T.C.

FIRAT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KRİPTOGRAFİK PROGRAMLAMADA SES STEGANOGRAFİSİ

YÜKSEK LİSANS TEZİ

Mehmet VURAL

152134103

MEKATRONİK MÜHENDİSLİĞİ

ANABİLİM DALI

DANIŞMAN:

Dr. Öğr. Üyesi Muharrem Tuncay GENÇOĞLU

AĞUSTOS 2019

T.C.

FIRAT ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

KRİPTOGRAFİK PROGRAMLAMADA

SES STEGANOGRAFİSİ

YÜKSEK LİSANS TEZİ

MEKATRONİK MÜHENDİSLİĞİ

ANABİLİM DALI

Bu tez, / / tarihinde aşağıda belirtilen jüri üyeleri tarafından oybirliği ile başarılı olarak değerlendirilmiştir.

Danışman : Dr. Öğr. Üyesi Muharrem Tuncay GENÇOĞLU

Üye : Prof. Dr. Bilal ALATAŞ

Üye : Doç. Dr. Muhammed Fatih TALU

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ...../...../..... tarih ve .....sayılı kararı ile onaylanmıştır.

ÖNSÖZ

Bu Tez çalışması, Fırat Üniversitesi Fen Bilimleri Enstitüsü Mekatronik Mühendisliği Anabilim Dalı Yüksek Lisans Programında hazırlanmıştır. Tezin araştırılmasında, yürütülmesinde ve oluşumunda engin bilgi ve tecrübelerinden yararlandığım ve bilgilendirmeleriyle çalışmamı şekillendiren saygıdeğer hocam ve tez danışmanım Dr. Öğr. Üyesi Muharrem Tuncay GENÇOĞLU'na sonsuz teşekkürlerimi sunarım.

Ayrıca bugünlere gelmemde en büyük pay sahibi aileme, her zaman manevi desteklerini hissettiğim sevgili eşim Sefanur VURAL'a tüm kalbimle teşekkür ederim.

Mehmet VURAL

İÇİNDEKİLER

ÖZET

Özellikle son yıllarda bilgi ve bilgisayar sistemlerinin güvenliği oldukça önemli bir konu olarak karşımıza çıkmaktadır. Son dönemlerde internetin yaygınlaşmasıyla veri alışverişi ve paylaşımı da artmıştır. Dünyanın birçok yerindeki insanlar tarafından metin, resim, video, ses gibi birçok veriyi içeren dosyalar, etkin bir şekilde paylaşılabilmektedir. Fakat bu iletişim ağı hayatı kolaylaştırdığı gibi çok ciddi güvenlik sorunlarını da beraberinde getirmiştir. İnternet gibi herkese açık olan bir ağdan gizli tutulan veriler paylaşıldığı veya gönderildiği zaman üçüncü şahıslar tarafından bu verilere kolaylıkla erişilebilir. Bu ciddi tehlikeler karşısında verilerin gizlenmesi ve değiştirilmesini engellemek için gelişen teknoloji ile birlikte yeni uygulamalar ve yeni sistem koruma mekanizmaları ortaya çıkmıştır. Kullanılan bu uygulamalarda kriptografi ve steganografi olarak birçok şifreleme algoritması oluşturulmuş ve verilerin güvenliğini daha da artırmak için var olan algoritmalar geliştirilmeye devam edildiği gibi hâlâ yeni algoritmalar da oluşturulmaktadır. Günümüz teknolojilerindeki gelişmelerle beraber, verilerin güvenliği gün geçtikçe daha zorlaşmaktadır. Güvenlik, herhangi bir haberleşme sisteminde en temel faktör olarak kabul edilmektedir. İletişim kanallarına yapılan saldırıların artması, gizli tutulması gereken verilere dışarıdan ulaşılması ve bazı sistemler yardımıyla bilginin kaynağından başka bir yere aktarılması veya bilginin değiştirilmesi, bilginin sahipleri için çok ciddi sorunlar oluşturmaktadır. Bilginin gizli kalmaması bireyleri, toplumları ve devletleri çok kötü şekillerde etkileyeceği için bu konu önemli ve tehlikeli bir tehdit oluşturmaktadır. Siber saldırıların yöntem ve tekniklerinin artması, sistemlerdeki mevcut verilere ulaşılmasını da kolaylaştırmıştır. Veri gizliliği ve bütünlüğünün önemi artmıştır. Verilere izinsiz kişiler tarafından ulaşıldığı durumlarda, verilerin çalınması, değiştirilmesi ve sisteme zarar verilmesi gibi olumsuz sonuçlar ortaya çıkmaktadır.

Bu tezde algoritmayla önce veriler şifrelenmiş daha sonra şifrelenen veriler ses dosyası içerisine gizlenmiştir. Verilerin bir yerden başka bir yere taşınması durumunda, veri gizlemesi yapıldığından, iletilen veri üçüncü şahıslar tarafından görülemeyecektir. Ancak çeşitli saldırılar ve analizler sonucunda gizlenen veriye izinsiz kişiler ulaşmak isteseler bile ulaşamayacaklardır. Çünkü gizlenen veri ilk olarak bir algoritmayla şifrelenmiş daha sonra ise ses içerisine gizleme yapılmıştır. Ses içerisindeki gizli veriye ulaşılsa bile şifreleme yapıldığı için anlamsız bir veri elde edilir. Bu çalışmada Taylor serisinin açılımı kullanılarak, algoritma dil analizi ve açık metin saldırılarına karşı güçlü hale getirilmiştir. Kriptografi ve Steganografi birleştirilerek elde edilen bu hibrit model ile kriptolojiye farklı bir bakış açısı ortaya konulmuştur. Önerilen algoritma ile yapılan uygulamalarda kriptoloji dolayısıyla siber güvenlik alanında matematiğin hayati önem taşıdığı bir kez daha ispatlanmıştır.

Anahtar Kelimeler: Ses Steganografi, Kriptografi, Veri Steganografi, Video Steganografi, Metin Steganografi

SUMMARY

SOUND STEGANOGRAPHY IN CRYPTOGRAPHIC PROGRAMMING

Especially in recent years, the security of information and computer systems is a very important issue. With the widespread use of the internet, data exchange and sharing have increased. Files containing text, pictures, videos, and audio can be effectively shared by people around the world. However, this network has made life easier and also brought serious security problems. These data are easily accessible by third parties when confidential data is shared or sent from a public network, such as the Internet. In the face of these serious dangers, new applications and new system protection mechanisms have emerged with developing technology to prevent data hiding and changing. In these applications, many cryptography and steganography algorithms have been created and existing algorithms have been developed in order to increase the security of data, and new algorithms are still being developed. With the advances in today's technology, the security of data is becoming more and more difficult. Security is considered the most fundamental factor in any communication system. Increasing attacks on communication channels, accessing data that needs to be kept confidential, and transferring or changing information from one source to another with the help of some systems create serious problems for the owners of the information. This issue poses a serious and dangerous threat since the lack of confidentiality will affect individuals, societies and states in a very bad way. Increased methods and techniques of cyber attacks have facilitated access to existing data in systems. The importance of data privacy and integrity has increased. When data is accessed by unauthorized persons, negative consequences such as theft, alteration and damage to the system occur.

The algorithm developed in this thesis first encrypts the data and then encodes the data into the audio file. If the data is moved from one location to another, the data will not be seen by third parties as the data is hidden. However, it will not be able to access the data hidden by unauthorized persons due to various attacks and analyzes. Because the hidden data was first encrypted with an algorithm and then hidden in the sound. Even if the confidential data in the voice is reached, it will get meaningless data because it is encrypted. By using Taylor, this algorithm has been strengthened against language analysis and open text attacks. With this hybrid model obtained by combining cryptography and steganography, a different perspective on cryptology has been put forward. In the applications made with the proposed algorithm, it has been proved once again that mathematics is vital in the field of cyber security due to cryptology.

Keywords: Voice Steganography, Cryptography, Data Steganography, Video Steganography, Text Steganography

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

SİMGELER VE KISALTMALAR

LSB (LeastSignificant Bit) : En Anlamsız Bit

DNA : Deoksiribo Nükleik Asit

KB : Kilobayt

TCP (Transmission Control Protocol) : Geçiş kontrol protokolü

IP (Internet Protokol) : İnternet Protokolü

SMS (Short Message Service) : Kısa Mesaj Servisi

HTML (Hypertext Markup Language) : Köprü Metni Biçimlendirme Dili

XML (Extensible Markup Language) : Genişletilebilir İşaretleme Dili

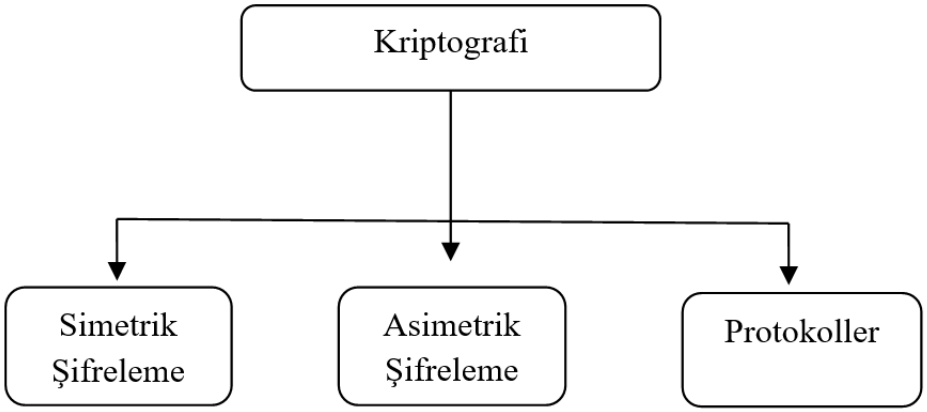
SSCE (Senior Secondary Certificate of Education) : Kıdemli Ortaöğretim Eğitim Sertifikası

1. GİRİŞ

Güvenlik, herhangi bir haberleşeme sisteminde en temel faktör olarak kabul edilmektedir. Günümüz teknolojilerindeki gelişmelerle beraber, iletişim kanallarına yapılan saldırıların artması, gizli tutulması gereken verilere dışarıdan ulaşılması ve bazı sistemler yardımıyla bilginin kaynağından başka bir yere aktarılması veya bilginin değiştirilmesi, bilginin sahipleri için çok ciddi sorunlar oluşturmaktadır. Bilginin gizli kalmaması bireyleri, toplumları ve devletleri çok kötü şekillerde etkileyeceği için bu konu önemli ve tehlikeli bir tehdit oluşturmaktadır.

Özellikle son yıllarda bilgi ve bilgisayar sistemlerinin güvenliği oldukça önemli bir konu olarak karşımıza çıkmaktadır. Son dönemlerde internetin yaygınlaşmasıyla veri alışverişi ve paylaşımı da artmıştır. Dünyanın birçok yerindeki insanlar tarafından metin, resim, video, ses vb. gibi birçok veriyi içeren dosyalar, etkin bir şekilde paylaşılabilmektedir. Fakat bu iletişim ağı hayatı kolaylaştırdığı gibi çok ciddi güvenlik sorunlarını da beraberinde getirmiştir. İnternet gibi herkese açık olan bir ağdan gizli tutulan veriler paylaşıldığı veya gönderildiği zaman üçüncü şahıslar tarafından bu verilere kolaylıkla erişilebilir. Bu ciddi tehlikeler karşısında verilerin gizlenmesi ve değiştirilmesini engellemek için gelişen teknoloji ile birlikte yeni uygulamalar ve yeni sistem koruma mekanizmaları ortaya çıkmıştır. Kullanılan bu uygulamalarda kriptografi ve steganografi olarak birçok şifreleme algoritması oluşturulmuş ve verilerin güvenliğini daha da artırmak için var olan algoritmalar geliştirilmeye devam edildiği gibi hâlâ yeni algoritmalarda oluşturulmaktadır.

Kriptografi, bir mesajın anlamını gizleme amacı taşıyan gizli yazı bilimidir [1, 2]. Şekil 1.1'de gösterildiği gibi kriptografi; Simetrik Şifreleme, Asimetrik Şifreleme ve Şifreleme Protokolleri olarak üç ana dala ayrılır.



Şekil 1.1. Kriptografi'nin dalları [1]

Simetrik şifrelemelerde kullanılan, simetrik algoritmalar, antik dönemlerden 1976 yılına kadar olan bütün şifreleme çeşitleri, yalnızca simetrik yöntemlere dayanıyordu. Simetrik şifreleme algoritmaları şifreleme ve deşifreleme işlemleri için tek bir gizli anahtar kullanmaktadır. Şifreleme işlemlerini gerçekleştirdikten sonra şifreli metni alıcıya gönderirken şifreli metinle birlikte gizli anahtarı da alıcıya güvenli bir şekilde göndermesi gerekmektedir. Simetrik şifreleme algoritmaları çok hızlı şifreleme ve deşifreleme işlemleri gerçekleştirebildiğinden dolayı günümüzde çok yaygın olarak kullanılmaktadır [1].

Asimetrik şifrelemede kullanılan, asimetrik algoritmalar, 1976'da Whitfield Diffie, Martin Hellman ve Ralph Merkle tarafından tamamen farklı bir şifre türü ile tanıtıldı. Simetrik şifreleme tekniğinde, bulunan anahtar dağıtım problemini çözmek için şifreleme ve çözme işlemlerinin her birisi için ayrı ayrı anahtar kullanma prensibine dayanan bir şifreleme sistemi geliştirilmiştir. Bu sistemde şifreleme işlemi herkes tarafından bilinen açık anahtarla yapılır. Şifreleme ve çözme işlemi birbirinin simetriği olmayan algoritmalarla gerçekleştirildiğinden dolayı da asimetrik şifreleme sistemi olarak bilinir [3].

Şifreleme protokolleri olarak adlandırılan Kriptografik Protokol Algoritmaları, basitçe iki veya daha fazla kişi arasındaki önceden belirlenmiş belli bir amaca yönelik haberleşme metodu olarak tanımlanabilir. Kriptografik protokol ile kriptografik algoritma içeren bir protokol kastedilir. Burada temel amaç verilerin bütünlüğünü ve gizliliğini sağlamaktır. Kriptografik algoritmalarda bir protokolün güvensiz olduğunu ispatlamak güvenliliği ispatlamaya göre çok daha kolaydır. Bir protokolü incelerken tıpkı algoritmalardaki gibi protokolü nasıl bir cihazda hayata geçireceğimizden çok temel çalışma prensipleri ile ilgileniliriz [4].

Korunmasını istediğimiz veriler; gönderilecek verinin bir anahtar ve belirtilen şifreleme algoritması yardımıyla anlaşılmaz hale getirilikten sonra gönderilir. Ancak bu şekilde şifrelenmiş verilerin zaman içinde kripto analistler tarafından kırılabilmesi, şifrelemenin güvenli iletişim için tek başına yeterli olmadığını göstermektedir. Bu nedenle şifreleme ve bilgi gizleme yöntemleri, şifreleme algoritmaları ile hibrit olarak kullanılmaya başlanılmıştır. Özellikle de Steganografi ile birlikte kullanılarak güvenli bir iletişimin yapılması mümkün olduğu gösterilmiştir.

Veri gizleme ve veri iletişim güvenliği oldukça önemli bir konudur. Veri gizlemede amaç iletişim esnasında üçüncü kişinin fark edemeyeceği şekilde yapılmasıdır. Kriptografik şifrelemede üçüncü kişi gizli bir verinin gönderildiğinden haberdardır. Ancak steganografik yöntemlerle 2 kişi arasında veri iletişimi yapıldığında üçüncü kişi arada gizli bir iletişim olduğunu fark edememektedir.

Bilgi gizleme ihtiyacı çok eski yıllara dayanmaktadır. O yıllarda kullanılan teknikler günümüz teknolojisindeki gelişmelerle beraber yeni yöntemlere dönüştürülmüş ve çok daha yeni yöntemler geliştirilmiş ve geliştirilmeye devam edilmektedir [5]. Bilgi gizlemede oldukça önemli olan Steganografi, sayısal ortamdaki verilerin, metin, ses, video ve görüntü dosyalarının içerisine gömülerek, korunması için son yıllarda sıklıkla kullanılmaktadır. Bazı tekniklerde hibrit model kullanılmaktadır. Veriyi ilk olarak şifreleyip daha sonra şifrelenen veriyi gizleme işlemi yapılmaktadır. Metin alfabedeki harflere karşılık gelen sayı değerleri şifrelenerek ses dosyası içerine gizleme yapılmasıdır. Böyle ilk olarak şifreleme yapıldı. Daha sonra gizleme yapılarak hibrit model kullanıldı [6].

Steganografi, teknolojinin gelişmesiyle birlikte birçok yöntemi kullanmaya başlamıştır. Bu gelişmeyle birlikte birçok steganalitik yöntem farklı algoritmalara ihtiyaç duymaya başlamıştır. Her steganografik yöntemde farklı bir algoritma ve farklı bir yöntem kullanılmıştır. Bu algoritmaları çözmek için de steganalitik yöntem çeşitlerine karşı steganaliz yöntemler geliştirilmiştir. Bir steganografi algoritması için geliştirilen steganaliz yöntem bir diğer steganografi algoritması için çalışmamaktadır. Her steganografik metodunun kendine özgü bir steganaliz yöntemi bulunmaktadır.

Bu tez çalışmasında iletişimin güvenliği sağlanırken kullanılan şifreleme tekniğinden ve veri gizleme tekniğinden bahsedilmiştir. Şifrelemede kullanılan algoritmaların dayanıklığı artırmak için matematiksel fonksiyonların kullanılabileceğini göstermek amacıyla taylor serisi tabanlı bir şifreleme algoritması geliştirildi. Daha sonra

önerilen algoritma ile şifrelenmiş verinin varlığını gizlemek için steganografi tekniklerinden ses steganografi kullanıldı. Bu çalışmanın birinci bölümünde kriptografik teknikleri ile steganografik teknikleri hakkında literatür taraması yapıldı. İkinci bölümde önerilen algoritma tekniği hakkında bilgiye yer verilmiştir. Üçüncü bölümde önerilen algoritma tekniğin uygulaması adımlar halinde yapıldı. Dördüncü bölümde deneysel sonuçlar hakkında bilgilere yer verildi. Beşinci bölümde önerilen yöntemin performans analizi yapıldı. Altıncı bölümde elde etimiz sonuçlar ve tartışmalara yer verildi.

2. KRİPTOGRAFİ VE STEGANOGRAFİ

2.1. Kriptografi

Kriptografi (Cryptography); mesajın açık halden anlamsız bir hale dönüştürülme işlemidir. Diğer bir değişle matematiksel yöntemler kullanarak mesajın şifrelenmesidir. Verilerin gizliliğini, bütünlüğünü, güvenliğini sağlar. Mesajın anlaşılabilir haline açık veya düz mesaj denilmektedir. Açık mesajın matematiksel yöntemler kullanarak bazı işlemlerde geçirildikten sonra elde edilen, mesajın anlamsız haline şifreli mesaj denilmektedir [7, 8, 9]. Günümüzde dijital iletişim kanalları kullanılmaktadır. Bu iletişim kanallarında birçoğumuz özel bilgimizi ve sırlarımızı paylaşıyoruz. Kesin olarak, her tür iletişim korunmasızdır ve manipülasyon için siber suçlulara açıktır. Dolayısıyla modern şifreleme acil bir ihtiyaç haline gelmiştir. Hassas veya gizli bilgileri korumak zorunlu hale gelmiştir. Bu nedenle, bilgiyi okunamayan bir formata dönüştürmemiz gerekir ki; yalnızca yetkili olanların erişimi mümkün olabilsin [10].

2.1.1. Kriptografi'nin Tarihçesi

Kriptografi, MÖ 1900 yıllarda Mısırlılar yazdığı kitabelerde standart dışı hiyeroglifler kullanmıştır. Mısırların yazdığı kitabelerdeki amaç mesajı gizli iletmekten öte anıt üzerinde yazan metnin daha esrarengiz görünmesini sağlamak aynı zamanda okunuşunu da zorlaştırmaktı. Eski Mısır'da bu yapılan çalışma için rebus olarak bir yöntem genellikle kullanılmış. Buna göre okunuş sırasında piktogramların (somut varlıkları bir resimle simgelemek) anlamlarını göz ardı edip sadece ses değerine önem vererek mesajı daha gizemli hale getirebiliyorlardı. Bu yazılan kitabeler bilinen ilk yazılı kriptografik belgelerin olduğunu göstermektedir.

MÖ. 590: Jeremiah'ın yazdığı kitabında yer alan İbranice yazılarda Atbash olarak da bilinen ve alfabeyi ters kullanılarak gerçekleştirilen basit bir yöntemdi. Bu kullanılan yöntem basit bir yerine koyma şifresi olarak kullanılmıştır. Atbash o yıllarda kullanılan İbranice şifreleme tekniklerinden biridir.

