T.C.

TOKAT GAZİOSMANPAŞA ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

MATEMATİK ANABİLİM DALI

YÜKSEK LİSANS PROGRAMI

AYRIŞIMLI ESNEK KÜMELER VE ESNEK ŞİFRELEME

YÜKSEK LİSANS TEZİ

Fikret ÇİFÇİ

Danışman: Prof. Dr. Naim ÇAĞMAN

TOKAT - 2025

ETİK SÖZLEŞME

Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü'nün tez yazım kılavuzuna dayanarak, Prof. Dr. Naim ÇAĞMAN danışmanlığında kaleme aldığım "Ayrışımlı Esnek Kümeler ve Esnek Şifreleme" başlıklı Yüksek Lisans tezimin bilimsel etik kurallarına uygun olarak hazırlandığını, özgün bir çalışma olduğunu, başkalarının eserlerinden faydalanırken bilimsel normlara uygun şekilde atıfta bulunduğumu, tezin içerdiği yeniliklerin ve sonuçların başka bir kaynaktan alınmadığını, kullanılan verilerde hiçbir tahrifat yapılmadığını ve tezin hiçbir bölümünün başka bir üniversitede başka bir tez olarak sunulmadığını beyan ederim.

31/12/2024

Fikret ÇİFÇİ

JÜRİ KABUL VE ONAY

Fikret ÇİFÇİ tarafından hazırlanan **"Ayrşımlı Esnek Kümeler ve Esnek Şifreleme"** adlı tez çalışmasının savunma sınavı 31.12.2024 tarihinde yapılmış olup aşağıda verilen Jüri tarafından Oy Birliği ile Tokat Gaziosmanpaşa Üniversitesi Lisansüstü Eğitim Enstitüsü Matematik Anabilim Dalı'nda Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Üyeleri (Unvanı, Adı Soyadı) **İmzası**

Üye (Danışman): Prof. Dr. Naim ÇAĞMAN

Üye (Başkan): Doç. Dr. Uğur ERKAN

Üye : Dr. Öğr. Üyesi Filiz ÇITAK

TEŞEKKÜR

Yüksek Lisans eğitimim boyunca bilgi ve tecrübelerini hiçbir zaman esirgemeyen bana karşı son derece anlayışlı davranan ve her türlü desteği veren sayın danışman hocam Prof. Dr. Naim ÇAĞMAN'a saygılarımı sunar teşekkür ederim. Çalışmalarımda tecrübelerinden faydalandığım Öğr. Gör. Hadi ESMERAY'a, Arş. Gör. Halil İbrahim KANAT'a teşekkür ederim.

Bu süreçte bana destek ve moral veren başta çok kıymetli eşim Ebru olmak üzere çocuklarım Yusuf, Ahmet Turan ve Elif Nil'e sevgilerimi sunar teşekkür ederim.

31.12.2024
Fikret ÇİFÇİ

ÖZET

AYRIŞIMLI ESNEK KÜMELER VE ESNEK ŞİFRELEME

Çifçi, Fikret
Yüksek Lisans, Matematik Anabilim Dalı
Tez Danışmanı: Prof. Dr. Naim ÇAĞMAN
Aralık 2024, xi + 62 sayfa

Bu tez çalışmasında, önce esnek kümleri ve esnek küme işlemlerini tanıttık. Esnek küme teorisi, belirsizlik içeren problemlerle başa çıkmak için 1999 yılında Molodtsov tarafından ortaya atılmıştır. O zamandan bu tarafa, karar verme uygulamaları başta olmak üzere esnek kümleri birçok alana uygulaması yapılmıştır. Sonra esnek kümlerin üzerinde tanımlandığı kümeyi denklik sınıflarına ayırarak ayrışımlı esnek kümleri tanımladık. Bunun için ayrı bir bölüm olarak denklik bağıntısı ve denklik sınıflarıyla ilgili temel tanım ve teoremleri ispatlarıyla verdik. Daha sonra ayrışımlı esnek kümlerine dayanarak esnek şifrelemeyi tanımladık. Yeni tanımladığımız esnek şifreleme daha komplike bir Sezar şifreleme türüdür. Şifreleme, verilerin veya bilgilerin belirli bir algoritma kullanılarak okunamaz hale getirilmesi işlemidir. Bu işlem, dijital dünyada bilgi güvenliğini sağlamak için hayati öneme sahiptir. Verilerin güvenli bir şekilde iletilmesi, saklanması ve işlenmesi gerektiği her durumda şifreleme kritik bir rol oynar. Hem bireysel kullanıcılar hem de kurumlar için şifreleme, siber saldırılara karşı en etkili savunmalardan birisidir. Esnek şifrelemenin matematiksel modellemesini yaptıktan sonra Python programlama dili kullanılarak bilgisayar programını yazdık. Son olarak esnek şifrelemenin mevcut şifrelemelerdeki yeri ve geleceği hakkında bilgiler verdiğimiz sonuç bölümüyle tezi bitirdik.

Anahtar Kelimeler: Denklik Sınıfı, Esnek Küme, Ayrışımlı Esnek Küme, Esnek Şifreleme, Sezar Şifreleme, Esnek Şifreleme Uygulama Programı

ABSTRACT
DECOMPOSED SOFT SETS AND SOFT ENCRYPTION

Çifçi, Fikret
Master's Degree, Department of Mathematics
Thesis Advisor: Prof. Dr. Naim ÇAĞMAN
December 2024, xi + 62 pages

In this thesis, we first introduced soft sets and soft set operations. Soft set theory was introduced by Molodtsov in 1999 to deal with problems involving uncertainty. Since then, soft sets have been applied to many areas, especially decision-making applications. We then defined decomposed soft sets by dividing the set on which soft sets are defined into equivalence classes. For this purpose, we gave the basic definitions and theorems related to equivalence relations and equivalence classes with their proofs as a separate section. We then defined soft encryption based on decomposed soft sets. The soft encryption we have newly defined is a more complicated type of Caesar encryption. Encryption is the process of making data or information unreadable using a certain algorithm. This process is of vital importance for ensuring information security in the digital world. Encryption plays a critical role in every situation where data must be transmitted, stored and processed securely. Encryption is one of the most effective defenses against cyber-attacks for both individual users and institutions. After doing the mathematical modeling of soft encryption, we wrote the computer program using the Python programming language. Finally, we finished the thesis with the conclusion section where we gave information about the place and future of soft encryption in existing encryptions.

Keywords: Soft Sets, Decomposed Soft Sets, Soft Encryption, Caesar Cipher, Computer Program

SİMGE ve SEMBOLLER

U: Evrensel küme

E: Parametreler kümesi

ϕ: Boş esnek küme

F: Esnek küme

$\tilde{E}$: Evrensel esnek küme

$\subseteq$: Altküme sembolü

$\tilde{\subseteq}$: Esnek altküme

F°: Esnek kümenin tümleyeni

$\tilde{\cup}$: Esnek birleşim

$\tilde{\cap}$: Esnek kesişim

KISALTMALAR

AES: Advanced Encryption Standard (İleri Düzey Şifreleme Standardı)

BCI: Bilinçli Cebir İlişki

BCK: Bilinçli Cebir Kavramı

BGYS: Bilgi Güvenliği Yönetim Sistemi

DES: Data Encryption Standard (Veri Şifreleme Standardı)

ECC: Elliptic Curve Cryptography (Eliptik Eğri Kriptografisi)

IEC: Uluslararası Elektroteknik Komisyonu

ISO: Uluslararası Standardizasyon Örgütü

M.Ö.: Milattan Önce

RC4: Rivest Cipher 4 (Rivest Şifresi 4)

RSA: Rivest Shamir Adleman (Asimetrik bir şifreleme algoritmasıdır)

XOR: Exclusive OR (Ayrı veya)

WS: Workflow System (İş Akış Sistemi)

ŞEKİL LİSTESİ

ÇİZELGE LİSTESİ

İÇİNDEKİLER

1. GİRİŞ

İnsanlık tarihi boyunca, önem arz eden bilgilerin gizlenmesi veya korunması ihtiyacı hep var olmuştur. Bu gereksinimler doğrultusunda, zamanla çeşitli gizleme yöntemleri geliştirilmiştir. Bu yöntemlerin birçoğu günümüzde kullanılmamakla birlikte, modern veri saklama sistemlerinin oluşumunda ilham kaynağı olmuştur. Bilgilerin gizlenmesi, genellikle stenografi ve kriptoloji olmak üzere iki ana kategori altında incelenir.

Stenografi, bilginin bir nesnenin renk kodları içine gizlenmesi işlemidir (Petitcolas ve ark., 1999). Bu terim, Yunanca kökenlidir ve "kaplanmış yazı" anlamına gelir (Anonim, 2024). Geçmişte, bilgi iletimi için resim ve kitap gibi fiziksel nesneler kullanılırken, teknolojinin ilerlemesiyle gizleme yöntemleri günümüzde daha gelişmiş bir hale gelmiştir. Örneğin, bir bilginin bilgisayar ortamında bir resmin renk kodları arasına gizlenmesi bir steganografi örneğidir.

Kriptoloji terimi, Yunanca 'krypto' (gizli) ve 'logos' (kelime) kelimelerinden türetilmiştir ve bilginin korunmasıyla ilgili bir bilim dalı olarak tanımlanır (Schneier, 2007). Bilgi güvenliğini sağlamak amacıyla uygulanan matematiksel teknikler genellikle kriptoloji çerçevesinde ele alınır. Bu alan, iki ana başlık altında incelenmektedir: kriptografi ve kriptanaliz.

Kriptografi, bilgiyi güvenli bir şekilde iletmek için şifreleme ve çözme işlemleriyle ilgilenen kriptolojinin bir alt dalıdır. Kriptanaliz ise şifrelenmiş metinlerin çözümlenmesi ile ilgilenir. Kriptografiyle uğraşanlara kriptograflar, kriptanalizle uğraşanlara ise kriptoanalistler denir (Dooley, 2018).

Kriptografi tekniği, dört ana bölümde incelenmektedir: Bunlar algoritma, anahtar, şifreli metin ve açık metindir. (Bauer, 2013). Veri gönderici, belirli bir algoritmayı kullanarak açık metni şifreler. Şifreleme işlemi sonucunda, anahtar ve şifreli metin ayrı kanallar aracılığıyla alıcıya iletilir. Alıcı, aynı algoritma ile anahtarı kullanarak şifreli metni çözer ve açık metne ulaşır. Şifreli metnin çözülmesi, algoritmanın karmaşıklığına ve anahtarın

uzunluğuna veya rastgeleliğine bağlı olarak değişir. Algoritma ve anahtarın gücü arttıkça, şifreli metnin çözülmesi de o kadar zorlaşır (Li ve Wang, 2022).

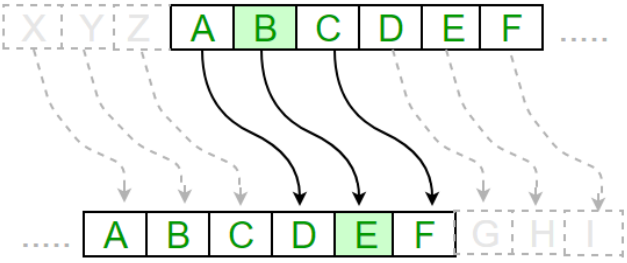
Şifreleme, insanlığın ilk zamanlarından beri farklı yöntem ve şekillerle kullanılmaktadır. Şifreleme örnekleri, Nil Nehri yakınlarındaki "Menett Khufu" kasabasında bulunan hiyerogliflerde görülmektedir (Rana, 2022). Bu hiyeroglifler incelendiğinde, daha önce hiç görülmemiş sembollere rastlanmıştır. Bu sembollerin, sadece belirli katipler tarafından okunabileceği için şifrelendiği düşünülmektedir.

Mezopotamya'da, M.Ö. 1500'lerde geliştirilen bir bileşim formülü, çivi yazısıyla şifrelendi ve daha sonra okunabilir hale getirildi (Bauer, 2013). M.Ö. 457'de, Anadolu'da Spartalılar tarafından Şekil 1-1 de görülen Scytale adı verilen bir aletle şifreleme yöntemi kullanıldı. Bu teknik, bir şerit kâğıdın belirli bir çapta silindire sarılması ve her şeride bir harf düşecek şekilde yatay olarak yazılmasıyla özetlenebilir. Şifrelenmiş metin, silindir çıkarıldığında karışık bir sıralama ile ortaya çıkar. Metni yeniden okuyabilmek için, şifreleme sırasında kullanılan çapta bir silindire ihtiyaç vardır. Bu nedenle, Scytale tarihteki bilinen ilk kripto aleti olarak kabul edilmektedir (Stanoyevitch, 2010).

Şekil 1-1: Scytale (Wikipedia (2024c))

M.Ö. 100 yılında doğan Sezar (Julius Caesar), şifrelemeyi etkin bir şekilde kullandı. Sezar şifresinin ilk yazılı kaydı, MÖ 1. yüzyılda yaşamış Romalı tarihçi Suetonius'un De Vita Caesarum (Sezar'ın Hayatı) adlı eserinde yer almaktadır (Tranquillus, 2013). Sezar, şifrelemek istediği metin için belirli bir alfabe kullandı ve her harfi alfabede kendisinden üç harf sonraki harf ile değiştirdi. Böylece alfabe, üç adım kaydırılarak bir şifreleme sağlandı. Sezar, generallerine bu teknikle şifreli mesajlar gönderdi. O dönemde kriptoanaliz yöntemlerinin gelişmemiş olması nedeniyle bu yöntemle iletilen mesajlar

güvenli bir şekilde korunabiliyordu. Şekil 1.2 de Sezar şifrelemesinin uygulanması gösterilmektedir (Topaloğlu ve ark., 2016; Wikipedia, 2024b).



Şekil 1-2: Sezar şifresi gösterimi (YazilimTurk (2024))

Bugün, sabit anahtarlı ve düşük güvenlik seviyelerine sahip bu yöntem, günümüz kriptoanaliz tekniklerinden olan geçiş sıklığı analiziyle kolaylıkla çözülebilmektedir.

Arap filozof Al-Kindi (801-873), modern şifrelemenin temellerini atmıştır. Mesaj gizleme ve çözme yöntemlerine dair günümüze ulaşan en eski eser, Al-Kindi'nin 'Risale I'stihraci'l-mu'amma' adlı çalışmasıdır. Al-Kındi, kriptografi alanında çığır açan ilk düşünürlerden biri olarak kabul edilir. "Risale I'stihraci'l-mu'amma" adlı çalışması, bu alandaki en eski ve kapsamlı eserlerden biridir. Eserinde, şifreleme ve deşifre süreçlerinin temel ilkelerini detaylı bir şekilde ele alır, çeşitli şifreleme yöntemlerini inceler ve kendi özgün şifreleme tekniklerini ortaya koyar (Piper ve Murphy, 2002). Bu eserinde, Al-Kindi ilk kez “yerine koyma” ve “yer değiştirme” işlemlerini tanımlamış ve bu iki işlemin bir arada kullanıldığı karma şifreleme sistemlerini açıklamıştır.

Şekil 1-3: Alberti diski (Wikipedia (2024d))

Leon Alberti, Sezar şifrelemesine benzer bir harf kaydırma tekniği kullanarak ilk kez çoklu alfabayla şifreleme gerçekleştirmiştir. Sezar şifresinde, alfabenin harfleri sırasıyla

üç harf kaydırılırken, Alberti'nin yönteminde kaydırma miktarı sabit değildir. Şifrelenecek metindeki her harfin kriptolu karşılığı Şekil 1.3'te gösterilen Alberti Diski yardımıyla belirlenir (Selleri, 2020). Şekil 1.3'te Alberta diski gösterilmektedir.

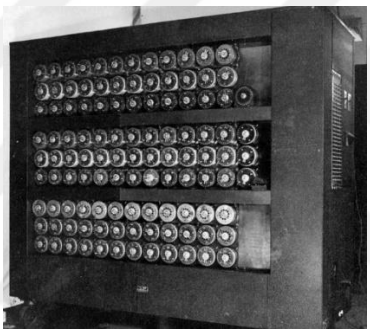
Vigenere şifrelemesi, 1800'lü yılların ortalarına kadar çözülmesi imkansız olarak kabul ediliyordu ve bu şifreleme tekniği Blaise de Vigenere tarafından geliştirilmiştir. 1586 yılında icat edilen bu teknik, Sezar şifrelemesine benzer bir alfabenin kaydırılması metoduna dayanmaktadır. Şifreleme işlemi sırasında Şekil 1.4'te gösterilen Vigenere sistemi "Vigenere Tablosu" adı verilen bu özel tabloyla gerçekleştirilir (Bhardwaj, 2012; Topaloğlu ve ark., 2016).

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

Şekil 1-4: Vigenere tablosu (Michigantech (2025))

Vernam, Vigenere şifrelemesi temel alınarak kendi adını taşıyan Vernam şifresini geliştirdi. Vernam şifresinde, her karakter için benzersiz bir kaydırma miktarı kullanılarak her karakter şifrelenir ve bu kaydırma miktarı bir daha kullanılmaz. Bu yöntemle oluşturulan Vernam şifresinin teorik olarak çözülmesi imkansızdır. Ancak, Vernam şifresinin uygulanması oldukça zordur çünkü her bir harfi şifrelemek veya çözmek için mesaj uzunluğunda bir şifreleme dizisinin iletilmesi gerekmektedir (Leung, 2000).

Şifreleme, devlet ve ordu güvenliğini sağlamak amacıyla ortaya çıktığı için, aynı zamanda teknolojinin gelişimiyle birlikte bir dizi güvenlik sorununu da beraberinde getirmiştir. Telsizin icadı, bilgi güvenliğini öncelik haline getirdi ve devletler, telsiz frekanslarını korumak amacıyla kriptografi tekniklerini geliştirmeye yöneldi. Birinci Dünya Savaşı sırasında, şifrelenmiş bir telgrafın çözülmesi savaşın gidişatını değiştirdi. İkinci Dünya Savaşı'nda ise, Alman ordusunun Şekil 1-6 da Enigma gibi şifreleme araçlarını aktif bir şekilde kullanması, şifrelemenin önemini daha da artırdı. Alman ordusundaki muhabere birimleri, mesajlarını Enigma cihazının tuş takımında yazarak şifreli metin elde ediyor ve bu şifreli mesajlar, telsiz operatörleri tarafından mors kodu aracılığıyla karşı tarafa iletiliyordu (Batey, 2017).



Şekil 1-5: Alan Turing tarafından geliştirilen Bombe cihazı (Ekonomikultur, 2024)

Enigma, mekanik ayarlarla kontrol edilen şifreleme işlemini elektronik olarak gerçekleştiren bir cihazdı. 1918'de Arthur Scherbius tarafından ticari amaçlarla geliştirilmiş ve başlangıçta bankalar arasındaki iletişimde kullanılmıştır. Polonyalılar, Fransızlar ve İngilizler, Enigma'nın şifresini çözmek için büyük çaba harcadılar; Polonyalılar ve Fransızlar, Almanlara yenilince, bu görevi tek başlarına İngilizler devraldılar.

Şekil 1-6: Enigma cihazı (Wikipedia, 2024a)

İngilizler, Enigma'yı kırmak için çok sayıda matematikçi ve mühendisi çalıştırdılar. Zorlu koşullar nedeniyle Enigma'nın kırılması gecikti, ancak Alan Turing tarafından geliştirilen ve Şekil 1-5'te "Bombe" olarak adlandırılan bir cihaz sayesinde sonunda kırıldı. Bu cihaz, kriptoanaliz saldırıları için ilk araç olarak kabul edilmektedir (Denniston ve ark., 2020).

Transistörlerin icadıyla birlikte şifreleme alanında da büyük bir ilerleme kaydedildi (Bardeen ve Brattain, 1948). Teknolojinin gelişmesiyle veri işleme kapasiteleri arttı ve bu da kriptoanaliz çalışmalarının temelini oluşturdu. Kriptoanaliz alanındaki bu ilerlemeler, yeni kripto algoritmalarının geliştirilmesine olanak sağladı. Ancak, kripto algoritmalarının güvenilirliği sadece anahtar uzunluğuna değil, aynı zamanda matematiksel çözümleme yöntemlerine karşı da dayanıklı olmalıdır. Şifrelemenin tarih boyunca var olduğu düşünüldüğünde, bilginin gizliliğine duyulan ihtiyaç devam ettiği sürece şifreleme teknikleri de sürekli olarak gelişecektir. Ancak, bu gelişmeler hem teknolojinin ilerlemesine hem de kriptoanalistlerin yeteneklerine bağlıdır. Son on yılda, bir zamanlar 20 yıl sürebilecek olan Kaba Kuvvet saldırıları günümüz teknolojisi ile çok daha kısa sürede çözülebilmektedir. Ayrıca, kriptoanalistlerin çalışmaları sayesinde şifreleme algoritmalarının zayıf noktaları keşfedilerek şifreler çözülebilmektedir. Bu gelişmeler, daha güçlü ve karmaşık şifreleme algoritmalarının gerekliliğini ortaya koymaktadır.

