KLASİK TEKNİKLER KULLANILARAK BİR KRİPTOGRAFİ ALGORİTMASI GELİŞTİRİLMESİ VE DES ALGORİTMASI İLE PERFORMANS ANALİZLERİNİN KARŞILAŞTIRILMASI

ÜLKÜ ÜLKER

YÜKSEK LİSANS TEZİ

BİLGİSAYAR EĞİTİMİ

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

OCAK 2014

ANKARA

Ülkü ÜLKER tarafından hazırlanan KLASİK TEKNİKLER KULLANILARAK BİR KRİPTOGRAFİ ALGORİTMASI GELİŞTİRİLMESİ VE DES ALGORİTMASI İLE PERFORMANS ANALİZLERİNİN KARŞILAŞTIRILMASI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Aysun COŞKUN
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği/çokluğu ile Bilgisayar Eğitimi Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

Başkan : Doç. Dr. Suat ÖZDEMİR.....

Üye : Doç. Dr. Aysun COŞKUN.....

Üye : Yrd. Doç. Dr. Aslıhan TÜFEKÇİ.....

Tarih : 07 / 01 / 2014

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Ülkü ÜLKER

KLASİK TEKNİKLER KULLANILARAK BİR KRİPTOGRAFİ ALGORİTMASI GELİŞTİRİLMESİ VE DES ALGORİTMASI İLE PERFORMANS ANALİZLERİNİN KARŞILAŞTIRILMASI

(Yüksek Lisans Tezi)

Ülkü ÜLKER

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ
Ocak 2014

ÖZET

Bu çalışmada bilgi güvenliği ile ilgili genel bilgiler verilmiş ve sonrasında bilgi güvenliğini sağlama yollarından biri olan kriptoloji bilimi üzerinde durulmuştur. Kriptoloji biliminin tarihsel gelişimi ve bazı kriptoanaliz yöntemleri anlatılmıştır. Kriptoloji algoritmaları yöntem ve tekniklerinden bahsedilmiştir. Tez çalışmasının uygulama kısmında klasik teknikler kullanılarak bir kriptografi algoritması geliştirilmiş ve güvenirliğini test etmek için bire-bir harf frekans analizi uygulaması yapılmıştır. Geliştirilen algoritma hem paragraf hem kelime düzeyinde şifreleme işlemi yapmaktadır. Algoritmanın deşifreleme işlemleri için biyometrik bir özellik olan ses kullanılmıştır. Geliştirilen algoritmanın performans analiz değerlendirmesi yapılırken modern bir teknik olan DES algoritması ile karşılaştırması verilmiştir. Visual Studio ortamında C# dili kullanılarak yapılan işlemler için demo program hazırlanmıştır.

Bilim Kodu : 702.1.011
Anahtar Kelimeler: bilgi güvenliği, harf frekans analizi, kriptografi, kriptoloji, performans analizi
Sayfa Adedi : 171
Tez Yöneticisi : Doç. Dr. Aysun COŞKUN

DEVELOPING A CRYPTOGRAPHY ALGORITHM USING CLASSICAL TECNIQUES AND THEN COMPARISON OF PERFORMANCE ANAYSIS WITH DES ALGORITHM

(M.Sc. Thesis)

Ülkü ÜLKER

GAZİ UNIVERSITY
INFORMATICS INSTITUTE
January 2014

ABSTRACT

In this study provides general information about information security, and then focused on the science of cryptography is one way to ensure the security of information. The historical development of the science of cryptology is examined. Some methods of the cryptanalysis are described. Methods and techniques of the cryptography algorithms are mentioned. A cryptography algorithm is developed using the techniques of classical cryptography. To test the reliability of the algorithm, frequency analysis was conducted in exactly the case. The algorithm has both paragraph and word level encryption process. Biometric feature such as voice is used for decryption operations. Analysis of the algorithm performance assessment, conducted for encryption operations. Performance analysis of the proposed algorithm and the results were compared with DES which is a modern cryptography technique. For the simulation of the developed algorithm, a demo program has been prepared using the C# programming language with the MS Visual Studio environment.

ScienceCode : **702.1.011**
KeyWords :**information security, letter frequency analysis, cryptography, cryptology, performance analysis**
PageNumber : **171**
Adviser :**Assoc. Prof. Dr. Aysun COŞKUN**

TEŞEKKÜR

Yüksek lisans eğitimi ders döneminde tanıdığım ve tezimin konusunu ile tanışmama vesile olan değerli hocam Prof. Dr. Sayın Şeref SAĞIROĞLU'na, yüksek lisans eğitimime başladığımdan bu yana anlayış ve hoşgörüsüyle yardımlarını esirgemeyen çok değerli hocam Doç. Dr. Sevgili Halil İbrahim BÜLBÜL'e, süreçte psikolojik olarak destek olan çok değerli hocam Yrd. Doç. Dr. Sayın Zihni KOÇ'a, beni her zaman anlayışla ve sabırla dinleyen sevgili hocam Yrd. Doç. Dr. Sayın Nursel YALÇIN'a, yüksek lisans eğitimi süresince karşılaştığım tüm problemlerde kıymetli yardım ve desteklerini esirgemeyen çok değerli hocam Araştırma Görevlisi Saygıdeğer M. Hanefi CALP'e, lisans eğitimim süresince mesleki anlamda bana sağladığı kıymetli katkılarından dolayı kıymetli hocam Öğr. Gör. Sayın Zafer Dinç'e, tez danışmanım Doç. Dr. Sayın Aysun COŞKUN'a, süreçte her türlü sıkıntımda yanımda olarak yardım ve desteklerini esirgemeyen çok değerli abim sevgili Ümit Şen'e,

Kendisini tanımaktan mutluluk duyduğum sevgili arkadaşım Araştırma Görevlisi Sayın Rıdvan NABİKOĞLU'na,

Sevgili arkadaşım Araştırma Görevlisi Sayın Ömer ŞİMŞEK ve sevgili eşi Manolya ŞİMŞEK'e,

Hayatımda bilerek ya da bilmeyerek bir şekilde yer almış olan, akademik anlamda çalışmalar yapabilmem için beni bu yola sevk eden ve bu yolda ilerlemem için destek olan tanıdığım ve tanımadığım tüm insanlara,

Bana desteği ve anlayışıyla güç veren tüm arkadaş, dost ve kardeşlerime,

Hiçbir fedakârlıktan kaçınmadan beni yetiştiren AİLEME,

Olumlu ya da olumsuz tüm destek ve katkılarından dolayı TEŞEKKÜRLER.

Sevgili *AİLEME*…

İÇİNDEKİLER

Sayfa

Sayfa

Sayfa

ÇİZELGELERİN LİSTESİ

Çizelge **Sayfa**

ŞEKİLLERİN LİSTESİ

Şekil **Sayfa**

RESİMLERİN LİSTESİ

Resim **Sayfa**

Devam **Sayfa**

Devam **Sayfa**

Devam **Sayfa**

SİMGELER VE KISALTMALAR

Simgeler	Açıklama
≈	Yaklaşık

Kısaltmalar	Açıklama
AES	(Advanced Encryption Standart) Gelişmiş şifreleme standardı
ASEKAL-21	Aselsan'ın ürettiği ulusal şifreleme algoritması
ASELSAN	Askeri Elektronik Sanayi
DES	(Data Encrypt Standart) Veri şifreleme standardı
DSA	(Digital Signature Algorithm) Dijital imza algoritması
EMP	Elektromanyetik Darbe (Electromagnetic Pulse)
FEAL	Fast Date Encipherment Algorithm
HERF	Yüksek enerji radyo frekansı (High Energy Radio Frequncy)
IDEA	(International Data Encryption Algorithm) Uluslarası veri şifreleme algoritması
.NET	Network bileşenleri platformu
UEKAE	Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü
RC5	Rivest's Cipher / Rone's Code5
SAFER	(Secure and Fast Encryption Routine) Güvenli ve hızlı şifreleme yöntemi
TÜBİTAK	Türkiye Bilimsel ve Teknolojik Araştırma Kurumu

1. GİRİŞ

Kriptoloji alanında yeni üretimlerin sınırlı sayıda kalmış olması ve bu konu ile ilgilenen kurumların sayısının azlığı bu konuya dikkat çekilmesinin gerekliliğini ortaya çıkarmıştır. Konu ile ilgili yapılan çalışmaların çoğu var olan sistemlerin incelenmesi, iyileştirilmesi, birbiri ile karşılaştırılması boyutundadır. Var olan sistemler ulusal olduğu sürece ulus güvenliği açısından dış kaynaklı bir tehdit oluşturmayacağı için bu sistemlerin incelenip geliştirilmesi olumlu katkılar sağlayabilir. Ancak dış kaynaklı sistemlerin incelenerek, geliştirilmesi ve ulusal işlemlerde kullanılması sistemlerde büyük bir güvenlik tehdidi oluşturacaktır. Bu nedenle kriptoloji alanında yapılan çalışmalar ulusal düzeyde ve yeni üretimler olmalıdır. Giderek büyüyen internet ağı ve günlük faaliyetlerin elektronik ortama taşınması da bilgiye ulaşımı kolaylaştırdığı için bir tehdit olarak görülmelidir. Bilgiye ulaşım kolaylaştıkça bilginin güvenliğini sağlamak zorlaşmakta ve karmaşık bir hal almaktadır. Bu durum bilginin bulunduğu ortamda nasıl tutulması gerektiği problemini de ortaya çıkarmıştır. Bu anlamda da kriptoloji bilimine önem verilmelidir. Kriptoloji alanında çalışacak yetkili personelin yetiştirilmesi ve desteklenmesi gerekmektedir. Alan uzmanı personele ve ulusal çalışmalara duyulan ihtiyaç giderek artmaktadır. Savaşların elektronik ortamda bilgi savaşı olarak gerçekleşmeye başladığı günümüzde kriptoloji alanında yeni üretimlerin gerçekleştirilmesi bilgi güvenliği için önemlidir.

Çalışmadaki amaç, yeni bir kriptografi algoritması geliştirebilmektir. Yerine koyma yönteminde şifre alfabeyi bilmeyen birinin şifreli metni çözebilmesi olası görülmemektedir. Bu yöntemlerin kullanıldığı dönemlerde bilgisayarlar olmadığı için kağıt üzerinde tek tek deneyerek doğru sonuca ulaşmak neredeyse imkansızdır. Fakat daha sonra bu kadar çok denemeye gerek kalmadan dilin yapısal özelliklerinden faydalanarak bu tarz şifreleri çözmek mümkün olmuştur. Bu yöntem frekans analizi olarak bilinmektedir. Günümüzde ise bilgisayarlar ile bu şifreler kolayca kırılabilmektedir. Buradan

yola çıkılarak harf frekans analizine karşı güvenilir bir algoritma üretilmeye çalışılmıştır.

Konu alanında yeni üretimler gerçekleştirebilmek için matematik, dilbilim gibi bilimsel alt yapıların gerekliliği söz konusu olmakla birlikte araştırmacıların yetenek, hayal gücü, veriler arasında ilişki kurabilme ve ilişkileri çözümleyebilme becerileri de önemlidir. Büyük adımlar atabilmek için küçük adımları ortaya koyabilmek gerekir. Araştırmacıların var olan sistemleri incelemelerinin yanı sıra yeni üretimlere yönelmeleri kaçınılmaz bir ihtiyaçtır. Siber savaşın hız kazandığı günümüzde bilgi güvenliğinin oldukça önemli hale gelmiş olması sonucu, bilginin bulunduğu ortamlarda olduğundan farklı bir şekilde tutulması ve bu sistemlerin kendilerini sürekli yenileme ihtiyacı içerisinde olması kriptografi alanında yeni üretimlere ihtiyaç olduğunu göstermiştir.

Çalışma kapsamında konu ile ilgili tanıtıcı bilgilere yer verilmiş ve küçük bir uygulama projesi geliştirilmiştir. Kriptoloji algoritmalarının temelini klasik teknikler oluşturmaktadır. Bu çalışma kapsamında da klasik teknikler kullanılarak bir şifreleme algoritması geliştirilmiştir. Şifreleme algoritmaları üzerlerinde yapılan çözümleme analizlerinden dolayı daima geliştirilmeye açıktır. Çalışma kapsamında geliştirilen algoritma da geliştirilmeye açıktır. Geliştirilen algoritmanın simülasyonu için Visual Studio 2010 paket programı C# dili kullanılarak Windows Formlardan oluşan demo program hazırlanmıştır. Algoritmanın çözüm anahtarı olarak biyometrik bir özellik olan ses tercih edilmiştir. Algoritmanın güvenirliğini tespit etmek için harf frekans analizi saldırısı incelemeleri yapılmıştır. Hazırlanan algoritma klasik tekniklere dayandığı için harf frekans analizi saldırısının daha uygun olacağı düşünülmüş ve bu nedenle harf frekans analizi saldırısı tercih edilmiştir. Algoritmanın modern tekniklere göre durum incelemesini yapmak için, DES algoritması ile performans analizlerinin karşılaştırması yapılmış ve performans analizlerinden elde edilen sonuçlara yer verilmiştir.

Kriptografi algoritmaları genel olarak yer değiştirme, yerine koyma ve cebirsel yöntemler olmak üzere üç temel yönteme sahiptir. Geliştirilen algoritmada bu üç temel yöntem kullanılmıştır. Bu yöntemler kullanılarak üretilen farklı kriptografi teknikleri vardır. Geliştirilen algoritmada bu farklı tekniklerden Sezar şifreleme tekniğinin harf kaydırma(öteleme) prensibi, Alberti Diskinin harf kaydırma miktarının sabit olmaması özelliği, ENİGMA'nın üst üste şifreleme işlemi yapması, vigenére şifreleme tekniğinin farklı şifre alfabe prensibi, klasik tekniklerin çoğunun ortak özelliği olan mod alma işlemi özellikleri, bir araya getirilmiş ve uygulanmıştır.

Geliştirilen algoritmanın modern kriptografi tekniklerine dayanmaması en büyük dezavantajdır. Demo program için Windows XP SP3 işletim sistemine sahip Dell Insprion 6000 dizüstü bilgisayar kullanılmıştır. Yapılan kaynak taramasında ulaşılabilen dökümanlar incelenmiş ancak bazı kaynak dökümanlara erişim izni olmadığı için ulaşılamamıştır. Çalışma esnasında konu ile ilgilenen kurum ya da kurum yetkilileri ile görüşülmemiştir. Çalışma, veritabanlarında ve ağ üzerindeki ulaşılabilen dökümanlar ile kitap, dergi, tez, makale, bildiri gibi basılı materyaller ve dijital verilerin incelenmesi ile gerçekleştirilmiştir. Çalışma donanımsal boyuta taşınmamıştır, ancak çözüm anahtarı olarak ses özelliğinin kullanılması sebebiyle donanımsal boyuta taşınabilir olduğu düşünülmektedir.

Tez çalışmasının ikinci bölümünde bilgi ve bilgi güvenliğine ait genel bilgiler yer almaktadır. Kriptoloji bilimi, bilgi güvenliğini sağlamak için kullanılmaktadır. Bu nedenle bilgi ve bilgi güvenliğinin genel çerçevesi ikinci bölümde verilmiştir.

Tez çalışmasının üçüncü bölümünde kriptoloji bilimine ait genel bilgiler sunulmuştur. Kriptoloji bilimi alt bilim dalları olan kriptografi ve kriptoanaliz anlatılmıştır. Tez çalışmasının temelini oluşturan klasik teknikler örnekleriyle birlikte açıklanmıştır.

Tez çalışmasının dördüncü bölümünde araştırmanın yöntem ve teknikleri anlatılmıştır. Kriptografi algoritmalarının üretilmesinde kullanılan üç temel yöntem örnekleri ile açıklanmıştır. Algoritmanın yazılımsal olarak hazırlanmasında kullanılan programlama dili ve kullanılan bilgisayarın donanımsal özelliklerinden bahsedilmiştir.

Tez çalışmasının beşinci bölümünde geliştirilen algoritma ve demo programın tanıtımı yapılmıştır. Algoritmanın işlem basamakları örnek ve çizelgelerle açıklanmıştır. Demo program için ekran görüntüleri ve aktivite diyagramları verilmiştir.

Tez çalışmasının altıncı bölümünde algoritmanın harf frekans analizine karşı güvenirlik incelemesi yapılmıştır. Geliştirilen kriptografi algoritmalarının güvenirlik incelemeleri yapılmalıdır. Tez kapsamında geliştirilen algoritma klasik teknikler incelenerek hazırlandığı için güvenirlik incelemesinde en uygun yöntem harf frekans analizi saldırısıdır. Altıncı bölümde harf frekans analizi incelemeleri ve elde edilen bulgular yer almaktadır.

Tez çalışmasının yedinci bölümünde algoritmaların performans analizi ile ilgili tanıtıcı bilgiler verilmiştir. Geliştirilen algoritmanın kendi içinde (kelime ve paragraf düzeyinde şifreleme işlemleri) performans analiz değerlendirmeleri yapılmıştır. Aynı şekilde modern bir kriptografi algoritması olan DES algoritmasının performans analizi yapılmış ve performans analizlerinden elde edilen tüm sonuçlar karşılaştırılmıştır.

Tez çalışmasının son bölümünde konu dahilinde yapılan tüm çalışmalardan elde edilen sonuçlar değerlendirilmiş ve değerlendirmeler dahilinde çeşitli önerilerde bulunulmuştur.

2. BİLGİ VE BİLGİ GÜVENLİĞİ

Bilgi günümüzdeki en büyük güç haline gelmiştir ve güvenliğini sağlamak her geçen gün daha da zorlaşmaktadır. Bu bölümde bilginin ne olduğundan ve öneminden, bilgi güvenliğinin öneminden bahsedilmiştir. Ayrıca bilgi savaşı ve bilgi güvenliğini tehdit eden unsurlardan kısaca söz edilmiştir.

2.1. Bilgi ve Önemi

Veri, kurumların günlük işlerinin, kayıtlarının ve birikimlerinin sonucu elde edilen ve karar verme aşamasında da kullanılabilen işlenmiş veya işlenmemiş bilgi anlamına gelmektedir [1]. Bilgi genellikle, bireyler veya kurumlar tarafından bir sorunun çözümü, herhangi bir çalışmanın başlatılması ya da bitirilmesi gibi faaliyetler sonucunda ortaya çıkarılan anlamlandırılmış verilerin bütününü ifade etmektedir [2].

Bilgi, bakış açısına göre farklı anlamlar kazanan içerik bütünüdür. Çok boyutlu bir varlıktır. Ham gerçeklerin işlenmiş hali yani işlenmiş veridir. Bilinmezliğin ve belirsizliğin karşıtıdır. İnsanlar ve makineler arasındaki anlamlı sinyaller bütünüdür. Bilgi, gelişen dünyaya uyum sağlayabilme ve hayatta kalabilme yeteneğinin öncüsüdür. Bilgiye bu açıdan bakıldığında insanlar hatta toplumlar arasında rekabeti arttırdığı için bilgi güçtür [3].

Bilgi güç anlamına geldiği için, içerisinde yönetme yeteneğini de içerir. Bu nedenle toplumlar, birbirlerine üstünlük sağlayabilmek için ağır ve mekanik silahları kullanmak yerine 1990'lı yılların başlarında yeni bir kavram olarak dünya literatürüne giren, maddi ve manevi açıdan daha az kayıpla gerçekleştirilebilen sessiz savaşı yani *Bilgi Savaşı'nı* (information warfare) tercih etmeye başlamıştır. Çok eskiden ağır silahlarla yapılan savaşların kazanılmasında yardımcı rolü oynayan bilgi casusluğu, günümüzde savaşların ana kahramanıdır. Bilginin güvenli şekilde elde tutulma süresi ve sürekli olarak güncellenmesi toplumların üstünlüklerinin ya da özgürlüklerinin

devamı anlamını taşıdığı için bilginin iyi korunması toplumların hayatta kalması için en önemli unsur haline gelmiştir [3,4].

Bir bilginin değerli mi yoksa değersiz mi olduğunu ortaya koyan değer, bilgiyi elinde tutan kişi ya da kurumların yaşamsal faaliyetleri için önem derecesidir. Ülkeler bazında düşündüğümüzde bir ulusun güvenliği ve devamı için, o ulusun elindeki milli istihbarat verilerini bilgiye dönüştürmesi ve elde ettiği bilgileri en iyi şekilde koruması gerekmektedir [5,6].

2.2. Bilgi Güvenliği ve Önemi

Güvenlik can ve mal varlıklarının her türlü tehdit ve tehlikelerden korunması şeklinde tanımlanabilir [1].

Yapılmış olan *güvenlik* tanımlaması ve Bölüm 2.1'de yapılan *bilgi* tanımlamasından hareket ederek bilgi güvenliği, verinin toplanması, son kullanıcıya ulaşması, saklanması ve kullanımı aşamalarında her türlü tehdit ve tehlikelerden korunması, bu amaçla önceden alınacak tedbirler ve saldırı halinde yapılabilecek işlemlerin tümünü kapsayan bir disiplin olarak tanımlanabilir [1]. Daha açıklayıcı bir ifadeyle bilgi güvenliği; elektronik ortamlarda verilerin veya bilgilerin saklanması ve taşınması esnasında bilgilerin bütünlüğü bozulmadan, izinsiz erişimlerden korunması, sürekli olarak erişilebilirliğin sağlanması için güvenli bir bilgi işleme platformu oluşturma çabasıdır [3, 7].

Günümüzde günlük hayata dair bütün işlemler sanal ortam üzerinden gerçekleştirilebilmektedir. Hemen her evde internet kullanımının yanı sıra kablosuz ağlar ve mobil cihazların da yaygınlaşması ile sanal ortamın insan hayatına dahil olmadığı tek bir an bile yoktur. Bu durum insanoğlu için büyük kolaylıklar sağlamakla birlikte ona büyük sorumluluklar da yüklemiştir.

Sanal ortam kavramı her geçen gün biraz daha genişlemektedir. Bu genişlemeyi sağlayan en önemli özellik internet kavramıdır. Bilgisayar, telefon, elektrik hatları gibi pek çok ortamda kullanılmaya başlayan internet

tüm dünyadaki harita sınırlarını ortadan kaldırmıştır. Bilginin korunması daha güç hale gelmiş ve her geçen sürede yeni yöntemlere ihtiyaç duyulmuştur. Geçmişten günümüze konu ile ilgili gelişmelere bakıldığında bu alandaki gelişmelerin devasa olduğu ve fakat buna rağmen halen yeterli olmayıp sürekli kendini yenileme ihtiyacı içinde olduğu görülmektedir. Kurumsal ve kişisel bilgilerin elektronik ortamlarda arşivlendiği ve elektronik ortamlar üzerinden hizmet sağlanmasının yaygınlaştığı göz önünde bulundurulursa bilgi güvenliğinin önemi daha iyi ortaya çıkar [8].

Bilgi güvenliğinde mükemmel güvenlikten bahsedilemez; çünkü bu durum söz konusu değildir. Bu yüzden problemleri tespit edebilmek ve bu problemlere karşı önlem üretebilmek, güvenlik zaaflarıyla ilgili senaryolar üretip buna göre güvenlik yöntemi geliştirmek tercih edilmelidir [6,9].

İyi bir bilgi güvenliği, bütün güvenlik çözümlerinin bir araya getirilmesiyle oluşur. Dolayısıyla, verinin güvenliğini sağlamak için birden çok güvenlik çözümüne ihtiyaç vardır. Tek bir güvenlik çözümü veri ve bilgisayar sistemlerinin güvenliğini tam olarak sağlayamaz ve yeterli bir güvenlik sistemi olarak düşünülemez [1]. Kriptoloji bilimi, bilgi güvenliğini sağlamanın yollarından birisidir. Kriptoloji bilimi ile ilgili genel bilgiler Bölüm 3’te verilmiştir.

Bilgi güvenliği bireysel bilgi güvenliği, kurumsal bilgi güvenliği ve ulusal bilgi güvenliği olmak üzere üç başlık altında incelenebilir:

2.2.1. Bireysel (kişisel) bilgi güvenliği

Kişilerin elektronik ortamdaki bilgileri, yazılımların içerisinde gizlenen kötü amaçlı kodlar ile ele geçirilebilir, tahrip edilebilir veya kötü amaçlı kullanılabilir [10]. Kişisel bilgilerin; örneğin kimlik bilgileri, banka hesap bilgileri, kredi kartı bilgileri, telefon numarası, adres, fotoğraflar gibi bilgilerin başka kişilerce istenmeyen durumlarda kullanılması ve bu istenmeyen durumların geniş kitlelere ulaşması günümüzün en büyük sorunlarından birisidir. Kimlik bilgileri ile pornografik sitelerde kullanıcı hesabı açmak, fotoğrafları cinsel içerikli

sitelerde uygunsuz kullanmak, banka hesaplarını boşaltmak, kredi kartı bilgileri ile alışveriş yapmak yaşanan olumsuzluklardan bazılarıdır [11].

2.2.2. Kurumsal bilgi güvenliği

Her birey bilgi sistemleri üzerinden hizmet alırken veya hizmet sunarken kurumsal bilgi varlıklarını doğrudan veya dolaylı olarak kullanmaktadır. Bu hizmetler kurumsal anlamda bir hizmet alımı olabileceği gibi, bankacılık işlemleri veya bir kurum içerisinde yapılan bireysel işlemler de olabilir. Kurumsal bilgi varlıklarının güvenliği sağlanmadıkça, kişisel güvenlik de sağlanamaz [12].

Kurumsal bilgi güvenliği, kurumların bilgi varlıklarının tespit edilerek zafiyetlerinin belirlenmesi ve istenmeyen tehdit ve tehlikelerden korunması amacıyla gerekli güvenlik analizlerinin yapılması ve önlemlerin alınması olarak düşünülebilir [12].

Kurumsal bilgi güvenliği insan faktörü, eğitim, teknoloji gibi birçok faktörün etki ettiği tek bir çatı altında yönetilmesi zorunlu olan karmaşık süreçlerden oluşmaktadır. Bu süreçlerin yönetilmesi, güvenlik sistemlerinin uluslararası standartlarda yapılandırılması ve yüksek seviyede bilgi güvenliğinin sağlanması amacıyla tüm dünyada kurumsal bilgi güvenliğinin yönetiminde standartlaşma çalışmaları hızla sürmektedir [12].

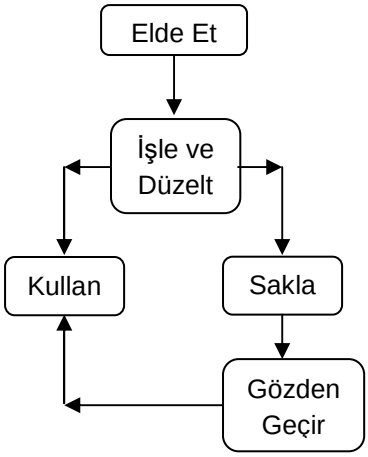
2.2.3.Ulusal bilgi güvenliği

Kamu veya özel kuruluşlara ait ağ destekli bilgi sistemlerinin birbirleriyle etkileşimi her geçen gün artmakta ve bu sistemler ülke bilgi sistemlerinin altyapısını oluşturmaktadır. Ülke bilgi sistemleri ülke güvenliği açısından önemli olan stratejik ülke bilgilerinin, ilgili kurumların kendi içinde veya başka kurumlar arasında paylaşılmasını ve kullanılmasını sağladığından ülke bilgi sistemlerinin güvenliğinin yüksek seviyede sağlanması ülke güvenliği açısından önemlidir. Kişilerin ve kurumların sahip oldukları önemli bilgilerin

yer aldığı ülke bilgi sistemleri istihbarat veya terör amaçlı yapılabilecek siber saldırılara karşı yüksek seviyede korunması gerekmektedir. [3].

2.3. Bilgi Savaşı

Bilgi üzerinde birbiriyle bağlantılı bir işlem zinciri vardır. Öncelikli olarak bilgi elde edilir. Elde edilen bilgi işlenir, gerekli düzeltmeler yapılır ve kullanılabilir hale getirilir. Kullanılabilir hale getirilen bilgi daha sonra gerektiğinde kullanılmak üzere saklanır. Son adımda bilgi tekrar gözden geçirilerek gereken düzeltmeler yapılır ve kullanılmak üzere iletilir. Bu zincir bilgi savaşlarının temelini oluşturmaktadır. Günümüzde bu işlemleri yapabilmek ve sistemleri ele geçirmek için siber ordular kurulmuştur. Gelişen teknoloji ve değişen şartlara göre bilgi savaşları da elektronik ortama taşınmış ve sanal ortamda siber savaş olarak kendini göstermiştir [3,4].



Şekil 2.1. Bilgi işlem zinciri

Bilgi savaşı, askeri boyutu olmakla birlikte daha çok bilgi sistemlerini çökertmeye yönelik sanal dünyada gerçekleşen savaşları tanımlar. Bu nedenle siber savaş ya da net savaşı olarak da ifade edilebilmektedir [13].

Bilgi sistemleri, bilginin iletimi, paylaşımı, saklanması, yedeklenmesi için oluşturulmuş altyapılardır [13]. Bu altyapılar kullanılarak; insan zayiatı verilmeden, bina, tesis ve diğer alt yapılara hasar oluşturmadan yapılan

savaşa bilgi savaşı denilmektedir. Elektronik ortamda hayat bulur. Sayısal veriler üzerinden yürütülür. Sanal ortamdaki bilgi savaşı özellikle yazılım ağırlıklıdır. Askeri, politik, ekonomik ya da kişisel bilgilerin elde edilmesi amacıyla bir bilgisayar sistemine ya da ağına *gizlice zararlı bilgisayar yazılımı* yerleştirilmesidir. Saldırgan, bir ülkenin silahlı kuvvetleri, bir terörist organizasyon, uluslararası bir şirket ya da bir şahıs olabilir [10].

Bilgiyi elinde tutan her zaman bir adım öndedir ve olayları istediği gibi yönlendirebilir. Bu nedenle siber savaş kavramı geliştikçe bilgiye verilen önem giderek artmıştır. Siber savaşta taraflardan biri bilgiyi elde etmek isterken diğeri bilgiyi korumak zorundadır. Bu nedenle taraflar karşıya saldıran ya da kendisini savunan taraf olarak her şekilde siber savaş ordusuna ihtiyaç duymuştur. Bu ihtiyaç ülkelerin savunma bakanlıkları bünyesinde yeni nesil askerler olan siber savaş orduları kurulmasına öncülük etmiştir. Elektronik ortamda gerçekleşen siber savaşa karşı eğitimler önem kazanmıştır [14].

Virüs programlarının sürekli gelişmesi, düşmanın bilgisayar sistemlerini etkisiz hale getirmek amacıyla virüs kullanılmasını gündeme getirmiştir. Ülkelerin, düşmanın sivil altyapısını imha etme yeteneğine sahip yıkıcı *yazılımlar* geliştirmesi ihtimali çok yüksektir. Bu yazılımlar ile enerji ağı köreltilebilir, ülkenin parasal kaynakları emilebilir, ulaşım ve iletişim altyapısı çökertilebilir. Bu tip askerî uygulamalar, bilinen bilgisayar virüslerinden farklıdır. Geliştirme programlarının çoğu son derece gizli tutulur [10].

Yapılan herhangi siber saldırı sonucu karada ya da havada mücadele veren askerler ellerindeki silahlar üzerindeki kontrolü kaybedebilmektedir. Ülke içerisindeki elektrik, su, doğalgaz gibi şehir şebekeleri, telefon, uydu, telsiz gibi haberleşme sistemleri kısmi ya da tam olarak işlevselliğini yitirebilmektedir. Siber savaşçılar, tüm dünyayı saran ağ yapısı sayesinde elektronik sistemlere sızarak tüm sistemleri kilitleyebilmektedir. Bu durumun önüne geçmek için uygulanabilecek yöntemlerden biri bilgileri açık bir şekilde sisteme girmek yerine bilgileri gizleyerek(şifreleyerek) sisteme yerleştirmektir.

Ayrıca yetkili olmayan kişilerin sisteme girişini zorlaştıran anahtarlar kullanılmalıdır. Ülkelerin kendi sistemlerini oluşturmaları da herhangi siber saldırıda kendilerini savunmaları için zaman kazanmalarını sağlayacaktır [13,14].

Bilgininin gizliliğinin korunması için yapılabilecek öncelikli işlem kripto sistemleri kullanmaktır. Kripto sistemlerinin milli imkânlar kullanılarak hazırlanması çok önemlidir. Türkiye'de askeri sistemlerde kullanılan bütün kriptolar milli olarak geliştirilmektedir. Gelecekte oluşabilecek bir savaşta, kullanılacak yazılımların kontrolünü, komuta kontrol ağına girebilecek düşman kuvvetleri ele geçirmeyi amaçlayacaktır. Savunma sistemlerinde teknoloji hızın ve hassasiyetin arttırılması yönünde geliştirilmektedir. Gelecekte düşman hakkında daha önce bilgi edinen, bu bilgileri istihbarat merkezlerine daha hızlı ulaştıran, bu istihbaratları daha hızlı düzenleyip komutana sunan, komutan emirlerini birliklere ve silah sistemlerine daha hızlı ulaştıran, hedefi daha hızlı ve hassas olarak vuran taraf savaşı kazanacaktır. Bu nedenle içinde bulunulan yüzyıl ve sonrası için bilgi savaşları önemlidir. [10].

Bilgiyi koruyabilmenin en iyi yolu onu tehlikeye atacak durum ve olayları en iyi şekilde kavramak, bu tehdit ve tehlikelere karşı tedbir almaktır. Bilgi güvenliğini tehdit eden unsurlar Bölüm 2.4'te verilmiştir.

2.4. Bilgi Güvenliğini Tehdit Eden Unsurlar

Kurum ve şahısların sahip oldukları tüm değer ve bilgilere; izinsiz erişmek, zarar vermek, verileri silmek veya değiştirmek, maddi/manevi kazanç sağlamak için bilişim sistemleri kullanılarak yapılan her türlü hareket dijital saldırı olarak tanımlanabilir. Tehditler ve saldırılar kendisini yeniledikçe bilgi güvenliği de kendini yeniler ve kendisine yeni sınırlar çizer [2,6,7].

Bilgi güvenliğini tehdit eden unsurlar:

- Başkası tarafından ele geçirilme(gizliliğin ihlali)

- Bilgide değişiklik yapmak(bütünlük ihlali)
- Erişilebilirlik/kullanılabilirlik problemi

olmak üzere genel olarak sınıflandırılabilir [7]. Bu tehdit unsurları Bölüm 2.5'te tanımlanmış olan izinsiz erişim, engelleme veya zarar verme, değişiklik yapma ve üretim saldırıları sonucu ortaya çıkar.

2.4.1. Gizlilik ihlali

Bölüm 2.3'te verilen Şekil 2.1'deki süreçlerin birisinde ya da birden fazlasında gizlilik ihlali yaşanabilir. Gizlilik ihlali, yetki tanımı yapılmış kişilere ait kullanıcı adı ve şifre bilgilerinin yetki tanımı olmayan kişilerce izinsiz olarak elde edilmesi ve kullanılması sonucu bilgiye izinsiz erişim sağlanmasıdır [13,15].

2.4.2. Bütünlük ihlali

Bilginin saklandığı yerde ya da iletimi sırasında yetkisiz bir şekilde bir kısmının ya da tamamının değiştirilmesidir [13,15].

2.4.3. Erişilebilirlik/kullanılabilirlik problemi

Bilgiye ihtiyaç duyulduğu anda bilginin kullanıma hazır olması durumu erişilebilirlik özelliğini tanımlamaktadır. Yetki tanımlaması yapılmış kişi ve kurumların ihtiyaç duyduklarında bilgiye ulaşmalarının mümkün olmaması, engellenmesi bilgi güvenliğini tehdit eden unsurlar arasındadır [13, 15].

2.5. Bilgi Güvenliğini Tehdit Eden Saldırı Türleri

Bilgi güvenliğini tehdit eden saldırı türleri genel olarak dört başlık altında incelenmektedir.

2.5.1. İzinsiz erişim

Bu saldırı türünde, saldırgan bilgiye (yazılım, donanım ve veri) yetkisi olmadığı halde erişebilmektedir. Aynı bilgiye yetkili kullanıcılar da olağan şekilde erişebilirler, yani bilginin kendisinde bir bozulma yoktur. Bununla

birlikte o bilgiye erişmesi beklenmeyen kişilerin bunu yapabilmesi, saldırı olarak nitelendirilir (örn: ağ koklama). En tehlikeli tehditlerin başında gelir. Pasif ataktır [6, 9, 16].

2.5.2. Engelleme veya zarar verme:

Bu saldırı türünde, bilgiye erişim engellenir. Bilgi ya kaybolmuş, silinmiştir; ya da kaybolmamıştır ama ulaşılamaz durumdadır. Bilgi kaybolmamış ve ulaşılabilir durumdaysa sadece yetkili kullanıcılar tarafından kullanılamaz durumdadır (örn: DoS veya DDoS gibi erişim reddi saldırıları). Aktif ataktır [6, 9, 16].

2.5.3. Değişiklik yapma

Bu saldırı türü, bilginin yetkili kullanıcıya ulaşmadan önce saldırganın amaçları doğrultusunda bilgide değişiklik yapmasını içerir. Program kodları, durgun veri veya aktarılmakta olan veri üzerinde yapılması mümkündür (örn: virüsler ve truva atları). En tehlikeli ataklardan birisidir. Aktif ataktır [6, 9, 16].

2.5.4. Üretim

Bu saldırı türü, gerçekte olmaması gereken verinin üretilmesini içerir. Üretilen veri, daha önceki gerçek bir verinin taklidi olabileceği gibi, gerçeğe uygun tamamen yeni bir veri şeklinde de olabilir (örn: sahte veri, ya da veri taklidi). Aktif ataktır [6, 9, 16].

Bu saldırıları gerçekleştirmek için virüsler, kurtlar, Truva atı, mantık bombaları, arka kapılar, mikroçipler, nano makineler, HERF silahları, EMP bombaları ve gelişen teknolojiyle ortaya çıkan yeni saldırı mekanizmaları kullanılmaktadır [13].

2.6. Bilgi Güvenliğini Sağlama Yolları

Bilgi güvenliğini sağlamak için genel olarak aşağıdaki işlemler yapılabilir:

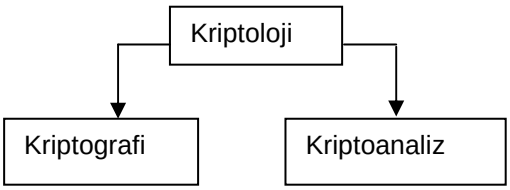
- Bilgi bulunduran ortamların korunması
- Fiziksel güvenlik tedbirlerinin önemsenmesi
- İnsan faktörünün dikkatlice ele alınması
- Güvenlik duvarlarının kullanılması
- Anti virüs yazılımları ile sistemlerin korunması
- Sayısal imza ve açık anahtar alt yapısının kullanılması
- Atak tespit sistemlerinin kurulması
- Şifreleme metotlarının uygulanması
- Güvenlik politikalarının belirlenmesi ve uygulanması [6]

Bu tez kapsamında kriptoloji bilimi kriptografi alt bilim dalı üzerinde çalışılmıştır.

3. KRİPTOLOJİ

Şifre bilimine kriptoloji denilmektedir. Kriptoloji, bilgilerin şifrelenmesi ve şifrelenmiş bilgilerin çözümlenmesi için metotlar geliştirir. Yunanca *saklanmış / gizli* anlamına gelen *kriptos* ve *söz, bilgi, bilim* gibi birçok anlamı olan *logos* kelimelerinden türetilmiştir [17,18].

Şekil 3.1'de görüldüğü gibi kriptoloji bilimi iki alt bilim dalına ayrılmaktadır:



Şekil 3.1. Kriptoloji bilimi alt bilim dalları

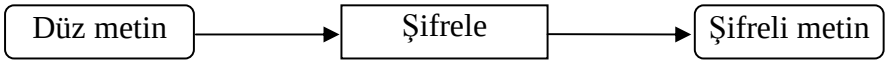
Şekil 3.1'de görülen kriptoloji bilimi alt bilim dalı Kriptografi, açık olan bilgiyi anlaşılamaz hale getirmek için kullanılan şifreleme bilimidir. Kriptoanaliz ise şifrelenmiş bir metnin analiz süreçleri üzerinde duran ve açık metni elde etmek için kullanılan yöntem ve teknikleri içeren kriptoloji alt bilim dalıdır [6].

3.1. Kriptografi Nedir

Kriptografi; bilgi güvenliğinde gizlilik, kimlik denetimi, bütünlük gibi kavramların sağlanması için çalışan matematiksel yöntemler bütünüdür. Bu yöntemler, bilginin bulunduğu ya da kullanıldığı ortamda karşılaşılabilecek saldırılardan bilgiyi, bilgi göndericisini ve alıcısını koruma amacı taşır [17].

Kriptografi (şifreleme), düz metinin içeriğini, anahtar bilgisi olmadan okunamayacak hale getirerek iletilen metinin güvenliğini sağlar. Anahtar, metini şifrelemek ya da şifrelenen metini düz metin haline getirmek için kullanılan sayı, harf, kelime, sembol, fonksiyon gibi herhangi bir veri parçası olabilir [19].

Şekil 3.2 ile şifreleme işlemi sembolik olarak anlatılmaktadır. Şifreleme işlemi kullanılan yönteme göre çok farklı şekillerde gerçekleştirilebilir. Düz metinin geçirdiği değişimler sonucu şifreli metine ulaşılır:



Şekil 3.2. Şifreleme işlemi [6]

Şekil 3.2'de görüldüğü gibi kriptografi verilerin açık halden kapalı yani gizli hale getirilmesi işlemidir. Verilerin gizliliğini, bütünlüğünü, güvenliğini sağlar. Bu işlemi yapan kişilere kriptograf denir. Eldeki metnin anlaşılabilir haline düz metin ya da açık metin adı verilir. Düz metinin farklı işlemlerden geçirilerek anlaşılamayacak bir yapıya dönüştürülmesi sonucunda elde edilen yeni haline şifreli metin adı verilir [17,19].

Bir şifreleme algoritmasının taşıması gereken temel özelliklerden bazıları aşağıda tanımlanmıştır:

- *Gizlilik:* Bilgi sadece ilgili kişiler tarafından anlaşılmalıdır, bilgi anlaması istenmeyen kişiler tarafından anlaşılmamalıdır [15, 20, 21].
- *Bütünlük:* Bilginin herhangi bir değişikliğe uğrayıp uğramadığını ifade eder. Bilginin saklanması ya da iletilmesi esnasında yetkili kişiler dışında başka kişilerce değiştirilememesidir [15, 20, 21].
- *İnkar edememe:* Bilgiyi üreten ya da ileten kişinin daha sonrasında bunu inkar edememesi durumudur [15, 20, 21].
- *Kimlik doğrulama:* Bilginin iletiminde karşılıklı iki tarafta da kimlik doğrulama kullanılırsa üçüncü bir kişinin bilgiye ulaşmasını engellemek için kimlik doğrulama bilgilerine ulaşmaması gereklidir [15,21].

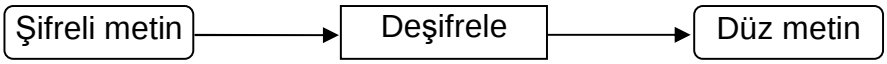
- *Erişilebilirlik/Süreklilik:* Yetkili kişilerin ihtiyaç duyduğu anda bilgiye ulaşabilmesidir [15, 20].

Bu temel özelliklere göre, kriptografi verilerin gizlenmesinin yanı sıra veri bütünlüğü, kimlik kanıtlama ve inkar edememe konuları ile de ilgilidir.

3.2. Kriptoanaliz Nedir

Kriptoanaliz şifrelenmiş metinden düz metni elde etmek için uygulanan yöntemler bütünüdür. Şifreli bir metinden düz metinin elde edilmesi işlemi deşifreleme olarak adlandırılır. Kriptoanalizle uğraşanlara ise kriptoanalist denir. İyi bir kriptoanalist aynı zamanda iyi bir kriptografi uzmanı olmalıdır. Çünkü şifrelerin nasıl çözülebileceğini bulmak için bu şifrelerin nasıl oluşturulabileceğini iyi bilmek gereklidir. Tarih boyunca kriptoanalistlerin başarısı kriptografları yeni ve daha güvenli sistemler tasarlamaya zorlamıştır [17].

Şekil 3.3 ile şifrelenen metinin deşifreleme işlemi sembolik olarak anlatılmaktadır. Deşifreleme işlemi için farklı yöntemler mevcuttur, uygun olan yöntem seçilerek düz metine ulaşılabilir:



Şekil 3.3. Şifre çözme işlemi [6]

Kriptoanaliz ile şifrelenmiş metini çözmek için kullanılacak anahtar bilinmeden şifrelenmiş mesaj çözülmeye çalışılır. Kriptoanaliz çalışmaları ilk kez ilkel şifrelerin kırılma çabalarıyla ortaya çıkmıştır. Basit şifre sistemlerini çözmek için olası tüm alfabeleri denemek yerine daha kolay şekilde çözebilmek için matematik, istatistik ve dilbilim alanlarında yeterli bilgiye sahip olmak gereklidir [3, 22, 23].

Günümüze ulaşan ve kriptoanalizin ilk örneklerini içeren eser, 9. yüzyılda yaşayan Arap filozofu Al-Kindi'nin yazdığı *Kriptografik Mesajların Deşifresi* (Risâle fi’stihrâci’l-mu’ammâ) isimli eserdir. Bu eser İstanbul'da, Süleymaniye Osmanlı Arşivi'nde bulunmaktadır. Eser ile frekans analizi kavramı ortaya çıkmıştır. Al-Kindi'nin kriptoanaliz tekniğine göre, hangi dilde yazıldığı bilinen şifreli bir mesajı çözmek için, aynı dilde yazılmış uzun bir metin bulup her bir harfin kullanım sıklığını hesaplamak gereklidir. Düz metinde en sık kullanılan harf, şifreli metinde en sık kullanılan harfe denk gelecek şekilde eşleme yapılmaktadır [3, 22, 23].

3.3. Kriptolojinin Önemi

Bilgi günümüz rekabet ortamında giderek daha değerli hale gelmektedir. İletişim teknolojilerindeki gelişmeler bilgiyi saklama ve iletme açısından işleri zorlaştırmakta ve yeni yöntemlere duyulan ihtiyaç artmaktadır. Uydu ve internet üzerinden yapılan tüm iletim ve haberleşmenin dinlenme ve izlenme olasılığı artmaktadır. Gizliliği mühim olan şahsî, ticarî, politik ve askerî belgelerin elektronik ortamlarda taşınması ve iletilmesi ile kriptografinin önemi daha da artmıştır. Sanal ortam ve internet ağının giderek yaygınlaşması teknoloji kullanımının oldukça geniş bir kitleye hitap etmesini sağlamıştır. Bu durum kişisel, kurumsal ve ulusal bilgi güvenliği için büyük tehditler oluşturmuştur. Sanal ortam ve internet ağının oluşturduğu yeni bir kavram olan siber dünya bilgiye ulaşımı ne kadar kolaylaştırdıysa güvenlik riskini de bir o kadar arttırmıştır. Bilgiyi olduğundan farklı şekilde yani gizleyerek kullanmak bu riskleri azaltmak için kullanılabilecek yöntemlerden biridir. Bilgiyi gizlemenin yollarından birisi kriptoloji bilimine ait kriptografi alt bilim dalıdır [6,10, 24-26].

Kriptografi yani şifreleme bilimi günümüzde güncel hayatın pek çok alanında kullanılmaktadır. Örneğin bankamatikler, rezervasyon sistemleri, e-posta iletileri, telefon bankacılığı, normal bankacılık işlemleri, cep telefonları, uydu sistemleri, füze sistemleri, savaş uçakları, askeri bilgiler, kriminal olaylar, sağlık sistemleri, mimari projeler, fatura sistemleri, şehir şebekeleri, e-ticaret,

e-devlet, e-imza, üyelik sistemleri gibi daha pek çok alanda kullanılabilmektedir. Kişilere ve kurumlara kolaylık sağlamak için tüm güncel faaliyetlerin elektronik ortama aktarılışı tehlikelerin giderek büyümesine sebep olmaktadır. Herhangi bir bilgi sızıntısı ya da değişikliğinin kötü niyetli kullanımının doğurabileceği sonuçlar düşünülürse günümüzde bilgi güvenliğinin artan şekilde hayati önem taşır hale geldiği görülebilir. Bilgi güvenliğini sağlamanın yollarından biri kriptoloji bilimi ile ilgilidir. Elektronik ortamlarda kriptografi olmadan güvenli iletişim gerçekleştirilemez. Kriptografi, elektronik ortamlarda kimlik doğrulamayı ve gizliliği sağlar. Bu nedenle kriptografi olmadan güvenli elektronik iletişim sağlanamaz [27-30].

Kişisel bilgilerin; örneğin kimlik bilgileri, banka hesap bilgileri, kredi kartı bilgileri, telefon numarası, adres, fotoğraflar gibi bilgilerin başka kişilerce istenmeyen durumlarda kullanılabilmesi ve bu istenmeyen durumların geniş kitlelere ulaşması günümüzün en büyük sorunlarından birisidir. Kimlik bilgileri ile pornografik sitelerde kullanıcı hesabı açmak, fotoğrafları cinsel içerikli sitelerde uygunsuz kullanmak, banka hesaplarını boşaltmak, kredi kartı bilgileri ile alışveriş yapmak yaşanan olumsuzluklardan bazılarıdır. Kurumsal bilgilerin kurum dışına sızdırılması ya da çalınması kurumların itibarını zedeleyen veya maddi anlamda büyük zararlar veren sonuçlarla sonlanmaktadır. Ulusal düzeyde bilgi güvenliği ulusların devamı için büyük bir önem taşımaktadır. Günümüzde ağır silahlarla yapılan mücadele yerini siber dünyadaki mücadeleye bırakmaya başlamıştır. Siber savaş olarak nitelendirilen bu durum ulusal bilgilerin güvenli bir şekilde korunması ihtiyacını arttırmıştır. Bu ve bunlar gibi sebeplerle siber dünyada kullanılan, paylaşılan, iletilen ya da saklanan veriler açık bir şekilde sistemlere girilmemeli, gerekiyorsa başka formlara çevrilerek ya da gizlenerek sanal ortama aktarılmalıdır [10, 11, 12].

Bu sonu gelmeyen döngüde ülkeler arası bilgi hırsızlığı yaygınlaşmıştır. Siber dünya, ülkeler ve bölgeler arasındaki harita sınırlarının ortadan kalkmasını sağlamıştır. Bu durum gerçek dünyada aşılamayan harita sınırlarının elektronik ortamda bilgi hırsızlığı ile aşılabilmesini kolaylaştırmıştır. Son

dönemde yaşanan kurumsal web sitelerinin ve sayfalarının yetkisiz(izinsiz) kişilerce erişimi sonucu açığa çıkmaması gereken pek çok doküman herkesin ulaşabileceği şekilde açığa çıkarılmıştır. Bu durum oldukça düşündürücüdür, çünkü ülkelerin milli güvenliği için akıl almaz bir tehlikenin her an gelebileceğini göstermektedir. Bilgilerin şifrelenmiş halde saklanması bilgi güvenliğini sağlamak için tek başına yeterli olmamakla birlikte olası tehlikeleri daha aza indirgemek için gereklidir [8,10].

Günlük hayatta işlemlerimizi kendimiz hallederken duymuş olduğumuz güven duygusunun sanal ortamdaki güvencesi kriptoloji bilimi ile sağlanabilir. Kişiler, sahtekarlık ve aldatılma kaygısı taşımadan elektronik ortamda tüm işlemlerini yapabilmelidir [30]. Günlük hayatın pek çok alanına giren kriptografinin en önemli boyutu ulusal bilgi güvenliğini sağlaması, siber saldırılarda bilgi hırsızlığı girişimlerini boşa çıkarmasıdır. Bu nedenlerle kriptoloji alanındaki çalışmalar önemlidir.

3.4. Kriptolojinin Tarihçesi

Geçmiş dönemlere bakıldığında kriptolojinin daha çok askeri ve diplomatik iletişimde (haberleşmede) bilgi güvenliğini sağlamak için kullanıldığı görülmüştür [22]. Günümüzde ise cep telefonu, bilgisayar sistemleri ve uydu sistemleri gibi oldukça yaygın bir kullanım alanı bulunmaktadır.

Literatürde bulunan en ilkel şifreleme algoritmaları şifreleme yönteminin sadece alıcı ile gönderen arasında bilindiği algoritmalardır. Gizli algoritma ile şifrelenmiş metnin çözümlenmesi için şifreleme algoritmasının tersine çevrilmesi yeterlidir. Bu tip algoritma ilk defa Roma İmparatorluğu döneminde Sezar tarafından generallerine mesaj göndermek için kullanılmıştır. Her bir karakteri belirli miktar kaydırma(öteleme) prensibine dayanmaktadır. Bu şifreleme türünün diğer örnekleri ise *yer değiştirme* ve a*lfabe değiştirme* şifreleridir [22, 23].

