T.C.

SAKARYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

SES VERİLERİNE SIKIŞTIRILMIŞ VE ŞİFRELENMİŞ HAM VERİLERİN GÖMÜLMESİ

YÜKSEK LİSANS TEZİ

Tuncay AKBAL

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Tez Danışmanı : Yrd. Doç. Dr. A.Turan ÖZCERİT

Haziran 2008

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SES VERİLERİNE SIKIŞTIRILMIŞ VE ŞİFRELENMİŞ HAM VERİLERİN GÖMÜLMESİ

YÜKSEK LİSANS TEZİ

Tuncay AKBAL

Enstitü Anabilim Dalı : Elektronik ve Bilgisayar Eğitimi

Bu tez 02 / 06 /2008 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Yrd.Doç.Dr. A. Turan ÖZCERİT	Doç.Dr. İsmail ERTÜRK	Yrd.Doç.Dr. Ahmet ZENGİN
Jüri Başkanı	Üye	Üye

TEŞEKKÜR

Kablosuz haberleşmede veri gizliliğinin ön plana çıktığı günümüzde, veri gizleme teknikleri bilgi güvenliği konusunda sunduğu yaklaşımlarla önemini giderek arttırmaktadır. “Sırörtme” (Steganography) yöntemleri yazılımlarla desteklenmekte ve veri gizliliği için güçlü yazılımlar oluşturulmaktadır. Bu noktadan hareketle, gerçek zamanlı ses haberleşmesinde, gömülü gizli veri ya da dosyaların sıkıştırılıp ardından şifrelenerek gönderilmesi yönünde tez çalışmaları yapılmış olup, geliştirilen yazılımlar sunulmaktadır.

Yüksek lisans eğitimim süresince değerli birikimlerini bana aktaran, tezimin başlangıcından bitimine kadar her aşamasında sorunlarımı dinleyen ve çalışmalarıma yön veren ve değerli zamanını sorunlarımın çözümüne ayıran tez danışmanım Sayın Yrd. Doç. Dr. A. Turan ÖZCERİT’e, tez ile ilgili araştırmaların yapılmasından, uygulamaların ve tezin yazılmasına kadar yardımlarını ve birikimlerini benimle paylaşan değerli arkadaşlarım Cemil ASLAN, Yıldıray YALMAN ve onun değerli eşi Neslihan YALMAN’a teşekkürlerimi sunarım.

Bugünlere gelmemi sağlayan annem Şahizer ve babam Şahabettin AKBAL’a, ben okurken maddi ve manevi desteklerini esirgemeyen kardeşlerim Funda, Çiğdem, Demet ve Kübra’ya ve üzerimde emeği olan herkese teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR LİSTESİ

C : Mesaj iletim süresi (s)

D : Mesajın varma sınır değeri (s)

d : Mesajın yük büyüklüğü (bayt)

R : En kötü durum gecikme süresi (s)

T : Mesajın üretim aralık zamanı (s)

t : Mesajın kuyruğa atılmasından veri yolu erişimini kazanmasına kadar geçen süre (s)

k_{m_i} : i. ses çerçevesine eklenecek modüle gizli anahtar işareti

k : Gizli anahtar işareti

w_j : j. gömülecek veri (damga) bilgisi

L : Gömülen verinin uzunluğu

f() : Doğrusal olmayan veri gömme fonksiyonu

s_i : Orijinal i. ses çerçevesi

$s_{i_{WM}}$: Veri gömülü olan i. ses çerçevesi

g() : Doğrusal olmayan gömülü veriyi çözme algoritması

$\hat{w}_j$: Öngörü ile çözülen j. gömülecek veri biti

a : Kuantalama aralığı

Q : Kuanta seviyesi sayısı

n : İşaretin kodlandığı bit sayısı

x(t) : Mesaj işareti

$x_q(t)$: Kuantalanmış örnek işareti

A_{max} : Maksimum genlik

A_{min} : Minumum genlik

f_N : Örnekleme frekansı

ACK : Acknowledgement (Alındı Bilgisi)

AP : Access Point (Erişim Noktası)

ARPANet : Advanced Research Project Agency Network
ASCII : American Standard Code for Information Interchange
BSS : Basic Service Set (Temel Servis Seti)
CAN : Controller Area Network
CCK : Complementary Code Keying
CRC : Cyclic Redundancy Check
CSMA/CD : Carrier Sense Multiple Access with Collision Detection
CTS : Clear to Send (Göndermeye Açık)
DAVIC : The Digital Audio Visual Council
DCF : Distrubuted Coordination Function
DIFS : Distrubuted Coordination Function Inter Frame Space
DQPSK : Differantial Quadrature Phase Shift Keying
DS : Digital Signal (Sayısal Sinyal)
DSSS : Direct Sequence Spread Spectrum
DVD : Digital Versatile Disc
EBCDIC : Extended Binary Coded Decimal Interchange Code
EIFS : Extended Inter Frame Space
FHSS : Frequency Hopping Spread Spectrum
HiperLAN : High Performance Radio Local Area Network
IAPP : Inter Access Point Protocol
IBM : International Business Machines Company
IFS : Inter Frame Space (Çerçeveler arası boşluk)
ISO : International Standards Organization
ISM : Industries, Scientific, Medical
ITU : International Telecommunications Union
KLAN : Kablosuz Yerel Alan Ağları
LAN : Local Area Network (Yerel Alan Ağı)
LSB : Least Significant Bit (En Az Değerlikli Bit)
MAC : Medium Access Protocol (Ortam Erişim Kontrolü)
MPDU : MAC Protocol Data Unit
MPEG : Moving Picture Experts Group
OFDM : Orthagonal Frequency Division Multiplexing
PAM : Pulse Amplitude Modulation (Darbe Genlik Modülasyonu)

PCF : Point Coordination Function

PCM : Pulse Code Modulation (Darbe Kod Modülasyonu)

PLCP : Physical Layer Convergence Procedure

PMD : Physical Medium Dependent

RF : Radyo Frekansı

RGB : Red Green Blue (Kırmızı Yeşil Mavi)

RIFF : Resource Interface File Format

RTS : Request to Sent (Gönderme İstemi)

SDMI : The Secure Digital Music Initiative

SIFS : Short Inter Frame Space

TCP/IP : Transfer Control Protocol / Internet Protocol

U-NII : Unlicensed National Information Infrastructure

WIPO : World Intellectual Property Organization

WECA : Wireless Ethernet Company Alliance

WLAN : Wireless Local Area Network

SSGM : Sayısal Ses İçerisinde Gizli Metin

SSGD : Sayısal Ses İçerisinde Gizli Dosya

Gömü Verisi (Dosyası) : Gönderilmek istenen gizli veri/dosya

Örtü Verisi (Dosyası) : Gömü verisinin/dosyasının gömüleceği taşıyıcı veri/dosya

Örtülü Veri (Dosya) : İçerisinde gömü verisi/dosyası bulunan örtü verisi/dosyası

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

ÖZET

Anahtar kelimeler: Sayısal Ses, Veri Gizleme, Sırörtme, Şifreleme, Sıkıştırma

Günümüzde veri gizleme (steganografi), Veri Şifreleme (Cryptography) ve Veri Sıkıştırma (Data Compression) teknikleri özellikle kablosuz iletişim sistemleri içerisinde giderek artan bir öneme sahip olmaktadır. Çoklu ortam ve bilgi güvenliği uygulamaları gibi güncel gereksinimler ile veri gizleme üzerine yapılan çalışmalar da yoğun bir talep ve ilgi görmektedir.

Bu tezde sunulan projenin temel amacı; sayısal ses içerisinde gönderilecek verinin daha güvenli ve hızlı iletimi için şifreleme ve sıkıştırma yöntemleri kullanarak gizli veri transferinin kablosuz ortamda gerçekleştirilmesidir. Bu nedenle kablosuz ortamda transfer edilen sayısal ses içerisine veri/dosya gömme uygulaması gerçekleştirilmiştir. Aynı zamanda gerçek zamanlı ses haberleşmesi yapmaktadır.

Tez çalışmalarında donanım aracı olarak kablosuz haberleşme yapabilen iki adet değişik özelliklere sahip bilgisayar ve bir adet Erişim Noktası (Access Point), yazılım aracı olarak ise Borland Delphi 7.0 programlama dili kullanılmaktadır.

Örnek çalışmaların sonucunda, elde edilen sonuçlar sunularak başarım değerlendirmeleri yapılmaktadır.

EMBEDDING COMPRESSED AND ENCRYPTED RAW DATA INTO AUDIO FILES

SUMMARY

Key Words: Supplier : Digital Voice, Data Hiding, Steganography, Data Compression, Cryptology

Techniques for information hiding (Steganography), Cryptography, Data Compression have nowadays become increasingly more sophisticated and widespread. Researches on information embedding, have received considerable attention for a decade due to its potential applications in multimedia and information security.

The main objective of this research presented is to design and implement encrypted and compressed hidden data transfer within digital voice for wireless communications. For this reason, the application is used for file embedding within digital voice. Furthermore, this application enables a conventional wireless realtime voice communication.

In this research, two different computers and an access point are utilized. These computers are equiped with wireless communication tools and software components. The application software is developed with Borland Delphi 7.0 programming language.

Examples of application results of the softwares have been presented and their performances have evaluated.

BÖLÜM 1. GİRİŞ

Temeli antik çağlara kadar dayanan gizli haberleşme, teknoloji değişip geliştikçe şekil ve yöntem açısından da faklılıklar göstermektedir. Bununla birlikte önemini devamlı korumaktadır. Gizliliğin öneminin had safhaya ulaştığı uygulamalarda; gizli bilgilerin, üçüncü kişilerin eline geçmeden ilgili hedefe gönderilmesi amaçlanır.

Veri gizleme aynı zamanda "steganografi/sırörtme" (steganography) adını almaktadır ve şifreleme ile yakından ilişkilidir. Şifrelemenin amacı, mesajları anlaşılmaz hale getirerek gizli anahtara sahip olmayan yetkisiz kişilerin mesajı yeniden elde ederek orijinal haline getirmesini önlemektir. Bazen şifreli mesajları değiştirmek yerine, haberleşmenin maskelenmesi yoluyla güvenlik ve gizliliğin elde edilmesi durumu arzu edilebilir. Bu problem sırörtme tekniğini ön plana çıkarmaktadır. Tarihte ilk sırörtme teknikleri özel mürekkep veya kimyasal maddeler kullanarak görünmeyen yazılar elde etmeyi içermektedir. O dönemde metin içinde gizli mesajlar oldukça yaygındır. Kelime veya cümlelerin ilk harflerini referans alarak, bazı masum görünümlü kelimelerle gizli bir mesaj iletilmekteydi. Günümüzde veriyi gizleme amacıyla, değişik taşıyıcıları belirli oranda kullanmak doğal görünümü değiştirmemektedir. Sayısal resimler, videolar ve ses işaretleri bu amaç için idealdir.

Günümüzdeki yaygın şekliyle sayısal veri gizleme ile ilgili en önemli çalışma ilk olarak 1954 yılında; Emil Hembrooke'un sahip olduğu Muzak şirketinin, müzik kayıtlarına sahiplik bilgisini içeren kod yerleştirmek için almış olduğu patenttir [1,2].

80'li yıllarda İngiltere Başbakanı Margaret Thatcher'in kabinesinde kabine içi bilgileri sızdıran bakanın kim olduğunun tespit edilmesi amacı ile kelime işlem programı her bakan için ayrı ayrı tanımlayıcı bilgi ekleyecek şekilde programlanmış ve sorumlu bakanın kim olduğu ortaya çıkarılmıştır [3].

En basit ve en yaygın sırörtme tekniği “En düşük değerlikli bit”e (Least Significant Bit embedding; LSB) gömme tekniğidir. Burada önerilen işlem genellikle sayısal resimler veya ses dosyaları içerisindeki belirli bit bloklarının en düşük değerlikli bitini, gürültü (orijinal dosya açısından gürültü şeklinde beliren bu durum aslında gizli veriyi ifade eder) tarafından maskelenerek değiştirilmesidir. Aslında renkli resim kullanımında, mesaj gizleme için daha fazla oda/piksel mevcuttur; çünkü, her bir piksel kırmızı, yeşil ve maviden oluşan üçlü bir birleşimdir. Yine iki veya daha fazla “en düşük değerlikli bit” yer değiştirilerek her bir pikselin veri gizleme kapasitesi artırılır; ancak, aynı zamanda istatistiksel olarak çözünebilirlik riski de arttırılmış olur. Böylece her bir özel Sırörtme tekniğinin güvenli çalışması önemlidir ve neden güvenli olduğu tartışılır. Resim veya ses gürültüsüne bağlı değişiklikler ile karmaşık şüpheler oluşturularak, resmin veya elde bulunan sayısal ses bilgisinin herhangi bir istatistiksel model ile kolayca anlaşılamaması başarılı bir şekilde sağlanır. Bununla beraber kullanılan sırörtme yönteminin ve uygulama şeklinin üçüncü kişiler tarafından bilinmesi veri güvenliğini ortadan kaldıracaktır. Sırörtme veri güvenliğini tek başına sağlamakta yeterli olamaması tekrar şifrelemeyi gündeme getirmektedir.

Tarihin bilinen ilk şifreleme yöntemi yer değiştirme ve harf değiştirme yöntemidir. Bu yöntemlerden ilki bir yazıdaki harflerin yerlerini değiştirerek, ikincisi ise harfleri başka harflerle değiştirerek gerçekleştirilir. Bu şifrelemeyi kullanan belki de en ünlü teknik Sezar Şifresi' dir: bu şifrede, her harf o harften birkaç sonraki harf kullanılarak yazılır. Örneğin, üç harf atlamalı Sezar Şifresi'nde "deneme" yerine "gğrğpğ" yazılır.

Şifreleme, özellikle herkese açık ortamlarda veri güvenliği için daha önemli olmaktadır. Kablosuz haberleşme ortamlarında yapılan yayın niteliğinden ötürü birçok kişi bu bilgilere ulaşabilir. Yapılmak istenen haberleşmenin güvenliğini arttırmak için ilgili veriye sırörtne ve şifeleme yapılması faydalı olacaktır.

Kablosuz ortamın sınırlı bant genişliği ve yayın niteliğinden ötürü, gönderilmeden önce görüntüye sıkıştırma ve şifreleme yapılması gerekmektedir. Diğer taraftan, kablosuz cihazlardaki sınırlı enerjiyi etkin kullanmak önemlidir [4].

1587 yılında İngiliz Kraliçesini devirmek için adamlarıyla haberleşmede kullandığı basit değiştirme yöntemi çözülen İskoçya Kraliçesi, bu hatasını idam edilerek ödemiştir [5].

Günümüzde özellikle gerçek zamanlı uygulamalarda veri iletim zamanı önem kazanmaktadır. Verinin hızlı biçimde iletilmesi için gönderilecek verinin gönderilmeden önce sıkıştırılması gönderme zamanını azaltacaktır. Böylelikle sırörtme ve şifreleme işlemlerinden dolayı kaybedilen zaman "Kayıpsız Veri Sıkıştırma Algoritmaları" kullanılarak azaltılır. Özellikle taşınır haberleşme cihazlarından veri gönderimi ve veri alımı için harcanacak enerjiyi azaltarak, bataryanın daha uzun zaman kullanılmasını sağlayacaktır.

Bu tez çalışmasında, iki bilgisayar arasında kablosuz ve gerçek zamanlı olarak ses iletişimi yapılırken, sayısal ses verilerine kullanıcının istediği bilgilerin gömülerek gönderilmesi sağlanmaktadır. Gönderme işleminden önce veri güvenliğini arttırmak için şifreleme yöntemi OTP (One Time Pad) ve veri gönderme zamanını azaltmak için kayıpsız veri sıkıştırma algoritması olan Zlib kullanılmıştır. Bu işlemin sonucunda ses verilerini alan bilgisayarda örtülü veri içerisinden gömü verisinin (dosyasının) ayırt edilmesi, şifrelenen verinin çözülmesi ve sıkıştırılan verinin açılma uygulaması da yapılmaktadır.

1.1. Literatürde Yapılan Çalışmaların Özetleri

Muzak şirketinin, 1954 yılında müzik kayıtları içerisine sahiplik bilgisini içeren kod yerleştirmek için almış olduğu patentle birlikte, telif haklarının korunmasına yönelik ses bilgileri içerisine veri gömme tekniği üzerine çalışmalar yoğunlaşmıştır. Bu durumun sadece kayıtlı ses verilerine değil, gerçek zamanlı ses verilerine de uygulanabileceği tartışılmaktadır. Örneğin hava trafik kontrolünde daha güvenli iletişim için ses bilgileri içerisine veri gömülmesinden bahsedilmekte ve buda Data in Voice (DiV) olarak adlandırılmaktadır [40].

1990'ların başında imge damgalama kavramı gelişmiş; Tanaka ve arkadaşları faks gibi ikili imgelerin korunması kavramını ortaya atmışlardır [35]. 1993 yılında Tirkel

ve arkadaşları gerçekleştirdikleri veri gömme tekniğine, daha sonra “watermark”(damgalama) olarak birleştirilecek olan “water mark” ismini vermişlerdir [6].

Özellikle müzik dosyaları için telif haklarının korunması amacıyla “Audio Watermarking” adı verilen çalışmalar genel olarak gömülü verilerin sezilemezliği üzerine yoğunlaşmıştır.

Dünya çapında telif haklarının korunması ve düzenlenmesi ile ilgili çalışmalar yapan ve hükümetler üstü bir kuruluş olan WIPO (World Intellectual Property Organization) sayısal veri gömme sistemlerinin yasal alanlarıyla ilgili çalışmalarını sürdürmektedir [7].

Sırörtmenin uygulandığı taşıyıcı verilerin, süzülerek içerisindeki gizli verinin elde edilmesi işlemi sıraçma olarak adlandırılır. Cheng ve arkadaşları elektronik metin resimleri için ilgili resim içerisinde veri gömülü olup olmadığını sezen bir sıraçma tekniği geliştirmişlerdir [8].

Sırörtme uygulamalarının yapılabilmesi için mutlaka taşıyıcı veri kullanılması gerekmektedir. Bunlar ses, resim, video vb. olabilir. Bunlardan birine örnek olarak, Adlı ve Nakao “.midi” uzantılı dosyalar için üç farklı sırörtme algoritması geliştirmişlerdir [9]. Xu ve arkadaşları da sıkıştırılmış video görüntülerine sırörtme algoritması önermişlerdir [10].

I.Dünya savaşında Almanların çözemeyeceği “bir kerelik bloknot” yöntemi Amerikan Telefon ve Telgraf şirketinin bir çalışan olan Gilbert Vernam tarafından geliştirilmiş ve savaş boyunca Amerika Birleşik Devletleri’nin mesaj güvenliğini sağlamıştır. Bu sistemde, şifrelenecek metin ASCII kodundaki karakterlere dönüştürülür ve bir kez şifreyi çözmede kullanılacak gizli anahtar, mesajı okuyan kişi tarafından imha edilirdi. Böylece tek seferlik mesajlaşmalar güvenli bir iletişimi oluştururdu [5].

Teknik Birlik dergisinin 1988 yılında yayınlanan 3. sayısında Türkiye Petrolleri Anonim Ortaklı'nın, Ankara'da kullanmakta olduğu VAX 11/780 minibilgisayarları ile Super VAX II mikro bilgisayarı ve Mikro VAX 2000 bilgisayarları arasında yerel alan ağı oluşturulmuş ve iletişim verilerini herhangi bir kişi tarafından okunabilmesi mümkün olan PTT hatları tarafından sağlandığı belirtilmiştir. Bu yayında gizlilik derecesi olan bilgilerin şifrelenmesi için Umumi Anahtar Şifrelemesinden faydalanılarak veri şifreleme programı hazırlanarak yerel alan ağı üzerinden şifreli veri iletimi yapılmıştır [11].

Kenneth Barr ve Krste Asanovi´c 2003 yılındaki yayınladıkları "Energy Aware Lossless Data Compression" adlı makalede kablosuz haberleşme yapılırken veri gönderilmeden önce sıkıştırma işleminin yapılmasının enerji boyutundan getireceği kazançları göstermişlerdir [12].

Tim Ho Tin Woo 2004 yılında "A Scalable, Secure, and Energy-Efficient Image Representation for Wireless Systems" isimli yüksek lisans tezinde, görüntü sıkıştırması yapmanın iletim için enerjiyi koruyacağını ve şifreleme yapmanın görüntüyü güvenli kılacağını göstermiş, sıkıştırma için enerji harcamasını azaltmak ve kullanıcıya farklı güvenlik düzeyleri vermek için kısımsal şifreleme kullanmak gerektiğini söylemiştir [4].

Christian Kratzer, Jana Dittmann, Thomas Vogel, Reyk Hillert 2008 yılında yayınladıkları makalede iki kullanıcı arasında sesli görüşme yapılırken sessizlik algılaması (Silence Detection), şifreleme ve sırörtme yöntemlerini kullanarak gizli bilgi gönderme işlemini gerçekleştirmişlerdir. Gönderilecek bilgiler, ses bilgilerinin son bitlerine(LSB) gömülmüştür [13].

LiWu Chang and Ira S. Moskowitz, yaptıkları "Critical Analysis of Security in Voice Hiding Techniques" adlı çalışmalarında ses içerisine veri gömme uygulaması yapmak için 4 farklı yöntem incelemişler ve bu yöntemler arasında birim zamanda gönderilecek veri miktarı bakımından en iyi yöntemin düşük değerlikli bitlere veri gömülmesi olduğunu göstermişlerdir.Ancak düşük değerlikli bitlere veri gömmenin kısmi güvenlik sağlayacağını söylemişlerdir [14].

Yalman, 2007 yılında “Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi” isimli yükek lisans tezinde sırörtme yöntemi kullanarak iki bilgisayar arasında gerçek zamanlı ses verilerinin içerisine metin ve dosya gömme uygulaması yapmıştır. Bu çalışmada sayısal ses verileri içerisine gizli veri gömülebileceğini göstermiş, bilgisayar donanımlarının başarımlarını incelemiştir [15].

1.2. Tez Çalışmasının Amacı ve Motivasyonu

II. Dünya Savaşı sırasında Almanlar bir mikro-noktalama aleti geliştirmişlerdir. Bu alet aracılığıyla gizli bir mesaj, resimleme tekniğinden faydalanılarak, örneğin “i” harfindeki veya başka bir noktalama işaretindeki noktanın boyutuna indirgenip bir kağıda işlenebilmiştir. Mesajı alan kişi tarafından ise tüm bu noktalar birleştirildiğinde gizli mesaj ortaya çıkarılmaktadır. Bu aletler, teknik çizimleri de kapsayan büyük miktarda yazılı veri aktarımını gerçekleştirebilecek potansiyele sahiptir ve bütün bunları da bilgileri çok etkili bir şekilde saklı tutarak yapabilmektedir. Neticede bu sanat günümüzde; insanlığa, bilgilerin gizlice iletilmesi konusunda çağlar boyu yardımı dokunmuş bir bilime dönüşmüştür. Modern sırörtme teknik olarak, bir veriyi (mesaj) bir nesnenin içine gizli biçimde yerleştirmeyi esas almaktadır. Öyle ki, sadece belirlenen alıcı, kendisine iletilmek istenen mesajı nesneden seçebilmekte ve diğer gözlemcilerin o nesnenin içindeki mesajın varlığından haberleri olmamaktadır. Şifreleme biliminin bir kolu olarak da görülen sırörtme, bu önemli özelliğiyle şifreleme bilimini bir adım ileri taşımaktadır. Şifreleme işlemi güvenilirliği sağlasa da bir bakıma mesajın gizliliğini sağlayamamaktadır. Bu uygulamalarda bilgi sadece gönderen ve alanın anlayabileceği şekilde şifrelenirken, sırörtme uygulamalarında bilgi sadece gönderen ve alanın varlığını bildiği şekilde saklanmakta, bazen de şifrelenip ekstra koruma sağlanmaktadır [15]. Bu bilgilerden hareketle sırörtme bağımsız bir bilim dalı olarak karşımıza çıkmaktadır.

11 Eylül'de yaşanan trajik olaylarda teröristlerin ileri teknolojileri kullandığı saptandıktan sonra, sırörtme oldukça popüler hale gelmiştir. Çünkü teröristlerin ECHELON tipi sistemleri devre dışı bırakarak aralarında gizlice haberleşmek için bu

teknolojiden yararlandıkları söylentisi tüm dünyada yayılmıştır ve bu konu üzerinde 2000'li yılların başından itibaren yoğun çalışmalar yapılmaktadır.

Yukarıdaki bilgiler ve gelişmeler de göz önünde bulundurulduğunda, özellikle Internet üzerinden yapılan haberleşmelerde zararsız görülen dosyaların (metin, resim, ses, video vb.) içerisine gizli bilgilerin gömülebileceği ya da yerel bir ağda kablolu veya kablosuz şekilde gerçekleştirilen haberleşme ve dosya alışverişlerinde sırörtmenin kullanılabileceği, gizleme işlemi yapılmadan önce sıkıştırma yapmanın dosya gönderme zamanı ve enerji tasarrufu konusunda getireceği fayda ve bu gizleme ve sıkıştırma işlemine bağlı olarak güvenliği sağlanan bilginin, şifrelenerek daha güvenli hale getirilebileceği gerçeği bu tez çalışmasının temel motivasyonunu oluşturmaktadır.

Literatürde sunulan çalışmalarda, genel olarak boyutu belli kayıtlı dosyalar üzerinde veri gömme uygulamaları, şifreleme çalışmaları veya sıkıştırılmış veri dosyalarının kablosuz ortamda gönderimi yapılmaktadır. Bu tezde sunulan çalışmada, gerçek zamanlı olarak elde edilen sayısal ses verileri sıkıştırma, şifreleme ve veri gömme algoritması içerisinden geçirilerek hedef noktaya kablosuz ortamda gönderilmektedir. Bu durumdan haberdar olan bir alıcı yazılım yardımı ile sıkıştırılmış veri açılarak şifre çözme algortiması yardımıyla çözülmekte ve gömülü veriler ayrıştırılmaktadır.

1.3. Tez Organizasyonu

Yapılan çalışmaların sunulduğu bu tez 8 ana bölümden oluşmaktadır.

Bölüm 1 Giriş: Bu bölümde tez çalışmasına konu olan problemin tanımı, çalışmanın amacı, literatürdeki ilgili problemle ilgili yapılan çalışmaların özetleri ve tez çalışmasının amacı ve motivasyonu hakkında bilgi sunulmaktadır.

Bölüm 2 Kablosuz Yerel Alan Ağları: Kablosuz ağların üstünlükleri ve kullanım alanlarından bahsedilmekte ve tez çalışmasının bir parçası olan IEEE 802.11

kablosuz yerel alan ağlarının protokol mimarisi, standartları, çerçeve biçimleri ve ortam erişim mekanizmasına ayrıntılı olarak değinilmektedir.

Bölüm 3 Şifreleme Teknolojileri : Şifreleme teknikleri hakkında bilgi verilmektedir. Simetrik ve Asimetrik Şifreleme hakkında bilgi verilmektedir. Damgalama (Watermarking) ve Sırörtme (Steganography) ile ilgili terimlerin tanımlamaları yapılmaktadır. Sırörtme teknikleri hakkında bilgi verilmektedir. Ayrıca sırörtme biliminde sıkça rastlanan bir yöntem olan gömü verisini elde etme (sıraçma) anlatılmaktadır.

Bölüm 4 Sıkıştırma Teknolojileri : Veri artıklık türleri, statik ve adaptif sıkıştırma algortimaları hakkında bilgi verilmektedir. Zlib sıkıştırma algortiması hakkında bilgi verilmektedir.

Bölüm 5 Verilerin Sayısallaştırılması: Metin kodlama standartları ASCII ve EBCDIC anlatılmakta ve tablolar halinde karakterlerin sayısal karşılıkları verilmektedir. Aynı zamanda ilgili alt bölümde analog ses bilgisinin örneklenerek sayısal ses verisi haline getirilmesi süreci anlatılmakta ve bilgisayarda PCM ses verisi formatının detayları incelenmektedir.

Bölüm 6 Sıkıştırılmış ve Şifrelenmiş Gizli Dosya Transferinin Gerçekleştirilmesi : Geliştirilen uygulamaların kullanıcı arayüzleri, çalışma prensipleri anlatılmakta ve akış diyagramları verilmektedir. Ayrıca programların nasıl kullanılacağı hakkında bilgi sunulmaktadır.

Bölüm 7 Geliştirilen Yazılım Uygulamaları: Geliştirilen yazılımların uygulama başarımları ve çalışmalar sırasında karşılaşılan problemler hakkında tespitler vurgulanmaktadır. Buna ek olarak farklı boyut ve tipteki dosyaların gönderilme ve alınma sürelerine ilişkin sonuçlar tablolar ve grafikler yardımıyla karşılaştırılmaktadır.

Bölüm 8 Tartışma ve Değerlendirmeler: Elde edilen veriler tartışılmakta ve sonuçlar irdelenmektedir. Karşılaşılan problemlerin aşılabilmesi için çözüm önerileri

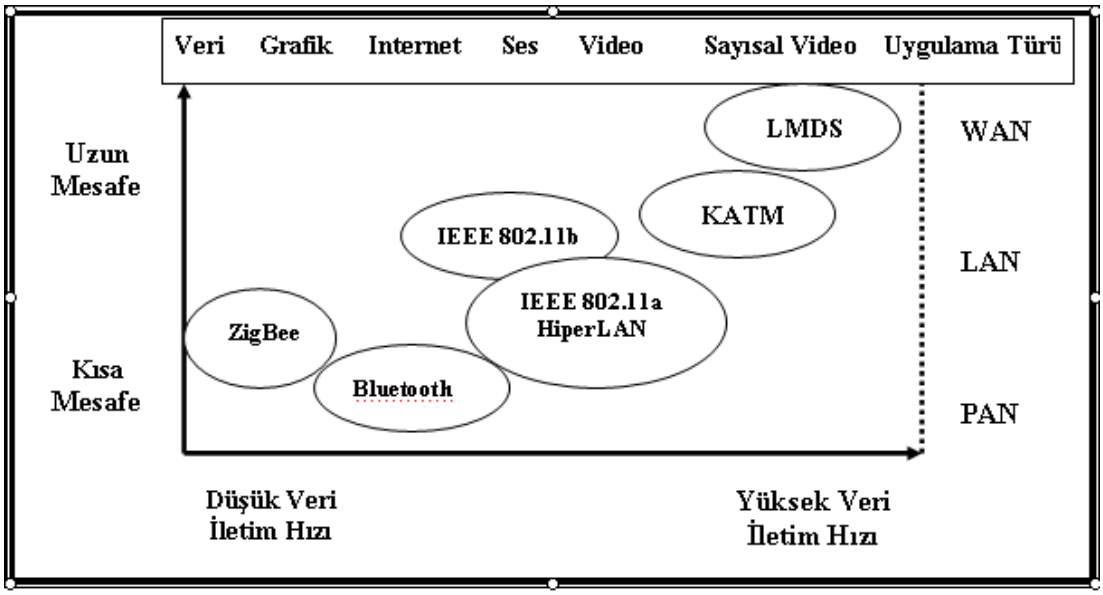
sunulmakta, ilgili çalışmaların maliyet-etkin hale getirilebilmesi için yapılabilecekler hakkında önerilerde bulunulmakta ve bundan sonra yapılabilecek çalışmalar önerilmektedir.

BÖLÜM 2. KABLOSUZ YEREL ALAN AĞLARI

2.1. Giriş

Kablosuz ağlar, haberleşmek için radyo frekans (RF) teknolojilerini kullanan terminallerin oluşturduğu sistemlerdir. Bunlar kablo kullanan eşleniklerinden farklı olarak kurulum kolaylığı, ölçeklenebilirlik, hareketlilik, üretkenlik, ileriye yönelik maliyet kazancı ve mevcut ağ yapısını genişletme gibi birçok üstünlük sunmaktadır. Bunlara karşın, kablosuz iletim ortamının doğasından kaynaklanan yüksek bit hata oranı ve sınırlı bant genişliği gibi önemli zayıflıklara da sahiptir [16-18].

Farklı uygulamalar ve ihtiyaçları karşılamak üzere birçok kablosuz ağ teknolojisi geliştirilmiş ve geliştirilmektedir. Şekil 2.1'de günümüzde mevcut ve geliştirilmekte olan kablosuz ağ standartlarının, destekledikleri uygulama türlerine, veri iletim hızlarına, kapsama alanlarının büyüklüğüne ve coğrafik ağ yapılarına göre yapılan sınıflandırmaları özetlemektedir [17].



Şekil 2.1. Kablosuz ağlar ve uygulamadaki yerleri

IEEE 802.11 Kablosuz Yerel Alan Ağları (KLAN), kablolama sınırlamaları olmaksızın Ethernet ve Token Ring gibi geleneksel LAN teknolojilerinin tüm özelliklerini ve yararlarını sağlar. Bu nedenle mevcut yerel alan ağlarının kablosuz ortam üzerinden haberleşen şekli olan kablosuz yerel alan ağları hava üzerinden Ethernet olarak da adlandırılır [37].

Bu bölümde tez çalışmasının bir parçası olan ve yaygın olarak kullanılan IEEE 802.11 Kablosuz Yerel Alan Ağları (Wireless Local Area Network) incelenmektedir. Tez çalışmasının amaçlarından birisi de kablosuz olarak ses iletişiminin gerçekleştirilmesidir. Bu amaçla bu bölümde Kablosuz ağlar ile ilgili temel bilgiler verilmektedir.

2.2. Kablosuz Yerel Alan Ağlarının Üstünlükleri

Kablosuz ağların üstünlükleri aşağıdaki şekilde sıralanabilir:

1. Hareketlilik: Kablosuz ağlar, ağ kullanıcılarına kapsama alanı içerisinde kalmak şartı ile hangi noktada olurlarsa olsunlar, hareket halinde dahi gerçek zamanlı bilgi iletişimi imkanı sağlar.
2. Kurulum hızı ve basitliği: Kablosuz ağ sistemlerinin kurulumu hızlı ve kolaydır. İletişim radyo dalgaları ile sağlandığından klasik LAN'lardaki gibi kablo çekme zorunluluğu yoktur.
3. Kurulum esnekliği: Kablosuz ağ teknolojisi kablolu LAN'ların erişemeyeceği (fiziki olarak) yerlere (noktalara) ulaşımı sağlar.
4. İleriye yönelik maliyet kazancı: Kablosuz ağların ilk kurulum maliyetleri nispeten kablolu bir ağdan daha fazla olmakla birlikte çalışma evresi sarfiyatı çok düşüktür. Uzun vadeli kazançları, çok yer değiştirme gerektiren dinamik ortamlarda ortaya çıkar.
5. Genişletilebilirlik: Kablosuz iletişim ortamı sayesinde dinamik bir yapıya sahip kablosuz ağlar ile kurulan sistemler kolaylıkla tekrar düzenlenebilir ve alan genişletilebilir. Aynı zamanda kurulu kablolu yapıların da genişlemesini sağlarlar. En az iki düğümün bir araya gelmesiyle oluşabileceği gibi erişim noktası kullanarak haberleşen düğüm sayısı yüzler hatta binleri bulabilir [19, 20] .

2.3. Kablosuz Yerel Alan Ağlarının Kullanım Alanları

Kablosuz yerel alan ağları, kablolu ağların kullanıldığı tüm yerlerde kullanılabilmektedir. Aşağıda KLAN'ların kullanım alanlarına birkaç örnek verilmektedir.

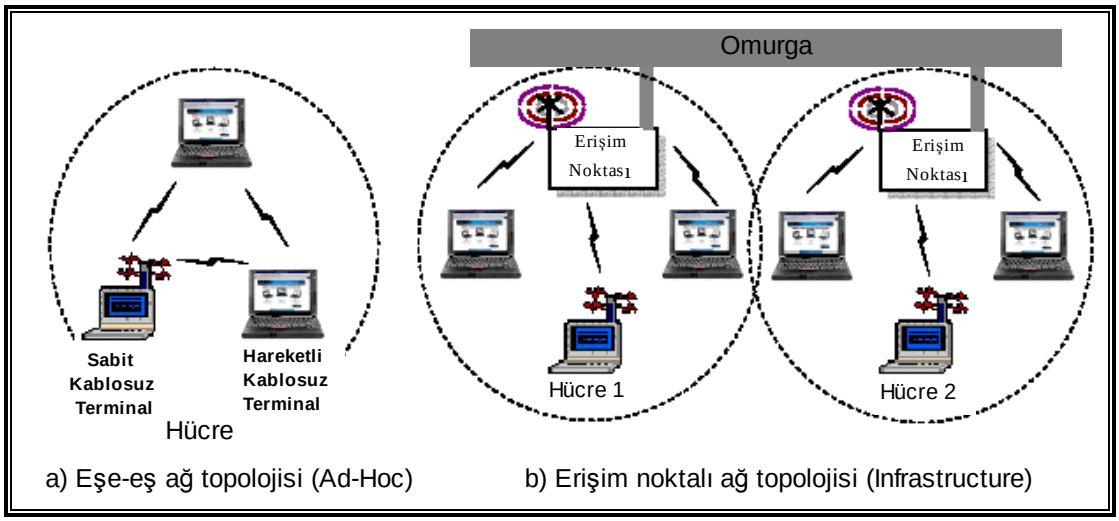
1. Endüstri: Gerçek zamanlı kontrol, dağıtık kontrol sistemleri, otomasyon sistemleri, veri tabanı bağlantısı, kent bilgi sistemleri bağlantısı.
2. Ofis ortamı: Video konferans, bilgisayar çevre birimlerinin haberleşmesi.
3. Hastane: Uzaktan görüntüleme, medikal görüntüler, hasta takibi.
4. Eğitim merkezleri: Bilgi erişim, uzaktan eğitim.
5. Taşıtlar: Araç tanıma sistemleri, araç içi kontrol uygulamaları.

