PARA PROBLEMİ DİNAMİK PROGRAMLAMA ÇÖZÜMÜ VE DIFFIE-HELLMAN PROTOKOLÜ İLE POLİALFABETİK ASİMETRİK ŞİFRELEME: KRİPTODİNAMİK ŞİFRELEME TABLOSU YAKLAŞIMI

POLYALPHABETIC ASYMMETRIC ENCRYPTION WITH COIN PROBLEM DYNAMIC PROGRAMMING SOLUTION AND DIFFIE-HELLMAN PROTOCOL: A CRYPTO DYNAMIC ENCRYPTION TABLE APPROACH

ANTALYA
İL MİLLİ EĞİTİM MÜDÜRLÜĞÜ

Sıla AVCI[1] **Ömer KINAY**[2] **Enis Kerem ÇAKMAK**[3] **Arzu ÇAĞLAR**[4]

[1,2]Orta Doğu Teknik Üniversitesi, Ankara/Türkiye
[1,2]Middle East Technical University, Ankara/Turkey
[3]Bilkent Üniversitesi, Ankara/Türkiye
[3]Bilkent University, Ankara/Turkey
[4]Antalya Bilim ve Sanat Merkezi, Antalya/Türkiye
[4]Antalya Science and Art Center, Antalya/Turkey

silahunter07@gmail.com omerkinay05@gmail.com eniskeremcakmak@gmail.com arzuycglr@gmail.com
ORCID: 0000-0002-3452-4342 **ORCID:** 0000-0003-4493-6313 **ORCID:** 0000-0002-4104-1023 **ORCID:** 0000-0002-2982-4770

MAKALE BİLGİSİ/ARTICLE INFORMATION

Geliş Tarihi / Date Received
20.01.2024

Kabul Tarihi/Date Accepted
10.03.2026

Yayın Tarihi/Date Published
Temmuz/July 2026

Yayın Sezonu/Pub Date Season
Aralık-Haziran/December-June

ATIF / CITE as

Avcı, S., Kınay, Ö., Çakmak, E.K., Çağlar, A. (2026). "Para Problemi Dinamik Programlama Çözümü ve Diffie-Hellman Protokolü ile Polialfabetik Asimetrik Şifreleme: Kriptodinamik Şifreleme Tablosu Yaklaşımı" / "Polyalphabetic Asymmetric Encryption with Coin Problem Dynamic Programming Solution and Diffie-Hellman Protocol: A Crypto Dynamic Encryption Table Approach". Bilar: Bilim Armonisi Dergisi 9 (1): 7-28 doi: 10.37215/bilar.1422973

https://dergipark.org.tr/tr/pub/bilar

Copyright © Published by Antalya İl Millî Eğitim Müdürlüğü Since 2018, Antalya, 07100 Turkey. All rights reserved.

PARA PROBLEMİ DİNAMİK PROGRAMLAMA ÇÖZÜMÜ VE DIFFIE-HELLMAN PROTOKOLÜ İLE POLİALFABETİK ASİMETRİK ŞİFRELEME: KRİPTO DİNAMİK ŞİFRELEME TABLOSU YAKLAŞIMI

POLYALPHABETIC ASYMMETRIC ENCRYPTION WITH COIN PROBLEM DYNAMIC PROGRAMMING SOLUTION AND DIFFIE-HELLMAN PROTOCOL: A CRYPTO DYNAMIC ENCRYPTION TABLE APPROACH

ANTALYA
İL MİLLİ EĞİTİM MÜDÜRLÜĞÜ

ÖZET

Bilgi güvenliğinin kritik önem taşıdığı günümüz dijital çağında hassas verilerin korunması için sürekli gelişen kriptografik çözümlere ihtiyaç duyulmaktadır. Bu çalışma, klasik bir kombinatorik problem olan Para Problemi'nin dinamik programlama yaklaşımıyla çözümünü kullanarak Kripto Dinamik Şifreleme Tablosu (KDŞT) adında özgün bir şifreleme tablosu önermektedir**.** Geliştirilen bu dinamik tablo, her bir açık metin karakterine birden fazla şifreli karşılık atama yeteneğiyle polialfabetik bir yapı sunarak frekans analizi saldırılarına karşı direnci artırmaktadır. Sistemin asimetrik anahtar değişimi mekanizması için **Diffie-Hellman Anahtar Alışverişi Protokolü** entegre edilmiştir. Bu entegrasyon, taraflar arasında güvenli ve paylaşılan bir gizli anahtarın oluşturulmasını sağlayarak sistemin anahtar yönetimi altyapısını güçlendirmektedir. Elde edilen bulgular, geliştirilen modelin metinleri hatasız bir şekilde şifreleyip deşifre edebildiğini ve bu sayede başarılı sonuçlar elde edildiğini göstermiştir. Bu çalışma, dinamik tablo oluşturma, güçlü anahtar değişimi ve polialfabetik özellikleriyle modern kriptografiye özgün bir katkı sunmakta, aynı zamanda daha küçük asal sayılarla çalışabilme potansiyeliyle geleneksel asimetrik şifrelemedeki bazı performans kısıtlamalarına çözüm arayışlarına yeni bir bakış açısı getirmektedir.

Anahtar Kelimeler: Kriptografi, Asimetrik Şifreleme, Dinamik Programlama, Para Problemi, Diffie-Hellman, Kripto Dinamik Şifreleme Tablosu

ABSTRACT

In today's digital age, where information security is of critical importance, there is a continuous need for evolving cryptographic solutions to protect sensitive data. This study proposes a novel encryption table, named **Crypto Dynamic Encryption Table (CDET)**, developed by applying the dynamic programming approach to solve the classical Coin Problem (also known as the Change-Making Problem). This dynamic table offers a polyalphabetic structure by enabling multiple encrypted representations for each plaintext character, thereby enhancing resistance against frequency analysis attacks. For the asymmetric key exchange mechanism, the **Diffie-Hellman Key Exchange Protocol** has been integrated into the system. This integration ensures the secure establishment of a shared secret key between communicating parties, fortifying the system's key management infrastructure. The findings demonstrate that the proposed model can encrypt and decrypt texts accurately, yielding successful results. This work contributes uniquely to modern cryptography through its dynamic table generation, robust key exchange, and polyalphabetic properties. Furthermore, it offers a new perspective in addressing some performance limitations of traditional asymmetric encryption by demonstrating the potential for operations with smaller prime numbers.

Keywords: Cryptography, Asymmetric Encryption, Dynamic Programming, Coin Problem, Diffie-Hellman, Crypto Dynamic Encryption Table

Bu çalışma 9-10 Aralık 2023 tarihlerinde Bilim Armonisi Uluslararası Gençlik Kongresinde Bildiri olarak sunulmuştur.

1. GİRİŞ

Dijital çağda bilginin gizliliği, bütünlüğü ve erişilebilirliği, bireysel mahremiyetten ulusal güvenliğe kadar geniş bir yelpazede kritik bir öneme sahiptir. Bu bağlamda, kriptografi, bilginin yetkisiz erişimden korunmasında temel bir disiplin olarak öne çıkmaktadır. Kriptografi, açık metinlerin *(M)* şifreleme anahtarları *(K)* kullanılarak şifreli metinlere *(C)* dönüştürülmesi *($E_K(M)=C$)* ve ardından deşifreleme anahtarları *(D)* ile tekrar açık metne çevrilmesi *($D_K(C)=M$)* prensibine dayanır (Stallings 2017). Kripto sistemler temelde, tek bir anahtarın şifreleme ve deşifrelemede kullanıldığı simetrik algoritmalar ile farklı şifreleme *(K_1)* ve deşifreleme *(K_2)* anahtarlarının kullanıldığı *($E_{K1}(M)=C$ ve $D_{K2}(C)=M$)* asimetrik algoritmalar olarak iki ana kategoriye ayrılmaktadır. Özellikle asimetrik şifreleme sistemleri, farklı anahtarlar kullanarak güvenli iletişimi mümkün kılmakta ve anahtar dağıtımı problemine etkin çözümler sunmaktadır (Menezes vd. 1996). Kriptografik algoritmaların güvenliği doğrudan kullanılan anahtarın gizliliğine ve karmaşıklığına bağlıdır.

