T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SES ALGILAMA YÖNTEMİ İLE TEK KULLANIMLIK ANAHTAR (ONE TIME PAD) ÜRETİMİ

YÜKSEK LİSANS TEZİ

JABRAYİL HASANOV
Y1413.010028

Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Köksal MUŞ

ARALIK 2016

T.C.
İSTANBUL AYDIN ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Bilgisayar Mühendisliği Ana Bilim Dalı Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı **Y1413.010028** numaralı öğrencisi **Jabrayil HASANOV**'un **"SES ALGILAMA YÖNTEMİ İLE TEK KULLANIMLIK ANAHTAR (ONE TİME PAD) ÜRETİMİ"** adlı tez çalışması Enstitümüz Yönetim Kurulunun 21.11.2016 tarih ve 2016/27 sayılı kararıyla oluşturulan jüri tarafından oybirliği ile Tezli Yüksek Lisans tezi olarak kabul....edilmiştir.

Öğretim Üyesi Adı Soyadı	İmzası
Tez Savunma Tarihi :05.12.2016	
1)Tez Danışmanı: Yrd. Doç. Dr. Köksal MUŞ	.....
2) Jüri Üyesi : Prof. Dr. Ali GÜNEŞ	.....
3) Jüri Üyesi : Yrd. Doç. Dr. Burak ŞİŞMAN	.....

Not: Öğrencinin Tez savunmasında **Başarılı** olması halinde bu form **imzalanacaktır**. Aksi halde geçersizdir.

YEMİN METNİ

Sunduğum "SES ALGILAMA YÖNTEMİ İLE TEK KULLANIMLIK ANAHTAR (ONE TIME PAD) ÜRETİMİ" isimli yüksek lisans tez çalışmam dâhilindeki bütün bilgilerin akademik kurallara dayanarak elde edildiğini ve tez yazım kuralları çerçevesinde yazılarak sunulduğunu, ayrıca tez yazım aşamasında yararlandığım bütün kaynakların Bibliyografyada gösterilenlerden oluştuğunu belirtir ve onurumla beyan ederim.

Jabrayil HASANOV

ÖNSÖZ

Yüksek lisans eğitimine başladığım dönemden beri akademik bilgi ve becerileri bizlere yüksek düzeyde aşılayan, derslerimle veya araştırmamla ilgili karşılaştığım tüm sorunlarda her türlü destek ve yardımlarını esirgemeyen çok değerli hocam Prof. Dr. Ali GÜNEŞ'e, eğitim süresince benim kriptoloji ilmine ilgimin artmasında çok büyük katkı sağlayan değerli hocam Yrd. Doç. Dr.Vasiliya UZUN'a, aynı süreçde araştırma yaptığım bu alanda benim bilgi ve becerilerimin gelişmesinde büyük katkı sahibi olan kıymetli hocam Prof. Dr. Ahmad BABANLI'ya, yüksek lisans eğitimim süresince anlayış ve hoşgörüsüyle her türlü yardım ve desteyini esirgemeyen değerli hocam Saygıdeğer Yrd. Doç. Dr. Metin ZONTUL'a, tezimin araştırma aşamasında karşılaştığım her türlü problemlerin çözülmesi için çok değerli katkılarda bulunan ve her türlü yardım ve desteklerini esirgemeyen kıymetli hocam ve tez danışmanım Yrd. Doç. Dr. Köksal MUŞ'a,

Psikolojik anlamda bana kıymetli desteklerde bulunan, kendisini tanımaktan mutluluk duyduğum kız arkadaşım Şiringül ABDULLAYEVA'ya,

Yüksek lisans eğitimim süresince bana her türlü yardım ve desteklerini esirgemeyen tüm arkadaşlarıma,

Tüm hayatım boyunca benim yetişmemde, eğitimimde ve her türlü sorunlarımın çözülmesinde büyük katkıları olan çok değerli AİLEME,

Tüm yardım ve desteklerinden dolayı TEŞEKKÜR ediyorum.

Bu tez sevgili aileme ve arkadaşlarıma adanmıştır.

Aralık, 2016 Jabrayil HASANOV

İÇİNDEKİLER

Sayfa

ÖNSÖZ ..... **v**
İÇİNDEKİLER ..... **xi**
KISALTMALAR ..... **xiii**
ÇİZELGE LİSTESİ ..... **xv**
ŞEKİL LİSTESİ ..... **xvii**
ÖZET ..... **xix**
ABSTRACT ..... **xxi**
1 GİRİŞ **1**
2 BİLGİ VE ÖNEMİ **3**
2.1 Bilgi Güvenliği 4
2.2 Bilgi Güvenliği Unsurları 5
2.2.1 Gizlilik 5
2.2.2 Bütünlük 6
2.2.3 Erişebilirlik 6
2.3 Haberleşme Güvenliği 7
2.4 Bilgi Güvenliği Standartları 7
2.4.1 Standartların kullanım nedenleri 7
2.4.2 Yürürlükteki standartlar 8
2.5 Bilgi Güvenliği Tehdit Eden Unsurlar 11
2.5.1 İzinsiz erişim 11
2.5.2 Engelleme veya zarar verme 11
2.5.3 Değişiklik yapma 11
2.5.4 Üretim 11
2.6 Bilgi Güvenliğini Sağlama Araçları 12
3 KRİPTOLOJİ **13**
3.1 Kriptografi Nedir? 13
3.2 Kriptanaliz Nedir 15
3.3 Kriptolojinin tarihçesi 16
3.4 Kriptografi Algoritmalarının Sınıflandırılması 23
3.4.1 Klasik kriptografi teknikler 24
3.4.1.1 Sezar şifreleme 24
3.4.1.2 Alberti diski 25
3.4.1.3 Vigenere şifreleme 25
3.4.1.4 Hill şifreleme 29
3.4.1.5 Playfair şifreleme 30
3.4.1.6 Enigma 31
3.4.2 Modern kriptografi teknikler 32
3.4.2.1 DES –veri şifreleme standartı 32
3.4.2.2 AES – ileri şifrleme standartı 44
3.4.3 Blok şifreleme 49

3.4.4 Akış şifreleme 50
3.4.5 Asimetrik (açık anahtarlı) şifreleme teknikleri 51
3.4.5.1 RSA algoritması 52
3.4.5.2 DSA algoritması 54
3.4.5.3 Diffie – Helman açık anahtar dağıtımı 54
3.5 Kriptanaliz Teknikleri Ve Saldırı Çeşitleri 56
3.5.1 Sadece şifeli metin saldırısı (Ciphertext Only) 56
3.5.2 Bilinen açık metin saldırısı (Known Plaintext) 57
3.5.3 Seçilmiş açık metin saldırısı (Chosen Plaintext) 57
3.5.4 Seçilmiş şifreli metin saldırısı (Chosen Ciphertext) 57
3.5.5 Seçilmiş açık veya şifreli metin saldırısı (Adaptive chosen plaintext or ciphertext) 57
3.5.6 İlişkili anahtar atağı 58
3.5.7 Kaba güç (Brute force) saldırısı 58
3.5.8 Ortadaki adam saldırısı (Man-in-the-Middle) 58
3.6 Tek Kullanımlık Anahtar (One Time Pad) 58
3.6.1 Tek Kullanımlık anahtar (one time pad) nedir? 59
3.6.2 Tek kullanımlık anahtarın (One Time Pad) güvenliği 59
3.6.3 Vernam şifreleme (one time pad) 60
4 ARAŞTIRMADA KULLANILAN YÖNTEMLER, TEKNİKLER VE MATERYALLER 63
4.1 Yöntemler 63
4.1.1 Ses tanıma yöntemi 63
4.1.2 XOR Operatörüne tabi tutma yöntemi 64
4.2 Teknik ve Materyaller 66
5 GELİŞTİRİLEN ALGORİTMANIN VE DEMONUN TANITIMI 69
5.1 Algoritmanın Tanıtımı 69
5.2 Demonun Tanıtımı 74
6 GELİŞTİRİLEN ALGORİTMANIN PERFORMANS DEĞERLERİ 85
6.1 İşlemci Kullanım (CPU sampling) Verileri 85
6.2 Bellek Kullanımı (.Net memory allocations) Verileri 89
7 SONUÇ 93
KAYNAKLAR 95
EKLER 99
ÖZGEÇMİŞ 105

KISALTMALAR

ISO :International Standarts Organization
DES :Data Encrypt System
RSA :Rivest-Shamir-Adleman
AES :(Advanced Encryption Standard), Gelişmiş şifreleme Standardı
RC2 :Rivest's Cipher / Rone's Code2
RC5 :Rivest's Cipher / Rone's Code5
FEAL :Fast Date Encripherment Algorithm
NSA :Ulusal Güvenlik Ajansı
NIST :Ulusal Teknoloji ve Standardlar Enstitüsü
SPN :Substitutio-Permutation Networks
DSA :Digital Signature Standard
RC4 :Rivest Cipher 4 (Ron Rivest Simetrik Şifreleme Algoritması)
PC :Personal Computer (Kişisel Bilgisayar)
IEEE :Institute of Electrical and Electronics Engineers (Elektrik Elektronik Mühendisleri Enstitüsü)

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1 Vigenere Örnek Çözüm Tablosu **27**
Çizelge 3.2 Vigenere Örnek Çözüm Tablosu **27**
Çizelge 3.3 Vigenere Örnek Çözüm Tablosu **28**
Çizelge 3.4 Tükçe alfabe için oluşturulmuş Vigenére Tablosu **29**
Çizelge 3.5 Başlanğıç Permütasyonu (IP) **35**
Çizelge 3.6 Genişletme Permütasyonu (E) **37**
Çizelge 3.7 S – Kutusu 1 ..... **38**
Çizelge 3.8 S – Kutusu 2 ..... **38**
Çizelge 3.9 S – Kutusu 3 ..... **39**
Çizelge 3.10 S – Kutusu 4 ..... **39**
Çizelge 3.11 S – Kutusu 5 ..... **39**
Çizelge 3.12 S – Kutusu 6 ..... **39**
Çizelge 3.13 S – Kutusu 7 ..... **40**
Çizelge 3.14 S – Kutusu 8 ..... **40**
Çizelge 3.15 P Permütasyon tablosu (P) **40**
Çizelge 3.16 Ters Permütasyon tablosu **41**
Çizelge 3.17 Permutasyon seçimi tablosu (PC-1) **42**
Çizelge 3.18 PC – 2 Permütasyon Tablosu **43**
Çizelge 3.19 Alt Anahtar üretim zamanı sola ve sağa kaydırma tablosu **43**
Çizelge 3.20 Blok Şifreleme Algoritmalarının Genel Yapısı **49**
Çizelge 4.1 A kısımlarının XOR işlemini içeren tablo. **65**
Çizelge 4.2 Tek Kullanımlık Anahtarın üretim tablosu **66**
Çizelge 5.1 Örnek Şifreleme İçin Girdilerin Bit Karşılıkları **72**
Çizelge 5.2 Tanımlanan Sesin bit kısımlarının XOR işlemi **73**
Çizelge 5.3 Tek Kullanımlık Anahtarın üretildiği tablo **73**
Çizelge 5.4 Şifreleme işleminin yapıldığı tablo **74**
Çizelge 5.5 Şifre çözme işleminin yapıldığı tablo **74**

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 Gizlilik, Bütünlük ve Erişebilirlik Üçlü Nitelikleri (Otgonjargal, 2013) **5**
Şekil 3.1 Kriptolojinin alt bilim dalları **13**
Şekil 3.2 Genel Şifreleme İşlemi **14**
Şekil 3.3 Şifre çözme işlemi **15**
Şekil 3.4 Şifreleme Algoritmalarının sınıflandırılması (Bayar, 2012). **24**
Şekil 3.5 DES algoritmasının Blok diyagramı (Şen, 2006) **35**
Şekil 3.6 Bir DES Döngüsünde Gerçekleştirilen İşlemler (Şen, 2006). **36**
Şekil 3.7 F Fonksiyonu, F(R,K)'nın hesaplanması (Bayar, 2012) **37**
Şekil 3.8 Anahtar Üretim Tablosu (Şen, 2006) **42**
Şekil 3.9 AES Algoritması Blok Diyagramı (128 bit anahtarlı) (Sakallı, 2006) **46**
Şekil 3.10 AES S Kutusu (Hexadecimal notasyonda xy byte için) (Yerlikaya, 2006). **47**
Şekil 3.11 ShiftRows Örneği a. Öteleme Öncesi, b. Öteleme Sonrası (Yerlikaya, 2006) **48**
Şekil 3.12 Sütunların Karıştırılması Dönüşümü (Ülkü, 2014). **48**
Şekil 3.13 a Feistel Mimarisi, **b** SPN Mimarisi **50**
Şekil 3.14 Senkron Bir Akış Şifreleme Algoritmasının Genel Yapısı **51**
Şekil 3.15 Asimetrik Şifreleme Algoritmalarının Genel Yapısı (Yerlikaya, 2006) .. **52**
Şekil 3.16 Diffie-Hellman Anahtar Paylaşım Protokolü (Tefon, 2013) **56**
Şekil 5.1 Tek Kullanımlık Anahtarın (One Time Pad) Akış Diyagramı **70**
Şekil 5.2 Şifreleme Diyagramı **71**
Şekil 5.3 Şifre Çözme Diyagramı **71**
Şekil 5.4 Şifreleme İşleminin Yapıldığı Formun Aktivite Diyagramı **75**
Şekil 5.5 Demo'nun Tüm İşlemlerin Gerçekleştirildiği Arayüz Formu **76**
Şekil 5.6 Ses tanıtımına başlandıktan sonra arayüzün görünümü **77**
Şekil 5.7 Anahtar Üret butonuna tıklama sonucunda arayüzün görünümü **78**
Şekil 5.8 Anahtar kelime boyutu tanımlanan sesin yarı boyutundan büyük olduğunda Anahtar Üret buttonuna tıklanınca çıkan arayüz ve uyarı mesajı. **79**
Şekil 5.9 Şifrele butonuna tıklandığında şifrelemede yapılan işlemlerin arayüz üzerinde görünümü. **80**
Şekil 5.10 Tek Kullanımlık Anahtar boyutu açık metin boyutundan küçük olduğunda yapılan şifreleme işlemleri ve uyarı mesajının verildiği arayüz. **81**
Şekil 5.11 Form üzerinde yapılan şifre çözme işleminin aktivite diyagramı **82**
Şekil 5.12 Şifre çözme işlemi yapıldığında arayüzün görünümü. **83**
Şekil 6.1"Yarın Projeyi bitirelim" açık metninin şifrelenmesine ait performans grafiği **85**
Şekil 6.2"Yarın Projeyi bitirelim" açık metninin şifrelenmesinde sistemi darboğaz eden sınıflar **86**
Şekil 6.3 Sistemde darboğaz oluşturan fonksiyonlar **86**
Şekil 6.4 Sistemde darboğaz oluşturan fonksiyonların %'leri **87**

Şekil 6.5 Sınıf ve fonksiyonların bulundukları satırları gösteren CPU kullanımı detayları **87**
Şekil 6.6 Şifreleme işlemleri için kullanılan fonksiyonların CPU kullanım oranları **88**
Şekil 6.7 CPU kullanımına ait detaylı süre bilgisi **88**
Şekil 6.8 CPU kullanımına ait süre bilgisi **89**
Şekil 6.9 Bellek kullanımının grafiki **89**
Şekil 6.10 Bellek kullanım %'leri yüksek olan fonksiyonlar **89**
Şekil 6.11 Bellek kullanımı en yüksek olan veri tipleri **90**
Şekil 6.12 Fonksiyon ve sınıfların bellek kullanımına ait detaylar **90**
Şekil 6.13 Function Details seçeneğine ait bellek kullanım detayları **91**

SES ALGILAMA YÖNTEMİ İLE TEK KULLANIMLIK ANAHTAR (ONE TIME PAD) ÜRETİMİ

ÖZET

Tek kullanımlık şifreler rastgele anahtarla üretildiği zaman mükemmel güvenlik sağlarlar. Anahtarın ele geçirilmesi için gereken işlem gücü, bütün anahtarları denemekten daha az değildir. Bunun yanında şifrelenecek metinle aynı uzunluktaki bir anahtarı güvenli şekilde saklamanın, düz metni sağlamayla aynı iş gücüne karşılık geldiği için bahsedilen şekliyle kullanılmamaktadır. Yani, anahtarı güvenli şekilde saklamakla düz metni güvenli şekilde saklamak arasında fark olmadığı için tam güvenlik sağlamasına rağmen kullanılmamaktadır. Bu sebeple, modern kriptoloji teknikleri temel olarak kısa anahtarlarla, uzun, rastgele gözüken (sözde rastgele - pseudo random) bit dizileri üreterek güvenli şifreleme algoritmaları oluşturarak saklanması gereken anahtar boyutunu mümkün olduğunca küçültebilme problemiyle ilgilenirler.

Bu çalışmada amaç küçük anahtarla üretilmeye çalışılan sözde rastgele bit dizisini, ses tanıma sistemi kullanılarak uzun cümlelerle oluşturmaktır. Tanımlanan algoritmanın detaylı tanımı ve performans analiz değerlendirilmesi de yapılmıştır. Visual Studio 2015 programı ortamında C# dili kullanarak algoritma için tanıtım programı hazırlanmıştır.

Anahtar Kelimler: Tek Kullanımlık Anahtar Üretimi, Ses Tanıma Yöntemi, Kriptoloji

GENERATION OF ONE TIME PAD USING SPEECH RECOGNITION SYSTEM

ABSTRACT

When disposable generated with a random key they provides excellent security, It's not less than trying all the keys. The processing power required to seize the key besides that, keep a key in the same length as the text to be encrypted security, providing solid text with the same job power it is not used as mentioned because it corresponds. It means no difference is used there between keeping the key secure despite providing full security therefore modern cryptography techniques are mainly based on short keys, and long, random (So-called random - pseudo random) Bit arrays to create secure encryption algorithms Key size to store as much as possible They are interested in the problem of diminishing. The purpose of the this study is production of small-key and Running pseudo-random bit sequence, Using voice recognition system to create long sentences. Defined the detailed description of the algorithm and performance analysis evaluation has been done. Program has been prepared. In the Visual Studio 2015 program environment and Using C# language for Introduction to algorithm.

Keywords: disposable key production, Speech Recognition, Cryptology

1 GİRİŞ

Kullanışa başlanması M.Ö.'ki yıllara dayanan kriptoloji biliminin tarihen insan hayatının özel bilgileri, harp sırları ve haberleşme gibi geniş alanda bilgilerin kötü amaçlı insanlardan saklanması, karşı tarafa güvenli şekilde ulaştırılması ve ulaştırılacağı kişi dışında kimseye sızmaması amacıyla kullanıldığı bilinmektedir. Tarih boyunca insan hayatı ve yaşam için önemli olan bilgilerin sürekli tehditlere maruz kalması, bu tehditlerle karşılaşmamak ve bilgi güvenliğini sağlamak amacıyla bilimin vazgeçilmezi olan kriptolojinin zaman zaman geliştirilmesini, yenilenmesini geliştirilmeye açık tutulmasını sağlamıştır. XX yüzyılın başlarından başlayarak bilgilerin telsizler ortamında aktarılması, haberleşmenin makinalar ortamına geçmesi bilgileri büyük tehditler altında bırakmış ve bilginin tehditlerden korunması için artık makineler ortamında bilgi güvenliğini sağlamak amacıyla eski kriptoloji şifreleme sistemlerinin geliştirilmesi ve yeni kriptoloji sistemlerin hazırlanmasını zorunlu kılmıştır. Artık gelişen kriptolojide klasik tekniklerde kullanılan alfabeye göre yer değiştirme, kaydırma, mod alma ve diğer eski yöntemler kullanışsız hale gelmiştir. Makineler aracılığıyla bilgilerin, en küçük bilgi birimi olan bitleri arasında geliştirilen yeni kriptoloji şifreleme sistemleri ile şifreleme yaparak güvenliğinin sağlandığı yeni makineler dönemi başlamıştır.

Günümüzde büyük hacimde bilgilerin hayatımızın vazgeçilmezi olan elektronik ortamda ve ağlar üzerinden aktarılması, depolanması ve kullanışı bizi, bu bilgilerin çalınması, değiştirilmesi, bozulması ve özel bilgilerin ele geçirilmesi gibi büyük güvenlik sorunları ve tehditlerle karşı karşıya bırakıyor. Bu bilgilerin korunması sebebi ile kriptoloji bilimi sürekli geliştirilmekte ve geliştirilmeye açık olmaktadır. Ancak hala bilgiler tehditlere maruz kalmakta ve yeni kriptografi sistemlerin üretilmesine ihtiyaç duyulmaktadır. Yeni geliştirilen kriptoloji yöntemlerinin daha önceki yöntemlerin incelenerek ve değiştirilerek üretilmesi bu alanda üretimin kısıtlı olduğunun göstergesidir. Var olan sistemlerin incelenmesi ve değiştirilerek yeni kriptoloji sistemlerin geliştirilmesi ulusal açıdan olumlu olurken, dış kaynaklı bir kriptoloji sisteminin geliştirilerek ulusal bilgi güvenliği için kullanılması bu bilgilerin

yüksek güvenliğini sağlamamaktadır. Bu nedenle bilgi güvenliğinin sağlanması için kriptoloji bilimine daha büyük önem verilmeli ve yeni kriptoloji sistemlerinin üretilmesi gerekmektedir.

Bu tez çalışmasında Tek Kullanımlık Anahtar (One Time Pad) yöntemli yeni bir kriptografi algoritma geliştirilmesi amaçlanmıştır. Tek Kullanımlık Anahtarla şifreleme yöntemlerinde anahtarın rastgele olması zorunluğundan yola çıkarak geliştirilen algoritmada anahtarın rastgele gözüken (sözde rastgele- pseudo random) olması için ses tanıtma yöntemleri kullanılmıştır. Tek Kullanımlık Anahtar yöntemli şifreleme algoritmalarından bilindiği gibi bu algoritmalarda en önemli kısım anahtarın rastgele (sözde rasgele - pseudo random) üretilmesidir. Anahtarın rastgele üretilmesi bu tip algoritmaların güvenliğini yüksek tutmaktadır. Klasik tekniklerle şifreleme yapan ve günümüzde geliştirilen birçok kiriptoloji algoritmaların kolayca kırıldığı bilinmektedir. Bu nedenle algoritmanın geliştirilmesi aşamasında bilinen Tek Kullanımlık Anahtar yöntemleri ve modern kriptoloji şifreleme teknikleri detaylı olarak incelenmiştir. Geliştirilen algoritmanın simülasyonu için Visual Studio 2015 programı üzerinden C# dili kullanarak Windows Form arayüzlü demo programı hazırlanmıştır. Demo programı için donanımsal olarak Windows 8. 1 Pro işletim sistemine sahip hp ProBook 4530s dizüstü bilgisayar ve demo programının sesi algılayabilmesi için bir adet mikrofon kullanılmıştır. Çalışma, kitap, dergi, tez, makale, bildiri gibi basılı materyaller, ağ ve veri tabanları üzerinde ulaşılabilen makaleler, dergiler, araştırmalar ve diğer dökumanların incelenmesi ile yapılmıştır.

Tez projesi yedi bölüm üzerinden anlatılmaktadır. İkinci bölümde bilgi, bilginin önemi ve bilgi hakkında genel bilgiler yer almaktadır. Üçüncü bölümde kriptoloji ve tarihi, kriptolojinin ayrıldığı kriptografi ve kriptanaliz dalları, kroptografinin klassik ve modern teknikleri örnekler ile birlikte ve bazı kriptanaliz yöntemleri yer almaktadır. Dördüncü bölümde geliştirilen algoritmanın geliştirme sürecinde kullanılan yöntemler, teknik ve materialler, konuşma tanıma yöntemleri yer almaktadır. Beşinci bölümde geliştirilen algoritmanın demo programının arayüzlerle adım adım işleyişi tanımlanmış ve gösterilmiştir. Altıncı bölömde demo programının CPU kullanımı ve bellek kullanımının detaylı şekilde performans analizleri yapılmış, incelenmiş ve değerlendirilmiştir. Son bölümde tez projesi dâhilindeki tüm çalışmaların sonuçları değerlendirilmiştir.

2 BİLGİ VE ÖNEMİ

Menşeyini Latince "informare" kelimesinden alan bilgi (information) kelimesi, her hangi bir şeyi şekillendirmek, şekil vermek anlamı gibi kabul edilmektedir (Vural, 2007).

Sözlük anlamıyla bilgi;

• Öğrenme, araştırma ve gözlem yoluyla elde edilen her türlü gerçek, malumat ve kavrayışın tümü,

• İnsan aklının erebileceği olgu, gerçek ve ilkelerin bütünü, bilim, malumat,

• İnsan zekâsının çalışması sonucu ortaya çıkan düşünce ürünü, malumat, vukuf,

• Genel olarak ve ilk sezi durumunda zihnin kavradığı temel düşünceler ve

• Kurallardan yararlanarak kişinin veriye yönelttiği anlam olarak tanımlanmaktadır (TDK, 2016).

Literatürde bilgi farklı şekilde tanımlanmaktadır:

Bilgi, insanın varlığı tanıma ve anlama isteği sonucunda oluşan, düşünebilen süje ile obje arasındaki ilişkidir. Suje; bilgiye yönelen, öğrenen, araştıran, bilen insandır. Obje; bilgiye konu olan, bilinen somut varlıkların tümüdür. İnsan başka bir varlığı düşündüğünde, araştırdığında, öğrendiğinde suje, başkası tarafından araştırıldığında ise objedir (Vural, 2007), (MEB, 2013). Türk Dil Kurumu bilgiyi şu şekilde tanımlamıştır: "İnsan zekâsının çalışması sonucunda ortaya çıkan düşünce ürünü, malumat, vukuf". (Muharremoğlu, 2013).

Yaşamın varlığından beri bilgi, önemini hayatın çeşitli alanları üzerinde tutmakta ve zaman geçtikçe daha da artırmaktadır. Bilgi, geçmişlerden günümüze kadar çeşitli alanlarda hayatın süzgecinden geçerek, daha da gelişmiş, büyümüş ve çeşitli araçlarla günümüze kadar ulaşmıştır. Bilginin eski çağlardan günümüze ulaşmasında ve yaygınlaşmasında M.Ö' lere dayanan taş kitabeler, eserler, destanlar, masallar, XII yüzyıldan sonra ise medreseler, üniversiteler, kitaplar araç olarak en önemli role sahip olmuşlar. XX yüzyılın başlarından başlayarak ise artık bilginin saklanılması,

korunması ve gelecek nesillere aktarılması yönünde hazırda devam eden çok büyük teknoloji gelişmeler sonucunda bilgi çağı adı verilen yeni dönemi yaşamaktayız. Yaşadığımız döneme adını kazıyan bilgi günümüzde insanların yaşam tarzına yön verdiği gibi, gelecekte de bu önemini koruyacak ve daha da ilerilere taşıyacaktır (Vural, 2007).

2.1 Bilgi Güvenliği

Bilgi güvenliği; bilginin mevcut tehditlerden korunması, gereken teknolojinin doğru şekilde kullanılarak bilgiye mümkün tüm ortamlarda, istenmeyen kişiler tarafından ulaşılmasını önlemektir. Başka bir tanımla bilgi güvenliği; elektronik ortamlarda verilerin veya bilgilerin saklanması ve taşınması zamanı bilgilerin bütünlüğü bozulmadan, izinsiz erişimlerden korunması için, güvenli bir bilgi işleme platformu oluşturma çabalarının tümüdür (Haklı, 2012).

Bilgi, bir kurum için sunduğu ürünler kadar önem taşıdığından dolayı uygun bir ortamda korunmalıdır. Bilginin farklı kurumlar arasında paylaşılması arttıkca, bilgi paylaşımındakı çeşitli risklerin olma ihtimali ve bilgiyi korumaya yönelik gereksinimler de artmaktadır. Bilindiği gibi eskilerden beri tarih boyunca askeri alanda özel bilgilerin, harp sırlarının düşmandan korunması için, zamanla gelişerek günümüze kadar gelen çeşitli araçlar, yöntemler ve sistemler kullanılmıştır.

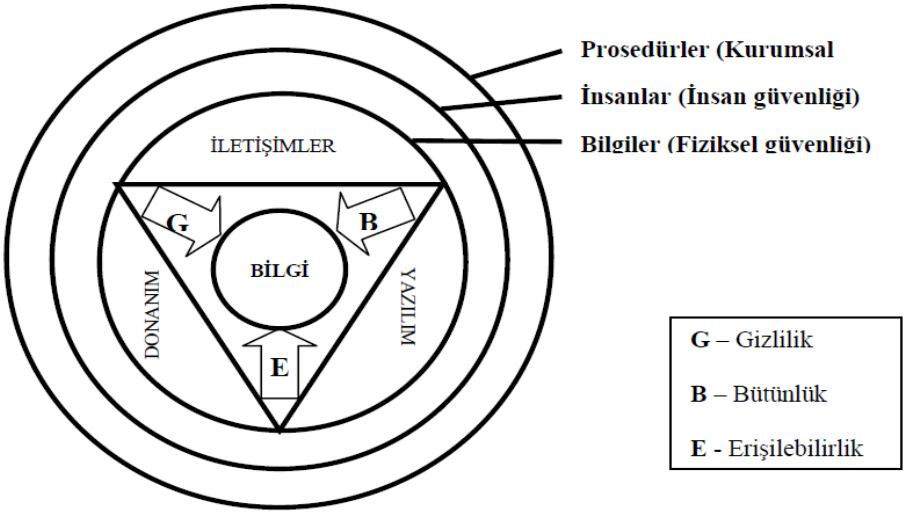
Bilgi, baskılı olarak veya kağt dokümanları üzerinde yazılı, elektronik ortamlarda saklanan, posta ya da elektronik posta yolu ile aktarılabilen, insanlar arasında sözle ifade edilebilen birçok çeşitli biçimlerde bulunabilmektedir. Bilginin kullanıma yararlı olarak tutulabilmesi için mutlaka uygun bir ortamda korunması gereklidir. Aktif kullanımda olan bilgilerin büyük bir kısmı artık bilgisayarlar aracılığıyla elektronik olarak saklanmakta, korunmakta ve güvenli biçimde diğer bilgisayarlarla paylaşılabilmektedir.

Bilgileri çeşitli risklerden ve tehlikelerden korumak ve güvenliğini sağlamak, bilginin iş risklerini azaltmak ve iş kazançlarını artırmak demektir. Örneğin bir firmanın veya şirketin özel işçi bilgileri, müşteri bilgileri, finansal ve yeni ürün bilgileri gibi özel bilgileri, iş kaybına ve iflasa götürebilecek güvenlik açığı sebebiyle rakip şirketlerin eline geçmemelidir.

Bilgi güvenliğinin sağlanması için kurumsal yapı, politika, süreç, yazılım ve donanımların bir arada işlevlerini içeren denetimler dizisi gerçekleştirilmektedir. Kurumun veya şirketin iş güvenliğinin sağlanması amacıyla, özel bir ortamda bahsetdiğimiz denetimlerin kurulması, uygulanması, izlenmesi, incelenmesi ve geliştirilmesi gerekmektedir. Tüm bu güvenlik yöntemlerinin diğer iş yönetimi süreçleri ile birlikde her hangi bir aksaklığa veya güvenlik açıklığına yol verilmeden yapılması gerekmektedir (Otgonjargal, 2013), (Akay, 2014).

2.2 Bilgi Güvenliği Unsurları

Bilgi güvenliğinin sağlanabilmesi için genel olarak bilginin gizliliği, bütünlüğü ve erişebilirliğinin sağlanması gerekmektedir. Bilgi güvenliğini sağlamak için gereken bu üç niteliğin tanımı aşağıdakı şekilde gösterilmiştir.



Şekil 2.1 Gizlilik, Bütünlük ve Erişebilirlik Üçlü Nitelikleri (Otgonjargal, 2013)

2.2.1 Gizlilik

Gizlilik, bilginin işlenmesi, iletilmesi, paylaşılması veya her hangi bir prosesi zamanı bilgi sahibi tarafından istenmeyen kurumlar veya kişilerden korunmasını sağlamak anlamını vermektedir.

Gizlilik Uluslararası Standartlar Örgütü (ISO) tarafından "Bilgiye sadece yetkilendirilmiş kişilerce ulaşılabilmesi" olarak nitelenir (Aslandağ, 2010). Başka bir değişle gizlilik, bilgiye sadece izni olan kişilerin ulaşması ve yetkisiz kişilerin ulaşmamasını sağlamak anlamına gelmektedir. Hazırda büyük bilişim firmaları bilginin gizliliğinin korunması amacıyla şifreleme yöntemlerini kullanmaktadırlar. Her kurum

için bilginin gizliliği ve güvenliği aynı önemde olmaya bilir. Bilginin güvenliği ve gizliliği özellikle bankalar ve kamu kuruluşları için çok önemlidir.

Gizlilik bazı krumlar için yasa ve kanun açısından zorunlu olmaktadır. Örneğin avukat - müvekkil ilişkisi ya da doktor hasta ilişkisi gibi mesleki bilgiler kanun ile koruma altına alınmıştır. Bazı durumlarda ise taraflar birtakım bilgileri sözleşme ile birbirlerine verirken gizlilik anlaşmaları yaparlar. Her iki durumda da gizlilik büyük önem taşımaktadır (Aslandağ, 2010), (Yıldız, 2007), (Altun, 2014).

2.2.2 Bütünlük

Bütünlük, bilgilerin aktarıldığı zaman, depolanırken, doğru, tam olması ve yetkisiz kişiler tarafından değiştirilmemesi veya yok edilmemesi anlamına gelir. Başka bir değişle bütünlük, bilginin yetkisiz kişiler tarafından silinmesi, değiştirilmesi ve yok edilmesine karşı korunmanın garantilenmesidir. Bilginin bütünlüğünün yanlışlıkla ya da kısıtlı olarak kayp edilmemesi için, kurum dâhilindeki tüm var olan bilgilere bu bilgilerden sorumlu olacak kişiler atanmalıdır. Kurumun dahilinde kendine özen olarak bilgilerin korunması ve saklanması için ayarladığı tedbirlere uygun denetlemeler yapılmalıdır.

Bilginin kaza olarak yada bilerekten değiştirilmesi veya bozulması, bu bilginin bozulmuş olduğu gerçeğini değiştiremez. Bunun için bilginin koruması tedbirlerini gerçekleştirdikte her iki riski göze almak gerekmektedir. Bilginin bütünlüğü ve doğruluğu için "Kesinlilik, Doğruluk ve Geçerlilik" unsurları sağlanması mutlaktır (Otgonjargal, 2013), (Aslandağ, 2010).