2.4. Kablosuz LAN Ağ Topolojileri

IEEE 802.11 KLAN, hücresel mimariye dayalı ağ topolojilerini destekler. İletişim ortamı hücre olarak adlandırılan küçük alanlara bölünür. Her bir hücre Temel Servis Seti (Basic Service Set, BSS) olarak adlandırılır. Kablosuz ağlar, eşe-eş ya da birebir ağ topolojisi (Ad-Hoc veya independent BSS) ve erişim noktalı ağ topolojisi (Infrastructure BSS) olmak üzere iki tür ağ topolojisini desteklemektedir.

Eşe-eş ağ, gerçekleştirilmesi hızlı ve kolay olan, geçici bağlantılar sağlamak üzere kurulan bir ağ yapısıdır. Aynı protokolü kullanan en az iki kablosuz terminalin bir araya gelmesi ile oluşur. Herhangi bir erişim noktası olmaksızın tüm kullanıcılar birbirleri ile iletişim kurarlar [16, 18, 19].

Erişim noktalı ağlar, yalnızca kablosuz terminallerin kendi aralarında haberleşmesine imkan veren eşe-eş ağların genişletilmesini, kurulu kablolu yerel alan ağları ile bütünleşmesini ve kablolu yerel alan ağları üzerinden sunuculara ulaşılabilmesini sağlar. Bu yapıda hücre içerisinde iletişim koordinasyonu sağlayan erişim noktaları (Access Point, AP) kullanılır. AP'ler kablolu ve kablosuz yerel alan ağları arasında köprü görevi gördüğü gibi kablosuz yerel alan ağlarının bant genişliklerini arttırarak daha çok sayıdaki kablosuz terminalin daha uzun mesafeli haberleşmelerini de sağlar [22].



Şekil 2.2. KLAN topolojileri: a) Eşe-eş b) Erişim noktalı ağ

2.5. TCP/IP

TCP/IP, ilk defa ABD'de ARPANet (Advanced Research Projects Agency Network) adı altında, askeri bir proje olarak geliştirildi. Önceleri askeri amaçlı düşünülen proje, üniversiteler tarafından da kullanılmaya başlandı. Ardından ABD'nin dört bir yanında birbirinden bağımsız geliştirilen ağlar, tek bir omurga altında NSFNet olarak adlandırıldı ve ulusal boyutu aşarak dünyaya yayıldı. İnternet'in ve ağ sistemlerinin oluşup yaygınlaşması da bu döneme denk gelmektedir.

Ağ işletim sistemlerine ek olarak ağı yönetmek, denetlemek, bağlantı uyumluluğu sağlamak açısından protokol olarak adlandırılan kurallar kümesi kullanılır. TCP/IP (Transport Control Protocol/Internet Protocol) tüm dünyada en yaygın olarak kullanılan protokol kümesidir. Eğer farklı ağ işletim sistemlerine veya protokolüne sahip LAN'lar birbirine bağlamak istenirse, büyük olasılıkla TCP/IP kullanılır. Çünkü, hemen her işletim sistemi TCP/IP'ye uyumlu yazılımlara sahiptir.
TCP/IP protokolünde tüm bilgisayarlar 32 bitlik "özgün" bir IP numarasına sahip olacak şekilde adreslenirler (bunun anlamı: internete aynı anda bağlı olabilecek bilgisayar sayısının en fazla 2^{32} = 4.294.967.296 olabileceğidir). Bunu bir örnekle ele alırsak, internet üzerinde 3.559.735.317 sayısı ile adreslenmiş bir bilgisayar düşünelim. Bu sayının onaltılık sayı sistemindeki karşılığının D42D4015 olduğunu kolaylıkla hesaplayabiliriz. Bu şekilde bir gösterimin hemen hiç kimseye bir şey

ifade etmeyeceği oldukça açık bir şekilde görülmektedir. Bunun yerine 32 bitlik adres, 8 bitlik adresler halinde 4'e ayrılıp (D4 2D 40 14 şeklinde), daha alışıldık bir sayı sistemiyle çalışabilmek için onluk sayı sistemine çevrilir.($(D4)_{16} = (212)_{10}$, $(2D)_{16} = (45)_{10}$, $(40)_{16} = (64)_{10}$ ve $(15)_{16}=(21)_{10}$). Bu gösterim son olarak aralara konan bir nokta ile birleştirilir ve sonuçta IP numarası olarak tanımlanan notasyona ulaşılır. Yani internet üzerinde 3.559.735.317 sayısı ile adreslenmiş bilgisayar 212.45.64.21 IP nolu bilgisayardır.

TCP/IP'de, iletilen veriler katmanlara göre paketlenerek gönderilir ve alıcıda bu paketler teker teker açılıp orijinal veriye ulaşılır. Bu yöntem, iletilen veri, iletim şekli ve iletişim yolunu birbirinden ayırarak birlikte çalışmayı kolaylaştırır.

Tez çalışmaları çerçevesinde geliştirilen uygulamalarda iki adet kullanıcı arayüzü mevcuttur. Bunlardan biri Ses Gönderici Modül, diğeri ise Ses Alıcı Modüldür. Kablosuz ağ üzerinde iki bilgisayardan her biri bu modüllerden birine sahiptir ve bu modüller yardımı ile kablosuz ses haberleşmesini gerçekleştirmektedir. Modüllerin ses haberleşmesini yapmaları IP numaraları kullanılarak gerçekleştirilmiş olup, IP numaralarında yaşanacak herhangi bir sorun ses haberleşmesinin gerçekleşmemesine sebep olmaktadır.

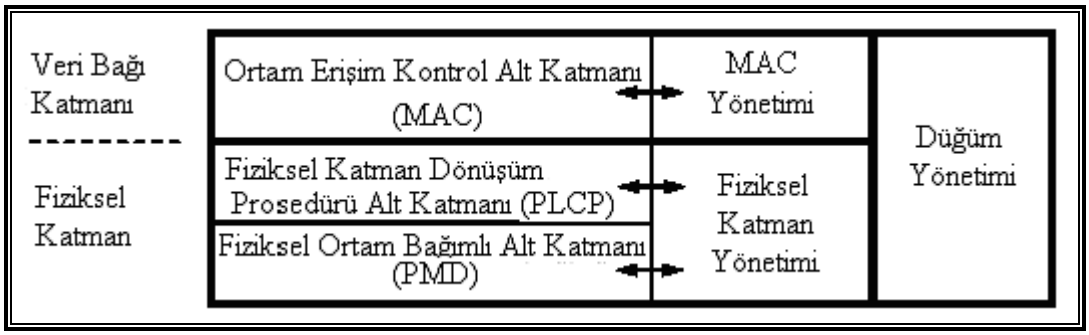
2.6. IEEE 802.11 Standardı

IEEE 802.11 KLAN standardı Amerikan Elektrik ve Elektronik Mühendisleri Enstitüsü (IEEE: The Institute of Electrical and Electronic Engineers) tarafından 1997 yılında geliştirilmiştir. IEEE, KLAN standartlarını IEEE 802.11x şeklinde tanımlamış ve bu alanda yeni standartlar geliştirmek üzere bir grup oluşturmuştur.
Kablolu yerel alan ağlarındaki Ethernet bağlantılarını kablosuz ortam üzerinden sağlayan IEEE 802.11 standardı, kablosuz yerel alan ağı standartları ailesinin temelini oluşturmaktadır. Zamanla farklı ihtiyaçlar ve farklı veri iletişim hızlarını karşılamak için birçok alt standart geliştirilmiştir.

IEEE 802.11 standardı 2.4 GHz lisanssız ISM (Industries, Scientific, Medical) bandında FHSS, DSSS ve infrared fiziksel bağlantı seçenekleri ile 2 Mbit/s'e ve 5 GHz bandında ise 54 Mbit/s'e kadar veri iletim hızlarına ulaşabilmektedir [21].

2.6.1. IEEE 802.11 protokol mimarisi

IEEE 802.11 standardı protokol mimarisi OSI referans modelinin Fiziksel ve Veri Bağı katmanlarını kapsar. Şekil 2.3'de IEEE 802.11 standardı temel referans modeli görülmektedir [21].



Şekil 2.3. IEEE 802.11 temel referans modeli

Fiziksel katman, kablosuz iletişim ortamı (medya) ile Ortam Erişim Kontrol (MAC) alt katmanını birbirine bağlayan arayüzdür. Fiziksel Katman Dönüşüm Prosedürü (Physical Layer Convergence Procedure, PLCP) ve Fiziksel Ortam Bağımlı (Physical Medium Dependent, PMD) olmak üzere iki alt katmandan meydana gelmektedir.

PMD alt katmanı, kablosuz ortam karakteristiklerini (DSSS, FHSS veya DFIR) ve kablosuz ortam yoluyla veri iletimi için gerekli metotları (modülasyon, kodlama vb.) tanımlar. PLCP katmanı ise, MAC katmanından gelen paketleri PMD alt katmanı için düzenler. Aynı zamanda MAC katmanı için taşıyıcı sezme ve kanal tahsis (carrier sensing and channel assessment) işlemini gerçekleştirir [21].

MAC katmanı, kablosuz ortamın kullanıcılar arasında etkin olarak paylaştırılmasını yani kullanıcıların ortama erişim mekanizmasını tanımlar. Bunun yanı sıra veri paketlerinin parçalanması (fragmentation), hata iyileştirme, hareketlilik yönetimi,

güç tasarrufu ve şifreleme gibi işlemleri de gerçekleştirir. MAC tüm fiziksel katman türleri (DSSS, FHSS, DFIR) için ortak olmakla birlikte veri iletim hızları farklılık gösterir.

Fiziksel katman yönetimi, farklı bağlantı şartlarının uyarlanması fonksiyonlarını, MAC yönetimi ise senkronizasyon, güç yönetimi, birliktelik (association) ve tekrar birliktelik fonksiyonlarını içerir. Düğüm yönetimi, fiziksel ve MAC yönetim katmanlarının etkileşiminden sorumludur [21].

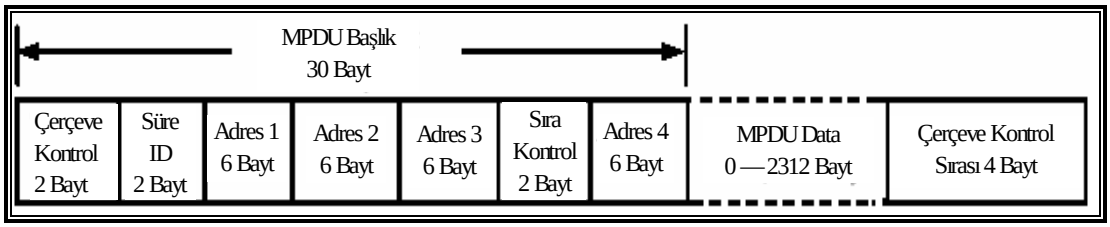
2.6.2. IEEE 802.11 çerçeve formatları

KLAN, MAC katmanında farklı amaçlar için kullanılan üç temel çerçeve biçimi (MAC Protocol Data Unit, MPDU) vardır [23]. Bunlar:

a. Veri çerçeveleri,
b. Kontrol Çerçeveleri (RTS, CTS, ACK) ve
c. Yönetim çerçeveleri (işaretleşme).

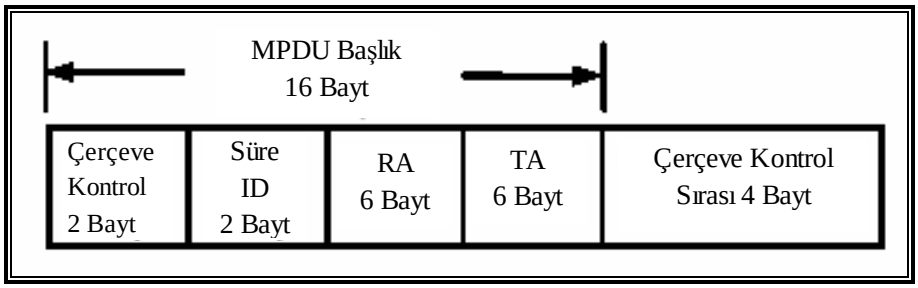
MAC veri çerçeve biçimi: Veri ve yönetim çerçeveleri için kullanılır. Şekil 2.4'de genel MAC çerçeve biçimi görülmektedir. MPDU başlık kısmındaki bölümler ve işlevleri şunlardır:

Çerçeve Kontrol: Dağıtık sisteme gönderilen/alınan paketlerin kontrolü, güç yönetimi, paket ayırma, şifreleme, kimlik belirleme (authentication).
Süre ID: Tahsis edilen vektörün süresi, güç koruma modunda çalışan düğümün tanımlanması.
Adres 1-4: BSS ID, hedef, kaynak, verici/alıcı için adresler.
Sıra Kontrol: Paket ve paket parçacıkları için sıra numarası.



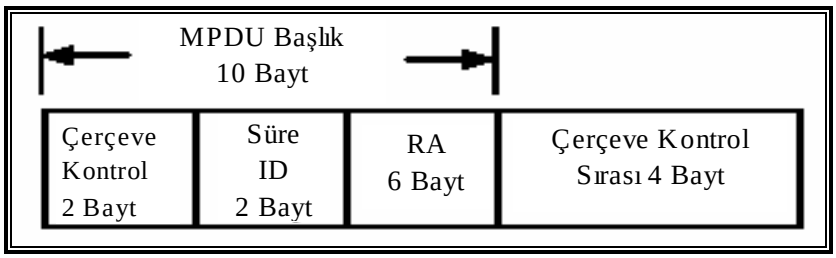
Şekil 2.4. MAC (MPDU) genel çerçeve biçimi

RTS (Request to Send) kontrol çerçeve biçimi: Süre alanında bir sonraki veri/yönetim çerçevesinin iletimi için gerekli zaman tanımlanmaktadır (Şekil 2.5). RA, bir sonraki veri/yönetim çerçevesini alacak düğümün adresini içerirken, TA ise RTS çerçevesini gönderen düğümün adresini içermektedir.



Şekil 2.5. RTS çerçeve biçimi

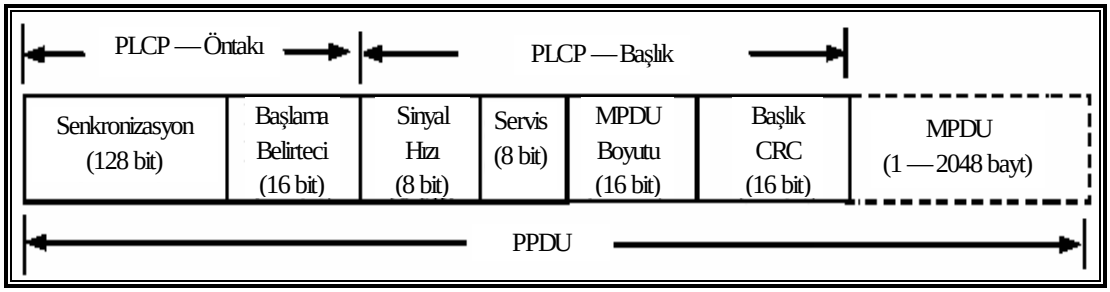
CTS (Clear to Send) kontrol çerçeve biçimi: CTS, RTS çerçevesine yanıt olarak gönderilir (Şekil 2.8). RA, alanına alınan RTS çerçevesindeki TA alanı adres bilgisi yüklenir. Süre alanına ise RTS çerçevesindeki süre alanındaki değerden CTS göndermek için gerekli zaman ve SIFS (Short Inter Frame Space) değerlerinin çıkarılması sonucu kalan değer yüklenir.



Şekil 2.6. CTS çerçeve biçimi

ACK kontrol çerçeve biçimi: ACK çerçeve CTS çerçevesi ile aynı biçimdedir. RA alanına hedef düğüm adresi, süre alanına da alınan çerçevenin süre alanındaki değerden ACK çerçeve göndermek için gerekli zaman ve onun çerçeve iletim boşluğu (SIFS) süresi çıkartılarak kalan değerler yüklenir.

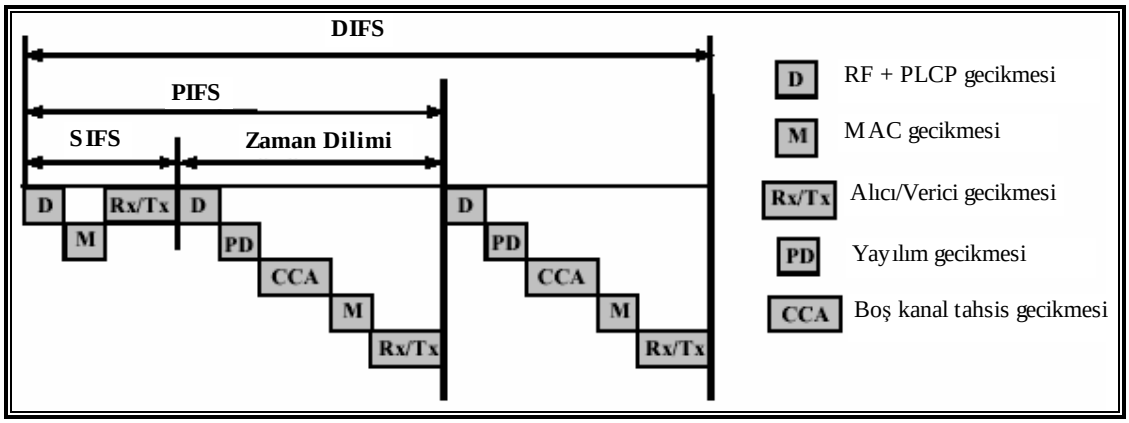
MAC katmanında oluşturulan çerçeveler iletilmek üzere fiziksel katmana gönderilir. Kullanılan fiziksel katmana (DSSS, FHSS, kızılötesi vb.) göre MAC çerçevesine bazı ilaveler yapılır. Şekil 2.7'de görülen IEEE 802.11b DSSS Fiziksel Katman Dönüşüm Prosedürü (PLCP) çerçeve biçimi; PLCP öntakısı (Preamble), PLCP başlık (header) ve MAC çerçevesinden oluşmaktadır. Öntakı alanı senkronizasyon, kanal tahsisi ve çerçeve zamanlaması için gerekli başlangıç bilgisini içerir. Başlık alanı ise kullanılan modülasyon tekniğini (DBPSK, DQPSK vb), veri iletim hızı bilgisini, gönderilen MAC çerçevesinin boyutu ve başlık alanındaki hata kontrolü (CRC) için gerekli bilgileri içerir.



Şekil 2.7. IEEE 802.11b DSSS PLCP çerçeve biçimi

2.6.3. Çerçeveler arası boşluk (Inter Frame Space, IFS)

Çerçeveler arasındaki zaman aralıkları, çerçeveler arası boşluk olarak adlandırılır ve kablosuz veri iletim hızlarından bağımsızdır. IFS her bir fiziksel katman için sabittir. IEEE 802.11 standardında MAC protokolünün ortama erişimi belirlemesinde çerçeveler arasındaki boşluk çok önemlidir. Çünkü çerçeveler arası boşluklar, ortama erişimi belirleyen Backoff algoritmasının çalışma süresini etkilemektedir [21]. IEEE 802.11 ortama erişim için farklı öncelikler sağlamak için dört farklı çerçeveler arası boşluk tanımlar (Şekil 2.8).



Şekil 2.8. Çerçeveler arası boşluk tanımlamaları

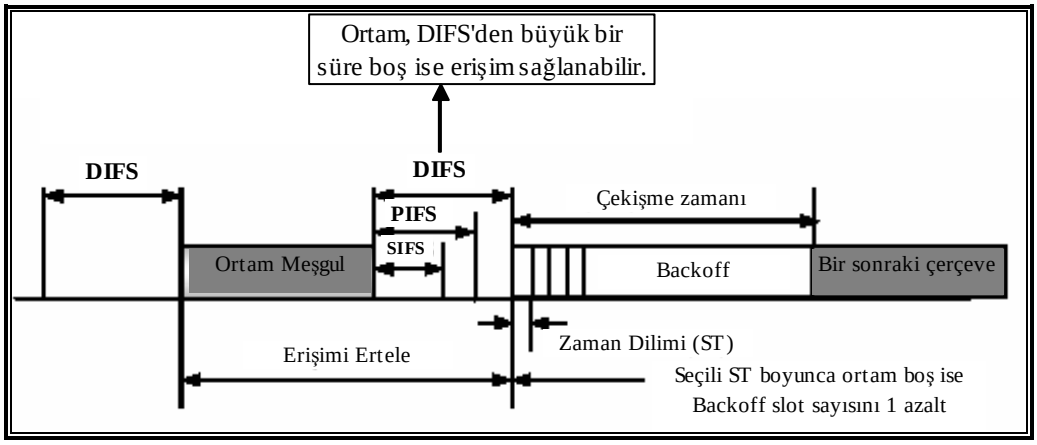
a. Zaman dilimi (Slot time): Backoff algoritmasında, her zaman diliminde ortamın meşgul olup olmadığı kontrol edilir.
b. En kısa çerçeveler arası boşluk (Short IFS, SIFS): Acil yanıt gönderiminde (ACK, RTS, CTS çerçevelerinin) kullanılır. SIFS, kullanılan fiziksel katmana bağlı olarak sabit bir değerdir. Ortam erişimini kazanmış bir düğüm, SIFS aralıklara yüksek öncelikli olarak iletimini gerçekleştirir.
c. Nokta eşgüdüm fonksiyon çerçeveler arası boşluk (Point Coordination Function IFS, PIFS): PCF erişim mekanizmasında ortam erişimini kazanmak için kullanılır. PIFS, SIFS ve Zaman Dilimi sürelerinin toplamına eşittir.
d. Dağıtık eşgüdüm fonksiyon çerçeveler arası boşluk (Distributed Coordination Function IFS, DIFS): Ardışık veri paketleri arasındaki minimum gecikmedir. Ortamın boş olduğundan kesinlikle emin olmak için düğümler DIFS süresi boyunca erişimlerini ertelerler. DIFS, PIFS ve Zaman Dilimi sürelerinin toplamına eşittir.
e. Genişletilmiş IFS (Extended IFS, EIFS): En uzun çerçeveler arası boşluktur. Hatalı paket alan düğüm tarafından kullanılır.

Örneğin DSSS kullanılan bir sistemde SIFS = 10 µs, Zaman Dilimi ise 20 µs'dir. FHSS kullanılan bir sistemde ise SIFS = 28 µs, Zaman Dilimi ise 50 µs'dir [23].

2.6.4. Ortam erişim mekanizması (MAC)

Ortam erişim mekanizmaları, sınırlı bant genişliğine sahip kablosuz iletim ortamını kullanıcılar arasında etkin olarak paylaştırmayı sağlayan kurallar bütünüdür. IEEE 802.11 MAC katmanında çekişme esaslı dağıtık eşgüdüm fonksiyonu (Distributed Coordination Function, DCF) ve çekişmeden bağımsız nokta eşgüdüm fonksiyonu (Point Coordination Function, PCF) olmak üzere iki farklı erişim mekanizması kullanılabilir [21, 24].

Şekil 2.9'da IEEE 802.11 ortam erişim mekanizmasının genel çalışma prensibi görülmektedir. PIFS yalnızca PCF erişim yöntemi ile çalışan düğümlerde kullanılır. PCF erişim noktası kullanılan sistemlerde geçerlidir. Bu yöntemde çekişmeden bağımsız olarak ortam boş olduğu sürece düğüm PIFS aralıkları ile çerçeve iletimini gerçekleştirir. DIFS ise DCF erişim mekanizmasını kullanan düğümlerde çerçeve iletimi arasındaki minimum süredir [21, 24].



Şekil 2.9. IEEE 802.11 ortam erişim mekanizmasının genel çalışması

2.6.5. Hata sezme

802.11 standardında bir paketin doğru olarak iletilip iletilmediği, ACK alındı paketlerinin gönderimi ile belirlenir. Bir paket doğru olarak alındığında vericiye bir ACK gönderilir. ACK SIFS'den sonra gönderilir. SIFS, DIFS'den küçük olduğundan herhangi yeni bir paketin gönderim zamanından önce alındı bilgisi gönderilmiş olur.

ACK gelmez ise kaynak düğüm, paketin bozulduğunu (hata oluştuğunu) varsayar ve tekrar gönderir. Tekrar gönderme işleminin daha üst katman yerine MAC katmanı tarafından gerçekleştirilmesi, kaybedilen çerçevelerin daha hızlı şekilde yeniden transferine (elde edilmesine) olanak sağlar [21].

ACK her ne kadar güvenli paket iletimi için kullanılsa da yayın (broadcast) modunda veya çoklu gönderim durumunda çok sayıda ACK gönderimi, çarpışma meydana getireceğinden pratik bir yöntem değildir.

2.6.6. IEEE 802.11 alt standartları

IEEE 802.11x ailesinin temelini IEEE 802.11 standardı oluşturmaktadır. Bu standart 2,4 GHz lisanssız ISM bandında FHSS, DSSS ve kızıl ötesi uygulama seçenekleri ile 2 Mbit/s'e kadar veri ilerim hızlarını destekleyebilmektedir. Gelişen teknoloji ile birlikte farklı ihtiyaçları karşılamak üzere farklı iletim hızları ve farklı fiziksel katman seçenekleri ile IEEE 802.11 standardını esas alan alt standartlar geliştirilmiştir. IEEE tarafından geliştirilen bu standartlar ANSI ve ISO tarafından da kabul edilmiştir. Bu alt standartların en yaygınları IEEE 802.11a, IEEE 802.11b ve IEEE 802.11g'dir.

2.6.6.1. IEEE 802.11a standardı

IEEE 802.11 ailesi içerisinde yeni nesil kablosuz LAN standardıdır. 2,4 GHz'deki band genişliğini kullanan değişik uygulamalara, 5 GHz'lik frekans bandını tanımlayarak alternatif oluşturmaktadır. 5, 15-5, 25 GHz, 5, 25-5, 35 GHz ve 5, 725-5, 825 GHz frekansları arasında 300 MHz'lik bir frekans bandında çalışır. IEEE 802.11.a standardı, 5 GHz lisanssız U-NII (Unlicensed National Information Infrastructure) bandda OFDM modülasyonu kullanarak veri iletim hızını kanal (üst üste binmeyen 8 kanal kullanır) başına 54 Mbit/s'e kadar çıkarmıştır. 8 Mbit/s, 9 Mbit/s, 12 Mbit/s, 24 Mbit/s, 38 Mbit/s, 48 Mbit/s ve 54 Mbit/s veri iletim hızlarını destekleyen bu standart çoklu ortam uygulamaları ve veri aktarımının yoğun olduğu uygulamalar için daha uygun olacaktır [16, 20].

DSSS yerine OFDM tekniğinin kullanılması daha iyi başarım ve daha geniş kapsama alanı sunmakla birlikte daha fazla güç harcaması gerektirir. IEEE 802.11a HiperLAN2 standardına rakip olarak geliştirilmiştir.

2.6.6.2. IEEE 802.11b standardı

Uygulamada en yaygın kabul gören standarttır. 802.11b standardı 2,4 GHz ISM bandında çalışır ve modülasyon tekniği olarak yalnızca DSSS kullanır. 1 Mbit/s, 2 Mbit/s, 5,5 Mbit/s ve 11 Mbit/s veri iletim hızlarını destekler. Kablosuz yerel alan ağlarının 2,4 Ghz ISM bandını mikrodalga fırın ve Bluetooth gibi ürünler ile paylaşması, olası parazitlerden dolayı veri kayıplarına ve veri iletim hızlarının düşmesine neden olabilmektedir.

Farklı firmaların 802.11b ürünleri arasındaki birlikte çalışabilirliğin bugün WiFi Alliance olarak bilinen WECA (Wireless Ethernet Company Alliance) tarafından onaylanması ile IEEE 802.11b bir endüstri standardı haline gelmiştir. Bu kurumun amacı WiFi ürünlerinin işlerliğini sertifikalandırmak ve IEEE 802.11b'yi global bir standart yapmaktır [37].

2.6.6.3. IEEE 802.11g standardı

Bu standardın kullanımındaki amaç, mevcut IEEE 802.11b standardı üzerinden veri iletim hız artırımını sağlamaktır. 802.11b'de olduğu gibi 2,4 GHz bandı kullanılmakla birlikte 54 Mbit/s'lik veri iletim hızı sağlar. OFDM ve CCK (Complementary Code Keying) modülasyon tekniklerinin her ikisini de destekler. Günümüzde 802.11b'nin yerini almak üzeredir.

Anılan 802.11 standartları ile bu standartlar ailesi üzerinde yapılan çalışmalar Tablo 2.1'de gösterilmektedir.

Tablo 2.1. IEEE 802.11 standart ailesi

Standart	Özellikleri
IEEE 802.11	Orijinal Wlan Standardı.1—2 Mbit/S Veri İletim Hızlarını Destekler.
IEEE 802.11a	5 Ghz U-Nıı Bandında Çalışsan Yüksek Hızlı Klan Standardı. Kanal Başına 54 Mbit/S Veri İletim Hızını Desteklemektedir.
IEEE 802.11b	2,4 Ghz Ism Bandında 11 Mbit/S Veri İletim Hızını Destekler.
IEEE 802.11e	Ieee Klan Yapıları İçin Servis Kalitesini Arttırmak Ve Yönetmek.
IEEE 802.11f	Ap'ler Arasında Haberleşme Protokolüdür (Inter Access Point Protocol, Iapp)
IEEE 802.11g	802.11b Standardı Üzerinde Kurulan Bu Standard 2,4 Ghz'de 54 Mbit/S Veri İletim Hızına Ulaşabilmektedir.
IEEE 802.11h	IEEE 802.11a için dinamik kanal seçimi ve iletim gücü kontrolü sağlar.
IEEE 802.11i	IEEE 802.11X ile kombine güvenlik özellikleri sunmaktadır.
IEEE 802.11n	2007'nin ortalarında standartlaşma çalışmalarının tamamlanması beklenmektedir. Kablosuz yerel alan ağları içerisinde en yüksek veri iletim hızını (540 Mbit/s) ve çalışma mesafesini (kapalı ortam 50 m) desteklemesi planlanmaktadır. 802.11n, diğer 802.11 standartlarına MIMO (Multiple Input Multiple Output) eklenilerek geliştirilmektedir.
IEEE 802.11X	IEEE ağları için güvenlik çerçeve standardı.
WISPR (Wireless ISP Roaming)	Kablosuz Ethernet Uyumluluğu Topluluğu tarafından geliştirilen, kablosuz kamusal ağlar arasında dolaşım için tavsiyeler bütünüdür

2.7. Sonuç

Kablosuz iletişim ortamının sınırlamalarına rağmen, kablosuz yerel alan ağlarının kullanımı; kurulum kolaylığı ve basitliği, esnekliği, ileriye yönelik maliyet kazancı, hareketlilik ve mevcut yerel alan ağ yapısını genişletme gibi üstünlüklerinden dolayı gün geçtikçe artmaktadır.

Günümüzde mevcut ve geliştirilmekte olan bir çok kablosuz iletişim teknolojisi olmasına rağmen, kablosuz Ethernet olarak adlandırılan IEEE 802.11 standardının en

büyük üstünlüğü, oldukça yaygın bir kullanım oranına sahip (%95 civarında) standart kablolu Ethernet yapısı ile sağladığı kolay entegrasyondur.

Veri iletim hızı uygun olmayan teknolojilerle gerçekleştirilecek bir ses haberleşmesinde veri gönderimindeki gecikmeler gerçek zamanlı ses haberleşmesine imkan vermeyecek, dolayısı ile uygulamanın hayata geçirilmesini önleyecektir. Burada sunulan tez çalışmalarında bu durum göz önünde bulundurularak, gerçek zamanlı ses haberleşmesinde ideal bir çözüm olan IEEE 802.11g standardını destekleyen donanımsal aygıtlar kullanılmakta ve uygulamalar bu temelden hareketle geliştirilerek çalıştırılmaktadır. Özellikle kablosuz haberleşme ortamlarında gönderilen/alınan verilerin herkese açık olması bu verilerin güvenliğini tehlikeye sokmaktadır. Bu tez çalışmasında gönderilen/alınan verilerin güvenliğini sağlamak için şifreleme, veri iletimi esnasında gönderilen/alınan veri paketlerinin sayısını düşürmek ve dolayısıyla veri iletim süresini düşürmek için sıkıştırma uygulanmıştır. Böylece kablosuz iletim ortamının zayıflıklarına (düşük veri iletim hızı, veri güvenliğinin düşük olması gibi) bir çözüm getirilmeye çalışılmıştır.

BÖLÜM 3. ŞİFRELEME TEKNOLOJİLERİ

3.1. Giriş

Temeli antik çağlara kadar dayanan gizli haberleşme, teknoloji değişip geliştikçe şekil ve yöntem açısından da faklılıklar göstermektedir. Bununla birlikte önemini devamlı korumaktadır. Gizliliğin öneminin had safhaya ulaştığı uygulamalarda; gizli bilgilerin, üçüncü kişilerin eline geçmeden ilgili hedefe gönderilmesi amaçlanmaktadır.

Ancak herhangi bir nedenden ötürü gönderilecek olan bilginin üçüncü şahısların eline geçse dahi bu kişilerin bu bilgilere ulaşması istenmemektedir. Buda karşımıza Şifreleme (Cryptography) bilimini çıkarmaktadır.

Bir takım gizli mesajların yetkilendirme prensibinden hareketle yalnızca ilgililer tarafından okunması, diğer kişiler tarafından ise ya “gizlenmiş veriden” haberdar dahi olmaması ya da haberi olsa dahi gömülü bilgiyi elde edememesi arzu edilir. Bu maksatla veri gizleme (data hiding) teknikleri bünyesinde Sırörtme(steganografi) bilim dalı kullanılmaktadır [25].

Özellikle kablosuz haberleşme ortamlarında gönderilecek veri miktarı haberleşmenin başarımını etkilemektedir. Dosya/Veri alış verişlerinde ilgili bilgilerin sıkıştırılması gönderilecek/alınacak veri miktarını azaltarak daha iyi başarım sağlayacaktır.

3.2. Şifreleme Biliminin (Cryptology) Temelleri

“Cryptography” kelimesi gizli yazı anlamına gelen, “secret(crypto-)” ve “writing(-graphy)” kelimelerinden türetilmiştir. Özel, gizli içeriğe sahip bilgi veya mesajların

anlamlı olarak, kaynak veya alıcıdan başka üçüncü kişilerin eline geçmesini önlemek amacıyla kullanılan tüm teknikleri içeren bir bilim dalıdır. Bu maksatla, gelişmiş algoritma teknikleri kullanılmaktadır. Alıcıda elde edilen mesajın kaynaktaki (orijinali) ile aynı olmasını sağlamak, doğruluğunu ispatlamak yine bu algoritma tasarıları ile sağlanmaktadır. Şifreleme bilimi, mesajların gizli tutulması sanatıdır. Şifre çözme, şifreli verilerin kırılması sanatıdır, örneğin doğru anahtarı bilmeden düz metnin elde edilmesi. Şifreleme bilimi ile uğraşanlara kriptograf ve şifre çözümü uygulamacılarına kriptanalist denir.

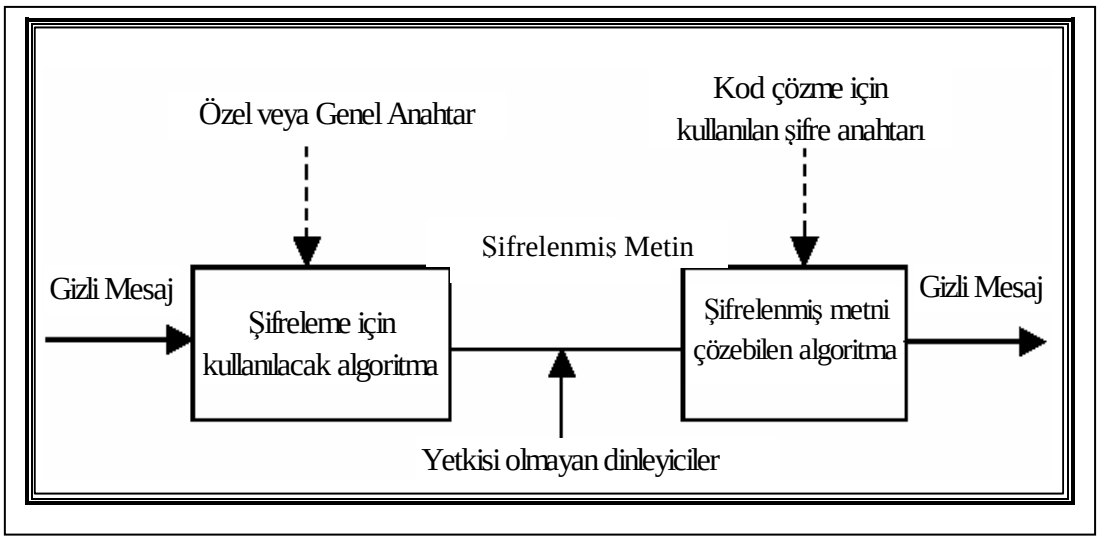
Şifreleme, kod-kırma sanatı kriptanaliz alanını da içine alan şifreleme biliminin bir dalıdır. Şifrelemede, kriptosistem ya da şifre denilen bir algoritma kullanılarak mesaj ve genellikle anahtar olarak bilinen bazı ilave bilgiler birleştirilir ve bir kriptogram üretilir. Bu tekniğin tümü şifreleme olarak adlandırılır. Bir kriptosisteme güvenli denebilmesi için anahtar olmadan kriptogramın kilidini çözmek imkansız olmalıdır [26].