Bilgi güvenliğinin artan önemi, yeni ve daha dirençli şifreleme algoritmalarının geliştirilmesi ihtiyacını doğurmaktadır. Mevcut şifreleme yaklaşımları, özellikle kuantum bilgisayarların ortaya çıkmasıyla potansiyel güvenlik açıklarına karşı savunmasız kalma riski taşımaktadır (Bernstein vd. 2017). Ayrıca, bazı geleneksel algoritmalar, büyük anahtar boyutları veya karmaşık matematiksel operasyonlar nedeniyle performans darboğazları yaratabilmektedir. Bu durum hem güvenliği artıran hem de hesaplama yükünü optimize eden yenilikçi şifreleme tekniklerine olan talebi artırmaktadır.

1.1. Araştırma Problemi ve Çalışmanın Amacı

Mevcut şifreleme algoritmaları ve polialfabetik yaklaşımlar belirli güvenlik seviyeleri hususunda da genellikle sabit veya önceden tanımlanmış şifreleme tabloları kullanır. Bu statik yapı, uzun vadede veya gelişmiş saldırı senaryolarında sistemin tahmin edilebilirliğini artırma riski taşımaktadır. Özellikle, şifreleme tablosunun oluşturulma sürecinin dinamik olmaması veya anahtarın tablo ile esnek bir şekilde etkileşime girmemesi, şifrelemenin adaptasyon kabiliyetini ve dolayısıyla direncini sınırlayabilir. Bu bağlamda, geleneksel yöntemlerin statik tablo dezavantajlarını ortadan kaldırarak, şifreleme tablosunun oluşturulmasını matematiksel bir problem çözümüne dayandıran, dinamik ve esnek bir yapıya sahip yeni bir şifreleme tekniğine ihtiyaç duyulmaktadır. Ayrıca, bu dinamik tablonun asimetrik bir anahtar değişim protokolü ile entegrasyonu, sistemin genel güvenlik mimarisini güçlendirecektir. Bu doğrultuda, çalışmamızın temel araştırma sorusu 'Para Problemi'nin dinamik programlama çözümü temelli ve Diffie-Hellman entegreli bir asimetrik şifreleme tekniği, geleneksel yöntemlere kıyasla daha dinamik ve güvenli bir alternatif sunabilir mi?' olarak belirlenmiştir.

Bu çalışmanın temel amacı, klasik bir kombinatorik problem olan Para Problemi (Coin Change Problem) ile Dinamik Programlama yaklaşımını birleştirerek özgün bir asimetrik şifreleme tekniği önermektir. Geliştirilen bu yöntemde, para probleminin dinamik programlama ile çözümü sonucunda elde edilen yapı, bir Kripto Dinamik Şifreleme Tablosu (KDŞT) oluşturmak için temel alınmıştır. Tablonun harf frekansları ve modüler aritmetik ile zenginleştirilmesiyle, her bir harfin birden fazla indis ile temsil edilebilmesi sağlanarak Polybius şifresine benzer bir polialfabetik yapı elde edilmiştir. Ayrıca, güvenli anahtar paylaşımını sağlamak amacıyla Diffie-Hellman anahtar

değişim algoritması sisteme entegre edilmiştir. Önerilen bu yaklaşım, şifreleme sürecinde esneklik ve anahtar alanı büyüklüğü açısından potansiyel bir katkı sunmayı hedeflemektedir.

1.2. Literatür Özeti

Para Problemi ve Dinamik Programlama: Kombinatoryal problemlerden biri olan ve sayıların parçalanması problemi olarak da bilinen para problemi, matematik ve bilgisayar bilimleri alanında uzun yıllardır incelenen temel konulardan biridir (Aho vd. 1974). Bu problem, belirli bir toplamın, verilen bir dizi madeni para veya sayı birimiyle kaç farklı şekilde oluşturulabileceğini araştırır. Dinamik Programlama, karmaşık problemleri alt problemlere bölerek ve bu alt problemlerin çözümlerini depolayarak optimum çözümü bulan güçlü bir algoritmik paradigmalar bütünüdür (Cormen vd. 2009). Para probleminin dinamik programlama ile çözümü, özellikle en az sayıda madeni para kullanma veya farklı kombinasyonları sayma gibi optimizasyon problemlerinde yaygın olarak uygulanmaktadır (Skiena 2008). Örneğin, klasik çalışmalardan Nabiyev (2007) ve birçok ders kitabında, dinamik programlamanın bu tür kombinatoryal problemlere nasıl uygulandığı detaylıca açıklanmıştır.

Kombinatorik ve Matematiksel Problemlerin Kriptografide Kullanımı: Kriptografi tarihi boyunca, matematiksel ve kombinatoryal problemler, şifreleme algoritmalarının temelini oluşturmuştur. Sayı teorisi, cebir ve olasılık teorisi, modern kripto sistemlerin (RSA, ElGamal, ECC gibi) ayrılmaz bir parçasıdır (Katz ve Lindell 2014). Örneğin, eliptik eğri kriptografisi, sonlu cisimler üzerindeki matematiksel zorluklara dayanmaktadır. Bazı araştırmalar, çeşitli kombinatoryal yapıları şifreleme veya anahtar üretimi için kullanma potansiyelini araştırmıştır. Topaloğlu vd. (2016) tarafından yapılan bir çalışmada, bilgi güvenliği kapsamında yeni bir veri şifreleme algoritması tasarımı ve gerçekleştirilmesi ele alınmış, bu tür matematiksel yapıların kriptografik uygulamalara entegrasyonuna örnek teşkil etmiştir. Ancak, para problemi veya dinamik programlama tabanlı yapıların doğrudan asimetrik şifreleme anahtarı veya şifreleme tablosu olarak kullanıldığı çalışmalar literatürde sınırlıdır.

Diffie-Hellman Anahtar Değişimi ve Asimetrik Şifreleme: Diffie-Hellman (DH) anahtar değişimi, 1976 yılında Whitfield Diffie ve Martin Hellman tarafından tanıtılan, açık anahtar kriptografisinin öncüsü bir protokoldür (Diffie ve Hellman 1976). Bu algoritma, daha önceden hiçbir gizli bilgi paylaşılmamış olsa dahi, iki tarafın güvenli olmayan bir iletişim kanalı üzerinden ortak bir gizli anahtar oluşturmasına olanak tanır. DH algoritması, ayrık logaritma probleminin hesaplama zorluğuna dayanmaktadır ve birçok modern kriptografik protokolün (örneğin TLS, SSL) temelini oluşturmuştur (Yerlikaya vd. 2006). Algoritmanın temel amacı, paylaşılan anahtarı saldırganlardan korumaktır. Şahin (2018) DH'ın matematiksel prensiplerini ve işleyişini detaylı olarak açıklamaktadır. Ancak, klasik DH algoritması, tek başına kimlik doğrulama sağlamadığı ve "ortadaki adam" saldırılarına (man-in-the-middle attack) karşı hassas olduğu için genellikle daha kapsamlı protokoller içinde kullanılır (Koblitz 1994). Bu çalışmada DH algoritması, önerilen şifreleme tekniği için ortak gizli anahtarın güvenli bir şekilde oluşturulmasında merkezi bir rol oynamaktadır.

2. MATERYAL ve METOT

Bu bölümde, önerilen asimetrik şifreleme tekniğinin geliştirilmesinde kullanılan materyaller ve yöntemsel adımlar detaylandırılmıştır. Sistemin temel bileşenleri olan KDŞT'nun oluşturulması, DH protokolü ile ortak gizli anahtarın belirlenmesi ve şifreleme/deşifreleme sürecinin altyapısı açıklanmaktadır.

