a * c)$ ve $(b + c) * a = (b * a) + (c * a)$ sağdan ve soldan dağılma özelliği vardır.
Bu özellikler sağlanırsa H kümesine "+" ve "*" işlemlerine göre halka denir ve $(H, +, *)$ şeklinde gösterilir.

Cisim: F kümesi boştan farklı bir küme ve "+" ve "*" işlemleri F üzerinde tanımlanan ikili işlemler olsun. "+" işleminin birim elemanını 0_F ile gösterelim. Eğer aşağıdaki şartlar sağlanıyorsa $(F, +, *)$ cebirsel yapısına bir cisim denir.

$F1)$ $(F, +)$ değişmeli gruptur.
$F2)$ $(F \setminus \{0_F\}, *)$ değişmeli bir gruptur.

Örnek olarak toplama ve çarpma işlemleriyle $\mathbb{Q}$, $\mathbb{R}$ ve $\mathbb{C}$ halkalarının birer cisim oldukları söylenebilir.

3.2. Asal Sayılar

Çarpanlara Ayırma Problemi olarak karşımıza çıkan problemin zorluğu üzerine dayanan şifreleme algoritmasının güvenilir olması için yeteri kadar büyük ve asal sayı üretilmesi çalışmanın esas konusu olacaktır. Bu kısımda, asal sayıların özelliklerinden başlayarak, asal sayı çeşitleri ve asal sayıların önemi hakkında bir kaynak taraması yapılacaktır.

Tanım 3.2.1. 1 den büyük, 1 ve kendisinden başka böleni olmayan tam sayılara "asal sayı" denir. Asal olmayan sayılara ise "bileşik sayı" denir.

Örnek 3.2.1. 2, 3, 5, 7, 11, 13, 17,… asal sayılardır.

- 0 ve 1 asal sayı olarak kabul edilmez.
- Asal sayılar kümesinin sonsuz elemanı vardır.
- En küçük asal sayı 2 dir ve 2'den başka çift asal sayı yoktur.
- En büyük asal sayı, 2020 yılında Mersenne Asalları Büyük İnternet Araştırması (GIMPS) projesindeki gönüllüler ile sürdürülen çalışmalar sonucunda bulundu. 24.862.048 basamaklı bilinen en büyük asal sayı $2^{82589933} - 1$ sayısıdır (PrimeGrid, 2005).

Aralarında Asal Sayılar: a ve b iki tam sayısının 1 den başka ortak böleni yoksa bu sayılara aralarında asal sayılar denir ve $\text{obeb}(a, b) = 1$ ile gösterilir.

Aritmetiğin Esas Teoremi: Her $n \geq 2$ tam sayısı asal sayıların çarpımı şeklinde tek türlü yazılır. Diğer bir ifadeyle, n sayısı, p_k sayıları farklı asal sayıları e_k sayıları pozitif tam sayıları göstermek üzere,

$$n = p_1^{e_1} \cdot p_2^{e_2} \cdot \ldots \cdot p_k^{e_k}$$

şeklinde yazılır.

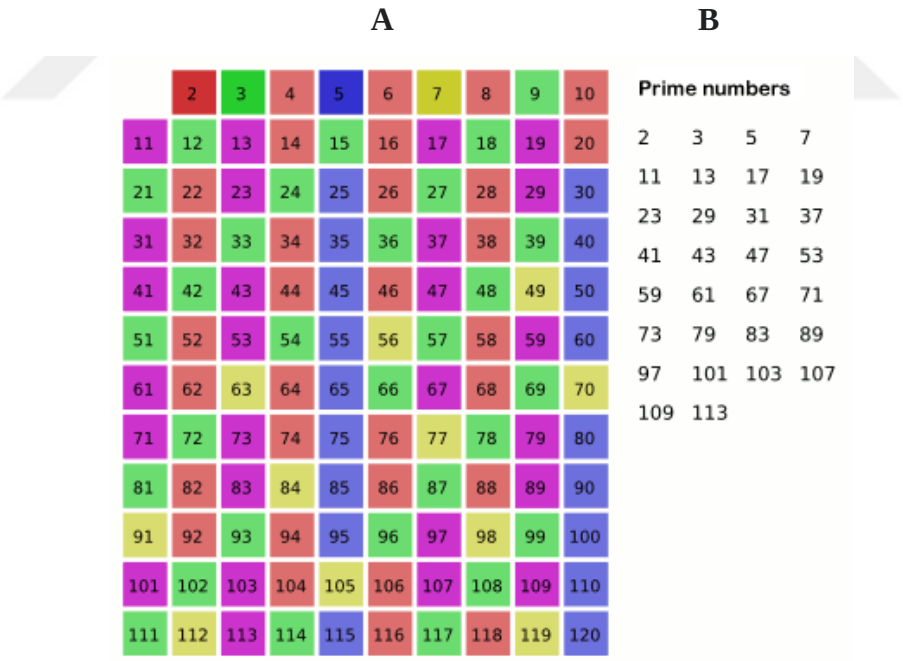
3.2.1. Asal Sayıların Tarihi

Asal sayılar ve özellikleri detaylı olarak ilk kez M.Ö. 500-300 yılları arasında antik Yunanlı Pythagoras okulunun matematikçileri tarafından incelenmiştir. M.Ö. 200 yılında Eratosthenes, asal sayıları bulmak için aşağıda verilen kalbur yöntemini geliştirmiş ve bu yönteme "Sieve of Eratosthenes" adını vermiştir.

Eratosthenes Kalburu: Eratosthenes kalburu, belirli bir tam sayıya kadar olan asal sayıların bulunması için kullanılan pratik olmayan bir yöntemdir. Antik Yunanistan'da Eratosten tarafından geliştirilmiştir. Çalışma sistemini açıklayalım.

1. Bir tabloya 2'den başlayarak, istenilen büyüklükte bir tam sayıya kadar olan tüm tam sayılar yazılır. Bu tabloya A tablosu diyelim.
2. Yanına bir liste oluşturulur ve A'daki ilk asal sayı olan 2 listeye eklenir. Bu listeye B listesi diyelim (resmin sağında bulunan liste).
3. A'dan 2 ve 2'nin tüm katları silinir.
4. A'da kalan ilk tek sayı asaldır. Bu sayı B'ye eklenir.
5. Bu sayı ve tüm katları A'dan silinir.
6. A tablosunda herhangi bir sayı kalmayıncaya kadar 4. ve 5. Adımlar tekrarlanır.
7. Tabloda kalan sayılar B listesine yazılır.

1'den seçilen sayıya kadar olan asal sayılar bulunmuş olur.



Şekil 3.2.1.1. 1'den 120 ye kadar olan asal sayıları bulma (URL-18, 2021)

Metot: $a > 1$ bir tam sayı olsun. Bu sayı eğer bölünebilir bir sayı ise $c < b < a$ ve $1 < c < a$ olmak üzere $a = b \cdot c$ şeklinde yazılabilir. Bütünlüğü bozmadan $c \leq b$ olduğu varsayılsın. O zaman,

$$c^2 \leq c \cdot b = a \rightarrow c \leq \sqrt{a}$$

Aritmetiğin esas teoremini kullanarak c sayısını bölen ve $p \leq c \leq \sqrt{a}$ koşulunu sağlayan bir p sayısı bulunur. Öyle ki burada p asal sayısı c sayısını böldüğü ve c sayısı da a sayısını böldüğü için p, a sayısını da böler.

Algoritma 3: Eratosthenes Kalburu algoritmasının gösterimi

Girdi: $a > 1$ tam sayı
A, 0 dan a ya tam sayı dizisi
$A[0] = 1$
$A[1] = 1$

for $i = 2,3,4, \ldots,$ için $\sqrt{a}$ yi aşmayanlar :
  if $A[i]$ is 0 :
    for $j = i^2, i^2 + i, i^2 + 2i, i^2 + 3i, \ldots, a'$yı aşmayanlar :
      $A\,[j] = 1$
    end for
  end if
end for

(Turan & Nacar, 2016)

Eratosten algoritmasında A dizisinin her biri başlangıçta asal olarak kabul edildikten sonra ilk iki dizi elemanı ($A[0]$ ve $A[1]$) asal olmadığından 1 olarak işaretlenir ve 2'den başlayarak her sayının kendi karesinden küçük ve 1'den büyük olan katları için dizideki sıra değeri 1 olarak işaretlenir. Elde edilen dizideki 1'den farklı olan her 0 değeri bir asal sayıyı temsil etmektedir.

Eratosthenes Kalburu örneğini verelim.

Örnek 3.2.1.1. $a = 181$ sayısının asal olup olmadığını Eratosthenes Kalburu ile inceleyelim. 181 sayısı $13^2 = 169$ ve $14^2 = 196$ sayıları arasındadır.

$$13 < \sqrt{181} < 14$$

181 sayısını bölebilecek asal sayılar 2, 3, 5, 7, 9, 11, 13 olabilir. Bu sayıların 181 sayısını bölüp bölmediği kontrol edilir. Hiç birisi 181 sayısını bölmediği için 181 asal sayıdır.

3.2.2. Asal Sayı Çeşitleri

Asal sayılar konusunda matematik ve bilgisayar başta olmak üzere birçok bilim dalına faydası olacak çözülmeyi bekleyen birçok problem bulunmaktadır. Bu bölümde asal sayı çeşitlerinden bazılarına yer verilmiş ve örneklerle açıklanmıştır.

Fermat Asalları. $F_n = 2^{2^n} + 1,\ (n \in \mathbb{N})$ şeklinde yazılan sayılara Fermat Sayıları denir. Örneğin,

$$F_0 = 3$$
$$F_1 = 5$$
$$F_2 = 17$$
$$F_3 = 257$$
$$F_4 = 65537$$
$$\vdots$$

Fermat 1732 yılına kadar bu şekilde yazılan tüm sayıların asal olduğunu sanıyordu. Fermat sayılarının asal olduğunu ispatlamaya çalıştı ama başaramadı.

1732 yılında Leonhard Euler $F_5 = 4294967297$ sayısını $F_5 = 641 \cdot 67004$ şeklinde çarpanlara ayırarak bu sanıyı çürütmüştür (Nesin, 2019). Daha sonra, Lucas, F_6 Fermat sayısının asal olmadığını kanıtlamıştır ve 1880 yılında Landry bu sayının asal çarpanlarını

$$F_6 = 18446744073709551617 = 274177 \cdot 67280421310721$$

olarak bulmuştur.

Wilfrid Keller 1980 yılında F_{9448} Fermat sayısının asal olmadığını göstermiştir. Bu Fermat sayısı $19 \cdot 2^{9450} + 1$ sayısına bölünmektedir.

1984 yılında yine W. Keller F_{23471} sayısının asal olmadığını ispatlamıştır. Bu sayının 10^{7000} den fazla basamağı vardır ve $5 \cdot 2^{23473} + 1$ sayısına tam bölünür (Nesin, 2019). Sonuç olarak $n > 4$ için asal bir Fermat sayısı olup olmadığı hala açık bir problemdir.

Palindromik Asallar. Soldan ve sağdan okunuşları aynı olan sayılara palindromik sayılar ve bu şekilde yazılan asal sayılara da palindromik asal sayılar denir (Honaker & Caldwell, 1999). Bazı palindromik asal sayılar; 2, 3, 5, 7, 11, 101, 131, 151, 181, 191, 313, 353, 373, 383, 727, 757, 787, … sayılarıdır. Bilinen en büyük palindromik asal sayı 1888529 basamaklı, $10^{1234567} - 20342924302 \cdot 10^{617278} - 1$ sayısı olup 18 Ekim 2021'de Ryan Propper ve Serge Batalov tarafından bulunmuştur.

Genelleştirilmiş Fermat Asalları. a^{2^n}+1 şeklinde olan asallardır. Bu sayının asal olabilmesi için a bir çift sayı olmak zorundadır. Bilinen en büyük genelleştirilmiş Fermat asalı 31 Ekim 2018 yılında bulunan $1059094^{1048576} + 1$ sayısıdır ve 6317602 basamaklıdır (PrimeGrid, 2005).

Sophie Germain Asalları. p ve $2p + 1$ sayılarının ikisi birden asal sayı ise p asal sayılarına Sophie Germain asalı denir (Takashi, 2000). 2, 3, 5, 7, 11, 23, 29, 41, 53, 83, 89, … sayılar Sophie Germain asallarına örnek verilebilir.

Faktöriyel Asallar. $n! \pm 1$ şeklindeki sayılara Faktöriyel asallar denir. 25 Temmuz 2016 da bulunan $208003! - 1$ asal sayısı 1.015.843 basamaklı en büyük faktöriyel asal sayısıdır (PrimeGrid, 2005).

Cullen Asalları. $n \cdot 2^n + 1$ şeklindeki asallara Cullen asal sayıları denir. Bilinen en büyük Cullen asalı $6679881 \cdot 2^{6679881} + 1$ sayısı olup, 20010852 basamaklıdır.

$n \cdot b^n + 1$ şeklindeki asallara da Genelleştirilmiş Cullen asalları denir. Bilinen en büyük G. Cullen asalı 8 Ağustos 2021 tarihinde bulunan $2525532 \cdot 73^{2525532} + 1$ sayısı olup, 4.705.888 basamaklıdır (PrimeGrid, 2005).

Mersenne Asalları. Asal bir p sayısı için $M_p = 2^p - 1$ şeklinde yazılan sayılara Mersenne sayıları denir. Eğer M_p Mersenne sayısı asal ise Mersenne asal sayısı denir (Robinson, 1954).

$p = 2$ için $M_2 = 2^2 - 1 = 3$
$p = 3$ için $M_3 = 2^3 - 1 = 7$
$p = 5$ için $M_5 = 2^5 - 1 = 31$
$p = 7$ için $M_7 = 2^7 - 1 = 127$
⋮

3, 7, 31, 127, … sayılarına Mersenne asal sayıları denir. 7'den sonraki ilk asal sayı olan 11 için M_{11} sayısı asal sayı değildir. Çünkü $M_{11} = 2047 = 23 \cdot 89$ dir.

Her p asal sayısı için M_p sayısının asal olmadığı görülmektedir. Mersenne asalların sonsuz sayıda olup olmadığı bilinmemektedir. Şu ana kadar bulunan en büyük asal sayı 7 Aralık 2020 de bulunan 24.862.048 basamaklı $2^{82589933} - 1$ Mersenne asal sayısıdır (PrimeGrid, 2005).

Carmichael Sayıları. Fermat'ın küçük teoremine göre, n sayısının asal sayı olabilmesi için $a^n - a$ 'yı bölmesi gerekmektedir, fakat bu bölme işlemini sağlayan asal olmayan sayılar da vardır. Bu sayılara Carmichael sayıları denir. İlk olarak 1910 yılında R. D. Carmichael bu kriterlere uyan sayıların bir kısmını bulmuştur. Carmichael sayılarından bazıları 561, 1105, 1729, 2465, 2821, 6601, 8911, 10585, 15841, 29341, … sayılarıdır.

Wilson Asalları.
Wilson Teoremi: p asal sayısı için, $(p-1)! \equiv -1 \; mod \; p$ dir. Literatürde, bu teorem yardımıyla elde edilen Wilson katsayısı $W(p) = \frac{(p-1)!+1}{p}$ olarak bilinmektedir. $W(p) \equiv 0 \; mod \; p$ veya $(p-1)! \equiv -1 \; mod \; p$ eşitliğini sağlayan p sayısına Wilson asal sayısı denir. $p = 5$, $p = 13$ Wilson asallarına örnektir (Ribenboim, 2004).

3.2.3. Asal Sayıların Önemi

Matematiğin temel yapısı olan sayılara baktığımızda her sayının asal sayıların çarpımından oluştuğu görülmektedir. Sonsuz olduğu bilinen asal sayılar hakkında birçok teorem ispatlanmış ve asal sayı bulmak için çeşitli algoritmalar geliştirilmiştir.

Asal sayıların gizeminin hala çözülememesi ise matematik ve bilgisayar bilimi için bu alana olan ilgiyi artırmaktadır. Açık anahtarlı kriptosistemlerin güvenilir olması için yeteri kadar büyüklükte asal sayılar kullanmak gerekmektedir.

Kriptografik uygulamalarda bazı şifreleme ve imzalama sistemlerinin güvenli olabilmesi için çok büyük asal sayılara ihtiyaç vardır. Küçük sayıların asal olup olmadığını belirleyebilmek mümkün olsa da sayıların büyüklüğü arttıkça bunun belirlenmesi uzun sürmektedir. Büyük sayıların asal olup olmadıklarını anlamak için daha gelişmiş asallık testleri gerekmektedir. Asal sayılar üzerine yapılan çalışmaların başında asallık testleri gelmektedir.

3.3. Kuadratik Rezidüler

p tek asal sayı olmak üzere $\mathbb{Z}_p$ toplama ve çarpma işlemleri ile birlikte bir cisim oluşturur. Bu cisim kısaca $\mathbb{Z}_p$ veya F_p ile gösterilebilir. $a \in F_p^*$ elamanlarının kuadratik rezidü olup olmadığı belirlenebilmektedir. Eğer $x^2 \equiv a \bmod p$ denkliğinde $\sqrt{a} \in F_p^*$ olacak şekilde $x \in F_p^*$ çözümü varsa a, p modülüne göre kuadratik rezidüdür; aksi halde, $x^2 \equiv a \bmod p$ denkliğini sağlayan $\sqrt{a} \in F_p^*$ olacak şekilde bir $x \in F_p^*$ sayısı yok ise a, p modülüne göre kuadratik non-rezidüdür.

İlk olarak, $x^2 \equiv a \bmod p$ denkliğinin F_p^* cisminde bir çözümünün olup olmadığını, belirlemek için Legendre ve Jacobi sembolleri verilmektedir.

Legendre Sembolü. $p > 2$ asal, $a \in \mathbb{Z}^+$ ve $x \in \mathbb{Z}_p^*$ olmak üzere, Legendre sembolü $\left(\frac{a}{p}\right) = 1$ ise, a sayısı $\bmod p$ ye göre kuadratik rezidü olarak adlandırılır. Diğer bir ifade ile verilen a sayısı $\bmod p$ de bir x sayının karesi şeklinde yazılabilmektedir, bu durum $x^2 \equiv a \bmod p$ şeklinde ifade edilir. Eğer Legendre sembolü $\left(\frac{a}{p}\right) = -1$ ise, a sayısı $\bmod p$ ye göre kuadratik rezidü değildir (non-rezidü). Diğer bir ifade ile verilen a sayısı $\bmod p$ de x in karesi şeklinde yazılamamaktadır, bu durum $x^2 \not\equiv a \bmod p$ şeklinde ifade edilmektedir.

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{eğer } x^2 \equiv a \bmod p \text{ ise} \\ -1 & \text{eğer } x^2 \not\equiv a \bmod p \text{ ise} \\ 0 & \text{eğer } a \mid p \text{ ise} \end{cases}$$

Legendre sembolü, a, b pozitif tamsayıları ve $p > 2$ asal sayısı için aşağıdaki özellikleri sağlamaktadır.

1. Her kuadratik rezidünün iki kökü vardır.
2. $\left(\frac{a}{p}\right) = a^{\frac{p-1}{2}} \bmod p$ dir.
3. $\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right)$ dir.

Örnek 3.3.1. Verilen $a = 8$ ve $p = 41$ sayıları için Legendre sembolünü hesaplayalım.

$$\left(\frac{8}{41}\right) = \left(\frac{4}{41}\right) = \left(\frac{2}{41}\right) = \left(\frac{1}{41}\right) = 1$$

$x = 7$ için $8 \equiv 7^2 \bmod 41$ denkliği sağlanır, dolayısıyla $a = 8$ sayısı $\bmod 41$'de kuadratik rezidüdür.

Asal Olmayan (Kompozit) Modüle Göre Kuadratik Rezidüler. p, q asal sayılar olmak üzere $n = p \cdot q$ olsun. $\mathbb{Z}_n$ grubundaki kuadratik rezidülere bakacak olursak,

$$\mathbb{Z}_n \cong \mathbb{Z}_p \times \mathbb{Z}_q$$

izomorfizması ve Çin kalan teoremi yardımı ile verilen bir tam sayının $\bmod n$ ye göre kuadratik rezidü olup olmadığı kolayca bulunabilir. Aşağıda Legendre sembolünü daha verimli hesaplamak için Jacobi sembolü verilmektedir.

Jacobi Sembolü. Jacobi sembolü, Legendre sembolünün genelleştirilmiş halidir. Asal olmayan bir n sayısı ve $a \in \mathbb{Z}_n$ için $J = \left(\frac{a}{n}\right)$ olarak tanımlanmaktadır. $\mathbb{Z}$ de $x^2 \equiv a \bmod n$ denkliğinde çözümün olup olmadığını kontrol etmeye yarar.