Burada esnek küme teorisi kullanılarak farklı bir şifreleme algoritması oluşturacağız. Esnek küme teorisi, belirsizlik içeren problemlerle başa çıkmak için 1999 yılında Molodtsov tarafından öne sürüldü (Molodtsov, 1999). Molodtsov'un öncülüğünde, bu teori smooth fonksiyonları, oyun teorisi, Riemann integrasyonu, Perron integrasyonu, olasılık teorisi ve ölçüm teorisi gibi çeşitli konular üzerinde uygulandı. Maji ve ekibi ile Pawlak'ın yaklaşımlı küme teorisi kullanılarak bir karar verme probleminde esnek kümelerin bir uygulaması sunuldu ve esnek kümelerde bazı işlemler tanımlandı (Maji ve ark., 2003; Pawlak, 1982). Xiao ve arkadaşları esnek küme temelli iş rekabet kapasitesi için yapay bir hesaplama yöntemi üzerinde çalıştılar (Xiao ve ark., 2005). (Xiao ve ark., 2005) ile (Pei ve Miao, 2005), esnek tabanlı bilgi sistemleri üzerine çalışmalar sunarken, (Mushrif ve ark., 2006) esnek küme temelli sınıflandırmalar üzerinde çalıştı. Yang ve arkadaşları, klinik teşhisin karar analizi ve indüksiyonu konusunda esnek kümeleri ve

yaklaşımlı kümeleri temel alan bir çalışma gerçekleştirdiler (Yang ve ark., 2007). Esnek kümelerin cebirsel özellikleri birçok yazar tarafından incelendi. (Aktaş ve Çağman, 2007) esnek grupların yeni bir tanımını sunarak bazı temel özelliklerini belirlediler. (Jun, 2008) esnek BCK/BCI-cebirleri ve esnek alt cebir kavramlarını ortaya atarak, onların bazı temel özelliklerini türetti. (Jun ve Park, 2008) esnek kümeleri BCK/BCI-cebirleri üzerinde tartışırken, (Park ve ark., 2008) esnek WS-cebiri üzerine bir çalışma yaptı. (Feng ve ark., 2008) esnek küme teorisini kullanarak esnek halkalar çalışmasını sundular ve ilgili bazı özelliklerini incelediler. (Sun ve ark., 2008) esnek modüllerin tanımını verdi ve bu modülleri ve Molodtsov'un esnek küme tanımını kullanarak bazı temel özelliklerini inşa etti. (Çağman ve Enginoğlu, 2010a) esnek matris teorisini tanımladılar. (Atagün ve Sezgin, 2011) esnek küme işlemleri üzerine çalıştılar ve (Sezgin ve ark., 2011) esnek yakın-halkalar ve idealist esnek yakın-halkalar ile ilgili çalışmalar yaptılar. (Molodtsov ve ark., 2006) esnek küme teorisi üzerine dayalı bir analiz geliştirerek, esnek sayı, esnek türev, esnek integral gibi kavramları formüle ettiler. Bu analiz, (Kovkov ve ark., 2007) tarafından optimizasyon teorisi ile ilgili problemlere uygulandı. Esnek kümeler üzerindeki farklı işlemlerin temel analizi (Ali ve ark., 2009) tarafından açıklandı. (Çağman ve Enginoğlu, 2010b) Molodtsov'un esnek kümelerindeki işlemleri, daha fonksiyonel hale getirerek ve yeniden tanımlayarak farklı yeni sonuçlar elde ettiler. Bu yeni tanımları kullanarak, esnek karar verme yöntemleri geliştirdiler.

Bu tezin birinci bölümünde esnek kümler ve şifrelemeyle ilgili bir literatür taraması yapacağız. İkinci bölümde şifrelemeyle ilgi temel kavramları vereceğiz. Üçüncü bölümde esnek kümleri ve esnek küme işlemlerini tanıtacağız. Dördüncü bölümde denklik bağıntısı ve denklik sınıflarıyla ilgi tezin diğer bölümlerinde kullanacağımız gerekli bilgileri vereceğiz. Beşinci bölümde esnek kümlerin üzerinde tanımlandığı kümeyi denklik sınıflarına yeni bir kavram olarak ayrışımlı esnek kümleri tanımlayacağız. Altıncı bölümde ayrışımlı esnek kümlerine dayanarak esnek şifrelemeyi tanımladıktan sonra esnek şifrelemenin matematiksel modellemesini yapacağız. Yeni tanımladığımız esnek şifreleme daha komplike bir Sezar şifreleme türüdür. Sekizinci bölümde esnek şifrelemenin Python programlama dili kullanılarak bilgisayar programını yazacağız. Son bölüm olan dokuzuncu bölümde esnek şifrelemenin mevcut şifrelemelerdeki yeri ve geleceği hakkında bilgiler vererek tezi bitireceğiz.

2. ŞİFRELEME

Bu bölümde, şifrelemenin temelleri ele alınarak; temel tanımlar ve yöntemler üzerinde durulacaktır.

2.1 Bilgi Güvenliği

Bilgi güvenliği, bilginin yetkisiz erişim, kullanım, ifşa, bozulma, değiştirilme, kaybolma veya yok edilme gibi tehditlere karşı korunmasını hedefleyen bir disiplindir. Bu disiplin, bilginin gizliliğini, bütünlüğünü ve erişilebilirliğini güvence altına almayı amaçlar. Günümüzde bilgi güvenliği, bilginin dijitalleşmesi ve internetin yaygınlaşmasıyla her zamankinden daha büyük bir önem kazanmıştır. Dijital bilgilere erişim ve kullanım imkanlarının artması, beraberinde bilgi güvenliği risklerini de getirmiştir (Tekerek, 2008).

(Eminağaoğlu ve Gökşen, 2009) bilgi güvenliğini "bilginin yetkisiz erişim, kullanım, ifşa, bozulma, değiştirilme, kaybolma veya yok edilme gibi tehditlerden korunması" olarak tanımlamaktadır. (Menezes ve ark., 2018) ise bilgi güvenliğini "bilginin gizliliğini, bütünlüğünü ve erişilebilirliğini koruma bilimi ve sanatı" olarak tanımlamaktadır.

(Aghayev, 2017) yaptığı tez çalışmasında, bilgi güvenliğinin sağlanmasında mevcut şifreleme sistemleri ve bu sistemlerde karşılaşılan hatalar incelenerek şifreleme ile ilgili önerilere uygulamalı olarak yer vermiştir.

Uluslararası bilgi güvenliği standardı ISO/IEC 27001, bilgi güvenliği yönetim sisteminin (BGYS) gerekliliklerini belirler. Bu standart, bilgi güvenliğini korumak için bir dizi kontrol önleminin uygulanmasını öngörmektedir (Başar, 2022).

2.1.1 Gizlilik

Bilgi güvenliğinin temel unsurlarından biri olan gizlilik, yetkisiz ve izinsiz kişilerin bilgiye erişimini engelleme disiplinidir. Bu, bilginin sadece yetkili kişiler tarafından kullanılabilir ve görülebilir olmasını sağlar. Günümüzde bilgi güvenliği, internet yoluyla

yayılan virüsler, solucanlar, truva atları, zararlı yazılımlar, iç saldırganlar (internal hacker), yetersiz koruma açıkları ve yapılandırma hataları gibi tehditlerle karşı karşıyadır. Bu tür bilgi güvenliği ihlalleri, veri kayıplarına, mali zararlara, itibar kaybına, hizmetlerin kesintiye uğramasına ve gizli bilgilerin ele geçirilmesine neden olabilmektedir (Canbek ve Sağıroğlu, 2006; Tekerek, 2008).

Gizlilik, bir bireyin veya kuruluşun kişisel bilgilerinin kontrolünü elinde tutma ve bu bilgilerin kimler tarafından, ne zaman ve nasıl kullanılacağına karar verme hakkı olarak tanımlanmaktadır. Bu bağlamda gizliliği de içeren güvenlik kavramı; veri gizliliği, hizmet sürekliliği ve bütünlüğü, kötü amaçlı yazılımlara karşı koruma, bilgi bütünlüğünün korunması ve erişim kontrolü gibi birçok görevi barındırmaktadır (Eroğlu, 2018; Öztürk ve Zeybek, 2021).

(Kuner, 2010) 'a göre gizlilik, bireylerin kişisel bilgilerinin rızası olmadan toplanmasını, kullanılmasını veya ifşa edilmesini önleme olarak tanımlanır.

Bilgi güvenliğinde gizlilik, şifreleme yoluyla sağlanabilir. Şifreleme, bilginin yetkisiz kişiler tarafından okunmasını veya anlaşılmasını zorlaştıran bir matematiksel işlemdir. Şifreleme algoritmaları, bilginin okunabilir hale gelmesi için bir anahtar kullanır. Bu anahtar, yalnızca yetkili kişiler tarafından bilinir (Canbek ve Sağıroğlu, 2006).

2.1.2 Bütünlük

Bilgi güvenliğinin bütünlük unsuru, bilginin göndericiden alıcıya eksiksiz ve doğru bir şekilde iletilmesini ve saklanmasını sağlar. Bütünlük, bilginin göndericiden alıcıya eksiksiz ve doğru bir şekilde ulaşmasını ifade eder; bu kavram, bilginin doğruluğu ve eksiksiz olması olarak tanımlanır (Baykara ve ark., 2013).

Şifreleme, bilgi bütünlüğünü korumak için kullanılan önemli bir araçtır. Şifrelenmiş bilgiler, sadece yetkili kişiler tarafından çözülebilir. Bu sayede, bilginin değiştirilmesi veya yok edilmesi engellenmiş olur (Şahin, 2015).

Bütünlük, bilgi güvenliğinin gizlilik ve erişilebilirlik unsurları ile de yakından ilişkilidir. Bilginin gizliliği korunmazsa, bu durum bütünlüğünü de etkileyebilir. Aynı şekilde, bilgiye erişilemezse, bu durum da bütünlüğünü etkileyebilir (Baykara ve ark., 2013; Şahin, 2015).

Bütünlük unsuru, bilginin eksiksiz ve doğru bir şekilde iletilmesini ve saklanmasını sağlar. Bütünlük, bilgi güvenliğinin önemli bir unsurudur ve şifreleme, bütünlüğü korumak için kullanılan önemli bir araçtır (Baykara ve ark., 2013).

2.1.3 Erişebilirlik

Bilgi güvenliğinin erişilebilirlik unsuru, yetkili kişilerin ihtiyaç duydukları bilgilere zamanında ve doğru bir şekilde erişebilmelerini sağlar. (Menezes ve ark., 2018)'e göre erişilebilirlik, "bilginin ihtiyaç duyulduğunda kullanılabilir olması" anlamına gelir. ISO/IEC 27001 standardında ise erişilebilirlik, "bilginin yetkili bir kullanıcı tarafından talep edildiğinde erişilebilir ve kullanılabilir olması" olarak tanımlanır (Yılmaz, 2014). Erişilebilirlik, bilgi güvenliğinin önemli bir unsurudur ve şifreleme, erişilebilirliği korumak için kullanılan önemli bir araçtır (Çubukçu, 2018).

Bilgi güvenliği, gizlilik, bütünlük ve erişilebilirlik gibi temel unsurların yanı sıra kimlik doğrulama, inkar edilemezlik, süreklilik, fiziksel güvenlik ve insan faktörü gibi unsurları da kapsayan bir disiplindir. Bilgi güvenliğini korumak için teknik çözümler ve bilinçli kullanıcıların birlikte hareket etmesi gerekir (Çubukçu, 2018; Menezes ve ark., 2018).

2.2 Şifrelemede Kullanılan Temel Terimler

Bu bölümde, şifreleme de kullanılan terimler ve tanımlara değinilmiştir.

Tanım 2.2.1. Açık Metin: Şifrelenmemiş, anlaşılabilir ve okunabilir metinlere açık metin veya düz metin denir (Aslan, 2013; Beşkirli ve ark., 2019).

Tanım 2.2.2. Şifreli Metin: Açık metnin şifrelenerek anlamsız hale getirilmiş biçimine şifreli metin denir (Aslan, 2013).

Tanım 2.2.3. Şifreleme: Açık metni algılanamayacak veya okunamaz hale getirme, şifreli metne dönüştürme işlemine şifreleme denir (Aghayev, 2017; Beşkirli ve ark., 2019).

Tanım 2.2.4. Şifre Çözme: Şifreli metni çözüp açık metne dönüştürme işlemine şifre çözme adı verilir (Aghayev, 2017; Toyran, 2007).

Tanım 2.2.5. Anahtar: Bilgiyi şifrelemek veya deşifrelemek için kullanılan ve algoritmanın bilinmeyen kısmını oluşturan bölümüne anahtar denir (Beşkirli ve ark., 2019).

Tanım 2.2.6. Simetrik Algoritmalar: Şifreleme ve deşifreleme işlemlerinde aynı anahtarın kullanıldığı algoritmalardır. Tek anahtarlı şifreleme veya gizli anahtarlı algoritmalar denir (Kodaz ve Botsalı, 2010).

Tanım 2.2.7. Asimetrik Algoritmalar: Şifreleme ve deşifreleme işlemlerinde farklı anahtarların (Açık anahtar ve Gizli anahtar) birlikte kullanıldığı algoritmalardır. Açık anahtarlı algoritmalar olarak da bilinir (Yerlikaya ve ark., 2006).

Tanım 2.2.8. Alıcı, Gönderici: Haberleşmede bilgiyi alan kişiye veya şifreli metni deşifre eden kişiye alıcı denir. Bir haberleşmede bilgiyi meşru olarak gönderen kişiye ve açık metni şifreleyen kişiye gönderici denir. (Akleylek ve ark., 2011).

Tanım 2.2.9. Kanal: Bilginin bir kullanıcıdan diğerine iletimi için gereken fiziksel iletişim ortamına kanal denir (Kodaz ve Botsalı, 2010; Yerlikaya ve ark., 2006).

Tanım 2.2.10. Kod: Bilginin kısaltılarak kaydedildiği veya tanımlandığı karakter dizisi veya bilgisayarın tanıyacağı formda özel semboller kullanılarak bilginin gösterilmesi ya da tanımlanmasına kod denir (Şahin, 2015).

Tanım 2.2.11. Şifreleme Algoritması: Şifreleme ve deşifreleme işlemlerinde kullanılan matematiksel işlemlerin bütününe şifreleme algoritması denir (Şahin, 2015).

Tanım 2.2.12. Kriptanaliz: Bir şifreleme sisteminin girdi veya çıktıların inceleyerek, bilgi ve anahtar olmaksızın açık metne ulaşma işlemine kriptanaliz denir. Aynı zamanda kod kırma olarak da adlandırılır. Kriptanalizin uygulayıcılarına kriptanalist denir (Abdirahman, 2018).

Tanım 2.2.13. Veri Bütünlüğü: Bilginin göndericiden alıcıya gitmesi esnasında yanlışlıkla ya da bilerek değiştirilmesinin önlenmesine veri bütünlüğü denir (Avaroğlu, 2022).

Tanım 2.2.14. Saldırgan, Saldırı: Alıcı ve gönderici arasındaki haberleşmede mesaj alıcı veya gönderici olmayıp güvenliği kırmaya çalışan zararlı kişiye saldırgan denir. Bir şifreleme algoritmasının veya şifreli metnin bir kısmını veya tamamını kırmak için yapılan başarısız veya başarılı teşebbüslere saldırı adı verilir (Ersan, 2017).

2.3 Şifreleme Çeşitleri

Şifreleme, bilgi güvenliğinin temel taşıdır. Bilgiyi yetkisiz erişime karşı korumak için şifreleme algoritmaları kullanılır. Bu algoritmalar, anahtar adı verilen bir parametre kullanarak bilgiyi şifreler ve tekrar okunabilir hale getirir. Anahtar kullanımına göre, gizli anahtarlı (simetrik) ve açık anahtarlı (asimetrik) algoritmalar olmak üzere şifreleme iki türde uygulanmaktadır (Kodaz ve Botsalı, 2010; Yerlikaya ve ark., 2006).

2.3.1 Gizli anahtarlı (Simetrik) algoritmalar

Simetrik şifreleme algoritmaları, veri güvenliği sağlamak amacıyla şifreleme ve şifre çözme işlemleri için tek bir gizli anahtar kullanan yöntemlerdir. Bu yapı, şifreleme sürecinin matematiksel temelini basit ve etkili bir hale getirir; dolayısıyla, simetrik şifreleme, veri şifrelemesi açısından oldukça pratik bir çözüm sunar.

Simetrik şifrelemenin temel avantajı, şifreleme ve şifre çözme işlemlerinin yüksek hızda gerçekleştirilmesidir. Bu durum, özellikle büyük veri setlerinin hızlı bir biçimde korunması gereken senaryolarda oldukça önemlidir. Sistemde kullanılan gizli anahtar, yalnızca şifrelemeyi ve şifre çözmeyi gerçekleştiren taraflar arasında paylaşılmalıdır. Ancak, gizli anahtarın güvenli bir şekilde iletilmesi, simetrik şifrelemenin en zorlu yönlerinden birini oluşturmaktadır. Anahtarın güvenli bir biçimde iletilmemesi durumunda, başkalarının şifreli veriye erişimi kolaylaşır ve veri güvenliği riske girmiş olur.

Simetrik şifreleme algoritmaları, çeşitli şifreleme standartlarına dayanmakta olup, Advanced Encryption Standard (AES) gibi yaygın kullanılan algoritmalar, endüstri genelinde önemli bir yer tutmaktadır. Bu standartlar, güvenilirlik ve hız açısından etkili çözümler sunarak, ticari uygulamalardan bireysel kullanıma kadar geniş bir yelpazede tercih edilmektedir. Bununla birlikte, bu algoritmaların kullanımı, gizli anahtar yönetiminde yaşanan zorluklar nedeniyle bazı sınırlamalara tabidir.

Yaygın olarak kullanılan simetrik algoritmalar arasında AES, DES, RC4 ve Blowfish yer alır (Kodaz ve Botsalı, 2010; Schneier, 2007; Yeşilbaş, 2016). Şekil 2-1 de Simetrik algoritma yapısı görülmektedir.

Şekil 2-1: Simetrik algoritma yapısı

Algoritmanın çalışması:

Gizli anahtarlı (simetrik) şifreleme algoritması, şifreleme ve şifre çözme işlemlerinde aynı anahtarı kullanan bir süreçtir. Bu yöntemde, veriyi gönderen taraf, orijinal veriyi (açık metin) ve gizli bir anahtarı kullanarak şifreleme algoritmasını çalıştırır ve ortaya

çıkan şifreli veriyi (şifreli metin) alıcıya iletir. Alıcı taraf ise aynı gizli anahtarı kullanarak bu şifreli metni deşifre eder ve verinin orijinal haline ulaşır. Simetrik algoritmalar, yüksek hızda çalışmaları nedeniyle büyük veri miktarlarını şifrelemede avantaj sağlarken, güvenlik açısından anahtarın yalnızca iki taraf arasında güvenli bir şekilde paylaşılması gerekliliği önemlidir (Yerlikaya ve ark., 2006).

2.3.2 Açık anahtarlı (Asimetrik) algoritmalar

Açık anahtarlı şifreleme algoritmaları, simetrik şifreleme algoritmalarından köklü bir farklılık sergilemektedir. Bu tür şifreleme yöntemleri, şifreleme ve şifre çözme işlemleri için iki ayrı anahtar kullanır: açık anahtar ve özel anahtar.

Asimetrik algoritmalar olarak da bilinen açık anahtarlı şifreleme sistemlerinde, şifreleme için kullanılan açık anahtar ile şifre çözme için kullanılan özel anahtar birbirinden farklıdır. Anahtar çiftlerini üreten algoritmaların matematiksel özellikleri nedeniyle, her birey için oluşturulan açık-özel anahtar çiftleri benzersizdir. Yani, her kullanıcının sahip olduğu anahtar çifti yalnızca o kullanıcıya özeldir ve başka bir kullanıcı tarafından kullanılamaz. Özel anahtar, pratik bir zaman dilimi içerisinde açık anahtardan hesaplanamaz; bu da güvenliği artıran önemli bir özelliktir.

