T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENTİTÜSÜ
MATEMATİK ANABİLİM DALI

STEGANOGRAFİ VE BLOK ZİNCİRİ YARDIMI İLE BİLGİ SAKLAMA VE GÜVENLİĞİ SAĞLAYAN BİR MERKEZSİZ UYGULAMA

YÜKSEK LİSANS TEZİ

ZİYADİN KARAKOÇ

DANIŞMAN
DOÇ. DR. SELÇUK TOPAL

OCAK 2024
BİTLİS

T.C.
BİTLİS EREN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENTİTÜSÜ
MATEMATİK ANABİLİM DALI

STEGANOGRAFİ VE BLOK ZİNCİRİ YARDIMI İLE BİLGİ SAKLAMA VE GÜVENLİĞİ SAĞLAYAN BİR MERKEZSİZ UYGULAMA

YÜKSEK LİSANS TEZİ

ZİYADİN KARAKOÇ
ORCID 0000-0001-5106-0060

DANIŞMAN
DOÇ. DR. SELÇUK TOPAL

OCAK 2024
BİTLİS

T.C.

BİTLİS EREN ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YÜKSEK LİSANS TEZ ÇALIŞMASI

ETİK BEYANI

Lisansüstü Eğitim Enstitüsü Matematik Anabilim Dalı Yüksek Lisans öğrencisiyim. Hazırlamış olduğum "**Steganografi ve Blok Zinciri Yardımı ile Bilgi Saklama ve Güvenliği Sağlayan Bir Merkezsiz Uygulama**" başlıklı tezde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi; tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu; tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi; kullanılan verilerde herhangi bir değişiklik yapmadığımı; bu tezde sunduğum çalışmanın özgün olduğunu bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim. …/01/2024

İmza

Ziyadin KARAKOÇ

T.C.

BİTLİS EREN ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

TEZ YAZIM KILAVUZU UYGUNLUK BEYANI

"**Steganografi ve Blok Zinciri Yardımı ile Bilgi Saklama ve Güvenliği Sağlayan Bir Merkezsiz Uygulama"** başlıklı yüksek lisans tezi Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Tez Yazım Kılavuzuna uygun olarak hazırlanmıştır. …/01/2024

Tezi Hazırlayan	**Danışman**
İmza	İmza
Ziyadin KARAKOÇ	Doç. Dr. Selçuk TOPAL

Matematik Anabilim Dalı Başkanı

İmza

Doç. Dr. Derya ARSLAN

T.C.

BİTLİS EREN ÜNİVERSİTESİ

LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

TEZ ONAY SAYFASI

Ziyadin KARAKOÇ tarafından hazırlanan "**Steganografi ve Blok Zinciri Yardımı ile Bilgi Saklama ve Güvenliği Sağlayan Bir Merkezsiz Uygulama**" adlı tez çalışması, 23/01/2024 tarihinde yapılan sınavla aşağıdaki jüri tarafından oybirliği ile Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Matematik Anabilim Dalı' nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

JÜRİ:	**İMZA**
Danışman: Doç. Dr. Selçuk TOPAL Gebze Teknik Üniversitesi	_____
Üye: Doç. Dr. Derya ARSLAN Bitlis Eren Üniversitesi	_____
Üye: Doç. Dr. Sefa Anıl SEZER İstanbul Medeniyet Üniversitesi	_____

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü Yönetim Kurulunun…../…../…. tarih ve …… sayılı kararıyla jüri tarafından kabul edilmiş bu çalışmanın yüksek lisans tezi olarak kabulü onaylanmıştır.

…../…../…..

Doç. Dr. Mehmet Bakır ŞENGÜL

Enstitü Müdürü

T.C.