Şifreleme terminolojisinde mesaj düz metin (plaintext) veya temiz metin(cleartext) olarak adlandırılır. Mesajın içeriğini diğer kişilerden saklamak için kodlamaya şifreleme(encryption) adı verilir. Şifrelenmiş mesaja, şifreli-mesaj(ciphertext) denir. Şifreli mesajdan düz metni elde etme işlemine çözme (decryption) adı verilir. Şifreleme ve çözme genelde bir anahtar kullanılarak yapılır ve çözme işlemi ancak doğru anahtarın bilinmesiyle gerçekleştirilebilir

Haberleşme ağlarında bir merkezden diğer bir merkeze gönderilen ve alındığı veya gönderildiği yerde saklanan bilgilerin korunması, yetkilendirilmemiş kişilerin bu bilgilere ulaşmasının önlenmesi, günümüz bilgi teknolojilerinde şifrelemeye (encryption-decryption & encipher-decipher) ayrılan zaman ve önemi sürekli olarak artırmaktadır. Internet & Intranet uygulamalarında; e-posta, banka işlemleri, kişisel işlem ve bilgilerin saklanması, sayısal imza ve kimliklerin üretimi, veri tabanı dosyalarının korunumu, video şifreleme, elektronik oyun ve program şifrelemesi, faks ve telefon şifrelemesi vb. uygulamaları sıkça kullanılır durumdadır.

Ticari güvenlik, askeri ve devlet güvenliği alanlarında bütün kurumların ortak hedefi, sahip oldukları önemli veya gizli bilgilerin güvenli ortamlarda saklanması ve sadece yetkisi olan kişilerin, yetkileri oranında bu bilgilere ulaşabilmelerini sağlamaktır.

Başlangıçta sadece askeri veya uluslararası/diplomatik mesajların korunarak güvenli bir şekilde alıcıya aktarılması ihtiyacını karşılamak amacıyla ortaya çıkan şifreleme teknikleri günümüzde bu alanlardaki özelliğini hala korumakla birlikte, özellikle ticari uygulamalardaki gereksinim de küçümsenmeyecek boyutlara ulaşmıştır.



Şekil 3.1. Genel şifreleme ve şifre çözme blok diyagramı

Şekil 3.1'de genel bir şifreleme sisteminin blok diyagramı görülmektedir. Sistemin amacı gizli bilginin görünümünü değiştirerek saklamak olduğundan, yetkisiz birisi tarafından bu bilgiler (veriler) anlaşılamamaktadır.

Bütün veri gizleme teknikleri, veri gömme algoritması ve bir algılayıcı fonksiyondan meydana gelir. Gömme algoritması gömü verisini (gizli mesajı) bir örtü verisine (veya taşıyıcıya) gömmek için kullanılır ve örtülü veri elde edilir. Gömme süreci bir "anahtar" mekanizmasıyla korunmaktadır. Bu yüzden yalnızca yetkili kişiler gizli anahtar ile gömü verisine ulaşabilmektedirler.

Algılayıcı fonksiyon örtülü veriye uygulanarak gömü verisi yeniden elde edilebilir. Halen sayısal resimlerde veri gizleme ile sınırlı olarak konu genişletilmektedir. Her

bir veri gizleme tekniği, planlanan uygulama tarafından dikte edilen belirli özelliklere sahip olmalıdır (örneğin, “taşıyıcı ve gizli mesaj arasında bir ilişki var mı?”, “kaç alıcı söz konusudur?”, gibi).

Sayısal ses içerisine veri gömme/şifreleme ve şifre çözme işlemleri üç temel adımla özetlenebilir:

1) Taşıyıcı işaretin “*i.*” ses çerçevesine gömülecek “ k_{m_i} ” gizli anahtar işaretinin belirlenmesi:

Ses çerçevesine gömülecek anahtar işareti tipik olarak; “*k*” gizli anahtar bilgisi ve “ ω_j ” gizli veri bilgisinin bir fonksiyonu olarak üretilir [6].

$$k_{m_i} = f(\omega_j, k) \quad j=1, 2, \ldots\ldots, L \tag{3.1}$$

(3.1) ifadesinde, *L* eklenecek gömü verisinin uzunluğunu belirtmekte olup, “ ω_j ” ise “*i.*” çerçeveye eklenecek gömü verisinin “*j.*” bitini göstermektedir. “*f*” fonksiyonu sadece “*k*” ve ω_j ’nin değil, gömü verisinin eklendiği orijinal ses çerçevesi s_i ’nin de bir fonksiyonu olarak tanımlanabilir [6].

Netice olarak hedeflenen sistemde “*f*” üç değişkenli doğrusal olmayan bir fonksiyondur.

$$k_{m_i} = f(\omega_j, k, s_i) \tag{3.2}$$

2) k_{m_i} modüle edilmiş gizli anahtar işaretinin taşıyıcı “ s_i ” işaretine gömülmesiyle örtülü veri “ $s_{i_{WM}}$ ” işaretinin elde edilmesi:

$s_{i_{WM}}$ ’nin elde edilmesi s_i ve k_{m_i} ’ye bağlı “ $f_1(s_i, k_{m_i})$ ” fonksiyonunun belirlenmesi olarak tanımlanabilir.

$$s_{i_{WM}} = f_1(s_i, k_{m_i}) \tag{3.3}$$

3) Gizli anahtar ve orijinal taşıyıcı işaret yardımıyla gömü verisinin elde edilmesi, bir başka deyişle veri çözme:

$\hat{w}_j$ öngörü ile çözülen *j*. gizli veri biti olmak üzere, (3.4) eşitliğinde *g( )* doğrusal olmayan fonksiyonu veri çözme(algılama) işlemini modellemektedir.

$$\hat{w}_j = g(s_i, s_{i_{WM}}, k) \tag{3.4}$$

Alıcıda orijinal ses işaretinin bilinmemesi durumunda *g()* fonksiyonu (3.5) eşitliğinde görüldüğü gibi iki değişkenli bir fonksiyon olarak tanımlanır.

$$\hat{w}_j = g(s_{i_{WM}}, k) \tag{3.5}$$

Modern Şifreleme bilimi temelde üç görevi yerine getirmek için kullanılır. Verinin okunmasını ve değiştirilmesini engelleme ve verinin belirtilen kişi tarafından gönderildiğinin garanti altına alınması. Bir hacker yada kötü niyetli herhangi bir kişi Internet gibi güvenli olmayan ortamlarda iki merkez arasındaki haberleşmeleri dinleyebilir ve bu haberleşmelerde gönderilen veriler üzerinde işlemler yapabilir. Bu işlemler genelde üç şekilde olabilir.

Verinin gitmesini engelleme(intercept), veriyi sadece okuma(read), veriyi değiştirme (modify). Bu tip işlemlerin yapılmasını engellemek amacıyla şifreleme bir bilim olarak çalışmaktadır ve temelde üç görev üstlenmiştir.

3.2.1. Veri güvenliği (Confidentiality)

İki merkez arasında gönderilen verinin üçüncü kişiler tarafından okunmasını engelleme(A/Symmetric Crypto). Bu basit şekilde normal yoldan gönderilen bir mektubun, alıcı kişiye giderken yolda herhangi bir kişi tarafından okunmasını (mesela postacı) engelleme amacı güder. Yazdığınız mektup düzmetin(plaintext) şeklindedir ve herhangi bir kişi tarafından zarfın açılması halinde, göndermiş olduğunuz mektup okunabilir. Şifreleme, bu düzmetni şifreleyerek yazma işlemini gerçekleştirmenizi sağlar. Bu sayede mektubunuz yolda giderken herhangi bir kişi tarafından açılması halinde yazı düzmetin olmadığı için okunması engellenecektir. (Bu şifreleme için simetrik ya da asimetrik yöntemler kullanılır.)

3.2.2. Veri bütünlüğü (Data integrity)

İki merkez arasında gönderilen verinin üçüncü kişiler tarafından değiştirilmesini engelleme. Normal yoldan göndermiş olduğunuz mektup yine üçüncü kişiler tarafından yolda sizin yazdığınız şeklin dışında başka bir şekle dönüştürülerek yolculuğuna devam ettirilebilir. Yazdığınız mektup düzmetin (plaintext) şeklindedir ve içerik okunabilmektedir. Okunabilen bu düzmetin kötü niyetli kişiler tarafından yolda değiştirilerek sizin gönderdiğiniz alıcıya, aslında söylemediğiniz şeyleri siz söylemiş gibi göstermek amacıyla yollanabilir. Şifreleme, gönderilen bu düz metin üzerinde işlem yaparak sayısal bir sonuç oluşturur. Bu sonuç, gönderilen yazının üzerinde en ufak bir değişiklik yapılırsa algoritma aynı olduğundan değişecektir.

Gönderici ve alıcı tarafından aynı yazı üzerinde aynı algoritmayla oluşturulan sayısal sonuçlar birbirinin aynı olmak zorundadır. Eğer sayısal sonuçlar tutmuyorsa gönderilen metin değiştirilmiştir şeklinde düşünülebilir. Çünkü aynı metin üzerinde yapılacak bir değişiklik aynı sayısal sonucu çıkarmayacaktır. Kullanılan algoritma sayesinde farklı metinler üzerinde aynı sayısal sonucun çıkartılması neredeyse imkansızdır.

3.2.3. Kimlik denetimi (Authentication)

İki merkez arasında gönderilen verinin, alıcı tarafında, belirtilen gönderici tarafından gönderildiğinden emin olunması gerekir(Digital Signature). Size ulaşan herhangi bir mektubun üzerinde bulanan gönderici ismi her zaman doğru olmayabilir. Kötü niyetli kişiler tarafından gönderici isimleri farklı yazılarak kişilere mektup yollanabilir (spam gibi). Şifreleme bilim olarak bu mektuplar üzerine özel imzalar (signature) ekleyerek, mektubu gönderen kişinin gerçekten mektubu gönderen kişi olduğundan emin olmanızı sağlayabilir. Gönderilen zaman dilimine göre özel algoritmalarla oluşturulan bu imzalar, alıcı kişi tarafından belirli yöntemlerle doğrulanabilir (Bu imza oluşturma ve doğrulama işlemi “Digital Signature” olarak adlandırılır).

3.3. Şifreleme Algoritmaları

Bazı şifreleme metotları algoritmanın gizliliğine dayanır; böyle algoritmaların sadece tarihi önemi vardır ve gerçek-dünya ihtiyaçları için yeterli değildir. Tüm modern algoritmalar şifreleme ve çözme işlemlerini kontrol etmek için bir anahtar kullanırlar; bir mesaj sadece kullanılan anahtar şifreleme anahtarıyla uyuştuğunda çözülebilir.

Anahtar temelli algoritmaların iki çeşidi vardır; simetrik (veya gizli-anahtar) ve asimetrik (veya açık-anahtar) algoritmalar. Aralarındaki fark, simetrik algoritmalar şifreleme ve çözme işlemleri için aynı anahtarı kullanırken (veya çözme anahtarı şifreleme anahtarından kolayca türetilebilir), asimetrik algoritmalar şifreleme ve çözme için farklı anahtar kullanırlar ve çözme anahtarı şifreleme anahtarından elde edilemez.

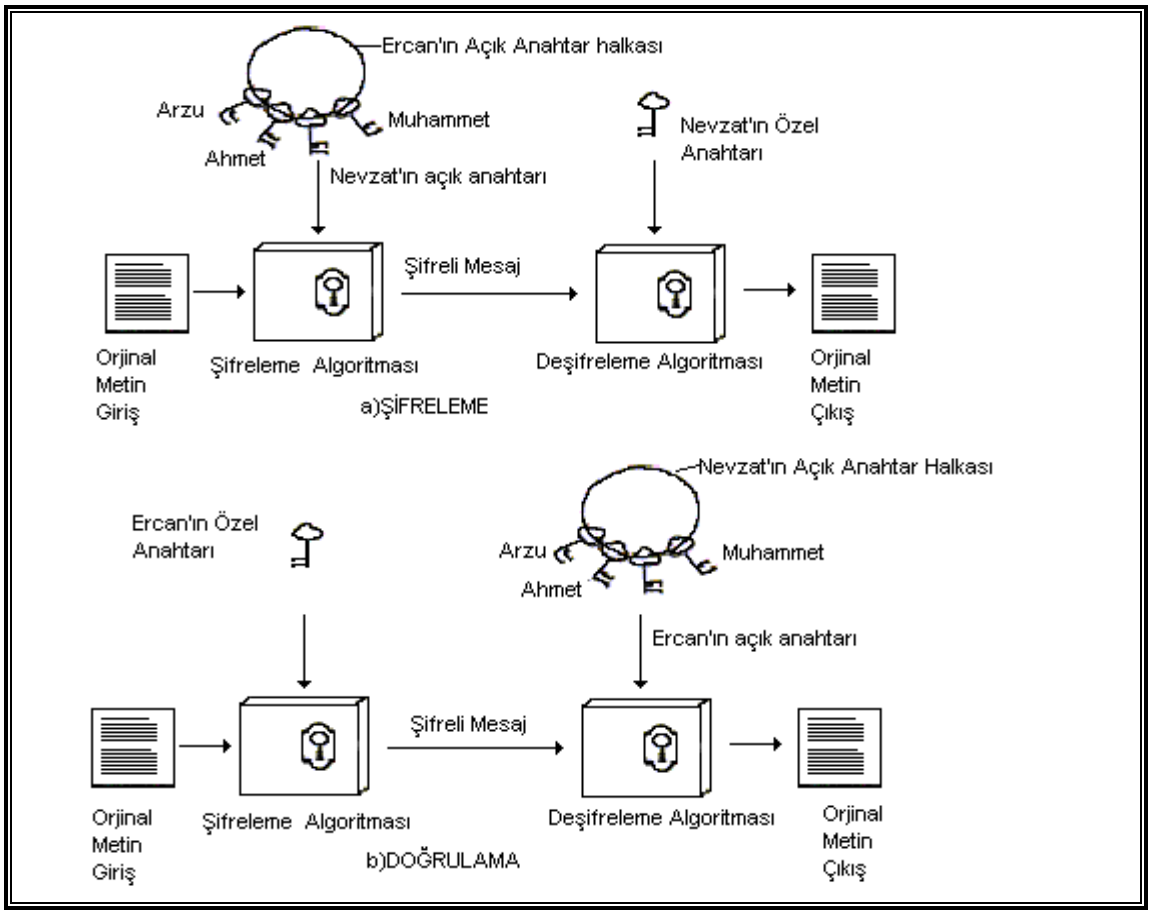
3.3.1. Açık anahtarlı şifreleme sistemleri(Asimetrik şifreleyiciler)

Açık anahtarlı şifreleme tekniği Diffie, Merklle ve Hellman tarafından keşfedildi. Açık anahtarlı şifreleme tekniğini, diğer şifreleme sistemlerinin hepsinden ayıran en önemli özellik; Açık anahtarlı şifreleme algoritmasının, yerine koyma metodu ve

permütasyondan daha çok matematiksel işlemlere dayanmasıdır. En önemlisi açık anahtar şifreleme tekniği iki ayrı anahtar kullanımı gerektirir. Bu da asimetrik olmasını sağlar. Böylece tek anahtar kullanılan simetrik geleneksel şifrelemeden daha güvenli olur. Güvenli olmasının nedeni şifre çözmek için kullanılan anahtarın paylaşılmamasıdır. İki anahtar kullanımı, anahtar dağıtımı ve kimlik doğrulama gibi güven ve gizlilik gerektiren büyük sonuçları meydana getirmiştir. Açık anahtar şifreleme sistemleri, şifreleme için açık anahtarı ve şifre çözme için özel anahtarı kullanır.

Açık anahtarlı algoritmalar, şifreleme için bir anahtar ve şifre çözme için farklı fakat ilişkili anahtarı kullanırlar. Yalnızca şifreleme algoritmasının ve şifreleme anahtarının bilgisi verildiğinde, şifre çözme anahtarını tespit etmek mümkün olmamalıdır. RSA gibi bazı algoritmalarda, her iki anahtar da şifreleme ve şifre çözme için kullanılabilir. Bir anahtar şifreleme için kullanılmış ise eş anahtarı şifre çözme için kullanılır.

Açık anahtarlı şifreleme tekniğinde, şifreleme işlemleri Şekil 2.1.a'da örnek olarak gösterilmiştir. Her kullanıcı şifreleme ve şifre çözme işlemleri için bir çift anahtar üretir. Her kullanıcı şifreleme için kullanılan anahtarını, herkes tarafından erişilebilecek bir dosyaya kaydederek açık anahtarını yayınlar. Eş anahtarı özel olarak saklanır. Bu da şifre çözme işleminde kullanılan özel anahtardır. Eğer A, B'ye mesaj yollamak istiyorsa, B'nin açık anahtarını kullanarak mesajı şifreler. B mesajı kabul ettiği zaman kendi özel anahtarını kullanarak onun şifresini çözecektir. B dışında hiçbir alıcı mesajı çözemez. Çünkü B'nin özel anahtarına yalnızca B sahiptir.



Şekil 3.2. Açık anahtar şifreleme (a- Şifreleme işlemi b-Doğrulama işlemi)

Şekil 3.2.a'da Ercan isimli kullanıcının Nevzat isimli kullanıcıya şifreli mesaj göndermesi işlemi görülmektedir. Ercan; Ahmet, Nevzat, Muhammet ve Arzu'nun açık anahtarlarına sahiptir. Mesaj göndereceği kişi Nevzat olduğu için, göndermek istediği mesajı Nevzat'ın açık anahtarını ve şifreleme algoritmasını kullanarak şifrelenmiş mesajı elde ederek Nevzat'a gönderir. Mesajı alan Nevzat ise kendi özel anahtarını kullanarak şifreli mesajı çözer ve orijinal mesaja ulaşır. Yalnız burada dikkat etmemiz gereken husus doğrulamadır. Nevzat'ın açık anahtarına sahip olan herkes Nevzat'a şifreli mesaj gönderebilir. Mesajın gerçekten Ercan'a ait olup olmadığının doğrulandığını Şekil 3.2.b'de görebiliriz. Ercan göndermek istediği mesajı kendi özel anahtarı ile şifreler. Nevzat mesajı Ercan'ın açık anahtarı ile deşifreler. Ercan'ın özel anahtarı ile şifrelendiği için Ercan tarafından gönderildiği doğrulanır. Çünkü Ercan'ın özel anahtarına yalnızca kendisi sahiptir. Mesajı Ercan'dan başkası şifreleyemez. Diyelim ki mesaj herhangi bir rakip tarafından ele geçirildi ve mesajı çözmeyi başardı. Mesajı okuyabilir ama Ercan'ın özel anahtarına

sahip olmadığı için mesajı tekrar şifreleyip Nevzat'a gönderemez. Eğer Ercan'ın özel anahtarını da ele geçirmeyi başardı ise Ercan için iletişim güvensizleşmiştir. Tekrar anahtar çifti üretmesi gerekmektedir.

Örnekten de anlaşıldığı üzere tüm kullanıcılar açık anahtarlara sahiptir ve onları kullanabilir. Fakat özel anahtar yalnızca sahibi tarafından kullanılır. Bundan dolayı dağıtılmaya ihtiyacı yoktur. Kullanıcılar kendi özel anahtarlarını kontrol ettiği sürece iletişim güvenlidir. Bir kullanıcı istediği zaman özel anahtarını değiştirebilir ve açık anahtarını yayınlayabilir.

Tablo 3.1. Geleneksel ve açık anahtarlı şifreleme

GELENEKSEL ŞİFRELEME	**AÇIK ANAHTARLI ŞİFRELEME**
Çalışması için ihtiyaçları : 1- Şifreleme ve şifre çözmede kullanılan anahtar ve algoritma. 2- Gönderen ve alıcı , algoritma ve anahtarı paylaşmalılar.	**Çalışması için ihtiyaçları :** 1-Bir algoritma, şifre çözme ve şifreleme için 1 çift anahtar kullanır. 2- Gönderici ve alıcıdan her birisi , eşlenen çift anahtarların birine sahip olmalıdır. (Aynı biri değil)
Güvenlik için ihtiyaçları : 1-Anahtar gizli tutulmalı. 2-Eğer diğer bilgi mevcut olmazsa mesajı çözmek olanaksız veya en azından elverişsiz olmalı. 3-Algoritmanın bilgisi ve şifreli mesajın örnekleri anahtarı belirlemek için yetersiz olmalı.	**Güvenlik için ihtiyaçları :** 1-İki anahtardan biri elde gizli tutulmalıdır. 2- Eğer diğer bilgi mevcut olmazsa mesajı çözmek olanaksız veya en azından elverişsiz olmalı. 3-Anahtarlar ve şifreli mesajın örneklerinin biri ve algoritmanın bilgisi diğer anahtarı belirlemek için yetersiz olmalı.

3.3.1.1. RSA Şifreleme Sistemi

Diffie ve Hellman'ın öncülük ettiği çalışmalar, şifrelemeye yeni bir yaklaşım getirdi. Açık anahtar sistemlerinin ihtiyaçlarını karşılayan bir şifreleme algoritmasında görüş birliğine vardılar. 1977'de Ron Rivest , Adi Shamir ve Len Adleman tarafından, MIT de gerçekleştirildi ve ilk olarak 1978'de (A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, February 1978) basıldı.

RSA nasıl çalışır ?

Orjinal mesaj bloklar halinde şifrelenir. Şifreleme ve şifre çözme, herhangi bir orjinal mesaj bloğu M ve şifreli mesaj bloğu C için birbirlerini takip eden yapılardır.

$C=M^e \bmod n$

$M=C^d \bmod n$

$= (M^e)^d \bmod n$

$=M^{e.d} \bmod n$

Gönderici ve alıcının her ikisi de n'nin değerini bilmek zorundadır. Gönderici e'nin değerini bilir ve yalnız alıcı d'nin değerini bilir. Böylece KU={e,n} açık anahtarı ve KR={d,n} özel anahtarı ile birlikte açık anahtar şifreleme sistemi oluşturulur. Bu algoritmanın açık anahtar şifrelemesinin şartlarını yerine getirmesi için aşağıdaki ihtiyaçlar sağlanmalıdır.

1- n>M'nin sağlandığı tüm durumlarda $M^{ed} = M \bmod n$ iken e,d ve n'nin değerleri tespit edilebilmelidir.

2- n>M 'nin tüm değerleri için M^e ve C^d 'yi hesaplamak kolay olmalıdır.

3- Verilen e ve n ile d'nin ne olduğunun tespit edilmesi mümkün olmamalıdır.

Şuanda ilk ihtiyacı düşünelim.

$M^{ed} = M \bmod n$

p ve q iki asal sayı olsun. n ve m iki tamsayı olmak üzere n=p.q ve 0<m<n olduğu durumda, keyfi seçilmiş bir k tamsayısı aşağıdaki gibi bir denklem oluşturur.

$$m^{k\Phi(n)+1} = m^{k(p-1)(q-1)+1} \equiv m \bmod n$$

$\Phi(n)$ sayısı n'den daha küçük pozitif bir tam sayıdır ve n ile kendi aralarında asaldır. p ve q asal sayı olmak üzere $\Phi(pq)= (p-1)(q-1)$ 'dir.

Böylece, eğer $e.d=k.\Phi(n)+1$ olursa aşağıdaki denkliklerden bahsedebiliriz.

$$ed \equiv 1 \bmod \Phi(n)$$

$$d \equiv e^{-1} \bmod \Phi(n)$$

e ve d , mod $\Phi(n)$'de çarpmaya göre ters elemandır.

Dikkat etmemiz gereken bir husus da, modüler aritmetiğin kurallarına göre eğer d (ve sonucunda e'de) ile $\Phi(n)$ aralarında asal olması durumunda bu gerçekleşir. Bu durumda, OBEB($\Phi(n)$,d)=1'dir.

p,q iki asal sayı (özel , seçilmiş)

n=p.q (Açık , hesaplanmış)

e , OBEB($\Phi(n)$,e) =1 ; 1<e< $\Phi(n)$ (Açık , seçilmiş)

d, $e^{-1} \bmod \Phi(n)$ (özel , hesaplanmış)

{d,n} çiftinden özel anahtar, {e,n} çiftinden açık anahtar oluşur. A kullanıcısının, açık anahtarını yayınladığını varsayalım. B kullanıcısı, A kullanıcısına mesaj M'yi göndermek istiyor. B kullanıcısı, $C=M^e \pmod n$'yi hesaplar ve C'yi gönderir. Şifreli mesaj alındıktan sonra A kullanıcısı tarafından $M=C^d \pmod n$ hesaplanarak şifre çözülür.

e ve d 'yi aşağıdaki denkliği sağlayacak şekilde seçtik.

$d \equiv e^{-1} \bmod \Phi(n)$

e ve d'nin mod $\Phi(n)$ 'de çarpmaya göre ters eleman olmasından dolayı;

$e.d \equiv 1 \bmod \Phi(n)$

Bu yüzden e.d , $k\Phi(n)+1$ 'in bir şeklidir. İki asal sayı olan p ve q, birer tamsayı olan n (n = p.q) ve M (0 < m < n) alınarak ispatlanmıştır.

$M^{k.\Phi(n)+1} = M^{k.(p-1).(q-1)} \equiv M \bmod n$

Bundan dolayı $M^{ed} \equiv M \bmod n$;

$C = M^e \bmod n$

$M = C^d \bmod n \equiv (M^e)^d \bmod n \equiv M^{ed} \bmod n \equiv M \bmod n$

Anahtar Üretimi

1- p,q seç (p ve q ikisi de asal)

2- n = p x q

3- $\Phi(n)=(p-1)(q-1)$ hesapla

4- e tamsayı seç : OBEB($\Phi(n)$,e) =1; 1< e< $\Phi(n)$

5- d'yi hesapla : $d= e^{-1} \bmod \Phi(n)$

6- Açık Anahtar : KU={e,n} ; Özel Anahtar KR={d,n}

	Şifreleme	**Deşifreleme**
Orjinal mesaj	M<n	C
Şifreli mesaj	$C=M^e \pmod n$	$M=C^d \pmod n$

Şekil 3.3. RSA algoritması

Şekil 3.3'te RSA algoritması özetlenmektedir. Şekil 3.4'te bulunan örnek için anahtarların üretim adımları aşağıda verilmiştir.

1- İki asal sayı seçilir p=7 ve q=17

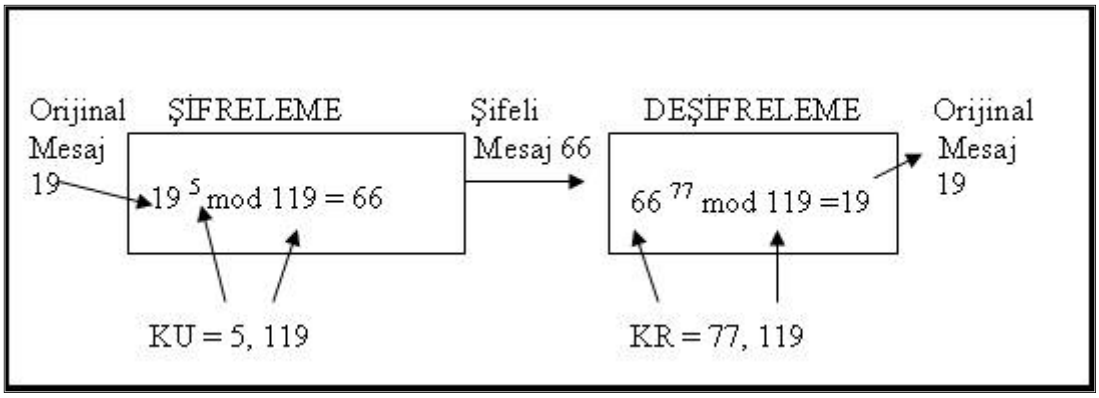
2- n = p.q = 7x17 = 119 hesaplanır.

3- $\Phi(n) = (p-1)(q-1) = 96$ hesaplanır.

4- Öyle bir e seç ki; e $\Phi(n)=96$ ya asal olsun aynı zamanda $\Phi(n)$ den küçük olsun . Bu durumda e=5'tir.

5- Öyle bir d belirleyelim ki d.e = 1 mod 96 ve d<96 olsun. Doğru değer d=77 'dir. Çünkü 77x5 = 385 = 4 x 96 + 1

6- olsun . Bu durumda e=5'tir.



Şekil 3.4. RSA algoritmasına örnek

Sonuçta anahtarlar; açık anahtar KU ={5,119} ve özel anahtar KR={77,119} olur. Şekil 3.4 deki örnek, M=19 orjinal mesajı için anahtarların kullanımını gösterir. Şifreleme için 19'un beşinci kuvveti alınır ve 2476099 sayısı elde edilir. Bu sayının 119 tarafından bölünmesi sonucunda elde edilen kalan bulunur. Kalan olarak bulunan 66 sayısı bizim şifreli mesajımızdır. $C=M^e$ (mod n) formülünü uygulayarak $C= 19^5$ mod 119 ≡ 66 mod 119 şifreli mesajımızı elde ettik. Elde edilen şifreli mesajımızı, çözmek için $M=C^d$ (mod n) formülünü uygularız. $M= 66^{77}$ mod 119 ≡ 19 mod 119 işleminin sonucunda orijinal mesajımız olan 19 sayısını elde ederiz.

Asimetrik sistemlerde başka insanlardan mesaj almak isteyen kişi, çarpanlarına ayrıldığında "gizli anahtarı" da ele verecek olan bir sayıyı içeren "açık anahtar" ı başka insanlara açıklamalıdır. Bu sistemlerde, genellikle, şifreleme anahtarı olarak açık anahtar, şifre çözme anahtarı olarak gizli anahtar kullanılmaktadır. Gizli anahtar bilindiğinde, ilgili açık anahtarla şifrelenen herhangi bir mesaj tehlikede olacaktır. Söz edilen sayının günümüz bilgisayarları ve bilgisiyle çarpanlarına ayrılması neredeyse imkansızdır. Ancak kuantum bilgisayarlar, özellikle çarpanlara ayırma işlemini oldukça kolay olarak yapabilmektedir. Bu nedenle bugün, açık anahtarlı kripto sistemlerin güvenliği, gerçek bir kuantum bilgisayarının yapılamamış olduğu ve ilerde de yapılamayacağı varsayımına bağlıdır. Ama teknolojik gelişmedeki hız göz önüne alınırsa bu oldukça riskli bir varsayımdır [39].

3.3.2. Gizli anahtarlı şifreleme sistemleri (Simetrik şifreleyiciler)

Gizli anahtar algoritmaları hem şifreleme hem de şifre çözmek için aynı anahtarı kullanırlar (veya biri diğerinden türetilebilir). Bu, veri şifreleme için daha kestirme, matematiksel açıdan daha az problem çıkaran bir yaklaşımdır ve uzun zamandan beri kullanılmaktadır.

3.3.2.1. Vigenere şifresi

m sabit pozitif bir tamsayı olsun. $P = C = \kappa = (Z_{28})^m$ tanımlansın.

Bir $K= (k_1,k_2,....,k_m)$ anahtarı için,

$e_k(x_1,x_2,....,x_m) = (x_1+k_1, x_2+k_2,....., x_m+k_m)$ ve

$d_k(y_1,y_2,....,y_m) = (y_1-k_1, y_2-k_2,....., y_m-k_m)$ tanımlanır.

Şekil 3.5. Vigenere şifresi

Kaydırma şifresinde bir anahtar seçildikten sonra her alfabetik karakter tek bir alfabetik karaktere haritalanır. Bu nedenle, bu şifreleme yöntemlerine monoalfabetik denir. Şekil 3.5'da Vigenere Şifresi olarak adlandırılan monoalfabetik olmayan bir şifreleme yöntemi gösterilmiştir.

Daha önceden tanımladığımız A -> 0, B -> 1, ..., Z -> 25 bağlantıları kullanılarak, her K anahtarını; anahtar kelime (keyword) denilen m uzunluklu bir alfabetik dizi ile bağlayabiliriz. Vigenere Şifresi, her mesaj elemanı m alfabetik karaktere eşit olduğu zaman bu karakterleri şifreler.

Örnek :
m = 6 ve anahtar kelime de CIPHER olsun. Bunun sayısal eşdeğeri,
K = (2,8,15,7,4,17)'dir. Mesaj'ın aşağıdaki diziden oluştuğunu varsayalım:

Thiscryptosystemisnotsecure.

Bu mesajın elemanlarını aşağıdaki gibi mod 26'ya göre 6'lı gruplar halinde yazıp, anahtar kelimeyi ekleriz:

19	7	8	18	2	17	24	15	19	14	18	24
2	8	15	7	4	17	2	8	15	7	4	17
21	15	23	25	6	8	0	23	8	21	22	15

18	19	4	12	8	18	13	14	19	18	4	2
2	8	15	7	4	17	2	8	15	7	4	17
20	1	19	19	12	9	15	22	8	25	8	19

20	17	4
2	8	5
22	25	19

Bu alfabetik eşitliğin şifreli dizisi :

21	17	23	25	6	8	0	23	8	21	22	15	20	1	19	19	12	9	15	22	8
V	P	X	Z	G	I	A	X	I	V	W	P	U	B	T	T	M	J	P	W	I
25	8	19	22	25	19															
Z	I	T	W	Z	T															

Şifre çözmek için, aynı anahtar kelimeyi kullanırız ancak, toplama yerine mod 26'a göre çıkarma yaparız. Vigenere şifresinde, m uzunluğundaki anahtar kelimelerin sayısı $26^{m'}$dir. m uzunluğunda anahtar kelimeye sahip bir Vigenere şifresinde, bir alfabetik karakter; m alfabetik karakterlerden birine haritalanabilir. (anahtar kelimenin m farklı karakter içerdiğini varsayarsak.) Böyle bir şifreleme sistemine polialfabetik denir. Genel olarak, şifre çözme polialfabetik sistemlerde, monoalfabetik sistemlerde olduğundan daha zordur.

3.3.2.2. One Time Pad (OTP)

1917'de Joseph Mauborgne ve Gilbert Vernam gizli anahtarlı şifreleme sistemi olan "one-time pad"i buldular. 1917 de G. Vernam tarafından keşfedilen Vernam şifreleyicisi OTP'nin iyi bir örneğidir. Bu şifreleyici oldukça basittir. Düzmetin mesajını içeren bit dizgesi alınır ve mesajın uzunluğuyla aynı uzunlukta bir anahtar ile kullanılır. Ancak anahtarın bir bölümü asla ikinci kez kullanılmamalıdır(aksi halde şifreleyici kırılabilir). Bundan sonra şifrelenecek mesajın uzunluğunda tam rasgele bir anahtar dizisi seçilerek mesaj ve anahtara XOR işlemi uygulanır.

ÖRNEK :

Gönderilecek Mesaj = "Ardımda kalan yerler anlaşıyor baharla" olsun.

Bu ifadenin ASCII tablosundaki sayısal karşılığı :

65 114 100 253 109 100 97 32 107 97 108 97 110 32 121 101
114 108 101 114 32 97 110 108 97 254 253 121 111 114 32 98
97 104 97 114 108 97

şeklindedir.

Anahtar (key) = B8DBKMKKV8RVB45DM3F5P7*HNDL981N!H94/FL olarak belirlenmiş olsun.

ANAHTAR'in ASCII tablosundaki sayısal karşılığı :

66 56 68 66 75 77 75 75 86 56 82 86
66 52 53 68 77 51 70 53 80 55 42 72
78 68 76 57 56 49 78 33 72 57 52 47
70 76

şeklindedir.

Mesaj, anahtar ile özelVeya(XOR) işlemine tabi tutulur ve Şifreli Metin elde edilir. Buna göre :

Şifrelenmiş Metin'in ASCII tablosundaki Sayısal karşılığı :

3	74	32	191	38	41	42	107	61	89	62	55
44	20	76	33	63	95	35	71	112	86	68	36
47	186	177	64	87	67	110	67	41	81	85	93
42	45										

Şifreli Mesajın Karaktersel Karşılığı :

♥J ¬ &)*k=Y>7,¶L!?_#GpVD$/│▒@WCnC)QU]*-

Şeklinde yer alır. Alıcı, şifreli mesaj kendisine ulaştıktan sonra ANAHTAR ile tekrar XOR işlemi gerçekleştirerek asıl mesajı elde eder. Şimdi şifreli mesajın göndericiden alıcıya giderken yolda üçüncü şahısların eline geçtiğini düşünelim. ANAHTAR bilinmediğinden dolayı üçünücü şahıslar mesajın uzunluğu kadar farklı karakterlerden oluşacak ANAHTAR tahmin etmeli ve bunu şifreli mesaja uygulamalıdırlar. ASCII tablosunda 256 farklı karakter yer almaktadır ve mesajın uzunluğu 38 olduğu için ortaya çıkacak olası ANAHTAR sayısı 38^{256}'dir. Bu da 2,658280061817465806594588029521 9e+404 olası ANAHTAR var demektir. Ayrıca 3.kişiler bu tahminleri yaparken şifreli mesajla yanlış Anahtarları XOR işlemine tabi tutacak olurlarsa yine de birden faz anlamlı mesaj elde edeceklerdir.
Örneğin : yukarıda sayısını vermeye çalıştığımız olası anahtarlar içinde aşağıdaki anahtar da mevcuttur.