Bu algoritmalara "açık anahtarlı" denmesinin nedeni, şifreleme anahtarının genel kullanıma açık olmasıdır. Başka bir deyişle, herhangi bir kişi, bir iletinin güvenli bir şekilde şifrelenmesi için açık anahtarı kullanabilir; ancak yalnızca ilgili özel anahtarı elinde bulunduran kişi, bu şifrelenmiş iletinin çözümünü yapabilir. Dolayısıyla, bu sistemde açık anahtar genellikle "şifre anahtarı" olarak adlandırılmakta, özel anahtar ise "şifre çözüm anahtarı" olarak anılmaktadır. Özel anahtar bazen "gizli anahtar" olarak da adlandırılsa da, simetrik algoritmalarla karışıklığı önlemek amacıyla bu terim genellikle tercih edilmemektedir.

Açık anahtarlı şifreleme algoritmaları, özellikle veri güvenliğinin arttırılması, dijital imzaların oluşturulması ve kimlik doğrulama işlemleri gibi alanlarda geniş bir uygulama yelpazesine sahiptir. Bu teknolojinin sağladığı güvenlik ve esneklik, modern iletişim sistemlerinde önemli bir rol oynamakta ve birçok çevrimiçi hizmetin temelini oluşturmaktadır.

Yaygın olarak kullanılan asimetrik algoritmalar arasında RSA, Elliptic Curve Cryptography (ECC) ve Diffie-Hellman Key Exchange yer alır (Kodaz ve Botsalı, 2010; Okumuş, 2012; Schneier, 2007; Yerlikaya ve ark., 2006). Şekil 2-2 de algoritmanın yapısı görülmektedir.

Şekil 2-2: Asimetrik algoritma yapısı

Algoritmanın çalışması:

Açık anahtarlı (asimetrik) şifreleme algoritması, her kullanıcının bir açık ve bir gizli anahtar çiftine sahip olduğu bir yapıda çalışır. Bu sistemde, açık anahtar herkesin erişimine açıkken, gizli anahtar yalnızca sahibinde bulunur. Şifreleme işlemi, gönderici tarafından alıcının açık anahtarını kullanarak gerçekleştirilir ve bu işlem sonucunda elde edilen şifreli metin, güvenli olmayan bir ortamda bile iletilebilir. Alıcı taraf, yalnızca kendisine ait gizli anahtarı kullanarak bu şifreli metni çözebilir ve orijinal veriyi elde eder. Bu yapı, açık anahtarların genel erişilebilirliği sayesinde veri gönderenlerin güvenli bir anahtar paylaşımına ihtiyaç duymadığı avantajı sunar. Ayrıca, açık anahtar şifrelemesi, dijital imza gibi kimlik doğrulama yöntemleri için de kullanılarak veri bütünlüğünü ve kaynak doğruluğunu sağlamaktadır. Gizli anahtarın yalnızca alıcıda bulunması, şifre çözme işleminin yalnızca bu tarafça gerçekleştirilmesini garanti eder (Kodaz ve Botsalı, 2010; Yerlikaya ve ark., 2006).

2.4 Bazı Şifreleme Algoritmaları

Günümüzde hem simetrik hem de asimetrik şifreleme teknikleri, güvenli iletişim sağlamak ve veri hırsızlığı gibi tehditleri önlemek için yaygın olarak tercih edilmektedir. Şifreleme algoritmaları, dijital verilerin gizliliğini ve bütünlüğünü korumak amacıyla kullanılan önemli araçlardır. Bu algoritmalar, veri güvenliğini sağlamak için çeşitli

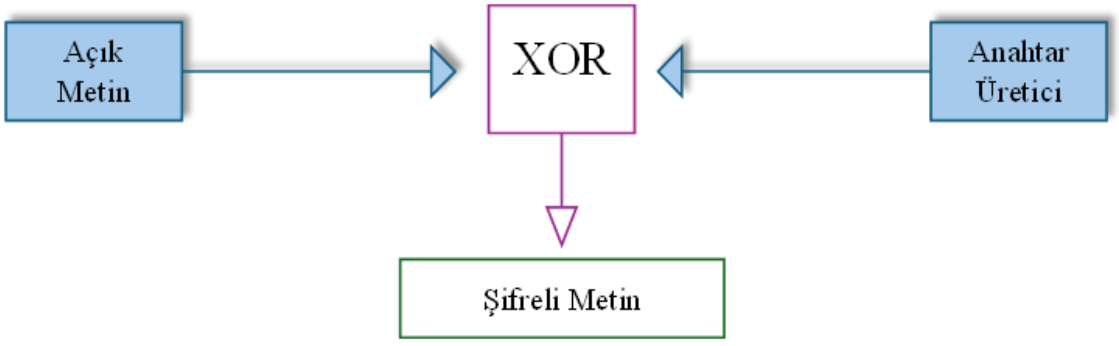
matematiksel yöntemler kullanarak veriyi karmaşık bir yapıya dönüştürmektedir (Beşkirli ve ark., 2019; Kodaz ve Botsalı, 2010; Topaloğlu ve ark., 2016). Bu algoritmaların nasıl kullanıldığını gösteren bazı örnekler aşağıda yer almaktadır.

2.4.1 Akan şifresi

Akış şifreleme algoritmaları, verileri şifrelemek için bir anahtar dizisi içerir ve XOR işlemine tabi tutulur. Bu anahtar dizisi, gizli bir anahtar kullanılarak bir anahtar üretici tarafından oluşturulur. Daha sonra, anahtar dizisi ve girdi mesajının her bir biti ayrı ayrı XOR işlemine tabi tutulur. Bu işlem, metni şifrelenmiş hale getirir. Akış şifreleme algoritmalarının güvenliği, anahtar dizisinin rastgele olmasına bağlıdır. Anahtar dizisindeki bitler tekrarlı olmamalı ve ölçülebilir bir düzende bulunmamalıdır. Aksi takdirde, şifreleme algoritması kırılmaya açık hale gelir (Paar ve Pelzl, 2010). Anahtar dizisi üretmek için iki temel yöntem kullanılır:

Tam Rastgele (True Random) Dizi: Bu yöntemde, dizideki her bit birbirinden bağımsız olarak üretilir. Bir örnek olarak yazı tura atışı verilebilir. Yazı tura atışında her iki olasılık da eşittir ve bir sonraki atış bir öncekinden etkilenmez (Paar ve Pelzl, 2010).

Pseudo Rastgele (Pseudo Random) Dizi: Bu yöntemde, dizinin her biti kendinden önce gelen bitlere bağlıdır. Aynı zamanda her bit kendinden sonra gelen biti de etkiler. Bu yöntemde, bir başlangıç değeri ve bir algoritma kullanılır. Başlangıç değeri, dizideki ilk biti belirler. Algoritma ise, her bir sonraki biti hesaplamak için kullanılır (Paar ve Pelzl, 2010). Şekil 2-3 te Akan şifre algoritma yapısı görülmektedir.



Şekil 2-3: Akan şifresi algoritma yapısı

$m_1 m_2 m_3$ ... m_x mesajımızı $a_1 a_2 a_3$ ... a_x anahtarı ile şifrelediğimizde $s_t = m_t \; XOR \; a_t$ olur.

2.4.1.1.*Örnek:* Açık metin ikilik tabanda "1011100010111" verilsin. Aynı ikilik tabanda ve uzunluğunda anahtar "1110111010111" seçilsin. Açık metni anahtar kullanarak şifreleyelim.

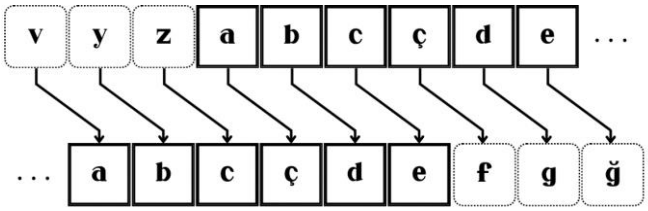
$$m_t = 1011100010111$$

$$a_t = 1110111010111$$

$$m_t \; XOR \; a_t = 0101011000000$$

2.4.2 Sezar şifresi

Sezar şifrelemesi hem yer değiştirme hem de harf değiştirme prensiplerini kullanan bir algoritmadır. Bu algoritmada alfabe, belirli bir sayıda ötelenir ve açık metindeki her harf, ötelenmiş alfabedeki karşılığıyla değiştirilir. Bu işlem, metnin şifrelenmesini sağlar. Şifreli metni tekrar açık metne dönüştürmek için ise öteleme işleminin tersi uygulanır. Bu işlem deşifre olarak adlandırılır (Stallings, 2006).



Şekil 2-4: Sezar şifresi gösterimi

Açık metin ve şifreli metindeki harflerin tamsayı değerlerini sırasıyla M ve S ile gösterelim. Bu durumda,

$$S \equiv M + 3(mod29),\ 0 \leq S \leq 28$$

ve

$$M \equiv S - 3(mod29),\ 0 \leq M \leq 28$$

olur (Paar ve Pelzl, 2010; Schneier, 2007; Stallings, 2006).

a	b	c	ç	d	e	f	g	ğ	h	ı	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

Çizelge 1: Türk alfabesinin mod 29'a göre tam sayı karşılıkları

2.4.2.1 ***Örnek:***

Sezar şifreleme metoduna göre Çizelge 1'e göre **"bulanık mantık"** cümlesini şifreleyelim. Burada her harfin alfabedeki karşılığı olan sayısal değeri alınır.

b	u	l	a	n	ı	k	m	a	n	t	ı	k
1	24	14	0	16	10	13	15	0	16	23	10	13

Bu tablodaki değerler üzerinde

$S \equiv M + 3(mod29), 0 \leq S \leq 28$ bağıntısı kullanıldığında; her harf üç karakter ötelenir ve

d	y	o	ç	p	k	n	ö	ç	p	v	k	n
4	27	17	3	19	13	16	18	3	19	26	13	16

"dyoçpknöçpvkn" şifreli metin elde edilir.

2.4.3 Değiştirme şifresi

0, … ,28'e kadar Çizelge1 deki harflerin tam sayı karşılıkları olsun.

$\mu: \{0,1,2, \dots ,28\} \rightarrow \{0,1,2, \dots ,28\}$ gibi tanımlı olan tüm permütasyonların kümesi P olsun. M_t açık mesajdaki harflerin tam sayı karşılığı ve S_t şifreli mesajdaki harflerin tam sayı karşılığı ise

$$e_\mu(M_t) = \mu(M_t) \text{ ve } d_\mu(S_t) = \mu^{-1}(S_{t)} \text{ olarak tanımlanır.}$$

2.4.3.1. ***Örnek:*** Çizelge 1'deki harflerin tam sayı karşılıklarına bağlı kalmak üzere anahtarımız μ ve μ nün her değeri sırasıyla metnimizin harflerine karşılık gelecek şekilde karşılaştırılır. μ nun harf karşılığındaki değer kadar öteleme yapılır.

$\mu = (14, 21, 11, 27, 7, 24, 19, 4, 28, 10, 1, 17, 0, 23, 16, 6, 12, 5, 26, 20, 15, 9, 8, 13, 3, 2, 25, 22, 18)$

olsun. "bulanık mantık" olarak verilen metni şifreleyelim.

$e_\mu(M_t) = \mu(M_t)$ fonksiyonuna göre Çizelge 2'deki alfabemiz ve μ ye göre karşılığını hesaplayalım.

Alfabe	**a**	**b**	**c**	**ç**	**d**	**e**	**f**	**g**	**ğ**	**h**	**ı**	**i**	**j**	**k**	**l**
µ	l	s	İ	y	g	u	p	d	z	ı	b	o	a	t	n
Alfabe	**m**	**n**	**o**	**ö**	**p**	**r**	**s**	**ş**	**t**	**u**	**ü**	**v**	**y**	**z**	
µ	f	j	e	v	r	m	h	ğ	k	ç	c	ü	ş	ö	

Çizelge 2: Anahtarların alfabedeki harf karşılıkları

Yeni şifreleme alfabemize göre girdiğimiz açık metnin şifreli karşılığı "sçnljbtfljkbt" şeklinde olur.

2.4.4 Vigenere şifresi

Vigenere şifreleme yöntemi, tek bir alfabe yerine birden fazla alfabe kullanarak metinleri gizlemeyi amaçlamaktadır. Vigenere şifrelemesi, düz metindeki her bir harfi şifrelerken farklı bir kaydırma miktarı kullanarak Sezar şifresine göre daha karmaşık bir yapı sağlar. Bu yöntem, özellikle uzun anahtar dizileri kullanıldığında, aynı harfin her defasında farklı bir harfe dönüşmesini sağlar ve böylece kripto analizini zorlaştırmaktadır. Şifreleme ve deşifreleme işlemi genellikle aşağıdaki adımlarla gerçekleştirilir:

- Adım1: Öncelikle bir anahtar kelime veya anahtar dizisi seçilir. Bu anahtar, açık metin kadar uzun değilse, açık metin uzunluğuna erişene kadar kendini tekrar eder.
- Adım2: Her bir açık metin harfi, anahtarın ilgili harfinin değeri kadar kaydırılır. Örneğin, "A" harfi 0, "B" harfi 1 değerine sahip olacak şekilde bir alfabe sıralaması yapılır.
- Adım3: Açık metindeki her bir harf, anahtar kelimenin karşılık gelen harfi kadar kaydırılarak yeni bir harfe dönüştürülür.

Şifreleme işlemi matematiksel olarak şöyle ifade edilebilir:
M açık metnindeki harflerin tamsayı karşılığı, K anahtar metnindeki harflerin tamsayı karşılığı ve S şifreli metnindeki harflerin tamsayı karşılığı ise

$$e_k(M) = M + K(mod29)$$

ve

$$d_k(S) = S - K(mod29)$$ şeklinde olur.

Çizelge 1 yardımıyla açık metin ve anahtar için alt alta tamsayı değerleri yazılır. K anahtarı M metninin her karakteriyle eşleştirildiğinde şifreleme işlemi sağlanmış olur. Eğer kullandığımız anahtar açık metnin karakter sayısından az ise anahtar tekrar edilerek eşleştirme sağlanır (Paar ve Pelzl, 2010; Stallings, 2006).

2.4.4.1 ***Örnek****:* Anahtar metnimiz "hüseyin" olsun. Anahtar kelimemizin her harfinin çizelge1 deki tam sayı karşılığı alınır. $K = (9,25,21,5,27,11,16)$ olur. Açık metnimiz "tamam bekliyorum" olsun. Metnin tam sayı karşılığı ise $X = (23,0,15,0,15,1,5,13,14,11,27,17,20,24,15)$ olur.

Belirlediğimiz anahtar 7 karakterli olduğu için 7'li gruplar halinde açık metinle birlikte alt alta yazılır. Alfabemiz 29 karakterli olduğu için $mod\ 29$'a göre toplamları alınır. Bu toplamlar,

X	23	0	15	0	15	1	5	13	14	11	27	17	20	24	15
K	9	25	21	5	27	11	16	9	25	21	5	27	11	16	9
e_k	3	25	7	5	13	12	21	22	10	3	3	15	2	11	24

olur. Oluşan e_k değeri Çizelge 1'e göre harflere dönüştürüldüğünde şifreli metin "çügekjsşıççmciu" olur.

2.4.5 RSA şifresi

RSA şifrelemesi, 1977 yılında Ron Rivest, Adi Shamir ve Leonard Adleman tarafından geliştirilen, günümüzde yaygın olarak kullanılan bir asimetrik şifreleme algoritmasıdır. Bu algoritma, güvenliği büyük asal sayılar üzerinde yapılan faktörizasyon probleminin zorluğuna dayanmaktadır. RSA, her kullanıcıya ait bir açık anahtar ve bir gizli anahtar çifti oluşturur. Açık anahtar, veriyi şifrelemek için kullanılırken, gizli anahtar yalnızca veriyi deşifre etmek için kullanılır. Bu özellik, güvenli bir iletişim sağlayabilmek için sadece açık anahtarın paylaşılmasına ve gizli anahtarın korunmasına olanak tanır. RSA algoritması, hem şifreleme hem de dijital imza işlemleri için yaygın olarak kullanılır ve özellikle internet üzerinden güvenli veri iletimi, kimlik doğrulama ve dijital imza

uygulamalarında önemli bir rol oynamaktadır. Algoritmanın güvenliği, kullanılan anahtar uzunluğuna bağlı olarak artmaktadır. RSA, teorik olarak güvenli olsada, anahtar uzunluğu arttıkça işlem süreleri de artmakta, bu da pratikte hız ve verimlilik sorunları yaratmaktadır. Bu nedenle, RSA'nın yanında başka hızlı şifreleme algoritmaları, özellikle veri şifrelemesi için tercih edilmektedir (Rivest ve ark., 1978; Yerlikaya ve Aslanyürek, 2019) .

Günümüzde RSA, internet üzerinden yapılan güvenli veri transferleri, dijital imza ve kimlik doğrulama gibi birçok alanda kritik bir rol oynamaktadır. Artan bilgisayar gücü ve gelişen kriptografik saldırılar göz önüne alındığında, RSA'nın uzun vadede güvenliğini korumak için kuantum bilgisayarlara karşı dirençli yeni algoritmalar geliştirilmesi üzerinde çalışmalar devam etmektedir (Yerlikaya ve Aslanyürek, 2019).

Algoritmanın çalışma prensibi şu şekildedir:
Alıcı Anahtar Oluşturma:

- İki tane farklı ve aynı uzunlukta asal sayı p ve q seçilir
- $n = p \,.\, q$ değeri hesaplanır.
- $\Phi(n) = (p-1).(q-1)$ değeri hesaplanır.
- $2 < e < \varphi(n)$ ve $ebob(\varphi(n), e) = 1$ olacak şekilde bir e tamsayısı seçer.
- Açık anahtar (e) için $\Phi(n)$ ile birbirinin tersi olacak bir tamsayı seçer.
- Gizli anahtar (d) için $d.e \equiv 1\ (mod\ \Phi(n))$ eşitliği sağlanacak şekilde ve $1 < d < \varphi(n)$ aralığında d değeri hesaplar.
- Alıcının açık anahtarı (n, e) ve özel anahtarı d'dir.

Şifreleme:

- Öncelikle alıcının açık anahtarı (n, e)'yi alır.
- m mesajını $[0, n-1]$ aralığında yazar. Şifreli metni (E) $E \equiv me\ (mod\ n)$ hesaplar. Ve şifreli metni elde eder.

Deşifreleme:

- Alıcı d gizli anahtarını kullanarak $E \equiv c^d\ (mod\ n)$ değerini hesaplar.
- Şifreli metin (c) elde edilir.
- Orijinal mesajı (m) şu şekilde hesaplayın: $m \equiv c^d\ (mod\ n)$ işlemini uygulayarak metne ulaşılır.

2.4.5.1. *Örnek*

Ali isimli kişi, Bekir isimli kişiye "Merhaba" mesajını göndermek istiyor. Yukarıdaki adımları kullanarak mesajın nasıl şifrelenip çözüleceği aşağıdaki adımlarda gösterilmiştir.

1. Anahtar Oluşturma:

$p = 61, q = 53$

$n = pxq = 61x53 = 3233$

$\varphi(n) = (p-1)(q-1) = 60x52 = 3120$

- **Açık Anahtar:** $e = 17$ (Aralarındaki asal olan bir sayı)
- **Özel Anahtar:** $d \equiv e^{-1}\ mod\ \varphi(n)$, burada $d = 2753$

2. Mesaj Şifreleme:

- **Mesaj:** "Merhaba"
- **ASCII Değerleri:** [77, 101, 114, 104, 97, 98, 97]
- **Şifreleme Formülü:**

$$c = m^e\ mod\ n$$

- **Şifreli Değerler:**
  - $M\ (77)$: $c = 1616$
  - $e\ (101)$: $c = 2285$
  - $r\ (114)$: $c = 1820$
  - $h\ (104)$: $c = 890$
  - $a\ (97)$: $c = 2527$
  - $b\ (98)$: $c = 1597$
  - $a\ (97)$: $c = 2527$

Sonuç olarak, şifreli metin: $c = [1616,2285,1820,890,2527,1597,2527]$

3. Mesajın Şifre Çözümü:

- **Şifre Çözümleme Formülü:**

$$m = c^d\ mod\ n$$

- Her bir şifreli sayı için:
  - $\boldsymbol{c = 1616}$: $m = 77\ (ASCII: M)$
  - $\boldsymbol{c = 2285}$: $m = 101\ (ASCII: e)$

- $\boldsymbol{c = 1820}$: $m = 114$ $(ASCII: r)$
- $\boldsymbol{c = 890}$: $m = 104$ $(ASCII: h)$
- $\boldsymbol{c = 2527}$: $m = 97$ $(ASCII: a)$
- $\boldsymbol{c = 1597}$: $m = 98$ $(ASCII: b)$
- $\boldsymbol{c = 2527}$: $m = 97$ $(ASCII: a)$

Bekir, "1616,2285,1820,890,2527,1597,2527" mesajını **"Merhaba"** olarak bulmuştur.