Ebced hesabı İbrani-Süryani, Grek ve Latin harf-sayı sistemidir. Daha sonra bu sistem Arapça için de kullanılmıştır. Ebced hesabında, alfabenin her

harfine bir sayı değeri verilir. Bir sözcüğü oluşturan harflerin toplam sayı değeri, anlatılmak istenen bir olayın tarihine denk getirilir. Böylece ebced hesabıyla belirli bir tarihi anlatan sözcüklere veya satırlara bakıldığında herhangi bir rakam görülmez, ancak kelimelerin sayısal değerleri hesaplandığında bir sonuca ulaşılabilir. Bu yöntem şifreleme sistemleri için de esin kaynağı olmuştur [23].

M.Ö. 400 yıllarında Spartalılar tarafından geliştirilmiş olan *scytale* sistemi geçmişteki kripto sistemlere örnek verilebilir. Scytale ilk kriptografik cihazdır [22]. Bu sistemde bir mesajı şifrelemek için uzun bir parşömen yada papirüs, silindirik bir sopa etrafına sarılmıştır. Gizlenecek mesajın kelimeleri dikey ve her bir şerit turunda bir harf gelecek şekilde yazılmıştır. Daha sonra şerit açılmış ve böylece anlamsız harflerin oluşturduğu metin ortaya çıkmıştır. Mesajın çözülebilmesi için şifreleme işleminde kullanılan silindirle *aynı çapa* sahip silindir kullanılması gereklidir. Silindirin çap değeri kripto anahtarına denk gelir [23].

M.Ö. 480 yılında Yunanlılar ve Persler arasındaki savaşta *steganografi* adı verilen bir teknik kullanılmıştır. Steganografi eski Yunanca'da *gizlenmiş yazı* anlamına gelmektedir. İran'da bulunan bir Yunanlı, Yunanlılara karşı düzenlenmesi planlanan Pers istilasını kölesinin saçını kazıtarak, kafa derisinin üzerine dövme ile yazmış ve saçları uzadıktan sonra kölesini Atina'ya yollamıştır. İranlılar tarafından yakalanmadan Atina'ya giden köle, saçını tekrar kazıtıp mesajı iletmiş, bu sayede Yunanlılar Pers istilasına hazırlanmış ve savaşı kazanmıştır [3, 22, 23].

Tarihsel süreç incelendiğinde mesajların gizlenmesinin yanı sıra şifrelenmiş mesajların çözümlenmesi ve açığa çıkarılmasının da oldukça önemli olduğu görülmüştür. Şifreli mesajın çözümü için gerekli anahtara sahip olmadan, şifrelenmiş bir mesajı deşifreleyen bilime kriptoanaliz denilmektedir. İlkel şifrelerin kırılma çabalarıyla ortaya çıkmıştır. Basit şifre sistemlerini olası tüm şifre alfabelerini denemeksizin daha kolay bir yoldan çözmek için matematik, istatistik ve dilbilim alanlarında yeterli bilgiye sahip olmak gereklidir [3, 22].

İlk kriptoanaliz çalışmaları 9. yüzyılda yaşamış Arap filozofu Al-Kindi'nin yazdığı *Kriptografik Mesajların Deşifresi* (Risâle fi'stihrâci'l-mu'ammâ) isimli yazı ile ortaya çıkmıştır ve Al-Kindi bu yazısında frekans analizi kavramını ortaya atmıştır. Bu yazı İstanbul'da, Süleymaniye Osmanlı Arşivi'nde bulunmaktadır. Al-Kindi'nin kriptoanaliz tekniğine göre yazıldığı dil bilinen şifreli bir mesajı çözmek için, aynı dilde yazılmış yeterince uzun bir metin bulup her bir harfin kullanım sıklığını hesaplamak gereklidir. Metinde en sık kullanılan harf, şifreli mesajda en sık kullanılan harfe denk gelmektedir. Frekans analizinin keşfi tek alfabeli şifre sistemlerini güvensiz hale getirmiştir. Bu nedenle şifreleme tekniğinde birden fazla alfabe kullanılmıştır ve böylece çok alfabeli şifre sistemleri ortaya çıkmıştır [22, 23].

1404-1472 yılları arasında yaşamış olan Leon Alberti, 1466-1467 yılları arasında ilk kez çoklu alfabe kullanarak kriptolama yapmıştır. Leon Alberti, Sezar şifreleme yöntemine benzer olarak harf kaydırma tekniğini uygulamıştır. Sezar şifrelemeden farklı yönü kaydırma miktarının sabit olmaması ve bu miktarın kullanıcının kararına göre belirlenmesidir. Harflerin kaydırılmasında *Alberti Diski* kullanılmıştır. İç çemberi sabit, dış çemberi hareketli bu disk yardımıyla, her harfin istenilen miktarda ötelenmiş hali görülebilir [3, 22].

1553 yılında Giovan Batista Belaso adlı bir İtalyan uzun yıllar kırılamayan çok alfabeli şifreyi geliştirmiştir. Bu şifre,1586 yılında Fransız bir diplomat tarafından biraz daha geliştirilerek *Vigenére Şifresi* adını alan şifreleme tekniğinin orjinal halidir. Bu şifrede açık metindeki her bir harf ayrı bir şifre alfabeyle şifrelenir. Hangi şifre alfabenin seçileceğine anahtar sözcüğe bakıp karar verilir. Böylece açık metindeki aynı sözcükler için farklı şifre metinler oluşur. Bu durum frekans analizinin basit şifrelerdeki gibi tek başına uygulanmasına engel olur. Uzun yıllar güvenilirliğini koruyan bu şifre 1854-1863 yılları arasında İngiliz matematikçi Charles Babbage ve Avusturya ordusunda görevli kriptograf Friedrich Kasiski tarafından kırılmıştır [3, 22, 23].

1790 yılında Thomas Jefferson, *Jefferson Diski* sistemini geliştirmiştir. Jefferson Diski, İngiliz alfabesine göre düzenlenmiştir. 26 diskten oluşmaktadır. Alfabedeki tüm harfler her diskin üzerine rastgele biçimde yerleştirilmiştir. Aynı özellikleri taşıyan Jefferson Diski, hem alıcı hem de göndericide bulunmak zorundadır. Şifre metni alan alıcı elindeki Jefferson Diski ile şifre metni oluşturup, geri kalan sıralardaki anlamlı metni çıkarmış olur [22, 23].

Charles Wheatstone ve Baron Lyon Playfair'in 5x5'lik bir matris kullanarak geliştirdiği *Wheatstone-Playfair şifresi* 20.yüzyılın başlarına kadar güvenilirliğini korumuştur. Bu şifreleme sistemi İngiliz alfabesi için tasarlanmıştır, *I* ve *J* harfleri bir arada düşünülerek 5x5'lik matrisin her hücresine bir harf gelecek şekilde tüm alfabe matrise sığdırılmıştır. Playfair şifresi 20. yüzyılın başlarına kadar güvenilirliğini korusa da daha sonraları harf ikililerinin frekans dağılımı kullanılarak kriptoanalizi elde edilmiştir [22, 23].

Hollandalı dilbilimci ve kriptograf Auguste Kerckhoffs 1883 yılında yayınladığı *La Cryptographie Militarie* isimli makalesinde bir şifreleme sisteminin sahip olması gereken özelliklerden bahsetmiştir. Bu özelliklere göre sistem güvenliği tam olmalıdır. Şifre sistemlerinin güvenli ve pratik olması en önemli özelliklerden biridir. Kerckhoffs'un şifreleme sistemlerinin güvenliği ile ilgili yayınladığı makalede bahsettiği özellikler *Kerckhoff Prensipleri* olarak bilinmektedir. Bu prensipler 1883'ten sonra tasarlanan şifre sistemleri için önemli bir yol gösterici olmuştur. Kerckhoff Prensipleri aşağıda görülmektedir [22, 23]:

- Sistem pratik ve matematiksel bir gerçekliğe dayanmalıdır.
- Sistem gizliliğe dayanmamalıdır. Yani sistem hakkındaki her şey herkes tarafından bilinmelidir.
- Sistemde kullanılan anahtarlar taraflar arasında kolayca, üçüncü kişinin değiştirmesine izin verilmeden değiştirilebilmelidir.

- Sistemin kullanılabilmesi için fazla sayıda insana ihtiyaç duyulmamalıdır.
- Sistemin uygulaması ve anlaşılması kolay olmalı ve şifreleme sisteminin güvenliği, şifreleme algoritmasını gizli tutmaya dayanmamalıdır. Güvenlik; yalnızca anahtarı gizli tutmaya dayanmalıdır [22, 23].

19. yüzyılın sonlarına doğru teknoloji geliştikçe şifrelemenin önemi daha da artmıştır. Şifrelerin kullanım alanlarının yaygınlaşması bu önemin artmasında etkili bir paya sahiptir. İtalyan fizikçi Markoni'nin güvenli iletişime ihtiyaç duyulan telsizi icat etmesi ile yeni sorunlar ortaya çıkmıştır. Savaş alanında haberleşmeyi çok kolaylaştıracak olan bu alet, bilgilerin güvenli bir şekilde iletilmesi zorunluluğunu bir kez daha ispatlamıştır. Mesaj hedeflenen alıcının yanı sıra düşmana da kolaylıkla ulaşabilecek hale gelmiştir. Bu sebeple telsizle iletişim mesajın güvenli bir şekilde şifrelenmesini gerektirmiştir [22, 23].

I.Dünya Savaşı döneminde telsiz iletişimi güvenliği için birçok şifreleme yöntemi geliştirilmiştir. Dönemin ünlü şifreleme yöntemlerinden biri Alman *ADFGVX şifresi*dir. 1918 tarihli büyük Alman saldırısından hemen önce tasarlanmış olan bu yöntem, yer değiştirme ve ters çevirme işlemlerinin karışımından oluşmaktadır. Bu şifreleme yöntemi Fransızlar tarafından kırılmış ve Almanlar savaşı kaybetmiştir [22, 23].

Birinci Dünya Savaşı döneminde en etkili kriptoanaliz, İngiltere'nin 1917'de Almanya Dışişleri Bakanı Arthur Zimmermann'ın Meksika Başkanı'na çekmiş olduğu telgrafı deşifre etmesidir. İngiltere üzerinden şifrelenmiş olarak iletilen telgrafın kriptoanalizciler tarafından deşifre edilmesi sonucu açığa çıkarılan bilgiler Amerika'nın savaşa katılmasında etkili olmuştur [3, 22, 23].

Matematiğin sistematik olarak kriptografide kullanılması 1917 yılında Amerika'da başlamıştır. *Vernam şifresi* olarak bilinen ve Vigenére şifreleme sistemini temel alan yeni bir şifreleme tekniği geliştirilmiştir [22, 23].

Vernam şifreleme yönteminde her şifrelemeden sonra anahtarın değiştirilmesinin sistemi çok daha güvenli hale getireceği düşüncesi *tek seferlik şifre(one time pad)* yöntemini ortaya çıkarmıştır. Tek seferlik şifre yöntemi günümüzde de kullanılmaktadır [22, 23].

Tek seferlik şifre yönteminin *anahtar iletim sorunu* vardır. Bu bir dezavantajdır. Mesaj uzunluğunda bir anahtarın güvenli bir kanaldan iletilmesi gereklidir. Her seferinde anahtarın değişeceği düşünülürse pratik olmayan masraflı bir sistem olduğu görülür. Buna rağmen Amerika ile Rusya devlet başkanları arasındaki kırmızı hatta tek seferlik şifreyle iletişim yapıldığı bilinmektedir [3, 22, 23].

1918'de Birinci Dünya Savaşı'nda Almanya tarafından *Enigma* adlı bir kripto cihaz kullanılmıştır. Polonyalı kriptoanalistler 1938 yılına kadar yapmış oldukları çalışmalarla Enigma ile şifrelenen mesajları çözmeyi başarmıştır. Bunun üzerine 1939 yılında Enigma'nın yapısı daha da karmaşıklaştırılmıştır [3, 22, 23, 31].

1929 yılında Leste Hill tarafından bulunan *Hill şifresi* çok alfabeli şifreleme sistemlerinin bir başka örneğidir ve çok alfabeli şifreleri daha pratik bir hale getirmesi bakımından oldukça önemlidir. Claude Shannon'un 1949 yılında öne sürdüğü güvenli bir şifreleme sisteminin karıştırma işlemini iyi yapması gerekliliği bu sistem tarafından sağlanmıştır. Lester Hill ve ortağı 6x6'lık bloklarla Hill şifresinin uygulanabileceği bir makine geliştirmiştir, fakat bu makineye çok fazla talep olmamıştır [22, 23].

Enigma şifrelerini çözmek için 1943 yılında *Colossus makinesi* geliştirilmiştir. İlk elektronik bilgisayar olarak kabul edilen Colossus'un geliştirilmesi ile bilgisayarın babası denilebilecek *Eniac* ortaya çıkmıştır. Enigma'yı çözmek için kullanılan Eniac, Avrupa'da Hitler'in gücünü kırmıştır [22, 23, 31].

Modern kriptografinin başlangıcı harflerin elektronik ortamda bitlerle ifade edilmesi ile olmuştur. Şifreleme işlemleri bitlerle yapılmaya başlanmıştır [22].

1970 yılında IBM laboratuarlarında *Demon* adıyla 64 bit anahtar kullanan yeni bir şifreleme sistemi geliştirilmiştir. Daha sonra adı *Lucifer* olarak değiştirilmiştir. Bu sistem 1975 yılında Amerika'da Birleşik Bilgi İşleme Standardı olarak seçilmiştir [22, 23].

Amerika Milli Standartlar Bürosu elektronik haberleşmenin yaygınlaşması ile herkesin kullanabileceği bir kriptografik algoritmaya ihtiyaç duyulduğunu açıklamıştır. Lucifer algoritması üzerinde yapılan bazı değişikliklerle bu ihtiyaç giderilmiş ve *Veri Şifreleme Standardı (Data Encrypt System(DES))* ortaya çıkmıştır [22, 23, 31].

DES bir algoritma olarak harflerin bit karşılıklarını kullanmaktadır. Genel bir ifadeyle DES, ikilik tabanda olan düz metni 64 bitlik parçalar, yani bloklar halinde, 56 bitlik anahtar kullanarak şifreler. Algoritma gücünü şifreleme anahtarının gizli ve rastgele seçilmiş olmasından almaktadır. DES'in en önemli özellikleri yayılma ve karıştırma özellikleridir. Yayılma özelliği, anahtarın düz metin ve şifreli metine bağlı olmasını sağlamaktadır. Bu sayede sistem karmaşıklaşmakta ve kriptoanalizi zorlaşmaktadır. Karıştırma özelliği, her anahtar için düz metin ve şifreli metinin yapıları arasında istatistiksel bağlantı olmamasını sağlamaktadır. Böylece düşmanın eline düz metin ve o düz metinin şifreli hali geçse de aralarında bir bağlantı kurulamayacağından anahtar hakkında tahminde bulunulamaz [6, 22, 23].

1976 yılında *Diffie-Hellman anahtar değişim algoritması* ile sayılar teorisinin uzun yıllardır bilinen ama kullanılmamış olan özellikleri ortaya çıkmıştır. Diffie ve Hellman, alıcı ve göndericinin gizli bir anahtar üzerinde anlaşmaları için bir araya gelmelerinin zorunlu olmadığını kanıtlamıştır. Diffie-Hellman anahtar değişim algoritması, sayılar teorisi ile şifreleme işleminin herkese açık bir şekilde yapılabileceğini göstermiştir. Böylece kriptografide yeni bir kavram olan *Açık Anahtarlı Kriptografi* literatüre girmiştir [3, 6, 22, 23].

Açık anahtar kriptografisi temel olarak herkesin birbirini tanımadan gizli bir şekilde haberleşebilmesi demektir. Bu tanım kriptografi için bir dönüm noktası

olmuştur. Açık anahtar kriptografisi geliştirilene kadar tasarlanan bütün sistemlerde kullanılan anahtarların taraflarca bilinmesi gerekirken geliştirilen bu yeni sistemle kişilerin sadece kendilerinin bildiği özel anahtarları olmuştur [3, 6, 22, 23].

Açık anahtar kriptografi düşüncesinden etkilenen Ronald Rivest, Adi Shamir ve Leonard Adleman, 1977'de *RSA*(Rivest-Shamir-Adleman) adını verdikleri yeni bir açık anahtar şifreleme sistemini bulmuştur. RSA, Diffie-Hellman anahtar değişim algoritmasının felsefesini içeren ilk şifreleme sistemidir. RSA sistem olarak matematiğin konusu olan çarpanlara ayırma problemine dayanmaktadır. RSA'da hem açık hem de gizli anahtar kullanılmaktadır [3, 6, 22, 23, 32].

Tüm bu şifreleme yöntemlerini çözmek için kriptoanaliz çalışmaları sürekli devam etmiştir. DES ve benzeri yapıyı kullanan diğer simetrik şifrelerin güvenilirliği 1990'lı yıllara kadar sürmüştür. Yeni geliştirilen lineer ve diferansiyel kriptoanaliz yöntemleri bu sistemlerin bazı anahtarlar için güvensiz olduklarını ortaya koymuştur. Kaba kuvvet atağına ek olarak düz metin ve seçilen düz metin ataklarının kullanılmaya başlanması, şifrelerin dayanıklılığını zorlamaya başlamıştır. Kaba kuvvet atağında, olası bütün anahtarlar şifre metin üzerinde denenir ve düz metine ulaşılmaya çalışılır. Bu pratik bir yöntem değildir. Bilinen düz metin atağında ise kriptoanalist aynı şifreleme anahtarı kullanılarak elde edilmiş şifrelenmiş düz metinlerin çeşitli şifre metinlerine sahiptir. Amaç istatistiksel saptamalar yaparak anahtarı bulmak ve şifrelenen başka mesajları çözme işleminde kullanmaktır. Seçilmiş düz metin atağında, kriptoanalist istediği düz metini seçerek bunun şifreli halini elde edebilir. Bu atak bilinen düz metin atağından daha güçlüdür [22, 23, 33, 34].

İkinci Dünya Savaşı'nda Alman ajanları *mikro nokta yöntemi* adı verilen bir şifreleme yöntemi kullanmışlardır. Bu yöntemde bir sayfalık metin şifrelenir ve fotoğraf teknikleriyle çapı bir milimetreye yakın bir nokta boyutuna küçültülür.

Daha sonra mikro nokta görünüşte önemsiz bir mektubun son noktası üzerine gizlenir [3, 22, 23].

1960'larda ise mor ötesi boya ile yazı yazabilen spreyler ve kalemler çıkmıştır. Bu sprey ve kalemlerle yazılan yazılar ancak mor ötesi bir ışıkta okunabilmektedir. Bu steganografide kullanılan yaygın yöntemlerden biridir [3, 22].

Görüldüğü gibi geçmişten bugüne, bilgi ve bilgi güvenliği giderek artan bir değerle insan hayatındaki önemini korumuştur. 1990'lı yılların başlarında yeni bir kavram olan *Bilgi Savaşı* (information warfare) dünya literatüründeki yerini almıştır. Dr. Andy Jones'un, *ComputerFraud& Security* adlı yayında yazmış olduğu *Information Warfare* adlı makalesinde belirttiği gibi internet ve geniş bir ağ ile örülü bu dünya için bilgi savaşı kavramı kıyamet alameti gibi gelmiştir.

1996-1997'li yıllarda silah olarak virüs, kurtçuk ve Truva atı gibi kötücül yazılımların kullanımı oldukça yaygınlaşmıştır. Böylece siber ordu olarak anılan yeni nesil askerlere ihtiyaç duyulmuştur. 1990'ların sonunda *bilgi savaşı* kavramına *bilgi işlemleri* kavramı eklenmiştir. 2003 yılında Amerika Savunma Bakanlığı, Bilgi İşlemleri Yol Haritasını hazırlamıştır. Birleşik Krallık Savunma Bakanlığı, etkili karar verebilmek ve çevik hareket edebilmek için gerekli bilgi ve istihbaratı sağlamak amacıyla Network Enabled Capability (NEC) (ağ destekli yetenek) kavramını açıklamıştır [4,35]. Tüm bu gelişmeler rakibinden ne kadar hızlı hareket edersen elindeki güçleri(bilgiyi) o kadar etkili, verimli ve yerinde kullanırsın düşüncesi ile ortaya çıkmıştır.

Şifreleme sistemlerindeki karmaşanın giderek artması günümüzde kuantum şifreleme tekniklerinin gelişmesini sağlamıştır. Kuantum şifreleme verilerin aynı anda hem 1 hem de 0 biti ile ifade edilebilmesini sağlamaktadır [3, 36].

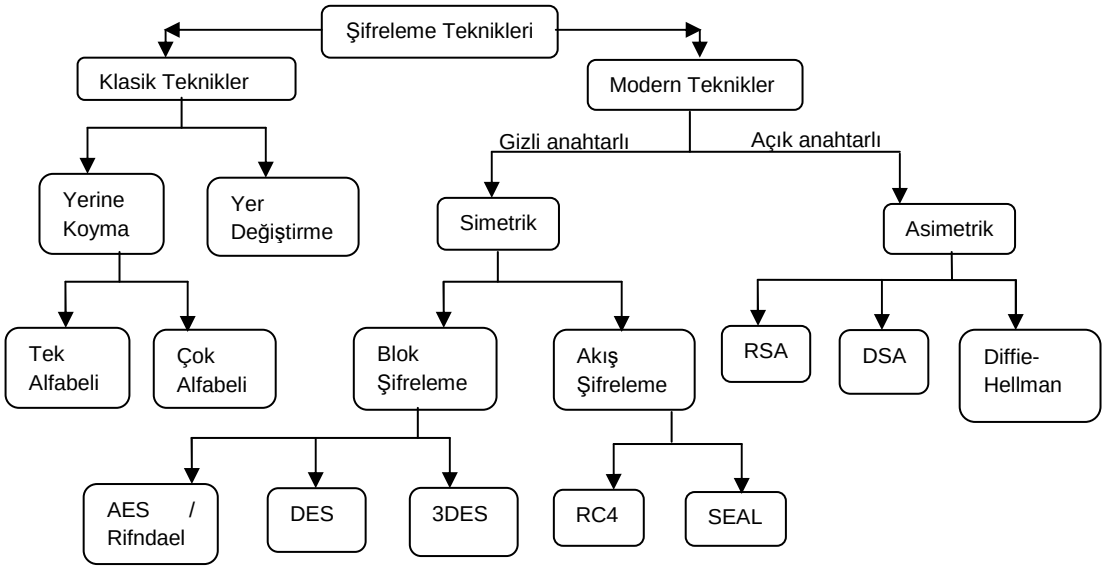
2009 yılında güvenilir yeni bir algoritma üretebilmek amacıyla Belçika Katholieke Üniversitesi'nde ilk Kriptografi olimpiyatları düzenlenmiştir [21].

Dünyada kriptografi alanında eğitim ve araştırmaya yönelik ilk ders kitapları 1987 yılında çıkmıştır. Bu alandaki bilimsel çalışmaların temeli ülkemizde 1990'lı yılların başlarında Orta Doğu Teknik Üniversitesi (ODTÜ) Matematik Bölümü'nde seçmeli derslerin verilmesi ile başlamıştır. Daha sonra ODTÜ Matematik Bölümü ve TÜBİTAK Marmara Araştırma Merkezi Temel Bilimler grubu ile ortak çalışmalar yapılmıştır. Bu çalışmalar sonucunda 1995 yılında TÜBİTAK bünyesinde bulunan Ulusal Elektronik ve Kriptoloji Araştırma Enstitüsü (UEKAE) birimi kurulmuştur. Bu kurum ülkemizde kriptoloji konusunda faaliyet gösteren ilk araştırma merkezidir. 2002 yılında ise ODTÜ Uygulamalı Matematik Enstitüsü bünyesinde Kriptografi Bölümü açılmıştır [23].

3.5. Şifreleme/ Kriptografi Algoritmalarının Sınıflandırılması

Gelişen teknoloji ile şifreleme için uygulanan yöntemler de farklılık göstermiştir. Kriptografi algoritmaları klasik ve modern olmak üzere iki ana kategoriye ayrılabilir [37, 38]:

- Klasik teknikler
- Modern teknikler



Şekil 3.4.Şifreleme algoritmalarının sınıflandırılması [39]

3.5.1. Klasik teknikler

Klasik şifreleme teknikleri daha çok kağıt-kalem kullanılarak gerçekleştirilebilecek algoritmalardır [40]. Elektronik bilgisayarlar ortaya çıkmadan önce kullanılmışlardır. ENIGMA, II. Dünya Savaşı esnasında kullanılan en önemli klasik tekniklerden biridir [41].

Klasik kriptografi sadece askeri işlemlerde ve ileri akademik kurumlarda kullanılmıştır [38]. Klasik teknikler, algoritması gizli olan şifreleme tekniklerini içermektedir. Algoritmanın gizli tutulması gerektiği için güvenirliği düşük algoritmalardır [19].

Sezar, playfair, hill, vigenére, ENİGMA sistemleri klasik tekniklerden bazılarıdır [37,38].

Sezar şifreleme

Alfabedeki harflerin sıra numaraları belirli bir miktar kaydırılarak şifreli karşılıkları elde edilir. Tek alfabeli şifreleme tekniğidir. Bu teknik öteleme şifrelemesi olarak da bilinir. Eğer harflerin ötelenme miktarı 3 olursa Sezar şifreleme tekniği olur [40,42]. Matematiksel olarak ifade edilmek istenirse; "$E_k(m) = (m + k) \bmod 26$" şeklindedir. *m*, açık metindeki sıradaki harfin sıra numarasıdır. *k*, harfleri öteleme miktarıdır. *mod26*'daki 26 İngiliz Alfabesindeki toplam karakter sayısıdır. Eğer öteleme şifresi Türkçe Alfabe için kullanılacaksa bu değer 29 olmalıdır [40]. Örnek 3.1'de Türkçe alfabe için hazırlanmış küçük bir uygulama verilmiştir.

Örnek 3.1.

a b c ç d e f g ğ h ı i j k l m n o ö p r s ş t u ü v y z
0 1 2 3 4 ……26 27 28

Düz metin : a r ı → 0, 20, 10

k= 3 için;

$E_k(0) = (0+ 3) \bmod 29 = 3$
$E_k(20) = (20 + 3) \bmod 29 = 23$
$E_k(10) = (10 + 3) \bmod 29 = 13$

Şifreli metin : 3, 23, 13 → ç t k

olarak elde edilir.

Alberti diski

1404-1472 yılları arasında yaşamış olan Leon Alberti, 1466-1467 yılları arasında ilk kez çoklu alfabe kullanarak kriptolama yapmıştır. Bu yöntemde, Sezar şifreleme yöntemine benzer olarak harf kaydırma tekniği uygulanmıştır. Sezar şifrelemeden farklı olarak kaydırma miktarı sabit olmayıp, kullanıcının kararına göre belirlenmektedir. Kriptolanacak metinde her harfin kriptolu karşılığı, Alberti Diski yardımıyla bulunmaktadır. İçteki çemberi sabit, dıştaki çemberi onun etrafında dönebilen bu disk yardımıyla, her harfin istenilen miktarda ötelenmiş hali kolaylıkla görülebilmektedir [22].

Vigenére şifreleme

1553 yılında Giovan Batista Belaso adlı bir İtalyan uzun yıllar kırılamayan çok alfabeli şifreyi geliştirmiştir. 1586'da bu şifrenin biraz daha gelişmiş şeklini tasarlayan ve sunan Fransız diplomat Bliase de Vigenére'den dolayı Vigenére şifresi adını almıştır. Bu teknikte düz metindeki her bir harf ayrı bir şifre alfabeyle şifrelenir. Şifre alfabenin seçiminde anahtar kelimeye göre karar verilir. Anahtar kelimenin farklı seçilmesi, düz metindeki aynı kelimeler için farklı şifreli metinler oluşmasını sağlar. Bu durum frekans analizinin tek alfabeli şifreleme tekniklerindeki gibi iyi sonuç vermesini engeller. Uzun yıllar güvenilirliğini koruyan bu şifre 1854-1863 yılları arasında İngiliz matematikçi Charles Babbage ve Avusturya ordusunda görevli kriptograf Friedrich Kasiski tarafından kırılmıştır [3,22]. Vigenére şifreleme için alfabedeki harflerin yer

aldığı bir tablo kullanılır. Çizelge 3.1'de Türkçe alfabe için hazırlanmış Vigenére tablosu görülmektedir.

Orijinal Vigenére şifreleme tekniğinde sadece *Vigenere Square* olarak adlandırılan tablo kullanılır. Bu tablo şifreleme ve deşifreleme işlemlerinde sabit olarak (tablodaki harflerin yerleri değişmez) kullanılır [43]. Şifreleme işleminde kullanılmak üzere seçilen anahtar kelime düz metinin uzunluğu kadar kendisini tekrar eder. Anahtar kelimedeki harfler Vigenére tablosundaki en sol sütundan, düz metindeki harfler ise en üstteki satırdan seçilir. Matris üzerinde çizilen dik doğruların kesiştiği yerdeki harf şifreli metnin harfini temsil eder. Harfler eşleştirilir ve yer değiştirilir. Bu eşleştirme ve yer değiştirme işlemi her harf için uygulanır [44]. Örnek 3.2'de Vigenére şifreleme tekniğinin uygulaması görülmektedir:

Örnek 3.2. Düz metin *bilgisayar eğitimi* ve anahtar kelime *kripto* olarak belirlenmiştir. Şifreleme işleminde Çizelge 3.1 kullanılmıştır.

Anahtar: k r i p t o k r i p t o k r i p t
Düz metin: b i l g i s a y a r e ğ i t i m i
Şifreli metin: l c ü v e h k ö i ı z ü u l ş e e

Vigenére şifreleme tekniğinde anahtar kelimedeki harflerin sayısal karşılıkları, düz metindeki harflerin kaydırılma miktarını verir [45]. Türkçe alfabe (29 harf var) için uygulanmış Örnek 3.2'de düz metindeki *B* harfinin sayısal karşılığı 1, anahtar kelimedeki *K* harfinin sayısal karşılığı 13'tür. *B* harfi 13 harf kaydırıldığında sayısal karşılığı 14 olan *L* harfi şifreli karşılığı olur. Matematiksel olarak ifade edilirse [45];

Anahtar kelime= (a_1 , a_2, ……a_i) ve Düz metin = (d_1, d_2, ….., d_i) olmak üzere $f_i(d) = (d + a_i) \bmod 29$ olur.

Çizelge 3.1. Tükçe alfabe için oluşturulmuş Vigenére Tablosu

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

Vernam şifreleme

1917 yılında Gilbert Vernam tarafından geliştirilmiştir. Telgraf hatlarındaki haberleşmenin şifrelenmesinde kullanılmıştır. Vernam şifreleme tekniği Vigenére şifreleme tekniğine benzer. Farkı ikili sayı sistemini (0 ve 1) kullanması ve düz metinin XOR (*exclusive-or*) işleminden geçmesidir. Rastgele belirlenen kendisini tekrarlamayan anahtar dizisi ile şifreleme işlemi yapılır. Şifreleme işleminde ikili sayı sisteminde yazılmış *Baudot* adı verilen 5 bitlik karakter dizileri kullanılır. Her harf için Baudot kodu farklıdır. Rastgele belirlenen anahtar dizinin her bir karakterine karşılık gelen Baudot koduna düz metinin her bir karakterinin Baudot kodu eklenerek(XOR) yeni şifreli karakter dizisi elde edilir. XOR işleminin kuralına göre, her iki bit aynı ise sonuç olarak 0 biti, her iki bit farklı ise sonuç olarak 1 biti üretilir. Düz metin $m_1m_2\ldots$, anahtar dizi $k_1k_2\ldots$ ve şifreli metin $c_1c_2\ldots$ ile temsil edildiğinde, şifreli metin $c_i=(m_i + k_i)mod2$ ile elde edilir. [45, 46]. Örnek 3.3'te Türkçe alfabe için yapılmış bir örnek görülmektedir:

Örnek 3.3.
Düz metin karakteri = A, Baudot kodu 11000
Anahtar dizi karakteri = D, Baudot kodu 10010
Şifreli metin karakteri = I , Baudot kodu 01010

A ve D harflerinin Baudot kodları ikilik sisteme göre toplanır ve şifreli karakterin Baudot kodu elde edilir [45]. Aynı şekilde m= 010 , k= 110 alınırsa c= m XOR k = 100 olarak bulunur [47].

Hill şifreleme

Hill şifrelemede şifreleme anahtarı olarak bir katsayılar matrisi (K) kullanılır. Katsayılar matrisinin elamanları ile düz metindeki karakterlerin sayısal karşılıklarından elde edilen matrisin elamanları çarpılır. Düz metin uzunluğu 2 birim olan karakter gruplarına ayrılır. Düz metinin uzunluğu d = 2, düz metin m_1m_2 , şifreli metin c_1c_2 olmak üzere katsayılar matrisisin eleman sayısı 4'tür

ve 2X2'lik bir matristen oluşur [38,45]. Alfabedeki harf sayısı *n* kabul edildiğinde Hill şifreleme tekniği, matematiksel olarak ifade edilmek istenirse;

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \end{pmatrix} \bmod n\,, \quad \text{şeklinde olur [45].}$$

$c_1 = (k_{11}m_1 + k_{12}m_2) \bmod n$ ve $c_2 = (k_{21}m_1 + k_{22}m_2) \bmod n$, olarak hesaplanır [45]. Örnek 3.4'te Türkçe Alfabe kullanılarak uygulanmış Hill şifreleme örneği görülmektedir [40]:

Örmek 3.4. Düz metin = BİLGİN → Bİ (1,11), LG (14,7), İN (11,16)

$K = \begin{pmatrix} 5 & 12 \\ 9 & 3 \end{pmatrix}$ kabul edilmiştir.

Bİ için; $c_1 = (5*1 + 12*11) \bmod 29 = 21$, $c_2 = (9*1 + 3*11) \bmod 29 = 13$

$c_1c_2 = (21, 13) \rightarrow$SK

LG için; $c_1 = (5*14 + 12*7) \bmod 29 = 9$, $c_2 = (9*14 + 3*7) \bmod 29 = 2$

$c_1c_2 = (9, 2) \rightarrow$HC

İN için; $c_1 = (5*11 + 12*16) \bmod 29 = 15$, $c_2 = (9*11 + 3*16) \bmod 29 = 2$

$c_1c_2 = (15, 2) \rightarrow$MC

Şifreli metin = SKHCMC olarak elde edilir.

Playfair şifreleme

Bu teknik adını İngiliz bilim adamı Lyon Playfair'den almıştır. Yerine koyma yönteminin bir türüdür. Şifreleme tekniğini 1854'te Charles Wheatstone bulmuştur, Lyon Playfair geliştirmiştir. Bu teknik Birinci Dünya Savaşı'nda İngilizler tarafından kullanılmıştır. 5X5'lik matris düzeni ile şifreleme işlemini gerçekleştirir. Şekil 4.1'de playfair için matris düzeni gösterilmiştir[45]:

H	A	R	P	S
I	C	O	D	B
E	F	G	K	L
M	N	O	T	U

Şekil 3.5. Playfair için anahtar diziyi gösteren matris.

Düz metindeki her harf ikilisi (m_1m_2), takip eden kurallarla şifrelenir [45]:

1. m_1 ve m_2 aynı satırdaysa, m_1 ve m_2'nin bulundukları sütunun sağında olduğu kabul edilen ilk sütunda sırasıyla m_1 ve m_2'nin sağındaki karakterler c_1 ve c_2'yi (şifreli haller) verir [45].
2. m_1 ve m_2 aynı sütundaysa, m_1 ve m_2'nin bulundukları satırın altındaki ilk satırda sırasıyla m_1 ve m_2'nin altındaki karakterler c_1 ve c_2'yi (şifreli haller) verir [45].
3. m_1 ve m_2 farklı satır ve sütundaysa, m_1 ve m_2'nin bulundukları yer matrisin köşesi kabul edilir. c_1, m_1'in olduğu satırın diğer köşesindeki karakter olur. c_2, m_2'nin olduğu satırın diğer köşesindeki karakter olur [45].
4. Eğer $m_1 = m_2$ olursa, düz metinde kullanılmayan bir harf m_1 ve m_2'nin arasına yerleştirilir. Böylece aynı iki harfin yan yana gelmesi önlenmiş olur[45].
5. Eğer düz metin ikili harf gruplarına ayrıldığında tekli harf kalırsa düz metinde kullanılmamış bir harf yanına eklenir [45].

Örnek 3.5'te playfair şifreleme tekniğine ait bir uygulama görülmektedir [45]:

Örnek 3.5. RENAISSANCE kelimesinin şifreli halini elde etmek için Şekil 3.5'te görülen matrise göre işlem adımları uygulanmıştır. Her işlem adımı için tek bir örnek gösterilmiştir. Teknik gereği öncelikle kelime ikişerli harf gruplarına ayrılır.

RE NA IS SA NC E*X*

Tekniğe ait 5. Maddeye göre en sonda tek kalan E harfinin yanına kelimede kullanılmayan herhangi harf (örneğin X harfi) koyulur.

RE ikilisi için, R harfi matrisin 1. Satır 3.sütun elemanı iken E harfinin 3.satır 1.sütun elemanı olduğu görülmektedir. Bu durumda 3.madde kullanılır. R ve E köşe eleman olursa 3X3 boyutunda matris oluşur. R ile aynı satırdaki diğer köşe elemanı H, E ile aynı satırdaki diğer köşe elemanı G şifre harf olarak seçilir.

H A R
I C O
E F **G**

NA ikilisi için, N ve A harfleri aynı sütundadır. Bu durumda tekniğin 2.maddesi uygulanır. N, 4.satırdadır. Bir alt satırdaki yani 5.satırdaki W harfi şifre harftir. A, 1.satırdadır. Bir alt satırdaki yani 2.satırdaki C şifre harftir.

SA ikilisi için, S ve A harfleri aynı satırdadır. Bu durumda tekniğin 1.maddesi uygulanır. S harfi 5.sütundadır ve sağındaki sütun 1.sütundur. S'nin sağındaki harf H seçilir. A harfi 2.sütundadır ve sağındaki sütun 3.sütundur. A'nın sağındaki harf R seçilir.

Örnek 3.5'te görülen kelimenin diğer harf ikilileri için de aynı işlemler uygulanır ve sonuçta HG WC BH HR WF GV şifreli metni elde edilir.

Enigma

Enigma İkinci Dünya Savaşı döneminde Almanlar tarafından kullanılmıştır[3]. Düz metinleri mekanik ortamda şifrelemek için geliştirilen bir makinedir. Enigmada 26 harften oluşan bir klavye (tuş yatağı) bulunmaktadır. Tuş yatağının arkasına tuşları aydınlatan lambalar yerleştirilmiştir. Şifreleme işlemini hareketli olan üç adet şifreleme çarkı (rotor) ve hareketsiz olan bir adet çark gerçekleştirir. Sabit olan çark yansıtıcı çark olarak adlandırılır. Hareketli çarkları ters düz eden bir mil(dingil) kullanılır. Hareketli çarkların

kenarlarında alfabenin harfleri vardır. En üstteki harfler küçük bir kapaktan görülebilir durumdadır. Hareketli olan her çarkın bir yüzüne 26 ortak merkezli sabit bağlantı diğer yüzüne 26 yaylı bağlantı yerleştirilir. Sabit ve yaylı bağlantıların birbiri ile yalıtımlı kablolar aracılığı sayesinde düzensiz olarak iletişim kurması sağlanır. Sabit çark üzerinde sadece yaylı bağlantılar vardır ve bu bağlantılar düzensiz olarak çiftler halinde birbirleri ile bağlanmıştır. Enigma şifreleme makinesinin esas gizemi dört çark arasında kablolar ile sağlanan iletişimdedir. [48,49] Seçilebilecek 26 harf için 26 kablo yuvası bulunur ve her bir kablo 2 yuvaya takılır. Seçilebilecek kablo sayısı p kabul edilirse, $0 \leq p \leq 13$ olur. Kabloların yerleştirileceği yuvaların seçimi için;

$\binom{26}{2p}$ farklı kombinasyon vardır. [50].

3.5.2. Modern teknikler

Bilgisayar sistemlerinin gelişmesi ile şifreleme sistemleri bilgisayar programları ile gerçekleştirilmeye başlanmıştır. Modern teknikler kağıt-kalem sistemlerinden daha karmaşık yapıya sahiptir [40]. Harflerin elektronik ortamda bitlerle ifade edilmesi sonucu şifreleme işlemleri bitlerle yapılmaya başlanmıştır. Genel olarak simetrik algoritmalar, asimetrik algoritmalar ve hash algoritmalar olmak üzere üç sınıfta toplanabilir [51, 52].

Simetrik (gizli anahtarlı) şifreleme teknikleri

Tek bir anahtar kullanılır. Aynı anahtarla hem şifreleme işlemi hem de şifreyi çözme işlemi yapılır. Bu nedenle simetrik şifreleme sistemlerinde şifreleme işleminde kullanılan anahtar gizlidir. Anahtar bilgisi sadece metini gönderen ve alan kişilerde olmalıdır. Anahtarın başka kişilerce bilinmesi iletilecek metinin güvenliğini tehlikeye atar. Simetrik şifreleme algoritmalarına XOR(özel veya), DES, 3DES, IDEA, RC2, RC5, Blowfish, FEAL, SAFER, Skipjack, Lucifer, ASEKAL-21 gibi algoritmalar örnek verilebilir [19, 53].

Simetrik (gizli anahtarlı) şifreleme blok şifreleme ve akan şifreleme olmak üzere ikiye ayılır [19]:

- *Blok şifreleme*: Şifreleme işlemi bloklar halinde yapılır. Blok şifreleme algoritmaları akan şifreleme algoritmalarından daha güvenilirdir [19]. Blok şifreleme algoritmaları yayılma ve karıştırma özelliklerine dayanır. Karıştırma işlemi şifreli ve açık metin arasındaki ilişkiyi gizler. Yayılma işlemi ise açık metin üzerinde uygulanan işlemlerinin şifreli metinden anlaşılmamasını sağlar. Blok şifrelemeyi kullanabilmek için NIST tarafından dört farklı model tanımlanmıştır [38]:
    - *Elektronik kod kitabı(ECB-Electronic Code Book) modeli:* Açık metin bloklara ayrılır ve her blok aynı anahtarla şifrelenir. Açık metinde birden fazla aynı b-bitlik blok olursa üretilen şifreli metin blokları aynı olur. Bu nedenle uzun metinlerde bu modeli kullanmak mesajın güvenliği açısından tehlikelidir [38].
    - *Kapalı metin zincirleme modeli (CBC-Cipher Block Chaining Mode):* ECB modelinin açıklarını kapatmak için geliştirilmiştir. EKK'dan farkı, açık metinde birbirini tekrar eden aynı bloklar için farklı şifreli bloklar üretmesidir. Bu farklılığı sağlamak için açık metnin ilk bloğuna XOR işlemi uygulanır [38].
    - *Şifreyi geri besleme modeli (CFB-Cipher Feedback Mode):* Bu modelde kaydırmalı yazmaç kullanılır. XOR işlemi ve sola kaydırma işlemleri açık metnin tüm birimleri şifrelenene kadar devam eder ve kaydırmalı yazmaca geri beslenen şifreli metindir [38].
    - *Çıktıyı geri besleme modeli (OFB-Output Feedback Mode):* CFB modeline benzer. OFB modelinde geri beslenen şifreli metin değil, şifreleme fonksiyonunun çıkışıdır [38].

DES algoritması

DES algoritması blok şifreleme tekniği kullanılarak geliştirilmiştir. İlk modern şifreleme algoritması olarak kabul edilmektedir.

Kriptografi ilk geliştirildiği zamanlarda askeri işlemlerde ve diplomatik ilişkilerde kullanılmıştır. Ancak 1970'li yıllarda kriptografinin ticari sektörde de bir ihtiyaç olduğu görülmüştür. Kurumsal verilerin, endüstriyel casusların dinleme olasılığı olmadan uzak siteler arasında iletilebilmesi gerekmiştir. Kişisel verileri içeren veritabanlarının herhangi casusluk veya değişikliklere karşı korunması ihtiyacı doğmuştur. Bir ATM(automatic teller machine) ile merkez bilgisayar arasındaki iletişim bu duruma örnek verilebilir. Banka müşterileri IBM'den ATM verilerinin şifrelenmesi için bir sistem geliştirmesini istemiştir. Bu problem DES algoritmasının gelişmesinde başlangıç noktası olmuştur [54].

Şimdiki adı NIST (National Instıtute of Standards and Technology) olan Ulusal Standart ve Teknoloji Enstitüsü, standart bir kriptografi algoritması üretilmesi için bir duyuru yapmıştır. IBM (International Business Management) geliştirdiği Lucifer algoritmasını öne sürmüştür. ABD (Amerika Birleşik Devletleri) NSA (National Security Agency, ulusal güvenlik ajansı) önerilen algoritmayı yani Lucifer'i geliştirerek 1977 yılında DES (Data Encrypt Standart, veri işleme standardı) algoritmasını Bilgi İşleme Standardı olarak geliştirmiştir. DES algoritması veri şifrelemede yaygın bir kullanıma sahip olmuştur [55, 56].

DES şifreleme ve deşifreleme işlemlerinde aynı anahtarı kullanır, simetrik anahtarlı bir algoritmadır. DES algoritması şifreleme işlemini 64bitlik bloklar şeklinde yapar. Her adımda 16 döngü vardır. Her döngüde, metin sağ ve sol kısım olmak üzere 32bitlik iki bloğa ayrılır.

Veriler yerine koyma, permütasyon ve XOR işlemlerinden geçer [55,56] .

Permütasyon, yerine koyma ve XOR(mod 2) işlemlerine dayanan DES algoritmasında yerine koyma işlemleri S-Kutusu adı verilen tablolar aracılığı ile yapılır. 16. Döngüden sonra sağ ve sol bloklar tekrar birleştirilir ve şifreli metin elde edilir [57].

- *Akan şifreleme*: Şifreleme işleminde şifreleme algoritması belli bir anda sadece bir bit ya da bir bayt şifreleme yapabilir. RC4 algoritması örnek verilebilir [19].
    - *Tek kullanımlık sistem (OTP-One Time Pad):* Şifreleme işlemi için açık metinle aynı uzunlukta tam rastgele bir anahtar dizisi seçilir. Tam rastgele dizide her bir birbirinden bağımsızdır. Şifreleme ve deşifreleme işlemleri için şifreli metine ve anahtara XOR işlemi uygulanır [38].
    - *Dizi üreticiler:* Diziler dizi üreticiler tarafından kısa bir anahtar kullanılarak üretilir. Her bitin kendinden önceki bitlere bağlı olduğu sözde rastgele dizilerden faydalanır [38].

Asimetrik (açık anahtarlı) şifreleme teknikleri

Asimetrik şifreleme tekniklerinde iki farklı anahtar kullanılır. Birinci anahtar şifreleme işleminde kullanılır ve herkese açıktır. İkinci anahtar ise şifreyi çözme işleminde kullanılır ve sadece şifreli metini açması gereken kişiye özeldir [19, 33]. Asimetrik şifreleme algoritmalarına RSA, DSA, Diffie-Hellman, ElGamal algoritmaları örnek verilebilir [19].

Hash şifreleme teknikleri

Özetleme işlemi yaparlar. Mesajdan sabit uzunlukta bir dizi üretilir ve buna mesaj özeti denir. Üretilen özet metnin karakterini taşır. Şifrelenecek metindeki tek bir karakter değişikliği üretilecek özette büyük değişikliklere yol açar. Hash algoritmalarda özetleme tek yönlü olduğu için özetten asıl metine dönüş yapılamaz. Hash algoritmalarına MD4, MD5, SHA-1 örnek verilebilir[20].

Yakın geçmişte modern tekniklere *hibrid(hybrid)* ve *kuantum teknikler* olmak üzere yeni algoritmalar eklenmiştir.

Hybrid şifreleme teknikleri

Simetrik ve asimetrik şifreleme tekniklerinin birlikte kullanılması ile ortaya çıkmıştır. Her iki yöntemin üstün yönlerini birleştirerek zayıf yönlerini ortadan kaldırmayı amaçlar. Hibrid sistemlerde düz metinin şifrelenmesi işlemi simetrik algoritmalarla yapılır. Şifreleme anahtarı ise asimetrik algoritmalarla gizlenir. Mesaj ve anahtar bir arada karşı tarafa iletilir. Bilginin simetrik algoritma ile şifrelenmesi işlemlere hız kazandırırken, anahtarın asimetrik algoritma ile gizlenmesi güvenliği arttırır [38].

Kuantum şifreleme teknikleri

Tek kullanımlık anahtar kriptografi tekniğidir. Güvenli ve devamlı anahtar dağıtımını garantiler [38]. Kuantum mekaniğine ait belirsizlik ilkesi, foton polarizasyonu, dolaşıklık gibi yasalardan yararlanır [58].

3.6. Saldırı Teknikleri / Kriptoanaliz Tekniklerinin Sınıflandırılması

Kriptolu yani şifreli bir metin üzerinde, düz metine ulaşabilmek amacıyla yapılan tüm uygulamalar birer kriptoanaliz yani şifre çözme saldırısı ya da atağı olarak değerlendirilmektedir. Bu saldırı tekniklerinden bazıları genel olarak aşağıda açıklanmıştır [3,52].

3.6.1. Sadece şifreli metin saldırısı

Saldırıyı yapacak kişi mesajın içeriği ile ilgili hiç bir bilgiye sahip değildir. Sadece şifreli metini kullanarak düz metini elde etmeye çalışır. Uygulamada düz metine ilişkin tahminler yapılır. Pek çok klasik saldırıda şifreli metinin harf frekans analizi kullanılmaktadır [3, 19, 33, 34, 40].

Harf frekans analizi

Kriptoanaliz, şifreleme sistemlerinin kırılma çabaları sonucu ortaya çıkmıştır. Anahtar bilgisi olmadan, şifrelenmiş bir metini çözmek için mümkün olabilecek tüm şifre alfabeleri denemek gerekebilir. Bu durum oldukça zaman alıcıdır. Daha kolay bir yoldan çözümleme yapabilmek için matematik, istatistik ve dilbilim alanlarında yeterli bilgiye sahip olmak gereklidir. Tüm bu alanları içeren çalışmalardan ilki ünlü Arap filozofu Al-Kindi'nin yazdığı *Kriptografik Mesajların Deşifresi* isimli kitaptır [3, 22, 23]. Al-Kindi'ye ait olan bu eserde; kriptoanaliz metotları, kriptoanaliz şifreleri ve Arapça frekans analizi konuları yer almaktadır [22]. Al-Kindi'nin kriptoanaliz tekniğine göre, hangi dilde yazıldığı bilinen şifreli bir mesajı çözmek için, aynı dilde yazılmış uzun bir metin bulup her bir harfin kullanım sıklığını hesaplamak gereklidir. Düz metinde en sık kullanılan harf, şifreli metinde en sık kullanılan harfe denk gelecek şekilde eşleme yapılmalıdır [3, 22, 23].

Harflerin birebir eşlenmesi işlemi, sırasıyla tüm harfler için yapılır. Bu işlem bittikten sonra metindeki harfler ortaya çıkmış olur. Arap bilgini Al-Kindi, bu kriptoanaliz yöntemine frekans analizi adını vermiştir. Bir dile ait alfabedeki her harfin bir kullanım sıklığı yani bir frekansı vardır. Bir harfin ait olduğu dildeki frekansını tespit etmek için uzun metinler kullanılmalıdır. Harfin frekansı, o harfin metinde kaç kez kullanıldığının metindeki toplam harf sayısına bölümüdür. Bu sayı küçük sapmalar gösterebilir, fakat harflerin frekanslarının kendi aralarındaki öncelik sıraları genellikle değişmez [22, 23].

Aynı alfabenin kullanıldığı farklı dillerde harflerin kullanım sıklıkları aynı değildir. Örneğin Latin alfabesini kullanan Türkçe ve İngilizce dilleri incelenirse Türkçe'de en çok kullanılan harfin A, İngilizce'de ise E olduğu görülür. Frekans analizi çalışmalarında ilk olarak önemli olan şifrelenmiş metinin hangi dilde yazıldığını bilmektir. İkinci olarak o dile ait harf frekanslarının tespit edilmesi gereklidir. Şifreli metinde en çok kullanılan harf, metinin yazıldığı dilde en çok kullanılan harfe ya da harflerden birine karşılık gelecektir. Bu işlem şifre metinde en çok kullanılan harften en az kullanılan harfe doğru yapıldığı zaman, kriptoanalizde başarılı olma olasılığı yükselir [59-61].

