Gerçek Zamanlı Sayısal Ses İçerisinde Sıkıştırılmış ve Şifrelenmiş Veri Transferi

Tuncay Akbal[1] Yıldıray Yalman[2] A.Turan Özcerit[3]

[1,3] Elektronik ve Bilgisayar Eğitimi Bölümü, Sakarya Üniversitesi, Sakarya
[2] Elektronik ve Bilgisayar Eğitimi Bölümü, Kocaeli Üniversitesi, Kocaeli

[1] e-posta: tuncay_akbal@yahoo.com [2] e-posta: yildiray.yalman1@kocaeli.edu.tr
[3] aozcerit@sakarya.edu.tr

Özetçe

Günümüzde kablosuz iletişim sistemleri içerisinde giderek artan bir öneme sahip olan veri gizleme/sırörtme (steganografi), veri sıkıştırma (data compression) ve şifreleme (cryptology) teknikleri araştırmacıların yoğun ilgisini çekmektedir. Çoklu ortam ve bilgi güvenliği uygulamaları gibi güncel gereksinimler ile veri gizleme üzerine yapılan çalışmalar birleştirilerek daha güvenli iletişim uygulamaları geliştirilmesi amaçlanır. Bu noktadan hareketle sunulan bu çalışmanın temel amacı; kablosuz ortamda gerçek zamanlı sayısal ses içerisinde gizli veri transferi sağlarken, güvenli ve hızlı iletim için sıkıştırma, şifreleme yöntemlerinden faydalanmaktır. Bu amaca yönelik olarak kablosuz ortamda transfer edilen sayısal ses içerisine veri/dosya gömme yazılımı geliştirilmiştir. Sayısal ses verilerine gizli bilgileri gömmek için literatürde sunulan klasik LSB (Least Significant Bit) yaklaşımını kullanan bu yazılım, kullandığı sıkıştırma ve şifreleme teknikleri ile gizli haberleşmeyi daha güvenilir ve hızlı hale getirmektedir.

1. Giriş

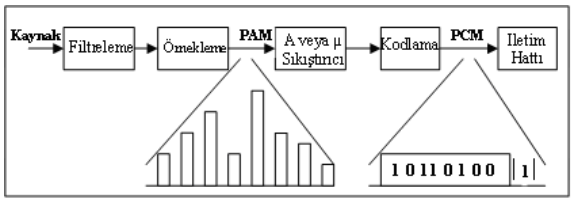
Gizli bilginin üçüncü kişilerin eline ulaşmaksızın hedef noktaya ulaştırılması istendiğinde, gizli haberleşme ihtiyacı ortaya çıkar ve bu ihtiyacın nasıl karşılanacağının cevabı büyük önem taşır. Şifreleme bu ihtiyacı karşılama bakımından akla gelen ilk yöntemlerden biridir. Ancak verinin güvenliğini tam olarak korumak için şifreleme her zaman yeterli olamamaktadır. Çünkü şifreleme teknikleri ile değişime uğratılmış mesajın/bilginin varlığından haberdar olan üçüncü kişiler bir takım ataklar ve çözme teknikleri ile başarıya ulaşacaklardır. Üçüncü kişilerin gizli bilgiden haberlerinin olmamasının sağlanması bilgi güvenliği için büyük önem arz eder. Bu durumun doğal sonucu olarak, sırörtme (steganografi) tekniklerine başvurulması gerekir. Sırörtme uygulamalarında bilgi sadece gönderen ve alanın varlığını bildiği şekilde saklanmakta, bazen de şifrelenip fazladan koruma sağlanmaktadır [1]. Bu uygulamalar özellikle band genişliği az ve veri güvenliği düşük olan kablosuz haberleşme ortamlarında, gizli veri gönderim zamanını arttırmaktadır. Bunun için gönderilecek olan verinin sıkıştırılması gönderme zamanı, işlemci yükü ve kullanılan mobil cihazların enerji giderleri açısından kazanç sağlamaktadır [2].

Sayısal veriler içerisine gizli bilgi yerleştirmek için birçok yöntem geliştirilmiştir. Bu yöntemler arasında sıkça kullanılan en düşük değerlikli bitlere (Least Significant Bits: LSBs) veri gömülmesinin birim zamanda gönderilecek veri miktarını olabildiğince yükseltmek ve taşıyıcı sinyalde minimum bozulmaya sebep olmak için daha uygun olduğu genel kabul gören bir yaklaşımdır [3, 4].

Bildiri bölümleri şöyle organize edilmiştir: Takip eden ilk üç bölümde analog ses sinyallerinin sayısal ses verilerine dönüştürülmesi, veri sıkıştırma ve şifreleme bilimleri sunulmaktadır. Beşinci bölümde sırörtme ve damgalama bilimi hakkında bilgi verilmektedir. Son iki bölümde ise geliştirilen sistemin uygulama örneklerine yer verilmekte ve elde edilen sonuçlara ilişkin değerlendirmeler sunulmaktadır.

2. Analog Ses Sinyallerinin Sayısal Ses Verilerine Dönüştürülmesi

Sayısal işaretlerin, gürültüden etkilenmemesi ve tümleşik devre teknolojisinin gelişmesi, sayısal verinin işlenmesinin (iletilme, sıkıştırma) nispeten daha ucuz olması sonucunu doğurmuştur. Bilginin iletimi, saklanması ve işlenmesi sırasında sayısal formatın analog formata tercih edilmesi yaygın bir durum haline gelmiştir. Ancak analog 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. Darbe Kod Modülasyonu (PCM: Pulse Code Modulation) yukarıda açıklanan probleme bir çözüm önerisi olarak 1970'li yıllarda ortaya çıkmış ve günümüzde bu amaç için en çok kullanılan sayısallaştırma tekniği olmuştur. PCM, analog işaretlerin belirlenmiş sayısal forma dönüştürülmesinde kullanılan bir tekniktir. Bu teknikte analog işaretten sayısal bilgiye ve sayısal bilgiden analog işarete dönüşüm sırasında oluşan örnekleme kayıpları oldukça küçüktür. Bu nedenle, örnekleme kayıplarından etkilenme oranı yüksek olan işaretlerin (konuşma işaretleri gibi) sayısal formda iletilmesi amacıyla, PCM günümüzde sıklıkla tercih edilmektedir (Şekil 1). Sayısal ortamda örneklenerek elde edilen ses verileri temel olarak ".wav" dosya tipindedir. Bu bildiri kapsamında geliştirilen yazılımlar "wav" dosya tipinde kodlanmış ses verileri üzerinde işlem yapmaktadır.