3. ESNEK KÜMELER

Bu bölümde, Molodstov (1999) tarafından tanımlanan esnek kümeleri, temel tanım ve temel teoremleri tanıtacağız. Bu bölümdeki tanım ve teoremler (Maji ve ark., 2003), (Çağman ve Enginoğlu, 2010), (Çağman, 2024), (John, 2021) kaynaklarından derlenmiştir. Teoremlerin ispatları kaynaklarında ispat edildiği için burada verilmeyecektir.

Tanım 3.1: U bir evrensel küme, U'nun kuvvet kümesi $P(U)$ ve E parametreler kümesi olsun. $f: E \to P(U)$ olmak üzere

$$F = \{(e, f(e) : e \in E\}$$

ikililer kümesine U üzerinde tanımlı bir **esmek küme** denir. Burada verilen küme değerli f fonksiyonuna F esnek kümesinin **yaklaşım fonksiyonu**, $f(e)$ değer kümesine **e-yaklaşım kümesi** denir. $f(e) = \emptyset$ ise $(e, f(e))$ esnek kümenin elemanı olarak gösterilmez.

Esnek kümeler genellikle *F, G, H*, … gibi büyük harflerle ve yaklaşım fonksiyonları sırasıyla *f, g, h*, … gibi onların küçük harfleriyle gösterilir.

Esnek kümelerin esnekliği, U evrensel kümesindeki terimleri niteleyen E parametre kümesindeki parametrelere göre kişiden kişiye değişecek şekilde kümelere ayrılmasıdır. Burada, U kümesindeki terimleri iyi tanıyan ve f yaklaşım fonksiyonunun görüntü kümelerine karar verenlere **karar vericiler** denir.

Örnek 3.2. $U = \{u_1, u_2, u_3\}$ isimli filmler kümesi, E parametre kümesi $E = \{sıkıcı, güzel, komik, uzun\}$ olsun. Bir X karar vericisine göre f: $E \to P(U)$ fonksiyonu

$$f(sıkıcı) = \{u_1, u_2\},\ f(güzel) = \{u_1, u_3\},\ f(komik) = \emptyset\ ,\ f(uzun) = \{u_2\}$$

şeklinde tanımlı ise *F* esnek kümesi aşağıdaki gibi yazılır.

$$F = \{(sıkıcı, \{u_1, u_2\}), (güzel, \{u_1, u_3\}), (uzun, \{u_2\})\}$$

Burada $f(komik) = \emptyset$ olduğu için *F* esnek kümesinde (*komik*, $\emptyset$) elemanının küme içinde yazılmadığına dikkat ediniz. Başka bir *Y* karar vericisine göre eğer $g: E \to P(U)$ fonksiyonu

$$g(sıkıcı) = \{u_1, u_3, \},\ g(güzel) = \{u_2, u_3\},\ g(komik) = \{u_2\}\ ,\ g(uzun) = \{u_1\}$$

şeklinde tanımlı ise *G* esnek kümesi aşağıdaki gibi olur.

$$G = \{(sıkıcı, \{u_1, u_3\}), (güzel, \{u_2, u_3\}), (komik, \{u_2\}), (uzun, \{u_1\})\}$$

Tanım 3.3. *F*, *U* üzerinde bir esnek küme olsun. Eğer $\forall e \in E$ için $f(e) = \emptyset$ oluyorsa *F* esnek kümesine **boş esnek küme** denir ve ϕ ile gösterilir.

Tanım 3.4. *F*, *U* üzerinde bir esnek küme olsun. Eğer $\forall e \in E$ için $f(e) = U$ oluyorsa *F* esnek kümesine **evrensel esnek küme** denir ve $\tilde{E}$ ile gösterilir.

Teorem 3.5. *U* üzerinde tanımlı,

i. Boş olan esnek küme tektir.
ii. Evrensel esnek küme tektir.

İspat. Tanım 3.3 ve 3.4'ten açıktır.

Tanım 3.6. *U* üzerinde tanımlı *F* ve *G* esnek kümeleri verilsin. Eğer $\forall e \in E$ için

$$f(e)\ = g(e)$$

oluyorsa *F* ve *G* esnek kümelerine **eşit esnek kümeler** denir ve *F* = *G* ile gösterilir.

Tanım 3.7. *U* üzerinde tanımlı F ve *G* esnek kümeleri verilsin. Bu durumda $\forall e \in E$ için

$$f(e) \subseteq g(e)$$

oluyorsa F esnek kümesine G esnek kümesinin bir **esnek alt kümesidir** denir ve $F_A \cong F_B$ şeklinde ifade edilir.

Teorem 3.8. *U* üzerinde tanımlı *F*, *G* ve *H* esnek kümeleri verilsin. Bu durumda;

i. $F \cong \tilde{E}$
ii. $\phi \cong F$
iii. $F \cong F$
iv. $F \cong G \ ve \ G \cong H \Rightarrow F \cong H$

Tanım 3.9. *U* üzerinde tanımlı bir *F* esnek kümesi verilsin. **F esnek kümesinin tümleyeni** F^o ile gösterilir ve yaklaşım fonksiyonu $\forall e \in E$ için aşağıdaki gibi tanımlanır.

$$f^o(e) = U - f(e)$$

Tanım 3.10. *U* üzerinde tanımlı F ve *G* esnek kümeleri verilsin. F ve G **esnek kümlerinin birleşimi** $F \tilde{\cup} G$ şeklinde gösterilir ve $F \tilde{\cup} G = H$ ise yaklaşım fonksiyonu $\forall e \in E$ için aşağıdaki gibi tanımlanır.

$$h(e) = f(e) \cup g(e)$$

Teorem 3.11. *U* üzerinde tanımlı *F*, *G* ve *H* esnek kümeleri verilsin. Bu durumda;

i. $F \tilde{\cup} F = F$
ii. $F \tilde{\cup} \phi = F$
iii. $F \tilde{\cup} \tilde{E} = \tilde{E}$
iv. $F \tilde{\cup} F^{\circ} = \tilde{E}$
v. $F \tilde{\cup} G = G \tilde{\cup} F$
vi. $(F \tilde{\cup} G) \cup H = F \tilde{\cup} (G \tilde{\cup} H)$

Tanım 3.12. *U* üzerinde tanımlı *F* ve *G* esnek kümeleri verilsin. F ve G **esnek kümlerinin kesişimi** $F \widetilde{\cap} G$ şeklinde gösterilir ve $F \widetilde{\cap} G = H$ ise yaklaşım fonksiyonu $\forall\, e \in E$ için aşağıdaki şekilde tanımlanır.

$$h(e) = f(e) \cap g(e)$$

Teorem 3.13. *U* üzerinde tanımlı *F*, *G* ve *H* esnek kümeleri verilsin. Bu durumda;

i. $F \widetilde{\cap} F = F$
ii. $F \widetilde{\cap} \phi = \phi$
iii. $F \widetilde{\cap} \tilde{E} = F$
iv. $F \widetilde{\cap} F^o = \phi$
v. $F \widetilde{\cap} G = G \widetilde{\cap} F$
vi. $(F \widetilde{\cap} G) \widetilde{\cap} H = F \widetilde{\cap} (G \widetilde{\cap} H)$
vii. $F \widetilde{\subseteq} G \Rightarrow F \widetilde{\cup} G = F_B$ ve $F \widetilde{\cap} G = F$

Teorem 3.14. *U* üzerinde tanımlı *F*, *G* ve *H* esnek kümeleri verilsin. Bu durumda;

i. $F \widetilde{\cup} (G \widetilde{\cap} H) = (F \widetilde{\cup} G) \widetilde{\cap} (F \widetilde{\cup} H)$
ii. $F \widetilde{\cap} (G \widetilde{\cup} H) = (F \widetilde{\cap} G) \widetilde{\cup} (F \widetilde{\cap} H)$

Örnek 3.15. Evrensel küme $U = \{u_1, u_2, u_3, u_4, u_5\}$ ve parametreler kümesi $E = \{e_1, e_2, e_3, e_4\}$ olmak üzere *U* üzerinde tanımlı iki esnek küme

$$F = \{(e_1, \{u_2, u_4\}), (e_2, \{u_1, u_3\})\} \text{ ve } G = \{(e_2, \{u_1, u_2\}), (e_3, \{u_1, u_4\}), (e_4, U)\}$$

olsun. Bu durumda aşağıdaki esnek kümelerini gösterebiliriz.

1. $F^o = \{(e_1, \{u_1, u_3, u_5\}), (e_2, \{u_2, u_4, u_5\}), (e_3, U), (e_4, U)\}$
2. $F \widetilde{\cup} G = \{(e_1, \{u_2, u_4\}), (e_2, \{u_1, u_2, u_3\}), (e_3, \{u_1, u_4\}), (e_4, U)\}$
3. $F \widetilde{\cap} G = \{(e_2, \{u_1\})\}$
4. $(F \widetilde{\cup} G)^{\circ} = \{(e_1, \{u_1, u_3, u_5\}), (e_2, \{u_4, u_5\}), (e_3, \{u_2, u_3, u_5\})\} = F^o \widetilde{\cap} G^o$
5. $(F \widetilde{\cap} G)^{\circ} = \{(e_1, U), (e_2, \{u_2, u_3, u_4, u_5\}), (e_3, U), (e_4, U)\} = F^o \widetilde{\cup} G^o$

4. DENLİK BAĞINTISI ve DENKLİK SINIFLARI

Bağıntı, küme veya kümelerin elemanları arasında bağ kurmayla uğraşır. Burada, kullanacağımız bir küme üzerinde tanımlı olan denklik bağıntını ve özelliklerini vereceğiz. Bu bölüm (Çağman, 2024) kaynağından derlenmiştir.

Tanım 4.1. $i = 1, 2, \dots, n$ ve $2 \leq n$ olmak üzere *n* tane A_i kümeleri verilsin. $A_1 \times A_2 \times \dots \times A_n$ kartezyen çarpımının her alt kümesine $A_1 \times A_2 \times \dots \times A_n$ üzerinde tanımlı bir ***n*-li bağıntı** denir ve bağıntılar genelde *α, β, γ, ...* gibi Yunan harfleriyle gösterilecektir.

Burada sadece **ikili bağıntı** üzerinde duracağız. Bu nedenle, aksi belirtilmedikçe ikili bağıntı yerine kısaca sadece **bağıntı** diyeceğiz.

Buna göre, AxB kartezyen çarpımının alt kümesi olan bir β bağıntısına $A\ x\ B$ üzerinde veya *A*'dan *B*'ye bir bağıntı denir ve

$$\beta = \{(x, y)\ \in AxB : p(x, y)\}$$

biçiminde yazılır. Eğer $B = A$ ise *A*'dan *A*'ya her bağıntıya *A* üzerinde bir bağıntı veya kısaca *A*'da bir bağıntı denir.

Örnek 4.2. $A = \{1,2\}$ ve $B = \{1,2,3\}$ olsun. Bu durumda,
$\alpha = \{(x, y) : x < y\} = \{(1,2), (1,3), (2,3)\}$ bağıntısı *A*'dan *B*'ye tanımlıdır.
$\beta = \{(x, y) : y = x - 1\} = \{(2,1), (3,2)\}$ bağıntısı *B*'dan *A*'ye tanımlıdır.
$\gamma = \{(x, y) : x \neq y\} = \{(1,2), (2,1)\}$ bağıntısı *A* üzerinde tanımlıdır.
$\delta = \{(x, y) : x + y = 4\} = \{(1,3), (2,2), (3,1)\}$ bağıntısı *B* üzerinde tanımlıdır.

Tanım 4.3. Bir *A* kümesi üzerinde tanımlı bir β bağıntısına;

a) $\forall x \in A$ için $(x, x) \in \beta$ oluyorsa yansıyan denir,
a*) ya da $I_A \subseteq \beta$ ise **yansıyan** denir.
b) $\forall (x, y) \in \beta$ ise $(y, x) \in \beta$ oluyorsa **simetrik** denir.

c) $\forall(x,y) \in \beta,\ x \neq y$ ise $(y,x) \notin \beta$ oluyorsa **ters simetrik** denir,

c*) ya da $\forall(x,y),(y,x) \in \beta$ ise $x = y$ oluyorsa ters simetriktir (1).

d) $\forall(x,y),(y,z) \in \beta$ ise $(x,z) \in \beta$ oluyorsa **geçişken** denir.

(1): c ve c* birbirine denk iki ters simetri tanımıdır. Bu tanımların denk olduklarını $p \Longrightarrow p \equiv \neg\, p \vee q$ denkliğini kullanarak gösterelim.

$$
\begin{aligned}
[(x,y) \in \beta \ \wedge\ x \neq y] \Longrightarrow (y,x) \notin \beta &\equiv \neg\,[(x,y) \in \beta \ \wedge\ x \neq y] \vee (y,x) \notin \beta \\
&\equiv (x,y) \notin \beta \vee (x = y) \vee (y,x) \notin \beta \\
&\equiv (x,y) \notin \beta \vee (y,x) \notin \beta \vee (x = y) \\
&\equiv \neg\,[(x,y) \in \beta \ \wedge\ (y,x) \in \beta] \vee (x = y) \\
&\equiv [(x,y) \in \beta \ \wedge\ (y,x) \in \beta] \Longrightarrow (x = y)
\end{aligned}
$$

Tanım 4.4. Yansıyan, simetrik ve geçişken olan her bağıntıya bir **denklik bağıntısı** denir. Bir β denklik bağıntısı için $(x,y) \in \beta$ ise x elemanı β bağıntısı ile y elemanına denktir denir ve $x\beta y$ şeklinde gösterilir.

Tanım 4.5. A kümesinde β bir denklik bağıntısı olsun. A'nın bir x elemanına β ile denk olan elemanların kümesine x'nin **denklik sınıfı** denir ve

$$x_\beta = \{y : (x,y) \in \beta\}$$

şeklinde yazılır. Bu tanımdan

$$y \in x_\beta \Leftrightarrow (x,y) \in \beta \qquad 3.2$$

denkliği elde edilir. Denklik sınıfları için x_β yerine literatürde $[x], [x]_\beta, \bar{x}_\beta$ ve $\bar{x}$ gösterimleri de kullanılır.

Örnek 4.6. $\mathbb{Z}$ üzerinde $\beta = \{(x,y) : x \equiv y (mod\ 3)\}$ bir denklik bağıntısı olduğunu gösterdikten sonra denklik sınıflarını bulalım.

$\forall x \in \mathbb{Z}$ için $x \equiv x (mod\ 3) \Rightarrow (x,x) \in \beta$ olduğundan β yansıyandır;

$\forall (x,y) \in \beta \Rightarrow x \equiv y (mod\ 3)$,

$\Rightarrow y \equiv x (mod\ 3)$,

$\Rightarrow (y,x) \in \beta$ olduğundan β simetriktir.

$\forall (x,y),(y,z) \in \beta \Rightarrow x \equiv y (mod\ 3), y \equiv z (mod\ 3)$,

$$\Rightarrow x + y \equiv y + z(mod\ 3),$$
$$\Rightarrow (x + y) - y \equiv (y + z) - y(mod\ 3),$$
$$\Rightarrow x \equiv z(mod\ 3),$$
$$\Rightarrow (x, z) \in \beta \text{ olduğundan } \beta \text{ geçişkendir.}$$

O halde β yansıyan, simetrik ve geçişken olduğu için bir denklik bağıntısıdır. Şimdi de her $x \in \mathbb{Z}$ için $x_\beta = \{y : y \equiv x(mod\ 3)\}$ denklik sınıflarını bulalım.

$$0_\beta = \{y : y \equiv 0(mod\ 3)\} = \{\dots, -6, -3, 0, 3, 6, \dots\}$$
$$1_\beta = \{y : y \equiv 1(mod\ 3)\} = \{\dots, -4, -1, 1, 4, 7, \dots\}$$
$$2_\beta = \{y : y \equiv 2(mod\ 3)\} = \{\dots, -5, -2, 2, 5, 8, \dots\}$$

Şeklinde $0_3, 1_3$ ve 2_3 olarak elde edilir. 3_3 ve sonrası için denklik sınıflarını bulmaya devam edersek:

$$3_3 = \{y : y \equiv 3(mod\ 3)\} = \{\dots, -6, -3, 0, 3, 6, \dots\} = 0_3$$
$$4_3 = \{y : y \equiv 4(mod\ 3)\} = \{\dots, -4, -1, 1, 4, 7, \dots\} = 1_3$$

Görüldüğü gibi bulunan denklik sınıfları yeni bir denklik sınıfı değil, sadece bulunan denklik sınıflarının bir tekrarı olarak ortaya çıkar.

Teorem 4.7. β, A kümesinde bir denklik bağıntısı olsun. $\forall x, y \in A$ olmak üzere x_β ve y_β denklik sınıfları için aşağıdakiler doğrudur:

i. $\forall x \in A$ için $x \in x_\beta$
ii. $(x, y) \in \beta \Leftrightarrow x_\beta = y_\beta$
iii. $x_\beta \neq y_\beta \Leftrightarrow x_\beta \cap y_\beta = \emptyset$

İspat:

I. β yansıyan olduğundan $\forall x \in A$ için $(x, x) \in \beta$ olur. Denklik sınıfı tanımından $x \in x_\beta$ çıkar.

II. $(\Rightarrow) : x\beta = y\beta \Rightarrow (x, y) \in \beta$ olduğunu gösterelim;

$x_\beta = y_\beta \quad \Rightarrow z \in x_\beta, z \in y_\beta,$ (Kümelerde eşitlik)

$\Rightarrow (z, x) \in \beta, (z, y) \in \beta,$ (Denklik sınıfı tanımı)

$\Rightarrow (x,z) \in \beta, (z,y) \in \beta,$ (β simetrik)

$\Rightarrow (x,y) \in \beta,$ (β geçişken)

$(\Leftarrow): (x,y) \in \beta \Rightarrow x_\beta = y_\beta$ olduğunu gösterelim;

$z \in x_\beta \Rightarrow (z,x) \in \beta,$ (Denklik sınıfı tanımı)

$\Rightarrow (z,x) \in \beta, (x,y) \in \beta,$ (Verilen Şart)

$\Rightarrow (z,y) \in \beta,$ (β geçişken)

$\Rightarrow z \in y_\beta,$ (Denklik sınıfı tanımı)

Benzer şekilde $z \in y_3 \Rightarrow z \in x_\beta$ gösterilirse $z \in y_\beta$ olacağından $x_\beta = y_3$ elde edilir.

III. Burada $p \equiv q \equiv \neg q \Rightarrow \neg p$ denkliğini kullanarak önermelerin karşıt terslerini ispat edelim.

$(\Rightarrow): x_\beta \neq y_3 \Rightarrow x_\beta \cap y_3 = \emptyset$ önermesi yerine $x_\beta \cap y_3 \neq \emptyset \Rightarrow x_\beta = y_3$ önermesini ispat edelim.

$x_\beta \cap y_3 \neq \emptyset \Rightarrow z \in x_\beta \cap y_3, \exists z \in A,$ (∩'in özelliği)

$\Rightarrow z \in x_\beta \wedge z \in y_3,$ (∩'in tanımı)

$\Rightarrow (z,x) \in \beta \wedge (z,y) \in \beta,$ (Denklik sınıfı tanımı)

$\Rightarrow (x,z) \in \beta \wedge (z,y) \in \beta,$ (β simetrik)

$\Rightarrow (x,y) \in \beta,$ (β geçişken)

$\Rightarrow x_\beta = y_3,$ (iii)

$(\Leftarrow): x_\beta \cap y_3 = \emptyset \Rightarrow x_\beta \neq y_3$ yerine $x_\beta = y_3 \Rightarrow x_\beta \cap y_3 \neq \emptyset$ önermesini ispat edelim.