MÖ. 487: Yunanlılar ilk askeri kriptografi şeklini bulmuşlardır. Askerler, Skytale adı verilen tahta bir çubuğu kullanmışlardır. Bu kullanılan Skytale etrafına şerit şeklinde parşömen ya da deri sarılabilen bir tahta çubuğudur.

MÖ. 300: Artha-sastra Kautilya tarafından kaleme alınmış bir kitaptır. Hindistan'da kaleme alınmıştır. Bu kitap istihbarat bilgilerine ulaşılmasını sağlayan kodların deşifre edilmesi anlamına gelen çeşitli kriptoanaliz tekniklerini önermektedir.

MÖ. 100-44: JuliaCeasar devlet iletişiminde 'Sezar şifrelemesi' olarak bilenen ve mesajdaki her harfin alfabede kendisinden sonra 3 harf ötelenmesiyle oluşturduğu şifreleme tekniğini kullanmıştır.

725- 790: İlk Arap sözlüğünün de mucidi olan Abu Abd al-Rahman, Bizans imparatoru için Yunanca yazılmış bir şifrelenmiş yazı nasıl çözdüğünü anlatan bir kitap yazmıştır (Bu kitap şuan nerede olduğu bilinmemektedir).

1000-1200: Gaznelilerden günümüze kadar gelen bazı belgelerin şifreli metinler olduğu tespit edilmiştir.

1586: Blaise de Vigenere kriptografi ile ilgili bir kitap yazdı. Bu kitapta ilk kez açık metin ve şifreli metin için otomatik anahtarlama tekniğinden söz edilmiştir. Çağımızda bu yöntem DES, CBC ve CFB kiplerinde hala kullanılmaya devam etmektedir. 790: Pensilvanya Üniversitesi'nde Thomas Jefferson matematikçi olarak wheel (tekerlek) şifresini buldu. Bu şifreleme yöntemi M-138-A Strip Cipher makinesinin gelişmesinde rol aldı. ABD donanması 2.Dünya savaşında M-138-A Strip Cipher makinesini kullandı.

1854: Charles Wheatstone playfair şifresini bulmuştur. Şifrenin adını arkadaşı olan Lyon Playfair den almıştır. 1861: Friedrich W. Kasiski, tekrarlan harf gruplarını kullanan çok alfabeli şifrenin ilk genel şifre çözümünü veren bir kitap yazdı. Böyle yıllarca güçlü olarak görülen çok alfabeli şifre artık zayıf duruma gelmiştir. 1. Dünya savaşı 1914 başladı. Bu savaşta güçlü bir askeri kriptografisi kullanılmaya başlanmıştır. İngilizler iletişim sırasında mesajlarını kriptografik araçlar vasıtasıyla gerçekleştirmişlerdir. Fransızlar ise 4 basamaklı kelimeleri ifade eden sayıları kullanmışlardır.

1919: Joseph Mauborgne ve Gilbert Vernam mükemmel şifreleme yöntemi olan "one-time pad" i buldular.

1920-1930: William Frederick Friedman, Riverbank Laboratuarlarını kurdu, ABD için kriptoanaliz yaptılar, 2. Dünya savaşında Japonlar'ın Purple Machine şifreleme yöntemini kırmayı başardılar.

1923: Arthur Scherbius Enigma makinasını bulmuştur. Almanya devleti bu makineyi inceliyip geliştirirek ve 2. Dünya savaşında askeri iletişimi şifrelemek için kullandı. Bu makina Alan Turing ve kurduğu ekip tarafından deşifre edildi.

1933: Japonlar mesajları şifrelemek için Purple makinasını yaptılar. Bu makine rotorlar kullanmıştır. Rotorlar alfabedeki harflerin yerini değiştirerek çalışmaktadırlar.

1952: ABD'de Resmi olarak Ulusal Güvenlik Teşkilatı (NSA) 04.11.1952 kurulmuştur.

1970: Horst Feistel (IBM) DES'in iskeletini oluşturan Lucifer algoritmasını geliştirdi.

1976: Açık Anahtarlı Kriptografi algoritmaları geliştirildi. Whitfield Diffie ve Martin Hellman New Direction in Crytography (Kriptografi'de Yeni Yöntemler) kitabında açık anahtarlı kriptografiyi yayınladı.

1976: DES (Data Encryption Standard), ABD tarafından FIPS 46 (Federal Information Processing Satndard) standardı olarak belirtildi.

1978: Ronald L. Rivest, Adi Shamir ve Leonard M. Adleman: RSA Algoritmasını yaptılar.

1985: Neal Koblitz ve Victor S. Miller ayrı yaptıkları çalışmalarda eliptik eğri kriptografik (ECC) sistemlerini açıkladılar.

1990: Xuejia Lai ve James Massey: IDEA algoritmasını buldular.

1991: Phil Zimmerman: PGP tekniğini inceleyip, geliştirdi ve yayınladı.

1995: SHA-1(Secure Hash Algorithm) özet algoritması NIST tarafından standart olarak ilan edildi.

1997: ABD'nin NIST (National Institute of Standards and Technology) kurumu DES'in yerine geçeçek bir simetrik algoritma için yarışma yaptılar.

2001: NIST'in yarışmasında birinci olan Belçikalı Joan Daemen ve Vincent Rijmen'e ait Rijndael algoritması, AES (Advanced Encryption Standard) adıyla standart hale getirildi [11].

2002-2019: Birçok şifreleme algoritmaları geliştirildi. Bazı özel şirketler sadece kendi verilerini korumak için algoritmalar geliştirdiler. Güvenlik nedeniyle bu algoritmaların tanıtımını bile yapılmadı. Günümüzde mevcut algoritmalar geliştirilmekte ve yeni teknikler kullanılarak yeni şifreleme algoritmalar oluşturulmaya devam etmektedir.

2.1.2. Kriptografi'nin Yöntemleri

Kriptografik yöntemler simetrik şifreleme, asimetrik şifreleme ve hibrit şifreleme olarak bilinir.

2.1.2.1. Simetrik Şifreleme Algoritmaları

Bu algoritmada şifreleme ve şifre çözmek için bir tane gizli anahtar kullanılmaktadır. Mesajı şifrelemekte kullanılan gizli anahtar ile şifreli mesajı çözmek için kullanılan anahtar aynıdır. Bundan dolayı şifreli mesajı çözmek için alıcı anahtarı bilmek zorundadır. Mesajı şifreledikten sonra şifreli mesajla birlikte anahtarı göndermesi gerekmektedir. Veya daha önce taraflar arasından anahtar konusunda mutabık olunmuştur. Simetrik şifrelemenin üstünlüklerinden en önemlisi olan şifreleme işleminin çok hızlı yapılmasıdır. Simetrik şifrelemeyi asimetrik şifrelemeyle mukayese edildiğinde hız konusunda simetrik algoritmalar asimetrik algoritmalara göre çok daha başarılıdır. Bundan dolayı simetrik algoritmanın basit işlemlerde elektronik cihazlarda uygulamak çok daha pratik oldu. Simetrik algoritmalarda kullanılan anahtarın uzunluğu ve dolayısıyla bit sayısı çok daha küçüktür.

2.1.2.2. Asimetrik Şifreleme Algoritmaları

Simetrik şifreleme algoritmalarında mesaj birden fazla kişilere iletilmesi durumlarda karşılaşılan en büyük problem anahtar mesajı çözecek kişilere iletilmesidir.

Simetrik şifreleme algoritmalarında anahtarın dağıtılması güvenlik açısından problemli olabilmektedir. Mesajı alan tüm kişilere farklı anahtarlar vermek istenildiğinde ise uygulamada birbirinden farklı anahtar bulunacağı için bu durum sorun teşkil edebilir. Bu sıkıntılara gidermek için asimetrik şifreleme algoritmaları geliştirilmiştir. Bu geliştirilen algoritmada anahtar ile şifre çözme anahtarı birbirinden tamamen değişiktir. Şifreleme işlemi yapan anahtara açık anahtar denilmektedir. Şifreyi çözülmesini sağlayan anahtar ise özel anahtar olmalıdır. Kullanılan açık anahtarlar herkese iletilebilir, fakat hangi anahtarın hangi kullanıcıya ait olduğu bilinmelidir. Bu sebepten sertifikalara başvurulmaktadır. Sertifika açık anahtar ile kullanıcının kimliği arasındaki bağlantının belgesidir. Şifreyi çözecek kullanıcıda özel anahtar bulunur. Bu anahtar sadece şifreyi çözen kişiye aittir. Açık anahtar ise gizli değildir. Bu sebepten güvenlik açısından simetrik şifrelemeye algoritması, asimetrik şifreleme algoritmasına göre başarısızdır. Simetrik şifrelemede ortaya çıkan anahtar dağıtımının önüne geçmiş olur Bu sebepten dolayı donanımsal uygunluk ve hız gibi konularda asimetrik şifreleme, simetrik şifreleme göre geride kalmıştır. Asimetrik algoritmalar çok büyük asal sayılar kullanarak güvenliğini sağlayabilmektedir. Ancak donanımsal yapılara uyum sağlaması zor olmaktadır. Hız açısından da çok büyük sıkıntılar yaşanmaktadır. Simetrik bir algoritmayı kullanan sistemler asimetrik algoritmaları kullanan sistemlere göre çok daha hızlıdır.

2.1.3. Kriptografide Kullanılan Mevcut Bazı Algoritmalar

DES (Data Encrytion Standard - Veri Şifreleme Standartı)

DES blok halinde şifreleme yapan bir algoritmadır. Şifrelemeyi metin boyutu belli olan bloklar halinde yapmaktadır. DES algoritması 56 bit uzunluğunda simetrik kriptolama tekniğidir. Fakat 64 bitlik anahtar uzunluğuna sahiptir. Kullanıma özel yeni bir anahtar oluşturduğu için bu durum DES algoritmasını güçlü hale getirmiştir. Günümüzde kullanılan teknolojiye göre yavaş ve 56 bitlik anahtar uzunluğu yetersiz olması DES algoritmasını zayıflatmıştır. DES algoritması 2000'li yıllarında ününü kaybetme sebebi kırılmasıdır ve günümüz teknolojisinin gerisinde kalmıştır. DES algoritmasının yetersiz kalması durumunda "Triple DES" veya "DES-3" olarak bilinen yeni bir algoritma geliştirilmiştir. SSH gibi günümüzde kullanılan çoğu uygulama DES-3'i kullanmaya başlamıştır. DES şifrelemesinin 3 kez yapılmasından dolayı DES-3 adını

almıştır. Bu sebepten ötürü DES-3, DES'e göre 3 kat daha yavaş çalışmaktadır. DES-3 şifreleme için uzunluğu 24 bayt olan bir anahtar kullanılmaktadır. Her bayt için 1 eşlik biti vardır, bu sebepten dolayı anahtarın uzunluğu 168 bittir. DES algoritmasının etkinliğini kaybetmesinin sebebi AES algoritmasının geliştirilmesidir. Çünkü AES şifreleme yöntemi DES şifreleme yöntemine göre 6 kat daha hızlıdır.

AES (Advanced Encrytion Standard - Gelişmiş Şifreleme Standartı)

Des algoritmasına oranla daha güvenli bir sistemdir. 2001 yılında geliştirilmiştir bunun sebebi DES algoritmasının kırılmasıdır. Belçikalı Joan Daemen ve Vincent Rijmen tarafından bulunmuştur. Bu sistem blok şifreleme algoritmasıdır. DES'in zayıf kalan yönlerini geliştirmek suretiyle yapılmıştır. Üç farklı anahtar uzunluğuna sahip olabilmektedir. Bunlar; 128 bit, 192 bit ve 256 bittir. AES şifreleme yönteminin DES şifreleme yöntemine göre güçlü yönleri donanımda ve yazılımda hızlı olması, daha kolay uygulanabilir olması ve çok daha az hafızaya gerek duymasıdır. Günümüzde bilinen tüm pratik ve doğrudan saldırılara karşı dayanıklı olduğu düşünülmektedir. En çok tercih edilen simetrik şifreleme algoritmasıdır.

Blowfish

Blowfish şifreleme yöntemi, 64-bit metin uzunluğuna ve 32 bit'ten 448 bit'e kadar anahtar uzunluğuna sahiptir. DES şifreleme yönteminin yetersiz kalmaya başlamasından sonra Blowfish şifreleme yöntemi geliştirilerek DES'in yerine geçmeyi hedeflemiştir. Blowfish şifreleme yöntemi için ram'de 4 kb yere ihtiyaç duyar. Bu sebepten dolayı en küçük sistemler olan akıllı kartlar gibi sistemlerde kullanılmaz. Rutin kullanıcı uygulamalarından olan e-posta konusundaki etkinliğiyle başarılı bir algoritma olarak değerlendirilmektedir. Blowfish şifreleme yöntemini cazip kılan özelliklerinden biri ise tamamen ücretsiz olmasıdır. Piyasadaki en hızlı şifreleme yöntemine sahip ve karmaşık olma özelliği taşımaktadır.

DH (Diffie-Helman)

DH şifreleme algoritması Diffie ve Helman tarafından 1976 yılında kullanıcılara sunmuştur. Asimetrik şifrelemenin ilk örneklerindendir. DH şifreleme algoritması iki kullanıcının öncesinde herhangi bir bilgi alışverişi yapmadan güvenli bir kanal

vasıtasıyla ortak bir şifrede karar kılmalarına yarayan bir protokoldür. Algoritmada anahtar değişiminin asıl amacı, iki kullanıcının belirtilen anahtarı güvenli bir şekilde karşı tarafa göndermeleri ve daha sonrada bu anahtarı kullanarak karşı kullanıcıya şifreli mesajları gönderebilmelerini sağlamaktır. DH şifreleme algoritması ile simetrik şifrelemenin büyük sorunlarından olan gizli anahtar dağıtım ve koruma büyük ölçüde aşılmıştır. Bu sebeple birlikte DH şifreleme algoritması sadece ortak gizli anahtarı belirlemekte kullanılmaktadır.

RSA (Rivest-Shamir-Adleman)

RSA algoritması 1977 yılında R.Rivest, A.Shamir ve L.Adleman isminde üç bilim adamının oluşturduğu yeni asimetrik şifreleme algoritmasıdır, anahtar paylaşımının yanında şifreleme ve şifre çözme işlemlerini de yapabilmektedir. RSA, güvenilirliği çok büyük tam sayılarla işlem yapmanın zorluğuna dayanan bir şifreleme tekniğidir. RSA genel anahtarlı şifreleme tekniğidir. Bu teknik çok büyük tamsayıları oluşturma ve bu sayıları işleminin zorluğu üzerinde durulmuştur. Daha güvenli bir yapı oluşturmak için asal sayılar kullanılmaktadır. RSA ticari amaçla da kullanılan bir algoritma çeşidirir, kullanım alanı olarak hem mesaj şifreleme hem de e-imza amacı ile de kullanılır. RSA algoritmasının hızlı olması ve sistemin güvenliğini sağlaması için kullanılan anahtarın sayısal büyüklüğü önemlidir. Eliptik eğri şifreleme algoritması kullanılarak yeterli güvenilirlik derecesine ulaşmak için bu algoritma ile belirlenmektedir. RSA ile günümüzde 1024 bitlik bir anahtar (yaklaşık 300 basamaklı bir sayı) basit uygulamalar için yeterli bir şifreleme tekniği olarak kullanılabilir. RSA algoritması, bir şifreleme algoritması için oldukça basit bir algoritmadır. Buna karşın sürekli çok büyük asal sayı oluşturmak oldukça zor bir işlemdir [12].

2.2. Steganografi

Steganografi kelimesi Yunanca “steganos: gizli, saklı” ve “grafi: çizim ya da yazım” kelimelerinden gelmektedir. Anlamı “kaplanmış yazı” (coveredwriting) demektir [13]. Kelimenin tam anlamıyla "gizli yazı" demektir. Steganografi, günümüzde bir takım elektronik ortamlara veri gömme ile ilişkilendirilir. Steganografi ve daha yaygın olarak kullanılan kriptografi arasındaki fark ise şudur; kriptografi verileri karıştırıp karartırken, steganografi verileri tamamen gizlemektedir. "Gizli" ya da “açık" veriler, ana bilgisayar dosyasındaki önemsiz küçük bitler değiştirerek gizlenir

[14]. Steganografi, bilgiyi diğer bilgilerle gizleme sanatıdır [15]. Şifreleme, verileri anlaşılmaz bir formata dönüştürür, gerçek verilere erişmeyi zorlaştırır, ancak iletişimin gizliliğini sağlayamaz [16] .

Steganografi'nin amacı bilginin varlığını gizlemek veya bilgiyi fark edilmeden başka verinin içerisine yerleştirmektir. Güvenli şekilde gönderilmek istenen veriyi, dikkat çekmeyen görünüme sahip bir başka ortamda gizleyerek üçüncü şahısların gönderilen mesajın varlığından haberdar olması engellenir. Bu işlemler metin, ses, resim, video dosyaları içerisine veriyi gizleyerek yapılır. Bu veriler herhangi bir metin dosyası olabileceği gibi, herhangi bir görüntü içerisine başka bir görüntüyü gizleme şeklinde de olabilmektedir. Yine aynı şekilde bir ses veya video dosyasının içine bir metin dosyası da saklanabilmektedir [17, 18, 19]. Steganografi verinin gizli ve güvenli bir şekilde iletilmesini sağlar. Amacı iki kişi arasındaki iletişimin üçüncü bir şahıs tarafından fark edilmesini engellemektir. Bilimsel ortamda Steganografi çalışmaları 1983 yılında Simmons tarafından "Prisoner Problem"in tanımlanması ile başlamaktadır [20]. Bu problemde Alice ve Bob hapishanededir ve hapishaneden kaçmak için planlar yapmaktadırlar. Fakat bu planların gardiyan Willie'ye fark ettirilmeden yapılması gerekmektedir. Eğer Willie bunu fark ederse kaçma planları suya düşecektir. Bu nedenle de çeşitli gizli haberleşme yöntemleri geliştirilmesi gerekmektedir.

Steganografi, kriptografiye yakın olmasına rağmen kriptografiden çok faklıdır. Kriptografi mesajın içeriğinin korunması ile ilgilenirken steganografi mesajın varlığının gizlenmesi ile ilgilenmektedir. Dolayısıyla steganografi bir şifreleme yöntemi değil şifrelemeyi tamamlayıcı bir öğedir [21].

2.2.1. Steganografinin Tarihçesi

Geçmişten bugüne kadar veriyi gizleme amacıyla birçok değişik teknik kullanılmış ve geliştirilmiştir. Steganografi'nin geçmişte bilinen ilk örneği olarak kabul edilen hikâye Herodot'a aittir. Eski Yunanistan'da Herodot M.Ö. 425-486 tarihleri arasında Histiaeus adlı çok güvendiği bir kölesinin kafasını kazıtmış ve dövme yaptırmıştır. Dövmede yer alan bu gizli bilgi kölenin saçları uzadığında gözükmemektedir. Böylece Persler fark etmeden bilgi iletimi sağlanmıştır. Eski çağlarda kullanılan bir başka yöntem ise 1. ve 2. Dünya Savaşı'nda kullanılan mektup aracılığıyla görünmez mürekkeplerle satır aralarında veriyi taşımaktır. Bu görünmez mürekkep içinde süt, meyve suyu gibi bileşenler içermekte olduğundan veri eline ulaşan kişi bu mektubu

ısıttığı zaman veri okunabilecek hale dönüşürdü. Daha sonra Almanlar 2. Dünya Savaşı'nda birbirlerine yolladıkları sıradan bir metnin içindeki her sözcüğün yalnızca baştan ikinci harflerine veriyi gizlemişler, bu ikinci harfleri yan yana getirince oluşan cümleyle mesajlarını iletmişlerdir. Aşağıda bir Alman casusun 2. Dünya Savaşı'nda gönderdiği bir metin örneği yer almaktadır:

"Apparently neutral's protest is thoroughly discounted and ignored. Isman hard it Blockade issue affects pretext for embargo on byproducts, ejecting suets and vegetable oils."

Her kelimenin ikinci harflerini aldığımızda oluşan mesaj şu şekildedir:

"Pershing sails from NY June 1."

Bu gizli verileri çözümleme yolları geliştikçe veriyi saklama yolları da zamanla gelişti. Almanların geliştirdiği microdot tekniğiyle veri normal boyutta sözcükler içeren bir metnin içine çok ufak yazılarak saklandı. 1999 yılında ise New York'taki MountSinai School of Medicine veriyi DNA içinde Sitozin, Timin, Adenin gibi bazların yerleri değiştirerek saklamayı başardı. Böylece bir milyon veri yalnızca bir gram DNA içinde saklanabilir oldu. Bilgisayar teknolojisi geliştikçe işletim sistemlerinde veri gizlenmeye ve taşınmaya başlandı. Windows 95 FAT 16 sisteminde 1 KB veri bile işlem görse 32 KB yer ayırdığını fark eden uzmanlar bu kullanılmayan alana veriyi gömmeye başladı. TCP/IP paketlerinin kullanılmayan yerlerine de bilgi gizlendikten sonra steganografi günümüzde çok büyük bir gelişme göstermiştir. Bilgisayar teknolojisi ve internetle yeni bir hayat bulan steganografi günümüzde yalnızca metin değil, ses ve video dosyalarına sayısal formda 1 ve 0 olarak kodlanarak uygulanmaktadır [22].