Bitlis Eren Üniversitesi Lisansüstü Eğitim Enstitüsü

Matematik Anabilim Dalı

STEGANOGRAFİ VE BLOK ZİNCİRİ YARDIMI İLE BİLGİ SAKLAMA VE GÜVENLİĞİ SAĞLAYAN BİR MERKEZSİZ UYGULAMA

Yüksek Lisans Tezi

Ziyadin KARAKOÇ

Danışman: Doç. Dr. Selçuk TOPAL

Ocak 2024

ÖZET

Bu tez çalışması, günümüzde artan internet kullanımıyla birlikte bilgi güvenliğini ve mahremiyetin sağlanması amacıyla geliştirilen bir yazılımı ele almaktadır. Steganografi; ses, video, görüntü ve metin içeren bilgilerin güvenli bir şekilde paylaşılmasını mümkün kılan bir yöntem olarak kullanılmaktadır. LSB tekniği ile rastgele bir biçimde resmin herhangi bir bölgesine bilgi gömme işlemi gerçekleştirilmiştir. Bu güvenli iletişim yönteminin yanı sıra, bilgi güvenliği ve mahremiyetinin korunması için IPFS uzantılı kullanıcı dostu arayüz yazılıma entegre edilmiştir. Bu entegrasyon, herhangi bir hostinge ihtiyaç duyulmadan aracısız bir ortam oluşturulmasına olanak tanımaktadır. Ayrıca, yazılımın güvenilirliği, şeffaflığı ve merkeziyetsizliği için blok zinciri ortamına kayıt yapılarak daha yüksek bir güvenlik seviyesi elde edilmiştir. Bu çalışma, steganografi, blok zinciri, IPFS ve LSB kavramlarını kullanarak iletişim ve veri gizliliği alanlarında etkili bir çözüm sunmaktadır.

Anahtar kelimeler: Steganografi, kriptografi, blok zinciri, şifreleme, bilgi güvenliği

Republic of Türkiye

Bitlis Eren University Graduate School

Department of Mathematics

A DECENTRALIZED APPLICATION PROVIDING INFORMATION STORAGE AND SECURİTY WITH THE HELP OF STEGANOGRAPHY AND BLOCKCHAIN

Master Thesis

Ziyadin KARAKOÇ

Supervisor: Assoc. Prof. Dr. Selçuk TOPAL

January 2024

ABSTRACT

This thesis study focuses on the development of software aimed at ensuring information security and privacy in response to the increasing use of the internet in contemporary times. Steganography is employed as a method enabling the secure sharing of information encompassing audio, video, images, and text. The LSB technique is utilized to embed information randomly into any region of an image, facilitating a secure communication method. In addition to this secure communication approach, an IPFS extension with a user-friendly interface has been integrated into the software to safeguard information security and privacy. This integration allows the creation of a decentralized environment without the need for any hosting services. Furthermore, to enhance the software's reliability, transparency, and decentralization, registration on a blockchain platform has been implemented, achieving a higher level of security. This study showcases an effective solution in the realms of communication and data privacy by employing the concepts of steganography, blockchain, IPFS, and LSB.

Keywords: Steganography, cryptography, blockhain, encryption, information security

TEŞEKKÜR

Yüksek lisans öğrenimim süresince bana hep destek olan, sadece bu çalışmada değil hayatta da yol gösteren değerli hocam ve tez danışmanım Sayın Doç. Dr. Selçuk TOPAL' a teşekkür ederim.

Ayrıca her zaman yanımda olan ve çevirilerde bana yardımcı olan eşim Serap KARAKOÇ' a, sonsuz teşekkürlerimi sunmayı bir borç bilirim.

İÇİNDEKİLER

ÇİZELGELERİN DİZİNİ

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

RESİMLER DİZİNİ

KISALTMALARIN DİZİNİ

ADD	: Ayrık Dalgacık Dönüşümü
CID	: Content ID (İçerik Kimliği)
EVM	: Ethereum Virtual Machine (Ethereum Sanal Makinasi)
IoT	: Internet of Things (Nesnelerin İnterneti)
IPFS	: Inter Planetary File System (Gezegenler Arası Dosya Paylaşımı)
LSB	: Least Significant Bit (En Önemsiz Bite Ekleme)
ML	: Machine Learning (Makine Öğrenimi)
MSE	: Mean Squared Error (Ortalama Kare Hatası)
NFT	: Nan Fungible Token (Nitelikli Fikri Tapu)
PSNR	: Peak signal-tonoise ratio (Tepe sinyal-gürültü oranı)
P2P	: Peer-to-Peer (Eşler Arası)
WWW	: World Wide Web