Şekil 1: Darbe kod modülasyonunun yapısı

Sayısallaştırma sürecinde meydana gelen kayıpların en aza indirgenmesi için analog ses sinyalinde yapılan örnekleme sayısının arttırılması çoğunlukla benimsenen bir yaklaşımdır. Örnekleme sayısına (8000, 11025, 16000, 22050, 32000 veya 44100) bağlı olarak ses iletişim kalitesi ve veri/dosya gömme kapasitesi açılarından toplam sistem başarımında önemli farklılıklar ortaya çıkmaktadır.

3. Veri Sıkıştırma (Data Compressing)

Diskler ve teypler gibi saklama birimleri üzerinde bulundurulan 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 mevcuttur. Bunlar karakter dağılımı, karakter tekrarı, çok kullanılan sözcükler ve konumsal artıklıktır [5]. Bu artıklıklar, gönderilecek verinin/dosyanın kayıpsız şekilde sıkıştırılarak farklı ortamlarda daha hızlı gönderilmesine olanak sağlarlar.

Verilerin hangi tür artıklık içerdiğinin önceden bilindiği durumlarda Statik Sıkıştırma Algoritması (SSA) kullanılır. Dosyanın türüne göre farklı yöntemler veya tablolar kullanılacaksa Dinamik Sıkıştırma Algoritması (DSA) kullanılması daha faydalı olabilmektedir. Boşluk sıkıştırma, Bit Dönüşümü Yöntemi (Bit Mapping), Tek Sembol Zinciri Kodlaması (Run-Length Encoding), Yarım Sekizli Paketleme (Half-Byte Packing) gibi teknikler SSA algoritmalarına, dinamik Huffman, LZ77 ve LZ78 gibi teknikler ise DSA algoritmalarına örnek verilebilir. LZ77 ve LZ78 algoritmaları harflerin birbirleriyle oluşturduğu kombinasyonlara ve bu kombinasyonların sinyal içindeki tekrarlarına bakarak sıkıştırma yaparken, Huffman algoritması ise dosya içerisinde sık rastlanan karakterlerin daha az bit ile, az rastlanan bitlerin ise daha fazla bit ile gösterilmesini sağlayarak çalışmaktadır. Takip eden alt bölümde LZ77 ve Huffman algoritmasının bir karışımı olan ve bu bildiride sunulmakta olan sistemin gizli verilerin sıkıştırılmasında kullandığı ZLib sıkıştırma algoritmasının detayları verilmektedir.

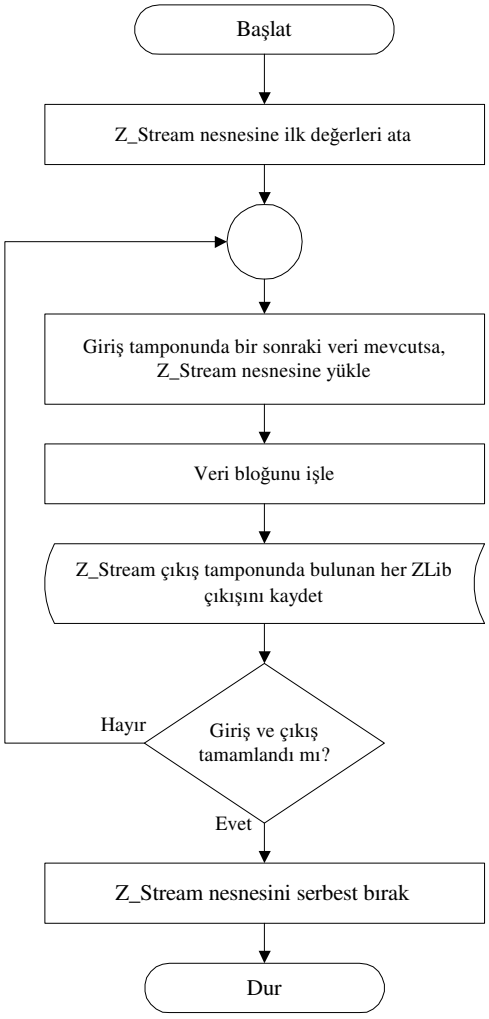
3.1. ZLib Sıkıştırma Algoritması

Mark Adler ve Jean-Loup Gailly tarafından geliştirilen ZLib, Deflate olarak bilinen bir algoritmayı biçimlendirmek için LZ77 ve Huffman kodlarını birleştirmektedir [6]. Şekil 2'de akış şeması verilen ZLib algoritması; bir Z_Stream nesnesi oluşturma, ZLib ile iletişim kurmak için Z_Stream nesnesini kullanarak giriş/çıkış yapma ve ardından Z_Stream nesnesini yok etme basamaklarından oluşur.

Sunulan çalışmada kullanılan üç farklı tipteki dosyanın, farklı sıkıştırma algoritmaları kullanılarak elde edilen nihai boyutları Tablo 1'de gösterilmektedir. Görüldüğü üzere ZLib sıkıştırma algoritması üç farklı tipteki dosya için en iyi sıkıştırma başarımını göstermektedir.

Tablo 1: Sıkıştırma algoritmalarının karşılaştırılması

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



Şekil 2: ZLib'in işlem basamaklarını içeren akış şeması

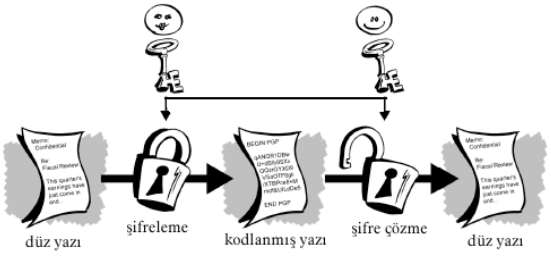
Geliştirilen yazılım, gerçek zamanlı sayısal ses verileri içerisine gömülecek olan gizli verileri ZLib algoritmasını kullanarak sıkıştırmaktadır. Bu sayede şifrelenip gönderilecek olan veri miktarı kayıpsız şekilde en küçük boyuta indirgenmektedir.