2.2.2. Steganografinin Alt Alanları

Steganografi, Dilbilim Steganografi (Linguistic Steganography) ve Teknik Steganografi (Technical Steganography) olmak üzere kendi içerisinde ikiye ayrılmaktadır [23]. Dilbilim steganografi, herhangi bir sayısal bilginin bazı dil bilgisine dayanan metinler içinde saklanmasına olanak tanıyan bir dizi teknik ve metot olarak tanımlanır. Burada gizlemek, yalnızca gelen metinin göze çarpmaması olarak anlaşılmamalı, yani ortografi, yazı tipi, morfoloji, söz dizimi, kullanılan dilin sözlüğü,

kelime sırası anlamıyla dışa dönük olarak karşılık gelen sıradan metin olmakla birlikte semantik bütünlük ve gramer doğruluğunu da muhafaza etmelidir [24].

Teknik Steganografi, Görünmez mürekkep, Microdotlar, Bilgisayar tabanlı (metin, ses, görüntü, resim dosyaları) vb. yöntemleri kullanarak veriyi gizleme yöntemidir [25].

2.2.3. Steganografi'nin Kullanım Alanları

Steganografide metin, ses, resim ve video olmak üzere dört farklı şekilde veri gizleme yapılmaktadır.

2.2.3.1 Metin Steganografi

Metin steganografisi, mevcut bir metnin biçimlendirilmesini değiştirmeden, bir metindeki değişen kelimelerle, rastgele karakter dizileri üretmeye veya okunabilir metinler üretmek için bağlamsız gramerler kullanarak herhangi bir şey içerebilir [26]. Metin steganografisi'nin görüntü, ses veya video dosyasında bulunan gereksiz bilgi eksikliğinden dolayı en karmaşık olduğu düşünülmektedir. Metin belgelerinin yapısı gözlemlediklerimizle aynıdır, resimdeki gibi diğer belge türlerinde belgenin yapısı gözlemlediğimizden farklıdır. Bu nedenle, bu tür belgelerde, ilgili çıktıda belirgin bir değişiklik yapmadan belgenin yapısında değişiklikler getirerek bilgileri gizleyebiliriz [27]. Görüntü veya ses dosyasında fark edilemeyen değişiklikler yapılabilir, ancak metin dosyalarında, sıradan bir okuyucu tarafından ek bir harf veya noktalama işareti bile gösterilebilir [25]. Metin dosyasının saklanması daha az hafıza gerektirir ve daha hızlı ve kolay iletişim olduğu için diğer tip steganografik yöntemlere göre tercih edilir [28]. Metin steganografi de farklı tekniklerde kullanılmıştır. İlk olarak metni ascii'ye kodlayıp ve ardından, belirnen algoritmaya göre dönüştürerek şifreleme yapıp ve elde edilen bu şifreli metni ise bir başka metin içerisine gizleme yapılmaktadır. Bu şekilde, metin steganografi yaklaşımı kullanılarak verilere daha yüksek güvenlik özelliği sağlanmaktadır [29].

Metin steganografisi üç ana kategoride sınıflandırılabilir. Rastgele İstatistiksel, Biçim Tabanlı ve Dilbilim yöntemleri [30].

Biçim Temelli Yöntemler

Biçim temelli yöntemler, verileri gizlemek için kapak metninin biçimlendirmesini kullanır ve değiştirir. Herhangi bir kelimeyi veya cümleyi değiştirmezler, bu yüzden kapak metninin ‘değerine’ zarar vermezler. Format tabanlı metin steganografi yöntemi açık alan metodudur [31]. Bu yöntemde, bilgiyi gizlemek için metne ekstra boşluklar eklenir. Bu beyaz boşluklar sonra eklenebilir. Cümle, paragraf veya kelimenin sonu, tek bir alan "0" olarak yorumlanır ve iki ardışık boşluk "1" olarak yorumlanır. Bir belgede az miktarda veri gizli olabilse de, bu yöntem, gizli verilerin varlığını ortaya koymadan hemen hemen her türlü metne uygulanabilir. Diğer iki format tabanlı yöntem ise kelime kaydırma ve satır kaydırma yöntemidir. Sözcük kaydırma yönteminde, bazı kelimelerin yatay hizalamaları, gömme bilgisi için Kelimeler arasındaki mesafeleri değiştirerek kaymaktadır [32]. Bu değişikliklerin yorumlanması zordur, çünkü sözcükler arasındaki mesafeler belgelerde çok sık görülür. Kelimelerin ve paragraflar arasındaki boşlukların manipüle edilmesi de bilgiyi saklamak için bir başka yöntemdir [33]. Satır kaydırma yönteminde, metnin bazı satırlarının dikey hizalamaları, içinde ileti gömmek için benzersiz bir gizli şekil oluşturmak üzere kaydırılmıştır [30].

Rastgele ve İstatistiksel Üretim Yöntemleri

Rastgele ve İstatistiksel Üretim, bilinen bir sade metin ile karşılaştırmayı önlemek için steganograflar kendi kapak metinlerini üretirler [26]. Bu yöntem, rastgele aranan karakter dizisinde bilgi gizlemektir. Başka bir yöntemde, sözcük uzunluğunun ve harf frekanslarının istatistiksel özellikleri, verilen dildeki gerçek kelimelerle aynı istatistiksel özelliklere sahip görünecek kelimeleri oluşturmak için kullanılır [34, 35].

Dilbilim Yöntemi

Dilsel yöntem, metnin dilsel özelliklerini değiştirmek için kullanır. Bu yöntem de bilgi gizlemek için bir yer olarak mesajın dilsel yapısı kullanır. Söz dizimsel yöntem, bir veriyi gömmek için virgül (,) ve nokta (.) gibi bazı noktalama işaretlerinin belgenin uygun yerlerine yerleştirildiği dilsel bir steganografi yöntemidir. Bu Yöntem, işaret taramasının yerleştirildiği yerlerin doğru bir şekilde tanımlanmasını gerektirir.

Dilbilimsel steganografi yöntemi olarak kullanılan bir diğer yöntem ise Semantik yöntemdir. Bu yöntemde, önceden seçilmiş bazı kelimelerin eşanlamları kullanılır. İçindeki bilgileri gizlemek için sözcüklerin yerini eşanlamlıları alır [30].

Metin Stenografisinde En Çok Tercih Edilen Bazı Yaklaşımlar

Satır Kaydırma

Bu yöntemde gizli mesaj, metin satırlarını bir dereceye kadar dikey olarak kaydırılarak gizlenir [29, 36]. İşaretlenen bir çizgi, işaretli çizginin hareket yönünü algılamak için her iki tarafında bir tane işaretlenmemiş kontrol çizgisine sahiptir [37]. Bit 0'ı gizlemek için bir satır yukarı kaydırılır ve bit 1'i gizlemek için satır aşağı kaydırılır [38]. Hattın yukarı veya aşağı kaymış olup olmadığının belirlenmesi, işaretli hattın merkez hattının ve kontrol hatlarının mesafesinin ölçülmesi ile yapılır [37]. Metin yeniden girilirse veya bir karakter tanıma programı kullanılıyorsa gizli bilgiler yok olur. Ayrıca mesafeler, mesafe değerlendirmesi özel araçlarını kullanarak da izlenebilir [29].

Sözcük Kaydırma

Bu yöntemde, gizli mesaj kelimeleri yatay olarak kaydırarak gizlenir, yani sırasıyla bit 0 veya 1'i temsil etmek için sola veya sağa gizlenir [34]. Kelime kayması, bir profili bir dalga formu olarak muamele eden ve orta bloğu sağa veya sola kaydırılan bir dalgalanmadan kaynaklanıp kaynaklanmadığına karar veren korelasyon yöntemi kullanılarak tespit edilir [37]. Bir metni doldurmak için kelimeler arasındaki uzaklık oldukça değiştiğinden, bu yöntem daha az tanımlanabilir. Ancak birisi mesafelerin algoritmasını biliyorsa stego metnini algoritmayla karşılaştırabilir ve farkı kullanarak gizli içeriği elde edebilir. Ayrıca, yeniden yazmak veya OCR programlarını kullanarak gizli bilgileri yok eder [29, 36]. Buna benzer geliştirilen bir algoritma ise metni matematiksel fonksiyonla şifreme işlemi yapılıyor. Daha sonra başka bir metin içerisine elde edilen değerin bit'tine göre;

1. Bit 0 sayısı ise kelimeler arasına iki boşluk bırakmak,
2. 1 ise kelimler arasında bir boşluk bırakmak,

Yukardaki adımlara göre metin steganografisi yapılmaktadır [38].

Sözdizimsel Yöntem

Bu teknik, 0 ve 1 bitlerini gizlemek için nokta (.), Virgül (,), vb. gibi noktalama işaretlerini kullanır. Ancak bu yöntemle ilgili sorun, noktalama işaretlerini eklemek için doğru yerlerin tanımlanmasını gerektirmesidir.

Bu nedenle okuyucuların noktalama işaretlerinin yanlış kullanıldığını fark edecekleri için bu yöntemi kullanırken özen gösterilmelidir [29, 36].

Beyaz Steg

Bu teknik gizli alanları gizlemek için boşluklar kullanır. Beyaz boşlukları kullanarak verileri gizlemek için üç yöntem vardır. Cümle-Arası Aralıkta, bit 0'ı gizlemek için tek boşluk konur ve biten her karakterin sonuna, gizlemek için iki boşluk yerleştirilir. Satır aralığının sonunda, sabit satır boşlukları her satırın sonuna eklenir. Örneğin, satır başına bir bit kodlamak için iki boşluk, iki biti kodlamak için dört boşluk vb. Kelime Aralığı Yazma tekniğinde, bir kelimeden sonra bir boşluk 0 bitini temsil eder ve bir kelimeden sonra 2 boşluk 1 biti temsil eder. Fakat boşluk tutarsız olarak şeffaf değildir [25].

Spam Metin

Bitleri gizlemek için HTML ve XML dosyaları da kullanılabilir. Başlangıç ve bitişte farklı etiketler varsa bit 0 yorumlanır, başlatmak ve kapatmak için tek bir etiket kullanılıyorsa, bit 1 yorumlanır. Başka bir teknikte bit 0, bir etikette boşluk eksikliği ile temsil edilir ve bit 1, bir etikete bir boşluk yerleştirerek temsil edilir [39].

SMS-Metin Uyarısı

SMS-Metin dili, SMS'de kullanılan kısaltılmış kelimelerin birleşimidir. İkili verileri tam biçimdeki kelimeyi veya onun kısaltılmış halini kullanarak gizleyebiliriz. Sözcükler ve bunlara karşılık gelen kısaltılmış formları içeren bir kod yazısı yapılır. Bit 0'ı gizlemek için kelimenin tam biçimi kullanılır ve bit 1'i gizlemek için kısaltılmış kelime biçimi kullanılır [28].

Özellik Kodlama

Özellik kodlamada gizli mesaj, metnin bir veya daha fazla özelliğini değiştirerek gizlenir. Bir ayrıştırıcı bir belgeyi inceler ve bilgileri gizlemek için kullanabileceği tüm özellikleri seçer [39]. Örneğin, i ve j harfleri nokta olabilir, f ve t harfleri içindeki vuruş

uzunluğu değiştirilebilir veya b, d, h, vb. harflerin yüksekliğini uzatarak veya kısaltarak yapılmasıdır [40, 41]. Bu yöntemin bir kusuru, bir OCR programı kullanılıyorsa veya yeniden yazma işlemi tamamlanırsa gizli içeriğin yok olacağıdır.

Katıştırma için Gizli Steganografik Kod (SSCE)

Bu teknik ilk olarak SSCE tablosunu kullanarak bir mesajı şifreler ve sonra belirli bir haritalama tekniği kullanarak İngilizce olmayan spesifik isimlerle yazıları bir araya getirerek şifre metnini bir kapak dosyasına gömer [42]. Bu gömme pozisyonları aynı SSCE tablosunu kullanarak şifrelenir ve stego dosyasıyla birlikte alıcıya güvenli bir şekilde iletilen başka bir dosyaya kaydedilir.

Kelime Eşleme

Bu teknik, gizli bir mesajı genetik operatörü çapraz kullanarak şifreler ve daha sonra belirli bir haritalama tekniği kullanarak çift veya tek uzunluklu kelimeler arasında boşluklar ekleyerek kapak dosyasında bir defada iki bit alarak çıkan şifre metnini gömer [43]. Gömme pozisyonları başka bir dosyaya kaydedilir ve stego nesnesi ile birlikte alıcıya iletilir.

MS Word Belgesi

Bu teknikte, bir dokümandaki metin parçaları, degradasyona uğramış olup, daha az yazma becerisine sahip bir yazarın eserini taklit eder ve gizli mesajlar dejenerasyonların seçimine gömülür ve değişiklikler izlenirken revize edilir. Veri gömme işlemi, stego belgesinin işbirlikçi yazının ürünü olduğu şekilde gizlenir [44].

Kriket Maç Karnesi

Bu yöntemde, veriler bir kriket maçı puan kartında, bit 1'i göstermek için sayıdan önce anlamsız bir sıfır ekleyerek ve numarayı 0 bitini temsil edecek şekilde bırakarak gizlenir [45].

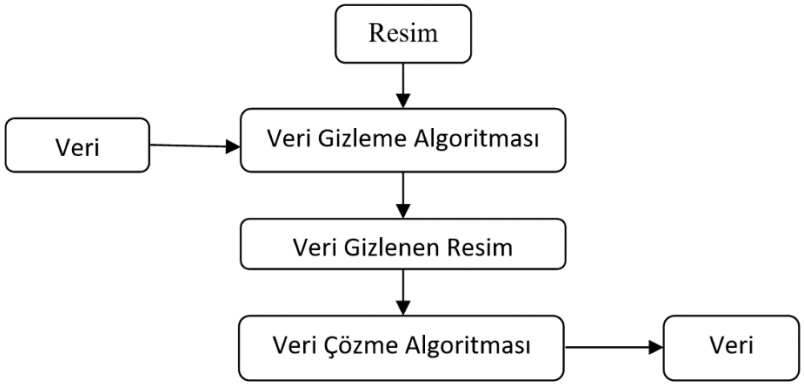
CSS (Basamaklı Stil Sayfası)

Bu teknik, bir iletiyi RSA genel anahtar kripto sistemini kullanarak şifreler ve şifreleme metni, her bir CSS stili özelliklerinde tam noktalı virgülden sonra satır sonuna kullanarak Geçişli Bir Stil Sayfasında (CSS) gömer. Noktalı virgülden sonraki boşluk, bit 0'ı katıştırır ve noktalı virgülden sonra bir sekme bit 1'i gömer [46].

2.2.3.2. Görüntü Steganografi

Steganografi konusunda yapılan çalışmalar ve geliştirilen tekniklerin en yaygın olarak kullanılanı görüntü steganografisidir. Bunun nedeni görüntülerin dağıtımı kolay olmasıdır. Kullanıldıkları algoritmalara göre farklılık göstermektedirler. Görüntü dosyalarının içerisine bir metin (text) gizlenebileceği gibi bir görüntü dosyasının içine bir başka görüntü de gizlenebilmektedir.

Şekil 2.1'de gösterilen gizli veriyi bir görüntüde gizleme işleminde iki dosya söz konusudur. Kapak resim ya da örtü verisi (coverimage) olarak adlandırılan ilk dosya, gizli veriyi gömecek resim dosyasıdır. İkinci dosya ise gizlenecek veridir. Bu mesaj da stego olarak isimlendirilmektedir. Mesaj, görüntü, açık metin (plaintext), şifreli metin (chiphertext) veya bit dizisi içinde saklanabilecek başka bilgi olabilir. Gömme işlemi sonucunda kapak resim ve gömülü mesajın oluşturduğu dosyaya "stego resim" adı verilir.



Şekil 2.1. Steganografi sistem yapısı

Görüntü steganografi de bilgiyi görüntü içine gizlemek için çeşitli yöntemler vardır. Bunlar şu şekilde sınıflandırılabilir.

- En önemsiz bite ekleme (LSB)
- Maskeleme ve Filtreleme
- Algoritmalar ve Dönüşümler [47, 48].

En önemsiz bite ekleme yöntemi uygulanması basit olduğundan dolayı en çok tercih edilen yöntemdir. Bu yöntemde; resmi oluşturan piksellerin her byte'nın en önemsiz bitinin yerine gizlenmesini istediğimiz verinin bitleri sırasıyla verinin başlangıcından itibaren birer birer yerleştirilmektedir [47].

Örneğin; resmin piksel değerlerinin binary karşılığının

10001001 11101001 11101001 10011011 10011011 10001001 00011111 00011101

Şeklinde olduğunu düşünelim. Buna M (01001101) karakterini eklersek resim

10001000 11101001 11101000 10011010 10011011 10001001 00011110 00011101

Şeklinde değiştirilmiş olacaktır.

Steganografik bir algoritma incelendiğinde üç temel unsur göz önüne alınmaktadır.

1. Değişimin fark edilememesi
2. Saklanabilecek veri miktarı
3. Dayanıklılık

Şekil 2.2'de belirli bir algoritmaya dayalı veri gizleme yöntemi kullanılmadan önce yani görüntünün içine veri gizlenmemiş haliyle görüntü yer almaktadır.

Şekil 2.2. Veri gizlenmemiş görüntü [19]

Şekil 2.3'de orijinal görüntünün içine bir algoritmaya bağlı kalınarak veri gizlenmiştir.

Şekil 2.3. Veri gizlenmiş görüntü [19]

Burada da görüldüğü gibi görüntülerde gözle görülür bir fark yoktur.

Algoritmalar kullanılarak görüntü üzerinde gerçekleştirilen değişiklikler insan gözü tarafından algılanamamalıdır. Aksi halde gizli verinin iletilmekte olduğu anlaşılacaktır. Bu durumda üçüncü kişilerin fark etmesiyle gizli veri olan görüntü üzerinde işlemler yaparak veriyi değiştirme olasılığı vardır. Steganografik yöntemlerin bu tür saldırılara karşı bir dayanıklılık göstermesi gerekir.

2.2.3.3. Ses Steganografi

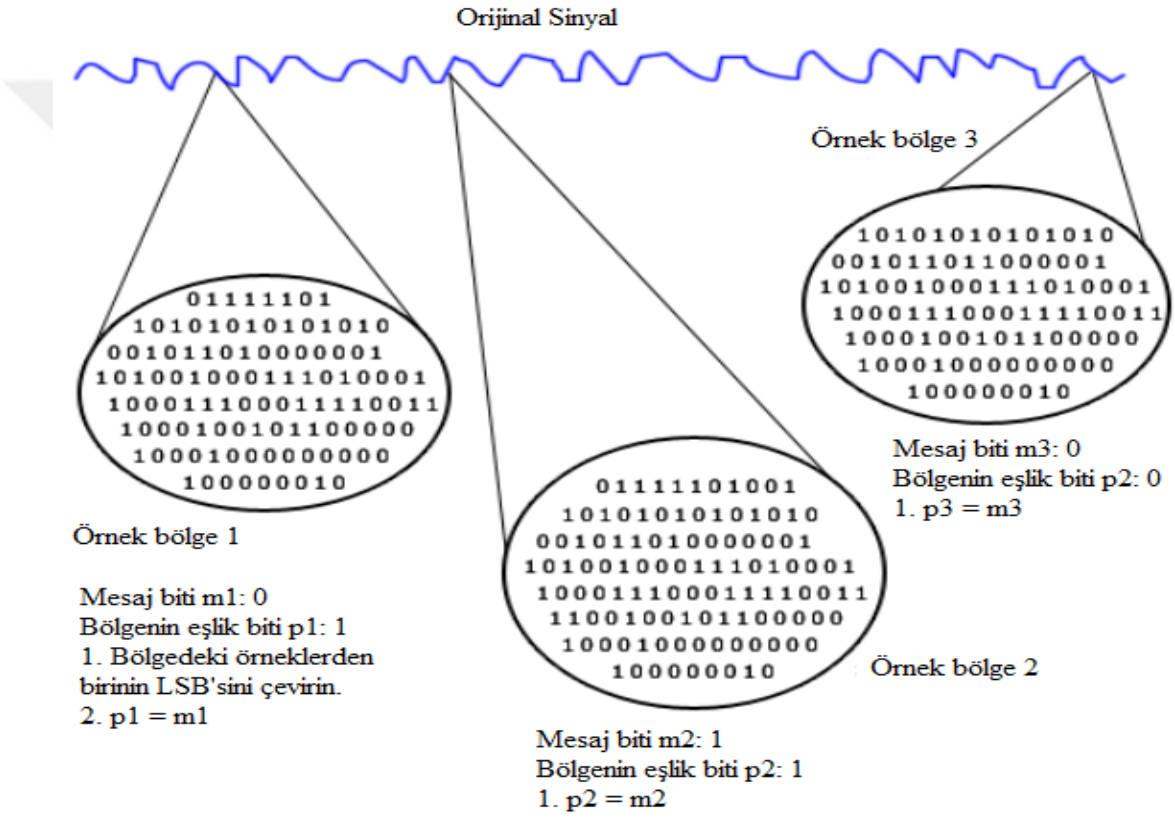
Sesle ilgili bilgileri veya mesajları, ses dosyasında yapılan değişikliklerin algısal olarak bölünmez şekilde gizlenmesi için pek çok teknikler kullanılmıştır. En çok tercih edilen bazı teknikler aşağıda kısaca açıklanmaktadır [49, 50].