2.1. Şifreleme Tablosunun Oluşturulması: Dinamik Programlama ile Para Problemi Yaklaşımı

Önerilen kriptografik sistemin temelini oluşturan KDŞT'nun oluşturulma süreci bu kısımda detaylandırılmıştır. Bu tablo, klasik bir kombinatorik problem olan Para Problemi'nin dinamik programlama yaklaşımıyla çözülmesi sonucunda elde edilen matris yapısının adaptasyonuyla geliştirilmiştir.

Dinamik programlama, karmaşık problemleri daha küçük ve örtüşen alt problemlere bölerek, bu alt problemlerin çözümlerini depolayıp yeniden kullanarak optimal çözüme ulaşmayı sağlayan güçlü bir algoritmik yaklaşımdır (Cormen vd. 2009). Bu prensip, problemin tekrar eden hesaplamalarını ortadan kaldırarak verimlilik sağlar.

Para Problemi, belirli bir toplam hedefine (örneğin, bir para miktarı) ulaşmak için verilen madeni para (veya sayı) birimlerinin kaç farklı kombinasyonunun kullanılabileceğini bulmayı amaçlayan klasik bir problemdir (Aho vd. 1974). Bu çalışmada, Para Problemi'nin dinamik programlama ile çözümü, bir frekans matrisi oluşturmak için kullanılmıştır. Bu matris, her bir para miktarının, belirli madeni para setleri kullanılarak kaç farklı şekilde oluşturulabileceğinin sayısını içerir.

2.1.1. Kripto Dinamik Şifreleme Tablosunun Oluşturulma Algoritması

KDŞT, n farklı madeni para birimi sembolik olarak *coins* = $\{c_1, c_2, c_3, \ldots, c_n\}$ ve maksimum hedef toplam m (örn. alfabenin büyüklüğü veya daha fazlası) kullanılarak oluşturulan bir matristir. Matris, $A_{i,j}$ ile gösterilir. Burada *i* mevcut madeni para birimlerinin sayısını *j* ise hedeflenen para miktarını temsil eder. Matrisin doldurulma prosedürü aşağıdaki gibidir:

1. **Matris Boyutlandırması:**

$n \times (m+1)$ boyutunda bir A matrisi oluşturulur. Yatay eksen (sütunlar), hedef para miktarlarını (*0* ile *m* arasında) temsil ederken, dikey eksen (satırlar), kullanılan madeni para birimlerini temsil eder.

2. **Başlangıç Koşulları:**

İlk satır (*i=0* veya tek kuruşluk madeni para için), her hedef miktar *j* için *1* ile doldurulur. Bu, *1* birimlik paraların herhangi bir miktarı sadece kendisiyle (tek bir yol) oluşturabileceği anlamına gelir:

$$A_{0,j} = 1 \; eğer \;\; 0 \leq j \leq m$$

3. Matrisin Doldurulması (Dinamik Programlama Adımları):

Matris, soldan sağa ve yukarıdan aşağıya doğru doldurulur. Her bir $A_{i,j}$ hücresi, aşağıdaki iki durumun toplamını temsil eder.

Durum 1: Mevcut madeni para (c_i) kullanılmazsa: Bu durumda, çözüm sayısı bir önceki madeni para biriminin kullanıldığı satırdaki aynı hedef miktar için olan değerden gelir. Yani

$$A_{i-1,j}$$

Durum 2: Mevcut madeni para (c_i) kullanılırsa: Bu durumda, hedef miktarın *(j)* mevcut Madeni paradan (c_i *)* çıkarılmasıyla elde edilen miktarın *($j-c_i$)* kalan madeni paralarla kaç farklı şekilde oluşturulabileceği bulunur. Bu, $A_{\mathrm{i,(j-}c_i)}$ değerine eşittir. Bu koşul sadece $j \geq c_i$ olduğunda geçerlidir. Bu iki durumun kombinasyonuyla, dinamik programlama formülü ile hesaplanır: $A_{i,j}$ hücresinin değeri aşağıdaki şekilde hesaplanır.

$$A_{i,j} = \begin{cases} A_{i,j}\,, & \text{eğer j} < \mathrm{c}_i \\ A_{(i-1),j} + A_{(i-1),1-c}\,, & eğer\ j \geq c_i \end{cases}$$

Çizelge1. {1, 5, 10} Kuruşluk Madeni Paralar ile 15 Kuruşun Oluşturulma Kombinasyonları Çizelgesi																
Para(j) / kuruş(i)	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
(i=0) 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
(i=1) 5	1	1	1	1	1	1+1	1+1	1+1	1+1	1+1	1+2	1+2	1+2	1+2	1+2	1+3
(i=2) 10	1	1	1	1	1	2	2	2	2	2	3+1	3+1	3+1	3+1	3+1	4+2

Bu süreç, matrisin tamamı dolana kadar devam eder. Örnek olarak, {1, 5, 10} kuruşluk madeni paralar kullanılarak 15 kuruşun farklı yollarla oluşturulması için bu algoritmanın uygulanması Çizelge 1'de gösterilmiştir.

Sonuç olarak, matrisin her bir hücresi, ilgili madeni para birimleri kullanılarak belirli bir toplamın kaç farklı şekilde oluşturulabileceğini gösteren bir frekans değeri içerir. Bu matris, makalenin ilerleyen bölümlerin de detaylandırılacak olan KDŞT'nun temelini oluşturur. Bu tablo, şifreleme sürecinde her bir karakter için birden fazla sayısal karşılık sunarak, polialfabetik bir şifreleme mekanizması sağlamaktadır.

2.2. Kripto Dinamik Şifreleme Tablosunun Polialfabetik Özellikleri ve Polybius Şifresi ile İlişkisi

Önerilen KDŞT, şifreleme mekanizması açısından bazı açılardan klasik Polybius şifresine benzerlik göstermekle birlikte, temel oluşturma prensibi ve sağladığı kriptografik esneklik bakımından önemli farklılıklar içermektedir.

Polybius şifresi, alfabedeki harfleri belirli bir ızgara (genellikle *5x5* boyutunda) üzerinde konumlandırarak, her harfi bulunduğu satır ve sütun numaralarıyla temsil eden bir yer değiştirme (substitution) şifresidir (Kahn 1967). Geleneksel Polybius şifresi, her harfe tek bir sayısal ikili karşılık atarken, bu çalışmada ele alınan polialfabetik versiyonları veya genişletilmiş versiyonları, alfabenin harf sayısına göre $n \times m$ boyutlarında ızgaralar kullanabilir ve hatta birden fazla kodlamaya izin verebilir (Topaloğlu vd. 2016).

KDŞT'nin Polybius şifresiyle olan benzerlikleri aşağıdaki gibidir:

İkili İndeksleme: Her iki sistemde deşifreleme için alfabedeki bir karakteri temsil etmek üzere sayısal çiftler (indeksler) kullanılır. Polybius'ta satır ve sütun numaraları, KDŞT'de ise Para Problemi matrisinden türetilen indisler kullanılır.

Polialfabetik Potansiyel: Polybius'un genişletilmiş versiyonları gibi, KDŞT de her bir harfin birden fazla farklı sayısal kombinasyonla temsil edilebilmesine olanak tanır. Bu durum, frekans analizi saldırılarına karşı direnci artıran polialfabetik şifreleme prensibini destekler. KDŞT'deki her hücre, belirli bir hedef miktarı (harf değeri) oluşturmanın farklı yollarının sayısını içerdiğinden, bu frekans değerleri şifreleme için farklı indeksler sunabilir.

Ancak, KDŞT'yi klasik Polybius şifresinden ayıran temel farklılıklar şunlardır:

Oluşturulma Mekanizması: Polybius şifresi genellikle statik bir ızgara üzerine harflerin doğrudan yerleştirilmesiyle oluşturulurken, KDŞT'nin temeli, Dinamik Programlama ile Para Problemi'nin matematiksel çözümüne dayanmaktadır. Bu, KDŞT'deki sayısal değerlerin ve bunların dağılımının, kombinatorik hesaplamaların bir sonucu olması anlamına gelir. Bu dinamik yapı, KDŞT'nin manuel olarak oluşturulan statik tablolara göre daha karmaşık ve önceden tahmin edilmesi zor bir yapıya sahip olmasını sağlar.