Yanlış ANAHTAR'ın ASCII Tablosundaki Sayısal Karşılığı

65	35	82	159	66	76	88	14	95	60	71	94
12	115	37	67	86	127	72	50	2	35	40	73
90	68	145	37	36	40	7	99	65	48	59	49
75	95	ise									

Yanlış ANAHTAR ile Şifreli Mesajı XOR işlemine tabi tutarsak aşağıdaki şekilde yanlış metin elde edilecektir.

Yanlış Mesaj = “Bir derebeyi gibi kurulmuş eski hanlar”

Yukarıda da görüldüğü gibi 3.şahıslar alıcı ve verici program algoritmalarına, şifrelenmiş mesaja sahip olsalar bile gerçek ANAHTAR’ı elde edemedikleri takdirde asıl mesaja ulaşmayacaklardır.

Sistemin üstünlükleri

Uzunluğu n bit olan bir mesaj için n bitlik bir anahtar dizisi seçilir. Mesaj şifrelenir ve gönderilir. Mesajı ele geçiren birisi olası bütün anahtarları (2^n tane) denese bile mesajı bulamaz. Çünkü bu işlemin sonunda n bitlik bütün kelimeleri bulur. Elinde birden fazla anlamlı mesaj olacağı için bu mesajların içinden gerçek mesajı tahmin etmek imkansızdır. Bu açıdan koşulsuz güvenli bir sistemdir.

Günümüz bilgisayarlarıyla yapılması çok zor, hatta neredeyse imkansız kabul edilen, çok büyük tamsayıların asal çarpanlarına ayrılması ve ayrık logaritma işlemlerini kuantum bilgisayarlar kolaylıkla ve verimli olarak yapabileceklerdir. Dolayısıyla, tahmin edileceği üzere, kuantum bilgisayarlar yapıldığında günümüzün güvenli kabul edilen açık anahtarlı şifreleme sistemlerinin güvenlikleri tehlikeye girecektir [39]. OTP gizli anahtarlı ve güvenilir bir çözüm sunma özelliğiyle açık anahtarlı şifreleme sistemlerine üstünlük sağlamaktadır.

Sistemin zayıflıkları

Uzun bir mesajı şifrelemek için uzun bir anahtar üretmek gerekir. Bu sistem tam rasgele bir anahtar dizisi kullanıldığından, uzun bir anahtar üretmek, bu anahtarı güvenli bir şekilde karşı tarafa iletmek ve saklamak zor olur. Ayrıca kullanılan anahtar tekrar kullanılamayacağı için, her seferinde başka bir anahtar üretilmesi gerekir. Bu nedenlerden dolayı sistemin kullanımı zordur.

Çözüm önerileri

Kuantum şifreleme tekniği yukarıda bahsedilen %100 güvenliği şimdilik sağlamaktadır. Yani alıcı ve verici arasındaki anahtar değişim kuralını güvenli hale getirir. Kuantum şifreleme tekniği temel bir fizik kanunu olan Heisenberg'in belirsizlik ilkesine dayanmaktadır. Bu ilkeye göre kuantum mekaniğinin temel öğesi olan bir fotonun aynı anda iki özelliği bilinemez. Bu da iletişim kanalındaki bir fotonun kopyalanmasını(klonlanmasını) imkansız hale getirmektedir. Günümüz teknolojisinde fiber optik ağ üzerindeki bir fotonun yeni bir kopyası çıkarılamaz. İşte kuantum kripto tekniği fotonun bu özelliğinden faydalanarak güvenli bir anahtar iletimi sağlar [27].

Yukarıda bahsettiğimiz Kuantum şifreleme tekniğinde şifreleme ve şifre çözme işlemleri için kullanılan ANAHTAR'ın gönderiminde fiber optik kablolama kullanılmıştı. Ancak bu sistemin zayıflığı her yerde kullanılamamasıdır. Bu tezde gerçekleştirdiğimiz uygulama OTP şifreleme algoritması için kullanılan şifreleme ve şifre çözme ANAHTAR'ını program içerisinde sadece Alıcı/Verici (Client / Server) programlarının bileceği, 1. ve 2. şahısların dahi o an için bilemeyecekleri eş zamanlı kullanılan anahtarlara dönüşümünü gerçekleştiriyor. Böylelikle başlangıçta iki tarafında bildiği ANAHTAR programın çalışması esnasında sürekli değişime uğruyor. Bu işlem hem aynı anahtarın birden fazla kez kullanımı hem de ANAHTAR'ın güvenli bir yoldan gönderilme zorunluluğunun ortaya koyduğu sorunun aşılmasını sağlıyor.

3.4. Damgalama ve Sırörtme

Sırörtme tekniklerinin ticari kullanımı yavaş yavaş sayısal "filigran"ın (watermarking) gelişmesini sağlamaktadır. Burada söz konusu olan gizli bilginin insan duyularından gizlenmesidir. 1990'ların başında imge filigrasyonu (damgalama) kavramı gelişmiş; Tanaka ve arkadaşları faks gibi ikili imgelerin korunması kavramını ortaya atmışlardır [35, 36]. 1993 yılında Tirkel ve arkadaşları gerçekleştirdikleri uygulamaya; daha sonra "watermark" olarak birleştirilecek "water mark" ismini vermişlerdir [6].

Steganografi iki parçadan oluşan Yunanca bir kelimedir. “Steganos” örtülü/gizli, “grafi”de yazım/çizim anlamına gelmektedir. Örtülü yazma sanatı olarak çevrilen “stego” aslında antik Yunan ve Herodot zamanına kadar uzanan derin bir geçmişe sahiptir. Herodot bu konuda birkaç olay anlatmaktadır. Örneğin, M.Ö. 5. yüzyılda, Yunan tiran Histiaeus'un, Susa Kralı Darius'un krallığında göz hapsine alındığı sırada, bir Anadolu şehri olan Milet'te yaşayan damadı Aristagoras'a gizli bir mesaj göndermek istemesiyle ilgilidir. Histiaeus, kölelerden birinin saçını kazıtır ve mesajı dövme şeklinde kölenin kafa derisine işler. Kölenin saçı yeteri kadar uzadığında, köle, Milet'e gönderilir. Köle yanında hiçbir şey götürmediği için Kral Darius bundan şüphelenmez. Köle oraya vardığında durumu anlatır ve saçları tekrar kazıtılan kölenin kafa derisinden Histiaeus'un mesajını içeren dövmesi ortaya çıkar.

Diğer örnek uygulamalar ise; odunların üzerine asitle yazılan mesajları balmumuyla gizlemek (Demaratus’un Spartalılar’ı uyardığı hikaye) ve mesajları tavşanların midesine kazımak gibi yöntemlerin kullanıldığı olaylardır. Eski Romalılar birbirleri arasında, meyve suyu veya süt gibi sıvılardan oluşturulan görünmez mürekkepler kullanarak yazışırlardı. Bu yazışma, gelişme göstererek günümüze kadar gelebilmiştir. Rönesans döneminde Johannes Trithemius'un Şifreleme ile ilgili kitapları üçleme olarak basılmıştır. Trithemius'un sırörtme metodu birbirini izleyen sütunlardaki kelimelerin ilk harflerini birleştirmeye dayalıdır; ve bir nevi akrostiş uygulamasıdır. “Steganographia” isimli yazısıyla terim geçerlilik kazanmış ve yaygın olarak o dönemde kullanılmaya başlanmıştır.

Bir takım gizli mesajların yetkilendirme prensibinden hareketle yalnızca ilgililer tarafından okunması, diğer kişiler tarafından ise ya “gizlenmiş veriden” haberdar olmaması ya da haberi olsa dahi gömülü bilgiyi elde edememesi arzu edilir. Bu maksatla veri gizleme (data hiding) teknikleri bünyesinde “steganografi” bilim dalı kullanılmaktadır [25]. Neticede bu sanat bugün; insanlığa, bilgilerin gizlice iletilmesi konusunda çağlar boyu yardımı dokunmuş bir bilime dönüşmüştür. Modern sırörtme teknik olarak, bir veriyi (mesaj) bir nesnenin içine gizli biçimde yerleştirmeyi esas alır. Öyle ki, sadece belirlenen alıcı kendine iletilmek istenen mesajı nesneden alır ve diğer gözlemcilerin o nesnenin içindeki mesajın varlığından haberleri olmaz. Şifreleme biliminin bir kolu olarak görülen sırörtme bu özelliğiyle onu bir adım ileri

taşır. Şifreleme algoritmaları güvenilirliği sağlasa da bir bakıma mesajın gizliliğini sağlamaz. Şifreleme uygulamalarında bilgi sadece gönderen ve alanın anlayabileceği şekilde şifrelenirken, sırörtme uygulamalarında bilgi sadece gönderen ve alanın varlığını bildiği şekilde saklanır, bazen de şifrelenip çift kat koruma sağlanır. Veriler genelde metin ve resim; taşıyıcı nesneler ise metin, ses, resim ve video görüntüleri olabilir.

Sırörtme uygulamaları iki temel prensip üzerine kurulmuştur. Bunlardan ilki sayısal hale getirilmiş resim veya ses dosyalarının, diğer türlerden farklı olarak, sahip oldukları fonksiyonlarını yitirmeden değiştirilebilmeleri ilkesidir. İkincisi ise, insanın, renk veya ses kalitesinde meydana gelen küçük değişiklikleri ayırt edememesine dayanmaktadır. Bunun mantığı da gereksiz bilgiler taşıyan nesnelerin içindeki bilgileri, başka bilgi parçacıklarıyla yer değiştirmektir.

3.4.1. Gizli bilginin araştırılması (Sıraçma)

Sırörtme tekniğinin amacı gizli bir mesajın veya bir gömü verisinin, şüphelerden sakınarak transferinin gerçekleştirilmesidir. Eğer kuşkular artarsa, gizli mesajın ortaya çıkarılması kaçınılmaz olur. Bu mesajların keşfedilerek faydasız hale getirilmesi sanatı literatürde stego-analiz (steganalysis) sanatı olarak bilinmektedir ve bu sanatın gelişmesi amacıyla çeşitli algoritmalar geliştirilmektedir.

Sunulan bu tez çalışmasında sırörtme tekniğinin uygulandığı ses paketleri içerisindeki gömü verisi/dosyası geliştirilen kod çözücü algoritma ile elde edilmekte ve kullanıcıya bilgi verilmektedir. Sırörtme uygulamasında üçüncü kişilerin gömü verisini elde etme (sıraçma) işlemini yapamaması için, verinin ne şekilde gömüldüğünün gizli tutulması, gömü verilerinin güvenliği açısından önem arz etmektedir.

3.4.2. Sırörtme metotları

Mesajların örtü verisi içerisine ne şekilde yerleştirildiği çok büyük önem arz eder. Gömü verisinin/dosyasının hangi bitlere yerleştirildiği, hangi veri bloklarının içerisine konumlandığı, şifreleme yapılıp yapılmadığı gibi parametreler sıraçmanın ilgi alanına girer.

Internet, haberleşmenin artan geniş bandında bilginin kitlelere dağıtılma vasıtası olarak kullanılmaktadır. Böyle bilgiler, kitle haberleşmesini sağlamak üzere metin, resim ve ses dosyalarını kapsamaktadır. Bu uygulamalarda gizli bilginin taşınması bir çok farklı teknikle ve mükemmel taşıyıcılarla mümkün olabilir. Diğer taşıyıcılar gizli bilgi için depolama cihazları ve TCP/IP paketleri içerirler. İlk yaklaşım metin içerisinde bilginin gizlenmesi olacaktır. Bilgisayarlar bilgi gizlemede daha fazla kapasite imkanı sağlamaktadır.

Sunulan bu tez çalışmasında, ses verilerini TCP/IP paketleri şeklinde ağ üzerindeki diğer bilgisayara gönderildiğinden, uygulama yerel alan ağında çalışabildiği gibi, Internet üzerinden de çalışabilmektedir. Bu durum ilgili uygulamanın kullanım alanının genişlemesi anlamına da gelmektedir.

Bir belgenin yerleşim planı bilgiyi açığa çıkarır. Belgeler, kelimeler ve çizgilerin pozisyonlarının modülasyonu ile işaretlenerek tanımlanabilir. Boşlukların eklenmesi ve görünmeyen karakterler gizli bilginin geçişine bir metot oluşturur. Görülecek ilginç bir yol bir HTML dosyasına ekstra kesme çizgileri ve boşluklar eklemektir. Web listeleyici bu ekstra çizgi ve boşlukları göz ardı eder, ancak web sayfasının kaynağı açığa çıkarılarak ekstra karakterler gösterilir. Bilginin metin içerisinde gizlenmesi için bir çok metot vardır. Bu metotlar en küçük değerlikli bit (Least Significant Bit, LSB) veya gürültü ekleme, resmin işlenmesi ve sıkıştırma algoritmaları ve parlaklık gibi resim özelliklerinin değiştirilmesi metotlarıdır. Diğer algoritmaların zaafından ve resim işleme veya onun bileşenlerinde bilgi gizleme katsayılarından yararlanarak resim içinde bilgi gizlemenin metotlarını daha güçlü yapmaktadır. Bu metotlar mesajları resmin belirgin alanlarına gizlerler ve sıkıştırma,

kesme ve bazı resim işleme saldırılarına karşı LSB yaklaşımından daha güçlü kılarlar [25].

Bu tez çalışmasının da temelini teşkil eden sayısal ses verilerine sırörtme, teorik ve pratik olarak mümkündür. Çünkü ses içerisine küçük yankılar veya kulak tarafından algılanamayan sinyaller eklenebilir ve bunlar daha yüksek genlikte bir ses bileşeni tarafından maskelenebilir [28, 29].

3.5. Sonuç

Gizliliğin öneminin had safhaya ulaştığı uygulamalarda; gizli bilgilerin, üçüncü kişilerin eline geçmeden ilgili hedefe gönderilmesi amaçlanmaktadır. Ancak herhangi bir nedenden ötürü gönderilecek olan bilginin üçüncü şahısların eline geçse dahi bu kişilerin bu bilgilere ulaşması istenmemektedir. Buda karşımıza şifreleme bilimini çıkarmaktadır.

İyi şifreleme sistemleri öyle tasarlanmalıdırlar ki bunları kırmak, mümkün olduğu kadar zor olsun. Bir sistem tasarımcısı için sistemi kırılabilir bırakmanın hiç bir özrü yoktur. Güvenliği aşmak için kullanılabilecek her mekanizma açığa çıkarılmalı, belgelenmeli ve son kullanıcının dikkatine sunulmalıdır

OTP şifreleme algoritması için kullanılan şifreleme ve şifre çözme anahtarını program içerisinde sadece Alıcı/Verici (Client / Server) programlarının bileceği, 1. ve 2. şahısların dahi o an için tahmin edemeyecekleri eş zamanlı kullanılan anahtarlara dönüşümünü gerçekleştiriyor. Böylelikle başlangıçta iki tarafında bildiği ANAHTAR programın çalışması esnasında sürekli değişime uğruyor. Bu işlem her veri kümesi için ayrı anahtar kullanımını sağlıyor. Böylelikle anahtarın güvenli bir yoldan iletilmesi zorluğu aşılmış oluyor.

Şifreleme bilimi güvenilirliği sağlasa da bir bakıma mesajın gizliliğini sağlamaz. Şifreleme algoritmalarının kullanıldığı uygulamalarda bilgi sadece gönderen ve alanın anlayabileceği şekilde şifrelenirken, sırörtme uygulamalarında bilgi sadece gönderen ve alanın varlığını bildiği şekilde saklanır, bazen de şifrelenip çift kat

koruma sağlanır. Gizli veriler genelde metin ve resim; taşıyıcı nesneler ise metin, ses, resim ve video görüntüleri olabilir.

Yapılan bu tez çalışmasında, sırörtme uygulaması gerçekleştirmek üzere, gizli bilgileri belli bir düzene göre sayısal ses içerisine gömen ses gönderici yazılımlar ve bu gizli bilgileri gelen sayısal ses verilerinden ayrıştıran ses alıcı yazılımlar geliştirilmiş olup, bu yazılımların detayları ilerleyen bölümlerde sunulmaktadır.

BÖLÜM 4. SIKIŞTIRMA TEKNOLOJİLERİ

4.1. Giriş

Diskler ve teypler gibi saklama birimleri üzerinde saklanan, ya da bilgisayar haberleşme hatlarından iletilen veriler, önemli ölçüde artıklık içerirler. Veri sıkıştırma algoritmalarının amacı, bu artıklıkları kodlayarak bilgi kaybı olmaksızın, veri yoğunluğunu artırabilmektir. Dört tür artıklık türü mevcuttur [30].

4.2. Veri Artıklık Türleri

4.2.1. Karakter dağılımı

Herhangi bir karakter katarında, bazı karakterler diğerlerine göre daha sık kullanılırlar. Özellikle sekiz bitlik ASCII kodlarının kullanıldığı özel bir dosyada, karakterlerin 3/4 ü kullanılmayabilir. Sonuçta her bir sekiz bitlik paketin, iki bitinden tasarruf edilmiş olur. Bir envanter kaydında, sayısal değerler çok daha yaygındır (ikili ya da ondalık sayılar istatistiği değiştirebilir) ve kendilerine ayrılmış alandaki sınırlamalar, dosyadan dosyaya, önemli ölçüde değişebilecek karakter dağılımına neden olabilir. Örneğin, ambar yerlerinin adreslenmesi için alfabetik veya sayısal değerlerin kullanılması, envanter dosyasındaki dağılımı değiştirebilir. Benzer şekilde, envanter kaydında yer alan açıklayıcı metinler, her karakter için gereken ortalama bit sayısını etkileyecektir.

4.2.2. Karakter tekrarı

Eğer bir karakter katarı, tek bir karakterin tekrarından oluşuyor ise veri, olduğundan daha yoğun bir şekilde kodlanabilir. Bu tür katarlar metin tipi dosyalarda fazla yer

almamaktadır. Bununla beraber, formatlı iş dosyalarında kullanılmayan alanlar oldukça fazla miktarda yer almaktadır. Bir envanter kaydında, kısmen kullanılan alfabetik alanlardaki boşluk katarlarına, sayısal alanlarda sıfır katarlarına ve kullanılmayan alanlarda null katarlarına oldukça sık rastlanır. Grafik görüntüler, özellikle iş grafiklerindeki çizgiler çoğunlukla homojen boşluklardan oluşmuştur.

4.2.3. Çok kullanılan sözcükler

Belli karakter dizileri, diğerlerine göre daha fazla sıklıkta kullanılırlar ve bu nedenle, olması gerekenden daha az bit sayısı ile temsil edilebilirler. Örneğin, ingilizcede ZE gibi pekçok karakter çiftinin oluşma olasılığı, tek harfin oluşma olasılığından fazladır ve daha az bit ile kodlanabilirler. Benzer şekilde, GC gibi oluşma olasılığı az olan karakter çiftlerinin, daha uzun bit kombinasyonları ile kodlanması, bit kullanımını iyileştirecektir.

Bazı tip dosyalarda olduğu gibi (programlama dilleri kaynak programlarını içeren dosyalar, metin dosyaları, v.b.) belli anahtar kelimeler diğerlerinden fazla miktarda yer alır. Örneğin, bu tezde "sıkıştırma” ve “şifreleme” sözcükleri sıklıkla kullanılmaktadır. Envanter kayıtlarında, depo isimleri gibi belli alanlar tekrar tekrar kullanılır. Sayısal alanlar ise, sadece sayısal olan ve hiçbir harf, özel işaret içermeyen dizilerden oluşur. Eğer bir metin bu tip bîr artıklık içeriyor ise, 8 bitlik kodların kullanılması yerine, 4 veya daha az bitlik kodların kullanılması ile temsil edilebilir.

4.2.4. Konumsal artıklık

Eğer belli karakterler, her bir veri bloğu içinde, önceden tahmin edilebilir yerlerde bulunuyorsa, bunlar kısmi olarak artıklık taşımaktadır. Dikey bir çizgi içeren bir resimde, çizgi her taramada aynı yerde görüneceğinden, daha az bit dizileri ile kodlanabilir. Bir envanter dosyada belli kayıtlar, hemen hemen aynı değişkenlere sahiptir. Diğer taraftan metin tipi dosyalar durumsal artıklık içermezler.

Bu dört tip artıklık bazı durumlarda çakışmaktadır. Örneğin, içinde sıfırların fazla miktarda yer aldığı tamsayılardan oluşan bir kayıt, ilk üç tip artıklık türünü de içermektedir.

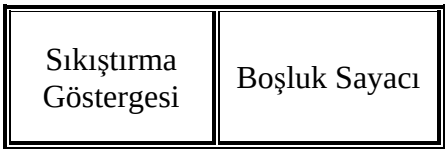
4.3. Statik Sıkıştırma Algoritmaları

Bu tip sıkıştırma algoritmaları, sıkıştırılacak verinin önceden bilinmesini gerektirmektedir. Verilerin hangi tür artıklık içereceği önceden bilineceğinden, sıkıştırma işlemi buna göre yapılır [31].

Boşluk Sıkıştırma

Boşluk sıkıştırma geliştirilen en eski yöntemlerden biridir. İsminden de anlaşılacağı gibi, boşluk sıkıştırma, veriler içindeki boşluk ve null karakterlerini bularak, bu karakterleri, kodlama yoluyla sıkıştıran bir veri sıkıştırma algoritmasıdır. Şekil 4.1'den de formatı görüleceği gibi boşluk ve null karakterlerinin yerine özel sıralı karakter çiftleri yerleştirilir.

Bu karakterlerden birincisi, sıkıştırmanın başladığını gösteren özel bir karakter, ikincisi ise boşluk karakterlerinin sayısını gösteren bir sayaçtır. Bu sayacın büyüklüğü, verinin özelliğine göre seçilebilir. Örneğin sekiz bitlik bir sayaç kullanılıyor ise, en fazla peş peşe yer alan 256 boşluk karakteri kodlanabilir (Sıfır değeri 1 olarak sayılırsa).



Şekil 4.1. Boşluk sıkıştırma formatı

Bu iki karakter katarı veri dizisi içinde iletildiğinde, alıcı taraf boşluk sıkıştırmanın başladığını gösteren özel karakteri arar. Bu karakterin bulunmasıyla, alıcı bir sonraki karakteri sıkıştırılmış boşluk karakterlerinin sayısı olarak değerlendirir. Bu bilgiden

yararlanarak orijinal veri alıcı tarafta tekrar elde edilmiş olur. Şekil 4.2'de bu sıkıştırma yöntemine bir örnek görülmektedir.

Kodlanacak veri dizisi : ABbbbbbDEF

Kodlanmış veri dizisi : A B S_C 5 D E F

S_C : Sıkıştırma Göstergesi

B : Boşluk Karakteri

Şekil 4.2. Boşluk sıkıştırma örneği

Bu yöntemle veri dizi içinde 3 ya da daha fazla boşluk karakteri tekrarlanmıyorsa, hiçbir tasarruf sağlanamayacaktır. Bu da algoritmaya bir sınırlama getirmektedir.

Yarım Sekizli Paketleme (Half-Byte Packing)

Yarım sekizli paketleme, sayısal karakterler ile bazı özel karakterlerin yer aldığı karakter dizilerinin sıkıştırılmasında kullanılmaktadır. EBCDIC karakter kümesinde Tablo 4.1' de gösterildiği gibi sayısal karakterlerin ilk dört bitleri aynı değere sahiptir

Tablo 4.1. EBCDIC sayısal karakter kodları

Bit yapısı	Sayısal karakter
1111 0000	0
1111 0001	1
1111 0010	2
1111 0011	3
1111 0100	4
1111 0101	5
1111 0110	8
1111 0111	7
1111 1000	8
1111 1001	9

Eğer veri katarları sayısal EBCDIC kodlarından oluşuyorsa ve bu katarlar karakter tekrarı içermiyorlarsa, karakter tekrarı kodlama yöntemiyle sıkıştırılamazlar. Oysa karakterlerin ilk dört bitleri tekrarlandığından iki sayısal karakter tek bir karakter olarak paketlenirse sıkıştırma sağlanmış olur.

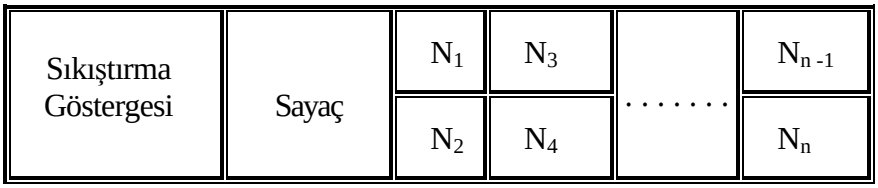
Veri karakterlerinin sayısal karakterler yanında bazı özel karakter içermesi durumunda da yarım sekizli paketleme başarım sağlamaktadır. Tablo 4.2' den de görüleceği gibi ASCII tipi kodlarda sayısal karakterlerin yanı sıra asteriks, dolar işareti, parantez işaretleri gibi özel karakterin de ilk dört bitleri aynıdır. Bu sayısal ve özel karakterlerin birlikte kullanıldığı mali kayıtlarda yarım sekizli paketleme iyi bir sıkıştırma başarımı sağlamaktadır.

Tablo 4.2. Yarım sekizli paketlemeye uygun ASCII tablosu

Bit Yapısı	**Karakter**
0011 0000	0
0011 0001	1
0011 0010	2
0011 0011	3
0011 0100	4
0011 0101	5
0011 0110	8
0011 0111	7
0011 1000	8
0011 1001	9
0010 1101	-
0010 0101	%
0010 0110	&
0010 1010	*
0010 1110	.

Tek sembol zinciri kodlamasındakine benzer şekilde burada da sıkıştırmanın başladığını göstermek için veriler içinde kullanılmayan özel bir karakter kullanılır. Daha sonra sırasıyla karakter sayısını gösteren bir sayaç ve paketlenmiş veri dizileri yer alır. Şekil 4.3'de format yapısı görülmektedir.

N_j, N_2, N_3,…., N_n paketlenmiş 4'er bitlik verileri göstermektedir. Formattan da anlaşılacağı gibi, sıkıştırmanın gerçeklenebilmesi için 5 ya da daha fazla sayıda, yukarıda anlatılan özelliklere uyan karakterlerin tekrarlanması gerekmektedir.



Şekil 4.3. Yarım sekizli paketleme formatı

Şekil 4.4'de bu sıkıştırma yöntemi kullanılarak gerçekleştirilen bir örnek verilmektedir.

Orijinal Veri Katarı : 5 9 3 0 7 1 8

Sıkıştırılmış Veri Katarı :

xxxxxxxx 00000111 0101 1001 0011 0000 0111 0001 1000

Özel Karakter : xxxxxxxx Sayaç : 00000111

Şekil 4.4. Yarım sekizli paketleme örneği

4.4. Adaptif Sıkıştırma Algoritmaları

Sıkıştırma işlemleri için sabit tablo bilgisi yerine dosyanın yapısına uygun olarak sözcük veya karakter taraması yapan algoritmalardır.

4.4.1. Huffman sıkıştırma algoritması

Huffman algoritması; dosya içerisinde sık rastlanan karakterlerin daha az bit ile, az rastlanan bitlerin ise daha fazla bit ile gösterilimi vasıtasıyla çalışmaktadır. Böylelikle dosya tipine bağlı olarak yerden tasarruf sağlamak mümkündür. Genelde metin dosyaları için kullanılır.

Huffman tekniğinde semboller (karakterler) ASCII'de olduğu gibi sabit uzunluktaki kodlarla kodlanmazlar. Her bir sembol değişken sayıda uzunluktaki kod ile kodlanır.
Bir veri kümesini Huffman tekniği ile sıkıştırabilmek için veri kümesinde bulunan her bir sembolün ne sıklıkta tekrarlandığını bilmemiz gerekir. Örneğin bir metin dosyasını sıkıştırıyorsak her bir karakterin metin içerisinde kaç adet geçtiğini bilmemiz gerekiyor. Her bir sembolün ne sıklıkta tekrarlandığını gösteren tablo frekans tablosu olarak adlandırılmaktadır. Dolayısıyla sıkıştırma işlemine geçmeden önce frekans tablosunu çıkarmamız gerekmektedir. Bu yönteme Statik Huffman tekniği de denilmektedir. Diğer bir teknik olan Dinamik Huffman tekniğinde sıkıştırma yapmak için frekans tablosuna önceden ihtiyaç duyulmaz. Frekans tablosu her bir sembolle karşılaştıkça dinamik olarak oluşturulur. Dinamik Huffman tekniği daha çok haberleşme kanalları gibi hangi verinin geleceği önceden belli olmayan sistemlerde kullanılmaktadır. Bilgisayar sistemlerindeki dosyaları sıkıştırmak için statik huffman metodu yeterlidir. Nitekim bir dosyayı baştan sona tarayarak her bir sembolün hangi sıklıkla yer aldığını tespit edip frekans tablosunu elde etmek çok basit bir işlemdir.

Huffman sıkıştırma tekniğinde statik yöntem seçilmişse iki yaklaşım vardır. Birinci yaklaşım, metin dosyasının diline göre sabit bir frekans tablosunu kullanmaktır. Örneğin Türkçe bir metin dosyasında "a" ve "e" harflerine çok sık rastlanırken "ğ" harfine çok az rastlanır. Dolayısıyla "ğ" harfi daha fazla bitle "a" ve "e" harfi daha az bitle kodlanır. Frekans tablosunu elde etmek için kullanılan diğer bir yöntem ise metni baştan sona tarayarak her bir karakterin frekansını bulmaktır. İkinci yöntem daha gerçekçi bir çözüm üretmekle beraber metin dosyasının dilinden bağımsız bir çözüm üretmesi ile de ön plandadır. Bu yöntemin zayıf yönü ise sıkıştırılan verilerde geçen sembollerin frekansının da bir şekilde saklanma zorunluluğunun olmasıdır.

Sıkıştırılan dosyada her bir sembolün frekansı da saklanmalıdır. Bu da küçük boyutlu dosyalarda sıkıştırma yerine genişletme etkisi yaratabilir. Ancak bu durum Huffman yönteminin kullanılabilirliğini zedelemez. Nitekim küçük boyutlu dosyaların sıkıştırılması gerekli değildir. Frekans tablosunu metin dosyasını kullanarak elde ettikten sonra yapmamız gereken "Huffman Ağacını" oluşturmaktır. Huffman ağacı hangi karakterin hangi bitlerle temsil edileceğini (kodlanacağını) belirlemeye yarar.

Huffman Ağacının Oluşturulması

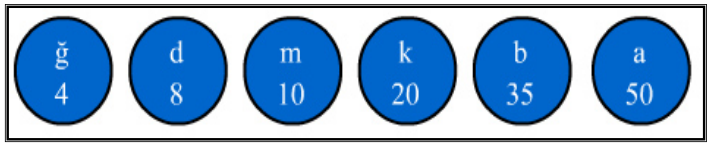
Bir huffman ağacı aşağıdaki adımlar izlenerek oluşturulabilir.Bu örnekte aşağıdaki frekans tablosu kullanılacaktır.

Tablo 4.3. Örnek metin dosyasında geçen karakterlerin frekans tablosu

Sembol(Karakter)	**Sembol Frekansı**
a	50
b	35
k	20
m	10
d	8
ğ	4

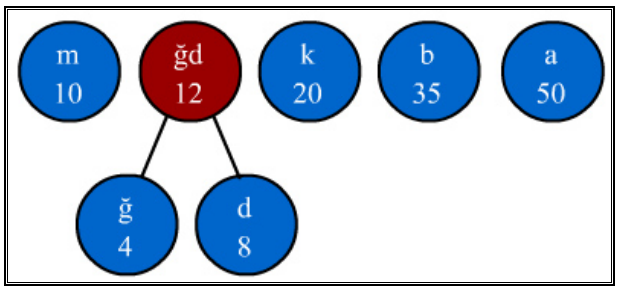
Bu tablodan, elimizdeki metin dosyasında "a" karakteri 50 defa, "b" karakteri 35 defa .... "ğ" karakteri 4 defa tekrarlanmaktadır. Amaç, her bir karakteri hangi bit dizileriyle kodlayacağımızı bulmaktır.

1 - Öncelikle "Huffman Ağacını" ndaki en son düğümleri(dal) oluşturacak bütün semboller frekanslarına göre şekil 4.5’deki gibi küçükten büyüğe doğru sıralanırlar.



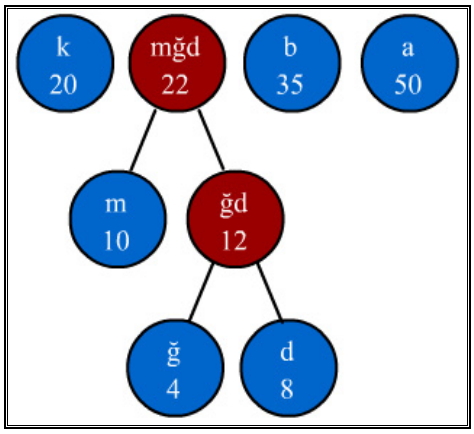
Şekil 4.5. Semboller ve takrar sayıları

2 - En küçük frekansa sahip olan iki sembolün frekansları toplanarak yeni bir düğüm oluşturulur. Ve oluşturulan bu yeni düğüm diğer varolan düğümler arasında uygun yere yerleştirilir. Bu yerleştirme frekans bakımından küçüklük ve büyüklüğe göredir. Örneğin yukarıdaki şekilde "ğ" ve "d" sembolleri toplanarak "12" frekansında yeni bir "ğd" düğümü elde edilir. "12" frekanslı bir sembol şekilde "m" ve "k" sembolleri arasında yerleştirilir. "ğ" ve "d" düğümleri ise yeni oluşturulan düğümün dalları şeklinde kalır. Yeni dizimiz aşağıdaki şekil 4.6 daki gibi olacaktır.



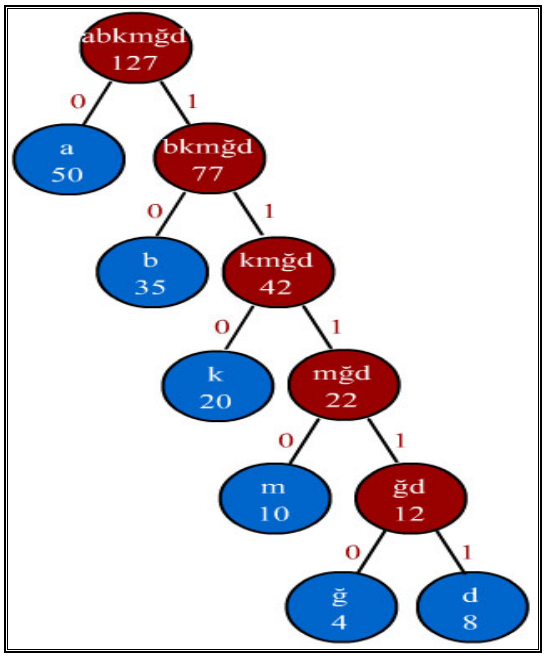
Şekil 4.6. Huffman ağacı ilk düğümü

3 - İkinci adımdaki işlem tekrarlanarak en küçük frekanslı iki düğüm tekrar toplanır ve yeni bir düğüm oluşturulur. Bu yeni düğümün frekansı 22 olacağı için "k" ve "b" düğümleri arasına yerleşecektir. Yeni dizimiz aşağıdaki şekil 4.7’deki gibi olacaktır.



Şekil 4.7. Huffman ağacı ikinci düğümü

4- İkinci adımdaki işlem en tepede tek bir düğüm kalana kadar tekrar edilir. En son kalan düğüm Huffman ağacının kök düğümü (root node) olarak adlandırılır. Son düğümün frekansı 127 olacaktır. Huffman ağacının son halini oluşturduğumuza göre her bir sembolün yeni kodu belirlenebilir. Sembol kodlarını oluştururken Huffman ağacının en tepesindeki kök düğümden başlanır. Kök düğümün sağ ve sol düğümlerine giden dala sırasıyla "0" ve "1" kodları verilir. Sırası ters yönde de olabilir. Bu tamamen seçime bağlıdır. Ancak ilk seçtiğiniz sırayı bir sonraki seçimlerde korumanız gerekmektedir. Bu durumda "a" düğümüne gelen dal "0", "bkmğd" düğümüne gelen dal "1" olarak seçilir. Bu işlem ağaçtaki tüm dallar için yapılır. Dalların kodlarla işaretlenmiş hali şekil 4.8'de gösterilmiştir.