1. GİRİŞ

Dünya genelinde akıllı teknolojilerin kullanımı hızla artmakta ve bu artış, akıllı teknolojilerin insan hayatında daha etkin bir rol oynamasına neden olmaktadır. Akıllı teknolojilerin yaygın kullanımı; akıllı şehirler, ulaşım sistemleri, sağlık teknolojileri ve akıllı evler gibi alanlarda gelişen yapıları da beraberinde getirmiştir. Özellikle Nesnelerin İnterneti (Internet of Things, IoT) ile birlikte, akıllı teknolojinin üst düzey güvenlik sistemlerine olan ihtiyacı daha çok ön plana çıkmaktadır. 2016 yılında Agecon Search şirketinin yayımladığı bir araştırma, 2021 yılının sonuna kadar yaklaşık 46 milyar cihazın Nesnelerin İnternetine bağlanacağını öngörmüştü [1]. Karel Teknoloji Yazılımı' nın yaptığı araştırmalar, 2021' den önce 10 milyar ile 11 milyar arası cihazın internete bağlandığını ve bu sayının 2022 'de 50 milyarı aşacağını belirtmişti [2]. 2019 yılında ise sadece 8. 6 milyar IoT cihazı bulunurken, 2023 yılının sonlarına doğru bu sayının 42.2 milyara ulaşması beklenmektedir [3]. Akıllı teknolojilerin ve Nesnelerin İnterneti' nin gelişimiyle birlikte, güvenlik ihtiyaçları da artmıştır. Bu bağlamda, insanlar dijital güvenliklerini sağlamak için kriptografi ve steganografi gibi iki yaygın yöntemi kullanmaktadır.

Kriptografi, gizlenen veriyi şifreleyerek koruma sağlarken, steganografi ise verinin varlığını gizlemeyi amaçlar. Her iki yöntem de asıl amacı, gizlenen veriyi üst düzey bir güvenlik sistemi ile korumaktır. Ancak, günümüzün teknolojik dünyasında, yalnızca kriptografik veya steganografik bir yaklaşımın tamamen güvenli olduğunu düşünmek yanıltıcı olabilir. Özellikle resmi kurumlar, üniversiteler, iş dünyası ve sivil bireyler gibi farklı kullanıcılar, belirli kişilerle paylaşmak istedikleri özel bilgilere sahip olabilirler. Bu bilgiler; bir toplantının dijital verileri, özel şifrelerin dijital bir hesapta saklandığı bilgiler veya devlet büyüklerinin savaş sırasında dijital ortamda iletişim kurduğu veriler gibi hassas olabilir. Önemli olan, gizlenen bilginin istenen kaynağa en güvenilir ve fark edilmeden iletilmesidir. Bu tez çalışması, özellikle bu zorlu soruna bir çözüm bulmaya odaklanmaktadır.

Steganografi, tarih boyunca bilgi gizleme yöntemi olarak kullanılmış önemli bir kavramdır. Temel amacı, gizlenen bilginin varlığını saklamaktır. Ancak günümüz dijital dünyasını düşündüğümüzde, bir bilginin sadece steganografik yöntemlerle korunması ve istenilen kişi veya kurumlara sağlam bir şekilde ulaştırılması giderek daha çok zorlaşmaktadır.