İçsel Karmaşıklık ve Anahtar Dinamiği: Polybius'ta anahtar, ızgaranın düzeni veya anahtar kelime olabilirken, KDŞT'de anahtar, hem kullanılan madeni para birimleri seti hem de belirlenen maksimum hedef miktarı gibi parametrelere bağlıdır. Ayrıca, bu parametrelerin Diffie-Hellman protokolü ile dinamik olarak belirlenebilmesi, Polybius'a kıyasla çok daha yüksek bir anahtar alanı ve karmaşıklık sunar. KDŞT, basit bir karakter-indeks eşlemesinden ziyade, matematiksel bir fonksiyonun çıktısı olarak üretilen frekans dağılımını kullanır.

Esneklik ve Genişletilebilirlik: KDŞT'nin dinamik yapısı, kullanılan madeni para setleri ve hedef miktarlar değiştirilerek tablonun boyutunun ve içerdiği frekans dağılımlarının kolayca ayarlanabilmesine olanak tanır. Bu, farklı güvenlik gereksinimlerine veya alfabe boyutlarına göre adapte edilebilir bir sistem sunar.

Bu farklılıklar, KDŞT'yi basit bir Polybius benzeri şifreden ziyade, modern kriptografik prensiplere uygun, matematiksel bir temel üzerine inşa edilmiş daha gelişmiş bir şifreleme yaklaşımı olarak konumlandırmaktadır.

2.3. Ortak Gizli Anahtar Oluşturma: Diffie-Hellman Protokolü

Bu alt bölümde, önerilen kripto sistemin güvenli anahtar değişimi için kullanılan DH anahtar değişim protokolü detaylandırılmıştır. Açık anahtar kripto sistemler, simetrik şifrelemede karşılaşılan anahtar dağıtımı problemine çözüm sunmak amacıyla geliştirilmiştir. Bu sistemler, iletişimde bulunan tarafların önceden herhangi bir gizli bilgi, açık kanallar üzerinden gizli anahtarlar oluşturmasına olanak tanır (Diffie ve Hellman 1976). Güvenlikleri, tek yönlü fonksiyonların hesaplama zorluğuna dayanır; bir yönde hesaplama kolayken, ters yönde hesaplama pratik olarak imkânsızdır.

Diffie-Hellman protokolü, ayrık logaritma probleminin zorluğuna dayanır. Protokolün matematiksel temelini oluşturan bazı kavramlar aşağıda tanımlanmıştır:

Tanım 1 (*Euler Φ Fonksiyonu*): Herhangi bir pozitif tamsayı n için *Euler* Φ fonksiyonu $\Phi(n)$, n'den küçük ve n ile aralarında asal olan pozitif tam sayıların sayısını ifade eder. Matematiksel

olarak, $\Phi(n) = \{m \in \mathrm{Z}^{+}: 0 < m < n \; ve \; ebob(m,n) = 1\}$ şeklinde tanımlanır.

Önerme 2.3.1. (Primitif (İlkel) Kök): Bir p asal sayısı için, eğer *g*'nin mertebesi

$\Phi(p) = p - 1$ ise g sayısı $\mathbb{Z}_p^*$ grubunun bir primitif kökü olarak adlandırılır. Yani, $p - 1$'den küçük hiçbir pozitif tam sayı k için $g^k \equiv 1 \; (mod \; p)$ olmamasıdır.

Diffie-Hellman Anahtar Değişim Protokolü Adımları:

Taraflar (Alice ve Bob olarak isimlendirilebilir), ortak bir gizli anahtar oluşturmak için aşağıdaki adımları izlerler:

1. **Ortak Parametrelerin Belirlenmesi:** İletişim kuracak taraflar (örneğin Alice ve Bob), herkes tarafından bilinebilecek büyük bir asal sayı p ve $\mathbb{Z}_p^*$ grubunun bir primitif elemanı $g$üzerinde anlaşırlar. $\mathbb{Z}_p^*$ devirli çarpımsal bir gruptur.

2. **Alice'in Gizli Anahtarı ve Genel Değeri:** Alice, $0 < a < p - 1$ olacak şekilde rastgele bir tamsayı a seçer. Bu a sayısı Alice'in gizli anahtarıdır. Ardından, u değerini hesaplar ve Bob'a gönderir:

$$u = g^a (mod \; p)$$

3. **Bob'un Gizli Anahtarı ve Genel Değeri:** Bob, $0 < b < p - 1$ olacak şekilde rastgele bir tamsayı b seçer. Bu b sayısı Bob'un gizli anahtarıdır. Ardından, v değerini hesaplar ve Alice'e gönderir:

$$v = g^b (mod \; p)$$

4. **Ortak Gizli Anahtarın Hesaplaması (Bob Tarafında):** Bob, Alice'ten aldığı u değerini kendi gizli anahtarı b ile üsleyerek ortak gizli anahtar k'yı hesaplar:

$k = u^b (mod \; p)$ yerine koyma yapıldığında:
$k = (g^a)^b (mod \; p) \equiv g^{ab} (mod \; p)$

5. **Ortak Gizli Anahtarın Hesaplaması (Alice Tarafında):** Alice, Bob'dan aldığı v değerini kendi gizli anahtarı a ile üsleyerek ortak gizli anahtar k'yı hesaplar:

$k = v^a (mod \; p)$ yerine koyma yapıldığında:
$k = (g^a)^b (mod \; p) \equiv (g^b)^a (mod \; p) \equiv g^{ab} (mod \; p)$

Böylece, $u^b (mod \; p) \equiv v^a (mod \; p) \equiv g^{ab} (mod \; p)$ eşitliği sayesinde Alice ve Bob, güvenli olmayan bir kanal üzerinden iletişim kurarak aynı gizli anahtar k 'yı elde etmiş olurlar (Şahin 2018). Bu çalışmada, DH protokolü, KDŞT tabanlı şifreleme sürecinde kullanılacak olan ortak gizli anahtarın güvenli bir şekilde oluşturulması için temel mekanizma olarak entegre edilmiştir. Elde edilen k anahtarı, şifreleme ve deşifreleme algoritmalarındaki belirli parametreleri dinamik olarak belirlemek veya tablo seçiminde kullanılmak üzere değerlendirilebilir.

2.4. Şifreleme ve Deşifreleme Süreci Altyapısı

Bu bölümde, önerilen kripto sistemin temel işlem altyapısı açıklanmaktadır. Şifrelenmemiş, anlaşılabilir mesaja açık metin, bu metnin şifreleme algoritmaları aracılığıyla dönüştürülmüş, anlaşılamayan formuna ise şifreli metin denir. Açık metinden şifreli metni elde etmek için kullanılan kurala şifreleme fonksiyonu (veya "kapama fonksiyonu"), şifreli metinden tekrar açık metni elde etmek için kullanılan kurala ise deşifreleme fonksiyonu (veya "açma fonksiyonu") denir.

Çalışmamızda kullanılacak alfabe ($\sum$), belirli bir doğal ya da yapay dilin karakter setini temsil etmektedir. Bu alfabenin eleman sayısı ($\sum$) ile sembolize edilir ve bu çalışmada ($|\sum| = 23$) harf olarak belirlenmiştir. Dolayısıyla, kullanılan alfabe sonlu bir kümedir. Şifreleme ve deşifreleme süreçleri bu sonlu alfabe üzerinde tanımlanan dönüşümlere dayanmaktadır.

3. BULGULAR

Bu bölümde, önerilen asimetrik şifreleme sisteminin deneysel bulguları ve operasyonel çıktıları sunulmuştur. Materyal ve Metot bölümünde açıklanan algoritmalar ve yöntemler kullanılarak elde edilen KDŞT'nun oluşturulması, DH anahtar değişim protokolünün uygulanması ve şifreleme/deşifreleme süreçleri somut örnekler üzerinden gösterilmektedir.