Şekil 4.8. Huffman ağacı bit kodlaması yapılmış hali

5- Her bir sembol dalların ucunda bulunduğu için ilgili yaprağa gelene kadar dallardaki kodlar birleştirilip sembollerin kodları oluşturulur. Örneğin "a" karakterine gelene kadar yalnızca "0" dizisi ile karşılaşırız. "b" karakterine gelene kadar önce "1" dizisine sonra "0" dizisi ile karşılaşırız. Dolayısıyla "b" karakterinin yeni kodu "10" olacaktır. Bu şekilde bütün karakterlerin sembol kodları çıkarılır. Karakterlerin sembol kodları tablo 4.4'de gösterilmiştir.

Tablo 4.4. Karakterlerin belirlenen bit grubu karşılıkları

Frekans	**Sembol(Karakter)**	**Bit Sayısı**	**Huffman Kodu**
50	a	1	0
35	b	2	10
20	k	3	110
10	m	4	1110
8	d	5	11111
4	ğ	5	11110

Sıkıştırma öncesi gereken bit sayısını bulmak için : Her bir karakter eşit uzunlukta temsil edildiğinden toplam karakter sayısı (50+35+20+10+8+4) = 127 olarak bulunur. Orjinal veriyi sıkıştırmadan saklarsak 127*8 = 1016 bit gerekmektedir.

Huffman algoritmasını kullanarak sıkıştırma yapılırsa kaç bitlik bilgiye ihtiyaç duyacağımızı hesaplayalım : 50 adet "a" karakteri için 50 bit, 35 adet "b" karakteri için 70 bit, 20 adet "k" karakteri için 80 bit....4 adet "ğ" karakteri için 20 bite ihtiyaç duyarız. (bkz. Tablo 3.5) Sonuç olarak gereken toplam bit sayısı = 50*1 + 35*2 + 20*3 + 10*4 + 8*5 + 4*5 = 50 + 70 + 60 + 40 + 40 + 20 = 280 bit olacaktır.

Sonuç olarak 1016 bitlik ihtiyacımızı 280 bite indirgemiş olduk. Böylece yaklaşık olarak %72 gibi bir sıkıştırma gerçekleştirilmiş oldu. Gerçek bir sistemde sembol frekanslarını da saklamak gerektiği için sıkıştırma oranı %72'ten biraz daha az olacaktır. Bu fark, genelde sıkıştırılan veriye göre çok küçük olduğu için ihmal edilebilir.

Huffman kodunun çözülmesi

Örnekte verilen frekans tablosuna sahip bir metin içerisindeki "aabkdğmma" veri kümesinin sıkıştırılmış hali her karakter ile karakterin kodu yer değiştirilerek aşağıdaki gibi elde edilir.

a	a	b	k	d	ğ	m	m	a		
0	0	10	110	1111	11110	1110	1110	0	→	00101101111111110111011100

Eğer elimizde frekans tablosu ve sıkıştırılmış veri dizisi varsa işlemlerin tersini yaparak orjinal veriyi elde edebiliriz. Şöyle ki; sıkıştırılmış verinin ilk biti alınır. Eğer alınan bit bir kod sözcüğüne denk geliyorsa, ilgili kod sözcüğüne denk düşen karakter yerine koyulur, eğer alınan bit bir kod sözcüğü değilse sonraki bit ile birlikte ele alınır ve yeni dizinin bir kod sözcüğü olup olmadığına bakılır. Bu işlem dizinin sonuna kadar yapılır ve huffman kodu çözülür. Huffman kodları tek çözülebilir kod olduğu için bir kod dizisinden farklı semboller elde etmek olanaksızdır. Yani bir huffman kodu ancak ve ancak bir şekilde çözülebilir.

4.4.2. Zlib sıkıştırma algoritması

LZ77 ve LZ78, kayıpsız sıkıştırma algoritmalarıdır. Shannon tipi sembol tabanlı entropi sıkıştırma algoritmalarından farklı olarak harfleri ayrı ayrı sıkıştırmazlar. Bunun yerine, harflerin birbirleriyle oluşturduğu kombinasyonlara ve bu kombinasyonların sinyal içindeki tekrarlarına bakar. Dizinin sonunda tekrarlanmış olan en uzun alt diziyi bulmak amaçlanır. Uygulamada, LZ78 algoritmasının sıkıştırma sonuçları, LZ77 algoritmasınınkiler kadar iyi olmamakla birlikte, sözlük tabanlı olmasından dolayı LZ78 daha hızlı çalışmaktadır [32].

Zlib, Mark Adler ve Jean-loup Gailly tarafından yazılmıştır. ZLIB'i yazmalarının nedenlerinden biri de PNG formatındaki grafikleri sıkıştırmaktı. Zlib, deflate olarak bilinen bir algoritmayı biçimlendirmek için LZ77 ve Huffman kodlarını birleştirir [12].

ZLIB'in ara yüzü birkaç basit fonksiyonun çağrılmasıyla sınırlandırılmıştır. Sıkıştırma ve çözme süresince verilen bütün saptamalar aşağıda gösterildiği şekliyle Delphi içinde z_stream yapısı içerisinde kılıflanır [34].

```
TZStreamRec = packed record
    next_in: PChar;          // sonraki giriş byte
    avail_in: Integer;       // next_in içinde bulunan byte sayısı
    total_in: Integer;       // şimdiye kadar okunan giriş byte sayısı

    next_out: PChar;         //  Çıkışa koyulacak sonraki çıkış byte
    avail_out: Integer;      // next_out da geriye kalan boş yer
    total_out: Integer;      // şimdiye kadar okunan çıkış byte sayısı

    msg: PChar;              // en son hata mesajı,eğer hata yoksa NULL dur
    internal: Pointer;       // uygulamalar tarafından görünür değildir*/

    zalloc: TAlloc;          // içsel durumlara yer ayırmak için kullanılır
    zfree: TFree;            // içsel durumları serbest bırakmak için kullanılır
```

```
  AppData: Pointer;     // zalloc ve zfree arasında geçen özel veri nesnesi
  data_type: Integer;   //  data hakkındaki en iyi tahmin : ascii veya binary
  adler: Integer;           // adler32 sıkıştırılmamış veri değeri
  reserved: Integer;    // gelecekteki kullanım için
end;  [33]
```

Bu kütüphaneyi kullanarak bir dosyayı veya bir nesne verisini sıkıştırmak ve açmak işlemi 3 aşamadan meydana gelir :

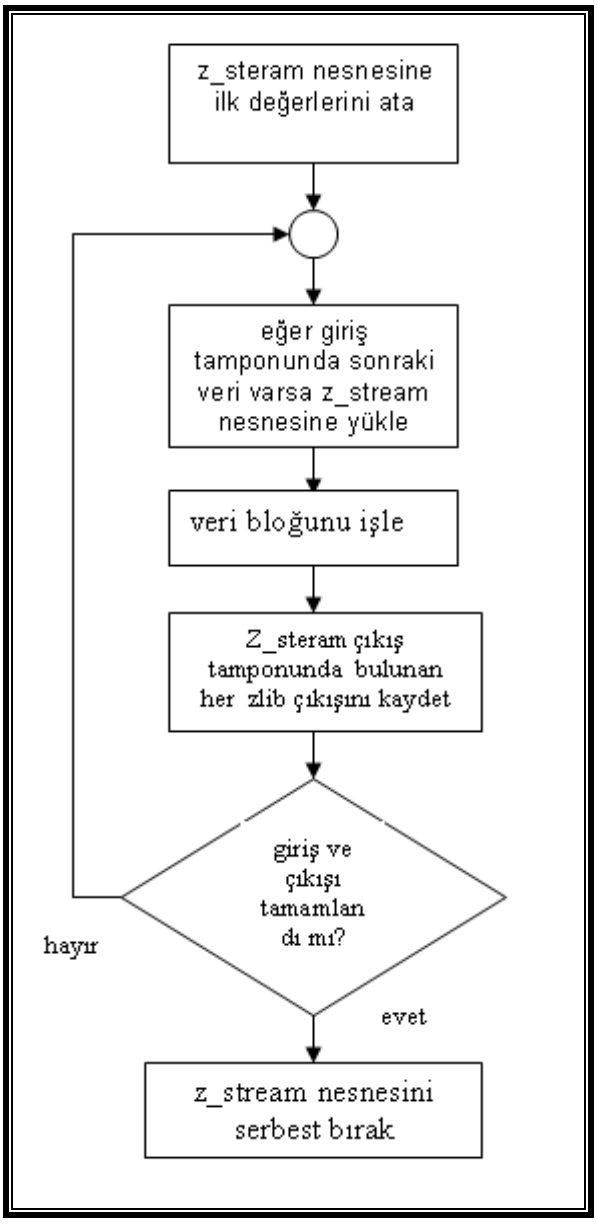
1. Bir z_stream nesnesi oluşturmak
2. Giriş ve çıkışı işlemek,z_stream nesnesini kullanarak ZLIB ile iletişime geçmek
3. z_stream nesnesini yok etmek.

Sıkıştırma işleminin 1. ve 3. adımları geleneksel(conventional) fonksiyonlar kullanılarak yapılır. Sıkıştırma veya açma işlemini başlatan ve sonlandıran fonksiyon prototipleri şu şekildedir:
deflateInit( ), inflateInit( ), deflateEnd( ), inflateEnd( ) ;

2. Adım, z_stream nesnesinin parametre olarak geçtiği inflate() veya deflate() çağrılarının tekrarı ile yapılır. Tüm işlem durumlarını bu nesne içerir, böylece kütüphaneye tamamen giriş yapmaya izin veren genel bayraklar veya değişkenler yoktur. Tek bir nesne içindeki işlem durumunu depolamak, API fonksiyona (zlib API) aktarılması gereken parametrelerin sayısını düşürür.

Sıkıştırma veya açma işlemi gerçekleştirilmek istendiğinde, ZLIB kendisine ait herhangi bir girdi/çıktı gerçekleştirmez. Bunun yerine z_stream nesnesine sağladığınız giriş arabellek işaretçisinden bilgi okur. Basit bir şekilde next_in üyesinin giriş verisinin sonraki bloğuna bir işaretçi belirleyebilirsiniz ve avail_in üyesine eldeki baytların sayısını yerleştirebilirsiniz. Bunun gibi ZLIB next_out üyesine belirlediğiniz bir bellek tamponuna çıkış verisini yazar. Çıkış baytlarını yazdığından, ZLIB avail_out üyesini 0 (sıfır) a düşene kadar azaltır [34]. Zlib'in bir dosyayı sıkıştırma ve açma işlemi şekil 4.9'da gösterilmiştir.

Bir Internet sitesinden (http://www.delphifr.com) elde edilen delphi programlama dili kullanılarak yapılmış olan TestCompression.exe adlı program LZ, LZW, HUFFMAN ve ZLIB sıkıştırma algoritmalarını test ederek aralarında karşılaştırma yapma imkanı sağlamıştır. Bu tez çalışmasında kullanılan üç farklı tipteki dosyanın sıkıştırılma algoritmaları kullanılarak elde edilen nihai boyutları Tablo 4.5'de gösterilmiştir. Görüldüğü üzere ZLIB sıkıştırma algoritması üç farklı tipteki dosya için en iyi sıkıştırma başarımını göstermiştir.



Şekil 4.9. ZLIB'in bir dosya veya nesneyi sıkıştırma ve açma işlemi [34]

Tablo 4.5. Sıkıştırma algoritmalarının karşılaştırılması

Dosya Adı	Dosya Boyutu	Sıkıştırma Algoritması			
		HUFFMAN	LZ	LZW	ZLIB
demo.mp3	38 KB	38 KB	47 KB	52 KB	37 KB
sndrec32.exe	122 KB	95 KB	91 KB	84 KB	60 KB
svega.wav	1781 KB	1150 KB	1059 KB	969 KB	909 KB

4.5. Sonuç

Kullanmış olduğumuz veri depolama birimlerinde dosya biçimlerine göre farklılıklar gösteren artıklıklar mevcuttur. Dört tip artıklık türü vardır: Karakter Dağılımı, Karakter Tekrarı, Çok Kullanılan Sözcükler, Konumsal Artıklık.

Delphi programlama dili kullanılarak yapılmış olan TestCompression.exe adlı program ZLIB sıkıştırma algoritmasının farklı tip dosyalarda iyi bir sıkıştırma oranına sahip olduğunu göstermiştir.

BÖLÜM 5. VERİLERİN SAYISALLAŞTIRILMASI

5.1. Giriş

Bilgisayar teknolojilerine dair tüm uygulamalar temel olarak ikili sayı sisteminde çalışırlar ve bilgiler bu sistem temelinde depolanırlar. Kullanıcıların yazdığı metinlerin ikili sayı sistemindeki karşılıklarını belirlemek için bir takım kodlama standartları mevcuttur. Bu kod standartlarına göre metinler sayısal bilgilere dönüştürülmektedir. Bununla birlikte bu bilgiler değişik ortamlarda iletilmek istendiğinde bir takım yöntemlerle analog sinyallere çevrilerek gönderilirler.

Bu bölümde, bu tez çalışmasında faydalanılan kodlama standartları ve analog sinyal çevrimleri anlatılarak sırörtme, sıkıştırma ve şifreleme yapılarak gerçekleştirilen geliştirilen uygulama hakkında bilgiler verilerek, alt bölümlerde bu uygulamalanın çalışma prensipleri ve algoritmaları detaylı şekilde anlatılmaktadır.

5.2. Metin Kodlama Standartları

Bu tez çalışmasında sayısal ses verileri içerisine veri gömme uygulamasını gerçekleştiren yazılımlar, metin kodlama standartlarından ASCII kullanılarak kodlama yapan metinleri ikili sisteme çevirmekte ve elde ettiği verileri sayısal ses paketlerine gömerek iletmektedir. Geliştirilen standartlar sadece ASCII kodlama sisteminden ibaret değildir. Aşağıda bu kod standartları hakkında bilgiler verilmektedir.

5.2.1. ASCII kodu

1988 yılında ANSI (American National Standards Institute) tarafından ortaya atılan ASCII, bilgisayar ağ ve sistemlerinde bilginin gösterilmesi/temsil edilmesi amacıyla kullanılan bir kod standardıdır. 7 bit olarak 0—127 arasında 128 değişik karakteri kapsamaktadır. Her bir karakter aşağıdaki tabloda gösterildiği gibi 7 bitlik bir kod ile ifade edilir. Örneğin “a” harfi; 7 bit ASCII kodunda $(1100\ 001)_{ascii}$ olarak ifade edilmektedir. Benzer şekilde “8” rakamı 011 1000, “+” işareti 010 1011 ASCII kodları ile ifade edilmektedir. Standart sembollerin dışında bir takım sembol ve şekillerin de ilave edilmesi ile 0—255 arasında genişletilmiş ASCII kodu oluşturulmuştur. Tablo 5.1 ASCII kodlarını göstermektedir.

Tablo 5.1. ASCII kod tablosu
(Standard No.X3-1988 of the ANSI, American National Standards Institute)

	0	1	2	3	4	5	6	7
0	NUL	DLE	space	0	@	P	`	p
1	SOH	DC1 XON	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3 XOFF	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	|
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	del

5.2.2. Genişletilmiş ASCII kodları

7-bit ASCII kodunun bazı karakterler için yetersiz kalmasıyla birlikte 7 bit, 8 bite çıkarılarak toplam 256 farklı kod ve Tablo 5.2'de görülen karakterler elde edilmiştir.

Tablo 5.2. ASCII kodlarının 8-bit olarak karakter karşılığı

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	Ç	160	A0	á	192	C0	└	224	E0	α
129	81	ü	161	A1	í	193	C1	┴	225	E1	ß
130	82	é	162	A2	ó	194	C2	┬	226	E2	Γ
131	83	â	163	A3	ú	195	C3	├	227	E3	π
132	84	ä	164	A4	ñ	196	C4	─	228	E4	Σ
133	85	à	165	A5	Ñ	197	C5	┼	229	E5	σ
134	86	å	166	A6	ª	198	C6	╞	230	E6	µ
135	87	ç	167	A7	º	199	C7	╟	231	E7	τ
136	88	ê	168	A8	¿	200	C8	╚	232	E8	Φ
137	89	ë	169	A9	⌐	201	C9	╔	233	E9	Θ
138	8A	è	170	AA	¬	202	CA	╩	234	EA	Ω
139	8B	ï	171	AB	½	203	CB	╦	235	EB	δ
140	8C	î	172	AC	¼	204	CC	╠	236	EC	∞
141	8D	ì	173	AD	¡	205	CD	═	237	ED	ø
142	8E	Ä	174	AE	«	206	CE	╬	238	EE	ε
143	8F	Å	175	AF	»	207	CF	╧	239	EF	∩
144	90	É	176	B0	░	208	D0	╨	240	F0	≡
145	91	æ	177	B1	▒	209	D1	╤	241	F1	±
146	92	Æ	178	B2	▓	210	D2	╥	242	F2	≥
147	93	ô	179	B3	│	211	D3	╙	243	F3	≤
148	94	ö	180	B4	┤	212	D4	╘	244	F4	⌠
149	95	ò	181	B5	╡	213	D5	╒	245	F5	⌡
150	96	û	182	B6	╢	214	D6	╓	246	F6	÷
151	97	ù	183	B7	╖	215	D7	╫	247	F7	≈
152	98	ÿ	184	B8	╕	216	D8	╪	248	F8	°
153	99	Ö	185	B9	╣	217	D9	┘	249	F9	∙
154	9A	Ü	186	BA	║	218	DA	┌	250	FA	·
155	9B	¢	187	BB	╗	219	DB	█	251	FB	√
156	9C	£	188	BC	╝	220	DC	▄	252	FC	ⁿ
157	9D	¥	189	BD	╜	221	DD	▌	253	FD	²
158	9E	₧	190	BE	╛	222	DE	▐	254	FE	■
159	9F	ƒ	191	BF	┐	223	DF	▀	255	FF	

Başlangıçta haberleşme işlemleri için tasarlanmakla birlikte bilgisayar uygulamalarında geniş yer bulmuştur. 8-bit ikili sayı 256 farklı koddan birisi olarak sunulmaktadır. Böylece, örneğin onluk (decimal) karşılıkları bir dizi halinde "72, 69, 76, 76, 79" kullanıldığında, ASCII kod karşılığı olarak "h, e, l, l, o" kelimesini

oluşturmaktadır. ABD ve İngiltere dışında diğer ülke dillerindeki karşılanmayan karakterler sebebiyle biri diğeriyle uyumsuz US-ASCII dışında birtakım farklı ulusal genişletilmiş kodlar türemiştir. Bu duruma bir son vererek bir standardizasyona gitmek üzere 16-bit (2 Bayt) 65536 karakter kümesinden oluşan UNICODE geliştirilmiştir. İçerisinde harf, rakam, özel karakterler ve diğer dilbilimsel sembol ve karakterleri içermekte olup günümüzün en önemli dillerinde kullanılmaktadır. İngilizce için Latin Alfabesi'ni, Rusça için Kril Alfabesini, Yunanca, İbranice ve Arapça alfabelerini; Avrupa, Afrika, Hint Yarımadası, Asya (Japonya, Kore, Çin) dillerine ait harf ve sembolleri kapsar.

Bu tez çalışmasın geliştirilen uygulama, sayısal ses verileri içerisine dosya/veri gömme uygulamasıdır. Bu uygulamada, ses iletişimi yapılırken Ses Gönderici Modül kullanılarak gönderilen dosyadaki tüm veriler ASCII kod karşılıkları bulunduktan sonra ikili sisteme çevrilmekte ve elde edilen veriler ses çerçeveleri içerisine gömülerek gönderilmektedir. Bu işlemler zamanlayıcı yardımı ile her saniye tekrar edilmektedir. Aynı şekilde Ses Alıcı Modülde ise gelen ses paketleri içerisinden alınan ikili sistemdeki bilgiler onluk sisteme ardından da ASCII kod karşılıklarına göre dosya haline getirilmekte ve kullanıcıya sunulmaktadır.

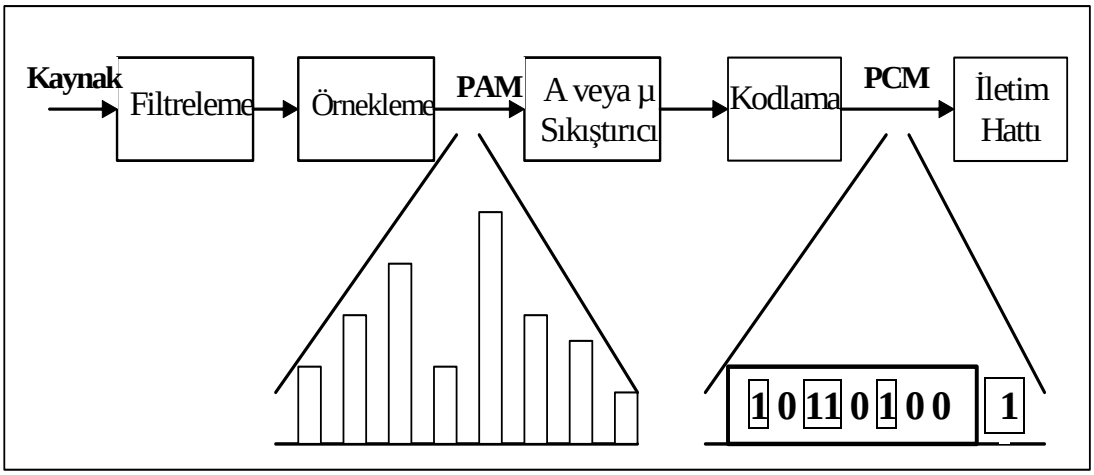
5.3. Analog Ses Sinyallerinin Sayısal Veriye Çevrilmesi

Darbe Kod Modülasyonu (Pulse Code Modulation: PCM), analog işaretlerin belirlenmiş sayısal forma dönüştürülmesini sağlayan bir tekniktir. Bu teknikte analog işaretten sayısal bilgiye ve sayısal bilgiden analog işarete dönüşüm sırasında oluşan örnekleme kayıpları oldukça küçüktür. Bu nedenle PCM örnekleme kayıplarından oldukça etkilenen (konuşma gibi) işaretlerin sayısal formda iletilmesini sağlayan önemli bir tekniktir.

5.3.1. Darbe kod modülasyonu

Sayısal işaretlerin, gürültüden etkilenmemesi ve tümdevre teknolojisinin gelişmesi ile sayısal verinin işlenmesinin (iletilme, sıkıştırma) nispeten daha ekonomik olması artık bilgi iletimi, saklanması ve işlenmesi sırasında sayısal formatın analog formata

göre tercih edilmesini doğurmuştur. Ancak analog formdaki kaynak bilgisinin sayısal forma dönüştürülmesi sırasında meydana gelen örnekleme ve kodlama hatalarından dolayı alıcıda elde edilen bilgideki bozulma bir problem olarak ortaya çıkmaktadır. Özellikle kaynak verisinin konuşma işaretleri olması, alıcıdaki bozulmayı daha da belirgin hale getirmekte ve sayısal formun konuşma bilgisi için kullanılmasını engellemektedir. PCM yukarıda açıklanan probleme bir çözüm önerisi olarak 1970'li yıllarda ortaya çıkmış ve günümüzde bu amaç için en çok kullanılan sayısallaştırma tekniği olmuştur. PCM'de önce analog işaret örneklenir, sonra kuantalanır ve son olarak da kodlanır. Şekil 5.1'de PCM yapısının şeması görülmektedir.



Şekil 5.1. PCM yapısının şeması

Örnekleme :

Örnekleme devresi, analog giriş sinyalini belirlenen frekansta periyodik olarak örnekleyerek çıkışa PAM sinyali olarak aktaran devredir. Burada Nyquist teoremi dikkate alındığında, işaret bantgenişliğinin iki katı frekansında örnekleme yapılmalıdır. Yani ses işareti 4 KHz kabul edildiğinde, saniyede 8000 (2x4000=8000) örnek alınmalıdır. İşaretin örneklenmesi örnekle-tut devreleri yardımı ile yapılmaktadır. Teorik olarak, Nyquist frekansının kullanılması örtüşmeye yol açmadığı halde, pratikte örnekleme frekansı minimum Nyquist sınırından biraz yüksek tutulur. Örneğin PCM kanallarından iletilecek sesin örnekleme frekansı, ITU-T tarafından f_N=8 KHz olarak belirlenmiştir.

A veya μ kuantalayıcı :

A veya μ tipi kuantalama yaklaşımı özellikle ses haberleşmesi uygulamalarında ses işaretlerinin sayısal işaretlere dönüştürülmesi için kullanılmaktadır. Bu tip bir kuantalama işlemine sıkıştırıcı-genleştirmeli kuantalama da denilir. Kuantalama işleminde düzgün dağılımlı olmayan giriş işareti, bir sıkıştırıcıdan geçirilerek daha sık rastlanan düşük genlikli değerlerin arası açılırken, daha düşük olasılıklı yüksek genlikli değerlerin arası sıkıştırılmakta ve bu sayede giriş düzgün dağılımlı hale getirilmektedir. Böylece düzgün dağılımlı hale getirilmiş işaret, doğrusal bir kuantalayıcı ile nicelenmektedir. Alıcıda PCM kod çözümü yapılırken, kuantalanan işaret seviyelerinin sıkıştırıcının tam tersi bir işlev yerine getiren bir genleştiriciden geçirilmesi sonucu işaret değerleri normal seviyelerine geri getirilmektedir.

PCM kodlayıcı :

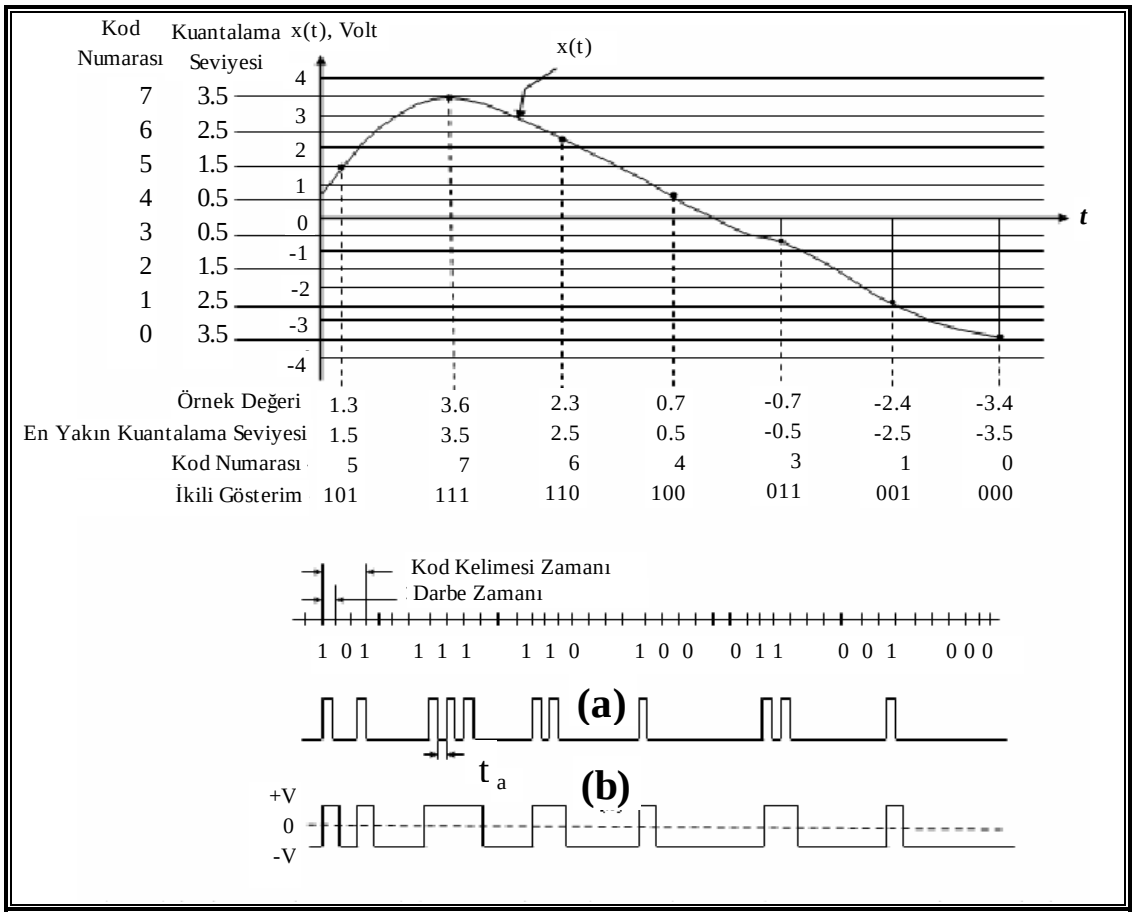
PCM Kodlayıcıda, kuantalanmış örnek değerleri sayısal kod sözcükleri şekline çevrilmektedir. Haberleşmede kuantalanmış örnek değerleri, 8-bitlik ikili kodlar olarak ifade edilmektedir. PCM kullanılan bir haberleşme sisteminde, örneklenen işaret 256 seviyeli olarak kuantalanmakta ve 8-bit ile temsil edilmektedir. Bundan dolayı PCM'li sayısal veri transferi için 8x8000 = 64000 bps yani, 64 Kbps taşıma kapasitesine sahip bir veri iletim kanalına ihtiyaç vardır. Elde edilen bu işarete DS-0 (Digital Signal-0) denir.

Örnekleme sonucu elde edilen genlik değerleri hala analogdur ve minimum genlik ile maksimum genlik arasında herhangi bir değeri alabilirler. Bu örnekleri alıcıya değişmeden iletmek için sonsuz sayıda bit kullanmak gerekir. Oysa, kullanılacak genlik değerlerinin sayısı sınırlı olursa, kullanılacak bit sayısı da sınırlı olur. Daha sonra kuantalanmış işaret belli bir sayı sistemine göre kodlanır. Kuantalanmış işarete, bir kod kelimesi karşı düşürülür. İkili sayı sisteminde, 1 olan yerlerde örneğin $+V$ genliğinde bir darbe, 0 olan yerlerde ise boşluk göndererek bu kodu iletmek mümkün olur (Şekil 5.2(a)). Daha farklı bir iletim şekli olarak, 1 olan yerlerde $+V$ genliğinde bir darbe, 0 olan yerlerde $-V$ genliğinde bir darbe de iletilebilir (Şekil 5.2(b)). Bu darbelerin genişlikleri kanala uygun şekilde seçilerek, iki darbe arasında güven

aralığı (t_g) bırakılabilir. Kuantalama işleminde kullanılan kuanta seviyesi sayısı arttıkça işaret daha iyi temsil edilir. Buna karşılık bir örneği iletmek için gereken bit sayısı artar. İşaret *n* bit ile kodlanıyorsa, kuanta seviyesi sayısı $Q=2^n$ olmalıdır. Kuantalanmak istenen işaretin maksimum genliği A_{max}, minimum genliği A_{min} ise ve işaretin bu aralıkta değişen genlik değerleri *Q* adet eşit kuanta seviyesine bölünmek isteniyorsa, kuantalama aralığı veya adımı şöyledir;

$$a = \frac{A_{max} - A_{min}}{2^n} \tag{5.1}$$

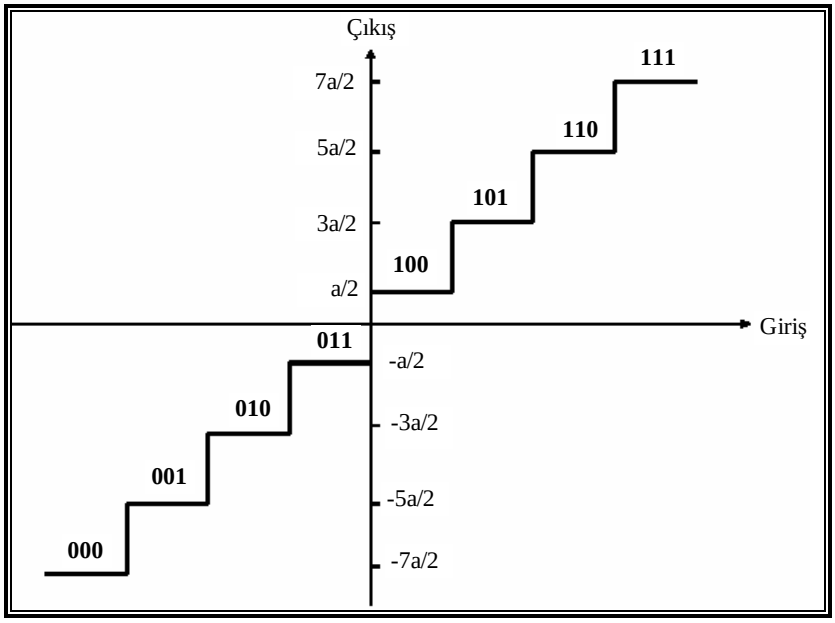
Şekil 5.2'de Q = 8, n = 3 ve a = 1 için örnek bir işleyiş görülmektedir.



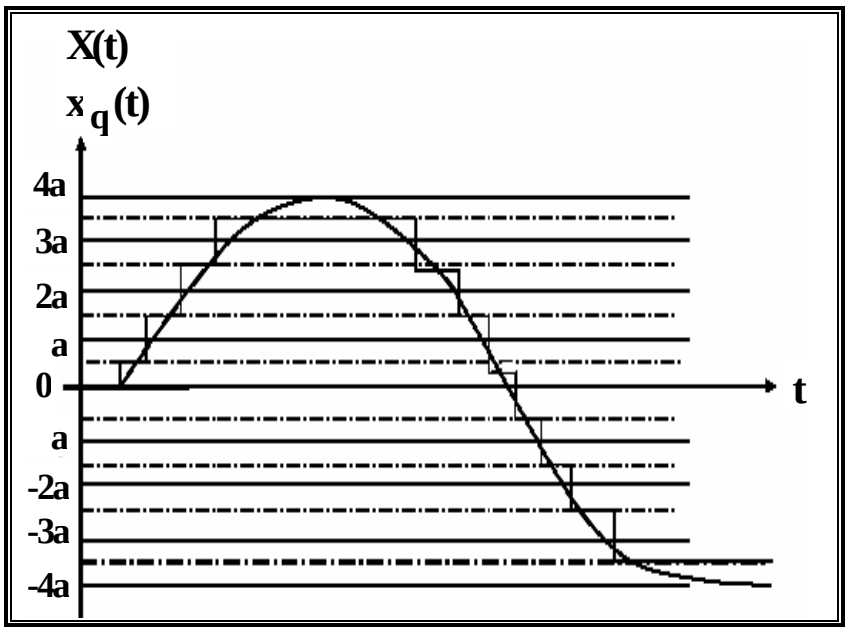
Şekil 5.2. Analog bir işaretin örneklenmesi ve karşılığı olan PCM işaretinin gösterimi

Kuantalama aralıklarının eşit seçildiği kuantalayıcılara düzgün kuantalayıcı adı verilir. Şekil 5.3'de düzgün bir kuantalayıcının giriş-çıkış eğrisi ve karşı düşen kod

kelimeleri görülmektedir. Terslenebilir(reversible) bir işlem olmayan kuantalama sonucunda bir bilgi kaybı olmaktadır. Kuantalanmış örnek işaret X(t), mesaj işareti $x_q(t)$'nin yaklaşık bir değeri olduğundan bir bozulma söz konusudur (Şekil 5.4).