$x_\beta = y_3 \Rightarrow z \in x_\beta \wedge z \in y_3 \exists z \in A,$ (Kümelerde eşitlik)

$\Rightarrow z \in x_\beta \cap y_3, \exists z \in A,$ (∩'in tanımı)

$\Rightarrow x_\beta \cap y_3 \neq \emptyset,$ (∩'in özelliği)

Tanım 4.8. *A* kümesinde β bir denklik bağıntısı oldu. *A* kümesinde β'nın oluşturduğu denklik sınıflarının kümesine *A* kümesinin β bağıntısına göre bölüm kümesi denir ve aşağıdaki gibi yazılır.

$$A/\beta = \{x_\beta : x \in A\}$$

Örnek 4.9. Örnek 4.13'da $\mathbb{Z}$ üzerinde $\beta = \{(x, y) : x \equiv y(mod\ 3)\}$ denklik bağıntının denklik sınıfları:

$$0_\beta = \{\dots, -6, -3, 0, 3, 6, \dots\} = -3_\beta = 3_\beta = -6_\beta = 6_\beta = \cdots,$$
$$1_\beta = \{\dots, -4, -1, 1, 4, \dots\} = -1_\beta = 4_\beta = -7_\beta = 7_\beta = \dots$$
$$2_\beta = \{\dots, -5, -2, 2, 5, \dots\} = -3_\beta = 5_\beta = -5_\beta = -8_\beta = 8_\beta = \dots$$

Yani, $\mathbb{Z}$ kümesinin β denklik bağıntısına göre bölüm kümesi aşağıdaki şekilde ifade edilebilir:

$$\mathbb{Z}/\beta = \{\{\dots, -6, -3, 0, 3, 6, \dots\}, \{\dots, -4, -1, 1, 4, \dots\}, \{\dots, -5, -2, 2, 5, \dots\}\}$$

olarak bulunur. Bölüm kümeleri eşit denklik sınıflarından birer temsilci alarak aşağıdaki gibi birbirine eşit kümelerden biri ile ifade edilebilir.

$$\mathbb{Z}/\beta = \{-3_\beta, -2_\beta, -1_\beta\} = \{0_\beta, 1_\beta, 2_\beta\} = \{3_\beta, 4_\beta, 5_\beta\} = \dots$$

Teorem 4.10. Bir kümenin bir denklik bağıntısına göre bölüm kümesi bu kümenin bir ayrışımıdır. Yani, bir A kümesi üzerinde tanımlı β bir denklik bağıntısı ise $\forall x, y \in A$ için aşağıdaki ayrışım şartları doğrudur:

i. $\forall x \in A$ için $x_\beta \neq \emptyset$,

ii. $\forall x, y \in A$ için ya $x_\beta = y_\beta$ ya da $x_\beta \cap y_\beta = \emptyset$,

iii. $\bigcup_{x \in A} x\beta = A$.

İspat:

I. Teorem 4.14-i)'den $x \in x_\beta$ olduğundan $x_\beta \neq \emptyset$ çıkar.

II. $\forall x, y \in A$ olmak üzere bir y_β denklik sınıfı için ya $x \in y_\beta$ ya da $x \notin y_\beta$ olur. Şimdi bu iki durumu sırasıyla irdeleyelim:

$x \in y_\beta \Leftrightarrow (x, y) \in \beta$ (Denklik sınıfı tanımı)

$\Leftrightarrow x_\beta = y_\beta$ (Teorem 4.14-iii)

$x \notin y_\beta \Leftrightarrow \neg (x \in y_\beta)$ (¬'nin tanımı)

$\Leftrightarrow \neg$ (x, y) ∈ β (Denklik sınıfı tanımı)

$\Leftrightarrow \neg (x_\beta = y_\beta)$ (Teorem 4.14-iii)

$\Leftrightarrow x_\beta \neq y_\beta$ (¬'nin tanımı)

$\Leftrightarrow x_\beta \cap y_\beta = \emptyset$ (Teorem 4.14-iv)

III. Bu ispat için $A \subseteq \bigcup_{x \in A} x\beta$ ve $\bigcup_{x \in A} x\beta \subseteq A$ olduklarını gösterelim.

$x \in A \Leftrightarrow (x, x) \in \beta,$ (β yansıyan)

$\Leftrightarrow x \in x\beta,$ (Denkliğin tanımı)

$\Rightarrow x \in \bigcup_{x \in A} x\beta,$ (∪'ın özelliği)

$y \in \bigcup_{x \in A} x_\beta \Leftrightarrow \exists x \in A, y \in x_\beta,$ (∪'ın özelliği)

$\Leftrightarrow (y, x) \in \beta$ (Denkliğin tanımı)

$\Rightarrow x \in A$ (Denkliğin tanımı)

Örnek 4.11. $\mathbb{Z}$ 'de $\beta = \{(x, y) : x \equiv y (mod\ 3)\}$ denklik bağıntısına göre

$$0_\beta = \{\ldots, -6, -3, 0, 3, \ldots\} = \{3x : x \in \mathbb{Z}\} = 3\mathbb{Z}$$
$$1_\beta = \{\ldots, -4, -1, 1, 4, \ldots\} = \{3x + 1 : x \in \mathbb{Z}\} = 3\mathbb{Z} + 1$$
$$2_\beta = \{\ldots, -5, -2, 2, 5, \ldots\} = 3x + 2 : x \in \mathbb{Z}\} = 3\mathbb{Z} + 2$$

denklik sınıflarından oluşan $\mathbb{Z}/\beta = \{0_\beta, 1_\beta, 2_\beta\}$ bölüm kümesi $\mathbb{Z}$ tamsayılar kümesinin bir ayrışımdır. Gerçekten,

1. $0_\beta \neq \emptyset, 1_\beta \neq \emptyset, 2_\beta \neq \emptyset,$
2. $0_\beta \cap 1_\beta = \emptyset, 0_\beta \cap 2_\beta = \emptyset, 1_\beta \cap 2_\beta = \emptyset,$
3. $0_\beta \cup 1_\beta \cup 2_\beta = 3\mathbb{Z} \cup (3\mathbb{Z} + 1) \cup (3\mathbb{Z} + 2) = \mathbb{Z}.$

Teorem 4.12. $\{A_i\}_{i\in I}$ ailesi, A kümesinin bir ayrışımı ise ailenin her bir elemanı denklik sınıfı olacak biçimde A kümesinde bir denklik bağıntısı tanımlanır. Yani Teorem 4.18'in tersi doğrudur.

İspat. $\{A_i\}_{i\in I}$ ailesi, A kümesinin bir ayrışımı ise

i. Her $i \in I$ için $A_i \neq \emptyset$
ii. Her $i,j \in I, i \neq j$ için $A_i \cap A_i = \emptyset$ ve
iii. $A = \bigcup_{i\in I} A_i$

Ayrışım şartları sağlanır. Bu şartlar altında A kümesinde tanımlı

$$\beta = \{(x,y) : x,y \in A_i, \exists i \in I\}$$

bağıntısının bir denklik bağıntısı olduğunu gösterelim. Buna göre:

$$\begin{aligned} \forall x \in A \Rightarrow x &\in \bigcup_{i\in I} A_i\,, \quad (A = \bigcup_{i\in I} A_i\,) \\ &\Rightarrow x \in A_i, \exists i \in I, && \text{(Aile birleşiminin tanımı)} \\ &\Rightarrow (x,x) \in \beta, && (\beta\text{'nın tanımı}) \end{aligned}$$

olduğundan ***β* yansıyandır.**

$$\begin{aligned} \forall (x,y) \in \beta &\Rightarrow x,y \in A_i, \exists i \in I, && (\beta\text{'nın tanımı}) \\ &\Rightarrow (y,x) \in \beta, && (\beta\text{'nın tanımı}) \end{aligned}$$

olduğundan ***β* simetriktir.**

$$\begin{aligned} \forall (x,y),(y,z) \in \beta \quad &\Rightarrow \quad x,y \in A_i,,y,z \in A_j, \exists i,j \in I, && (\beta\text{'nın tanımı}) \\ &\Rightarrow \quad A_i = A_j,, \\ &\Rightarrow \quad x,z \in A_i, \exists i \in I, \\ &\Rightarrow \quad (x,z) \in \beta, && (\beta\text{'nın tanımı}) \end{aligned}$$

olduğundan ***β* geçişkendir.**

5. AYRIŞIMLI ESNEK KÜMELER

Bu bölümde, yeni bir kavram olarak ayrışımlı esnek kümeleri tanımlayacağız.

Tanım 5.1. *U* bir evrensel olsun. Parametre kümesi de *U* olan ve *U* üzerinde tanımlı bir $F = \{(u, f(u) : u \in U\}$ esnek kümesine, *U* üzerinde bir **ayrışımlı esnek küme** denir ancak ve ancak *f* yaklaşım fonksiyonunun görüntülerinden oluşan küme ailesi *U* kümesinin bir ayrışımıdır, yani *f* yaklaşım fonksiyonunun görüntülerinden oluşan kümeler aşağıdaki şartları sağlar.

a. $\forall u \in U,\ f(u) \neq \emptyset$
b. $\forall u, v \in U,\ f(u) = f(v)$ veya $f(u) \cap f(v) = \emptyset$
c. $\bigcup_{u \in U} f(u) = U$

Teorem 5.2. Bir *U* kümesi üzerinde tanımlı bir *F* ayrışımlı esnek kümenin *f* yaklaşım fonksiyonunun görüntülerinden oluşan küme ailesi *U* kümesinin bir ayrışımı ise ailenin her elemanı bir denklik sınıfı olacak biçimde *U* kümesinde bir denklik bağıntısı tanımlanır.

İspat: Teorem 4.12'nin ispatıyla hemen hemen aynıdır

Teorem 5.3. *U* üzerinde tanımlı bir $F = \{(u, f(u) : u \in U\}$ ayrışımlı esnek kümesine verilsin. Buna göre;

i. $\forall u \in U$ için $u \in f(u)$
ii. $u \in f(v) \Leftrightarrow v \in f(u)$
iii. $u_1, u_2 \in f(v) \Leftrightarrow f(u_1) = f(u_2)$
iv. $f(u) \neq f(v) \Leftrightarrow f(u) \cap f(v) = \emptyset$

İspat: Teorem 4.7'nin ispatıyla hemen hemen aynıdır.

***Örnek* 5.4.** $U = \{u_1, u_2, u_3, u_4, u_5, u_{6,}, u_7, u_8, u_9, u_{10}\}$ bir iş yerindeki görevlilerin kümesi olsun. *U* üzerinde farklı ayrışımlı esnek kümeler tanımlayalım.

1. Her görevlimin işe geldiğini kabul edersek, çalışanların işe beraber geldikleri gruplara göre bir *F* ayrışımlı esnek kümesinin yaklaşım fonksiyonu

$$f: U \to P(U),\ \ f(u) = \{v \in U: u \text{ ile beraber işe gelenler.}\}$$

şeklinde verilsin. Eğer,

- u_1 görevlisi işe özel arabasıyla yalnız gelmişse

$$f(u_1) = \{u_1\}$$

- u_4, u_6, u_7, u_8, u_9 ve u_{10} görevlileri servisle beraber gelmişse

$$f(u_4) = f(u_6) = f(u_7) = f(u_8) = f(u_9) = f(u_{10}) = \{u_4, u_6, u_7, u_8, u_9, u_{10}\}$$

- u_2, u_3 ve u_5 görevlileri yürüyerek beraber gelmişse

$$f(u_1) = f(u_2) = f(u_3) = f(u_5) = \{u_1, u_2, u_3, u_5\}$$

şeklinde tanımlanırsa *U* üzerinde *F* ayrışımlı esnek küme aşağıdaki şekilde yazılır.

$$F = \left\{ \begin{array}{c} (u_1, \{u_1\}), (u_2, \{u_1, u_2, u_3, u_5\}), (u_3, \{u_1, u_2, u_3, u_5\}), \\ (u_5, \{u_1, u_2, u_3, u_5\}), (u_4, \{u_4, u_6, u_7, u_8, u_9, u_{10}\}), \\ (u_6, \{u_4, u_6, u_7, u_8, u_9, u_{10}\}), (u_7, \{u_4, u_6, u_7, u_8, u_9, u_{10}\}), \\ (u_{10}, \{u_4, u_6, u_7, u_8, u_9, u_{10}\}), (u_8, \{u_4, u_6, u_7, u_8, u_9, u_{10}\}), \\ (u_9, \{u_4, u_6, u_7, u_8, u_9, u_{10}\}) \end{array} \right\}$$

2. Her görevlimin işyeri yemekhanesinde öğle yemeğini yediğini kabul edersek, çalışanların aynı masada beraber yemek yedikleri gruplara göre aynı *U* kümesi

üzerinde bir G ayrışımlı esnek kümesi yazılabilir. Burada G ayrışımlı esnek kümesinin yaklaşım fonksiyonu aşağıdaki gibi olur.

$$g: U \to P(U),\ \ g(u) = \{v \in U: u \text{ ile beraber aynı masada yemek yerler.}\}$$

Her bir çalışan için denklik sınıflarına ayırma işlemi 1. örneğe benzer şekilde yapılır.

3. Her görevlimin çalıştığını kabul edersek, çalışanların aynı iş bölümünde beraber çalıştıkları gruplara göre aynı U kümesi üzerinde bir H ayrışımlı esnek kümesi yazılabilir. Burada H ayrışımlı esnek kümesinin yaklaşım fonksiyonu aşağıdaki gibi olur.

$$h: U \to P(U),\ \ g(u) = \{v \in U: u \text{ ile beraber iş bölümünde çalışanlar.}\}$$

Her bir çalışan için denklik sınıflarına ayırma işlemi 1. örneğe benzer şekilde yapılır.

6. ESNEK ŞİFRELEME

Bu bölümde, ayrışımlı esnek kümenin özelliklerini kullanarak esnek şifrelemeyi tanımlayacağız.

Tanım 6.1. Karakterler kümesi U olsun. U üzerinde tanımlı bir ayrışımlı esnek kümesini kullanarak aşağıdaki şekilde bir metnin şifrelenmesine **esnek şifreleme** denir.

Esnek Şifreleme Algoritması:

1. U karakter kümesi belirlenir.
2. U üzerinde bir F ayrışımlı esnek kümesi tanımlanır.
3. Şifrelemek için şifrelenecek metnin i-inci karakteri hangi denklik sınıfında olduğu bulunur ve bundan i tane sonraki harf şifreli metinin i-inci karakteri olarak belirlenir. Gerektiğinde modüler aritmetik kullanılır.
4. İlk 3 madde bütün karakterler için yapılarak şifreleme tamamlanır.

***Örnek* 6.2** "su sıvıdır" metnini esnek şifrelemeyle şifreleyelim.

1. Karakter kümesi $U = \{d, e, f, g, h, ı, i, j, p, r, s, u, \quad, v\}$ olsun.
2. U üzerinde bir F ayrışımlı esnek kümesinin yaklaşım fonksiyonu aşağıdaki gibi tanımlansın.

$$f(d) = f(e) = f(f) = f(g) = f(h) = \{d, e, f, g, h\}$$
$$f(ı) = f(i) = f(j) = \{ı, i, j\}$$
$$f(p) = f(r) = f(s) = f(u) = f(v) = f(\quad) = \{p, r, s, u, \quad, v\}$$

3. "su sıvıdır" metinin harf harf şifreleyelim:

- "s" harfi, "su sıvıdır" cümlesinin birinci karakteri olduğu için f(s)'in şifrelenmiş metindeki yerine $f(s)$ kümesinde "s" harfinden sonraki ilk harf olan "u" olur.

- “u” harfi, “su sıvıdır” cümlesinin ikinci karakteri olduğu için f(u)’in şifrelenmiş metindeki yerine $f(u)$ kümesinde “u” harfinden sonraki ikinci harf olan “v” olur.

- Boşluk, “su sıvıdır” cümlesinin üçüncü karakteri olduğu için f( )’in şifrelenmiş metindeki yerine $f(\ )$ kümesinde boşluktan sonraki üçüncü harf olan “r” olur.

- “s” harfi, “su sıvıdır” cümlesinin dördüncü karakteri olduğu için f(s)’in şifrelenmiş metindeki yerine $f(s)$ kümesinde “s” harfinden sonraki dördüncü harf olan “p” olur.

- “ı” harfi, “su sıvıdır” cümlesinin beşinci karakteri olduğu için f(ı)’in şifrelenmiş metindeki yerine $f(\imath)$ kümesinde “ı” harfinden sonraki beşinci harf olan “j” olur.

- “v” harfi, “su sıvıdır” cümlesinin altıncı karakteri olduğu için f(v)’in şifrelenmiş metindeki yerine $f(v)$ kümesinde “v” harfinden sonraki altıncı harf olan “v” olur.

- “ı” harfi, “su sıvıdır” cümlesinin yedinci karakteri olduğu için f(ı)’in şifrelenmiş metindeki yerine $f(\imath)$ kümesinde “ı” harfinden sonraki yedinci harf olan “i” olur.

- “d” harfi, “su sıvıdır” cümlesinin sekizinci karakteri olduğu için f(d)’in şifrelenmiş metindeki yerine $f(d)$ kümesinde “d” harfinden sonraki sekizinci harf olan “g” olur.

- “ı” harfi, “su sıvıdır” cümlesinin dokuzuncu karakteri olduğu için f(ı)’in şifrelenmiş metindeki yerine $f(\imath)$ kümesinde “ı” harfinden sonraki dokuzuncu harf olan “ı” olur.

- “r” harfi, “su sıvıdır” cümlesinin onuncu karakteri olduğu için f(r)’in şifrelenmiş metindeki yerine $f(r)$ kümesinde “r” harfinden sonraki onuncu harf olan “v” olur.

Buradan “su sıvıdır” cümlesinin şifrelenmiş hali “uvrpjvigıv” olur.

Esnek Deşifreleme Algoritması:

1. Aynı U karakter kümesi üzerinde tanımlı aynı F ayrışımlı esnek kümesi kullanılır.
2. Şifrelenmiş metni çözmek için şifreli metnin i-inci karakteri hangi denklik sınıfında olduğu bulunur ve bundan i tane önceki karakter çözülen metinin i-inci karakteri olarak belirlenir. Gerektiğinde modüler aritmetik kullanılır.
3. İlk 2 madde bütün karakterler için yapılarak deşifreleme tamamlanır.

***Örnek* 6.3**. Burada aynı ayrışımlı esnek kümeyi kullanarak Örnek 6.2’da şifrelediğimiz “uvrpjvigıv” cümlesini çözümleyelim.

1. $U = \{d, e, f, g, h, ı, i, j, p, r, s, u, \quad, v\}$ karakter kümesi üzerinde tanımlı aynı F ayrışımlı esnek kümesinin yaklaşım fonksiyonunu kullanılır.

$$f(d) = f(e) = f(f) = f(g) = f(h) = \{d, e, f, g, h\}$$
$$f(ı) = f(i) = f(j) = \{ı, i, j\}$$
$$f(p) = f(r) = f(s) = f(u) = f(v) = f(\quad) = \{p, r, s, u, \quad, v\}$$

2. Şifrelenmiş “uvrpjvigıv” metinin harf harf çözelim:

- “u” harfi, “uvrpjvigıv” cümlesinin birinci karakteri olduğu için f(u)’nun şifrelenmiş metindeki yerine $f(u)$ kümesinde “u” harfinden önceki ilk harf olan “s” olur.

- “v” harfi, “uvrpjvigıv” cümlesinin ikinci karakteri olduğu için f(v)’nin şifrelenmiş metindeki yerine $f(v)$ kümesinde “v” harfinden önceki ikinci harf olan “u” olur.

- “r” harfi, “uvrpjvigıv” cümlesinin üçüncü karakteri olduğu için f(r) ’nin şifrelenmiş metindeki yerine $f(r)$ kümesinde “r” harfinden önceki üçüncü karakter olan “ ” boşluk olur.

- “p” harfi, “uvrpjvigıv” cümlesinin dördüncü karakteri olduğu için f(p)’nin şifrelenmiş metindeki yerine $f(p)$ kümesinde “p” harfinden önceki dördüncü harf olan “s” olur.

- “j” harfi, “uvrpjvigıv” cümlesinin beşinci karakteri olduğu için f(j)’nin şifrelenmiş metindeki yerine $f(j)$ kümesinde “j” harfinden önceki beşinci harf olan “ı” olur.

- “v” harfi, “uvrpjvigıv” cümlesinin altıncı karakteri olduğu için f(v)’nin şifrelenmiş metindeki yerine $f(v)$ kümesinde “v” harfinden önceki altıncı harf olan “v” olur.

- “i” harfi, “uvrpjvigıv” cümlesinin yedinci karakteri olduğu için f(i)’nin şifrelenmiş metindeki yerine $f(i)$ kümesinde “i” harfinden önceki yedinci harf olan “ı” olur.

- “g” harfi, “uvrpjvigıv” cümlesinin sekizinci karakteri olduğu için f(g)’nin şifrelenmiş metindeki yerine $f(g)$ kümesinde “g” harfinden önceki sekizinci harf olan “d” olur.