Jacobi sembolü, a, b pozitif tamsayıları ve m, n tek pozitif tam sayıları için aşağıdaki özellikleri sağlamaktadır.

1. Jacobi sembolü çarpma özelliğine sahiptir.
$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right) \cdot \left(\frac{b}{n}\right)$$
2. $\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$ dir.

3. Jacobi sembolü için ikinci dereceden karşılıklılık yasası

$$\left(\frac{m}{n}\right) = (-1)^{\frac{(m-1)(n-1)}{4}} \left(\frac{n}{m}\right)$$

şeklinde ifade edilir.

Jacobi sembolünün hesaplanması sonucunda $\left(\frac{a}{n}\right) = 1$ ise a, $\bmod n$ ye göre kuadratik rezidüdür. Diğer bir ifade ile verilen a sayısı $\bmod n$ de bir x sayının karesi şeklinde yazılabilmektedir, bu durum $x^2 \equiv a \bmod n$ şeklinde ifade edilmektedir. Eğer Jacobi sembolü $\left(\frac{a}{n}\right) = -1$ ise, a sayısı $\bmod n$ ye göre kuadratik rezidü değildir. Diğer bir ifade ile verilen a sayısı $\bmod n$ de x in karesi şeklinde yazılamamaktadır, bu durum $x^2 \not\equiv a \bmod n$ şeklinde ifade edilmektedir.

$$\left(\frac{a}{n}\right) = \begin{cases} 1 & \text{eğer } x^2 \equiv a \bmod n \text{ ise} \\ -1 & \text{eğer } x^2 \neq a \bmod n \text{ ise} \\ 0 & \text{eğer } a \mid n \text{ ise} \end{cases}$$

Örnek 3.3.2. $a = 1201$ ve $n = 1453$ için Jacobi sembolünü hesaplayalım.

$$\left(\frac{1201}{1453}\right) = \left(\frac{252}{1201}\right) = \left(\frac{126}{1201}\right) = \left(\frac{63}{1201}\right) = \left(\frac{4}{63}\right) = \left(\frac{2}{63}\right) = \left(\frac{1}{63}\right) = 1$$

$\left(\frac{1201}{1453}\right) = 1$ dir. $x = 419$ için $1201 \equiv 419^2 \bmod 1453$ denkliği sağlanır. $a = 1201$ sayısı $\bmod 1453$'de kuadratik rezidüdür.

4. RSA ALGORİTMASI

Bu bölümde büyük asal sayıların kullanıldığı en önemli şifreleme sistemlerinden biri olan RSA algoritması incelenerek, anahtar üretimi, şifreleme ve şifre çözme adımlarına yer verilmiştir. Ayrıca, RSA algoritmasının önemi ve güvenliği üzerinde durulmuş ve örneklerle desteklenmiştir. Bu bölümde algoritmalar verilirken, (Paar & Pelz, 2009) ve (Rivest, Shamir, & Adleman, 1978) çalışmalarından yararlanılmıştır.

RSA 1977 yılında Ron Rivest, Adi Shamir ve Leonard Adleman tarafından geliştirilmiş olan açık anahtarlı bir şifreleme sistemidir (Rivest, Shamir, & Adleman, 1978). RSA algoritmasının güvenliği çarpanlara ayırma probleminin zorluğuna dayanmaktadır. Daha açık söylemek gerekirse, yeteri kadar büyük iki asal sayının çarpımından oluşan tam sayıyı çarpanlara ayırmanın zorluğuna dayanmaktadır. Algoritmanın güvenirliliği kullanılan asal sayıların büyüklüğü ile doğru orantılıdır; fakat şifreleme ve şifre çözme işlemlerinde yavaş olması dezavantaj oluşturmaktadır. Bunun yanı sıra RSA şifreleme ve şifre çözme işlemlerinde zaman açışından dezavantaj oluşturmasının asıl sebebi şifreleme işleminin üstel işlem olmasından kaynaklanmaktadır. Bir mesajı şifreleme işlemi yaparken mesajın e-inci kuvveti alınırken, şifreli mesaj çözülürken d-inci kuvveti alınmaktadır. Modüler üs alma işlemi de zaman gerektiren bir hesaplama olduğu için algoritmanın yavaş çalışmasına neden olmaktadır. Öncelikle RSA şifreleme sisteminde kullanılacak olan parametreleri verelim.

Tablo 4.1. RSA Parametreleri

n	Modülüs parametresi (Açık)
e	Üs parametresi (Açık Anahtar)
d	Üs parametresi (Gizli Anahtar)
p, q	İki farklı asal sayılar (Gizli)
$\varphi(n)$	Euler Totient Fonksiyonu (Gizli)

4.1. RSA Anahtar Üretimi

RSA algoritmasını kullanmak için öncelikle anahtar üretimi yapılması gerekmektedir. Temel olarak, bir açık anahtar ve buna karşılık gelen bir gizli anahtar üretimi yapılmaktadır. Anahtar üretmek için aşağıdaki adımlar izlenmektedir.

RSA'da Anahtar Üretimi Adımları

1. p ve q iki farklı büyük asal sayıları seçilir.
2. $n = p \cdot q$ değeri hesaplanır.
3. $\varphi(n) = (p-1) \cdot (q-1)$ değeri hesaplanır.
4. $1 < e < \varphi(n)$ ve obeb $\left(e, \varphi(n)\right) = 1$ olacak şekilde rastgele bir e sayısı seçilir.
5. $e \cdot d \equiv 1 \bmod \left(\varphi(n)\right)$ denkliğini sağlayan d sayısı hesaplanır.

Burada (n, e) açık parametreler ve (p, q, d) gizli parametrelerdir. Açık anahtar e değeri obeb $(e,\ \varphi(n)) = 1$ olarak seçildiği için $e \in \mathbb{Z}_n^*$ dir ve $d = e^{-1}$ gizli anahtar olarak seçilmektedir. Burada d gizli anahtarı ters Öklid algoritması ile hesaplanmaktadır.

Şifreleme ve şifre çözme anahtarları oluşturulurken $n = p \cdot q$ çarpımının içindeki p ve q asalları yeterince büyük seçilmelidir. Diğer bir ifade ile n bilinse bile p ve q asalları hesaplanamayacak kadar büyük olmalıdır.

4.2. RSA Şifreleme Algoritması

RSA şifreleme ve şifre çözme işlemleri, $\mathbb{Z}_n$ halkası üzerinde yapılır ve modüler hesaplama merkezi bir rol oynar. RSA, m mesajını şifreler, burada m mesajının sayısal karşılığı $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$ de bir eleman olarak kabul edilmektedir.

RSA algoritmasında şifreleme fonksiyonu E ile şifre çözme fonksiyonu D ile gösterilir. Düz metin m ve şifreli metin c ile gösterilir. Burada c ve m değerleri $\mathbb{Z}_n$ halkasının elemanları olarak görülebilir. m düz metin ve e açık anahtarı için RSA şifreleme fonksiyonu,

$$c = E(m) \equiv m^e \bmod n$$

ve d gizli anahtarı için RSA şifre çözme fonksiyonu,

$$m = D(c) \equiv c^d \bmod n$$

şeklinde hesaplanmaktadır.

RSA Şifreleme (Encryption): A kişisi, m mesajını şifreleyerek B kişisine göndermek istesin. RSA şifreleme işlemi için A kişisi aşağıdaki adımları izlemektedir.

RSA Şifreleme Adımları

- İlk olarak B kişisinin açık anahtarı olan (n, e) ikilisini alır.
- $c = E(m) \equiv m^e \bmod n$ değerini hesaplar.
- c şifreli mesajını B kişisine gönderir.

RSA Şifre Çözme (Decryption): B kişisi, A kişisinden gelen şifreli c mesajını çözmek için aşağıdaki işlemi gerçekleştirmektedir.

RSA Şifre Çözme İşlemi

A kişisinden gelen şifreli c mesajını çözmek isteyen B kendi d gizli anahtarını kullanarak;

- $m = D(c) \equiv c^d \bmod n$ değerini hesaplar ve m mesajına ulaşır.

RSA algoritmasının çalışma prensibini bir örnekle açıklayalım. Çalışmanın devamında şifreleme anahtarı k_E ile şifre çözme anahtarı k_D ile gösterilecektir.

Örnek 4.2.1. Alice tarafından Bob'a "F" harfi şifrelenip gönderilsin. Şifrelenecek olan F harfinin sayısal dönüşümü için ASCII tablosu kullanılacaktır.

Tablo 4.2.2.1. Harflerin ASCII Kod Karşılıkları

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

Alice F harfinin ASCII karşılığı olan $m = 70$ mesajını şifreleyip göndersin ve Bob gelen şifreli mesajı çözerek $m = 70$ mesajına ulaşsın. Şifreleme ve şifre çözme adımlarını verelim. Anahtar üretimi için Bob aşağıdaki adımları izler.

1. $p = 347$ ve $q = 491$ şeklinde iki farklı asal sayı seçer.

2. $n = p \cdot q = 170377$ değerini hesaplar.

3. $\varphi(n) = (347 - 1) \cdot (491 - 1) = 169540$ değerini hesaplar.

4. $1 < e < \varphi(n)$ ve obeb $(e, \varphi(n)) = 1$ olacak şekilde rastgele bir $e = 17$ seçer.

5. $d \equiv e^{-1} = 9973 \pmod{169540}$ değerini hesaplar.

Burada, $e = 17$ açık anahtar ve $d = 9973$ gizli anahtardır. Bob şifreleme de kullanılacak (e, n) ikilisini Alice ile paylaşır.

RSA Şifre Algoritması	**İşlem**
$k_E = (e, n)$ $c = E_{k_E}(m) = m^e \bmod n$	$k_E = (17,\ 170377)$ $c = E_{k_E}(70) = 70^{17} \bmod 170377$

Alice $c = 70^{17} \bmod 170377 = 11855$ şifreli mesajını elde eder ve Bob'a gönderir. Bob Alice tarafında gelen şifreli c mesajını kendi gizli d anahtarı ile çözerek m mesajına ulaşır.

RSA Deşifreleme	**İşlem**
$k_D = (d, n)$ $m = D_{k_D}(c) = c^d \bmod n$	$k_D = (9973, 170377)$ $m = D_{k_D}(\mathrm{c}) = 11855^{9973} \bmod 170377$

Bob, açık mesaj $m = c^d = 11855^{\ 9973} \bmod 170377 \equiv 70$ elde eder.

Örnek 4.2.2. Alice tarafından Bob'a "PRESTİJ" düz metni şifreli olarak gönderilsin. RSA algoritmasıyla şifreleme ve şifre çözme işlemlerini yapalım. Bu örnekte işlem kolaylığı için p ve q asal sayılarını küçük seçiyoruz.

Anahtar üretimi için Bob aşağıdaki adımları izler.

1. $p = 41$ ve $q = 29$ şeklinde farklı iki asal sayı seçer.

2. $n = p \cdot q = 41 \cdot 29 = 1189$ değerini hesaplar.

3. $\varphi(n) = (p-1)\cdot(q-1) = 40 \cdot 28 = 112$ değerini hesaplar.

4. $1 < e < \varphi(n)$ ve obeb $\left(e, \varphi(n)\right) = 1$ olacak şekilde rastgele bir $e = 11$ sayısı seçer.

5. $e \cdot d \equiv 1 \bmod \varphi(n)$ denkliğini sağlayan d = 611 sayısı hesaplanır.

Açık anahtar: $e = 11$
Gizli anahtar: $d = 611$

RSA Şifre Algoritması	**İşlem**
$k_E = (e, n)$ $c = E_{k_E}(m) = m^e \bmod n$	$k_E = (11, 1189)$ $c = E_{k_E}(m) = m^{11} \bmod 1189$

Tablo 2.1.1. de verilen Türkçe harflerin sıra sayı karşılıklarından faydalanılarak aşağıdaki tablo oluşturulur. Harflerin sıra karşılığı yazılırken tek basamaklı sayı başına 0 eklenerek iki basamağa tamamlanır.

Şifrelenecek Metin	**P**	**R**	**E**	**S**	**T**	**İ**	**J**
Sayısal Karşılığı	**19**	**20**	**05**	**21**	**23**	**11**	**12**

Bir harfi şifrelemek için,

$$E_{k_E}(m) = m^{11} \bmod 1189$$

şifreleme işlemi yapılır. Harf değerleri için şifreleme işlemi yapalım.

$E_{k_E}(19) = 19^{11} \bmod 1189 \equiv 1100$
$E_{k_E}(20) = 20^{11} \bmod 1189 \equiv 964$
$E_{k_E}(05) = 05^{11} \bmod 1189 \equiv 651$
$E_{k_E}(21) = 21^{11} \bmod 1189 \equiv 635$
$E_{k_E}(23) = 23^{11} \bmod 1189 \equiv 310$
$E_{k_E}(11) = 11^{11} \bmod 1189 \equiv 1083$
$E_{k_E}(12) = 12^{11} \bmod 1189 \equiv 887$
değerleri elde edilir.

Bulunan $E_{k_E}(m)$ değerlerinde 4 basamaktan az olan sayı değerlerinin başına 0 eklenerek 4 basamağa tamamlanır.

Şifrelenen Metin	**P**	**R**	**E**	**S**	**T**	**İ**	**J**
Şifreli metin $c = E_{k_E}(m)$	**1100**	**0964**	**0651**	**0635**	**0310**	**1083**	**0887**

Alice şifreli metin olarak **1100 0964 0651 0635 0310 1083 0887** sayılarını Bob'a gönderir. Bob şifreli halde verilen bu mesajı çözerken aşağıdaki adımları izler. RSA deşifreleme işlemi aşağıda verildiği gibi yapılmaktadır.

RSA Deşifreleme	**İşlem**
$k_D = (d, n)$ $m = D_{k_D}(c) = c^d \bmod n$	$k_D = (611, 1189)$ $m = D_{k_D}(c) = c^{11} \bmod 1189$

Şifreli metni çözmek için, $D_{k_D}(c) = c^{611} \bmod 1189$ şifre çözme işlemi kullanılır.

$D_{k_D}(c) = 1100^{611} \bmod 1189 \equiv 19$
$D_{k_D}(c) = 0964^{611} \bmod 1189 \equiv 20$
$D_{k_D}(c) = 0651^{611} \bmod 1189 \equiv 05$
$D_{k_D}(c) = 0635^{611} \bmod 1189 \equiv 21$
$D_{k_D}(c) = 0310^{611} \bmod 1189 \equiv 23$
$D_{k_D}(c) = 1083^{611} \bmod 1189 \equiv 11$
$D_{k_D}(c) = 0887^{611} \bmod 1189 \equiv 12$

Şifreli Metin	**1100**	**0964**	**0651**	**0635**	**0310**	**1083**	**0887**
$m = D_{k_D}(c)$	**19**	**20**	**05**	**21**	**23**	**11**	**12**
Metin Karşılığı	**P**	**R**	**E**	**S**	**T**	**İ**	**J**

Bob Alice tarafından gelen şifreli metnini çözerek "PRESTİJ" metnine ulaşır.

RSA algoritması şifrelemenin yanı sıra dijital imzalar için de kullanılabilmektedir. Dijital imzalar, dijital mesajların veya belgelerin gerçekliğini doğrulamak için verilen bir şemadır. Geçerli bir dijital imza, alıcıya, mesajın bilinen bir gönderici tarafından oluşturulduğuna (kimlik doğrulama) ve böylece onu göndermeyi reddedemeyeceğine (reddetmeme) ve mesajın aktarım sırasında değiştirilmediğine (bütünlük) inanması için verilmektedir.

4.3. RSA Algoritması'nın Hızı

RSA algoritmasında güvenilirliğini artırmak için seçilen sayılar çok büyük asal sayılardır. Şifreleme de ve deşifreleme de kullanılan bu asal sayılar güvenirliği artırırken aynı zamanda işlem süresini de artırmaktadır. Bunun yanında asıl yavaşlamanın sebebi şifreleme ve şifre çözme işlemlerinde üs alma işlemidir. İşlem süresini azaltmak ve daha hızlı üs alma işlemi yapabilmek için şifreleme ve şifre çözme hızını artırmaya yönelik çeşitli algoritmalar geliştirilmiştir.

4.3.1. RSA Şifreleme Hızını Arttıran Algoritmalar

RSA şifreleme işlemi için modüler üs alma işlemi yapıldığından şifreleme süresi uzamaktadır. Bu bölümde RSA şifreleme algoritmasının hızını arttıran algoritmalardan bazılarını vereceğiz. Bu algoritmalar sayesinde modüler üs alma işlemi daha hızlı yapılabilmekte ve RSA şifreleme hızı arttırılabilmektedir.

Algoritmalardan ilki Montgomery Modüler Çarpım Algoritmasıdır. 1985 yılında Peter Montgomery tarafından tasarlanan RSA'nın şifreleme hızını arttıran bir algoritmadır

(Montgomery, 1985). İkincisi, ikili Modüler Üs Alma Algoritması şifreleme algoritmasında şifreleme hızını artırmak için tasarlanmış bir algoritmadır (Koç, 1994). İkili Arama Algoritması, RSA'da şifrelenecek mesaj olan m' yi iki ayrı bit halinde bularak, şifrelemeyi hızlandırmak için kullanılan bir algoritmadır. Hızlı Mod Alma Algoritması RSA şifreleme algoritmasında şifreleme işleminin hızını artırmak için tasarlanmıştır.

Küçük Açık Anahtar İle Hızlı Şifreleme (Fast Encryption with Short Public Exponents)

Açık anahtar e ile RSA şifreleme işlemi yapılırken basit ve çok güçlü bir teknik kullanılabilir. Burada açık anahtar e çok küçük bir değer olarak seçilebilir. Tablo 5.4.1. de verilen $e = 3$, $e = 17$ ve $e = 2^{16} + 1$ değerleri özel olarak seçilen ve yaygın olarak kullanılan açık anahtarlardır. Tabloda verilen #SQ kare alma sayısını ve #MUL çarpma sayısını gösterir.

Tablo 4.3.1. Küçük Açık Anahtar ile RSA şifreleme işleminin karmaşıklığı

Açık Anahtar e	**e İkili Taban Gösterimi**	**#MUL+ #SQ**
3	$(11)_2$	3
17	$(10001)_2$	5
$2^{16}+1$	$(10000000000000001)_2$	17

Burada asıl anlatılmak istenen, şifreleme de e nin ikili taban gösteriminde ne kadar çok "0" varsa o kadar çarpma işlemi yok demektir ve bu sebepten dolayı şifreleme hızlıdır (Paar & Pelz, 2009). Bu yüzden açık anahtar e yi seçerken ikili taban gösterimindeki 0 sayılarının çok olmasına dikkat edilir ki şifreleme hızı artsın.

Tekrarlanan Kare Alma Algoritması (Repeated Squaring Algorithm). Tekrarlanan kare alma algoritması, bir tam sayı kuvvetini hızlı bir şekilde hesaplayan algoritmadır.

Algoritma 4: Tekrarlanan Kare Alma Algoritması $g^m \bmod n$

Girdi: $n, g, m = (m_{l-1}m_{l-2} \dots m_0)_2$
Çıktı: $g^m \bmod n$
$s \leftarrow g$
if $m_0 = 0$ **then**
    $r \leftarrow 1$
end if
if $m_0 = 1$ **then**
    $r \leftarrow g$
end if
for $i = 1\ to\ \ l - 1\ \ do$
    $s \leftarrow s^2 \bmod n$
    if $m_i = 1$ **then**
        $r \leftarrow rs \bmod n$
    end if
end for
return r

Algoritma 4 de verilen tekrarlanan kare alma algoritmasının zaman karmaşıklığı $O(log^3 n)$ işlemdir (Akyıldız, Cenk, & Sınak, 2021).

Örnek 4.3.1.1. $x \equiv 13^{73} \bmod 147$ değerini hızlı mod alma algoritması ile hesaplayalım.

$$13^1 \equiv 13 \bmod 147$$
$$13^2 \equiv 22 \bmod 147$$
$$13^4 \equiv (13^2)^2 \equiv 22^2 \equiv 43 \bmod 147$$
$$13^8 \equiv (13^4)^2 \equiv 43^2 \equiv 85 \bmod 147$$
$$13^{16} \equiv (13^8)^2 \equiv 85^2 \equiv 22 \bmod 147$$
$$13^{32} \equiv (13^{16})^2 \equiv 22^2 \equiv 43 \bmod 147$$
$$13^{64} \equiv (13^{32})^2 \equiv 43^2 \equiv 85 \bmod 147$$
$$13^{73} \equiv 13^{64} \cdot 13^8 \cdot 13^1 \equiv 85 \cdot 85 \cdot 13 \equiv 139 \bmod 147$$

olur. Diğer bir ifade ile $13^{73} \equiv x \bmod 147$ denkliğinde $x = 139$ olur.