Şekil 5.3. Düzgün kuantalama eğrisi

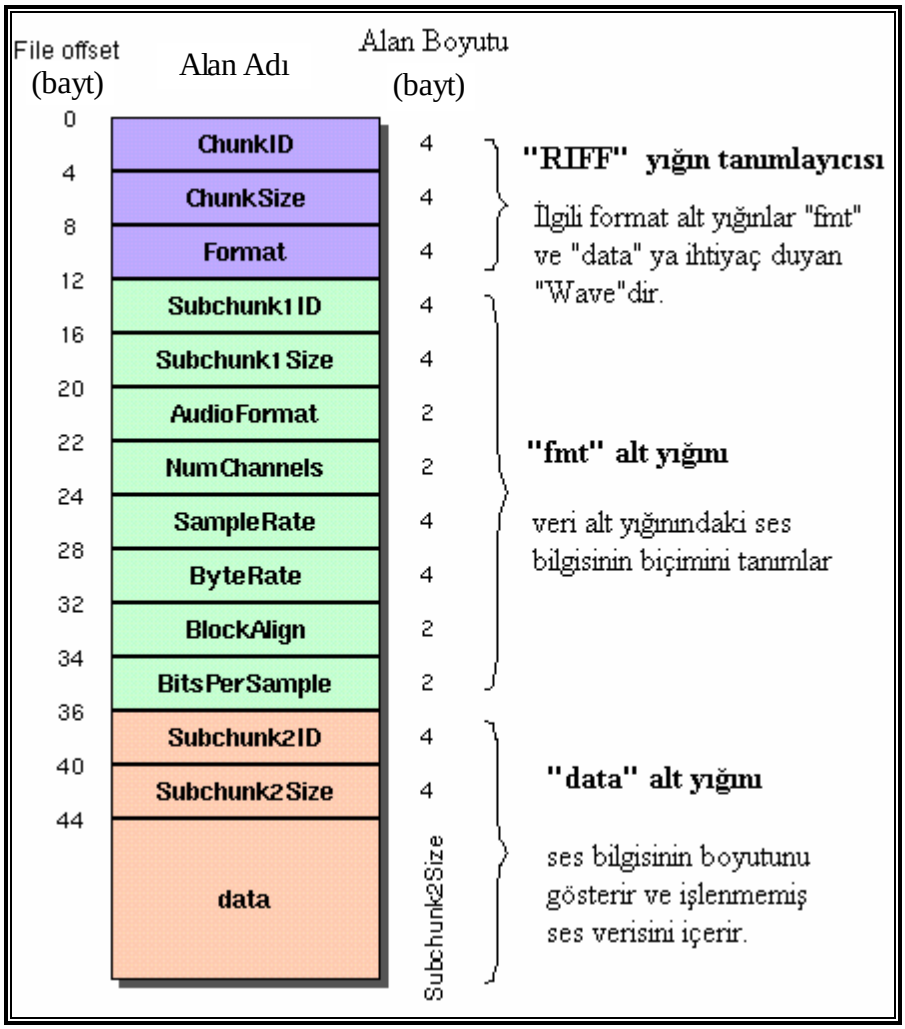


Şekil 5.4. Analog işaret ile kuantalanmış işaret arasındaki hata

Kuantalama hatası etkisi bir toplamsal gürültünün etkisine eşdeğerdir. Bu nedenle bu bozulma kuantalama gürültüsü olarak da adlandırılır. Bu bilgilerden hareketle, tez çalışmalarında hedeflenen amaca ulaşmak için yapılan uygulamalar, zaten kuantalama hatası sebebi ile bozulmaya uğrayan ses verilerini değiştirerek bozulmayı daha da arttıracak bir etki oluşturmaktadır. Bu sebeple bozucu etkinin çok az olması için veri gömme işlemlerinde, sayısal ses bilgilerinin en düşük değerlikli bitleri kullanılmaktadır. Yukarıdaki anlatılanlar göz önünde bulundurulduğunda gerçek zamanlı olarak sayısal ses verilerine ulaşılıp, bu verilerin son bitlerinin değiştirilerek alıcıya gönderilmesi işleminde bozulma oranı ile birlikte, verilerin aktarım hızı ve yapılan örnekleme sayısı çok büyük önem kazanmaktadır.

5.3.2. PCM ses verisi formatı

Ses kartından alınan analog sesler, bilgisayar ortamında PCM yöntemi ile sayısallaştırılmaktadır. Standart olarak alınan ses bilgileri “.wav” dosya tipindedir. Bu tip bir dosya yapısının ilkel versiyonu ise Microsoft’un “.riff” (Resource Interface File Format) uzantılı dosya yapısıdır. “Wave” ses dosyasının yapısı Şekil 5.5’de sunulmaktadır [38].



Şekil 5.5. Kurallara uygun wave dosya formatı

Yukarıda görüldüğü üzere ilgili ses dosyasının yapısı temel olarak üç bölümden oluşmaktadır. Bunlar “RIFF” yığın tanımlayıcısı, “fmt” alt yığını ve “data” alt yığınıdır. Tablo 5.3’de RIFF yığın tanımlayıcı bilgileri görülmektedir.

Tablo 5.3. RIFF yığın tanımlayıcısı

BOYUT (BAYT)	AD	AÇIKLAMA
4	ChunkID	ASCII biçimindeki “RIFF” yazısını içerir.
4	ChunkSize	Yığın boyutunu içerir.
4	Format	ASCII biçiminde “WAVE” bilgisini içerir.

İnsanların soldan sağa veya sağdan sola doğru okunan farklı alfabelere sahip olmaları gibi işlemciler de baytları saklarken en büyük değerlikli (MSB) baytın solda veya sağda olmasına göre sınıflandırılır.

“fmt” alt yığını ses verisinin formatını tanımlamaktadır. “fmt” alt yığını bilgileri Tablo 5.4’da görülmektedir.

Tablo 5.4. “fmt” alt yığını

BOYUT (BAYT)	AD	AÇIKLAMA
4	Subchunk1ID	“fmt” yazısını içerir.
4	Subchunk1Size	PCM için 18’dir.
2	Audioformat	PCM için 1’dir. Sıkıştırma tiplerinin bazıları için 1’den faklı değerler mevcuttur.
2	NumChannels	Mono için 1, Stereo için 2’dir.
4	SampleRate	8000, 44100 gibi örnek oranlarıdır.
4	ByteRate	= SampleRate x NumChannels x BitsPerSample/8
2	BlockAlign	= NumChannels x BitsPerSample/8
2	BitsPerSample	8 bit için 8, 16 bit için 16.

“data” alt yığını verinin boyutunu ve güncel ses bilgisini içermektedir. “data” alt yığını bilgileri Tablo 5.5’de görülmektedir.

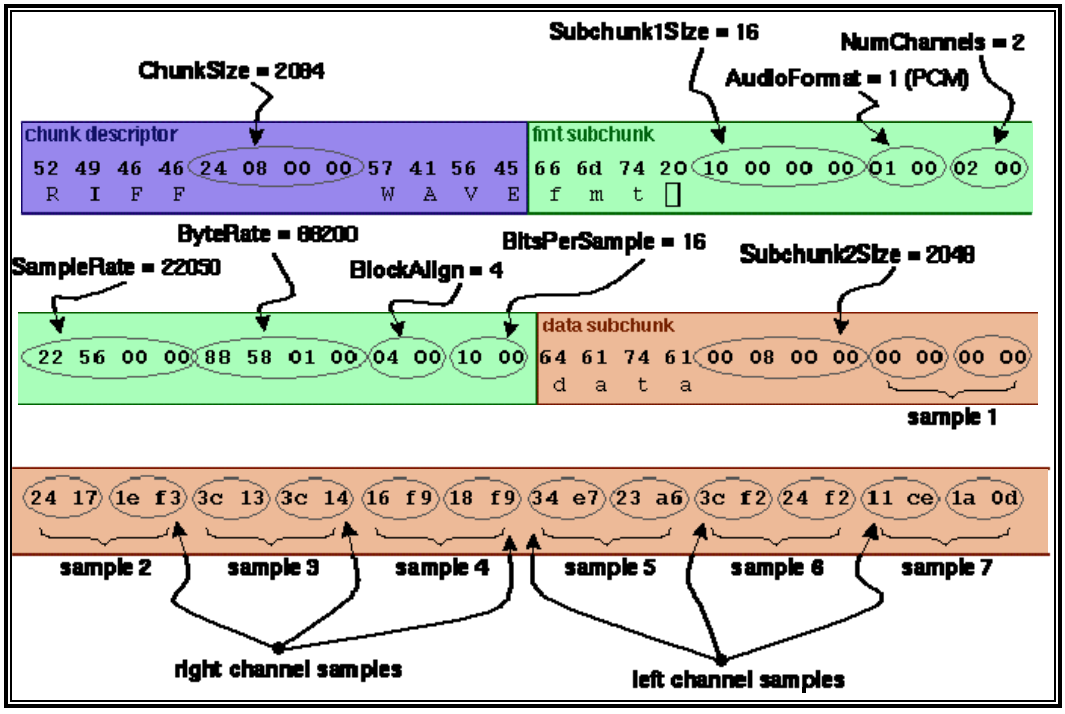
Tablo 5.5. “data” alt yığını

BOYUT (BAYT)	AD	AÇIKLAMA
4	SubChunk2ID	“data” yazısını içerir.
4	SubChunk2Size	= NumSamples x NumChannels x BitsPerSample/8 (Bu bilgi, data bölümünde bulunan veri boyutu bilgisidir.)
n	Data	Güncel ses bilgisi (44.bayttan itibaren).

Yapılan tez çalışmalarında, Şekil 5.5'de görülen "data" alt yığınındaki veriler üzerinde değişiklikler yapılarak sırörtme uygulaması gerçekleştirilmektedir. RIFF yığın tanımlayıcısı ve fmt alt yığını üzerinde yapılacak herhangi bir değişiklik orijinal ses verisinin yapısını bozacağından ses haberleşmesinin gerçekleşmesine engel teşkil etmektedir. Aşağıda onaltılık (hexadecimal) sayı sisteminde verilmiş 72 bayt boyutunda Wave dosya bilgisi verilmektedir:

52 49 48 48 24 08 00 00 57 41 58 45 88 8d 74 20 10 00 00 00 01 00 02 00
22 58 00 00 88 58 01 00 04 00 10 00 84 81 74 81 00 08 00 00 00 00 00 00
24 17 1e f3 3c 13 3c 14 18 f9 18 f9 34 e7 23 a8 3c f2 24 f2 11 ce 1a 0d

Şekil 5.6'da bu bilgilerin bir ses dosyasında yerleşimi verilmektedir.



Şekil 5.6. Örnek bir ses dosyası

5.4. Sonuç

İnsan sesi analog veriler içermektedir. Bilgisayarlar bu verileri ancak sayısallaştırarak depolayabilmekte veya çalabilmektedir. Analog ses verilerinin

sayısallaştırılması süreci bu tez çalışması açısından önem arz etmektedir. Anlaşılacağı üzere bu sayısallaştırma sürecinde birim zamanda alınan örnek sayısı ile seste meydana gelecek olan bozulma arasında ters orantı söz konusudur. Bu nedenle birim zamanda ne kadar fazla örnek alınırsa gerçek sese o kadar yaklaşılmaktadır.

Veriler bilgisayarlarda ikili sistemdeki sayılar şeklinde (0 veya 1) ifade edilmektedir. Bu zorunluluk karakterlerin de sayısal karşılıklarının oluşturulması için kodlama sistemlerinin geliştirilmesine sebep olmuştur. Bunlardan en çok kullanılanı ASCII kodlama sistemidir. Yapılan tez çalışmasında geliştirilen uygulama sayısal ses verileri içerisine dosya/veri gömme uygulamasıdır. Bu uygulamada kablosuz ses iletişimi yapılır iken Ses Gönderici Modül kullanılarak gönderilmek istenen dosya/veri'deki tüm ifadeler ASCII kod karşılıkları bulunduktan sonra ikilik sisteme çevrilmekte ve elde edilen veriler ses çerçeveleri içerisine gömülerek gönderilmektedir.

BÖLÜM 6. SIKIŞTIRILMIŞ VE ŞİFRELENMİŞ GİZLİ DOSYA TRANSFERİNİN GERÇEKLEŞTİRİLMESİ

6.1. Giriş

Bu bölümde, sırörtme, sıkıştırma ve şifreleme yapılarak gerçekleştirilen uygulama hakkında bilgiler verilerek, alt bölümlerde bu uygulamaların çalışma ilkeleri ve algoritmaları detaylı şekilde anlatılmaktadır.

Bu tez çalışması, gerçek zamanlı kablosuz sayısal ses haberleşmesinin üzerine yapılmıştır. Gerçek zamanlı kablosuz ses haberleşmesi yapılırken gönderilecek sayısal ses bilgilerine kullanıcının istediği herhangi bir veri kümesini / dosyayı gömme uygulamasıdır.

Yapılan çalışmada kablosuz haberleşme özelliğine sahip olan iki dizüstü bilgisayar (P4 3.2 GHz, 384 MB RAM ve Celeron 1.7 GHz, 224 MB RAM) ve bir adet erişim noktası (U.S. Robotics 54 MBit/s) kullanılmıştır.

İlgili uygulamalar Borland Delphi 7.0'da geliştirilmiştir ve temel olarak Network Multi Medya (NMM) bileşeninden (component) faydalanılmıştır. Sıkıştırma uygulamasında, Borland Delphi 7.0 içerisinde gelen ZLIB sıkıştırma fonksiyonları kullanılmıştır. Şifreleme uygulaması, OTP (One Time Pad- Tek Zamanlı Deste) yöntemi kullanılarak gerçekleştirilmiştir. Görüntü özelliklerinin Windows XP'ye uyumlu olması açısından dosya gönderme uygulamasında "Jedi" bileşeni kullanılmaktadır. İlgili uygulamalarda "main.pas" dosyalarının dışında bir çok "pas" uzantılı dosya kullanılmaktadır. Bu dosyaların önemlileri şunlardır.

a) NMMP2PVoiceServer.pas (Gelen verinin hangi aşamada olduğunu tespit etmektedir).

b) NMMVoiceClient.pas, (Giden verinin hangi aşamada olduğunu tespit etmektedir).
c) NMMAudioPlayThread.pas (Gömülü verileri ayırt etme işlemi yapmaktadır).
d) NMMAudioRecordThread.pas (Verileri gömme işlemi yapmaktadır).

Bu dosyalar temel olarak yanlarında belirtilen işlevlere sahip olup, kaynak kodları Ek-A ve Ek-B'de verilmektedir. İzleyen alt bölümlerde tez çalışması olarak gerçekleştirilen uygulamalar detaylı bir şekilde anlatılmaktadır.

6.2. Sayısal Ses İçerisinde Gizli Gömü Verilerinin/Dosyalarının (Ssgd) Kablosuz Transferi İçin Geliştirilen Yazılım

Gönderilecek olan dosya seçimi yapıldıktan sonra ilgili dosya önce sıkıştırma, sonra şifreleme işlemelerine tabi tutulur. Ses kartından alınan ilgili örneklerin en küçük değerlikli bitine, gömülmek istenen dosyanın ilgili bitleri yerleştirilerek gönderilmektedir (Ses Gönderici Modül). Aynı şekilde bu verileri sezerek gömülmüş olan veriyi tespit edip açan ve şifreyi çözen algoritma da geliştirilmiş olup detayları takip eden alt bölümlerde anlatılmaktadır (Ses Alıcı Modül).

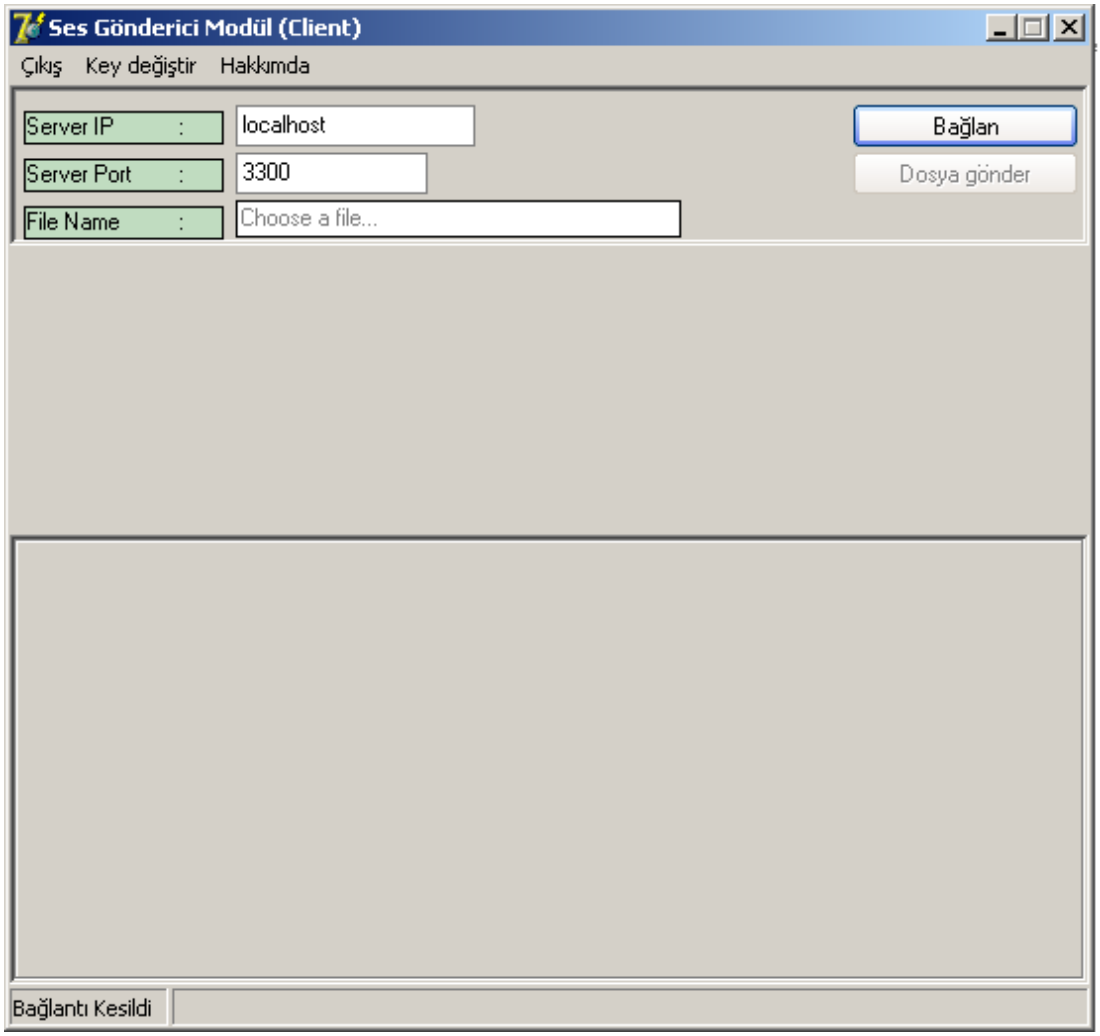
Geliştirilen uygulamada ses verileri değiştirilirken, meydana gelecek olan bozulmanın algılanamayacak seviyede olması için çalışılmıştır.

6.2.1. SSGD kablosuz transferi için geliştirilen yazılımın kullanıcı arayüzleri

Uygulamanın başlatılması için ilk olarak erişim noktası (Access Point, AP) ile birbirine bağlanmış olan iki adet bilgisayardan birinde Ses Gönderici(Client) modül diğerinde ise Ses Alıcı (Server) modül çalıştırılarak kablosuz sayısal ses haberleşmesinde temel adım atılmış olmaktadır.

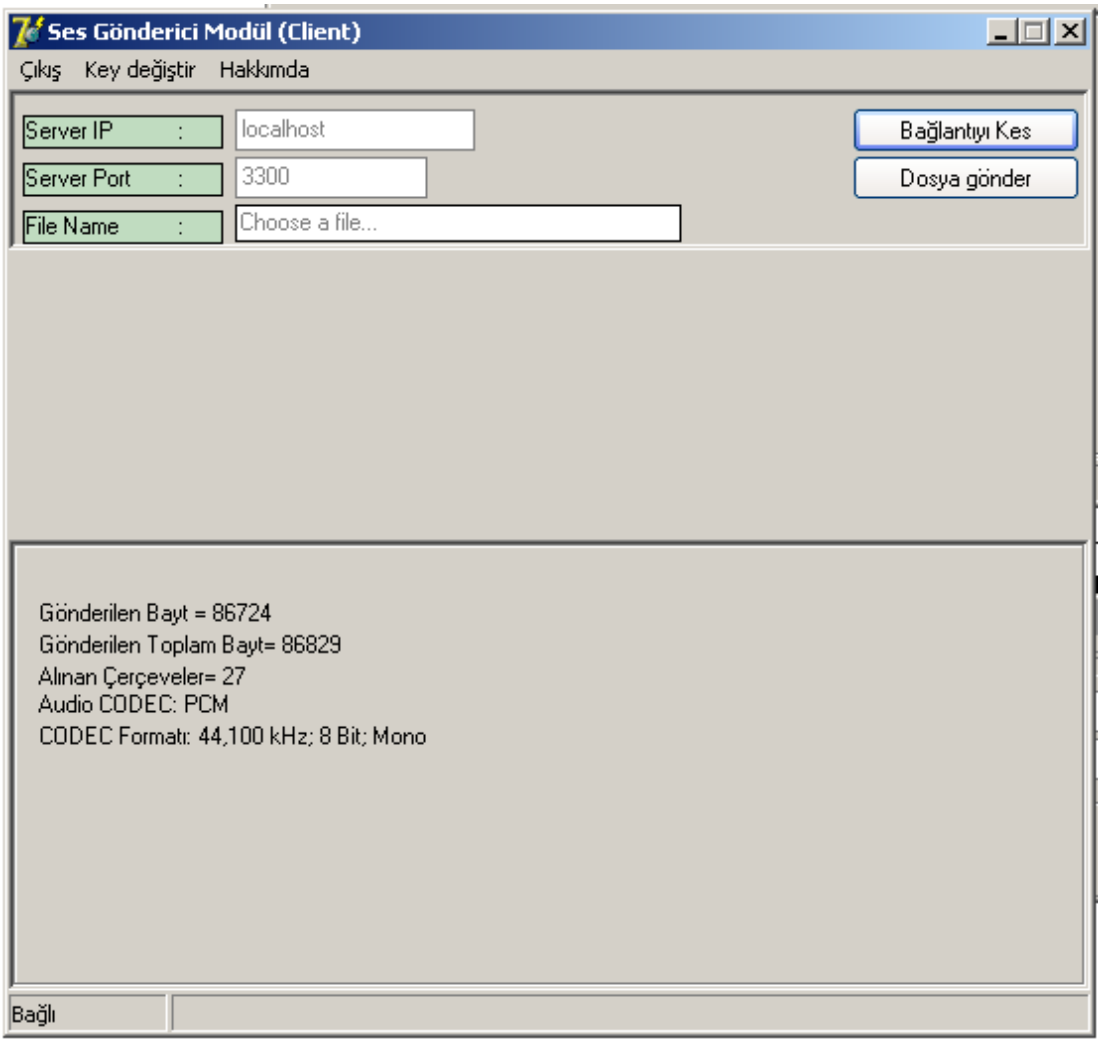
Ses Gönderici Modül ilk açıldığı anda IP adresi "localhost", port ise "3300" olarak belirlenmiş şekilde ekrana gelmektedir. Kullanıcı "Server IP Adresi" olarak

belirtilmiş olan kısma, ses bilgilerini alacak olan bilgisayarın IP adresini yazmalıdır. Ses Gönderici Modül çalıştığında Şekil 6.1'deki ekran çıktısı elde edilmektedir.



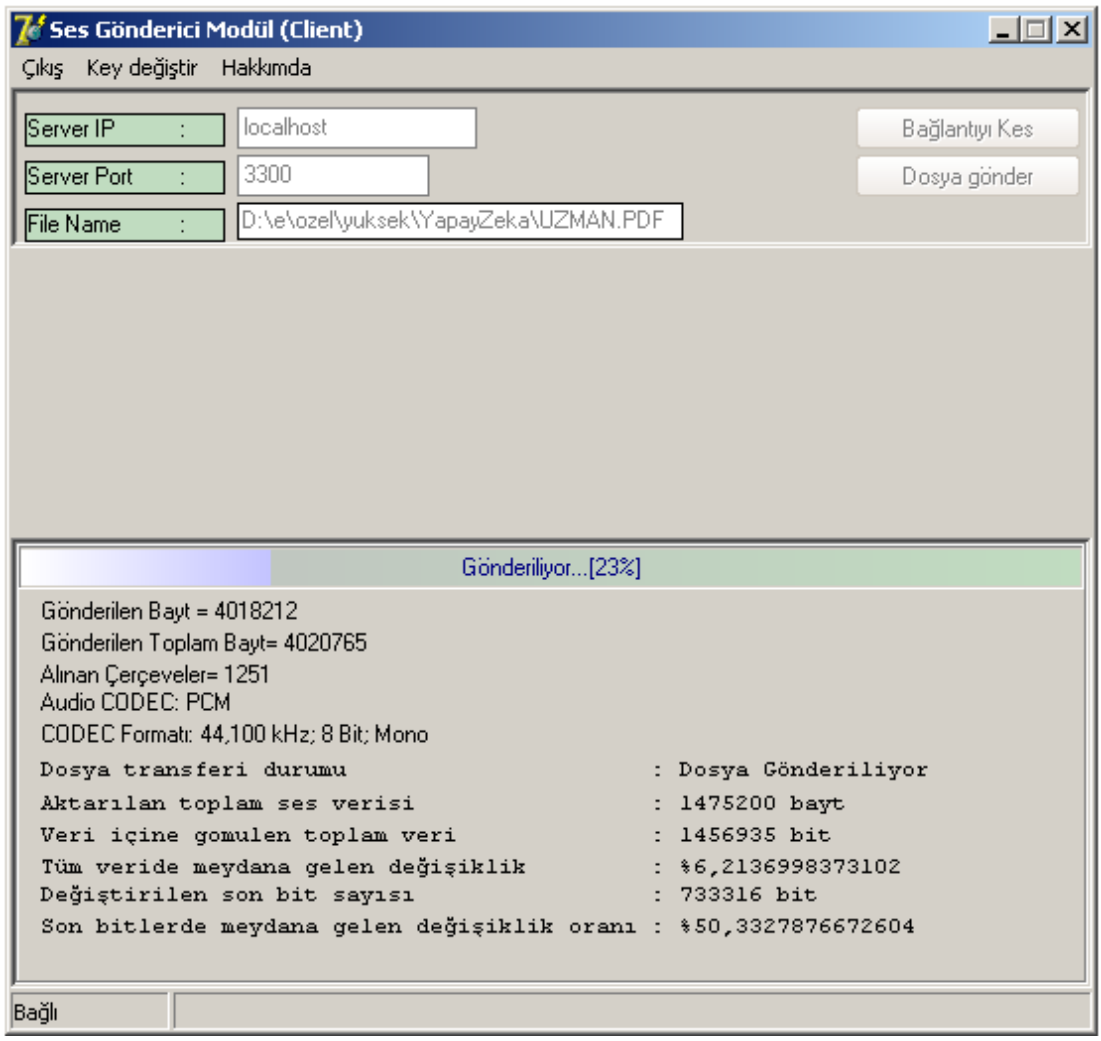
Şekil 6.1. Ses gönderici modülün başlangıç görünümü

Kullanıcı, sunucu adresini yazıp bağlan butonuna bastığında ses iletişimi başlatılır ve ilgili istatistiki bilgilerde görünmeye başlar. Şekil 6.2'de bu durumu gösteren bir ekran çıktısı verilmektedir.



Şekil 6.2. Ses gönderici modülün çalışma görünümü

Kullanıcı, herhangi bir anda dosya gönder yazan butona tıklayarak istediği bir dosyayı gönderebilmektedir. Dosya gönderimi yapılmaya başlandığı andan itibaren, dosya gönderim oranını belirten bir durum belirteci de ekranda görünmektedir. Bununla birlikte dosya gönderimi boyunca, taşıyıcı ses verilerindeki yüzdelik değişimi ve diğer istatistiki bilgiler de ekrana gelmektedir. Şekil 6.3'de bu durumu gösteren bir ekran çıktısı verilmektedir.

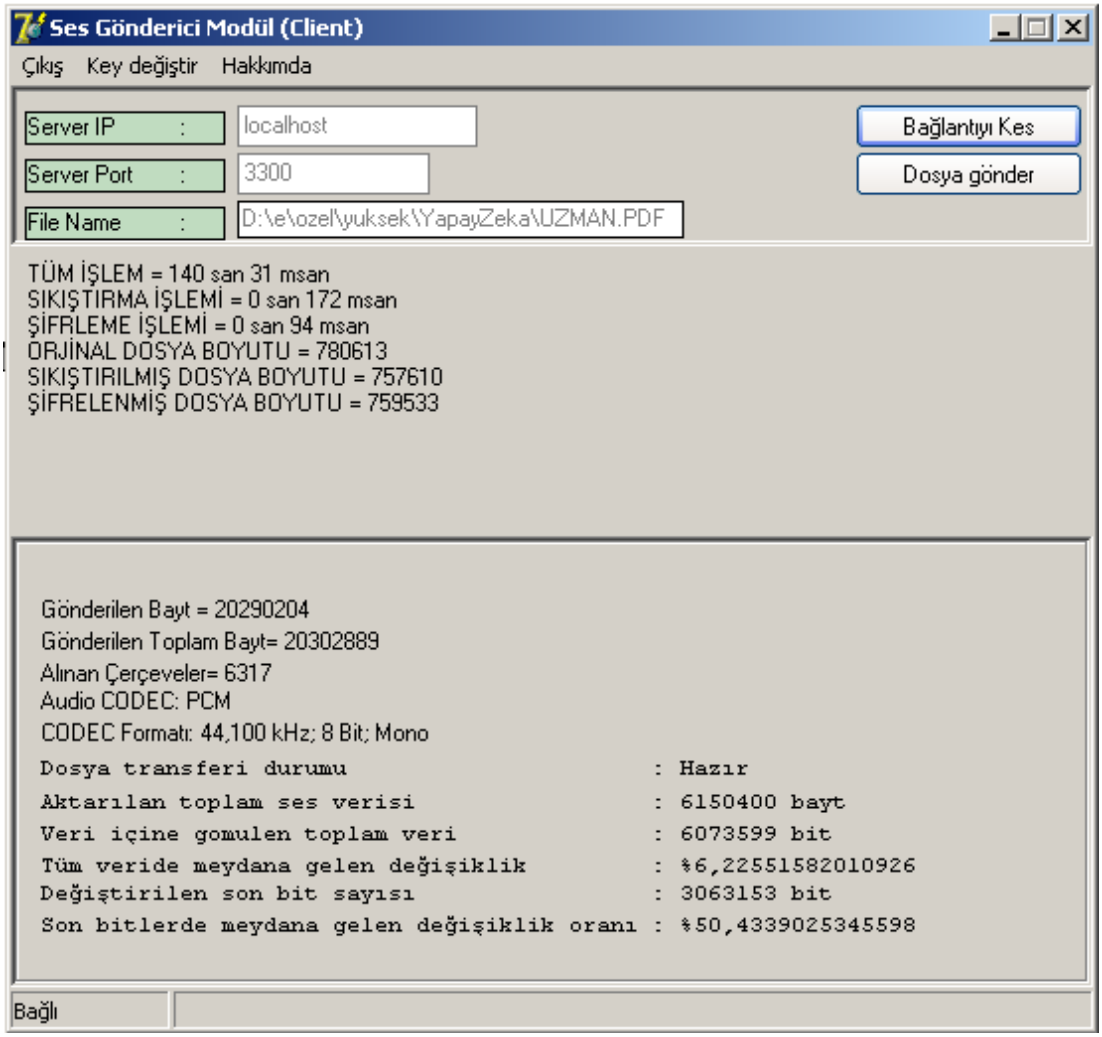


Şekil 6.3. Ses gönderici modülün veri aktarımı yapıldığı andaki görünümü

Yukarıdaki şekilde görülen istatistikler dosya gönderimi boyunca değişmektedir. İlgili dosyanın gönderimi sırasında değişimlerin hangi oranda olduğu, yani net sonuçlar dosya gönderimi tamamlandıktan sonra belli olmaktadır.

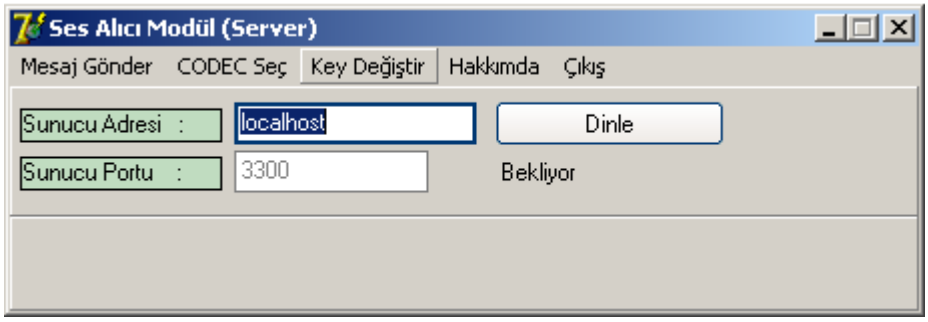
Şekil 6.4'de bir dosya gönderimi işlemi tamamlandıktan sonra elde edilen son görüntü ve istatistikler verilmektedir. Ekran görünümünden de anlaşılacağı üzere ilgili dosya gönderilirken, taşıyıcı ses verilerinin % 50,43 oranındaki kısmı son bitlerinin değişimi anlamında farklılaşmış görünmektedir. Ancak bu ses verileri matrissel anlamda (tüm veride meydana gelen değişiklik) düşünüldüğünde, taşıyıcı ses verilerinin bit düzeyinde % 6,22'si değişmiş olarak yansımaktadır. Dosyanın boyutu 780613 bayt olup, sıkıştırıldıktan sonraki boyutu 757610 bayt ve şifrelendikten sonrada 759533 bayt olmuştur. Sıkıştırma işleminden sonra dosya

boyutunun biraz daha artmasındaki sebep şifreleme bilgisinin dosya içerisine kaydedilmiş olmasındandır. Tüm gönderme işlemi 140 saniye 31 milisaniye sürmüştür. Sıkıştırma işlemi 172 milisaniye, şifreleme işlemi de 94 milisaniye sürmüştür.



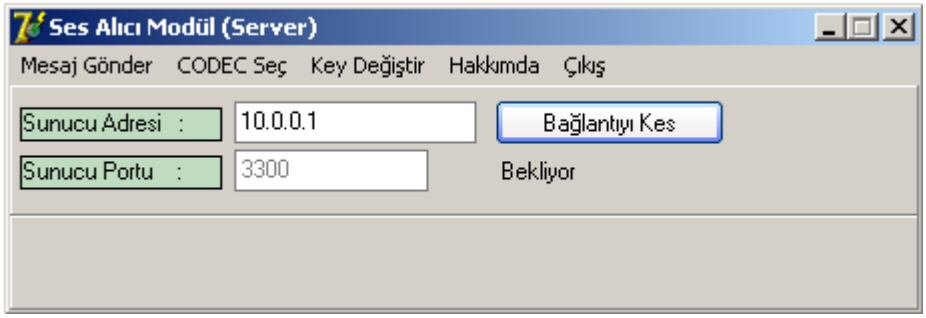
Şekil 6.4. Ses gönderici modülün veri aktarımı yapıldıktan sonraki görünümü

Ses Alıcı modül açıldığında ise kullanıcı ilk olarak Şekil 6.5'deki görüntüyü elde etmektedir.



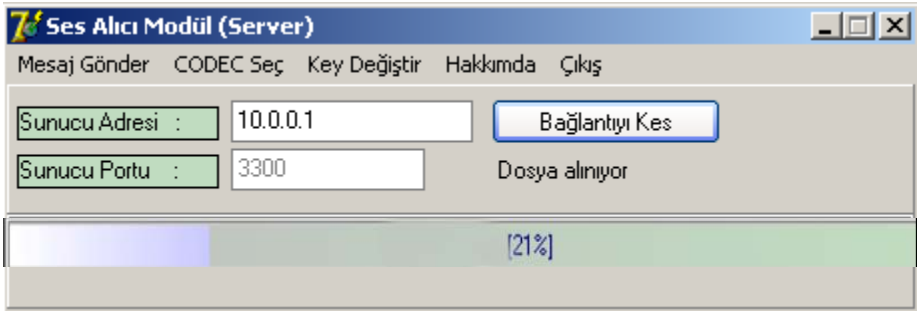
Şekil 6.5. Ses alıcı modülün çalıştırıldığı andaki görünümü

Bu modülde ilk ekran görünümünde Sunucu Adresi kısmı “localhost”, Sunucu Portu kısmı ise “3300” olarak belirlenmiş olarak ekrana gelmektedir. Kullanıcı “Dinle” butonuna bastığında; modülün, IP numarasını sistemden alarak ilgili kutucuğa otomatik olarak yazmasını sağlamaktadır. Şekil 6.6’da bu işlemin yapılarak, ses iletişiminin başlatıldığı bir ekran çıktısı verilmektedir.



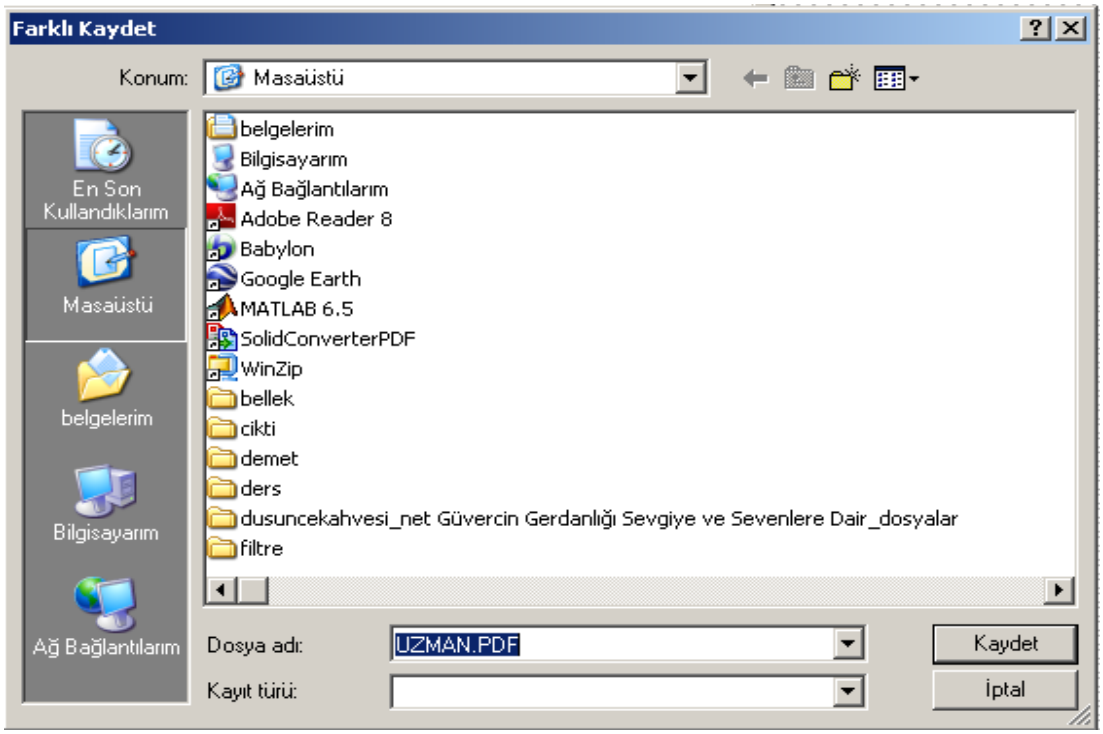
Şekil 6.6. Ses alıcı modülün iletişim başladığındaki görünümü

Eğer Ses Gönderici Modül kullanıcısı iletişim esnasında herhangi bir dosyayı göndermeye başlarsa, Ses Alıcı Modül bundan kullanıcısını haberdar etmekte ve durum hakkında bilgi vermektedir. Şekil 6.7’de bu durum ile ilgili bir ekran çıktısı verilmektedir.