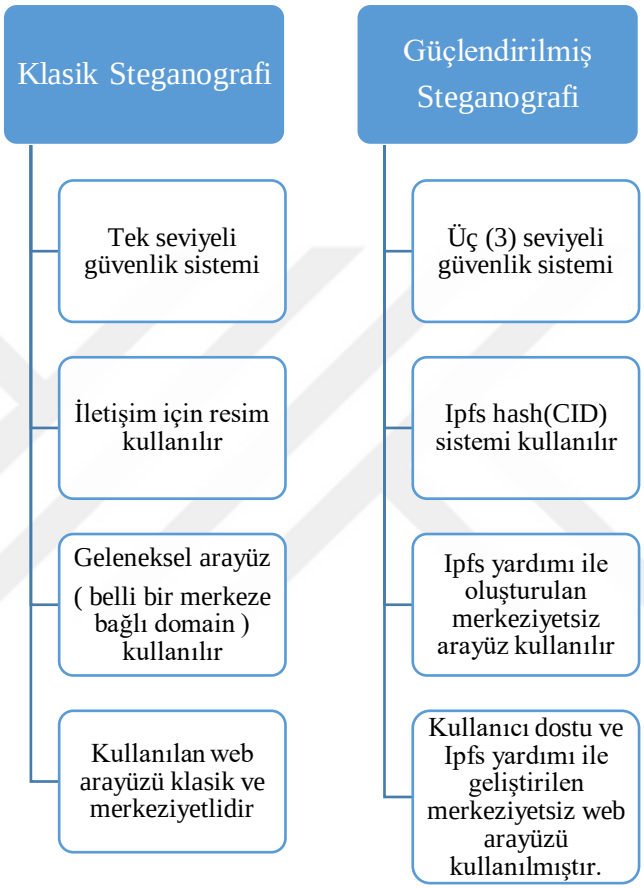
Mevcut literatür incelendiğinde, bilginin gizlenmesi gereken durumlarda sadece steganografik yöntemlerin kullanıldığı veya en fazla ikinci bir yöntemin devreye sokulduğu görülmüştür [4-24]. Ancak bu tür yaklaşımlar, bilginin güvenilirliğini istenen düzeyde sağlamada yetersiz olabilir. Bu tez çalışmasında, steganografik bir yapı oluşturmanın ötesinde, bir Inter Planetary File System (IPFS) protokolü geliştirilmiştir. Bu Ipfs protokolü kullanılarak kullanıcı dostu ve merkeziyetsiz bir arayüz oluşturulmuştur. Ipfs, merkezi olmayan bir Peer to Peer (P2P) yani eşler arası dosya paylaşım sistemidir ve her saklanan dosya için benzersiz bir içerik karması oluşturulur (hash). Bu yaklaşım, veri güvenliğini artırırken depolama ve hesaplama maliyetlerini de azaltır [25].

Günümüz en büyük dijital sorunu olan güvenlik açığını azaltmak için geliştirilen bu projede bireyler kullanıcı dostu arayüze girişlerini metamask kimliği ile sağlarlar. Sisteme giriş yapan bireyler gerçekleştirmek istedikleri eylemi seçip işlemlerini yaptıktan sonra sistem tarafından bir blok zincir mantığı ile çalışan bir kod elde ederler. Bu koda kısaca Ipfs hash' i diyebiliriz. Ipfs' nin çalışma mantığı da blok zincirinin çalışma mantığına dayanır.

Blok zinciri teknolojisi, bir şirket veya uygulama değil, dijital verileri belgelemenin tamamen yeni bir yoludur. Bu teknoloji; sosyal ağlar, iletişim uygulamaları, oyunlar, veri değiş tokuşları, depolama platformları, oy kullanma sistemleri, tahmin pazarları, çevrimiçi mağazalar ve daha birçok alanda kullanılabilir blok zinciri uygulamalarının geliştirilmesine olanak sağlar. Bu açıdan bakıldığında, blok zinciri, internete benzer bir potansiyele sahiptir ve birçok uzman bu teknolojiyi "web 3.0" olarak adlandırmıştır [26]. Bir blok zincirine kaydedilen bilgiler çok çeşitli olabilir. Örneğin; para transferleri, mülkiyet belgeleri, işlemler, kişilerin kimlik bilgileri, iki taraf arasındaki antlaşmalar veya bir cihazın elektrik faturası gibi. Ancak bu verilerin blok zincirine kaydedilmesi için çeşitli aygıtlardan onay alınması gerekir. Bu onay sürecine "mutabakat" da denir ve bir şeyin bir blok zincirinde depolanmasına karar verildiğinde, bu kayıt tartışmasız bir şekilde orada kalır ve kaydı yapanların bilgisi ve izni olmadan değiştirilemez ve kaldırılamaz [27]. Bu nedenle, bu yöntemle dijital ortamda güvenilirlik düzeyi artmıştır.

Bu tez çalışmasında blok zinciri teknolojisi ve steganografik yapılar bir araya getirilerek üst düzey bir güvenlik sistemi oluşturulmaya çalışılmıştır. Normal bir steganografik yapı kullanıldığında bir resme bir bilgi, kayıt edilir ve bilginin kayıtlı

olduğu resim (image) indirilip paylaştırılır. Bizim geliştirdiğimiz projede bilgi bir resme kayıt edildikten sonra resim paylaşılmaz. Bunun yerine resmin hash (CID) kodu paylaşılır böylece hem verinin bozulmasının önüne geçilir hem de güvenlik açığı en asgari düzeye indirilir. Şekil 1. 1' de Klasik Steganografi ile bizim kurduğumuz Güçlendirilmiş Stegaografik yapının karşılaştırılması gösterilmiştir.