En Önemsiz Bite Kodlaması

Yaygın olarak kullanılan bu metot, gizli veriyi içeren bir bayt dizisini gizlemek için kapak dosyasının bazı baytlarındaki en düşük anlamlı biti değiştiren LSB (LeastSignificant Bit) algoritmasıdır. Bu genellikle LSB değiştirilmesinin 24 bitlik bitmap'lerde olduğu gibi belirgin bir bozulmaya neden olmadığı durumlarda etkili bir teknik olarak kullanılmaktadır.

Parçalı Kodlama

Ses steganografisinde Parite kodlaması güvenli bir tekniktir. Bir sinyali tek tek örneklere bölmek yerine, bu yöntem bir sinyali ayrı örneklere böler ve gizli mesajın her bir bitini bir eşlik bitine gömer. Seçili bir bölgenin eşlik biti, şifrelenecek gizli bitle eşleşmezse, süreç bölgedeki örneklerin birinin LSB'sini tersine çevirir. Böylece, gönderen, gizli bit kodlamada daha fazla seçeneğe sahiptir. Şekil 2.4'te eşlik kodlama prosedürü görülmektedir.



Şekil 2.4. Parçalı kodlaması [51]

Faz Kodlaması

Faz kodlama tekniği, bir başlangıç ses bölümünün fazını, gizli bilgiyi temsil eden bir referans fazıyla değiştirerek çalışır. Kalan bölümler fazı, bölümler arasındaki göreli fazı korumak için ayarlanır. Sinyal / parazit oranı açısından, Faz kodlaması, en etkili kodlama yöntemlerinden biridir. Her bir frekans bileşeni arasındaki faz ilişkisinde

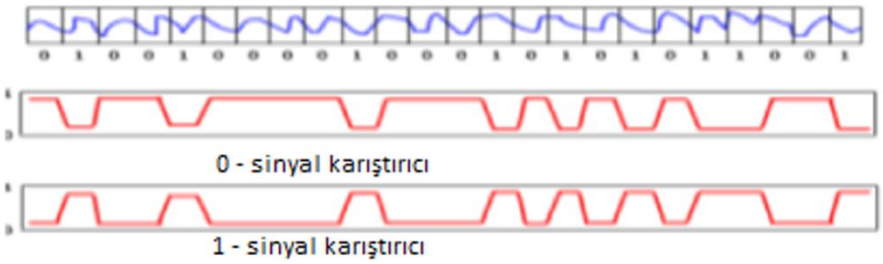
belirgin bir değişiklik olduğunda fark edilerek faz dağılımı meydana getirir. Bununla birlikte, fazın modifikasyonu yeterince küçük olduğu için duyulamaz bir kodlama elde edilebilir [52, 53].

Yaygın Spektrum

Sesli steganografi de, temel yayılma spektrumu yöntemi, gizli bilgiyi ses sinyalinin frekans spektrumuna yaymaya çalışır. Bu ileti bitlerini rasgele tüm ses dosyasına yayan LSB'nin uygulanmasını kullanan sisteme benzer. Bununla birlikte, LSB kodlamasının aksine yaygın spektrum yöntemi, gerçek sinyalden bağımsız bir kod kullanarak gizli bilgiyi, ses dosyasının frekans spektrumuna yaymaktadır [54]. Sonuç olarak, nihai sinyal, gerçekte iletim için gerekenin üzerinde bir bant genişliği kaplar.

Yankı Gizleme

Yankı gizleme tekniği, ayrı bir işarete yankı getirerek gizli bilgileri gizli bir dosyaya yerleştirir. Yankı gizleme, diğer yöntemlerle karşılaştırıldığında yüksek bir veri iletim hızı ve üstün sağlamlık sağlamanın avantajlarına sahiptir. Orijinal sinyalden sadece bir yankı çıkarsa, gizli bilgilerin yalnızca bir biti kodlanabilir. Bu nedenle, kodlama işlemi başlamadan önce orijinal sinyal bloklara ayrılır. Kodlama işlemi tamamlandıktan sonra, bloklar son sinyali oluşturmak için bir araya getirilir [55, 56]. Bu işlem Şekil 2.6'da gösterilmiştir.



Şekil 2.5. Yankı gizleme [51]

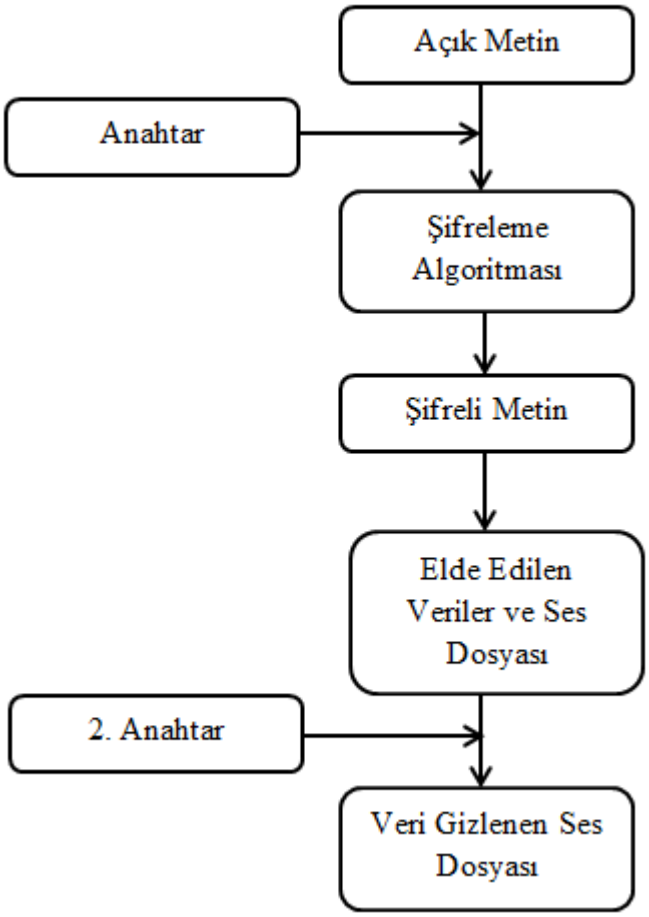
2.2.3.4. Video Steganografi

Video Steganografi, her türlü dosyayı bir video dosyasına gizlemek için kullanılan bir tekniktir. Çerçeveler ekranda çok hızlı olarak görüntülendiğinden, video dosyasındaki değişiklik, insan görsel sistemi tarafından tespit edilmesi çok daha zordur.

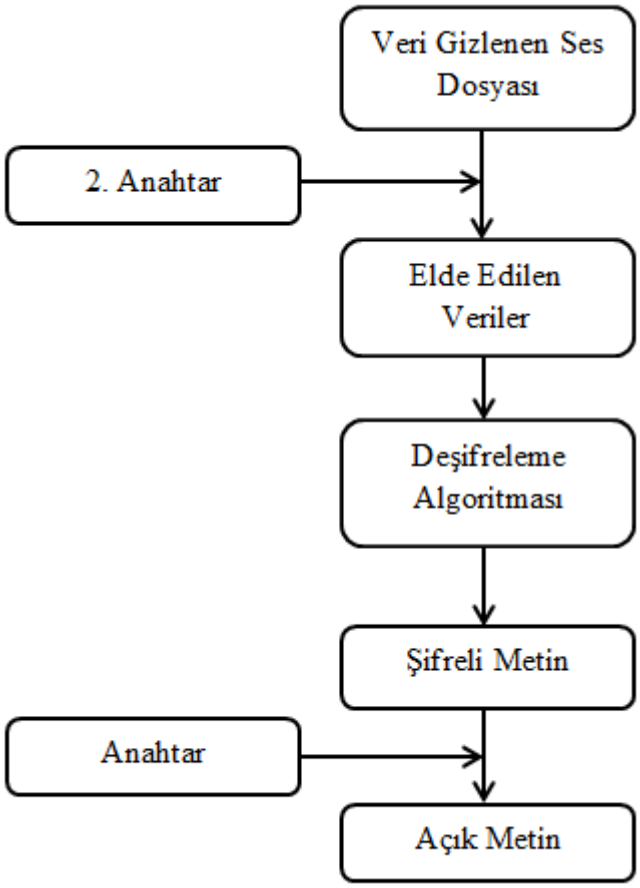
Ayrıca, video kareleri keskin odaklı görüntüler veya keskin olmadığından, steganografi ile indüklenen piksel renk varyasyonları çerçeveye kolayca karışır. Video tabanlı steganografinin kullanımı, boyutu ve bellek gereksinimleri nedeniyle diğer multimedya dosyalarından daha uygun olabilir. Videonun 2 bileşeni vardır ve bunlar bir ses akışı ve resim akışıdır. Bu nedenle görüntülerdeki ve sesteki mevcut tekniklerden çoğu, video dosyalarına da uygulanabilir. Video üzerinde steganografi de ana hatları itibariyle ses üzerindeki steganografiye benzemektedir. En yaygın kullanılan teknik en önemsiz bite gömmedir [57].

3. ÖNERİLEN YÖNTEM

Bu şifreleme algoritmasında Taylor serisi kullanılarak, ilk olarak sekiz karakter uzunluğunda bir anahtar ile mesaj, bir dizi kurallara tabi tutularak şifrelenir. Daha sonra elde edilen veriler şifrelenmiş mesajı alacak kişiye özel bir kod ile sesin en anlamsız bit ’ine gizleme yapılarak gönderilir. Şekil 3.1’de şifreleme diyagramı, Şekil 3.2’de deşifreleme diyagramı gösterilmiştir.



Şekil 3.1. Şifreleme diyagramı



Şekil 3.2. Deşifreleme diyagramı

şeklindedir.

3.1. Şifreleme Algoritması

Tanım 3.1.

t>0 için f (t) tanımlansın. $|f(t)| \leq Me^{\propto t}$ olması durumunda eğer α, *M>0* sayıları varsa, f üstel sıradadır yazılır. (1)

Eğer *f (t)* üstel fonksiyonsa, o halde $t \rightarrow \infty$ *için f (t)* $= \infty$ yazılır [59].

Tanım 3.2.

t≥0 için f (t) verilsin ve fonksiyonun α, üstel sıra özelliğini sağladığını varsayalım ve t, s ∈R. f (t)'nin Laplace dönüşümü olarak tanımlanır [57].

$$F(s) = \int_0^\infty e^{-st} f(t)dt \qquad (3.1)$$

Tanım 1 ve 2'yi kullanarak Laplace dönüşümünü genişletip yeni bir dönüşüm fonksiyonu tanımlayalım.

Tanım 3.3.

Her *t≥0 için f (t)* dönüşümü şu şekilde tanımlanır.

$$H(h) = T[f(t)]$$
$$= \int_0^{\infty} \frac{1}{h} e^{\frac{-t}{h}} f(t)dt. \qquad (3.2)$$

(Genişletilmiş Kuvvet Serisi Dönüşümü) f (t)'nin ters dönüşümünü tanımlamak için $f^{-1}(t) = T^{-1}[F(h)]$ yazılır. Elde edilen bu genişletilmiş güç serisi dönüşümü olan aşağıdaki denklemler standart sonuçlara sahiptir [59].

1. $T\{t^n\} = \frac{n!}{s^{n+1}} \Rightarrow T^{-1}\left\{\frac{1}{s^{n+1}}\right\} = \frac{t^n}{n!}$
2. $T\{t^n e^{st}\} = \frac{n!.h^n}{(1-sh)^{n+1}} \Rightarrow \quad T^{-1}\{\frac{h^n}{(1-sh)^{n+1}}\} = \frac{t^n.e^{st}}{n!} \quad (t\geq 0) \qquad (3.3)$

Taylor serisi e^t ile genişletilir. Matematiksel ilişkiyi genellemek için bu değer t^3 ile çoğaltılırak şifreleme algoritmasında kullanılır.

Şifreleme algoritmasında ilk olarak mesaj, anahtarla bir dizi kurallar kullanılarak şifreleme işlemi yapıldıktan sonra, elde edilen veriler, ses steganografisi ile gizleme işlemine tabi tutulur. Şifreleme algoritmasının çalışma prensibi aşağıdaki gibidir:

- Şifrelenecek mesaj, sekizli bloklar halinde anahtarla işleme alınır.
- Kullanılacak olan anahtar sekiz karakterli olmak zorundadır.
- Şifrelemede Taylor serisi kullanılmış olup, hesaplamalarda mod 256 kullanılmıştır.
- Anahtarın tümü ve mesaj sekizli bloklar halinde Binary'e dönüştürülerek işleme tabi tutulur.

1. Adım:

Anahtarın 1. karakteri ve Mesajın 1. karakteri binary'e dönüştürüldükten sonra;

a) Eğer anahtarın 1. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 1. karakteri olan en son bit, en başa alınır (a_1).
- Mesajın 1. karakteri olan en baştaki bit, en sona alınır.

b) Eğer anahtarın 1. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 1. karakteri olan baştaki bit, en sona alınır (a_1).
- Mesajın 1. karakteri olan en sondaki bit, en başa alınır (m_1).

➢ Anahtarın 1. karakterin dönüşüme uğramış hali (a_1) ile mesajın 1. karakterin dönüşüme uğramış hali (m_1) XOR'lanarak mesajın ilk karakteri 1. adımda şifrelenir (x_1).

2. Adım:

Anahtarın 2. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 2. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 2. karakteri olan en son bit, en başa alınır (a_2).
- Mesajın 1. karakteri olan, 1. adımın sonunda şifrelenmiş değerin (x_1) en baştaki biti, en sona alınır (m_2).

b) Eğer anahtarın 2. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 2. karakteri olan baştaki bit, en sona alınır (a_2).
- Mesajın 1. karakteri olan, 1. adımın sonunda şifrelenmiş değerin (x_1) en sondaki biti, en başa alınır (m_2).

➢ Anahtarın 2. karakterin dönüşüme uğramış hali (a_2) ile mesajın 2. adımda dönüşüme uğramış hali (m_2) XOR'lanarak mesajın 1. karakteri 2.adımda şifrelenir (x_2).

3. Adım:

Anahtarın 3. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 3. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 3. karakteri olan en son bit, en başa alınır (a_3).
- Mesajın 1. karakteri olan, 2. adımın sonunda şifrelenmiş değerin (x_2) en baştaki biti, en sona alınır (m_3).

b) Eğer anahtarın 3. karakterindeki 0 sayısının adedi, 1 sayısın adedinden az ise;

- Anahtarın 3. karakteri olan baştaki bit, en sona alınır (a_3).

- Mesajın 1. karakteri olan, 2. adımın sonunda şifrelenmiş değerin (x_2) en sondaki biti, en başa alınır (m_3).

➢ Anahtarın 3. karakterin dönüşüme uğramış hali (a_3) ile mesajın 3.adımda dönüşüme uğramış hali (m_3) XOR'lanarak mesajın 1. karakteri 3. adımda şifrelenir (x_3).

4. Adım:

Anahtarın 4. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 4. karakterindeki 0 sayısı adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 4. karakteri olan en son bit, en başa alınır (a_4).
- Mesajın 1. karakteri olan, 3. adımın sonunda şifrelenmiş değerin (x_3) en baştaki biti, en sona alınır (m_4).

b) Eğer anahtarın 4. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 4. karakteri olan baştaki bit, en sona alınır (a_4).
- Mesajın 1. karakteri olan, 3. adımın sonunda şifrelenmiş değerin (x_3) en sondaki biti, en başa alınır (m_4).

➢ Anahtarın 4. karakterin dönüşüme uğramış hali (a_4) ile mesajın 4. adımda dönüşüme uğramış hali (m_4) XOR'lanarak mesajın ilk karakteri 4. adımda şifrelenir (x_4).

5. Adım;

Anahtarın 5. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 5. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 5. karakteri olan en son bit, en başa alınır (a_5).
- Mesajın 1. karakteri olan, 4. adımın sonunda şifrelenmiş değerin (x_4) en baştaki biti, en sona alınır (m_5).

b) Eğer anahtarın 5. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 5. karakteri olan baştaki bit, en sona alınır (a_5).
- Mesajın 1. karakteri olan, 4. adımın sonunda şifrelenmiş değerin (x_4) en sondaki biti, en başa alınır (m_5).

➢ Anahtarın 5. karakterin dönüşüme uğramış hali (a_5) ile mesajın 5. adımda dönüşüme uğramış hali (m_5) XOR'lanarak mesajın ilk karakteri 5. adımda şifrelenir (x_5).

6. Adım;

Anahtarın 6. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 6.karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 6. karakteri olan en son bit, en başa alınır (a_6).
- Mesajın 1. karakteri olan, 5. adımın sonunda şifrelenmiş değerin (x_5) en baştaki biti, en sona alınır (m_6).

b) Eğer anahtarın 6. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 6. karakteri olan baştaki bit, en sona alınır (a_6).
- Mesajın 1. karakteri olan, 5. adımın sonunda şifrelenmiş değerin (x_5) en sondaki biti, en başa alınır (m_6).

➢ Anahtarın 6. karakterin dönüşüme uğramış hali (a_6) ile mesajın 6. adımda dönüşüme uğramış hali (m_6) ile XOR'lanarak mesajın ilk karakteri 6. adımda şifrelenir (x_6).

7. Adım;

Anahtarın 7. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 7. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 7. karakteri olan en son bit, en başa alınır (a_7).
- Mesajın 1. karakteri olan, 6. adımın sonunda şifrelenmiş değerin (x_6) en baştaki biti, en sona alınır (m_7).

b) Eğer anahtarın 7. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 7. karakteri olan baştaki bit, en sona alınır (a_7).
- Mesajın 1. karakteri olan, 6. adımın sonunda şifrelenmiş değerin (x_6) en sondaki biti, en başa alınır (m_7).

➢ Anahtarın 7. karakterin dönüşüme uğramış hali (a_7) ile mesajın 7. adımda dönüşüme uğramış hali (m_7) ile XOR'lanarak mesajın 1.karakteri 7. adımda şifrelenir (x_7).

8. Adım;

Anahtarın 8. karakteri binarye dönüştürüldükten sonra;

a) Eğer anahtarın 8. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 8. karakteri olan en son bit, en başa alınır (a_8).
- Mesajın 1. karakteri olan, 7. adımın sonunda şifrelenmiş değerin (x_7) en baştaki biti, en sona alınır (m_8).

b) Eğer anahtarın 8. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 8. karakteri olan baştaki bit, en sona alınır (a_8).
- Mesajın 1. karakteri olan, 7. adımın sonunda şifrelenmiş değerin (x_7) en sondaki biti, en başa alınır (m_8).

➢ Anahtarın 8. karakterin dönüşüme uğramış hali (a_8) ile mesajın 8. adımda dönüşüme uğramış hali (m_8) ile XOR'lanarak mesajın 1.karakteri 8. adımda şifrelenir (x_8).