2.2.3 Erişebilirlik

Erişilebilirlik, bilginin yetkili kurum veya kişi tarafından gerektiği zaman kullanılabilir ve erişilebilir olma özelliğidir.

Bilgi, yetkili kişiler tarafından zamanında ulaşılabilir olmalı ve ulaşım sırasında gerektiği zaman kaynak paylaşımına izin verilen biçimde olmalıdır. Kullanılabilirlik, yetkili kişiler tarafından erişilebilir olmasıyla beraber hemde ulaşılabilir olması demektir (Haklı, 2012).

Bir diğer tanımla erişilebilirlik veya kullanılabilirlik bilginin ihtiyaç duyulduğu zaman kullanıma ve erişime hazır olmasıdır. Ortaya çıkan sorun veya problemlerden aslı olmaksızın bilginin erişilebilir ve kullanılabilir olması gereklidir. Erşim

kullanıcının hakları dahilinde olmalıdır. Her kullanıcı bilgiye erişim hakkına sahip olduğu zaman süresinde bilgiye erişebilmeli ve kullanabilmelidir. Güvenlik hizmetlerinin ve sistemin güvenlik performansının düzenli biçimde izlenmesi ve kontrol edilmesi, kullanılabilirliği garantilemek için uygulanacak önleyici tedbirlerdir (Aslandağ, 2010).

2.3 Haberleşme Güvenliği

Bilginin karşılıklı alışverişi ve paylaşımı zamanı, güvenli haberleşme ortamı oluşturmak üzere yapılan çabaların ortak ismi haberleşme olarak adlandırılmaktadır. Heberleşme zamanı bilginin fiziksel olarak güvenliğinin sağlanması, haberleşme güvenliği için yeterli olmamaktadır. Haberleşme sırasında bilginin aktarılması ve paylaşımı zamanı hedefe ulaşmadan önce üçüncü şahıs ve ya düşman tarafından ele geçrilmesi, değiştirilmesi riski her zaman aktualdır. Tarih boyunca haberleşme güvenliğinde kullanılan yöntemler ve teknikler değişmese de, bu güvenliği sağlamak için kullanılan yöntemler her zaman gelişmiştir. Haberleşme güvenliğinin sağlanmasında kriptografi ve stenografi tekniklerin evezsiz rolü vardır (Vural, 2007).

2.4 Bilgi Güvenliği Standartları

Güvenlik sürecinin temelini, bir kurumun varlıklarını genel olarak korumaya alınması için tasarlanmış bir güvenlik politikası oluşturmaktadır. Bir kurum kendi bilgi güvenliği yönetim sisteminde, kendi dahilindeki bilgi kontrol araçlarını bir araya getirebilir, uygulama ilkelerini benimseyebilir ve var olan mümkün güvenlik standartlarından birini seçip sertifika almak için çalışabilir ya da bunların bir karışımını izleyebilir. Bütün bu çabaların amacı var olan bilgileri yeterli şekilde güvenç altına almaktan ibarettir (Aksu, 2014).

2.4.1 Standartların kullanım nedenleri

Organizasyonların bilgi güvenliğinin sağlanmasında karşılaştıkları en temel zorluklar şu şekilde sralanabilir (Poşul, 2014):

- Yüksek entegrasyon gerektiren bilgi teknolojileri temelli sistemlerin geniş bir alanda kullanılmaya başlanılması,
- Hızla değişen teknoloji ile temellendirilmiş bilgisayarların,uygulamaların ve ağ yapılarının yüksek tehdit altıda bulunmaları,

- Teknolojik sistemlerin sürekli saldırıya maruz kalmaları,
- İşletmelerin düşük maliyetli ve yüksek verimli sistemlere ihtiyaç duymaları,
- Yasalve düzenleyici zorunlulukların bilgi güvenliği adına getirdiği yükümlülükler,
- Kurumların kaynak,beceri ve uzmanlık bakımından bilgigüvenliğini sağlamadakı yetersizlikleri

Yukarıda gösterilen zorunlukları göze alarak kurumlar, bilgi güvenliğini kapsamlı şekilde ele alan ve güvenliğin sağlanmasına dair kuralları içeren geniş standartların oluşturulması için çabalara başlamışlar (Poşul, 2014).

2.4.2 Yürürlükteki standartlar

HIPAA: Organizasyonlar kişilerin sağlık bilgilerini elektronik ortamda başka bir yere aktardıkları zaman, HIPAA standartlarının kriterlerini dikkata almak zorundadırlar. Bu organizasyonlara hastaneler, tıp hizmeti veren şirketler, tıbbi malzemeler satan veya kiralayan şirketler, eczaneler vb. kurumlar dahil olabilir. HIPAA, kişilerin sağlık bilgilerini koruma altına almak için, mahremiyet ve güvenlik taleplerine uygun olarak geliştirilmiş fiziksel, tekniksel ve idari önlemler içeren standartlar toplusudur (Aksu, 2014).

Fiziksel Önlemler: Ağ üzerinde çalışırken cihazların gereken güvenlik gereksinimlerine dikkat edilmeli, özellikle sağlık bilgileri içeren cihazlar sık sık konturol edilmeli veya ulaşım sağlanmalı ve böyle özel bilgiler içeren cihazlara yetkilendirilmiş kişiler tarafından ulaşım sağlanmalı ve kontrol edilmelidir.

Tekniksel Önlemler: Bu tür önlemlerde, özel bilgileri içeren cihazların, saldırı türlerine karşı korunması, sağlık bilgilerinin internet ortamında aktarılması zamanı şifreleme tekniklerinin kullanılması verilerin değiştirilmeyeceği ve silinmeyeceğine dair güvenliği karanti altına almakla riski yok etmeye çalışmış olur.

İdari Önlemler: Gizli saklanılan birimlerin denetlenmesi ve gereken prosedürlerin yazılması ve ayarlanması için yönetici olarak bir kişinin yetkilendirilmesi, gizli tutulan sağlık bilgilerine izin verilen kişiler dışında kimsenin ulaşmamasını sağlamak, bir başka tanıma göre çalışanların sınıflandırılması ve çalışanları gereken bilgilerle bilgilendirerek farkındalık yaratılması, gerektiği zaman verilerin acil geri

çekilebilmesi ve üzerinde gereken işlemlerin yapılması, saldırıların farkına varmak ve zararları tespitetmek gerekmektedir (Aksu, 2014).

Gram-Leach-Bliley Act: Bu standard güvenlik şirketleri, bankalar ve sigorta şirketlerinde müşterilerin özel bilgilerini koruma altına almak, gizliliğini ve güvenliğini sağlamak için yapılmış bir standartdır. Bilgi güvenliği ve mahremiyet için üç önemli prinsib vardır (Poşul, 2014).

Mali mahremiyet kuralı: Mali organizasyonlar, müşterilerinin mahremiyet bilgilerini korumak amacıyla, bu bilgilerin nasıl ve kimler tarafından kullanılabileceğini, nerede tutulacağını, güvenliği için nelerin gerektiğini belirten arşivlerin hazırlanmasını, müşteri bilgilerinin güvensizliği ortaya çıktığında müşterinin hangi haklara sahip olduğunun belirtilmesini, kurumun güvenlik ve koruma ile ilgili yaptığı ayarlamalar ve değişikliklerin müşteriye sunularak onayının alınmasını yapmaları gerekmektedir.

İhtiyat kuralı: Organizasyonların, müşterilerin mahrem bilgilerini koruma altına almak için hangi yöntemlerin kullanılması ve nasıl önlemlerin yapılmasına dair yazılı planlamaların hazırlanması ve gereken sayıda kişilerin bu iş için yetkilendirilmesi, her türlü risklere karşı önlemler yapılması ve bilgilerin korunması için risk yönetim merkezleri yapılması, bilgilerin izlenmesi, denetlenmesi, test edilmesi, toplanması, sunulması ve kullanılması için yaptığı politika değişikliğidir.

Veri çalımının engellenmesi kuralı: Diğer müşterilerin mahrem bilgilerine erişim izni olmayan kişiler için ulaşımın engellenmesidir.

BASEL: Bankaların sermaye yeterliliklerini ölçmek ve değerlendirmek için geliştirilmiş standartlar bütünüdür. Amacı bakalarının risk yönetimlerini etkin hale getirmek, piyasa düzenlemesini geliştirmek, sermaye yeterliliklerini, ölçümlerini artırarak etkili bir bankacılık sistemi yarartmaktır. Bilişim sistemlerinde riskin ortadan kaldırılmayacağı ancak gerçekleşme olasılığının minimize edilebileceği felsefesine dayanır. Bilgi güvenliğinin sağlanması için aşağıdakı maddelere önem verilir (Poşul, 2014).

Erişim kontrölü: Organizasyon dâhilinde, bilişim teknolojilerine erişim için belirlemelerin yapılması ve kimlerin nasıl erişeceğini belirtmek, kullanıcılar için hesapların oluşturulması, kaynaklara erişim haklarının ayarlanması, yönetici hesaplarının oluşturulması, kullanıcıların şifre ve hesaplarına dair standartların hazırlanmasını içerir.

İş sürekliliğinin sağlanması: Tekniksel hatalar, yazılım veya donanım hataları, elektirik kesilmesi gibi doğal ve bilinmeyen hatalara karşı gereken risk önlemlerinin yapılması, böyle hataların oluşacağı durumlar için risk yönetim merkezlerinin yapılması, bu durumlarda yasal sorumlulukların ve müşteri davamiyetinin sağlanması, organizasyona aid tüm gerekli bilgilerin güvenliğinin sağlanması ve korunması için kopyalarının önceden başka sunucularda saklanılmasıdır.

Değişiklik yöntemi: Gereken değişikliklerin tanımlanarak olası etkiler için belirlemelerin yapılması, sistemlerin hangi tarihlerde güncleneceğinin tespitini yaparak bunlar için sorumluluk taşıyacak kişilerin yetkilendirilmesi, olası hatalı durumlar için geridönüşüm prosedürlerinin oluşturulmasıdır.

Güvenlik yöntemi: Erişim hakkı olmayanların engellenmesi, bilginin korunmasını sağlamak, değiştirilmesi, silinmesi ve yok edilmesine karşı önlemeler yapmak, saldırıların engellenmesi, verinin korunmasını sağlamak için kontröl ve ölçümlerin tespit edilerek gereken tedbirlerin yapılmasıdır.

Payment Card Industry Data Security System: Organizasyonların kredi kartı işlemleri, ödemeleri ve başka işlemleri sırasında güvenliğin sağlanması ve özel bilgilerin korunması amacıyla yapılmış kurallar toplusudur (Poşul, 2014).

Güvenli bir internet ağının oluşturulması ve bakımı: Kart sahibinin özel bilgilerini koruyarak güvenliğini sağlamak amacıyla güvenlik duvarlarının yapılması ve dış kaynak organizasyonların güvenlik standartlarına itibar edilmesidir.

Kart sahibi bilgilerinin korunması: Kart sahibinin depoda tutulan bilgilerinin güvenliğinin sağlanması ve korunması amacıyla, bu bilgilerin herkese açık ağ üzerinde akması zamanı şifrelenmesidir.

Saldırılara karşı yönetim birimlerinin oluşturulması: Saldırılara ve tehditlere karşı güçlü anti-virüslerin hazırlanması ve güvenilirliği yüksek olan sistemler ve uygulamalar geliştirilerek kullanıma sunulmasıdır.

Erişim kontrol ölçütlerinin zorlaştırılması: Kullanıcının kendine açık ve bilinmesi gereken bilgilerine erişiminin kısıtlanması, bilgisayar aracılığıyla erişim için kullanıcıya yeni kimlik numarası verilmesi ve kart sahibinin kendi bilgilerine erişimin engellenmesidir.

Düzenli olarak ağın izlenmesi ve test edilmesi: Kullanıcı bilgilerine ve ağ kaynaklarına erişimin izlenmesi, güvenlik sistemlerinin ve uygulamalarının düzenli olarak denetlenmesi, test edilmesidir.

Bilgi güvenliği politikası geliştirilmesi: Veri güvenliğini kapsayan politikaların oluşturulmasıdır. (Poşul, 2014)

2.5 Bilgi Güvenliği Tehdit Eden Unsurlar

Bilgi güvenliğini tehdit eden unsurlar genel olarak aşağıdakı şekilde incelenmektedir.

2.5.1 İzinsiz erişim

Bu tür saldırılarda, bilgiye (yazılım, donanım, veri) erişim yetkisi olmayan kişiler (saldırgan) erişebilmektedir. Aynı bilgiye yetkili kurumlar veya kişiler de erişebilmekte. Bu ise bilgide heç bir bozulma veya değiştirlmenin olmadığı anlamına gelir. Bilgiye hem yetkili kişilerin, hemde erişme izni olmayan, beklenmeyen kişilerin ulaşabilmesi saldırı olarak kabul edilir ve bu en tehlikeli saldırı olarak değerlendirilir (Ülkü, 2014).

2.5.2 Engelleme veya zarar verme

Bu saldırı çeşitinde bilgi kayb edilerek, silinerek yada ulaşılmaz hale getirilerek erişimi engellenir. Ya da saldırgan, bilgiye zarar vermemişse de bilgiye erişim yetgisi olan kişilerin erişimini engellemiştir ve onlar için bilgi kullanılmaz haldedir (örn: DOS veya DDOS gibi erişim reddi saldırıları). Aktif ataktır (Ülkü, 2014).

2.5.3 Değişiklik yapma

Bu saldırı türünde, bilgi yetkili kişiye ulaşmadan önce saldırgan bilgiye ulaşar ve isteğine göre bilgi üzerinde değişlik yapar. Genel olarak program kodları, durgun veri veya başka bir yere aktarılmakta olan bilgiler üzerinde uygulanır (örn: virüsler ve truva atları). En tehlikeli ataklardan biridir. Aktif ataktır (Ülkü, 2014).

2.5.4 Üretim

Bu saldırı türünde, gerçekte olmayan ve gerçek bilgiye uygun yeni bilgi üretilir. Üretilen yeni bilgi, orjinal bilginin taklidi olması ile beraber, gerçek bilgiye uygun

tamamen yeni bir bilgi biçiminde de olabilir (örn: sahte veri, ya da veri taklidi). Aktif ataktır (Ülkü, 2014).

Bu saldırı türlerini gerçekleştirmek amacıyla virüsler, kurtlar, Truva atı, mantık bombaları, arka kapılar, mikroçipler, nano makineler, HERF silahları, EMP bombaları ve gelişen teknolojiyle meydana gelen yeni saldırı mekanizmaları kullanılmaktadır (Topal, 2004).

2.6 Bilgi Güvenliğini Sağlama Araçları

Bilgi güvenliğini sağlama yolları aşağıdaki şekilde sıralanmaktadır (Aksu, 2014):

Fiziksel güvenlik: Fiziksel tedbirler ile (güvenli ortam vb.) güvenliğin sağlanması,

Kullanıcı doğrulaması yöntemleri: Akıllı kart, tek kullanımlık parola, vb. kullanıcı doğrulama araçlarının kullanımı,

Şifreleme: Güvensiz ağlar üzerinden geçen verilerin güvenliği için şifreleme yapan donanımların kullanılması,

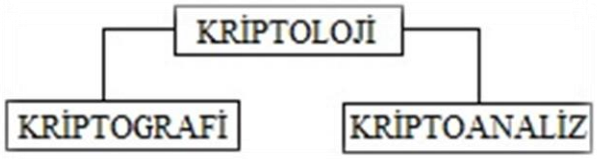
Yönetsel önlemler: Güvenlik politikaları; kurumsal, konuya özel ve sisteme özel güvenlik politikaları oluşturulması,

Standartlar ve prosedürler: Konfigürasyon yönetimi, yedekleme ve yedekleme ortamlarını saklama, olaya müdahale, iş sürekliliği ve felaket kurtarma prosedürleri,

- Elektronik imza,
- Anti-virüs sistemleri,
- Güvenlik duvarları,
- Yedekleme,
- Erişim denetimi,
- Birey eğitimleri ve farkındalık yaratma. (Aksu, 2014)

3 KRİPTOLOJİ

Kriptoloji yunan kökenlidir, kyrtops (gizli, saklı) ‘κρυπτός’’ ve logos (bilim) ‘’λόγος’’ sözcüklerinin birleşiminden oluşmuştur, kısaca şifre bilimi demektir. Kriptoloji, önemli bilgilerin şifrelenmesi, aynı zamanda şifrelenmiş bilgilerin deşifre edilmesi için geliştirilmiş metotlar, teknikler bütünüdür. Kriptoloji, kriptografi ve kriptanaliz olarak iki alt bilim dalından oluşmaktadır. Şekil 3.1’de Kriptolojinin alt bilim dalları verilmiştir.



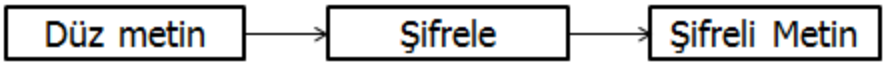
Şekil 3.1 Kriptolojinin alt bilim dalları

Kriptografi alt bilim dalı, açık olan bilgiyi (düz metin) anlaşılamaz hale (şifre metin) getirmek için kullanılan şifreleme bilimidir. Kriptanaliz ise şifrelenmiş bir metinin üzerinde anahtarı bilmeden açık metini elde etmek için kullanılan metot ve teknikler toplusu olan kriptoloji alt bilim dalıdır.

3.1 Kriptografi Nedir?

Kriptografi, Yunanca ***kryptos*** (gizli, saklı) ve ***graphein*** (yazmak) sözcüklerinin birleşmesinden oluşmaktadır. Kriptografi, gizlilik, kimlik denetimi, bütünlük gibi bilgi güvenlik kavramlarının sağlanması için uğraşan matematiksel yöntemler bütünüdür (Çimen, Akleylek, & Akyıldız, 2007). Kriptografik yöntemler bilgiyi, bilgi göndericisini ve alıcısını, bilginin saklanması ya da iletilmesi zamanında tüm saldırılardan korur. Başka bir deyişle kriptografi, okunabilecek bir bilgiyi, istenmeyen kişiler tarafından okunmasını önlemek için okunmaz hale getiren matematiksel yöntemler bütünüdür. Kriptografi bilim dalı ile uğraşan kişilere kriptograf denilmekte. Kriptografi, bilgiyi anahtar adı verilen ifadeyle şifreler ve anahtar ifade olmadan bilginin okunmasını imkânsız hale getirir. Şifreleme işleminde

kullanılan anahtar (key) ifadesi, gizliliği gereken bilgiyi şifrelemek ve ya şifrelenmiş bilgiyi deşifre etmek için kullanılan sayı, harf, kelime, sembol olabilir (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007). Şekil 3.2'de şifreleme işleminin genel anlatımı gösterilmektedir.



Şekil 3.2 Genel Şifreleme İşlemi

Önce düz metin şifreleme işlemine giriyor sonra şifreli metin oluşuyor. Şifreleme işlemleri farklı şekilde farklı tekniklerle yapılabilir. Düz metin, metinin şifrelemeden önceki okunulabilir halidir. Düz metin şifreleme işlemlerinden geçirildikten sonra oluşan okunulmaz hali şifreli metin adlanır.

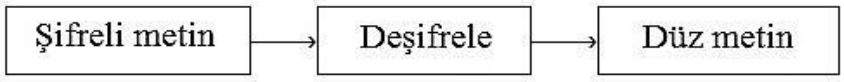
Bir şifreleme algoritmasının taşıması gereken bazı temel özellikleri aşağıda tanımlanmıştır:

- *Gizlilik:* Bilgi sadece ilgili kişiler tarafından anlaşılmalıdır, bilgi anlaması istenmeyen kişiler tarafından anlaşılmamalıdır (Ülker & Coşkun, 2014) (Ülkü, 2014).
- *Bütünlük:* Bilginin herhangi bir değişikliğe uğrayıp uğramadığını ifade eder. Bilginin saklanması ya da iletilmesi esnasında yetkili kişiler dışında başka kişilerce değiştirilememesi veya değiştirilirse bunun açığa çıkmasıdır (Ülker & Coşkun, 2014), (Ülkü, 2014).
- *İnkâr edememe:* Bilgiyi üreten ya da ileten kişinin daha sonrasında bunu inkâr edememesi durumudur (Ülker & Coşkun, 2014), (Ülkü, 2014).
- *Kimlik doğrulama:* Bilginin iletiminde karşılıklı iki tarafta da kimlik doğrulama kullanılırsa, üçüncü bir kişinin bilgiye ulaşmasını engellemek için kimlik doğrulama bilgilerine ulaşmaması gereklidir (Ülkü, 2014). Ayrıca kimlik doğrulama, inkâr edememe ile de yakından bağlantılıdır.
- *Erişilebilirlik/Süreklilik:* Yetkili kişilerin ihtiyaç duyduğu anda bilgiye ulaşabilmesidir (Ülkü, 2014).

Verilen özelliklere göre kriptografi, bilgilerin gizlenmesiyle beraber veri bütünlüğü, kimlik kanıtlama ve inkâr edememe konuları ile de ilgilenir (Ülkü, 2014).

3.2 Kriptanaliz Nedir

Kriptanalizin görevi, değişik yöntemler uygulayarak kriptogrfik şifreleme sistemlerinin çözülmesi ve şifreli metinlerin okunulabilir hale getirilmesidir. Kriptografik şifreleme sistemlerinin incelenmesi ve sistemin ne kadar güvenli olduğunu değerlendirmek için kriptanaliz teknikleri kullanılır. Kriptanaliz yöntemleri ile okunulabilir hale getirmek için üzerinde çalışılan metin *şifremetin*, kullanılan yöntemler toplusu *deşifreleme işlemi*, elde edilen açık metin ise *düzmetin* olarak adlandırılmıştır. Kriptanalizle uğraşan bilim adamlarına kriptanalist denir. İyi bir kriptanalist aynı zamanda iyi bir kriptograf olmak zorundadır. Bir şifreleme algoritmasını deşifre etmek için, o algoritmanın şifreleme tekniklerini adım adım bilmek gerekir. Kriptanalistlerin kazandığı uğurlar tarih boyunca kriptografları daha güvenli algoritmalar geliştirmeleri için zorlamıştır (Çimen, Akleylek, & Akyıldız, 2007). Şekil 3. 3'de şifreli metinin düzmetine çevrilme adımları sembolik olarak verilmiştir.



Şekil 3.3 Şifre çözme işlemi

M.Ö. yapılan kriptografi şifrelerin kırılması ile başlayan kriptanaliz çalışmaları anahtar bilinmeden yapılan çalışmalardır. Kriptanalizin tarihine bakıldığında ilk çalışmaların 9. yüzyılda yaşayan Arap filozofu Al-Kindi'nin yazdığı *Kriptografik Mesajların Deşifresi* (Risâle fi'stihrâci'l-mu'ammâ) isimli eserde olduğu görülüyor. Bu eser İstanbul'da, Süleymaniye Osmanlı Arşivi'nde bulunmaktadır. Al-Kindi bu yazısında harf frekansı analizi kavramını ortaya çıkarmıştır. O, şifrelenmiş metinle aynı dilde olan yeterince uzun bir metin seçerek metindeki harflerin kullanım sıklığını hesaplamış ve şifreli metinde de aynı işlemi yapmıştır. Sonra seçilmiş metindeki en sık kullanılan harf, şifre metindeki en sık kullanılan şifrelenmiş harfe denk olarak kriptanaliz işlemini yapmıştır (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007).

3.3 Kriptolojinin tarihçesi

Kriptoloji yunancada gizli anlamına gelen *κρυπτός* (kriptos) ve yazı anlamında kullanılan *γράφειν* (grafein) sözcüklerinden oluşuyor (Konheim, 1981). Tarihini incelediğimizde çeşitli kaynaklardan edinilen bilgiler şifreleme işlemlerinin eskilere, miladdan önceki tarihe dayandığını gösteriyor. Önemli bilgilerin ortaya çıkması (askeri sırlar vs) ve bu bilgileri korumaya alma gereksinimi neticede kriptolojinin temel örneklerini ortaya çıkardı. O dönemlerde yalnızca askeri ve haberleşme alanında kullanılmasına rağmen teknoloji devriminin gelmesiyle büyük güvenlik sorunları ortaya çıktı ve bu da kriptografinin önemini çok fazla artırdı. Zamanımızda kriptoloji güvenlik acısından vazgeçilmeyecek şekilde kullanılmaktadır.

Eski zamanlarda kullanılan şifreleme sistemlerinin bir kısmı alfabedeki harfleri belli bir sayı ile ifade ederek oluşturuluyordu, örnek olarak en çok bilinenleri İbrani-Süryani, Grek ve Latin harf-sayı sistemidir. Daha sonra sözü geçen harf-sayı sistemi Arap alfabesine uygulanarak "*Ebcet hesabı*" adlandırılan bir sistem yapılmıştır. Ebcet hesabında, harflerin her birine 1'den 1000'e kadar sayısal değerler verilmiştir. İlk 9 harfe 1-den 9-a kadar, ikinci 9 harfe 10-dan 90-a kadar onluk değerler, üçüncü 9 harfe 100-den 900-e kadar yüzlük değerler ve sonuncu harfe 1000 değeri verilmiştir (Çimen, Akleylek, & Akyıldız, 2007).

Eski Yunan tarihçisi Heredotus'un yazdığına göre M.Ö. 480 yılında Yunanlar ve Persler arasındakı savaşta Stenografi (Yunanca "gizlenmiş yazı") adı verilen teknik kullanılmıştır. İranda yaşayan Yunanlı, kölelerinden birinin saçlarını kazıtarak yunanlara karşı düzenlenmiş Pers istilası planını onun kafası üzerine yazarak saçları uzadıktan sonra Atina'ya gönderiyor ve yunanlar kölenin saçlarını keserek haberi okuyorlar. Bu haber sayesinde İranlıların planına karşı hazırlanan Yunanlılar savaşı kazanıyorlar (Çimen, Akleylek, & Akyıldız, 2007), (Kahn, 1967).

M.Ö. 5. yüzyılın başlarında askeri amaçla kullanılan ilk şifreleme sistemi yunanlıların *skytale* adı verdikleri bir kriptografik cihaz olmuştur. Şifreleme işlemini yapmak için bir sopa ve uzun bir papirüs gerekmekte idi. Papirüsü silindirik sopanın üzerine sardıktan sonra şifrelenecek mesaj uzununa sopanın üzerine yazılıyordu. Papirüs açıldıktan sonar ise üzerinde şifrelenmiş anlamsız metin oluşuyordu. Şifreçözme işlemini yapmak için de aynı ölçüde bir sopa gerekiyordu. Sopanın azıcık farklı olması doğru mesaja ulaşmanı engelliyordu. Mesajı okumak isteyen kişi

papirüsü aynı ölçütte sopanın üzerine sardıktan sonra anlamlı metine ulaşıyordu (Şen, 2006), (Nicholas, 2015), (Simon, 2001).

Skytaleden sonra Yunanlar tarafından (M.Ö. 205-123) tasarlanan şifreleme sistemi tarih sayfalarında yerini bulmuş Polybius'un *dama tahtası şifrelemesi* olmuştur. Polybius'un şifreleme sisteminde alfabe olarak Yunan ve Roma alfabesi kullanılıyordu. Sistem 5x5'lik matristen oluşuyordu ve her harfe iki sayı karşılık geliyordu. Sayılardan birincisi satırı, ikincisi sütunu göstermekteydi (Çimen, Akleylek, & Akyıldız, 2007).

Diğer önemli gelişmelerden biri M.Ö. 60-50 yılları arasında haberleşme amacıyla askeri alanda kullanılan Büyük Roma İmparatoru Julius Caesar (Sezar)'a ait şifreleme sistemidir. Sezar, komutanları ile iletişimi sağlamak için *Sezar şifrelemesi* olarak adlandırılan şifreleme sistemini kullanıyordu. Bu şifreleme sistemi ile her hangi bir metni şifrelemek için harf değiştirme işlemi yapılıyordu. Her harf düz alfabenin üç harf sola kaydırılmasından oluşan şifre alfabedeki karşılığıyla, yani kendisinden sonrakı üçüncü harfle değiştiriliyordu. Şifrelenmiş metni alan kişi şifre çözme işlemini yapmak için şifre metindeki her harfi alfabe sayının üç eksiği kadar sola kaydırdıktan sonra düz alfabedeki aynı yerde duran harfi almaktaydı. Böylece şifre çözme işlemi de kolaylıkla yapılabiliyordu (Sulak, Turan, & Demiröz, 2013), (Nicholas, 2015).

Zamanımızda olduğu gibi geçmiş dönemlerde de şifreleme işlemlerine olan ihtiyaç kadar o şifreleme sistemlerini, şifre metinleri çözmeye de ihtiyaç duyuluyordu ve bu işlerle ilgilenenlerin sayı artmaktaydı. İlk şifre çözme işlemlerini Arap filozofu Al-Kindi yazdığı "*Kriptografik Mesajların Deşifresi"* (Risâle fi'stihrâci'l-mu'ammâ) isimli yazısında araştırmıştır. Al-Kindi araştırma yaptığı bu yazıda kriptanaliz araştırmalarının temelini koymuş, frekans analizi kavramını ortaya atmıştır. Çalışması İstanbul'da, Süleymaniye Osmanlı Arşivi'nde bulunmaktadır. Al-Kindi'nin araştırdığı frekans analizine göre şifre metinin yazıldığı dil bilindikten sonra aynı dilde yazılmış yeteri kadar uzun bir metindeki harflerin kullanım sıklığına bakıldığında en çok kullanılan harf şifre metindeki en çok kullanılan harfe denk gelmekteydi. Bu araştırma tekniği ile Al-Kindi tek alfabeli şifreleme sistemlerini güvensiz hale getirmiştir ve şifreleme sistemleri için yeni dönem başlamıştır. Tek alfabeli sistemler güvensiz hale geldiğinden kriptograflar çok alfabeli şifreleme

sistemleri geliştirmeyi düşünmeye başlamışlardır (Şen, 2006), (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007).

İlk Çok alfabeli şifreleme sistemini Leon Alberti (1404-1472) 1466-1467 yıllarında harf kaydırma tekniğini kullanarak geliştirmiştir. Harf kaydırma işlemi *Alberti diski* ile yapılan bu sistemde harflerin kaydırılma miktarı kullanıcının isteğine göre belirlenmekteydi. Bu diskin iç çemberi sabit, dış çemberi ise hareket ettirilebilir ve harflerin değişik miktarda ötelenmiş hali görülebilirdi (Ülkü, 2014), (Nicholas, 2015).

Yeni çok alfabeli şifrelerin geliştirilmesine bakılmaksızın onlar uzun yıllar güvenliği koruyamıyordu ve güvensiz hale geliyordu. İlk uzun zaman kırılmayan çok alfabeli şifreleme sistemi 1553 yılında Giovan Batista Belaso adlı bir İtalyan kriptograf tarafından geliştirilmiştir. 1586 yılında Blaise De Vigenere bu sistemi biraz daha geliştirerek uzun zaman kırılmayan ve *Vigenére Şifresi* adlandırılan yeni bir şifreleme sistemi geliştirdi. Bu şifreleme sistemi tek alfabeli şifreleme sistemlerinden çok farklı idi ve bu sisteme frekans analizini uygulamak mümkün değildi. Bu sistemle düz metindeki her harf için farklı alfabe oluşturularak şifreleme işlemi yapılıyordu. Şifre alfabeler anahtar kelimeye göre seçildiğinden düz metindeki aynı sözler için şifre metinde farklı sözler karşılık geliyordu ve böylece frekans analizi ile bu sorunu çözmek mümkün olmuyordu. Vigenere bu şifreyi keşfetmekle çok güvenilir ve uzun yıllar kırılmayan bir şifreleme sistemi geliştirmişti. Bu sistem iki yüz yıldan fazla kırılmaz bir sistem olarak kaldı. 18. yüzyılın sonlarında Vigenre şifresi Babbage ve Kasiski tarafından kırıldı ve güvenilirliğini yitirdi. Bu analizislerin dikkatini çeken belirli bir döngüden (anahtar kelimedeki harf sayı) sonra aynı şifre alfabenin kullanılması olmuştur (Ülkü, 2014), (Nicholas, 2015), (Kahn, 1967).

1790.yılda Tomas Jefferson "*Jefferson Diski*" adı verilen yeni bir şifreleme sistemi oluşturdu. İngiliz alfabesi 26 harften oluştuğundan dolayı Jefforson Diski harflerin rastgele yer alıdığı 26 diskten ibaret idi. Disk üzerinde anahtar kelime ve şifremetin yazıldıkdan sonra disk karıştırılıyordu ve çok anlamsız hale gelen şifremetin oluşmuş oluyordu. Düzmetine ulaşmak isteğen kişi diskin aynısı üzerinde anahtar kelimeni oluşturduktan sonra düzmetine ulaşıyordu. Şifremetin ve anahtar kelme düşman eline geçtiğinde de düzmetine ulaşmak için şifreleme yapılan diskin aynısını kullanması gerekmeyteydi. Böylece her disk diğerinden farklı olduğu için gönderilen haber

alıcıya güvenli şekilde ulaştırılıyordu (Nicholas, 2015), (Çimen, Akleylek, & Akyıldız, 2007).

1854 yılında Charles Wheatstone ve Baron Lyon Playfair 5x5 lik bir matris kullanarak *Wheatstone-Playfair şifresi'*ni tasarlıyorlar. İngiliz alfabesi için tasarlanan bu sistemde 25 hücre olduğu için İ ve J harfleri bir arada ve her hücreye bir harf gelmekle alfabenin tüm harfleri hücrelere giriliyor. İngilizlerin askeri alanda kullandıkları Playfair şifreleme sistemi 1900'lü yılların başlarına kadar güvenliğini sağlasa da sonrasında harf ikililerinin frekans dağılımı kullanılarak deşifre edilmiştir (Çimen, Akleylek, & Akyıldız, 2007).

Tarihte şifreleme sistemlerinin güvenliğinden ve praktikliğinden ilk defe Hollandali kriptograf Auguste Kerskhoffs 1883 yılında yayınladığı "*La Cryptographie Militarie"* makalesinde bahs etmiştir. Yazısında bir şifreleme sisteminin anahtar sözcüğünden başka her şeyi bilinse bile güvenilirliğini sağlaması gerektiğini belirtmiştir. Bu makale yayınlandıktan sonra tasarlanan şifreleme sistemleri için aşağıda gösterilen Kerskhoff Prensipleri ortaya konmuştur (Çimen, Akleylek, & Akyıldız, 2007).