Tekrarlanan kare alma algoritması şifreleme hızını arttırmak için oldukça etkilidir. Bu nedenle RSA şifreleme ve Miller-Rabin asallık testi gibi bazı kriptografik algoritmalarda kullanılmaktadır.

4.3.2. RSA Şifre Çözme Hızını Arttıran Algoritmalar

RSA şifre çözme işlemi için modüler üs alma işlemi yapıldığından şifre çözme işleminin süresi uzamaktadır. Bu bölümde RSA şifre çözme hızını arttıran algoritmalardan olan Çin kalan teoremini vereceğiz. Çin kalan teoremi algoritması sayesinde RSA şifre çözme işlemi daha hızlı yapılabilmektedir.

Çin Kalan Teoremi (The Chinese Remainder Theorem)

Sun Tzu tarafından M.S. 3. Yüzyılda bulunan Çin kalan teoremi (ÇKT) matematik ve kriptografi için önemli bir teoremdir. RSA algoritması ve Eliptik eğri tabanlı sistemler gibi kriptografik hesaplamalarda kullanılan çarpma algoritmalarının verimliliğini arttırmaktadır.

Teorem 4.3.2.1. $n_1, n_2, \dots, n_k$ ikişer ikişer aralarında asal sayılar ve $r_1, r_2, \dots, r_k$ tam sayılar olmak üzere,

$$x \equiv r_1 \bmod n_1$$
$$x \equiv r_2 \bmod n_2$$
$$\vdots$$
$$x \equiv r_k \bmod n_k$$

denklik sisteminin $\bmod n_1 \cdot n_2 \cdots n_k$ da tek çözümü vardır.

İspat. $n = n_1 \cdot n_2 \cdots n_k$ olsun.
$1 \le i \le k$ için $\left(\frac{n}{n_i},\ n_i\right) = 1$ olduğundan $\frac{n}{n_i} s_i \equiv r_i \bmod n_i$ denklik sistemini sağlayan s_i vardır.
$x \equiv \sum_{i=1}^{k} \frac{n}{n_i}\, s_i$ olsun. $i \neq j$ için $n_i \mid \frac{n}{n_j}$ olduğundan $x \equiv \frac{n}{n_i}\, s_i \equiv r_i \bmod n_i$ olur.

Diğer bir ifade ile x tüm denklikleri sağlar. Tümevarım yöntemi ile,

- $x \equiv r_1 \bmod n_1$ denkliğinin tek bir çözümü vardır.
- İlk olarak $k-1$ denklik sisteminin tek çözümü olduğunu kabul edelim.

- k denklik sisteminin tek çözümü olduğunu göstermek istiyoruz. x_1 ve x_2 iki çözüm olsun. Bu durumda,

$$x_1 \equiv x_2 \bmod n_1 n_2 \dots n_{k-1}$$
$$x_1 \equiv x_2 \bmod n_k$$

olduğundan $n_1 n_2 \dots n_{k-1} \mid (x_1 - x_2)$ ve $n_k < (x_1 - x_2)$ olur. Diğer bir ifadeyle, $x_1 \equiv x_2 \bmod n_1 n_2 \dots n_k$ olur.

Çin Kalan Teoremi algoritması Algoritma 5'de verilmiştir.

Algoritma 5: Çin Kalan Teoremi Algoritması

Girdi: $x \equiv a_i \bmod n_i,\ i = 1, 2, \dots, k$ öyle ki obeb $(n_i, n_j) = 1$ her $i \neq j$

Çıktı: $x_0 = \sum_{i=1}^{k} a_i M_i N_i \bmod N$

1: $N \leftarrow 1$
2: **for** $i = 1$ **to** k **do**
3: $\quad N \leftarrow N n_i$
4: **end for**
5: $x_0 \leftarrow 0$
6: **for** $i = 1$ **to** k **do**
7: $\quad N_i \leftarrow N / n_i$
8: $\quad M_i \leftarrow N_i^{-1} \bmod n_i$
9: $\quad x_0 \leftarrow x_0 + a_i M_i N_i \bmod N$
10: **end for**
11: **return** x_0

Çin kalan teoremi algoritması (Algoritma 5) $n_i \leq m$ ve $\log n < \log m = l$ olmak üzere $O(l^5)$ işlem karmaşıklığına sahiptir (Akyıldız, Cenk, & Sınak, 2021).

Çin Kalan Teoremi İle Hızlı Şifre Çözme. ÇKT ile şifre çözme işlemi hızlandırılabilir. Temel amacı uzun modül olan n ile hesaplama yapmak yerine iki kısa asal olan p ve q modülüne göre iki ayrı üs alma işlemi yapmaktır.

ÇKT Dönüşümü: x modülünü n modülünün iki faktörü olan p ve q'ya indirgenir. Elde edilen ifade x'in modüler gösterimi olarak adlandırılır.

$$c_p \equiv x \bmod p$$
$$c_q \equiv x \bmod q$$

ÇKT Hesaplama: x_p ve x_q kullanılarak aşağıdaki iki üs hesaplanmaktadır.

$$m_p = c_p^{d_p} \bmod p$$
$$m_q = c_q^{d_q} \bmod q$$

iki yeni üs aşağıdaki şekilde verilmektedir.

$$d_p \equiv d \bmod (p - 1)$$
$$d_q \equiv d \bmod (q - 1)$$

Ters Dönüşüm:

$$y \equiv [qM_i]m_p + [pN_i]m_q \bmod n$$

burada M_i ve N_i katsayıları şu şekilde hesaplanır:

$$M_i \equiv q^{-1} \bmod p$$

$$N_i \equiv p^{-1} \bmod q$$

ÇKT Yöntemi bankacılık uygulamalarında akıllı kartlar üzerindeki uygulamalar için önemlidir. Burada d gizli anahtarı içeren dijital imzalamaya ihtiyaç vardır. ÇKT yi imza için uygulayarak akıllı kart özelliği 4 kat daha hızlandırılabilir. Normalde 1024 bit olan RSA üssü kullanıldığında 3 saniye sürerse, ÇKT kullanılarak 0,75 saniyeye düşürülür. Bu ÇKT algoritmasının gerçek hayatta doğrudan kullanımının bir örneğidir (Paar & Pelz, 2009).

Örnek 4.3.2.1. Örnek 4.2.2.1. de elde edilen şifreli mesaj $c = 11855$ için RSA şifre çözme işlemini ÇKT kullanarak yapalım.

Örnek 4.2.2.1. de verilen $p = 347$ ve $q = 491$ asalları için $n = p \cdot q = 170377$ 'dir. Açık anahtar $e = 17$ ve gizli anahtar $d \equiv 9973$ dir.
c şifreli mesaj ÇKT dönüşümleri aşağıdaki gibidir. Şifreli mesajın dönüşümü;

$$c_p \equiv 57 \equiv 11855 \bmod 347$$
$$c_q \equiv 71 \equiv 11855 \bmod 491$$

şeklinde yapılır. $d = 9973$ gizli anahtarının dönüşümü

$$d_p \equiv 285 \equiv 9973 \bmod 346$$
$$d_q \equiv 173 \equiv 9973 \bmod 490$$

şeklinde yapılır. ÇKT dönüşümleri sonucunda şifre çözümü

$$m_p \equiv c_p^{d_p} = 57^{285} \equiv 70 \bmod 347$$
$$m_q \equiv c_q^{d_q} = 71^{173} \equiv 70 \bmod 491$$

şeklindedir. ÇKT için M_i ve N_i katsayıları

$$M_i = 491^{-1} \equiv 147 \bmod 347$$
$$N_i = 347^{-1} \equiv 283 \bmod 491$$

şeklindedir. ÇKT algoritmasına göre x düz metni aşağıdaki gibi hesaplanır.

$$m \equiv [q \cdot M_i]m_p + [p \cdot N_i]m_q \bmod n$$
$$m \equiv [491 \cdot 147]70 + [347 \cdot 283]70 \bmod 170377$$
$$m \equiv \mathbf{70} \bmod 170377$$

n'nin $t + 1$ bit olduğunu varsayılırsa, hem p hem de q yaklaşık $t/2$ bit uzunluğundadır. ÇKT üslerinde bulunan x_p, x_q, d_p ve d_q değerleri yaklaşık $t/2$ bit uzunluğa sahiptir. İki üs için kare ve çarpma algoritmasını kullanırsak, her biri ortalama olarak yaklaşık $\frac{3t}{4}$ modüler çarpma ve kare gerektirir. Sonuç olarak toplam işlem sayısı:

$$\#SQ + \#MUL = 2 \cdot \frac{3t}{4} = \frac{3t}{2}$$

Her çarpma ve kare alma, yalnızca $t/2$ bitlik bir uzunluğa sahip sayıları içerir. Çarpmanın karmaşıklığı bit uzunluğu ile kuadratik olarak azaldığından, her $t/2$ bitlik çarpma, bir t bitlik çarpmadan dört kat daha hızlıdır. Bu nedenle, ÇKT yoluyla elde edilen toplam hızlanma 4 kattır.

Bu hızlandırma yöntemi aynı zamanda akıllı kartlardaki uygulamalar için, örneğin yalnızca küçük bir mikroişlemciyle donatılmış bankacılık uygulamaları için oldukça önemlidir. Burada, genellikle gizli anahtar d'yi içeren dijital imzaya ihtiyaç duyulur. İmza hesaplaması için ÇKT'yi uygulayarak, akıllı kart dört kat daha hızlandırmaktadır. Diğer bir ifade ile normal 1024 bitlik bir RSA üs alma işlemi 3 saniye sürerse, ÇKT'yi kullanmak bu süreyi 0,75 saniyeye düşürür. Bu örnek gerçek hayattaki kullanımına en iyi örnek olarak verilebilir.

4.4. Çarpanlara Ayırma Metotları

Asal sayılar ve pozitif tam sayıların asal çarpanlara ayrılması, geçmişten günümüze insanların dikkatini çeken ve çalışmalar yapılan bir konu olmuştur. Bu sayıların kriptolojide kullanılmasıyla ilgili olarak önem kazanmış ve çalışmalar bu yönde artmıştır.

Şifreleme de kullanılacak olan sayıların çarpanlara ayırma metotlarına dayanıklı olması şifrelemenin güvenliği için gerekli bir kriterdir. Büyük sayılar için çarpanlara ayırma probleminin zorluğu RSA şifreleme sisteminin güvenliğini oluşturmaktadır.

Bu bölümde çarpanlara ayırma metotlarından Fermat çarpanlara ayırma algoritmasını ve Polard'ın Rho Heuristik algoritmasını vereceğiz.

4.4.1. Fermat Çarpanlara Ayırma Algoritması

Fermat çarpanlara ayırma algoritması iki kare farkı elde etmeye dayanan çarpanlara ayırma metotlarından biridir. Algoritmanın açıklamasını verelim.

n bir tek sayı olsun. O halde $n = p \cdot q$ ve $p \leq q \Rightarrow x^2 - y^2 = n$ denkleminin, $x + y = q$ ve $x - y = p$ eşitliklerini sağlayan $x = \frac{p+q}{2}$ ve $y = \frac{q-p}{2}$ gibi bir çözümü vardır. Dolayısıyla n sayısını $n = p \cdot q$ şeklinde çarpanlara ayırmak için $x^2 - y^2 = n$ denkleminin $x - y > 1$ olacak şekilde bir çözümünü bulmamız yeterlidir. $x^2 > n$ olacağı için $x > \sqrt{n}$ olacaktır. Fermat Çarpanlara Ayırma algoritması Algoritma 18'de verilmektedir.

Algoritma 18: Fermat Çarpanlara Ayırma Algoritması

Girdi: n bileşik tam sayı.
Çıktı: p ve q öyleki $n = p \cdot q$

for $x = \lfloor \sqrt{n} \rfloor + i, \quad i = 0, 1, 2, \ldots do$
  if $x^2 - n$ tam kare,
    $y \leftarrow \sqrt{x^2 - n}$
    return $p \leftarrow (x + y)$ ve $q \leftarrow (x - y)$
  end if
end for

Fermat çarpanlara ayırma algoritması için bir örnek verelim.

Örnek 4.4.2.1. $n = 1139$ sayısını Fermat çarpanlara ayırma algoritması ile çarpanlarına ayıralım. $n = 1139$ sayısının karekökünden başlayarak sayıya kadar olan bütün ihtimallerin farkı ile iki kare farkı elde edilir mi araştırılır.

Girdi: $n = 1139$
$\sqrt{1139} \cong 33{,}75$ ise $x > 33{,}75$ olacak şekilde x değerleri için algoritmayı çalıştıralım.
$x = 34$ için $y = \sqrt{34^2 - 1139} = \sqrt{17}$
$x = 35$ için $y = \sqrt{35^2 - 1139} = \sqrt{86}$
$x = 36$ için $y = \sqrt{36^2 - 1139} = \sqrt{157}$
$x = 37$ için $y = \sqrt{37^2 - 1139} = \sqrt{230}$
$x = 38$ için $y = \sqrt{38^2 - 1139} = \sqrt{305}$
$x = 39$ için $y = \sqrt{39^2 - 1139} = \sqrt{382}$
$x = 40$ için $y = \sqrt{40^2 - 1139} = \sqrt{461}$
$x = 41$ için $y = \sqrt{41^2 - 1139} = \sqrt{542}$
$x = 42$ için $y = \sqrt{42^2 - 1139} = \sqrt{625} = \sqrt{25^2} \Rightarrow y = 25$

$x = 42$ ve $y = 25$ için, $1139 = 42^2 - 25^2$ şeklinde yazılabilmektedir. İki kare farkı özelliğinden $1139 = (42 - 25) \cdot (42 + 25)$ olarak yazılabilmektedir. Bunun sonucunda, 1139 sayısının $p = 67$ ve $q = 17$ olarak çarpanları bulunmuş olur.

4.4.2. Polard'ın Rho Heuristik Algoritması

Pollard'ın Rho çarpanlara ayırma metodu, büyük asal sayıların hızlı bir şekilde çarpanlara ayrılmasını amaçlamaktadır. Veri güvenliği açısından oldukça önemli olan bu yöntemin çalışması aşağıdaki adımlardan oluşmaktadır.

- Çarpanlarına ayrılmak istenen sayı n olsun.
- Bulunacak çarpanlardan biri d olarak isimlendirilecektir ve d sayısı $d \mid n$ şartını sağlamalıdır.
- Algoritma sırasında kullanılacak iki değişken olan a ve b değerlerine 2 değeri atanarak başlanır. $a \leftarrow 2,\ b \leftarrow 2$
- Bir döngü içerisinde, sonucu bulana kadar aşağıdaki adımlar takip edilir:

$$a \leftarrow a^2 + 1 \bmod n$$
$$b \leftarrow b^2 + 1 \bmod n$$
$$b \leftarrow b^2 + 1 \bmod n$$

- a sayısının 1 kez karesi alınıp artırılıyorken, b sayısı iki kez aynı fonksiyona tabi tutulmaktadır. Dolayısıyla b sayısı, a ya göre daha hızlı ilerlemektedir.
- $d = \text{obeb}\ (a - b,\ n)$ değeri hesaplanır.
- eğer $1 < d < n$ şartını sağlayan bir d değeri bulunuyorsa algoritma başarıyla tamamlanmıştır, işlem sonlandırılır.
- eğer $d = n$ durumu oluşursa algoritma başarısız bir şekilde tamamlanmıştır.
- Yukarıdaki iki durum haricinde döngüye devam edilir (Alizade, 2014).

Algoritma 6: Pollard'ın Rho Algoritması

Girdi: n bileşik tam sayı.
Çıktı: n'nin faktörü.
$f: \mathbb{Z}_n \to \mathbb{Z}_n$
$a \leftarrow 2, \quad b \leftarrow 2, \quad d \leftarrow 1$
while $d = 1$ *do*
$a \leftarrow f(a) \bmod n$
$b \leftarrow f\left(f\ (b)\right) \bmod n$
$d \leftarrow \text{obeb}\ (a - b, n)$
end while
if $d = n$ **then return** "başarısız"
else return d
end if

Örnek 4.4.1.1. 2021 sayısını Pollard'ın Rho Algoritması ile çarpanlarına ayıralım.

Girdi: $n \leftarrow 2021,\ a \leftarrow 2,\ b \leftarrow 2$
$a \leftarrow 2^2 + 1 \bmod 2021$ ve $b \leftarrow 2^2 + 1 \bmod 2021$, $b \leftarrow 5^2 + 1 \bmod 2021$
$a = 5, b = 26$ için, $d = \text{obeb}\ (a - b, n)$, $d = \text{obeb}\ (5 - 26, 2021) = 1$

$a \leftarrow 5^2 + 1 \bmod 2021$ ve $b \leftarrow 26^2 + 1 \bmod 2021$, $b \leftarrow 677^2 + 1 \bmod 2021$
$a = 26, b = 1584$ için, $d = \text{obeb}\ (26 - 1584, 2021) = 1$

$a \leftarrow 26^2 + 1 \bmod 2021$ ve $b \leftarrow 1584^2 + 1 \bmod 2021$, $b \leftarrow 996^2 + 1 \bmod 2021$
$a = 677, b = 1727$ için, $d = \text{obeb}\ (677 - 1727, 2021) = 1$

$a \leftarrow 677^2 + 1 \bmod 2021$ ve $b \leftarrow 1727^2 + 1 \bmod 2021$, $b \leftarrow 1555^2 + 1 \bmod 2021$
$a = 1584, b = 910$ için, $d = \text{obeb}\ (1584 - 910, 2021) = 1$

$a \leftarrow 1584^2 + 1 \bmod 2021$ ve $b \leftarrow 910^2 + 1 \bmod 2021$, $b \leftarrow 1512^2 + 1 \bmod 2021$
$a = 996, b = 394$ için, $d = \text{obeb}\ (996 - 394, 2021) = 1$

$a \leftarrow 996^2 + 1 \bmod 2021$ ve $b \leftarrow 394^2 + 1 \bmod 2021$, $b \leftarrow 1641^2 + 1 \bmod 2021$
$a = 1727, b = 910$ için, $d = \text{obeb}\ (1727 - 910, 2021) = 1$

$a \leftarrow 1727^2 + 1 \bmod 2021$ ve $b \leftarrow 910^2 + 1 \bmod 2021$, $b \leftarrow 1512^2 + 1 \bmod 2021$
$a = 1555, b = 394$ için, $d = \text{obeb}\ (1555 - 394, 2021) = 43$

$1 < d < n$ şartını sağlayan bir $d = 43$ değeri bulundu. Bu sonuç 2021 sayısının çarpanlarından biridir. Diğer çarpan $2021/43 = 47$ olarak bulunabilir.

4.5. RSA Algoritması'nın Güvenliği

Açık anahtarlı şifreleme algoritmalarından biri olan RSA da iki farklı anahtar kullanılmaktadır. Açık anahtar ile şifrelenen düz metin sadece gizli anahtar ile deşifrelenmektedir. RSA algoritmasının güvenirliği çok büyük asal sayı seçmeye bağlıdır. Sistemin güvenliğini sağlamak için çarpanlarına ayırma algoritmalarına dayanıklı n sayısı oluşturmak önemlidir. Bunun için de p ve q asalları bir takım kriterlere uygun seçilmelidir. Seçilen asal sayılar ancak n sayısının büyüklüğü ile orantılı bir

güvenlik seviyesi sunmaktadır (Akyıldız, Çalık, Özarar, Tok, & Yayla, 2013). p ve q asal sayıları seçilirken dikkat edilmesi gereken kriterler aşağıda verilmiştir.

1) p ve q asallarının bit sayıları yaklaşık olarak eşit olmalıdır. Örneğin n sayısı 1024 bitlik bir değere sahip ise p ve q yaklaşık olarak 512 bit olmalıdır.
2) $p - q$ farkı çok küçük olmaması gerekmektedir.
3) p ve q güçlü asal sayı olmalıdır. Güçlü asal sayı olması için aşağıdaki koşullar sağlanmalıdır.
    - $p - 1$, r ile gösterilen asal faktöre eşittir.
    - $p + 1$ in büyük bir asal çarpanı vardır.
    - $r - 1$ büyük bir asal çarpana sahiptir.
4) Güvenliği k-bit olan bir sistemi kırmak için 2^k mertebesinde işlem yapmak gerekmektedir. Bu yüzden p ve q asalları,

$$2^{(k-1)/2} \leq p, q \leq (2^{k/2} - 1)$$

aralığında bir değer seçilmelidir.