Çizelge 3.2'de Türkçe alfabede kullanılan harflerin kullanım sıklıkları verilmiştir. Çizelge daha önce yapılmış olan çalışmalardan alınmış olduğu için harflerin kullanım sıklık değerleri daha önceki çalışmalarda incelenen metinlere göre tespit edilen değerleri içermektedir. İncelenen çalışmalarda harflerin kullanım sıklığı değerlerinin incelenen metinlere göre sayısal olarak değişiklik gösterebileceği, fakat harflerin kullanım sıklığı sıralamasında büyük bir farklılık olmadığı belirtilmiştir.

Çizelge 3.2. Türkçe harf frekansları [62]

Harf	**%**		**Harf**	**%**		**Harf**	**%**		**Harf**	**%**
A	11,82		D	4,63		O	2,47		Ğ	1,07
E	9		M	3,71		Ü	1,97		V	1
İ	8,34		Y	3,42		Ş	1,83		C	0,97
N	7,29		U	3,29		Z	1,51		Ö	0,86
R	6,98		T	3,27		G	1,32		P	0,84
L	6,07		S	3,03		Ç	1,19		F	0,43
I	5,12		B	2,76		H	1,11		J	0,03
K	4,7									

Çizelge 3.2'de görüldüğü üzere Türkçe'de en çok kullanılan sesli harfler A,E,İ ve en çok kullanılan sessiz harfler N,R,L,K,D harfleridir. En az kullanılan harfler C,Ö,P,F,J harfleridir. Tabloya dikkatli bakılırsa harflerin kullanım sıklığı

en fazla olandan en aza doğru sıralı olarak verildiği görülür. Sadece tekli harflere bakılarak yapılan frekans analizleri her zaman doğru çıkmayabilir. Bu gibi durumlarda harf ikililerine, üçlülerine de bakılmalıdır. Harf ikililerinde genellikle sesli harfler sessiz harflerle birlikte görülür.

Çizelge 3.3 ile Türk dil yapısında yan yana en sık kullanılan ikili harf grupları verilmiştir. Çizelge daha önce yapılmış olan çalışmalardan alınmış olduğu için harf ikililerinin kullanım sıklık değerleri daha önceki çalışmalarda incelenen metinlere göre tespit edilen değerleri içermektedir.

Çizelge 3.4 ile Türk dil yapısında yan yana en sık kullanılan üçlü harf grupları verilmiştir. Çizelge daha önce yapılmış olan çalışmalardan alınmış olduğu için harf üçlülerinin kullanım sıklık değerleri daha önceki çalışmalarda incelenen metinlere göre tespit edilen değerleri içermektedir.

Çizelge 3.3. Türkçe' de yan yana kullanılan harf ikililerinin (digram) kullanım sıklığı (100,000' de tekrarlama sıklığı) [62]

R	2273		Dİ	1021		NI	703		OL	586		AŞ	500		BE	433		KI	350
LA	2013		ND	980		AY	698		Sİ	578		NL	496		KE	424		RU	349
AN	1891		RA	976		YO	686		LI	576		TI	494		EY	421		Ğİ	347
ER	1822		AL	974		EK	683		RE	566		EM	494		ES	411		AZ	343
İN	1674		AK	967		RD	681		SI	565		ÜN	492		IK	407		İS	343
LE	1640		İL	870		TA	670		Mİ	564		DU	487		RL	393		Gİ	342
DE	1475		Rİ	860		AM	638		TE	562		GE	480		MI	392		Ğı	340
EN	1408		ME	785		DI	637		ET	560		AT	479		İK	379		AH	338
IN	1377		Lİ	782		SA	624		İM	541		SE	457		CA	379		YL	324
DA	1311		OR	782		İY	619		Tİ	537		ED	452		LD	362		ÜR	319
İR	1282		NE	738		Kİ	618		HA	528		UR	452		CE	361			
Bİ	1253		RI	733		UN	606		AS	527		ON	452		NU	359			
KA	1155		BA	718		NA	602		BU	516		KL	447		IŞ	355			
YA	1135		Nİ	716		AD	592		VE	508		IL	438		İZ	353			
MA	1044		EL	710		YE	588		IR	503		İŞ	434		LM	353			

Örneğin Çizelge 3.3'e göre incelenmiş olan 100,000 harflik bir metinde AR ikilisi 2273 kez tekrarlamıştır.

Çizelge 3.4. Türkçe' de yan yana kullanılan harf üçlülerinin(trigram) kullanım sıklığı (100,000' de tekrarlama sıklığı) [62]

LAR	1237		RIN	345		İYO	271		OLA	227		TAN	205		DİY	172
BİR	952		NLA	338		ELE	271		IĞI	226		NDİ	204		KLE	171
LER	949		DAN	338		İNE	266		EĞİ	223		KAL	204		VER	170
ERİ	764		IND	336		SİN	265		EME	223		ONU	201		EMİ	169
ARI	757		EDİ	326		ANL	263		INA	222		UNU	200		GÖR	169
YOR	643		ADA	321		KLA	262		ANA	220		END	199		RDI	169
ARA	521		AYA	316		ERE	262		KEN	218		ÇİN	198		SON	168
NDA	482		KAR	299		ALI	258		İÇİ	217		AĞI	194		ILA	167
İNİ	432		ALA	298		ELİ	256		IYO	217		ORD	194		BEN	166
INI	428		LAN	296		İYE	255		RLA	216		GEL	194		CAK	165
ASI	387		ENİ	294		BİL	246		MİŞ	213		MAN	192		İRİ	163
DEN	383		SIN	294		İLİ	245		YAN	212		ACA	192		EYE	163
NDE	383		İND	291		BAŞ	243		ECE	209		ÖYL	191		AŞI	162
RİN	372		ESİ	283		ARD	242		AYI	207		KAD	187		ÇIK	160
İLE	367		NIN	280		NİN	239		LMA	207		ERD	183		KAN	159
ANI	362		YLE	277		RDU	231		İĞİ	207		ORU	178			
AMA	357		ADI	273		MIŞ	229		EDE	206		RAK	177			

3.6.2. Bilinen düz metin saldırısı

Şifreli metini çözmek isteyen kişi, düz metnin bazı kısımlarına ya da tamamına ve ona karşılık gelen şifreli metine sahiptir. Elindeki bilgiyi kullanarak şifreli metin bloklarını çözebilir. Bilinen düz metin saldırısı yönteminde en sık kullanılan saldırı blok şifrelemeye karşı lineer kriptoanaliz saldırısıdır. Saldırının amacı şifreleme anahtarını bulmaktır [3, 19, 33, 34, 40].

Bu saldırı tekniği algoritma üzerinde uygulanmamıştır.

3.6.3. Seçilmiş düz metin saldırısı

Saldırıdaki amaç, şifreleme için kullanılan anahtarı belirlemektir. RSA gibi bazı kripto sistemler, seçilmiş-düz metin saldırısına karşı açıktır [19, 33, 34, 40]. Saldırganın(üçüncü kişi) şifreleme algoritmasını bildiği varsayılırsa yapılabilecek muhtemel saldırılardan biri *kaba kuvvet saldırısı (brute force)*

yaklaşımıdır. Kaba kuvvet saldırısında olası tüm anahtarların denenmesi gerekir. Bu yüzden saldırgan değişik istatistiksel uygulamalar, çeşitli testler yapmak zorundadır [19].

Bu saldırı tekniği algoritma üzerinde uygulanmamıştır.

3.6.4. Seçilen şifreli metin saldırısı

Seçilen şifreli metin saldırısında, kriptoanalist şifreli bir metin seçer ve şifreli metinle uyuşan düz metini bulmaya çalışır. Bu saldırı, deşifreleme için kullanılan özel bir cihazla şifreleme anahtarı olmaksızın yapılabilir [63].

Bu saldırı tekniği algoritma üzerinde uygulanmamıştır.

3.6.5. Uyarlanır seçili düz metin / şifreli metin saldırısı

Seçilen düz metin ya da seçilen şifreli metin saldırısı ile benzerdir. Tek fark uyarlanabilir seçili düz metin ya da şifreli metin saldırısında metinler rastgele seçilmeyip daha önceki şifre çözmelerde elde edilen bilgiler doğrultusunda seçilir [3,52].

Bu saldırı tekniği algoritma üzerinde uygulanmamıştır.

3.6.6. İlişkili anahtar atağı

Farklı anahtarlarla şifrelenmiş açık metinler setinin sonuçlarını değerlendirebilmeye bağlı bir saldırıdır [52].

Bu saldırı tekniği algoritma üzerinde uygulanmamıştır.

4.ARAŞTIRMANIN YÖNTEM, TEKNİK ve MATERYALLERİ

Tez kapsamında geliştirilen algoritma için ağırlıklı olarak yerine koyma ve yer değiştirme yöntemleri kullanılmıştır. Yerine koyma işlemi için harflerin Türkçe alfabedeki sıra numaraları kullanılmıştır. Bir sonraki adımda yer değiştirme yöntemi kullanılmış ve harflerin sıra numaraları arasında yer değiştirme işlemi gerçekleştirilmiştir. Yer değiştirme yöntemini uygulamak için permütasyon işleminden yararlanılmıştır. Bölüm 5.1'de permütasyon işleminden nasıl yararlanıldığı açıklanmıştır. Geliştirilen algoritmanın son adımları için cebirsel yöntemlerden olan modüler aritmetik kullanılmıştır. Bölüm 5.1'de modüler aritmetiğin algoritma üzerinde nasıl uygulandığı açıklanmıştır.

Bu yöntemler kullanılarak üretilen farklı kriptografi teknikleri vardır. Bu tekniklerden bazıları Bölüm 3'te verilmiştir. Geliştirilen algoritmada bu farklı tekniklerden Sezar şifreleme tekniğinin harf kaydırma(öteleme) prensibi, Alberti Diskinin harf kaydırma miktarının sabit olmaması özelliği, ENİGMA'nın üç kez şifreleme işlemi yapması, Vigenêre şifreleme tekniğinin farklı şifre alfabe prensibi, klasik tekniklerin çoğunun ortak özelliği olan mod alma işlemi özellikleri, bir araya getirilmiş ve uygulanmıştır.

4.1. Yöntemler

Şifreleme algoritmalarının üretilmesinde üç temel yöntem kullanılmaktadır [42, 64]:

- Yerine Koyma Yöntemleri (substitution)
- Yer Değiştirme Yöntemleri (transposition)
- Cebirsel Yöntemler

Bu yöntemler şifreleme işlemlerinde ayrı ayrı kullanılabildiği gibi birlikte de kullanılabilir [42, 64].

4.1.1.Yerine koyma yöntemi

Yerine koyma yöntemlerinde açık metindeki orijinal karakterler kimliklerini kaybeder, fakat bulundukları yer aynı kalır. Yani açık metindeki karakterlerin yerleri sabittir, karakterleri gizlemek için başka bir alfabenin karakterleri ya da sayılar kullanılır [42, 64]. Yerine koyma işlemleri maskeleme tekniği için kullanılır [29].

Yerine koyma yöntemleri; basit yerine koyma, homofonik yerine koyma, çok alfabeli yerine koyma, polygram yerine koyma olarak dört başlığa ayrılmaktadır. Basit yerine koymada düz ve şifreli metin harfleri arasında bire bir ilişki vardır. Düz metindeki her bir karakterin şifreli metinde tek bir karşılığı vardır. Homofonik yerine koyma, basit yerine koyma ile benzerdir. Tek fark, homofonik yerine koymada bire çok ilişki vardır, düz metindeki her bir harf şifreli metinde farklı harflerle şifrelenir. Çok alfabeli sistemler, düz metindeki karakterlerin şifreli karşılıklarına ulaşmak için çeşitli haritalar(planlar) kullanır. Polygram şifrelemede genellikle harf grupları için keyfi(rastgele) yerine koyma işlemi uygulanmaktadır [45]. Takip eden paragrafta yerine koyma yönteminin matematiksel olarak ifade edilişi görülmektedir:

A, $\{a_0 , a_1 , \ldots\ldots , a_{n-1} \}$ şeklinde n karakterli bir alfabe kümesi olsun. C, $\{f(a_0), f(a_1) , \ldots\ldots , f(a_{n-1})\}$ şeklinde n karakterli bir alfabe kümesi olsun. f:A→C , A kümesi ile C kümesi arasında her karakterin bire-bir eşlendiği bir fonksiyondur [45].

Örnek 4.1. Düz metin harfleri M harfi, şifreli metin $E_k(M)$ gösterimi ile temsil edilsin. Bu durumda şifreleme işlemi;

$M = m_1m_2m_3\ldots\ldots$

$E_k(M) = f(m_0), f(m_1)\ldots..$ şeklinde olur [45].

Örnek 4.1'in daha açık hale getirilmesi için Uygulama 4.1 verilmiştir.

Uygulama 4.1. Uygulama Türkçe alfabe için yapılmıştır.

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

Düz metin “merhaba”, *f*:A→C fonksiyonu ile şifrelendiğinde;

M = m e r h a b a
E_k(M) = g ı m o h a h

şifreli metin “gımohah” olarak elde edilir.

4.1.2. Yer değiştirme yöntemi

Yer değiştirme yöntemlerinde açık metindeki orijinal karakterler bulundukları yeri kaybeder, fakat kimlikleri değişmez. Metindeki harflerin yerine başka karakterler kullanılmaz, fakat karakterlerin sırası değiştirilir ve metin karmaşık hale getirilerek gizlenir [42, 64]. Yer değiştirme işlemleri gizleme tekniği için kullanılır [29].

Düz metindeki harfler belirli bir plana göre yeniden dizilir. Klasik olarak yeniden dizme işlemi bazı geometrik şekiller yardımı ile yapılır. İki aşamalı bir yöntemdir. İlk aşamada geometrik şeklin nasıl olacağına, ikinci aşamada ise harflerin şekil üzerine nasıl yerleştirileceğine karar verilmelidir [45].

Yer değiştirme yöntemi sütunlu(columnar) yer değiştirme ve periyodik yer değiştirme olmak üzere iki başlık altında incelenebilir. Sütunlu yer değiştirme yönteminde geometrik şekil genelde iki boyutlu bir matris dizisi olmaktadır. Örneğin düz metne ait harfler matrise satır satır yerleştirilirken, şifreli metne ait harfler sütun sütun seçilebilir. Sütunlu yer değiştirme yöntemi farklı dizi boyutları için de uygulanabilir. Örnek 4.2’de sütunlu yer değiştirme yöntemi açıklanmıştır. Periyodik yer değiştirme yönteminde bir periyot(her bir gruptaki harf sayısı) belirlenir. Düz metindeki harflerin yer değiştirme işlemini belirli bir permütasyon ile yapar. Matematiksel olarak ifade edilirse; *d,* permütasyon

işleminin yapılacağı karakter sayısını verirse ve Z karakter kümesini temsil ederse fonksiyon; f:$Z_d \rightarrow Z_d$ şeklinde bir permütasyon olur [45].

Düz metin; $M = m_1 \ldots m_d\, m_{d+1} \ldots. m_{2d} \ldots$

Şifreli metin; $E_k(M) = m_{f(1)} \ldots m_{f(d)}\, m_{d+f(1)} \ldots m_{d+f(d)}$

şeklindedir [45]. Örnek 4.3'te periyodik yer değiştirme yöntemi açıklanmıştır.

Örnek 4.2.

Şekil 4.1'de "bilgisayar" düz metninin 3X4'lük bir matrise satır satır yerleştirilişi gösterilmiştir.

	1	**2**	**3**	**4**
1	B	İ	L	G
2	İ	S	A	Y
3	A	R		

Şekil 4.1. Yer değiştirme yöntemi için uygulama örneği matrisi

Örnek 4.2'de geometrik şekil olarak 3X4'lük bir matris kullanılmıştır. Harf yerleşimi matris düzeninde görüldüğü gibi yapılmıştır. Sütunların hangi sırayla seçileceğinin belirli bir kuralı yoktur. Eğer sütunlar 3-1-4-2 sıralaması ile seçilirse;

Şifreli metin : LABİAGYİSR olarak elde edilir.

Örnek 4.3. d= 3 alınmıştır. Düz metin "bilgisayar" için;

i : 1 2 3

f(i) : 3 1 2

M = BİL GİS AYA R

E_k(M) = LBİ SGİ AAY R

Örnek 4.3'te görüldüğü üzere, metin 3'erli harf gruplarına ayrılmış ve her gruba kendi içinde permütasyon işlemi uygulanmıştır. Düz metinde 1 numaralı harf şifreli metinde üçüncü sıraya, düz metinde 2 numaralı harf şifreli metinde birinci sıraya yerleştirilmiştir. İşlemler bu şekilde devam ettirilmektedir. Eğer blok uzunlukları *d* değerinden daha küçükse permütasyon işleminde en yakın değere göre işlemler devam ettirilir [45].

4.1.3. Cebirsel yöntemler

Cebirsel yöntemlerde ise karmaşık yer değiştirme, yerine koyma ya da her iki yöntemi matematiksel dönüşümler yardımıyla kullanan yöntemler kullanılır [42, 64]. Matematiksel dönüşüm işlemini asal sayılar, modüler aritmetik, eliptik eğriler, sonlu cisimler, çarpma, üs alma gibi işlemler yardımıyla yapar [58].

4.2. Teknik ve Materyal

Tez çalışmasında geliştirilen algoritmanın similasyonu için donanımsal olarak Windows XP SP3 işletim sistemine sahip Dell Inspiron 6000 dizüstü bilgisayar ve bir adet mikrofon, yazılımsal olarak Microsoft Visual Studio 2010 programı C# dili ve DikteApi demo programı kullanılmıştır.

C# dili ile gerekli kodlamalar yapılmıştır. Arayüzlerin tasarımında *Windows Form* kullanılmıştır. Geliştirilen uygulama projesinin algoritma ile ilgili kısmı bellekte yaklaşık olarak 3MB yer kaplamaktadır. Projeye ait olan aktivite diyagramları ve performans analizi raporları da Visual Studio 2010 paket programı ile proje dosyası içinde hazırlanmıştır. Projenin toplam bellek kullanımı yaklaşık olarak 34 MB'dir.

DikteApi demo programı projeye konuşma tanıma özelliklerinin eklenebilmesini sağlamaktadır. Programın bilgisayara kurulumu gerçekleştirilmiştir. Program kendisini C dizini altında DikteAPIF20 dizinine yerleştirmektedir. Gerekli olan sistem dosyaları bu dizinde bulunmaktadır.

Kurulum esnasında sistem tarafından C:\Windows\system32 dizinine dikteAPIF20.dll dosyası yerleştirilmektedir.

DikteApi programına ait kütüphaneler C# ile hazırlanan projeye Project_Add Reference bölümünden eklenmiştir. Bilgisayar sisteminde .NET Framework 4.0 kurulu olduğu halde sistemde Framework ile ilgili hata çıkmıştır. NET Framework 2.0 ya da 3.5 sürümünün kurulması istenmiştir. 3.5 sürümü kurulmuş fakat sonuç alınamayınca 2.0 sürümü de kurulmuştur. Sistemin karşılaştığı bir diğer hata CrystalReport bileşen hatasıdır. Visual Studio 2010 paketi içinde hazır olarak gelmeyen Crystal Report bileşenleri internetten indirilerek bilgisayara kurulmuştur.

Yüksek Öğretim Kurumu Ulusal Tez Veritabanı, Springer, Elsevier, Scopus, IEEE Transaction, SCI ve SCI Expanded, EBSCO, Science Direct veritabanlarında yapılan kaynak taramasında ulaşılabilen dökümanlar incelenmiş ancak bazı kaynak dökümanlara erişim izni olmadığı ya da ücret ödemesi yapılmadığı için ulaşılamamıştır. Veritabanları ve ağ üzerinde ulaşılabilen dökümanlar ile kitap, dergi, tez, makale, bildiri, rapor gibi basılı materyaller ve dijital veriler incelenmiştir. Aynı şekilde Gazi Üniversitesi Kütüphanesi Online Katalog Taraması yapılmıştır. Blog ve sitelerde bulunan kişisel makale ve dökümanlar incelenmiş ve destekleyici bilgi olarak kullanılmıştır.

Geliştirilen algoritma klasik teknikler incelenerek hazırlandığı için algoritma üzerinde uygun saldırı tekniklerinden sadece harf frekans analizi saldırısı incelenmiştir. Diğer saldırı teknikleri incelenmemiştir.

Tez kapsamında geliştirilen algoritma sadece Türkçe için hazırlanmıştır. Herhangi başka bir dil üzerinde çalışmalar yapılmamıştır. Algoritma Türkçe alfabe kullanılarak geliştirildiği için harf frekans analizi değerlendirmeleri de sadece Türkçe için yapılmıştır.

Yazılım test tekniklerinden sadece sistem testleri başlığı altında incelenebilecek olan performans testi uygulanmıştır. Performans analizi için Microsoft Visual Studio C# ortamında Analyze menüsü kullanılmıştır. Performans analizi değerlendirmelerinde farklı uzunluktaki metinlerle yapılan çalışmalarda farklı değerler elde edildiği için tez kapsamında sunulan örnek metin 8 byte (64 bit) olarak sınırlandırılmıştır.

5. GELİŞTİRİLEN ALGORİTMA VE DEMONUN TANITIMI

Bu bölümde tez çalışmasında geliştirilmesi amaçlanan kriptografi algoritması ve algoritmanın similasyonu için hazırlanan demo programın tanıtımı bulunmaktadır. Bu çalışmanın kapsamındaki kriptografi algoritmasını geliştirmek için yerine koyma ve yer değiştirme işlemleri ile bazı matematiksel işlemler kullanılmıştır. Saldırı tekniklerinden sadece şifreli metin saldırısı için kullanılan harf frekans analizi saldırısı incelenmiş ve geliştirilen kriptografi algoritması üzerinde uygulanmıştır.

5.1. Algoritmanın Tanıtımı

Algoritma Türkçe Alfabe kullanılarak geliştirilmiştir. Şifreleme işlemi için farklı bir alfabe kullanılmamıştır. Algoritma geliştirilirken harflerin sıra numaralarından faydalanılmıştır.

Geliştirilen algoritmada kelimeler arası boşluklar atılmaktadır. Boşlukları temsil etmek için herhangi bir harf kullanılmamıştır. Bunun sebebi sabit karakterlerin, kelime uzunluklarının tahminen tespit edilmesini kolaylaştırmasıdır. Boşluk karakteri yerine dildeki kullanımı en az olan herhangi sabit karakter kullanımı harf frekans analizinde yanılma payını arttırabileceği gibi kelime uzunluklarının tahminini kolaylaştırarak şifreli metin üzerindeki kriptoanaliz çalışmalarını kolaylaştırabilmektedir. Bu nedenle boşluk karakterinin atılması tercih edilmiştir.

Algoritma, şifrelenmek için seçilen metin üzerinde iki farklı işlem gerçekleştirebilmektedir. Şifrelenmek üzere girilen metin, yapılan seçime göre kelime düzeyinde ya da paragraf düzeyinde şifrelenebilmektedir.

Kelime düzeyinde, şifrelenecek metinde kelimeler arası boşluklar atılmakta fakat her kelime kendi içinde işlemlerden geçmektedir. Kelimelerin uzunluğuna göre algoritma adım sayısı belirlenmektedir. Bir kelime üzerindeki işlem tamamlandığında algoritmadaki indis değerleri sıfırlanmakta ve işlemler yeniden başlamaktadır.

Paragraf düzeyinde, şifrelenecek metinde kelimeler arası boşluklar atılmaktadır. Uzun bir paragraf ya da birden fazla sayfadan oluşan herhangi dokümanda metin tek bir kelime olarak algılanmaktadır. Paragraf düzeyinde şifreleme işleminde algoritmadaki indis değerleri tek bir kez hesaplanır, işlemler başa dönmez ve algoritma tüm paragrafı tek bir kelime olarak kabul ederek çalışır.

Literatür taramalarında; boşlukların yerine sabit bir karakter konulmasının frekans analizinde yanılma payını arttırdığı, fakat bu yöntemin keşfedilmesinden sonra metinde en çok geçen harfin boşluk olarak tahmin edilmesiyle frekans analizini kolaylaştırdığı bilgisi ile karşılaşılmıştır. Bu nedenle boşlukların atılması uygun görülmüştür.

Aynı düz metin paragraf düzeyinde ve kelime düzeyinde şifrelendiğinde ortaya farklı şifreli metinler çıkmaktadır. Paragraf düzeyinde yapılan şifreleme işleminde indis değerleri sıfırlanmadan işlemler devam etmektedir ve bu durum harf çeşitliliğini etkilemektedir. Harf çeşitliliği azalmakta ve kendini tekrarlayan harf grupları artmaktadır. Bu sonuç, paragraf düzeyinde yapılan şifreleme işleminin harf frekans analizine karşı daha güvenilir olduğunu ortaya çıkarmıştır.

Düz metin şifreli hale getirildiğinde, düz metindeki birden fazla harf şifreli metinde aynı harfle temsil edilebilmektedir, aynı şekilde düz metindeki bir harf şifreli metinde birbirinden farklı harflerle temsil edilebilmektedir. Harflerin birbirinden farklı harflerle temsil edilmesi algoritmanın geri işlerliğini etkilemiş ve dolayısıyla şifreleme işlemi için uygulanan adımlar geri çevrilememiştir. Bu nedenle algoritma simetrik anahtarlı bir yapıda değildir. Bu durum biyometrik sistemlerden yararlanılabileceği fikrinin oluşmasına yol açmış ve şifrenin çözüm anahtarı olarak sesli komutlar kullanılmıştır. Bu konu algoritmanın similasyonu için hazırlanan demo programının tanıtımı bölümünde anlatılacaktır.

Yerine koyma yönteminde düz alfabedeki her bir karakter, alfabedeki harflerin rastgele karıştırılmasıyla elde edilen ve her harfin tek sefer kullanıldığı şifre alfabedeki karakterlerle birebir eşleştirilir [32].

Örnek 5.1. Yerine koyma işlemi

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

“YILDIZ” kelimesi yukarıda verilen tabloya bakılarak şifrelenmek istenirse; “FKNRKĞ” şifre metini elde edilir.

Yerine koyma yönteminin kullanılabilmesi için kullanıcı ve alıcının şifre alfabeye beraber karar vermeleri gerekmektedir. Yerine koyma yöntemi kullanılarak oluşturulan tüm olası şifre alfabe sayısı, mesajın iletildiği dilin alfabesindeki harf sayısına bağlıdır. Bu sayı Türkçe için 29!=8 841 761 993 739 701 954 543 616 000 000 tane farklı şifre alfabedir [33].

Geliştirilen algoritmada harflerin sayısal karşılıkları harflerin yerine kullanılmıştır. Şifreleme işleminin son adımında sayısal değerlere karşılık gelen harf değerleri kullanılmıştır.

Geliştirilen algoritma için aşağıda görülen işlem basamakları uygulanmaktadır:

1. Maksimum inversiyon değerini veren harf yerleşimini elde etmek.
2. Modüler aritmetiğe göre harf seçimi yapmak.
3. Şifrelenmiş metni elde etmek.

Kriptografi algoritması geliştirilirken Türkçe’deki 29 harf ve bu harflerin sıra numaraları kullanılmıştır. Yukarıda açıklandığı üzere alfabenin 29 harften oluşması şifreleme işlemi yapıldığında 29! şifre alfabe olasılığı demektir. Kelimelerdeki harflerin kayma miktarları sabit değildir. Kelimedeki harf sayısı n ile gösterilirse bu kelime için n! farklı harf yerleşimi elde edilir.

Aşağıda verilmiş olan çizelge 5.1'de harflerin sahip oldukları sıra numaraları görülmektedir:

Çizelge 5.1. Şifreleme işleminde kullanılacak harflerin sayısal karşılıkları

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

Kriptografi algoritmasının birinci adımı olarak harfler arasında maksimum inversiyon değerine ulaşılması hedeflenmiştir. Bunun için de permütasyon işlemi kullanılmıştır. Bir metnin içerdiği harflere göre maksimum inversiyon sayısı değişmektedir; örneğin kelime 4 harften oluşuyorsa 4! = 24 tane farklı harf yerleşimi elde edilir. Fakat geliştirilen algoritmaya göre bu yerleşimlerin içerisinde önemli olan maksimum inversiyonu sağlayan yerleşimdir. Aşağıda permütasyon ve maksimum inversiyon kavramları matematiksel olarak örneklenmiştir:

S = { 1, 2, 3, …, n } artan sırada düzenlenmiş 1-den n-ye kadar tamsayılar kümesi olsun. S'nin elemanlarının " $j_1 j_2 \ldots j_n$" şeklinde yeniden düzenlenmesine S'nin bir *permütasyonu* denir [65].

Örneğin S = { 1, 2, 3 } olsun. S'nin permütasyonları yazılmak istenirse ilk durumda S'nin n elemanından herhangi biri, ikinci durumda ise geriye kalan n-1 elemandan herhangi biri, üçüncü durumda ise geriye kalan n-2 elemandan herhangi biri yazılabilir. Bu durumda S'nin permütasyon sayısı n (n-1) (n-2) …. 2.1 = n! olarak bulunur [65].

Örnekteki S kümesinin eleman sayısı 3 olduğuna göre S'nin permütasyonlarının sayısı; 3! = 3.2.1 = 6 tanedir.

S'nin permütasyonları; 1 2 3, 1 3 2, 2 3 1, 2 1 3, 3 1 2, 3 2 1 şeklindedir. Eğer daha büyük bir tamsayı j_r , daha küçük bir tamsayı j_s'den önce gelirse $j_1 j_2 \ldots j_n$ permütasyonunun bir *inversiyonu* vardır [65].

Bu noktadan yola çıkılarak şifrelenecek metindeki harflerin sayısal değerlerinden yararlanılarak *maksimum inversiyon* değerine ulaşılması düşünülmüş ve böylece harfler arasında yer değiştirme işlemi gerçekleştirilmiştir. Kriptografi algoritmasının birinci adımı olan bu işlem için aşağıda küçük bir örnek gösterilmiştir:

"KİTAP" kelimesi örnek olarak alınırsa; kelime 5 harfli olduğu için toplamda 120 farklı permütasyon söz konusudur. Bu 120 dizilim içerisinde maksimum inversiyonu bulmak için sayısal değeri en büyük olandan en küçük olana doğru bir sıralama yapılması yeterlidir.

Çizelge 5.2'de geliştirilen algoritmanın ilk adımı için küçük bir örnek verilmiştir:

Çizelge 5.2. Algoritmanın ilk adımı için örnek uygulama

Harf	K	İ	T	A	P
Sıra No	13	11	23	0	19
Max İnversiyon	23	19	13	11	0
Yer değiştirme	T	P	K	İ	A

Çizelge 5.2'de maksimum inversiyon değerini veren harf yerleşimi gösterilmektedir. Maksimum inversiyon değerini veren yerleşimi elde ettikten sonra permütasyonda yer değiştiren harflerin sayısal değerlerini 28'e tamamlayan karşılıkları maskeleme işlemi için kullanılır.

Çizelge 5.3'te geliştirilen algoritmanın bir sonraki adımının uygulanışı görülmektedir:

Çizelge 5.3. Algoritmanın ikinci adımı için örnek uygulama

Harf	T	P	K	İ	A
Sıra No	23	19	13	11	0
28'e tümleyen sayısal değer	5	9	15	17	28
Yerine koyma	E	H	M	O	Z

Çizelge 5.3’te görüldüğü üzere 28’e tümleyen sayıya karşılık gelen harfler yapılan yeni yerleşimde minimum inversiyona göre dizilmiştir.

Algoritmanın elde edilmesindeki bir sonraki adım Çizelge 5.4’te görülmektedir:

Çizelge 5.4. Algoritmanın bir sonraki adımı için örnek uygulama

“EHMOZ” maksimum inversiyon	28	17	15	9	5
“EHMOZ” minimum inversiyon	5	9	15	17	28
TOPLAM	33	26	30	26	33

Elde edilen toplam değerler modüler işleme tabi tutularak şifrelenmiş metine ulaşılması gerçeklenmiştir. Burada önemli olan değişkenlerden biri de harflerin indis değerleridir. Çizelge 5.5’te bu adım için örnek bir uygulama görülmektedir.

Eğer toplam değer 28’den büyük ve 28’e eşit ise toplamın mod28’e göre değeri hesaplanır. Elde edilen sonuç, şifrelenmiş metini oluştururken kullanılacak olan harfin sayısal karşılığıdır. Bu sonucun gösterdiği sayısal değer düz alfabedeki sıra numaraları ile eşleştirilir ve eşleşen harf şifreleme işlemi için seçilir. Eğer toplam değeri 28’den küçükse o harfin indis değeri kullanılır, eğer indis değeri tek sayı ise minimum inversiyondaki harf, çift sayı ise maksimum inversiyondaki harf şifrelenmiş metini oluşturmak için seçilir. Çizelge 5.5’te yukarıdaki paragrafta anlatılan işlemler adım adım gösterilmektedir:

Çizelge 5.5. Algoritmanın üçüncü adımı için örnek uygulama

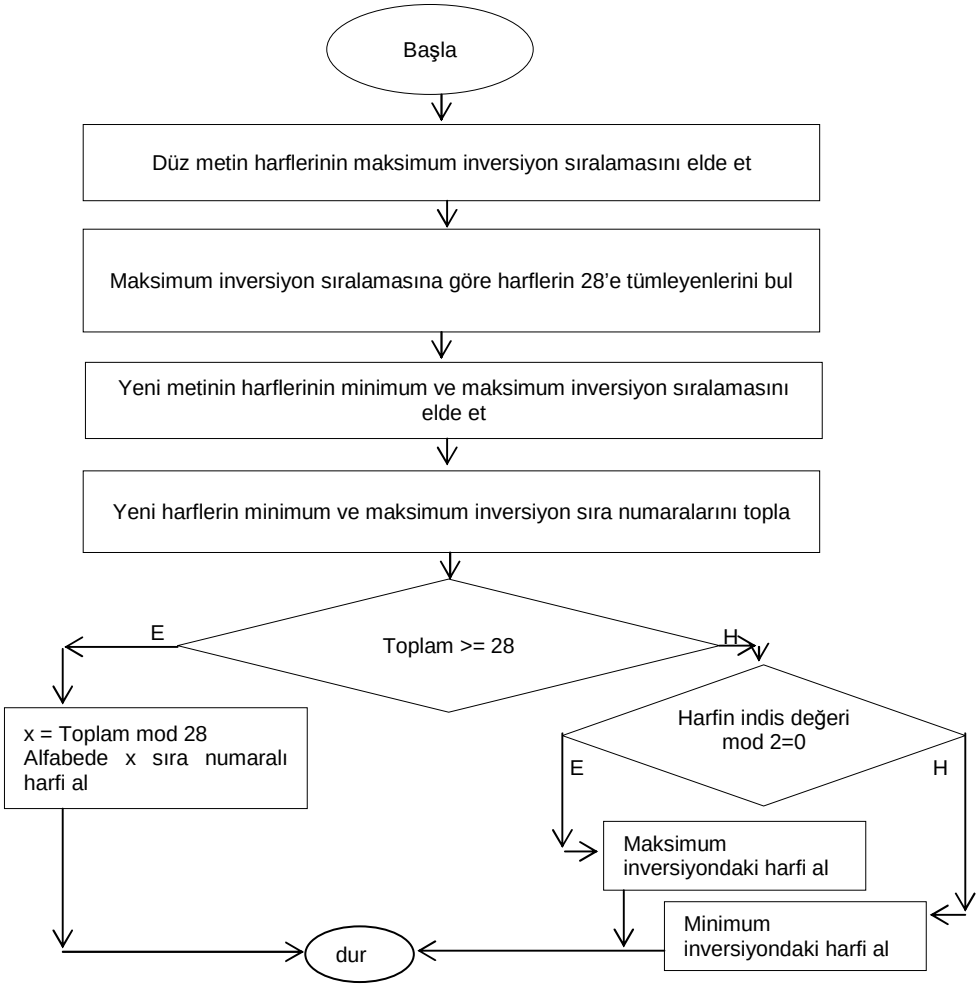
Harflerin indisleri	0	1	2	3	4
“EHMOZ” maksimum inversiyon	28	17	15	9	5
“EHMOZ” minimum inversiyon	5	9	15	17	28
Toplam	33	26	30	26	33
	33>28	26<28	30>28	26<28	33>28
	33>28	26<28	30>28	26<28	33>28
	33mod28=5 düz alfabedeki 5 numaralı harf	İndis değeri tek, minimum inversiyondaki harf	30mod28=2 düz alfabedeki 2 numaralı harf	İndis değeri tek, minimum inversiyondaki harf	33mod28=5 düz alfabedeki 5 numaralı harf
	E	**H**	**C**	**O**	**E**

Düz Metin: KİTAP

Şifrelenmiş Metin : EHCOE

Maksimum ve minimum inversiyon değerinin toplanıp mod 28’e göre değerlendirmesi aşaması VERNAM şifreleme algoritmasından esinlenilerek yapılmıştır. Üst üste iki kez yer değiştirme işlemi, ENİGMA makinesinin şifreleme mantığından esinlenilmiştir. Geliştirilen algoritmada bir harf birden fazla sayıda harf ile şifrelenebilmektedir. Aynı şekilde birden fazla sayıda sayıda harf bir harf ile şifrelenebilmektedir.

Geliştirilen algoritmanın genel akış diyagramı Şekil 5.1’de gösterilmiştir. Görülen akış diyagramı koşul değerlendirmesi bir adım(bir harf) için çalıştırılmış algoritmayı temsil etmektedir.



Şekil 5.1. Koşul değerlendirmesi bir harf için gösterilmiş akış diyagramı

Örnek 5.2'de geliştirilen algoritma ile kelime düzeyinde şifrelenmiş küçük bir düz metin görülmektedir:

Örnek 5.2. Şifreleme işleminde kullanılan düz metin

Düz metin : "Bilişim teknolojilerinin bizlere sağladığı kolaylıklar arttıkça, elektronik ortamların kullanımı yaygınlaşmakta, bilginin işlendiği, taşındığı, saklandığı ortamlara erişimler zamandan, mekandan bağımsız hale gelmektedir" [3].

Bilişim : 1,11,14,11,22,11,15

MAX İNV = Ş M L İ İ B → 22,15,14,11,11,11,1

Mod 28'e göre = 6,13,14,17,17,17,27 → FKLOOOY

Teknolojilerinin : 23,5,13,16,17,14,17,12,11,14,5,20,11,16,11,16
MAX İNV = TROONNNLLKJİİİEE
→23,20,17,17,16,16,16,14,14,13,12,11,11,11,5,5,5
Mod 28'e göre = 5,8,11,11,12,12,12,14,14,15,16,17,17,17,23,23,23
→EĞİİJJJLLMNOOOTTT

Bizlere : 1,11,28,14,5,20,5
MAX İNV = ZRLİEEB →28,20,14,11,5,5,1
Mod 28'e tamamlayan = 0,8,14,17,23,23,27→ AĞLOTTY

Sağladığı : 21,0,8,14,0,4,10,8,10
MAX İNV = SLIIĞĞDAA →21,14,10,10,8,8,4,0,0
Mod 28'e göre = 7,14,18,18,21,21,24,28,28→ GLÖÖSSUZZ

Kolaylıklar : 13,17,14,0,27,14,10,13,14,0,20
MAX İNV = YROLLLKKIAA →27,20,17,14,14,14,13,13,10,0,0
Mod 28'e göre = 1,8,11,14,14,14,15,15,18,28,28→ BĞİLLLMMÖZZ

Arttıkça : 0, 20,23,23,10,13,3,0
MAX İNV = TTRKIÇAA →23,23,20,13,10,3,0,0
Mod 28'e göre = 5,5,8,15,18,25,28,28→ EEĞMÖÜZZ

Elektronik : 5, 14,5,13,23,20,17,16,11,13
MAX İNV = TRONLKKİEE →23,20,17,16,14,13,13,11,5,5
Mod 28'e göre = 5,8,11,12,14,15,15,17,23,23→ EĞİJLMMOTT

Ortamların : 17,20,23,0,15,14,0,20,10,16
MAX İNV = TRRONMLIAA →23,20,20,17,16,15,14,10,0,0
Mod 28'e göre = 5,8,8,11,12,15,14,18,28,28→ EĞĞİJMLÖZZ

Kullanımı : 13,24,14,14,0,16,10,15,10
MAX İNV = UNMLLKIIA →24,16,15,14,14,13,10,10,0
Mod 28'e göre = 4,12,13,14,14,15,18,18,28→ DJKLLMÖÖZ

Yaygınlaşmakta : 27,0,27,7,10,15,14,0,22,15,0,13,23,0
MAX İNV = YYTŞMMLKIGAAAA →27,27,23,22,15,15,14,13,10,7,0,0,0,0
Mod 28'e göre = 1,1,5,6,13,13,14,15,18,21,28,28,28,28→
BBEFKKLMÖSZZZZ

Bilginin : 1,11,14,7,11,16,11,16
MAX İNV = NNLİİİGB →16,16,14,11,11,11,7,1
Mod 28'e göre = 12,12,14,17,17,17,21,27→ JJLOOOSY

İşlendiği : 11,22,14,5,16,4,11,8,11
MAX İNV = ŞNLİİİĞED →22,16,14,11,11,11,8,5,4
Mod 28'e göre = 6,12,14,17,17,17,20,23,24→ FJLOOORTU

Taşındığı : 23,0,22,10,16,4,10,8,10
MAX İNV = TŞNIIIĞDA →23,22,16,10,10,10,8,4,0
Mod 28'e göre = 5,6,12,18,18,18,20,24,28→ EFJÖÖÖRUZ

Saklandığı : 21,0,13,14,0,16,4,10,8,10
MAX İNV = SNLKIIĞDAA →21,16,14,13,10,10,8,4,0,0
Mod 28'e göre = 7,12,14,15,18,18,20,24,28,28→ GJLMÖÖRUZZ

Ortamlara : 17,20,23,0,15,14,0,20,0
MAX İNV = TRROMLAAA →23,20,20,17,15,14,0,0,0
Mod 28'e göre = 5,8,8,11,13,14,28,28,28→ EĞĞİKLZZZ

Erişimler : 5,20,11,22,11,15,14,5,20
MAX İNV = ŞRRMLİİEE →22,20,20,15,14,11,11,5,5
Mod 28'e göre = 6,8,8,13,14,17,17,23,23→ FĞĞKLOOTT

Zamandan : 28,0,15,0,16,4,0,16
MAX İNV = ZNNMDAAA →28,16,16,15,4,0,0,0
Mod 28'e göre = 0, 12,12,13,24,28,28,28→ AJJKUZZZ

Mekandan : 15,5,13,0,16,4,0,16
MAX İNV = NNMKEDAA →16,16,15,13,5,4,0,0
Mod 28'e göre = 12,12,13,15,23,24,28,28→ JJKMTUZZ

Bağımsız : 1,0,8,10,15,21,10,28
MAX İNV = ZSMIIĞBA →28,21,15,10,10,8,1,0
Mod 28'e göre = 0,7,13,18,18,20,27,28→AGKÖÖRYZ

Hale : 9,0,14,5
MAX İNV = LHEA →14,9,5,0
Mod 28'e göre = 14,19,23,28→ LPTZ

Gelmektedir : 7,5,14,15,5,13,23,5,4,11,20
MAX İNV = TRMLKİGEEED →23,20,15,14,13,11,7,5,5,5,4
Mod 28'e göre = 5,8,13,14,15,17,21,23,23,23,24→ EĞKLMOSTTTU

Uygulama Örneği 5.1, Uygulama Örneği 5.2 ve Uygulama Örneği 5.3'te görülen sonuçlar, şifreleme işleminin kelime düzeyinde yapılması sonucu elde edilmiştir:

Uygulama Örneği 5.1. Düz metindeki kelimelerin maksimum inversiyonlarına göre yeni metin:

ŞMLİİİB TROONNNLLKJİİİEE ZRLİEEB SLIIĞĞDAA YROLLLKKIAA TTRKIÇAA TRONLKKİEE TRRONMLIAA UNMLLKIIA YYTŞMMLKIGAAAA NNLİİİGB ŞNLİİİĞED TŞNIIIĞDA SNLKIIĞDAA TRROMLAAA ŞRRMLİİEE ZNNMDAAA NNMKEDAA ZSMIIĞBA LHEA TRMLKİGEEED

Maksimum inversiyona göre harf dizilimleri yapıldıktan sonraki işlem 28'e tümleyen harf karşılıklarını bulmaktır. Uygulama Örneği-2'de maksimum inversiyona göre dizilmiş harflerin yeni dizilişleri verilmiştir:

Uygulama Örneği 5.2. Kelimelerin maksimum inversiyonlarına göre oluşturulan metine göre harflerin 28'e tümleyen harflere göre yeni dizilişi:

FKLOOOY EĞİİJJJLLMNOOOTTT AĞLOTTY GLÖÖSSUZZ BĞİLLLMMÖZZ EEĞMÖÜZZ EĞİJLMMOTT EĞĞİJMLÖZZ DJKLLMÖÖZ BBEFKKLMÖSZZZZ JJLOOOSY FJLOOORTU EFJÖÖÖRUZ GJLMÖÖRUZZ EĞĞİKLZZZ FĞĞKLOOTT AJJKUZZZ JJKMTUZZ AGKÖÖRYZ LPTZ EĞKLMOSTTTU

Çizelge 5.6'da geliştirilen algoritmada uygulanan adımların tümü görülebilir:

Çizelge 5.6. Algoritmadaki tüm adımların uygulanışı

Düz Metindeki Kelime	B	İ	L	İ	Ş	İ	M
Maksimum İnversiyon	Ş(22)	M(15)	L(14)	İ(11)	İ(11)	İ(11)	B(1)
Mod 28'e göre Harfler	F	K	L	O	O	O	Y
Harflerin indisleri	0	1	2	3	4	5	6
FKLOOOY Minimum inversiyon	6	13	14	17	17	17	27
FKLOOOY maksimum inversiyon	27	17	17	17	14	13	6
Toplam	33	30	31	34	31	30	33
	33>28	30>28	31>28	34>28	31>28	30>28	33>28
	33mod28= 5 düz alfabedeki 5 numaralı harf	30mod28= 2 düz alfabedeki 2 numaralı harf	31mod28= 3 düz alfabedeki 3 numaralı harf	34mod28= 6 düz alfabedeki 6 numaralı harf	31mod28= 3 düz alfabedeki 5 numaralı harf	30mod28= 2 düz alfabedeki 2 numaralı harf	33mod28= 5 düz alfabedeki 5 numaralı harf
Şifreli Hali	E	C	Ç	F	Ç	C	E

Çizelge 5.6'da düz metinin ilk kelimesi olan "Bilişim" kelimesinin geliştirilen şifreleme algoritması ile şifreleme işlemleri sonrasındaki yeni hali görülmektedir.

Uygulama Örneği 5.3. Örnek 5.2'deki düz metin için kelime düzeyinde şifreleme işleminden elde edilen son sonuçlar

BİLİŞİM → ŞMLİİİB →FKLOOOY → ECÇFÇCE
Teknolojilerinin→ TROONNNLLKJİİİEE→ EĞİİJJJLLMNOOOTTT → AÇAABANAAMABAAÇA
Bizlere → ZRLİEEB → AĞLOTTY → ZÇHFHÇB
Sağladığı → SLIIĞĞDAA → GLÖÖSSUZZ→ GLLIJILLG
Kolaylıklar →YROLLLKKIAA → BĞİLLLMMÖZZ → BĞBBBABBBĞB
Arttıkça →TTRKIÇAA → EEĞMÖÜZZ → EEEEEEEE
Elektronik →TRONLKKİEE→ EĞİJLMMOTT → AÇAJBBKAÇA
Ortamların →TRRONMLIAA → EĞĞİJMLÖZZ → EĞRİNKLÖĞE
Kullanımı → UNMLLKIIA → DJKLLMÖÖZ → DCÇBABÇCD
Yaygınlaşmakta → YYTŞMMLKIGAAAA → BBEFKKLMÖSZZZZ → BBEFEÇBBÇEFEBB
Bilginin → NNLİİİGB → JJLOOOSY → İEÇFFÇEİ
İşlendiği → ŞNLİİİĞED → FJLOOORTU → CGFFFFFGC
Taşındığı → TŞNIIIĞDA → EFJÖÖÖRUZ → ECDĞĞĞDCE
Saklandığı → SNLKIIĞDAA → GJLMÖÖRUZZ → GJIGĞĞGIJG
Ortamlara → TRROMLAAA → EĞĞİKLZZZ → EĞĞİMLĞĞE
Erişimler → ŞRRMLİİEE → FĞĞKLOOTT → BÇRCACİÇB
Zamandan → ZNNMDAAA → AJJKUZZZ → AJJHHJJA
Mekandan→ NNMKEDAA → JJKMTUZZ → JJHIIHJJ
Bağımsız → ZSMIIĞBA → AGKÖÖRYZ → AFEĞĞEFA
Hale → LHEA → LPTZ → LLLL
Gelmektedir → TRMLKİGEEED → EĞKLMOSTTTU → BÇĞHĞFĞHĞÇB

<u>Kelime düzeyinde şifreleme işleminden sonra elde edilen şifreli metin:</u>
ECÇFÇCEAÇAABANAAMABAAÇAZÇHFHÇBGLLIJILLGBĞBBBABBBĞBE
EEEEEEEAÇAJBBKAÇAEĞRİNKLÖĞEDCÇBABÇCDBBEFEÇBBÇEFEBBİ
EÇFFÇEİCGFFFFFGCECDĞĞĞDCEGJIGĞĞGIJGEĞĞİMLĞĞEBÇRCACİ
ÇBAJJHHJJAJJHIIHJJAFEĞĞEFALLLLBÇĞHĞFĞHĞÇB

Paragraf düzeyinde şifreleme işlemi sonucu elde edilen şifreli metin:

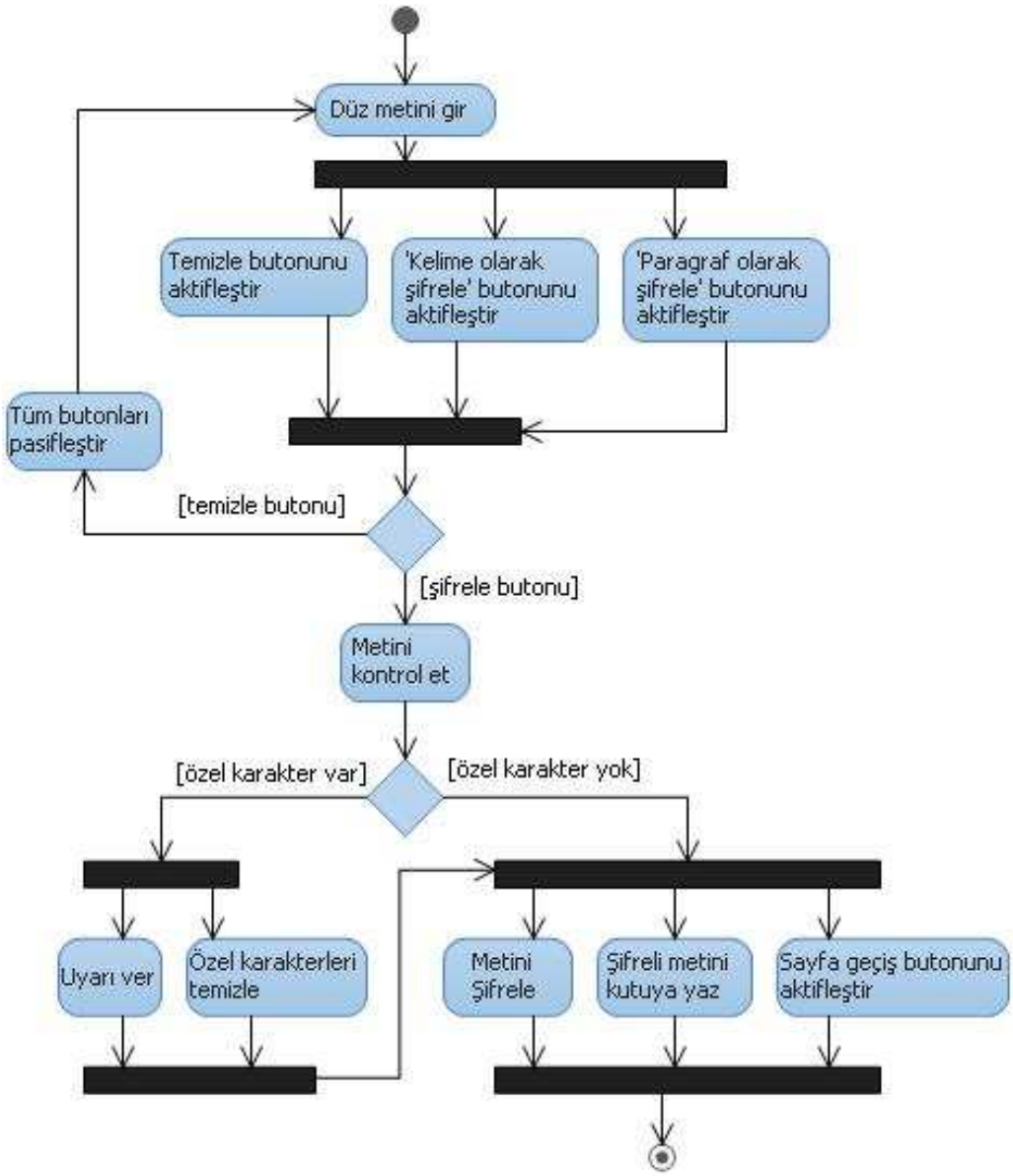
AAABBBDEEEEEEEEEFFFFFGGGĞĞGGGGEDDDDDGGFFFFGGGGGG
GGGGEEEDDDEEEDÇÇÇÇÇDDDDDDDDDÇÇÇÇÇÇÇÇÇDDDDDDÇ
CCCCÇDDDDDDÇÇÇÇÇÇÇÇÇDDDDDDDDDÇÇÇÇÇDEEEDDDEEE
GGGGGGGGGGFFFFGGDDDDDEGGGGĞĞGGGFFFFFEEEEEEEEEDBB
BAAA

Kelime ve paragraf düzeyinde şifrelenerek elde edilen metinler incelendiğinde kelime düzeyinde ECÇFÇCE olarak şifrelenmiş BİLİŞİM kelimesi paragraf düzeyinde AAABBBD olarak şifrelenmiştir. İki metin üzerinde yapılan incelemeler harf frekans analizi saldırısında paragraf düzeyinde şifreleme işleminin daha güvenilir olduğunu göstermiştir.