- Sistem pratik ve matematiksel bir gerçekliğe dayanmalıdır.
- Sistemde kullanılan anahtarlar taraflar arasında kolayca, üçüncü kişinin değiştirmesine izin verilmeden değiştirilebilmelidir.
- Sistem telegraf uygulamasında kullanılabilmelidir.
- Sistemin kullanılabilmesi için fazla sayıda insana ihtiyaç duyulmamalıdır.
- Sistemin uygulaması ve anlaşılması kolay olmalıdır.
- Şifreleme sisteminin güvenliği, şifreleme algoritmasını gizli tutmaya dayanmamalıdır. Yani sistem hakkındaki her şey herkes tarafından bilinse bile güvenilirliğini korumalıdır. Güvenlik; yalnızca anahtarı gizli tutmaya dayanmalıdır (Çimen, Akleylek, & Akyıldız, 2007).

1800'lü yılların sonlarına doğru teknolojinin gelişmesi kriptolojinin önemini daha da artırmakta idi. O dönem teknolojisi için çok önemli bir gelişme olan İtalyan fizikçi Markoni'nin keşfettiği telsiz cihazı kablo kullanılmadan iller arası haberleşme imkânını sağlıyordu. Bu aletin keşfi askeri alan için savaşlarda haberleşmeyi kolaylaştırması bakımından çok önemli idi. Ancak bu alet haberleşmeyi kolaylaştırmasıyla beraber büyük güvenlik sorunları da getirdi. Telsizin her yana yayılma özelliği olduğundan düşmanın da mesaja ulaşma olasılığı ortaya çıkıyordu.

Telsizin bu zayıflığı bilginin güvenli bir sistemle şifrelenmesi ihtiyacını ortaya çıkardı. Yeni yöntemler arama ihtiyacı kriptografinin gelişimine de katkıda bulunmuştur (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007).

Birinci dünya savaşında telsizle güvenli haberleşmeyi sağlamak için kriptografların birçok yeni şifreleme sistemi geliştirmesine rağmen, bunların hepsi güvenliğini koruyamadı ve kırıldı. O dönemlerde almanlar tarafından kullanılan ve daha güvenli sayılan şifreleme sistemlerinden biri 1918 yılında tasarlanan ADFGVX sistemi idi. Buna karşılık Fransızların en ünlü kriptanalistlerinin almanların kırılmaz saydığı şifreleme sistemini çözmek için büyük çabaları vardı. Onlardan en ünlüsü sayılan George Painvin durmadan çalışarak kırılmaz sayılan bu şifreleme sistemini çözdü ve almanların telsizle haberleşdikleri büyük miktarda şifreli mesajları ele geçirip deşifre etti. Kriptanalistlerin bu önemli çalışmaları almanların yenilgisine sebep oldu (Simon, 2001).

1917'de İngiltereli kriptanalistlerin Almanya Dışişleri Bakanı Arthur Zimmermann'ın Meksika Başkanı'na çekmiş olduğu telgrafı deşifre etmesi Birinci Dünya Savaşını değiştiren bir kriptanaliz oldu. Almanların Avrupa güçlerini ele geçirerek savaşı kazanma planları vardı. Savaş zamanı almanların gizli planından habersiz olan ve ingilizlerin müttefiki sayılan Amerka savaşa katılmayacaktı. Ama almanların İngiltere üzerinden gönderdiği mesajları deşifre eden ingiliz kriptanalistler haberi Amerikanın o zamanki başkanı Woodrow Wilson'a duyurdular ve Amerika savaşa katılma kararı aldı. Tarihe "Zimmermann Telegrafı" ismiyle giren bu olay Birinci Dünya Savaşının seyrini değiştirdi (Nicholas, 2015), (Simon, 2001), (Çimen, Akleylek, & Akyıldız, 2007).

Birinci Dünya Savaşı zamanı kriptanalistler kriptograflardan daha etkili olduklarını gösterdiler ve savaşı kazandılar. O sırada ünlü kriptograflar yeni kırılmaz ve dayanıklı bir güvenlik sistemi geliştirmek için çok çaba harcıyorlardı. 1917 yılında Amerikalı mühendis Gilbert Vernam, Vernam şifrelemesi adıyla tanınan yeni bir şifreleme tekniği geliştirdi. Vernam şifreleme sisteminde şifreleme adımları Vigenere şifrelemesine benzemekteydi. Vernam şifrelemesinin avantajı anahtarın düzmetinle aynı uzunlukda olması idi. Vigenere şifresinde yapılan Kriptanaliz işlemleri bu sistem için geçerli olmamaktaydı. Vernam Şifresine Kriptanaliz yapmak için önce şifrelemede kullanılan anahtardaki harfleri doğru bulmak gerekiyordu. Anahtar uzun

olduğundan dolayı zor bir anahtar seçildiği takdirde bu şifreleme sistemi kırılmaz bir güvenliğe sahip oluyordu (Nicholas, 2015), (Çimen, Akleylek, & Akyıldız, 2007).

Kağıt-kalemle yapılan şifreleme sistemlerinin Birinci Dünya Savaşı zamanı güvenilirliğini sağlayamamasının ardından bir çok devlet şifrelemenin artık makinalarla yapılması gerektiğini düşündüler. 1918 yılında alman mühendis Arthur Scherbius makinalarla şifrelemenin temelini koydu ve şifresinin kırılmayacağını düşündüğü Enigma adlı makinayı tasarladı. O dönem için makina fiyatının çok yüsek olduğundan alıcılar tarafından iyi karşılanmadı.

Makinasının beklenen ilgiyi görmemesinin ardından Scherbius Alman ordusuyla anlaşma kararı aldı. İngilizlere ait bir belgeyi Alman ordusuna sunduktan sonra 1926 yılında makinalarını onlara satmaya başladı. 1943 yılında ingilizler ilk elektron bilgisayar sayılan Colossus adı verdikleri şifre çözme makinasıyla Enigmayı çözmeyi başardılar. Daha sonra Enigmanın şifrelediği mesajları çözmek için şu anda kullanılan bilgisayarların temeli sayılan Eniac'ı yaptılar (Simon, 2001), (Çimen, Akleylek, & Akyıldız, 2007).

1929 yılında Leste Hill çok alfabeli şifreleme sisteminin daha pratik hali olan *Hill şifresini* tasarladı. Bu sistemde, Claude Shannon'un 1949 yılında öne sürdüğü "güvenli bir şifreleme sistemi karıştırma işlemini iyi yapmalıdır" fikri sağlanmıştır. Bu sistemde her harfin bir sayı karşılığı vardır ve şifrelenecek metin alt gruplara bölünerek seçilmiş anahtar matrisleriyle şifreleniyor. Sonradan Leste Hill ortağıyla beraber 6x6'lık bloklarla Hill şifresini uygulayabilen yeni bir makine geliştirmiştir, ancak bu makine ilgiyle karşılanmamış ve çok fazla talep görmemiştir (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007).

Harflerin bitlerle ifade edildiği ilk modern kriptografi 1970 yılında IBM labaratuvarında önceler Demon adı verilen, sonraki zamanlarda Lucifer adı verilen 64 bitlik anahtarlı yeni şifreleme sistemi geliştirildi. Bu sistem 1975 yılında Amerika'da Birleşik Bilgi İşleme Standardı olarak seçilmiştir (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007).

1973 yılında elektronik haberleşmenin yaygınlaşması ile Amerika Milli Standartlar Bürosu Herkes tarafından kullanabilecek bir şifreleme sistemine ihtiyaç duyulduğunu ilan etmiştir. Yeni tasarlanacak algoritmanın adı da büro tarafından "Veri Şifreleme Standardı (Data Encrypt System(DES))" olarak belirlenmişti (Ülkü, 2014), (Çimen, Akleylek, & Akyıldız, 2007).

1975 yılında yayınlanan DES algoritması, 1976 yılında Veri Şifreleme Standardı olarak kabul edildi. DES algoritması Feistel yapısı aracılığıyla Shannon'un karıştırma önerisinin özelliklerini sağlamaktadır. Vernam şifresinden başlayarak tasarlanan yeni modern şifreleme sistemlerinde bilgisayar aracılığıyla harflerin bitlerle şifrelenmesi işlemi kullanılmaya başlandı. DES algoritması ikilik tabandakı bir düzmetni 64 bitlik bloklar halinde parçalayıp 56 bitlik anahtarla şifreleme yapıyor. Anahtarın gizliliği ve rastgele seçilmesi bu sistemin güvenilirliğini sağlıyor. Şifreleme anahtarını elde edebilen kişi rahatlıkla şifre çözme anahtarını da elde ederek düzmetine ulaşması mümkündür. DES algoritmasının Shannon'un yayılma ve karıştırma özelliğini sağlaması onun en önemli tarafıdır. Karıştırma işlemi her anahtar için düzmetin ve şifremetin arasında istatistiksel bağlantı olmamasını sağlar. DES için bu özellikler güvenlik açısından çok önemlidir (Ülkü, 2014), (Simon, 2001).

1976 yılında alıcı ve gönderici arasında anahtar paylaşımı işleminin daha kolay bir şekilde yapılması için Whitfield Diffie ve Martin Hellman yeni bir algoritma yaptılar. Diffie-Hellman anahtar değişim algoritması, sayılar teorisi yardımıyla şifrelemede açık anahtarın kullanılabileceğini kanıtlamış ve böylece açık anahtarlı kriptografinin temelini koymuşlar. Bununla da anahtar paylaşımı zorunluğu ortadan kalkmış ve bu algoritmayla şifreleme yapan kişiler kendilerine ait özel anahtarlarını kullanmışlar (Ülkü, 2014), (Nicholas, 2015), (Simon, 2001).

Diffie-Hellman anahtar değişim algoritmasının tasarlanmasından sonra artık şifrelemede herkes kendi özel anahtarını kullana bilicekti. Bu gelişme kriptografları daha da çok çalışmaya ve yeni açık anahtarlı bir şifreleme algoritması tasarlamaya sevk ediyordu. 1977 yılında Ronald Rivest, Adi Shamir ve Leonard Adleman, RSA (Rivest-Shamir-Adleman) adı verdikleri açık anahtarlı bir şifreleme algoritması tasarladılar. RSA algoritması Diffie-Hellman anahtar değişimini içeren ilk şifreleme sistemi oldu. Bu algoritma matematikte zor problemlerden sayılan çarpanlara ayırma problemine dayanarak tasarlanmıştır (Nicholas, 2015), (Çimen, Akleylek, & Akyıldız, 2007).

Günümüzde de kullanılan RSA şifreleme sistemi kullanılmaya başladığı dönemlerde uygulanması kolay ve kırılması zor olmasına bakılmaksızın güvenlik açısından daha da karmaşıklaştırılıyordu. Bit uzunluğunun 128 bitten 512 bite çıkması büyük bir matematiksel hesaplama işlemi gerektiriyordu. RSA'nın bu zayıf yönlerini göze alan

kriptograflar yeni açık anahtarlı şifreleme sistemi tasarlamağı düşünüyorlardı. 1985 yılında Neal Koblitz ve Victor S. Miller tarafından tasarlanan Elliptik Eğri algoritması RSA'ya göre daha az bit kullandığından hızlıydı ve en az RSA kadar güvenliğe sahipti (Çimen, Akleylek, & Akyıldız, 2007).

1990 yıllarında DES ve ona benzer yapıdakı şifreleme sistemlerinin güvenliği sona ermiştir. Kriptanalistlerin yeni geliştirdikleri Linear ve Differensial Kriptanaliz yapıları bu sistemlerin bazı anahtarların kullanımında güvensiz olduklarını kanıtlamıştır (Çimen, Akleylek, & Akyıldız, 2007).

1997 tarihinde Amerka Ulusal Standardlar ve Teknoloji Enstititüsü yeni bir Gelişmiş Şifreleme Standartı (AES) yarışması ile DES'in yerini alacak yeni şifreleme sistemi seçeceğini duyurmuştur. 2000 yılında yapılan başvurular arasından diğerlerinden daha güvenilir ve hızlı olan, iki Belçikalı kriptograf Joan Daemen ve Vincent Rijmen'in tasarladığı AES algoritması seçilmiştir. AES algoritması şimdiye kadar güvenliğini korumakta ve kullanılmaktadır (Çimen, Akleylek, & Akyıldız, 2007).

Kriptolojiyle ilgili yapılan ilk konfrans bilim adamları tarafından 1981 yılında California Santa Barbara Üniversitesi'nde CRYPTO 81 adı altında gerçekleşmiştir. Bu konfransla beraber ilk defa 1982 yılında Almanya'da düzenlenen EUROCRYPT, ilk defa 1990 yılında Avusturalya'da düzenlenen ASIACRYPT, ilk defa 2000 yılında Hindistan'da düzenlenen INDOCRYPT isimli konferanslarda her sene gerçekleştirilmektedir. Bunlardan başka 2005 yılında ODTÜ Uygulamalı Matematik Enstitüsünde düzenlenmesine başlanılmış Ulusal Kriptoloji Sempozyumu da her yıl gerçekleştirilmektedir (Çimen, Akleylek, & Akyıldız, 2007).

Dünyada kriptografi alanında eğitim ve araştırmaya yönelik ilk ders kitapları 1987 yılında çıkmıştır (Ülkü, 2014).

3.4 Kriptografi Algoritmalarının Sınıflandırılması

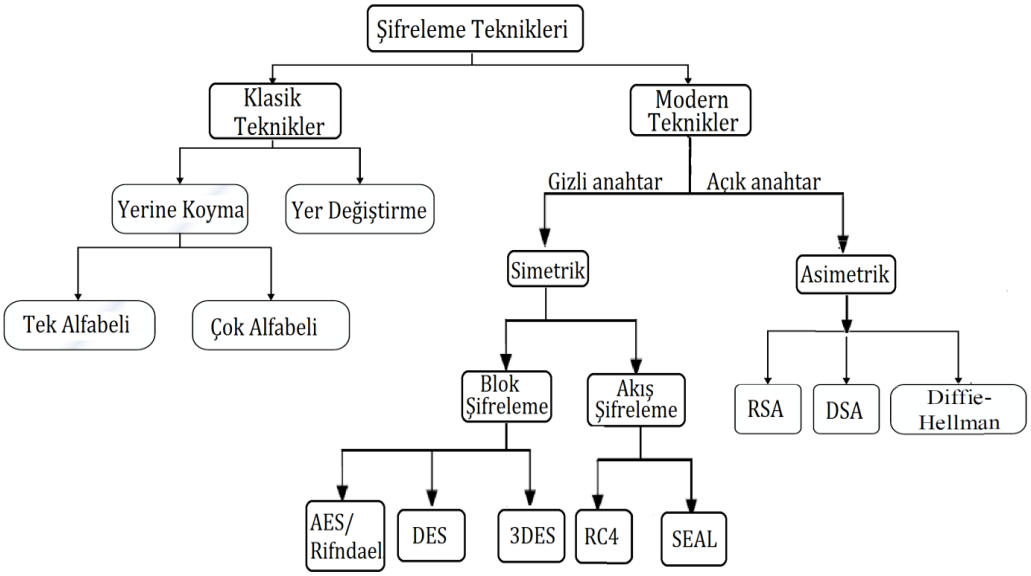
Tarih boyunca kriptolojinin yenilenmesi ve son yüzyılda ise teknolojinin hızla gelişmesi nedeniyle yeni şifreleme yöntemlerinin ortaya çıkması sonucunda, kriptografi iki ana kısma ayrıldı (Bayar, 2012).

- Klasik teknikler
- Modern teknikler

3.4.1 Klasik kriptografi teknikler

Bilgisayarların bulunmasından önce, kâğıt-kalemle gerçekleştirilen klasik kriptografi teknikler kullanılıyordu. Klasik tekniklerin en önemlisi II. Dünya Savaşı zamanında kullanılan ENIGMA olmuştur (Tuncal, 2008). Klasik kriptografi teknikler daha çok askeri alanda ve akademik kurumlarda kullanılmıştır.

Klasik kriptografi tekniklerde kullanılan algoritmalar, gizli saklanıldığından dolayı



Şekil 3.4 Şifreleme Algoritmalarının sınıflandırılması (Bayar, 2012).

güvenilirliği düşük algoritmalar olarak değerlendiriliyor (Menezes, Oorschot, & Vanstone, 1996).

Başlıca klasik teknikler, Sezar, Playfair, Hill, Vigenere, Albert Diski, Vernam, Enigma'dır (Dalkıç & Akın , 2005).

3.4.1.1 Sezar şifreleme

Sezar şifreleme tekniğinde, alfabedeki harflerin sıra numaraları 3 karakter sola kaydırılarak yeni alfabe oluşturulur. Bu teknik tek alfabeli şifreleme tekniğidir ve ötelenme şifreleme tekniği olarak da bilinmektedir (Ülkü, 2014) (Başar, 2004). Matematiksel olarak "$E_k(m) = (m + k) \bmod 26$" şeklindedir. Burada *m*, düz metindeki harfin düz alfabedeki sıra numarasıdır. *k* harflerin kaydırılma miktarıdır. E_k

şifrelenmiş metindeki harflerin düz alfabedeki sıra numarasıdır. 26 ise İngiliz alfabesindeki toplam harf sayısından gelmektedir. Türkçe alfabe için uygulanırsa bu sayı 29 olarak kullanılmalıdır (Ülkü, 2014).

Örnek olarak Türkçe alfabe için hazırlanmış küçük bir uygulama verilmiştir.

Ö$\boldsymbol{rnek}$:

$a\ b\ c$ 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 $metin:\ savaş \rightarrow\ 21, 0, 26, 0, 22$

$k = 3\ için;$

$E_k(21) = (21 + 3)\ mod 29 = 24$

$E_k(0) = (0 + 3)\ mod 29 = 3$

$E_k(26) = (26 + 3)\ mod 29 = 0$

$E_k(0) = (0 + 3)\ mod 29 = 3$

$E_k(22) = (22 + 3)\ mod 29 = 25$

$Şifreli\ metin:\ 24, 3, 0, 3, 25 \rightarrow\ uçaç$ü

"Savaş" sözü şifrlendikte "uçaçü" olarak elde edilmiştir.

3.4.1.2 Alberti diski

İlk Çok alfabeli şifreleme sistemini 1404-1472 yılları aralığında yaşamış Leon Alberti 1466-1467 yıllarında harf kaydırma tekniğini kullanarak geliştirmiştir. Harf kaydırma işlemi *Alberti diski* ile yapılan bu sistemde, harflerin kaydırılma miktarı kullanıcının isteğine göre belirlenmekteydi. Bu diskin iç çemberi sabit, dış çemberi ise hareket etdirilebilir ve hariflerin değişik miktarda ötelenmiş hali görülebilirdi (Ülkü, 2014), (Nicholas, 2015).

3.4.1.3 Vigenere şifreleme

İlk uzun zaman kırılmayan çok alfabeli şifreleme sistemi 1553 yılında Giovan Batista Belaso adlı bir İtalyan kriptograf tarafından geliştirilmiştir. 1586 yılında Blaise De Vigenere bu sistemi biraz daha geliştirerek *Vigenére Şifresi* adlandırılan yeni bir şifreleme sistemi geliştirdi. Bu şifreleme sistemi tek alfabeli şifreleme sistemlerinden çok farklı idi ve bu sisteme frekans analizini uygulamak mümkün değildi. Bu sistemle düz metindeki her harf için farklı alfabe oluşturularak şifreleme işlemi yapılıyordu. Şifre alfabeler anahtar kelimeye göre seçildiğinden düz metindeki aynı

sözler için şifre metinde farklı sözler karşılık geliyordu ve böylece frekans analizi ile bu sorunu çözmek mümkünsüz hale geliyordu. Vigenere bu şifreyi keşf etmekle çok güvenilir ve uzun yıllar kırılmayan bir şifreleme sistemi geliştirmiştir. Bu sistem iki yüz yıldan fazla kırılmaz bir sistem olarak kaldı. Sonralar bu şifreleme sistemi de (18.yüzyılın sonları) Babbage ve Kasiski tarafından kırıldı ve güvenilirliğini itirdi. Bu analizislerin dikkatini belirli bir periottan (anahtar kelimedeki harf sayı) sonra aynı şifre alfabenin kullanılması olmuştur (Ülkü, 2014), (Nicholas, 2015), (Çimen, Akleylek, & Akyıldız, 2007), (Kahn, 1967).

Çizelge 3. 1'de Türkçe alfabe için oluşturulmuş Vigenere Tablosu verilmiştir.

Türkçe alfabe için yapılmış örnek:

Ö$\boldsymbol{rnek}$:

$Anahtar: Vigenere$

Düz $metin$: "$Ayna\ ilk\ defa\ do$ğ$uda\ yapıldı$" $olsun.$

Anahtar kelime 8 harf olduğundan düzmetini 8 harften oluşan kelimelere bölelim.

$AYNA$İ$LKD - EFADO$Ğ$UD - AYAPILDI$

Anlamsız kelimeleri oluştu. Şimdi işlemi yapalım.

Matematiksel olarak işlem ifade edilirse;

Anahtar kelime $= (a_1, a_2, \dots \dots a_i)$ ve Düz metin $= (d_1, d_2, \dots \dots, d_i)$

olmak üzere, $f_i(d) = (d_i + a_i) mod\ 29$ olur.

$VIGENERE\ (26, 11, 7, 5, 16, 5, 20, 5)$

$AYNA$İ$LKD\ (0, 27, 16, 0, 11, 14, 13, 4)$

$f1 = (26 + 0) mod 29 = 26$

$f2 = (11 + 27) mod 29 = 9$

$f3 = (7 + 16) mod 29 = 23$

$f4 = (5 + 0) mod 29 = 5$

$f5 = (16 + 11) mod 29 = 27$

$f6 = (5 + 14) mod 29 = 19$

$f7 = (20 + 13) mod 29 = 4$

$f8 = (5 + 4) mod 29 = 9$

Çizelge 3.1 Vigenere Örnek Çözüm Tablosu

Düz m.	A	Y	N	A	İ	L	K	D
d_i	0	27	16	0	11	14	13	4
a_i	26	11	7	5	16	5	20	5
f_i	26	9	23	5	27	19	4	9
Şifre m.	V	H	T	E	Y	P	D	H

$VIGENERE\ (26, 11, 7, 5, 16, 5, 20, 5)$

$EFADOĞUD\ (5, 6, 0, 4, 17, 8, 24, 4)$

$f1 = (26 + 5) mod 29 = 2$

$f2 = (11 + 6) mod 29 = 17$

$f3 = (7 = 0) mod 29 = 7$

$f4 = (5 + 4) mod 29 = 9$

$f5 = (16 + 17) mod 29 = 4$

$f6 = (5 + 8) mod 29 = 13$

$f7 = (20 + 24) mod 29 = 15$

$f8 = (5 + 4) mod 29 = 9$

Çizelge 3.2 Vigenere Örnek Çözüm Tablosu

Düz m.	E	F	A	D	O	Ğ	U	D
d_i	5	6	0	4	17	8	24	4
a_i	26	11	7	5	16	5	20	5
f_i	2	17	7	9	4	13	15	9
Şifre m.	C	O	G	H	D	K	M	H

$VIGENERE\ (26, 11, 7, 5, 16, 5, 20, 5)$

$AYAPILDI (0, 27, 0, 19, 10, 14, 4, 10)$

$f1 = (26 + 0) mod 29 = 26$

$f2 = (11 + 27) mod 29 = 9$

$f3 = (7 + 0) mod 29 = 7$

$f4 = (5 + 19) m0d 29 = 24$

$f5 = (16 + 10) mod 29 = 26$

$f6 = (5 + 14) mod 29 = 19$

$f7 = (20 + 4) mod 29 = 24$

$f8 = (5 + 10) mod 29 = 15$

Çizelge 3.3 Vigenere Örnek Çözüm Tablosu

Düz m.	A	Y	A	P	I	L	D	I
d_i	0	27	0	19	10	14	4	10
a_i	26	11	7	5	16	5	20	5
f_i	26	9	7	24	26	19	24	15
Şifre m.	V	H	G	U	V	P	U	M

Oluşturulan şifreli metin: “VHTEYPDHCOGHDKMHVHGUVPUM”

Çizelge 3.4 Tükçe alfabe için oluşturulmuş Vigenére Tablosu

	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
	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

3.4.1.4 Hill şifreleme

1929 yılında Lester S. Hill tarafından bulunmuş Hill şifresi polialfabetik şifreleme sistemi olarak tanımlanıyor. Türkçe alfabe için $P = C = (Z_{29})^m$, m pozitif tam sayı olmak üzere tanımlansın. Temel fikir düz metindeki m sayısındaki alfabetik karakterlerin, m sayıda lineer kombinasiyonunu almaktır.

Örneğin, $m = 2$ olsun, düzmetin elemanı $x = (x_1 , x_2)$, şifremetin elemanı ise $y = (y_1 , y_2)$ şeklinde yazıla bilir. Burada y_1 ve y_2 'i, x_1 ve x_2 'nin lineer kombinasiyonu olacaktır (Tuncal, 2008).

m = 2 ise, K anahtarı için denklem (3. 1)'deki gibi şifreleme işlemi yapılacak.

$$(y_1, y_2) = (x_1 , x_2) \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \tag{3.1}$$

Genel olarak $x = (x_1, \dots x_m)$, $y = (y_1, \dots, y_m$ve K anahtarı için denklem (3.2)'deki gibi mxm'lik matris kullanılacak:

$$(y_1, y_2, \dots, y_m) = (x_1, x_2, \dots, x_m) \begin{bmatrix} k_{11} & \cdots & k_{1m} \\ \vdots & \ddots & \vdots \\ k_{m1} & \cdots & k_{mm} \end{bmatrix} \quad (3.2)$$

Başka bir şekilde şifreleme için seçilmiş K matrisi ve onun tersi olan K^{-1} matrisi kullanılarak şifreleme ve şifre çözme işlemini (3. 3) formülleri ile de göstere biliriz:

$$y = x_K \, mod 29 \quad şifreleme$$
$$x = y_{K-1} \, mod 29 \quad şifreçözme \quad (3.3)$$

Ö**rnek**:

$Anahtar \quad K = \begin{bmatrix} 4 & 7 \\ 9 & 6 \end{bmatrix} \; olsun.$

CEBİ düzmetinini şifreleyelim. $m = 2$ olduğundan düz metni ikişer $CE - B$İ olarak ayıralım. Turkçe alfabenin harf sırasına göre $CE\,(3,5)$ ve Bİ $(1,11)$ sayısal karşılığını elde ederiz.

$$(3,5) \begin{bmatrix} 4 & 7 \\ 9 & 6 \end{bmatrix} = (12 + 45, 21 + 30) = (28, 22) \rightarrow ZŞ$$

$$ve$$

$$(1,11) \begin{bmatrix} 4 & 7 \\ 9 & 6 \end{bmatrix} = (4 + 99, 7 + 66) = (16, 15) \rightarrow NM$$

CEBİ düzmetini şifrelenmiş halde $ZŞNM$ anlamsız metinine dönüştü. Şifre çözme işleminde ise K anahtar matrisinin tersi bulunarak ters işlem yapılıyor.

3.4.1.5 Playfair şifreleme

1854 yılında Charles Wheatstone'nin bulduğu bu şifreleme tekniğini sonralar Lyon Playfair geliştirdiği için Playfair şifreleme adlandırmışlar. Birinci Dünya Savaşı zamanında ingilizler bu tekniği kullanmışlar. Şifreleme işlemi İngiliz alfabesi ile oluşturulmuş 5X5'lik matris ile gerçekleştirilir.

Düzmetindeki her harf ikilisi (m_1, m_2) aşağıdakı kurallarla şifrelenmektedir (Ülkü, 2014):

1. m_1 ve m_2 harfleri aynı satırdaysa onların şifreli karşılığı, m_1 ve m_2 'nin bulundukları sütunun sağındakı ilk sütunlarda yerleşen sırasıyla c_1 ve c_2 harfleri oluyor.

2. m_1 ve m_2 harfleri aynı sütundaysa onların şifrelenmiş karşılıkları, m_1 ve m_2'nin bulundukları satırın altındakı ilk satırlarda yerleşen sırasıyla c_1 ve c_2 harfleri olur.

3. m_1 ve m_2 harfleri farklı satır ve sütunlardaysa onların şifrelenmiş karşılıkları, m_1 ve m_2'nin bulundukları yer matrisin köşesi kabul edilir ve m_1'in bulunduğu satırın diğer köşesindeki karakter c_1 ve m_2'nin bulunduğu satırdakı diger köşedeki karakter ise $c_2$olur.

4. Eger $m_1 = m_2$ olursa, düz metinde kullanılmamış bir harf m_1 ve m_2' nin arasına yerleştirilir ve aynı harflerin yan yana gelmesi önlenir.

5. Düz metin ikili harf gruplarına ayrıldığında tek harf kalarsa o zaman düz metinde kullanılmamış herhangi bir harf yanına eklenir.

3.4.1.6 Enigma

Enigma İkinci Dünya Savaşı zamanı askeri alanda sırları korumak ve gizli haberleşmek amacıyla geliştirilmiş şifreleme makinesidir. Orjinal mesaj mekanik olarak 26 harften oluşan klavye ile şifrelenmektedir. Klavyede tuşları aydınlatmak amacıyla her tuşun arkasına ışıklar yerleştirilmiştir. Şifreleme işlemini gerçekleştirmek için üç adet şifreleme rotorü ve bir adet yansıtıcı hareketsiz rotor kullanılır. Kenarlarında alfabe yerleştirilen hareketli çarkları ters düz çevire bilen bir mil kullanılır. Üstteki harfler küçük bir kapaktan gözükebilmektedir. Haraketli olan çarkların bir yüzüne 26 sabit ve ortak merkezli bağlantı yerleştirilirken diğer yüzüne 26 yaylı bağlantı yerleştirilir ve bu bağlantıların birbirleri ile izolasyonlu kablolar aracılığıyla düzensiz iletişimi sağlanır. Sabit çark üzerinde çiftler olarak birbirleri ile bağlanılmış sadece yaylı bağlantılar mevcuttur. Dört çark arasında kablolar aracılığıyla kurulan iletişim, Enigma şifreleme makinesinin en önemli adımıdır. Her bir kablo iki yuvaya takılmakla 26 harf için 26 kablo yuvası bulunmaktadır. Kablo sayısını p kabul edersek, $0 \leq p \leq 13$ olur ve kabloları yuvaya yerleştirmek için $\binom{26}{2p}$ kombinasyon sayıda seçim vardır (Ülkü, 2014).

3.4.2 Modern kriptografi teknikler

Bilgisayarın ortaya çıkması ile iletişimin elektronik ortamda yapılmasına başlanılması, güvenlik açısından yeni sorunların oluşmasına neden oldu ve modern şifreleme kavramı ortaya çıktı (Çimen, Akleylek, & Akyıldız, 2007). Modern kriptografi, bilgisayar ve haberleşme güvenliğinin temel taşı olarak görülmektedir. Modern şifreleme algoritmaları kısa ama karmaşık yapıdadırlar. Bahs ettiğimiz kriptografi tekniklerin tüm algoritmalarında bir anahtar kullanılır ve şifreli metin yalnızca kullanılan anahtarla uyuştuğunda çözülür.

Modern kriptografi tekniklerin iki alt ana tekniği vardır. Bunlar simetrik anahtarlı sistemler (gizli anahtarlı şifreleme) ve asimetrik anahtarlı sistemler (açık anahtarlı şifreleme) olarak ikiye ayrılmaktadır (Menezes, Oorschot, & Vanstone, 1996).

Simetrik (gizli anahtarlı) şifreleme teknikleri

Simetrik şifreleme tekniklerinde şifreleme işlemi için bir gizli anahtar kullanılmaktadır. Aynı gizli anahtar şifre çözme işlemi için de kullanılır. İletilecek metinin güvenliği için anahtarı yalnızca mesajı gönderen ve alan kişi bilmelidir. Bu tekniğe dayanan algoritmalar kullanırken şifrelenmiş metinle beraber anahtarı da güvenli şekilde alıcıya ulaştırmak gerekir. Bu yöntem çok kullanılan ve matematiksel açıdan daha az problemli bir yöntemdir. Simetrik şifreleme algoritmalarına örnek olarak XOR (özel veya), DES, 3DES, IDEA, RC2, RC5, Blowfish, FEAL, SAFER, Skipjack, Lucifer, ASEKAL-21 gibi algoritmalar gösterilebilir (Ülkü, 2014), (Kodaz & Botsalı, 2010).

Simetrik (gizli anahtarlı) şifreleme blok şifreleme ve akış şifreleme olmak üzere iki kategoriye ayrılmaktadır (Sakallı, 2006).

3.4.2.1 DES –veri şifreleme standartı

Gelişim süreci

1970'lerde IBM tarafından Lucifer adıyla geliştirilen DES algoritması NSA (Ulusal Güvenlik Ajansı) ve NIST (Uluslararası Standartlar Enstitüsü ve Teknolojisi) tarafından destek almıştır (Kodaz & Botsalı, 2010).

1970'li yıllara kadar askeri alanda haberleşme için özel kriptografi kodlamalar kullanılsa da, insanlar arasında yaygın olmamasından dolayı çok az insan bu bilim hakkında bilgilere sahip olmuştur.

1972'de Ulusal Standart ve Teknoloji Enstitüsü (NIST) olarak bilinen Ulusal Standartlar Bürosu (NBS), veri haberleşmesinin güvenliğini sağlamak üzere yeni bir kriptografi algoritması geliştirilmesi için özel standartlar belirtmiştir. Bu standartlara dayanan yeni bir kriptografi algoritma geliştirilmeli idi. 1973'te NBS belirttiği koşulları sağlayan kriptografi algoritmanın geliştirilmesi için bilim adamlarına aşağıdakı kriterlerden oluşan bir duyuruda bulunmuştur.

- Algoritma üst seviyede güvenlik saglamalıdır.
- Algoritma tamamen tanımlanmalı ve anlaşılması zor olmamalıdır.
- Algoritmanın güvenliği anahtara dayanmalıdır; güvenlik algoritmanın gizliliğine bağlı olmamalıdır.
- Algoritma tüm kullanıcılar tarafından eldeedilebilir olmalıdır.
- Algoritma çeşitli uygulamaların kullanımına uyarlanabilmelidir.
- Algoritma elektronik aygıtların ekonomik olarak gerçekleştirilmesini sağlamalıdır.
- Algoritma verimli olarak kullanılabilmelidir.
- Algoritma onaylanabilmeli ve test edilebilmelidir.
- Algoritma ihraç edilebilir olmalıdır.