RSA sisteminin güvenlik seviyesi, sistemi kırmak için gereken hesaplama gücünün büyüklüğüne bağlıdır. RSA da güvenlik n parametresine bağlı olarak yapılması gereken işlem miktarının hesaplanmasıyla elde edilmektedir (Akyıldız, Çalık, Özarar, Tok, & Yayla, 2013). RSA algoritmasını kırmak demek şifreleme de ve şifre çözme de kullandığımız $p,\ q$ asalları, e, c, d ve $\varphi\ (n)$ değerleri için, m düz metnini faktör etmek demektir. m'nin faktör edilme adımları için (Koç, 1994) çalışmasından faydalanılabilir.

RSA Standartları

Günümüzde RSA algoritmasının, SSL, S-HTTP, S-MIME, S/WAN ve STT gibi birçok uygulama alanı bulunmaktadır. Web sitelerinde kredi kartı kullanımında güvenlik sertifikası yönetiminde de RSA algoritması kullanılmaktadır. Bu kısımda RSA algoritması için bazı standartlar verilmiştir. Standartların çalışma adımları için (Wong, 2021) çalışmalarından faydalanılabilir.

RSA-PKCS#1 v1.5: PKCS#1 kriptografide 90'lı yılların başında, RSA Laboratories tarafından yayınlanan Public-Key Cryptography Standards (PKCS) adı verilen standartların ilkidir.

PKCS#1 standardı, RSA açık ve gizli anahtarlarının sahip olması gereken matematiksel tanımları ve özellikleri tanımlamaktadır. RSA PKCS#1 v1.5 standardı şifrelemeden önce mesaja bir dizi rasgele bayt ekleyen bir padding tanımlayarak mesajın boyutunu en üst düzeye çıkarmaktadır.

PKCS#1 standardı bilinen bazı sorunları düzeltirken, 1998'de Bleichenbacher, PKCS#1 sürüm 1.5 üzerinde, bir saldırganın standart tarafından belirtilen padding ile şifrelenmiş mesajların şifresini çözmesine izin veren pratik bir saldırı bulunmuştur. Bu saldırı bir milyon mesaj gerektirdiğinden, meşhur bir şekilde "Milyon Mesaj Saldırısı (Million Message Attacks)" olarak adlandırılmaktadır.

Bleichenbacher, RSA açık anahtarlı şifreleme sistemine karşı BB'98 saldırısı tasarlamıştır. BB'98 saldırısı RSA PKCS#1 şifreleme standardına karşı seçilen şifreli metin saldırısı (CCA) dır. 1998'de Daniel Bleichenbacher, SSL protokolünün bir sürümü

de dahil olmak üzere PKCS#1 ile uyum içinde RSA şifrelemesi kullanan sistemlere karşı pratik bir saldırı göstermiştir. Bleichenbacher saldırısı iyi bilinmesine rağmen, bugün hala kullanımda olan ve şifreleme için RSA PKCS#1 v1.5'i uygulayan birçok sistem vardır.

OAEP ile Asimetrik Şifreleme: 1998'de, aynı PKCS#1 standardının 2.0 sürümü, RSA için Optimal Asymmetric Encryption Padding (OAEP) adı verilen yeni bir padding şemasıyla piyasaya sürülmüştür. PKCS#1 v1.5'in aksine OAEP, Bleichenbacher'ın saldırısına karşı savunmasız değildir ve bu nedenle günümüzde RSA şifrelemesi için kullanılacak güçlü bir standarttır.

Algoritmanın merkezinde bir Maske Üretim Fonksiyonu (Mask Generation Function) (MGF), bir girdiyi rastgele seçmek ve büyütmek veya küçültmek için kullanılmaktadır. Ayrıca, şema bir şifreli metni değiştirerek iyi biçimlendirilmiş bir düz metin elde etmeyi imkansız hale getirdiğinden, Bleichenbacher saldırısının artık çalışılmaması gerektiği düşünülmektedir.

OAEP'yi güvenli bir şekilde uygulamak, PKCS#1 v1.5'e kıyasla çok daha basittir. Ayrıca yıllar içinde daha iyi yapılar önerilmiş ve standartlaştırılmıştır. Örneğin, daha güçlü güvenlik kanıtlarına sahip olan ve güvenli bir şekilde uygulanması çok daha kolay olan RSA-KEM standardıdır. Günümüzde RSA kullanan çoğu protokol ve uygulama, hala güvenli olmayan PKCS#1 v1.5 veya OAEP'yi uygulamaktadırlar.

5. ASAL SAYI TEST YÖNTEMLERİ

Asal sayılar açık anahtarlı kriptografide önemli bir yere sahip olduğu için bir sayının asal olup olmadığını belirlemeye yönelik birçok çalışma yapılmıştır. Verilen tek bir sayının asal olup olmadığını belirlemek için kullanılan yöntemlere "asal sayı test yöntemleri" denir. Asal sayı test yöntemleri bu çalışmada kısaca asallık testleri olarak adlandırılacaktır.

Bu bölümde Lucas-Lehmer ve AKS kesin (deterministic) asallık testlerine ve Fermat, Slovay-Strassen, Miller-Rabin, Lehmann ve Frobenius olasılıksal (probabilistic) asallık testlerine yer verilecektir. Ayrıca, günümüze kadar olan çalışmalar ile ilgili bilgi verilecek ve asal sayı testlerinin önemi vurgulanacaktır.

Asal sayı test yöntemlerinin sağlaması gereken temel kriterler aşağıda verilmektedir.

- **Doğruluk**: Algoritma her zaman doğru sonucu vermelidir.
- **Genellik**: Algoritma tüm sayılar için çalışmalıdır.
- **Hız:** Algoritma, polinom zamanda çalışmalıdır.

Deterministik. Verilen bir algoritma da herhangi bir rastgelelik bulunmayan ve her zaman aynı girdilerle aynı sonucu veren, sonucun kesin olarak belirlenebildiği algoritmalara deterministik algoritmalar denir.

Teorem 5.1. (Asal Sayı Teoremi). Asal Sayı Teoremi 1791 yılında Gauss tarafından herhangi bir sayıya kadar kaç tane asal sayı olduğunu hesaplamak için sunulan bir teoremdir. $\pi(n)$ fonksiyonu,

$$\pi(n) \approx \frac{n}{\log n}$$

yaklaşımı ile n sayısından küçük asal sayıların sayısını vermektedir.

Bu teorem asal sayıların oldukça yaygın olduğunu söylemektedir. Örneğin, 2^{1024} ten küçük asal sayıların sayısı yaklaşık olarak 2^{1014} dür. Asal Sayı Teoremi ayrıca rastgele bir sayının olasılığının tahmin edilmesini sağlar. Eğer p rastgele seçilen bir sayı ise p nin asal olma olasılığı yaklaşık olarak $\frac{1}{\log p}$ dir (Smart, 2016).

5.1. Kesin Asallık Testleri

Kesin asallık testleri bir sayının asal olup olmadığını kesin olarak belirlemek için kullanılan testlerdir. Bu testler çalışırken olası asallık testlerinden daha fazla zamana ihtiyaç duyarlar. Bu bölümde Lucas-Lehmer asallık testini ve AKS (Agrawal, Kayal, Saxena) asallık testini vereceğiz.

5.1.1. Lucas-Lehmer Testi

Lucas-Lehmer Testi 1856 yılında Lucas tarafından yapılan çalışmaların 1930 yılında Lehmer tarafından geliştirilmesi ile ortaya çıkmıştır. Mersenne sayılarının asal sayı olup olmadığını belirlemek için kullanılan bir testtir (Pomerance, 2010).

n tek bir asal sayı olmak üzere $M_n = 2^n - 1$ Mersenne sayısının asallığını belirlemek için başlangıç şartı $s_0 = 4$ olan s_i dizisi tanımlayalım.

$$s_i = \begin{cases} 4 & \text{eğer } i = 0, \\ s_{i-1}^2 - 2 & \text{aksi halde,} \end{cases}$$

Bu dizinin birkaç terimi $s_0 = 4,\ s_1 = 4^2 - 2 = 14,\ s_2 = 14^2 - 2 = 194$ ... şeklindedir. $n > 2$ ve $M_n = 2^n - 1$ Mersenne sayısı için Lucas-Lehmer testi,

$$M_n \text{ asal sayıdır } \Leftrightarrow\ {s_i}^2 - 2 \equiv 0 \mod M_n$$

şeklinde ifade edilir (Crandall & Pomerance, 2005). Bu testin algoritması aşağıda verilmiştir.

Algoritma 8: Lucas-Lehmer Testi

$n > 2$ için, $M_n = 2^n - 1$ sayısının asal olup olmadığını belirleme
Girdi: $s_0 = 4$ ve $M_n = 2^n - 1$
Çıktı: M_n asal veya bileşik
for $(i = 0; i < n - 2; i++)$
$s_{i+1} = ((s_i \cdot s_i) - 2 \mod M_n$
if $s_{i+1} == 0$ **return** ASAL **else return** BILESIK
end if
end for

Lucas-Lehmer asallık testi algoritmasının örneklerini verelim.

Örnek 5.1.1.1. $n = 7$ için M_7 Mersenne sayısının asal olup olmadığını Lucas-Lehmer testi ile test edelim. $M_7 = 2^7 - 1 = 127$ dir. $n - 2 = 5$ kez testi çalıştıralım.

$S_0 = 4 \equiv 4 \mod 127$
$S_1 = 14 \equiv 14 \mod 127$
$S_2 = 194 \equiv 67 \mod 127$
$S_3 = 37634 \equiv 42 \mod 127$
$S_4 = 1416317954 \equiv 111 \mod 127$
$S_5 = 2005956546822746114 \equiv 0 \mod 127$
$S_5 \equiv 0 \mod 127$ olduğu için, M_7 Mersenne sayısı asal sayıdır.

Örnek 5.1.1.2. $n = 11$ için $M_{11} = 2047$ Mersenne sayısının asal olup olmadığını Lucas-Lehmer testi ile test edelim. $M_{11} = 2^{11} - 1 = 2047$ dir. $n - 2 = 9$ kez algoritmayı çalıştıralım.
$S_0 = 4 \mod 2047 \equiv 4$
$S_1 = (4^2 - 2) \mod 2047 \equiv 14$
$S_2 = (14^2 - 2) \mod 2047 \equiv 194$
$S_3 = (194^2 - 2) \mod 2047 \equiv 788$
$S_4 = (788^2 - 2) \mod 2047 \equiv 701$
$S_5 = (701^2 - 2) \mod 2047 \equiv 119$
$S_6 = (119^2 - 2) \mod 2047 \equiv 1877$
$S_7 = (1877^2 - 2) \mod 2047 \equiv 240$
$S_8 = (240^2 - 2) \mod 2047 \equiv 282$
$S_9 = (282^2 - 2) \mod 2047 \equiv 1736$

Sonuç olarak, $S_9 \not\equiv 0 \mod 2047$ olduğu için M_{11} Mersenne sayısı bileşik sayıdır.

5.1.2. AKS Testi

2002'de Manindra Agrawal, Neeraj Kayal ve Nitin Saxena tarafından önerilen AKS asallık testi hem polinom zamanda çalışan hem de kesin sonuç veren bir asallık testidir. Bu test verilen bir tek sayının asal sayı olup olmadığını polinom zamanda kesin olarak belirlemektedir. Dolayısıyla, bu test yöntemi açık anahtarlı kriptografi için oldukça önemlidir.

Antik Yunan'da başlayan asallık serüveni Hindistan'da 3 bilgisayar mühendisi tarafından geliştirilerek devam ettirilmiştir. M. Agrawal, N. Kayal ve N. Saxena 2004 yılında hem deterministik hem de polinom zamanda çalışan bir asal sayı test yöntemi geliştirmişledir. Bu test AKS testi olarak adlandırılmıştır.

Asallık testinin hem deterministik olması hem de polinom zamanda çalışması çok önemlidir. Bunlardan ilki deterministik olma, diğer bir ifade ile algoritmanın her aşamasının ve sonuçlarının kesinlik içermesi anlamına gelir. Polinom zamanlılık ise algoritmanın karmaşıklığının girdi olarak verilen değerin bit sayısına bağlı bir polinom tarafından sınırlanmasıdır. AKS testi aşağıdaki gibi açıklanabilir.

$a \in \mathbb{Z},\ n \in \mathbb{N}\ n \geq 2$ ve $(a,n) = 1$ için $(x-a)^n \equiv (x^n - a) \bmod n$ dir. Bu denklem, Fermat'ın küçük teoreminin genişletilmiş halidir ve n ile aralarında asal a değerleri bulunmaktadır. Algoritmayı vermeden önce iki fonksiyonu tanımlayalım.
$\varphi(r)$ Euler Totient fonksiyonudur.
$O_n\ (r)$ ifadesi $\text{obeb}(n,r) = 1$ olmak üzere r sayısının n modülüne göre mertebesidir (multiplicative order). Diğer bir ifade ile, $r^k \equiv 1 \bmod n$ denkliğini sağlayan en küçük k değeridir, $O_n\ (r) = k$.

Algoritmanın çalışması aşağıda verilmiştir:

Girdi: : $n > 1$
- $a > 0$ ve $b > 1$ için,
- eğer $n = a^b$ eşitliği sağlanıyorsa, n sayısı bileşik sayıdır.
- $O_r\ (n)\ > \log_2 n)$ denklemini sağlayan en küçük r değeri bulunur.
- $1 < \text{obeb}\ (a,n) < n$ denklemini sağlayan bir $a \leq r$ değeri bulunabiliyorsa sayı bileşik sayıdır.
- eğer $n \leq r$ ise n asal sayıdır.
- $1 < a < |\sqrt{\varphi(r)} \cdot (\log(n)|$ değerine kadar olan değerler için
$(x+a)^n \neq x^n + a\ (\text{mod}\ x^r - 1,\ n)$ eşitsizliği sağlanıyorsa sayı bileşik sayıdır.

Çıktı: Sayı asal veya bileşiktir.

AKS asallık testi algoritması için aşağıdaki örnek verilebilir.

Örnek 5.1.2.1. $r = 2$ ve $n = 91$ sayıları için 91 sayısının AKS testine göre asal olup olmadığını araştıralım.
- $r = 2$ ve $n = 91$ sayıları için $(\log_r n)$ değeri hesaplanır. $\log_2\ (91)$ = 6,5078 dir.
- $\text{obeb}\ (91,2)$ = 1 dir.
- $O_{91}(2) = 12$ değeri bulunur. Bulunan bu değer için en küçük r değeri hesaplanır. $O_n(r) > 6{,}5078$ büyüklüğündeki en küçük sayı bulunana kadar işlem devam eder.
- $r = 7$ için $\text{obeb}\ (91{,}7)\ =\ 7 \neq 1$ olduğu için $n = 91$ sayısı bileşik sayıdır.

AKS testinin algoritması Algoritma 9 da verilmiştir.

Algoritma 9: AKS Testi

Girdi: $n > 1$
Çıktı: n asal veya bileşik sayıdır
if ($n = a^b$ olmak üzere $a \in \mathbb{N}$ ve $b > 1$) **return** "BILESIK";
    $O_n(r) > \log_2 n$ olacak şekilde en küçük r bulunur.
if ($1 <$ obeb $(a, n) < n$ olacak şekilde $a \leq r$ varsa) **return** "BILESIK" **end if**
    if ($n \leq r$) **return** "BILESIK" **end if**
for $a = 1$ **to** $\sqrt{(\varphi(r) \cdot \log(n))}$ do
    if $((x + a)^n \neq x^n + a \pmod{x^r - 1,\ n})$ **return** "BILESIK " **end if**
    end if
end for
(Agrawal, Kayal, & Saxena, 2004)

Örnek 5.1.2.2. $r = 31$ için $n = 524287$ sayısının AKS testine göre asal olup olmadığını araştıralım.

- $r = 31$ ve $n = 524287$ sayıları için $\log_{31}(524287) = 3{,}8351$ dir.
- obeb $(524287, 31) = 1$ dir.
- $O_{524287}(31) = 524286$ değeri bulunur. Bulunan bu değer için en küçük r değeri hesaplanır. $O_n(r) > 3{,}8351$ büyüklüğündeki en küçük sayı bulunana kadar işlem devam eder.
- $r = 5$ için obeb $(524287{,}5) = 1$
- $r = 7$ için obeb $(524287{,}7) = 1$
- $r = 11$ için obeb $(524287{,}11) = 1$
  ⋮
- $r = 524286$ değerine kadar olan tüm asal r sayıları için obeb $(31, r) = 1$ olduğu için $n = 524287$ sayısı asal sayıdır.

5.2. Olası Asallık Testleri

Bu bölümde olası asallık testleri incelenecek ve bu testlerin algoritmaları verilecektir. Ayrıca, bazı önemli sonuçlara ve örneklere yer verilecektir.

Olası Asallık Testleri (OAT) bir sayının bileşik sayı veya yüksek olasılıkla asal sayı olduğunu gösterir. Kriptografide kullanılacak olan asal sayıları belirlemek için önce n rassal sayısı üretilir. Daha sonra üretilen sayı asallık testinden geçirilir. OAT'de test sayısı artırılarak hata payı düşürülebilir. En çok kullanılan olası asallık testleri Fermat testi, Solovay-Strassen testi ve Miller Rabin testidir.

Öncelikle bu bölümü anlamamızı kolaylaştıracak bazı olasılıksal tanımları vereceğiz. Tanımlar verilirken (Akdeniz, 2015) kaynağından yararlanılmıştır.

Örnek Uzay: Bir deneyin tüm olanaklı sonuçlarının kümesine örnek uzay denir. S ile gösterilir.

***Olay*:** Örnek uzayın herhangi bir alt kümesine olay denir.

Rasgele olay: Gerçekleşmesi rastlantıya bağlı olan olaya rasgele olay denir.

Ayrık olaylar: $A \cap B = \emptyset$ ise A ve B olayları ayrıktırlar; diğer bir ifade ile A ve B nin kesişimi boş kümedir.

Bir olayın olasılığı**:** Bir deneyin eşit olasılıklı N tane sonucu olsun. Bu sonuçların M tanesinden herhangi biri gerçekleştiğinde A olayı gerçekleşmiş olsun. A olayının $P\ (A)$ ile gösterilen gerçekleşme olasılığı,

$$P\ (A) = \frac{\text{Gerçekleşen Sonuçların Sayısı}}{\text{Tüm Sonuçların Sayısı}} = \frac{M}{N}$$

Eşit Olasılıklı Olaylar**:** Bir örnek uzayındaki tüm olayların ortaya çıkma olasılığı eşit ise bu olaylara eşit olasılıklı olaylar denir.

Bağımsız Olaylar**:** Eğer bir olayın ortaya çıkması öteki olayın ortaya çıkma olasılığını etkilemiyorsa, olaylar bağımsız olaylardır.

Bağımlı Olaylar**:** Bir olayın ortaya çıkması diğerinin ortaya çıkması olasılığını etkiliyorsa bağımlı olaylardır.

Rassal Sayılar**:** Herhangi bir kurala bağlı olmadan dizilen sayılardır.

Şahit (Witness) Kavramı. Olası asallık testleri, şahit kavramına dayanmaktadır. n sayısının bileşik sayı olduğuna şahitlik eden 1 ile n arasındaki sayılara "şahit sayıları" denir. Şahit sayıların yoğunluğu (d değeri) olası asallık testlerine göre değişmektedir (Segre, 2000).

Olası asallık testlerinde incelenen bir sayının i iterasyon sonunda asal olarak belirlenmesine rağmen bileşik olma olasılığı da vardır. Şahit sayıların yoğunluğu d olarak kabul edilirse, bir sayının bileşik olma olasılığı (1) ile asal olma olasılığı ise (2) ile hesaplanmaktadır.

$$P\ (\text{bileşik sayı}) = (1-d)^i \qquad (1)$$
$$P\ (\text{asal sayı}) = 1 - P(\text{bileşik sayı}) \qquad (2)$$
$$= 1 - (1-d)^i$$

OAT sonucunda hata payının daha da düşmesi için iterasyon değeri arttırılabilir (Schneier, 1996).

$E_n(t)$ Hata Oranı: n sayısı asal olup olmadığı test edilecek bir sayı olmak üzere $0 \neq a \in \mathbb{Z}_n$ sayıları için t kez tekrarlanan bir algoritmada seçilen a sayıları n sayısının bileşik olduğu sonucunu verdiği kabul edilirse, testin n sayısının asal olduğunu söyleme olasılığı en fazla $\frac{1}{2^t}$' dir. Bu değere n sayısının hata olasılığı denir ve $E_n(t)$ ile gösterilir. Sonuç olarak n sayısı $E_n(t) = \frac{1}{2^t}$ hata oranı ile olası asal sayıdır.