5.2. Demonun Tanıtımı

Demo olarak geliştirilen proje *kriptografi* olarak adlandırılmıştır. Program için dört arayüz oluşturulmuştur. Arayüzler için C# programı altında windows form türü seçilmiştir. Arayüzlere sırası ile *sfr, sfrack, sfrhrf, sfrdsf* isimleri verilmiştir. Programın kaynak kodlarında bazı bölümlerde kısaltmalar kullanılmıştır. Örneğin Button'lar *btn*, listBox'lar *lb*, textBox'lar *tbx*, RichTextBox'lar *rtb* olarak düzenlenmiştir.

Formlara ait aktivite diyagramları, açıklamalar ve ekran görüntüleri takip eden sayfalarda görülebilir. Programın ilk formuna ait aktivite diyagramı Şekil 5.2'de verilmiştir:



Şekil 5.2. Şifreleyiş işlemlerinin yapıldığı sfr formuna ait aktivite diyagramı

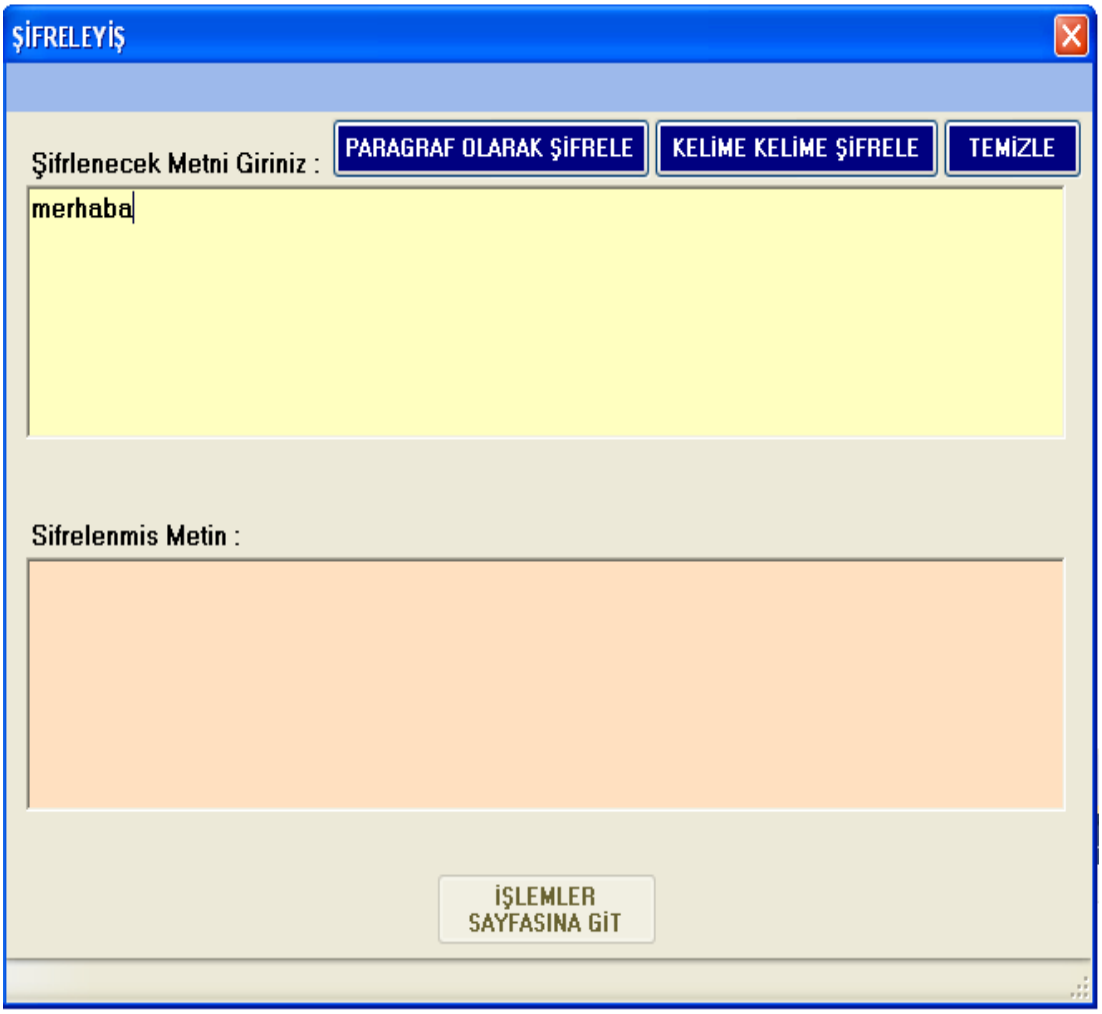
Şekil 5.2'de verilen akış diyagramında görüldüğü gibi işlemler düz metinin girişi ile başlamaktadır. Aktifleşen butonlardan seçilen butona göre uygun işlem yapılmaktadır. Şifrele butonu tıklandığında geliştirilen algoritmanın işlem basamakları uygulanmaktadır. Özel karakter kontrolü yapabilmek için özel bir dizi tanımlanmıştır. Dizi elemanları; [,], (,), {, }, boşluk karakteri, virgül karakteri , ., :, ;, !, ?, ‘,’, “,”, \, /, #, ^, $, %, &, =, *, -, _, @, <, >, |, +, ~, q, Q, w, W, x, X, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 olarak belirlenmiştir. Şifreleme

işleminde kullanılacak olan alfabenin elemanları da sabit bir dizi olarak tanımlanmıştır. Bu sabit dizi Türkçe alfabedeki Latin harflerinden oluşmaktadır. Eğer metin özel karakter içeriyorsa ekranda "*Şifrelemek istediğiniz metinde tanımlanmış alfabedeki karakterlerden farklı karakterler bulunmaktadır, metin yeniden düzenlendikten sonra şifreleme işlemi gerçekleştirilecektir.*" uyarı mesajı verilmektedir. Metin özel karakter dizisine göre kontrol edilip yeniden düzenlendikten sonra şifreleme işlemi gerçekleştirilmektedir. Şifrelenmiş metin ilgili kutuya atandığında diğer sayfalara geçiş butonu aktifleşmektedir. Sayfalar arası gezintiyi sağlayan nesneler ve diğer arayüzler demo programı ve geliştirilen algoritmayı adım adım tanıtmak için hazırlanmıştır.

Şekil 5.2'de görülen aktivite diyagramı Resim 5.1, Resim 5.2, Resim 5.3 ve Resim 5.4'ü temsil etmektedir. Bu dört resim şifreleme işlemlerinin yapıldığı ilk forma aittir. Resim 5.1'de program ilk çalıştırıldığında ekrana gelen arayüz görülmektedir. Resim 5.2'de düz metin girişi yapıldığında resmin üst kısmındaki butonların aktifleştiği görülebilir. Resim 5.3'de şifreleme butonunun kullanımı sonucu şifreli metnin kutuya aktarıldığı ve alttaki butonun aktifleştiği görülebilir. Resim 5.4'te ise ekrana çıkan uyarı mesajı görülmektedir.

Resim 5.1. Şifreleyiş işleminin yapıldığı arayüz

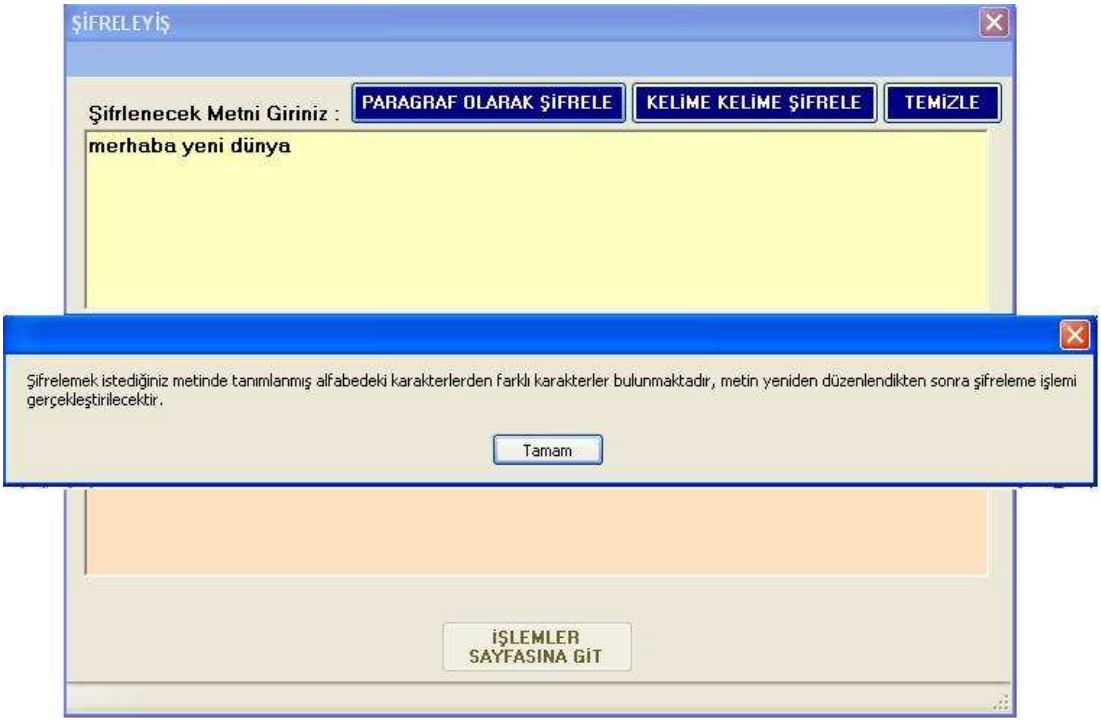
Resim 5.1'de görülen arayüz programın giriş sayfasını oluşturan *sfr* formudur. Form üzerinde iki adet RichTextBox ve üç adet Button kullanılmıştır. Form ilk yüklendiğinde tüm butonlar etkileşime kapalıdır. Metin girişleri ve butonların tıklanması durumlarına göre arayüzde görülen üç butonun etkileşim özellikleri değişmektedir. Üstteki RichTextBox düz metinin girişi için kullanılırken alttaki RichTextBox metnin şifrelenmiş halini göstermektedir. Arayüzün en altında statusStrip1, en üstünde de menuStrip1 bileşenleri görsel olarak kullanılmıştır.



Resim 5.2. Düz metinin girişi

Metin girişi yapıldıktan sonra ŞİFRELE ve TEMİZLE butonları aktif hale gelmektedir. ŞİFRELE butonu *btn2*, TEMİZLE butonu *btn3*, İşlemler Sayfasına Git butonu *btn4* olarak adlandırılmıştır. Şifreleme butonuna tıklanınca girilen metin üzerinde özel karakter ve sayı kontrolü yapılmakta ve sonrasında şifreleme işlemi gerçekleştirilmektedir. Temizle butonu her iki RichTextBox'ın içeriğini boşaltmaktadır. İşlemler Sayfasına Git butonu ile bir sonraki arayüze geçiş sağlanmıştır.

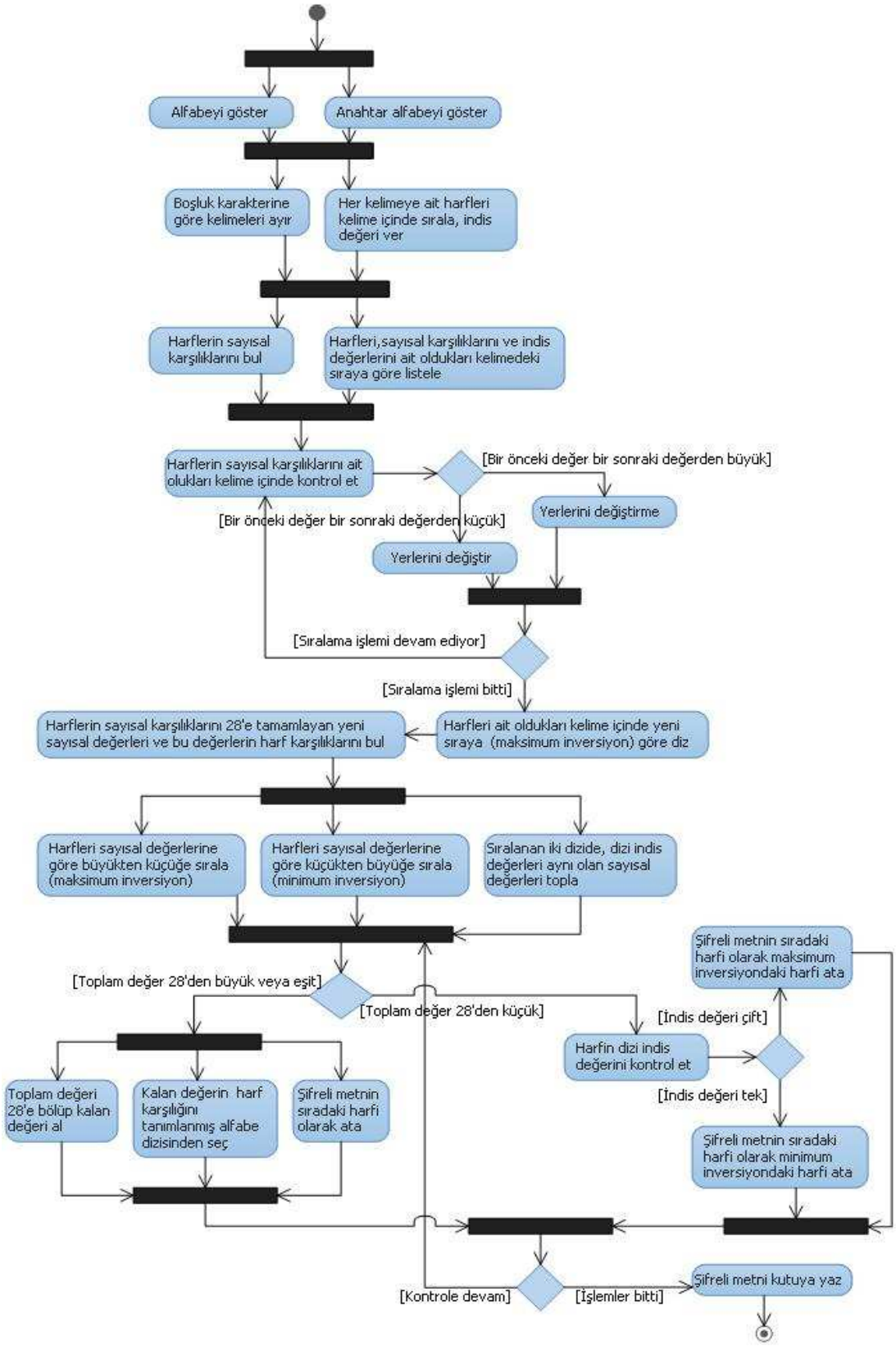
Resim 5.3. Şifrele butonu tıklandıktan sonra arayüzün yeni görünümü



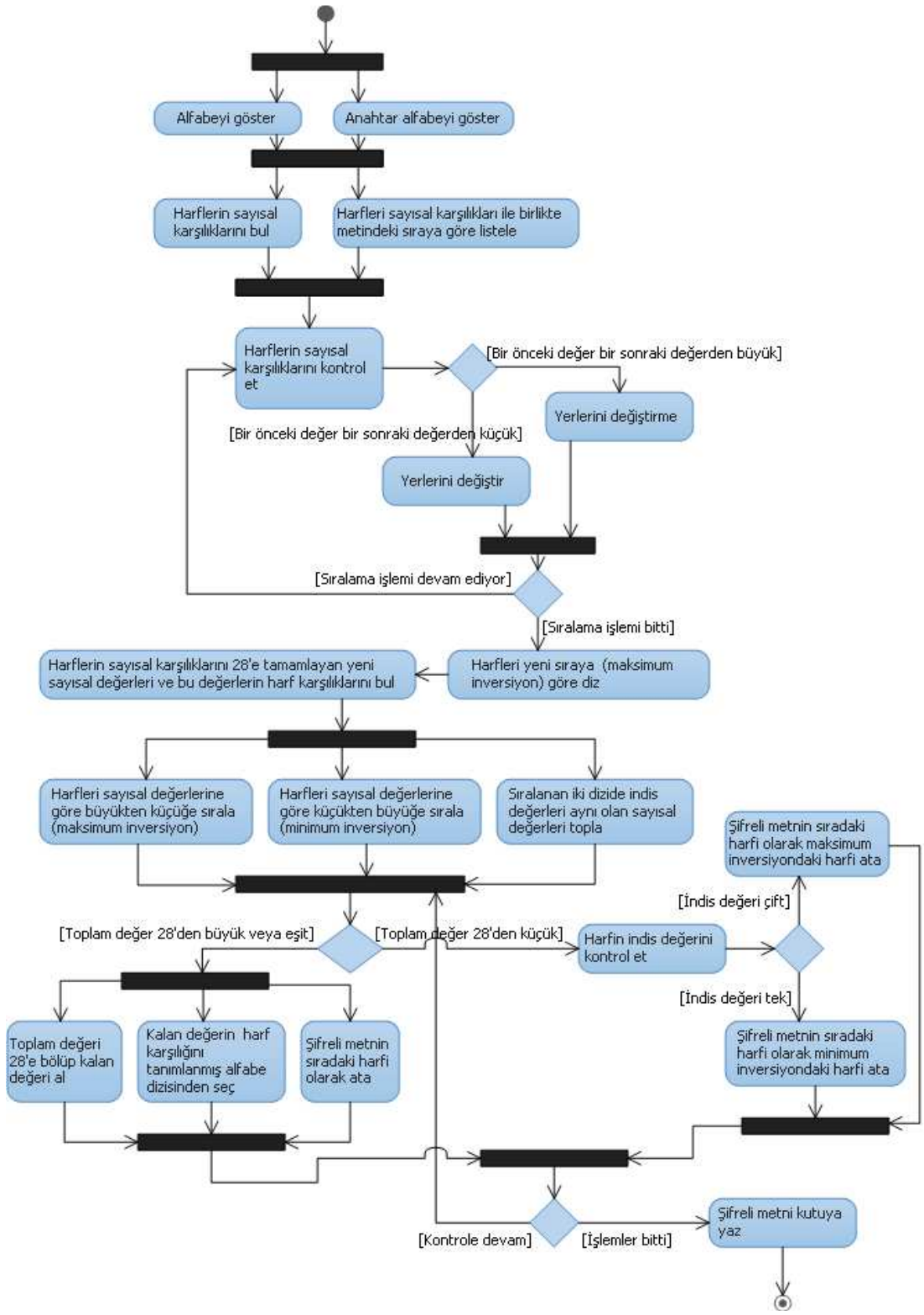
Resim 5.4. Özel karakter ve sayı içeren metinler için şifrele butonu tıklanınca çıkan uyarı mesajı

Resim 5.4’te görülen uyarı mesajındaki *Tamam* butonuna tıklandıktan sonra girişi yapılan metin içerisindeki tüm özel karakterler, sayılar ve Türk alfabesindeki 29 harfin dışındaki farklı harfler temizlenmekte, geriye kalan metin şifrelenmektedir.

Programın ikinci formunda kelime düzeyinde yapılan şifreleme işlemine ait aktivite diyagramı Şekil 5.3’te, paragraf düzeyinde yapılan şifreleme işlemine ait aktivite diyagramı Şekil 5.4’te görülmektedir:

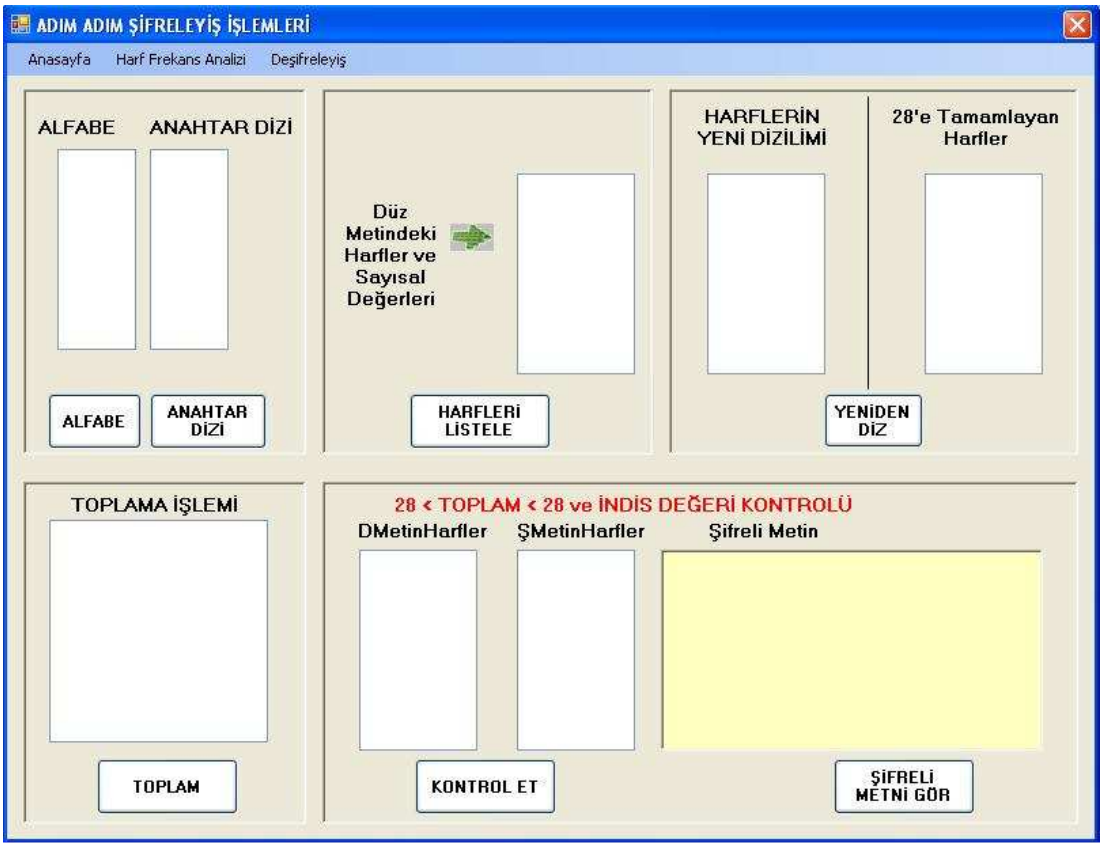


Şekil 5.3. Adım adım şifreleme işlemlerinin yapıldığı sfrack formuna ait aktivite diyagramı(kelime düzeyinde şifreleme işlemi)



Şekil 5.4. Adım adım şifreleme işlemlerinin yapıldığı sfrack formuna ait aktivite diyagramı(paragraf düzeyinde şifreleme işlemi)

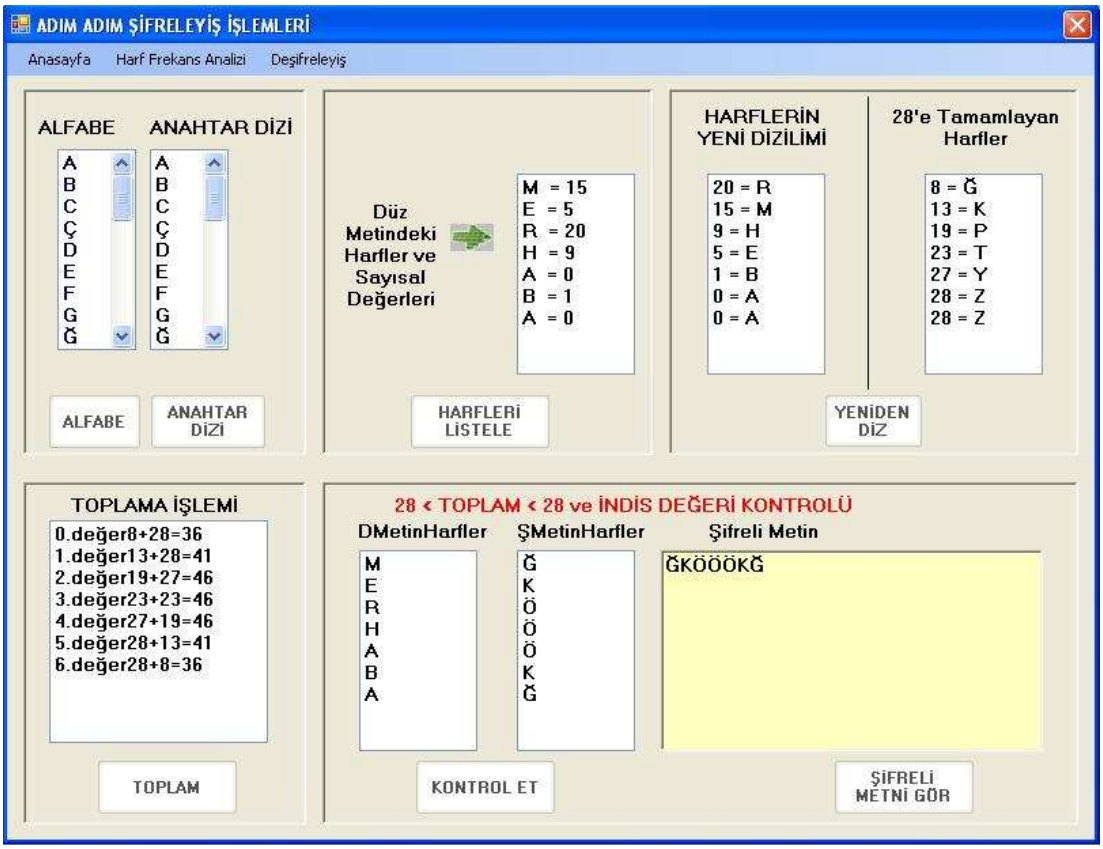
Resim 5.5 ve Rasim 5.6 programın ikinci formuna aittir. Şekil 5.3 ve Şekil 5.4 ile verilen aktivite diyagramlarında anlatılan işlemlerin arayüzlerini temsil etmektedir.



Resim 5.5. Adım adım şifreleyiş işlemlerini gösteren arayüz

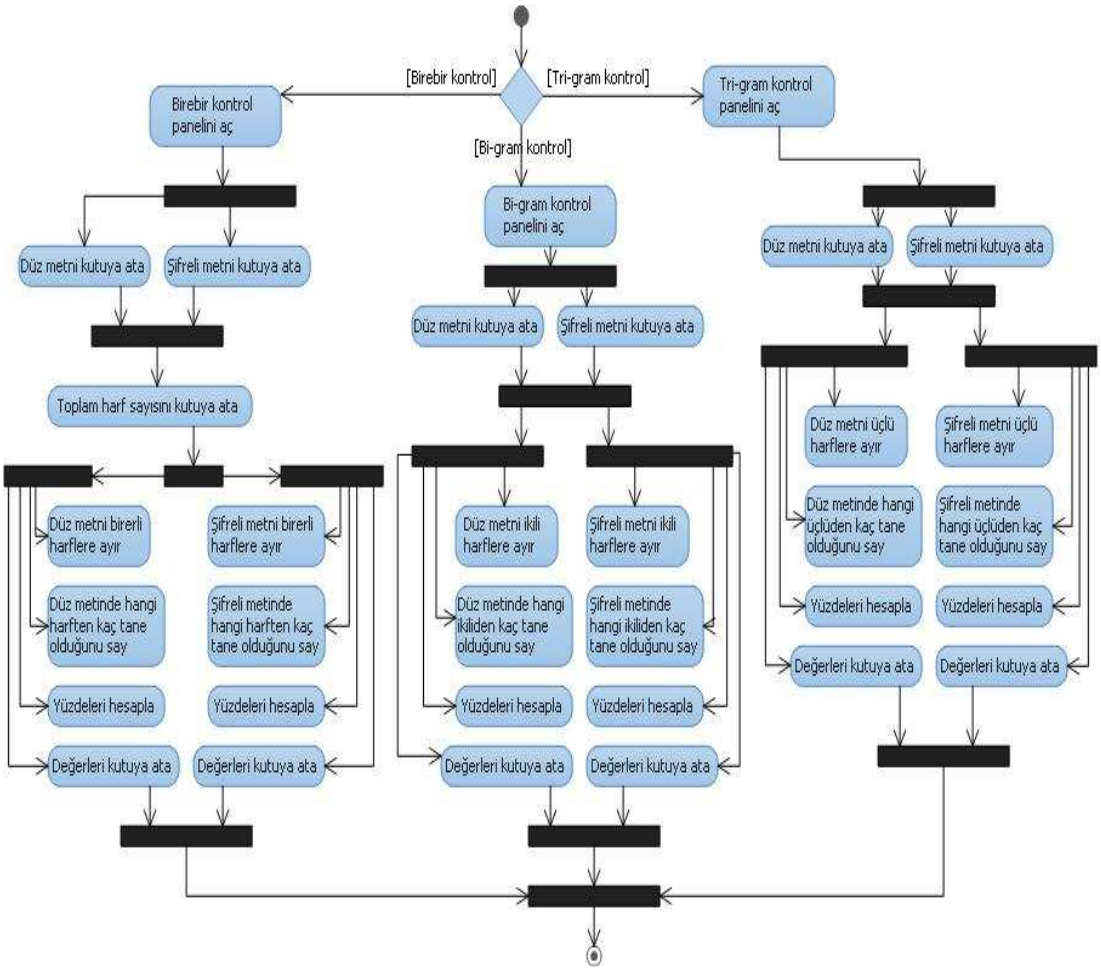
Resim 5.5'te kullanılan windows form *sfrack* olarak adlandırılmıştır. Form üzerinde on iki adet label, altı adet panel, sekiz adet listBox, sekiz adet buton ve bir adet RichTextBox nesnesi kullanılmıştır. ALFABE butonu ile düz metni oluşturan dile ait alfabe(Türkçe alfabe), ANAHTAR DİZİ butonu ile şifreleyiş işlemi için kullanılan alfabe(Türkçe alfabe) atayışları yapılmıştır. HARFLERİ LİSTELE butonu sfr formunda girişi yapılan düz metindeki harfleri ve sayısal karşılıklarını ilgili listboxa aktarmaktadır. YENİDEN DİZ butonu ile harfler maksimum inversiyon sıralamasına göre sıralanmaktadır. HARFLER butonu bir önceki adımda sıralatılan harflerin sayısal karşılıklarının 28'e tamamlayan karakterlerini bulmak için kullanılmıştır. Türkçe'de 29 harf vardır, A harfine

indis değeri olarak sıfır verildiği için Z harfinin indisi 28 kabul edilmiştir. Bu nedenle hesaplamalar 28 sayısına göre yapılmıştır. TOPLAM butonu ile bir önceki adımda hesaplanan harfler tersten ve düzden yazılarak sayısal karşılıkları toplatılmaktadır. KONTROL ET butonu hesaplanmış olan toplam değerlerin modüler aritmetiğe göre ve harflerin kelime içindeki indis değerlerine göre durumlarını değerlendirir. Değerlendirme sonucu bulunan harf değerlerini ilgili listboxlara atar ve şifreleyiş işlemi biter. Elde edilen şifreli metin ŞİFRELİ METNİ GÖR butonu ile richtextboxa atanır. Diğer sayfalara geçiş için menuStrip1 nesnesi kullanılmıştır. MenuStrip1 nesnesine Resim 5.5 ve Resim 5.6'da görülen üç adet ToolStripMenuItem eklenmiştir.



Resim 5.6. Şifreleyiş işlemlerinin adım adım uygulanışı

Programın harf frekans analizlerinin yapıldığı sfrhrf adlı formuna ait aktivite diyagramı Şekil 5.5'te görülebilir:



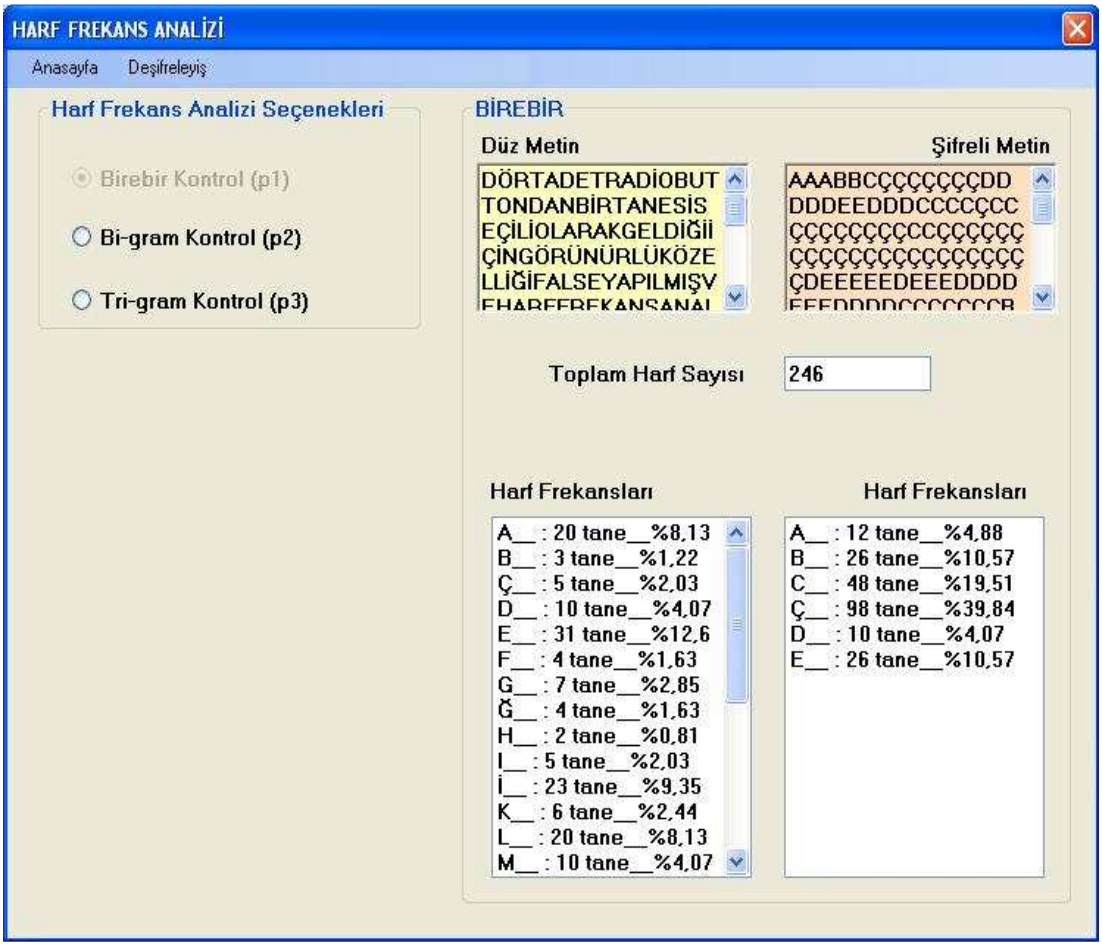
Şekil 5.5. Harf frekans analizi formuna ait aktivite diyagramı

Şekil 5.5'te görülen aktivite diyagramı Resim 5.7, Resim 5.8, Resim 5.9, Resim 5.10' da görülen arayüzlerdeki işlemleri kapsamaktadır.

Harf frekans analizi ile ilgili çalışmaları gösteren ana arayüz Resim 5.7'de görülebilir. Resim 5.8, Resim 5.9 ve Resim 5.10 ana arayüz üzerinde yapılan seçimlere göre dönüt veren arayüzleri göstermektedir.

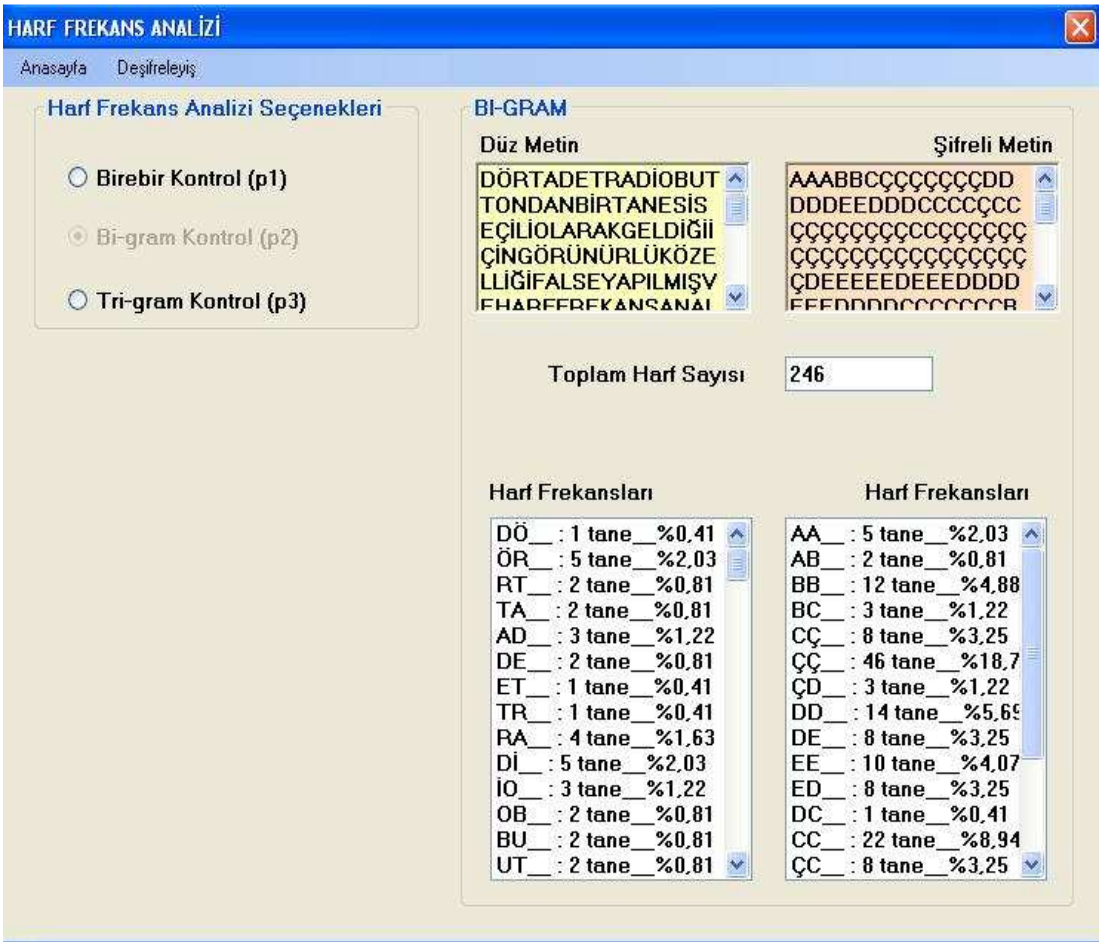
Resim 5.7. Harf frekans analizi giriş arayüzü

Resim 5.7'de görülen arayüzde bir adet groupBox, dört adet radioButton, bir adet menuStrip1 nesneleri kullanılmıştır. MenuStrip1 nesnesine sayfalar arası geçiş için uygun olan ToolStripMenuItem seçenekleri eklenmiştir. Dört adet radioButton'dan bir tanesi seçili olarak geldiği için görünürlük özelliği *false* yapılmış ve harf frekans analizi seçeneklerinin seçilmemiş olarak gelmesi sağlanmıştır. Form üzerinde görülmeyen diğer nesneler radioButton'ların seçilme durumuna göre görünür hale gelmektedir. Bu nesneler; üç adet groupBox, altı adet richTextBox, üç adet textBox, altı adet listBox, on beş adet label nesnesidir. Bu nesnelerin kullanım amaçları Resim 5.8, Resim 5.9 ve Resim 5.10'da sırası ile açıklanacaktır.



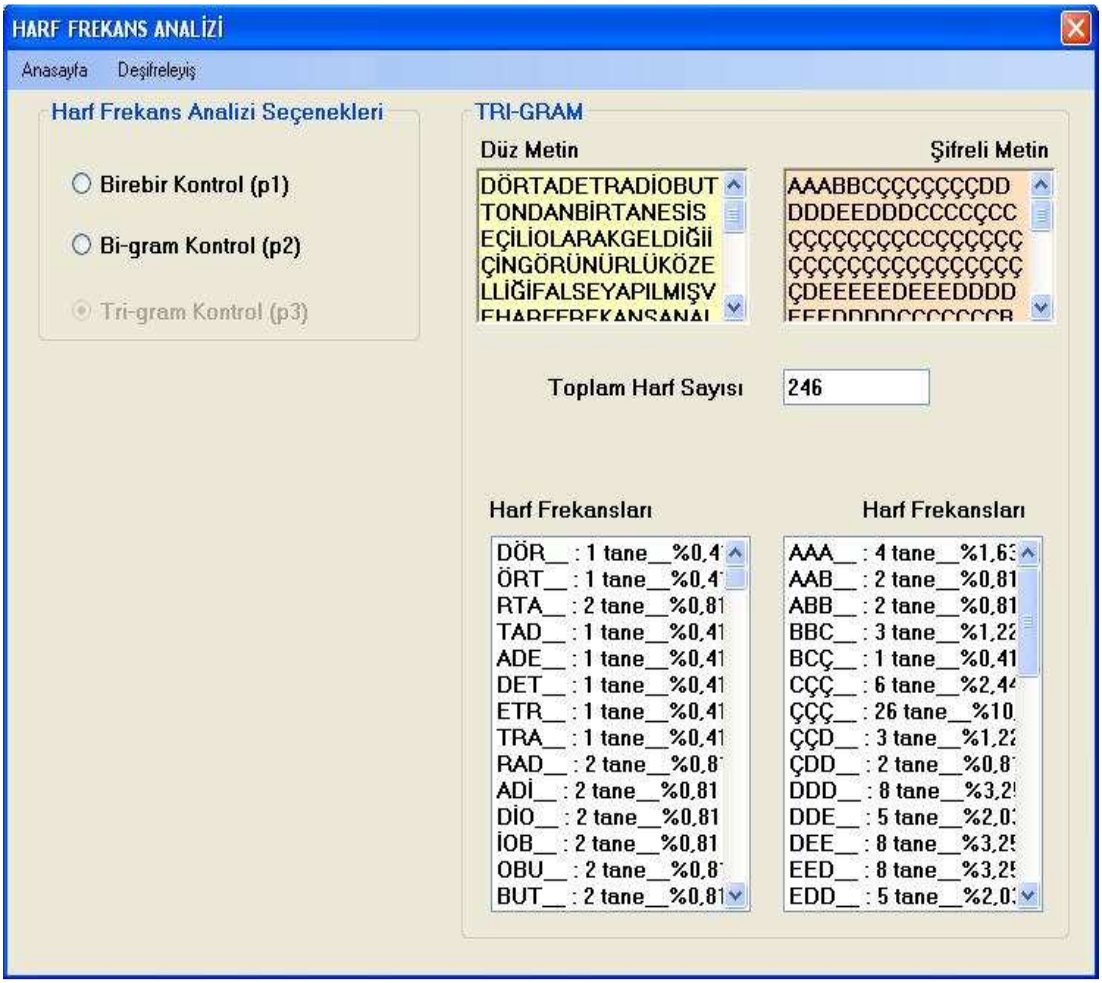
Resim 5.8. Harf frekans analizi birebir kontrol seçeneği arayüzü

Resim 5.8'de görülen BİREBİR başlıklı groupBox, üzerindeki tüm nesneleri tek bir nesne olarak kullanabilmek için kullanılmıştır. Bi-gram ve Tri-gram seçenekleri için de aynı şekilde groupBox kullanılmıştır. Seçilen radioButton'a göre uygun groupBox'ın görüntülenmesi ile istenilen sonuçlar ekrana yansıtılmıştır. Bu kısım için uzun bir metin seçilmiştir, bunun amacı harf frekans değerlerinin yüzdelik olarak daha net elde edilebilmesidir. Herhangi radioButton seçildiğinde uygun işlemler yapılarak elde edilen sonuçlar ilgili listBox'a atanmaktadır. RadioButton'lar arası geçişlerde etkileşim özelliği butonların seçilmiş olma durumuna göre ayarlanmıştır. TextBox ile özel karakter, sayı ve Türkçe alfabede bulunmayan Latin harflerinden temizlenmiş olan boşluk içermeyen metinin karakter sayısı tutulmaktadır.



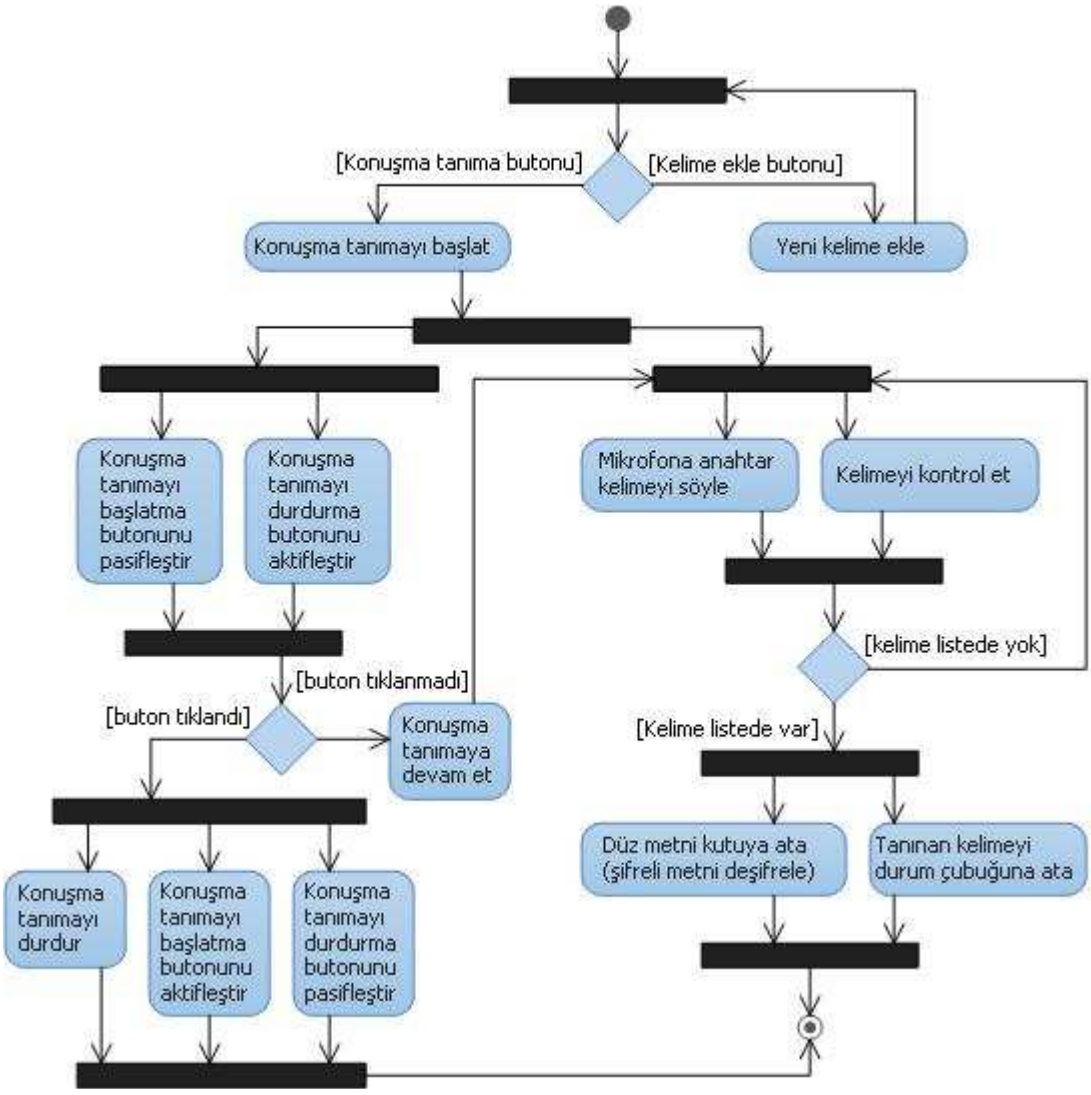
Resim 5.9. Harf frekans analizi bi-gram kontrol seçeneği arayüzü

Resim 5.9'da görülen arayüzde metin içerisinde geçen harf ikililerinin sıklık değerleri hesaplanmaktadır. Bu işlem hem şifreli hem de düz metin için gerçekleştirilmektedir. Bu işlemler yapılırken metindeki harf ikilileri şu şekilde incelenmiştir. Örneğin PARKTA kelimesindeki harf ikilileri, PA_AR_RK_KT_TA şeklindedir. Metin hecelere ayrılmamıştır, bir önceki ve bir sonraki harfler ikişerli olarak alınmıştır. Aynı işlemler harf üçlüleri için uyarlanarak tri-gram seçeneği altında uygulanmıştır. Tri-gram değerleri Resim 5.10'da görülebilir.



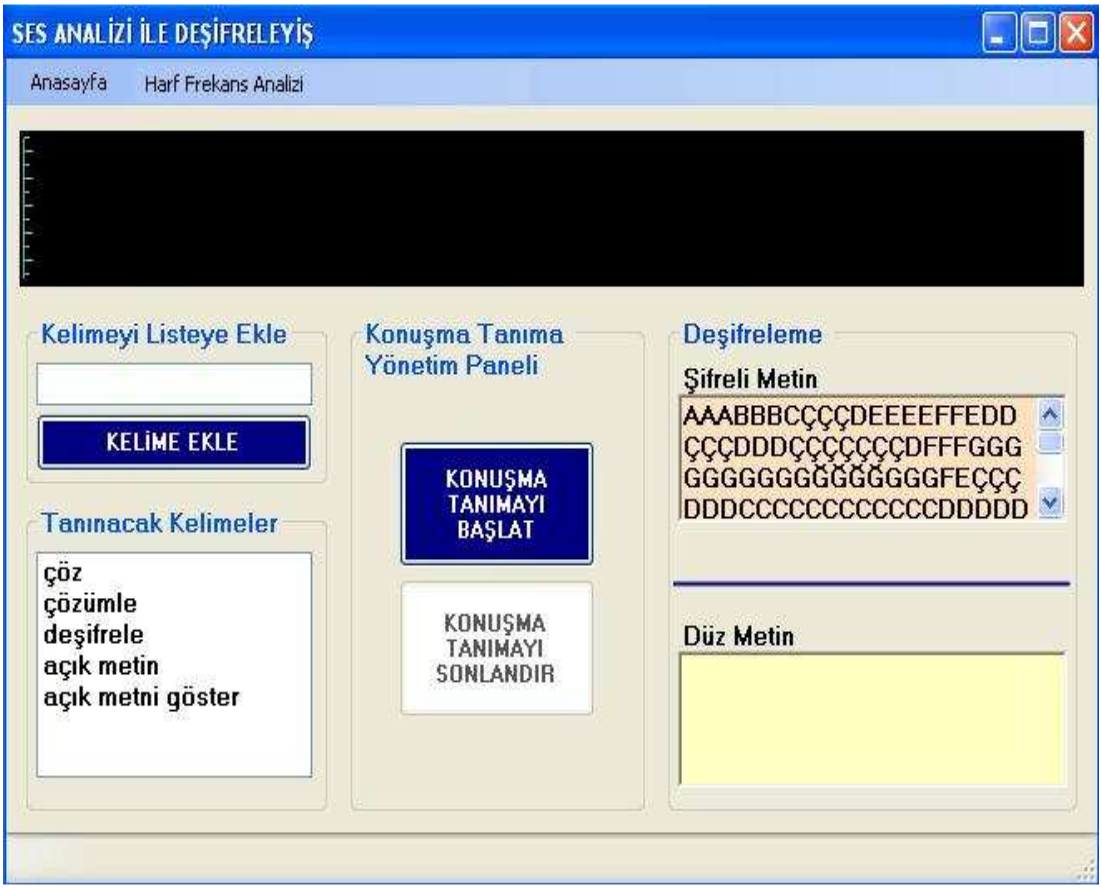
Resim 5.10. Harf frekans analizi tri-gram kontrol seçeneği arayüzü

Şifrelenmiş metnin çözümü için biyometrik özellik olan ses kullanılmıştır. Deşifreleme işlemlerine ait işlemler sfrdsf formunda bulunmaktadır. Sfrdsf formuna ait aktivite diyagramı Şekil 5.5'te görülebilir. Şekli takip eden Resim 5.11, Resim 5.12 ve Resim 5.13 deşifreleme işlemlerini gösteren arayüzlere aittir.