3.1. Kripto Dinamik Şifreleme Tablosu Oluşturulması ve Özellikleri

Daha önce Materyal ve Metot bölümünde açıklanan dinamik programlama algoritması kullanılarak, belirli bir alfabe boyutu için para problemi çözümü tablosu oluşturulmuştur. Bu çalışmada alfabemizin *23* harften oluşması sebebiyle, hedef miktar olarak maksimum *23*'e kadar olan sayılar için kombinasyonlar hesaplanmıştır.

İlk olarak, kullanılan para birimleri setini belirtin, örneğin $\{1, 5, 10, \dots\}$ madeni para birimleri ile *23*'e kadar olan miktarlar için dinamik programlama matrisi hesaplanmıştır. Bu hesaplamaların bir kısmı Çizelge 2'de gösterilmiştir.

Çizelge 2. 23 Birimlik Hedef Miktar için Dinamik Programlama Matrisi Hesaplama Örneği

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1+1	1+1	1+2	1+2	1+3	1+3	1+4	1+4	1+5	1+5	1+6	1+6	1+7	1+7	1+8	1+8	1+9	1+9	1+10	1+10	1+11	1+11
3	1	1	2	2+1	3+1	3+2	4+3	4+4	5+5	5+7	6+8	6+10	7+12	7+14	8+16	8+19	9+21	9+24	10+27	10+30	11+33	11+37	12+40	12+44

Bu kısmi hesaplamaların tamamlanmasıyla, *23* sayısının dinamik algoritma kullanılarak çözümünü içeren Çizelge 3 elde edilmiştir:

Çizelge 3. 23 Sayısının Dinamik Algoritma Kullanılarak Çözüm Çizelgesi

i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12
3	1	1	2	3	4	5	7	8	10	12	14	16	19	21	24	27	30	33	37	40	44	48	52	56
4	1	1	2	3	5	6	9	11	15	18	23	27	34	39	47	54	64	72	84	94	108	120	136	150
5	1	1	2	3	5	7	10	13	18	23	30	37	47	57	70	84	101	119	141	164	192	221	255	291
6	1	1	2	3	5	7	11	14	20	26	35	44	58	71	90	110	136	163	199	235	282	331	391	454
7	1	1	2	3	5	7	11	15	21	28	38	49	65	82	105	131	164	201	248	300	364	436	522	618
8	1	1	2	3	5	7	11	15	22	29	40	52	70	89	116	146	186	230	288	352	434	525	638	764
9	1	1	2	3	5	7	11	15	22	30	41	54	73	94	123	157	201	252	318	393	488	598	732	887
10	1	1	2	3	5	7	11	15	22	30	42	55	75	97	128	164	212	267	340	423	530	653	807	984
11	1	1	2	3	5	7	11	15	22	30	42	56	76	99	131	169	219	278	355	445	560	695	863	1060
12	1	1	2	3	5	7	11	15	22	30	42	56	77	100	133	172	224	285	366	460	582	725	905	1116
13	1	1	2	3	5	7	11	15	22	30	42	56	77	101	134	174	227	290	373	471	597	747	935	1158
14	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	175	229	293	378	478	608	762	957	1188
15	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	230	295	381	483	615	773	972	1210
16	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	296	383	486	620	780	983	1225
17	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	384	488	623	785	990	1236
18	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	489	625	788	995	1243
19	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	490	626	790	998	1248
20	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	490	627	791	1000	1251
21	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	490	627	792	1001	1253
22	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	490	627	792	1002	1254
23	1	1	2	3	5	7	11	15	22	30	42	56	77	101	135	176	231	297	385	490	627	792	1002	1255

Elde edilen bu çözüm çizelgesine (Çizelge 3), şifreleme tablosu olarak kullanılabilmesi için modüler aritmetik işlemi uygulanmıştır. Bu çalışmada, alfabemizin *23* harf olmasından dolayı *(mod 23)* işlemi uygulanmıştır. Bu dönüşüm sonucu elde edilen tablo Çizelge 4'te sunulmaktadır:

Çizelge 4. Çizelge 3'e (mod 23) Uygulanması Sonucu Elde Edilen Değerler																							
i/j	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12
3	1	1	2	3	4	5	7	8	10	12	14	16	19	21	1	4	7	10	14	17	21	2	6	10
4	1	1	2	3	5	6	9	11	15	18	0	4	11	16	1	8	18	3	15	2	16	5	21	12
5	1	1	2	3	5	7	10	13	18	0	7	14	1	11	1	15	9	4	3	3	8	14	2	15
6	1	1	2	3	5	7	11	14	20	3	12	21	12	2	21	18	21	2	15	5	6	9	0	17
7	1	1	2	3	5	7	11	15	21	5	15	3	19	13	13	16	3	17	18	1	19	22	16	20
8	1	1	2	3	5	7	11	15	22	6	17	6	1	20	1	8	2	0	12	7	20	19	17	5
9	1	1	2	3	5	7	11	15	22	7	18	8	4	2	8	19	17	22	19	2	5	0	19	13
10	1	1	2	3	5	7	11	15	22	7	19	9	6	5	13	3	5	14	18	9	1	9	2	18
11	1	1	2	3	5	7	11	15	22	7	19	10	7	7	16	8	12	2	10	8	8	5	12	2
12	1	1	2	3	5	7	11	15	22	7	19	10	8	8	18	11	17	9	21	0	7	12	8	12
13	1	1	2	3	5	7	11	15	22	7	19	10	8	9	19	13	20	14	5	11	22	11	15	8
14	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	14	22	17	10	18	10	3	14	15
15	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	0	19	13	0	17	14	6	14
16	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	20	15	3	22	21	17	6
17	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	16	5	2	3	1	17
18	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	17	6	4	6	6	1
19	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	17	7	5	8	9	6
20	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	17	7	6	9	11	9
21	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	17	7	6	10	12	11
22	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	17	7	6	10	13	12
23	1	1	2	3	5	7	11	15	22	7	19	10	8	9	20	15	1	21	17	7	6	10	13	13

Şifreleme sürecinde kullanılacak alfabedeki harflerin sayısal karşılıkları, geleneksel eşleşmelerden farklı olarak Çizelge 5'te belirtildiği gibi belirlenmiştir. Bu eşleşme, anahtarın bir parçası olarak düşünülebilir.

Çizelge 5. Harflerin Sayı Karşılıkları																					
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	T	U	V	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	0

Son olarak, Çizelge 4'te elde edilen modüler değerlerin, Çizelge 5'te belirlenen harf karşılıkları ile eşleştirilmesi sonucunda KDŞT oluşturulmuştur. Bu tablo, her bir harfin farklı sayısal indis kombinasyonları ile temsil edilmesini sağlayarak polialfabetik bir yapı sunar ve frekans analizi saldırılarına karşı direnci artırır. KDŞT, Çizelge 6'da gösterilmiştir:

Çizelge 6. Dinamik Programlama ile Para Problemi Çözümü Yaklaşımından Oluşturulan Kripto Dinamik Şifreleme Tablosu (KDŞT)

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

Şifreli metin, KDŞT'den alınan indis çiftlerinin ardışık birleşimi biçimindedir: $i_1 j_1 i_2 j_2 i_3 j_3$ ... Burada *i* satır indisini, *j* sütun indisini ifade eder. İndislerin tek haneli olması durumunda, her bir indisin başına "0" ekleyerek dört haneli bir yapı elde edilir (örn., $i < 10$ ise *i* indisi "0_i" şeklinde düzenlenir*;* $j < 10$ ise *j* indisi "0_j" şeklinde düzenlenir). Bu, her bir şifreli karakter için dört basamaklı bir blok *(iijj*) oluşturur. Bu biçimlendirme, şifreli metnin deşifreleşirken ikili gruplara ayrılmasını kolaylaştırır ve klasik Polybius şifrelemesindeki yer değiştirme yöntemine benzer şekilde indislerin kullanımını sağlar.