Birbirinden bağımsız a sayıları seçilerek algoritmalar tekrarlanırsa n sayısının bileşik sayı sonucunu vermesi testin pozitif olduğu anlamına gelmektedir. Bu durumun tersi olan, n sayısının asal sonucunu vermesi testin negatif olduğu anlamına gelmektedir. O halde test a seçeneklerinin en az %50 si için pozitiftir (Gallier & Quaintance, 2019). Testin negatif sonuç vermesi N ile pozitif sonuç vermesi P ile gösterilirse,
(Testin t kez tekrarı sonucu)

$$N = N^t$$
$$= (1-P)^t$$

$$\leq \left(1-\frac{1}{2}\right)^t$$
$$= \left(\frac{1}{2}\right)^t$$

dir. Sonuç olarak, n sayısı $E_n(t) = \frac{1}{2^t}$ hata oranı ile olası asal sayıdır. $0 \neq a \in \mathbb{Z}_n$ olmak üzere her a değeri aynı sabit olasılıktadır. Bu yüzden bu a sayıları tekdüze dağılımlıdır (sürekli, uniform). Olası asallık testleri için bu oran birbirinden farklılık göstermektedir.

5.2.1. Fermat Olası Asallık Testi

Fermat olası asallık testi, olası asallık testlerinin temelini oluşturan ilk testtir. Fermat tarafından 1640 yılında önerilmiş olan Fermat'ın küçük teoremine dayanmaktadır. Öncelikle Fermat'ın küçük teoremini verelim. Teoremin ifadesi ve ispatı (Koblitz, 1994) çalışmalarında ayrıntılı verilmiştir.

Teorem 5.2.1.1. (Fermat'ın Küçük Teoremi). p bir asal sayı ve a bir tam sayı olsun. Bu durumda p asal sayısı, $a^p - a$ sayısını böler. Dolayısıyla p asal sayısı a sayısını bölmüyorsa, $a^{p-1} - 1$ sayısını böler. Basit bir ifade ile eğer p sayısı asal ise $a^{p-1} \equiv 1 \bmod p$ dir.

İspat. $p \nmid a$ olduğunu kabul edelim. $0a, 1a, 2a, 3a, \dots, (p-1)a$ sayılarının p modülünün kalan kümesi olduğunu varsayalım. $ia \; ve \; ja$ aynı kalan sınıfında $ia \equiv ja \bmod p$ olması gerekir. Ama $p \mid (i-j)a$ ve a, p ile bölünemediği için $p \mid i-j$ sonucunu elde ederiz. Bu durumda i ve j nin her ikisi de p den küçük olduğu için $i = j$ dir. $a, 2a, \dots, (p-1)\,a$ tamsayılarının, $\bmod p$ olarak kabul edildiğinde $1, 2, \dots, p-1$'i yeniden düzenleyelim. Böylece, birinci dizideki sayıların çarpımı, ikinci dizideki sayıların çarpımı olan $a^{p-1}\,(p-1)! \equiv (p-1)! \bmod p$ ile kongrüenttir. Böylelikle $p \mid ((p-1)!\;(a^{p-1}-1))$ dir. $(p-1)!$, p ile bölünmediğinden $p \mid (a^{p-1}-1)$ olması gerekir. Son olarak, $a^{p-1} \equiv 1 \bmod p$ kongrüansının her iki tarafını da a ile çarparsak, a'nın p ile bölünemediği durumda önermenin ifadesindeki ilk kongrüansı elde ederiz.

NOT: Teorem 5.2.1.1. de verilen ifadenin karşıtı doğru değildir. Açıkça ifade etmek gerekirse, $p \nmid a$ için $a^{p-1} \equiv 1 \bmod p$ olması p sayısının asal olmasını gerektirmez.

Sonuç 5.2.1.1. Eğer p bir asal sayı ise herhangi bir a sayısı için $a^p \equiv a \bmod p$ olur.

İspat. $p \mid a$ ise $ap \equiv 0 \equiv a \bmod p$. Eğer $p \nmid a$ ise Fermat Teoremi'nden,

$$a^{p-1} \equiv 1 \bmod p$$

elde edilir. Bu kongrüansın her iki tarafı a ile çarpılırsa, $a^p \equiv a \bmod p$ elde edilir.

Fermat'ın küçük teoreminin denk ifadesi: Eğer $a^{p-1} \not\equiv 1 \bmod p$ ise p sayısı bileşik sayıdır. Diğer bir ifade ile herhangi bir n sayısının asal olması için $a \in \mathbb{Z}_n$ olmak üzere $a^{(n-1)} \equiv 1 \bmod n$ denkliğinin sağlanması gerekmektedir, fakat yeterli değildir. Fermat'ın küçük teoremine dayanan Fermat olası asallık testini açıklayalım. Çalışmanın devamında Fermat olası asallık testi, Fermat testi olarak ifade edilecektir.

Fermat Testi : $n > 3$ tek tam sayısı ve $a \in \mathbb{Z}_n$ sayısı verilsin öyleki $\text{obeb}\,(a, n) = 1$ olmak üzere eğer

$a^{(n-1)} \equiv 1 \bmod n$ denkliği sağlanmıyorsa, n sayısına bileşik sayı denir,
$a^{(n-1)} \equiv 1 \bmod n$ denkliği sağlanıyorsa, n sayısına olası asal sayı denir.

Aşağıdaki adımlar t kez tekrarlanarak n sayısının bileşik sayı olduğu veya t sayısına bağlı olarak belirli bir hata oranı ile olası asal sayı olduğu sonucuna varılır.

- Rastgele bir $a \in \mathbb{Z}_n$ sayısı seçilir.
- obeb (a, n) değeri hesaplanır.
- eğer obeb $(a, n) \neq 1$ ise, n sayısı bileşik sayıdır.
- eğer obeb $(a, n) = 1$ ise, $b = a^{n-1} \bmod n$ hesaplanır.
- eğer $b \neq 1$ ise n sayısı bileşik sayıdır.
- eğer $b = 1$ ise n sayısı olası asal sayıdır.

Bu testin algoritması Algoritma 10 da verilmiştir. Bu algoritma n sayısının asal olup olmadığını test etmek için t tane farklı a sayısı için çalıştırılmaktadır ve $E_n(t)$ hata oranı ile çalışmaktadır.

Algoritma 10: Fermat Testi

Girdi: n ve $t \in \mathbb{Z}^+$
Çıktı: n sayısı bileşik sayıdır ya da $E_n\ (t)$ hata oranı ile asal sayıdır.
1: **for** $\{2, \dots, n-2\}$ kümesinden rastgele t tane a
2: $d \leftarrow$ obeb (a, n)
3: **if** $d > 1$ **then return** "bilesik" **break**
4: **else** $b \leftarrow a^{n-1} \bmod n$
5: **end if**
6: **if** $b \neq 1$ **then return** "bilesik" **break end if**
7: **end for**
8: **return** "n olası asal"

Bu testte kullanılan Fermat şahidi ve Fermat yalancısı sayılarının tanımını verelim.

Tanım 5.2.1.1. (Fermat şahidi) n bileşik bir tek tam sayı olsun. $a \in \mathbb{Z}_n$ için $a^{n-1} \equiv 1 \bmod n$ denkliği sağlanmaz ise bu a sayısı n sayısının bileşik sayı olduğunun kanıtıdır ve bu a sayısı Fermat şahidi olarak adlandırılır.

Tanım 5.2.1.2. (Fermat yalancısı) n bileşik tek tam sayı olsun. . $a \in \mathbb{Z}_n$ için $a^{n-1} \equiv 1 \bmod n$ denkliği sağlanırsa bu a sayısına Fermat yalancısı denir.

Hata Oranı: Bileşik n sayısı için, testin t kez tekrarlanması sonucu n sayısının asal olduğunu söyleme olasılığı en fazla $E_n(t) = \frac{1}{2^t}$ dir. Bu olasılık testin hata oranı olarak adlandırılır.

Zaman Karmaşıklığı: Fermat testi algoritmasının karmaşıklığı t paremetresine bağlı olarak aşağıdaki gibi verilebilir. 2. Adımda çalışan Öklid algoritmasının (bkz. Algoritma 1) maliyeti $O(log^3\ n)$ işlemdir. 4. adımda tekrarlanan kare alma algoritmasının (bkz. Algoritma 4) maliyeti $O(log^3\ n)$ işlemdir. Dolayısıyla t defa çalıştırılan Fermat testinin karmaşıklığı $O\ (t\ log^3\ n)$ işlemdir. Sonuç olarak bu olası asallık testi polinom zamanda çalışan bir test yöntemidir.

Fermat asallık testi algoritmasının örnekleri aşağıda verilmiştir.

Örnek 5.2.1.1. 91 sayısının asal olup olmadığını Fermat testi ile test edelim.
Girdi: $n = 91$ ve $t = 7$ kez test yapalım.
1: $\{2, \ldots, 89\}$ kümesinden rastgele 7 tane a
2: $a = 2$ için $a^{n-1} = 2^{90} \equiv 64 \bmod 91$
3: $a = 7$ için $a^{n-1} = 7^{90} \equiv 77 \bmod 91$
4: $a = 13$ için $a^{n-1} = 13^{90} \equiv 78 \bmod 91$
5: $a = 21$ için $a^{n-1} = 21^{90} \equiv 77 \bmod 91$
6: $a = 35$ için $a^{n-1} = 35^{90} \equiv 14 \bmod 91$
7: $a = 45$ için $a^{n-1} = 45^{90} \equiv 64 \bmod 91$
8: $a = 77$ için $a^{n-1} = 77^{90} \equiv 14 \bmod 91$
Çıktı : "91 bileşik sayıdır."

Burada $a = 2, 7, 13, 21, 35, 45, 77$ sayıları Fermat şahididir.

Fermat'ın küçük teoremine göre, p bir asal sayı ve a bir tam sayı olmak üzere p asal sayısı, $a^p - a$ sayısını böler. Fakat bu bölme işlemini sağlayan ve asal olmayan Carmichael sayıları vardır. 561 bileşik Carmichael sayısına Fermat testini uygulayalım.

Örnek 5.2.1.2. 561 sayısının asal olup olmadığını Fermat testini kullanarak test edelim.
Girdi : $n = 561$ ve $t = 7$ kez test yapalım.
1: $\{2, \ldots, 559\}$ kümesinden rastgele 7 tane a
2: $a = 13$ için $a^{n-1} = 13^{560} \equiv 1 \pmod{561}$
3: $a = 29$ için $a^{n-1} = 3^{560} \equiv 1 \pmod{561}$
4: $a = 52$ için $a^{n-1} = 52^{560} \equiv 1 \pmod{561}$
5: $a = 76$ için $a^{n-1} = 76^{560} \equiv 1 \pmod{561}$
6: $a = 125$ için $a^{n-1} = 125^{560} \equiv 1 \pmod{561}$
7: $a = 128$ için $a^{n-1} = 128^{560} \equiv 1 \pmod{561}$
8: $a = 142$ için $a^{n-1} = 142^{560} \equiv 1 \pmod{561}$
Çıktı : "561 olası asal sayıdır"
Fakat $561 = 3 \cdot 11 \cdot 17$ bileşik bir sayıdır, dolayısıyla $a = 13, 29, 52, 76, 125, 128, 142$ sayıları 561 sayısı için Fermat yalancısıdır.

Örnek 5.2.1.3. 6972593 sayısının asal olup olmadığını Fermat testi ile test edelim.
Girdi: $n = 6972593$ ve $t = 7$ kez test yapalım.
1: $\{2, \ldots, 6972591\}$ kümesinden rastgele 7 tane a
2: $a_1 = 2$ için $a^{n-1} = 2^{6972592} \equiv 1 \bmod 6972593$
3: $a_2 = 465$ için $a^{n-1} = 465^{6972592} \equiv 1 \bmod 6972593$
4: $a_3 = 1285$ için $a^{n-1} = 1285^{6972592} \equiv 1 \bmod 6972593$
5: $a_4 = 34567$ için $a^{n-1} = 34567^{6972592} \equiv 1 \bmod 6972593$
6: $a_5 = 47637$ için $a^{n-1} = 47637^{6972592} \equiv 1 \bmod 6972593$
7: $a_6 = 455137$ için $a^{n-1} = 455137^{6972592} \equiv 1 \bmod 6972593$
8: $a_7 = 5662497$ için $a^{n-1} = 5662491^{6972592} \equiv 1 \bmod 6972593$
Çıktı : "6972593 olası asal sayıdır"

Fermat testi $a = 2, 465, 1285, 34567, 47637, 455137, 5662497$ taban sayılarına göre n sayısı için olası asal sayı sonucunu verdi. Dolayısıyla, Fermat testi n sayısının $E_n(7) = \frac{1}{2^7} = \frac{1}{128}$ hata oranı ile olası asal sayı olduğunu söyler. $n = 6972593$ sayısı gerçekten de asal bir sayıdır.

5.2.2. Solovay- Strassen Testi

Solovay-Strassen testi Robert Solovay ve Volker Strassen tarafından ortaya çıkarılan açık anahtar kriptografisinde kullanılmış ilk olası asallık testidir. Slovay-Strassen testi tek bir n sayısının asal olup olmadığını belirlemek için $J(a,n)$ ile gösterilen Jacobi sembolüne dayanmaktadır. Jacobi sembolü aşağıdaki gibi tanımlanır (bkz. Bölüm 3.3.).

$$\left(\frac{a}{n}\right)=\begin{cases}1 & \text{eğer } x^2 \equiv a \bmod n \text{ ise}\\ -1 & \text{eğer } x^2 \not\equiv a \bmod n \text{ ise}\\ 0 & \text{eğer } a \mid n \text{ ise}\end{cases}$$

Teorem 5.2.2.1. (Euler kriteri) Eğer n tek asal sayı ise, $(a,n)=1$ şartını sağlayan her a pozitif tam sayısı için $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n$ sağlanır. Bu denklik sağlanmaz ise n sayısı bileşik sayıdır. Euler kriterine bağlı Solovay-Strassen olası asallık testini açıklayalım. Çalışmanın devamında Solovay-Strassen olası asallık testi, Solovay-Strassen testi olarak ifade edilecektir.

Solovay-Strassen Testi : n tek bileşik sayı ve $a \in \mathbb{Z}_n$ olmak üzere eğer
$a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \bmod n$ ise n sayısı bileşik sayıdır,
$a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n$ ise n sayısı a tabanına göre olası asal sayıdır.

Solovay - Strassen testinde öncelikle asal olup olmadığı test edilecek n sayısı ve testin kaç kez tekrar edileceğini gösteren t parametresi belirlenir. Algoritmanın aşamaları aşağıda verildiği gibidir.

- n sayısından küçük rastgele bir a sayısı seçilir.
- eğer obeb $(a,n) \neq 1$ ise n sayısı bileşik sayıdır.
- $b = a^{(n-1)/2} \bmod n$ hesaplanır.
- Jacobi sembolü olan $J = \left(\frac{a}{n}\right)$ hesaplanır.
- eğer $b \neq J$ ise n sayısı bileşik sayıdır.
- eğer $b = J$ ise n olası asal sayıdır. Burada, n'nin asal olmama olasılığı %50 den fazla olamaz (Schneier, 1996).

Bu algoritma n sayısının asal olup olmadığını test etmek için t tane farklı a sayısı için çalıştırılmaktadır ve n sayısı için bileşik sayı veya $E_n(t) = 1/2^t$ hata oranı olası asal sayı sonucunu vermektedir. Bu testin algoritması Algoritma 11 de verilmiştir.

Bu testte kullanılan Euler şahidi ve Euler yalancısı sayılarının tanımını verelim.

Tanım 5.2.2.1. (Euler Şahidi) n tek bileşik sayı ve a sayısı $\mathbb{Z}_n$ kümesinde bir eleman olmak üzere, eğer $a^{(n-1)/2} \not\equiv \left(\frac{a}{n}\right) \bmod n$ ise, a sayısına n sayısının "Euler şahidi" denir.

Tanım 5.2.2.2. (Euler Yalancısı) n tek bileşik sayı ve a sayısı $\mathbb{Z}_n$ kümesinde bir eleman olmak üzere, eğer $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n$ ise, a sayısına n sayısının "Euler yalancısı" denir.

Algoritma 11: Solovay-Strassen Testi

Girdi: n ve $t \in \mathbb{Z}^+$
Çıktı: n sayısı bileşik ya da $E_n(t)$ hata oranı ile asal
1: **for** rastgele t tane $0 \neq a \in \mathbb{Z}_n$
2: $b \leftarrow a^{\frac{n-1}{2}} \bmod n$
3: **if** $b \neq \pm 1$ **return** "bileşik" **break end if**
4: $J \leftarrow \left(\frac{a}{n}\right)$
5: **if** $b \neq J \bmod n$ **return** "bilesik" **break end if**
6: **end for**
7: **return** "n olası asal"

Hata Oranı: Test edilecek n sayısı tek bileşik tam sayı ise Euler yalancısı $\mathbb{Z}_n^*$'ın alt grubudur. Bu ise $\mathbb{Z}_n^*$'ın elemanlarının yarısının Euler yalancısı olmadığını verir. Bu nedenle bileşik n sayısı için, testin t kez tekrarlanması sonucu n sayısının asal olduğunu söyleme olasılığı $E_n(t) = \frac{1}{2^t}$ dir. Bu olasılık testin hata oranı olarak adlandırılır.

Zaman Karmaşıklığı: Solovay-Strassen testi algoritmasının karmaşıklığı t paremetresine bağlı olarak aşağıdaki gibi verilebilir. 2. Adımda çalışan tekrarlanan kare alma algoritmasının (bkz. Algoritma 4) maliyeti $O(log^3 n)$ işlemdir. 4. Adımda çalışan Jacobi sembolünün maliyeti $O(\ log^3 n)$ işlemdir. Dolayısıyla t defa çalıştırılan Solovay-Strassen testinin karmaşıklığı $O\ (t\ log^3 n)$ işlemdir. Sonuç olarak bu olası asallık testi polinom zamanda çalışan etkili bir test yöntemidir.

Örnek 5.2.2.1. $n = 41041$ sayısının asal olup olmadığını Solovay-Strassen testi ile belirleyelim.

1: $a_1 = 22$ için $b = a_1^{(n-1)/2} \bmod n = 22^{20520} \bmod 41041 = 18656$
$b \neq \pm 1$ **return** "n "bileşik"

2: $a_2 = 2$ için $b = a_1^{(n-1)/2} \bmod n = 2^{20520} \bmod 41041 = 1$ ve $J \leftarrow \left(\frac{2}{41041}\right) = -1$
$b \neq J$ **return** "n bileşik"

Burada, $a_1 = 22$ ve $a_2 = 2$ taban sayıları $n = 41041$ sayısının bileşik sayı olduğunu söylediği için bu sayılar n bileşik sayısı için Euler şahidi'dir. Diğer taraftan $a_3 = 12$ ve $a_4 = 124$ taban sayıları kullanıldığı zaman

3: $a_3 = 12$ için $b = a_1^{(n-1)/2} \bmod n = 12^{20520} \bmod 41041 = 1$ ve $J \leftarrow \left(\frac{12}{41041}\right) = 1$
$b = J$ **return** "n olası asal"

4: $a_4 = 124$ için $b = a_1^{(n-1)/2} \bmod n = 124^{20520} \bmod 41041 = 1$ ve $J \leftarrow \left(\frac{124}{41041}\right) = 1$
$b = J$ **return** "n olası asal"

olası asal sonucu elde edilmektedir. Dolayısıyla $a_3 = 12$ ve $a_4 = 124$ taban sayıları n bileşik sayısı için Euler yalancısıdır.

Örnek 5.2.2.2. $n = 3021377$ sayısının asal olup olmadığını Solovay-Strassen testi ile test edelim.
Girdi: $n = 3021377$ ve $t = 7$ kez test yapalım.
1: $\{2,3,\dots,3021376\}$ kümesinden 7 tane rastgele a seçilir
2: $a_1 = 2$ için $2^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{2}{3021377}\right) = 1$
3: $a_2 = 287$ için $287^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{287}{3021377}\right) = 1$
4: $a_3 = 800$ için $800^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{800}{3021377}\right) = 1$
5: $a_4 = 2387$ için $2387^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{2387}{3021377}\right) = 1$
6: $a_5 = 37517$ için $37519^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{37517}{3021377}\right) = 1$
7: $a_6 = 754131$ için $754131^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{754131}{3021377}\right) = 1$
8: $a_7 = 2321479$ için $2321479^{1510688} \equiv 1 \bmod 3021377$ ve $\left(\frac{2321479}{3021377}\right) = 1$
Çıktı: "3021377 olası asal sayıdır"

Solovay-Strassen testi $a = 2, 287, 800, 2387, 37517, 754131, 2321479$ taban sayılarına göre n sayısı için olası asal sayı sonucunu vermektedir. Bu test 7 tane a değeri için çalıştırıldığı için $n = 3021377$ sayısı $E_n(7) = \frac{1}{2^7} = \frac{1}{128}$ hata oranı ile olası asal sayıdır. Gerçekten de $n = 3021377$ sayısı asal bir sayıdır.