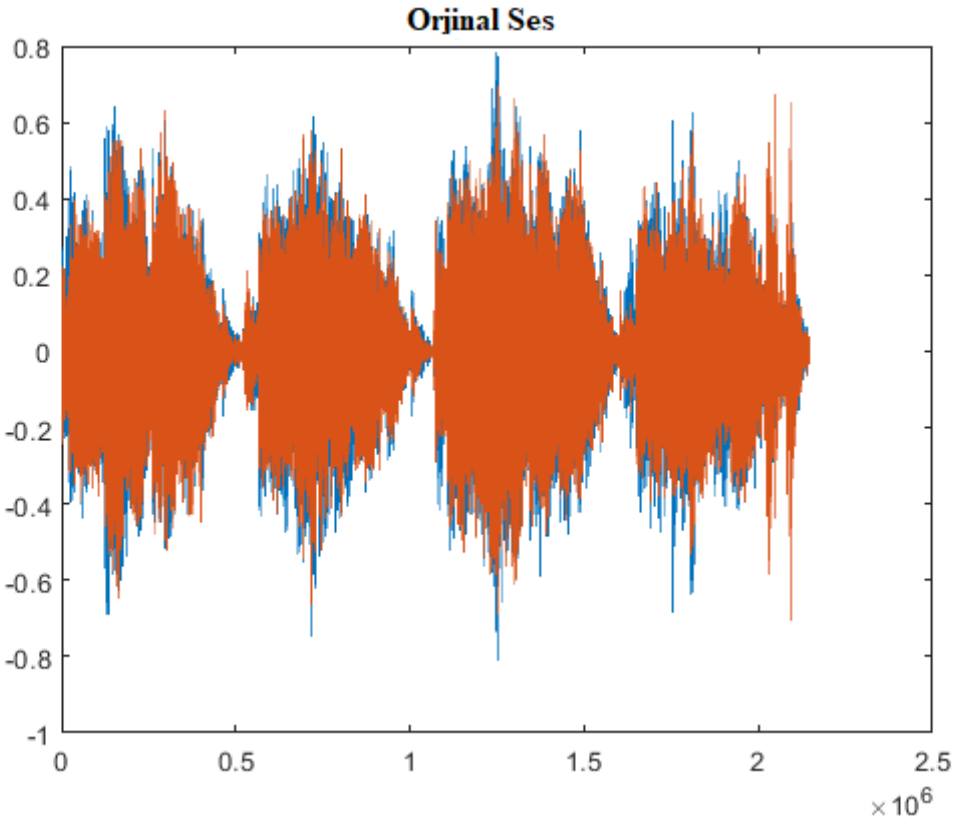
➢ 8. adım sonunda mesajın 1. karakterinde şifrelenme işlemi tamamlanmış olur. Mesajın 2-3-4-5-6-7-8 karakterleri bu sekiz adım şifrelemeye başlamadan önce kendisinden bir önceki karakterin şifrelenen hali ile XOR'lama işlemi yapılmaktadır. Daha sonra 8 adım aynı şekilde devam edilmektedir. Mesaj şifreleme sekizli bloklar halinde şifrelenmektedir.

➢ Her adımın sonunda elde edilen değerin, daha önceden rastgele oluşturulan Tablo 3.1'de belirtilen karşılığına denk gelen değeri (ilk 4 sayıyı satır, son 4 sayıyı sütun olarak) almaktadır.

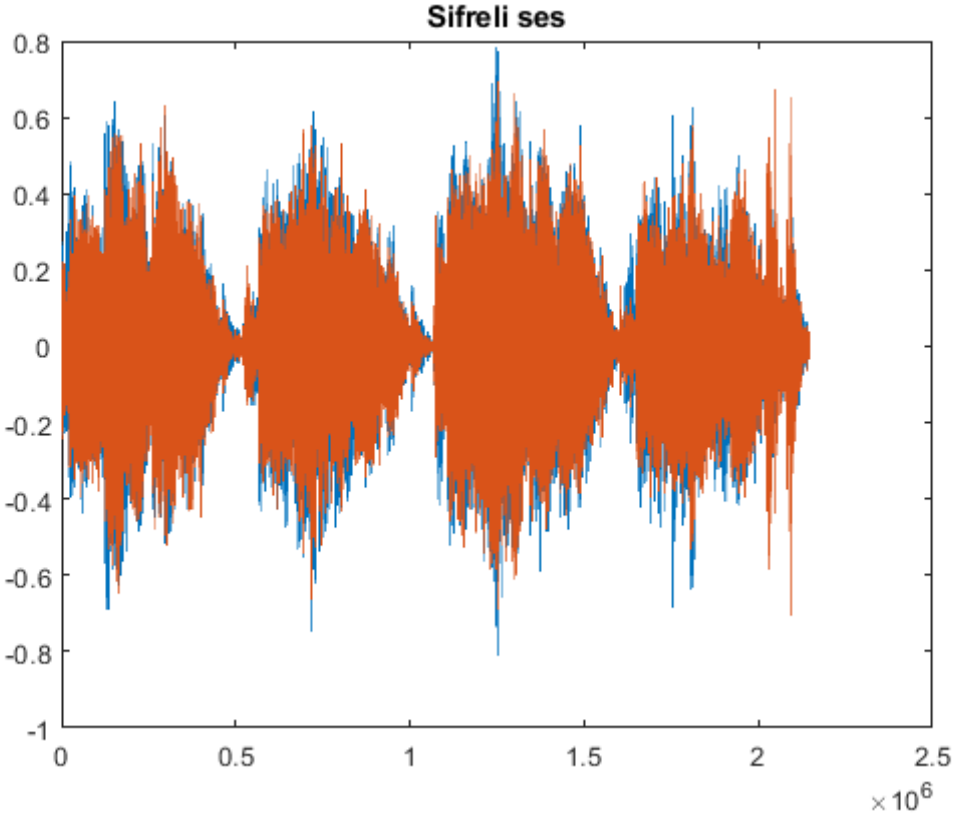
Tablo 3.1 S-box listesi

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	181	120	169	38	245	76	242	230	39	72	78	47	33	239	249
0001	49	7	37	64	68	160	237	32	58	48	14	203	35	114	110	143
0010	119	23	6	12	220	109	44	61	215	202	159	45	29	250	157	235
0011	24	184	170	22	150	28	149	133	129	198	219	13	145	56	63	204
0100	212	97	201	5	15	177	234	122	50	0	113	102	253	106	36	168
0101	192	65	57	104	226	3	174	101	84	151	42	128	140	60	224	112
0110	207	53	46	95	131	243	87	118	175	164	69	55	178	247	79	126
0111	158	134	217	229	139	73	93	30	254	92	142	59	27	52	248	153
1000	240	121	189	196	138	165	130	228	11	144	34	147	25	194	137	100
1001	180	135	99	222	156	241	161	208	18	20	80	197	67	105	244	124
1010	221	74	211	167	85	115	183	251	111	51	16	108	200	233	205	66
1011	41	152	214	125	163	31	86	62	155	166	176	26	75	21	188	172
1100	232	96	4	216	238	54	107	210	171	9	195	103	8	88	141	10
1101	117	2	91	123	154	43	191	236	162	116	185	81	127	19	173	193
1110	82	252	246	83	190	187	186	136	223	71	70	218	182	225	89	146
1111	40	209	98	179	255	132	17	199	77	231	213	206	227	90	94	148

- Algoritmada kullanılacak olan Laplace dönüşümü için şifreleme algoritmasında Taylor serisi kullanılmıştır.
- Mod 256 alınmıştır.
- Daha önceden taraflar anlaşmalı olarak mesajı alacak kişiye özel, standart bir değeri (T.C. kimlik numarası, sicil numarası vb.), *.txt uzantılı bir dosyanın içine kayıt edip, bu dosyayı ikinci anahtar olarak kullanırlar. Ses Steganografisi'nin LSB (LeastSignificant Bit) en anlamsız bit tekniğini kullanarak herhangi bir ses dosyasının içine, elde edilen bölüm - kalan değerlerini gizlerler. Gizlenen veri, ses dosyasının en anlamsız bit'inde gizlendiği için, değiştirilen ses kulakla duyulabilecek bir frekans özelliği göstermeyecektir. Ayrıca ses frekansında herhangi bir değişiklik gözlenmeyecektir. Şekil 3.3 ve 3.4'de orijinal ses ve şifreli ses görülmektedir.



Şekil 3.3. Orijinal ses



Şekil 3.4. Şifreli ses

3.2. Şifre Çözme Algoritması

Şifre çözme işlemi, şifreleme algoritmasındaki adımların ters yönde uygulanması yöntemi ile şifrenin çözülmesidir. Şifre çözme algoritmasının çalışma prensibinin kuralları aşağıdaki gibidir:

- Şifreli mesajın çözümü için, şifreli mesaj sekizli bloklar halinde anahtarla işleme tabi tutulmaktadır.
- Anahtarın tümü, şifreli mesajı sekizli bloklar halinde Binary'e dönüştürerek uygulanmaktadır.
- Ses Steganografisi'nin LSB tekniği kullanarak, gizlenen bölüm - kalan değerlerine ulaşmak için, taraflar arasında önceden mutabık kalınan standart değer *.txt uzantılı dosyanın içine kayıt edilir ve bu dosyaya ikinci anahtar girilir.
- $A_n = \frac{K_n - K'_n}{256}$

$x_0 = 256 * bölüm_0 + kalan_0$

$x_1 = 256 * bölüm_1 + kalan_1$

$$x_2 = 256 * bölüm_2 + kalan_2$$
$$x_3 = 256 * bölüm_3 + kalan_3$$
$$x_4 = 256 * bölüm_4 + kalan_4$$
$$x_5 = 256 * bölüm_5 + kalan_5$$
$$x_6 = 256 * bölüm_6 + kalan_6$$
$$x_7 = 256 * bölüm_7 + kalan_7$$

$$\sum_{n=0}^{\infty} K_n(n+3)!\frac{h^{n+3}}{n!} \tag{3.4}$$

$$\sum_{n=0}^{\infty} K_n(n+3)!\frac{h^{n+3}}{n!} = x_0h^3 + x_1h^4 + x_2h^5 + x_3h^6 + x_4h^7 + x_5h^8 +$$

$$x_6h^9 + x_7h^{10} \tag{3.5}$$

Yukarıdaki Taylor denklemi sonucunda elde edilen her bir değer için Tablo 3.2'de karşılıkları bulunur.

Tablo 3.2. S-box listesi

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	181	120	169	38	245	76	242	230	39	72	78	47	33	239	249
0001	49	7	37	64	68	160	237	32	58	48	14	203	35	114	110	143
0010	119	23	6	12	220	109	44	61	215	202	159	45	29	250	157	235
0011	24	184	170	22	150	28	149	133	129	198	219	13	145	56	63	204
0100	212	97	201	5	15	177	234	122	50	0	113	102	253	106	36	168
0101	192	65	57	104	226	3	174	101	84	151	42	128	140	60	224	112
0110	207	53	46	95	131	243	87	118	175	164	69	55	178	247	79	126
0111	158	134	217	229	139	73	93	30	254	92	142	59	27	52	248	153
1000	240	121	189	196	138	165	130	228	11	144	34	147	25	194	137	100
1001	180	135	99	222	156	241	161	208	18	20	80	197	67	105	244	124
1010	221	74	211	167	85	115	183	251	111	51	16	108	200	233	205	66
1011	41	152	214	125	163	31	86	62	155	166	176	26	75	21	188	172
1100	232	96	4	216	238	54	107	210	171	9	195	103	8	88	141	10
1101	117	2	91	123	154	43	191	236	162	116	185	81	127	19	173	193
1110	82	252	246	83	190	187	186	136	223	71	70	218	182	225	89	146
1111	40	209	98	179	255	132	17	199	77	231	213	206	227	90	94	148

1. Adım:

Anahtarın 8. karakteri binarye dönüştürüldükten (a_8) sonra, Tablo 3.2'de gelen değer (m_x) ile işleme tabi tutulmaktadır;

a) Eğer anahtarın 8. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 8. karakteri olan en sondaki bit, en başa alınır (a_8).
- Anahtar ile Tablo 3.1'de gelen 1. değeri XOR'lama işlemi yapılır (m_8).
- Anahtarın 8. karakterin dönüşüme uğramış hali (a_8) ile Tablo 3.2'de gelen 1. değeri dönüşüme uğramış hali (m_8) XOR'landıktan sonra gelen değerin en son biti, en başa alınarak mesajın 1. karakteri 1. adım şifre çözümlemesi yapılır.

b) Eğer anahtarın 8. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 8. karakterin en baştaki biti, en sona al (a_8).
- Anahtar ile Tablo 3.1'de gelen 1. değeri XOR lama işlemi yapılır (m_8).
- Anahtarın 8.karakterin dönüşüme uğramış hali (a_8) ile Tablo 3.2'de gelen 1.değeri dönüşüme uğramış hali (m_1) ile XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1.karakteri 1. adım şifre çözümlemesi yapılmaktadır (m_7).

2. Adım:

Anahtarın 7. karakteri binarye dönüştürüldükten (a_7) sonra, 1. adım sonunda elde edilen veri (m_7) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 7. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 7. karakterin en son biti, en başa alınır (a_7).
- Anahtar ile 1. adım sonunda elde edilen veriyle XOR'lama işlemi yapılır (m_6).
- Anahtarın 7. karakterin dönüşüme uğramış hali (a_7) ile 1. adım sonunda elde verinin dönüşüme uğramış hali (m_7) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1.karakteri 2.adım şifre çözümlemesi yapılmaktadır (m_6).

b) Eğer anahtarın 7. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 7. karakterin en baştaki biti, en sona alınır (a_8).
- Anahtar ile 1. adım sonunda elde veriyle XOR lama işlemi yapılır (m_6).
- Anahtarın 7. karakterin dönüşüme uğramış hali (a_7) ile 1. adım sonunda elde verinin dönüşüme uğramış hali (m_7) XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 2. adım şifre çözümlemesi yapılmaktadır (m_6).

3. Adım:

Anahtarın 6. karakteri binarye dönüştürüldükten (a_6) sonra, 2. adım sonunda elde edilen veri (m_6) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 6. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;
   - Anahtarın 6. karakterin en son biti, en başa alınır (a_6).
   - Anahtar ile 2. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_5).
   - Anahtarın 6. karakterin dönüşüme uğramış hali (a_6) ile 2. adım sonunda elde verinin dönüşüme uğramış hali (m_6) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1.karakteri 3. adım şifre çözümlemesi yapılmaktadır (m_5).

b) Eğer anahtarın 6. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;
   - Anahtarın 6. karakterin en baştaki biti, en sona alınır (a_6).
   - Anahtar ile 2. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_5).
   - Anahtarın 6. karakterin dönüşüme uğramış hali (a_6) ile 2. adım sonunda elde verinin dönüşüme uğramış hali (m_5) XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 3. adım şifre çözümlemesi yapılmaktadır (m_5).

4. Adım:

Anahtarın 5. karakteri binarye dönüştürüldükten (a_5) sonra, 3. adım sonunda elde edilen veri (m_5) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 5. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;
   - Anahtarın 5. karakterin en son biti, en başa alınır (a_5).
   - Anahtar ile 3. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_4).

- Anahtarın 5. karakterin dönüşüme uğramış hali (a_5) ile 3. adım sonunda elde verinin dönüşüme uğramış hali (m_5) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1. karakteri 4. adım şifre çözümlemesi yapılmaktadır (m_4).

b) Eğer anahtarın 5. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 5. karakterin en baştaki biti, en sona alınır (a_5).
- Anahtar ile 3. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_4).
- Anahtarın 5. karakterin dönüşüme uğramış hali (a_5) ile 3. adım sonunda elde verinin dönüşüme uğramış hali (m_5), XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 4. adım şifre çözümlemesi yapılmaktadır (m_4).

5. Adım:

Anahtarın 4. karakteri binarye dönüştürüldükten (a_4) sonra, 4. adım sonunda elde edilen veri (m_4) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 4. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;

- Anahtarın 4. karakterin en son biti, en başa alınır (a_4).
- Anahtar ile 4. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_3).
- Anahtarın 4. karakterin dönüşüme uğramış hali (a_4) ile 4. adım sonunda elde verinin dönüşüme uğramış hali (m_4) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1. karakteri 5.adım şifre çözümlemesi yapılmaktadır (m_3).

b) Eğer anahtarın 4. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 4. karakterin en baştaki biti, en sona alınır (a_4).
- Anahtar ile 4. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_3).
- Anahtarın 4. karakterin dönüşüme uğramış hali (a_4) ile 4. adım sonunda elde verinin dönüşüme uğramış hali (m_4) XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 5. adım şifre çözümlemesi yapılmaktadır (m_3).

6. Adım:

Anahtarın 3. karakteri binarye dönüştürüldükten (a_3) sonra, 5. adım sonunda elde edilen veri (m_3) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 3. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;
   - Anahtarın 3. karakterin en son biti, en başa alınır (a_3).
   - Anahtar ile 3. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_2).
   - Anahtarın 3. karakterin dönüşüme uğramış hali (a_3) ile 5.adım sonunda elde verinin dönüşüme uğramış hali (m_3) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1.karakteri 6.adım şifre çözümlemesi yapılmaktadır (m_2).

b) Eğer anahtarın 3. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;
   - Anahtarın 3. karakterin en baştaki biti, en sona alınır (a_3).
   - Anahtar ile 5. adım sonunda elde veriyle XOR'lama işlemi yapılır (m_2).
   - Anahtarın 3. karakterin dönüşüme uğramış hali (a_3) ile 5. adım sonunda elde verinin dönüşüme uğramış hali (m_3) XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 6. adım şifre çözümlemesi yapılmaktadır (m_2).

7. Adım:

Anahtarın 2. karakteri binarye dönüştürüldükten (a_2) sonra, 6. adım sonunda elde edilen veri (m_2) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 2. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;
   - Anahtarın 2. karakterin en son biti, en başa alınır (a_2).
   - Anahtar ile 6. adım sonunda elde veriyle edilen XOR'lama işlemi yapılır (m_1).
   - Anahtarın 2. karakterin dönüşüme uğramış hali (a_2) ile 6. adım sonunda elde verinin dönüşüme uğramış hali (m_2) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1. karakteri 7. adım şifre çözümlemesi yapılmaktadır (m_1).

b) Eğer anahtarın 2. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;

- Anahtarın 2. karakterin en baştaki biti, en sona alınır (a_2).
- Anahtar ile 6. adım sonunda elde edilen veriyle XOR'lama işlemi yapılır (m_1).
- Anahtarın 2. karakterin dönüşüme uğramış hali (a_2) ile 6. adım sonunda elde verinin dönüşüme uğramış hali (m_2) XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 7. adım şifre çözümlemesi yapılmaktadır (m_1).

8. Adım:

Anahtarın 1. karakteri binarye dönüştürüldükten (a_1) sonra, 7. adım sonunda elde edilen veri (m_1) ile işleme tabi tutulmaktadır.

a) Eğer anahtarın 1. karakterindeki 0 sayısının adedi, 1 sayısının adedinden fazla veya eşit ise;
   - Anahtarın 1. karakterin en son biti, en başa alınır (a_1).
   - Anahtar ile 7. adım sonunda elde veriyle edilen XOR'lama işlemi yapılır (m).
   - Anahtarın 1. karakterin dönüşüme uğramış hali (a_1) ile 7. adım sonunda elde verinin dönüşüme uğramış hali (m_1) XOR'landıktan sonra gelen değerin en sondaki biti, en başa alınarak mesajın 1. karakteri 8. adım şifre çözümlemesi sonucunda mesajın 1. karakterine ulaşılmaktadır (m).

b) Eğer anahtarın 1. karakterindeki 0 sayısının adedi, 1 sayısının adedinden az ise;
   - Anahtarın 1. karakterin en baştaki biti, en sona alınır (a_1).
   - Anahtar ile 7. adım sonunda elde edilen veriyle XOR'lama işlemi yapılır (m).
   - Anahtarın 1. karakterin dönüşüme uğramış hali (a_1) ile 7. adım sonunda elde verinin dönüşüme uğramış hali (m_1) XOR'landıktan sonra gelen değerin en baştaki biti, en sona alınarak mesajın 1. karakteri 8. adım şifre çözümlemesi sonucunda mesajın 1. karakterine ulaşılmaktadır (m).

Mesajın 1. karakterine ulaşıldıktan sonra 2-3-4-5-6-7-8. karakterlerini yukarıdaki 8 adım uygulamadan önce kendisinden bir önceki karakterin Tablo 3.1'de ulaşılan değerle XOR'lama işlemi yapıldıktan sonra 8 adım uygulanırsa mesajın sekizli blokları şifre çözümüne ulaşılmaktadır.

4. UYGULAMA

Yapacağımız uygulamada, kullanılan algoritmada belirtilen kurallara göre ilk olarak veri şifreleme işlemi yapılacak sonra ise şifrelenen veri ses dosyasında gizleme yapılacaktır. Şifre çözme ise ses içerisinde şifreli mesaj çıkarılacak ve adımlar halinde şifreli mesaj çözülecektir.

4.1. Şifreleme

Anahtar = F1r@tb3y Mesaj = Steganog

F=01000110	1=00110001	r =01110010	@=01000000
t=01110100	b=01100010	3=00110011	y=01111001
S=01010011	t=01110100	e=01100101	g=01100111
a=01100001	n=01101110	o=01101111	g=01100111

Uygulamada anahtar olarak kullanılan karakterlerin 0 ve 1 sayıların adedi Tablo 4.1'de gösterilmiştir.