Bu koşulları sağlayan hiçbir teklif olmadığı için 1974 yılında NBS yeni bir istek yayımlamıştır. Diğerinden farklı olarak bu isteğe olumlu teklifler gelmiş ve Lucifer algoritması basit işlemler uygulanarak donanımsal biçimde gerçekleştirilmiştir.
1975 yılında NBS algoritmanın lisansını ve detaylarını halkla paylaşarak onların algoritma hakkındakı fikirlerini öğrenmeye çalışmıştır. Ardından 1976 yılında bu algoritma federal bir veri şifreleme standardı olarak nitelendirilmiş ve devletin gizli tutulmayan haberleşmesinde uygun görülmüştür. Algoritma resmi olarak 1977 yılında "Veri Şifreleme Standardı" olarak halka sunulmuştur. 1980 yılında ise DES'in detaylarını içeren FIPS PUB 81 yayımlanmıştır. NSA DES'i donanımsal olarak geliştirdiği halde NBS çok detaylı anlatmıştır. Buda DES'in daha da geliştirilmesini sağlamıştır (Şen, 2006), (Yerlikaya, 2006).

DES algoritması

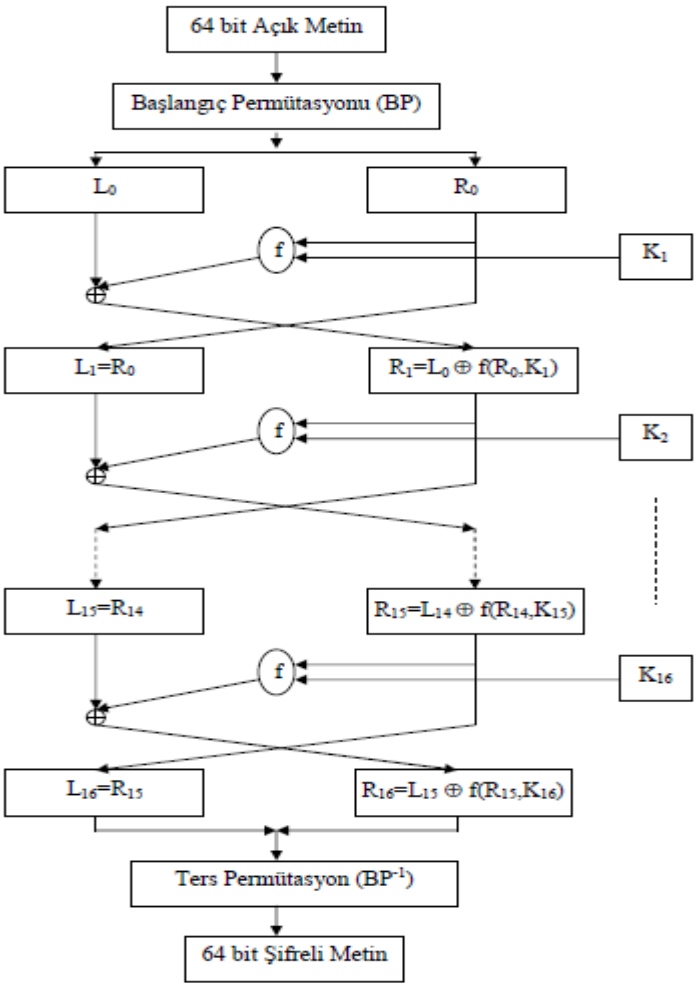
İlk modern kriptografi şifreleme tekniği sayılan DES algoritmasında blok şifreleme teknikleri kullanılmıştır. Hem şifreleme hem de şifre çözme işlemi yapan algoritma,

iki 64 bitlik giriş verisi alır. Bu verilerden biri düzmetin (şifremetin) diğeri şifreleme (şifre çözme) anahtarıdır. Veriler işleme sokulduktan sonra yeni 64 bitlik şifrelenmiş (deşifre edilmiş) veri üretilir. Seçilmiş 64 bitlik anahtarın sadece 56 biti algoritmada işleme girer, kalan 8 bit ise eşlenik kontrolü (odd parity check) için kullanıma alınır. Bu bitler her baytda en az önemi olan bitler olarak kabul edilir. DES algoritmasında şifreleme (şifre çözme) işlemleri için işleme sokulan 56 bitlik anahtardan her döngü için 16 adet yeni anahtar oluşturulur. Algoritmada işleme giren 56 bitlik anahtar güvenliğin daha iyi sağlanılması için gerektiği zaman değiştirilebilir. DES'de 4 çeşit önemli işlem kullanılır. Bunlar:

- Perimutasyon işlemleri
- Yerinekoyma işlemleri
- Mod işlemi
- (XOR) işlemidir.

Algoritmada yerinekoyma işlemlerini bir-birinden farklı bitlerin yer aldığı S-kutuları oluşturur. S-kutularının girişleri 6 bitden, çıkışları ise 4 bitden oluşmaktadır. Algoritmada döngü sayısı 16 olduğu için tekrarlanabilir biçimde olması donanımsal açıdan algoritmanın daha kolay gerçekleştirilmesini sağlamaktadır.

Agoritma 64 bitlik düzmetini aldıktan sonra başlangıç permütasyonundan işleme sokar. Başlangıç permütasiyonunda işlem yapıldıktan sonra 64 bitlik veri 32 bitlik sağ ve sol kısımlara ayrılır. Sonra f funksiyonun birleştirdiği, her biri için farklı anahtar kullanılan ardıcıl 16 döngüde işlem yapılır. Sonuncu döngü tamamlandığında sağ ve sol döngü birleştirilerek başlangıç işlemin tersi olan ters permütasyon işlemine sokulur ve 64 bitlik şifreli metin üretilir (Şen, 2006), (Yerlikaya, 2006).



Şekil 3.5 DES algoritmasının Blok diyagramı (Şen, 2006)

Başlangıç permütasyonu (IP)

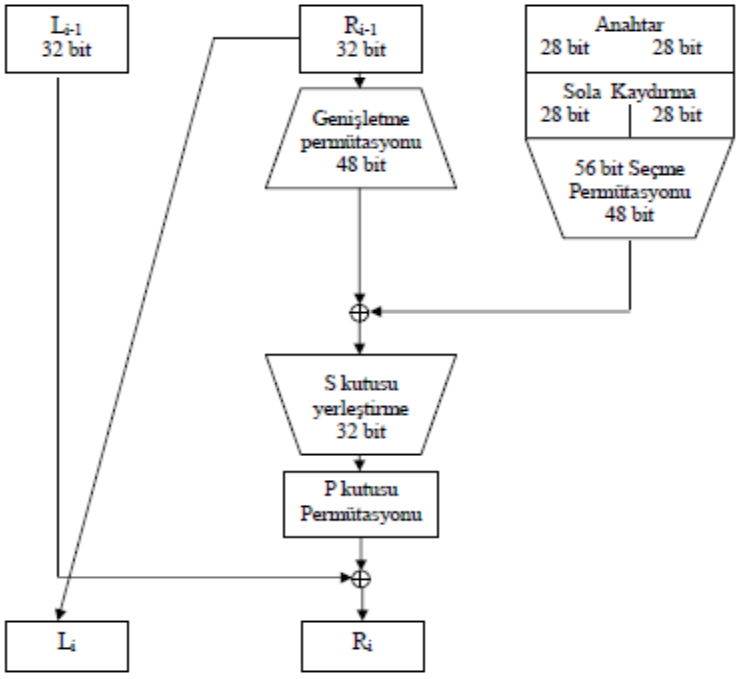
Algoritmanın ilk adımında 64 bitlik veri şifreleme işlemine girdikte ilk olarak başlangıç permütasyonuna tabi tutulur. Çizelge 3.5'te, veri başlangıç permütasyonuna tabi tutulduktan sonra bitlerin karıştırılmış hali gözükmektedir. Başlangıç permütasyonunda ilk bit 58. bit, ikinci bit 50. bit ve son olarak 64. bit ise 7. bit değerini almaktadır.

Çizelge 3.5 Başlangıç Permütasyonu (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Bir DES döngüsünde gerçekleştirilen işlemler

Şifrleme işleminde anahtar uzunluğu, ilk döngüden başlayarak her döngüde sola kaydırılır ve bit uzunluğu 56 bitden 48-e düşürülür. Sağ taraftakı 32 bit, genişletme permütasyonu aracılığıyla 48 bite dönüştürülür. Genişletme permütasyonundan çıkan 48 bit 48 bitlik anahtarla XOR işlemine sokulur. Elde edilen sonuç bir sonraki işlem için 8 tane S-kutularına tabi tutulur. Çıkışta üretilen 32 bitlik veri, P permütasyonunda işleme girir. Seri olarak uygulanan bu dört işlem birleşerek f fonksiyonunu oluşturur. f fonksuyonu çıkışta sol yandan gelen 32 bitlik veri ile XOR işlemine girir. İşlem sonucunda üretilen 32 bitlik veri sağ yarı kısımı, sağdakı eski 32 bitlik veri ise sol yarı kısımı oluşturur. Her döngü için bu işlemler tekrarlanarak 12816 döngülü DES algoritmasının işlemleri gerçekleştiriliyor (Şen, 2006), (Yerlikaya, 2006).

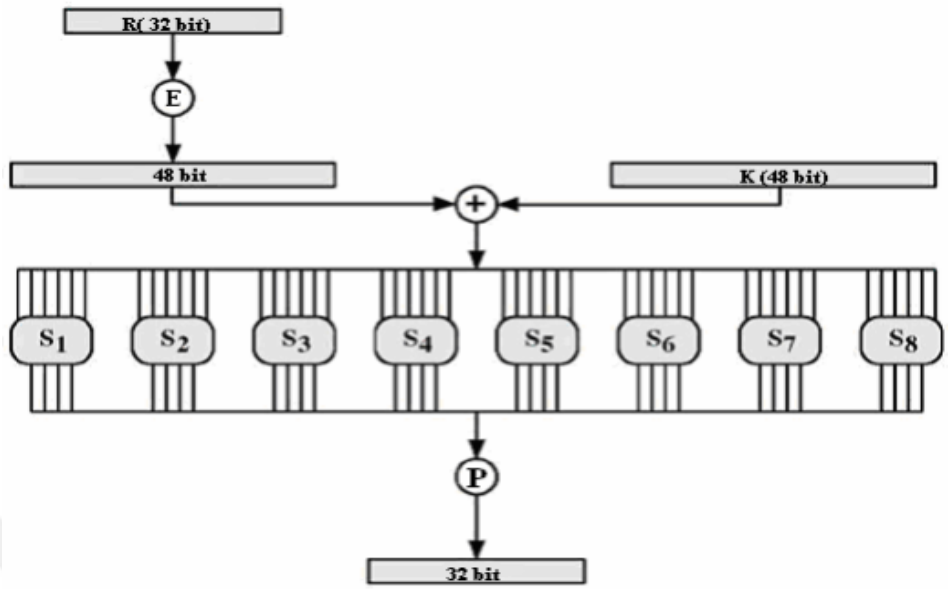


Şekil 3.6 Bir DES Döngüsünde Gerçekleştirilen İşlemler (Şen, 2006).

F fonksiyonu

F fonksiyonu algoritma içerisindeki her döngü için uygulanmaktadır. Alt anahtar oluşturma etapından gelen 48 bitlik alt anahtar ile F fonksiyonuna XOR işlemi uygulanır. Bu fonksiyon, E genişletme permütasyonu, S Kutusu ve P permütasyon

tablosundan oluşur. F fonksiyonunun tek bir döngü için genişletilmiş şeması aşağıdaki gibidir:



Şekil 3.7 F Fonksiyonu, F(R,K)'nın hesaplanması (Bayar, 2012)

Genişletme permütasyonu

Çizelge 3.6'da genişletme permütasyonu (E) yer almıştır. Çizelgede R (girişteki) bölmesinin 32 bit sayısının yerleri ve genişletilmiş hali gözükmektedir. Örneğin R bölmesinin 32. biti E genişletme çizelgesinde 1., 1. biti E çizelgesinde 2. ve 48. bit değeri yerine yazılır. Çizelgedeki altı çizili bitler genişletme permütasyonundakı 48 biti oluşturmak için sonradan ilave edilmiştir. Bu bitlerin yardımıyla R bloğundakı 32 bit, E genişletme permütasyonundakı 48 bite tamamlanır (Şen, 2006)

Çizelge 3.6 Genişletme Permütasyonu (E)

Genişletme permütasyonu(E)					
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

S-kutuları

Des algoritması 8 adet S kutusundan oluşmaktadır. S-kutuları 6 bitlik veri girişinin 4 bitlik çıkışını sağlayan ve 4 satır 16 sütundan oluşan bir tablodur. Genişletme permütasyonu işleminden sona oluşan 48 bitlik verinin 48 bitlik anahtarla XOR işlemine sokulmasından sonra 6 bitlik veriden oluşan 8 tane S kutusu bloklarına ayrılır. 6 bitden oluşan giriş verisinin ilk ve 4.bitinin oluşturduğu sayı satır numarasını, ortada kalan 4 bit ise sütun numarasını vermektedir. DES algoritması için en önemli işlem S-kutularının oluşturulmasıdır. Çünkü DES'in diğer işlemlerine göre S-kutularının yerleştirilmesi daha karmaşık, daha zor ve algoritmanın güvenliği için en önemli işlemdir. S-kutularının yerleştirilmesinden sonra yaranan 4 bitlik veriler birleşerek 32 bitlik blok meydana getirir ve P kutusu permütasyonunda bu bloktan faydalanılır (Şen, 2006), (Yerlikaya, 2006).

Çizelge 3.7 S – Kutusu 1

S – kutusu 1															
14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Çizelge 3.8 S – Kutusu 2

S - kutusu 2															
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
13	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

Çizelge 3.9 S – Kutusu 3

S - kutusu 3															
10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12

Çizelge 3.10 S – Kutusu 4

S - kutusu 4															
7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14

Çizelge 3.11 S – Kutusu 5

S – kutusu 5															
2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3

Çizelge 3.12 S – Kutusu 6

S – kutusu 6															
12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

Çizelge 3.13 S – Kutusu 7

S – kutusu 7															
4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12

Çizelge 3.14 S – Kutusu 8

S – kutusu 8															
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

P kutusu permütasyonu

P kutusu permütasyonu işlemi S-kutularından bir sonrakı aşamadır. S-kutusu işlemleri tamamlandıktan sonra ortaya çıkan veriye P permütasyon işlemleri uygulanır ve 32 bitlik yeni bir veri üretilir. Çizelge 3.15'te S – kutularından çıkan bitlerin yerleşmesi gösterilmiştir.

Çizelge 3.15 P Permütasyon tablosu (P)

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	31	10
19	13	30	6
22	11	4	25

Ters permütasyon

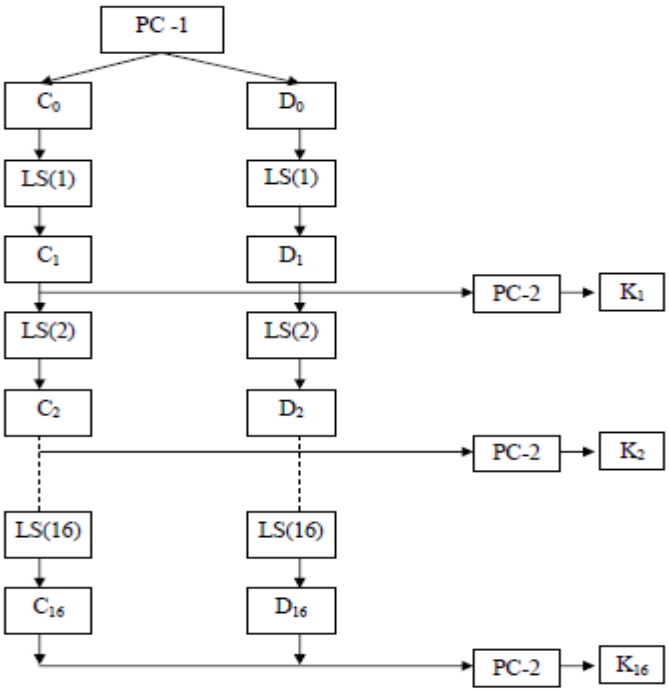
16 kez döngü işlemleri yapıldıktan sonra başlangıç permütasyonun tersi olan ters permütasyon işlemi uygulanır. Çizelge 3.16'da bitlerin yerleştirilme şekli gösterilmiştir.

Çizelge 3.16 Ters Permütasyon tablosu

Ters Permütasyon							
40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Anahtarların oluşturulması

DES algoritmasında şifreleme işlemi için başlangıç olarak 64 bitlik anahtar kullanılmaktadır. 64 bitlik anahtarın kullanımından sonra döngüler içerisinde kullanılacak alt anahtarlar üretilir. Yeni anahtar üretici tablo Şekil 3.8'de gösterilmiştir.



Şekil 3.8 Anahtar Üretim Tablosu (Şen, 2006)

PC – 1 Permütasyon tablosu

64 bitlik anahtarın PC1'e tabi tutulması sonucunda 56 bitlik anahtar üretilir. Üretilen 56 bitlik anahtar iki 28 bitlik bloklara ayrılır. Anahtarın 64 bitden 56 bite düşürülmesi zamanı 8, 16, 24, 32, 40, 48, 56, 64. bitler parite bitler olarak kabul edilir. 28 bitlik bloklar C ve D blokları olarak adlandırılır (Şen, 2006), (Denning, 1982).

Çizelge 3.17 Permutasyon seçimi tablosu (PC-1)

PC-1						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

PC – 2 Permütasyon tablosu

C ve D bölümleri sola döndürüldükten sonra birleştirilerek PC-2 permütasyon tablosunda işleme sokulur. Her bir döngü için uygulanan PC-2 permütasyonu 56 bitlik anahtarı, ana döngüde kullanılacak 48 bitlik alt anahtara düşürür. Çizelge 3.18'de PC-2 permütasyon tablosu verilmiştir.

Çizelge 3.18 PC – 2 Permütasyon Tablosu

PC - 2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Kaydırma Tablosu

Alt anahtarlar oluşturulduğu zaman 28 bitlik bloklar her döngüde 1 ya da 2 defa sola kaydırılıyor. Şifre çözme zamanı ise 2. döngüden başlayarak her döngüde 1 ve ya 2 defa sağa kaydırma işlemi yapılıyor. Her döngüde sola ve sağa kaydırma tablosu aşağıda verilmiştir.

Çizelge 3.19 Alt Anahtar üretim zamanı sola ve sağa kaydırma tablosu

Sola ve Sağa Kaydırma Tablosu																
Döngü sayısı	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Sola döndürülecek bit sayısı	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Sağa döndürülecek bit sayısı (şifreçözmede)	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Şifre çözme işlemi

DES Algoritmasında şifreleme işlemi için kullanılan algoritmanın aynısı şifre çözme işlemi için de kullanılmaktadır. Şifreleme ve şifre çözme algoritmalarındakı tek fark şifrelemedeki K_1 anahtarının şifre çözmede K_{16}, K_2'nin K_{15} vb.olmasıdır.

3.4.2.2 AES – ileri şifrleme standartı

AES'in gelişim süreci

Günümüz teknolojisi ile çok büyük sayıda anahtarların (2^{56} sayıda- olası tüm anahtarlar) denenmesi kısa bir zaman süresinde gerçekleştirilmektedir. Günümüzde bu nedenleri göz önüne alarak DES'in 3 defa art arda uygulanmasından oluşan 3DES algoritması kullanılır. Bu algoritma sayesinde anahtar uzunluğu 112 bite çıkarılsa da, algoritma çok yavaş bir algoritmadır (Dalkıç & Akın , 2005). Bu nedenlerden dolayı NIST (Ulusal Teknoloji ve Standartlar Enstitüsü) günümüz şartlarına cevap veren anahtar ve blok uzunluklu algoritma geliştirmek için 2 Ocak 1997 tarihinde 3 yıllık maraton yarışmanın duyrusunu yayınlamıştır. Bu yarışmaya 1998 yılında gerçekleştirilen AES1 konferansında dünya kiriptografi üyelerinin yaptığı 15 algoritma sunulmuştur. 1999 yılında, sunulan algoritmalar için, güvenlik, maliyet, algoritma ve uygulama karakteristikleri olan koşullara göre rastgelelik ve ansi c uyarlamalarının etkinlik testleri ve analizleri gerçekleştirilmiştir. Yapılan araştırma ve analizler sonucunda 5 algoritma (MARS, RC6, Rijndael, Serpent, Twofish) AES2 konfransında yarışma finalistleri olarak seçilmişler. Seçilen 5 algoritma yeni belirlenen koşullara göre testlerden geçirilmiş ve bunun sonucu 2000. yılın nisan ayında AES3 konferansında tartışılmıştır. NIST yapılan testlerin sonucu üzerinde incelemeler yaparak 2 Kasım 2000 tarihinde AES için Rijndael algoritmasının seçildiğini ilan etmiştir. 26 Aralık 2001 tarihinde temeli Rijndael algoritması olan FIBS PUB 197 numaralı İleri Sifreleme Standardı (Advanced Encryption Standard) yayınlanmıştır (Sakallı, 2006), (Güncan, 2002), (Rogaway & Coppersmith, 1994).

AES algoritması

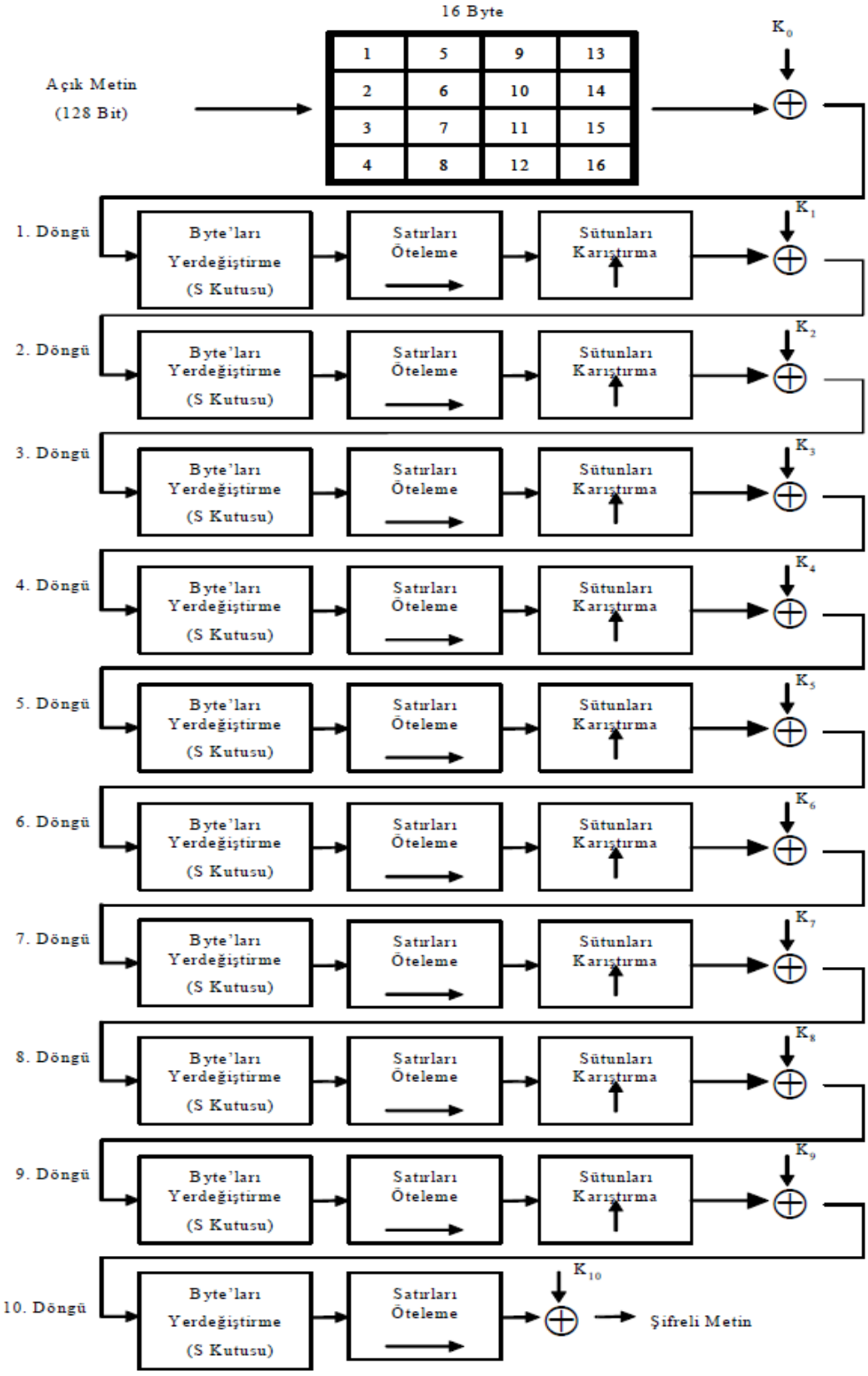
AES (Rijndael-Gelişmiş Şifreleme Standardı) algoritması 128 bit veri bloklarını 3 farklı (128, 192, 256) anahtarla şifreleyebilen bir simetrik şifreleme sistemidir. Şifreleme sisteminde kullanılan her farklı anahtar için farklı sayıda döngü ile şifreleme yapılmaktadır. 128 bit anahtar için 10, 192 bit anahtar için 12, 256 bit anahtar için 14 döngü ile şifreleme gerçekleştirilmektedir. AES algoritmasının her döngüsü 4 katmandan oluşmaktadır. Algoritmada ilk olarak 128 bitlik veri bloğu 4x4 baytlık matrise dönüştürülür. Bu işlem yapıldıktan sonra her döngü için aşağıda sıralanan 4 işlem uygulanır (Sakallı, 2006), (Yerlikaya, 2006), (Montgomery, 1985).

- Subbytes Fonksiyonu (baytların yerdeğiştirmesi)
- ShiftRows dönüşümü (satırların ötelenmesi)

- MixColumns Fonksiyonu (sütunların karıştırılması)
- AddRoundKey dönüşümü (döngüye anahtar ekleme dönüşümü)

Birinci işlemde 16 bayt değerinin her biri 8 bit girişli ve 8 bit çıkışlı S kutusuna sokulur. Bu işlemin ardından 4x4'lük matrisde satırlar ötelenir. Sütunların karıştırılması işleminde sütunlardan her hangi biri için o sütundakı değerler karıştırılır. Döngünün sonuncu işleminde döngü için kullanılan anahtar ile XOR işlemi yapılır (Sakallı, 2006), (Bayar, 2012), (Keliher, 2003).

Şekil 3. 9'da 128 bitlik veriyi 128 bitlik anahtar ile şifreleyen AES algoritmasının blok şeması verilmiştir.



Şekil 3.9 AES Algoritması Blok Diyagramı (128 bit anahtarlı) (Sakallı, 2006)

Subbytes fonksiyonu (baytların yerdeğiştirmesi)

Baytların yerdeğiştirilmesi doğrusal olmayan bir yerdeğişme işlemidir ve bir S kutusu kullanmakla bağımsız olarak durumun her bayt'ı üzerinde çalışması sağlanılır. Bu adımda durumun her bayt'ı değiştirme tablosuyla S kutusuna gönderilir ve yeni bir bayta dönüştürülür. S kutusu tersine çevrilebilir bir S kutusudur. Baytların yerdeğiştirilmesi zamanı 16 bayt değerinin her biri 8 bit girişli 8 bit çıkışlı S kutusuna sokulur. S kutusu değerleri, Galois alanında GF(2^8), " $x_8 + x_4 + x_3 + x + 1$" 8 bitlik polinomu için tersi alındıktan sonra lineer dönüşüm işlemi yapılarak oluşturulmuştur (Sakallı, 2006), (Yerlikaya, 2006).

		Y															
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**a**	**b**	**c**	**d**	**E**	**f**
x	**0**	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	Ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	B2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	F3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	E4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Şekil 3.10 AES S Kutusu (Hexadecimal notasyonda xy byte için) (Yerlikaya, 2006).

ShiftRows dönüşümü (satırların ötelenmesi)

Bu dönüşümde ilk satırda öteleme yapılmazken 2. satırda bir, 3. satırda iki, 4. satırda üç kere öteleme yapılmaktadır. Şekil 3.11'de ShiftRows dönüşümü gösterilmektedir.

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

a.

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

b.

Şekil 3.11 ShiftRows Örneği a. Öteleme Öncesi, b. Öteleme Sonrası (Yerlikaya, 2006)

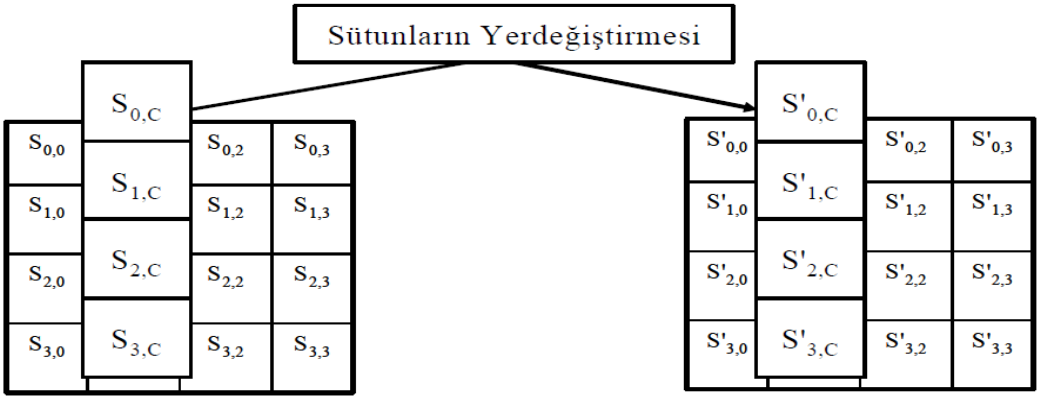
MixColumns fonksiyonu (Sütunların karıştırılması)

Sütunların karıştırılması katmanında sütun sütun işlem yapılır. Bu katmanda sütunlar GF(2^8)'de kabul edilir ve sabit bir a(x) polinomu ile mod (x^4 + 1)'e göre çarpılarak sonuç elde edilir.

$$a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\} \quad (3.4)$$

32 bit sütun değerini gösteren S(x) ve yerdeğişmeye uğrayan sütunların 32 bit değerini gösteren S′(x) ifadeleri arasında aşağıdaki eşitlik mevcutdur:

$$S'(x) = a(x) \otimes S(x) \quad (3.5)$$



Şekil 3.12 Sütunların Karıştırılması Dönüşümü (Ülkü, 2014).

AddRoundKey dönüşümü (döngüye anahtar ekleme dönüşümü)

Bu döngüde ana anahtardan işlem yapılan döngü için üretilen alt anahtarla durumun XOR işlemi gerçekleştirilir.

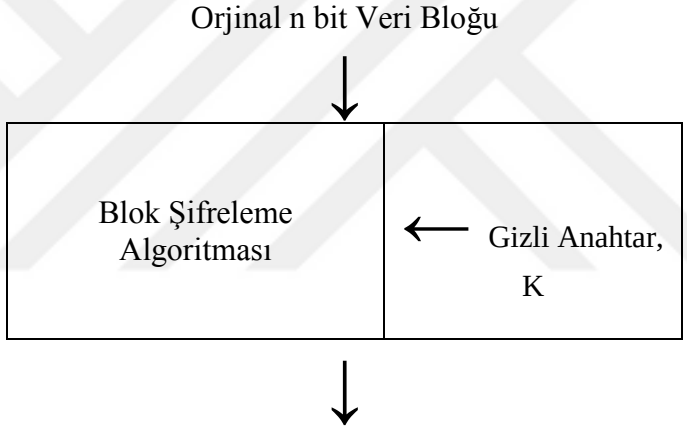
Deşifreleme işlemi

AES algoritması için şifre çözme işlemi, şifreleme işlemindeki fonksiyonların tersleri alınarak yapılmaktadır (Sakallı, 2006), (Güncan, 2002), (Daemen & Rijmen, 2000).

3.4.3 Blok şifreleme

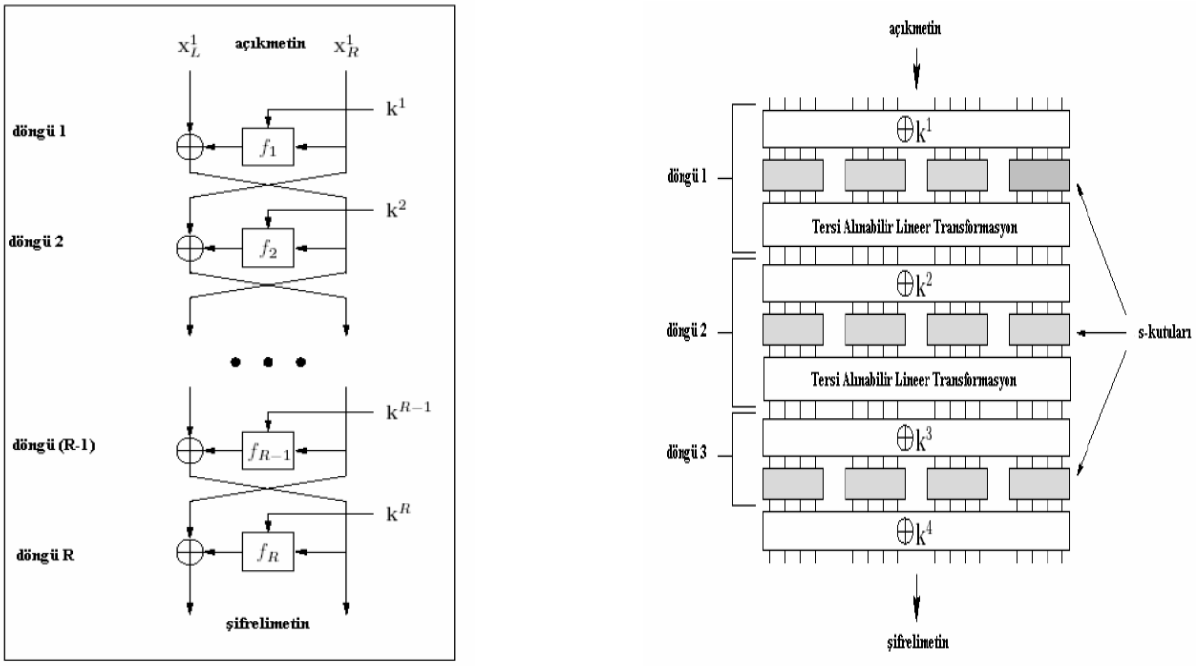
Blok şifreleme algoritmaları ikili kodda (0 ve 1'ler) şifreleme yapan, yeni ham veriyi bitler bloğu şeklinde alan algoritmalardır. Bu algoritmalar sabit uzunluklu ham bit bloklarını, seçilmiş anahtar yardımıyla aynı uzunluklu şifreli bit bloklarına dönüştürmektedirler. Genel olarak bit bloklarının uzunluğu 32, 64 ve 128 bit şeklinde seçilmiştir. Aşağıdakı resimde blok şifreleme algoritmalarının genel yapısı verilmiştir (Sakallı, 2006).