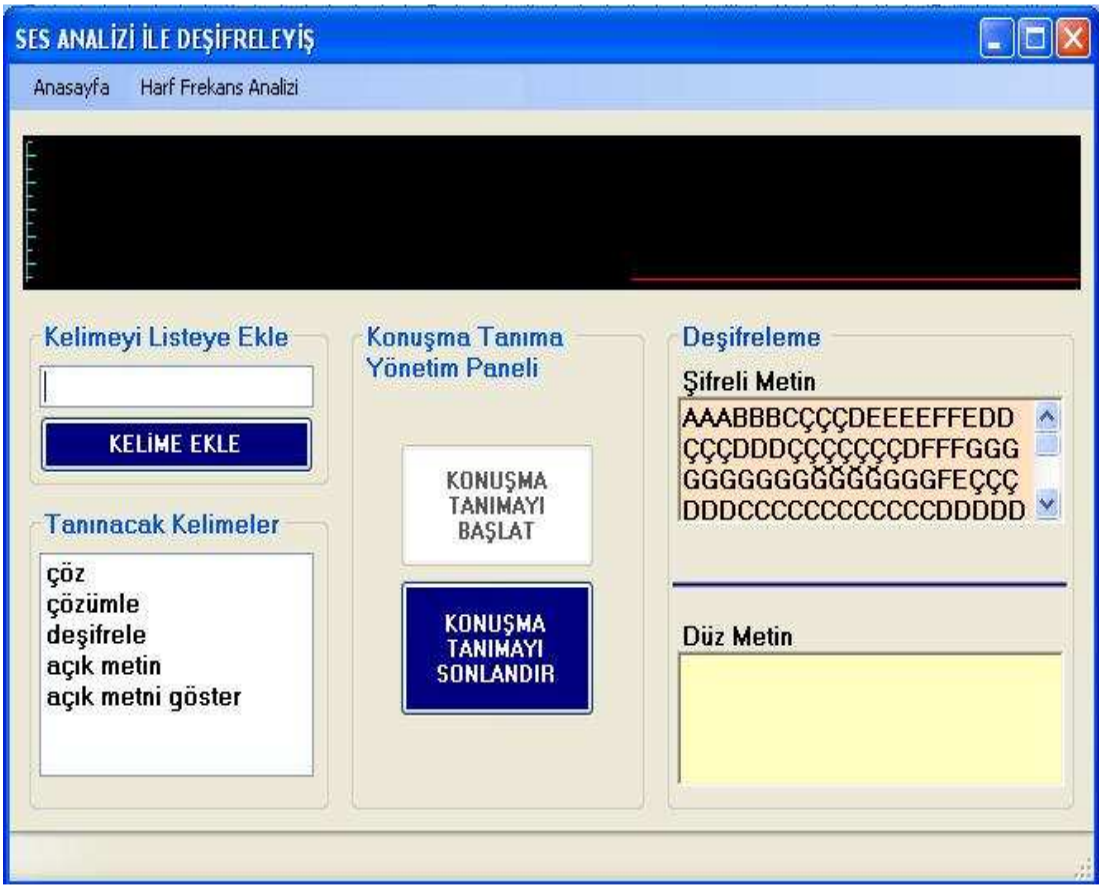
Şekil 5.6. Deşifreleme işlemlerinin yapıldığı forma ait aktivite diyagramı

Diyagramdan anlaşılacağı üzere form üzerinde deşifreleme işlemi için konuşma tanıma işlemleri ve tanınacak kelimelerin listeye eklenmesi olmak üzere iki farklı işlem yapılabilmektedir. Gerekli açıklamalar takip eden resimlerle (Resim 5.11, Resim 5.12, Resim 5.13) birlikte verilecektir.



Resim 5.11. Deşifreleyiş işlemleri arayüzü

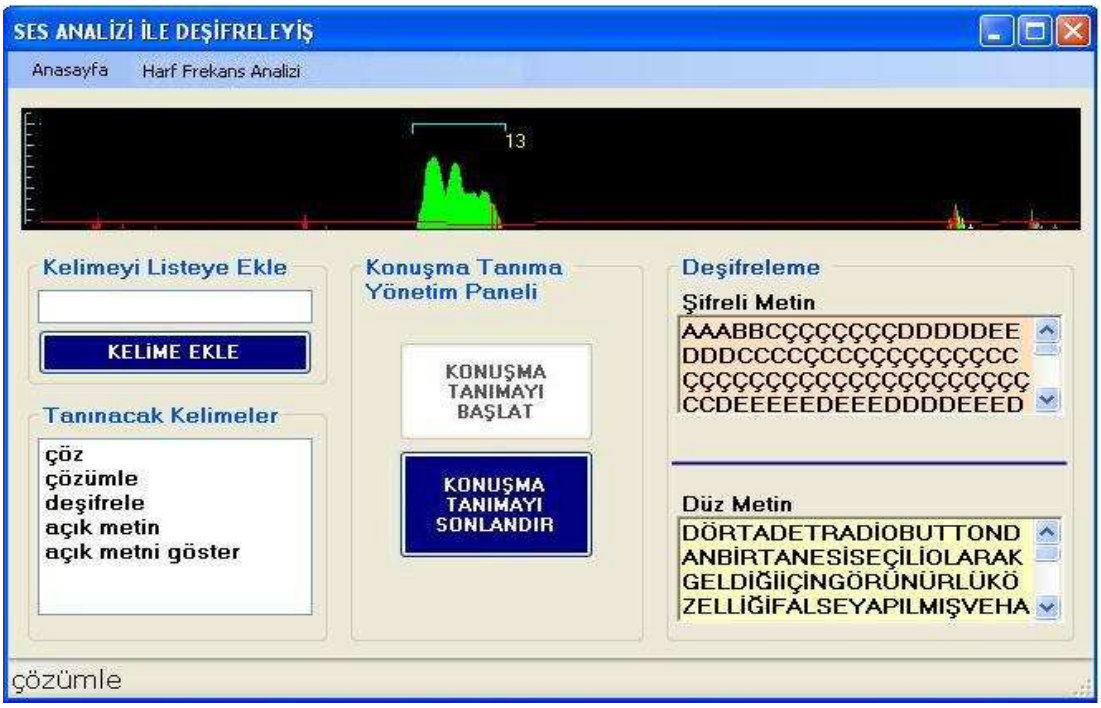
Resim 5.11'de görülen formda dört groupBox, üç Button, iki RichTextBox, bir textBox, bir listBox, bir statusStrip1, bir menuStrip1 nesneleri ile DikteApi demo programına ait ActiveX bileşenlerinden axDikteApiF20X1 ile axDikteApiF20DisplayX1 nesneleri kullanılmıştır. MenuStrip1 nesnesine sayfalar arası geçişi sağlamak için uygun ToolStripMenu'ler eklenmiştir.



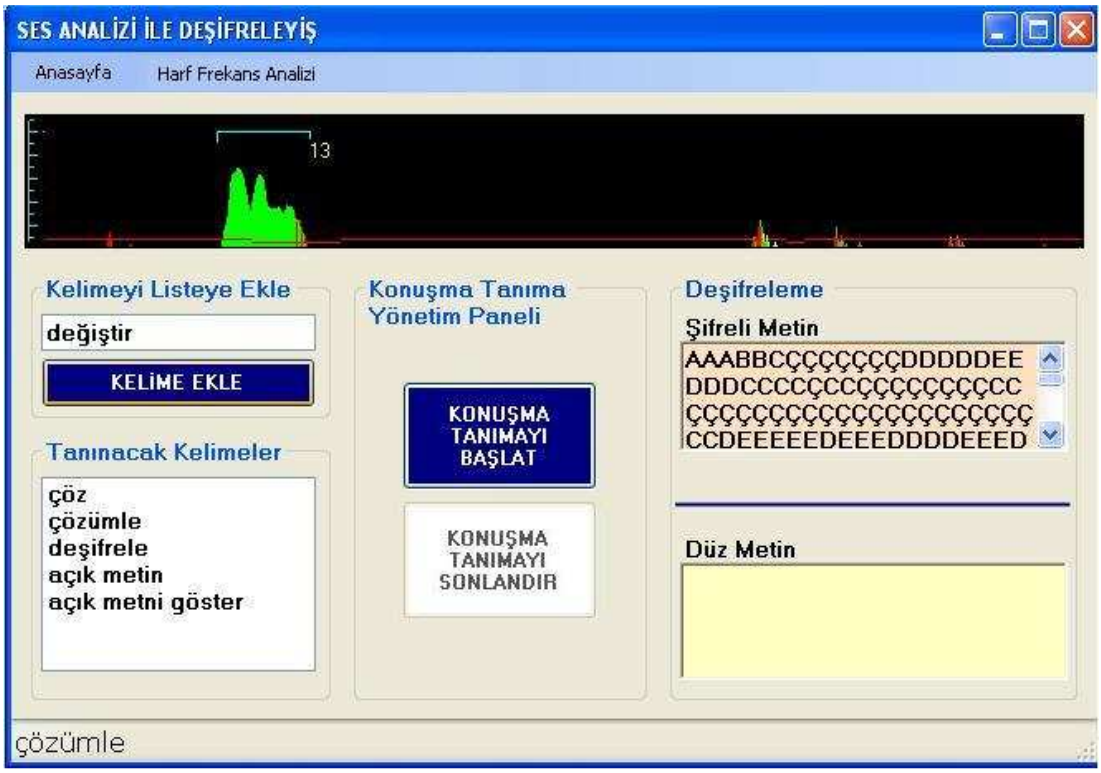
Resim 5.12. Deşifreleyiş arayüzü konuşma tanıma işlemi

KONUŞMA TANIMAYI BAŞLAT butonu ile sesli komutların algılanması için sistem aktif hale getirilmektedir. Sistemin aktif ya da pasif hale getirilmesi için arayüz üzerinde görünmeyen axDikteApiF20X1 nesnesi kullanılmaktadır. Butona tıklanınca bu nesne ile ilgili işlemler yaptırılmaktadır. Mikrofon aracılığıyla algılanan sesli komut, axDikteApiF20DisplayX1 nesnesi üzerinde ses dalgaları oluşturmaktadır. Tanınan sesli komut, en alttaki statusStrip1 nesnesine aktarılmaktadır. KONUŞMA TANIMAYI SONLANDIR butonu ile aktif hale getirilen konuşma tanıma sistemi pasif hale getirilmektedir.

Resim 5.13'te mikrofon aracılığı ile sesli olarak verilen komutun uygulanışı görülmektedir. Komutun algılanışı ile şifreli metin çözümlenmekte ve Düz Metin başlıklı richtextboxa metnin açık hali aktarılmaktadır.



Resim 5.13. Konuşma tanımanın uygulanışı



Resim 5.14.Tanınacak kelimeler listesine yeni kelime eklenişi

Resim 5.14’te görülen textBox ve listBox, şifrelenmiş metini çözmek için kullanılabilecek sesli komutların eklenmesi için kullanılmaktadır. listBox içinde görülen beş komut sistemde sabit olarak atanmıştır, sesli komutların tanınması için kullanılan DikteApi demo programının yirmi kelimelik kapasitesi bulunmaktadır. textBox aracılığı ile on beş adet sesli komut tanınacak kelimeler listesine eklenebilir.

6. HARF FREKANS ANALİZİNDEN ELDE EDİLEN BULGULAR

Şifrelenmiş metin üzerinde harf frekans analizinden sağlıklı veriler elde edebilmek için uzun metinler üzerinde çalışmak gereklidir. Türk dil yapısına göre harflerin frekans değerleri daha önce yapılan çalışmalarda tespit edilmiştir. Bu nedenle yapılan çalışmada bu bilgiler doğrultusunda küçük bir örnek üzerinde düz metin ile şifreli metin arasında frekans analiz değerlendirmesi yapılmıştır. Harf frekans analizi ile ilgili genel bilgiler Bölüm 3.6.1'de anlatılmıştır.

Harf frekans analizinde bilinmesi gereken en önemli bilgi her dilin kendi özelliklerine göre harf kullanım dağılımı olduğu bilgisidir. Türkçe'de de bazı harflerin daha çok kullanıldığı daha önce yapılan çalışmalarda tespit edilmiştir. Bu çalışmalara göre Türkçe'de en çok görülen harf A dır. Onu sırası ile E, İ, N ve R izlemektedir. Bu veriler Bölüm 3'te bulunan Çizelge 3.2'de görülebilir.

Aşağıda verilen örnekte A harfi için harf frekans hesaplaması görülmektedir:

Örnek: Ayşe pazara gitti ve elma aldı.

Toplam harf sayısı: 25

A harfinin sayısı : 6

A harfinin cümledeki frekansı : 6/25 = 0,24

Frekans analizinin uzun metinler incelenerek yapılması elde edilen sonuçların birbiri ile tutarlı olmasını sağlar. Ayrıca birebir eşleme dışında harf ikililerine, üçlülerine de bakılabilir. Yan yana gelen harf sıralılarının frekans değerlerine göre, şifreli metinde en çok yan yana gelen sıralıların yerine yerleştirilmesiyle de çözümleme yapılabilir.

Bu çalışmada birebir karşılaştırma yapılmıştır, harf ikilileri ve üçlülerinin frekansları hesaplanmış ancak düz ve şifreli metin üzerinde bu karşılaştırmalar yapılmamıştır. Ayrıca Türkçe için harf frekans analizi

çalışmaları daha önceden yapılmış bazı çalışmalarda bulunduğu için dil üzerinde yeni bir araştırma yapılmamış ve frekans analizi için varolan veriler kullanılmıştır.

Çizelge 6.1'de küçük bir metin üzerinde yapılan incelemeler görülmektedir:

Düz metin : "Bilişim teknolojilerinin bizlere sağladığı kolaylıklar arttıkça, elektronik ortamların kullanımı yaygınlaşmakta, bilginin işlendiği, taşındığı, saklandığı ortamlara erişimler zamandan, mekandan bağımsız hale gelmektedir." [3]

İlk önce metindeki harflerin sayısı bulunmalıdır:

Toplam harf sayısı: 194

Çizelge 6.1. Düz metindeki harf dağılımları

Harf	Harf Sayısı	Kullanım%'si		Harf	Harf Sayısı	Kullanım %'si
a	26	13,40		m	10	5,15
b	4	2,06		n	16	8,25
c	___	___		o	6	3,09
ç	1	0,52		ö	___	___
d	7	3,61		p	___	___
e	14	7,22		r	12	6,19
f	___	___		s	3	1,55
g	3	1,55		ş	5	2,58
ğ	6	3,09		t	9	4,64
h	1	0,52		u	1	0,52
ı	15	7,73		ü	___	___
i	17	8,76		v	___	___
j	1	0,52		y	3	1,55
k	11	5,67		z	3	1,55
l	20	10,31				

Çizelge 6.1'de görülen frekans değerleri hesaplanırken harfin metindeki sayısı metindeki toplam harf sayısına bölünmüş ve sonuç yüz ile çarpılmış ve Çizelge 6.1'de yüzdelik değerler gösterilmiştir. Çizelge 6.1'de görüldüğü üzere örnek metinde en çok kullanılan harfler; a, l, e, i, ı, k, n, r, m harfleridir. Örnek olarak kullanılan düz metindeki harflerin frekans analizine göre en çok kullanılandan en az kullanılana doğru sıralaması;

A, L, İ, N, I, E, R, K, M, T, D, Ğ=O, Ş, B, G=S=Y=Z, Ç=H=J=U

şeklindedir.

Kelime kelime şifrelenmiş şifreli metin

ECÇFÇCE AÇAABANAAMABAAÇA ZÇHFHÇB GLLIJILLG BĞBBBABBBĞB EEEEEEEE AÇAJBBKAÇA EĞRİNKLÖĞE DCÇBABÇCD BBEFEÇBBÇEFEBB İEÇFFÇEİ CGFFFFFGC ECDĞĞĞDCE GJIGĞĞGIJG EĞĞİMLĞĞE BÇRCACİÇB AJJHHJJA JJHIIHJJ AFEĞĞEFA LLLL BÇĞHĞFĞHĞÇB

Yukarıda görülen şifreli metin her kelimenin tek tek şifrelenmesi sonucu elde edilen sonuçların yan yana yazılması ile elde edilmiştir. Metin paragraf olarak şifrelenmemiştir. Şifreli metine ait harf frekans analizi ile ilgili bulgular aşağıda görülen Çizelge 6.2'de yer almaktadır:

Toplam Harf Sayısı : 194

Çizelge 6.2. Kelime kelime şifrelenmiş şifreli metindeki harf dağılımları

Harf	Harf Sayısı	Kullanım%'si		Harf	Harf Sayısı	Kullanım %'si
a	21	10,82		M	2	1,03
b	25	12,89		N	2	1,03
c	10	5,15		O	___	___
ç	18	9,28		Ö	1	0,52
d	4	2,06		P	___	___
e	24	12,37		R	2	1,03
f	14	7,22		S	___	___
g	8	4,12		Ş	___	___
ğ	19	9,79		T	___	___
h	8	4,12		U	___	___
ı	6	3,09		Ü	___	___
i	5	2,58		V	___	___
j	12	6,19		Y	___	___
k	2	1,03		Z	1	0,52
l	10	5,15				

Şifreli metindeki harflerin frekans analizine göre en çok kullanılandan en az kullanılana doğru sıralaması;

B, E, A, Ğ, Ç, F, J, C=L, G=H, I, İ, D, K=M=N=R, Ö=Z

şeklindedir.

Eğer düz metindeki kelimeler ve şifrelenmiş halleri incelenirse bir harfin birden fazla harf ile şifrelendiği görülebilir. Örneğin düz metindeki “B” harfi şifrelenmiş metin içerisinde aynı anda “E,Z,İ,A” harfleri ile şifrelenmiştir. Bu durum harf frekans analizi ile kriptografi algoritmasının çözülmesini zorlaştırmaktadır.

Aynı zamanda düz metindeki kelimelerin şifreli halleri incelendiğinde düz metindeki birden fazla harf şifreli metinde tek bir harfle temsil edilmektedir. Örneğin HALE kelimesinin şifreli hali LLLL olarak bulunmuştur. Bu durum harf frekans analizinde elde edilen bulgularda yanılma payının oldukça yüksek olması ihtimalini doğurmaktadır.

Şifreli metnin harf frekans analizine karşı güvenirliği incelenerek geliştirilen algoritmanın güvenirliği bu kısımda tespit edilmeye çalışılmıştır.

Türkçe’nin yapısında en çok kullanılan harf “A” olduğu ve şifreli metinde de en çok geçen harf “B” olduğuna göre şifreli metindeki B harfleri yerine A harfi yazılmalıdır.

Şifreli metindeki “E” harfi yerine “L” harfi, “A” harfi yerine “İ” harfi yazılmalıdır.

Şifreli metindeki “C” ve “L” harflerinin dağılımları aynı oranda çıktığı için deneme yanılma yapılmalıdır. Öncelikle “C” harfi yerine “K” yazılmalı eğer anlamsız olduysa “L” harfi yerine “K” yazılmalıdır. K ve M harflerinin oranları birbirine çok yakın olduğu için şifreli metinde C ya da L harfi yerine M harfi de denenebilir.

Aşağıda görülen metinde, altı farklı şekillerde çizili olan bölümler, şifreli metini çözmek için yapılan bazı harf değişikliklerini göstermektedir:

LCÇFÇCL İÇİİAİNİİMİAİİÇİ ZÇHFHÇA GLLIJILLG AĞAAAİAAAĞA LLLLLLLL İÇİJAAKİÇİ LĞRİNKLÖĞL DCÇAİAÇCD AALFLÇAAÇLFLAA İLÇFFÇLİ CGFFFFFGC LCDĞĞĞDCL GJIGĞĞGIJG LĞĞİMLĞĞL AÇRCİCİÇA İJJHHJJİ JJHIIHJJ İFLĞĞLFİ LLLL AÇĞHĞFĞHĞÇA

Bu metinde aynı harfin pek çok kez arka arkaya gelmiş olması bu metin üzerinde harf frekans analizinin etkili olmayacağını göstermektedir, çünkü Türkçe'nin yapısal özelliklerine göre en çok tekrarlanan harf *A* olmasına rağmen şifreli metinde en çok kullanılan harf(B) yerine *A* yazıldığında birbirinden farklı birçok harf *A* ile temsil edilmiş olur ve bu tekrarların çözülmesi harf frekans analizi ile mümkün değildir. *A* harfi birden fazla harfi temsil etmiştir. Şifreli metindeki tek bir harfin birden fazla harfi temsil etmesi ve düz metindeki bir harfin şifreli metinde birden fazla harfle temsil edilmesi geliştirilen kriptografi algoritmasının en önemli avantajıdır.

Paragraf olarak şifrelenmiş şifreli metin:

AAABBBDEEEEEEEEEFFFFFGGGĞĞGGGGEDDDDDGGFFFFGGGGGG
GGGGEEEDDDEEEDÇÇÇÇÇDDDDDDDDDÇÇÇÇÇÇÇÇÇDDDDDDÇ
CCCCÇDDDDDDÇÇÇÇÇÇÇÇÇDDDDDDDDDÇÇÇÇÇDEEEDDDEEE
GGGGGGGGGGFFFFGGDDDDDEGGGGĞĞGGGFFFFFEEEEEEEEEDBB
BAAA

Paragraf olarak şifrelenmiş metine ait frekans değerleri Çizelge 6.3'te verilmiştir:

Çizelge 6.3. Paragraf olarak şifrelenmiş şifreli metindeki harf dağılımları

Harf	Harf Sayısı	Kullanım%'si
a	6	3,09
b	6	3,09
c	4	2,06
ç	36	18,56
d	7	3,61
e	32	16,49
f	18	9,28
g	38	19,59
ğ	4	2,06

Çizelge 6.2 ve Çizelge 6.3 incelendiğinde geliştirilen algoritma ile paragraf olarak şifreleme yapılarak elde edilen şifreli metinin, kelime kelime şifreleme yapılarak elde edilen şifreli metine göre daha karmaşık bir yapıya sahip olduğu görülmüştür. Paragraf olarak şifreleme işlemi yapıldığında şifreli metindeki harf çeşitliliği azalmaktadır. Bu durum harf frekans analizi saldırısını olumsuz yönde etkilerken geliştirilen algoritmanın güvenirliğini olumlu yönde etkilemektedir.

7.GELİŞTİRİLEN ALGORİTMA VE DES ALGORİTMASININ PERFORMANS ANALİZLERİNİN KARŞILAŞTIRILMASI

Performans analizi algoritmanın verimliliği ve kullanışlılığını tespit etmek için yapılır. Bir algoritmanın performans analizi ile algoritmanın bellekte kapladığı alan, çalışma zamanı gibi özellikler ölçülebilir [21].

Performans analizi iç ve dış faktörlere bağlı olarak değişkenlik gösterebilir. İç ve dış faktörler Çizelge 7.1'de görülmektedir [66]:

Çizelge 7.1. Performans analizini etkileyen faktörler

İç Faktörler	**Dış Faktörler**
Çalıştırmak için gereken zaman	Girdi verisinin büyüklüğü
Çalıştırmak için gereken bellek alanı	Bilgisayarın hızı
	Derleyicinin kalitesi

Şifreleme algoritmalarının performans ölçütleri genel olarak aşağıda belirtildiği gibidir [21, 53, 67, 68]:

- Şifreleme sistemin kırılabilme süresinin uzunluğu,
- Şifreleme ve çözme işlemlerine harcanan zaman (Zaman Karmaşıklığı)
- Şifreleme ve çözme işleminde ihtiyaç duyulan bellek miktarı (Bellek Karmaşıklığı)
- Bu algoritmaya dayalı şifreleme uygulamalarının esnekliği
- Bu uygulamaların dağıtımındaki kolaylık ya da algoritmaların standart hale getirilebilmesi
- Algoritmanın kurulacak sisteme uygunluğu

Performans testleri, sistem çıktılarının belirlenen ve kabul edilebilecek olan zaman dilimi içerisinde, üretilebildiğini değerlendirebilmek için gerçekleştirilen

testlerdir. Performans testi, yapılan işlem zamanlarının hesaplanması ve sistemin dar boğazlarının tespit edilmesini sağlar [69].

Algoritmanın özellikle işlem zamanı boyutu için yapılan analizin sağlıklı sonuç verebilmesi ilgili algoritmanın en kötü durumu için performans ölçümü yapılmasını gereklidir. Bu şekilde o algoritmanın gerçek performansı büyük ölçüde tespit edilmiş olur [70].

Performans ölçütlerinden bir diğeri algoritma çalışırken ihtiyaç duyulan bellek gereksinimidir. Bellekte değişkenler, ara sonuçlar ve sonuçlar, bir takım sabitler ve algoritma bilgileri yer almaktadır. İşlem esnasında maksimum bellek kullanımı tespit edilerek ilgili algoritmanın ihtiyaç duyduğu bellek miktarı değeri elde edilir. Performansı yüksek bir algoritmanın bellek kullanımı az olmalıdır. İdeal bir algoritma hem hızlı olmalı hem de az bellek harcamalıdır [70].

Tez çalışmasında, performans analizleri Windows XP SP3 işletim sistemine sahip Dell Inspiron 6000 dizüstü bilgisayar kullanılarak yapılmıştır.

Geliştirilen algoritmanın performans analizini gerçekleştirmek için Microsoft Visual Studio 2010 C# ortamı kullanılmıştır. Algoritmanın şifreleme işlemi için harcadığı zaman, hafıza ve işlemci kullanımları incelenmiştir.

Öncelikle Build ya da Analyze menüsünden Run Code Analysis seçeneği kriptografi projesi üzerinde çalıştırılmıştır. Daha verimli sonuçlar elde edebilmek için programın verdiği uyarılar tespit edilmiş ve bir kısmı düzeltilebilmiştir.

Analyze menüsündeki Launch Performans Wizard seçeneği kullanılarak programın performans değerleri tespit edilmiştir. Bu bölümde ekrana dört farklı seçenek gelmektedir. CPU Sampling seçeneği ile, CPU üzerindeki işlem yükü tespit edilmiştir. Instrumentation seçeneği ile harcanan zaman ve çağrılan fonksiyon sayısı tespit edilebilir. Bu bölümde işlem süreleri

incelenmiştir. .NET Memory Allocation seçeneği ile bellek kullanımı tespit edilmiştir. Concurrency seçeneği thread uygulamaları için kullanılabilir. Geliştirilen demoda thread uygulaması bulunmadığı için bu bölümle ilgili herhangi inceleme yapılmamıştır.

7.1. Geliştirilen Algoritmaya Ait Performans Değerleri

Geliştirilen algoritmaya ait performans değerleri hem paragraf hem de kelime düzeyinde incelenmiştir. *CPU kullanımı*, *bellek kullanımı* ve işlemler için geçen *işlem zamanı* açısından performans analizleri yapılmıştır.

Performans analizi, farklı uzunluklardaki metinler için yapılmış olup metinin karakter sayısının performans değerlerini etkilediği görülmüştür. Bu nedenle tez çalışmasında 8 byte (64 bit) veri üzerinde yapılan analiz sonuçları sunulmuştur.

Performans analizi uygulamalarında *Kalemlik* kelimesi kullanılmıştır. 64bitlik bir veri olduğu için örnek olması amacıyla seçilmiştir. Seçilmesinde herhangi bir özel durum söz konusu değildir.

Yapılan incelemeler sadece şifreleme işlemi için sunulmuştur.

7.1.1. Paragraf düzeyinde performans analizi değerleri

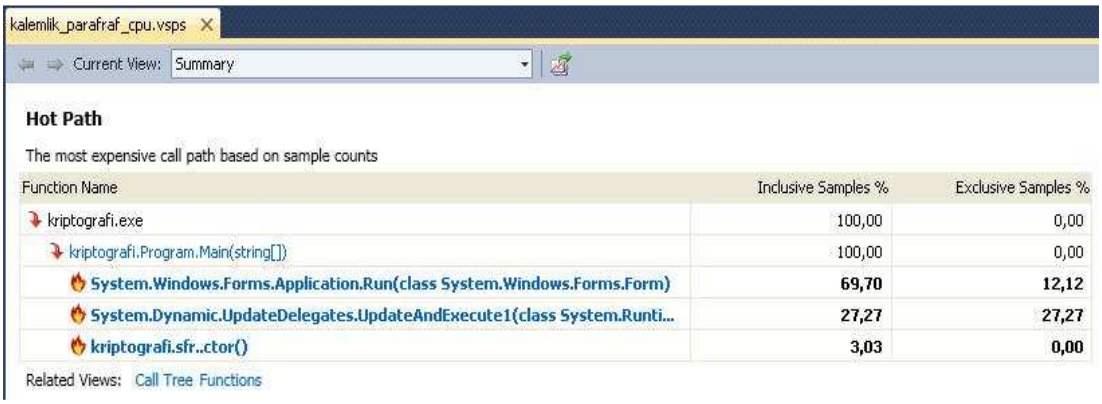
CPU sampling verileri

Resim 7.1’de CPU Sampling seçeneğine ait sonuçlar görülmektedir:

Resim 7.1. “kalemlik” (8 byte) kelimesinin paragraf düzeyinde şifrelenmesine ait CPU grafiği

Toplam 33 (sınıf, fonksiyon) için CPU kullanımı Resim 7.1’deki grafikte görülmektedir. CPU kullanımı program ilk çalıştırıldığında %100’e ulaşmakta, daha sonra şifreleme işlemlerinde kullanılan fonksiyon ve sınıflara göre (metin kutularının içeriğinin değişmesi, butonlara tıklanması, döngülerin çalıştırılması gibi) değişkenlik göstermektedir.

Resim 7.2’de sistemi en çok zorlayan sınıflar ve sistem darboğazı oluşturan fonksiyonlar görülmektedir:

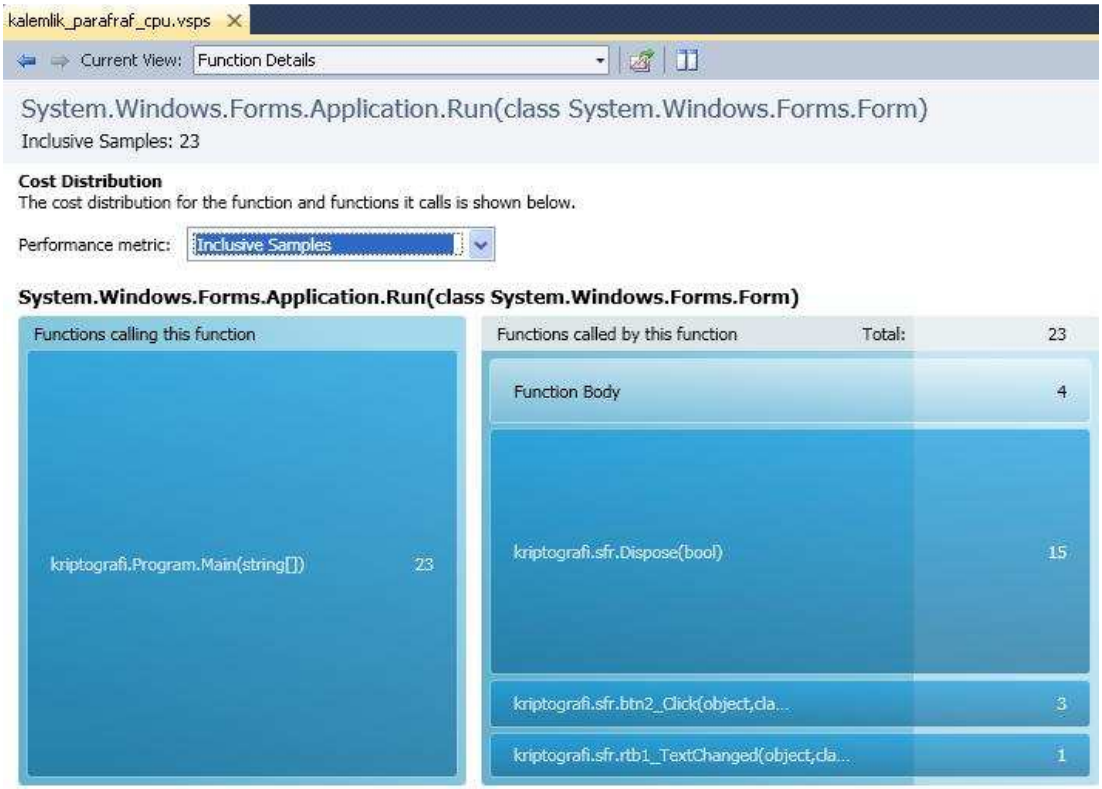


Resim 7.2.“kalemlik” kelimesinin paragraf düzeyinde şifrelenmesinde sistemi zorlayan sınıflar

Current View kısmında farklı seçenekler bulunmaktadır. *Summary*, CPU ile ilgili kullanım bilgilerinin özet şeklinde ve en genel haliyle görülmesini sağlar.

Hot Path bölümünde *Inclusive Samples* yüzdesi yüksek olan sınıflar sistemi en çok zorlayan sınıflardır. *Exclusive Samples* bölümünde ise performans darboğazı yaratan fonksiyonlar en yüksek yüzdeye sahiptir [68]. Bu fonksiyonları daha detaylı görebilmek için yanlarında ateş simgesi bulunan yazıların üzerine tıklamak yeterlidir ya da *Current View* kısmında *Function Details* seçeneği ile detaylara ulaşmak mümkündür.

Resim 7.3'te Application Run sınıfına ait fonksiyonlar görülmektedir:

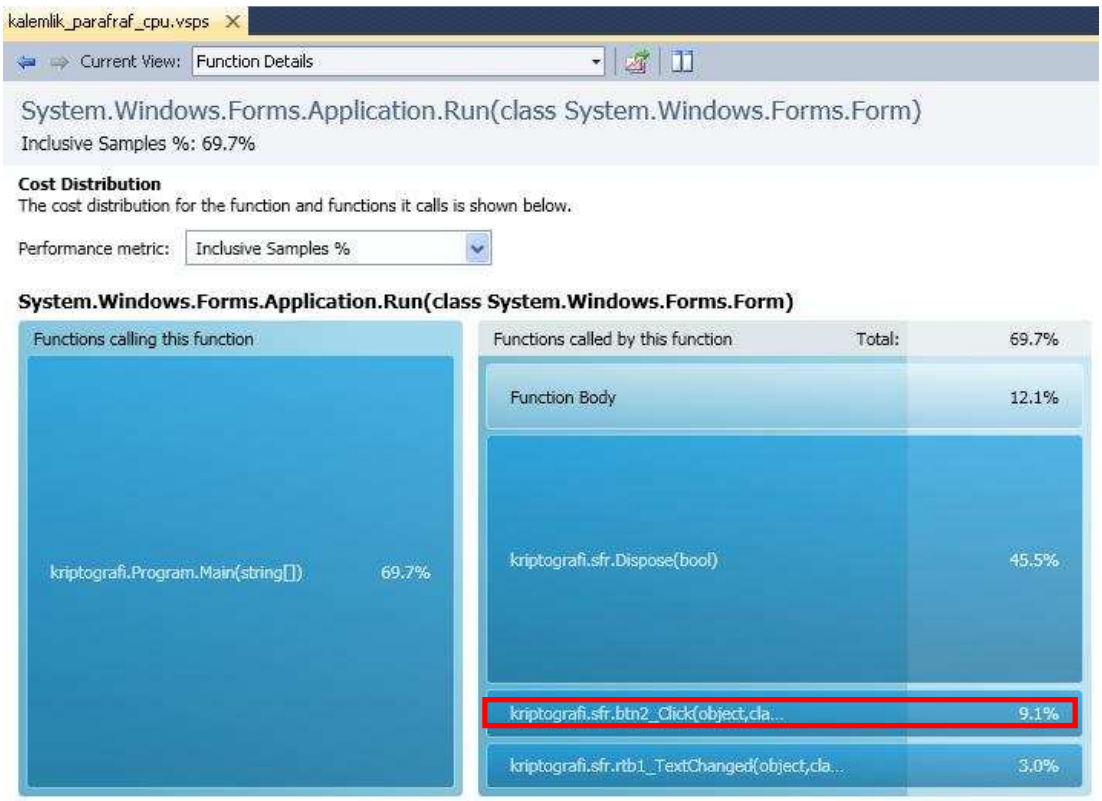


Resim 7.3. Application Run sınıfında darboğaz oluşturan fonksiyonlar

Resim 7.3'te görüldüğü üzere Application Run sınıfında toplam 23 fonksiyon çağırılmaktadır. Rtb1 nesnesi düz metinin girişini sağlayan arayüz elemanıdır. Btn2 nesnesi paragraf olarak şifreleme işlemini yapan butondur. Bu nesnelere ait çağırılan fonksiyon sayıları Resim 7.3'te görülmektedir. Sağ

taraftaki her bir kutuya tıklandığında detaylı bilgilere ulaşılmaktadır. Örnek oluşturması açısından Application Run sınıfına ait verilerin detaylı görüntüleri Resim 7.4 ile verilmiştir.

Çağırılan fonksiyonların %'lik dilimleri Resim 7.4'te görülmektedir. Bu kısma ulaşmak için *Performance metric* kutusundan *Inclusive Samples %* seçimini yapmak yeterlidir.



Resim 7.4. Application Run sınıfında darboğaz oluşturan fonksiyonların %'leri

Resim 7.4'te görülen değerlere göre sistemde en büyük darboğazı oluşturan fonksiyon Dispose metodudur. Resim 7.4'te çerçeve ile gösterilen *btn2_Click* olayı şifreleme algoritmasının çalıştırılmasını sağlayan olaydır. Şifreleme algoritmasının CPU kullanımı, geliştirilen demo programın toplam CPU kullanımının %9,1'ini oluşturmaktadır.

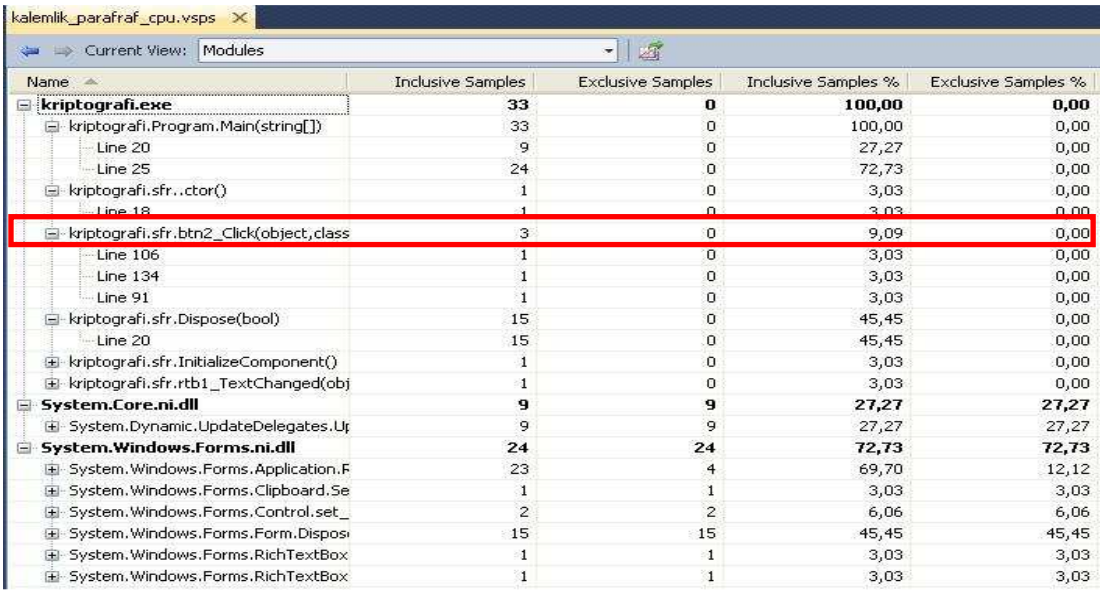
Application Run metoduna ait toplam değerlerin genel özeti Resim 7.5'te verilmiştir:

Function Performance Details

Metric	Exclusive	In Calls	Inclusive Total
Collected Samples	4 (12.1%)	19 (57.6%)	23 (69.7%)

Resim 7.5. Application Run sınıfına ait genel değerler

Current View kısmında *Modules* seçeneği seçilirse sistemi zorlayan sınıfların hangileri olduğu ve fonksiyonların hangi satırlarda yer aldığı bilgisine ulaşılabilir. Resim 7.6'da Modules seçeneği ile daha detaylı olarak elde edilen bilgiler görülmektedir:


kalemlik_parafraf_cpu.vsps

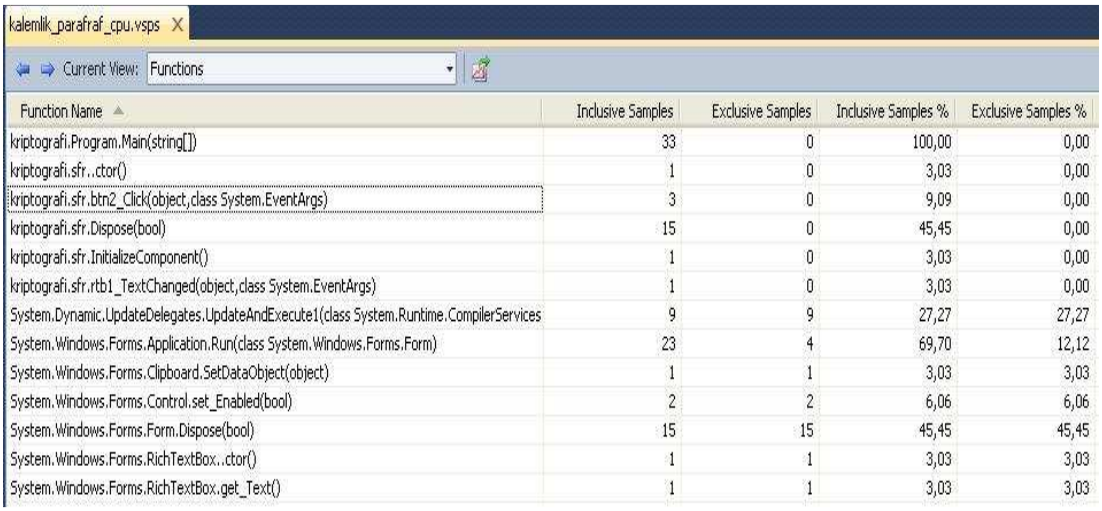
Current View: Modules

Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
kriptografi.exe	**33**	**0**	**100,00**	**0,00**
kriptografi.Program.Main(string[])	33	0	100,00	0,00
Line 20	9	0	27,27	0,00
Line 25	24	0	72,73	0,00
kriptografi.sfr..ctor()	1	0	3,03	0,00
Line 18	1	0	3,03	0,00
kriptografi.sfr.btn2_Click(object,class	3	0	9,09	0,00
Line 106	1	0	3,03	0,00
Line 134	1	0	3,03	0,00
Line 91	1	0	3,03	0,00
kriptografi.sfr.Dispose(bool)	15	0	45,45	0,00
Line 20	15	0	45,45	0,00
kriptografi.sfr.InitializeComponent()	1	0	3,03	0,00
kriptografi.sfr.rtb1_TextChanged(obj	1	0	3,03	0,00
System.Core.ni.dll	**9**	**9**	**27,27**	**27,27**
System.Dynamic.UpdateDelegates.Up	9	9	27,27	27,27
System.Windows.Forms.ni.dll	**24**	**24**	**72,73**	**72,73**
System.Windows.Forms.Application.F	23	4	69,70	12,12
System.Windows.Forms.Clipboard.Se	1	1	3,03	3,03
System.Windows.Forms.Control.set_	2	2	6,06	6,06
System.Windows.Forms.Form.Dispos	15	15	45,45	45,45
System.Windows.Forms.RichTextBox	1	1	3,03	3,03
System.Windows.Forms.RichTextBox	1	1	3,03	3,03

Resim 7.6. Sınıf ve fonksiyonların bulundukları satırları gösteren CPU kullanımı ile ilgili detaylı bilgi

Resim 7.6'da çerçeve içinde görülen alan şifreleme algoritmasının çalıştırıldığı *btn_Click* olayına ait CPU kullanımını göstermektedir. Geliştirilen algoritmanın, demo programın CPU kullanımının sadece %9,09≈%9,1'ini kullandığı tespit edilmiştir. Resim 7.4, Resim 7.6 ve Resim 7.7'de de bu değerler görülmektedir.

Resim 7.7, Current View'de Functions seçeneğine ait verileri göstermektedir.



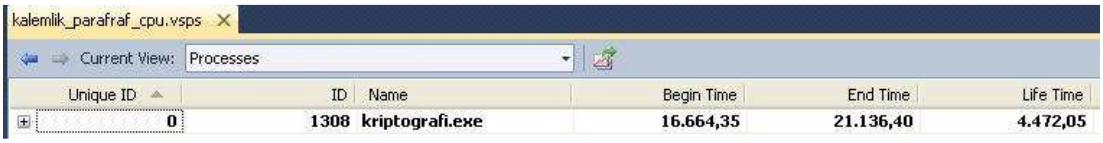
kalemlik_parafraf_cpu.vsps

Current View: Functions

Function Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
kriptografi.Program.Main(string[])	33	0	100,00	0,00
kriptografi.sfr..ctor()	1	0	3,03	0,00
kriptografi.sfr.btn2_Click(object,class System.EventArgs)	3	0	9,09	0,00
kriptografi.sfr.Dispose(bool)	15	0	45,45	0,00
kriptografi.sfr.InitializeComponent()	1	0	3,03	0,00
kriptografi.sfr.rtb1_TextChanged(object,class System.EventArgs)	1	0	3,03	0,00
System.Dynamic.UpdateDelegates.UpdateAndExecute1(class System.Runtime.CompilerServices	9	9	27,27	27,27
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	23	4	69,70	12,12
System.Windows.Forms.Clipboard.SetDataObject(object)	1	1	3,03	3,03
System.Windows.Forms.Control.set_Enabled(bool)	2	2	6,06	6,06
System.Windows.Forms.Form.Dispose(bool)	15	15	45,45	45,45
System.Windows.Forms.RichTextBox..ctor()	1	1	3,03	3,03
System.Windows.Forms.RichTextBox.get_Text()	1	1	3,03	3,03

Resim 7.7. Şifreleme işleminde kullanılan fonksiyonların CPU kullanım oranları

CPU kullanımına ait süreler Current View bölümünde *Marks* seçeneği ile detaylı olarak görülmektedir. *Processes* seçeneği ile CPU kullanımına ait süreler genel olarak gösterilmektedir. Resim 7.8'de geliştirilen demo programın CPU kullanımına ait sürelerin genel gösterimi görülmektedir:



kalemlik_parafraf_cpu.vsps

Current View: Processes

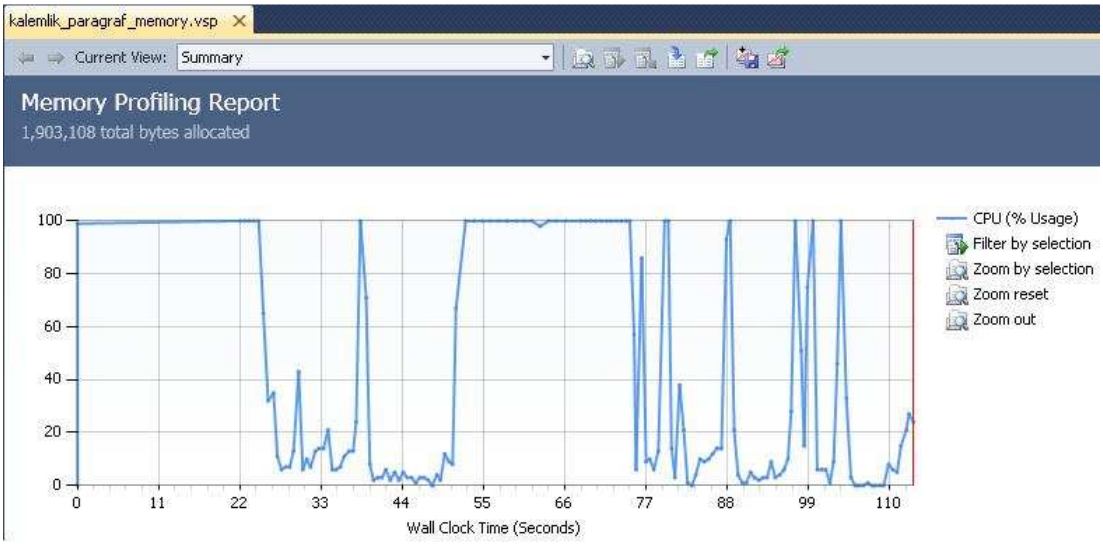
Unique ID	ID	Name	Begin Time	End Time	Life Time
0	1308	kriptografi.exe	16.664,35	21.136,40	4.472,05

Resim 7.8. CPU kullanımına ait süre bilgisi

Resim 7.8'den görüldüğü üzere demo programın CPU kullanım süresi 4,472,05 milisaniye olarak tespit edilmiştir.

.Net memory allocations verileri (bellek kullanımı verileri)

Resim 7.9'da bellek kullanımına ait grafik görülmektedir:



Resim 7.9. Paragraf düzeyinde şifreleme işlemleri bellek kullanımı

Resim 7.9'dan görüldüğü üzere *kalemlik* verisinin şifrelenmesi işlemini gerçekleştiren kriptografi.exe programı toplamda 1,903,108 byte miktarında bellek kullanmıştır. Resim 7.10'da bellek kullanımı ile ilgili genel bilgiler görülmektedir:

Functions Allocating Most Memory

Functions with the highest exclusive bytes allocated

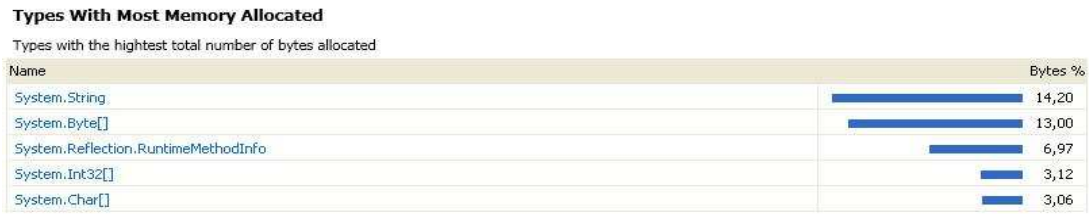
Name	Bytes %
System.Dynamic.UpdateDelegates.UpdateAndExecute1(class System.Runtime.CompilerServices.CallSite,!!0)	30,01
System.Dynamic.UpdateDelegates.UpdateAndExecuteVoid2(class System.Runtime.CompilerServices.CallSite,!!0,!!1)	19,42
System.Windows.Forms.RichTextBox.get_Text()	10,55
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	10,43
System.Windows.Forms.Form..ctor()	6,54

Resim 7.10. Bellek kullanımı en yüksek olan fonksiyonlar

Resim 7.10'daki veriler incelendiğinde CPU üzerinde en fazla yük oluşturan fonksiyonların aynı zaman da bellek kullanımının en fazla olduğu görülmektedir. Resim 7.10'da görülen fonksiyonlar aynı zamanda sistemde darboğaz problemi oluşturan fonksiyonlardır. Satırlara tek tek tıklanarak bellek kullanımını arttıran fonksiyonlar hakkında detaylı verilere ulaşılabilir.

Resim 7.12, Resim 7.13 ve Resim 7.14, detaylı verilerin gösterilmesi amacıyla verilmiştir.