3.2. Ortak Gizli Anahtarın Oluşturulması ve Kullanımı

Kapalı anahtarın güvenli bir şekilde oluşturulması için Materyal ve Metot bölümünde detaylandırılan DH Anahtar Alışverişi Protokolü uygulanmıştır. Bu protokol, iletişim kuran tarafların açık bir kanal üzerinden ortak bir gizli anahtar üzerinde anlaşmasını sağlar. Bu çalışmada, alfabemizin büyüklüğü ($|\sum| = 23$) harf olması sebebiyle, DH protokolünde kullanılacak asal sayı *p=23* olarak belirlenmiştir. İlk adım olarak, ($p = 23$)'ün bir primitif kökü (ilkel kök) araştırılmıştır. Bir sayının ilkel köklerini bulma süreci Çizelge 7'de gösterilmiştir:

Çizelge 7. Asal Sayı p=23 için İlkel Kök Arama Çizelgesi

2^x(mod 23)	3^x(mod 23)	5^x(mod 23)
2^1≡2(mod 23)	3^1≡3(mod 23)	5^1=5(mod 23)
2^2≡4(mod 23)	3^2≡9(mod 23)	5^2=2(mod 23)
2^3≡8(mod 23)	3^3≡4(mod 23)	5^3=10(mod 23)
2^4≡16(mod 23)	3^4≡12(mod 23)	5^4=4(mod 23)
2^5≡9(mod 23)	3^5≡13(mod 23)	5^5=20(mod 23)
2^6≡18(mod 23)	3^6≡16(mod 23)	5^6=8(mod 23)
2^7≡13(mod 23)	3^7≡2(mod 23)	5^7=17(mod 23)
2^8≡3(mod 23)	3^8≡6(mod 23)	5^8=16(mod 23)
2^9≡6(mod 23)	3^9≡18(mod 23)	5^9=11(mod 23)
2^{10}≡12(mod 23)	3^{10}≡8(mod 23)	5^{10}=9(mod 23)
2^{11}≡1(mod 23)	3^{11}≡1(mod 23)	^{11}=22(mod 23)
		5^{12}=18(mod 23)
		5^{13}=21(mod 23)
		5^{14}=13(mod 23)
		5^{15}=19(mod 23)
		5^{16}=3(mod 23)
		5^{17}=15(mod 23)
		5^{18}=6(mod 23)
		5^{19}=7(mod 23)
		5^{20}=12(mod 23)
		5^{21}=14(mod 23)
		5^{22}=1(mod 23)

Yapılan hesaplamalar sonucunda, 2 ve 3 ilkel kök değildir ama 5 bir ilkel köktür. $\Phi\big(\Phi(23)\big) = \Phi(22) = 10$ adet ilkel kök bulunmaktadır.
Bunlar; $1 \leq k < 22\ ve\ ebob(k, 22) = 1$ olmak üzere *k* değerleri için $g^k(mod\ p)$ hesaplanarak bulunur.
$p = 23$ için ilkel kökler: *5, 7, 10, 11, 14, 15, 17, 19, 20, 21* olarak belirlenmiştir.
Diffie-Hellman protokolünün uygulanmasına yönelik bir örnek aşağıda verilmiştir:
Belirlenen Ortak Parametreler: Asal sayı $p = 23$ ve ilkel kök $g = 7$ olarak taraflarca kabul edilmiştir. Bu parametreler açık kanalda paylaşılabilir veya önceden güvenli bir şekilde kararlaştırılabilir.

Alice'in İşlemleri: Alice, $1 < a < 21$ koşulunu sağlayan gizli anahtarı $a = 5$ olarak seçer. Daha sonra, *u* değerini hesaplar ve Bob'a gönderir:

$$u = 7^5(mod\ 23) \equiv 16807(mod\ 23) \equiv 17(mod\ 23)$$

Bob'un İşlemleri: Bob, $1 < b < 21$ koşulunu sağlayan gizli anahtarı $b = 8$ olarak seçer. Daha sonra, v değerini hesaplar ve Alice'e gönderir:

$$v = 7^8(mod\ 23) \equiv 5764801(mod\ 23) \equiv 12(mod\ 23)$$

Ortak Gizli Anahtarın Hesaplaması (Bob Tarafında): Bob, Alice'ten aldığı $u = 17$ değerini kendi gizli anahtarı $b = 8$ ile kullanarak ortak anahtar k'yı hesaplar:

$$k = u^b(mod\ 23) \equiv 17^8(mod\ 23) \equiv 20(mod\ 23)$$

Ortak Gizli Anahtarın Hesaplaması (Alice Tarafında): Alice, Bob'dan aldığı $v = 12$ değerini kendi gizli anahtarı $a = 5$ ile kullanarak ortak anahtar k'yı hesaplar:

$$k = v^a(mod\ 23) \equiv 12^5(mod\ 23) \equiv 20(mod\ 23)$$

Her iki taraf da aynı ortak gizli anahtar $k = 20$ değerini elde etmişlerdir. Bu k anahtarı, yalnızca Alice ve Bob tarafından bilinmektedir. Eğer hesaplama sonucunda $k = 0$ veya $k = 1$ gibi zayıf bir anahtar elde edilirse, protokol gereği farklı $a\ ve\ b$ değerleri seçilerek anahtar değişimi tekrarlanır.

Elde edilen k anahtarı, şifreli metin harflerini belirli bir dönüşümden geçirmek için kullanılmıştır. Bu dönüşüm, her bir harfin sayısal karşılığının k anahtarı ile modüler çarpımı şeklinde gerçekleştirilir:

$$(Şifrelenmiş\ Sayı) = k \times (Harfin\ Sayı\ Karşılığı)(mod\ 23)$$

Bu dönüşüm, Çizelge 5'teki harf karşılıkları kullanılarak aşağıdaki gibi hesaplanmıştır:

(A) $20 \times 1 \equiv 20(mod\ 23) \Rightarrow 1 \rightarrow 20$

(B) $20 \times 2 \equiv 17(mod\ 23) \Rightarrow 2 \rightarrow 17$

(C) $20 \times 3 \equiv 14(mod\ 23) \Rightarrow 3 \rightarrow 14$

(D) $20 \times 4 \equiv 11(mod 23) \Rightarrow 4 \rightarrow 11$

(E) $20 \times 5 \equiv 8(mod\ 23) \Rightarrow 5 \rightarrow 8$

(F) $20 \times 6 \equiv 5(mod\ 23) \Rightarrow 6 \rightarrow 5$

(G) $20 \times 7 \equiv 2(mod\ 23) \Rightarrow 7 \rightarrow 2$

(H) $20 \times 8 \equiv 22(mod\ 23) \Rightarrow 8 \rightarrow 22$

(İ) $20 \times 9 \equiv 19(mod\ 23) \Rightarrow 9 \rightarrow 19$

(J) $20 \times 10 \equiv 16(mod\ 23) \Rightarrow 10 \rightarrow 16$

(K) $20 \times 11 \equiv 13(mod 23) \Rightarrow 11 \rightarrow 13$

(L) $20 \times 12 \equiv 10(mod\ 23) \Rightarrow 12 \rightarrow 10$

(M) $20 \times 13 \equiv 7(mod\ 23) \Rightarrow 13 \rightarrow 7$

(N) $20 \times 14 \equiv 4(mod\ 23) \Rightarrow 14 \rightarrow 4$

(O) $20 \times 15 \equiv 1(mod\ 23) \Rightarrow 15 \rightarrow 1$

(P) $20 \times 16 \equiv 21(mod\ 23) \Rightarrow 16 \rightarrow 21$

(R) $20 \times 17 \equiv 18(mod\ 23) \Rightarrow 17 \rightarrow 18$

(S) $20 \times 18 \equiv 15(mod\ 23) \Rightarrow 18 \rightarrow 15$

(T) $20 \times 19 \equiv 12(mod\ 23) \Rightarrow 19 \rightarrow 12$

(U) $20 \times 20 \equiv 9(mod\ 23) \Rightarrow 20 \rightarrow 9$

(V) $20 \times 21 \equiv 6(mod\ 23) \Rightarrow 21 \rightarrow 6$

(Y) $20 \times 22 \equiv 3(mod\ 23) \Rightarrow 22 \rightarrow 3$

(Z) $20 \times 23 \equiv 0(mod\ 23) \Rightarrow 23 \rightarrow 0$

3.3. Şifreleme ve Deşifreleme Sürecinin Uygulanması

Bu bölümde, belirlenen KDŞT ve ortak gizli anahtar k kullanılarak şifreleme ve deşifreleme süreçleri adım adım bir örnek üzerinden gösterilmiştir.