Çizelge 3.20 Blok Şifreleme Algoritmalarının Genel Yapısı



Simetrik şifreleme algoritmalarından olan blok şifreleme algoritmaları, Şannonun ortaya koyduğu karıştırma ve yayılma tekniklerine dayanmaktadır. Karıştırma işlemi orjinal metinle şifreli metin arasındakı ilişkiyi yok etmeye çalışırken, yayılma ise açık metin üzerinde yapılan işlemlerin şifreli metin üzerinde sezilmemesini sağlar. Karıştırma yerdeğiştirme, yayılma ise doğrusal dönüşüm (lineer transformasion) işlemleri ile gerçekleştirilir. Bu tür şifreleme sistemlerinde yerdeğiştirme işlemi S kutuları ile, yayılma işlemi ise bayt ve bit aracılığıyla gerçekleştirilen lineer transformasionla sağlanılır. S kutuları doğrusal olmayan kutulardır ve bu tür şifreleme algoritmalarında doğrusallığın olmaması en önemli şartlardan biridir (Sakallı, 2006).

Blok şifreleme algoritmalarının iki esas ana mimarisi vardır. Bunlar *Fiestel ağları* ve *yerdeğiştirme-Permütasyon* (SPN-Substitutio-Permutation Networks) ağlarıdır (Tuncal, 2008). Bu mimariler bir kaç şifreleme işleminin birleşmesinden oluşur. Tekrarlanan şifreler aynı şifreleme adımının bir kaç defa yapılmasını içerir ve bu şifreler ürün şifrelerdir. Blok şirelemenin feistel mimarisini kullanan algoritmalara DES'i, SPN mimarisini kullanan algoritmalara ise AES'i örnek olarak göstermek olar. Feistel mimarisi bir döngüde verinin yarsını işlerken, SPN mimarisi bir döngüde tam veriyi işlemeği sağlar. Bu işlemde Feistel mimarisi SPN'e göre dezavantajlı olsa da uygulama kısmında daha iyi avantaj sağlayan tarafları vardır. Feistel mimarisi kullanılan bir şifreli metinin deşifre olunması için anahtarların ters çevrilmesi yeterli oluyor. Aşağıakı şekilde Feistel ve SPN mimarileri srasıyla verilmiştir (Sakallı, 2006).



a **b**

Şekil 3.13 a Feistel Mimarisi, **b** SPN Mimarisi

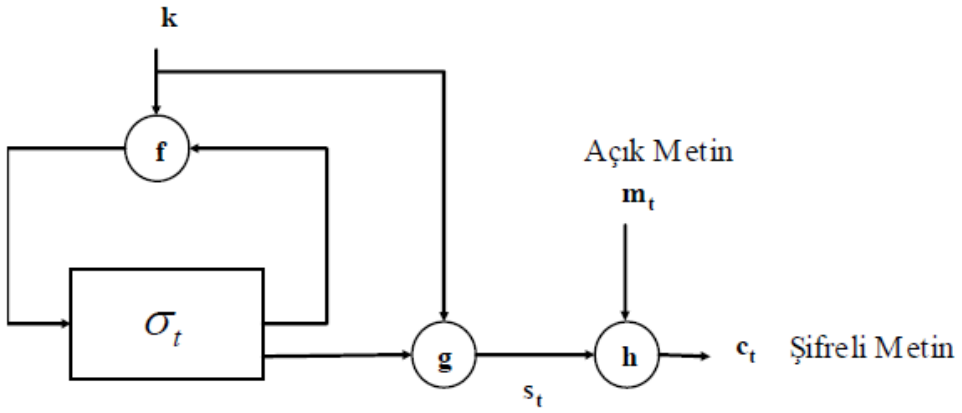
3.4.4 Akış şifreleme

Akış şifreleme sistemleri açık metin bloğunu bitler şeklinde alır, şifrelenmiş veriyi de bitler şeklinde üretmektedir. Bu tip şifreleme sistemlerinde şifreli metin, orjinal metin ($M = m1, m2, \ldots, ms$) ve anahtar bitlerinin ($K = k1, k2, \ldots, ks$) (3.6) ifadesinde gösterildiği gibi XOR ($mod\ 2'ye$ göre) işlemine sokulması sonucu ile elde edilir (Sakallı, 2006).

$$ci = mi \oplus ki, \quad i = 1, \ldots s \tag{3.6}$$

Şifre çözme işlemi de aynı teknikle gerçekleştirilmektedir. Akış şifreleme sistemlerinin ne kadar güvenilir olması doğrudan anahtar üreticisine bağlıdır. Bu sebepten anahtar dizisi olarak tamamen rastgele ve bir kereliğine kullanılan veri seçilmelidir. Bu tip şifreleme algoritmalarında en önemli şartlardan biride açık metin uzunluğunun anahtar uzunluğuna eşit olmasıdır. Üretilen anahtarın kendini tekrarlamaması ve sonrakı üretilen anahtarların önceki anahtar bitleri aracılığıyla elde edilmemesi, anahtar üreticileri için sağlanması gereken en önemli şartdır (Şen, 2006), (Sönmez, 2002).

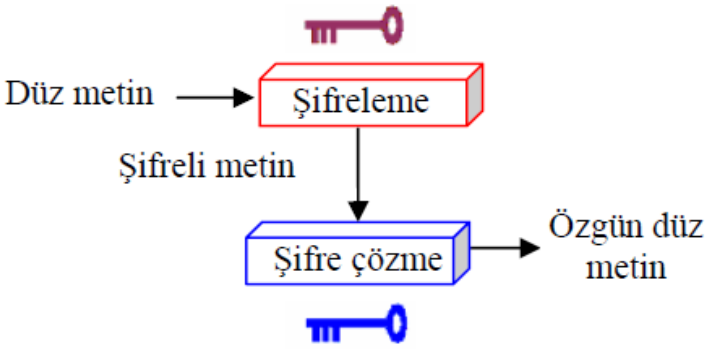
Akış şifreleme algoritmaları zamanla değişen bir fonksiyon kullanarak tek karakter üzrinde işlemi gerçekleştirerler. Bu özelliği sağlayan algoritmalara örnek olarak RC4 ve SEAL algoritmalarını göstermek olar (Başar, 2004). Şekil 3.14'te senkron bir akış şifreleme sisteminin genel yapısı gösterilmiştir. Açık metin (m_t), ve anahtar dizisinin (s_t) h fonksiyonuna girişinin sonucunda şifreli metin (c_t) üretilmektedir (Sakallı, 2006).



Şekil 3.14 Senkron Bir Akış Şifreleme Algoritmasının Genel Yapısı

3.4.5 Asimetrik (açık anahtarlı) şifreleme teknikleri

Asimetrik şifreleme algoritmalarında iki farklı anahtar kullanılarak şifreleme ve şifre çözme işlemi gerçekleştirilir. Şifreleme işlemi için kullanılan anahtar herkese açık olurken, şifre çözme işlemi için kullanılan anahtar sadece şifre çözme işlemini gerçekleştiren kişi tarafından bilinir. Asimetrik şifreleme algoritmalarına RSA, DSA, Diffie-Hellman, ElGamal algoritmaları örnek olarak gösterilebilir (Yılmaz, 2010).



Şekil 3.15 Asimetrik Şifreleme Algoritmalarının Genel Yapısı (Yerlikaya, 2006)

3.4.5.1 RSA algoritması

RSA (RIVEST-SHAMIR-ADLEMAN) asimetrik şifreleme sistemi 1977 yılında R.Rivest, A.Shamir ve L.Adleman tarafından tasarlanmış ve asimetrik şifreleme algoritmalarına uygun olarak geliştirilmiştir. Bu algoritma sayısal imza ve açık anahtarlı şifrelme sistemi işlemlerini gerçekleştirmek için güvenli şekilde kullanılır (Yerlikaya, 2006).

Public-key şifreleme yöntemlerinin temel uygulamalarından olan RSA algoritmasının en önemli tarafı, mesajlaşmak isteyen iki kişinin kendi aralarında önceden görüşme yapmadan iletişim kurmaları için yüksek güvenli bir ortam sağlamasıdır. Matematiksel açıdan çalışma yöntemleri çok basit gözüken RSA algoritması, biri herkese açık diğeri gizli olan iki anahtarla şifreleme yapmaktadır. Kendi mesajını karşı tarafa iletmek isteyen kişi mesajı şifreler, herkese açık anahtarı yayımlar ve mesajı gönderir. Ancak mesajı gizli anahtara sahib olan biri çöze bilir. Gizli anahtarsa sahibinde bulunduğu için, herkes bu anahtarı bilmeden mesajını yüksek güvenli bir şifreyle gizleyebilir. Bu yöntemle daha önce birbirini hiç görmeyen, hiç tanımayan kişiler, mesajlarına hiçkimsenin ulaşamayacağı, yüksek güvenli bir ortamda mesajlaşa bilirler (Gamal, 1988).

RSA algoritmasının yapısı

RSA algoritmasının matematiksel açıdan genel yapısı aşağıda gösterilmiştir. Şifreleme işleminin ilk aşamasında orjinal metin $[0, n-1]$ arasındakı pozitiv tam sayı blokları haline dönüştürülür (Gamal, 1988).

Şifreleme işlemi matematiksel yöntemlerle aşağıdakı ardıcıllıkla devam etdirilir:

1. İlk önce birbirinden farklı p ve q adlı çok büyük iki asal sayı seçilir ve bunların çarpımına N adı verilir.

$$N = p.q$$

2. Sonrakı adımda $(p-1).(q-1)$ çarpımı hesaplanır ve bu sayı Z olarak gösterilir.

$$Z = (p-1).(q-1)$$

3. Bu adımda ise Z ile ortak böleni olmayan, 1-den büyük ve N-den küçük bir E sayısı seçilir.Açık anahtar (E,n) olarak belirlenir.

$$D = E - 1 mod\ Z$$

D sayısı bulunur ve gizli anahtar (D,n) olarak belirlenir.

4. Gönderilecek mesajı M olarak isimlendirsek, M dâhilinde ki harflerin alfabe sıralamasındakı değerleri N-den küçük olmalıdır. Ardından M mesajı n bitlik kısımlara ayrılır.

$$M = M(1) + M(2) + \cdots + M(n)$$

Ayrilan her bir kısım için $M(i)^E$ mod N işlemi uygulanır ve C(i) şifreli metini elde edilir.

$$C(i) = M(i)E mod\ N$$

Böylece şifreleme işlemi tamamlanmış olur ve şifreli mesaj güvenli şekilde karşı tarafa iletilir.

Şifreleme sisteminin güvenliği, N ve E değerlerini herkes bilse de $M(i)^E$ mod N değerini ele keçiren birinin p ve q değerlerini bilmeden M'yi (orjinal mesaja) bulamayacağına dayanır (Çimen, Akleylek, & Akyıldız, 2007), (Yerlikaya, 2006).

Şifreçözme

Şifreçözme işlemi aşağıdakı adımlarla gerçekleştirilir:

1. Şifreli mesajı alan kişi, yukarıda bahs etdiğimiz N, E sayılarını ve gizli olarakda p, q asal sayılarını, ayrıca daha önce hesapladığımız Z sayısını bilmektedir. İlk olarak alıcı $E.D = 1 mod Z$'den şifreçözmede kullanacağı D anahtarını bulur.

$$D = E - 1 mod\ Z$$

2. Daha sonra şifrelenmiş mesajın, yeni C(i) değerinin D dereceden kuvvetini bulur ve modN'e göre hesaplar.

$$C(i)D\ mod\ N\ =>\ (M(i)E)Dmod\ N\ =>\ M(i)E.Dmod\ N\ =\ M$$

Böylece alıcı orjinal mesaja ulaşmış olur (Çimen, Akleylek, & Akyıldız, 2007).

3.4.5.2 DSA algoritması

DSA (Digital Signature Standard) algoritması NIST (National Institute of Standards and Technology) tarafından sayısal imza standardı olarak yayınlanmıştır."Discrete Logarithm" problemine dayanan DSA algoritması Diffie-Hellman ve RSA algoritmaları gibi asimetrik (açık anahtarlı) kriptografi sistemdir. RSA'dan farklı olarak DSA algoritması şifreleme amacı ile kullanılmamakta ve sadece imzalama için kullanılmaktadır.

DSA algoritması matematiksel olarak aşağıdakı şekilde çalışmaktadır:

p, bit uzunluğu 512 ve 1024 arasında olan bir asal sayı

q, bit uzunluğu 160 olan ve $p-1$ sayısını bölen bir asal sayı

g, $(p-1)$'den küçük herhangi bir h sayısı için $g = h(p-1)q\ (mod\ p)$ eşliğini sağlayan ve 1'den farklı herhangi bir sayı olmak üzere p, q ve g sayıları uygun yöntemler kullanılarak bulunur (Yılmaz, 2010).

3.4.5.3 Diffie – Helman açık anahtar dağıtımı

Kriptografinin temel yapılarından biri, mesaj alıcı ve göndericinin bir araya gelerek anahtar belirlemesi zorunluğu olmuştur. Diffie ve Helman bu zorunluğu aradan kaldırarak, alıcı ve gönderici bir araya gelme zorunluğu olmadan, birbirleri ile güvenli ortamda anahtar paylaşabilecekleri bir sistem geliştirmişlerdir. Oluşturulan bu sistem kriptografide açık anahtarlı şifreleme sistemlerinin temelini koymuştur. Açık anahtarlı şifreleme sistemlerinde anahtar iki kısımdan oluşmaktadır. Bu kısımlardan biri herkes tarafından bilindiği halde, diğer kısımı yalnızca alıcı tarafında bilinmektedir. Anahtar paylaşımına ihtiyaç duyulmadığından bu tür kriptosistemler açık anahtarlı kriptosistemler olarak bilinmektedir.

Diffie-Helman anahtar dağıtımı sisteminin matematiksel yapısı aşağıdaki gibidir:

Aralarında gizli anahtar belirtmek isteyen iki kişi düşünelim. Bunlardan biri Ali diğeri Buşra olsun. Ali ve Buşra herkes tarafından bilinen p asal sayısı ve bir g sayısı belirtsinler. Anahtar alışverişi sırasıyla aşağıdakı gibi devam etmektedir.

Ali $0 < a < p - 1$ şartını sağlayan bir a tam sayısı (gizli) seçer, g^a (mod p) işleminin sonucunu bulur ve bu sayıyı (u) Buşraya gönderir.

$$u = g^a (mod\ p) \tag{3.7}$$

Buşra 0 < b < p-1 şartını sağlayan bir b tam sayısı (gizli) seçer, g^b(mod p) işleminin sonucunu bulur ve bu sayıyı (v) Aliye gönderir.

$$v = g^b (mod\ p) \tag{3.8}$$

Ali v^a(mod p) ifadesini hesaplar ve gereken gizli anahtarı bulur.

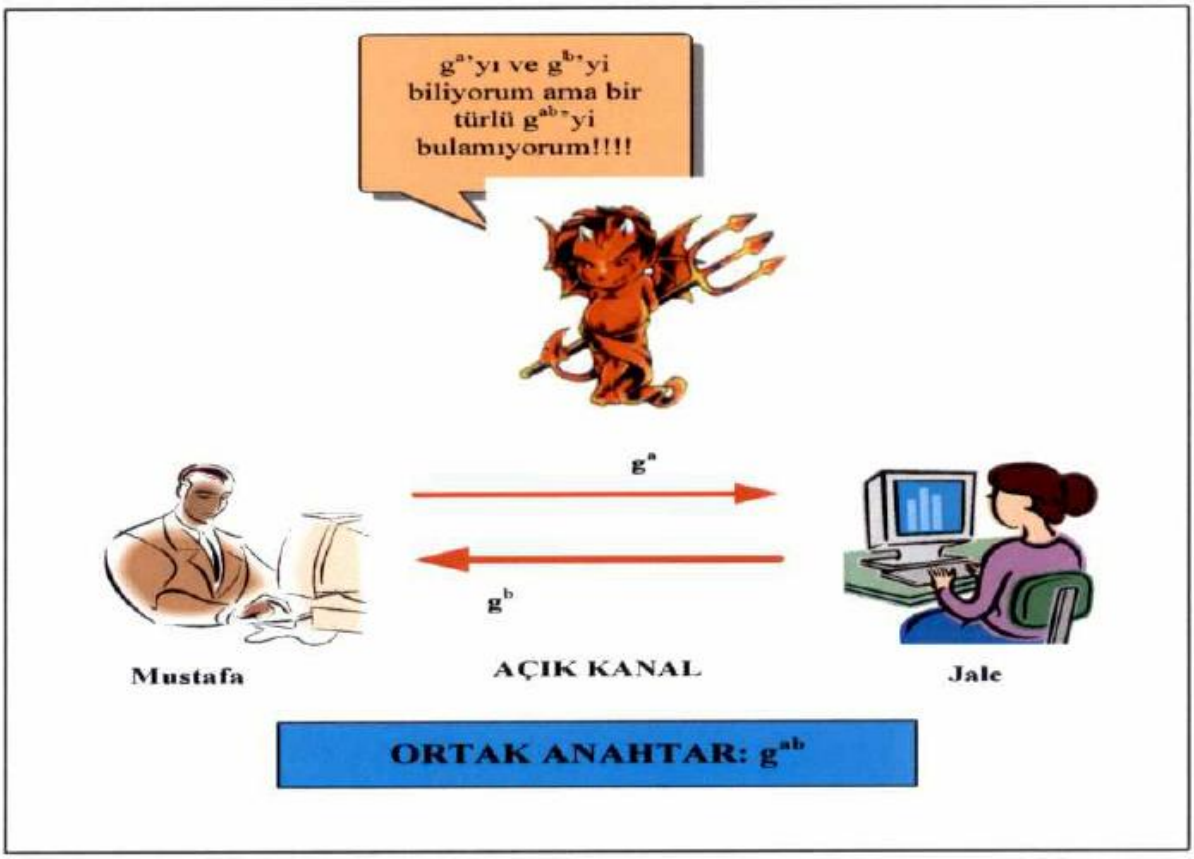
$$v^a \equiv (g^b)^a \equiv g^{b.a}\ (mod\ p) \tag{3.9}$$

Buşra u^b(mod p) ifadesini hesaplar ve gereken gizli anahtarı bulur.

$$u^b \equiv (g^a)^b \equiv g^{a.b}\ (mod\ p) \tag{3.10}$$

Sonuçda ortak gizli anahtar $k = g^{a.b} (mod\ p)$ sayısı olmuş olur.

Dolayısıyla, $u^b\ (mod\ p) = k = v^a\ (mod\ p)$ eşitliği DPL logaritma problemine dayanır ki, bununda çözümünün çok zor olduğu bilinmektedir (Tefon, 2013).



Şekil 3.16 Diffie-Hellman Anahtar Paylaşım Protokolü (Tefon, 2013)

3.5 Kriptanaliz Teknikleri Ve Saldırı Çeşitleri

Kriptanaliz, açık metini veya anahtarı elde etmek için kullanılan teknikler bütünüdür. Yapılan kriptanaliz işlemine saldırı denilmektedir. Bir saldırıyı yapacak olan kişi orjinal mesajın içeriğiyle ilgili hiç bir bilgiye sahip değildir. Mesajın şifrelendiği algoritmada ise kullanılan gizli anahtar dışında, tüm detaylarla ilgili bilgiye sahip olmaktadır. Bir kriptanaliz yapan kişinin maksatı, her hangi bir şifre çözme metodu aracılığıyla şifreli mesajın şifrelendiği yöntemleri tespit etmek ve gizli anahtarı elde ederek orjinal mesaja ulaşmaktır. Kriptanaliz yöntemlerinin effektiv ve güclü bir yöntem olması için, kriptolama ile ilgili gereken tüm ön bilgilere sahip olmak gerekmektedir. İyi bir kriptanalist olmak isteyen kişi, önce iyi bir kriptograf olmak zorundadır (Bağcıoğlu, 2007), (Akay, 2014), (Singh, 2013). En çok kullanılan saldırı çeşitleri aşağıda verilmektedir:

3.5.1 Sadece şifeli metin saldırısı (Ciphertext Only)

Bu saldırı en güçlü kriptanaliz atağı olarak bilinmektedir. Şifreli metnin kriptanalizini yapan kişi orjinal mesajın içeriği ile ilgili hiçbir bilgiye sahip olamdan, sadece şifreli metin üzerinden düzmetine ulaşmaya çalışmaktadır. Şifrelme

yöntemleri ve düzmetinle ilgili yapılan tahminlerle yola çıkarak saldırı gerçekleştirilir (Ülkü, 2014), (Şen, 2006).

3.5.2 Bilinen açık metin saldırısı (Known Plaintext)

Bir şifreli metinin kriptanalizini yapmak isteyen kişi orjinal metinin bazı kısımlarına ve aynı orjinal metinin şifrelenmiş haline sahip olmalıdır. Gönderilen mesajların bir kısımı tahmin edilir, ya da elde edilmiş açık metin kısımları toplanarak saldırı yapılabilir. Kullanılan bu bilgiler aracılığıyla şifreli metinin blokları çözülebilir. Bu saldırı türünde en sık kullanılan saldırı blok şifreleme sistemlerine karşı kullanılan lineer kriptoanaliz saldırısıdır. Saldırıda şifreleme anahtarını bulmak amaçlanıyor (Ülkü, 2014), (Şen, 2006), (Soyalıç, 2005), (Yılmaz, 2010).

3.5.3 Seçilmiş açık metin saldırısı (Chosen Plaintext)

Analiz yapan kişi bilinmeyen anahtar aracılığıyla gereken şifrelenmiş düz metin kısımlarına ulaşa bilmektedir. Saldırgan, şifreleme için kullanılan algoritmanın güvenli olarak yerleştirildiği yöntemleri elde edebilir. Saldırgan aktif mesajlaşma zamanı sistemde yer alabilmektedir. Bu saldırı çeşiti, açık metinin anahtar hakkında daha çok bilgiye sahip olmağı sağlayan bloklarını seçme imkânı olduğu için bilinen düz metin saldırısından daha güçlü saldırı tekniğidir. Bazı kriptografi algoritmalar, özellikle RSA algoritması, seçilmiş açık metin saldırılarına karşı açık olmaktadır. Bu tip şifreleme sistemleri kullanıldığı zaman, saldırganın düzmetinin şifrelenmiş haline ulaşmaması için, uygulamanın tasarım yöntemlerine ciddi dikkat edilmelidir (Şen, 2006), (Soyalıç, 2005).

3.5.4 Seçilmiş şifreli metin saldırısı (Chosen Ciphertext)

Saldırgan, bu saldırı tekniğini kullanarak seçdiği şifreli metin kısmı ile uyuşan düz metin kısmına ulaşmaya çalışar. Bu saldırı çeşiti şifre çözme tekniğine ya da algoritmasına ulaşılarak yapılan saldırı tekniğidir. Saldırgan bir şifreli metin kısımı seçerek, ulaşdığı şifre çözme için kullanılan algoritma aracılığıyla seçdiği şifreli metin kısmını deşifre eder ve düz metine ulaşmaya çalışar (Şen, 2006), (Yılmaz, 2010).

3.5.5 Seçilmiş açık veya şifreli metin saldırısı (Adaptive chosen plaintext or ciphertext)

Bu saldırı tipi seçilen düzmetin ve seçilen şifreli metin saldırıları ile yakın yöntemlere sahiptir. Bu saldırı çeşitleri arasındakı fark seçilen açık veya şifreli metin

saldırısında metinler rastgele değil, önceki deşifreleme işlemlerindeki bilgilerden yararlanarak seçilmesidir. Bu saldırı tekniği seçilen açık metin ve seçilen şifreli metin saldırısının daha da güçlendirilmiş saldırı tipidir (Ülkü, 2014), (Sakallı, 2006), (Şen, 2006).

3.5.6 İlişkili anahtar atağı

Saldırgan bu saldırı tekniği ile farklı anahtarlar kullanarak şifrelenmiş metinler setinin sonuçlarını gözaltında tuta bilme ve idare edebilme yeteneğine sahiptir (Sakallı, 2006).

3.5.7 Kaba güç (Brute force) saldırısı

Bu saldırı çeşiti, şifreli metinin okunabilir hale getirilene kadar, tahmin edilen olası tüm anahtarları tek tek deneyerek atak yapan saldırı tekniğidir. Bu tür anahtar bulma yönteminin karşısını almak için gereken metotlardan biri anahtarların yeterince büyük seçilmesidir (Soyalıç, 2005).

3.5.8 Ortadaki adam saldırısı (Man-in-the-Middle)

Bu saldırı tekniği genelde anahtar değişimi protokölleri ile ilgili bir saldırı çeşitidir. Bu saldırı tekniği aracılığıyla saldırgan, kendini güvenli iletişim için kendi aralarında anahtar değişikliği yaparak (örneğin Diffie-Hellman anahtar değişmi) iletişimde olan iki kişi arasına sokar ve tarafların her ikisine sanki diğer tarafmış gibi davranır. Dolayısıyla, her iki kişinin özel anahtarına ulaşarak onlarla anahtar değişimi yapar. Her iki kişi kendi özel anahtarları aracılığıyla güvenli iletişim yapacaklarını düşünmelerine rağmen saldırgan her iki özel anahtara ulaşmaktadır. Anahtarları elde eden saldırgan, iletişim zamanı uygun anahtar ile uygun şifreli metni deşifre edecek, sonra ise diğer anahtarla şifreleme yaparak karşı tarafa iletecektir. İletişimde olan kişiler de saldırgandan habersiz olarak kendi aralarında güvenli iletişim yaptıklarını sanacaklardır. Aslında ise saldırgan iletişim zamanı her iki kişinin bir birine aktardığı herşeye ulaşmaktadır (Yılmaz, 2010).

3.6 Tek Kullanımlık Anahtar (One Time Pad)

Tek kullanımlık anahtar yöntemi (one time pad) ilk defa Gilbert Vernam tarafından geliştirilmiş bir şifreleme sistemidir. Güvenlik açısından bu yöntemin en önemli tarafı tek kullanımlık anahtarın açık metin uzunluğu boyutunda ve tamamen rastgele bitlerden oluşmuş olmasıdır. Açık metni ele geçiren ve kırma işlemini

gerçekleştirmek isteyen saldırgan hiçbir bilgiye ulaşmamalıdır (OTP, 2014). Kullanılacak anahtar açık metin boyutunda ve yalnız bir defa kullanmak için üretilmelidir. Anahtar tamamen rastgele olduğunda ve doğru kullanıldığı takdirde kırma işleminin imkânsız olduğundan dolayı mükemmel güvenlik (perfect secrecy) sağlar (Frank, 1882). Tek kullanımlık anahtarlar yüksek güvenlik sağlasalar da, pratiklik açısından yaygın olarak kullanılmamaktadır. İlk kez 1882 yılında Frank Miller tarafından tanımlanan tek kullanımlık anahtar (one time pad) 1917 yılında yeniden geliştirildi ve patentini aldı (Frank, 1882).

3.6.1 Tek Kullanımlık anahtar (one time pad) nedir?

Bilinen ilk güvenli şifreleme yöntemlerinden biri tek kullanımlık anahtarla yapılan şifreleme yöntemidir. XOR işlemleri kullanılarak basit adımlarla şifreleme ve deşifre etme işlemi yapılan algoritmaların genel matematiksel yapısı aşağıdaki formüllerle gösterile bilir (Bağcıoğlu, 2007).

$$p = Ş(a, m) = a \oplus m$$

$$m = D(a, p) = a \oplus p$$

Formüllerde yer alan a-anahtar, m-açık metin, p-şifreli metin, Ş-şifreleme algoritması, D-şifre çözme algoritması anlamına gelmektedir.

Tek kullanımlık anahtarla şifreleme yöntemi yalnızca XOR işlemi kullanılarak yapıldığından dolayı çok hızlı olsa da, kullanımı oldukça zordur. Buna sebep olarak anahtarın en az açık metin uzunluğu kadar olması zorunluluğunu göstere biliriz. Eğer internet üzerinden mesajlaşan iki kişi anahtarı bir-birlerine güvenli bir şekilde ilete biliyorlarsa, aynı yöntemle şifreleme işlemine ihtiyaç duyulmadan açık metni de güvenli olarak bir-birine ilete bilirler. Bu yüzden anahtarın açık metinle aynı uzunlukta olması tek kullanımlık şifreleme yönteminin kullanılmasını anlamsız kılıyor (Bağcıoğlu, 2007).

3.6.2 Tek kullanımlık anahtarın (One Time Pad) güvenliği

Tek kullanımlık anahtarın güvenliğini matematiksel olarak aşağıdaki formüllerle inceleye biliriz (Bağcıoğlu, 2007):

Her m açık metin ve p şifreli metin için, m'nin herhangi bir anahtar ile şifrelendiğinde p'yi oluşturması olasılığı, m'yi şifrelemek için kullanıldığında p'yi oluşturacak anahtar sayısının toplam anahtar sayısına bölümüne eşittir.

$$\forall m, p: O[Ş(a, m) = p] \ = \ |\{a: Ş(a, m) = p\}| \ / \ |A|$$

Tek kullanımlık şifrenin çalışma prensibi göz önünde bulundurulduğunda belirli bir açık metni belirli bir şifreli metine dönüştürecek yalnızca bir tek anahtar olduğu ve bu anahtarın da $m \oplus p = m \oplus (a \oplus m) = a$ formülü ile bulunduğu görülür. O halde tek kullanımlık anahtarlar için:

$$\forall m, p: O[Ş(a, m) = p] \ = \ 1 \ / \ |A|' dır.$$

Bu olasılığın açık metin m ve şifreli metin p için aynı olması tek kullanımlık anahtar yönteminin mutlak güvenli olduğunu kanıtlıyor. Böyle olduğu takdirde tek kullanımlık anahtar yöntemi üzerine her hangi şifreli metin saldırısının olmadığı kanaatine gele biliriz ve bu da mutlak güvenliğin bahsettiğimiz şifreleme yöntemi için en önemli taraf olduğunu gösteriyor. Ancak şifreli metin saldırısına karşı dayanıklı olan tek kullanımlık anahtar yöntemi üzerinde farklı saldırı çeşitleri de mevcuttur (Bağcıoğlu, 2007).

3.6.3 Vernam şifreleme (one time pad)

1918 yılında Gilbert Vernam'ın geliştirdiği one time pad (Vernam) şifreleme tekniği sınırsız teknolojik güce sahip olan düşmanlara karşı bile yüksek güvenlik sağlaya bilen bir şifreleme sistemidir. Başka bir tanımla düşmanın ne kadar hesaplama gücüne sahip olmasından bağımsız olarak, tek kullanımlık şifre ile şifrelenmiş metni kırması mümkün değildir. Aynı anahtarın hem şifreleme hem de şifreçözme işleminde kullanıldığından, vernam şifreleme algoritması simetrik şifreleme tekniklerine dahil olmaktadır. Şifreleme işleminde anahtar rastgele seçildiğinden, elde edilen şifreli metin de rastgele bitlerden oluşur ve bu da anahtarla ilgili bilgi elde edilmesini imkansız yapar. Şifreleme işlemini gerçekleştirmek için düz metinle aynı uzunlukta rastgele anahtar seçilir ve düzmetinle beraber anahtardakı harfler de sayılarla işaretlenir. Düz metinle anahtar İngilizce'de Mod26, Türkçe'de Mod29'a göre toplanır (Şenay, 2012).

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*
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28

Örnek olarak “İSTANBUL AYDIN ÜNİVERSİTESİ” açık metnini tek kullanımlık anahtar (one time pad) yöntemi ile şifrelersek bu metindeki karakterlerin sayısal karşılığı “ 11 21 23 00 16 01 24 14 00 27 04 10 16 25 16 11 26 05 20 21 11 23 05 21 11 ” şeklinde olur. Şifreleme işlemi için, düz metinle aynı uzunlukta, ya da daha da uzun rastgele anahtar seçilir. Örneğin “FEN BİLİMLERİ ENSTİTÜSÜ BİLGİSAYAR

MÜHENDİSLİĞİ BÖLÜMÜ” anahtarının düz metinle aynı uzunluktakı kısmını alıp kullanalım. 25 harflik düz metini şifrelemek için anahtardan aldığımız ilk 25 harf “06 05 16 01 11 14 11 15 14 05 20 11 05 16 21 23 11 23 25 21 25 01 11 14 07”olur. Anahtar ve açık metinin sayısal karşılıklarına aşağıdakı formülü uygularsak sonuç olarak şifreli metin oluşur. Formülde Ş şifreli metini, A anahtarı, D düz metni temsil etmektedir.

$$Ş = (D + A)mod29$$

Düz metin: İ$STANBUL\ AYDIN$ ÜN$İVERSİTES$İ

Anahtar: $FEN\ BİLİMLER$İ $ENSTİTÜS$Ü $BİLGİSAYAR$

Düz metin: 11 21 23 00 16 01 24 14 00 27 04 10 16 25 16 11 26 05 20 21 11 23 05 21 11

Anahtar: 06 05 16 01 11 14 11 15 14 05 20 11 05 16 21 23 11 23 25 21 25 01 11 14 07

$$(11 + 6)mod29 = 17$$

$$(21 + 5)mod29 = 26$$

$$(23 + 16)mod29 = 10$$

$$(0 + 1)mod29 = 1$$

$$\vdots$$

Şifre metin: 17 26 10 01 27 15 06 00 14 03 24 21 21 12 08 05 08 28 16 13 7 24 16 06 18

sonucuna ulaşırız. Bu dizinin harf karşılığını oluşturarak “$OVIBYMFALÇUSSJĞEĞZNKGUNF$Ö” şifreli metinini elde ederiz.

Şifre çözme işlemi zamanı, aynı anahtarla şifrelemede yapılan işlemlerin tersi yapılarak düzmetin elde edilmektedir. Anahtar ve şifreli metinin alfabedeki sayısal karşılıklarına aşağıdakı formül uygulandığında düzmetin karakterlerinin alfabedeki sayısal karşılığına ulaşılmaktadır. Formülde D düzmetini, Ş şifreli metini, A(ş) şifre çözme anahtarını temsil etmektedir.

$$D = (\ Ş + A(ş)\ ) mod 29$$

Şifre çözme anahtarını elde etmek için, şifrelemede kullanılan anahtarın alfabedeki harf sayısından (29) çıkarılması gerekmektedir. Çıkarma işlemi yapıldığında,

Anahtar: 06 05 16 01 11 14 11 15 14 05 20 11 05 16 21 23 11 23 25 21 25 01 11 14 07

$$29 - 6 = 23$$

$$29 - 5 = 24$$

$$29 - 16 = 13$$

$$\vdots$$

anahtar 2: 23 24 13 28 18 15 18 14 15 24 9 18 24 13 08 06 18 06 04 08 04 28 18 15 22

şifre çözme anahtar dizini (anahtar 2) elde edilmektedir.