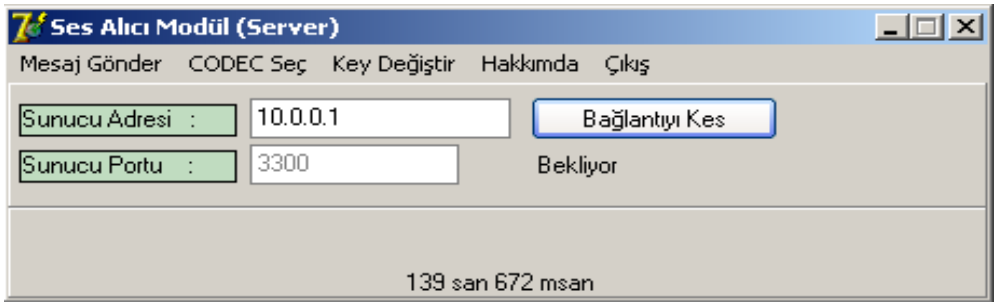


Şekil 6.7. Ses alıcı modülün veri aktarımı anındaki görünümü

Dosya aktarımı tamamlandıktan sonra kullanıcıya gelen dosyanın adı da görünerek kayıt yapması sağlanmakta ve aktarım tamamlanmaktadır. Bu işlem Şekil 6.8'de gösterilmiştir. Şekil 6.9'da kayıt işlemi tamamlandıktan sonra Ses Alıcı modül ekranı görünmektedir. Bu ekranda toplam dosya transferi süresi de görünmektedir ve bu örneğimizde tüm işlem 139 saniye 672 milisaniye sürmüştür.

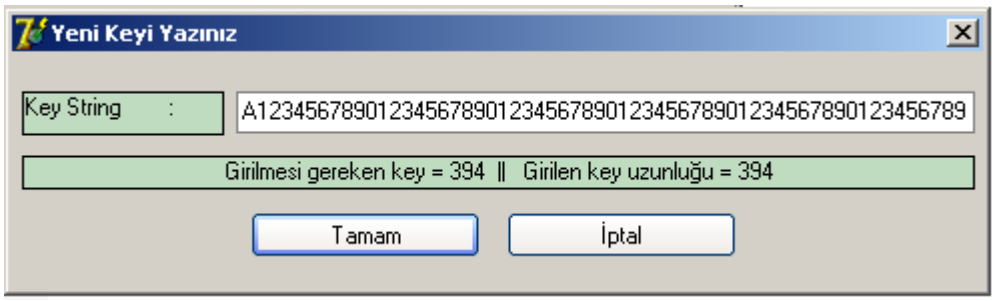


Şekil 6.8. Ses alıcı modülün dosya kayıt ekranı



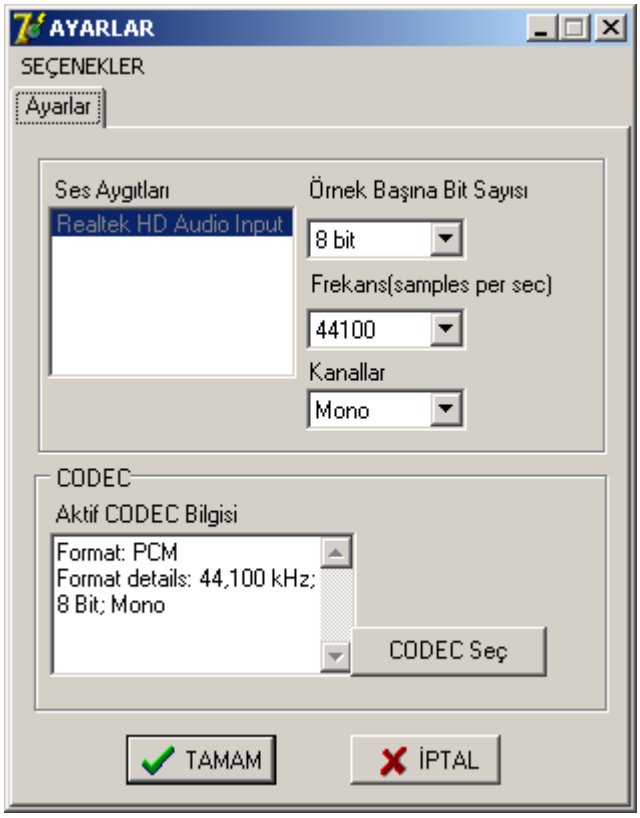
Şekil 6.9. Ses alıcı modülün dosya transferi bitmesinden sonraki ekran

Gizli haberleşme yapan kişiler şifreleme işlemi için kullandıkları ANAHTAR'ı "Key Değiştir" menüsü yardımıyla yeniden düzenleyebilirler. Şekil 6.10'da ilgili Anahtar değişimi ekranı gösterilmiştir. Her iki modülde aynı ekran görüntüsüne sahiptir.



Şekil 6.10. Şifreleme işlemi için kullanılacak anahtarı için değiştirme ekranı

Menülerde "Codec Seç" seçeneği mevcuttur. Bu menü yardımı ile CODEC seçimi yapılarak, analog ses bilgilerinin örnekleme sayısı ve her bir örneğin kaç bit ile gösterileceği (8 veya 16 bit) seçilebilmektedir. İlgili değerler seçilerek "Tamam" butonuna basıldıktan sonra iletişim başlatılır (Şekil 6.11).



Şekil 6.11. CODEC seçimi yapılmasını sağlayan menü

6.2.2. SSGD kablosuz transferi için geliştirilen yazılımın ses gönderici modülünün çalışma prensibi ve akış diyagramı

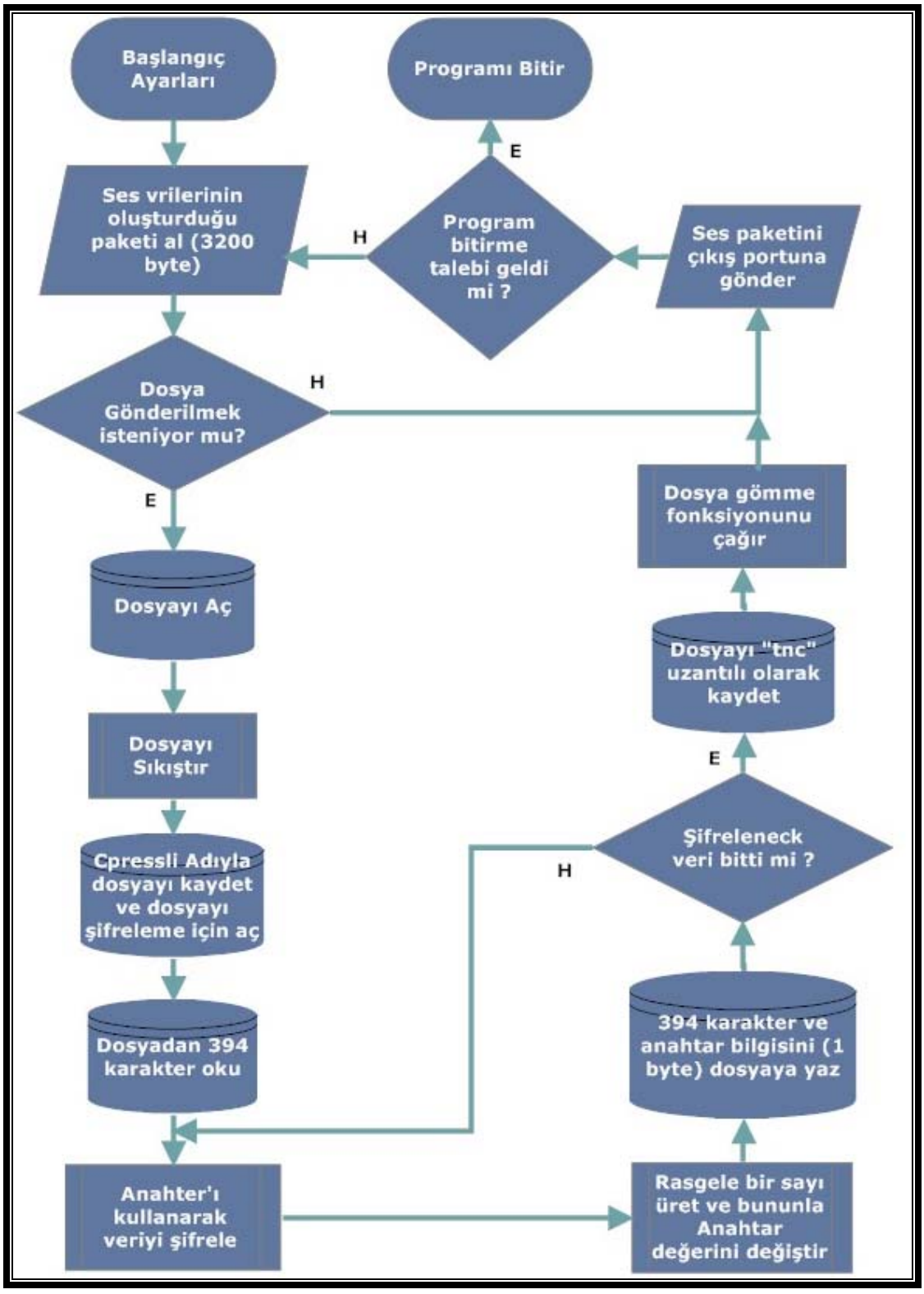
Uygulamanın bu bölümünde kaynak tarafından kablosuz ses iletimi yapılırken herhangi bir anda istenilen bir dosya gönderimi gerçekleştirilebilmektedir. İlerleme çubuğu yardımı ile dosya gönderimindeki son durum hakkında oransal olarak bilgi verilmektedir. İlgili dosyanın, ses bilgilerinin içerisine yerleştirilmesi işlemi “en düşük değerlikli bit” kullanılarak yapılmaktadır. Diğer bir deyişle, taşıyıcı ses verilerinin son bitleri, gömülecek dosyanın bitleri ile değiştirilmektedir. Birim zamanda gömülecek olan verinin diğer modül tarafından alınma süresi birim zamanda (1 saniye) ses kartından alınan ses bilgisi sayısına bağlı olarak değişmektedir. Dosya gönderimi tamamlandıktan sonra gönderme/alma, sıkıştırma ve şifreleme süreleri ilgili modülün üzerinde yazmaktadır.

Sistemden alınan ses bilgileri Ses Gönderici Modülde 3200 bayt boyuta sahip bir ara belleğe (tampon) alınmaktadır. İlgili tampon bellek dolduktan sonra işlemler gerçekleştirilmektedir. Eğer gömülecek herhangi bir dosya ya da veri yok ise ara bellekteki ilgili ses verisi bloğu porttan gönderilmektedir.

Dosya gönderme uygulamasında ses bilgisinin son bitleri göz önüne alındığında bir seferde 395 karakter gönderebilmekteyiz. Eğer kullanıcı dosya gönderilmesi yönünde eylemde bulunmuş ise, gönderilecek dosya okunarak önce Borland Delphi 7.0 programının sağlamış olduğu ZLIB kütüphanesi yardımıyla oluşturduğumuz PRESS prosedürü ile sıkıştırılır ve “cpressli” adıyla kaydedilir. Elde edilen sıkıştırılmış dosyadan 394 karakter alınır, modüllerdeki anahtar yardımıyla OTP şifreleme algoritması kullanılarak şifrelenir ve anahtar, program tarafından üretilen rasgele bir sayı kullanılarak değiştirilir. Bu sayı 1 karakterlik yer kaplar. Toplamda sıkıştırılmış dosya 395 karakterlik bilgi kümeleri halinde ikinci bir dosya olarak “tnc” uzantılı olarak kaydedililr. Bu işlemin akış diyagramı şekil 6.12’de gösterilmiştir. Daha sonra giden ilk veri bloğuna; “başlat” anahtarı (uygulamada başlat anahtarı 16 bitten oluşan 0000111100001111 olarak belirlenmektedir), dosya adının boyutu, dosya adı ve dosyanın boyutu bilgisi gömülerek gönderilmektedir (Tablo 6.1). Ardından gelen 3200 baytlık veri bloklarına dosyanın tamamı gömülmektedir. İlgili veri bloğundaki her bayt bilginin son biti değiştirilerek gömme işlemi gerçekleştirilmektedir.

Tablo 6.1. Dosya ile ilgili bilgilerin ses çerçevesi içerisindeki yerleri

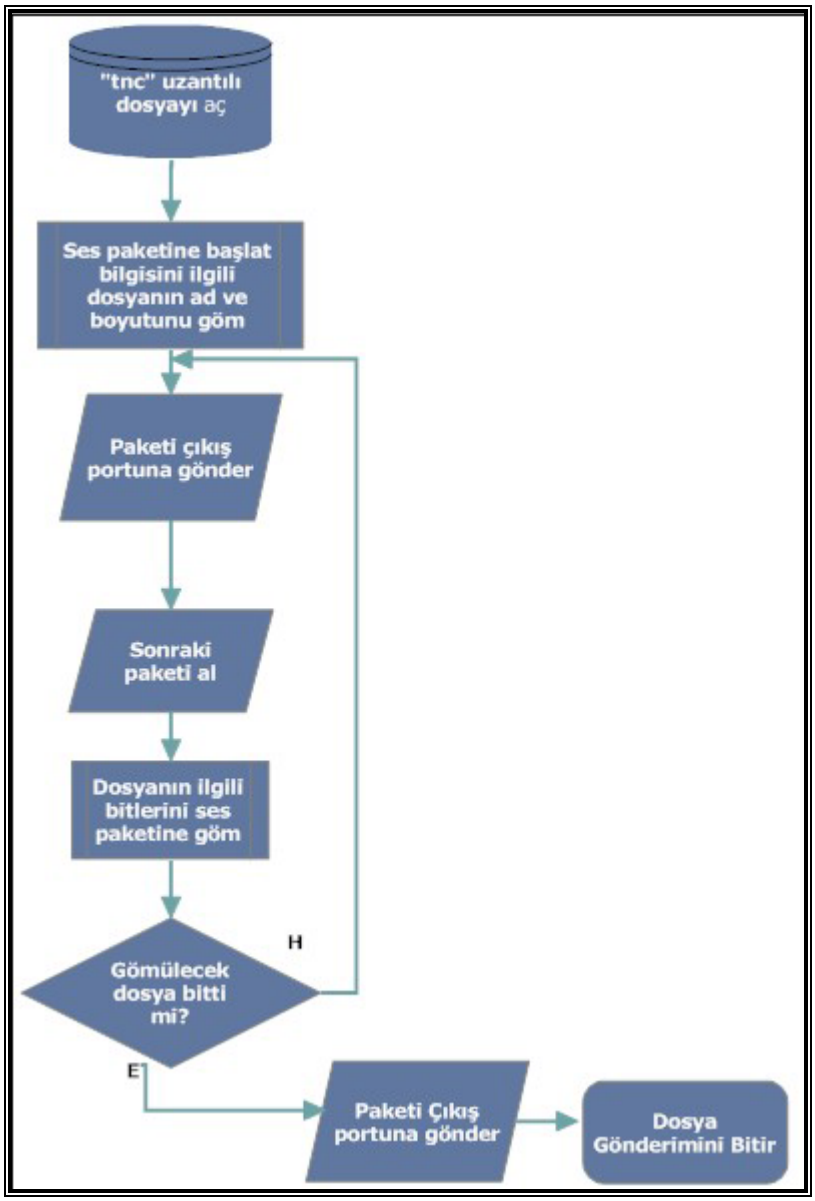
ÇERÇEVEDEKİ SIRA	GÖMÜLEN VERİ
20 — 35.bayt	Başlat biti (Burada 0000111100001111 olarak belirlenmiştir).
40 — 54.bayt	Dosya isminin uzunluğu bilgisi.
80 — 99.bayt	Gönderilecek dosyanın uzunluğu.
100 — n bayt	Dosyanın adı (Dosya ismine göre bitiş baytı(n) değişmektedir).



Şekil 6.12. Sıkıştırma ve şifreleme akış diyagramı

Dosyaya ilişkin bilgiler bir pakete gömülerek gönderildikten sonraki ilk pakete dosyanın bitleri gömülmeye başlanmaktadır. Ses bilgileri daha önceden de

belirtildiği gibi 3200 bayt boyutunda bir ara bellekte bulunmaktadır. Gömülerek gönderilecek olan dosya (bu dosya hangi tipte olursa olsun gönderilebilir) paketlerin 21—3180. baytlarının son bitlerine gömülmektedir. Diğer bir ifadeyle 3160 bayt ses verisinin son bitleri 3160 bit veri gömme alanı sağlamaktadır. Bu da paket başına 395 bayt veri gömme kapasitesi anlamına gelmektedir. Gömü Verisi (Dosyası), 395 baytlık parçalar halinde her pakete gömülmekte ve böylece dosya gönderme işlemi gerçekleştirilmiş olmaktadır (Şekil 6.13).



Şekil 6.13. Ses gönderici modülün akış diyagramı

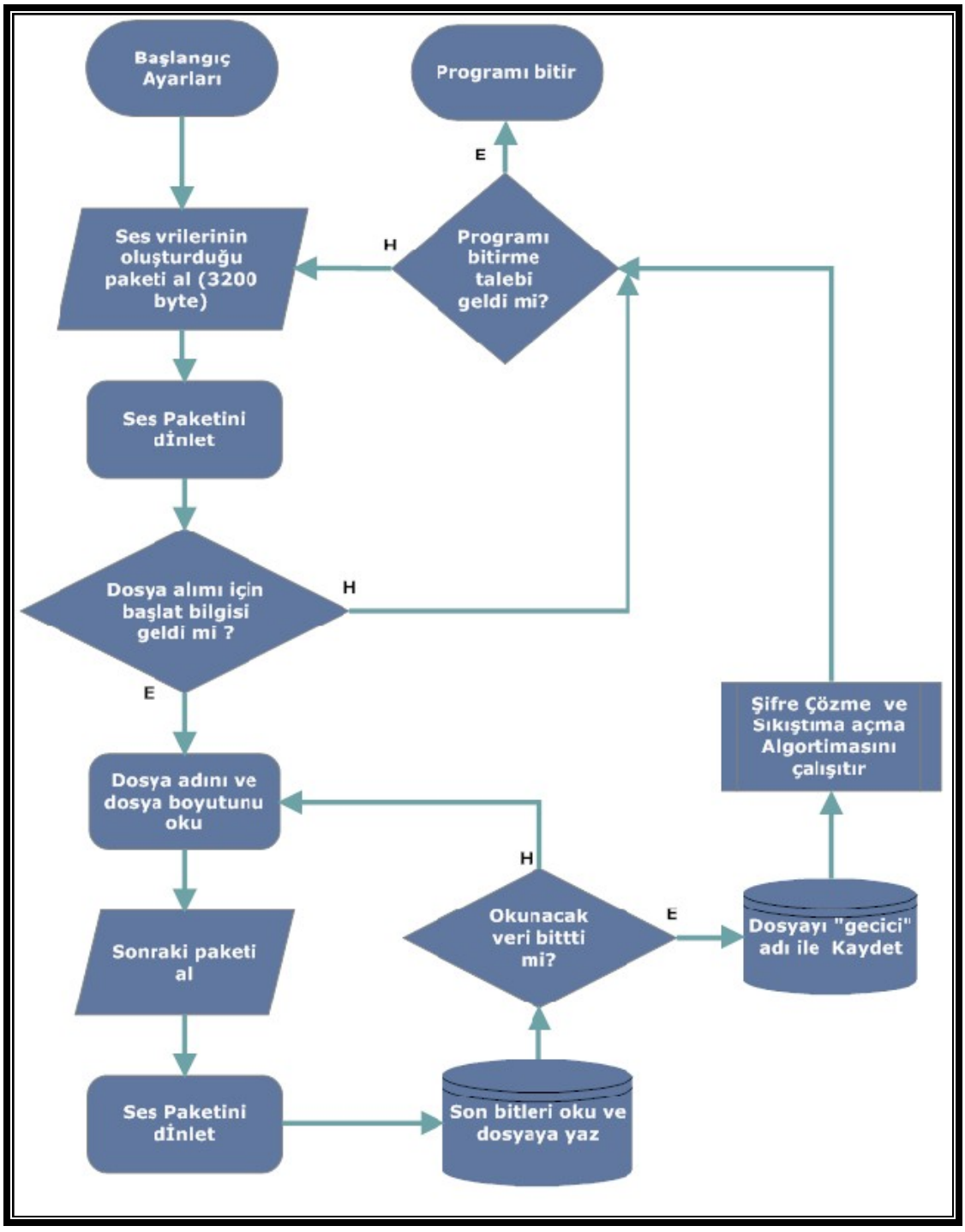
6.2.3. SSGD kablosuz transferi için geliştirilen yazılımın ses alıcı modülünün çalışma prensibi ve akış diyagramı

Uygulamanın, kablosuz olarak gönderilen ses bilgilerini alarak işleten ve eğer ses bilgileri içerisinde gömü verisi (dosyası) varsa bunu sezen ve kullanıcıya bildiren parçası bu bölümde açıklanmaktadır. İlgili bu yazılımdaki CODEC seç menusu yardımı ile kaynaktan birim zamanda (1 saniye) ses kartından kaç örnek alınacağı (8000, 11025, 18000, 22050, 32000 veya 44100) ve her bir örneğin kaç bit ile nitelendirileceği belirlenebilmektedir. Bu bilgiler oturumun açılması esnasında başlangıç ayarları yapılırken Ses Gönderici Modüle otomatik olarak gönderilmekte ve iletişim başlatılmaktadır.

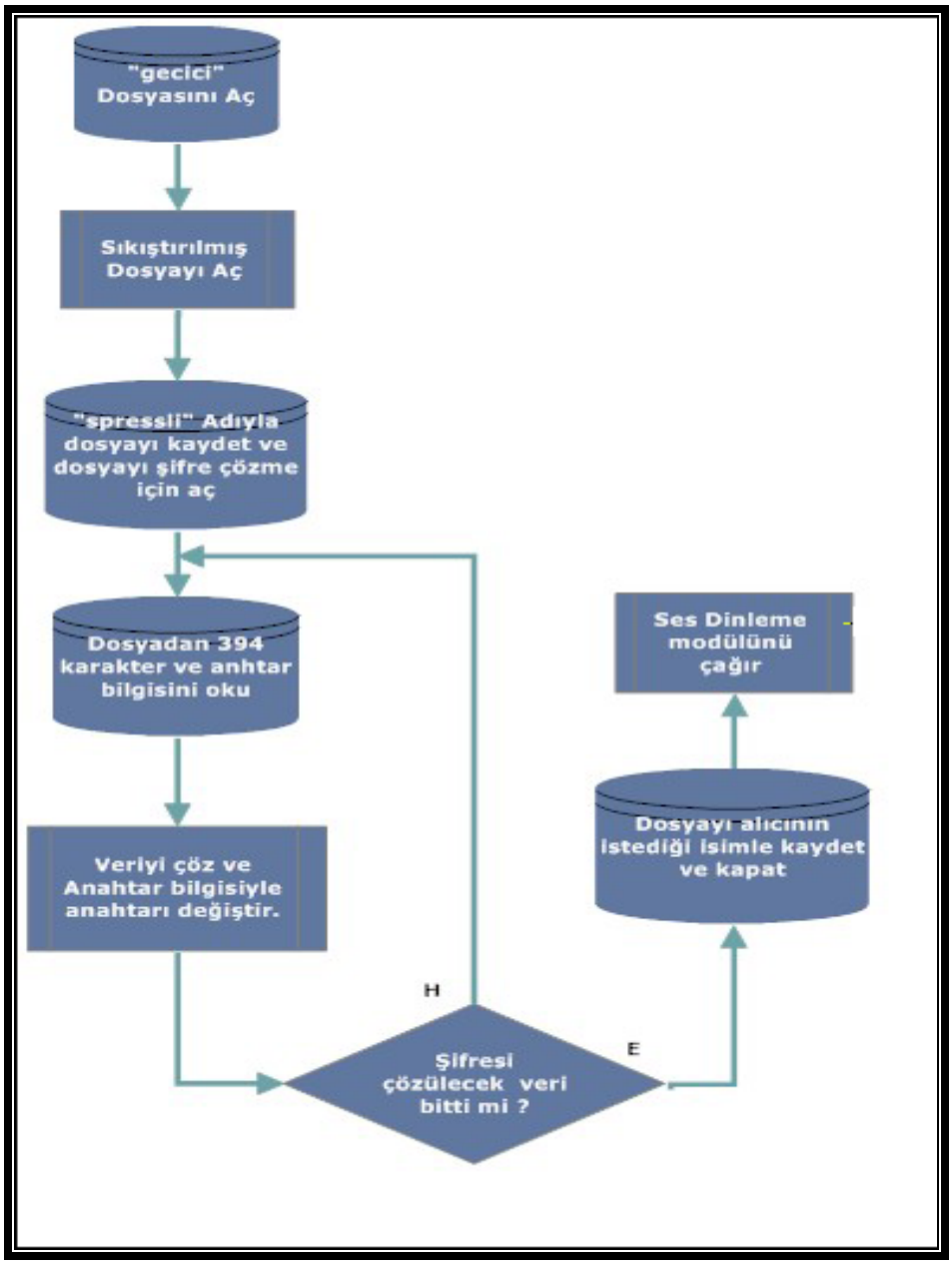
Sayısal ses verilerinden oluşan 3200 baytlık paket alındıktan sonra ilgili yazılım başlat bilgisinin gelip gelmediğini kontrol etmektedir (20—35. baytların son bitlerine bakarak). Başlat bilgisi gelse de gelmese de ses hoparlörlerde çalınmaktadır. Başlat bilgisinin gelmesi durumunda dosyaya ait bilgiler bu paketten okunarak, gelecek olan diğer paketlerden bu bilgiler ışığında, dosyanın okunması işlemi gerçekleştirilmektedir (Şekil 6.14).

Alınan dosya şifrelenmiş dosyadır. “gecici” adıyla dosya açılmaktadır. Bu dosyadan 395 karakterlik bloklar halinde okuma işlemi yapılmaktadır. Her 394 karakter asıl dosyaya ait olup son karakter bir sonraki 394 karakterlik bloğun şifresinin nasıl çözüleceğine ilişkin bilgi içerir. Şifre çözüldükten sonra dosya “spressli” ismi ile kaydedilir. Elde edilen dosya sıkıştırılmış dosyadır. Bu dosya tekrar ZLIB kütüphanesi kullanılarak açılır (Şekil 6.15). Bu işlemler bittikten sonra dosyanın hangi ad ile nereye kaydedileceği kullanıcıya sorulmakta ve aktarım işlemi sonlanmaktadır. Ancak ses iletişimi, kullanıcılar ses iletişimini kesmediği sürece devam etmektedir.

Ses alıcı yazılımında (ses gönderici de olduğu gibi) Ses Gönderici ile yazılı olarak haberleşmesini sağlayan “Mesaj Gönder” menusu mevcuttur. Bu sayede Ses Alıcı Modülü kullanan operatör istediği bir anda Ses Gönderici Modülünü kullanan operatöre yazılı mesaj gönderebilmektedir.



Şekil 6.14. Ses alıcı modülün akış diyagramı



Şekil 6.15. Sıkıştırılmış veriyi açma ve şifrelenmiş veriyi çözme işlemleri akış diyagramı

6.3. Sonuç

Bu tez çalışması için geliştirilen yazılımlarda, genel olarak daha önce kaydedilmiş metin, resim ve ses dosyalarına uygulanan sırörtme tekniğinin, gerçek zamanlı olarak yapılan kablosuz ses haberleşmesine de uygulanabileceği gösterilmektedir.

Geliştirilen uygulamada sayısal ses bilgilerine dosya gömme uygulaması gerçekleştirilmiştir. Bu işlemde kablosuz ses haberleşmesi yapılırken kullanıcının göndermek istediği herhangi bir dosya “Sıkıştırma”, “Şifreleme” ve “Sırörtme” işlemlerine tabi tutulduktan sonra sayısal ses bilgileri içerisine gömülmekte ve alıcı yazılım tarafından ilgili dosya ayırt edilerek alınmaktadır.

BÖLÜM 7. GELİŞTİRİLEN YAZILIM UYGULAMALARI

7.1. Giriş

Bu bölümde, geliştirilen yazılımların başarımlarının, çalıştırıldıkları donanımların hız ve kapasitelerinin büyüklüğüne ve diğer parametrelere ne kadar bağımlı olduğunu tespit etmek, sıkıştırma ve şifreleme uygulamalarının kablosuz ortamda dosya gönderim başarımlarına etkisini göstermek amacı ile iki farklı özelliğe sahip bilgisayarda değişik uygulamalar çalıştırılmış olup elde edilen sonuçlar tablolar ve grafikler halinde verilmektedir.

7.2. Sayısal Ses İçerisine Veri/Dosya Gömme ve Kablosuz İletimi

Kablosuz haberleşme uygulamalarında çeşitli nedenlerle veri kayıpları olmakta ve bu kayıplar geliştirilen bir takım yöntemlerle telafi edilmeye çalışılmaktadır. RF (Radyo Frekansı) etkisi sebebi ile bazı ortamlarda kablosuz olarak yapılan iletişimde istenmeyen ses bozuklukları meydana geldiği görülmektedir. Ancak uygulama çalışır iken gizli şekilde gönderilen verilerde herhangi bir bozulma meydana gelmediği tespit edilmektedir. Bunun temelinde uygulamaların geliştirildiği platform olan Borland Delphi 7.0'ın içerisindeki veri kayıplarının en aza indirgenmesini sağlayan bileşenlerden yararlanılmış olması yatmaktadır.

Ses içerisinde gömü verisi(dosyası) gönderimi (dosyalar hakkındaki bilgiler Tablo 7.1 ve 7.2'de verilmektedir) yapılmasını sağlayan modüller ile çeşitli örnek uygulamalar yapılmış olup, elde edilen sonuçlar Tablo 7.4'de gösterilmektedir. Sıkıştırma, Şifreleme ve Sırörtmenin birlikte uygulandığı örnek uygulamaları, boyutu ve dosya tipi farklı 3 adet dosya (Tablo 7.1) ve yaklaşık olarak aynı boyutlarda,

dosya türleri farklı üç adet dosya (Tablo 7.2) için gerçekleştirilmiştir. Kullanılan bilgisayarların teknik özellikleri ise Tablo 7.3'de verilmektedir.

Tablo 7.1. Ses içerisine gömülerek gönderilen gömü dosyaları-1

NO	GÖMÜ DOSYASI ADI	DOSYA UZANTISI	DOSYA BOYUTU (BYTE)
1	demo	mp3	38912
2	sndrec32	exe	124928
3	svega	wav	1823640

Tablo 7.2. Ses içerisine gömülerek gönderilen gömü dosyaları-2

NO	GÖMÜ DOSYASI ADI	DOSYA UZANTISI	DOSYA BOYUTU (BYTE)
1	Ares	doc	1479680
2	İstiklal	wav	1499382
3	Kutuphane	exe	1466408

İlgili dosyalar sıradan bilgisayar kullanıcılarının ulaşabileceği dosyalardır. "demo.mp3" 5 saniyelik bir ses dosyası olup, müzik dinlemek için kullanılan Winamp programı kurulduğunda elde edilebilmektedir. "sndrec32.exe" dosyası Microsoft Windows'un ses kayıt programıdır. "svega.wav" dosyası ise bilimsel çevrelerde Sırörtme uygulamalarında kullanılan 20 saniyelik bir referans ses dosyasıdır. "Ares.doc", "İstiklal.wav" ve "Kutuphane.exe" dosyaları ise yapılan uygulamanın aynı boyutlarda fakat farklı tip dosyalardaki etkisini göstermek için kullanılmıştır.

Tablo 7.3. Kullanılan bilgisayarların donanım özellikleri

BİLGİSAYAR ADI	İŞLEMCİ TİPİ	HIZ (GHZ)	BELLEK BOYUTU
PC_A	Intel(R)Pentium(R)4 CPU	3,2	384 MB RAM
PC_B	Intel(R)Celeron Mobile CPU	1,8	224 MB RAM

Özellikle gerçek zamanlı kablosuz haberleşme uygulamalarında kullanılan donanımların birim zamanda işlem yapabilme kapasitelerinin yüksekliği, uygulamaların sağlıklı şekilde gerçekleştirilmesi açısından hayati önem taşımaktadır. Tez çalışmasına konu olan uygulamaların geliştirilmesi aşamalarında bu ilke dikkate alınmış olup, kullanılan bilgisayarlar birbirinden farklı teknik özelliklere sahiptir. Bu sayede yapılan denemelerde elde edilen sonuçlar üzerine yoğunlaşıldığında bilgisayarların hız ve kapasitelerinin haberleşmenin başarımına ne kadar etki ettiği de ortaya çıkarılmaktadır. Dosyalara, sırörtme yöntemi kullanılarak gönderilmeden önce sıkıştırma ve şifreleme işlemleri uygulanır. Sıkıştırma için Borland Delphi 7.0'ın kendi kütüphanesinde bulunan ZLIB sıkıştırma fonksiyonları kullanılmıştır. Şifreleme için ise OTP (One Time Pad - Tek Zamanlı Deste) yöntemi kullanılmıştır. Uygulamada kullanılan dosyaların sıkıştırma ve şifreleme işlemlerinden sonraki boyutları Tablo 7.4'de gösterilmektedir.

Tablo 7.4. Kullanılan dosyaların sıkıştırılmış ve şifrelenmiş dosya boyutları

Dosya İsmi	**Orijinal Dosya Boyutu (Byte)**	**Sıkıştırılmış Dosya Boyutu (Byte)**	**Şifrelenmiş Dosya Boyutu (Byte)**
Demo.mp3	38912	36931	37025
Sndrec32.exe	124928	60582	60736
Svega.wav	1823640	930111	932472
Ares.doc	1479680	785459	787453
İstiklal.wav	1499382	730565	732420
Kutuphane.exe	1466408	1361429	1364885

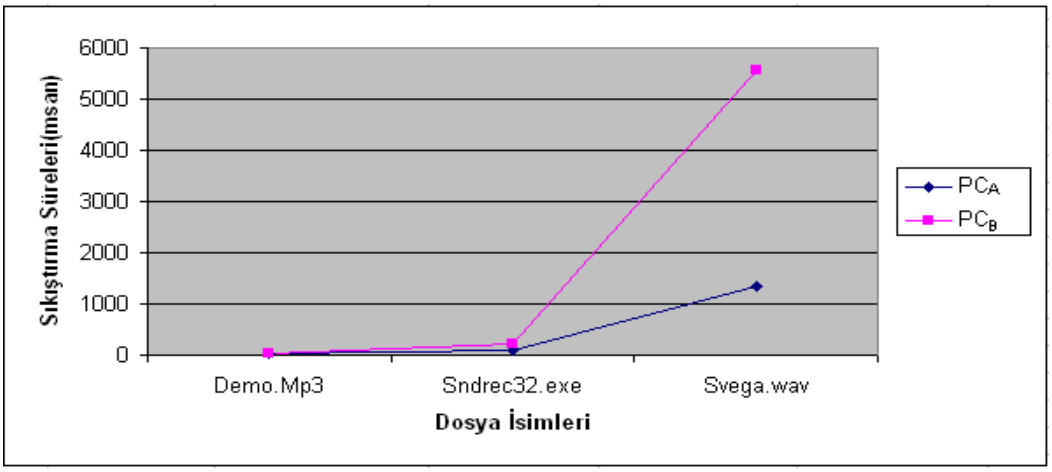
Özellikle kablosuz haberleşme sistemlerinde veri transferi için harcanan zaman önemlidir. Süre düştükçe bilgisayarların işlem yükü düşecek, performansları artacaktır. Bunun kullanılan batarya süresine de etkisi olumlu yöndedir. Yapısı itibariyle fazla artıklık içermeyen mp3, exe gibi dosyalarda sıkıştırma performansı düşük olacaktır. Şifrelemeden sonra dosya içerisine anahtar ile ilişkili bilgiler gömüldüğü için (her 394 byte için 1 byte) dosya boyutu küçük bir oranda artacaktır (1 / 394 oranında). Ancak bu dosya güvenliği için gözden çıkarılabilir bir fedakarlıktır. Dosya boyutu arttıkça ya da içerisinde artıklık fazla olan dosya tipleri

(wav, doc gibi uzantılı dosyalar) uygulandıkça sıkıştırmanın veri iletimine faydası fazla olacaktır.