4. Şifreleme Bilimi

Şifreleme, haberleşen iki veya daha fazla tarafın bilgi alışverişini emniyetli olarak yapmasını sağlayan ve gizli bilgiyi istenmeyen kişilerin anlayamayacağı hale getirerek korumayı esas alan, temeli matematiksel yöntemlere dayanan tekniklerin ve uygulamaların bütünüdür [1]. Şifrelemede, mesaj ve anahtar bilgileri birleştirilerek bir kriptogram üretilir**.** Bir şifreleme sistemine güvenli denilebilmesi için anahtar olmadan kriptogramın çözülmesinin neredeyse imkânsız olması gerekir.

Şekil 3'de genel olarak şifreleme ve şifre çözme blok diyagramı gösterilmektedir. Mesaj ve şifreleme anahtarı bir şifreleme algoritması içerisinden geçirilerek şifreli mesaj elde edilir. Şifrelenmiş mesaj yetkisi olmayan kişiler tarafından da elde edilebilen ancak düşük maliyetli ve veri iletim hızı yüksek bir ortamdan geçirilerek alıcıya ulaştırılır. Alıcı, şifreli

mesaj ve çözme anahtarını kullanarak asıl mesajı elde eder. Yetkisi olmayan kişilerin şifrelenmiş mesajı elde etmeleri durumunda bile çözme anahtarı kendilerinde olmadıkça asıl mesajı elde etmeleri çok zordur.



Şekil 3: Genel şifreleme ve şifre çözme diyagramı

Şifreleme, veri güvenliği (confidentiality), veri bütünlüğü (data integrity) ve kimlik denetimi (authentication) olmak üzere üç görev üstlenir. İki merkez arasında gönderilen verinin üçüncü kişiler tarafından okunmasının engellemesi, veri güvenliği; iki merkez arasında gönderilen verinin üçüncü kişiler tarafından değiştirilmesinin engellemesi, veri bütünlüğü; alıcının, gönderilen verinin kimin tarafından gönderildiğinden emin olması ise kimlik denetimi olarak adlandırılır. Belirtilen bu üç görevin gerçekleştirilme yüzdesi şifreleme sisteminin güvenlik kalitesi için belirleyici olmaktadır.

Gizli Anahtarlı Şifreleme (Simetrik Şifreleme) sistemlerinde şifreleme ve şifre çözme için aynı anahtar, Açık Anahtarlı Şifreleme (Asimetrik Şifreleme) sistemlerinde ise şifreleme için açık anahtar, şifre çözmek için ise gizli anahtar kullanılır. Geliştirilen yazılım simetrik şifreleme tekniklerinden One Time Pad (OTP) algoritmasını kullanarak sıkıştırılmış gizli verileri şifrelemektedir. Aşağıdaki alt bölümlerde kullanılan bu algoritmaya ait detaylar verilmektedir.

4.1. One Time Pad (OTP) Algoritması

Sunulan çalışmada, gizli anahtarlı şifreleme yöntemlerinden biri olan ve interaktif televizyon yayınlarından telefona, e-posta işlemlerinden finansal işlemlere kadar birçok alanda kullanılan OTP (One Time Pad) algoritması kullanılmaktadır [7]. 1917 yılında Joseph Mauborgne ve Gilbert Vernam tarafından geliştirilen bu algoritmanın ilk örneği Vernam şifreleyicisidir. Bu şifreleyici oldukça basit bir yapıya sahiptir. Şifreleme işleminde öncelikle düz metin mesajını içeren bit dizgesi alınır ve şifrelenecek mesajın uzunluğunda rasgele bir anahtar dizisi seçilerek mesaj ve anahtara XOR işlemi uygulanır. Ancak anahtarın bir bölümü asla ikinci kez kullanılmamalıdır (aksi halde şifreleyici kırılabilir).
Örneğin mesaj ve anahtar,

Mesaj = “güvenli bir şifreleme sistemi” (1)

Anahtar = C143AD/*67GJ&4#DS4341F4/^+%PK (2)

kabul edilsin. Bu durumda mesajın ve anahtarın ASCII karşılığı (kelime aralarındaki boşluklar ile birlikte) sırası ile Tablo 2 ve Tablo 3’ de görüldüğü gibi olacaktır.

Mesaj, anahtar ile bire bir XOR işlemine tabi tutulduğunda Tablo 4’te ve (3)’de görülen şifreli mesaj (Smsj) elde edilir.

Smsj = $═BV/(F◙T^5jÏ]`66XVYTfGF-_@=“ (3)

Tablo 2: Mesajın ASCII karşılığı

g	ü	v	e	n	l	i	
103	252	118	101	110	108	105	32

b	i	r	
98	105	114	32

ş	i	f	r	e	l	e	m	e	
254	105	102	114	101	108	101	109	101	32

s	i	s	t	e	m	i
115	105	115	116	101	109	105

Tablo 3: Anahtarın ASCII karşılığı

C	1	4	3	A	D	/	*	6	7
67	49	52	51	65	68	47	42	54	55

G	J	&	4	#	D	S	4	3	4
71	74	38	52	35	68	83	52	51	52

1	F	4	/	^	+	%	P	K
49	70	52	47	94	43	37	80	75

Alıcı, şifreli mesaj kendisine ulaştıktan sonra anahtar ile tekrar XOR işlemi gerçekleştirerek asıl mesajı elde eder. Şifreli mesajın göndericiden alıcıya giderken iletim kanalında yetkisiz kişilerin eline geçmesi durumunda bile, anahtar bilinmediğinden dolayı bu kişiler mesajın uzunluğu kadar farklı karakterlerden oluşacak anahtarı tahmin etmeli ve bunu şifreli mesaja uygulamalıdırlar. ASCII tablosunda 256 farklı karakter yer almaktadır ve yukarıda kullanılan mesajın uzunluğu 29 olduğundan, olası anahtar sayısı şifreyi çözmek isteyenlerin karşısına 256^{29} olarak çıkacaktır. Bu da olası anahtar sayısının 6,90174634679056378743475586277e+69 olacağı anlamına gelmektedir. Ayrıca üçüncü kişiler bu tahminleri yaparken şifreli mesajla yanlış anahtarları XOR işlemine tabi tuttuklarında birden fazla anlamlı mesaj elde edeceklerdir ve bu da doğru mesajın hangisi olduğu konusu da tereddüt yaşanmasına sebep olacaktır.