Şifre çözme işlemi gerçekleştirildiğinde,

Ş***ifreli metin***: *PVIBYMFALÇUSSJ*Ğ*E*Ğ*ZNKGUNF*Ö

Ş***ifreli metin***: 17 26 10 01 27 15 06 00 14 03 24 21 21 12 08 05 08 28 16 13 7 24 16 06 18

anahtar 2 : 23 24 13 28 18 15 18 14 15 24 9 18 24 13 08 06 18 06 04 08 04 28 18 15 22

$$(17 + 23) mod 29 = 11$$

$$(26 + 24) mod 29 = 21$$

$$(10 + 13) mod 29 = 23$$

$$\vdots$$

Açık metin : 11 21 23 00 16 01 24 14 00 27 04 10 16 25 16 11 26 05 20 21 11 23 05 21 11

açık metin dizini elde edilmektedir. Bu sayıların alfabedeki harf karşılığı açık metinin kendisini "İSTANBUL AYDIN ÜNİVERSİTESİ" -ni vermektedir.

4 ARAŞTIRMADA KULLANILAN YÖNTEMLER, TEKNİKLER VE MATERYALLER

Bu tez çalışmasında, Tek Kullanımlık Anahtarla şifreleme tekniklerinde kullanılan anahtarın tamamen rastgele olması ve metin boyutundan küçük olmaması zorunluğundan dolayı küçük anahtar kelime aracılığıyla rastgele (sözde rastgele) en az açık metin boyutlu Tek Kullanımlık Anahtar üretmek amaçlanmıştır. Tek Kullanımlık Anahtar üreten algoritma için ses tanıma yöntemi kullanılmıştır. Ses tanıma yöntemi ile Tek kullanımlık Anahtarı üretmek için önemli kısım olan sesin metin karşılığı elde edilmiştir. Daha sonra tanımlanan ses iki eşit kısıma bölünerek sonuçlar birbirleri ile XOR işlemine tabi tutulmuşdur. Ardından bir anahtar kelime seçilerek ilk XOR işleminden elde edilen sonuçla XOR işlemine sokulmuştur. İkinci XOR işleminin sonucunda Tek Kullanımlık Anahtar (One Time Pad) üretilmiştir. Algoritma geliştirildikten sonra üretilen anahtar şifreleme işleminde kullanılmıştır.

Algoritmanın güvenliğinin dayanıklı olması amacıyla Tek kullanımlık Anahtar için en önemli şart olan anahtarın rastgele olması şartı göz önüne alınmıştır. Anahtarın rastgele gözüken olması için rastgele ses ve bir anahtar sözcük kullanılmıştır. Araştırmada Tek Kullanımlık Anahtarın üretilmesi için Vernam Şifreleme Tekniğinin başlıca prensiplerine dayanılmıştır.

4.1 Yöntemler

Tek Kullanımlık Anahtar üreten (One Time Pad) algoritma için iki yöntem kullanılmıştır.

- Ses Tanıma Yöntemi
- XOR Operatörüne Tabi Tutma Yöntemi

4.1.1 Ses tanıma yöntemi

Tek Kullanımlık Anahtar (One Time Pad) yöntemiyle şifreleme yapılan algoritmalarda anahtarın rastgele olarak seçilmesi zorunluğu olduğundan dolayı Tek Kullanımlık Anahtarın üretilmesi kısımına büyük dikkat verilmiştir. Rastgele

gözüken Tek Kullanımlık Anahtarın üretilmesi amacıyla konuşulan sesin program tarafından tanınmasına çalışılmıştır. Konuşulan sesin tanıtılarak metin haline çevrilebilmesi için donanımsal olarak Windows 8.1 Pro işletim sistemli hp ProBook 4530s dizüstü bilgisayar ve bir mikrofon, yazılımsal olarak ise Microsoft Visual Studio 2015 C# dili ve ek olarak ses tanıma için C# dili üzerinde "Speech Recognition" kütüphanesi kurulmuş ve gereken nesne ve bileşenler kullanılmıştır. Programa mikrofon aracılığıyla söylenilen sesler program içerisinde belirtilmemiştir. Demo programına tanıtılmağa çalışılan ses rastgele kullanıla bilir ve demonun tanıtımı bölümünde de örnek olarak rastgele ses kullanılmış ve sesin program tarafından tanıtılmasına çalışılmıştır.

Mikrofon aktifleştirildiği zaman bir kaç saniyede binlerce bayt ses ala bilmektedir. Bu baytların bit karşılıkları ise on bin bitlerle ölçülmektedir. Projenin bu kısmında çaba gösterilerek yapılmağa çalışılan işlem, küçük zaman süresinde mikrofona söylenilen birkaç kelimelik sesin on bin bitlerle ölçülen ses verileri içerisinden seçilerek filtrelenebilmesi ve filtrelenmiş sesin metin karşılığını kullanarak rastgele anahtar üretebilmektir. Eğer birkaç saniyede mikrofonun aldığı sesi herhangi bir program aracılığıyla algılayıp yazıya çevirmeden bir dosyaya yazdırarsak karşımıza büyük hacmde bitlerden oluşan bir verinin çıktığını göre biliriz. Bu kadar büyük boyutta bit dizisinin gerekli olmadığından dolayı tüm sesleri değil, mikrofona söylenilen sesleri filtreleyip bire bir metin karşılığı kutuya yazdırılmıştır.

4.1.2 XOR Operatörüne tabi tutma yöntemi

Ses tanıma sonucunda elde edilen metin Anahtarın üretilmesi için XOR işlemine sokulmuştur. Bunun için önce bir anahtar kelime seçilmiştir. Ses tanıma için kullanılan ses boyutu, Tek Kullanımlık Anahtarla şifreleme yönteminde anahtarın boyutu en az şifrelenecek metin boyutu kadar olması zorunluğundan dolayı uzun seçilmektedir. Ses tanıma aşamasında girilen ses birebir metine dönüştürüldüğü için metin boyutu seste kullanılan kelimelerin toplam boyutu kadar olmaktadır. Anahtar kelimeni ses tanımadan elde edilen metinle XOR işlemine sokmak için kendi sonuna kendi eklenerek ses tanımadan elde edilen metin boyutuna getirilerek XOR işlemine tabi tutulmuştur. XOR işlemi sonucunda rastgele gözüken (sözde rastgele) Tek Kullanımlık Anahtar (One Time Pad) üretilmiştir. Bir örnek aracılığıyla yapılan işlemler matematiksel olarak açıklamaya çalışılmıştır.

Örnek: Ses tanımadan elde edilen metini A, anahtar kelimeyi ise B olarak isimlendirelim.

A: “İSTANBUL AYDIN ÜNİVERSİTESİ”

B: “BİLGİSAYAR”

İlk adımda A iki eşit kısıma bölünerek sonuçlar XOR işlemine sokulmaktadır.

Eğer A’nın karakter sayısı tek ise o zaman ilk harfi kendi sonuna eklenmektedir.

Kelimeler arası boşlukların olmaması için kelimeler birleştirilmektedir.

“İSTANBULAYDINÜNİVERSİTESİİ”

Kısım1: İSTANBULAYDIN

Kısım2: ÜNİVERSİTESİİ

Kısım1, Kısım2 ve B’nin bit karşılıkları ASCII kodlarına göre Çizelge 4. 1’de gösterilmektedir.

Çizelge 4.1 A kısımlarının XOR işlemini içeren tablo.

Kısım1:	11111110100111010100100000110011101000010101011001100100 00011011001100010010010011001110
Kısım2:	11111110011101111111010110100010110100101010011111111110101 00100010110100111111111111110011
İşlem:	(XOR)
Sonuç:	00000000111010101011110110010001011100111111001001100100010011100111100010110110110011110 1

XOR işlemini, *Sonuç*’u B uzunluğuna bölerek her kısımı B ile veya B’yi kendi sonuna ekleme sonucunda *Sonuç* uzunluğuna eşitleyerek gerçekleştire biliriz.

B’nin bit uzunluğunun sonuna, ilk bitinden başlayarak kendi bitlerini *Sonuç* bit uzunluğuna eşit olana kadar ekleyerek *Sonuç* ile XOR işlemi yapılmaktadır:

Çizelge 4.2 Tek Kullanımlık Anahtarın üretim tablosu

Sonuç:	00000000111010101011110110010001011100111111001001100010011100111100010110110110011110
B:	10000101111111001100100011111111110100111000001101100110000011010010100001011111110011001
İşlem:	(XOR)
Anahtar:	10000101000101100111010101101110101000000111000100000010010000011010100001110100100000111

Sonuç ve B kısımına ait bitlerin XOR işlemine tabi tutulması sonucunda Tek Kullanımlık Anahtar üretilmektedir.

Tek Kullanımlık Anahtar:
10000101000101100111010101101110101000000111000100000010010000011010100001110100100000111

4.2 Teknik ve Materyaller

Tez çalışmasında geliştirilen algoritmanın hazırlanması aşamasında donanımsal olarak Windows 8. 1 Pro işletim sistemli hp ProBook 4530s dizüstü bilgisayar ve bir mikrofon, yazılımsal olarak ise Microsoft Visual Stdio 2015 C# dili ve ek olarak ses tanıma için kullanılmış C# dili üzerinde "Speech Recognition" kütüphanesi kurulmuştur.

Gereken kodlar C# dili ile yazılmış ve ara yüz tasarımları Windows Form üzerinde yapılmıştır. Projeye ait olan tüm aktivite diyagramları Microsoft PowerPoint 2010 programı üzerinde yapılmıştır.

Speech recognition kütüphanesi projede konuşmayı tanımak amacıyla kurulmuştur. C# dili dahilinde Referance üzerinden "Add.Referance" bölümünden Assembliesde seçili olan Framework içeriğinden System.Speech 4.0.0.0 versiyonu projeye yerleştirilmiştir. Bilgisayar üzerinde Framework'ün ".Net Framework 4.5.2" sürümü kurulu olmuştur.

Tez çalışmasının kaynak taraması aşamasında Yüksek Öğretim Kurumu Ulusal Tez Veri tabanı, EBSCO ve diğer veri tabanlarında kaynak taraması yapılmış ve ulaşılan kaynaklar incelenmiştir. Veri tabanları ve ağ üzerinde ulaşılabilen dokümanlar ile kitap, dergi, tez, makale, rapor gibi basılı materyaller incelenmiştir. Kriptografi

alanına ait ulaşılabilinen kitaplar detaylı şekilde incelenmiştir. Aynı zamanda İstanbul Aydın Üniversitesi kütüphanesinde gereken kitaplar, jurnaller, dergiler ve makaleler incelenmiş ve kaynak olarak kullanılmıştır.

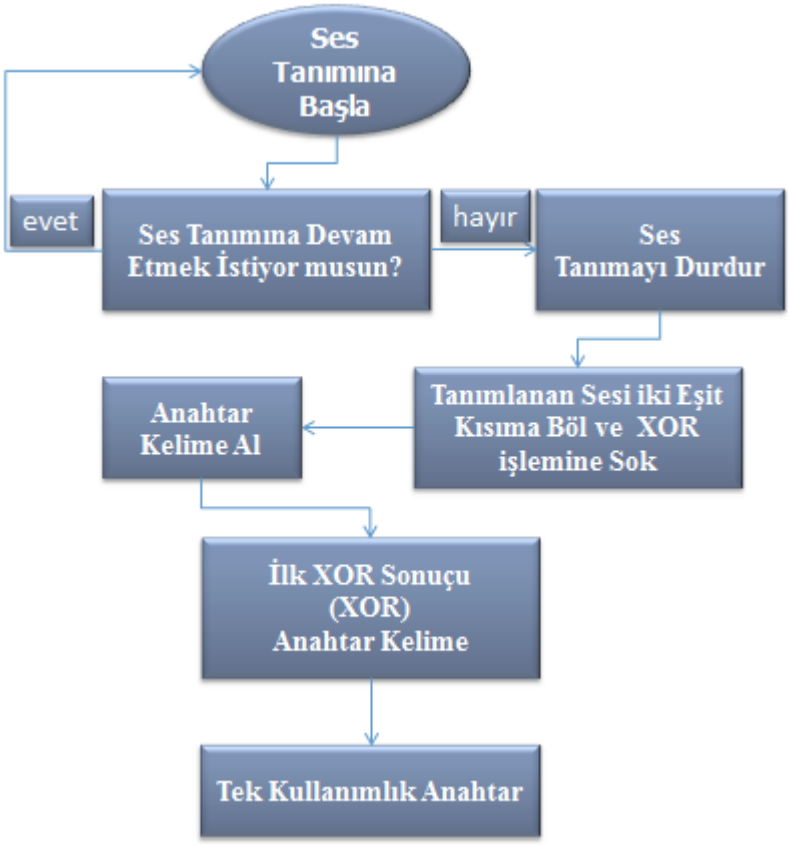
5 GELİŞTİRİLEN ALGORİTMANIN VE DEMONUN TANITIMI

Tez çalışmasında Tek Kullanımlık Anahtar (One Time Pad) tekniği kullanılan algoritmaların prensiplerine dayanarak yeni bir Tek Kullanımlık Anahtar (One Time Pad) üreten algoritma geliştirmeğe çalışılmıştır. Bu bölümde yapılan anahtar üretici algoritmanın tanıtımı ve şifreleme tekniğinin simülasyonu için Demo programının tanıtımı bulunmaktadır.

5.1 Algoritmanın Tanıtımı

Bu Araştırmada ses tanıma yöntemi ve XOR operatörü kullanımı ile Tek Kullanımlık Anahtar (One Time Pad) üreten yeni bir algoritma geliştirmek amaçlanmıştır. Tek kullanımlık anahtarla şifreleme tekniğinde anahtarın rastgele olması zorunlu olduğu için ses tanıma yöntemi ile rastgele gözüken anahtar üretmeye çalışılmıştır.

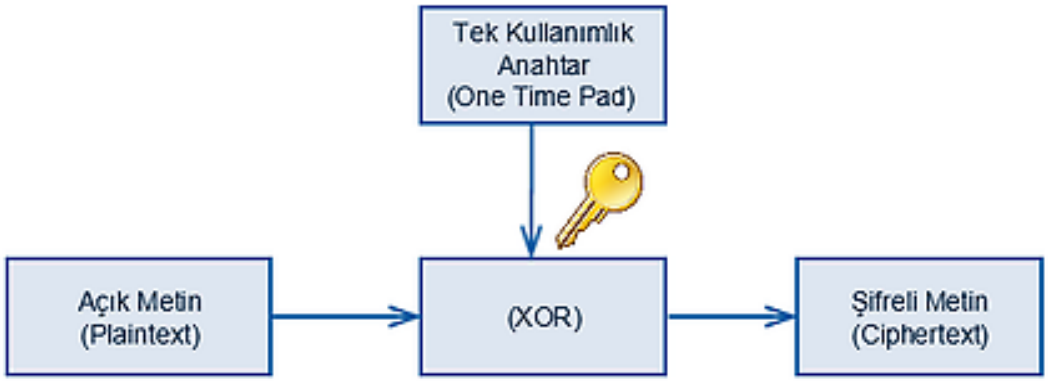
İlk adımda ses tanıma kısmında rastgele girilen ses programa tanıtılmıştır. Program dışardan girilen rastgele sesi alarak bire bir sesteki kelimelerin metnini vermektedir. Bunun için bir mikrofon kullanılmıştır. Mikrofon sesi alıyor, programa aktarıyor ve program sesi metne çeviriyor. Ses tanımından elde edilen metin iki eşit kısıma ayrılarak birbirleriyle XOR işlemine sokuluyor. Daha sonra XOR işleminden elde edilen sonucun ikilik tabanda ASCII kodları alınıyor. Bir sonrakı adımda rastgele gözüken (sözde ratgele) Tek Kullanımlık Anahtarı üretmek amacıyla bir anahtar kelime seçiliyor ve seçilen anahtar kelimenin ikilik tabanda ASCII kodları alınarak ilk XOR işleminden elde edilen sonuçla XOR işlemine sokuluyor. İkinci XOR işlemini gerçekleştirmek için anahtar kelimenin bit uzunluğu ilk XOR işleminden elde edilen sonuçun bit uzunluğuna eşitleniyor. Bit uzunluklarını eşitlemek için anahtar kelimenin bit karşılığının sonuna, ilk XOR işleminden elde edilen sonucun bit uzunluğuna eşit olana kadar kendi bitleri ekleniyor. İkinci XOR işleminin gerçekleşmesi sonucunda Tek Kullanımlık Anahtar (OneTime Pad) üretilmiş oluyor.



Şekil 5.1 Tek Kullanımlık Anahtarın (One Time Pad) Akış Diyagramı

Ses tanıma yönteminin kullanılmasında en önemli maksat rastgele gözüken anahtarın üretilmesidir. Anahtar Tamamen rastgele olduğu takdirde Tek kullanımlık Anahtarla şifreleme yöntemlerinin yüksek güvenliğe sahip olduğu kanıtlanmıştır (Frank, 1882), (Bağcıoğlu, 2007).

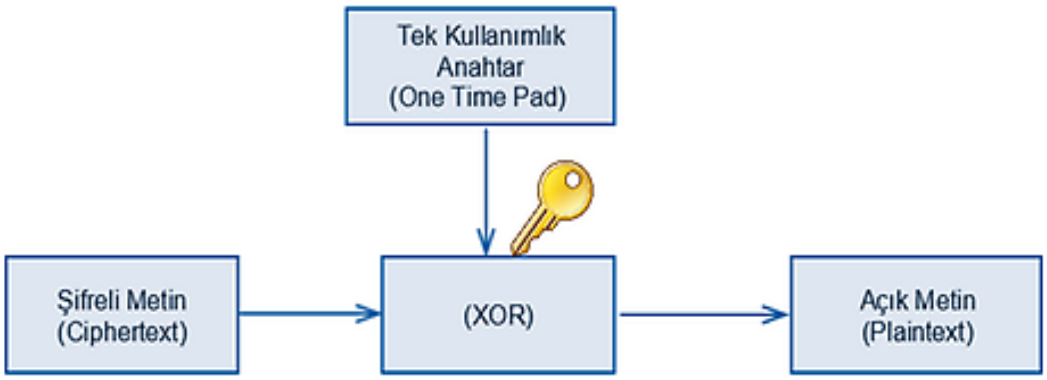
Şifreleme aşamasında anahtar üretildikten sonra şifrelenecek açık metinle XOR işlemine tabi tutuluyor. Şifrelemeyi gerçekleştirmek için açık metin boyutu Tek kullanımlık Anahtar boyutundan küçük ya da eşit olmalıdır. Eğer anahtar boyutu açık metin boyutundan büyükse, o zaman anahtarın ilk bitten başlayarak açık metin boyutu kadarı şifreleme için kullanılıyor.



Şekil 5.2 Şifreleme Diyagramı

Tek kullanımlık Anahtar (One Time Pad) ile açık metnin XOR işlemine sokulması sonucunda şifreli metin elde ediliyor.

Şifreleme için kullanılan Tek Kullanımlık Anahtar şifre çözme işlemi aşamasında da kullanılmaktadır. Şifre çözme işlemini gerçekleştirmek için Tek Kullanımlık Anahtar (One Time Pad) şifreleme işleminin sonucunda elde edilen şifrelenmiş metinle XOR işlemine tabi tutulmaktadır. XOR işleminin sonucunda şifre çözme işlemi bitmekte ve metnin deşifre edilmiş haline ulaşılmaktadır. Şekil 5. 3' de şifre çözme işleminin genel diyagramı verilmiştir.



Şekil 5.3 Şifre Çözme Diyagramı

Örnek:

Geliştirilen Proje için hazırlanmış Demo programının ses ve konuşma tanıma dil paketi İngiliz dili olduğu için örnekte sesin metin karşılığı İngilizce verilmektedir.

Girilen sesin metin karşılığı: I am learning English and Spanish

Anahtar Kelime: Anahtar

Açık Metin: Beni şifrele

Sesin metini, anahtar kelime ve açık metinin bit karşılıkları Çizelge 5. 1' de gösterilmektedir. Anahtarın üretilmesi için kelimeler arası boşluklar silinerek kelimeler birleştirilmektedir.

Girilen sesin metin karşılığı: IamlearningEnglishandSpanish

Anahtar Kelime: Anahtar

Açık Metin: Benişifrele

Çizelge 5.1 Örnek Şifreleme İçin Girdilerin Bit Karşılıkları

Girilen ses metninin bit karşılığı:	100100111000011101101110110011001011100001111001 011011101101001110111011001111000101110111011001 111101100110100111100111101000110000111011101100 100111001111100001100001110111011010011110011110 100011011010
Anahtar kelimenin bit karşılığı:	100000111011101100001110100011101001100001111000
Açık metinin bit karşılığı:	100001011001011101110110100111111111010011100110 111001011001011101100110 0101

İlk aşamada girilen ses metninin bit karşılığı iki eşit kısıma bölünerek birbirleri ile XOR işlemine sokulmaktadır.

Kısım1: *IamlearningEng*

Kısım2: *lishandSpanish*

Çizelge 5.2 Tanımlanan Sesin bit kısımlarının XOR işlemi

Kısım1:	1001001110000111011011101100110010111000011110010110111011010100111011101100111100010111011101100111110110
Kısım2:	1101100110100111100111101000110000111011101100100111001111000011000011101110110100111100111101000110110
İşlem:	(XOR)
Sonuç:	0100101000100000111000001000000100000011110010110001110100110001000111100010010101100001110100011110000000

Daha sonra XOR işleminden elde edilen *Sonuç* ile anahtar kelimenin bit karşılığını XOR işlemine sokarak Tek Kullanımlık Anahtarı elde etmek gerekmektedir.

İkinci XOR işlemini gerçekleştirmek için anahtar kelimenin bit sonuna kendi bitlerini ekleyerek bit boyutu *Sonuç'*un boyutu ile aynılaştırılmaktadır.

Çizelge 5.3 Tek Kullanımlık Anahtarın üretildiği tablo

Sonuç:	0100101000100000111000001000000100000011110010110001110100110001000111100010010101100001110100011110000000
İşlem:	(XOR)
Anahtar kelimenin bit karşılığı:	1000001110111011000011101000111010011000011110010100000011101110110000111010001110100110000111100101000011
Tek Kullanımlık Anahtar:	1100100110011011111011100000111110011011101100100101110111011101100001101100011111011111101100000111

Üretilen Tek Kullanımlık Anahtarı açık metinin bit karşılığı ile XOR işlemine sokmakla şifreleme işlemi gerçekleştirilmektedir. XOR işlemini gerçekleştirmek için anahtarın açık metin boyutu kadarı kullanılmaktadır.

Çizelge 5.4 Şifreleme işleminin yapıldığı tablo

Tek Kullanımlık Anahtar:	1100100110011011111011100000111110011011101100100101110111101111011101110000
İşlem:	(XOR)
Açık metnin bit karşılığı:	1000010110010111011101101001111111110100111001101110010110011010111011001100101
Şifreli metinin bit karşılığı:	0100110000001100100110001001000001101111010101001011100001111000110111010101

Şifre çözme işlemi aşamasında, şifreleme işleminde yapılan işlemlerin tersi yapılmaktadır. Açık metine ulaşmak için Tek Kullanımlık Anahtarla şifreli metin XOR işlemine sokulmaktadır.

Çizelge 5.5 Şifre çözme işleminin yapıldığı tablo

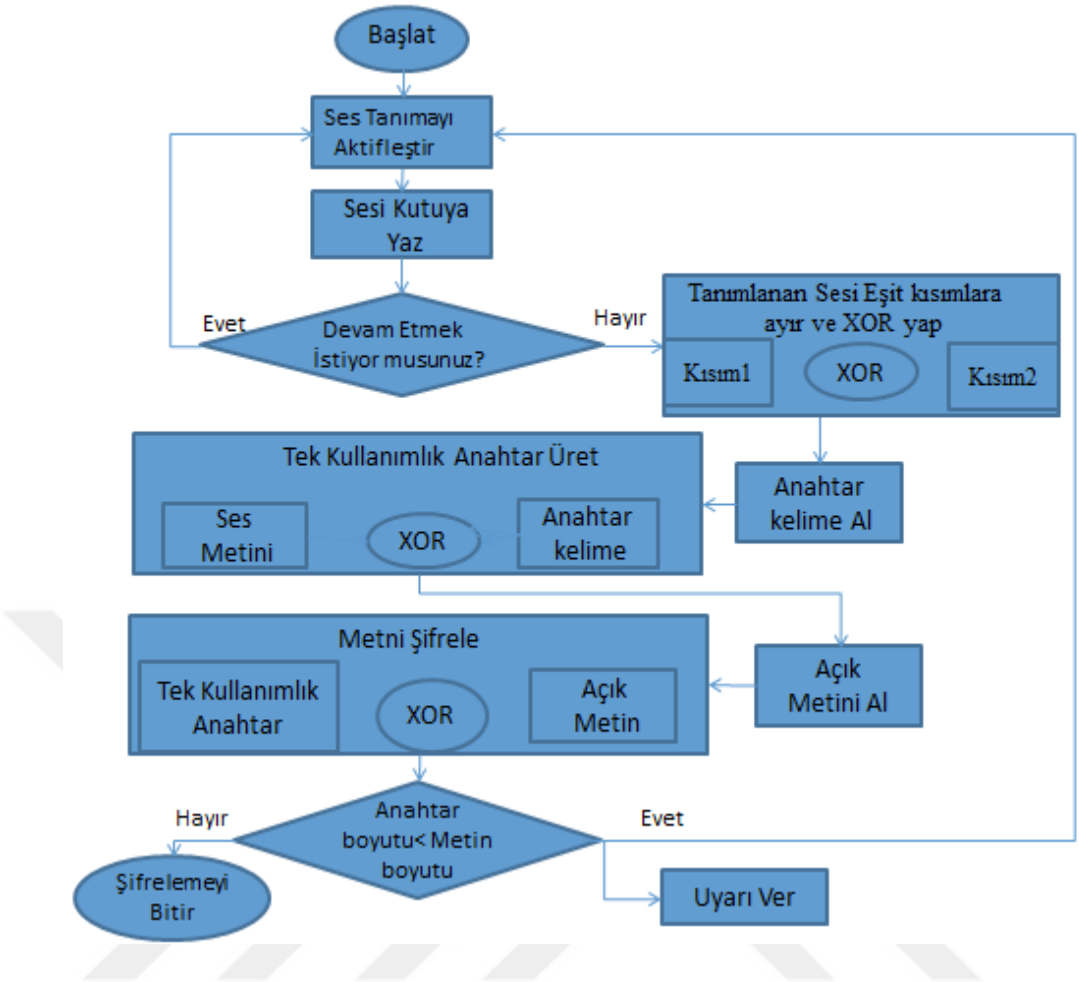
Tek Kullanımlık Anahtar:	1100100110011011111011100000111110011011101100100101110111101111011101110000
İşlem:	(XOR)
Şifreli metinin Bit karşılığı:	0100110000001100100110001001000001101111010101001011100001111000110111010101
Açık metinin bit karşılığı:	1000010110010111011101101001111111110100111001101110010110011010111011001100101

Şifre çözme işleminin sonucunda “Beni şifrele” açık metinine ulaşmış oluyoruz.

5.2 Demonun Tanıtımı

Geliştirilen proje için hazırlanmış demo programı One Time Pad olarak adlandırılmıştır. Programın tüm girdi ve çıktıları tek arayüzde yer almaktadır. Arayüz C# dili üzerinden windows form ile yapılmıştır. Programın kod kısımında ToolBox elemanları ve diğer komutlar kısaltılarak yazılmıştır. Örneğin label’ler lb, TexBox’lar txb, Buttonlar btn gibi adlandırılmıştır.

Formun aktivite diyagramı Şekil 5. 4’te verilmiştir:



Şekil 5.4 Şifreleme İşleminin Yapıldığı Formun Aktivite Diyagramı

Formun aktivite diyagramında gösterildiği gibi şifreleme işlemi ses tanıtmayla başlamaktadır. Demo programı çalıştırıldıktan sonra ses tanıma butonu ve mikrofon aktifleştiriliyor. Ardından sesin metin karşılığının yazılacağı ilk TexBox'a (txb_SesTex) tıklanıyor ve ses mikrofona söyleniyor. Alınan sesi ses tanıma komutları filtreliyor ve söylenilen sesi mikrofon aktifleştirildiği zaman duyulan diğer seslerden temizliyor ve filtrelenmiş sesi metin tipine dönüştürüyor. Aktivite diyagramından görüldüğü gibi eğer ses tanıtmaya devam etmek isteniyorsa ses girmeye devam ediliyor, ses tanıtma bittiyse program bir sonrakı işlemi yapmaktadır. Ardından program tanımlanan sesi TexBox'a yazdıktan sonra iki eşit kısıma bölerek kendi aralarında XOR işlemine sokmaktadır. Bir sonraki adımda program anahtar kelime istemektedir. Anahtar kelime ikinci TexBox'a (txb_anahtar) girildikten sonra anahtar üret butonu aktifleştiriliyor. Anahtar üret butonu ilk XOR işleminin sonucu ile girilen anahtar kelimeyi XOR işlemine sokuyor. XOR işlemi aşamasında program anahtar kelimenin bit karşılığının sonuna kendi bit karşılığını ekleyerek ilk XOR işleminden çıkan sonucun boyutuna eşitliyor ve XOR işlemini gerçekleştiriyor. XOR

işlemi sonucunda Tek Kullanımlık Anahtar (One Time Pad) üretilmektedir. Program üretilen anahtarı üçüncü TexBox'a (txb_OneTimePad) yazıyor. Anahtar üretildikten sonra şifrele butonu aktifleştiriliyor. Ardından dördüncü TexBox'a (txb_AcikMetin) şifrelenecek metin giriliyor. Şifrelenecek metin girildikten sonra, formun diyagramından görüldüğü gibi program şifrelenecek metinle üretilmiş Tek Kullanımlık Anahtar arasında XOR işlemini gerçekleştiriyor. XOR işlemi zamanı eğer anahtar boyutu metin boyutundan küçükse, program MessageBox aracılığıyla hatanın olduğunu belirtiyor ve işlemi başa döndürüyor. Eğer anahtar boyutu metin boyutundan küçük değilse o zaman şifreleme işlemini gerçekleştiriyor ve şifreli metini beşinci TexBox'a (txb_SifreMetin) yazıyor.

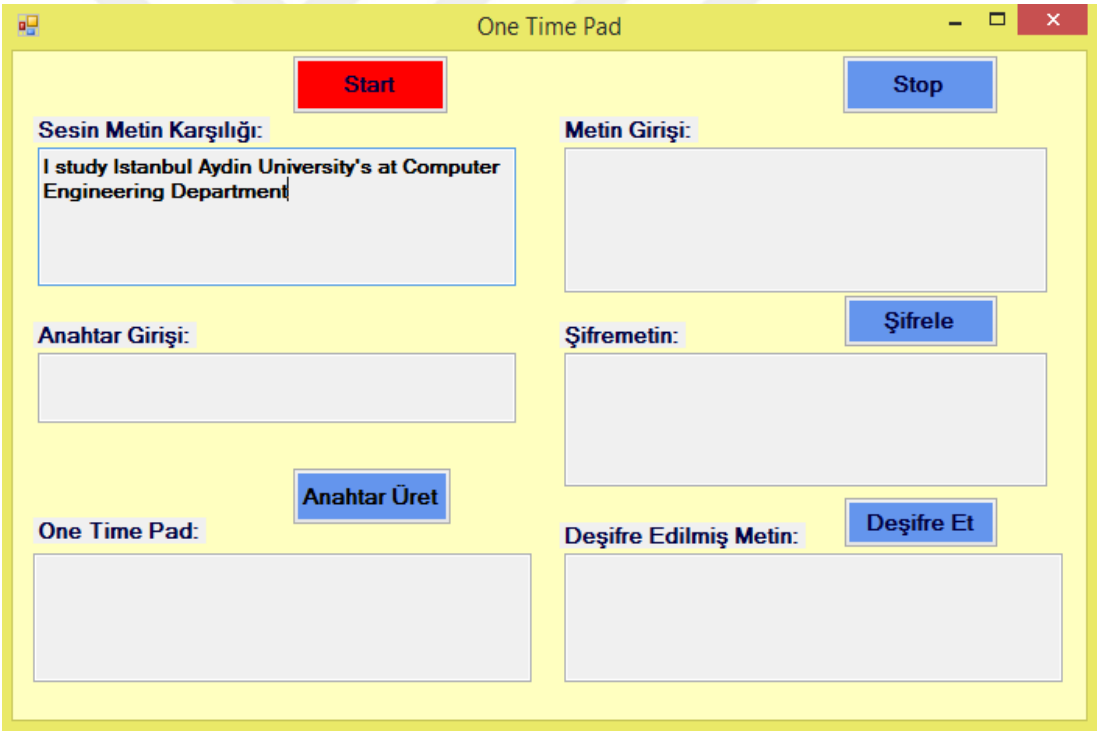
Şekil 5.5 Demo'nun Tüm İşlemlerin Gerçekleştirildiği Arayüz Formu

Şekil 5. 5'te görüldüğü gibi programın arayüz formunda 6 TexBox, 5 Button ve 6 label yer almaktadır. Butonlar ve TexBox'lar şifrelemenin akış diyagramına göre ardışık yerleştirilmiştir. Form yüklendiği zaman butonlar Start, Stop, Anahtar Üret, Şifrele ve Deşifre Et gibi ardışık aktifleşmektedirler. İlk TexBox (txb_SesTex) sesin metin karşılığını göstermek, ikinci TexBox (txb_anahtar) anahtar kelime girişi, üçüncü TexBox (txb_OneTimePad) üretilen Tek Kullanımlık Anahtarı göstermek,

dördüncü TexBox (txb_AcikMetin) şifrelenecek metin girişi, beşinci TexBox (txb_SifreMetin) şifrelenmiş metni göstermek, altıncı TexBox (txb_decription) ise deşifre edilmiş metni göstermek için kullanılmaktadır.

Start butonunu aktifleştirip sesin metin karşılığının yazılacağı TexBox'ı seçtiğimizde program girilen sesi filtreleyip konuşulan sesin bire-bir metin karşılığını seçili TexBox'a yazıyor. Form yüklendikten sonra aktifleştirilen button'nun rengi kırmızı renge çevriliyor ve diğer buttonların rengi pasif halde oldukları rengi alıyor. Konuşmayı durdurduğumuz zaman Start button'u aktif olsa da, konuşma dışında mikrofonun duyduğu sesi program algılamıyor. Stop button'una tıkladığımız zaman Start button'u pasifleşiyor ve program konuşmayı algılamıyor.