Şifreleme Süreci:

Alice'in şifrelemek istediği açık metnin "*IYI OKUMALAR*" olduğunu varsayalım.

1. **Ortak Anahtar Belirleme:** Alice ve Bob, daha önce detaylandırılan Diffie-Hellman Anahtar Alışverişi Protokolü'ne göre ortak gizli anahtar $k = 20$ üzerinde anlaşırlar.

2. **Harflerin Sayısal Dönüşümü (k Anahtarına Göre):** Alice, açık metindeki her harfi, Çizelge 5'teki orijinal sayı karşılıklarını ve ortak gizli anahtar $k = 20'$yi kullanarak 3.2 bölümünde hesaplanan dönüşüme göre yeni sayısal değerlere çevirir:

$$I\ (9)\ \rightarrow 19$$

$$Y\ (22)\ \rightarrow 3$$

$$I\ (9)\ \rightarrow 19$$

$$O\ (15)\ \rightarrow 1$$

$$K(11) \rightarrow 13$$

$$U\ (20)\ \rightarrow 9$$

$$M\ (13)\ \rightarrow 7$$

$$A\ (1)\ \rightarrow 20$$

$$L\ (12)\ \rightarrow 10$$

$$A\ (1)\ \rightarrow 20$$

$$R\ (17)\ \rightarrow 18$$

Bu dönüşüm sonucunda "*IYIOKUMALAR*" metni, "*19 3 19 1 13 9 7 20 10 20 18*" haline gelir. Bu sayısal dizi, eğer Çizelge 5'teki harf karşılıklarıyla yeniden eşleştirilirse "*TCTAMIGUJUS*" şeklinde okunabilir bir ara metin elde edilir.

KDŞT Kullanarak İndis Belirleme: Alice, bu yeni sayısal değerler için Çizelge 6'da (KDŞT) karşılık gelen indis çiftlerini *(j,i)* seçer. KDŞT, her bir sayısal değere (hedef miktara) karşılık gelen birden fazla indis çifti sunabildiğinden, şifreleyici harfin frekansına bağlı olarak farklı

indis dizileri seçebilir. Bu, aynı harfin metin içinde farklı şifreli gösterimlerle yer almasını sağlayarak frekans analizi saldırılarını zorlaştırır.

"T" (19. harf) için: 1207 ve 1011 indisleri kullanılmıştır. "U" (9. harf) için: 2008 ve 1613 indisleri kullanılmıştır. Örnek metin için belirlenen indisler (her bir indis tek haneli ise "*0*" eklenerek 4 haneli formatta gösterilir): "*120703031011230107052320192220081814161318 07*" Bu indis dizisi, Bob'a gönderilen şifreli metni oluşturur. Bu sürecin temel amacı, çizelgede her harfin birden fazla frekans değeri taşıması sayesinde, metin içerisinde geçen aynı harflerin farklı indislerle

şifrelenmesini sağlamaktır. Örneğin, "I" harfi için "1207" ve "1011" indisleri kullanılmıştır. Bu durum, şifrenin kırılmasını daha zor hale getirir.

Deşifreleme Süreci:

Bob, Alice'ten aldığı şifreli metni:

"*120703031011230107052320192220081814161318 07*"

deşifre etmek için aşağıdaki adımları izler:

1. **Dörtlü Gruplara Ayırma:** Bob, şifreli metni dörtlü gruplara ayırır: "*1207, 0303, 1011, 2301, 0705, 2320, 1922, 2008, 1814, 1613, 1807*"
2. **KDŞT Kullanarak Sayısal Değerlere Dönüştürme:** Her bir dörtlü indis grubunu Çizelge 6 (KDŞT) üzerinde bularak karşılık gelen sayısal değeri (hedef miktar) elde eder:

$$(12,07) \rightarrow 19$$

$$(03,03) \rightarrow 3$$

$$(10,11) \rightarrow 19$$

$$(23,01) \rightarrow 1$$

$$(07,05) \rightarrow 13$$

$$(23,20) \rightarrow 9$$

$$(19,22) \rightarrow 7$$

$$(20,08) \rightarrow 20$$

$$(18,14) \rightarrow 10$$

$$(16,13) \rightarrow 20$$

$$(18,07) \rightarrow 18$$

Bu adım sonucunda

"*19 3 19 1 13 9 7 20 10 20 18*"

sayısal dizisi elde edilir.

3. **Ortak Anahtar k ile Ters Dönüşüm:** Bob, ortak anahtar $k = 20$'yi ve Çizelge 5'i kullanarak her bir sayısal değeri orijinal harf karşılığına dönüştürür. Bu ise;

$k - 1 \times (Şifrelenmiş\ Sayı)(mod23)$ işlemiyle gerçekleştirilir. $k = 20$ için modüler tersi olan

$k-1(mod\ 23)$'ü bulmamız gerekmektedir. Yani, $20x \equiv 1(mod\ 23)$ denklemini sağlayan x değerini arıyoruz. Bu *x* değeri, $20-1(mod23)$ olarak ifade edilir.

Öklid Algoritması veya deneme yanılma yöntemiyle bu değer hesaplanabilir:

- $20 \times 1 \equiv 20(mod\ 23)$
- $20 \times 2 \equiv 40 \equiv 17(mod\ 23)$
- $20 \times 3 \equiv 60 \equiv 14(mod\ 23)$
- …
- $20 \times 15 \equiv 300\ (mod23).\ 300 = 13 \times 23 + 1.\ Yani\ 300 \equiv 1(mod\ 23).$

Bu durumda, $k-1=15$ olarak bulunur.

Şimdi, şifrelenmiş sayıları orijinal harf karşılıklarına dönüştürelim:

Örnek 1: Şifrelenmiş sayı *19*'un ters dönüşümü

- $19 \rightarrow 15 \times 19(mod\ 23)$
- $15 \times 19 = 285$
- $285(mod\ 23) \equiv 9(mod\ 23)$
- Çizelge 5'e göre, *9* değeri "*I*" harfine karşılık gelmektedir. (Bu, metinde belirtilen "*19*'un *I (9)* olması gerekiyordu" beklentimizle tutarlıdır.)

Örnek 2: Şifrelenmiş sayı *20*'nin ters dönüşümü

- $20 \rightarrow 15 \times 20(mod\ 23)$
- $15 \times 20 = 300$
- $300(mod\ 23) \equiv 1(mod\ 23)$
- Çizelge 5'e göre, 1 değeri "*A*" harfine karşılık gelmektedir.

4. **Orijinal harflere dönüştürme:** Şifrelenmiş sayıların orijinal harf karşılıklarına dönüştürülmesi için, $k = 20\ ve\ p = 23$ için k'nin modüler tersi olan $k-1 \equiv 15(mod\ 23)$ kullanılır.

$$(Yani\ 20 \times 15 \equiv 300 \equiv 1(mod\ 23)).$$

$$19 \rightarrow 15 \times 19(mod\ 23) \equiv 285(mod\ 23) \equiv 9(mod\ 23) \rightarrow I$$

$$3 \rightarrow 15 \times 3(mod\ 23) \equiv 45(mod\ 23) \equiv 22(mod\ 23) \rightarrow Y$$

$$19 \rightarrow 15 \times 19(mod\ 23) \equiv 9(mod\ 23) \rightarrow I$$

$$1 \rightarrow 15 \times 1(mod\ 23) \equiv 15(mod\ 23) \rightarrow O$$

$$13 \rightarrow 15 \times 13(mod23) \equiv 195(mod23) \equiv 11(mod23) \rightarrow K$$

$$9 \rightarrow 15 \times 9(mod\ 23) \equiv 135(mod\ 23) \equiv 20(mod\ 23) \rightarrow U$$

$$7 \rightarrow 15 \times 7(mod\ 23) \equiv 105(mod\ 23) \equiv 13(mod\ 23) \rightarrow M$$

$$20 \rightarrow 15 \times 20(mod\ 23) \equiv 300(mod\ 23) \equiv 1(mod\ 23) \rightarrow A$$

$$10 \rightarrow 15 \times 10(mod\ 23) \equiv 150(mod\ 23) \equiv 12(mod\ 23) \rightarrow L$$

$$20 \rightarrow 15 \times 20(mod\ 23) \equiv 1(mod\ 23) \rightarrow A$$

$$18 \rightarrow 15 \times 18(mod\ 23) \equiv 270(mod\ 23) \equiv 17(mod\ 23) \rightarrow R$$

Bu işlemler sonucunda, şifreli metin "*IYI OKUMALAR*" olarak doğru bir şekilde deşifre edilmiş olur. Bu örnek, KDŞT ve DH protokolünün birleşik kullanımının şifreleme ve deşifreleme sürecini nasıl sağladığını göstermektedir. Sistemin güvenliği hem dinamik programlama tablosunun karmaşıklığına hem de ayrık logaritma probleminin zorluğuna dayanmaktadır.