- “ı” harfi, uvrpjvigıv” cümlesinin dokuzuncu karakteri olduğu için f(ı)’nın şifrelenmiş metindeki yerine $f(\imath)$ kümesinde “ı” harfinden önceki dokuzuncu harf olan “ı” olur.

- “v” harfi, “uvrpjvigıv” cümlesinin onuncu karakteri olduğu için f(v)’nin şifrelenmiş metindeki yerine $f(v)$ kümesinde “v” harfinden önceki onuncu harf olan “r” olur.

Buradan “uvrpjvigıv” cümlesinin çözülmüş hali beklendiği gibi “su sıvıdır” olur.

7. ŞİFRELEME PROGRAMI

Bu bölümde, ayrışımlı esnek şifreleme metodunun uygulanmasına yönelik olarak geliştirilen bilgisayar programı tanıtılacaktır. Program yazılımına girmeden önce programın nasıl çalıştığını gösteren bir algoritmayı ve örneğini açıklayalım.

7.1 Esnek Şifreleme Program Kodları

Bu alt bölümde Esnek şifreleme metodunun Python bilgisayar programlama dilindeki kodları verilmiştir.

```
import random
def sezar_sifrele(metin):
    """Verilen metni Sezar şifresi ile şifreler, her harf kendi
indeks numarası kadar kaydırılır."""
    turk_alfabe = "abcçdefgğhıijklmnoöprsştuüvyz"
    sifreli_metin = ""
    parcali_alfabe = olustur_parcali_alfabe(turk_alfabe)

    # Ayrışım Esnek Kümeleri göster
    for i, grup in enumerate(parcali_alfabe):
        print(f"Ayrışım Esnek Küme_{i+1}: {grup}")

    index = 1  # Boşluklar da dahil olmak üzere indeksleme için
başlangıç değeri
    for i, harf in enumerate(metin):
        if harf.isalpha():
            if harf.isupper():
                baslangic_kodu = ord('A')
                sifreli_harf = chr((ord(harf) - baslangic_kodu +
index) % 29 + baslangic_kodu)
            else:
```

```
                for j, grup in enumerate(parcali_alfabe):
                    if harf in grup:
                        sifreli_harf = grup[(grup.index(harf) +
index) % len(grup)]
                        break
        else:
            sifreli_harf = harf
        sifreli_metin += sifreli_harf
        index += 1
    return sifreli_metin, parcali_alfabe

def sezar_coz(sifreli_metin, parcali_alfabe):
    """Verilen Sezar şifrelenmiş metni çözer, her harf kendi
indeks numarası kadar geri kaydırılır."""
    turk_alfabe = "abcçdefgğhıijklmnoöprsştuüvyz"
    cozulmus_metin = ""
    index = 1

    for i, harf in enumerate(sifreli_metin):
        if harf.isalpha():
            if harf.isupper():
                baslangic_kodu = ord('A')
                cozulmus_harf = chr((ord(harf) - baslangic_kodu
- index) % 29 + baslangic_kodu)
            else:
                for j, grup in enumerate(parcali_alfabe):
                    if harf in grup:
                        cozulmus_harf = grup[(grup.index(harf) -
index) % len(grup)]
                        break
        else:
            cozulmus_harf = harf
```

```
        cozulmus_metin += cozulmus_harf
        index += 1
    return cozulmus_metin

def olustur_parcali_alfabe(alfabe):
    """Türk alfabesini rastgele parçalara böler."""
    parcali_alfabe = []
    grup_sayisi = random.randint(3, 7)
    kalan_harfler = list(alfabe)
    for i in range(grup_sayisi):
        if i == grup_sayisi - 1:
            grup_uzunlugu = len(kalan_harfler)
        else:
            grup_uzunlugu = random.randint(1, len(kalan_harfler)
- grup_sayisi + i)
            if grup_uzunlugu == 0:
                grup_uzunlugu = 1
        grup = kalan_harfler[:grup_uzunlugu]
        parcali_alfabe.append(grup)
        kalan_harfler = kalan_harfler[grup_uzunlugu:]
    return parcali_alfabe

# Türk alfabesini ekranda göster
turk_alfabe = "abcçdefgğhıijklmnoöprsştuüvyz"
print("Türk Alfabesi:", turk_alfabe)

# Kullanıcıdan şifrelenecek metin girmesini iste
metin = input("Şifrelenecek metni girin: ")

sifreli_metin, parcali_alfabe = sezar_sifrele(metin)
print("Şifrelenmiş Metin:", sifreli_metin)
```

```
# Kümeleri bir kere göster
print("Bölünmüş Kümeler:", parcali_alfabe)

# Kullanıcıdan çözülecek metin girmesini iste
sifreli_metin = input("Çözülecek şifreli metni girin: ")

cozulmus_metin = sezar_coz(sifreli_metin, parcali_alfabe)
print("Çözülmüş Metin:", cozulmus_metin)
```

7.2 Programın Genel Yapısı ve Çalışma Şekli

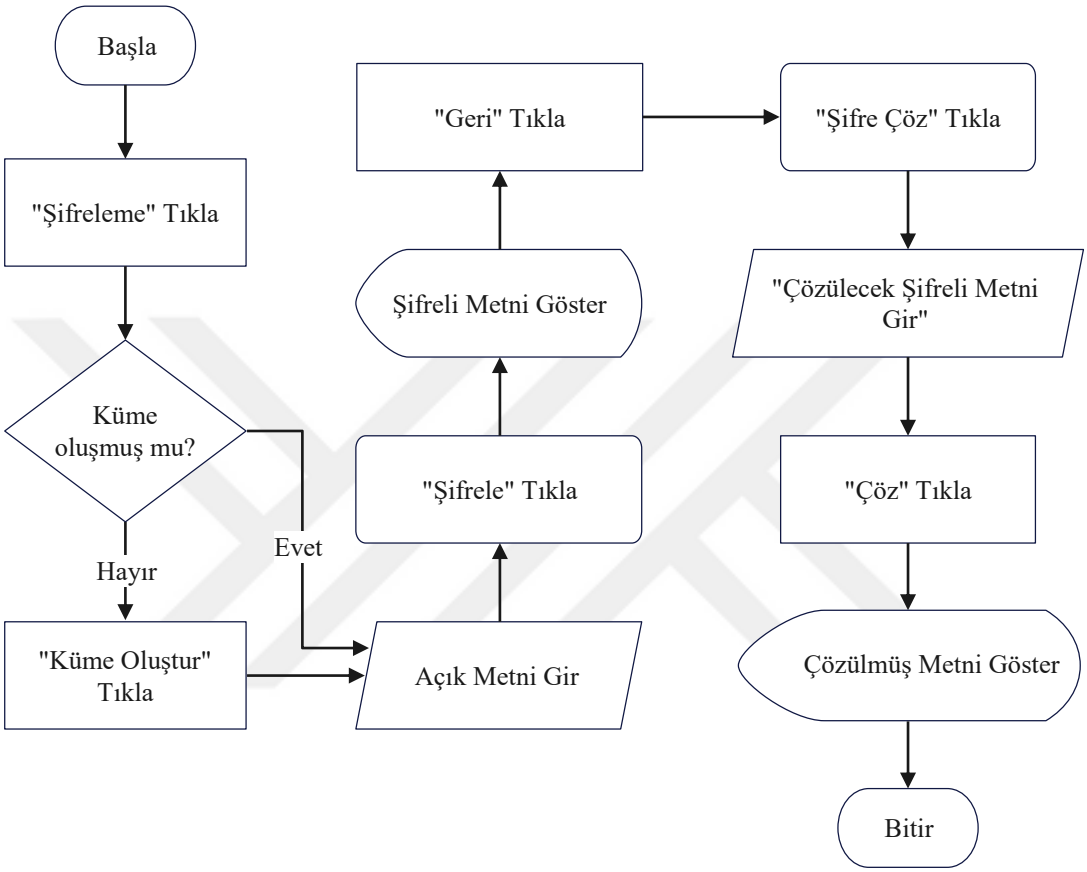
Program, “Esnek Şifreleme” adıyla oluşturulmuştur ve Microsoft tarafından açık kaynaklı olarak geliştirilen Visual Studio Code ortamında yazılmıştır. Yazılım, Python programlama dili kullanılarak geliştirilmiş ve Windows 10 ile Windows 11 işletim sistemlerinde çalışacak şekilde tasarlanmıştır. Programın kullanıcı arayüzü ve ilgili işlevleri, Python'un PyQt5 kütüphanesi ile gerçekleştirilmiştir.

7.2.1 Programın genel yapısı

Esnek Şifreleme programı, bir metin veya metin öbeğini şifrelemek için geliştirilmiş bir Sezar şifreleme yönteminin esnek kümeler ile uyarlanmış halidir. Oluşturulacak küme sayısı, maksimum yedi olarak belirlenmiştir. Küme sayısı, seçilen küme elemanlarından fazla olamaz. Ayrıca, küme sayısı 2'den az olamaz. Kullanılacak küme sayısı, şifrelemenin kapsamını belirlemek için ayarlanabilir. Boş kümeler esnek küme tanımına aykırı olduğundan programda kullanılamaz. Program, şifreleme ve çözme işlemlerini eş zamanlı olarak gerçekleştirmek üzere tasarlanmıştır. Bu kurallar, Esnek Şifreleme programının tutarlı ve güvenilir bir şekilde çalışmasını sağlamak amacıyla oluşturulmuştur.

7.2.2 Çalışma şekli

Arayüz programın çalışması Şekil 7-1'deki algoritma adımlarına göre gerçekleşir.

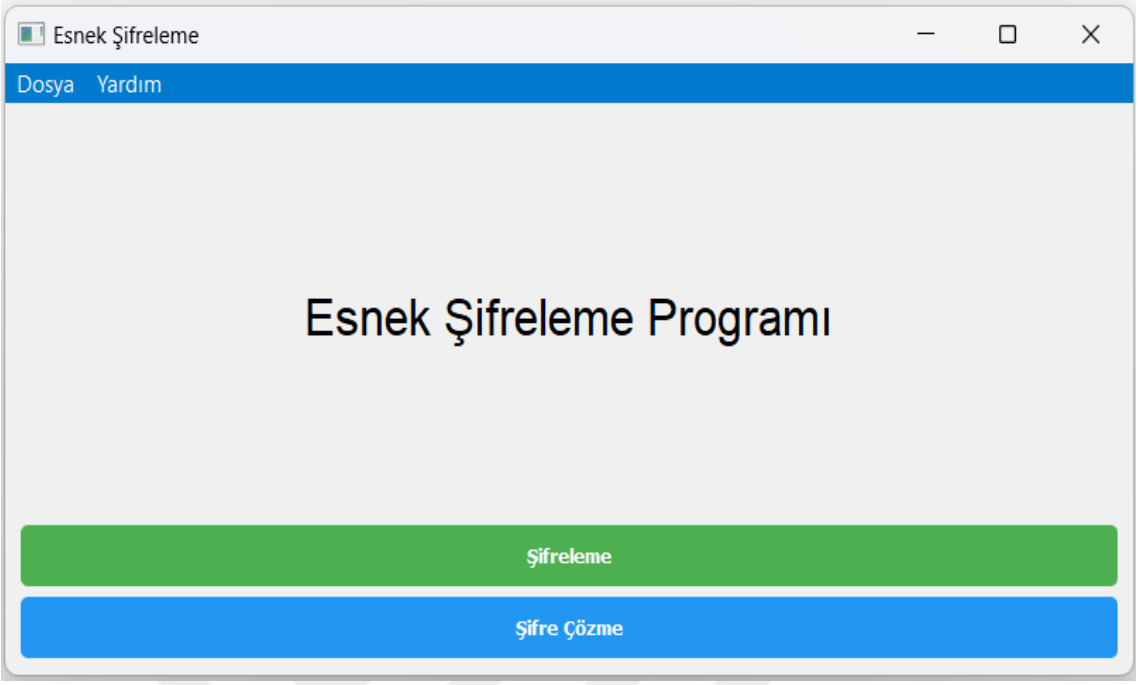


Şekil 7-1: Esnek şifreleme programı algoritması

7.3 Arayüz Uygulama Programın Tanıtımı

7.3.1 Program giriş arayüzü

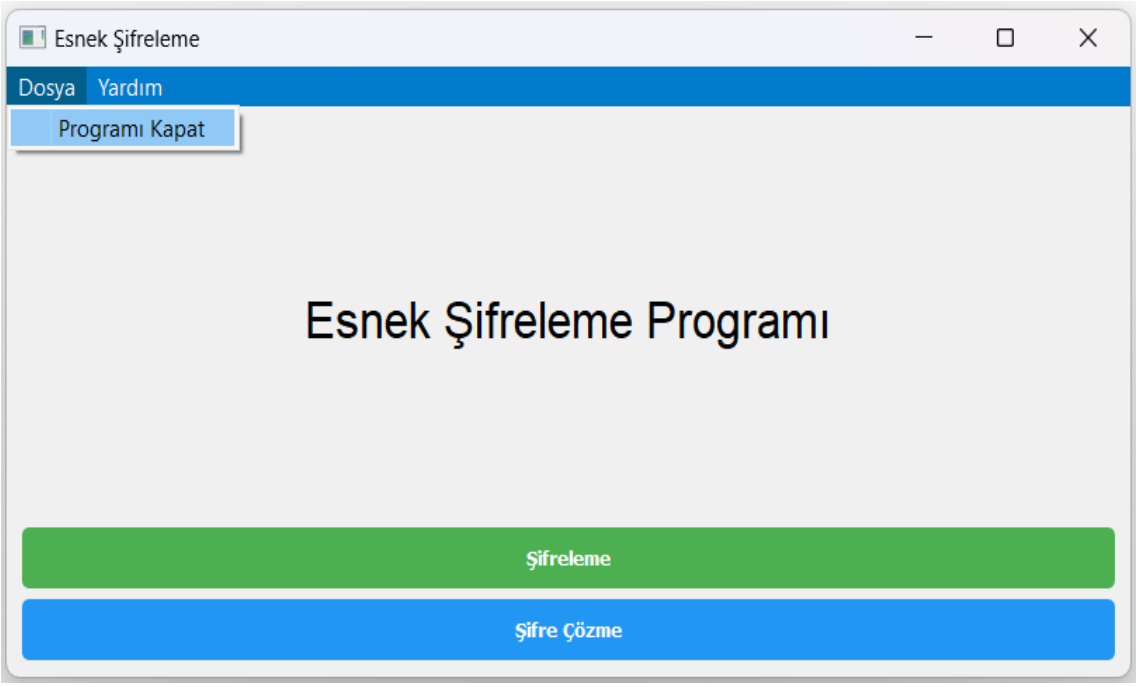
Şekil 7-2'deki giriş arayüzünde sol üst köşede "Dosya" ve "Yardım" menüsü bulunmaktadır. "Dosya" menüsünde "Programı Kapat" seçeneği yer alırken, "Yardım" menüsünde "Şifreleme Adımları" ve "Şifre Çözme Adımları" başlıkları altında yardım metinleri mevcuttur. Arayüzün alt kısmında, "Şifreleme" ve "Şifre Çözme" arayüzlerine geçiş sağlayan butonlar bulunmaktadır.



Şekil 7-2: Program arayüzü

Dosya Menüsü

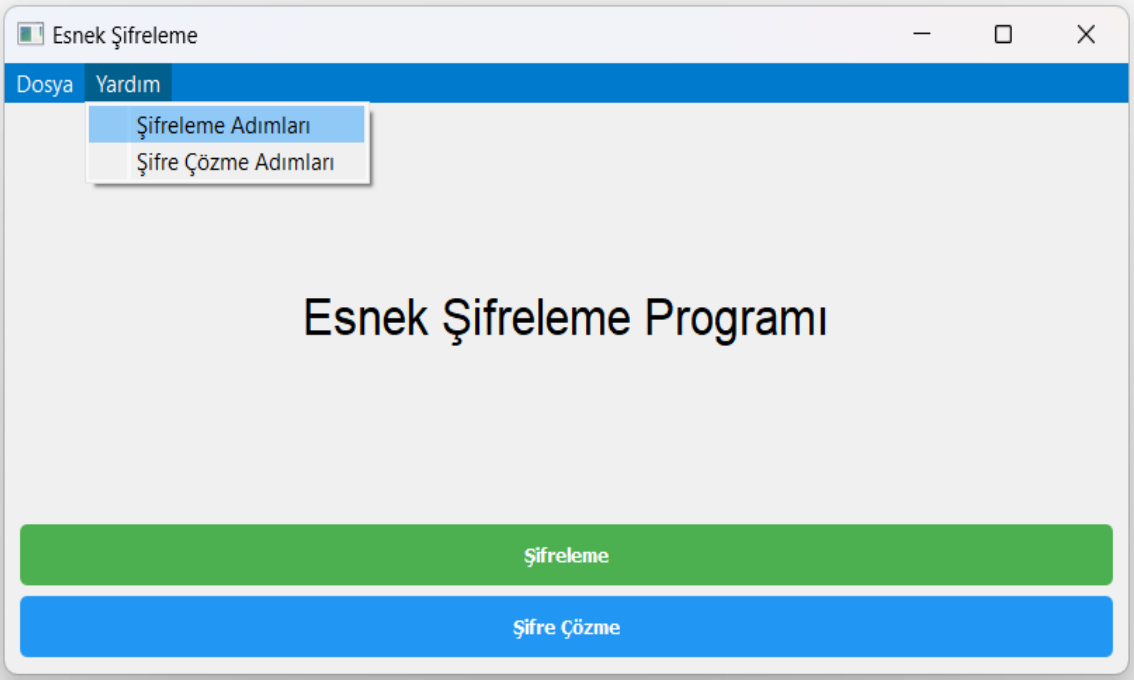
Şekil 7-3'de görüldüğü gibi **dosya menüsü,** bir başlıktan meydana gelir ve **programı kapat** seçeneği, programdan çıkmayı sağlar.



Şekil 7-3: Giriş arayüzü dosya menüsü

Yardım Menüsü

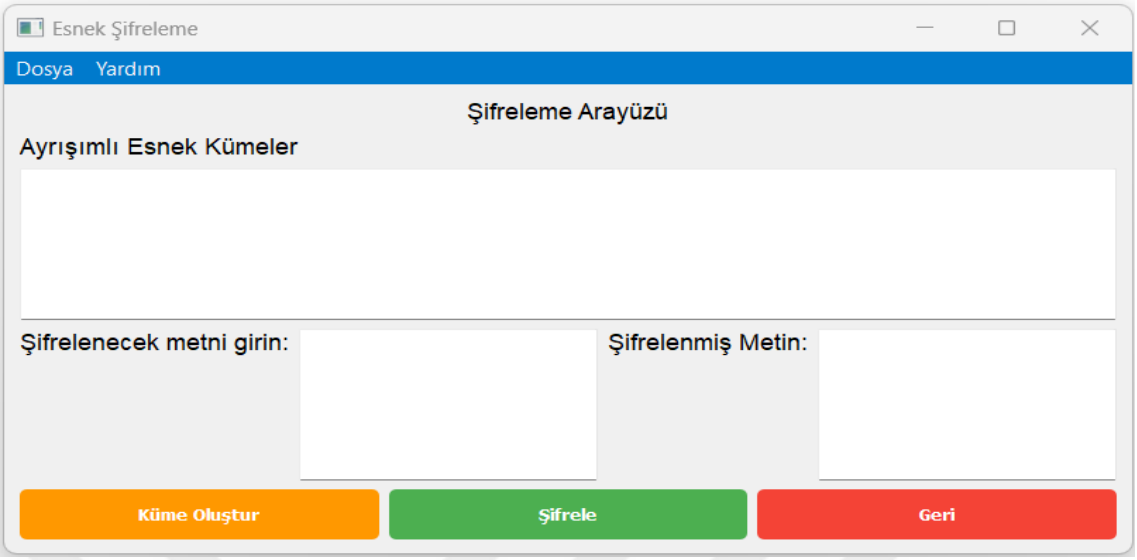
Şekil 7-4'de görüldüğü gibi **Yardım menüsü** iki başlıktan meydana gelir. Arayüzdeki tüm "yardım" menüleri aynı seçenekleri içerir. **Şifreleme Adımları**, metni şifrelemek için izlenmesi gereken kuralları içerir. **Şifre Çözme Adımları**, şifrelenmiş metni çözmek için gereken kuralları içerir.