5.2.3. Miller Rabin Olası Asallık Testi

Verilen tek bir sayının asallığını test etmek için kullanılan en yaygın yöntemlerden biri Michael Rabin tarafından Gary Miller'ın fikirlerine dayanarak geliştirilen hata oranı düşük olan Miller-Rabin (M-R) olası asallık testidir.

Verilen bir n tek sayısı için $n - 1 = 2^s\, r$ olacak şekilde s ve r sayıları hesaplanır. Eğer n sayısı asal sayı ise, $1 \le a \le n - 1$ aralığındaki a sayısı için,

$$a^r \equiv 1 \bmod n \text{ veya } 0 \le j \le s - 1 \text{ için } a^{2^j r} \equiv -1 \bmod n$$

eşitlikleri sağlanır. Denk olarak, eğer $\{0, s-1\}$ aralığındaki her j için $a^r \not\equiv 1 \bmod n$ ve $a^{2^j r} \not\equiv -1 \bmod n$ ise, n asal sayı değildir.

Miller-Rabin olası asallık testinde verilen bir tek n sayısının asal olup olmadığı test edilmektedir. Çalışmanın devamında Miller-Rabin olası asallık testi, Miller-Rabin testi olarak ifade edilecektir.

Miller-Rabin Testi : n tek sayı olmak üzere $n-1 = 2^s r$ öyleki r tek sayı olacak şekilde yazılabilir. a sayısı $\{0, s-1\}$ aralığında olmak üzere,

- eğer $\{0, s-1\}$ aralığındaki her j için $a^r \not\equiv 1 \bmod n$ ve $a^{2^j r} \not\equiv -1 \bmod n$ ise n sayısı bileşik sayıdır,
- eğer $a^r \equiv 1 \bmod n$ veya $a^{2^j r} \equiv -1 \bmod n$ denkliği $\{0, s-1\}$ aralığındaki herhangi bir j için sağlanıyorsa, n sayısına a tabanına göre olası asal sayı denir.

Bu testin algoritması Algoritma 12 de verilmiştir. Bu algoritma n sayısının asal olup olmadığını test etmek için t tane farklı a sayısı için çalıştırılmaktadır ve n sayısı için bileşik sayı veya $E_n(t)$ hata oranı ile olası asal sayı sonucu vermektedir.

Algoritma 12: Miller-Rabin Testi

Girdi: n ve $t \in \mathbb{Z}^+$
Çıktı: n sayısı bileşik ya da $E_n(t)$ hata oranı ile asal
1: $n-1 \leftarrow 2^s r$ (r tek tam sayı)
2: **for** rastgele t tane $0 \neq a \in \mathbb{Z}_n$
3: $b \leftarrow a^r \bmod n$
4: **if** $b \neq \pm 1$ **then**
5: **for** (1 den $s-1$ e kadar olan r ler için)
6: $c \leftarrow a^{2^j r} \bmod n$
7: **if** $c = -1$ ise 2. adıma dön **end if**
8: **end for**
9: **return** "bilesik"
10: **end if**
11: **end for**
12: **return** "n olası asal"

Bu testte kullanılan güçlü yalancı ve güçlü şahit sayılarının tanımını verelim.

Tanım 5.2.3.2. (Güçlü yalancı) n tek bileşik sayı ve $a \in \mathbb{Z}_n$ için, Eğer $a^r \equiv 1 \bmod n$ veya $\{0, s-1\}$ aralığındaki herhangi bir j için $a^{2^j r} \equiv -1 \bmod n$ denkliği sağlanıyorsa, a sayısına n sayısı için "güçlü yalancı" denir.

Tanım 5.2.3.1. (Güçlü şahit) n tek bileşik sayı ve $\mathbb{Z}_n$ kümesinden alınan bir a sayısı için, Eğer $a^r \not\equiv 1 \bmod n$ veya $j \in \mathbb{Z}_n$ için $a^{2^j r} \not\equiv -1 \bmod n$ denkliği sağlanıyorsa, a sayısına n sayısı için "güçlü şahit" denir.

Hata Oranı: Bileşik n sayısı için, testin t kez tekrarlanması sonucu n sayısının asal olduğunu söyleme olasılığı en fazla $E_n(t) = \frac{1}{4^t}$ dir. Bu olasılık testin hata oranı olarak adlandırılır. $E_n(t)$ hata oranı $t > 4$ için ihmal edilebilecek düzeyde küçük bir hatadır (Rosen, Michaels, Gross, Grossman, & Shier, 1999).

Zaman Karmaşıklığı: Miller-Rabin testi algoritmasının karmaşıklığı t paremetresine bağlı olarak aşağıdaki verilebilir. 3. adımda tekrarlanan kare alma algoritmasının (bkz. Algoritma 4) maliyeti $O(t\ log^3\ n)$ işlemdir. Dolayısıyla t defa çalıştırılan Miller-Rabin testinin karmaşıklığı $O\ (t\ log^3\ n)$ işlemdir. Sonuç olarak bu olası asallık testi polinom zamanda çalışan etkili bir bir test yöntemidir.

Miller-Rabin asallık testi algoritmasının örneklerini verelim.

Örnek 5.2.3.1. $n = 1105$ sayısının asal olup olmadığını test etmek için Miller-Rabin testini uygulayalım.

Girdi: $n = 1105$ ve $t = 4$
$n - 1 = 1104 = 2^4 \cdot 69$ burada $s = 4,\ r = 69$ dir.
$i = 1 \rightarrow\ \ a = 17$ için $a^r = 17^{69} \equiv 272 \bmod 1105$
$i = 2 \rightarrow\ \ a = 162$ için $a^r = 162^{69} \equiv 382 \bmod 1105$
$i = 3 \rightarrow\ \ a = 646$ için $a^r = 646^{69} \equiv 391 \bmod 1105$
$i = 4 \rightarrow\ \ a = 1035$ için $a^r = 1035^{69} \equiv 580 \bmod 1105$
Çıktı: 1105 sayısı bileşik sayıdır.

Burada $a = 17, 162, 646, 1035$ sayıları güçlü şahittir. Diğer taraftan, $n = 1105$ sayısı için farklı a sayılarını kullanalım.

Girdi: $n = 1105$ ve $t = 2$
$s = 4,\ r = 69$ olmak üzere,
$i = 1 \rightarrow \quad a_1 = 256$ için $a^r = 256^{69} \equiv 1 \bmod 1105$
$i = 2 \rightarrow \quad a_2 = 341$ için $a^r = 341^{69} \equiv 1 \bmod 1105$
Çıktı: 1105 olası asaldır.

Burada $a_1 = 256$ ve $a_2 = 341$ taban sayıları için 1105 sayısını olası asal olarak verdi. Fakat 1105 asal sayı değildir, dolayısıyla $a_1 = 256$ ve $a_2 = 341$ sayıları güçlü yalancılardır.

Örnek 5.2.3.2. $n = 2976221$ sayısının asal olup olmadığını Miller-Rabin testi ile test edelim.
Girdi: $n = 2976221$ ve $t = 4$ kez test yapalım.
$n - 1 = 2976220 = 2^2 \cdot 744055$ burada $s = 2$ ve $r = 744055$ dir.
$i = 1 \rightarrow\ a_1 = 197$ için $a^r = 197^{744055} \equiv 1 \bmod 2976221$
$i = 2 \rightarrow\ a_2 = 3488$ için $a^r = 3488^{744055} \equiv 1 \bmod 2976221$
$i = 3 \rightarrow\ a_3 = 42561$ için $a^r = 42561^{744055} \equiv 1 \bmod 2976221$
$i = 4 \rightarrow\ a_4 = 1721446$ için $a^r = 1721446^{744055} \equiv 1 \bmod 2976221$
Çıktı: "3021377 olası asal sayıdır"

Burada a_1, a_2, a_3, a_4 taban sayılarına göre n sayısı için olası asal sayı sonucunu verdi. Dolayısıyla, Miller-Rabin testi bu dört tane a değeri için n sayısının $E_n(4) = \frac{1}{4^4} = \frac{1}{256}$ hata oranı ile olası asal sayı olduğu sonucunu vermektedir. Gerçekten de $n = 2976221$ sayısı asal bir sayıdır.

5.2.4. Lehmann Olası Asallık Testi

Lehmann tarafından 945 yılında geliştirilen Lehmann olası asallık testi, Legendre tarafından geliştirilen Legendre sembolüne (bkz. Bölüm 3.3.) dayanmaktadır.

Verilen bir n sayısının asal olup olmadığını test etmek için aşağıdaki adımlar gerçekleştirilir.

- n sayısından küçük rastgele bir a sayısı seçilir.
- $b = a^{(n-1)/2} \bmod n$ hesaplanır.
- eğer sonuç 1 veya -1 ise, n sayısı bileşik sayıdır.
- eğer sonuç 1 veya -1 den farklı ise, n sayısı olası asal sayıdır (Schneier, 1996).

Bu testin algoritması aşağıda verilmiştir. Çalışmanın devamında Lehmann olası asallık testi, Lehmann testi olarak ifade edilecektir.

Algoritma 13: Lehmann Testi
Girdi: n ve $0 \neq a \in Z_n$
Çıktı: n sayısı bileşik ya da $E_n(t)$ hata oranı ile asal
1: $b = a^{(n-1)/2} \bmod n$
2: **if** $b = \pm 1$ ise **return** n sayısı bileşik sayıdır **end if**
3: **if** $b \neq \pm 1$ ise **return** n sayısı olası asal sayıdır **end if**

Hata Oranı: Bileşik n sayısı için, testin t kez tekrarlanması sonucu n sayısının asal olduğunu söyleme olasılığı en fazla $E_n(t) = \frac{1}{2^t}$ dir. Bu olasılık testin hata oranı olarak adlandırılır.

Zaman Karmaşıklığı: Lehmann testi algoritmasının karmaşıklığı t paremetresine bağlı olarak aşağıdaki gibi verilebilir. 2. adımda tekrarlanan kare alma algoritmasının (bkz. Algoritma 4) maliyeti $O(t\ log^3\ n)$ işlemdir. Dolayısıyla t defa çalıştırılan Lehmann testinin karmaşıklığı $O\ (t\ log^3\ n)$ işlemdir. Sonuç olarak bu olası asallık testi polinom zamanda çalışan bir test yöntemidir.

Lehmann asallık testi algoritmasının örneğini verelim.

Örnek 5.2.4.1. $n = 1279$ sayısının asal olup olmadığını Lehmann testi ile test edelim.

Girdi: $a < 1279$

$a_1 = 143$ için $a_1{}^{(n-1)/2} \bmod n = 143^{639} \bmod 1279 \equiv 1$
$a_2 = 167$ için $a_2{}^{(n-1)/2} \bmod n = 167^{639} \bmod 1279 \equiv 1$
$a_3 = 324$ için $a_3{}^{(n-1)/2} \bmod n = 324^{639} \bmod 1279 \equiv 1$
$a_4 = 521$ için $a_4{}^{(n-1)/2} \bmod n = 521^{639} \bmod 1279 \equiv 1$
$a_5 = 624$ için $a_5{}^{(n-1)/2} \bmod n = 624^{639} \bmod 1279 \equiv 1$
$a_6 = 743$ için $a_6{}^{(n-1)/2} \bmod n = 743^{639} \bmod 1279 \equiv 1$
$a_7 = 1011$ için $a_7{}^{(n-1)/2} \bmod n = 1011^{639} \bmod 1279 \equiv 1$

Çıktı: 1279 olası asal sayıdır.

Burada $a_1, a_2, a_3, a_4, a_5, a_6, a_7$ taban sayılarına göre n sayısı için olası asal sayı sonucunu vermektedir. Lehmann testi 7 tane a değeri için n sayısının $E_n(7) = \frac{1}{2^7} = \frac{1}{128}$ hata oranı ile olası asal sayı olduğu sonucunu vermektedir. Gerçekten de $n = 1279$ sayısı asal bir sayıdır.

5.2.5. Frobenius Olası Asallık Testi

Frobenius olası asallık testi sonlu cisimlere dayanmaktadır. Fermat, Solovay-Strassen ve Miller-Rabin testlerinin genelleştirilmesi ve güçlendirilmesi ile oluşturulduğu belirtilmektedir (Grantham, 1998).

$f\ (x) \in \mathbb{Z}\ [x]$ ifadesindeki $f\ (x)$ fonksiyonu d dereceli ve Δ diskriminantına sahip bir fonksiyon olsun. Ayrıca en büyük dereceye sahip terimin katsayısı da 1 olsun. n sayısının asal olup olmadığına bakmak için aranan ilk şart $\text{obeb}\ (n, f\ (0)\ \Delta) = 1$ eşitliğidir. Sonra da $\mathbb{Z}_n[x]$ kümesinde hesaplamalar yapılarak şu algoritma uygulanır.

Çarpanlara Ayırma Adımı: Bu adımda aşağıdaki hesaplamalar yapılır,

$f_0\ (x) = f\ (x) \bmod n$
$F_i\ (x) = \text{obeb}\ (x^n - x, f_{i-1}(x))$
$f_i\ (x) = f_{i-1}\ (x)/F_i(x) \quad (1 \le i \le d)$

Bu obeb hesaplarından bir tanesi bile bulunmadığı zaman "n bileşik sayıdır" sonucu verilerek algoritma sonlandırılır.

Frobenius Adımı: $2 \leq i \leq d$ için $F_i(x^n) \bmod F_i(x)$ hesaplanır. Aralıktaki herhangi bir i için 0'dan farklı bir sonuç çıkıyorsa "n birleşik sayıdır" diyerek algoritma sonlandırılır.

Jacobi Adımı: $S = \sum_{2/i} \deg(F_i(x))/i$ değeri için $(-1)^S \neq \left(\frac{\Delta}{n}\right)$ ifadesi sağlandığı zaman "n birleşik sayıdır" denir ve algoritma sonlandırılır.

Hata Oranı: Frobenius testi ile bir bileşik sayıyı asal olarak belirleme hata oranının $E_n(t) = \frac{1}{7710}$ olduğu ölçülmüştür (Grantham, 1998).

Zaman Karmaşıklığı: Frobenius testi algoritmasının karmaşıklığı t paremetresine bağlı olarak aşağıdaki gibi verilebilir. Çarpanlara ayırma adımında çalışan Öklid algoritmasının (bkz. Algoritma 1) maliyeti $O(log^3 n)$ işlemdir. Dolayısıyla t defa çalıştırılan Frobenius testinin karmaşıklığı $O(t\ log^3 n)$ işlemdir. Sonuç olarak bu olası asallık testi polinom zamanda çalışan bir test yöntemidir.

6. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu çalışmanın amacı olası asallık testlerinin performans analizlerini yapmak ve karşılaştırmaktır. Olası asallık testleri ile bir tek sayının bileşik sayı olduğu veya olası asal sayı olduğu belirlenebilir. Olası asallık testlerinden üç temel test olan Fermat, Solovay-Strassen ve Miller-Rabin asallık testleri karşılaştırılırken hata oranı ve çalışma zamanı kriterlerine bakılarak bir değerlendirme yapılacaktır.

- Carmichael sayıları asal olmamasına rağmen Fermat testini geçerler. Bu nedenle Fermat testinin güvenilirliği düşüktür.
- Solovay-Strassen testi, Jacobi sembol hesabından kaynaklanan çalışma zamanı artışı sebebiyle daha uzun sürmektedir.
- Fermat ve Slovay-Strassen testleri $1/2^t$ hata payıyla çalışırken Miller-Rabin testi $1/4^t$ hata payıyla daha gerçeğe yakın sonuçlar sunmaktadır.
- Miller-Rabin Testi ile birlikte Lucas veya Frobenius testleri birlikte kullanılarak hata payı çok aza indirilebilmektedir.
- Frobenius testi Miller-Rabin testinden üç kat daha yavaş çalışır. Fakat hata oranı $\frac{1}{7710}$ olduğu için yavaş olması bir dezavantaj oluşturmaz. Miller-Rabin testi 3 kez çalıştırıldığında hata oranının en fazla $1/4^3 = 1/64$ olur. Bu da Frobenius testinin aynı zaman aralığında Miller-Rabin testinden daha az hata oranı ile çalıştığını göstermektedir.

n sayısının bileşik sayı olduğuna şahitlik eden 1 ile n arasındaki sayılara "şahit sayıları" denir. Olası asallık testlerine göre, şahit sayıların yoğunluğu (d değeri) değişmektedir. Tablo 6.1. de olası asallık testlerinin şahit sayıların yoğunluk değerleri verilmiştir. Daha büyük d değerleri, itimat eşiğine daha hızlı yaklaşma demektir (Segre, 2000).

Tablo 6.1. OAT'de yoğunluk (d) değerleri

Olası Asallık Testleri	**Yoğunluk (d değeri)**
Fermat Testi	0.5
Lehmann Testi	0.5
Solovay-Strassen Testi	0.5
Miller-Rabin Testi	0.75

Tabloya göre, şahit sayılarının yoğunluğu en fazla Miller-Rabin testindedir. Miller-Rabin testinde yoğunluğun daha fazla olması daha az adımda seçilen eşiğe ulaşılabilir olması demektir.

Fermat yalancıları, Euler yalancıları ve güçlü yalancılar arasındaki ilişkiler aşağıda verilmektedir. n, tek bir bileşik tam sayı olsun.

i) a, n için bir Euler yalancısıysa, aynı zamanda Fermat yalancısıdır.

ii) a, n için bir güçlü yalancıysa aynı zamanda Euler yalancısıdır.

Bölüm 5'te $n = 91$ tek bileşik tam sayısı için Fermat yalancıların var olup olmadığı araştırılmış ve bazı Fermat yalancıları bulunmuştu. Aşağıdaki örnekte tüm Fermat, Euler ve güçlü yalancıları bulunmaktadır.

Örnek 6.1. $n = 91$ bileşik tam sayısı için Fermat, Euler ve güçlü yalancıları bulalım.

Fermat yalancılarını bulmak için aşağıdaki adımlar izlenir. n bileşik tam sayısı olmak üzere $a \in \{2, \dots, n-2\}$ için, $a^{n-1} \equiv 1 \bmod n$ ise a, n için Fermat yalancı sayısıdır.
$n = 91$ ve $a \in \{2, \dots, 89\}$ için,

$3^{90} \equiv 1 \bmod 91$ $4^{90} \equiv 1 \bmod 91$ $9^{90} \equiv 1 \bmod 91$
$10^{90} \equiv 1 \bmod 91$ $12^{90} \equiv 1 \bmod 91$ $16^{90} \equiv 1 \bmod 91$
$17^{90} \equiv 1 \bmod 91$ $22^{90} \equiv 1 \bmod 91$ $23^{90} \equiv 1 \bmod 91$
$25^{90} \equiv 1 \bmod 91$ $27^{90} \equiv 1 \bmod 91$ $29^{90} \equiv 1 \bmod 91$
$30^{90} \equiv 1 \bmod 91$ $36^{90} \equiv 1 \bmod 91$ $38^{90} \equiv 1 \bmod 91$
$40^{90} \equiv 1 \bmod 91$ $43^{90} \equiv 1 \bmod 91$ $48^{90} \equiv 1 \bmod 91$
$51^{90} \equiv 1 \bmod 91$ $53^{90} \equiv 1 \bmod 91$ $55^{90} \equiv 1 \bmod 91$
$61^{90} \equiv 1 \bmod 91$ $62^{90} \equiv 1 \bmod 91$ $64^{90} \equiv 1 \bmod 91$
$66^{90} \equiv 1 \bmod 91$ $68^{90} \equiv 1 \bmod 91$ $69^{90} \equiv 1 \bmod 91$
$74^{90} \equiv 1 \bmod 91$ $75^{90} \equiv 1 \bmod 91$ $79^{90} \equiv 1 \bmod 91$
$81^{90} \equiv 1 \bmod 91$ $82^{90} \equiv 1 \bmod 91$ $87^{90} \equiv 1 \bmod 91$
$88^{90} \equiv 1 \bmod 91$

O halde $n = 91$ için $a^{n-1} \equiv 1 \bmod n$ şartını sağlayan Fermat yalancıları; 3, 4, 9, 10, 12, 16, 17, 22, 23, 25, 27, 29, 30, 36, 38, 40, 43, 48, 51, 53, 55, 61, 62, 64, 66, 68, 69, 74, 75, 79, 81, 82, 87, 88 sayılarıdır.