Şekil 1.1. Klasik Steganografi ile Güçlendirilmiş Steganografi' nin karşılaştırılması.

1.1. İlgili Çalışmalar

Steganografik yapı kullanarak elde edilen şifrelenmiş resmi aynı yöntem ile tekrar steganografik bir resim oluşturarak dah a güvenli bir gizleme metodu oluşturmak istemişlerdir [28]. Finansal banka verilerinin güvenirliğini sağlamak için steganografik yapı ile beraber blok zinciri ve akıllı sözleşmeler tekniğini önermişlerdir [29]. Araştırmacılar geliştirdikleri blok zinciri tabanlı steganografik yapıyı istenilen gizli mesaj kaynağını gönderici ve alıcı tarafından tespit edilememesini sağlamışlardır. Buda söz konusu yapının blok zinciri sayesinde merkeziyetsizleştiğini gösterir [30]. Araştırmacılar Makine Öğrenimi üzerine veri paylaşımının güvenirliğini ve veri paylaşımının şefafflığını kontrol edebilmek için blok zincir teknolojisi entegre edilen bir steganografik

yapı önermişlerdir [31]. Bulut depolama sistemlerinde kullanılmak üzere steganografik yapı içeren bir Ipfs sistemi önerilmiştir [32]. Derin öğrenmedeki deep fake yani derin bilgi hırsızlığının önüne geçilmesi için blok zincirili tabanlı steganografik bir yapı geliştirmişlerdir [33]. Özellikle eğitim hayatı boyunca bireylerin eğitim kayıtlarının aktarılması ve güvenle saklanabilmesi için blok zinciri tabanlı steganografik bir yapı önermişlerdir [34]. Sağlık sektöründe büyük veri depolanmasını gerektiren durumlarda bilgi güvenirliğinin ve şeffaflığının sağlanması için Ipfs tabanlı steganografik bir yapı önermişlerdir [35]. Geliştirilen bir ses NFT' si istenen verinin gizlenmesi için steganografik bir yapı önermişlerdir [36]. Müzik dünyasında dijital veri ihlalinin önüne geçilmesi amacıyla akıllı kontrat içeren ethereum ağı ile çalışan steganografik bir yapı önerilmiştir [37]. Sağlık sektöründeki verilerin quantum sistemine ek olarak steganografik bir yapı kullanılarak güvenli yapı paylaşımını önermiştir [38]. Sağlık sektöründeki bilgi depolanmasının güvenilir bir şekilde aktarılması için IoT ile beraber steganografik yapı önermişlerdir [39]. Kriptografik veriler kullanarak tıbbi verileri çok katmanlı güvenlik sistemleri ile destekleyen bir yapı tasarlamışlardır [40]. İlgili konferans bildirisinde araştırmacılar gelişen sağlık sektöründeki verilerin özellikle de tele-tıp denilen e-sağlık hizmetlerindeki verilerin ihlale uğramaması için bulut depolama sisteminde steganografik yapıların kullanılmasını ve bu yapıların IoT ile desteklenmesini önermişlerdir [41]. Sağlık sektöründeki dijital verilerin güvenle aktarılması için dijital ortamda kullanılmak üzere steganografi destekli LSB tekniği kullanılarak DES yöntemi önerilmiştir [42]. Sağlık sektöründe hasta bireylerin kişisel bilgilerinin güvenli bir şekilde bir dijital bulutta kaydedilmesi için steganografik bir yöntem önermişlerdir [43]. Hastalara ait olan EKG gibi kayıtların depolanacağı güvenilir bir steganografik yapı geliştirmişlerdir [45]. Hastalara ait gizli bilgilerin Manyetik Rezonans görüntülerinin güvenle saklanabileceği Ayrık Dalgacık Dönüşümü (ADD) yöntemi ile steganografik yapı önermişlerdir [46]. E-reçete uygulamasının güvenirliğini sağlamak amacıyla, yayılım spektrum görüntüsü kullanarak gelişmiş şifreleme standardını ve steganografi algoritmasını entegre ederek bir sistem geliştirmişlerdir [47]. Eliptik eğri kriptografisi ve görüntü steganografisi yöntemlerini birleştirerek MR ve tıbbi kayıtların güvenli dijital ortamda saklanması için öneride bulunmuşlardır [48]. Tıbbi işlemlerde hastaların daha iyi güvenliğini ve gizliliğini sağlamak için Q faktörü dalgacık dönüşümü (TQWT) ve tekil değer ayrıştırma (SVD) teknikleri ile beraber steganografik teknikler önermişlerdir [49]. Hasta verilerinin güvenli bir şekilde hasta bakıcılara iletilmesi için gerekli hasta