Şekil 7-4: Giriş arayüzü yardım menüsü

7.3.2 Şifreleme arayüzü

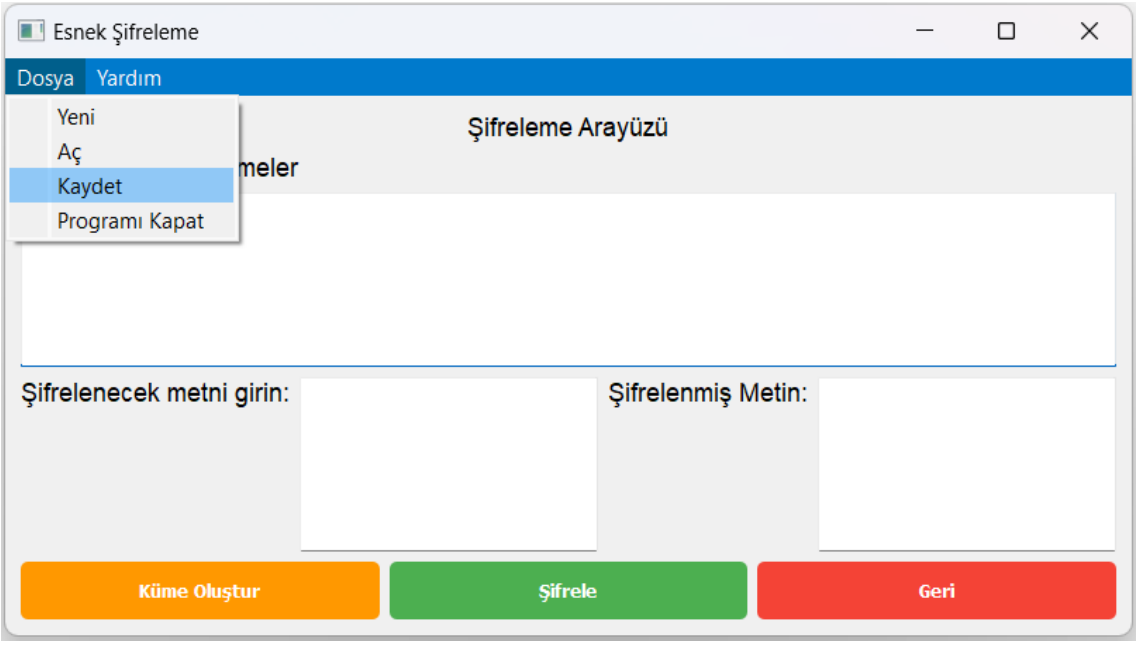
Şifreleme arayüzünde Şekil 7-5'te görüldüğü gibi sol üst köşede "Dosya" ve "Yardım" menüsü bulunur. "Dosya" menüsünde "Yeni", "Aç", "Kaydet" ve "Programı Kapat" seçenekleri vardır. "Yardım" menüsünde programın kullanımı ile ilgili bilgiler yer almaktadır. "Ayrışımlı Esnek Kümeler" başlığı altında "Küme Oluştur" butonuna tıklanınca oluşturulan kümeler görüntülenir. "Şifrelenecek metni girin" bölümüne şifrelenmek istenen metin girilir. "Şifrelenmiş Metin" alanında, şifrelenecek metnin şifrelenmiş görüntüsü yer alır. "Küme Oluştur" butonu rastgele sayıda ve rastgele eleman sayısında küme oluşturur. "Şifrele" butonu, "Şifrelenecek Metni Girin" bölümüne girilen metni şifreler. "Geri" butonu giriş arayüzüne dönmeyi sağlar.



Şekil 7-5: Şifreleme arayüzü

Dosya Menüsü

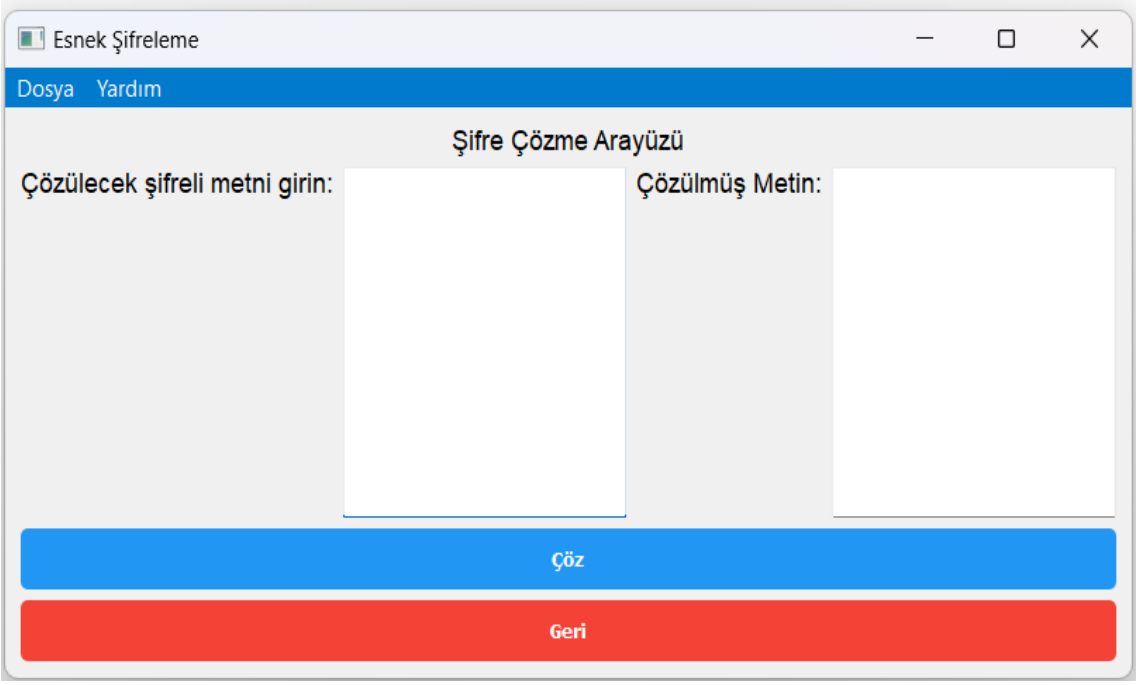
Şekil 7-6 da **Dosya menusu** dört başlıktan meydana gelmektedir. **Yeni**, şifrelenmiş yada şifrelenecek metin için boş bir çalışma sayfası açar. **Aç**, şifrelenmek istenen bir dosyayı şifrelenecek metin bölümüne ekler. **Kaydet**, açık metni, şifrelenmiş metni ve oluşturulan kümeleri kaydeder. Bu bize eşzamanlı kullanımda problem çıktığında şifreyi çözme imkânı sunar. **Programı kapat,** arayüz programından çıkmak için kullanılır.



Şekil 7-6: Şifreleme arayüzü dosya menüsü

7.3.3 Şifre çözme arayüzü

Şekil 7-7'deki Şifre çözme arayüzünde "Dosya" menüsü, giriş arayüzündeki gibi yalnızca "Programı Kapat" seçeneğini içermektedir. "Çözülecek Şifreli Metni Girin" alanında, şifrelenmiş metnin çözülmesi için gerekli metin girilmelidir. "Çözülmüş Metin" bölümünde ise, "Çözülecek Şifreli Metni Girin" kısmına yazılan metnin çözümlenmiş hali görüntülenir. Ekranın alt kısmında yer alan "Çöz" butonuna tıklandığında, girilen şifreli metin çözülür. "Geri" butonu ise kullanıcıyı giriş arayüzüne döndürür.



Şekil 7-7: Şifre çözme arayüzü

8. PROGRAMIN UYGULANMASI

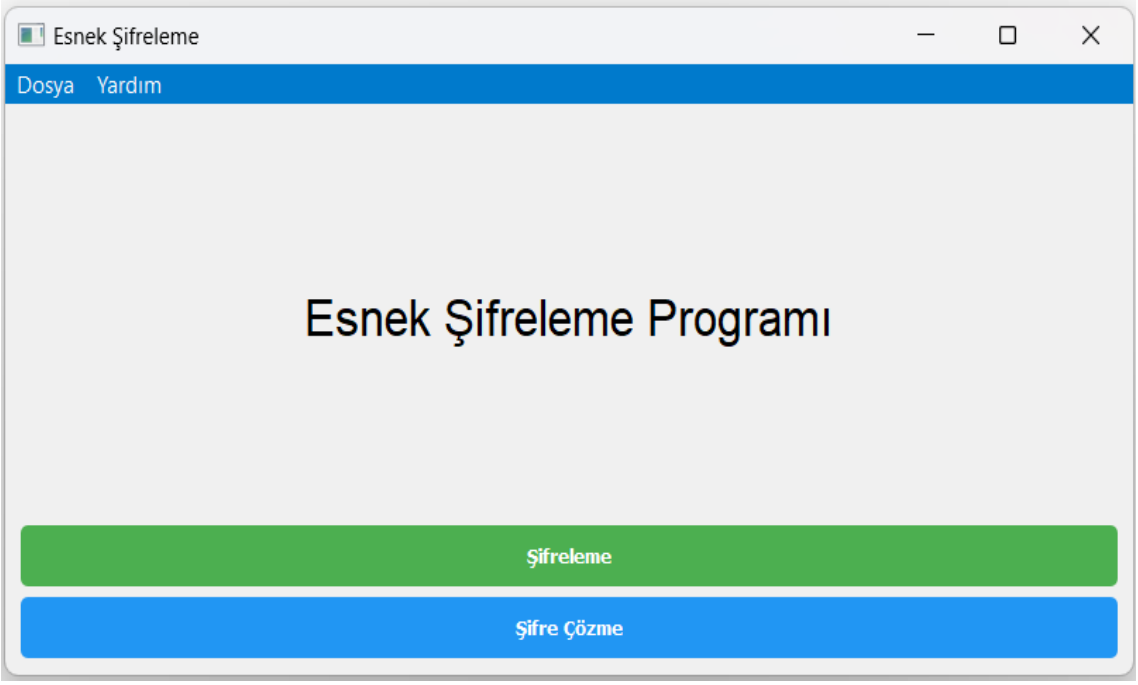
Bu bölümde Çizelge 3'te verilen açık metni Esnek Şifreleme programı ile şifreleyerek Çizelge 4'teki şifrelenmiş metne ulaşacağız. Daha sonra Çizelge 4'teki şifrelenmiş metni çözerek Çizelge 3'teki açık metne ulaşacağız.

Esnek Şifreleme programı ile Çizelge 3'te verilen açık metni şifreleyelim.

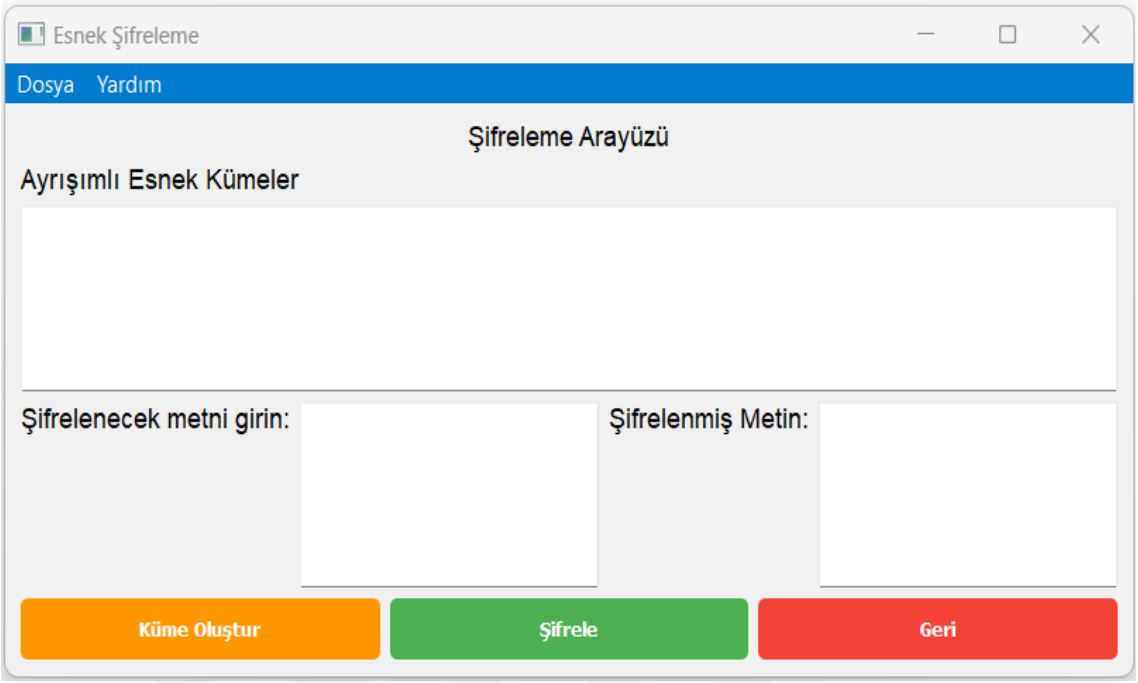
AÇIK METİN
Büyük veri, büyük miktarlarda yapılandırılmamış veya yapılandırılmış verilerin hızlı bir şekilde oluşturulması ve analiz edilmesi anlamına gelir. Bu veriler genellikle hacim, hız ve çeşitlilik (3V) ile tanımlanır.

Çizelge 3: Şifrelenecek açık metin

Şekil 8.1'de gösterilen program arayüzünde, kullanıcı öncelikle yeşil buton ile belirtilen **Şifreleme** butonuna basarak Şekil 8.2'de şifreleme işlemini başlatır.

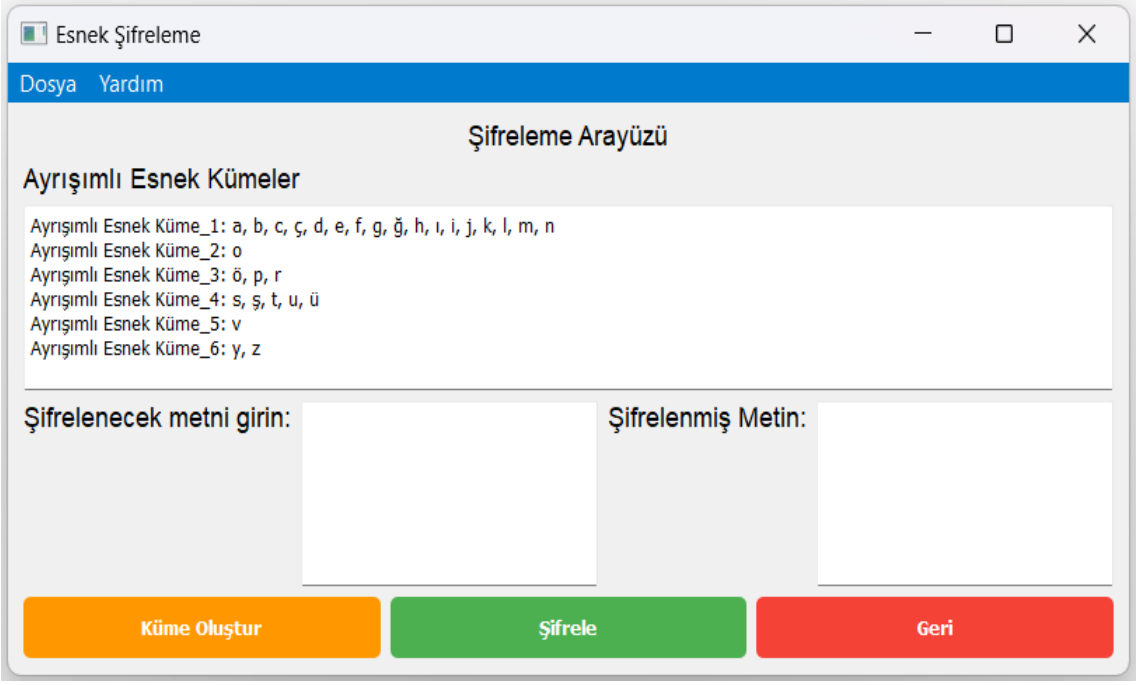


Şekil 8-1: Program arayüzü



Şekil 8-2: Şifreleme arayüzü

Şekil 8-2'de öncelikle turuncu renkli "küme oluştur" butonuna tıklanarak Şekil 8.3 'de gösterildiği gibi rastgele uzunlukta ve sayıda kümeler oluşturulur.

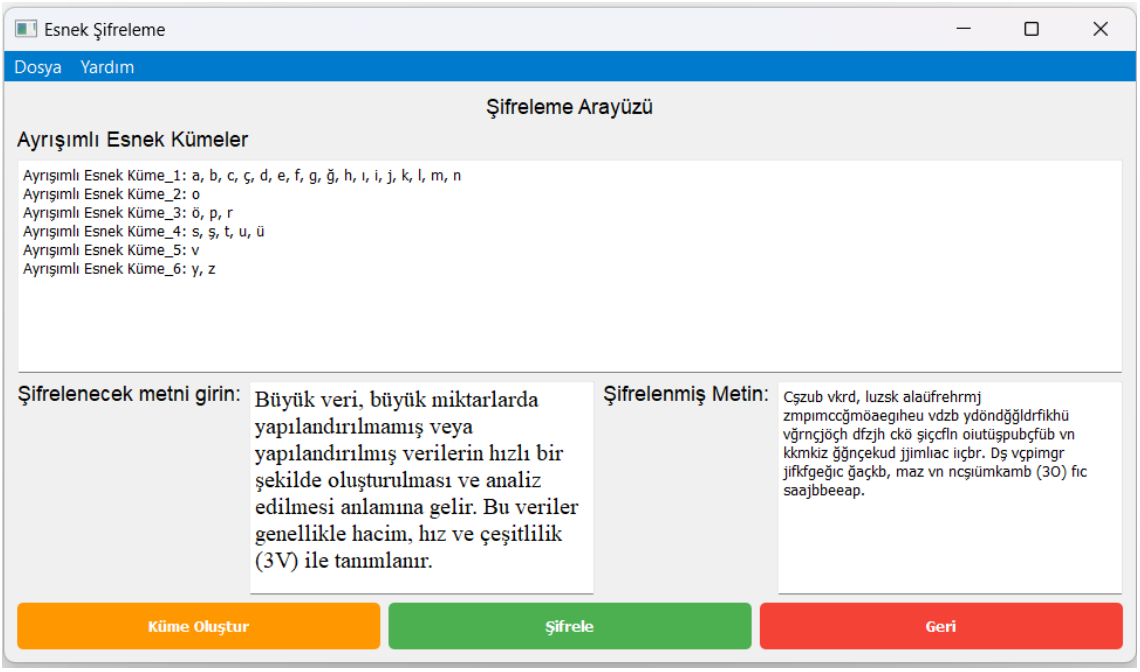


Şekil 8-3: Küme oluşturma

Daha sonra, Çizelge 3'te yer alan metin, kopyalanarak "Şifrelenecek Metni Girin" alanına, Şekil 8.4'te görüldüğü gibi yapıştırılır.

Şekil 8-4: Şifrelenecek metnin girilmesi

Yeşil buton ile belirtilen "Şifrele" butonuna tıklanarak, metin Şekil 8.5'teki gibi şifrelenir. Çizelge 4'te şifrelenen metin gösterilmiştir.



Şekil 8-5: Metnin şifrelenmesi

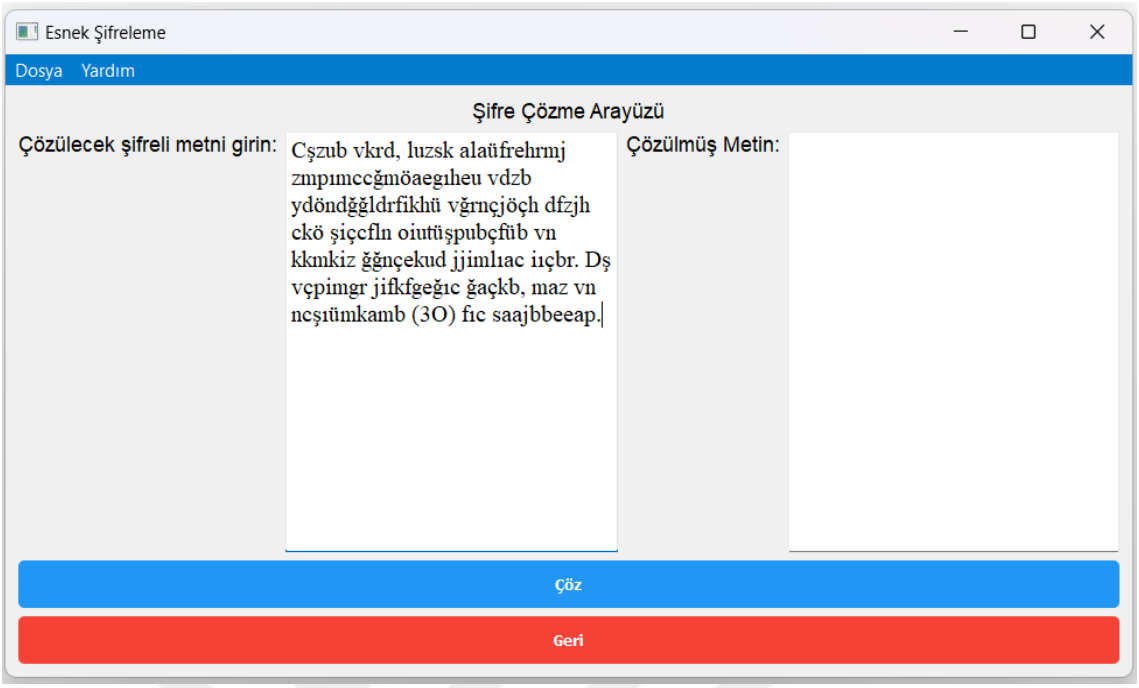
ŞİFRELİ METİN
Cşzub vkrd, luzsk alaüfrehrmj zmpımccğmöaegıheu vdzb ydöndğğldrfikhü vğrnçjöçh dfzjh ckö şiçcfln oiutüşpubçfüb vn kkmkiz ğğnçekud jjimlıac ııçbr. Dş vçpimgr jifkfgeğıc ğaçkb, maz vn ncşıümkamb (3O) fıc saajbbeeap.