Tablo 7.5. PC_A ve PC_B'nin sıkıştırma süreleri

Dosya İsmi	**Sıkıştırma İşlemi(msan)**	
	PC_A	**PC_B**
Demo.Mp3	16	20
Sndrec32.exe	94	231
Svega.wav	1359	5578

Tablo 7.5'de PC_A ve PC_B'nin uygulama dosyalarını sıkıştırma işlemi süreleri gösterilmiştir

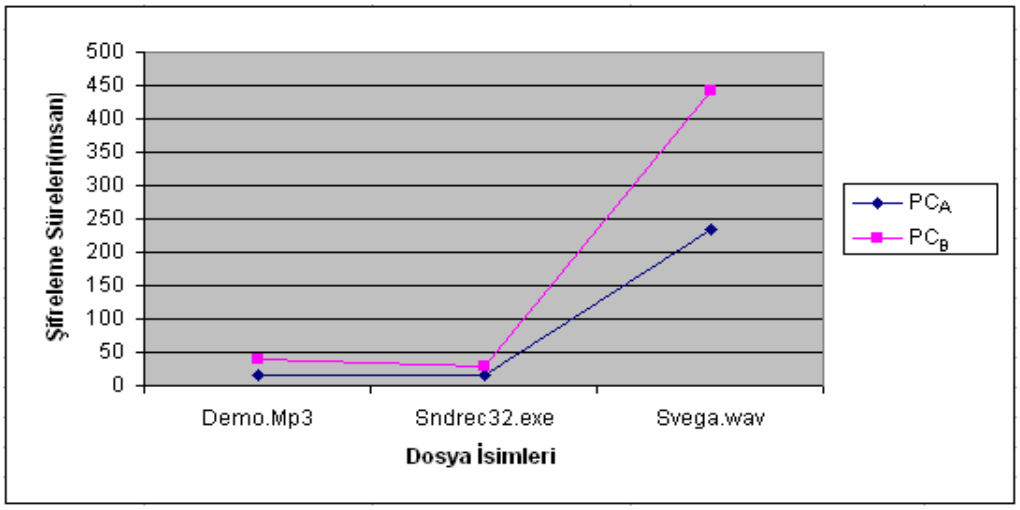


Şekil 7.1. PC_A ve PC_B'nin mili saniye türünden dosya sıkıştırma başarım grafikleri

Şekil 7.1'de, uygulama bilgisayarlarının dosyaları sıkıştırma süreleri grafik olarak gösterilmiştir. Daha hızlı olan PC_A'nın PC_B'den daha iyi bir başarım gösterdiği Tablo 7.5 ve Şekil 7.1'de görülmektedir.

Tablo 7.6. PC_A ve PC_B'nin şifreleme süreleri

Dosya İsmi	Şifreleme Süresi(msan)	
	PC_A	**PC_B**
Demo.Mp3	15	40
Sndrec32.exe	16	30
Svega.wav	234	441

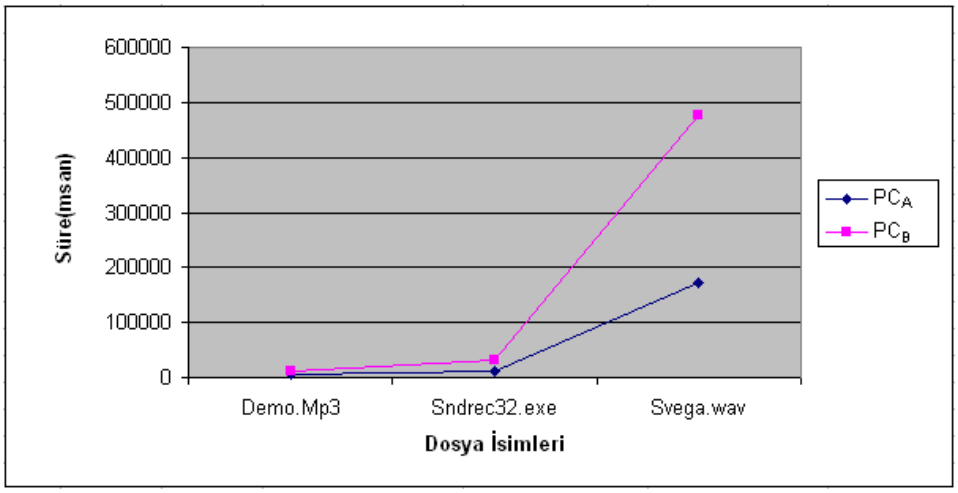


Şekil 7.2. PC_A ve PC_B'nin mili saniye türünden dosya şifreleme başarım grafikleri

Tablo 7.5'de PC_A ve PC_B'nin uygulama dosyalarını şifreleme süreleri milisaniye cinsinden gösterilmiştir. Daha hızlı olan PC_A ile PC_B arasında sıkıştırma ve şifreleme süreleri bakımından küçük dosyalarda fark az olmasına karşın dosya boyutu daha büyük olan "Svega.wav" dosyasında PC_A , şifreleme işleminde yaklaşık 2 kat, sıkıştırma işleminde ise 3 kat daha iyi başarım göstermiştir.

Tablo 7.7. PC_A ve PC_B'nin toplam dosya gönderme süreleri

Dosya İsmi	Toplam Gönderme Süresi	
	PC_A	**PC_B**
Demo.Mp3	7031	12328
Sndrec32.exe	11625	32808
Svega.wav	173093	477697

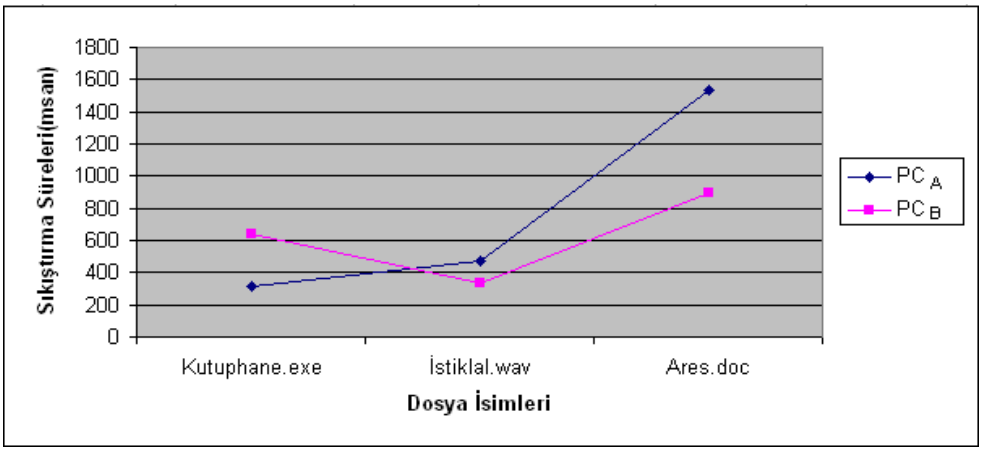


Şekil 7.3. PC_A ve PC_B'nin mili saniye türünden dosya gönderme başarım grafikleri

Tablo 7.6'de PC_A ve PC_B'nin uygulama dosyalarını toplam gönderme sürelerini milisaniye cinsinden göstermektedir. Bu değerler grafik olarak Şekil 7.3'de sunulmuştur. En küçük boyutlu dosya olan Demo.mp3 dosyasının gönderilmesinde PC_A'nın PC_B'den yaklaşık 5300 milisaniye(5.3 saniye) daha iyi başarım gösterdiği belirlenmiştir. Dosya gönderme ve alma işlemlerinde veri iletim hızı 54Mbs, 1sn'deki örneklem sayısı 44100'dür. Yukarıda, farklı boyutlarda 3 farklı dosya üzerindeki başarımlar gösterilmiştir. Aşağıdaki tablo ve grafiklerde yapılan uygulamanın yaklaşık aynı boyutlarda ama farklı tür dosyalar üzerindeki başarımları incelenmektedir.

Tablo 7.8. PC_A ve PC_B'nin sıkıştırma süreleri

Dosya İsmi	Sıkıştırma İşlemi	
	PC_A	**PC_B**
Kutuphane.exe	312	641
İstiklal.wav	469	331
Ares.doc	1531	891

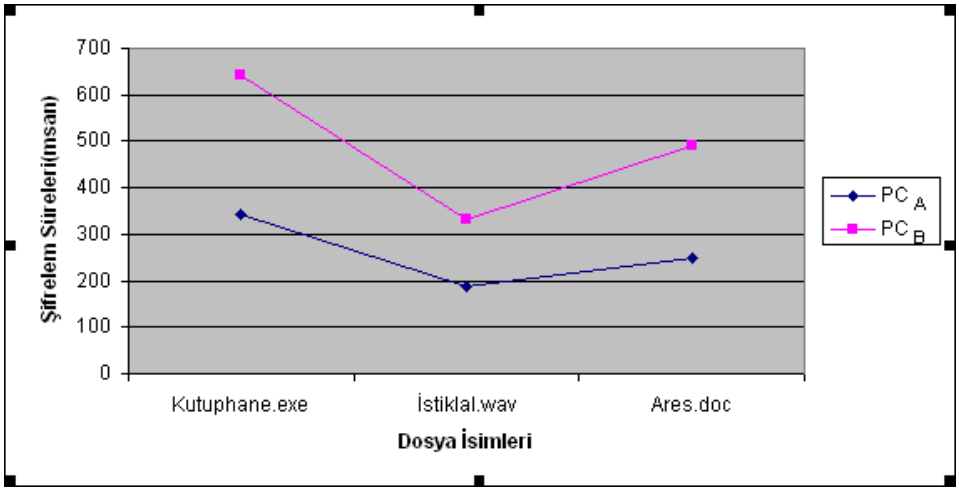


Şekil 7.4. PC_A ve PC_B'nin mili saniye türünden dosya sıkıştırma başarım grafikleri

Uygulaması yapılan boyutları bir birine yakın 3 farklı dosya tipinin sıkıştırma başarımlarını etkiledikleri görülmektedir. "Wav" uzantılı dosyalar sıkıştırılabilirlik açısından "exe" uzantlı dosyalardan daha elverişlidir. Bu elverişlilik dosya sıkıştırma başarımlarını da etkilemektedir.

Tablo 7.9. PC_A ve PC_B'nin şifreleme süreleri

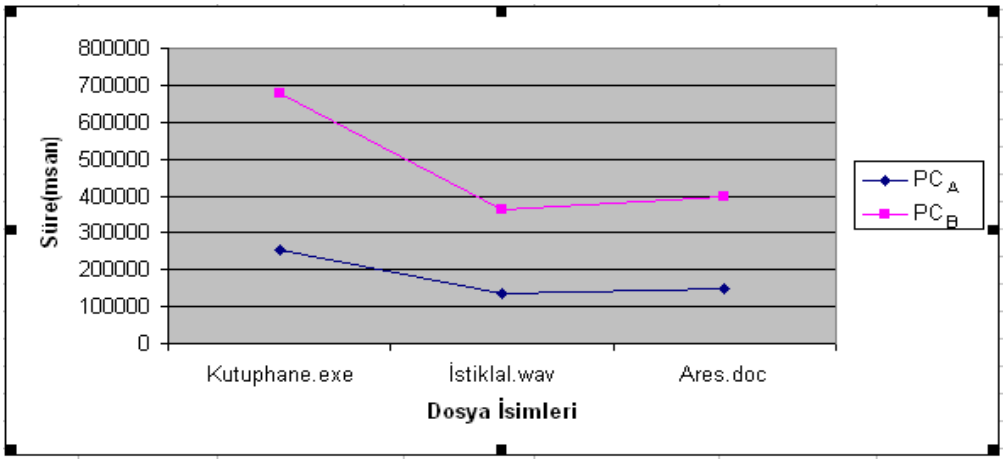
Dosya İsmi	Şifreleme Süresi	
	PC_A	**PC_B**
Kutuphane.exe	344	641
İstiklal.wav	188	331
Ares.doc	250	491



Şekil 7.5. PC_A ve PC_B'nin mili saniye türünden dosya şifreleme başarım grafikleri

Tablo 7.10. PC_A ve PC_B'nin şifreleme süreleri

Dosya İsmi	Toplam Gönderme Süresi	
	PCA	PCB
Kutuphane.exe	251656	678445
İstiklal.wav	135407	362832
Ares.doc	146781	396199



Şekil 7.6. PC_A ve PC_B'nin mili saniye türünden dosya gönderme başarım grafikleri

Yukarıdaki tablo ve şekillerde PC_A'nın PC_B'den daha iyi başarımlar göstermiştir. Ancak programın çalışması esnasında bilgisayara gelen kesme komutları,

bilgisayarın işlemcisinin ısınması gibi ortamdan veya bilgisayarın çalışma performansından kaynaklanan nedenler bütün bu uygulamaları etkilemektedir. Özellikle dosya gönderme işlemlerinde kablosuz ortam kullanıldığı için kablosuz iletişim ortamındaki cisimlerin etkisi, kablosuz haberleşme başarımın etkilemektedir.

Yalman, 2007 yılında yapmış olduğu "Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi" adlı yüksek lisans tezinde Tablo 7.11'deki sonuçları sunmuştur.

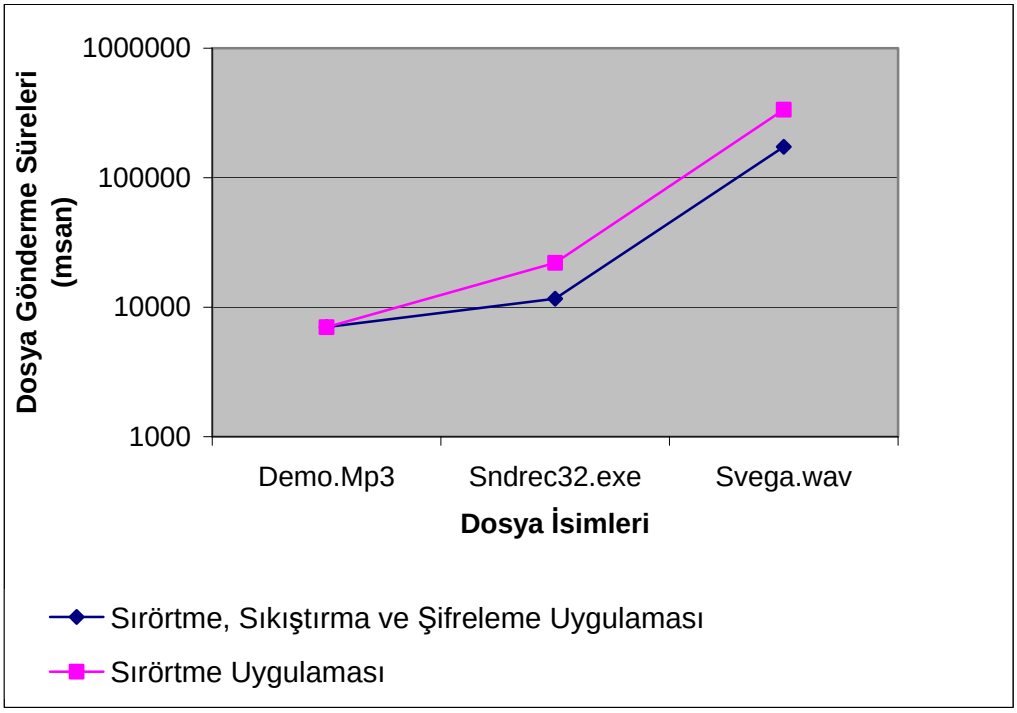
Tablo 7.11. PC_A ve PC_B'nin Dosya alma ve gömme süreleri (Yalman, 2007)

GÖMÜ DOSYASININ ADI	**DOSYA GÖMME SÜRESİ (SN)**		**GÖMÜLÜ DOSYAYI ALMA SÜRESİ (SN)**	
	PC_A	**PC_B**	**PC_A**	**PC_B**
Demo.mp3	7	12	13	10
sndrec32.exe	22	37	38	33
svega.wav	335	520	522	470

Bu tez çalışmasının başarımını değerlendirmek için aynı dosyalara sırörtme uygulamasına ek olarak, sıkıştırma ve şifreleme uygulanmıştır. Aşağıdaki tabloda bu dosyaların başarımları gösterilmiştir.

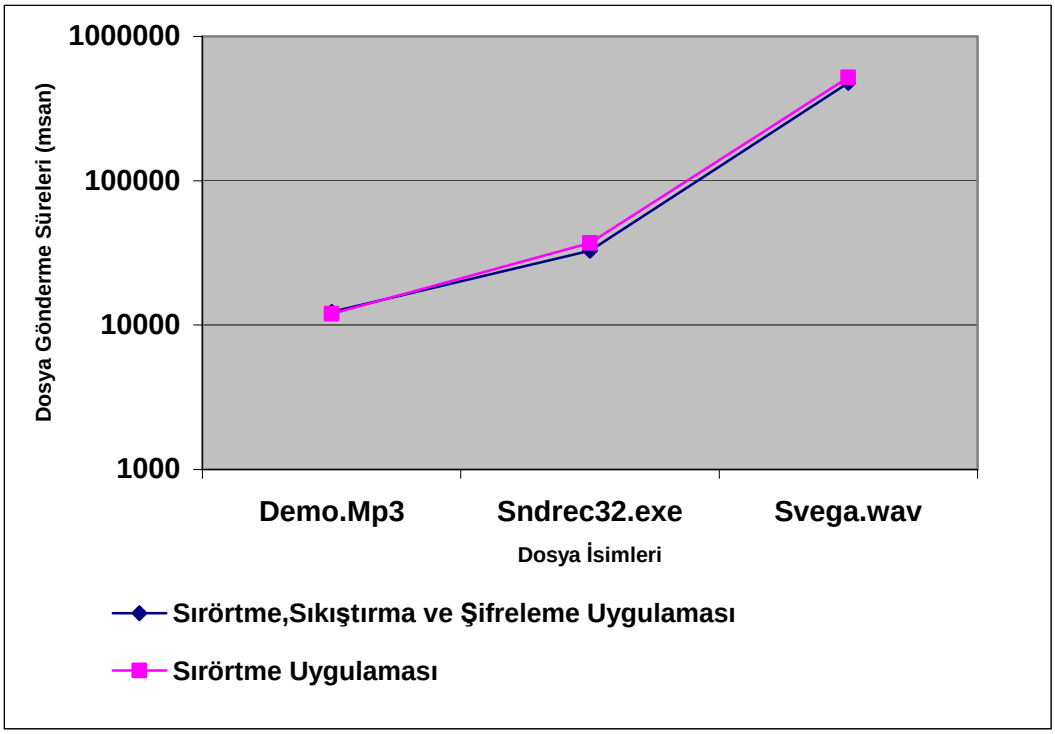
Tablo 7.12. PC_A ve PC_B'nin sırörtme, sıkıştırma ve şifreleme başarımı

Dosya İsmi	**Toplam Gönderme Süresi(msan)**		**Toplam Alma Süresi(msan)**	
	PC_A	**PC_B**	**PC_A**	**PC_B**
Demo.Mp3	7031	12328	11922	9914
Sndrec32.exe	11625	32808	31937	16023
Svega.wav	173093	477697	468781	234137



Şekil 7.7. Yapılan sırörtme, sıkıştırma ve şifreleme uygulaması ile sırörtme uygulaması dosya gönderme başarım grafikleri (PC_A)

Şekil 7.7'de yukarıda bahsedilen uygulamaların karşılaştırılmaları mili saniye türünden grafik olarak yapılımıştır. Bu grafik PC_A'nın dosya gönderme başarımlarını karşılaştırmaktadır.



Şekil 7.8. Yapılan sırörtme, sıkıştırma ve şifreleme uygulaması ile sırörtme uygulaması dosya gönderme başarım grafikleri (PC_B)

Şekil 7.8'de yukarıda bahsedilen uygulamaların karşılaştırılmaları mili saniye türünden grafik olarak yapılımıştır. Bu grafik PC_B'nın dosya gönderme başarımlarını karşılaştırmaktadır.

Dosya boyutu küçük olan Demo.mp3 dosyasında, yapılan uygulamaların başarım süresi birbirine yakın çıkmıştır. Ancak sıkıştırma ve şifreleme uygulaması, sırörtme uygulamasına ek güvenlik sağlamıştır. Sıkıştırma işlemi, şifreleme için kullanılan zamanı telafi etmektedir. Sıkıştırmanın sağlamış olduğu başarım dosya gönderme süresini de düşürmüştür.

Dosya boyutu arttıkça sırörtme, sıkıştırma ve şifreleme uygulamasının, sadece sırörtme uygulamasına üstünlükleri gözlemlenmiştir. Uygulamada sıkıştırma işleminden sonra şifreleme yapmak dosya boyutunu büyütmektedir. Ancak, veriyi ele geçiren kişi şifreyi çözebilse dahi anlamlı veri elde edemeyecektir. Bu işlem 3.şahısların gömü verisini elde etmesini zorlaştıracaktır.

BÖLÜM 8. SONUÇLAR VE ÖNERİLER

Sayısal ses içerisine sıkıştırılmış ve şifrelenmiş dosya gömme ve açma algoritmaları ve ara yüzlerinin geliştirildiği ve gerçekleştirildiği bu tez çalışmaları kapsamında elde edilen sonuçlar ışığında, konuya ilgi duyan araştırmacılara ve bilim camiasına aşağıdaki öneri/tartışma ve değerlendirmelerin sunulması uygun görülmektedir.

1. Ses haberleşmesinin yapıldığı esnada veri/dosya gömülmeye başlandığı andan itibaren insan kulağının oldukça zor algıladığı periyodik bir bozulma meydana gelmektedir. Bu durum hem paketlerdeki sayısal ses bilgilerinin değiştirilmesinden hem de paketlerin veriyi/dosyayı ayırt eden algoritmaya sokulması ile meydana gelen zaman kaybından kaynaklanmaktadır. Daha hızlı çalışabilecek algoritmaların geliştirilmesinin bu durumu olumlu yönde etkileyeceği anlaşılmaktadır.

2. Sayısal ses bilgileri içerisine veriler/dosyalar gömülür iken son bitler kullanılmıştır. Birim zamanda gömülen veri/dosya boyutunun arttırılması amacıyla her ses örneği 16 bit ile nitelendirilip gömülecek bit sayısı 2 ya da 3'e çıkarılabilir.

3. Gömü Dosyasının gönderilme süresi, ilgili dosyanın boyutu ile doğru orantılı olarak arttığından gömülecek bit sayısının azaltılması için sıkıştırma algoritması uygulanmış ve özellikle büyük boyutlu dosyalarda sadece sırörtme uygulamasına göre daha hızlı dosya gönderim ve alım zamanı elde edilmiştir. Ayrıca gönderilecek bit miktarı düştüğünden sadece sırörtme yapılan uygulamaya göre toplamda sesteki bozulma miktarı da düşmüştür.

4. Bilindiği üzere sırörtmede gizli bilgiler yalnızca kaynak ve alıcı algoritmanın bildiği şekilde gömülmektedir. Ancak bu gömme şeklinin üçüncü kişilerce

bilinme ihtimali de dikkate alındığında gömülecek verilere şifreleme uygulanmıştır. Kullanılan şifreleme algoritmasında gerçek zamanlı olarak alıcı ve gönderici modüllerdeki anahtar değişimleri verinin güvenliğini arttırmaktadır. Üçüncü şahısların anahtarı tahmin etmek istemeleri durumunda her pakette 394^{255} tane anahtar tahmin etmeleri gerekmektedir. 1 MB'lık bir dosyada yaklaşık olarak 2654 adet paket mevcuttur. Bu sonuçlar bu uygulama için şifre çözme işleminin zorluğunu göstermektedir.

5. Dosya gönderimi yapılırken ses verilerini dinleyen üçüncü kişiler gizli veriyi elde edebilmek için şu aşamalardan geçmeleri gerekmektedir:
   a- Yapılan sırörtme uygulamasının nasıl yapıldığını bilmeleri gerekir(Ses verisinin son bitlerine veri gömülmesi).
   b- Kullanılan dosya gönderim biçimini(başlat biti, dosya adının uzunluğu bilgisi gibi) bilmeleri gerekir.
   c- Dinlenen ses verilerinden elde ettikleri dosya şifreli olacağından, hangi şifreleme yöntemi(OTP), başlangıç şifresi ve şifre çözümü için anahtar bilgisinin nasıl olduğunu bilmeleri gerekir. Üçüncü bölümde anlattığımız OTP şifreleme yönteminin güçlü yanlarından biri, kişi yanlış anahtarı kullansa bile anlamlı veri elde etmektedir. Bu da asıl anahtara sahip olmayan birinin gerçek veriye ulaşıp ulaşamadığı bilgisini saklı tutacaktır.
   d- Şifre çözme işlemini başarıyla gerçekleştirdiklerinde eldeki dosya sıkıştırılmış dosya olacağından sıkıştırma algoritmasının kullandığı fonksiyonları (ZLIB kütüphanesi) tahmin etmeleri gerekir.
   e- Bütün bunlara(algoritma, kod, anahtar) sahip dışarıdan biri ancak diğer iki kişi dosya gönderimini başlatır başlatmaz paketleri ele geçirmelidir. Bu paketlerden herhangi birini(1 MB = 2654 paket) dinleyemez ise dosyanın tamamını çözmesi mümkün olmayabilir.

6. Bugün gerek bilgisayar ağlarında ve gerekse internet ortamında çok sayıda ses verisi mevcut olup, bunlardan hangisinin veri gizlediğini tahmin etmek oldukça zordur. Bu da sırörtmenin şifreleme bilimine karşı üstünlüğü olarak görülebilir. Ancak şifreleme için harcanan çok kısa bir süre (dosya boyutuna göre milisaniyeler mertebesinde olabilmektedir) veri güvenliğini arttırmaktadır.

Kullanılan tekniklerde kod çözme esnasında orijinal ses verisine ihtiyaç duyulmaması da bir diğer önemli özelliği oluşturmaktadır.

7. Günümüzde telif haklarının (copyright) korunması uygulamalarından olan sanal sayısal filigran (Digital Watermarking) teknolojilerinde sıkça sırörtme uygulamalarını görmek mümkündür. Bir kişiye ait olan orijinal bir çalışma (resim, ses vb.) başkaları tarafından izin alınmadan sahiplenilmesi yine bu tekniklerle önlenmektedir. Orijinal obje üzerine yerleştirilen gizli tanıtıcı veriler, nesnenin sahibine işaret etmektedir. Bu bakımdan da gerçekleştirilen çalışmaların oldukça faydalı ve uygulanabilir alanları olduğu görülmektedir

Bilgisayar donanım özellikleri (işlemci türü, işlemci hızı, RAM bellek) iyileştikçe yapılan uygulamanın daha sorunsuz çalıştığı tespit edilmiştir. Gelecekteki donanım konfigürasyonlarının çok daha gelişmiş olacağı göz önüne alındığında gömme ve gömülü veriyi/dosyayı ede etme süreleri daha da kısalacak ve böylece akıcı (streaming) görüntü/video uygulamalarının da veri gizleme amaçlı kullanımında alternatif olacağı düşünülmektedir.

İleride Yapılabilecek Çalışmalar :

1. Uygulamalar yapılırken RF etkisinin söz konusu olduğu ortamlarda seste bozulmalar meydana geldiği gözlemlenmiştir. Bu bozulmaların en aza indirgenmesini sağlamak için çeşitli algoritmalar geliştirilebilir.

2. Gerçekleştirilen örnek uygulamalarda aynı anda sadece bir tek dosya gömülerek gönderilmesi söz konusudur. Bu durum yapılacak olan bir algoritma ile geliştirilerek aynı anda birden fazla dosyanın gönderimi gerçekleştirilebilir.

3. Gönderici ve alıcının kimlik bilgileri bu uygulamada göz önüne alınmamıştır. Ancak eklenecek bir modül sayesinde göndericinin kim olduğu anlaşılabilir. Bu işlem, diğer şahısların programı ele geçirmeleri ve anahtar dosyalarını bulmaları halinde bile güvenliği arttırıcı olabilir. Alıcı, kimliğini bilmediği birinden gelen mesajı göz ardı edecektir.

4. Geliştirilen uygumla sesli görüşme yaptığı için geliştirilecek bir modül sayesinde ses verilerinden kimlik tespiti yapılabilir.Bu işlem veri gönderimi yapan kişi hakkında daha kesin bilgi verebilir.

5. Çalışmalar bilgisayara bağımlı olduğu için tam anlamıyla istenen hızda sonuçlara veya başarıma ulaşmak mümkün olmamaktadır. İlgili uygulamalar eğer bilgisayardan bağımsız platformda gerçekleştirilirse daha fazla verim elde edilecektir.

KAYNAKLAR

[1] COX, I. J., MİLLER, M. L., BLOOM, J. A., Watermarking Applications and Their Properties, Int. Conf. on Information Technology, Las Vegas, USA, 2000

[2] COX, I. J., MİLLER, M.L., The First 50 Years of Electronic Watermarking, Journal of Applied Signal Processing, Vol. 18, No.4, pp 128–132, 2002

[3] ANDERSON, R., J., PETİTCOLAS, F. A. P., On the Limits of Steganography, IEEE Journal of Selected Areas in Communications, Vol. 18, No.4, pp 474–481, 1998

[4] Tim Ho Tin Woo, A Scalable, Secure, and Energy-Efficient Image Representation for Wireless Systems, University of Waterloo in Electrical and Computer Engineering, Waterloo, Ontario, Canada, 2004

[5] http://bilimadami.net/blog/2008/10/24/sifrelemenin-tarihi/ 23.03.2008

[6] HARTUNG, F., KUTTER, M., Multimedya Watermarking Techniques, Proceedings of the IEEE, Vol.87, No.7, pp 1079–1107, 1999

[7] DELAİGLE, J. K., Protection of Intellectual Property of Images by Perceptual Watermarking, Doktora Tezi, Universitè Catholique de Louvain, 2000

[8] CHENG, J., KOT, A.C., LİUAND, J., CAO, H., Steganalysis of Data Hiding in Binary Text Images, Proceedings of the IEEE, pp 4405–4408, 2005

[9] ADLI, A., NAKAO, Z., Three Steganaography Algorithms for MIDI Files, IEEE Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, 2005

[10] XU, C., PİNG, X., ZHANG, T., Steganography in Compressed Video Stream, Proceedings of the First International Conference on Innovative Computing, IEEE, 2008

[11] ÖZEL,İ., ÇALIKOĞLU, D., Teknik Birlik Dergisi, , Cilt 1, Sayı3, s.37,Yıl 1988

[12] KENNETH BARR AND KRSTE ASANOVİ´C, Energy Aware Lossless Data Compression, The First International Conference on Mobile Systems, Applications, and Services, San Francisco, CA,May 2003

[13] CHRİSTİAN KRATZER, JANA DİTTMANN, THOMAS VOGEL, REYK HİLLERT, Design and Evaluation of Steganography for Voice-over-IP, Advanced Multimedia and Security Lab (AMSL) Otto-von-Guericke-Universitat, Magdeburg, Germany, 2006

[14] Chang, L., Moskowitz, I., Critical Analysis of Security in Voice Hiding Techniques, Information Technology Division, MaLl Code 5540, Center for High Assurance Computer Systems, Naval Research Laboratory, Washington, DC 20375 USA

[15] YALMAN,Y., Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi, Yüksek Lisans Tezi, s.4, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, 2007

[16] BAYILMIŞ, C., Kablosuz Bilgisayar Ağlarının Performans Analizi, Yüksek Lisans Tezi, Sakarya Üniversitesi Fen Bilimleri Enstitüsü, 2003

[17] BAYILMIŞ, C., ERTÜRK, I., ÇEKEN, C., Kablosuz Bilgisayar Ağlarının Karşılaştırılmalı İncelemesi, Gazi Üniversitesi Politeknik Dergisi, Cilt 7, Sayı 3, 201–210, 2004

[18] BAYILMIŞ, C., ERTÜRK, I., ÇEKEN, C., A Comparative Performance Evaluation Study of IEEE 802.3 Wired and IEEE 802.11 Wireless LANs for Multimedia Data Traffic, Journal of Naval Science and Engineering, 2, 1–12, 2004

[19] NİCOPOLİTİDİS, P., OBAİDAT, M., S., PAPADİMİTRİOU, G., I., POMPORTSİS, A., S., Wireless Networks, Wiley, 239–289, 2003.

[20] GAST, M., 802.11 Wireless Networks: The Definitive Guide, Second Edition, Q'Reilly, 2005

[21] BİNG, B., High–Speed Wireless ATM and LANs, Artech House Mobile Communications Library, 1–102, 2000

[22] ÇEKEN, C., ERTÜRK, I., BAYILMIŞ, C., Wireless Networks for Real–Time Multimedia Communications, Broadband Wireless and WiMAX, Comprehensive Report by International Engineering Consortium (IEC), 2004

[23] ANSI/IEEE Std 802.11, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standards 802.11, 70–90, 1999

[24] AAD, I., CASTELLUCCİA, C., Priorities in WLANs, Computer Networks, Vol. 41, 505–528, 2003

[25] AKAR, F., Veri Gizleme ve Şifreleme Tabanlı Bilgi Güvenliği Uygulaması, Doktora Tezi, Marmara Üniversitesi Fen Bilimleri Enstitüsü, 2005

[26] Toyran, M., Kuantum Kriptografi (Quantum Cryptography), Tübitak, Uekae, mtoyran@uekae.tubitak.gov.tr

[27] http://www.csharpnedir.com/makalegoster.asp?MId=223
01.05.2008

[28] FRANZ, E., JERİCHOW, A., MOLLER, S., PFİTZMANN, A., STİERAND, I., Computer Based Steganography:How It Works And Why Therefore Any Restrictions on Cryptography Are Nonsense, At Best, Proc.Information Hiding Workshop, pp. 7–21, 1998

[29] GRUHL, D., BENDER, W., LU., A., Echo Hiding, Proc. Information Hiding Workshop, pp. 295–315, 1998

[30] T. A. WELCH., A technique for high-performance data compression, IEEE Computer 17, pp. 8-19, 1984

[31] Held, G. Data Compression, John Wiley & Sons, 1988

[32] Gerek, Ö., Fidan, M., Mycıelskı78 Sıkıştırma Algoritması
The Mycıelskı78 Compressıon Algorıthm, Elektrik ve Elektronik Mühendisliği Bölümü, Anadolu Üniversitesi, Eskişehir

[33] Borland Delphi 7.0 Zlib.Pas Dosyası

[34] http://marknelson.us/1997/01/01/zlib-engine/ , Dr. Dobb's Journal January, 1997 , 01.05.2008

[35] MATSUİ, K., TANAKA, K., NAKAMURA, Y. , Digital Signature on Facsmile Document by Recursive MH Coding, International Symposium on Cryptography and Information Security (CIS89), 1989

[36] TANAKA, K., NAKAMURA, Y., MATSUİ, K., Embedding a Secret Information into a Dithered Multi-level Image, Proceedings of IEEE Military Communcations Conference, pp 218–220, 1990

[37] LEVİLLAİN, P., Wireless LAN for Enterprises, Alcatel Telecommunications Review,Vol. 4, 287–291, 2002

[38] WAVE PCM soundfile format, Stanford Üniversitesi, http://ccrma.stanford.edu/CCRMA/Courses/422/projects/WaveFormat/, 18 Aralık 2007

[39] TOYRAN, M., Kuantum Kriptografi, Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, 2003

[40] SAJATOVİC, M., PRİNZ, J., KROEPJ, A., Increasıng The Safety Of The Atc Voıce Communıcatıons By Usıng In-Band Messagıng, FREQUENTS Nachrich tenfechnik GmbH, IEEE, Vienna, Austria, 2003

[41] http://lkd.belgeler.org/man/man1/man1-rsync.html, 22.05.2008

EKLER

EK-A. Sayısal Ses İçerisinde Sıkıştırılmış, Şifrelenmiş Gizli Verilerin/Dosyaların Kablosuz Transferi İçin Geliştirilen Yazılımın Program Kodları CD İçerisinde Sunulmuştur (EK-A.doc).

EK-B. Geliştirilen Uygulama Yazılımlarının Çalıştırılabilir Dosyaları CD İçerisinde Sunulmuştur (VoiceClient.exe, VoiceServer.exe).

EK-C. Tez pdf Dosyası CD İçerisinde Sunulmuştur (Tez.pdf).

ÖZGEÇMİŞ

1980 yılında İstanbulda doğdu. Üsküp İlköğretim Okulu'nu bitirdikten sonra 1999 yılında Bağcılar Abdurrahman ve Nermin Bilimli Teknik Lisesi'nden mezun oldu. Aynı yıl girmiş olduğu ÖSS sınavı sonucunda Kocaeli Üniversitesi Bilgisayar Öğretmenliği'nde okumaya hak kazandı. Buradan 2004 yılında mezun oldu ve Hereke Endüstri Meslek Lisesi'nde Bilgisayar Öğretmeni olarak göreve başladı. 2005 yılında Sakarya Üniversitesi'nde Yüksek Lisans öğrenimi görmeye başladı. 2007 temmuz ayından itibaren İstanbul'da mezun olduğu Üsküp İlköğretim Okulu'nda Bilgisyar Öğretmeni olarak görev yapmaktadır.