Resim 7.11, bellek kullanımı en yüksek olan veri tiplerini göstermektedir:



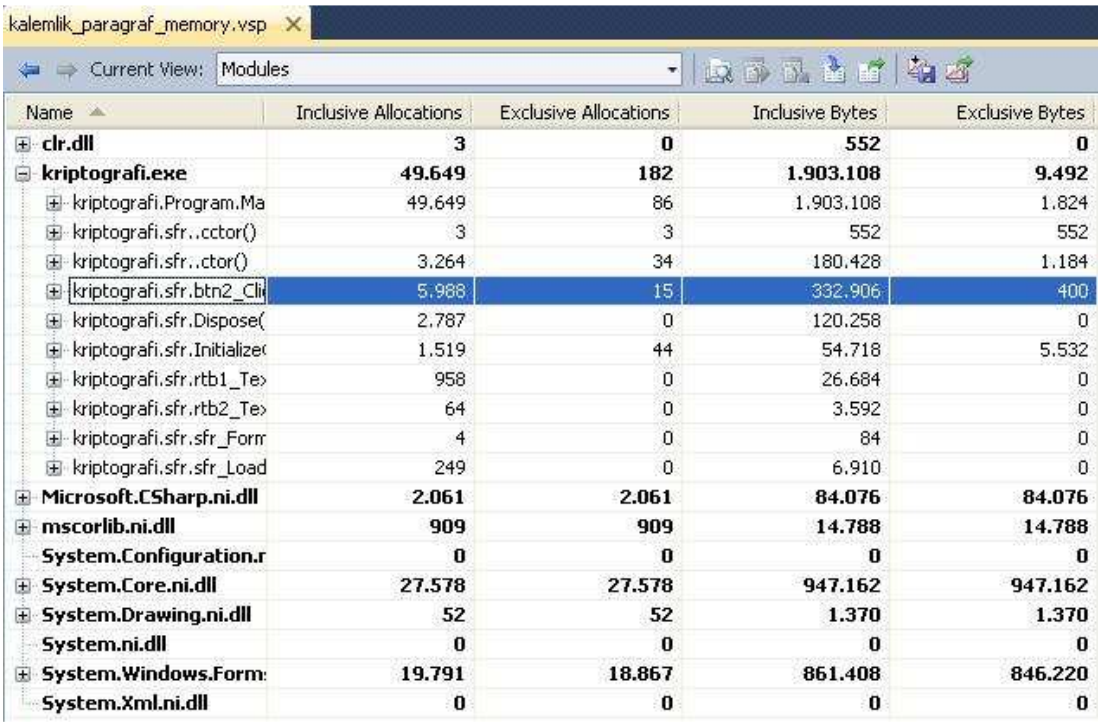
Types With Most Memory Allocated

Types with the hightest total number of bytes allocated

Name	Bytes %
System.String	14,20
System.Byte[]	13,00
System.Reflection.RuntimeMethodInfo	6,97
System.Int32[]	3,12
System.Char[]	3,06

Resim 7.11. Bellek kullanımı en yüksek olan veri tipleri

Modules seçeneği ile fonksiyon ve sınıfların bellek kullanımı ile ilgili detaylı bilgiye ulaşılabilir. Resim 7.12'de Modules seçeneği ile bellek kullanımına ait daha detaylı olarak elde edilen bilgiler görülmektedir:



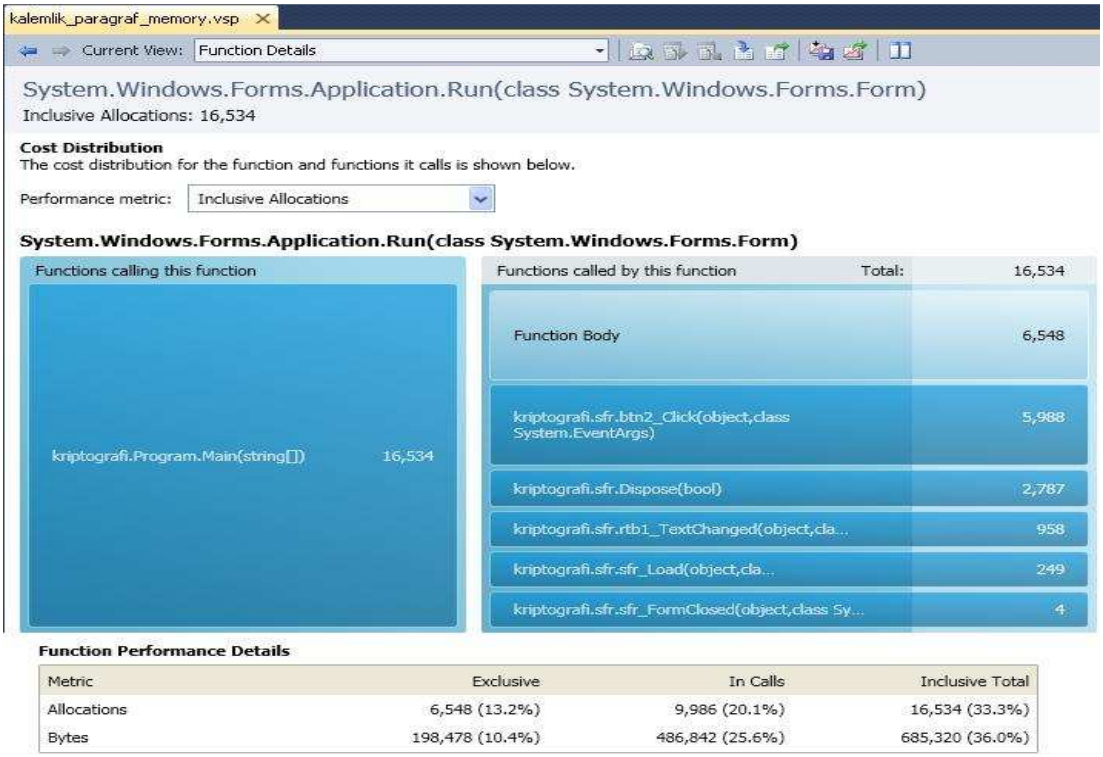
kalemlik_paragraf_memory.vsp

Current View: Modules

Name	Inclusive Allocations	Exclusive Allocations	Inclusive Bytes	Exclusive Bytes
clr.dll	**3**	**0**	**552**	**0**
kriptografi.exe	**49.649**	**182**	**1.903.108**	**9.492**
kriptografi.Program.Ma	49.649	86	1.903.108	1.824
kriptografi.sfr..cctor()	3	3	552	552
kriptografi.sfr..ctor()	3.264	34	180.428	1.184
kriptografi.sfr.btn2_Cli	5.988	15	332.906	400
kriptografi.sfr.Dispose(	2.787	0	120.258	0
kriptografi.sfr.Initialize(	1.519	44	54.718	5.532
kriptografi.sfr.rtb1_Te>	958	0	26.684	0
kriptografi.sfr.rtb2_Te>	64	0	3.592	0
kriptografi.sfr.sfr_Form	4	0	84	0
kriptografi.sfr.sfr_Load	249	0	6.910	0
Microsoft.CSharp.ni.dll	**2.061**	**2.061**	**84.076**	**84.076**
mscorlib.ni.dll	**909**	**909**	**14.788**	**14.788**
System.Configuration.r	**0**	**0**	**0**	**0**
System.Core.ni.dll	**27.578**	**27.578**	**947.162**	**947.162**
System.Drawing.ni.dll	**52**	**52**	**1.370**	**1.370**
System.ni.dll	**0**	**0**	**0**	**0**
System.Windows.Form	**19.791**	**18.867**	**861.408**	**846.220**
System.Xml.ni.dll	**0**	**0**	**0**	**0**

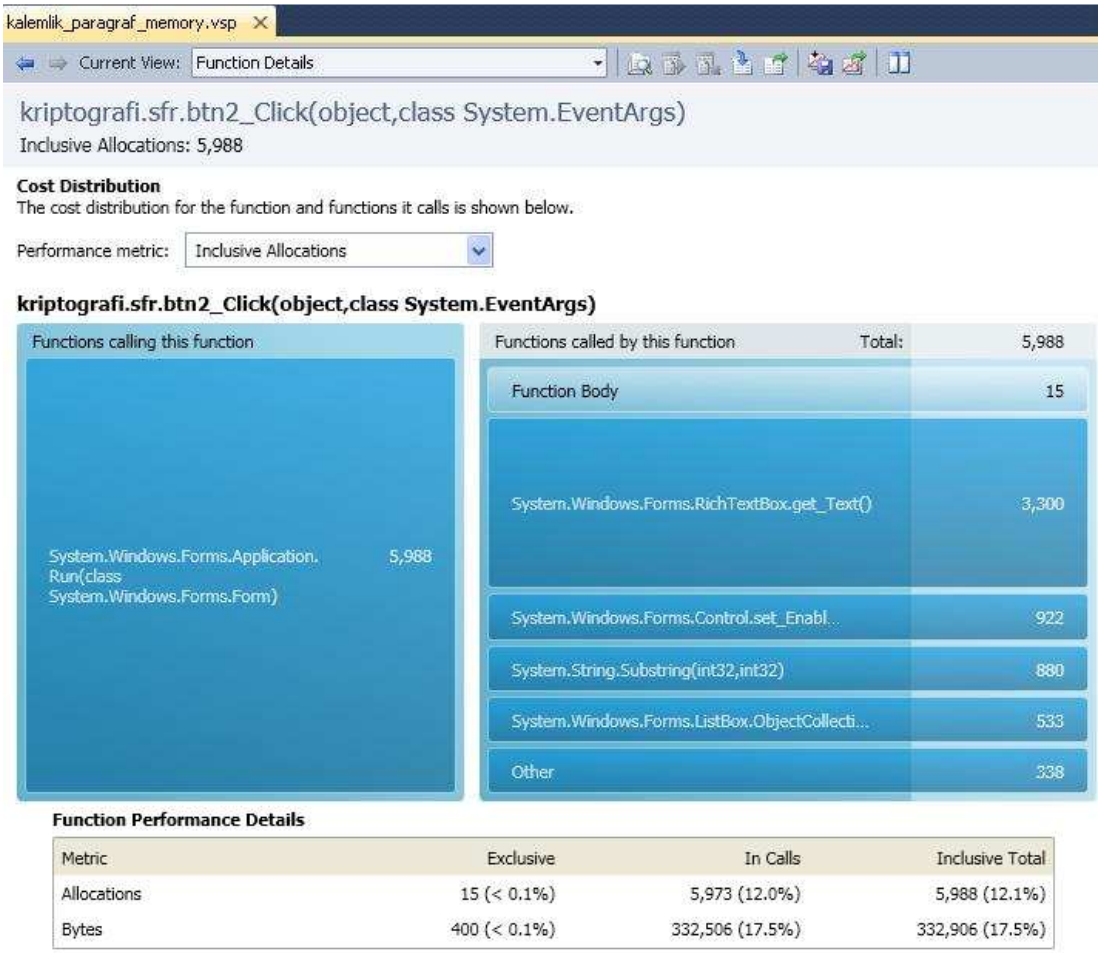
Resim 7.12. Fonksiyon ve sınıfların bellek kullanımına ait detaylar

Current View kısmında Functions ya da Allocation seçilirse tüm fonksiyonların tek tek görüntülendiği daha detaylı bilgi ekranına ulaşılabilir. Resim 7.12'de arka planı koyu görülen değerler şifreleme algoritmasına aittir. Şifreleme işlemleri *btn2_Click* olayında gerçekleşmektedir. kriptografi.exe programı 1,903,108 Byte'lık bellek alanı kullanırken bu kullanımın sadece 332,906 Byte'lık kısmı şifreleme işlemlerinde kullanılmaktadır.



Resim 7.13. Application Run sınıfına ait bellek kullanım detayları

Resim 7.13'te görüldüğü üzere byte cinsinden incelendiğinde Application Run sınıfına ait fonksiyonlar toplam bellek kullanımının %36,0'ına sahiptir. Şifreleme algoritmasına ait bellek verileri(*btn2_Click olayı*) Resim 7.14'te görülmektedir:



Resim 7.14. Şifreleme algoritmasına ait bellek kullanım oranları

Resim 7.14'te *btn2_Click* olayına ait alt olayların bellek kullanım değerleri görülmektedir. Bu değerlere göre, byte cinsinden incelendiğinde şifreleme algoritmasının kullandığı bellek, programın toplam bellek kullanımının %17,5'ini oluşturmaktadır.

İşlem zamanı verileri

Resim 7.15, işlem zamanına ait grafiği göstermektedir. *kalemlik* kelimesinin şifrelenmesi için geçen işlem zamanı süresince algoritmanın içermiş olduğu işlemlere göre CPU kullanımında dalgalanmalar oluşmaktadır.

Resim 7.15. Geliştirilen algoritmanın işlem zamanı-CPU grafiği

Resim 7.15'te görüldüğü gibi geliştirilen şifreleme algoritmasının şifreleme işlemini tamamlama süresi 12,211,08 milisaniye olarak tespit edilmiştir. Bu süre, arayüzün ekrana yüklenmesinden kapatılmasına kadar geçen toplam süredir.

Hot Path

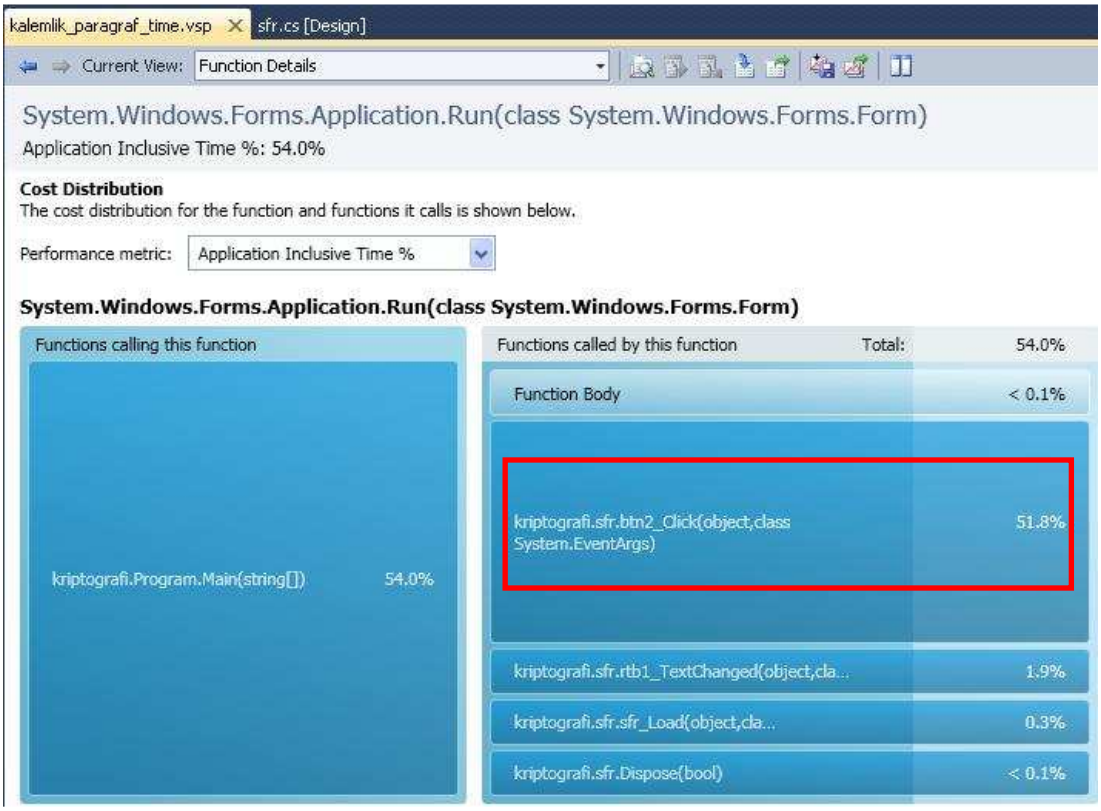
The most expensive call path based on execution times

Function Name	Elapsed Inclusive Time %	Elapsed Exclusive Time %
kriptografi.exe	100,00	0,00
kriptografi.Program.Main(string[])	100,00	0,29
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	**97,71**	**96,60**

Resim 7.16. En çok süre harcayan sınıf ve fonksiyonlar

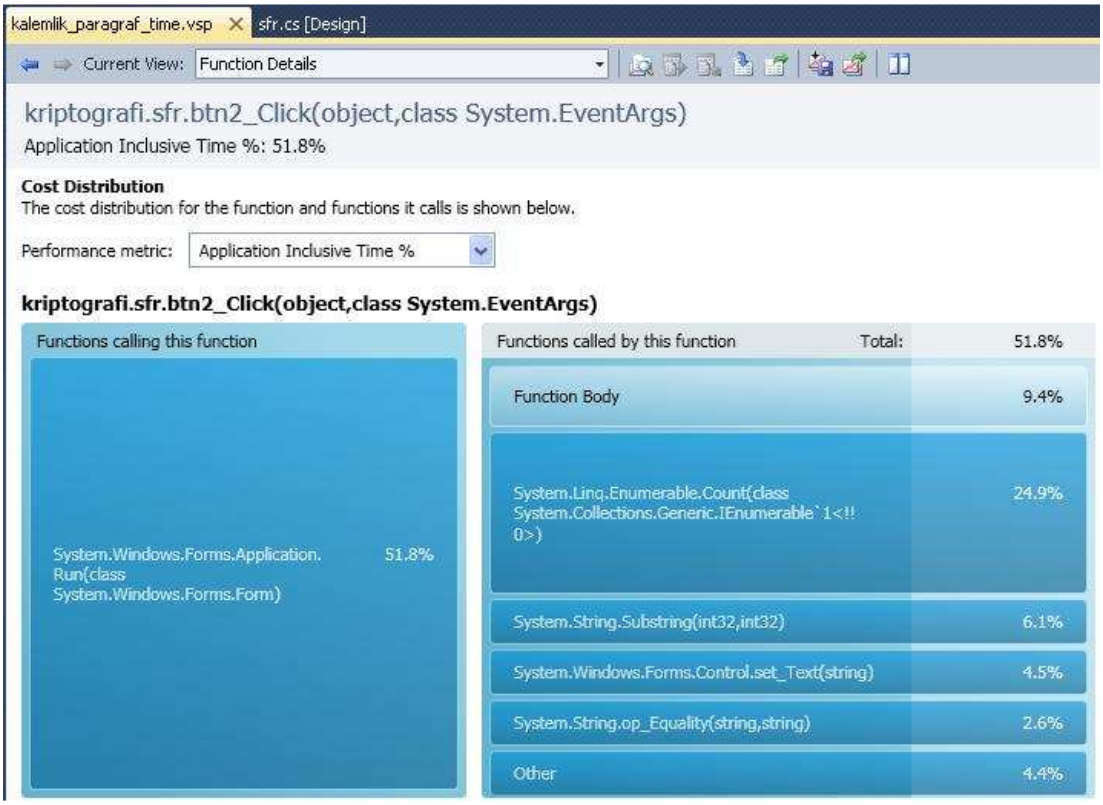
Resim 7.16, işlem süresi en uzun olan ve sistemi zorlayan sınıf ve fonksiyonları göstermektedir. Satırın üzerine tıklanınca detaylı bilgi veren ekrandan fonksiyonların ve olayların kendileri için kullandıkları işlem süreleri görülebilir.

Application Run metoduna ait detaylı bilgiler Resim 7.17 ve Resim 7.18'de görülmektedir:



Resim 7.17. Application Run metodu için gerekli işlem zamanı

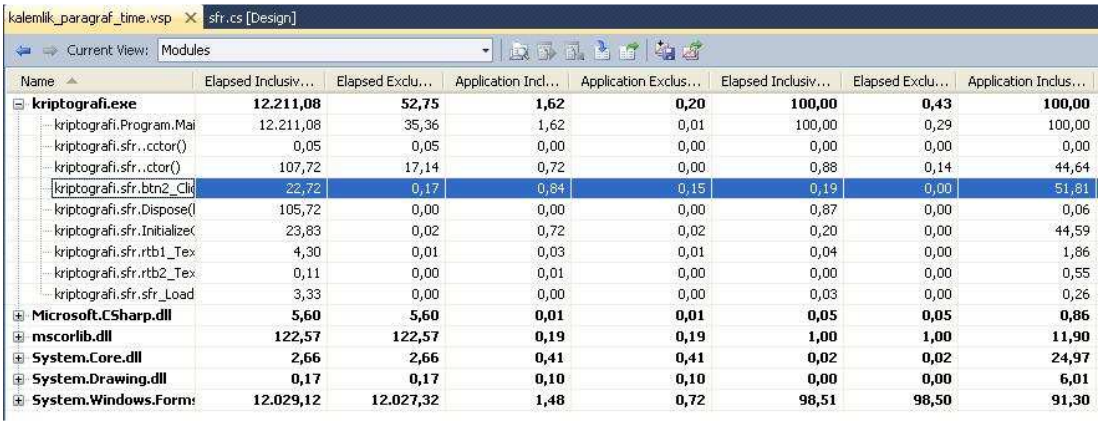
Application Run metodu toplam işlem zamanının %54'üne sahiptir. Sağ sütunda çerçeve içine alınmış *btn2_Click* olayı şifreleme algoritmasının çalışmasını sağlayan olaydır. Dolayısıyla şifreleme algoritmasının işlem zamanı %51,8 olarak tespit edilmiştir. Resim 7.18 *btn_Click* olayına ait fonksiyon ve olayları göstermektedir:



Resim 7.18. Şifreleme işleminde kullanılan fonksiyon ve olayların kendilerine ait işlem zamanı süreleri

Resim 7.18'den görüldüğü üzere algoritma sıralama işlemleri ve string denilen metin işlemlerinden oluşmaktadır. Sıralama işlemlerinde sayma, indis değerlerin hesaplanması, yer değiştirme, yerine koyma gibi işlemler yapılmaktadır. String işlemlerde ise sabit bir karakterin(harfin) aratılması, eşleştirilmesi, harf olup olmadığının kontrol edilmesi gibi işlemler bulunmaktadır. Bu işlemlerin %'lik değerleri Resim 7.18'de görüldüğü gibi tespit edilmiştir.

Resim 7.19'da şifreleme işlemlerine ait (*btn2_Click*) işlem zamanı değerleri milisaniye ve % olarak görülmektedir.


kalemlik_paragraf_time.vsp | sfr.cs [Design]

Current View: Modules

Name	Elapsed Inclusiv…	Elapsed Exclu…	Application Incl…	Application Exclus…	Elapsed Inclusiv…	Elapsed Exclu…	Application Inclus…
kriptografi.exe	**12.211,08**	**52,75**	**1,62**	**0,20**	**100,00**	**0,43**	**100,00**
kriptografi.Program.Mai	12.211,08	35,36	1,62	0,01	100,00	0,29	100,00
kriptografi.sfr..cctor()	0,05	0,05	0,00	0,00	0,00	0,00	0,00
kriptografi.sfr..ctor()	107,72	17,14	0,72	0,00	0,88	0,14	44,64
kriptografi.sfr.btn2_Cli	22,72	0,17	0,84	0,15	0,19	0,00	51,81
kriptografi.sfr.Dispose(I	105,72	0,00	0,00	0,00	0,87	0,00	0,06
kriptografi.sfr.Initialize(	23,83	0,02	0,72	0,02	0,20	0,00	44,59
kriptografi.sfr.rtb1_Tex	4,30	0,01	0,03	0,01	0,04	0,00	1,86
kriptografi.sfr.rtb2_Tex	0,11	0,00	0,01	0,00	0,00	0,00	0,55
kriptografi.sfr.sfr_Load	3,33	0,00	0,00	0,00	0,03	0,00	0,26
Microsoft.CSharp.dll	**5,60**	**5,60**	**0,01**	**0,01**	**0,05**	**0,05**	**0,86**
mscorlib.dll	**122,57**	**122,57**	**0,19**	**0,19**	**1,00**	**1,00**	**11,90**
System.Core.dll	**2,66**	**2,66**	**0,41**	**0,41**	**0,02**	**0,02**	**24,97**
System.Drawing.dll	**0,17**	**0,17**	**0,10**	**0,10**	**0,00**	**0,00**	**6,01**
System.Windows.Forms	**12.029,12**	**12.027,32**	**1,48**	**0,72**	**98,51**	**98,50**	**91,30**

Resim 7.19. Şifreleme algoritmasının işlem zamanı için milisaniye ve % değerleri

Resim 7.19'da görülen ilk dört sütun işlem zamanını milisaniye olarak göstermektedir. Son dört sütun ise %'lik değerleri vermektedir.

Resim 7.20, geliştirilen algoritmanın şifreleme işlemlerine ait toplam işlem zamanı süresini göstermektedir.

Function Performance Details

Metric	Exclusive	In Calls	Inclusive Total
Application Time	0.2 (9.4%)	0.7 (42.4%)	0.8 (51.8%)
Elapsed Time	0.2 (< 0.1%)	22.6 (0.2%)	22.7 (0.2%)

Resim 7.20. Şifreleme algoritmasının toplam işlem zamanı süresi

Resim 7.20'den görüleceği üzere algoritmanın işlem zamanı 0,8 milisaniye olarak tespit edilmiştir. Resim 7.20'de %'lik değer olarak, Resim 7.16'da de görüldüğü gibi toplam işlem zamanının %51,8'i şifreleme algoritmasına aittir. 0,8 milisaniyelik süre programın toplam işlem zamanının %51,8'ini oluşturmaktadır.

Çizelge 7.2'de performans analiz değerlerinden elde edilen değerler(işlem zamanı, bellek kullanımı, CPU kullanımı) yer almaktadır.

Çizelge 7.2. Geliştirilen algoritma ile 8 byte'lık metnin(kalemlik) paragraf düzeyinde şifrelenmesine ait performans değerleri

	İşlem zamanı (msec)	**Hafıza Kullanımı (Byte %)**	**CPU Kullanımı (%)**
Geliştirilen algoritma (paragraf düzeyinde şifreleme)	0,8	17,5	9,1

7.1.2. Kelime düzeyinde şifreleme işlemleri performans analiz değerleri

Paragraf düzeyinde şifreleme işlemlerinden farklı olarak her kelimenin harfleri için sahip olduğu indis değerleri vardır. Bu indis değeri her kelime için sıfırdan başlayarak, tüm kelimelere uygulanır.

CPU sampling verileri

Resim 7.21. "kalemlik" (8 byte) kelimesinin kelime düzeyinde şifrelenmesine ait CPU grafiği

Toplam 27 (sınıf, fonksiyon) için CPU kullanımı Resim 7.21'deki grafikte görülmektedir. CPU kullanımı program ilk çalıştırıldığında %100'e ulaşmakta, daha sonra şifreleme işlemlerinde kullanılan fonksiyon ve sınıflara göre (metin kutularının içeriğinin değişmesi, butonlara tıklanması, döngülerin çalıştırılması gibi) değişkenlik göstermektedir.

Current View kısmında farklı seçenekler bulunmaktadır. *Summary*, CPU ile ilgili kullanım bilgilerinin özet şeklinde ve en genel haliyle görülmesini sağlar.

Resim 7.22, sistemi zorlayan ve darboğaz oluşturan sınıf ve fonksiyonları göstermektedir.

Hot Path

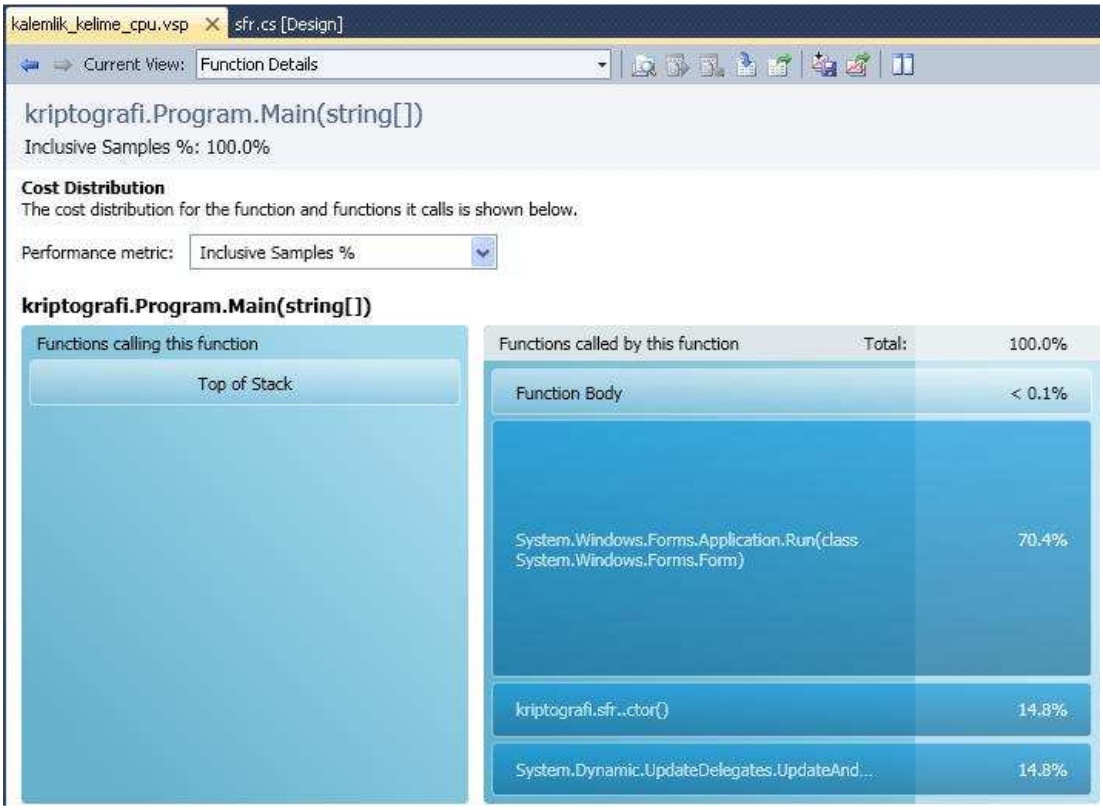
The most expensive call path based on sample counts

Function Name	Inclusive Samples %	Exclusive Samples %
kriptografi.exe	100,00	0,00
kriptografi.Program.Main(string[])	100,00	0,00
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	**70,37**	**14,81**
System.Dynamic.UpdateDelegates.UpdateAndExecute1(class System.Runti...	**14,81**	**14,81**
kriptografi.sfr..ctor()	**14,81**	**0,00**

Resim 7.22. “kalemlik” kelimesinin kelime düzeyinde şifrelenmesinde sistemi en çok zorlayan sınıflar

Hot Path bölümünde *Inclusive Samples* yüzdesi yüksek olan sınıflar sistemi en çok zorlayan sınıflardır. *Exclusive Samples* bölümünde ise performans darboğazı yaratan fonksiyonlar en yüksek yüzdeye sahiptir [68]. Bu fonksiyonları daha detaylı görebilmek için yanlarında ateş simgesi bulunan yazıların üzerine tıklamak yeterlidir ya da *Current View* kısmında *Function Details* seçeneği ile detaylara ulaşmak mümkündür.

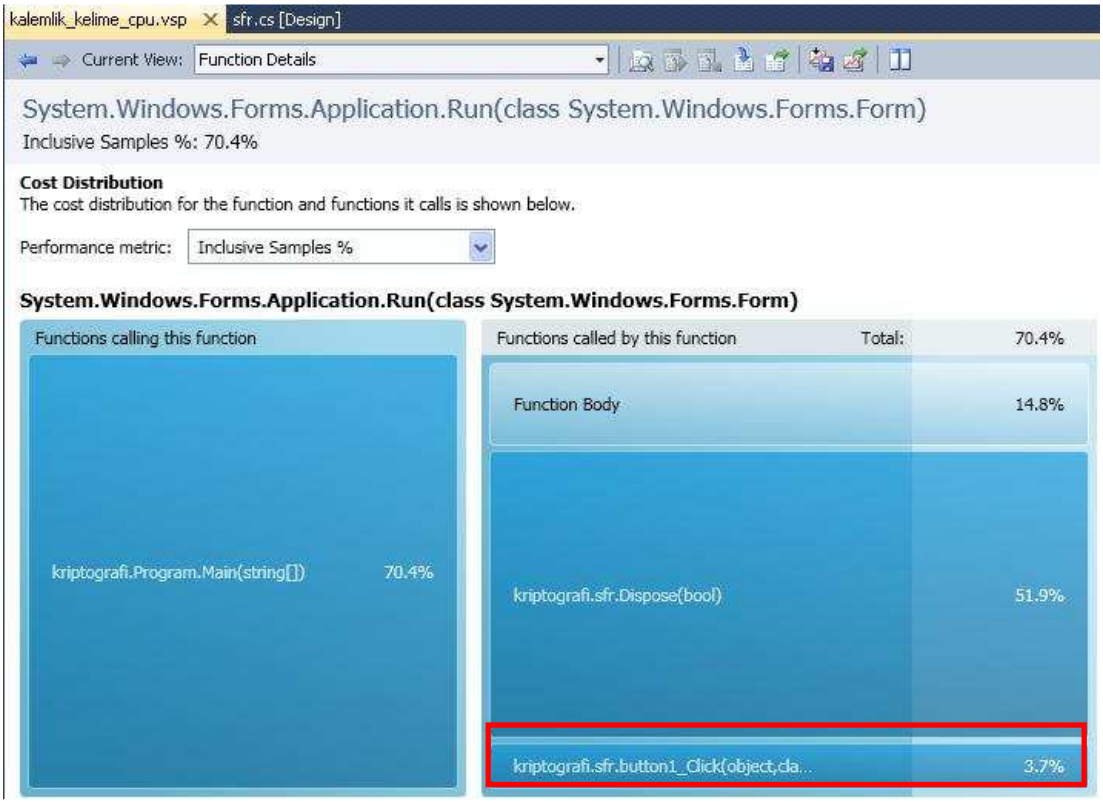
Resim 7.23’te fonksiyonların bireysel olarak CPU kullanım oranları görülmektedir.



Resim 7.23. Kelime düzeyinde şifreleme işlemine ait fonksiyonların CPU kullanım oranları

Şifreleme algoritmasının çalıştırıldığı *button1_Click* olayı Application Run sınıfına aittir. Resim 7.23'te görüldüğü üzere Application Run sınıfının CPU kullanım oranı programın toplam CPU kullanımının % 70,4'üdür.

Resim 7.24'te şifreleme işleminin gerçekleştiği *button1_Click* olayına ait CPU kullanımı durumu görülmektedir.



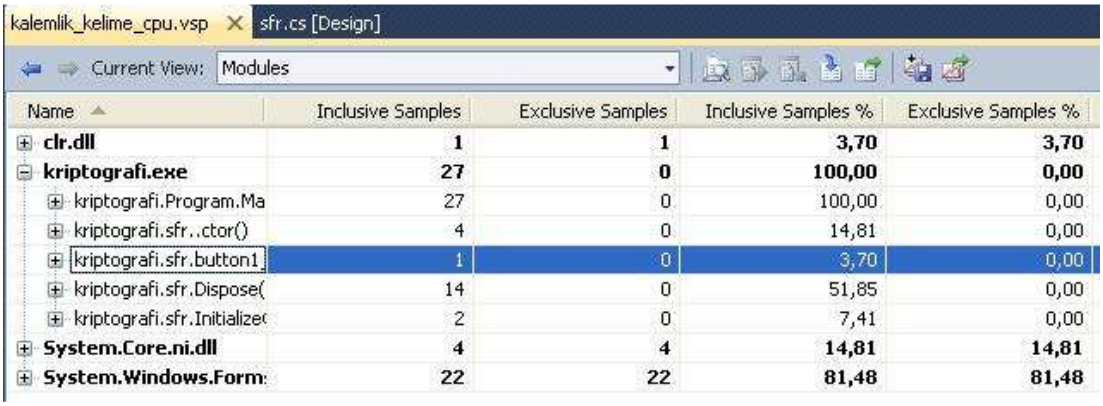
Resim 7.24. Şifreleme algoritmasına ait CPU kullanım oranı

Resim 7.24'ten görüldüğü üzere şifreleme algoritmasının CPU kullanımı programın CPU kullanımının sadece % 3,7'sini oluşturmaktadır. Bu durum Resim 7.25'te detaylı olarak görülmektedir.

Function Performance Details

Metric	Exclusive	In Calls	Inclusive Total
Collected Samples	0 (< 0.1%)	1 (3.7%)	1 (3.7%)

Resim 7.25. button1_Click olayına ait CPU kullanım detayı



kalemlik_kelime_cpu.vsp | sfr.cs [Design]

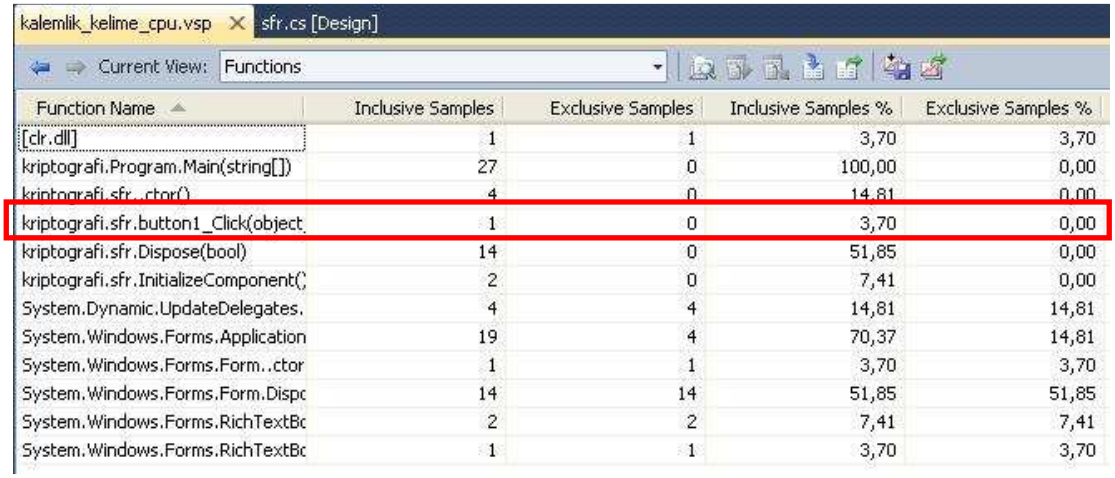
Current View: Modules

Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
clr.dll	**1**	**1**	**3,70**	**3,70**
kriptografi.exe	**27**	**0**	**100,00**	**0,00**
kriptografi.Program.Ma	27	0	100,00	0,00
kriptografi.sfr..ctor()	4	0	14,81	0,00
kriptografi.sfr.button1	1	0	3,70	0,00
kriptografi.sfr.Dispose(	14	0	51,85	0,00
kriptografi.sfr.Initialize	2	0	7,41	0,00
System.Core.ni.dll	**4**	**4**	**14,81**	**14,81**
System.Windows.Form	**22**	**22**	**81,48**	**81,48**

Resim 7.26. Kelime düzeyinde şifreleme işleminin CPU kullanımına ait detaylar

Resim 7.26'da arka zemini koyu görülen alan şifreleme algoritmasının çalıştırıldığı *button1_Click* olayına ait CPU kullanımını göstermektedir. Şifreleme algoritmasının CPU kullanımı, geliştirilen demo programın CPU kullanımının sadece %3,70'i olarak tespit edilmiştir.

Resim 7.27, Current View'de Functions seçeneğine ait verileri göstermektedir.



kalemlik_kelime_cpu.vsp | sfr.cs [Design]

Current View: Functions

Function Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
[clr.dll]	1	1	3,70	3,70
kriptografi.Program.Main(string[])	27	0	100,00	0,00
kriptografi.sfr..ctor()	4	0	14,81	0,00
kriptografi.sfr.button1_Click(object,	1	0	3,70	0,00
kriptografi.sfr.Dispose(bool)	14	0	51,85	0,00
kriptografi.sfr.InitializeComponent()	2	0	7,41	0,00
System.Dynamic.UpdateDelegates.	4	4	14,81	14,81
System.Windows.Forms.Application	19	4	70,37	14,81
System.Windows.Forms.Form..ctor	1	1	3,70	3,70
System.Windows.Forms.Form.Dispc	14	14	51,85	51,85
System.Windows.Forms.RichTextBc	2	2	7,41	7,41
System.Windows.Forms.RichTextBc	1	1	3,70	3,70

Resim 7.27. Kelime düzeyinde şifreleme işleminde kullanılan fonksiyonların CPU kullanım oranları

.Net memory allocations verileri (bellek kullanımı verileri)

Resim 7.28'de bellek kullanımına ait grafik görülmektedir:

Resim 7.28. Kelime düzeyinde şifreleme işlemleri bellek kullanımı

Resim 7.28'den görüldüğü üzere *kalemlik* verisinin şifrelenmesi işlemini gerçekleştiren kriptografi.exe programı toplamda 1,918,672 byte bellek kullanmıştır. Resim 7.29'da bellek kullanımında sistemi zorlayan fonksiyonlar ve oranları görülmektedir:

Functions Allocating Most Memory

Functions with the highest exclusive bytes allocated

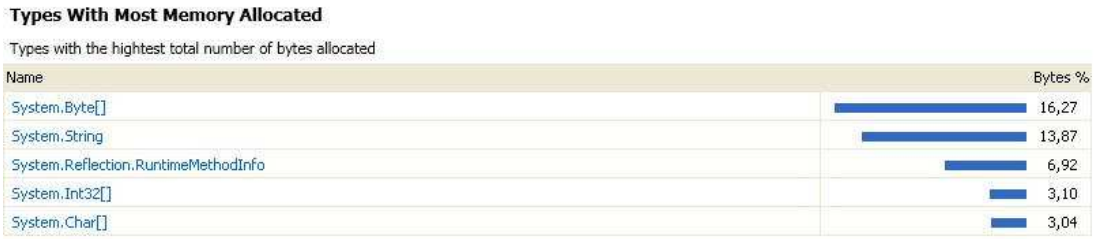
Name	Bytes %
System.Dynamic.UpdateDelegates.UpdateAndExecute1(class System.Runtime.CompilerServices.CallSite,!!0)	29,76
System.Dynamic.UpdateDelegates.UpdateAndExecuteVoid2(class System.Runtime.CompilerServices.CallSite,!!0,!!1)	19,26
System.Windows.Forms.RichTextBox.get_Text()	10,39
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	7,84
System.Windows.Forms.RichTextBox.set_Text(string)	7,52

Resim 7.29. Bellek kullanımı en yüksek olan fonksiyonlar

Resim 7.29'daki veriler incelendiğinde CPU üzerinde en fazla yük oluşturan fonksiyonların aynı zaman da bellek kullanımının en fazla olduğu görülmektedir. Resim 7.29'da görülen fonksiyonlar aynı zamanda sistemde

darboğaz problemi oluşturan fonksiyonlardır. Satırlara tek tek tıklanarak bellek kullanımını arttıran fonksiyonlar hakkında detaylı verilere ulaşılabilir.

Resim 7.30, bellek kullanımı en yüksek olan veri tiplerini göstermektedir:



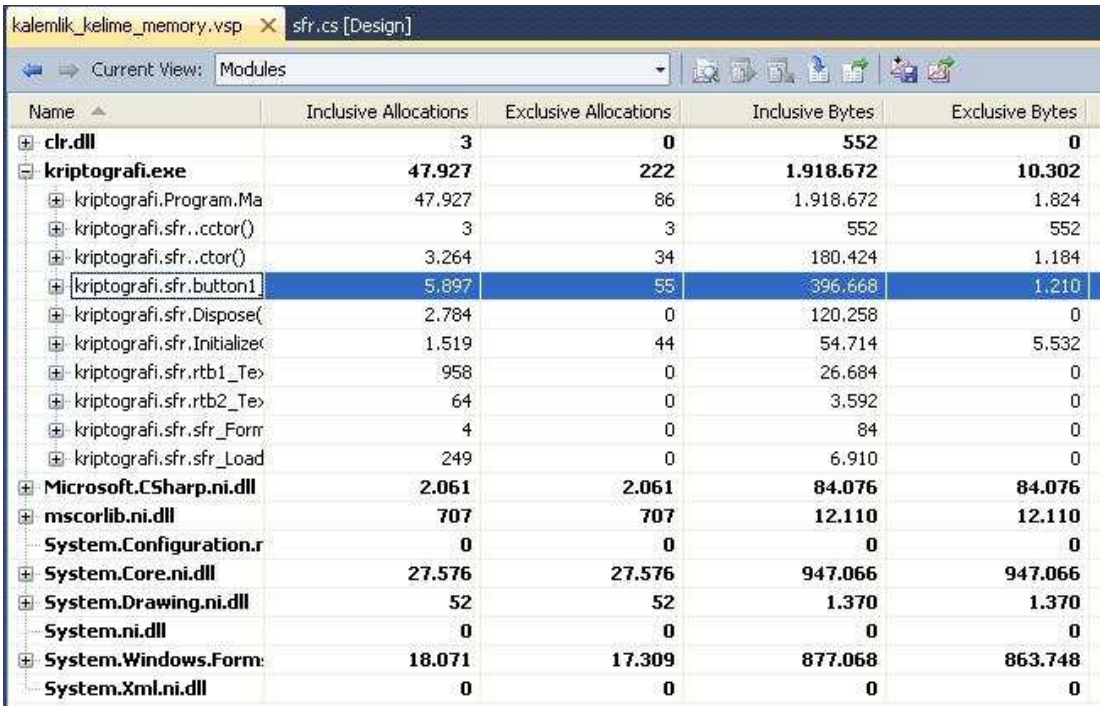
Types With Most Memory Allocated

Types with the hightest total number of bytes allocated

Name	Bytes %
System.Byte[]	16,27
System.String	13,87
System.Reflection.RuntimeMethodInfo	6,92
System.Int32[]	3,10
System.Char[]	3,04

Resim 7.30. Bellek kullanımı en yüksek olan veri tipleri

Modules seçeneği ile fonksiyon ve sınıfların bellek kullanımı ile ilgili detaylı bilgiye ulaşılabilir. Resim 7.31'de Modules seçeneği ile bellek kullanımına ait daha detaylı olarak elde edilen bilgiler görülmektedir:



kalemlik_kelime_memory.vsp | sfr.cs [Design]

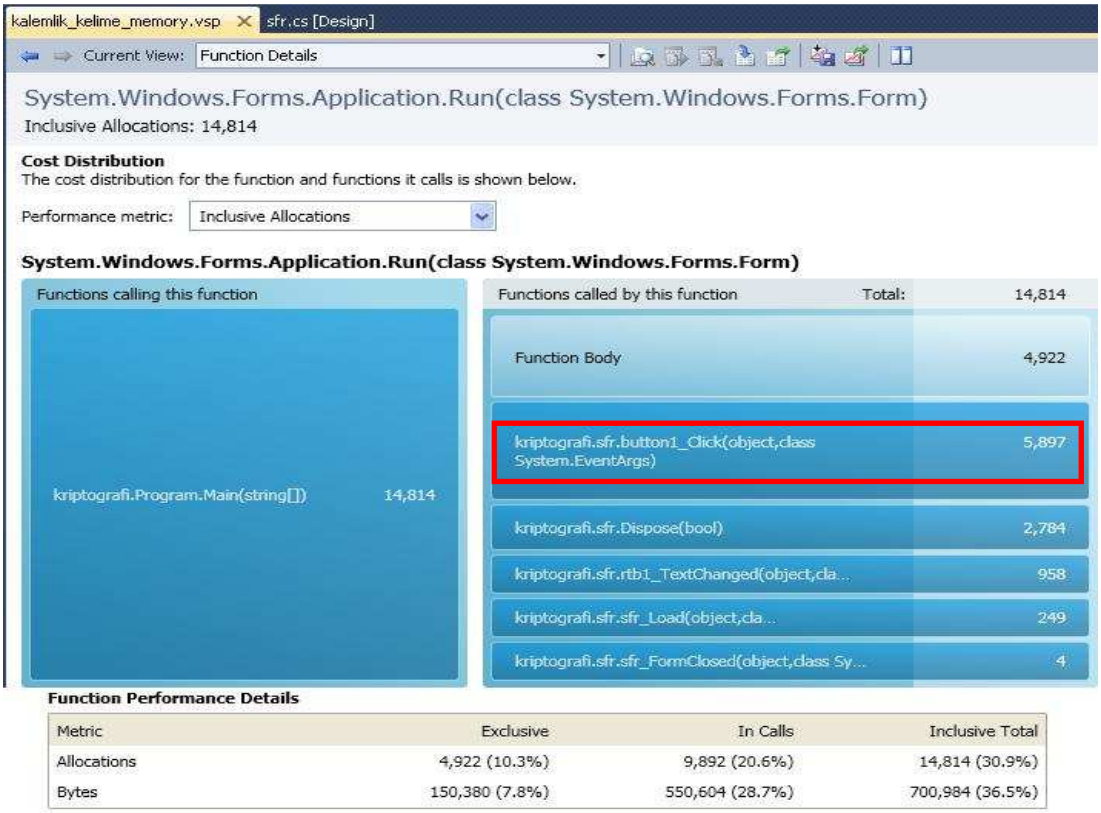
Current View: Modules

Name	Inclusive Allocations	Exclusive Allocations	Inclusive Bytes	Exclusive Bytes
clr.dll	**3**	**0**	**552**	**0**
kriptografi.exe	**47.927**	**222**	**1.918.672**	**10.302**
kriptografi.Program.Ma	47.927	86	1.918.672	1.824
kriptografi.sfr..cctor()	3	3	552	552
kriptografi.sfr..ctor()	3.264	34	180.424	1.184
kriptografi.sfr.button1_	5.897	55	396.668	1.210
kriptografi.sfr.Dispose(	2.784	0	120.258	0
kriptografi.sfr.Initialize(	1.519	44	54.714	5.532
kriptografi.sfr.rtb1_Te>	958	0	26.684	0
kriptografi.sfr.rtb2_Te>	64	0	3.592	0
kriptografi.sfr.sfr_Form	4	0	84	0
kriptografi.sfr.sfr_Load	249	0	6.910	0
Microsoft.CSharp.ni.dll	**2.061**	**2.061**	**84.076**	**84.076**
mscorlib.ni.dll	**707**	**707**	**12.110**	**12.110**
System.Configuration.r	**0**	**0**	**0**	**0**
System.Core.ni.dll	**27.576**	**27.576**	**947.066**	**947.066**
System.Drawing.ni.dll	**52**	**52**	**1.370**	**1.370**
System.ni.dll	**0**	**0**	**0**	**0**
System.Windows.Form	**18.071**	**17.309**	**877.068**	**863.748**
System.Xml.ni.dll	**0**	**0**	**0**	**0**

Resim 7.31. Fonksiyon ve sınıfların bellek kullanımına ait detaylar

Current View kısmında Functions ya da Allocation seçilirse tüm fonksiyonların tek tek görüntülendiği daha detaylı bilgi ekranına ulaşılabilir. Resim 7.31'de arka planı koyu görülen değerler şifreleme algoritmasına aittir. Şifreleme işlemleri *button1_Click* olayında gerçekleşmektedir. İlk iki sütun %'lik değerleri, son iki sütun Byte miktarlarını göstermektedir. Byte cinsinden incelendiğinde, kriptografi.exe programının bellek kullanımı 1,918,672 Byte iken bu kullanımın sadece 396,668 byte'lık kısmı şifreleme işlemlerinde kullanılmaktadır.