Tablo 4: Şifrelenmiş mesaj

Şifreli Mesaj ASCII Kodu	36	205	66	86	47	40
Şifreli Mesaj Karakter Kodu	$	═	B	V	/	(

Şifreli Mesaj ASCII Kodu	70	10	84	94	53	106
Şifreli Mesaj Karakter Kodu	F	◙	T	^	5	j

Şifreli Mesaj ASCII Kodu	216	93	69	54	54	88
Şifreli Mesaj Karakter Kodu	Ï	]	`	6	6	X

Şifreli Mesaj ASCII Kodu	86	89	84	102	71	70
Şifreli Mesaj Karakter Kodu	V	Y	T	f	G	F

Şifreli Mesaj ASCII Kodu	45	95	64	61	34
Şifreli Mesaj Karakter Kodu	-	_	@	=	“

4.2. OTP Algoritmasının Avantajları

OTP algoritmasında öncelikle uzunluğu n bit olan mesaj için n bitlik bir anahtar dizisi seçilir ve sonrasında mesaj şifrelenip gönderilir. Bu yöntemde şifreli mesajı ele geçiren üçüncü bir kişi, olası bütün anahtarları (256^n tane) denese bile gizli mesajı tam anlamıyla bulamamaktadır. Çünkü bu işlemin sonunda n bitlik bütün kelimeler elde edilmektedir. Üçüncü kişilerin elinde birden fazla anlamlı mesaj olacağı için, bu mesajların içinden gerçek mesajı tahmin etmek oldukça zordur. Bu açıdan OTP, koşulsuz güvenli bir sistem olarak kabul edilebilir.

Günümüz bilgisayarlarıyla yapılması çok zor, hatta imkânsız kabul edilen çok büyük tamsayıların asal çarpanlarına ayrılarak ayrık logaritma alınması işlemlerini kuantum bilgisayarların kolaylıkla ve verimli olarak yapabilecekleri öngörülmektedir. Dolayısıyla, kuantum bilgisayarlar yapıldığında günümüzün güvenli kabul edilen açık anahtarlı şifreleme sistemlerinin güvenlikleri tehlikeye girecektir [8]. OTP gizli anahtarlı ve güvenilir bir çözüm sunma özelliğiyle açık anahtarlı şifreleme sistemlerine üstünlük sağlamaktadır.

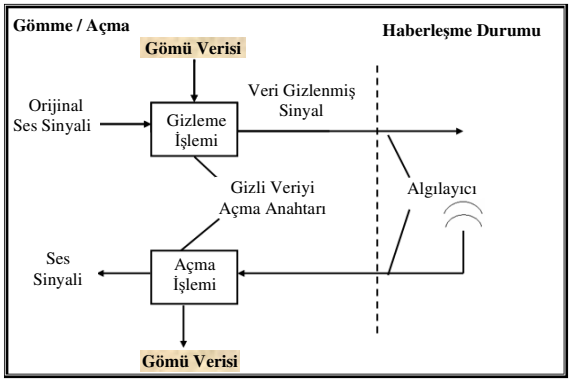
Bu bildiride sunulan uygulama, OTP şifreleme algoritması için kullanılan şifreleme ve şifre çözme anahtarını program içerisinde sadece Ses Gönderici/Ses Alıcı (Client / Server) modüllerinin bileceği ve iletişimi kuran kullanıcıların dahi o an için bilemeyecekleri eş zamanlı olarak kullanılan anahtarlar ile gerçekleştirmektedir. Başlangıçta iki tarafında bildiği anahtar programın çalışması esnasında sürekli değişime uğratılmaktadır. Böylece güvenli bir yoldan anahtar değiştirilmekte ve farklı anahtarlar kullanılmaktadır. Değişime uğratılmış olan anahtar hakkında sunucu ve istemci yazılımlar birbirleri ile eş zamanlı olarak haberleşmektedir.

5. Sırörtme ve Damgalama

Steganografi (Sırörtme) iki parçadan oluşan Yunanca bir kelimedir. "Steganos" örtülü/gizli, "grafi"de yazım/çizim anlamına gelmektedir. Örtülü yazma sanatı olarak dilimize çevrilebilen "stego" aslında antik Yunan ve Herodot zamanına kadar uzanan derin bir geçmişe sahiptir. Sırörtme tekniklerinin ticari kullanımı yavaş yavaş sayısal "filigran"ın (watermarking) gelişmesini sağlamıştır. Burada söz konusu olan gizli bilginin insan duyularından gizlenmesidir. 1990'ların başında imge filigrasyonu (damgalama) kavramı gelişmiş; faks gibi ikili imgelerin korunması kavramı ortaya atılmıştır [9, 10]. 1993 yılında yapılan bir uygulamaya ise daha sonra "watermark" olarak birleştirilecek "water mark" ismini verilmiştir [11].

Modern sırörtme teknik olarak, bir veriyi (mesaj) bir nesnenin içine gizli biçimde yerleştirmeyi esas alır. Şifreleme uygulamalarında bilgi sadece gönderen ve alanın anlayabileceği şekilde şifrelenirken, sırörtme uygulamalarında bilgi sadece gönderen ve alanın varlığını bildiği şekilde saklanır. Bu bilimde taşıyıcılara örtü verisi, iletilmek istenen gizli veriye gömü verisi, içerisine gizli veri gömülmüş olan nesneye/dosyaya ise örtülü veri denir. Gömü verileri genelde metin ve resim; örtü verisi ise metin, resim ve video görüntüleridir. Sayısal ses verileri üzerinde sırörtme, teorik ve pratik olarak mümkündür. Çünkü sese küçük yankılar veya kulağın algılayamadığı sinyaller eklenebilir ve daha yüksek genlikte bir ses bileşeni tarafından maskelenebilir [12, 13].