Euler yalancılarını bulmak için aşağıdaki adımlar izlenir. n bileşik tam sayısı olmak üzere $a = \{2, \dots, n-2\}$ için, $a^{(n-1)/2} \equiv \left(\frac{a}{n}\right) \bmod n$ ise a, n için Euler yalancı sayısıdır.
$n = 91$ ve $a \in \{2, \dots, 89\}$ için,

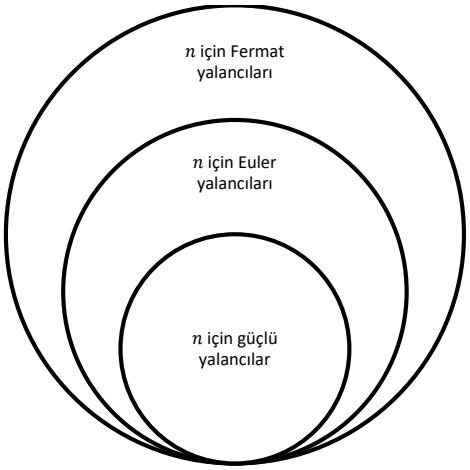
$4^{45} \equiv 1 \bmod 91$ $9^{45} \equiv 1 \bmod 91$ $10^{45} \equiv -1 \bmod 91$
$12^{45} \equiv -1 \bmod 91$ $16^{45} \equiv 1 \bmod 91$ $17^{45} \equiv -1 \bmod 91$
$22^{45} \equiv 1 \bmod 91$ $25^{45} \equiv 1 \bmod 91$ $29^{45} \equiv 1 \bmod 91$
$36^{45} \equiv 1 \bmod 91$ $38^{45} \equiv -1 \bmod 91$ $49^{45} \equiv 1 \bmod 91$
$53^{45} \equiv 1 \bmod 91$ $62^{45} \equiv -1 \bmod 91$ $64^{45} \equiv 1 \bmod 91$
$69^{45} \equiv -1 \bmod 91$ $74^{45} \equiv 1 \bmod 91$ $75^{45} \equiv -1 \bmod 91$
$79^{45} \equiv 1 \bmod 91$ $81^{45} \equiv 1 \bmod 91$ $82^{45} \equiv -1 \bmod 91$

O halde $n = 91$ için Euler yalancıları; 4, 9, 10, 12, 16, 17, 22, 25, 29, 36, 38, 49, 53, 62, 64, 69, 74, 75, 79, 81, 82 sayılarıdır.

Güçlü yalancıları bulmak için aşağıdaki adımlar izlenir. n bileşik tam sayısı olmak üzere $a \in \{2, \dots, n-1\}$ için, $n - 1 = 2^s \cdot r$ olmak üzere, $a^r \equiv 1 \bmod n$ ise a, n için güçlü yalancı sayıdır. $n = 91$ ve $a \in \{2, \dots, 90\}$ için,

$9^{45} \equiv 1 \bmod 91$ $16^{45} \equiv 1 \bmod 91$
$22^{45} \equiv 1 \bmod 91$ $29^{45} \equiv 1 \bmod 91$
$53^{45} \equiv 1 \bmod 91$ $74^{45} \equiv 1 \bmod 91$
$79^{45} \equiv 1 \bmod 91$ $81^{45} \equiv 1 \bmod 91$

O halde $n = 91$ için güçlü yalancılar; 9, 16, 22, 29, 53, 74, 79, 81 sayılarıdır.



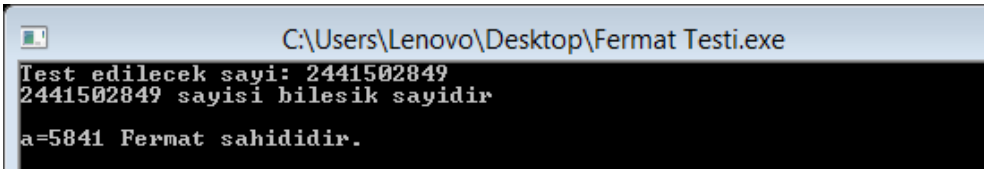
Şekil 6.1. n için yalancı sayıların küme gösterimi

Sonuç olarak, n için güçlü yalancı olan bir sayı aynı zamanda Fermat ve Euler yalancısıdır. Doğru sonuç verme kriterine bakıldığında, yalancı sayı sonucu en az Miller-Rabin testinde bulunmuştur. Bu sonuç Miller-Rabin testinin, Fermat ve Solovay-Strassen testlerinden daha güvenilir sonuç verdiğini göstermektedir.

Olası asallık testlerinin performans analizlerini verelim. Bu testlerin çalışma zamanı, bellek gereksinimi ve yapılan işlem sayısı kriterlerine bakılmaktadır. Bu bölümde olası asallık testlerinden olan Fermat, Solovay-Strassen ve Miller-Rabin olası asallık testlerinin performansları zaman açısından karşılaştırılmıştır. Olası asallık testlerinin çalışma zamanlarını karşılaştırabilmek için kullanılan derleyici 2.40 Ghz işlemcili, 4.00 GB RAM, 64 bit işletim sistemi ve x64 tabanlı işlemci özelliklerine sahip bir bilgisayarda çalıştırılmıştır.

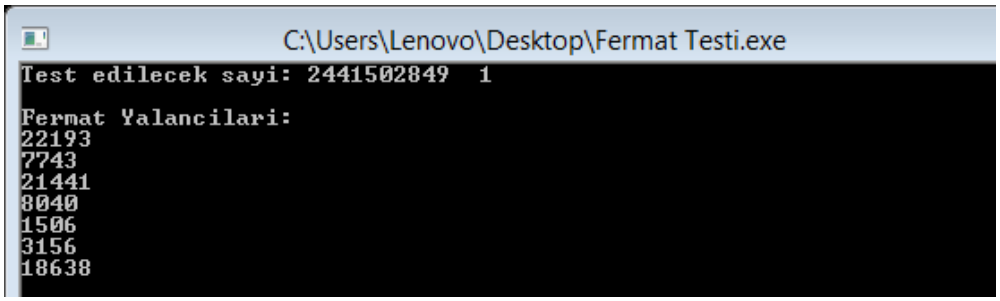
Örnek 6.2. 10 basamaklı $n = 2441502849$ bileşik sayısı için Fermat, Solovay-Strassen ve Miller–Rabin olası asallık testlerini çalıştıralım ve sonuçları karşılaştıralım.

Fermat testi $a = 5841$ taban sayısı için çalıştırıldığı zaman $n = 2441502849$ sayısını 0.0002 saniyede "bileşik sayı" olarak döndürmektedir. Bu durumda bu $a = 5841$ sayısı Fermat şahidi olarak adlandırılır.



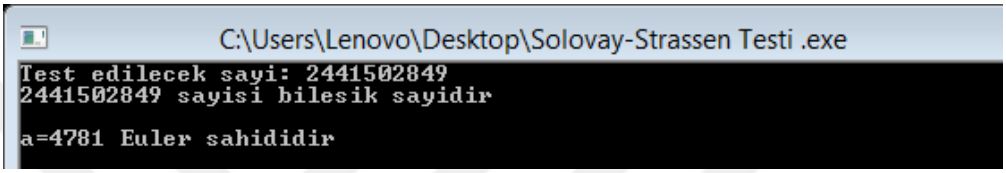
Şekil 6.2. Fermat testi ekran görüntüsü

Diğer taraftan, $n = 2441502849$ sayısı için $t = 7$ tane $a = 22193, 7743, 21441, 8040, 1506, 3156$ ve 18638 sayıları n sayısının olası asal sayı olduğunu söylemektedir. Dolayısıyla, n sayısı Fermat testine göre $E_n(7) = \frac{1}{2^7} = \frac{1}{128}$ hata oranı ile olası asal sayıdır. Gerçekte n sayısı bileşik sayı olduğu için, bu a sayıları Fermat yalancılarıdır.



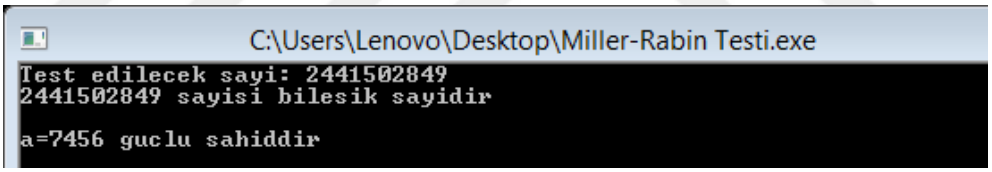
Şekil 6.3. Fermat yalancıları

Solovay-Strassen testi $a = 4781$ taban sayısı için çalıştırıldığı zaman $n = 2441502849$ sayısını 0.0003 saniyede "bileşik sayı" olarak döndürmektedir. Bu durumda bu $a = 4781$ sayısı Euler şahidi olarak adlandırılır.



Şekil 6.4. Solovay-Strassen ekran görüntüsü

Miller-Rabin testi $a = 7456$ taban sayısı ve $r = 19074241$ için çalıştırıldığı zaman $n = 2441502849$ sayısını 0.0002 saniyede "bileşik sayı" olarak döndürmektedir. Bu durumda bu $a = 7456$ sayısı güçlü şahit olarak adlandırılır.



Şekil 6.5. Miller-Rabin ekran görüntüsü

Bunun sonucunda Fermat ve Miller-Rabin testlerinin, Solovay-Strassen testine göre daha hızlı sonuca ulaştığı görülmektedir.

Örnek 6.3. 2-10 aralığında basamak değerine sahip olan Mersenne sayıları için testlerin çalışma süreleri için bir karşılaştırma yapalım. Testlerin "olası asal sayı" ya da "bileşik sayı" sonucunu verme süreleri değişkenlik göstermektedir ama performans açısından etkin olan yine Miller-Rabin testidir.

Fermat testi ile $t = 7$ tane a tabanı için 2-10 aralığında basamak değerine sahip 5 sayı test edilmiştir.

Tablo 6.1. Fermat Testi çalışma süresi

FERMAT TESTİ		
Basamak Sayısı	Mersenne Sayısı	Çalışma Süresi (sn)
2	63	0.00081
4	2047	0.00076
6	262143	0.00072
8	16777215	0.00061
10	1073741823	0.0002

Solovay-Strassen testi ile $t = 7$ tane a tabanı için 2-10 aralığında basamak değerine sahip 5 sayı test edilmiştir.

Tablo 6.2. Solovay-Strassen Testi çalışma süresi

SOLOVAY-STRASSEN TESTİ		
Basamak Sayısı	Mersenne Sayısı	Çalışma Süresi (sn)
2	63	0.00083
4	2047	0.00081
6	262143	0.00074
8	16777215	0.00072
10	1073741823	0.0003

Miller-Rabin testi ile $t = 4$ tane a tabanı için 2-10 aralığında basamak değerine sahip 5 sayı test edilmiştir.

Tablo 6.3. Miller-Rabin Testi çalışma süresi

MİLLER-RABİN TESTİ		
Basamak Sayısı	Mersenne Sayısı	Çalışma Süresi (sn)
2	63	0.00065
4	2047	0.00067
6	262143	0.00071
8	16777215	0.0006
10	1073741823	0.0002

Aşağıda elde edilen değerler kullanılarak oluşturulan bir grafik verilmiştir.

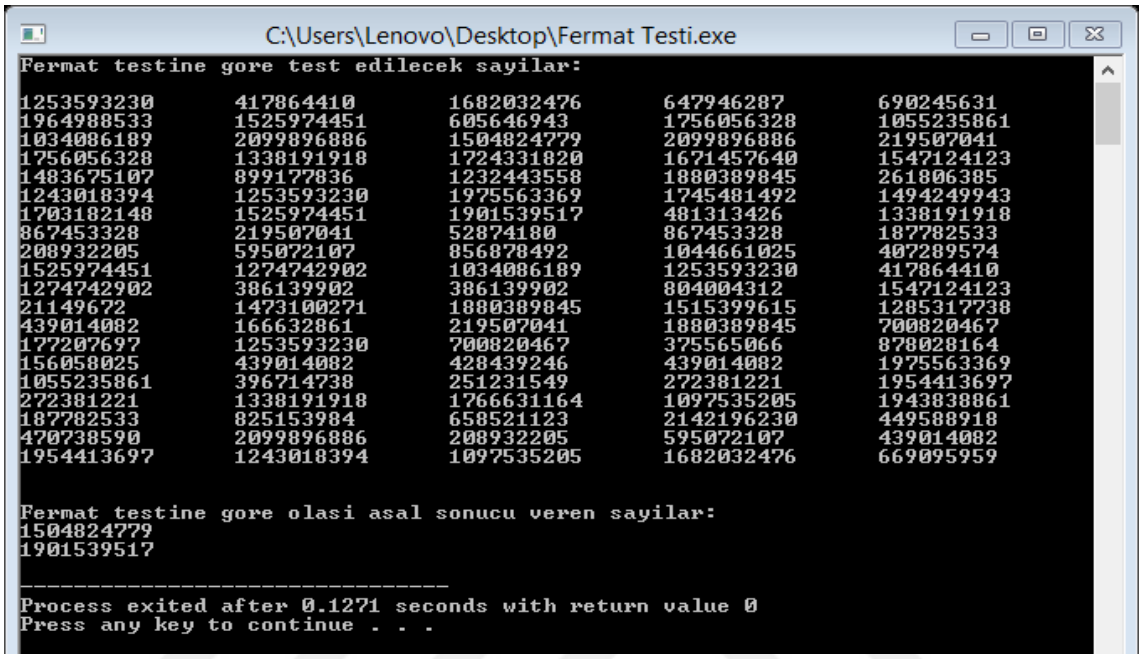
Şekil 6.5. Testlerin çalışma süreleri karşılaştırılması

Yukarıda verilen tablodaki değerler sonucunda olası asallık testlerinden güvenilir sonuca en hızlı ulaşan Miller-Rabin testidir.

Şimdi rastgele üretilen en az 100 adet sayı için bir karşılaştırma sunulacaktır.

Örnek 6.4. Aşağıda 10-15 basamak aralığındaki rastgele üretilen 100 sayı için, Fermat, Solovay-Strassen ve Miller–Rabin olası asallık testleri için bir karşılaştırma verilmektedir.

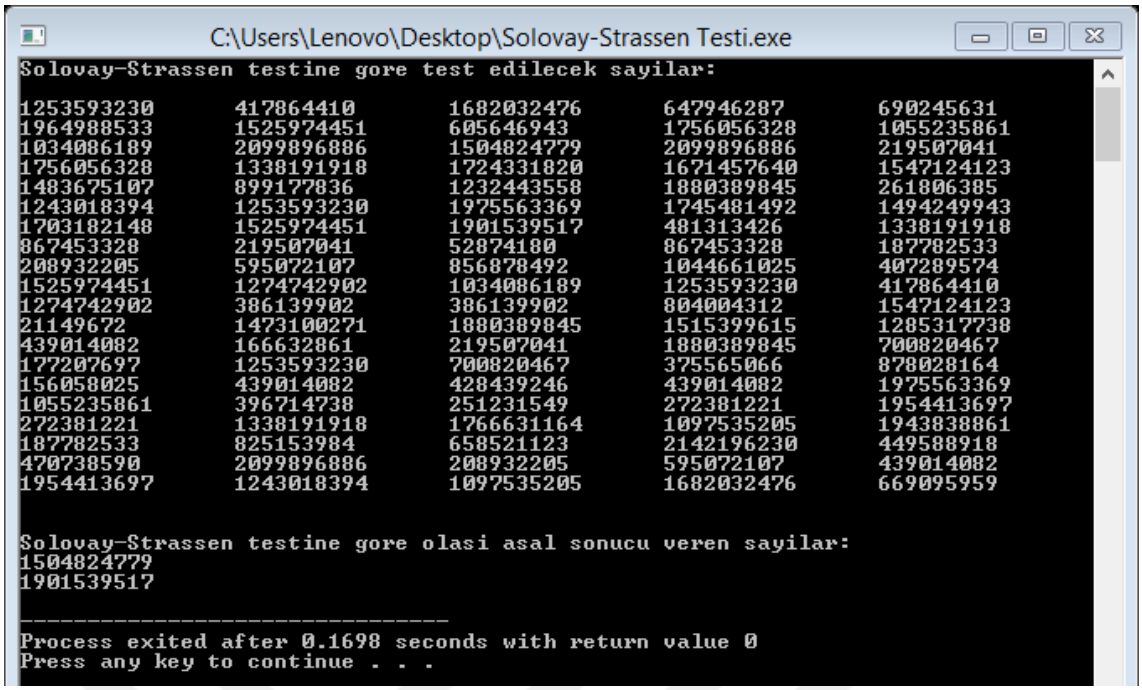
Üretilen sayılar Fermat testi ile test edildiği zaman 2 sayı için olası asal sonucu vermiştir. İşlem için geçen süre yaklaşık olarak 0.12 saniyedir. Çalışma ekranı aşağıda verilmiştir.



Şekil 6.6. Fermat testi ekran görüntüsü

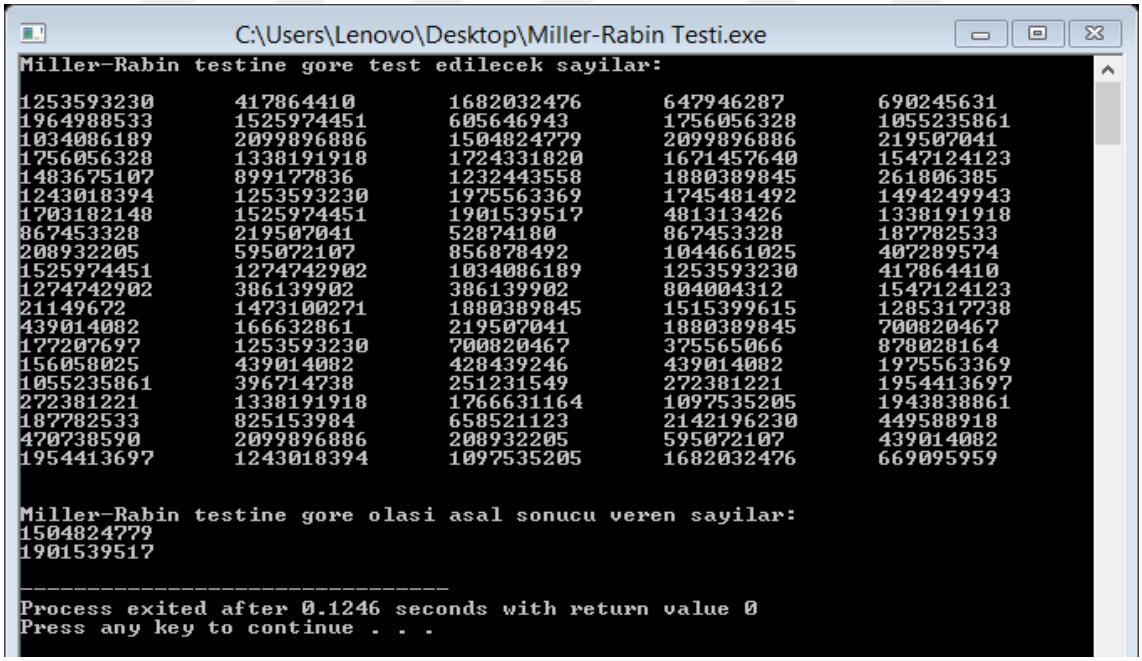
Fermat testine göre olası asal olarak belirlenen 1504824779 ve 1901539517 sayıları gerçekten de asal sayıdır.

Solovay-Strassen testi ile test edidiği zaman aynı 2 sayı için olası asal sonucu vermiştir. İşlem için geçen süre yaklaşık olarak 0.16 saniyedir. Solovay-Strassen testi Jocobi hesabı adımı olduğu için Fermat testinden daha uzun bir sürede sonuca ulaşmıştır. Çalışma ekranı aşağıda verilmiştir.



Şekil 6.7. Solovay-Strassen testi ekran görüntüsü

Miller-Rabin testi ile test edidiği zaman 2 sayı için olası asal sonucu vermiştir. İşlem için geçen süre yaklaşık olarak 0.12 saniyedir. Çalışma ekranı aşağıda verilmiştir.



Şekil 6.8. Miller-Rabin testi ekran görüntüsü

Rastgele üretilen 10-15 basamak aralığında 100-600 arası sayı adedi için Fermat, Solovay-Strassen ve Miller-Rabin testleri çalıştırılıp, çalışma sürelerinden elde edilen veriler Tablo 6.4., Tablo 6.5. ve Tablo 6.6. da verilmiştir.