Tablo 4.1. Anahtardaki 0 ve 1 sayılarının adetleri

		0 sayısı	1 sayısı
F	01000110	5	3
1	00110001	5	3
r	01110010	4	4
@	01000000	7	1
t	01110100	4	4
b	01100010	5	3
3	00110011	4	4
y	01111001	3	5

1. Adım:

F= 01000110 S=01010011 F_X=00100011

XOR m_1=10000101

F_X=00100011 S_X= 10100110 S_X=10100110

1 =00110001 m_1=10000101 1_X =10011000

XOR m_2 =10010011

1_X=10011000 m_X=00001011 m_X=00001011

r=01110010 m_2=10010011 r_X=00111001

XOR m_3= 00011110

r_X=00111001 m_X=00100111 m_X=00100111

@ =01000000 m_3=00011110 $@_X$= 00111100

XOR m_4= 00011100

$@_X$= 00100000 m_X= 00111100 m_X= 0100 1110

t=01110100 m_4= 00011100 t_X=1110 1000

XOR m_5=00000010

t_X=00111010 m_X=00111000 m_X=00111000

b = 01100010 m_5=00000010 b_X=00110001

XOR m_6=00110101

b_X=00110001 m_X=00000100 m_X=00000100

3= 00110011 m_6= 00110101 3_X=10011001

XOR m_7=11110011

3_X=10011001 m_X= 01101010 m_X= 01101010

y=01111001 m_7= 11110011 y_X=11110010

XOR m_8=00001011 (1. karakter)

y_X=11110010 m_X=11111001 m_X=11111001

2. Adım: Çıkan sonuç ile mesajın 2. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

t =01110100

XOR t_x= 01111111

m_8=00001011

F =01000110 t_x= 01111111 F_X=00100011

XOR m_1=11011101

F_X=00100011 m_X= 11111110 m_X= 11111110

1 =00110001 m_1=11011101 1_X = 10011000

XOR m_2 =00100011

1_X=10011000 m_X=10111011 m_X=10111011

r=01110010 m_2 = 00100011 r_X=00111001

XOR m_3=01111111

r_X=00111001 m_X=01000110 m_X= 01000110

@ =01000000 m_3= 01111111 $@_X$= 00100000

XOR m_4= 11011110

$@_X$= 00100000 m_X= 11111110 m_X= 11111110

t=01110100 m_4= 11011110 t_X=00111010

XOR m_5=10000111

t_X=00111010 m_X=10111101 m_X= 10111101

b =01100010 m_5=10000111 b_X=00110001

XOR m_6=00111110

b_X=00110001 m_X=00001111 m_X=00001111

3=00110011 m_6=00111110 3_X=10011001

XOR m_7=11100101

3_X=10011001 m_X= 01111100 m_X= 01111100

y=01111001 m_7=11100101 y_X=11110010

XOR m_8=00000000 (2. karakter)

y_X=11110010 m_X=11110010 m_X=11110010

3. Adım: Çıkan sonuç ile mesajın 3. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

e =01100101

XOR e_x=01100101

m_8=00000000

F =01000110 e_x= 01100101 F_X=00100011

XOR m_1=11101001

F_X=00100011 m_X= 11001010 m_X= 11001010

1 =00110001 m_1= 11101001 1_X = 10011000

XOR m_2 = 01001011

1_X=10011000 m_X=11010011 m_X=11010011

r=01110010 m_2 = 01001011 r_X=00111001

XOR m_3= 10101111

r_X=00111001 m_X=10010110 m_X=10010110

@ =01000000 m_3= 10101111 $@_X$= 1000 0000

XOR m_4= 01111111

$@_X$= 00100000 m_X= 01011111 m_X= 01011111

t=01110100 m_4= 01111111 t_X=00111010

XOR m_5=11000100

t_X=00111010 m_X= 11111110 m_X= 11111110

b =01100010 m_5=11000100 b_X=00110001

XOR m_6=10111000

b_X=00110001 m_X=10001001 m_X=10001001

3=00110011 m_6=10111000 3_X=10011001

XOR m_7=11101000

3_X=10011001 m_X=01110001 m_X=01110001

y=01111001 m_7=11101000 y_X=11110010

XOR m_8=10000110 (3. karakter)

y_X=11110010 m_X=01110100 m_X=01110100

Çıkan sonuç ile mesajın 4. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

g =01100111

XOR g_x=11100001

m_8=10000110

F =01000110 g_x= 11100001 F_X=00100011

XOR m_1=11100000

F_X=00100011 m_X= 11000011 m_X= 11000011

1 =00110001 m_1=11100000 1_X = 10011000

XOR m_2 =01011001

1_X=10011000 m_X=11000001 m_X=11000001

r=01110010 m_2 = 01011001 r_X=00111001

XOR m_3=10001011

r_X=00111001 m_X=10110010 m_X=10110010

@ =01000000 m_3= 10001011 $@_X$= 1000 0000

XOR m_4= 00110111

$@_X$= 00100000 m_X= 00010111 m_X= 00010111

t=01110100 m_4= 00110111 t_X=00111010

XOR m_5=01010100

t_X=00111010 m_X=01101110 m_X= 01101110

b =01100010 m_5=01010100 b_X=00110001

XOR m_6=10011001

b_X=00110001 m_X=10101000 m_X=10101000

3=00110011 m_6=10011001 3_X=10011001

XOR m_7=10101010

3_X=10011001 m_X=00110011 m_X=00110011

y=01111001 m_7=10101010 y_X=11110010

XOR m_8=10100111 (4. karakter)

y_X=11110010 m_X=01010101 m_X=01010101

Çıkan sonuç ile mesajın 5. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

a=01100001

XOR a_x=11000110

m_8=10100111

F =01000110 a_x= 11000110 F_X=00100011

XOR m_1=10101110

F_X=00100011 m_X= 10001101 m_X= 10001101

1 =00110001 m_1=0101110 1_X = 10011000

XOR m_2 = 11000101

1_X=10011000 m_X=01011101 m_X=01011101

r=01110010 m_2 = 11000101 r_X=00111001

XOR m_3= 10110010

r_X=00111001 m_X=10001011 m_X=10001011

@ =01000000 m_3= 10110010 $@_X$= 1000 0000

XOR m_4= 01000101

$@_X$= 00100000 m_X= 01100101 m_X= 01100101

t=01110100 m_4= 01000101 t_X=00111010

XOR m_5=10110000

t_X=00111010 m_X=10001010 m_X=10001010

b =01100010 m_5=10110000 b_X=00110001

XOR m_6=01010000

b_X=00110001 m_X=01100001 m_X=01100001

3=00110011 m_6=01010000 3_X=10011001

XOR m_7=00111001

3_X=10011001 m_X=10100000 m_X=10100000

y=01111001 m_7=00111001 y_X=11110010

XOR m_8=01101110 (5. karakter)

y_X=11110010 m_X=10011100 m_X=10011100

Çıkan sonuç ile mesajın 6. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

n =01101110

XOR n_x= 00000000

m_8=01101110

F =01000110 n_x= 00000000 F_X=00100011

XOR m_1=00100011

F_X=00100011 m_X= 00000000 m_X= 00000000

1 =00110001 m_1=00100011 1_X = 10011000

XOR m_2 = 11011110

1_X=10011000 m_X=01000110 m_X=01000110

r=01110010 m_2 = 11011110 r_X=00111001

XOR m_3=10000100

r_X=00111001 m_X= 10111101 m_X= 10111101

@ =01000000 m_3= 10000100 $@_X$= 1000 0000

XOR m_4= 00101001

$@_X$= 00100000 m_X= 00001001 m_X= 00001001

t=01110100 m_4= 00101001 t_X=0011 1010

XOR m_5=01101000

t_X=00111010 m_X=01010010 m_X=01010010

b =01100010 m_5=01101000 b_X=00110001

XOR m_6=11100001

b_X=00110001 m_X=11010000 m_X=11010000

3=00110011 m_6=11100001 3_X=10011001

XOR m_7=01011010

3_X=10011001 m_X=11000011 m_X=11000011

y=01111001 m_7=01011010 y_X=11110010

XOR m_8=11011111 (6. karakter)

y_X=11110010 m_X=00101101 m_X=00101101

Çıkan sonuç ile mesajın 7. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

o=01101111

XOR o_x=10110000

m_8=11011111

F =01000110 o_x= 11011111 F_X=00100011

XOR m_1=01000010

F_X=00100011 m_X= 01100001 m_X= 01100001

1 =00110001 m_1=01000010 1_X = 10011000

XOR m_2 =00011100

1_X=10011000 m_X=10000100 m_X=10000100

r=01110010 m_2 = 00011100 r_X=00111001

XOR m_3=00000001

r_X=00111001 m_X=00111000 m_X= 00111000

@ =01000000 m_3= 00000001 $@_X$= 00100000

XOR m_4= 00100010

$@_X$=00100000 m_X= 00000010 m_X= 00000010

t=01110100 m_4= 00100010 t_X=00111010

XOR m_5=01111110

t_X=00111010 m_X=01000100 m_X=01000100

b =01100010 m_5=01111110 b_X=00110001

XOR m_6=11001101

b_X=00110001 m_X=11111100 m_X=11111100

3=00110011 m_6=11001101 3_X=10011001

XOR m_7=00000010

3_X=10011001 m_X=10011011 m_X=10011011

y=01111001 m_7=00000010 y_X=11110010

XOR m_8=11110011 (7. karakter)

y_X=11110010 m_X=00000001 m_X=00000001

Çıkan sonuç ile mesajın 8. karakteri xor'landıktan sonra tekrardan anahtar ile işleme tabi tutulur.

g=01100111

XOR g_x=10010100

m_8=11110011

F =01000110 g_x= 10010100 F_X=00100011

XOR m_1=00001010

F_X=00100011 m_X=00101001 m_X= 00101001

1 =00110001 m_1=00001010 1_X = 10011000

XOR m_2 =10001100

1_X=10011000 m_X=00010100 m_X=00010100

r=01110010 m_2 =10001100 r_X= 00111001

XOR m_3=00100000

r_X=00111001 m_X=00011001 m_X= 00011001

@ =01000000 m_3=00100000 $@_X$= 00100000

XOR m_4= 01100000

$@_X$= 00100000 m_X= 01000000 m_X= 01000000

t=01110100 m_4= 01100000 t_X=00111010

XOR m_5=11111010

t_X=00111010 m_X=11000000 m_X=11000000

b =01100010 m_5=11111010 b_X=00110001

XOR m_6=11000100

b_X=00110001 m_X=11110101 m_X=11110101

3=00110011 m_6=11000100 3_X=10011001

XOR m_7=00010000

3_X=10011001 m_X=10001001 m_X=10001001

y=01111001 m_7=00010000 y_X=11110010

XOR m_8=11111010 (8. karakter)

y_X=11110010 m_X=00001000 m_X=00001000

4. Adım: Çıkan sonucu da daha önceden rastgele oluşturulan Tablo 4.2'de belirtilen karşılığını bul (ilk 4 sayıyı satır, son 4 sayıyı sütun olarak alınır).

Tablo 4.2. S-box listesi

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	181	120	169	38	245	76	242	230	39	72	78	47	33	239	249
0001	49	7	37	64	68	160	237	32	58	48	14	203	35	114	110	143
0010	119	23	6	12	220	109	44	61	215	202	159	45	29	250	157	235
0011	24	184	170	22	150	28	149	133	129	198	219	13	145	56	63	204
0100	212	97	201	5	15	177	234	122	50	0	113	102	253	106	36	168
0101	192	65	57	104	226	3	174	101	84	151	42	128	140	60	224	112
0110	207	53	46	95	131	243	87	118	175	164	69	55	178	247	79	126
0111	158	134	217	229	139	73	93	30	254	92	142	59	27	52	248	153
1000	240	121	189	196	138	165	130	228	11	144	34	147	25	194	137	100
1001	180	135	99	222	156	241	161	208	18	20	80	197	67	105	244	124
1010	221	74	211	167	85	115	183	251	111	51	16	108	200	233	205	66
1011	41	152	214	125	163	31	86	62	155	166	176	26	75	21	188	172
1100	232	96	4	216	238	54	107	210	171	9	195	103	8	88	141	10
1101	117	2	91	123	154	43	191	236	162	116	185	81	127	19	173	193
1110	82	252	246	83	190	187	186	136	223	71	70	218	182	225	89	146
1111	40	209	98	179	255	132	17	199	77	231	213	206	227	90	94	148

1. karakter = 00001011 - 78
2. karakter = 00000000 - 1
3. karakter = 10000110 - 130
4. karakter = 10100111 - 251
5. karakter = 01101110 - 79
6. karakter = 11011111 – 193
7. karakter =11110011 - 179
8. karakter =11111010 – 213

5. Adım: Algoritmada kullanılacak Laplace dönüşümü için şifreleme algoritmasında Taylor serisi kullanılmıştır. İlk olarak e^t ile birlikte genişletilmiş Taylor serisi alınır.

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^n(a)}{n!}(x-a)^n + \cdots$$

$$= \sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n \qquad (4.1)$$

Daha sonra, genişletilirse;

$$e^t = 1 + \frac{t}{1!} + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots$$
$$= \sum_{n=0}^{\infty} \frac{t^n}{n!} \quad (4.2)$$

t^3 ile (4.3) denklem elde edilir:

$$t^{3e^t} = t^3 + \frac{t^4}{1!} + \frac{t^5}{2!} + \frac{t^6}{3!} + \cdots$$
$$= \sum_{n=0}^{\infty} \frac{t^{n+3}}{n!} \quad (4.3)$$

Sonuç olarak,

$f(t) = \sum_{n=0}^{\infty} K_n \frac{t^{n+3}}{n!}$ elde edilir.

Şifrelenecek olan ”Stegonag” açık metni 78, 1, 130, 251, 79, 193, 179, 213 sayılarına denk gelir.

$K_0 = 78, K_1 = 1, K_2 = 130, K_3 = 251, K_4 = 79, K_5 = 193, K_6 = 179, K_7 = 213$

yazarak elde edilir.

$$f(t) = \sum_{n=0}^{\infty} K_n \frac{t^{n+3}}{n!}$$
$$= K_0 \frac{t^3}{0!} + K_1 \frac{t^4}{1!} + K_2 \frac{t^5}{2!} + K_3 \frac{t^6}{3!} + K_4 \frac{t^7}{4!} + K_5 \frac{t^8}{5!} + K_6 \frac{t^9}{6!}$$
$$+ K_7 \frac{t^{10}}{7!} \quad (4.5)$$

Eğer (4.5) denklemin iki tarafına genişletilmiş güç serisi dönüşümünü uygulanırsa aşağıdaki denklem elde edilir: şunu elde ederiz:

$$T[f(t)](h) = T[\sum_{n=0}^{\infty} K_n \frac{t^{n+3}}{n!}](h)$$

$$= T[K_0 \frac{t^3}{0!} + K_1 \frac{t^4}{1!} + K_2 \frac{t^5}{2!} + K_3 \frac{t^6}{3!} + K_4 \frac{t^7}{4!} + K_5 \frac{t^8}{5!} + K_6 \frac{t^9}{6!} + K_7 \frac{t^{10}}{7!}](h)$$

$$= 78.3!\,h^3 + 1.4!\,h^4 + 130.5!\frac{h^5}{2!} + 251.6!\frac{h^6}{3!} + 79.7!\frac{h^7}{4!} + 193.8!\frac{h^8}{5!} + 179.9!\frac{h^9}{6!} + 213.10!\frac{h^{10}}{7!}$$

$$\sum_{n=0}^{\infty} K_n(n+3)!\frac{h^{n+3}}{n!} \tag{4.6}$$

$$= 468h^3 + 24h^4 + 7800h^5 + 30120h^6 + 16590h^7 + 64848h^8 + 90216h^9 + 153360h^{10}$$

468, 24, 7800, 30120, 16590, 64848, 90216, 153360 dizisi'nin mod (256)'a göre:

Bölümler; 1, 0, 30, 117, 64, 253, 352, 599,

Kalanlar; 212, 24, 120, 168, 206, 80, 104, 16, veriler elde edildi.

6. Adım: Elde edilen veriler *txt uzantılı bir dosya vasıtasıyla 2. anahtar kullanarak Ses Steganografisi'nin LSB (LeastSignificant Bit) en anlamsız biti tekniği ile seçilen bir ses dosyasının içine gizleme yapılmaktadır.

4.2. Şifre Çözme

1. Adım: Verileri 2. anahtar olarak kullanılan *.txt uzantılı bir dosya seçilerek ses dosyasının içinden gizlenilen şifreli mesaj çıkartılır.

Yukardaki adımlar neticesinde;

a) Bölümler; 1, 0, 30, 117, 64, 253, 352, 599.

b) Kalanlar; 212, 24, 120, 168, 206, 80, 104, 16 şeklinde değer sonuçları elde edilmektedir.

2. Adım:

$$A_n = \frac{K_n - K_n'}{256}$$

$256 * 1 + 212 = 468$

$256 * 0 + 24 = 24$

$256 * 30 + 120 = 7800$

$256 * 117 + 168 = 30120$

$256 * 64 + 206 = 16590$

$$256 * 253 + 80 = 64848$$

$$256 * 352 + 104 = 90216$$

$$256 * 599 + 16 = 153360$$

468, 24, 7800, 30120, 16590, 64848, 90216, 153360 değerleri elde edilir.

$$\sum_{n=0}^{\infty} K_n (n+3)! \frac{h^{n+3}}{n!} \tag{4.7}$$

$$= 468h^3 + 24h^4 + 7800h^5 + 30120h^6 + 16590h^7 + 64848h^8 + 90216h^9 + 153360h^{10}$$

$$= 78.3!\, h^3 + 1.4!\, h^4 + 130.5!\, h^5 + 251.6!\, h^6 + 79.7!\, h^7 + 193.8!\, h^8 + 179.9!\, h^9 + 213.10!\, h^{10}$$

Ters Genişletilmiş Kuvvet Serisi Dönüşümünü (4.7) denkleminin her iki tarafına da uygularsak,

$$T^{-1}\left[\sum_{n=0}^{\infty} K_n (n+3)! \frac{h^{n+3}}{n!}\right] \tag{4.8}$$

elde edilir.

$$= T^{-1}[78.3!\, h^3 + 1.4!\, h^4 + 130.5!\, h^5 + 251.6!\, h^6 + 79.7!\, h^7 + 193.8!\, h^8 + 179.9!\, h^9 + 213.10!\, h^{10}\ ]$$

$$\sum_{n=0}^{\infty} K_n \frac{t^{n+3}}{n!} = 78.t^3 + 1.t^4 + 130.\frac{t^5}{2!} + 251.\frac{t^6}{3!} + 79.\frac{t^7}{4!} + 193.\frac{t^8}{5!} + 179.\frac{t^9}{6!} + 213.\frac{t^{10}}{7!}$$

Yukarıdaki işlemler sonuçunda 78, 1, 130, 251, 79, 193, 179, 213 bu değerlere ulaşıldı.