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 üzerine kurulduğu mantık ise gereksiz bilgiler taşıyan nesnelerin içindeki bilgilerin, başka bilgi parçacıklarıyla yer değiştirmesidir. Şekil 4'de sayısal ses içerisine sırörtme uygulamasının şematik yapısı görülmektedir. Sıkıştırılmış ve ardından şifrelenmiş gömü verisi bir fonksiyon yardımıyla orijinal sayısal ses sinyali içerisine gömülür. Veri gizlenmiş olan ses sinyali (örtülü veri) yetkisi olmayan kişilerin de dinleyebilecekleri bir ortamda (kablosuz) alıcıya ulaştırılır. Alıcı fonksiyonlar ses sinyali içerisinde tespit ettikleri gömü verisini bir çözücü algoritma yardımıyla geri elde ederler. Bu noktada süreç içerisinde dinlenen ses sinyali orijinal değildir. Ancak kulağın algılamakta zorlanacağı bir bozulmaya uğramıştır. Bu yüzden verinin gömüleceği sinyaldeki bozulmanın en aza indirgenmesi büyük önem taşımaktadır.



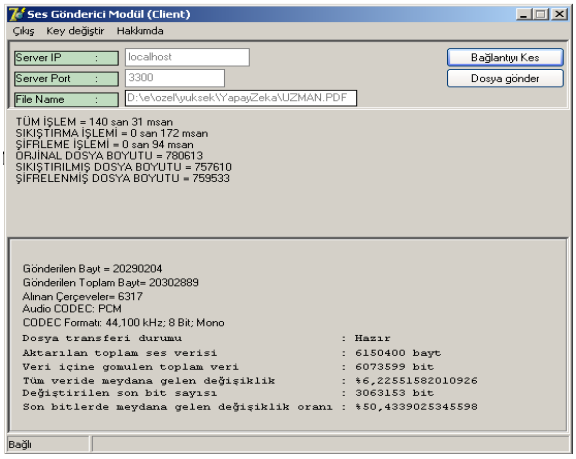
Şekil 4: Ses sinyali içerisine sırörtme uygulamasının gösterimi

Mesajların örtü verisi içerisine yerleştirilme şekli çok büyük önem taşır. Gömü verisinin/dosyasının hangi bitlere yerleştirildiği, hangi veri bloklarının içerisine konumlandığı, şifreleme ve sıkıştırma yapılıp yapılmadığı gibi parametreler veriyi elde etmenin (sıraçma, steganaliz) ilgi alanına girer. Sırörtme uygulamasında üçüncü kişilerin gömü verisini elde etme (sıraçma) işlemini yapamaması için, verinin gömülme şeklinin gizli tutulması, gömü verilerinin güvenliği açısından büyük önem arz eder.

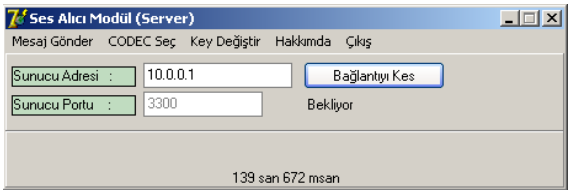
6. Geliştirilen Sırörtme Sisteminin Kullanıcı Arayüzleri ve Uygulama Örnekleri

Geliştirilen sırörtme sistemi Borland Delphi 7.0'da geliştirilmiş olup, temel olarak Network Multimedya (NMM) bileşeninden faydalanılmaktadır. Uygulama, Ses Gönderici Modülü (SGM) ve Ses Alıcı Modülü (SAM) olmak üzere, sırasıyla gizli veriyi/dosyayı sayısal ses örtü paketlerine geliştirilen algoritma ile gömen ve alıcıda bunları ayrıştırarak elde eden iki ana kısımdan oluşmakta ve ilgili modüller TCP/IP iletişim protokolünü kullanmaktadır [14]. Şekiller 5 ve 6'da, geliştirilen sırörtme uygulamasının SGM ve SAM kullanıcı arayüzleri verilmektedir.

Geliştirilen arayüzlerden SGM iletişim ile ilgili tüm istatistik bilgilerini (işlem zamanı, gömü verisinin transfer durumu, kullanılan ses formatının özellikleri, sıkıştırma ve şifreleme başarımı, örtü verisinde meydana gelen değişim vb.) kullanıcıya vermektedir.

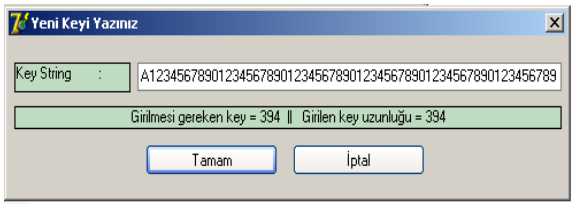


Şekil 5: Geliştirilen SGM modülünün kullanıcı arayüzü



Şekil 6: Geliştirilen SAM modülünün kullanıcı arayüzü

Gizli haberleşme yapan kullanıcılar şifreleme işlemi için kullandıkları anahtarı her iki modülde de yer alan "Key Değiştir" menüsü yardımıyla yeniden düzenleyebilmektedir (Şekil 7). Buna ek olarak menü çubuğunda "Codec Seç" seçeneği bulunmaktadır. Bu menü yardımı ile Codec seçimi yapılarak, analog ses bilgilerinin sayısal ses bilgilerine dönüştürülmesindeki örnekleme sayısı (8000, 11025, 16000, 22050, 32000 veya 44100) ve her bir örneğin kaç bit ile temsil edileceği (8 veya 16 bit) belirlenebilmektedir.