bilgilerinin QR kodlarına dönüştürülmesi ve bu kodların steganografik yapı ile işlenmesini önermişlerdir [48]. Sağlık sektöründe kullanılmak üzere dijital veri ihlalinin önlenmesi için IoT ile beraber steganografik bir LSB yöntemini önermişlerdir [50]. Sağlık sektöründe kullanılan çeşitli teknikler arasında blok zinciri, akıllı kontratlar, yayılım spektrumu görüntüsü, QR kodları, gelişmiş şifreleme standartları ve IoT gibi yeni teknolojiler yer almaktadır [51]. Bu teknolojilerin steganografi ile entegrasyonu, hasta verilerinin güvenliğini sağlamak ve yetkisiz erişimleri engellemek için önemli bir adım olarak görülmektedir. Steganografi, tıbbi verilerin gizliliğini korumak ve bu verilerin güvenli bir şekilde saklanmasını sağlamak için sağlık sektöründe yaygın bir şekilde kullanılan bir araç olmuştur. Bu çalışmaların ortak amacı, hasta verilerinin güvenliğini artırmak ve bu verilere yetkisiz erişimleri engellemek üzerinedir. Bu projenin geliştirilmesiyle sağlık alanında gereken güvenlik seviyesi sağlanabilir.

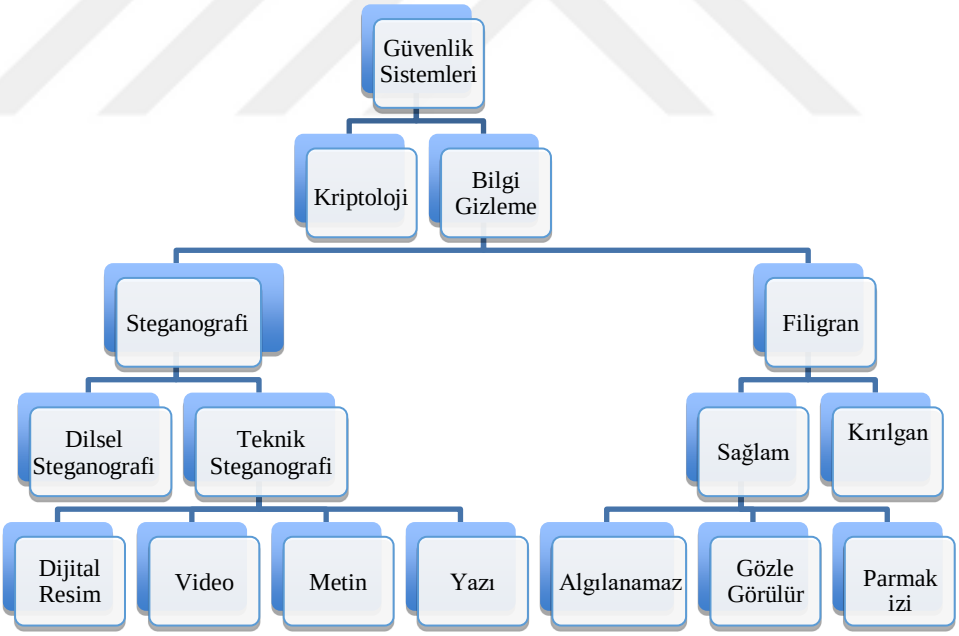
2. MATERYAL VE YÖNTEM

Tezin bu kısmında konuyla bağlantılı olan steganografi, en az öneme sahip bit (LSB), MSE, PSNR, streamlit, IPFS, web arayüzü, metamask, hash, Python, blok zinciri kavramlarının tanımları yapılacaktır. İhtiyaç duyulan kısımlar için dijital fotoğraflama ve kodlama verileri işlenecektir.