3. Adım: Yukarıdaki (4.8) denkleminden elde edilen her bir değer için Tablo 4.2'de karşılıkları bulunur.

1. karakter = 78 - 00001011

2. karakter = 1 - 00000000

3. karakter = 130 - 10000110

4. karakter = 251 - 10100111

5. karakter = 79 - 01101110

6. karakter = 193 - 11011111

7. karakter = 179 - 11110011

8. karakter = 213 – 11111010

4. Adım:

Anahtar = F1r@tb3y

1. karakter = 00001011 2. karakter = 00000000 3. karakter = 10000110

4. karakter = 10100111 5. karakter = 01101110 6. karakter =11011111

7. karakter = 11110011 8. karakter = 11111010

y=01111001 m_8 =00001011

XOR m_{7x} = 11111001 m_7 =11110011

y_x =11110010 y_x =11110010

3=00110011 m_7 =11110011

XOR m_{6x} = 01101010 m_6 =00110101

3_x =10011001 3_x =10011001

b=01100010 m_6 =00110101

XOR m_{5x} = 00000100 m_5 =00000010

b_x =00110001 b_x =00110001

t=01110100 m_5 =00000010

XOR m_{4x} = 00111000 m_4 =00011100

t_x =00111010 t_x =00111010

@=01000000 m_4 =00011100

XOR m_{3x} = 00111100 m_3 =00011110

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =00011110

XOR m_{2x} = 00100111 m_2 =10010011

r_x =00111001 r_x =00111001

1=00110001 m_2 =10010011

XOR m_{1x} = 00001011 m_1 =10000101

1_x =10011000 1_x =10011000

F=01000110 m_1 =10000101

XOR S_x =10100110 S=01010011 mesajın 1. karakteri

F_x =00100011 F_x =00100011

5. Adım:

y=01111001 m_8 =00000000

XOR m_{7x} = 11110010 m_7 =11100101

y_x =11110010 y_x =11110010

3=00110011 m_7 =11100101

XOR m_{6x} = 01111100 m_6 =00111110

3_x =10011001 3_x =10011001

b=01100010 m_6 =00111110

XOR m_{5x} = 00001111 m_5 =10000111

b_x =00110001 b_x =00110001

t=01110100 m_5 =10000111

XOR m_{4x} = 10111101 m_4 =11011110

t_x =00111010 t_x =00111010

@=01000000 m_4 =11011110

XOR m_{3x} = 11111110 m_3 =01111111

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =01111111

XOR m_{2x} = 01000110 m_2 =00100011

r_x =00111001 r_x =00111001

1=00110001 m_2 =00100011

XOR m_{1x} = 10111011 m_1 = 11011101

1_x =10011000 1_x =10011000

F=01000110 m_1 =11011101 t_x =01111111

XOR t_{xx} =11111110 XOR t=01110100

F_x =00100011 F_x =00100011 Sm_8 =00001011 mesajın 2. karakteri

6. Adım:

y=01111001 m_8 =10000110

XOR m_{7x} = 01110100 m_7 =11101000

y_x =11110010 y_x =11110010

3=00110011 m_7 =11101000

XOR m_{6x} = 01110001 m_6 =10111000

3_x =10011001 3_x =10011001

b=01100010 m_6 =10111000

XOR m_{5x} = 10001001 m_5 =11000100

b_x =00110001 b_x =00110001

t=01110100 m_5 =11000100

XOR m_{4x} = 11111110 m_4 =01111111

t_x =00111010 t_x =00111010

@=01000000 m_4 =01111111

XOR m_{3x} = 01011111 m_3 =10101111

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =10101111

XOR m_{2x} = 10010110 m_2 =01001011

r_x =00111001 r_x =00111001

1=00110001 m_2 =01001011

XOR m_{1x} = 11010011 m_1 =11101001

1_x =10011000 1_x =10011000

F=01000110 m_1 =11101001 e_x =01100101

XOR e_{xx} =11001010 XOR e=01100101

F_x =00100011 F_x =00100011 tm_8 =00000000 mesajın 3. karakteri

7. Adım:

y=01111001 m_8 =10100111

XOR m_{7x} = 01010101 m_7 =10101010

y_x =11110010 y_x =11110010

3=00110011 m_7 =10101010

XOR m_{6x} = 00110011 m_6 =10011001

3_x =10011001 3_x =10011001

b=01100010 m_6 =10011001

XOR m_{5x} = 10101000 m_5 =01010100

b_x =00110001 b_x =00110001

t=01110100 m_5 =01010100

XOR m_{4x} = 01101110 m_4 =00110111

t_x =00111010 t_x =00111010

@=01000000 m_4 =00110111

XOR m_{3x} = 00010111 m_3 =10001011

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =10001011

XOR m_{2x} = 10110010 m_2 =01011001

r_x =00111001 r_x =00111001

1=00110001 m_2 =01011001

XOR m_{1x} = 11000001 m_1 =1110000

1_x =10011000 1_x =10011000

F=01000110 m_1 =11000011 t_x =11100001

XOR g_{xx} =11000011 XOR g=01100111

F_x =00100011 F_x =00100011 em_8 =10000110 mesajın 4. karakteri

8. Adım:

y=01111001 m_8 =01101111

XOR m_{7x} = 10011101 m_7 =00111011

y_x =11110010 y_x =11110010

3=00110011 m_7 =00111011

XOR m_{6x} = 10100010 m_6 =01010001

3_x =10011001 3_x =10011001

b=01100010 m_6 =01010001

XOR m_{5x} = 01100000 m_5 =00110000

b_x =00110001 b_x =00110001

t=01110100 m_5 =00110000

XOR m_{4x} = 00001010 m_4 =00000101

t_x =00111010 t_x =00111010

@=01000000 m_4 =00000101

XOR m_{3x} = 00100101 m_3 =10010010

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =10010010

XOR m_{2x} = 10101011 m_2 =11010101

r_x =00111001 r_x =00111001

1=00110001 m_2 =11010101

XOR m_{1x} = 01001101 m_1 =10100110

1_x =10011000 1_x =10011000

F=01000110 m_1 =10100110 a_x =11000010

XOR a_{xx} =10000101 XOR t=01100101

F_x =00100011 F_x =00100011 gm_8 =10100111 mesajın 5. karakteri

9.Adım:

y=01111001 m_8 = 11011111

XOR m_{7x} = 00101101 m_7 =01011010

y_x =11110010 y_x =11110010

3=00110011 m_7 =01011010

XOR m_{6x} = 11000011 m_6 =11100001

3_x =10011001 3_x =10011001

b=01100010 m_6 =11100001

XOR m_{5x} =11010000 m_5 =01101000

b_x =00110001 b_x =00110001

t=01110100 m_5 =01101000

XOR m_{4x} = 01010010 m_4 =00101001

t_x =00111010 t_x =00111010

@=01000000 m_4 =00101001

XOR m_{3x} = 00001001 m_3 =10000100

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 = 10000100

XOR m_{2x} = 10111101 m_2 =11011110

r_x =00111001 r_x =00111001

1=00110001 m_2 =11011110

XOR m_{1x} = 01000110 m_1 =00100011

1_x =10011000 1_x =10011000

F=01000110 m_1 =00100011 n_x =00000000

XOR n_{xx} =00000000 XOR n=01101110

F_x =00100011 F_x =00100011 am_8 =01101110 mesajın 6. karakteri

10. Adım:

y=01111001 m_8 =11110011

XOR m_{7x} = 00000001 m_7 =00000010

y_x =11110010 y_x =11110010

3=00110011 m_7 =00000010

XOR m_{6x} = 10011011 m_6 =11001101

3_x =10011001 3_x =10011001

b=01100010 m_6 =11001101

XOR m_{5x} = 11111100 m_5 =01111110

b_x =00110001 b_x =00110001

t=01110100 m_5 =01111110

XOR m_{4x} = 01000100 m_4 =00100010

t_x =00111010 t_x =00111010

@=01000000 m_4 =00100010

XOR m_{3x} = 00000010 m_3 =00000001

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =00000001

XOR m_{2x} = 00111000 m_2 =00011100

r_x =00111001 r_x =00111001

1=00110001 m_2 =00011100

XOR m_{1x} = 10000100 m_1 =01000010

1_x =10011000 1_x =10011000

F=01000110 m_1 =01000010 o_x =10110000

XOR o_{xx} =01100001 XOR o=01101111

F_x =00100011 F_x =00100011 nm_8 =11011111 mesajın 7. karakteri

11. Adım:

y=01111001 m_8 =11111010

XOR m_{7x} = 00001000 m_7 =00010000

y_x =11110010 y_x =11110010

3=00110011 m_7 =00010000

XOR m_{6x} = 10001001 m_6 =11000100

3_x =10011001 3_x =10011001

b=01100010 m_6 =11000100

XOR m_{5x} = 11110101 m_5 =11111010

b_x =00110001 b_x =00110001

t=01110100 m_5 =11111010

XOR m_{4x} = 11000000 m_4 =01100000

t_x =00111010 t_x =00111010

@=01000000 m_4 =01100000

XOR m_{3x} = 01000000 m_3 =00100000

$@_x$ =00100000 $@_x$ =00100000

r=01110010 m_3 =00100000

XOR m_{2x} = 00011001 m_2 =10001100

r_x =00111001 r_x =00111001

1=00110001 m_2 =10001100

XOR m_{1x} = 00010100 m_1 =00001010

1_x =10011000 1_x =10011000

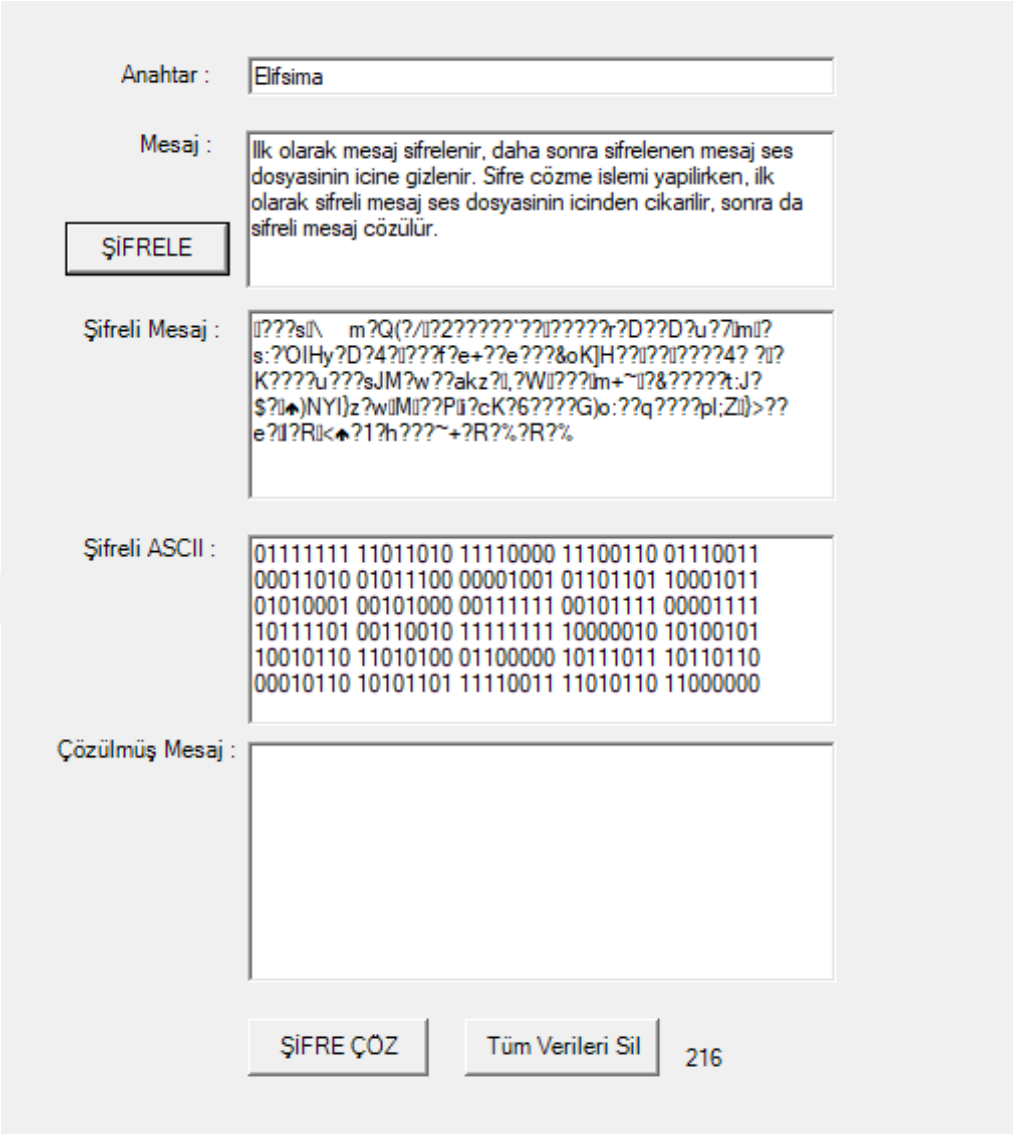
F=01000110 m_1 =00001010 g_x =10010100

XOR g_{xx} =00101001 XOR g=01100111

F_x =00100011 F_x =00100011 om_8 =11110011 mesajın 8. karakteri

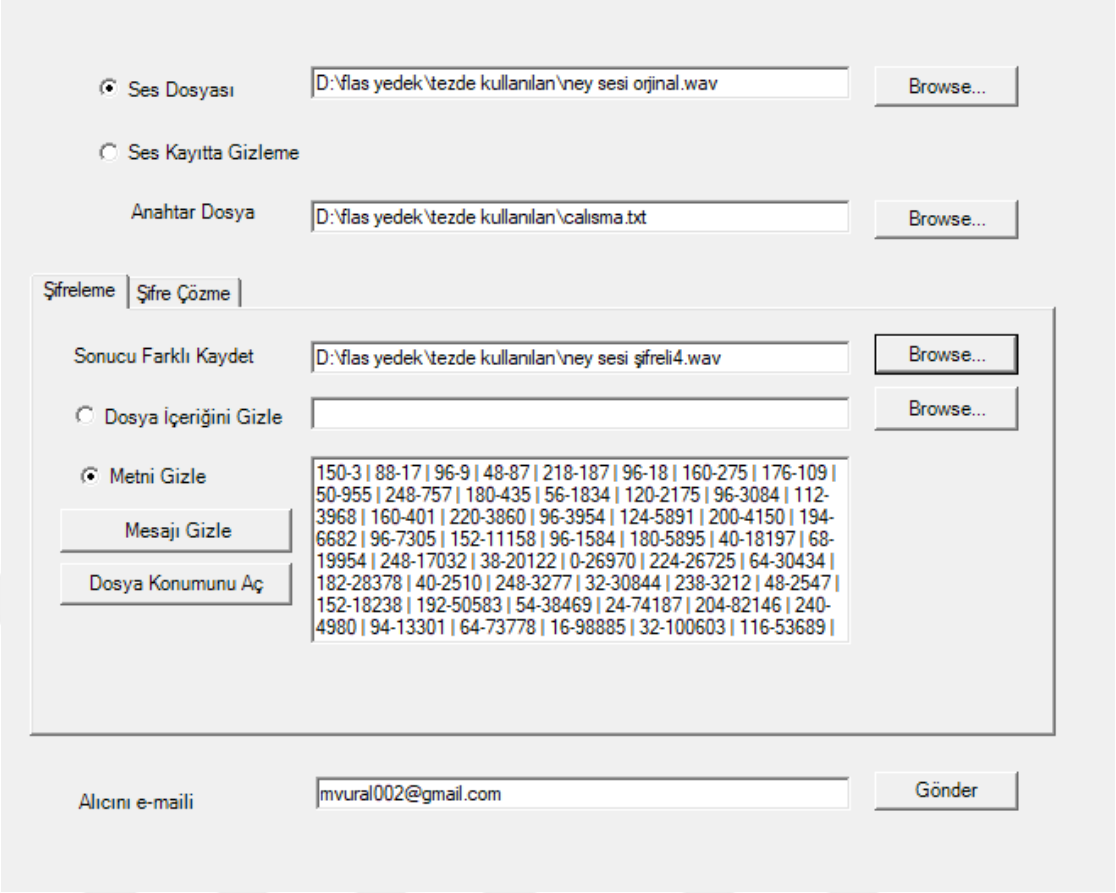
4.3 Deneysel Sonuçlar

İlk olarak mesaj şifrelenir, daha sonra şifrelenen mesaj ses dosyasının içine gizlenir. Şifre çözme işlemi yapılırken, ilk olarak şifreli mesaj ses dosyasının içinden çıkarılır, sonra da şifreli mesaj çözülür. Şekil 4.1'de önerilen algoritmaya dayalı veri şifreleme işlemi yapılmıştır.



Şekil 4.1. Şifreleme görüntüsü

Şekil 4.2'de önerilen algoritmaya dayalı şifrelenen veri, ses dosyası içine gizleme işlemi yapılmıştır.



Şekil 4.2. Şifrelenen veriyi ses içine gizleme

Anahtar Alanı: Bu alanda mesajı şifrelemek için anahtar girilir. Anahtarın uzunluğu 8 karakterden oluşmaktadır. Anahtar, 8 karakterden az veya fazla olur ise mesajı şifrelemek için kullanılan anahtar hatalı olacağından dolayı işlem hata verir.

Mesaj Alanı: Bu alana şifrelemek istenilen mesaj yazılır. Mesaj bölümü, boş olduğunda program çalıştırılırsa işlem hata verir.

Şifreli Mesaj: Mesajın şifrelenmiş halidir. Yani mesaj, bir dizi kurallara tabi tutulduktan sonra mesajın şifrelenmiş en son halidir.

Şifreli Ascii: Mesajın şifrelenmiş olan halinin ascii karşılığını gösterir.

Çözülmüş Mesaj: Şifreli mesajın çözülmüş halidir.

Ses Dosyası: Şifreli mesajı gizlemek için kullanılan ses dosyasının seçildiği alandır.

Ses Kayıtta Gizleme: Bu alanda şifreli mesaj, o anda ses kaydı yapılarak gizlenmek istenildiğinde kullanılır.

Anahtar Dosya: Şifreli mesajın ses dosyası içirisine gizlemek istenildiğinde anahtar dosya olarak kişi veya kuruma özel verilen bir kodun *.txt uzantılı bir dosyanın içine kayıt edildikten sonra bu dosya seçimi yapılarak gizleme işleminin yapıldığı alandır.

Sonucu Farklı Kaydet: Şifreli mesajın ses içerisine gizlendikten sonra nereye ve hangi isimle kayıt edileceği alandır.

Dosya İçeriğini Gizle: Gizlemek istenilen herhangi bir dosyanın seçiminin yapıldığı alandır.

Metni Gizle: Şifrelenen mesajın doğrudan geldiği alandır. Buradaki veri, ses dosyasına gizlenir.

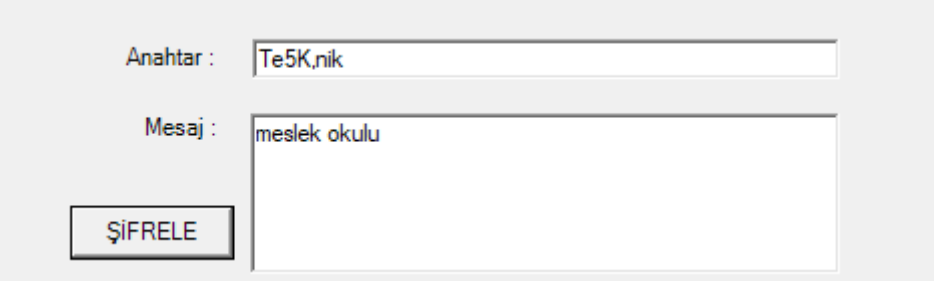
Dosya Konumunu Aç: Şifreleme işlemi sonunda gizlenen dosyaya ulaşılmak istenildiğinde dosyanın kayıtlı olduğu adresi belirten alandır.

Alıcının E-maili: Şifrelenerek gizlenen sesin, gönderilmek istenilen kişi veya kurum adresini açıklayan alandır. Şifrelenerek gizlenen ses, kime veya hangi kuruma gönderilmek isteniliyorsa, kişinin veya kurumun e-maili yazılır ve gönderme işlemi tamamlanır.

5. ÖNERİLEN YÖNTEMİN PERFORMANS ANALİZİ

Bu şifreleme algoritmasında Taylor serisi kullanılmış olup, ilk olarak sekiz karakter uzunluğunda bir anahtar ile mesaj, bir dizi kurallara tabi tutularak şifrelendi. Daha sonra elde edilen veriler şifrelenmiş mesajı alacak kişiye özel bir kod ile Sesin en anlamsız bit ’ine gizleme yapıldı. Bu teknikte Taylor serisi kullanılarak kripto analiz tekniklerinden olan dil frekans analizi saldırılarına karşı önlem alındı. Geliştirilen mesaj şifreleme ve gizleme tekniğine yönelik deneme amaçlı çeşitli saldırılar geçekleştirildi. Tekniğe karşı gerçekleştirilen bu saldırılar herhangi bir olumsuz sonuç vermedi. Bir kelimede veya bir cümle içinde kullanılan aynı harfin karşılığı farklı değerler çıkmaktadır. Bundan dolayı kripto saldırı tekniklerinden biri olan dil analizi yönteminin saldırılarına karşı önlem alınmış oldu. Dil analiz yöntemiyle yapılan saldırılarda mesajı çözmeye yönelik herhangi bir çözümleme elde edilmeyecektir.

Şekil 5.1’de önerilen algoritmaya dayalı şifrelenen verilere, dil analiz yöntemiyle saldırı yapılmıştır.

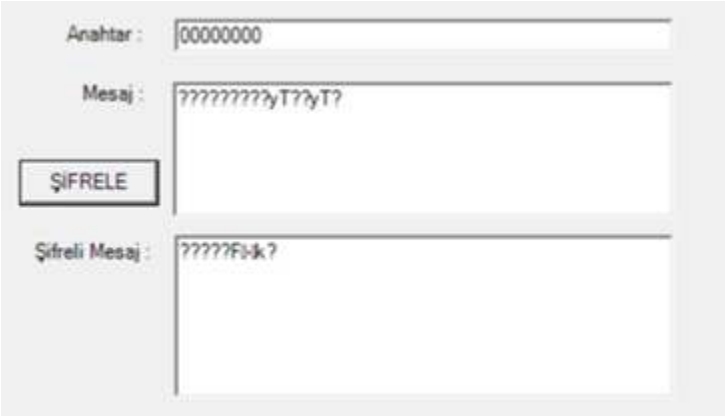


Şekil 5.1. Dil analiz görüntüsü

12-3 | 104-7 | 92-28 | 56-64 | 2-73 | 224-301 | 40-179 | 112-413 | 64-866 | 112-1268 | 252-1038 | 72-213 | 32-853 | 192-2703 | 16-4064 | 32-478 bu değerlerin inceleme sonucunda görüldüğü gibi “e”, ”u”, ”l” harfleri cümle içinde iki defa kullanılmasına rağmen bütün sonuçlar farklı değerleri göstermektedir.

Bir diğer kripto analiz yöntemi olan açık metin saldırılarına karşı ise metin önce şifrelenip, daha sonra ses içerisinde gizlendiğinden dolayı açık metin saldırıları yapılamamaktadır. Ses içerisinde gizlenen veriye ulaşılsa bile geliştirilen algoritma sayesinde açık metin saldırılarına karşı güçlü bir algoritma olduğu test edilmiştir.