Şekil 5. 6'da Start button'u aktifleştirildiği zaman renginin değişmesi ve konuşulan sesin program tarafından seçilen TexBox'a yazılması gözükmektedir.



Şekil 5.6 Ses tanıtımına başlandıktan sonra arayüzün görünümü

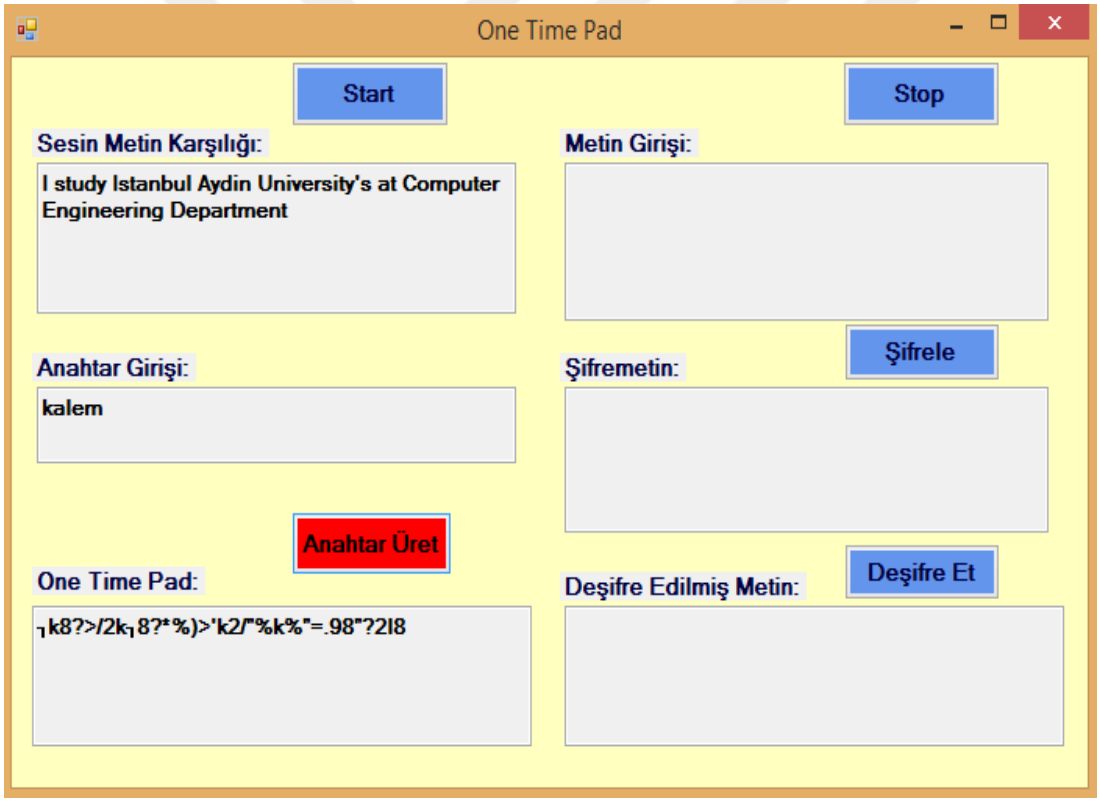
Programda ses tanıma ve konuşma dil paketi İngilizce dil paketi olduğu için Türkçe konuşmayı iyi algılamıyor. Program İngilizce "*I study at computer engineering department of Istanbul Aydin University*" sesini aldıktan sonra sesi filtreleyerek TexBox'a yazmaktadır.

Ses tanıtmanın ardından ikinci TexBox'a (txb_anahtar) anahtar kelime girilmektedir. Anahtar kelime girildikten sonra Anahtar Üret buttonuna tıklandığında program anahtar kelimeyle, tanımlanan sesin metin karşılığını XOR işlemine sokuyor ve işlemin sonucunu üçüncü TexBox'a (txb_OneTimePad) yazıyor. Program XOR işleminin sonucunu karakterler tipinde dönüştürüyor. İşlem sonucunda oluşan karakterler tipindeki ifade şifreleme için önemli olan Tek kullanımlık Anahtar (One Time Pad) olmaktadır.

Tanımlanan Ses: *I study at computer engineering department of Istanbul Aydin University*

Anahtar Kelime: *Kalem*

olsun.



Şekil 5.7 Anahtar Üret butonuna tıklama sonucunda arayüzün görünümü

Şekil 5.7'de karakterler tipinde üçüncü TexBox'a yazılan (k8?>/2k8?*%)>'k 2/"%k-%"=.98"?2l8) ifade Tek Kullanımlık Anahtar (One Time Pad) olmaktadır.

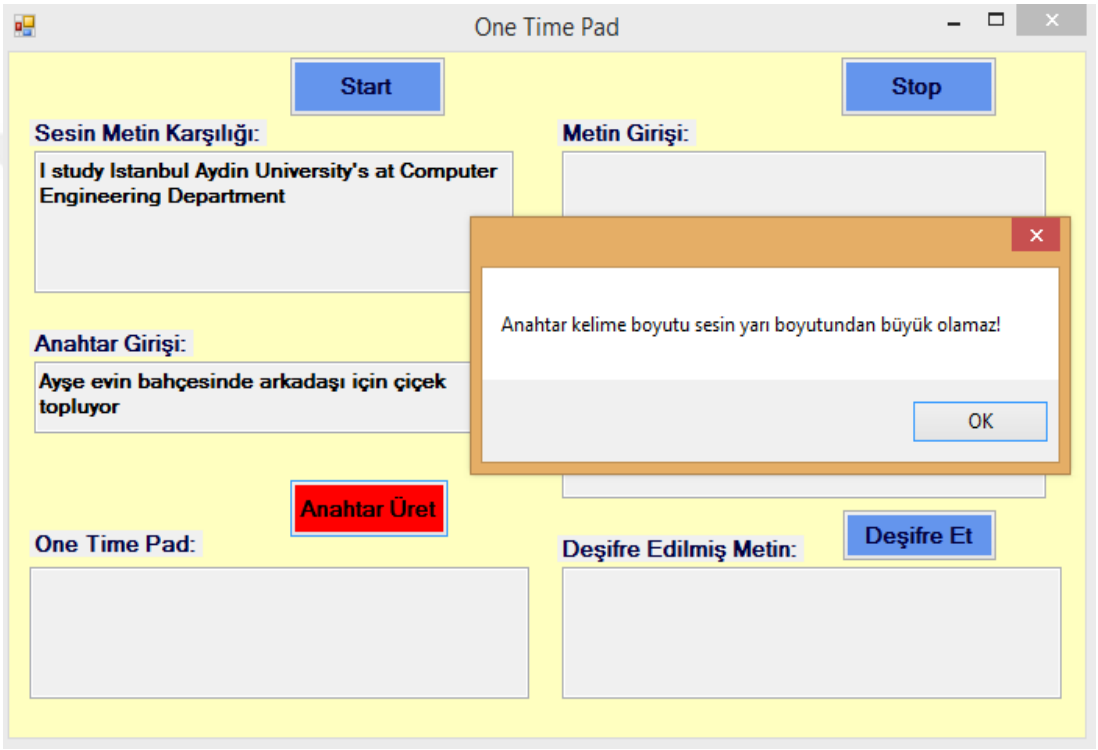
Tanımlanan sesin boyutu, iki eşit kısıma bölünerek Xor işlemi sokulduğdan dolayı iki defa küçülmektedir. Girilen anahtar kelimenin boyutu tanımlanan sesin metin

karşılığının yarı boyutundan büyük olduğunda, program MessageBox aracılığıyla "Anahtar kelimenin boyutu sesin boyutundan büyük olamaz" hatasını veriyor ve anahtar kelime ile tanımlanan ses metini arasında XOR işlemini gerçekleştirmiyor.

Tanımlanan Ses: *I study at computer engineering department of Istanbul Aydin University*

Anahtar Kelime: *Ayşe evin bahçesinde arkadaşı için çiçek topluyor*

olsun.



Şekil 5.8 Anahtar kelime boyutu tanımlanan sesin yarı boyutundan büyük olduğunda Anahtar Üret buttonuna tıklanınca çıkan arayüz ve uyarı mesajı.

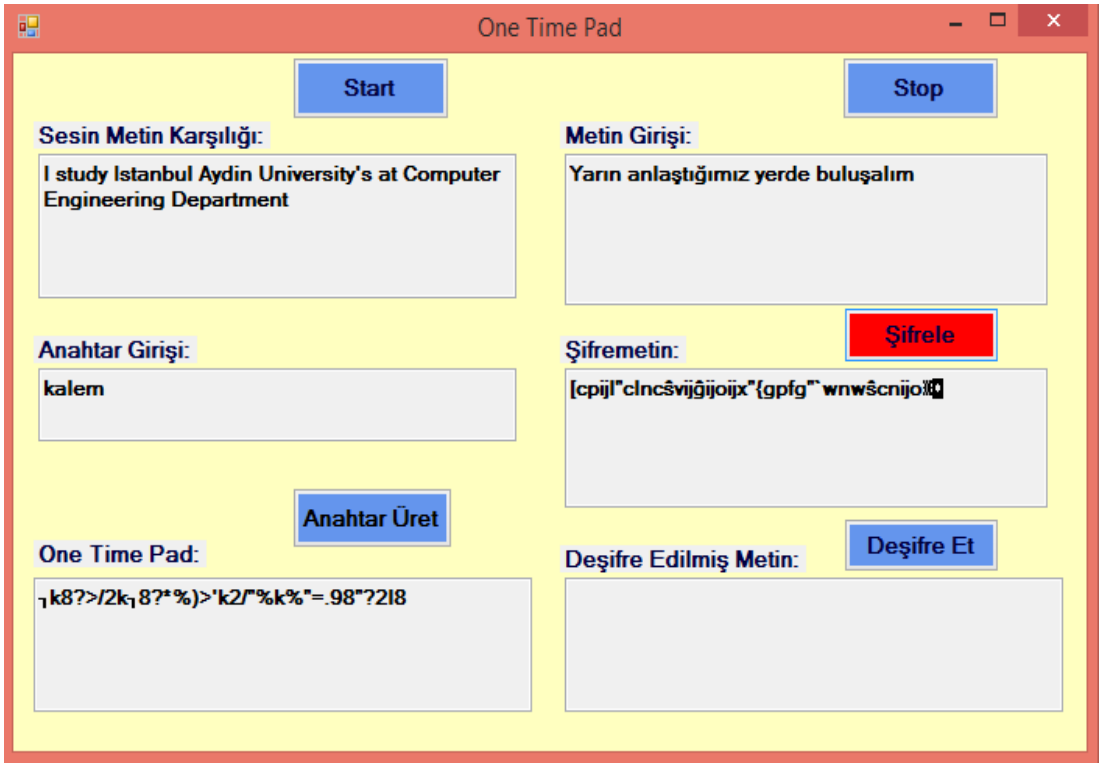
Tek Kullanımlık Anahtarın üretilebilmesi için, anahtar kelime boyutunun tanımlanan sesin metin karşılığının yarı boyutundan küçük girilmesi gerekmektedir. Bir sonraki adımda Tek Kullanımlık Anahtar (One Time Pad) üretildikten sonra şifreleme işlemini gerçekleştirmek amacıyla dördüncü TexBox'a (txb_AcikMetin) açık metin giriliyor. Bu adımda Tek Kullanımlık Anahtarla şifreleme yönteminde anahtar boyutunun açık metin boyutundan küçük olmaması prensibinin sağlanabilmesi için daha uzun konuşma sesi tanıtılmaktadır.

Konuşulan ses metini: *I study at computer engineering department of Istanbul Aydin University*

Anahtar kelime: *Kalem*

Şifrelenecek metin: *Yarın anlaştığımız yerde buluşalım*

olsun.



Şekil 5.9 Şifrele butonuna tıklandığında şifrelemede yapılan işlemlerin arayüz üzerinde görünümü.

Şifrele buttonuna tıklandığında program, üretilen Tek Kullanımlık Anahtarın ilk açık metin bitleri uzunluğunda olan kısmını alarak açık metnin bit karşılığı ile XOR işlemine sokuyor ve şifreleme işlemi gerçekleşmiş oluyor. Ardından XOR işlemi sonucunda elde edilen karakter tipindeki şifreli metni beşinci TexBox'a (txb_SifreMetin) yazıyor.

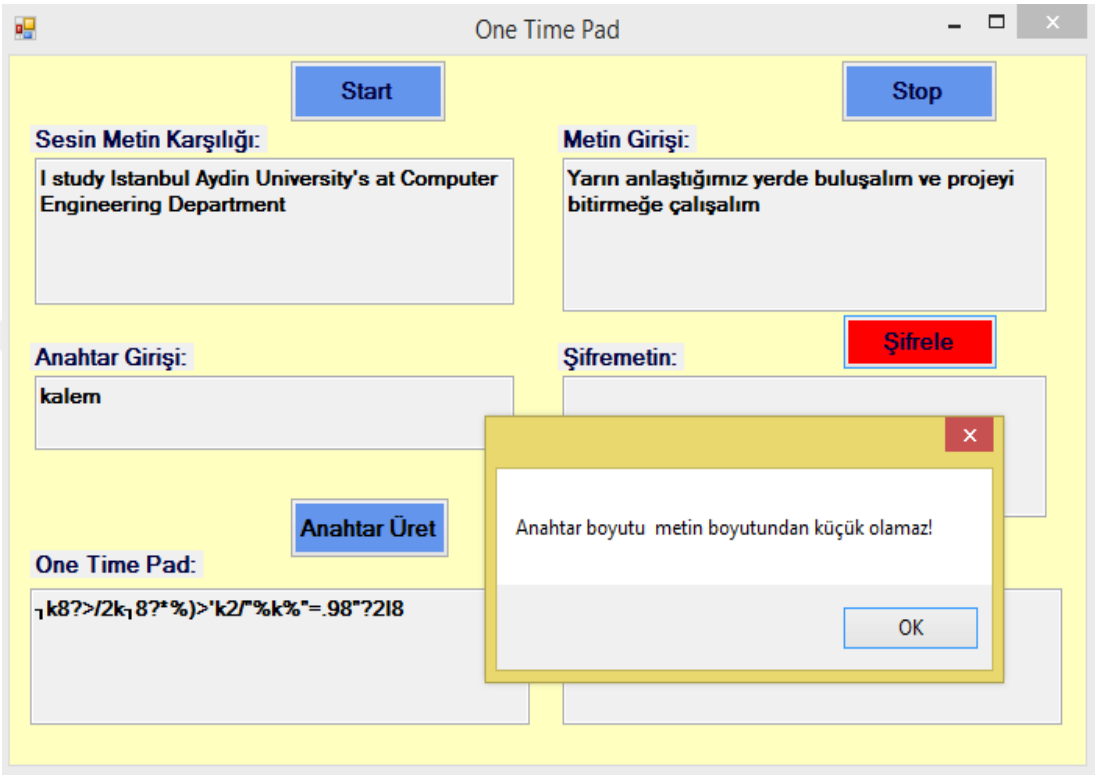
Şifreleme aşamasında üretilen Tek Kullanımlık Anahtarın boyutu açık metin boyutundan küçük olduğunda program şifrelemeyi gerçekleştirmiyor ve MessageBox aracılığıyla uyarı mesajı veriyor.

Konuşulan ses metni: *I study at computer engineering department of Istanbul Aydin University*

Anahtar kelime: *kalem*

Şifrelenecek metin: *Yarın anlaştığımız yerde buluşalım ve projeyi bitirmeğe çalışalım*

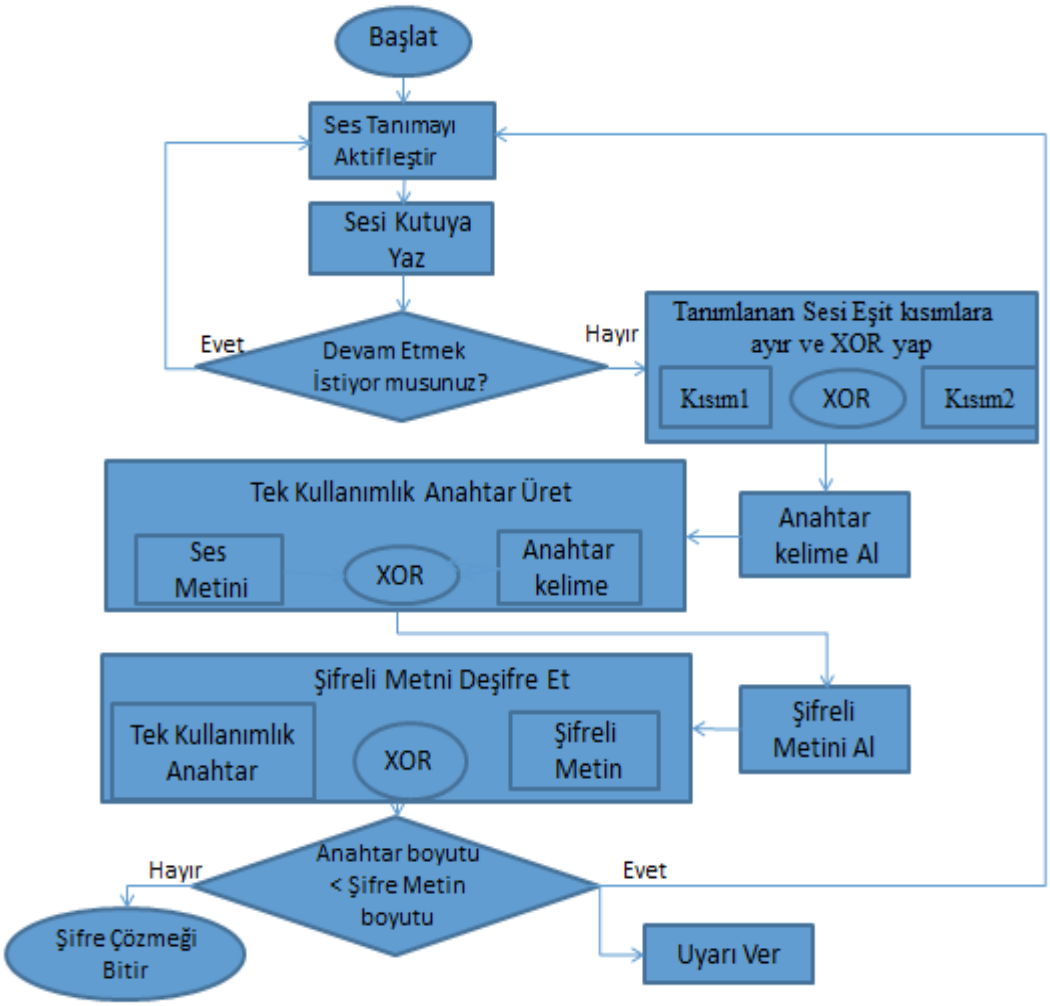
Olsun.



Şekil 5.10 Tek Kullanımlık Anahtar boyutu açık metin boyutundan küçük olduğunda yapılan şifreleme işlemleri ve uyarı mesajının verildiği arayüz.

Uyarı mesajını aldıktan sonra işlem başa dönmekte ve yeniden konuşma tanıma devam ettirilmektedir. Tek Kullanımlık Anahtarın boyutu Açık metin boyutundan büyük olana kadar konuşma tanıtması devam ettiriliyor. Tek Kullanımlık Anahtar Açık metinden büyük ya da eşit olduğunda şifreleme işlemi gerçekleşiyor.

Şifrelenmiş metinin çözülmesi için şifrelemede yapılan işlemlerin tersini yapmak yeterli olmaktadır. İşlemin gerçekleştirilmesi için şifreleme aşamasında Tek Kullanımlık Anahtar üretme amacıyla kullanılan konuşma sesi ve anahtar kelimenin aynısının kullanılması zorunludur. Şifre çözme işleminin yapıldığı formun aktivite diyagramı Şekil 5. 11'de gösterilmiştir.



Şekil 5.11 Form üzerinde yapılan şifre çözme işleminin aktivite diyagramı

Şifre çözme için kullanacağımız anahtarın şifrelemede kullanılan anahtarla aynı olmasından dolayı, diyagramdan görüldüğü gibi konuşma tanıtma ve Tek Kullanımlık Anahtar üretme adımlarında şifrelemede yapılan işlemlerin bire bir aynısı yapılmaktadır. Sonraki adımlarda ise şifreleme işlemindeki işlemlerin tersinin yapıldığı gözükmektedir.

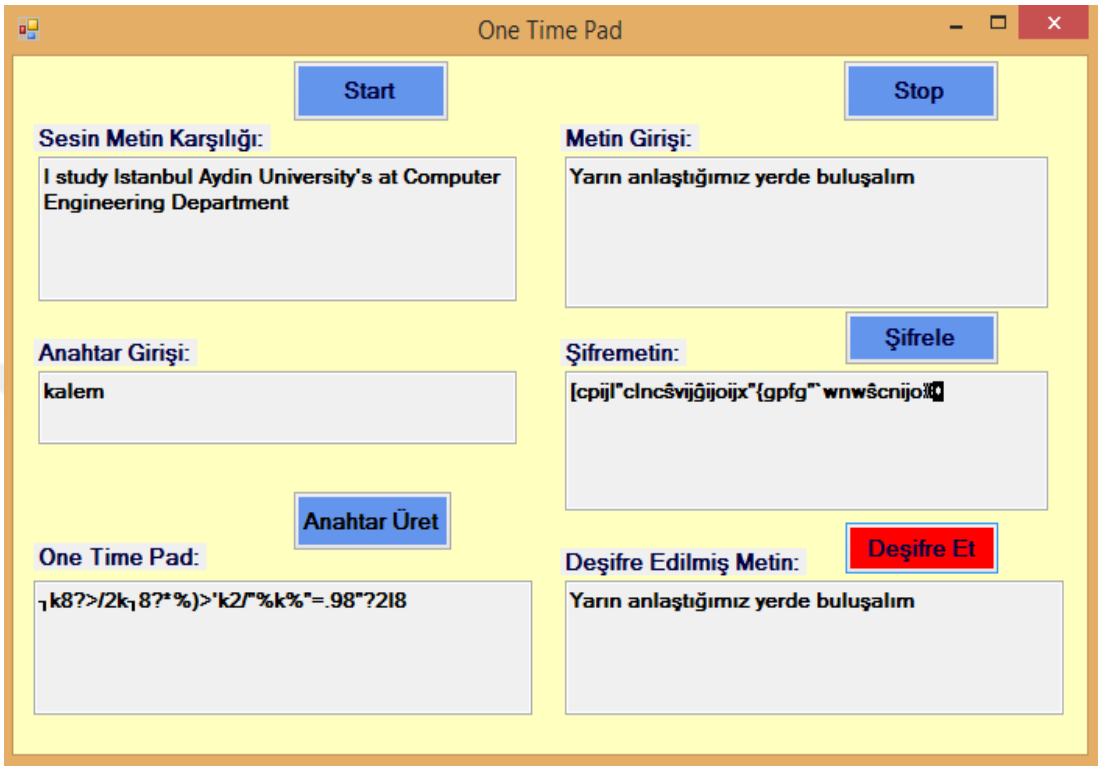
Şekil 5.12'de deneme amacıyla şifre çözme işleminin bir örnek üzerinden yapıldığı arayüz gösterilmektedir

Konuşulan ses metini: *I study Istanbul Aydin University's at Computer Engineering Department*

Anahtar kelime: *kalem*

Şifrelenecek metin: *Yarın anlaştığımız yerde buluşalım*

olsun. Bu örnekten elde edilen şifreli metnin şifre çözme işlemi yapılmaktadır. Şekil 5.12'de şifreleme işleminin sonucunda elde edilen şifreli metin ve Deşifre Et butonuna tıkladıktan sonra elde edilen açık metin gösterilmektedir. Şifreli metini deşifre etmek için Deşifre Et buttonuna tıklamak yeterli olmaktadır.



Şekil 5.12 Şifre çözme işlemi yapıldığında arayüzün görünümü.

Deşifre Et buttonuna tıklandığı zaman program üretilen Tek Kullanımlık Anahtarla şifreli metini XOR işlemine sokuyor ve işlemin sonucunu TexBox'a (txb_decription) yazıyor. Şekil 5.12'den görüldüğü gibi Deşifre Et buttonuna tıklandığı zaman XOR işleminin sonucu açık metinle bire bir aynı olarak program tarafından TexBox'a yazılıyor. Bu aşamada şifrelemede kullanılan Tek Kullanımlık Anahtarın aynısı kullanıldığı için anahtar boyutu şifreli metin boyutundan uzun olmaktadır ve bundan dolayı da program uyarı mesajı vermeden doğru biçimde çalışmaktadır.

6 GELİŞTİRİLEN ALGORİTMANIN PERFORMANS DEĞERLERİ

Çalışmada geliştirilen algoritmanın demo programına aid CPU kullanımı, bellek kullanımı ve işlemler için geçen zaman açısından performans analizleri yapılmıştır.

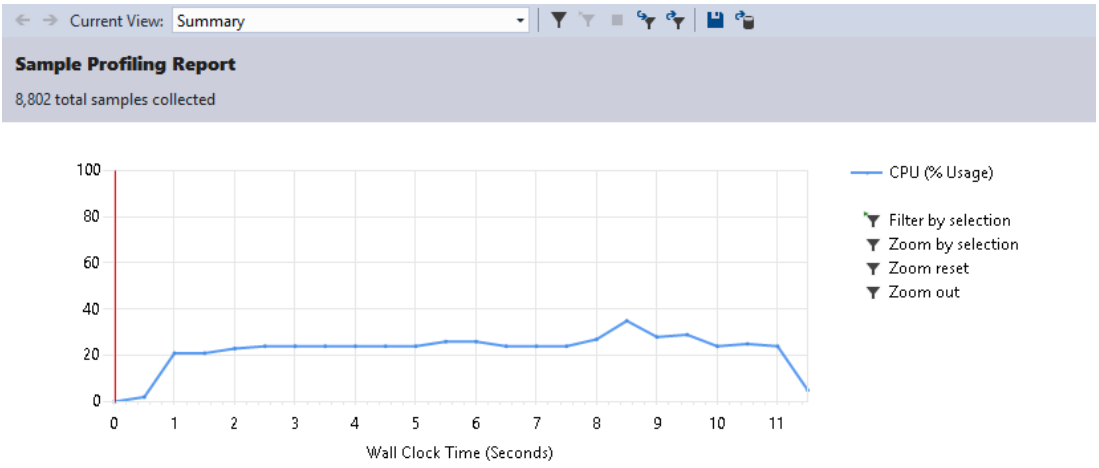
Performans analizlerinin yapılması için demo programı üzerinde örnek olarak "*I study at computer engineering department of Istanbul Aydin University*" ses metini, "*kalem*" anahtar kelimesi, "*yarın projeyi bitirelim*" açık metini kullanılmıştır.

Performans analizlerini gerçekleştirmek için debug'dan Start Diagnostic Tools Without Debugging seçeneği içeriğinden CPU Wizard'ı seçmek gerekmektedir.

6.1 İşlemci Kullanım (CPU sampling) Verileri

Current View içeriğinde farklı seçenekler bulunmaktadır. *Summary* seçeneğini seçtiğimizde CPU kullanım bilgilerinin genel hali görülmektedir.

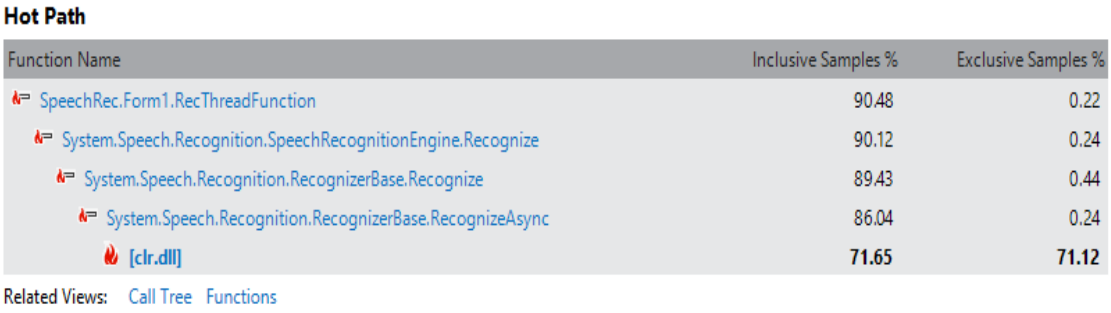
Şekil 6. 1' de CPU Sampling seçeneğinin içerdiği veriler görülmektedir:



Şekil 6.1"Yarın Projeyi bitirelim" açık metninin şifrelenmesine ait performans grafiği

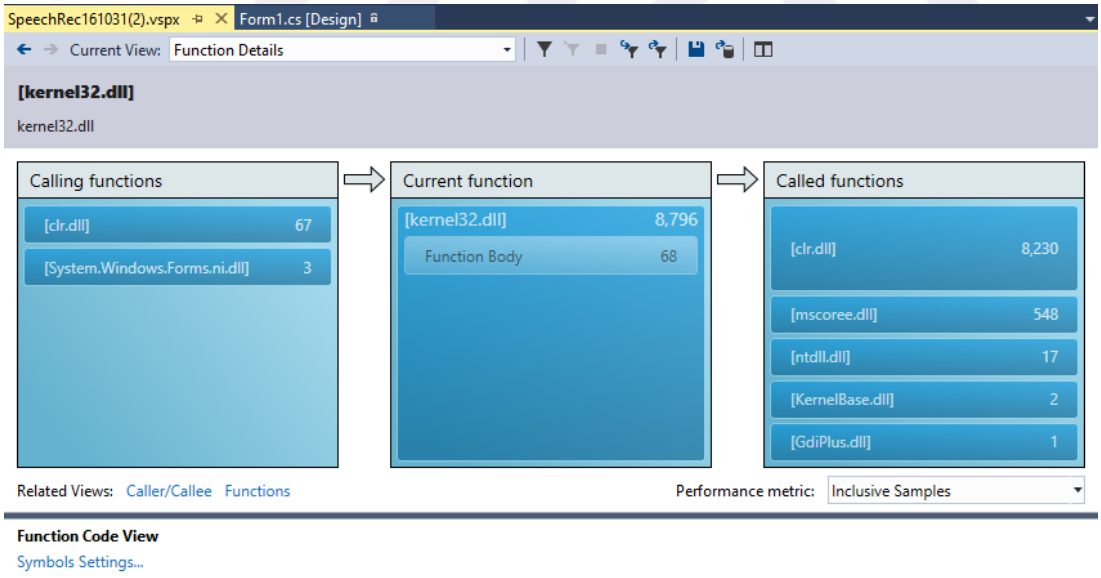
Program ilk çalıştırıldığı zaman CPU kullanımı %0'dan başlamakta, daha sonra şifreleme işleminde yapılan değişik adımlara göre CPU kullanım yüzdesi de değişmektedir.

Şekil 6. 2' de sistemi zorlayan sınıflar ve sistem üzerinde darboğaz oluşturan fonksiyonlar yer almaktadır.



Şekil 6.2"Yarın Projeyi bitirelim" açık metninin şifrelenmesinde sistemi darboğaz eden sınıflar

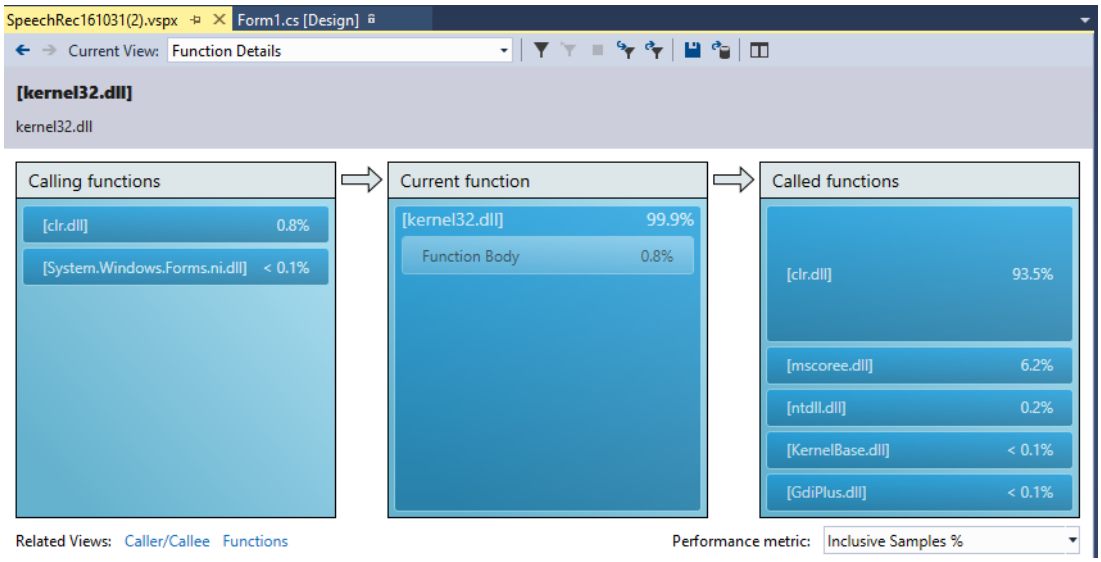
Hot Path bölümünde *Inclusive Samples* yüzdesi yüksek olan sınıflar sistemi en çok zorlayan sınıflar, *Exclusive Samples* bölümünde ise performans darboğazı oluşturan fonksiyonlar en yüksek yüzdeye sahiptir. Bu fonksiyonları detaylı şekilde göre bilmek için *Current View* kısmından *Function Details* seçeneği seçildikte detayları görmek mümkündür.