Çizelge 4: Şifrelenmiş metin

Şifrelenmiş metni çözmek için Şekil 8-5 te yer alan "Geri" butonuna tıklanılarak Şekil 8-1'deki program arayüzüne ulaşılır. Şekil 8-1'de mavi renkli "Şifre Çözme" butonuna tıklanılarak Şekil 7-6'daki Şifre çözme arayüzüne geçilir.

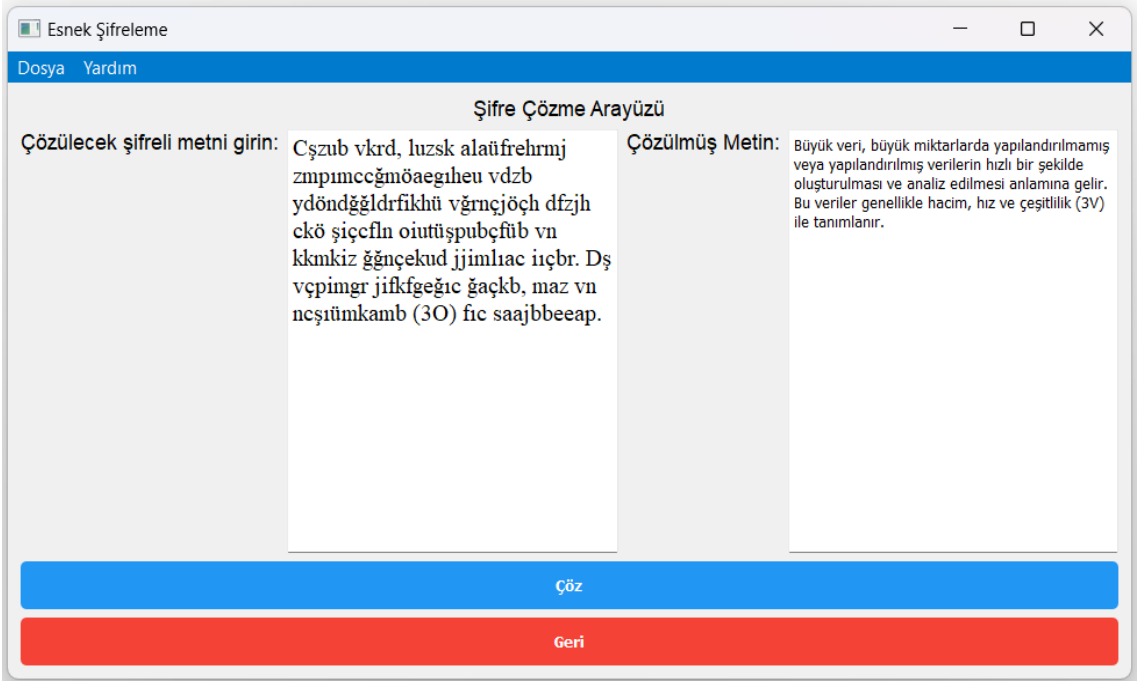
Şekil 8-6: Şifre çözme arayüzü

Çizelge 4'teki şifrelenmiş metni çözmek için Şekil 8-5'teki "Şifrelenmiş Metin" bölümünden kopyalanan metin, Şekil 8-7'de yer alan "Çözülecek Şifreli Metni Girin" alanına yapıştırılır.



Şekil 8-7: Çözülecek şifreli metnin istenmesi

Ardından, Şekil 8-7'de gösterilen mavi renkli "Çöz" butona tıklanarak, Şekil 8-8'deki gibi metin çözülür. Çizelge 3'teki açık metne ulaşılır.



Şekil 8-8: Şifreli metnin çözülmesi

9. SONUÇ ve TARTIŞMA

Bu tezin birinci bölümünde esnek kümler ve şifrelemeyle ilgili bir literatür taraması yaptık. İkinci bölümde şifrelemeyle ilgi temel kavramları verdik. Üçüncü bölümde esnek kümleri ve esnek küme işlemlerini tanıttık. Dördüncü bölümde denklik bağıntısı ve denklik sınıflarıyla ilgi tezin diğer bölümlerinde kullanacağımız gerekli bilgileri verdik. Beşinci bölümde esnek kümlerin üzerinde tanımlandığı kümeyi denklik sınıflarına yeni bir kavram olarak ayırarak ayrışımlı esnek kümleri tanımladık. Altıncı bölümde ayrışımlı esnek kümlerine dayanarak esnek şifrelemeyi tanımladık ve esnek şifrelemenin matematiksel modellemesini yaptık. Yeni tanımladığımız esnek şifreleme daha komplike bir Sezar şifreleme türüdür. Sekizinci bölümde esnek şifrelemenin Python programlama dili kullanılarak bilgisayar programını yazdık. Son bölüm olan dokuzuncu bölümde esnek şifrelemenin mevcut şifrelemelerdeki yeri ve geleceği hakkında bilgiler vererek tezi bitirdik.

Bu tezde sunulan ayrışımlı esnek kümeler, şifreleme sistemlerinin güvenliğini artırmaya yönelik yenilikçi bir yaklaşım ortaya koymaktadır. Esnek şifreleme tekniklerinin henüz gelişim sürecinde olduğu, ancak sahip olduğu yüksek potansiyel ile gelecekte veri güvenliği için önemli bir rol oynayacağı belirtilmelidir. Bu yeni yöntem, daha güçlü şifreleme çözümleri geliştirilmesine zemin hazırlamakta ve farklı uygulama alanlarında kullanım olanaklarını genişletmektedir.

Esnek şifreleme, veri güvenliğini artırmak için farklı tekniklerin ve algoritmaların kombinasyonunu kullanarak gelecekte önemli gelişmelere sahne olacaktır. Kuantum bilgisayarların etkisiyle kuantum dayanıklı algoritmalar, yapay zekâ destekli optimize çözümler, blockchain uygulamaları, IoT cihazlarının güvenliği, veri gizliliği yasalarına uyum ve hibrid güvenlik yaklaşımları esnek şifrelemenin potansiyel uygulama alanları arasında yer alıyor. Gelişen tehditlere karşı esnek şifreleme çözümlerinin kullanıcı verilerini güvenli bir şekilde koruma kabiliyeti, bu alandaki yeniliklerin önemini artıracaktır.

KAYNAKÇA

Abdirahman, A. A. (2018). *Kriptoanaliz problemlerinin çözümünde evrim stratejisi uygulaması* Sakarya Universitesi, Bilgisayar ve Bilişim Mühendisliği. Sakarya.

Aghayev, M. (2017). *Kriptoloji ve veri şifreleme teknikleri üzerine* Yüksek Lisans Tezi, Ege Üniversitesi Fen Bilimleri Enstitüsü, İzmir.

Akleylek, S., Yıldırım, H. M., ve Tok, Z. Y. (2011). Kriptoloji ve uygulama alanları: açık anahtar altyapısı ve kayıtlı elektronik posta. *Akademik Bilişim*, *11*, 2-4.

Aktaş, H., ve Çağman, N. (2007). Soft sets and soft groups. *Information sciences*, *177*(13), 2726-2735.

Ali, M. I., Feng, F., Liu, X., Min, W. K., ve Shabir, M. (2009). On some new operations in soft set theory. *Computers & mathematics with applications*, *57*(9), 1547-1553.

Anonim. (2024). *Oxford English Dictionary*. https://www.oed.com/search/dictionary/?scope=Entries&q=stenography (21.11.2024)

Aslan, B. (2013). *Blok şifreler için cebirsel ikili doğrusal dönüşüm tasarımı ve modern bir blok şifreye uygulanması,* Doktora, Trakya Üniversitesi,Bilgisayar Mühendisliği. Edirne.

Atagün, A. O., ve Sezgin, A. (2011). Soft substructures of rings, fields and modules. *Computers & mathematics with applications*, *61*(3), 592-601.

Avaroğlu, E. (2022). Bilgi Güvenliğinin Temel Yapı Taşı:Kriptoloji. *Düşünce Dünyasında Türkiz*, *8*(43), 53-65.

Bardeen, J., ve Brattain, W. H. (1948). The transistor, a semi-conductor triode. *Physical Review*, *74*(2), 230.

Başar, R. (2022). Türkiye'de ISO 27001 Bilg Güvenliği Yönetim Sistemi Sertifikasına Sahip Çeşitli Kuruluşların BGYS Yaklaşımları Üzerine Bir İnceleme. *Premium e-Journal of Social Sciences (PEJOSS)*, *6*(22), 303-310.

Batey, M. (2017), *Dilly: The man who broke Enigmas* (2st ed.), Biteback Publishing, s. 138.

Bauer, C. (2013), *Secret history: The story of cryptology* (1st ed.), Chapman and Hall/CRC, s. 113.

Baykara, M., Daş, R., ve Karadoğan, İ. (2013, 20-21 Mayıs). Bilgi güvenliği sistemlerinde kullanılan araçların incelenmesi. 1st International Symposium on Digital Forensics and Security (ISDFS'13), Elazığ.

Beşkirli, A., Özdemir, D., ve Beşkirli, M. (2019). Şifreleme yöntemleri ve RSA algoritması üzerine bir inceleme. *Avrupa Bilim ve Teknoloji Dergisi*, 284-291.

Bhardwaj, C. (2012). Modification of vigenere cipher by random numbers, punctuations & mathematical symbols. *Journal of Computer Engineering (IOSRJCE)*, *4*(2), 35-38.

Canbek, G., ve Sağıroğlu, Ş. (2006). Bilgi, bilgi güvenliği ve süreçleri üzerine bir inceleme. *Politeknik Dergisi*, *9*(3), 165-174.

Çağman, N., (2024). Soyut Matematiğin Temelleri, Kutlu Yayınevi, İstanbul.

Çağman, N., (2014). Contributions to the theory of soft sets, Journal of New Results in Science 3 (4), 33-41

Çağman, N., ve Enginoğlu, S. (2010). Soft set theory and uni–int decision making. *European journal of operational research*, *207*(2), 848-855.

Çubukçu, F. (2018), *Bilgi Güvenliği Yönetim Sistemi ISO27001: 2013 Uygulama Kılavuzu* (1st ed.), Pusula, s. 1-17.
Denniston, A., Park, B., ve Turing, A. (2020). *Enigma Machine*. https://spartacus-educational.com/Enigma_Machine.htm (21.11.2024)
Dooley, J. F. (2018), *History of cryptography and cryptanalysis* (M. Campbell-Kelly, Ed. 1st ed.), Springer, s. 195-212.
Ekonomikultur. (2024). *Alan Turing Kimdir? Savaşın Şifresini Çözen Matematikçi*. https://www.ekonomikultur.com/alan-turing-kimdir-savasin-sifresini-cozen-matematikci/ (22.11.2024)
Eminağaoğlu, M., ve Gökşen, Y. (2009). Bilgi güvenliği nedir, ne değildir? Türkiye'de bilgi güvenliği sorunları ve çözüm önerileri. *Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Dergisi 11*(4), 1-15.
Eroğlu, Ş. (2018). Dijital yaşamda mahremiyet (gizlilik) kavramı ve kişisel veriler: Hacettepe Üniversitesi bilgi ve belge yönetimi bölümü öğrencilerinin mahremiyet ve kişisel veri algılarının analizi. *Hacettepe Üniversitesi Edebiyat Fakültesi Dergisi*, *35*(2), 130-153.
Ersan, E. (2017). *Veri iletişiminde ağ dinleme saldırılarına karşı geliştirilmiş melez kriptosistem* Ankara Universitesi (Turkey), Bilgisayar Mühendisliği. Ankara.
Feng, F., Jun, Y. B., ve Zhao, X. (2008). Soft semirings. *Computers & mathematics with applications*, *56*(10), 2621-2628.
John, S.J. (2021). Soft Sets, Theory and Applications, Springer.
Jun, Y. B. (2008). Soft bck/bci-algebras. *Computers & mathematics with applications*, *56*(5), 1408-1413.
Jun, Y. B., ve Park, C. H. (2008). Applications of soft sets in ideal theory of BCK/BCI-algebras. *Information sciences*, *178*(11), 2466-2475.
Kodaz, H., ve Botsalı, F. M. (2010). Simetrik Ve Asimetrik Şifreleme Algoritmalarinin Karşilaştirilmasi. *Selcuk University Journal of Engineering Sciences*, *9*(1), 10-23.
Kovkov, D., Kolbanov, V., ve Molodtsov, D. (2007). Soft sets theory-based optimization. *Journal of Computer and Systems Sciences International*, *46*, 872-880.
Kuner, C. (2010). Regulation of transborder data flows under data protection and privacy law: past, present, and future. *TILT Law & Technology Working Paper*(016).
Leung, D. W. (2000). Quantum vernam cipher. *arXiv preprint quant-ph/0012077*.
Li, H., ve Wang, Y. (2022). The history of cryptography and its applications. *International Journal of Social Science and Education Research*, *5*(3), 343-349.
Maji, P. K., Biswas, R., ve Roy, A. R. (2003). Soft set theory. *Computers & mathematics with applications*, *45*(4-5), 555-562.
Menezes, A. J., Van Oorschot, P. C., ve Vanstone, S. A. (2018), *Handbook of applied cryptography*, CRC press, s.
Michigantech (2025), The Vigenère Cipher Encryption and Decryption, url:https://pages.mtu.edu/~shene/NSF-4/Tutorial/VIG/FIG-VIG-Table.jpg, 06.01.2025
Molodtsov, D. (1999). Soft set theory—first results. *Computers & mathematics with applications*, *37*(4-5), 19-31.
Molodtsov, D., Leonov, V. Y., ve Kovkov, D. (2006). Soft sets technique and its application. *Nechtkie Sistemy I Myagkie Vychisleniya*, *1*(1), 8-39.
Mushrif, M. M., Sengupta, S., ve Ray, A. K. (2006). Texture classification using a novel, soft-set theory based classification algorithm. Computer Vision–ACCV 2006: 7th

Asian Conference on Computer Vision, Hyderabad, India, January 13-16, 2006. Proceedings, Part I 7,

Okumuş, İ. (2012). *RSA Kritosisteminin Hızını Etkileyen Faktörler* Doktora tezi,Atatürk Üniversitesi. Erzurum.

Öztürk, İ., ve Zeybek, B. (2021). Dijitalleşme ve etik sorunlar: Nesnelerin interneti teknolojisini gözetim, gizlilik, güvenlik kapsamında değerlendirme. *İletişim Kuram ve Araştırma Dergisi*, *2021*(55), 1-15.

Paar, C., ve Pelzl, J. (2010), *Understanding cryptography* (2st ed. Vol. 1), Springer, s. 1-27.

Park, C.-H., Jun, Y.-B., ve Ozturk, M. A. (2008). Soft WS-algebras. *Communications of the Korean Mathematical society*, *23*(3), 313-324.

Pawlak, Z. (1982). Rough sets. *International journal of computer & information sciences*, *11*, 341-356.

Pei, D., ve Miao, D. (2005, 25-27 Jully). From soft sets to information systems. 2005 IEEE international conference on granular computing, Beijing, China.

Petitcolas, F., Anderson, R., ve Kuhn, M. (1999). Information hiding {a survey, in Proceedings of the IEEE, 87 (7): 1062 {1078. In: July.

Piper, F., ve Murphy, S. (2002), *Cryptography: A very short introduction* (Vol. 68), Oxford Paperbacks, s.

Rana, K. (2022). Cryptological Mathematics. *International Journal for Research in Applied Sciences and Biotechnology*, *9*(3), 107-112.

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

Schneier, B. (2007), *Applied cryptography: protocols, algorithms, and source code in C* (20st ed. Vol. 1), john wiley & sons, s.

Selleri, S. (2020). The roots of modern cryptography: Leon Battista Alberti's" De Cifris". *URSI Radio Science Bulletin*, *2020*(375), 55-63.

Sezgin, A., Atagün, A. O., ve Aygün, E. (2011). A note on soft near-rings and idealistic soft near-rings. *Filomat*, *25*(1), 53-68.

Stallings, W. (2006), *Cryptography and network security, 4/E* (50st ed.), Pearson Education India, s.

Stanoyevitch, A. (2010), *Introduction to Cryptography with mathematical foundations and computer implementations*, CRC Press, s.

Sun, Q.-M., Zhang, Z.-L., ve Liu, J. (2008). Soft sets and soft modules. Rough Sets and Knowledge Technology: Third International Conference, RSKT 2008, Chengdu, China, May 17-19, 2008. Proceedings 3,

Şahin, F. (2015). Modern Blok Şifreleme Algoritmaları. *İstanbul Aydın Üniversitesi Dergisi*, *7*(26), 23-40.

Tekerek, M. (2008). Bilgi güvenliği yönetimi. *KSÜ Doğa Bilimleri Dergisi*, *11*(1), 132-137.

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.

Toyran, M. (2007). Quantum cryptography. 2007 IEEE 15th Signal Processing and Communications Applications,

Tranquillus, G. S. (2013), *The Lives of the Twelve Caesars: Complete*, Simon and Schuster, s. 101.

Wikipedia. (2024a). *Enigma Makinesi*. https://tr.wikipedia.org/wiki/Enigma_makinesi (19 Kasım 2024)

Wikipedia. (2024b). *Sezar şifrelemesi*. https://tr.wikipedia.org/wiki/Sezar_%C5%9Fifrelemesi 01.10.2024

Wikipedia (2024c). Scytale. *Vikipedi, Özgür Ansiklopedi*. url:https://tr.wikipedia.org/w/index.php?title=Scytale&oldid=33587056., 01.6.2025

Wikipedia (2024d). Alberti şifresi. *Vikipedi, Özgür Ansiklopedi*. url:https://tr.wikipedia.org/w/index.php?title=Alberti_%C5%9Fifresi&oldid=33992550.,01.6.2025

Xiao, Z., Chen, L., Zhong, B., ve Ye, S. (2005). Recognition for soft information based on the theory of soft sets. Proceedings of ICSSSM'05. 2005 International Conference on Services Systems and Services Management, 2005.,

Yang, X., Yu, D., Yang, J., ve Wu, C. (2007). Generalization of soft set theory: from crisp to fuzzy case. Fuzzy Information and Engineering: Proceedings of the Second International Conference of Fuzzy Information and Engineering (ICFIE),

YazilimTurk (2024), Kriptoloji Nedir?, url: https://media.geeksforgeeks.org/wp-content/uploads/ceaserCipher.png, 01.06.2025

Yerlikaya, T., ve Aslanyürek, C. (2019). RSA algoritmasının şifreleme hızını arttıran algoritmalar ve performansları. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, *10*(3), 853-862.

Yerlikaya, T., Buluş, E., ve Buluş, N. (2006). Asimetrik Şifreleme Algoritmalarinda Anahtar Değişim Sistemleri. *Akademik Bilişim*, 9-11.

Yeşilbaş, E. (2016). *Cebirsel kriptoloji yöntemleri ve bazı uygulamaları* Recep Tayyip Erdoğan Üniversitesi/Fen Bilimleri Enstitüsü/Matematik.

Yılmaz, H. (2014). Ts Iso/Iec 27001 Bilgi Güvenliği Yönetimi Standardı Kapsamında Bilgi Güvenliği Yönetim Sisteminin Kurulması Ve Bilgi Güvenliği Risk Analizi. *Denetişim*(15), 45-59.