Şekil 5.2'de önerilen algoritmaya dayalı şifrelenen verilere, açık metin yöntemiyle saldırı yapılmıştır.

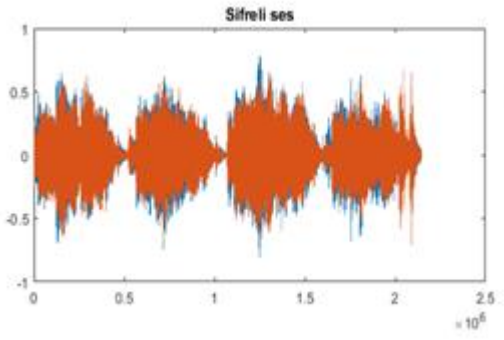


Şekil 5.2. Açık metin saldırı görüntüsü

Uygulama olarak "teknoloji" kelimesi önce şifreleniyor. Daha sonra şifrelenen mesaj ile anahtar bölümüne 00000000 değerler girildi. "Teknoloji" kelimesine ulaşılmadığı görülmektedir.

Şifrelenen mesaj, Sesin en anlamsız bit 'ine gizleme yapıldığı için sesin frekans analizi yapıldığında görüntü olarak göz ile görülebilecek herhangi bir değişikliğe rastlanmamaktadır.

Şekil 5.3'te önerilen algoritmaya dayalı şifrelenen veriler, LSB tekniği kullanılarak sesin içerisine gizlenmiştir.

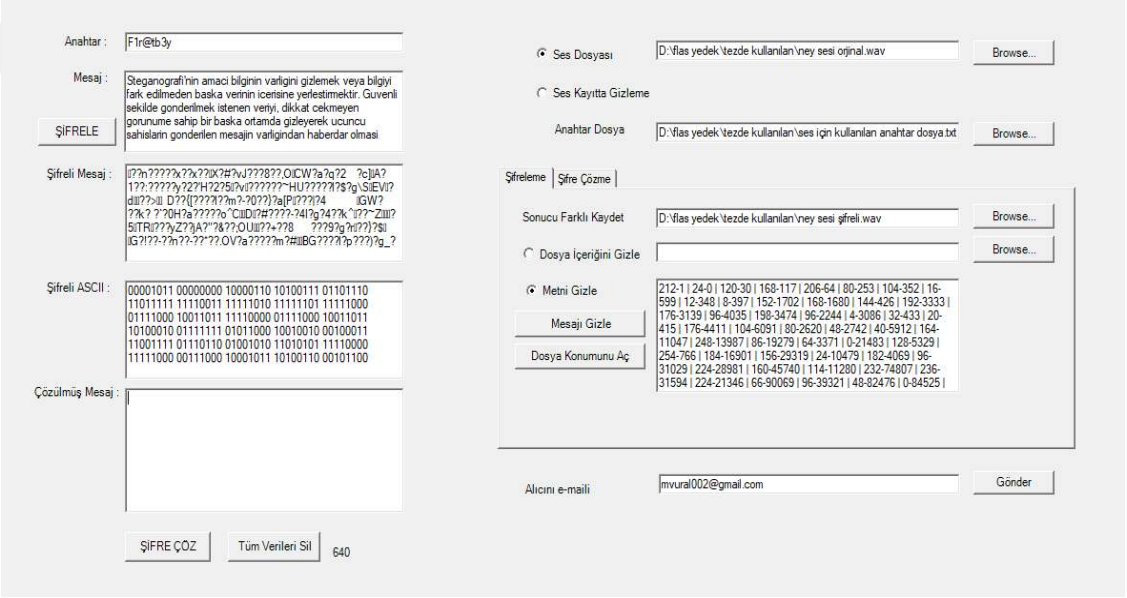


Şekil 5.3. Şifreli ses

Uygulama olarak "Steganografinin amacı bilginin varlığını gizlemek veya bilgiyi fark edilmeden başka verinin içerisine yerleştirmektir. Güvenli şekilde gönderilmek istenen veriyi, dikkat çekmeyen görünüme sahip bir başka ortamda

gizleyerek üçüncü şahısların gönderilen mesajın varlığından haberdar olması engellenir. Bu işlemler ile metin, ses, resim, video dosyaları içerisine veri gizleyebilmek mümkündür. Bu veriler herhangi bir metin dosyası olabileceği gibi, herhangi bir görüntü içerisine başka bir görüntüyü gizleme şeklinde de olabilmektedir. Yine aynı şekilde bir ses veya video dosyasının içine bir metin dosyası da saklanabilmektedir" metin parçası şifrelendi. Daha sonra bu şifrelenen metin "ney sesi. wav" adlı ses dosyasına gizlendi.

Şekil 5.4'te önerilen algoritmaya dayalı veriler ilk olarak şifrelenmiş, daha sonra da LSB tekniği kullanılarak sesin içerisine gizlenmiştir.



Şekil 5.4. Uygulama görüntüsü

Ses dosyasının içine kayıt edildiği gibi anlık olarak program üzerinden ses kaydedici açılarak şifrelenen mesaj, yeni kayıt edilen ses dosyasının içine gizleme de yapılabilmektedir.

Şifrelenen mesajı, bir dosyaya kayıt ettikten sonra isteğe göre şifreli mesaj dosya şeklinde (*.txt, *.xls, *.doc vb.) ses içerisine gizleme yapılabilmektedir.

Yapılan yazılım sayesinde program izinsiz kişiler tarafından ele geçirildiğinde, programın kim tarafın kullanıldığı ve verilerin kime gönderildiği tespit edilebilmektedir. Bu işlem bir güvenlik tedbiri olarak eklenmiştir. Bu takipler sadece izinsiz kişiler tarafından kullanıldığında yapılmaktadır.

6. SONUÇ VE TARTIŞMA

Kriptografi ve Steganografi birleştirilerek elde edilen bu hibrit model ile kriptolojiye farklı bir bakış açısı ortaya konulmuştur. Önerilen algoritma ile yapılan uygulamalarda kriptoloji dolayısıyla siber güvenlik alanında matematiğin hayati önem taşıdığı bir kez daha ispatlanmıştır. Ayrıca önerilen algoritma genişletilerek; siber savunmada kullanılabilecek önemli bir savunma sistemi haline getirilebilir. Hatta önerilen algoritmanın mobil uygulaması da geliştirilip, özellikle anlık haberleşmede kullanılmak üzere, ticari bir ürün olarak sunulabilir. Önerilen algoritma ile açık metin saldırılarına engel olunmuştur. Algoritma da taylor serisi kullanılarak, dil analiz saldırısı da engellemiştir.

Günümüz gelişen teknoloji ile veri güvenliğinin önemi de artmıştır. Gün geçtikçe siber saldırılar da artmaktadır. Yapılan bu siber saldırıların çeşitli amaçları bulunmaktadır. Yapılan siber saldırılar, verileri kopyalamak, değiştirmek veya zarar vermek amacıyla yapılmaktadır. Geliştirilen bu algoritma ile şifrelenen veriler gizlendiğinden dolayı yapılan saldırılarda verilere ulaşılamayacaktır. Gizlenen verilere ulaşılsa bile, veriler matematiksel fonksiyon kullanılarak şifrelendiğinden dolayı anlamsız veriler elde edilecektir. Geliştirilen bu algoritma veri güvenliğinde oldukça önemli katkı sağlayacaktır.

Geliştirilen bu algoritmada, kriptografi ve steganografi birleştirilerek hibrit model oluşturuldu. Bu hibrit model geliştirilmeye açıktır. Önerilen metodla şifrelenip gizlenen bir metne dil frekans saldırısı, açık metin saldırısı ve ses analiz yöntemleri kullanılarak saldırılar yapılmıştır. Algoritmanın güvenirliğini tespit etmek için başka saldırı tekniklerine de başvurulabilir. Algoritma yazılımsal olarak çalışıldı. Donanımsal ekipmanlara da entegre edilebilir. Yeni bir kripto cihazı üretilebilir. Geliştirilen algoritma, biyometrik şifreleme veya deşifreme işleminde de kullanılabilir.

KAYNAKÇA

[1] **Paar, C. and Pelzl, J.,** 2010. Understanding Cryptography, Springer - Verlag Berlin Heidelberg.

[2] **Koblitz, N.,** 1987. A course in number theory and cryptography. New York: Springer erlag.

[3] **Yerlikaya T., Buluş E., Arda D.,** 2005. AsimetrikKriptoSistemlerVeUygulamaları, *II. MühendislikBilimleriGençAraştırmacılarKongresi- MBGAK, İstanbul-TÜRKİYE.*

[4] **Keyman, E., Yıldırım, M.,** 2004. KriptolojiyeGirişDersNotları,*Uygulamalı MatematikEnstitüsü, ODTÜ, TÜRKİYE.*

[5] **Katzenbeisser, S., Petitcolas, F.A.P.,** 2000. Information Hiding Techniques for Steganography and Digital Watermarking, *Artech House, INC. 685 Canton Street Norwood, MA 02062.*

[6] **Vural, M., Gençoğlu, M.T.,** 2018. Embedded Audio Coding Using Laplace Transform For Turkish Letters, *Journal of the Technical University - Sofia Plovdiv branch, Bulgaria "Fundamental Sciences and Applications" Vol. 24, pp.109-116.*

[7] **İnternet:** Kriptoloji, 04.02.2013. *http://tr.wikipedia.org/wiki/Kriptoloji.*

[8] **R. Yılmaz,** 2010. "Kriptolojik Uygulamalarda Bazı İstatistik Testler", Yüksek Lisans Tezi, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü İstatistik Anabilim Dalı, Konya.*

[9] **Akyelek, S., Yıldırım, H.M., Tok, Z.Y.** 2011. "Kriptoloji ve Uygulama Alanları: Açık Anahtar Altyapısı ve Kayıtlı Elektronik Posta", Akademik Bilişim 2011, İnönü Üniversitesi, Malatya, 713-718.

[10] **Gençoğlu, M.T.**, 2019. Importance of Cryptography in Information Security, *IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 21, Issue 1, Ser. II (Jan* - Feb 2019), PP 65-68.

[11] **Ellison, C,** 2004. Cryptography Timeline. http://world.std.com/~cme/html/timeline.html, December 11, 2004.

[12] http://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/%C5%9Fifreleme-y%C3%B6ntemleri İTÜ Bilgi İşlem Daire Başkanlığı, Şifreleme Yöntemleri, 07 Eylül 2013.

[13] **Murray A.H., Burchfiled R.W (eds.),** 1933. The Oxford English Dictionary: Being a Corrected Re-issue, Oxford, England: Clarendon Pres.

[14] **Weiss, M.,** 2004. Principles of Steganography, Math 187: Introduction to Cryptography Professor Kevin O'Bryant.

[15] **Kadry, S. and Nasr, S.,** 2013. New Generating Technique for Image Steganography, *Lecture Notes on Software Engineering, Vol. 1, No. 2, May.*

[16] **Gençoğlu, M.T., Baleanu, D.,** 2017. Power Series Transform In Cryptology and ASCII, *International Workshop On Mathematical Methods In Engineering Cankaya University, Ankara, Turkey April 27-29.*

[17] **Memon, N., Wong, P.,** 1998. Protecting digital media content, *Communications of the ACM, vol 41, no. 7, pp. 34–43, July.*

[18] **Wang, H., Wang, S.,** 2004. Cyber Warfare: Steganography vs. Steganalysis, *Communications of the ACM, vol. 47, no. 10, October.*

[19] **Gençoğlu, M.T.,** 2017. Programming Encryption Algorithms with steganography, *International Conference on Engineering Techonology and Innovation, Sarajevo, 22-26 march, 58.*

[20] **Simmons G.,** 1984. The Prisoners' Problem and the Subliminal Channel, *crypto83 Advances in Cryptology, pp. 51-67, Aug 22 -24.*

[21] **Anderson, R.J., Petitcolas, F.A.P.,** 1998. On the Limits of Steganography, *IEEEJournal of Selected Areas in Communications,16(4):474-481, Special Issue on Copyright & Privacy Protection. ISSN 0733–8716, May.*

[22] **Johnson, N. F., Duric Z. veJajodia S.,** 2001. Information Hiding: Steganography and Watermark ing - Attacks and Countermeasures, Boston.

[23] **Johnson, N.F. and Rude, T.,** 2001. Introduction to Steganography Hidden Information*, Regional Computer Forensic Group (RCFG) and Mid-Atlantic Chapter of High- Technology Crime Investigation Association (HTCIA) George Mason University Computer Forensics Symposium (GMU 2001), Fairfax, VA, August 13-17.*

[24] **Fridrich, J.,** 2004. Information Hiding: 6th International Workshop, *IH 2004, Toronto, Canada, May 23-25 2004, Revised Selected Papers, Springer, New York, p 180.*

[25] **Bender, W., Gruel, D., Morimoto, N., Lu, A.** 1996. Techniques for data hiding, *IBM Systems Journal, vol. 35, no 3, pp. 313-336.*

[26] **Benett, K.,** 2004. Linguistic steganography- survey, analysis and robustness concerns for hiding information in text, *Purdue University, CERIAS Tech.*

[27] **Shahreza, M. S. and Shahreza, M. H. S.,** 2007. Text steganography in SMS, *Int. Conf. on Convergence Information Technology, pp. 2260-2265 2007.*

[28] **Gençoğlu, M.T.,** 2017. Combining Cryptography with Steganography, *ITM Web of Conferences **13**, DOI: 10.1051/itmconf 01010 2017.*

[29] **Shahreza, M. H. S. and Shahreza, M. S.,** 2006. A new approach to Persian/Arabic text steganography, In Proceedings of 5th IEEE/ACIS Int. Conf. on Computer and Information Science and 1st IEEE/ACIS Int. *Workshop on Component-Based Software Engineering, Software Architecture and Reuse, pp. 310-315.*

[30] **Bhattacharyya, S., Banerjee, I., and Sanyal, G.,** 2010. A Novel Approach of Secure Text Based Steganography Model using Word Mapping Method (WMM) *International Journal of Computer and Information Engineering 4:2.*

[31] **Sweldens, W.,** 1997. The lifting scheme. A construction of second generation wavelets. *SIAM J. Math. Anal., 29:511–546, 1997.*

[32] **Johnson, N.F. and Jajodia, S.,** 1998. Steganography: seeing the unseen. *IEEE Computer, 16:26–34, 1998.*

[33] **Sweldens, W., andCalderbank, R.,. Daubechies, I and Yeo, B.L.,** 1998. Wavelet transforms that map integers to integers. *Appl. Comput. Harmon. Anal., 5:332–369.*

[34] **Por, L. Y. and Delina, B.** 2008. Information hiding- a new approach in text steganography, 7 th WSEAS Int. *Conf. on Applied Computer and Applied Computational Science, pp. 689-695.*

[35] **Por, L. Y., Ang, T. F. and Delina, B.,** 2008. WhiteSteg- a new scheme in information hiding using text steganography, *WSEAS Transactions on Computers, vol.7, no.6, pp. 735-745.*

[36] **Shahreza, M. H. S. and Shahreza, M. S,** 2006. A new synonym text steganography, Int.*Conf. on Intelligent Information Hiding and Multimedia Signal Processing, 2006, pp. 1524-1526.*

[37] **Low, S. H., Maxemchuk, N. F., Brassil, J. T. and Gorman, L. O.,** 1995. Document marking and identification using both line and word shifting, *INFOCOM'95 Proceedings of the Fourteenth Annual Joint Conf. of the IEEE Computer and Communication Societies, pp. 853-860.*

[38] **Gençoğlu, M.T.,** 2019 Embedded image coding using laplace transform for Turkish letters, *Multimedia Tools and Applications, Volume 78, Issue 13, pp 17521–1753, 2019.*

[39] **Cummins, J., Diskin, P., Lau, S. and Parlett, R.,** 2004. Steganography and digital watermarking, *School of Computer Science, pp.1-24.*

[40] **Rabah, K.,** 2004. Steganography-the art of hiding data, *Information Technology Journal, vol.3, pp. 245-269.*

[41] **Brassil, J. T., Low, S., Maxemchuk, N. F. and Gorman, L. O.,** 1995. Electronic marking and identification techniques to discourage document copying, *IEEE Journal on Selected Areas in Communication, vol.1, pp. 1495-1504.*

[42] **Banerjee, I., Bhattacharyya, S. and Sanyal, G.,** 2011. Novel text steganography through special code generation, *Int. Conf. on Systemics, Cybernetics and Informatics, pp. 298-303.*

[43] **Bhattacharyya, S., Banerjee, I. and Sanyal, G.,** 2010. A novel approach of secure text based steganography model using word mapping method, *Int. Journal of Computer and Information Engineering, vol.4, pp. 96-103.*

[44] **Liu, T. Y. and Tsai, W. H.,** 2007. A new steganographic method for data hiding in Microsoft word documents by a change tracking technique, *IEEE Transactions on Information Forensics and Security, vol.2, no.1, pp. 24-30.*

[45] **Khairullah, M.,** 2011. A novel text steganography system in cricket match scorecard, *Int. Journal of Computer Applications, vol.21, pp. 43-47.*

[46] **Kabetta, H., Dwiandiyanta, B. Y. and Suyoto,** 2011. Information hiding in CSS: a secure scheme text steganography using public key cryptosystem, *Int. Journal on Cryptography and Information Security, vol.1, pp. 13-22.*

[47] **Amin, M.M. and Salleh, M. and Ibrahim, S. and Katmin,** 2003. Information hiding using steganography, *4th National Conference on Telecommunication Technology Proceedings, Shah Alam, Malaysia, Page(s):21 – 25.*

[48] **Sellars, D.,** 1999. An Introduction to Steganography, *Student Papers. http://www.cs.uct.ac.za/ courses/CS400W/NIS04/papers99/dsellars/index.html.*

[49] **Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.,** 1999. Information Hiding- A Survey, *Process of IEEE, vol.87, no.7, pp.1062-1078, July.*

[50] **Huang, X., Kawashima, R., Segawa, N., Abe, Y.,** 2006. The Real-Time Steganography Based on Audio-to-Audio Data Bit Stream, *Technical report of IEICE, ISEC, vol.106 pp.15-22, September.*

[51] **Jayaram, P., Ranganatha, H.R., Anupama. H,S.,** 2011. Information hiding using audio steganography – a survey. *The International Journal of Multimedia & Its Applications (IJMA) Vol.3, No.3, August.*

[52] **Cvejic, N.,Seppben, T.,** 2002. Increasing the capacity of LSB-based audio steganography. *FIN- 90014 University of Oulu, Finland.*

[53] **Qi, Y.C., Ye, L., Liu, C.,** 2009. Wavelet domain audio steganalysis for multiplicative embedding model. *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, Baoding, 12-15 July.*

[54] **Qiao, M., Sung, A.H., Liu, Q.,** 2009. Feature Mining and Intelligent Computing for MP3 Steganalysis. *International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, Şanghay, 25 September s. 627-630.*

[55] **Shahreza, S.S., Shalmani,. M.T.M.,** 2008. High capacity error free wavelet domain speech steganography. *Acoustics, Speech and Signal Processing. ICASSP. IEEE International Conference on 31 March-4 April ss. 223-227.*

[56] **Vapnik, V.N.,** 1998. Statistical Learning Theory, John Wiley & Sons.

[57] **Choudry, K. N. and Aakash, W.,** 2015. A Survey Paper on Video Steganography. *International Journal of Computer Science and Information Technologies, 6 (3), 2335-2338.*

[58] **M. Aydın, G. Gökmen, B. Kuryel, G. Gündüz,** Diferansiyel Denklemler ve Uygulamaları, *Barış Yayınları, 1990, pp. 332-349.*

[59] **Gençoğlu, M.T.,** Use of Integral Transform in Cryptology Science and Eng. J of Fırat Univ., *Vol. 28, No. 2, pp. 217-220,2016.*

ÖZGEÇMİŞ

Mehmet VURAL, 1987 yıllında Adıyaman Merkez Bağpınar köyünde doğmuştur. İlk ve orta eğitimini Adıyaman Merkez Bağpınar köyünde bulunan Bağpınar İlköğretim Okulunda, lise eğitimini ise Adıyaman'ın Besni ilçesinde bulunan Endüstri Meslek Lisesinde tamamlamıştır. 2007 yılında Cumhuriyet Üniversitesi Sağlık Hizmetleri Meslek Yüksek Okulu Anestezi ön lisans eğitimine başlamış ve 2009 yılında mezun olmuştur. 2009 yılında Fırat Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Öğretmenliği lisans eğitimine başlamış ve 2014 yılında mezun olmuştur. 2017 yılında Fırat Üniversitesi Teknoloji Fakültesi Yazılım Mühendisliği lisans eğitimine başlamış ve 2018 yılında mezun olmuştur. 2016 yılında Fırat Üniversitesi Teknoloji Fakültesi Mekatronik Mühendisliğinde yüksek lisans eğitimine başlamış ve bu eğitimini halen sürdürmektedir.