Şekil 7: Şifrelemede kullanılan anahtarın tanımlandığı ekran

Gizli verinin/dosyanın gönderilmeye başlanma anı SGM'yi kullanan kullanıcıya bağlıdır. SGM kullanıcısı gizli haberleşmeyi başlatmak istediğinde, iletişimde SAM'ye gönderilen ilk sayısal ses paketinin LSB değerleri içerisine 16 bitlik başlat örüntüsü ve gizli veriye/dosyaya ait diğer bilgiler (dosya adı, dosya boyutu vb.) gömülür. Gizli haberleşme yapıldığı sürece gömü verileri öncelikle ZLib algoritması ile sıkıştırılmakta sonrasında ise OTP algoritması ile şifrelenerek gönderilmektedir. Bu sayede gönderilmesi gereken veri miktarı azaltılmakta ve güvenlik en üst seviyeye çıkarılmaktadır.

Çalışmanın kullanımı öngörülen uygulamalar, değişik kullanıcıların erişimine açık, iletişim ortamını paylaşan ve çoğunlukla hareketli düğümlerden oluşmaktadır. Geliştirilen sırörtme uygulaması, internet üzerinden gerçek IP numaraları yardımıyla çalıştırılabildiği gibi, bir eşe-eş ağda da kolaylıkla kullanılabilmektedir. Uygulama örneklerinde, Tablo 5'de belirtilen değişik özelliklere sahip iki bilgisayar kullanılmıştır. Tablo 6'de ise gömü verisi/dosyası olarak kullanılan örnek dosyalara ait özellikler verilmektedir. İlgili dosyalar elde edilmesi kolay ve sıkıştırma performansı açısından farklılıklar gösterdikleri için seçilmiştir. İlerleyen paragraflarda, belirtilen bilgisayarlar ve değişik örnekleme sayıları için yapılan uygulamalar sonucunda gizli veri gömme/alma süreleri açısından değerlendirmeler sunulmaktadır.

Tablo 5: Uygulamalarda kullanılan bilgisayarların donanım özellikleri

PC ADI	İŞLEMCİ TİPİ	HIZ (GHZ)	BELLEK BOYUTU(MB)
PC_A	Intel(R) Pentium(R)4 CPU	3,2	384
PC_B	Intel(R) Celeron Mobile(R) CPU	1,6	224

Özellikle gerçek zamanlı kablosuz haberleşme uygulamalarında kullanılan donanımların birim zamanda işlem yapabilme kapasitelerinin yüksekliği, uygulamaların sağlıklı şekilde gerçekleştirilmesi açısından hayati önem taşımaktadır. Bu çalışmaya 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 odaklanıldığında bilgisayarların hız ve kapasitelerinin haberleşmenin başarımına ne kadar etki ettiği de ortaya çıkarılmaktadır.

Tablo 6: Gizli gömü dosyaları ve özellikleri

No	Gömü Dosyası Adı	Dosya Uzantısı	Dosya Boyutu (Byte)
1	demo	.mp3	38912
2	sndrec32	.exe	124928
3	svega	.wav	1823640

Dosyalara, sırörtme yöntemi kullanılarak gönderilmeden önce sırasıyla sıkıştırma ve şifreleme işlemleri uygulanır. Uygulamada kullanılan dosyaların sıkıştırma ve şifreleme işlemlerinden sonraki boyutları Tablo 7'de gösterilmektedir. Yapısı itibariyle fazla artıklık içermeyen ".mp3", ".exe" gibi dosyalarda sıkıştırma başarımı doğal olarak düşük olacaktır.

Tablo 7: Kullanılan dosyaların sıkıştırılmış ve şifrelenmiş boyutları

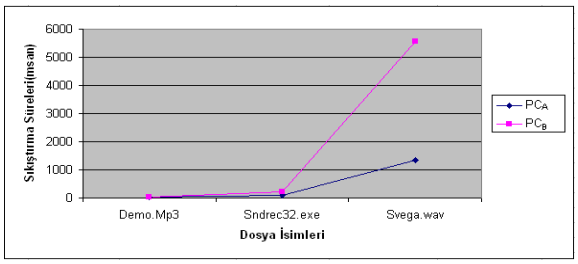
Dosya İsmi	Dosya Boyutu (Byte)	Sıkıştırılmış Dosya Boyutu (Byte)	Şifrelenmiş Dosya Boyutu (Byte)
Demo.mp3	38912	36931	37025
Sndrec32.exe	124928	60582	60736
Svega.wav	1823640	930111	932472

Özellikle kablosuz haberleşme sistemlerinde veri transferi için harcanan zaman önemlidir. Bu zaman azaldıkça bilgisayarların işlem yükü azalmakta, başarımları ise artmaktadır. Bu durumun kullanılan batarya süresine de etkisi olumlu yönde olacaktır. Şifrelemeden sonra dosya içerisine

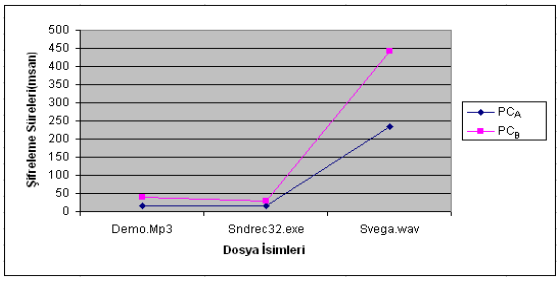
anahtar ile ilgili bilgiler gömüldüğü için (geliştirilen yazılımda her 394 byte için 1 byte) dosya boyutu küçük bir oranda artacaktır (1/394 oranında). Ancak bu dosya güvenliği için göze alınabilir bir fazlalıktır. Dosya boyutu arttıkça ya da içerisinde artıklık fazla olan dosya tipleri (wav, doc gibi uzantılı dosyalar) kullanıldıkça sıkıştırmanın veri iletimine faydası daha fazla olacaktır.