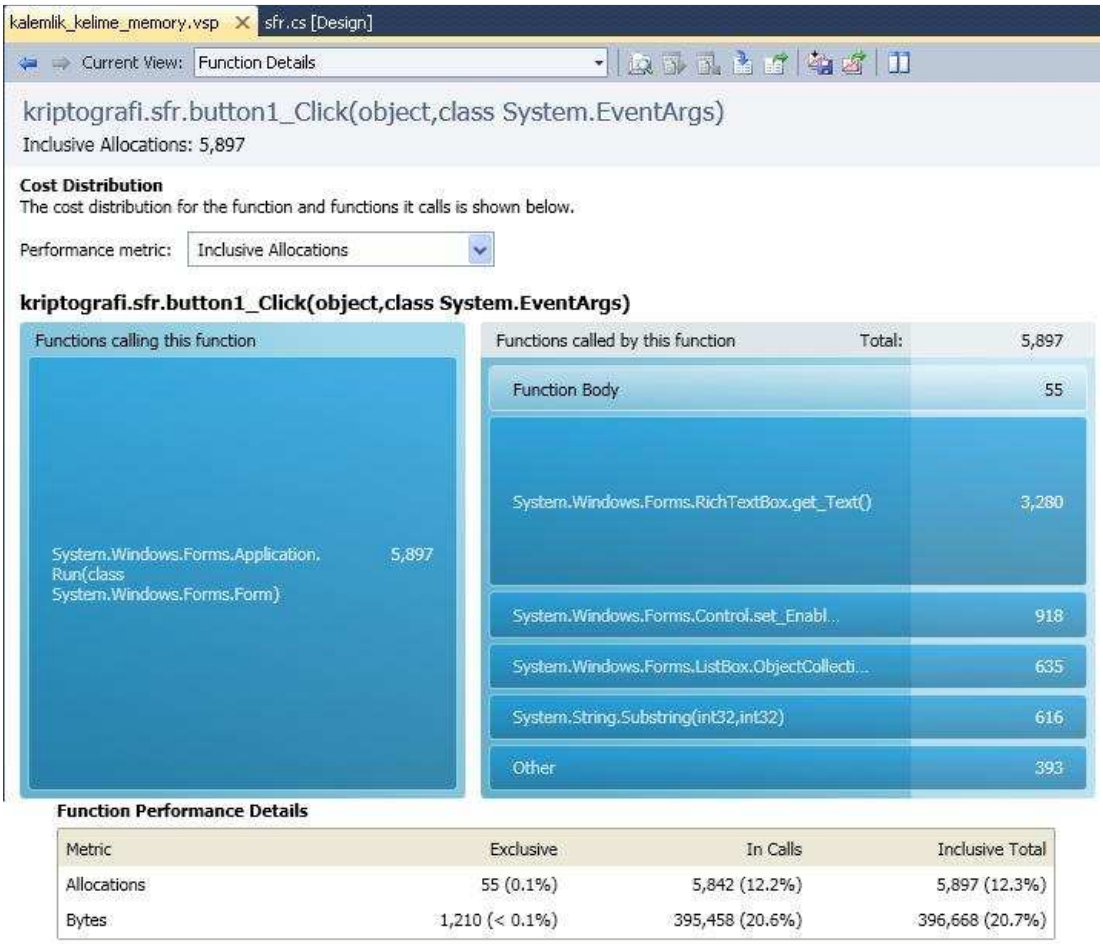
Resim 7.32 ve Application Run sınıfına ait bellek kullanım oranlarını göstermektedir.



Resim 7.32. Application Run sınıfına ait bellek kullanım detayları

Resim 7.32'de görüldüğü üzere, byte cinsinden incelendiğinde Application Run sınıfına ait fonksiyonlar toplam bellek kullanımının %36,5'ine sahiptir.

Şifreleme algoritmasına ait bellek verileri(*button1_Click olayı*) Resim 7.33'te görülmektedir:

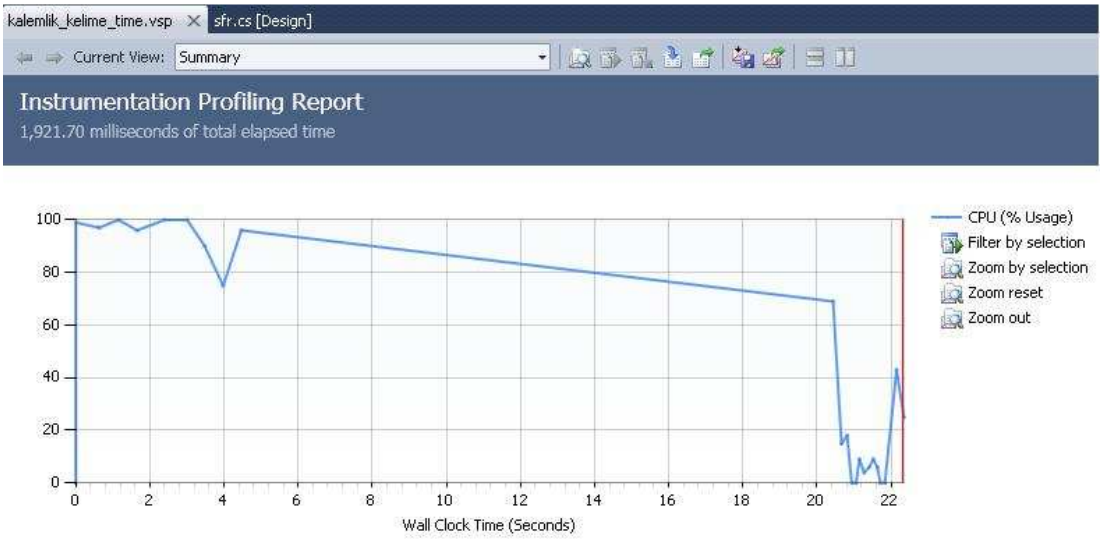


Resim 7.33. Şifreleme algoritmasına ait bellek kullanım oranları

Resim 7.33'te *button1_Click* olayına ait alt olayların bellek kullanım değerleri görülmektedir. Byte cinsinden incelendiğinde programın toplam bellek kullanımına göre şifreleme algoritmasının bellek kullanımı %20,7 olarak tespit edilmiştir.

İşlem zamanı verileri

Resim 7.34, programın işlem zamanına ait grafiği göstermektedir. *kalemlik* kelimesinin şifrelenmesi için geçen işlem zamanı süresince çağırılan sınıf ve fonksiyonlara göre CPU kullanımında dalgalanmalar oluşmaktadır.



Resim 7.34. Geliştirilen algoritmanın işlem zamanı-CPU grafiği

Resim 7.34'te görüldüğü gibi geliştirilen şifreleme algoritmasının şifreleme işlemini tamamlama süresi 1,921,70 milisaniye olarak tespit edilmiştir. Bu süre, arayüzün ekrana yüklenmesinden kapatılmasına kadar geçen toplam süredir. Algoritmanın kendisine ait süre bilgileri Resim 7.38'de görülebilir. Resim 7.35'te sistemi zorlayan sınıflar görülmektedir:

Hot Path

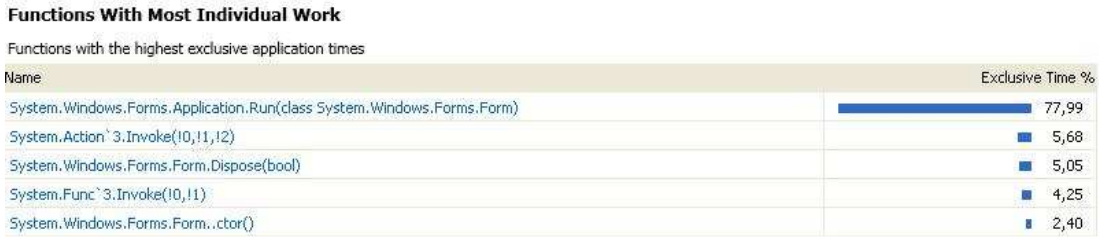
The most expensive call path based on execution times

Function Name	Elapsed Inclusive Time %	Elapsed Exclusive Time %
kriptografi.Program.Main(string[])	100,00	0,20
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	**84,55**	**77,99**
System.Action`3.Invoke(!0,!1,!2)	**5,68**	**5,68**
kriptografi.sfr..ctor()	**4,61**	**0,93**
System.Func`3.Invoke(!0,!1)	**4,25**	**4,25**

Resim 7.35. En çok süre harcayan sınıf ve fonksiyonlar

Resim 7.35, işlem süresi en uzun olan ve sistemi zorlayan sınıf ve fonksiyonları göstermektedir. Satırın üzerine tıklanınca detaylı bilgi veren ekrandan fonksiyonların ve olayların kendileri için kullandıkları işlem süreleri görülebilir.

Resim 7.36'da işlemler esnasında en yüksek işlem süresine sahip fonksiyonlar görülmektedir:



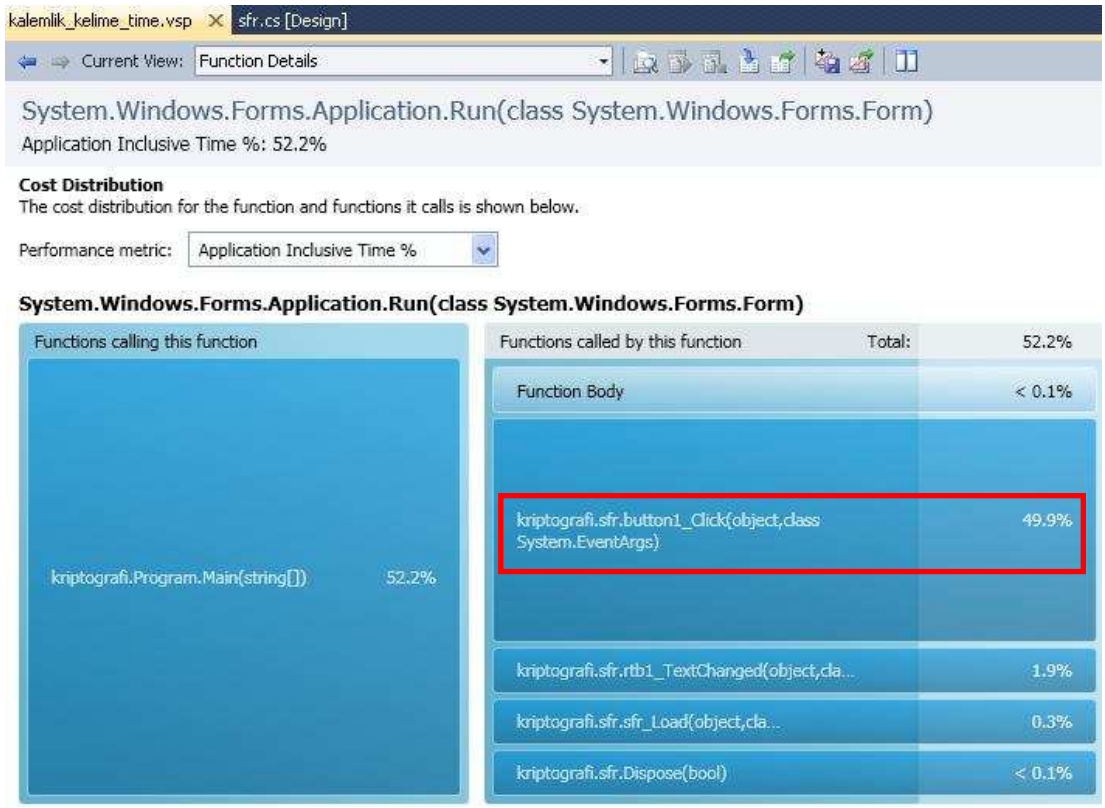
Functions With Most Individual Work

Functions with the highest exclusive application times

Name	Exclusive Time %
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	77,99
System.Action`3.Invoke(!0,!1,!2)	5,68
System.Windows.Forms.Form.Dispose(bool)	5,05
System.Func`3.Invoke(!0,!1)	4,25
System.Windows.Forms.Form..ctor()	2,40

Resim 7.36. En yüksek işlem süresine sahip fonksiyonlar

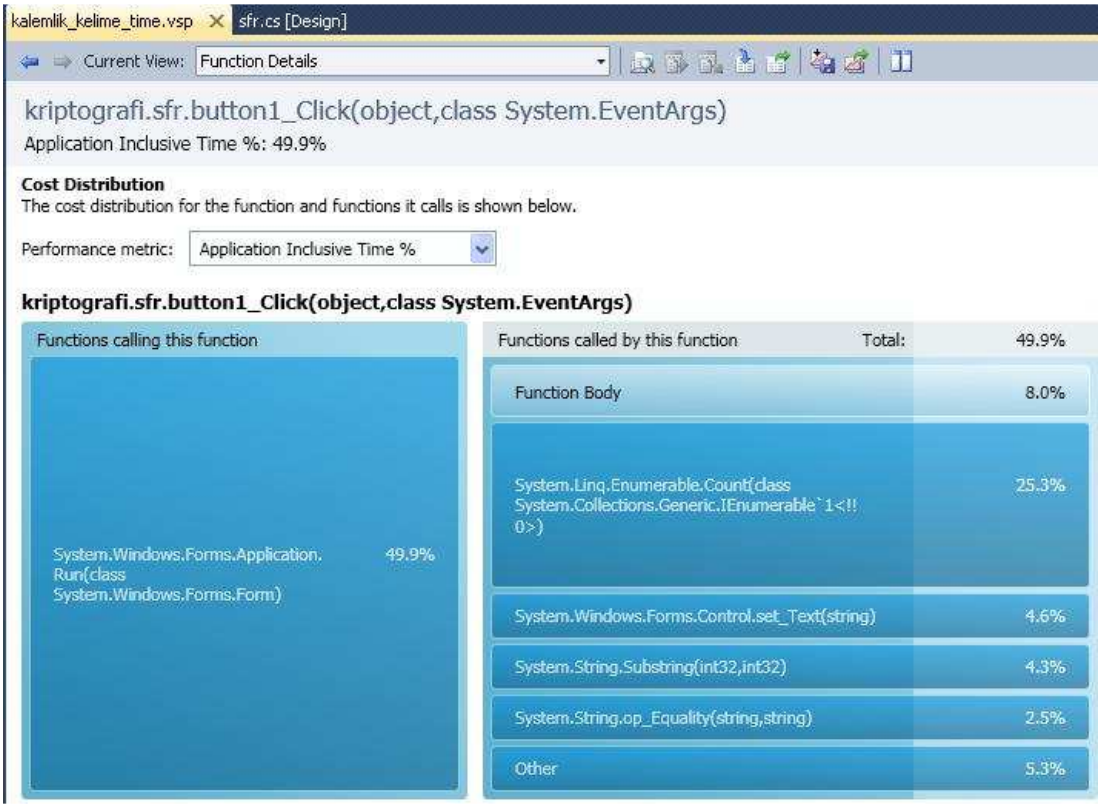
Resim 7.36'da görülen Application Run metodunun işlem süresi %77,90 oranındayken diğer fonksiyonların işlem süreleri daha küçüktür. Application Run metoduna ait detaylı bilgiler Resim 7.37 ve Resim 7.38'de görülmektedir:



Resim 7.37. Application Run metodu için gerekli işlem zamanı

Application Run metodu toplam işlem zamanının %52,2'sine sahiptir. Sağ sütunda çerçeve içine alınmış *button1_Click* olayı şifreleme algoritmasının çalışmasını sağlayan olaydır. Dolayısıyla şifreleme algoritmasının işlem

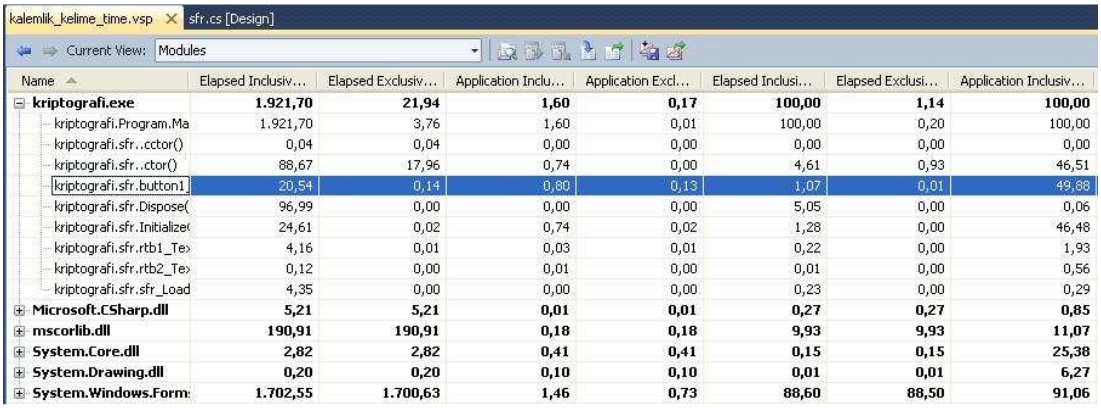
zamanı %49,9 olarak tespit edilmiştir. Resim 7.38 *button1_Click* olayına ait fonksiyon ve olayları göstermektedir:



Resim 7.38. Şifreleme işleminde kullanılan fonksiyon ve olayların kendilerine ait işlem zamanı süreleri (% olarak)

Resim 7.38'den görüldüğü üzere algoritma sıralama işlemleri ve string denilen metin işlemlerinden oluşmaktadır. Sıralama işlemlerinde sayma, indis değerlerin hesaplanması, yer değiştirme, yerine koyma gibi işlemler yapılmaktadır. String işlemlerde ise sabit bir karakterin(harfin) aratılması, eşleştirilmesi, harf olup olmadığının kontrol edilmesi gibi işlemler bulunmaktadır.

Resim 7.39'da şifreleme işlemlerine ait (*button1_Click*) işlem zamanı değerleri milisaniye ve % olarak görülmektedir.



kalemlik_kelime_time.vsp | sfr.cs [Design]

Current View: Modules

Name	Elapsed Inclusiv…	Elapsed Exclusiv…	Application Inclu…	Application Excl…	Elapsed Inclusi…	Elapsed Exclusi…	Application Inclusiv…
kriptografi.exe	**1.921,70**	**21,94**	**1,60**	**0,17**	**100,00**	**1,14**	**100,00**
kriptografi.Program.Ma	1.921,70	3,76	1,60	0,01	100,00	0,20	100,00
kriptografi.sfr..cctor()	0,04	0,04	0,00	0,00	0,00	0,00	0,00
kriptografi.sfr..ctor()	88,67	17,96	0,74	0,00	4,61	0,93	46,51
kriptografi.sfr.button1	20,54	0,14	0,80	0,13	1,07	0,01	49,88
kriptografi.sfr.Dispose(	96,99	0,00	0,00	0,00	5,05	0,00	0,06
kriptografi.sfr.Initialize	24,61	0,02	0,74	0,02	1,28	0,00	46,48
kriptografi.sfr.rtb1_Te	4,16	0,01	0,03	0,01	0,22	0,00	1,93
kriptografi.sfr.rtb2_Te	0,12	0,00	0,01	0,00	0,01	0,00	0,56
kriptografi.sfr.sfr_Load	4,35	0,00	0,00	0,00	0,23	0,00	0,29
Microsoft.CSharp.dll	**5,21**	**5,21**	**0,01**	**0,01**	**0,27**	**0,27**	**0,85**
mscorlib.dll	**190,91**	**190,91**	**0,18**	**0,18**	**9,93**	**9,93**	**11,07**
System.Core.dll	**2,82**	**2,82**	**0,41**	**0,41**	**0,15**	**0,15**	**25,38**
System.Drawing.dll	**0,20**	**0,20**	**0,10**	**0,10**	**0,01**	**0,01**	**6,27**
System.Windows.Form	**1.702,55**	**1.700,63**	**1,46**	**0,73**	**88,60**	**88,50**	**91,06**

Resim 7.39. Şifreleme algoritmasının işlem zamanı için milisaniye ve % değerleri

Resim 7.39'da görülen ilk dört sütun işlem zamanını milisaniye olarak göstermektedir. Son sütunlar ise %'lik değerleri vermektedir. Şifreleme algoritması için tespit edilen değerler arka zemini koyu renkle gösterilmiştir. İşlem zamanı %49,88≈%49,9 ve 0,80 milisaniye olarak tespit edilmiştir.

Çizelge 7.3'te performans analiz değerlerinden elde edilen değerler(işlem zamanı, bellek kullanımı, CPU kullanımı) yer almaktadır.

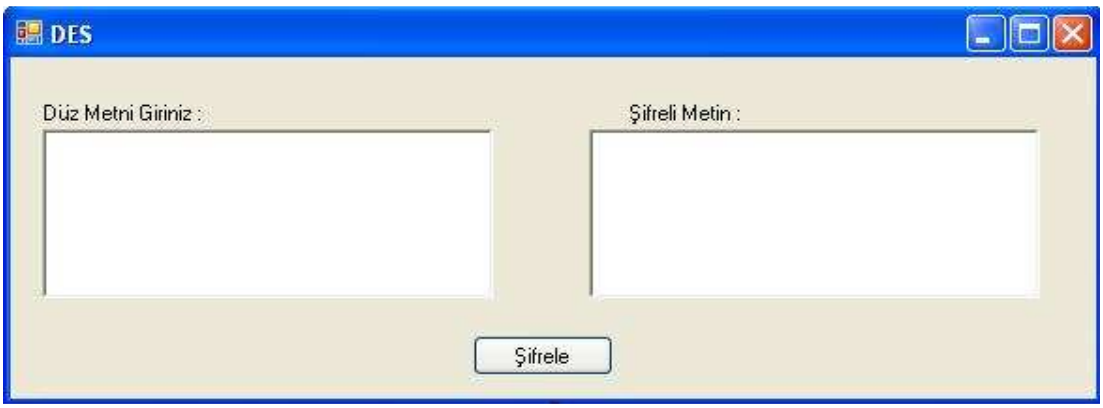
Çizelge 7.3. Geliştirilen algoritma ile 8 byte'lık metnin(kalemlik) kelime düzeyinde şifrelenmesine ait performans değerleri

	İşlem zamanı (msec)	**Hafıza Kullanımı (Byte %)**	**CPU Kullanımı (%)**
Geliştirilen algoritma (kelime düzeyinde şifreleme)	0,8	20,7	3,7

7.2. DES Şifreleme Algoritması Performans Analiz Değerleri

Performans analiz değerleri, *kalemlik* kelimesi kullanılarak elde edilmiştir. Geliştirilen algoritmada paragraf düzeyinde ve kelime düzeyinde şifreleme işlemleri ile DES algoritması ile şifreleme işlemleri için aynı kelime ve aynı bilgisayar donanım-yazılımı kullanılmıştır. DES algoritması için daha önceden başka çalışmalarda elde edilen sonuçlar kullanılmamıştır. Kullanılan

bilgisayar donanım ve yazılım özellikleri performans analizini etkilediği için daha sağlıklı analiz değerleri elde etmek adına DES algoritmasının, C#'de kaynak kodlarını içeren küçük bir Windows Form hazırlanmış ve analiz değerleri bu form aracılığı ile elde edilmiştir. Resim 7.40, hazırlanan formu göstermektedir.

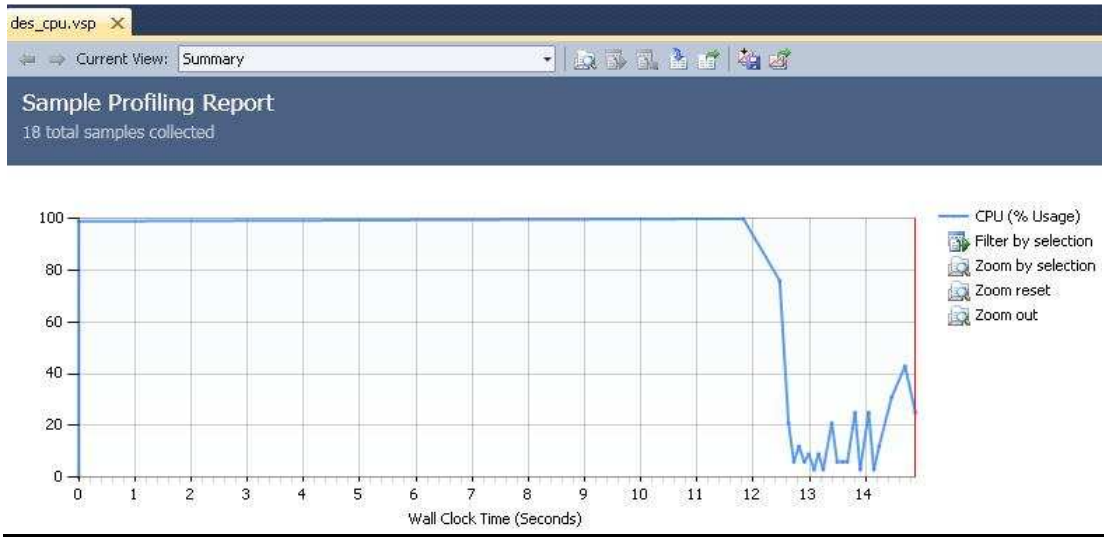


Resim 7.40. DES şifreleme algoritması windows formu

Resim 7.40'ta görülen formda, iki adet richtextbox, iki adet label ve bir adet buton nesneleri kullanılmıştır. Sol taraftaki richtextboxa girilen metine, *şifrele* butonu ile DES şifreleme algoritmasının işlem basamakları uygulanır. Metinin şifreli hali sağ taraftaki richtextboxa aktarılır.

CPU sampling verileri

DES algortimasının CPU kullanım grafiği Resim 7.41'de görülmektedir.



Resim 7.41. DES algoritması ile şifreleme işlemi CPU grafiği

Resim 7.41'de görüldüğü üzere CPU kullanımı program ilk çalıştırıldığında %100'e ulaşmakta, daha sonra şifreleme işlemlerinde kullanılan fonksiyon ve sınıflara göre (metin kutularının içeriğinin değişmesi, butonlara tıklanması, döngülerin çalıştırılması gibi) değişkenlik göstermektedir.

Resim 7.42, DES şifreleme algoritması çalışırken sistemi en çok zorlayan sınıf ve fonksiyonları göstermektedir.

Hot Path

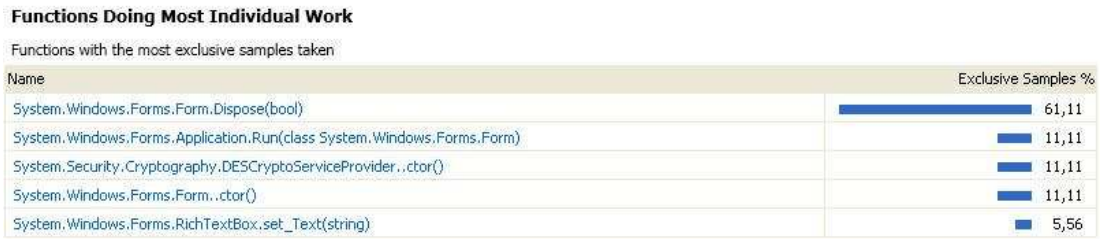
The most expensive call path based on sample counts

Function Name	Inclusive Samples %	Exclusive Samples %
des.exe	100,00	0,00
des.Program.Main()	100,00	0,00
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	**88,89**	**11,11**
des.Form1..ctor()	**11,11**	**0,00**

Resim 7.42. Sistemi en çok zorlayan sınıflar

Hot Path bölümünde *Inclusive Samples* yüzdesi yüksek olan sınıflar sistemi en çok zorlayan sınıflardır. *Exclusive Samples* bölümünde ise performans darboğazı yaratan fonksiyonlar en yüksek yüzdeye sahiptir [68]. Bu fonksiyonları daha detaylı görebilmek için yanlarında ateş simgesi bulunan yazıların üzerine tıklamak yeterlidir ya da Resim 7.41'de görülen *Current View* kısmında *Function Details* seçeneği ile detaylara ulaşmak mümkündür.

Resim 7.43'te CPU kullanımında fonksiyonların bireysel olarak oluşturdukları zorlukların % değerleri görülmektedir.



Functions Doing Most Individual Work

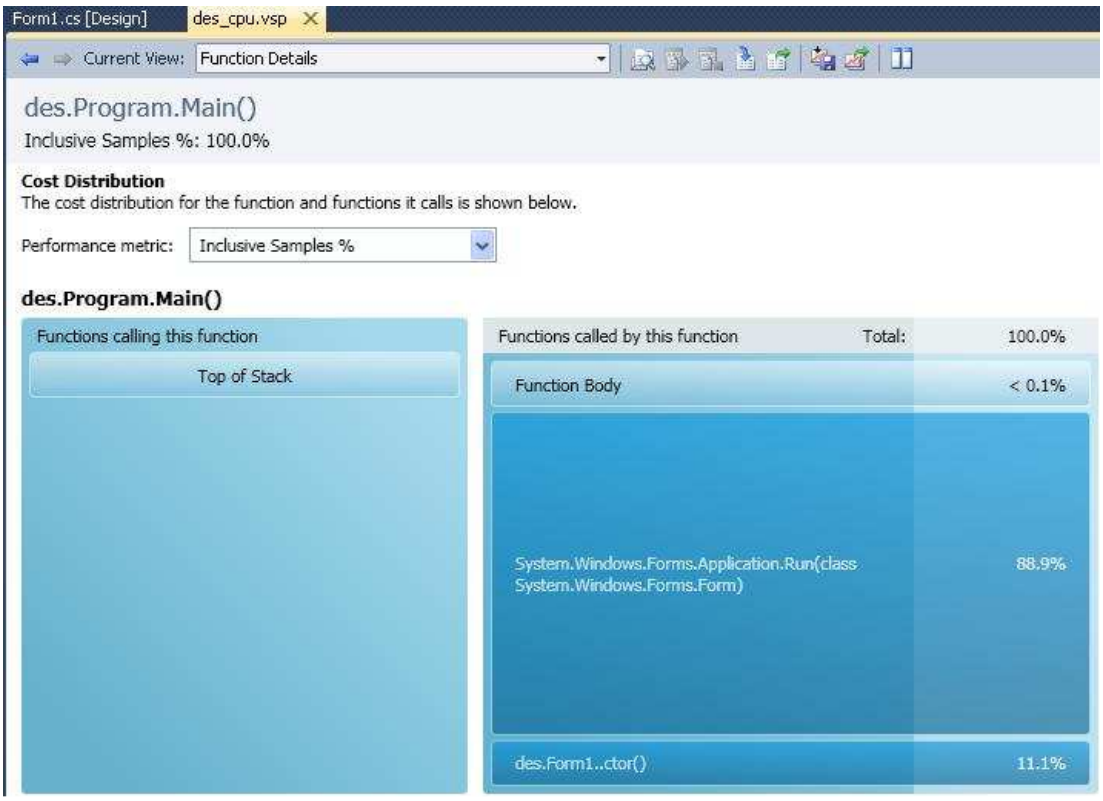
Functions with the most exclusive samples taken

Name	Exclusive Samples %
System.Windows.Forms.Form.Dispose(bool)	61,11
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	11,11
System.Security.Cryptography.DESCryptoServiceProvider..ctor()	11,11
System.Windows.Forms.Form..ctor()	11,11
System.Windows.Forms.RichTextBox.set_Text(string)	5,56

Resim 7.43. Sistemi en çok zorlayan fonksiyonlar

Resim 7.43'te görülen fonksyonlar, CPU kullanımında *exclusive samples* oranları kadar darboğaz oluşturmaktadır.

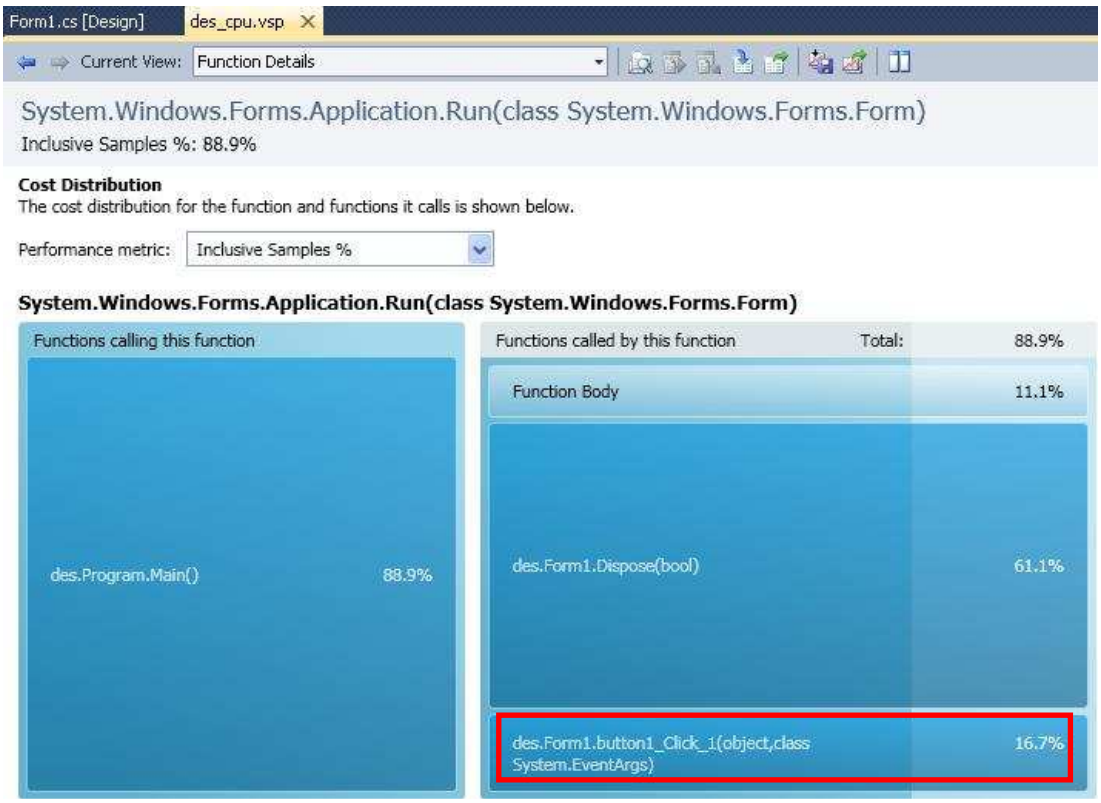
Resim 7.44, programa ait CPU kullanım detaylarını göstermektedir.



Resim 7.44. DES algoritması için hazırlanan programa ait CPU kullanımları

Şifreleme algoritmasının çalıştırıldığı *button1_Click* olayı Application Run sınıfına aittir. Resim 7.44'te görüldüğü üzere Application Run sınıfının CPU kullanım oranı programın toplam CPU kullanımının %88,9'udur. *des.Form1..ctor()* ile gösterilen %'lik değer form üzerindeki nesnelerin oluşturulmasına ait CPU kullanım değeridir.

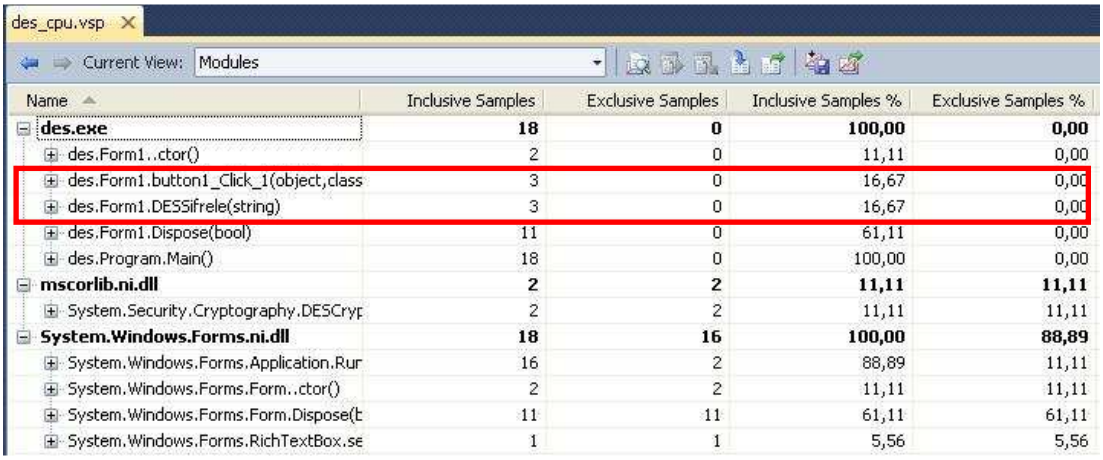
Resim 7.45'te şifreleme işleminin gerçekleştiği *button1_Click* olayına ait CPU kullanımı durumu görülmektedir.



Resim 7.45. Şifreleme algoritmasına ait CPU kullanım oranı

Resim 7.45'te çerçeve ile gösterilen *button1_click* olayı şifreleme algoritmasının CPU kullanım oranını göstermektedir. Programın toplam CPU kullanımının % 16,7'sinin şifreleme algoritmasına ait olduğu tespit edilmiştir.

Resim 7.46'da CPU kullanımına ilişkin detaylar görülmektedir.



des_cpu.vsp

Current View: Modules

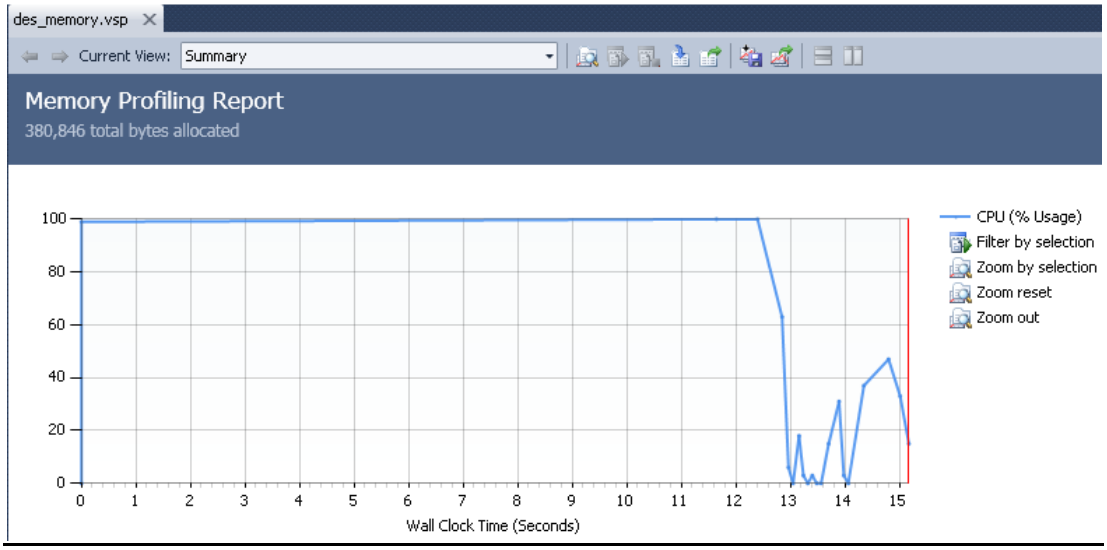
Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
des.exe	**18**	**0**	**100,00**	**0,00**
des.Form1..ctor()	2	0	11,11	0,00
des.Form1.button1_Click_1(object,class	3	0	16,67	0,00
des.Form1.DESSifrele(string)	3	0	16,67	0,00
des.Form1.Dispose(bool)	11	0	61,11	0,00
des.Program.Main()	18	0	100,00	0,00
mscorlib.ni.dll	**2**	**2**	**11,11**	**11,11**
System.Security.Cryptography.DESCryp	2	2	11,11	11,11
System.Windows.Forms.ni.dll	**18**	**16**	**100,00**	**88,89**
System.Windows.Forms.Application.Run	16	2	88,89	11,11
System.Windows.Forms.Form..ctor()	2	2	11,11	11,11
System.Windows.Forms.Form.Dispose(b	11	11	61,11	61,11
System.Windows.Forms.RichTextBox.se	1	1	5,56	5,56

Resim 7.46. CPU kullanımına ait detaylar

Resim 7.46'da çerçevelenmiş bölgede iki farklı olay vardır. Aslında *DESSifrele(string)* bir alt programdır. *button1_click()* olayı ile bu alt program çağırılır ve şifreleme işlemi gerçekleştirilir. Dolayısıyla sayısal değerlerin Resim 7.46'da aynı olduğu görülmektedir. DES şifreleme algoritmasının CPU kullanım oranı %16,67≈%16,7 olarak tespit edilmiştir.

.Net allocation memory verileri

Resim 7.47, DES şifreleme algoritması için hazırlanan programın toplam bellek kullanımına ait grafiği göstermektedir. CPU kullanım grafiği ile aynıdır. Farkı, bu kısımda veriler *Byte* ve % olarak verilmiştir. CPU verilerinin aktarıldığı bir önceki bölümde veriler sadece % olarak verilmiştir.



Resim 7.47. DES şifreleme algoritması bellek kullanımı

Resim 7.47'de *memory profiling report* kısmında görüldüğü üzere program başlama ve bitiş süreleri arasında 380,846 Byte'lık bellek kullanmıştır. Resim 7.48, program içerisinde en çok bellek kullanımına sahip fonksiyonları göstermektedir.

Functions Allocating Most Memory

Functions with the highest exclusive bytes allocated

Name	Bytes %
System.Windows.Forms.Form..ctor()	33,26
System.Windows.Forms.Form.Dispose(bool)	27,36
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	27,03
System.Windows.Forms.RichTextBox..ctor()	3,88
System.Windows.Forms.RichTextBox.set_Text(string)	2,27

Resim 7.48. Bellek tüketimi en fazla olan fonksiyonlar

Resim 7.48'deki veriler incelendiğinde CPU üzerinde en fazla yük oluşturan fonksiyonların aynı zaman da bellek kullanımının en fazla olduğu görülmektedir. Resim 7.48'de görülen fonksiyonlar aynı zamanda sistemde darboğaz problemi oluşturan fonksiyonlardır. Satırlara tek tek tıklanarak bellek kullanımını arttıran fonksiyonlar hakkında detaylı verilere ulaşılabilir.

Resim 7.49'da bellek kullanımı en yüksek olan veri tipleri görülmektedir.

Types With Most Memory Allocated

Types with the hightest total number of bytes allocated

Name	Bytes %
System.String	25,62
System.Byte[]	13,05
System.Char[]	7,55
System.Version	4,30
System.Collections.Generic.List`1	2,07

Resim 7.49. Bellek kullanımı en yüksek olan veri tipleri

Resim 7.49'da görülen veri tipleri aynı zamanda programda en çok kullanılan veri tipleridir. Programda en çok string veriler kullanılmıştır. Programın bellek kullanımına ait detaylı veriler Resim 7.50'de görülmektedir.

des_memory.vsp

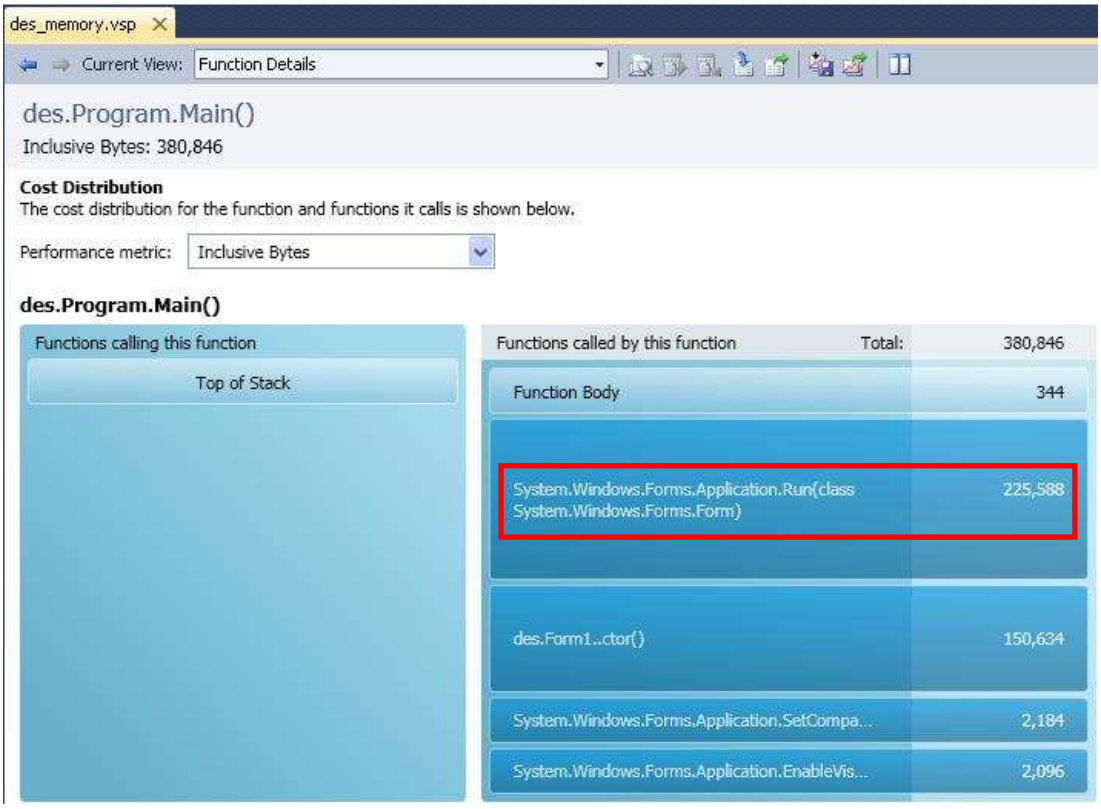
Current View: Modules

Name	Inclusive Allocations	Exclusive Allocations	Inclusive Bytes	Exclusive Bytes
des.exe	**8.040**	**23**	**380.846**	**1.788**
des.Form1..ctor()	2.411	9	150.634	304
des.Form1.button1_Click_1(obje	112	0	18.422	0
des.Form1.Byte8(string)	4	2	96	40
des.Form1.DESSifrele(string)	101	4	17.888	220
des.Form1.Dispose(bool)	2.603	0	104.216	0
des.Form1.InitializeComponent()	554	7	23.668	880
des.Program.Main()	8.040	1	380.846	344
mscorlib.ni.dll	**66**	**66**	**8.084**	**8.084**
System.Configuration.ni.dll	**0**	**0**	**0**	**0**
System.Drawing.ni.dll	**0**	**0**	**0**	**0**
System.ni.dll	**0**	**0**	**0**	**0**
System.Windows.Forms.ni.dll	**8.023**	**7.951**	**379.318**	**370.974**
System.Xml.ni.dll	**0**	**0**	**0**	**0**

Resim 7.50. Fonksiyon ve sınıfların bellek kullanımına ait detaylar

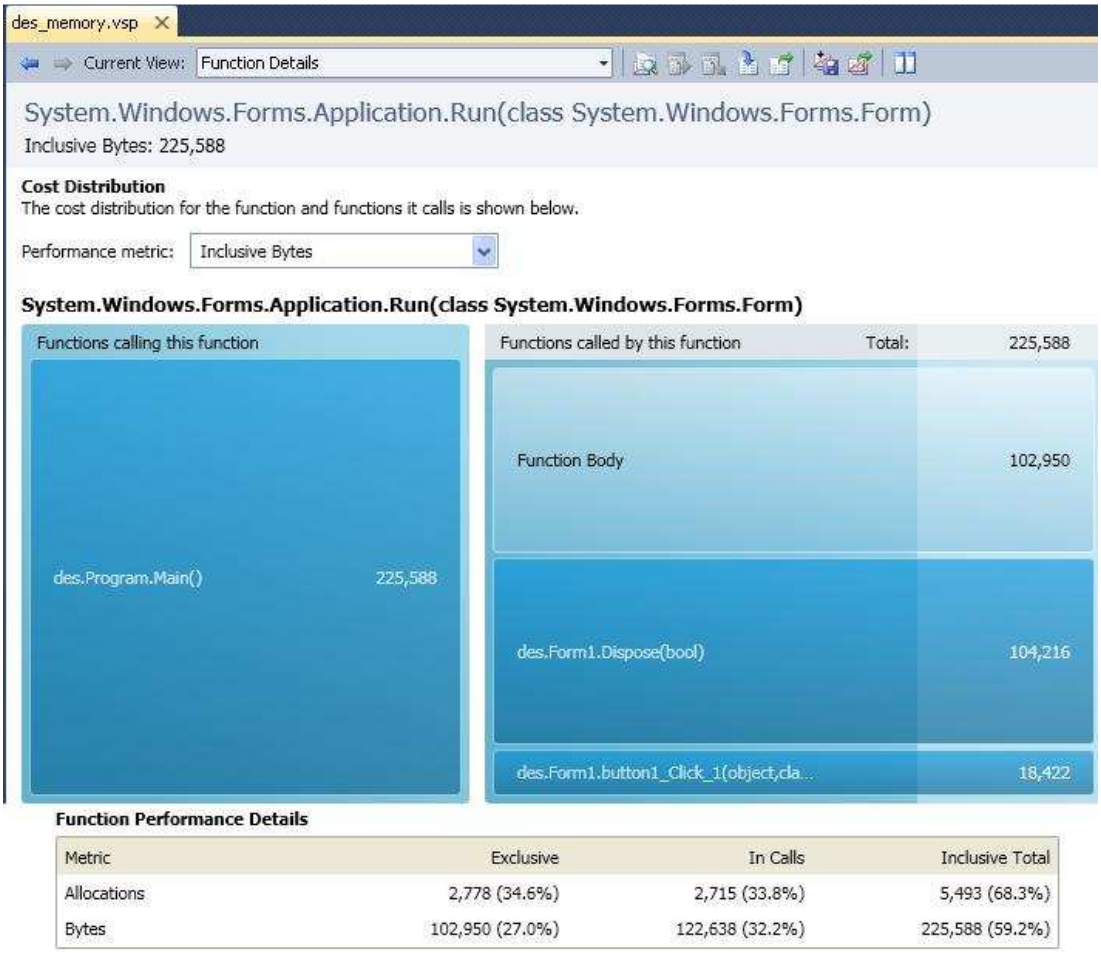
Resim 7.50'den görüldüğü üzere program toplamda 380,846 byte bellek kullanmaktadır. Bu miktarın yalnızca 18,422 byte'lık kısmının şifreleme algoritmasını çalıştırılmak için kullanıldığı tespit edilmiştir.

Resim 7.51, programın bellek kullanımı ile ilgili detaylı verileri göstermektedir.



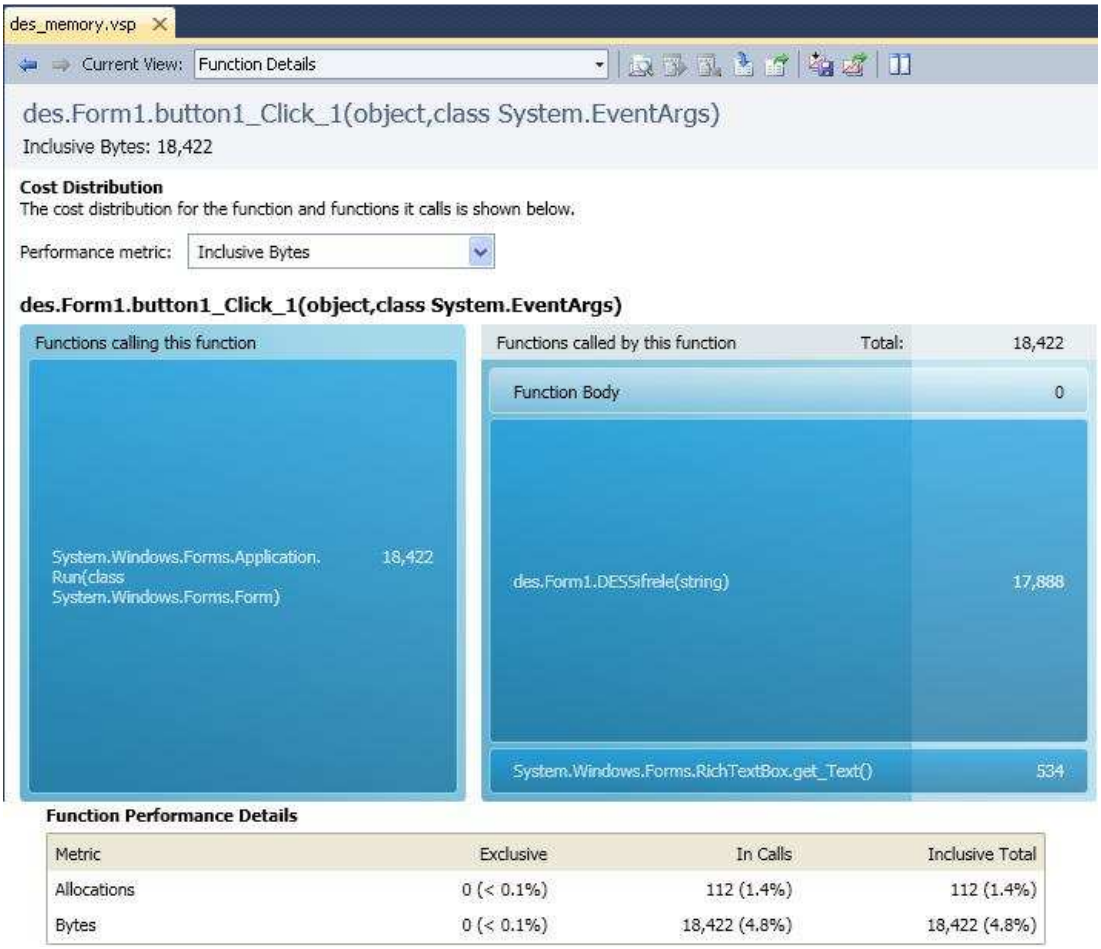
Resim 7.51. Programın bellek kullanım durumu detayı

Resim 7.51'den görüldüğü üzere 380,846 Byte'lık bellek tüketiminin 225, 588 Byte'lık kısmı Application Run sınıfına aittir. Application Run sınıfı aynı zamanda şifreleme işlemlerinin gerçekleştirilmesini sağlayan *button1_click()* olayını da içermektedir. Application Run sınıfına ait detaylı bilgi Resim 7.52'de, *Button1_click()* olayına ait detaylı veriler Resim 7.53'te görülmektedir.



Resim 7.52. Application Run sınıfına ait bellek kullanımı detayları

Application Run sınıfının byte cinsinden bellek tüketimi incelendiğinde, programın toplam bellek kullanımının % 59,2'sine sahip olduğu görülmüştür.