Tablo 6.4. Fermat Testi 100-600 sayı için çalışma süresi

FERMAT TESTİ	
Sayı Adedi	Çalışma Süresi (sn)
100	0.1271
200	0.1446
300	0.1588
400	0.2015
500	0.2320
600	0.2996

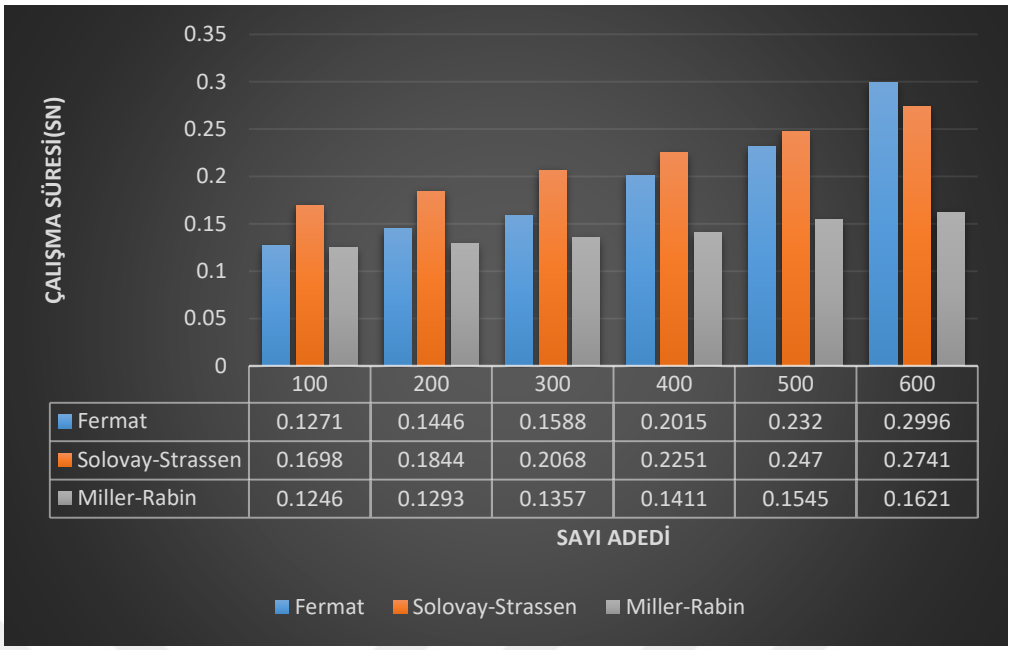
Tablo 6.5. Solovay-Strassen Testi 100-600 sayı için çalışma süresi

SOLOVAY-STRASSEN TESTİ	
Sayı Adedi	Çalışma Süresi (sn)
100	0.1698
200	0.1844
300	0.2068
400	0.2251
500	0.2470
600	0.2741

Tablo 6.6. Miller-Rabin Testi 100-600 sayı için çalışma süresi

MİLLER-RABİN TESTİ	
Sayı Adedi	Çalışma Süresi (sn)
100	0.1246
200	0.1293
300	0.1357
400	0.1411
500	0.1545
600	0.1621

Bu tablolardaki elde edilen veriler kullanılarak karşılaştırmalı bir grafik oluşturulmuş ve aşağıda verilmiştir.



	100	200	300	400	500	600
Fermat	0.1271	0.1446	0.1588	0.2015	0.232	0.2996
Solovay-Strassen	0.1698	0.1844	0.2068	0.2251	0.247	0.2741
Miller-Rabin	0.1246	0.1293	0.1357	0.1411	0.1545	0.1621

Şekil 6.6. Fermat, Miller-Rabin ve Solovay-Strassen testlerinin çalışma süreleri karşılaştırılması

Performans analizi için sayıların çalışma zamanlarına bakıldığında basamak değeri arttıkça çalışma zamanı da artmaktadır. Fermat testi Solovay-Strassen testine göre daha hızlı zamanda, Miller-Rabin testi de Solovay-Strassen testine göre daha hızlı zamanda sonuca ulaşmaktadır. Bunun sonucunda üç temel olası asallık testi çalışma zamanı kriterine göre karşılaştırıldığında Miller-Rabin testi sonuca en hızlı ulaşan ve hata oranı en düşük olan testtir. Hata payını en aza indirmek için Miller-Rabin testi ile birlikte Lucas-Lehmer ve Frobenius testleri birlikte kullanılmalıdır.

Çalışmanın bu bölümünde asallık test algoritmalarının performanslarını ölçmek ve karşılaştırmak için testlerin çalışma zamanları farklı sayı değerleri için incelenmiş ve karşılaştırılmıştır.

Bellek gereksinimi dediğimiz kriter, algoritmanın çalışması için ne kadarlık bir belleğe ihtiyacı olduğunu gösteren değerdir. Program kodun tutulması ve kodun çalışması olmak üzere iki farklı açıdan belleğe ihtiyaç duymaktadır. Olası asallık test yöntemlerinin algoritmalarının çalışması için kodlarının tutulması ayrı bir bellek gerektirir ve kodun çalışıp bir sayının asal olup olmadığının herhangi bir olası asallık testi ile test edilmesi ayrı bir bellek gereksinimidir.

Bir program çalışırken özelliğine göre ne kadar işlem yapıldığına bakılarak işlem sayısı kriteri elde edilir. Olası asallık testlerinde matematiksel dört işlemin yanında üs hesaplama, karekök hesabı, en büyük ortak bölen sayısının hesaplanması gibi farklı işlemler de yapılmaktadır. Tüm bu işlemlerin sayısı her olası asallık testi için ayrı ayrı bulunarak belirlenebilmektedir.

7. SONUÇLAR VE ÖNERİLER

7.1. Sonuçlar

Bu çalışmada kriptolojinin tarihsel gelişiminden başlayarak çalışma da kullanılacak temel matematiksel tanımlara ve esas konumuz olan asal sayıların test yöntemlerinin incelenmesine yer verilmiştir. Büyük sayıların asal olup olmadıklarını bulmak için ihtiyacımız olan kesin asallık testlerinden Lucas-Lehmer ve AKS testleri, olası asallık testlerinden ise Fermat, Slovay-Strassen, Miller-Rabin, Lehmann ve Frobenius testleri algoritmalarıyla birlikte ayrıntılı olarak incelenmiştir. Fermat, Solovay-Strassen ve Miller-Rabin olası asallık testlerinin performans analizleri 10-15 basamaklı rastgele 100 sayı için yapılmış ve çalışma süreleri karşılaştırılmıştır. Elde edilen veriler doğrultusunda Miller-Rabin olası asallık testinin Fermat ve Solovay-Strassen olası asallık testlerine göre hız ve performans kriterlerine bakılarak daha verimli olduğu görülmüştür. Ayrıca, açık anahtarlı şifreleme sistemi olan RSA kriptosistemi incelenmiş ve özellikle bu sistemin şifreleme ve şifre çözme hızlarını arttıran faktörler ve algoritmalar verilmiştir. Son olarak, çalışmada verilen bazı algoritmaların ve olası asallık test yöntemlerinin C++ dilinde kodları sunulmuştur.

7.2. Öneriler

Açık anahtarlı kriptografi'de bazı şifreleme ve imzalama algoritmalarının tasarımında yeteri kadar büyük asal sayılara ihtiyaç duyulmaktadır. Kullanılan asal sayıların büyüklüğüne bağlı olarak (sayılar büyüdükçe) kriptosistemin güvenliği artmaktadır, fakat diğer taraftan kriptosistemin yaptığı işlemin (Örn. şifreleme, şifre çözme, imzalama, vb.) hızı azalmaktadır. Fakat pratik kullanımda bu işlemlerin mümkün olduğunca hızlı olması gerekmektedir. Benzer şekilde, bu sistemlerin tasarımı için gerekli olan asal sayıların elde edilmesi de mümkün olduğunca pratik olması gerekmektedir. Bu yüzden, bu çalışmada olası asallık testlerinin performans analizleri yapılırken testlerin çalışma süreleri incelenmiş ve karşılaştırması yapılmıştır. Elde edilen sonuçlara göre, büyük sayıların asallığının test edilmesinde en hızlı çalışan ve hata oranı diğer olası asallık testlerinden düşük olan Miller-Rabin testi olduğu görülmüştür. Bu çalışmada yapılan performans analizinde, kullanılan sayıların basamak değerleri artırılarak algoritmaların işlem sayısı kriterleri için de bir inceleme yapılırsa, elde edilen sonuçların karşılaştırma analizine dahil edilmesiyle daha etkili bir karşılaştırma yapılabilir.

8. KAYNAKLAR

Agrawal, M., Kayal, N., Saxena, N., (2004), PRIMES is in P, *Annals of Mathematics*, 160(2), 781-793.

Ağçakaya, E. 2020, Asal Sayı Test Algoritmaları ve Kriptolojideki Uygulamaları Üzerine, Yüksek Lisans Tezi, *Van Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı*, Van.

Akdeniz, F. (2015). Olasılık ve İstatistik. Ankara: Akademisyen Kitabevi.

Akyıldız, E., Çalık, Ç., Özarar, M., Tok, Z., Yayla, O., "RSA Kriptosistemi Parametreleri için Güvenlik Testi Yazılımı", ISC Turkey 6. Uluslararası Bilgi Güvenliği ve Kriptoloji Konferansı, 23-24 Mayıs 2013, Ankara 2013, s. 124-126.

Akyıldız, E., Cenk, M., Sınak, A., (2021), Algorithms and Complexity in Cryptography, Kitap Bölümü, 1- 66.

Alizade, L., 2014, Fermat Sayılarının Asal Çarpanlarına Ayrılması ve Kriptoloji Uygulamaları, Yüksek Lisans Tezi, *Ege Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı*, İzmir.

Alptekin Bayam, K. A., & Örs, B. (2010). Differential power analysis resistant hardware implementation of the RSA cryptosystem. *Turkish Journal of Electrical Engineering & Computer Sciences*, **18**(1), 129-140.

Beşkirli, A., Özdemir, D. & Beşkirli, M., 2019, Şifreleme Yöntemleri ve RSA Algoritması Üzerine Bir İnceleme, European Journal of Science and Technology, (Special Issue), 284-291.

Bruen, Aiden A. & Forcinito, Mario A. (2011). Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century. John Wiley & Sons. s. 21. ISBN 978-1-118-03138-4. (3 Ocak 2021)

Ceran, E., Kiraz M.S., Uzunkol, O., (2017), RSA Sifreleme Sistemlerinin Kleptografik Arka Kapıları için Güvenlik ve Karmaşıklık Analizi, *Süleyman Demirel Üniversitesi,* Fen Bilimleri Enstitüsü Dergisi, 21 (2) : 631-643.

Crandal, R., & Pomerance, C. (2005). Prime Numbers A Computational Perspective. New York: Springer.

Çimen, C., Akleylek, S., Akyıldız, E., (2007), Şifrelerin Matematiği:Kriptografi, *Orta Doğu Teknik Üniversitesi*, 1. Baskı, ODTÜ Geliştirme Vakfı Yayıncılık, 5-49.

Daemen, J., Rijmen, V. (2002). The Design of Rijndael.

Eastlake, D., Jones, P. (2001). US Secure Hash Algorithm 1 (SHA1). United States: RFC Editor.

Erdoğan M., Yılmaz G., (2008), Çözümlü Problemlerle Soyut Cebir ve Sayılar Teorisi, 1. Baskı, Beykent Üniversitesi Yayınları, 1-22.

Erhan, M., 1993, RSA Algoritmasını Kullanan Şifreleme/Deşifreleme Yazılımının Tasarımı, Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü*, İstanbul, Türkiye, 83 s.

FIPS 46-3, Data Encryption Standard, Federal Information Processing Standard (FIPS), Publication 46-3, National Bureau of Standards, U.S. Department of Commerce, Washington D.C., October 25, 1999.

Gallier, J., &Quaintance, J. (2019), Notes on Primality Testing And Public Key Cryptography, Part 1: Randomized Algorithms Miller-Rabin and Solovay-Strassen Tests, Philadelphia, USA.

Grantham, J., A Probable Prime Test with High Confidence, 1998, 32-48.

Gül, S., 1997, RSA Tabanlı Halka Açık Anahtarlı Kriptosisteminin Uygulanması, Yüksek Lisans Tezi*, Ortadoğu Teknik Üniversitesi Fen Bilimleri Enstitüsü*, Ankara, 56 s.

Hassanpour, A., 2015, Asal Sayıların Şifreleme Teorisindeki Uygulamaları, Yüksek Lisans Tezi, *Atatürk Üniversitesi Fen Bilimleri Enstitüsü*, Erzurum, Türkiye, 85 s.

Hill, L.S., 1929, Cryptography in an Algebraic Alphabet, The American Mathematical Monthly, Vol.36, No. 6, 306-312.

Higgins, B.C., The Rabin-Miller Probabilistic Primality Test, Some Results on the Number of Non-Witnesses to Compositeness, 2000.

Honaker, G. L. and Caldwell, C. , "Palindromic prime pyramids," J. Recreational Math., 30:3 (1999-2000) 169-176.

Keyman, E., Yıldırım, M. (Der.), (2004), Kriptolojiye Giriş Ders Notları, *ODTÜ, Uygulamalı Matematik Enstitüsü, Kriptografi Bölümü*, Ankara.

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

Koca, N., 2020, Asal Sayıların Tespiti İçin Farklı Methot ve Uygulamaları, Yüksek Lisans Tezi, *Pamukkale Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı,* Denizli.

Koç, Ç.K. (1994). High-Speed RSA Implementation. *Technical report. RSA Laboratories TR201*.

Kodaz, H., (2003), RSA Şifreleme Algoritmasının Uygulaması, *Selçuk Üniversitesi Alaaddin Keykubat Kampüsü Bilgisayar Mühendisliği Bölümü*, Konya.

Külen, F., 2013, Kriptolojide Bazı Şifreleme Yöntemlerinde Cebirsel Yaklaşımlar, Yüksek Lisans Tezi, *Gaziosmanpaşa Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı*, Tokat.

Lenstra, H.W., 1987, Factoring Integers with Elliptic Curves. Annals of Mathematics, 126:649 – 673.

Marques, D., 2014. On Generalized Cullen and WoodallNumbers That areAlsoFibonacci Numbers. Journal of Integer Sequences. 17.

Menezes, A. & Van Oorschot, P., Handbook of Applied Cryptography, CRC Press, 1997, 169-185.

Montgomery, P., Modular multiplication without trial division, Mathematics of Computation, Vol. 44:519–521, 1985.

Nasibov, S., 2015, Kriptoloji Sistemleri ve Uygulamaları Üzerine, Yüksek Lisans Tezi, *Ege Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı*, İzmir.

Nesin, A., 2019, Matematik ve Korku, Nesin Yayıncılık, 132 (3 Ocak 2021)

Özçim, S.S., 2018, Polinom Zamanlı Bir Asallık Algoritması, Yüksek Lisans Tezi, *Ankara Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı*, Ankara.

Paar, C. & Pelzl, J., Understanding Cryptography, A Textbook for Students and Practitioners, Springer–Verlag (2009).

Pomerance, C. (2010). Primality Testing: Variations on A Theme of Lucas. Congressus.

PrimeGrid'in Genelleştirilmiş Fermat Prime Araması" (PDF) , PrimeGrid, (10 Ekim 2021).

Ribenboim, P., *The Little Book Of Bigger Primes*, Springer-Verlag : New York, (2004).

Rivest, R.L., Shamir, A., and Adleman, A. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2):120–126, 1978.

Robinson, R.M., 1954.Mersenne and Fermat Numbers.Proc. Amer. Math. Soc.,5, 842-846.

Rosen, K.H., Handbook Of Discrete And Combinatorial Mathematics, CRC Press, 1999, 288-292.

Rosen, K. H., 1984. Elementary Number Theoryand ItsApplications, Addison-Wesley, publishing Company.

Ruohonen, K., Mathematical Cryptology, 2014.

Schneier, B., Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C (cloth), John Wiley & Sons Inc, 1996.

Segre, A., Computer and Network Security, Iowa Üniversitesi “Data Security” Ders Notları, 2000.

Singh, S., 1998. Fermat's Enigma: TheEpicQuest to Solve the World's Greatest wongMathematical Problem.

Solovay, R. And Strassen, V., A fast monte-carlo test for primality. SIAM journal on Computing, 6(1):84–85, 1977.

Smart, N., P.(2016), Crptography Made Simple, Switzerland, Springer.

Stallings, W., (2003). Cryptography and Network Security, Third Edition. New Jersey.

Takashi, A., 2000. On Sophie Germain primes. Tatra Mt. Math. Publ., 20, 65-73.

Topaloğlu, N., Calp, M.H., Türk, B., (2016), Bilgi Güvenliği Kapsamında Yeni Bir Veri Şifreleme Algoritması Tasarımı ve Gerçekleştirilmesi, *Gazi Üniversitesi*, Bilişim Teknolojileri Dergisi, CİLT: 9, SAYI: 3, Ankara.

Turan, M., Nacar, M.A., (2016), Asal Sayıların Eratosten Kalburu Algoritması Kullanılarak Bulunmasında GPU ve CPU Başarımlarının Analizi, Araştırma Makalesi, *Adıyaman Üniversitesi*, Mühendislik Bilimleri Dergisi, Adıyaman.

Wells, D., "Prime Number: The Most Mysterious Figures in Math", United States Of America, (2005).

Wong, D., Real-World Cryptography, 2021.

Yerlikaya, T., Aslanyürek, C., (2019), RSA Algoritmasının Şifreleme Hızını Arttıran Algoritmalar ve Performansları, Araştırma Makalesi, *Dicle Üniversitesi*, DÜMF Mühendislik Dergisi, 10:3 :853-862, Diyarbakır.

Yerlikaya, T., Kara, O., 2017, Kriptolojide Kullanılan Asal Sayı Test Algoritmaları, Derleme Makale, *Trakya University Journal of Engineering Sciences, 18(1): 85-94*, Edirne.

Yeşilbaş, E., 2016, Cebirsel Kriptoloji Yöntemleri ve Bazı Uygulamaları, Yüksek Lisans Tezi, *Recep Tayyip Erdoğan Üniversitesi Fen Bilimleri Enstitüsü Matematik Anabilim Dalı*, Rize.

Yıltaş, D., 2003, Kriptolojide Kullanılan Asal Sayı Test Algoritmalarının Performans Açısından Karşılaştırılması, Yüksek Lisans Tezi, *İstanbul Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı*, İstanbul.

(URL-1), 2021, http://tarihiolaylar.com/tarihi-olaylar/rosetta-tasi-229 (3 Ocak 2021)

(URL-2), 2021, http://kuranarastirmalarim.com/icerik/arap-alf%C3%A2besinin-sira-ve-sayisal-degerleri-35.aspx (3 Ocak 2021)

(URL-3), 2021, tr.wikipedia.org/wiki/Sezar_sifrelemesi (3 Ocak 2021)

(URL-4), 2021, mathcenter.oxford.emory.edu/site/math125/transpositionCiphers (4 Ocak 2021)

(URL-5), 2021, tr.wikipedia.org/wiki/Türk_alfabesindeki_harflerin_kullanım_sıklıkları (4 Ocak 2021)

(URL-6), 2021, kulturservisi.com/p/600-yillik-voynich-elyazmasi-desifre-edildi (5 Ocak 2021)

(URL-7), 2021, bilimoloji.com/600-yillik-gizemli-el-yazmasi-voynichin-sirrini-yapay-zeka-cozdu (5 Ocak 2021)

(URL-8), 2021, en.wikipedia.org/wiki/Alberti_Cipher_disk (10 Ocak 2021)

(URL-9), 2021, en. wikipedia.org/wiki/Polygraphia_(book) (10 Ocak 2021)

(URL-10), 2021, en. wikipedia.org/wiki/Jefferson_disk (12 Ocak 2021)

(URL-11), 2021, e-bergi.com/y/enigma (15 Ocak 2021)

(URL-12), 2021, tarihiolaylar.com/tarihi-olaylar/eniac-ilk-elektronik-bilgisayar-449 (16 Ocak 2021)

(URL-13), 2021, sibervatan.org/makale/des-sifreleme/27 (18 Ocak 2021)

(URL-14), 2021, tr.wikipedia.org/wiki/Diffie-Hellman-anahtar-degişimi (20 Ocak 2021)

(URL-15), 2021, tr.wikipedia.org/wiki/RSA_(şifreleme_yönetimi) (25 Ocak 2021)

(URL-16), 2021, tr.wikipedia.org/wiki/Uluslararası_Veri_Şifreleme_Algoritması (25 Ocak 2021)

(URL-17), 2021, tr.wikipedia.org/wiki/AES (18 Ocak 2021)

(URL-18), 2021, tr.wikipedia.org/wiki/Eratosten_Kalburu (27 Ocak 2021)

(URL-19), 2021, uekae.bilgem.tubitak.gov.tr (30 Ekim 2021)