Şekil 6.3 Sistemde darboğaz oluşturan fonksiyonlar

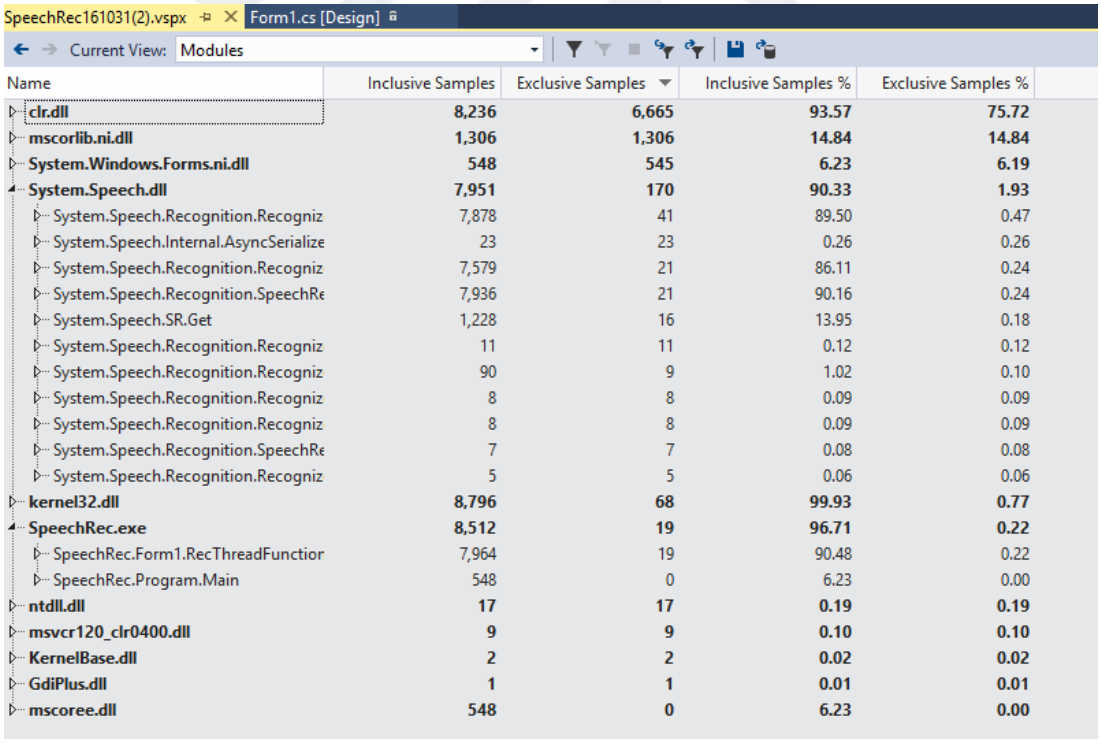
Şekil 6. 3' te çağrılan ve güncel fonksiyonların sayı görülmektedir.

Çağırılan ve güncel funksiyonların %'lik değerleri Şekil 6. 4' de görülmektedir. Bu kısıma ulaşmak için *Performance metric* seçeneğinden *Inclusive Samples* % seçimini yapmak gerekmektedir.



Şekil 6.4 Sistemde darboğaz oluşturan fonksiyonların %'leri

Sistemi zorlayan sınıf ve fonksiyonların hangiler olduğu ve hangi satırlarda yerleştiği bilgisine ulaşmak için *Current View* kutusundan *Modules* seçeneği seçilmektedir. Şekil 6. 5' te Modules seçeneğine ait detaylı bilgi görülmektedir.



Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
clr.dll	**8,236**	**6,665**	**93.57**	**75.72**
mscorlib.ni.dll	**1,306**	**1,306**	**14.84**	**14.84**
System.Windows.Forms.ni.dll	**548**	**545**	**6.23**	**6.19**
System.Speech.dll	**7,951**	**170**	**90.33**	**1.93**
System.Speech.Recognition.Recogniz	7,878	41	89.50	0.47
System.Speech.Internal.AsyncSerialize	23	23	0.26	0.26
System.Speech.Recognition.Recogniz	7,579	21	86.11	0.24
System.Speech.Recognition.SpeechRe	7,936	21	90.16	0.24
System.Speech.SR.Get	1,228	16	13.95	0.18
System.Speech.Recognition.Recogniz	11	11	0.12	0.12
System.Speech.Recognition.Recogniz	90	9	1.02	0.10
System.Speech.Recognition.Recogniz	8	8	0.09	0.09
System.Speech.Recognition.Recogniz	8	8	0.09	0.09
System.Speech.Recognition.SpeechRe	7	7	0.08	0.08
System.Speech.Recognition.Recogniz	5	5	0.06	0.06
kernel32.dll	**8,796**	**68**	**99.93**	**0.77**
SpeechRec.exe	**8,512**	**19**	**96.71**	**0.22**
SpeechRec.Form1.RecThreadFunctior	7,964	19	90.48	0.22
SpeechRec.Program.Main	548	0	6.23	0.00
ntdll.dll	**17**	**17**	**0.19**	**0.19**
msvcr120_clr0400.dll	**9**	**9**	**0.10**	**0.10**
KernelBase.dll	**2**	**2**	**0.02**	**0.02**
GdiPlus.dll	**1**	**1**	**0.01**	**0.01**
mscoree.dll	**548**	**0**	**6.23**	**0.00**

Şekil 6.5 Sınıf ve fonksiyonların bulundukları satırları gösteren CPU kullanımı detayları

Şekil 6. 5'de sınıf ve fonksiyonların CPU kullanımı detayları içerisinde, geliştirilen algoritma için yapılmış demo programında kullanılan Speech Recognition fonksiyonlarının CPU kullanımı detaylı şekilde görülmektedir.

Şekil 6. 6'da Current View kutusundan Functions seçeneğine ait verilerin detayları gösterilmektedir.

SpeechRec161031(2).vspx | Form1.cs [Design]

Current View: Functions

Function Name	Inclusive Samples	Exclusive Samples	Inclusive Samples %	Exclusive Samples %
[clr.dll]	8,236	6,665	93.57	75.72
[mscorlib.ni.dll]	1,306	1,306	14.84	14.84
[System.Windows.Forms.ni.dll]	548	545	6.23	6.19
[kernel32.dll]	8,796	68	99.93	0.77
System.Speech.Recognition.Recognizerl	7,878	41	89.50	0.47
System.Speech.Internal.AsyncSerialized\	23	23	0.26	0.26
System.Speech.Recognition.Recognizerl	7,579	21	86.11	0.24
System.Speech.Recognition.SpeechRec	7,936	21	90.16	0.24
SpeechRec.Form1.RecThreadFunction	7,964	19	90.48	0.22
[ntdll.dll]	17	17	0.19	0.19
System.Speech.SR.Get	1,228	16	13.95	0.18
System.Speech.Recognition.Recognizerl	11	11	0.12	0.12
[msvcr120_clr0400.dll]	9	9	0.10	0.10
System.Speech.Recognition.Recognizerl	90	9	1.02	0.10
System.Speech.Recognition.Recognizerl	8	8	0.09	0.09
System.Speech.Recognition.Recognizerl	8	8	0.09	0.09
System.Speech.Recognition.SpeechRec	7	7	0.08	0.08
System.Speech.Recognition.Recognizerl	5	5	0.06	0.06
[KernelBase.dll]	2	2	0.02	0.02
[GdiPlus.dll]	1	1	0.01	0.01
[mscoree.dll]	548	0	6.23	0.00
SpeechRec.Program.Main	548	0	6.23	0.00

Şekil 6.6 Şifreleme işlemleri için kullanılan fonksiyonların CPU kullanım oranları

Şekil 6. 7'de Current View kutusundan *Marks* seçeneği seçildiği zaman CPU kullanımına ait süreler detaylı olarak görülmektedir.

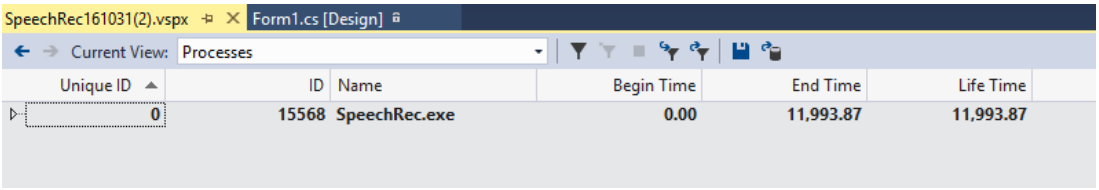
SpeechRec161031(2).vspx | Form1.cs [Design]

Current View: Marks

Mark ID	Mark Name	Timestamp	CPU Usage %
0	Start of Program	0.00	0
1	VSPXAutoMark	500.00	2
2	VSPXAutoMark	1,000.00	21
3	VSPXAutoMark	1,500.00	21
4	VSPXAutoMark	2,000.00	23
5	VSPXAutoMark	2,500.00	24
6	VSPXAutoMark	3,000.00	24
7	VSPXAutoMark	3,500.00	24
8	VSPXAutoMark	4,000.00	24
9	VSPXAutoMark	4,500.00	24
10	VSPXAutoMark	5,000.00	24
11	VSPXAutoMark	5,500.00	26
12	VSPXAutoMark	6,000.00	26
13	VSPXAutoMark	6,500.00	24
14	VSPXAutoMark	7,000.00	24
15	VSPXAutoMark	7,500.00	24
16	VSPXAutoMark	8,000.00	27
17	VSPXAutoMark	8,500.00	35
18	VSPXAutoMark	9,000.00	28
19	VSPXAutoMark	9,500.00	29
20	VSPXAutoMark	10,000.00	24
21	VSPXAutoMark	10,500.00	25
22	VSPXAutoMark	11,000.00	24
23	VSPXAutoMark	11,500.00	5
4,294,967,295	End of Program	12,447.86	0

Şekil 6.7 CPU kullanımına ait detaylı süre bilgisi

Şekil 6. 8'de, Processes seçeneği seçildiğinde geliştirilen algoritmanın demo programının CPU kullanımına ait süreler genel olarak görülmektedir.



Şekil 6.8 CPU kullanımına ait süre bilgisi

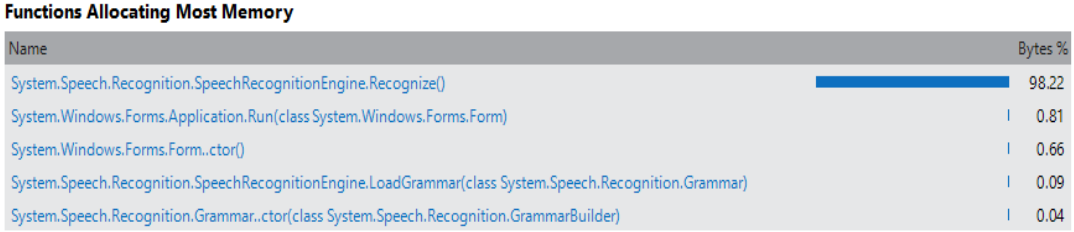
Şekil 6. 8'de görüldüğü gibi demo programının CPU kullanım süresi 11, 993, 87 millisaniye olarak görülmektedir.

6.2 Bellek Kullanımı (.Net memory allocations) Verileri

Şekil 6. 9'da bellek kullanım grafiği gösterilmektedir.

Şekil 6.9 Bellek kullanımının grafiki

Grafikten görülmek üzere seçilen örnek açık metinin şifrelenmesi için demo programı toplam olarak 20, 987, 628 byte miktarda bellek kullanmıştır. Şekil 6.10' da bellek kullanımı %'leri gösterilmektedir.



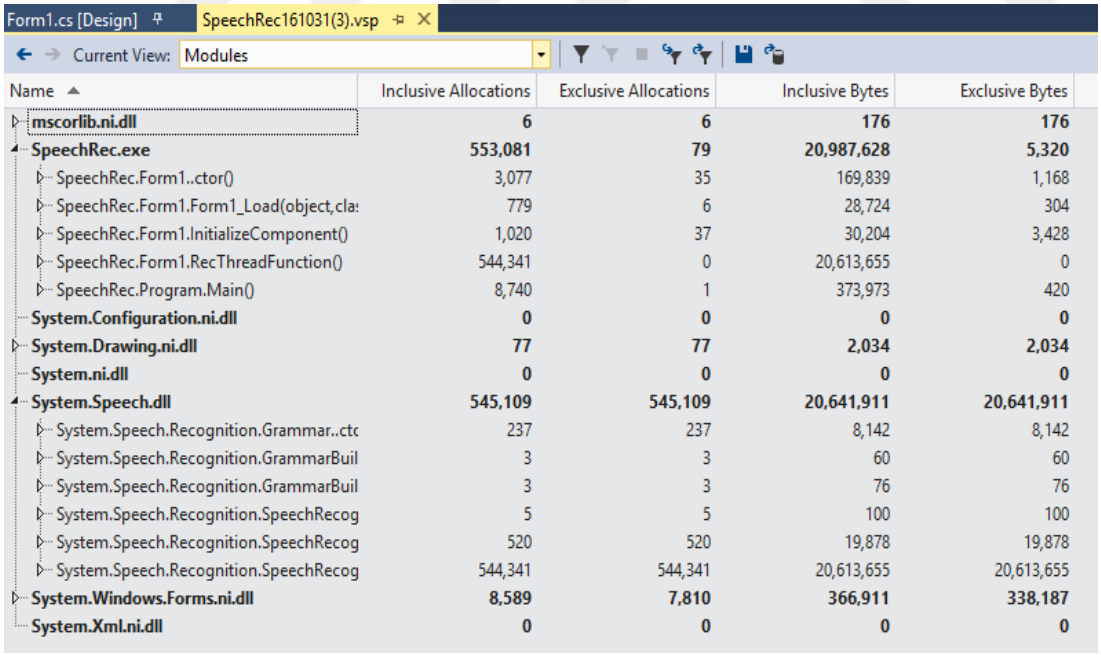
Şekil 6.10 Bellek kullanım %'leri yüksek olan fonksiyonlar

Programın CPU kullanımı ve bellek kullanımı incelendiğinde CPU kullanımında sistem üzerinde darboğaz oluşturan fonksiyonların Şekil 6. 10'dan görüldüğü gibi bellek kullanımınında en yüksek olduğu tespit edilmiştir. Satırlara tek tek tıklandığında bellek kullanımını arttıran fonksiyonların detaylı bilgilerine ulaşılmaktadır.

Bellek kullanımı en yüksek olan veri tipleri Şekil 6. 11' de gösterilmektedir.

Şekil 6.11 Bellek kullanımı en yüksek olan veri tipleri

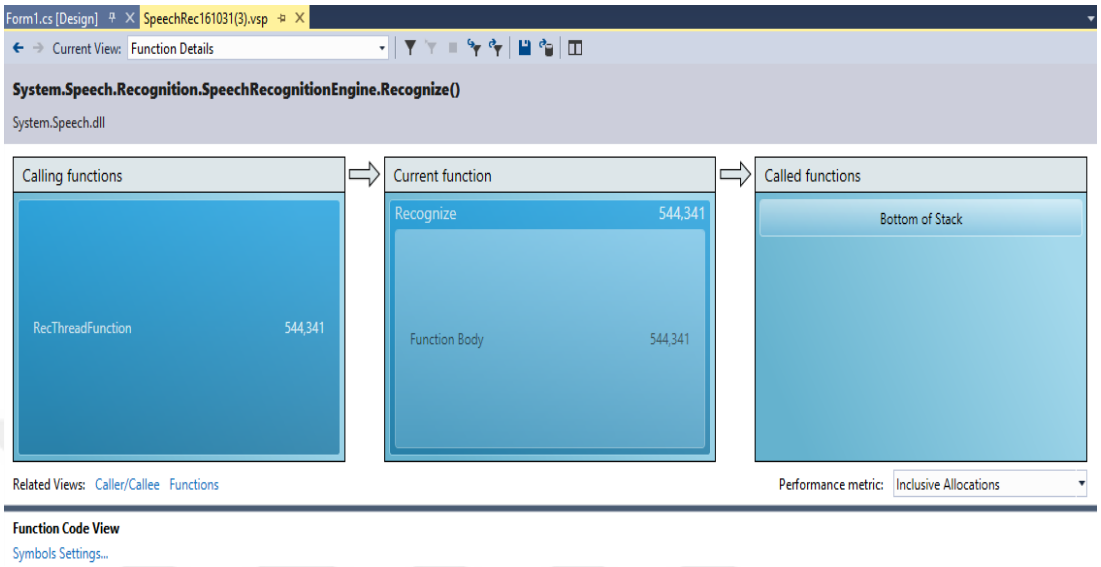
Current View bölümünden *Modules* seçeneği seçildiğinde zaman bellek kullanımına ait daha detaylı bilgilere ulaşılmaktadır. Şekil 6.12'de *Modules* seçeneği ile bellek kullanımının daha detaylı bilgileri görülmektedir.



Şekil 6.12 Fonksiyon ve sınıfların bellek kullanımına ait detaylar

Şekil 6.12'de görülmek üzere SpeechRec.exe 5.320 byte bellek kaplayarken System.Speech.dll 20,641,911 byte bellek kaplamaktadır. Current View kutusundan Allocation veya Function seçeneği seçildiğinde tüm fonksiyonların daha detaylı

bilgilerine ulaşılabilir. Şekil 6. 13'te *Function Details* seçeneği ve Performance metrics bölümünden *Inclusive Allocation* seçildiğinde fonksionların bellek kullanım detayları görülmektedir.



Şekil 6.13 Function Details seçeneğine ait bellek kullanım detayları

7 SONUÇ

Tez çalışmasında daha öncelerde kullanılan Tek Kullanımlık Anahtarla şifreleme teknikleri ve modern teknikler incelenerek, onlarla paralel özelliklere sahip modern tekniklere dayanan yeni bir Tek Kullanımlık Anahtar üretici geliştirilmiş ve şifreleme için kullanılmıştır. Geliştirilen algoritmanın performans değerlendirilmesi yapılmış, incelenmiş ve sunulmuştur.

Geliştirilen algoritmada amaç Tek Kullanımlık Anahtarla şifreleme yöntemleri için önemli koşul olan anahtarın rastgele olması şartının sağlanabilmesi ve rastgele gözüken (sözde ratgele-pseudo random) Tek Kullanımlık Anahtar üretebilmektir. Bu yöntemlerle şifreleme yapan algoritmaların güvenliğinin, anahtarın tek kullanımlık ve tamamen rastgele olmasına bağlı olması tez kapsamında geliştirilen algoritmanın geliştirilme sürecinde göz önüne alınmıştır. Geliştirilen anahtar üretici algoritmanın modern tekniklere uyum sağlayabilmesi için ses algılama ve ses tanıma yöntemi kullanılmıştır. Geliştirilen algoritma için C# dili kullanarak demo programı hazırlanmış ve windows form aracılığıyla arayüzü yapılmıştır. Demo programında ses algılama işleminin yapılması için C# dili üzerinden gereken nesne, bileşen ve kütüphaneler, donanımsal olarak ise Windows 8.1 Pro işlemciye sahip hp ProBook 4530s dizüstü bilgisayar ve bir adet mikrofon kullanılmıştır. Ses algılama aşamasında mikrofona söylenilen konuşma sesleri mikrofonun bi kaç saniyede ala bildiği dış seslerin içerisinden filtirelenmiş ve bire bir karşılığı alınarak Tek Kullanımlık Anahtar üretimi için kullanılmıştır. Bu aşamada mikrofona söylenilen sesler veya metin karşılıkları programın hiçbir kısmında önceden yer almamakta ve söylenilen sesler tamamen rastgele olmaktadır. C# dili üzerinde ses ve konuşma tanıtma için İngiliz dili paketi kurulu olduğu için ses tenıtma ingiliz dili üzerinden yapılmaktadır. Tek Kullanımlık Anahtarın üretilmesi için program tarafından sesin bire bir metin karşılığı alınmakta ve bir anahtar kelime seçilerek bu metinle XOR işlemine sokulmaktadır. XOR işlemi sonucunda oluşan bit dizisi Tek Kullanımlık Anahtar olmaktadır.

Tez kapsamında ses tanıtma yöntemi kullanarak geliştirilen algoritma için yapılan demo programı hiçbir donanımsal boyuta taşınmamıştır. Ancak bu algoritmanın dahada geliştirilerek donanımsal boyuta da taşınılabilinmesi düşünülmektedir. Bu sebepten ses tanıtma yöntemiyle geliştirilen bu algoritma her zaman geliştirilmeye açıktır.

Geliştirilen algoritma için hazırlanmiş demo programı aracılığıyla üretilen Tek Kullanımlık Anahtar, XOR operatoru kullanarak şifreleme ve deşifreleme işlemleri için kullanılmıştır. Üretilen rastgele anahtarla sadece XOR operatoru kullanarak değil, başka yöntemler kullanarak da şifreleme işlemleri yapıla bilir. Ancak algoritmada amaç rastgele gözüken (sözde rastgele-pseudo random) anahtar üretimi olduğu için algoritmanın şifreleme kısmında sadece XOR operatoru kullanılmıştır. Demo programının CPU kullanımı, Bellek kullanımının performans analizi incelenmiş ve detayları tez çalışmasında belirtilmiştir.

KAYNAKLAR

Akay, İ. G. (2014). "Bilgi Güvenliği Yönetim Sistemleri:Bilgi Güvenliği Uygulama Mülakatları". Bilecik: Bilecik Şeyh Edebali Üniversitesi Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, s: (10-25).

Aksu, P. K. (2014). " Hastane Bilgi Yönetim Sisteminin Bilgi Güvenliği Açısından Değerlendirilmesi ". İstanbul: Marmara Üniversitesi Sağlık Bilimler Enstitüsü Doktora Tezi, s:(40-41).

Altun, R. (2014). "Belirli Kısıtrlara Göre Bilgi Güvenliği İhlallerinin Tespiti". İstanbul: Beykent Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s: (1-14).

Aslandağ, K. (2010). "Bilgi Güvenliği Kavramı Ve Bilgi Güvenliği Yönetim Sistemleri İle Şirket Performansı İlişkisine Dair Bir Uygulama". Gebze: Gebze Yüksek Teknolojiler Enstitüsü Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, s: (17-20).

Bağcıoğlu, E. Ö. (2007). Tek Kullanımlık Şerit. ODTÜ bilgisayar topluluğu elektronik dergi .

Başar, M. S. (2004). "Yer Değiştirme Esaslı ve Rastgele Anahtarlı Yeni Bir Şifreleme Algoritması". Erzurum: Atatürk Üniversitesi Doktora Tezi, s: (1-17).

Bayar, E. (2012). "Modern Kriptosistemlerle Şifrelemenin Modellenmesi İle Veri Güvenliğinin Sağlanmasi". İstanbul: Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s:(1-44).

Çimen, C., Akleylek, S., & Akyıldız, E. (2007). "Şifrelerin Matematiği: Kriptografi" (15-66). Ankara: ODTÜ Yayıncılık.

Daemen, J., & Rijmen, V. (2000). "The Block Cipher Rijndael, Smart Card Research and Applications" (288-296). LNCS 1820, Springer-Verlag.

Dalkıç, G., & Akın , O. (2005). Anahtar Tabanlı Gelişmiş Rotor Makinesi". Gaziantep: Akademik Bilişim Konferansı.

Denning, D. E. (1982). Cryptography and Data Security, Purdue University. Boston: Addison-Wesley Longman Publishing .

Frank, M. (1882). Telegraphic code to insure privacy and secrecy in the transmission of telegrams. C.M. Cornwell.

Gamal, T. E. (1988). "A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. Advances in Cryptology: Proceedings of CRYPTO 84". Springer Verlag, pp: (10-18).

Güncan, M. (2002). "Kimlik Tabanlı Kriptosistemler ile Güvenli Veri Aktarımı". İstanbul: İTÜ Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.

Haklı, T. (2012). "Bilgi Güvenliği Standartları Ve Kamu Kurumları Bilgi Güvenliği İçin Bir Model Önerisi". Isparta: Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s: (5).

Kahn. (1967). The Codebreakers. ISBN 0-684-83130-9.

Keliher, L. (2003). Linear Cryptanalysis of Substitution-Permutation Networks. PHd Theis.

Kodaz, H., & Botsalı, F. (2010). "Simetrik ve Asimetrik Şifreleme Algoritmalarının Karşılaştırılması". Selcuk Üniversitesi Teknik Bilimler Meslek Yüksekokulu Teknik-Online Dergi, 9(1), s:(12-20).

Konheim, G. A. (1981). Cryptography: A Primer. New York: Wiley.

MEB. (2013, 12 06). "Bilgi Felsefesi".

Menezes, A., Oorschot, P., & Vanstone, S. (1996, 11 12). Handbook of Applied Cryptography. CRC Press, ss: 31-32.

Montgomery, P. (1985). "Modular Multiplication Without Trial Division". Mathematics of Computation, pp: (44).

Muharremoğlu, G. (2013). "Kurumsal Bilgi Güvenliğinde Zafiyet, Saldırı Ve Savunma Öğelerinin İncelenmesi". İstanbul: İstanbul Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s: (6-17).

Nicholas, G. M. (2015). " PAST, PRESENT, AND FUTURE METHODS OF CRYPTOGRAPHY AND DATA ENCRYPTION". Department of Electrical and Computer Engineering University of Utah.

Otgonjargal, G. (2013). " Bilgi Güvenliği Yönetim Sistemi ISO/IEC 27001 Ve Bilgi Güvenliği Risk Yönetimi ISO/IEC 27005 Standartlarının Uygulanması". İzmir: Ege Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s: (2-5).

OTP. (2014, 03 17). Crypto Museum, One-Time Pad. 09 14, 2016 tarihinde Cryptomuseum.com. adresinden alındı

Poşul, A. (2014). bilgi-guvenligi-standartlari.html. 06 27, 2016 tarihinde Ulusal Bilgi Güvenliği Kapısı: http://bilgiguvenligi.gov.tr/bt-guv.-standartlari/ adresinden alındı

Rogaway, P., & Coppersmith, D. (1994). A software-optimized encryption algorithm. In Ross Anderson, editor, Fast Software Encryption, Springer-Verlag, pp: (56-63).

Sakallı, M. T. (2006). "Modern şifreleme yöntemlerinin gücünün incelenmesi". Trakya: Trakya Üniversitesi Fen Bilimleri Enstitüsü Doktora Tezi, s: (3-7, 49-63).

Şen, Ş. (2006). "İndirgenmiş SPN (Substitution Permutation Network) Algoritması İçin Lineer Kriptoanaliz Uygulaması". Edirne: Trakya Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.

Şenay, V. (2012). "Kuantium Kriptografi". Eskişehir: Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s: (14-17).

Simon, S. (2001). The Code Book. Shinchosha.

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

Sönmez, R. (2002). "Veri Sifreleme Standardı (DES) ve Rivest Shamir Adleman (RSA) Güvenlik Algoritmalarının VLSI Tasarımı". Ankara: Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi.

Soyalıç, S. (2005). "Kriptografik Hash Fonksiyonları ve Uygulamaları". Kayseri: Erciyes Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s:(70-72).

Sulak, F., Turan, M., & Demiröz, B. (2013). Kriptoloji. Ankara: Atılım Üniversitesi Fen Edebiyat Fakültesi.

TDK. (2016, 06 17). T.C. Başbakanlık Atatürk Kültür, Dil ve Tarih Yüksek Kurumu. Türk Dil Kurumu:

http://www.tdk.gov.tr/index.php?option=com_gts&arama=gts&guid=TDK.GTS.576374b7079db6.78620412 adresinden alınmıştır

Tefon, M. (2013). "Elektronik Haberleşme Hizmeti İçinde Güvenli Ses/Veri Haberleşmesi Açısından Kriptolu Haberleşmenin İncelenmesi, Düzenlemeler, Öneriler ve Türkiye Analizi". Ankara: Bilgi Teknolojileri ve İletişim Kurumu Teknik Uzmanlık Tezi, s: (17-40).

Topal, H. (2004). "Siber Terör". İstanbul: İstanbul Üniversitesi Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, s:(15).

Tuncal, T. (2008). "Bilgisayar Güvenliği Üzerine Bir Araştırma ve Şifreleme-Deşifreleme Üzerine Uygulama". İstanbul: Maltepe Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s:(1-6, 15-42).

Ülker, Ü., & Coşkun, A. (2014). "Ulusal Bilgi Güvenliğine Yönelik Bir Kriptografi Algoritması Geliştirilmesi ve Harf Frekans Analizine Karşı Güvenirlik Tespiti". Ankara: Gazi Üniversitesi Bilişim Enstitüsü Yüksek Lisans Tezi.

Ülkü, Ü. (2014). Klasik Teknikler Kullanılarak Bir Kriptografi Algoritması Geliştirilmesi Ve DES Algoritması İle Performans Analizlerinin Karşılaştırılması. Ankara: Gazi Üniversitesi Bilişim Enstitüsü Yüksek Lisans Tezi, s:(15-60).

Vural, Y. (2007). "Kurumsal Bilgi Güvenliği Ve Sızma (Penetrasyon) Testleri". Ankara: Gazi Üniversiteti Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s:(17-35).

Yerlikaya, T. (2006). "Yeni Şifreleme Algoritmalarının Analizi". Edirne: Trakya Üniversitesi Fen Bilimleri Enstitüsü Doktora Tezi, s: (57-75).

Yıldız, B. (2007). "Bilgi Güvenliği Ve E-Devlet Kapsamında Kamu Kurumlarında Bilgi Güvenliği Yönetimi Standartlarının Uygulanması". Gebze: Gebze Yüksek Teknolojiler Enstitüsü Sosyal Bilimler Enstitüsü Yüksek Lisans Tezi, s:(25-26).

Yılmaz, R. (2010). "Kriptolojik Uygulamalarda Bazı İstatistik Testler". Konya: Selçuk Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, s: (1-30).

EKLER

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Speech.Recognition;
using System.Threading;

namespace SpeechRec
{
   public partial class Form1 : Form
   {

      public SpeechRecognitionEngine recognizer;
      public Grammar grammar;
      public Thread RecThread;
      public Boolean RecognizerState = true;
      public Form1()
      {
         InitializeComponent();
      }

      private void Form1_Load(object sender, EventArgs e)
      {
         GrammarBuilder build = new GrammarBuilder();
         build.AppendDictation();
         grammar = new Grammar(build);

         recognizer = new SpeechRecognitionEngine();
         recognizer.LoadGrammar(grammar);
         recognizer.SpeechRecognized += new
EventHandler<SpeechRecognizedEventArgs>(recognizer_SpeechRecognized);
         RecognizerState = true;
         RecThread = new Thread(new ThreadStart(RecThreadFunction));
         RecThread.Start();
      }
      public void recognizer_SpeechRecognized(object sender,
SpeechRecognizedEventArgs e)
      {
```

```
    if (!RecognizerState)
        return;

    this.Invoke((MethodInvoker)delegate
    {
        txb_sesTex.Text += (" " + e.Result.Text.ToLower());

    });

}
public void RecThreadFunction()
{
    while (true)
    {
        try
        {
            recognizer.Recognize();
        }
        catch
        {

        }

    }
}

private void button1_Click(object sender, EventArgs e)
{
    btn_sifrele.BackColor = Color.CornflowerBlue;
    button2.BackColor = Color.CornflowerBlue;
    button1.BackColor = Color.Red;
    button3.BackColor = Color.CornflowerBlue;
    button4.BackColor = Color.CornflowerBlue;
    RecognizerState = true;
    txb_anahtar.Clear();
    txb_sesTex.Clear();
    txb_OneTimePad.Clear();
    txb_SifreMetin.Clear();
    txb_decription.Clear();
    decription.Clear();
    SifreMetin.Clear();
}

private void button2_Click(object sender, EventArgs e)
{
    btn_sifrele.BackColor = Color.CornflowerBlue;
    button2.BackColor = Color.Red;
    button1.BackColor = Color.CornflowerBlue;
    button3.BackColor = Color.CornflowerBlue;
    button4.BackColor = Color.CornflowerBlue;
```

```
    RecognizerState = false;
}

private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    RecThread.Abort();
    RecThread = null;
    recognizer.UnloadAllGrammars();
    recognizer.Dispose();
    grammar = null;

}

List<char> key = new List<char>();
private void btn_AnahtarUret(object sender, EventArgs e)
{
    btn_sifrele.BackColor = Color.CornflowerBlue;
    button2.BackColor = Color.CornflowerBlue;
    button1.BackColor = Color.CornflowerBlue;
    button3.BackColor = Color.Red;
    button4.BackColor = Color.CornflowerBlue;

    int m = txb_sesTex.Text.Count();

    if(m%2>0){ m = m + 1;}

    string ifade = txb_sesTex.Text.Substring(0, m / 2);
    string ifade1 = txb_sesTex.Text.Substring(m/2,m / 2);

    int j=0;
    List<char> ses = new List<char>();
    foreach (var item in ifade)
    {
        int b  = (int)item ^ (int)ifade1[j];
        ses.Add((char)b);
    }

    if (!( m/2 < txb_anahtar.Text.Length))
    {

        int i = 0;
        foreach (var c in ses)
        {
            int a = (int)c ^ (int)txb_anahtar.Text[i];
            key.Add((char)a);
        }
        txb_OneTimePad.Text = new string(key.ToArray());

    }
    else
```

```
        {
            MessageBox.Show("Anahtar kelime boyutu tanımlanan sesin yarı
boyutundan büyük olamaz!");
        }


    }
    List<char> SifreMetin = new List<char>();
    private void btn_sifrele_Click(object sender, EventArgs e)
    {
        btn_sifrele.BackColor = Color.Red;
        button2.BackColor = Color.CornflowerBlue;
        button1.BackColor = Color.CornflowerBlue;
        button3.BackColor = Color.CornflowerBlue;
        button4.BackColor = Color.CornflowerBlue;
        if (!(key.ToString().Length < txb_acikMetin.Text.Length))
        {

            int i = 0;
            foreach (var b in txb_acikMetin.Text)
            {
                int a = (int)b ^ (int)key[i];
                SifreMetin.Add((char)a);
            }
            txb_SifreMetin.Text = new string(SifreMetin.ToArray());

        }
        else
        {
            MessageBox.Show("Anahtar boyutu  metin boyutundan küçük olamaz!");
        }
    }
    List<char> decription = new List<char>();
    private void button4_Click(object sender, EventArgs e)
    {
        btn_sifrele.BackColor = Color.CornflowerBlue;
        button2.BackColor = Color.CornflowerBlue;
        button1.BackColor = Color.CornflowerBlue;
        button3.BackColor = Color.CornflowerBlue;
        button4.BackColor = Color.Red;

        if (!(key.ToString().Length < SifreMetin.ToString().Length))
        {
            int i = 0;
            foreach (var c in SifreMetin)
            {
                int a = ((int)c ^ (int)key[i]);
                decription.Add((char)a);
            }
            txb_decription.Text = new string(decription.ToArray()).ToString();
```

```
            }
            else
            {
                MessageBox.Show("Anahtar boyutu metin boyutundan küçük olamaz!");
            }

        }
    }
}
```

ÖZGEÇMİŞ

Ad-Soyad : Jabrayil HASANOV
E-Posta : chesenov1989@gmail.com

KİŞİSEL BİLGİLER

Doğum Tarihi ve Yeri : 17/05/1989/ Azerbaycan, Celilabad
Medeni Durum : Bekar
Askerlik Durumu : Tamamlandı (2007-2009)

EĞİTİM BİLGİLERİ

Lisans : Azerbaycan Devlet Pedagoji Üniversitesi / Matematik ve Bilişim
Yüksek Lisans : İstanbul Aydın Üniversitesi / Bilgisayar Mühendisliği

BİLGİSAYAR BİLGİLERİ

C#
SQL
Html-CSS
Javascript
jQuery