Resim 7.53. Button1_click() olayına ait bellek kullanımı detayları

Button1_Click() olayının byte cinsinden bellek tüketimi incelendiğinde, programın toplam bellek kullanımının % 4,8'ine sahip olduğu görülmüştür.

İşlem zamanı verileri

Resim 7.54, işlem zamanına ait grafiği göstermektedir. *kalemlik* kelimesinin şifrelenmesi için geçen işlem zamanı süresince algoritmanın içermiş olduğu işlemlere göre CPU kullanımında dalgalanmalar oluşmaktadır.

Resim 7.54. DES algoritması işlem zamanı-CPU grafiği

Resim 7.54'te görüldüğü gibi geliştirilen şifreleme algoritmasının şifreleme işlemini tamamlama süresi 1,718,99 milisaniye olarak tespit edilmiştir. Bu süre, arayüzün ekrana yüklenmesinden kapatılmasına kadar geçen toplam süredir. Resim 7.55'de sistemi zorlayan sınıflar görülmektedir:

Hot Path

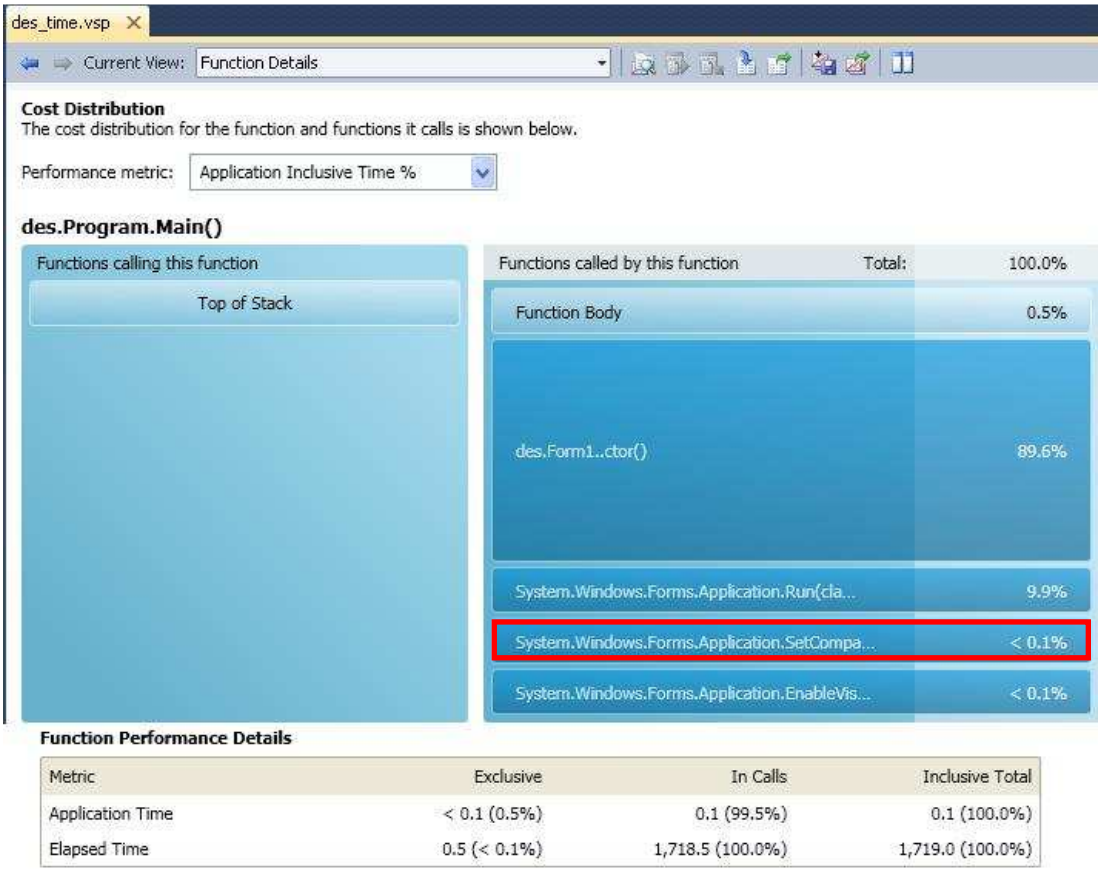
The most expensive call path based on execution times

Function Name	Elapsed Inclusive Time %	Elapsed Exclusive Time %
des.exe	100,00	0,00
des.Program.Main()	100,00	0,03
System.Windows.Forms.Application.Run(class System.Windows.Forms.Form)	95,82	85,58
des.Form1.Dispose(bool)	**5,86**	**0,00**
des.Form1.button1_Click_1(object,class System.EventArgs)	**4,38**	**0,08**

Resim 7.55. Sistemi en çok kullanan sınıflar

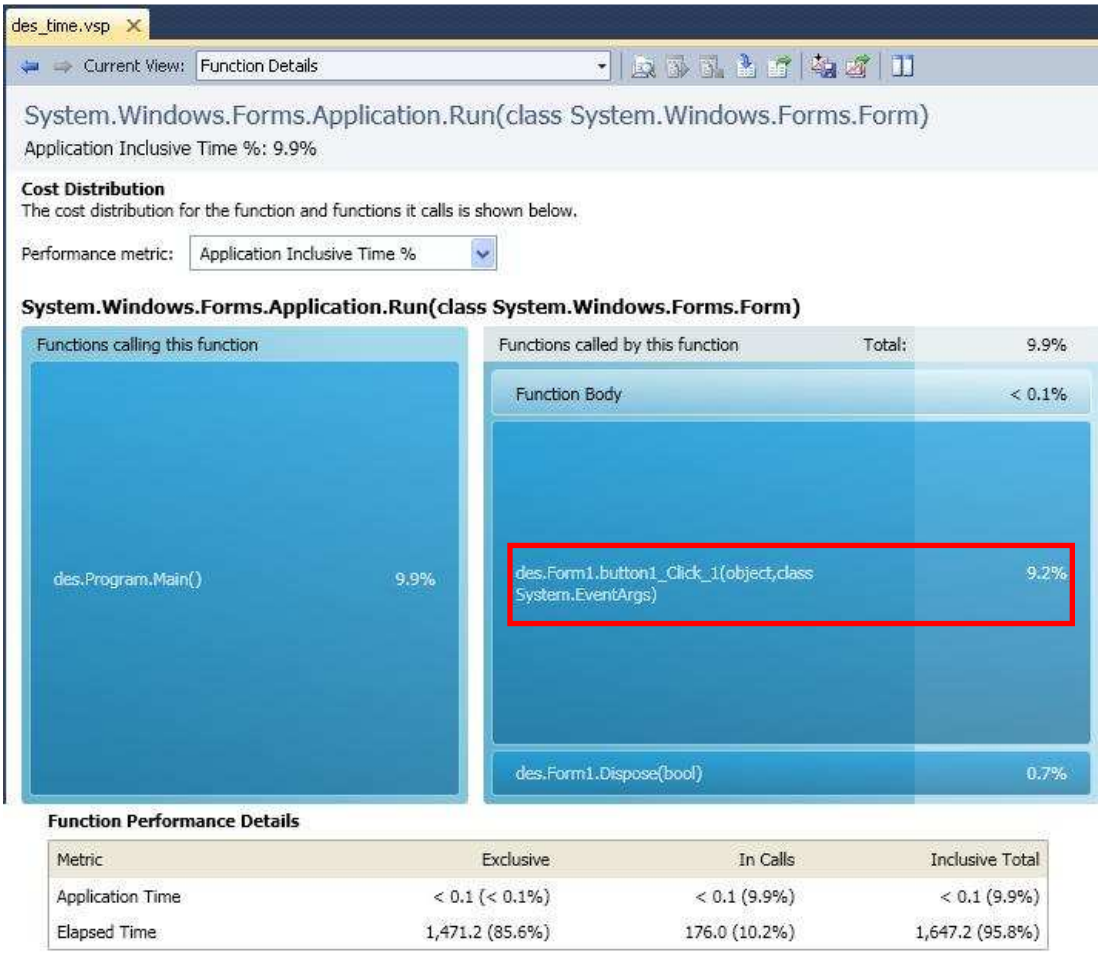
Resim 7.55'te işlem süresi en uzun olan sınıflar ve %'lik değerleri görülmektedir.

Detaylı bilgiler Resim 7.56, Resim 7.57 ve Resim 7.58'de görülmektedir:



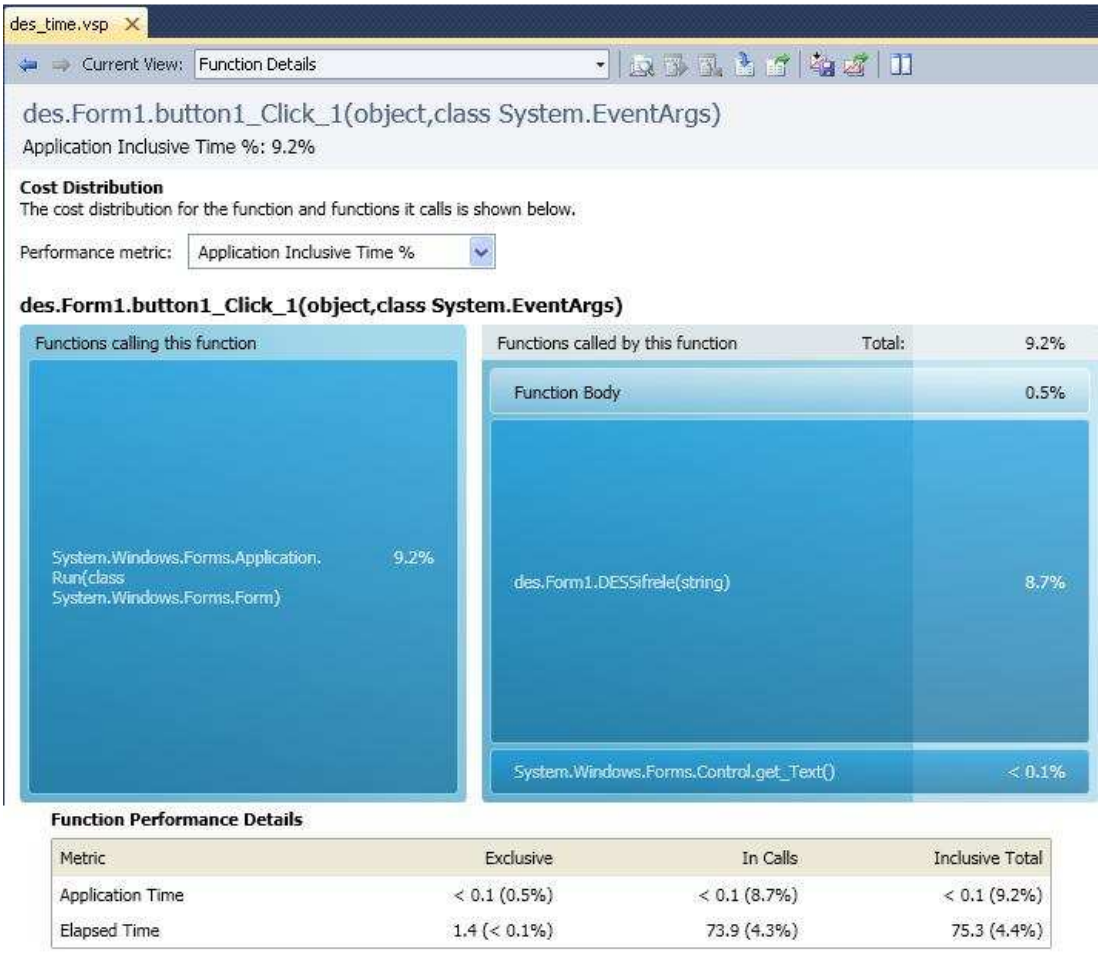
Resim 7.56. DES algoritması için hazırlanan programın süre kullanım detayı

Resim 7.56'da DES algoritmasının similasyonu için hazırlanan tek bir formdan oluşan programa ait işlem zamanı süre bilgileri görülmektedir. Programın tamamının süre kullanımı %100 olarak görülmekte ve alt sınıflara ait %'lik değerler Resim 7.56'da görülmektedir. Çerçeve içine alınan Application Run sınıfının işlem zamanı süresi toplam süre kullanımının %9,9'unu oluşturmaktadır. Resim 7.57'de detaylar görülmektedir.



Resim 7.57. Application Run sınıfına ait işlem zamanı süre bilgisi detayı

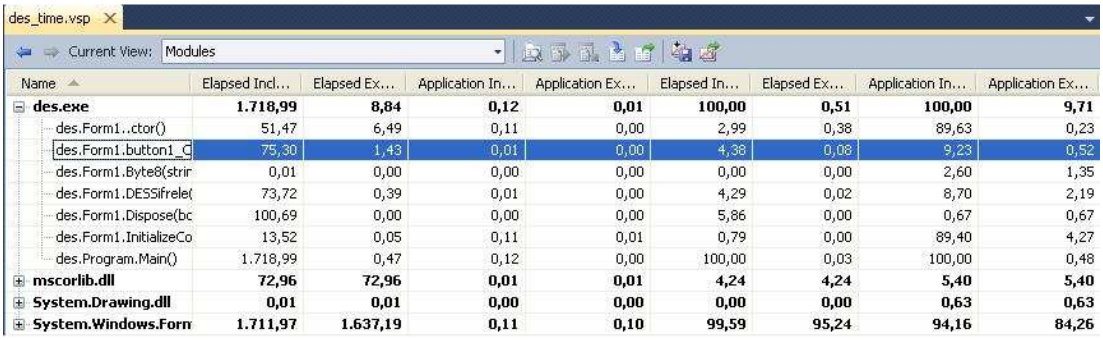
Application Run metodu toplam işlem zamanının %9,9'una sahiptir. Sağ sütunda çerçeve içine alınmış *button1_Click* olayı şifreleme algoritmasının çalışmasını sağlayan olaydır. Dolayısıyla şifreleme algoritmasının işlem zamanı %9,2 olarak tespit edilmiştir. Resim 7.58, *button1_Click* olayına ait fonksiyon ve olayların süre kullanım detaylarını göstermektedir:



Resim 7.58. Button1_Click() olayına ait (şifreleme algoritmasına ait) işlem zamanı süre kullanım bilgisi detayı (% cinsinden)

Resim 7.58'de görüldüğü üzere şifreleme algoritmasının uygulama zamanı %9,2 olarak tespit edilmiştir.

Resim 7.59'da şifreleme işlemlerine ait (*button1_Click*) işlem zamanı değerleri milisaniye ve % olarak görülmektedir.


des_time.vsp

Current View: Modules

Name	Elapsed Incl...	Elapsed Ex...	Application In...	Application Ex...	Elapsed In...	Elapsed Ex...	Application In...	Application Ex...
des.exe	**1.718,99**	**8,84**	**0,12**	**0,01**	**100,00**	**0,51**	**100,00**	**9,71**
des.Form1..ctor()	51,47	6,49	0,11	0,00	2,99	0,38	89,63	0,23
des.Form1.button1_C	75,30	1,43	0,01	0,00	4,38	0,08	9,23	0,52
des.Form1.Byte8(strir	0,01	0,00	0,00	0,00	0,00	0,00	2,60	1,35
des.Form1.DESSifrele(	73,72	0,39	0,01	0,00	4,29	0,02	8,70	2,19
des.Form1.Dispose(bc	100,69	0,00	0,00	0,00	5,86	0,00	0,67	0,67
des.Form1.InitializeCo	13,52	0,05	0,11	0,01	0,79	0,00	89,40	4,27
des.Program.Main()	1.718,99	0,47	0,12	0,00	100,00	0,03	100,00	0,48
mscorlib.dll	**72,96**	**72,96**	**0,01**	**0,01**	**4,24**	**4,24**	**5,40**	**5,40**
System.Drawing.dll	**0,01**	**0,01**	**0,00**	**0,00**	**0,00**	**0,00**	**0,63**	**0,63**
System.Windows.Forn	**1.711,97**	**1.637,19**	**0,11**	**0,10**	**99,59**	**95,24**	**94,16**	**84,26**

Resim 7.59. Şifreleme algoritmasının işlem zamanı için milisaniye ve % değerleri

Resim 7.59'da görülen ilk dört sütun işlem zamanını milisaniye olarak göstermektedir. Son sütunlar ise %'lik değerleri vermektedir. Şifreleme algoritması için tespit edilen değerler arka zemini koyu renkle gösterilmiştir. İşlem zamanı %9,23≈%9,2 ve 0,01 milisaniye olarak tespit edilmiştir.

Çizelge 7.4'te performans analiz değerlerinden elde edilen değerler(işlem zamanı, bellek kullanımı, CPU kullanımı) yer almaktadır.

Çizelge 7.4. DES algoritması ile 8 byte'lık metnin(kalemlik) şifrelenmesine ait performans değerleri

	İşlem zamanı (msec)	**Hafıza Kullanımı (Byte %)**	**CPU Kullanımı (%)**
DES ile şifreleme	0,01	4,8	16,7

7.3. Performans Değerlerinin Karşılaştırılması

DES algoritması modern olarak geliştirilen ve kabul edilen ilk algoritma olması özelliğinden dolayı seçilmiştir. Klasik teknikler kullanılarak geliştirilen algoritma ile işlemlerin gerçekleştirilmesinde benzerlik bulunmamasına rağmen işlem basamakları arasında benzerlik bulunmaktadır. DES algoritması bitlerle (1 ve 0) işlem yaparken, klasik yöntemler kullanılarak geliştirilen algoritma harflerin sıra numaraları ve harflerin kendisini kullanmaktadır.

Performans değerlerinin karşılaştırılmasında sağlıklı sonuçlar elde edebilmek için, her üç algoritmada 8 Byte'lık(64 bit) *kalemlik* kelimesi ile çalışılmıştır. Farklı kelimeler için farklı sonuçlar elde edilmektedir. Bu durumun, harflerin sayısal karşılıkları, indis değerleri ve modüler aritmetik işlemlerinden kaynaklanabileceği düşünülmektedir. Bölüm 7'de Çizelge 7.1'de performans değerlerini etkileyen faktörler verilmiştir. Bölüm 7'deki Çizelge 7.1'e göre, *girdi verisinin büyüklüğü* performans analizinde etkili bir değerdir. Bu nedenle sabit uzunlukta ve sabit bir kelime seçilmiştir. Her algoritma için 8 Byte'lık farklı kelimeler kullanılmamıştır, çünkü kelimelerdeki harflerin sayısal karşılıkları ve indis değerleri de girdi verisinin büyüklüğünü değiştirmektedir. Performans analizi çalışmalarında elde edilen sonuçlar değişkenlik gösterdiği için sabit bir kelime üzerinde çalışılması tercih edilmiştir.

Algoritmaların üçü de aynı bilgisayarda ve aynı yazılım ortamında test edilmiştir. Bölüm 7, Çizelge 7.1'de performansı etkileyen dış faktörler arasında *bilgisayarın hızı* ve *derleyicinin kalitesi* görülmektedir. Bu faktörlerin şifreleme işlemlerine olan etkisini sabit tutabilmek için algoritmaların üçü aynı ortamda test edilmiştir. Çizelge 7.5'te geliştirilen algoritmanın paragraf ve kelime düzeyinde şifreleme işlemleri ve DES algoritması ile şifreleme işlemine ait elde edilen değerler görülmektedir. Performans değerleri, işlem zamanı(milisaniye(msec)), memory (bellek kullanımı) ve CPU(işlemci iş yükü) olarak gösterilmiştir.

Çizelge 7.5'te, Bölüm 7'de verilmiş olan Çizelge 7.2, Çizelge 7.3 ve Çizelge 7.4'te gösterilen performans analizi sonuçları yer almaktadır.

Çizelge 7.5. Şifreleme işlemi için algoritmaların performans analiz değerleri

	İşlem zamanı (msec)	Memory (Byte %)	CPU (%)
Paragraf düzeyinde şfreleme	0,8	17,5	9,1
Kelime düzeyinde şifreleme	0,8	20,7	3,7
DES algoritması ile şifreleme	0,01	4,8	16,7

Çizelge 7.5'te *işlem zamanı* sütunundan görüldüğü üzere geliştirilen algoritmanın paragraf ve kelime düzeyinde şifreleme işlemlerini tamamlaması için geçen işlem zamanı süresi 0,8 milisaniye olarak tespit edilmiştir. Paragraf ve kelime düzeyinde şifreleme işlemleri için işlem zamanı açısından bir fark bulunmamıştır. Daha uzun verilerle çalışıldığında paragraf ve kelime düzeyinde şifreleme işlemlerinde farklılık tespit edilmiştir. DES algoritmasının işlem zamanı süresi ise 0,01 milisaniye olarak tespit edilmiştir. DES algoritması, klasik teknikler kullanılarak geliştirilen algoritmaya göre daha hızlı çalışmaktadır. Bunun sebebi DES algoritmasının bitlerle(1 ve 0) çalışması olabilir. Bilgisayar sisteminde de işlemler ikilik sistemde (1 ve 0) yapılır. Bu durum DES algoritması için işlem zamanı anlamında kazanç sağlamakta ve sürenin kısalmasına katkı sağlamaktadır. Bahçetepe'ye göre (2006), performans açısından hızlı çalışan bir algoritma, kullanım için daha idealdir. Bu durumda karşılaştırma sonuçlarına göre, işlem zamanı açısından bakıldığında DES algoritması performans sıralamasında birinci sırada, geliştirilen algoritma paragraf ve kelime düzeyinde şifreleme işlemleri ile ikinci sırada yer almaktadır.

Çizelge 7.5'te *Memory* sütunundan görüldüğü üzere geliştirilen algoritmanın paragraf düzeyinde şifreleme işlemlerinin tamamlaması için kullanılan bellek miktarı Byte cinsinden % değerlere göre %17,5 olarak tespit edilmiştir. Kelime düzeyinde şifreleme işlemlerinin tamamlaması için kullanılan bellek miktarı %20,7 olarak tespit edilmiştir. Paragraf ve kelime düzeyinde şifreleme işlemleri bellek kullanım durumları açısından karşılaştırıldığında paragraf düzeyinde şifreleme işleminin daha iyi sonuç verdiği görülmüştür. Bahçetepe (2006), modüler çarpma algoritmaları ve kriptolojide uygulamaları üzerine yazmış olduğu tez çalışmasında bellek tüketimi az olan algoritmanın performansının daha yüksek olduğunu belirtmiştir. Buna dayanarak bellek kullanım performansı açısından, paragraf düzeyinde şifreleme algoritmasının kelime düzeyinde şifreleme algoritmasına göre daha kullanışlı olduğu tespit edilmiştir. Bunun sebebi, kelime düzeyinde şifreleme işlemlerinde

algoritmanın birkaç adımında yapılan değişikliktir. Kelime düzeyinde şifreleme algoritmasının adım sayısı daha fazladır, çünkü her kelimedeki harflerin indis değerleri kelime düzeyinde tutulmaktadır. Paragraf düzeyinde şifrelemede indis değerlerinin tutulması işlemi, paragrafın ilk harfinden son harfine kadar tek bir dizide tutulmaktadır. Kelime düzeyinde şifreleme algoritmasında ise her kelime için indis değerlerini tutan ayrı bir dizi söz konusudur. DES algoritmasının bellek kulanım oranı %4,8 olarak tespit edilmiştir. Yapılan karşılaştırmada bellek kullanımı en düşük çıkan algoritma DES algoritmasıdır. Bu durumda performansı en yüksek olan algoritma DES'tir. Bellek kullanımı açısından performans sıralamasına bakıldığında, DES birinci, paragraf düzeyinde şifreleme ikinci ve kelime düzeyinde şifreleme üçüncü sıradadır. Bu sıralama aynı zamanda bellek kullanımı açısından algoritmaların kullanılabilirliğini göstermektedir.

Çizelge 7.5'te *CPU* sütunundan görüldüğü üzere geliştirilen algoritmanın paragraf düzeyinde şifreleme işlemlerinin tamamlaması için CPU üzerinde oluşturduğu iş yükü miktarı % 9,1 olarak tespit edilmiştir. Kelime düzeyinde şifreleme işlemlerinin tamamlaması için CPU üzerinde oluşturulan iş yükü miktarı %3,7 olarak tespit edilmiştir. DES algoritması için bu oran %16,7'dir. Paragraf ve kelime düzeyinde şifreleme işlemleri CPU üzerindeki iş yükü durumları açısından karşılaştırıldığında kelime düzeyinde şifreleme işleminin daha iyi sonuç verdiği görülmüştür. DES algoritması CPU kullanımı açısından en yüksek değere sahiptir. CPU üzerindeki iş yükü oranlarına göre performansı en iyi olandan en kötü olana doğru sıralama yapıldığında kelime düzeyinde şifreleme algoritması birinci, paragraf düzeyinde şifreleme algoritması ikinci ve DES algoritması üçüncü sıradadır.

Bellek kullanım oranları ve CPU kullanım oranları arasında ters ilişki ortaya çıkmıştır. CPU kullanımı en yüksek olan DES algoritmasının bellek kullanımı en düşüktür. Bu durumun sebebi, algoritmaların kaynak kodlarındaki adım ve işlem sayısı olabilir. Daha az kod satırı ile daha çok iş yapılması bu durumun temel nedeni olabilir.

Çizelge 7.6'da, performans değerlendirmesi yapılan algoritmaların kullanılabilirlik sıralamaları yani Çizelge 7.5'e göre, Bölüm 7.3'te yapılan değerlendirmenin özeti görülmektedir.

Çizelge 7.6. Performans değerlerine göre en iyi sıralaması

	İşlem zamanı (msec)	Memory (Byte %)	CPU (%)
Paragraf düzeyinde şfreleme	2	2	2
Kelime düzeyinde şifreleme	2	3	1
DES algoritması ile şifreleme	1	1	3

8. SONUÇ VE ÖNERİLER

Çalışmada şifreleme biliminde geçmişte kullanılan klasik teknikler incelenerek onlara paralel doğrultuda bir kriptografi algoritması geliştirilmiştir. Klasik tekniklerde etkili bir saldırı tekniği olan harf frekans analizi saldırısı için incelemeler yapılmış ve sonuçlarına yer verilmiştir. Ayrıca geliştirilen algoritmanın şifreleme işlemindeki performans değerleri incelenmiş ve sunulmuştur.

Tez kapsamında geliştirilen algoritmada sabit bir anahtar yoktur. Kelimenin içerdiği harf sayısı ve harflerin sayısal ağırlıklarına göre şifreleme anahtarı ve şifre alfabe değişmektedir. Yapılan işlemler incelendiğinde her harf için ayrı bir şifre alfabe oluştuğu görülmüştür.

Şifreleme algoritması, kelime düzeyinde ve paragraf düzeyinde olmak üzere iki farklı şekilde tasarlanmıştır. Algoritma ilk geliştirildiğinde sadece paragraf düzeyinde şifreleme işlemi için tasarlanmıştır. Daha sonra harf frekans analizinden elde edilen bulgulara göre; şifreli metindeki harf çeşitliliğinin az olması ve harf frekans analizi çalışmalarının daha kapsamlı yapılabilmesi için geliştirilen algoritmanın kelime düzeyinde de şifreleme işlemi yapmasına karar verilmiş ve uygulanmıştır.

Sezar şifreleme tekniğinin harf kaydırma(öteleme) prensibi, Alberti Diskinin harf kaydırma miktarının sabit olmaması özelliği, ENİGMA'nın üst üste şifreleme işlemi yapması, vigenére şifreleme tekniğinin farklı şifre alfabe prensibi, klasik tekniklerin çoğunun ortak özelliği olan mod alma işlemi özellikleri, geliştirilen algoritmada bir araya getirilmiş ve uygulanmıştır.

Algoritmada düz ve şifreli metinde karşılıklı harfler sabit değildir, düz metindeki bir harf şifreli metinde birden fazla harfi temsil edebilmektedir. Şifreli metindeki bir harf de düz metinde birden fazla harfi temsil edebilmektedir. Bu açıdan, geliştirilen algoritma çoklu alfabeli sistemlere benzemektedir. Bu durum algoritmanın deşifreleme işlemi kısmını etkilemiştir

ve sonucunda algoritmanın deşifreleme işleminde biyometrik bir özellik olan ses çözüm anahtarı olarak kullanılmıştır.

Geliştirilen algoritmada uygulanan harf frekans analizi incelemeleri, paragraf düzeyinde yapılan şifreleme işleminin kelime düzeyindeki şifreleme işlemine göre daha güvenilir olduğunu ortaya koymuştur. Paragraf düzeyinde yapılan şifreleme işleminde harf çeşitliliğinin kelime düzeyinde yapılan şifreleme işlemine göre daha az olması harf frekans analizi saldırısında yanılma payını arttırmıştır. Şifreli metin için yapılan çözümleme sayısı (olasılıklar) artmıştır. Kelime düzeyinde şifreleme işleminde harf çeşitliliği daha fazladır. Bu durum, harf frekanslarının %'lik ağırlıklarının çözümleme yaparken daha fazla yardımcı olmasını sağlamıştır. Şifreli metindeki harf çeşitliliğinin az olması frekans analizini zorlaştırmıştır.

Algoritmanın şifreleme, deşifreleme ve harf frekans analizi incelemeleri için arayüzler hazırlanmış ve butonlar aracılığı ile arayüzler arasında gezinme imkanı sağlanmıştır.

Performans analizi işlemlerinde Çizelge 7.1'de görülen iç faktörler incelenmiş, dış faktörlerin performans üzerindeki etkileri incelenmemiştir. Araştırmacılar için dış faktörlerin performans üzerindeki etkileri araştırma konusu olabilir.

Performans analizi incelemelerinde farklı uzunluktaki kelimelerin CPU iş yükü, bellek kullanımı ve işlem süresi açısından farklı değerlere sahip oldukları görülmüştür. Bu nedenle tez çalışmasında örnek olması amacıyla performans analizi işlemlerinde 8 Byte'lık sabit bir kelime kullanılmıştır. Performans analizi incelemeleri sadece şifreleme işlemleri için yapılmış olup CPU üzerindeki iş yükü, bellek tüketimi, işlem süresi açısından incelenmiştir. CPU üzerindeki iş yükü açısından en ideal algoritma kelime düzeyinde şifreleme yapan geliştirilen algoritmadır. Paragraf düzeyinde şifreleme algoritması ikinci, DES algoritması üçüncü olarak tespit edilmiştir. Bellek tüketimi açısından en ideal algoritma DES algoritması, ikinci paragraf

düzeyinde şifreleme algoritması, üçüncü kelime düzeyinde şifreleme algoritması olarak tespit edilmiştir. İşlem süresi açısından DES algoritması en ideal algoritma çıkmıştır. Kelime ve paragraf düzeyinde şifreleme algoritmalarında işlem süresi aynı tespit edilmiştir. Bunun sebebi 8 Byte'lık tek bir kelimenin kullanılmış olması olabilir. Her iki algoritma için aynı uzunluğa sahip farklı kelimeler içeren metin kullanıldığında farklı sonuçlar elde edilmektedir. Örneğin, *merhaba yeni dünya* girdisi için kelime düzeyinde şifreleme algoritmasının işlem süresinin daha kısa olduğu görülmüştür.

Geliştirilen algoritma üzerinde harf frekans analizi incelemeleri yapılmış ve bu saldırıya karşı güvenilir bir algoritma olduğu tespit edilmiştir. Ancak farklı saldırı türleri için herhangi bir inceleme yapılmamıştır. Algoritmanın güvenirliğini tespit etmek için başka yöntemler de kullanılabilir.

Şifreleme işleminin deşifreleme anahtarı olarak ses, parmak izi gibi biyometrik özellikler kullanılabilir. Uygulama projesinde biyometrik bir özellik olarak ses tercih edilmiştir. Tez kapsamındaki çalışma donanımsal boyuta taşınarak yeni bir kripto cihazı üretilebilir.

Yapılan çalışma ve incelemeler sadece Türkçe alfabe için yapılmıştır. Çalışma, farklı alfabeler için de geliştirilebilir. Algoritmaya farklı dil seçenekleri eklenebilir. Aynı şekilde algoritma üzerindeki harf frekans analizi incelemeleri faklı dillerde de uygulanabilir.

En güvenli sistemler bile çözüldüğü için farklı algoritmalar geliştirilmeli ve var olanların güvenlik derecesi arttırılmalıdır, yeni üretimler gerçekleştirilmelidir. Klasik teknikler kullanılarak hazırlanmış bu algoritma geliştirilmeye açıktır.

KAYNAKLAR

1.Türkiye Bilişim Derneği, “Bilişim Sistemleri Güvenliği El Kitabı Sürüm 1.0” (2006).

2.Vural, Y., Sağıroğlu, Ş., “Ülke Bilgi Güvenliği”, ***3.Uluslararası Katılımlı Bilgi Güvenliği ve Kriptoloji Konferansı***, Ankara, 3-8 (2008).

3.Canbek, G., Sağıroğlu, Ş., “Bilgi ve Bilgisayar Güvenliği Casus Yazılımlar ve Korunma Yöntemleri”, ***Grafiker Yayınları***, Ankara, 16-22,33,198-204, (2006).

4.Jones, A., “Information Warfare-what has been happining?”, ***Computer Fraud & Security***, 4-7 (November 2005).

5.İnternet: Tübitak Bilgem Ulusal Bilgi Güvenliği Kapısı, “BGYS-0001 Bilgi Güvenliği Yönetim Sistemi Kurulum Kılavuzu”, www.bilgiguvenligi.gov.tr/bilgi-guvenligi-yonetimi-dokumanlari/uekae-bgys-0001-bilgi-guvenligi-yonetim-sistemi-kurulum-kilavuzu.htm (2013).

6.Sağıroğlu, Ş., Alkan, M., “Bilgi Güvenliği Bilimi(Kriptoloji)”, Her Yönüyle Elektronik İmza, ***Grafiker Yayınları***, Ankara, 1-19, 25-39 (2005).

7.Vural, Y., “Kurumsal Bilgi GÜvenliği ve Sızma(Penetrasyon) Testleri”, Yüksek Lisans Tezi, ***Gazi Üniversitesi Fen Bilimleri Enstitüsü***, Ankara, 17-21,38-43 (2007).

8.Tekerek, M., “Bilgi Güvenliği Yönetimi”, ***KSU Fen ve Mühendislik Dergisi***, 11(1): 132-137 (2008).

9.İnternet : “Bilişim Ağı Hizmetlerinin Düzenlenmesi ve Bilişim Suçları Kanun Tasarısı, Kişisel Verilerin Korunması Hakkında Kanun Tasarısı”, http://bt-stk.org.tr/bilisim-hizmetler-suclari.html.

10.Mavzer, Ş., “Milli Olan Yazılımların ve Milli Olmayan Yazılımların Bilgi Güvenliğine Etkileri: Karşılaştırmalı Bir Çalışma”, Yüksek Lisans Tezi, ***TC Karaharp Okulu Savunma Bilimleri Enstitüsü***, Ankara, 4-8,37,43-56 (2006).

11.Civelek, Y. D., “Kişisel Verilern Korunması ve Bir Kurumsal Yapılanma Örneği”, Uzmanlık Tezi, ***TC Başbakanlık Devlet Planlama Teşkilatı Müsteşarlığı Bilgi Toplumu Dairesi Başkanlığı***, Ankara, 30-57 (2011).

12.Vural, Y., Sağıroğlu, Ş., “Kurumsal Bilgi Güvenliği ve Standartları Üzerine Bir İnceleme”, ***Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi***, 23(2): 507-522 (2008).

13.Topal, H., “Siber Terör”, Yüksek Lisans Tezi, ***İstanbul Üniversitesi Sosyal Bilimler Enstitüsü***, İstanbul, 1-26, 53-54 (2004).

14.İnternet: V. Uluslararası Bilgi Güvenliği ve Kriptoloji Konferansı, “Konferans Sonuç Bildirgesi”, www.iscturkey.org/ISCTURKEY2012/index.html, (2013).

15.Aslandağ, K., “Bilgi Güvenliği Kavramı ve Bilgi Güvenliği Yönetim Sistemleri ile Şirket Performansı İlişkisine Dair Bir Uygulama”, Yüksek Lisans Tezi, ***Gebze Yüksek Teknoloji Estitüsü, Sosyal Bilimler Enstitüsü***, Gebze, 13-21 (2010).

16.Güven, M., İnternette Güvenlik ve Hacker Cracker Meselesi, 1.Baskı, ***Grafiker Yayınları***, Ankara, 30-58 (2004).

17.İnternet:Wikipedia, “Kriptoloji”, http://tr.wikipedia.org/wiki/Kriptoloji, (2013).

18.Yerlikaya, T., Buluş, E., Buluş, N., “Kripto Algoritmalarının Gelişimi Ve Önemi”, ***Akademik Bilişim Konferansı***, Denizli, (2006).

19.Yılmaz, R.,“Kriptolojik Uygulamalarda Bazı İstatistik Testler” , Yüksek Lisans Tezi, ***Selçuk Üniversitesi Fen Bilimleri Enstitüsü***, Konya, 1-31 (2010).

20.Yıldırım, K., “Veri Şifrelemesinde Simetrik ve Asimetrik Anahtarlama Algoritmalarının Uygulanması (Hybrid Şifreleme)”, Yüksek Lisans Tezi, ***Kocaeli Üniversitesi Fen Bilimleri Enstitüsü***, Kocaeli, 1-23, 36-54 (2006).

21.Günden, Ü., “Şifreleme Algoritmalarının Performans Analizi”, Yüksek Lisans Tezi, ***Sakarya Üniversitesi Fen Bilimleri Enstitüsü***, Sakarya, 1-18, 28-34, 77-78 (2010).

22.Çeşmeci, M.Ü., “Kriptoloji Tarihi” , ***UEKAE Dergisi***, 1(1): 20-31, (2009).

23.Çimen, C., Akleylek, S., Akyıldız, E., “Şifrelerin Matematiği Kriptografi 6.Basım”, ***ODTÜ Yayıncılık***, Ankara, 1,5-66, (2008).

24.Dutta, A., McCrohan, K., “Management’s Role In Information Security in a Cyber Economy”, ***California Management Review***, 45(1): 67-87, (2002).

25.Gordon, L.A., Loeb, M.P., “The Economics of Information Security Investment”, ***ACM Transactions on Information and System Security***, 5(4): 438–457, (2002).

26.Whitman, M.E., “Enemy At The Gate: Threats To Information Security”, ***Communıcatıons Of The Acm***, 46(8): 91-95, (2003).

27.İnternet: Frank Lin, “Cryptography’s Past, Presents, And Future Role In Society”, http://engineering.wustl.edu/contentfiles/ecc/Lin.pdf (2013).

28.Liikanen, E., “Trust and Security in Electronic Communications: The Europen Approach”, ***Information Security Solutions Europe Conference(ISSE 99)***, Berlin, (1999).

29.Bauer, F.L., Decrypted Secrets Methods and Maxims of Cryptology Third Edition, ***Springer-Verlag Berlin Heidelberg***, Germany, 1-7 (2002).

30.İnternet: “Importance of Crptography”, https://www.cryptochallenge.com/home/importance (2013).

31.Kara, O., “2. Dünya Savaşı'ndan Günümüze Kriptoloji: Enigma'dan AES'e Şifreleme”, ***Bilim ve Teknik Dergisi***, (500): 28-33 (2009).

32.Akleylek, S., Yıldırım, M.H., Tok, Z.Y., “Kriptoloji ve Uygulama Alanları:Açık Anahtar Altyapısı ve Kayıtlı Elektronik Posta”, ***Akademik Bilişim Konferansı*** , Malatya, (2011).

33.Soyalıç, S., “Kriptografik Hash Fonksiyonları ve Uygulamaları”, Yüksek Lisans Tezi, ***Erciyes Üniversitesi Fen Bilimleri Enstitüsü***, Kayseri, 1-31 (2005).

34.İnternet: ODTÜ Bilgisayar Topluluğu Elektronik Dergisi, “Şifrelerin Büyülü Dünyası”, e-bergi.com/2008/Nisan/Kriptoloji (2013).

35.Jones, A., “Cyber Terrorism:Fact or Fiction”, ***Computer Fraud & Security***, 4-7, (June 2005).

36.Güvenoğlu, E., “Görüntü Şifreleme Algoritmaları ve Performans Analizleri”, Yüksek Lisans Tezi, ***Trakya Üniversitesi Fen Bilimleri Enstitüsü***, Edirne, 1-12, 35 (2006).

37.Dalkılıç, G., Akın, O., “Anahtar Tabanlı Gelişmiş Rotor Makinesi”, ***Akademik Bilişim Konferansı***, Gaziantep, (2005).

38.Bayar, E., “Modern Kriptosistemlerle Şifrelemenin Modellenmesi İle Veri Güvenliğinin Sağlanması”, Yüksek Lisans Tezi, ***Marmara Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 1-32 (2012).

39.Singh, G., Supriya, “A Study of Encryption Algorithms (RSA, DES, 3DES and AES) for Information Security”, ***International Journal Of Computer Applications****, 67(19): 33-38 (*2013).

40.Buluş, H.N., “Temel Şifreleme Algoritmaları ve Kriptoanalizlerinin İncelenmesi”, Yüksek Lisans Tezi, ***Trakya Üniversitesi Fen Bilimleri Enstitüsü***, Edirne, 1, 20, 23-43, 61-73 (2006).

41.Tuncal, T., “Bilgisayar Güvenliği Üzerine Bir Araştırma ve Şifreleme-Deşifreleme Üzerine Uygulama”, Yüksek Lisans Tezi, ***Maltepe Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 1-6, 16-44 (2008).

42.Başar, M.S., “Yer Değiştirme Esaslı ve Rasgele Anahtarlı Yeni Bir Şifreleme Algoritması”, Doktora Tezi***, Atatürk Üniversitesi Sosyal Bilimler Enstitüsü***, Erzurum, 1-17 (2004).

43.Singh, Y.K., “Generalization Of Vigenere Cipher”, ***ARPN(Asian Research Publishing Network) Journal Of Engineering and Applied Sciences***, 7(1): 39-44 (2012).

44.Saeed,F., Rashid, M., “Integrating Classical Encryption with Modern Technique”, ***IJCSNS International Journal of Computer Science and Network Security****,* 10(5): 280-285 (2010).

45.Denning, R. D.E., “Encyption Algorithms”, Cryptography and Data Security, ***Addison-Wesley Publishing***, United States of America, 1-3, 59-79, 86-89 (1982).

46.Ritter, T., “Substitution Cipher With Pseudo-RandomShuffling: The DynamicSubstitution Combiner”, ***Cryptologia***, 14(4): 289-303 (1990).

47.Papanikolau, N., “An Introduction to Quantum Cryptography”, ***Crossroads The ACM Student Magazine***, 11(3), (2005).

48.Rejewski, M., “Mathematical Solution Of The Enigma Cipher”, ***Cryptologia***, 6(1): 1-18 (1982).

49.Rejewski, M., “An Application of the Theory of Permutations in Breaking the Enigma Cipher”, ***Applicaciones Mathematicae***, 16(4), (1980).

50.Miller, A.R., “The Cryptographic Mathematics Of Enigma”, ***Cryptologia***, 19:(1): 65-80 (1995).

51.Demir, N., Dalkılıç, G., “Anahtar Bağımlı Bir Şifreleme Algoritması (IRON)”, ***Akademik Bilişim Konferansı***, Kütahya, 499-503 (2007).

52.Sakallı, M.T., “Modern Şifreleme Yöntemlerinin Gücünün İncelenmesi”, Doktora Tezi, ***Trakya Üniversitesi Fen Bilimleri Enstitüsü***, Edirne, 1-10, 45-48 (2006).

53.Kodaz, H., Botsalı, F.M., “Simetrik ve Asimetrik Şifreleme Algoritmalarının Karşılaştırılması”, ***Selçuk Üniversitesi Teknik Bilimler Meslek Yüksekokulu Teknik-Online Dergi***, 9(1): 10-23 (2010).

54.Coppersmith, D., “The Data Encryption Standart (DES) and its strength against attack”, ***IBM Journal of Research and Development***, 38(3): 243-244 (1994).

55.Patterson, C., “High Performance DES Encryption in Virtex™ FPGAs using JBits™”, ***8. IEEE Symposium on Field-Programmable Custom Computing Machines***, California, (2000).

56.Syed, F., “Children of DES: A Look at the Advanced Encryption Standart”, ***Elsevier Network Security,*** 2000(9): 11-12 (2000).

57.Ciğer, İ., “Data Şifreleme Algoritmaları ve Performans Analizi”, Yüksek Lisans Tezi, ***İstanbul Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 1-3, 11-13 (2012).

58.İnternet: Toyran, M., Pedersen, T.P., Hasekioğlu, A., Can, M.A., Berber, S., “Bilgi Güvenliğinde Kuantum Teknikler”, http://www.emo.org.tr/ekler/ebc1c93c2e9ad67_ek.pdf , (2013).

59.Arda, D., Buluş, E., “Türk Alfabesi ve Yapısal Özellikleri Kullanılarak Tek Alfabeli Yerine Koymada Şifreleme ve Kriptanaliz”, ***20. Türkiye Bilişim Kurultayı***, İstanbul, (2003).

60.Arda, D., Buluş, E., Yerlikaya, T., “Türkiye Türkçesi’nin Bazı Dil Karakteristik Ölçütlerini Kullanarak Vigenere Şifresi ile Şifreleme ve Kriptanaliz”, ***ELECO’2004 Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu ve Fuarı***, Bursa, (2004).

61.Arda, D., Buluş, E., Yerlikaya, T., “Simetrik Kriptosistemlerden Çok Alfabeli Yerine Koyma Metodunun Türkiye Türkçesi'nin Yapısal Özelliklerini Kullanarak Kriptanalitik İncelenmesi”, ***Ağ ve Bilgi Güvenliği Ulusal Sempozyumu-ABG2005***, İstanbul, (2005).

62.Dalkılıç, M.E., Dalkılıç, G., “On The Cryptographic Patterns and Frequencies in Turkish Language”, ***Advances in Information Systems***, ***Springer-Verlag Berlin Heidelberg***, 2457: 144-153 (2002).

63.İnternet: Eric Condrad, “Types Of Cryptographic Attacks”, http://www.giac.org/cissp-papers/57.pdf (2013).

64.Nabiyev, V., “Uygulama Problemleri”, Yapay Zeka, 3.Baskı, ***Seçkin Yayıncılık***, Ankara, 253-264 (2010).

65.Kolman, B., Hill, D., “Determinantlar”, Uygulamalı Lineer Cebir, 7.Baskı, Prof. Dr. Ömer Akın, ***Palme Yayıncılık***, Ankara, 315-350 (2002).

66.İnternet: Bartın Üniversitesi Yönetim Bilişim Sistemleri Bölümü, “karmasiklik.ppt”, http://iibf.bartin.edu.tr/ybs/files/karmasiklik.ppt (2013).

67.Yerlikaya, T., “Yeni Şifreleme Algoritmalarının Analizi”, Doktora Tezi, ***Trakya Üniversitesi Fen Bilimleri Enstitüsü***, Edirne, 1-3, 6-9, 51-58 (2006).

68.Yerlikaya, T., Buluş, E., Buluş, N., “Asimetrik Şifreleme Algoritmalarında Anahtar Değişim Sistemleri” , ***Akademik Bilişim Konferansı***, Denizli, (2006).

69.Öztürk, M.M., “Uzaktan Eğitimde Ölçme Değerlendirme Sistemi Tasarımı ve Yazılım Test Teknikleri İle Performans Analizi”, Yüksek Lisans Tezi, ***Sakarya Üniversitesi Fen Bilimleri Enstitüsü***, Sakarya, 5-29, 71-73 (2012).

70.Bahçetepe, H., “Modüler Çarpma Algoritmalarının İncelenmesi ve Kriptolojide Uygulanması”, Yüksek Lisans Tezi, ***İstanbul Üniversitesi Fen Bilimleri Enstitüsü***, İstanbul, 71-72 (2006).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, Adı	: ÜLKER, Ülkü
Uyruğu	: TC
Doğum tarihi ve yeri	: 02.01.1987 Bigadiç
Medeni hali	: Bekar
Telefon	: 05376876460
e-mail	: ulkeruu@hotmail.com

Eğitim

Derece	Eğitim Birimi	Mezuniyet
Lisans	Gazi Üniversitesi Endüstriyel Sanatlar Eğitim Fakültesi Bilgisayar Eğitimi Bilgisayar Öğretmenliği Bölümü	2009
Lise	Balıkesir Muharrem Hasbi YDA Lisesi	2005

İş Deneyimi

Yıl	Yer	Görev
2010	Balıkesir/Kepsut	Ücretli BT Öğretmenliği
2010-2011	Ankara	Ücretli BT Öğretmenliği
2011	Ankara	Özel Dershanede Yazılım Eğitmenliği
2012	Diyarbakır	Dicle Ünv. Ziya Gökalp Eğt. Fak. BÖTE Bölümü Araştırma Görevlisi

Yabancı Dil

İngilizce

Geliştirdiği Ders Projeleri

Görme duyusu açısından dezavantajlılar için MS Access veritabanı, MS Visual Studio C# programı ve DikteAPI Demo programı kullanılarak hazırlanan ses komutları ile e-posta iletim programı.

İlköğretime yeni başlayan öğrenciler için Delphi 7.0 programı kullanılarak OleContainer nesnesi ile görsel olarak desteklenmiş metin editörü programı.

Okul öncesi eğitim öğrencileri için ToolBook programı kullanılarak hazırlanan rakamları(0-9) ve okunuşlarını öğreten program.

Sql veritabanı ve MS Visual Studio C# programı kullanılarak hazırlanan Şişe Su Dağıtım Otomasyonu.

Sql veritabanı ve Delphi 7.0 programı kullanılarak hazırlanan Şişe Su Dağıtım Otomasyonu.

MS Accses veritabanı ve Delphi 7.0 programı kullanılarak hazırlanan Yayınevi Otomasyonu.

Yayınlar

1. Yalçın, N., Ülker, Ü., "Görme Engelliler İçin Ses Analizi İle E-posta İletimi", ***Gazi Üniversitesi Bilişim Enstitüsü Bilişim Teknolojileri Dergisi***, 4(3): 37-45 (2011).

2. Coşkun, A., Ülker, Ü., "Ulusal Bilgi Güvenliğine Yönelik Bir Kriptografi Algoritması Geliştirilmesi ve Harf Frekans Analizine Karşı Güvenirlik Tespiti", ***Gazi Üniversitesi Bilişim Enstitüsü Bilişim Teknolojileri Dergisi***, 6(2): 31-39 (2013).

Katıldığı Konferans,Sempozyum ve Seminerler

International Perspectives on New Aspects of Learning in Teacher Education International Conference (IPALTE) (Dinleyici)	Ekim2013
Siber Güvenlik Konferansı (Dinleyici)	Aralık 2011
DDoS ve BotNet Etkinliği (Dinleyici)	Şubat 2011
Güvenli İnternet Günü 2011 (Dinleyici)	Şubat 2011
IPV6 Konferansı (Dinleyici)	Ocak 2011

Sertifikalar / Katılım Belgeleri

MCSA *(Microsoft Certified Solutions Associate)*

Windows Server 2008 (MCSA)	Nisan 2012

MCTS *(Microsoft Certified Technology Specialist)*

Microsoft Certified Technology Specialist	
Windows Server 2008 Active Directory, Configuration	Mayıs 2011
Microsoft Exchange Server 2007, Configuration	Mayıs 2011
Windows Server 2008 Network Infrastructure, Configuration	Nisan 2011
Windows 7, Configuration	Mart 2011

MCITP *(Microsoft Certified IT Professional)*

Microsoft Certified IT Professional	
Server Administrator on Windows Server 2008	Mayıs 2011
Enterprise Messaging Administrator on Exchange 2007	Mayıs 2011

Meridyen Bilişim Akademi Kişisel Gelişim Eğitim Seminerleri Katılım Sertifikası *(CV Hazırlama, Mülakat Teknikleri, Sektörel Konuşma, Beden Dili)* Mayıs 2011

Hızlı Okuma Derneği Katılım Belgesi Mayıs 2011

Liderlik Okulu

Hızlı Okuma Sertifikası	Mayıs 2011
Liderlik ve Takım Çalışması Sertifikası	Şubat 2011
Yeni Liderlik Yaklaşımları Sertifikası	Şubat 2011
Kriz Yönetimi Sertifikası	Şubat 2011
Motivasyon Sertifikası	Şubat 2011
Etkili Dinleme Sertifikası	Şubat 2011
Proje Yönetimi Sertifikası	Şubat 2011

IPV6 Konferansı Katılım Belgesi Ocak 2011

Teşekkür Belgeleri

Kepsut Çok Programlı Lise Müdürlüğü Teşekkür Belgesi Mart 2010

Hobiler

Kitap okumak, el sanatları ile uğraşmak, resim yapmak, tarihi ve turistik bölgeleri gezmek.