4. SONUÇ ve TARTIŞMA

Bu bölümde, çalışmamızın temel araştırma sorusu olan 'Para Problemi'nin dinamik programlama çözümü temelli ve DH entegreli bir asimetrik şifreleme tekniği, geleneksel yöntemlere kıyasla daha dinamik ve güvenli bir alternatif sunabilir mi?' sorusuna bulgularımız ışığında yanıt verilmekte ve elde edilen sonuçlar detaylı tartışılmaktadır.

Bu çalışmada, klasik Para Problemi'nin dinamik programlama prensipleri ile çözülmesi temeline dayanan özgün bir asimetrik şifreleme tekniği geliştirilmiştir. Geliştirilen sistemin temelini oluşturan KDŞT, harflerin şifrelenmesinde çoklu indis seçenekleri sunarak polialfabetik bir yapı sağlamaktadır. Asimetrik anahtar değişimi için ise DH Anahtar Alışverişi Protokolü başarıyla entegre edilerek, taraflar arasında güvenli bir ortak gizli anahtarın (k anahtarı) oluşturulması sağlanmıştır.

Elde edilen bulgular, geliştirilen model ile belirlenen herhangi bir metnin hatasız bir şekilde şifrelenebildiğini ve tekrar deşifre edilebildiğini göstermiştir. Bu başarı, sistemin operasyonel tutarlılığını ve doğruluğunu kanıtlamaktadır. KDŞT'nun esnek yapısı, belirlenecek herhangi bir alfabenin harf sayısına göre ($|\ \Sigma\ | = p$) ve istenen ($n \times m$) boyutlarında (burada $n, m \in \mathbb{Z}$) oluşturulabilme potansiyeli sunmaktadır. Tablonun ($mod\ p$) için düzenlenmesi ve harf frekanslarının hesaplanarak kullanılması, şifreleme algoritmasının adaptasyon kabiliyetini ve güvenlik düzeyini artırmaktadır.

Geliştirilen bu özgün teknik, klasik şifreleme yaklaşımlarından farklılaşan önemli özelliklere sahiptir. Özellikle, KDŞT'nin dinamik programlama ile matematiksel bir temel üzerine inşa edilmesi, statik tablolara kıyasla daha yüksek bir karmaşıklık ve tahmin edilemezlik sunar. Her bir açık metin karakterinin, şifreleme tablosundaki birden fazla indis çifti ile temsil edilebilmesi, geleneksel tek-tek eşleşmeli şifrelerin (örneğin temel Polybius şifresi) aksine frekans analizi gibi istatistiksel saldırılara karşı daha güçlü bir direnç sağlamaktadır. Anahtarın (k anahtarı) DH protokolü ile güvenli bir şekilde oluşturulması ve şifreleme sürecine dâhil edilmesi, sistemin anahtar yönetimi ve genel güvenliğini desteklemektedir. Çözücü, şifreli metindeki indis çiftlerini alarak, önce KDŞT üzerinden karşılık gelen sayısal değerleri, ardından bu değerleri ortak gizli k anahtarı ile ters dönüşüme tabi tutarak açık metni elde eder.

Geliştirilen bu modelin, özellikle asimetrik şifreleme algoritmalarının temel prensibi olan çok büyük asal sayıların kullanımının bazen hız kaybına neden olabileceği günümüz bilgi güvenliği ortamında önemli bir potansiyeli bulunmaktadır. Bu çalışma, daha küçük asal sayıların kullanımına izin veren, hem simetrik hem de asimetrik şifreleme prensiplerini entegre edebilecek yeni şifreleme tekniklerinin geliştirilmesine yönelik bir açılım sunmaktadır. Bu niteliğiyle, farklı kriptografik yöntemlerle dönüşümlere ve geliştirmelere açık bir temel oluşturmaktadır.

Gelecek Çalışmalar:

Bu çalışmanın bulguları ışığında, sistemin performansı ve güvenlik analizi üzerine daha detaylı araştırmalar yapılabilir. Özellikle:

- Farklı alfabe boyutları ve madeni para setleri kullanılarak KDŞT'nin boyutu ve karmaşıklığının performansa etkisi incelenebilir.

- Geliştirilen şifreleme tekniğinin standart güvenlik analizleri (örneğin, zaman karmaşıklığı, brute-force saldırılarına karşı direnç, bilinen açık metin saldırıları) derinlemesine yapılabilir.

- Sistemin, mevcut popüler asimetrik veya hibrit şifreleme algoritmalarıyla (örneğin RSA, ECC) karşılaştırmalı performans ve güvenlik analizleri gerçekleştirilebilir.

- Geliştirilen modelin gerçek dünya uygulamaları için bir prototip yazılımı geliştirilerek kullanım kolaylığı ve pratikliği değerlendirilebilir.

KAYNAKLAR

Aho, A. V., Hopcroft, J. E., ve Ullman, J. D. (1974). The design and analysis of computer algorithms. Addison –Wesley.

Bernstein, D. J., Lange, T., ve van Someren, N. (2017). "The dance of the elliptic curves: An introduction to post-quantum cryptography". In G. Althunian, P. R. (Ed.), Post-Quantum Cryptography (ss. 1 –26). Springer.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., ve Stein, C. (2009). Introduction to algorithms (3. baskı). MIT Press.

Diffie, W., ve Hellman, M. E. (1976). "New directions in cryptography". IEEE Transactions on Information Theory, 22(6): 644 –654.

Kahn, D. (1967). The code breakers: The story of secret writing. Macmillan.

Katz, J., ve Lindell, Y. (2014). Introduction to modern cryptography (2. baskı). CRC Press.

Koblitz, N. (1994). A course in number theory and cryptography (2. baskı). Springer-Verlag.

Menezes, A. J., van Oorschot, P. C., ve Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.

Nabiyev, V. (2007). Teoriden uygulamalara. Seçkin Yayıncılık. Ankara –Türkiye.

Skiena, S. S. (2008). The algorithm design manual (2. baskı). Springer Science ve Business Media.

Stallings, W. (2017). Cryptography and network security: Principles and practice (7. baskı). Pearson.

Şahin, M. (2018). Kriptolojiye giriş. Ankara Üniversitesi Açık Ders Malzemeleri. Erişim Adresi: https://acikders.ankara.edu.tr/course/view.php?id=26. Son Erişim Tarihi: 05.07.2026.

Topaloğlu, N., Calp, M. H., ve Türk, B. (2016). "Bilgi güvenliği kapsamında yeni bir veri şifreleme algoritması tasarımı ve gerçekleştirilmesi". Bilişim Teknolojileri Dergisi, 9(3): 291 – 298.

Yerlikaya, T., Buluş, E., ve Buluş, N. (2006). "Asimetrik şifreleme algoritmalarında anahtar değişim sistemleri". Akademik Bilişim Konferansı Bildirileri, 9 – 11 Şubat 2006, Denizli.