Geliştirilen modüllerde şifreleme için başlangıçta her iki tarafta da aynı anahtar mevcuttur. Anahtar içerisinde her türden karakter kullanılabilmekte ve veri güvenliği için gönderilen her paket farklı bir anahtar ile şifrelenmektedir. Bu değişim için her iki bilgisayarda da anahtar değişim fonksiyonu bulunmakta, SGM modülündeki fonksiyon kullanılarak anahtar değişim parametresi üretilmektedir. Bu parametre gönderilecek olan paketin içine gömüldükten sonra, SAM modülü bu parametre ve anahtar değişim fonksiyonu yardımıyla mevcut anahtarı tespit etmektedir. Son olarak elde edilen anahtar kullanılarak gelen paketteki şifreli veri çözülmektedir. Ancak ilk anahtarın rastsal sayı üreteçleri (Secure PseudoRandom Number Generator, Secure—PRNG) kullanılarak elde edilmesinin, daha güvenli bir iletişim imkânı sağlayacağı öngörülmektedir.

Şekil 8'de, uygulama bilgisayarlarının dosyaları sıkıştırma süreleri, Şekil 9'da şifreleme süreleri ve Şekil 10'da ise toplam dosya gönderim süreleri grafiksel olarak gösterilmektedir.



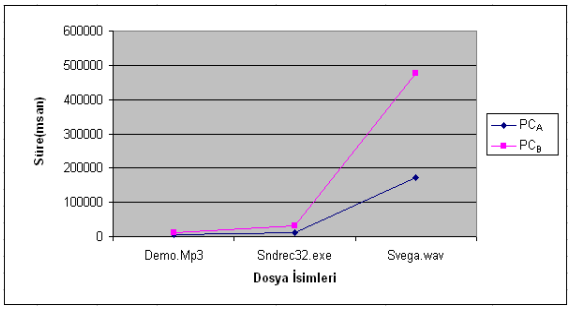
Şekil 8: PC_A ve PC_B'nin milisaniye türünden dosya sıkıştırma başarım grafikleri



Şekil 9: PC_A ve PC_B'nin milisaniye türünden dosya şifreleme başarım grafikleri

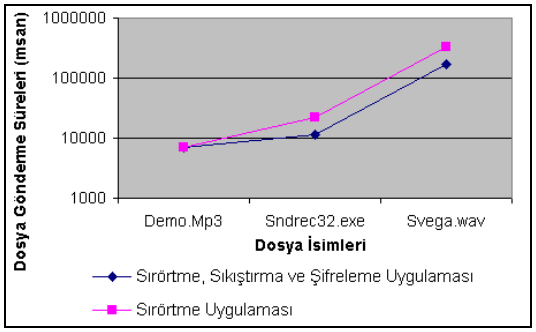
Daha hızlı olan PC_A'nın sıkıştırma ve şifreleme başarımı açısından PC_B'den daha iyi olduğu görülmektedir. Daha hızlı olan PC_A ile PC_B arasında sıkıştırma ve şifreleme süreleri bakımından küçük dosyalar açısından fark az olmasına karşın dosya boyutu daha büyük olan "Svega.wav" dosyasında PC_A, şifreleme işleminde yaklaşık 2 kat, sıkıştırma işleminde ise 5 kat daha iyi başarım göstermektedir.

Dosya gönderme açısından bakıldığında ise yaklaşık 3 kat daha iyi bir başarım söz konusudur. Bu durum dosya boyutu büyüdükçe sıkıştırma ve şifrelemede kullanılan algoritmaların gereksinim duyduğu bellek ve işlem hızı ihtiyacının artmasından ve dolayısı ile bilgisayar donanım özelliklerine olan bağlılığın nispeten yükselmesinden kaynaklanmaktadır.



Şekil 10: PC_A ve PC_B'nin milisaniye türünden dosya gönderme başarım grafikleri

Şekil 11'de, bu bildiride sunulan çalışma ile [3]'te sunulan sayısal ses paketlerine gömü verisini sıkıştırma ya da şifreleme işlemi yapmadan gömen klasik bir sırörtme yazılımının dosya gönderme süreleri açısından karşılaştırması yapılmaktadır.



Şekil 11: Yapılan sırörtme, sıkıştırma ve şifreleme uygulaması ile klasik sırörtme uygulaması başarım karşılaştırması

Dosya boyutu küçük olan Demo.mp3 dosyasında, yazılımların başarım süresi birbirine yakın çıkmıştır. Ancak sunulan yazılım, klasik sırörtme uygulamasından daha güvenli bir iletişim ortamı sunmaktadır. Sıkıştırma işlemi, şifreleme için kullanılan zamanı telafi etmektedir. Sıkıştırmanın sağlamış olduğu başarım dosya gönderme süresini de düşürmektedir. Daha büyük boyutlu dosyalar için (Sndrec32.exe ve Svega.wav) karşılaştırma yapıldığında ise geliştirilen yazılımın klasik sırörtme uygulamasına olan üstünlüğü daha belirgin hale gelmektedir.

7. Sonuçlar ve Değerlendirmeler

Sunulan çalışmada, gömü dosyalarına öncelikle sıkıştırma ardından şifreleme yapılmaktadır. Elde edilen veri, gerçek zamanlı sayısal ses paketleri içerisine sırörtme yapılarak gömülmekte ve kablosuz ortamda iletilmektedir. SGM ile eşzamanlı iletişim halinde olan SAM elde ettiği sayısal ses paketlerini çalmakta, diğer yandan bu ses paketleri içerisinde bulunan gömü verilerini süzerek kaydetmektedir Sunulan çalışmada geliştirilen yazılımın farklı donanım özelliklerine sahip bilgisayarlardaki uygulama örneklerinden elde edilen başarımları incelenmektedir. Yapılan uygulama örneklerinden elde edilen temel bulgular şunlardır:

- 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 klasik sırörtme uygulamasına göre daha fazla başarım elde edilmiştir. Ayrıca gönderilecek bit miktarı düştüğünden klasik sırörtme uygulamasına kıyasla sesteki toplam bozulma miktarı da düşmüştür.
- Sırörtmede gizli bilgiler yalnızca kaynak ve alıcı algoritmanın bildiği şekilde gömülmektedir. Bu gömme şeklinin üçüncü kişilerce bilinme ihtimali de dikkate alınarak gömülecek verilere şifreleme uygulanmıştır. Kullanılan şifreleme algoritmasında gerçek zamanlı olarak alıcı ve gönderici modüllerdeki anahtar değişimleri verinin güvenliğini arttırmaktadır.
- Dosya gönderimi yapılırken ses verilerini dinleyen üçüncü kişilerin gizli veriyi elde edebilmek için şu aşamalardan geçmeleri gerekmektedir:
    1. Yapılan sırörtme uygulamasının hangi teknik ile gerçekleştirildiğinin bilinmesi gerekir (LSB, son iki bite veri gömme vb.).
    2. Kullanılan dosya gönderim biçimini (gizli haberleşmeyi başlat bilgisinin, gizli dosya adının, gizli dosya boyutunun hangi paketlere gömüldüğü gibi) bilmeleri gerekir.
    3. Dinlenen ses verilerinden elde ettikleri dosya şifreli olacağından, hangi şifreleme yöntemi (OTP), başlangıç anahtarı ve şifre çözümü için anahtar bilgisinin her veri paketi için nasıl değiştirildiğini bilmeleri gerekir.
    4. Şifre çözme işlemini başarıyla gerçekleştirdiklerinde eldeki dosya sıkıştırılmış dosya olacağından sıkıştırma algoritmasını ve bu algoritmanın kullandığı fonksiyonları (ZLib kütüphanesi) tahmin etmeleri gerekir.
    5. Algoritma, kod ve anahtara sahip üçüncü kişinin, iletişimi gerçekleştiren diğer iki kişinin haberleşmesi ile senkronize olması gereklidir. Gerçek zamanlı sayısal ses paketlerinden herhangi birinin (1 MByte sayısal ses bilgisi 2654 paket ile gönderilmektedir) dinlenememesi halinde dosyanın tamamını çözülmesi mümkün olmayacaktır.
- Bugün gerek bilgisayar ağlarında ve gerekse internet ortamında çok sayıda ses verisi mevcut olup, bunlardan hangisinin veri gizlediğini tahmin etmek oldukça zor hatta imkânsızdır. Bu da sırörtmenin şifreleme bilimine karşı üstünlüğü olarak görülebilir. Ancak sırörtme içerisinde, şifreleme için harcanan çok kısa bir süre (dosya boyutuna göre milisaniyeler mertebesinde olabilmektedir) gömü verisinin güvenliğini oldukça arttırmaktadır. Kullanılan tekniklerde kod çözme esnasında orijinal ses verisine ihtiyaç duyulmaması da geliştirilen yazılımın bir diğer önemli özelliğini oluşturmaktadır.
- Bilgisayar donanım özellikleri (işlemci türü, işlemci hızı, RAM bellek) iyileştikçe yapılan uygulamanın daha sorunsuz çalıştığı tespit edilmiştir. Daha gelişmiş bilgisayarlar ile veri gömme ve gömülü veriyi/dosyayı elde etme süreleri daha da kısalacak ve böylece akıcı (streaming) görüntü/video uygulamalarının da veri gizleme uygulamalarına iyi birer alternatif olacakları öngörülmektedir.
- Sayısal ses bilgileri içerisine gömü verisi yerleştirilirken en düşük değerlikli bitler kullanılmıştır. Birim zamanda gömülen veri/dosya boyutunun arttırılması amacıyla her ses örneği 16 bit ile nitelendirilip gömülecek bit sayısı 2'ye hatta her sekizlinin (16 bitlik örnek için örnek başına 2 sekizli mevcuttur) son iki bitine veri gömülerek 4'e çıkarılabileceği öngörülmektedir. Ancak bu uygulamanın, birim zamanda alınan sayısal ses bilgisi 44100'e çıkarıldığında ve çok yüksek işlem gücü olan bilgisayarlarda gerçekleştirildiğinde olumlu sonuçlar vereceği aşikârdır.

8. Kaynakça

[1] Yalman, Y., Ertürk, İ., "Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi", *Politeknik Dergisi,* 11(4): 319–327, 2008.

[2] Barr, K., Asanovic, K., "Energy Aware Lossless Data Compression", *The First International Conference on Mobile Systems, Applications and Services*, San Francisco, CA, Mayıs 2003.

[3] Yalman, Y., "Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi", Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, 2007.

[4] Chang, L., Moskowitz, I., "Critical Analysis of Security in Voice Hiding Techniques", *Information Technology Division Center for High Assurance Computer Systems*, Naval Research Laboratory, Washington DC, USA.

[5] Welch, T. A., "A Technique for High-Performance Data Compression", *IEEE Computer Science*, p. 8–19, 1984.

[6] Barr, K., Asanovic, K., "Energy Aware Lossless Data Compression", *The First International Conference on Mobile Systems, Applications, and Services*, San Francisco, May 2003.

[7] Pohlmann, N., Reimer, H., Schneider, W., "OTP and Challenge/Response Algorithms for Financial and E–Government Identity Assurance", *Securing Electronic Business Processes*, p. 281–290, 2008.

[8] Toyran, M., "Kuantum Kriptografi", Yüksek Lisans Tezi, *İstanbul Teknik Üniversitesi F.B.E*, 2003.

[9] Matsui, K., Tanaka, K., Nakamura, Y., "Digital Signature on Facsimile Document by Recursive MH Coding", *International Symposium on Cryptography and Information Security (CIS89)*, 1989.

[10] Tanaka, K., Nakamura, Y., Matsui, K., "Embedding a Secret Information into a Dithered Multi-level Image", *Proceedings of IEEE Military Communications Conference*, p. 218–220, 1990.

[11] Hartung, F., Kutter, M., "Multimedia Watermarking Techniques", Proceedings of the IEEE, 87(7):1079–1107, 1999.

[12] Franz, E., Jerichow, A., Moller, S., Pfitzmann, A., Stierand, I., "Computer Based Steganography: How It Works and Why Therefore Any Restrictions on Cryptography Are Nonsense", *Proc. Information Hiding Workshop*, p. 7–21, 1998.

[13] Gruhl, D., Bender, W., Lu., A., "Echo Hiding", *Proc. Information Hiding Workshop*, p. 295–315, 1998.

[14] Akbal, T., "Ses Verilerine Sıkıştırılmış ve Şifrelenmiş Ham Verilerin Gömülmesi", Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, 2008.