2.1. Ön Bilgiler ve Temel Kavramlar

2.1.1. Steganografik Yapı ve Bu Yapının Kısa Tarihi

Steganografinin tanımını yapmadan önce, güvenlik sistemlerinin genel bir şemasını paylaşmak istiyorum. Şekil 2. 1' de belirtildiği gibi, güvenlik sistemlerinde birçok yöntem bulunmaktadır. Bu projede, steganografik yöntem olarak teknik steganografnin dijital resim steganografisi kullanılmıştır.



Şekil 2.1. Güvenlik sistemlerinin şematize edilmesi

Günümüz de bilgi güvenliğini sağlamak için sıkça kullanılan iki dijital yöntem vardır. Bunlar; kriptografi ve steganografidir. Kriptografi, mesajların içeriğindeki verileri gizlemekle ilgili iken steganografi ise mesajların varlığını gizlemekle ilgilidir [52]. Steganografi, içinde gizli mesaj veya bilgiler bulunan bir veriyi, alıcıdan başka

kimsenin fark edemeyeceği bir bicimde gönderme sanatıdır. Latincede ‘steganos’ kelimesi ‘görünmeyen’, steganografi de ‘gizlenmiş yazı’ anlamına gelmektedir [53].

Steganografi, eski çağlardan itibaren gizli iletişim için yoğun bir şekilde kullanılmıştır. Yunan tarihçi Heredot' un eserinde, İran' da bulunan bir casusun Pers istilasını Yunanistan' a nasıl ilettiğini anlatmaktadır. Hikayeye göre, casus, kölesinin saçını kazıtmış ve istila uyarısını kafa derisine kazımıştır. Daha sonra yapılacak olan, kölenin saçının yazıyı kapatacak kadar uzamasını beklemek ve köleyi Yunanistan' a göndermektir. Kölenin bilmesi gereken tek şey "kafamı kazıyın" olacaktır. Yine aynı kaynakta, avcı kılığında bir ulağın, avladığı hayvanın karnına parşömen saklayarak Yunanistan'a girmesi anlatılmaktadır. Steganografinin kullanımı sadece Yunanistan ile sınırlı değildir. Çin kaynaklarında da meyve sepetlerinin gizli iletişim için nasıl kullanıldığı anlatılmaktadır. Meyve sepetindeki her meyvenin birbirine göre pozisyonu farklı bir anlam ifade eder. Antik dönemdeki bu basit uygulamalar, steganografinin gizli iletişimdeki insanlık kadar eski olduğunu göstermektedir. Ancak, çarpıcı kullanımı ikinci dünya savaşı sırasında ortaya çıkmıştır. Alman casuslar, kimyasal bir maddeyle beyaz bir mendile yazdıkları bazı yöntemlerle keşfedilmiştir. Casuslar, gizli mesaj içeren bu mendili belirli noktalara atmakta; alıcı ise yine kimyasal maddeler kullanarak bu mesajı okumaktadır. Ayrıca, ikinci dünya savaşı sırasında Almanlar "mikrofilm" teknolojisi kullanarak "mikro noktalar" (microdot) adı verilen yöntemi geliştirmişlerdir. Bu teknikte, A4 büyüklüğündeki herhangi bir belge veya çizim, bir dizi işlemden sonra daktilo yazısında kullanılan bir noktaya kadar küçültülmektedir. Bu yöntemle, masum içerikli bir sayfa içerisindeki i ve j harflerinin noktalarına oldukça büyük miktarda veri saklamak mümkün olmuştur [54]. Steganografik yapı oluşturmak için eski ve geleneksel yöntemlerden ve günümüz dijital yöntemlerinden faydalanılmıştır. Biz, steganografik yapımızı oluşturup güçlendirmek için tamamen güncel ve dijital yöntemler kullandık. Hakkında bilgi verilen steganografik tekniklerin genel tarihsel süreci Çizelge 2. 1' de gösterilmiştir.

Çizelge 2 1. Steganografinin tarihsel süreci

M.Ö 400 Antik Çağlar	Antik Yunan'da kölelerin saçlarının kazınıp, saç derisine mesajın yazılması, kölenin saçları uzayıp varacağı yere gitmesi ve saçların tekrar kazınması [38] Antik Yunan'da balmumu kaplı tabletlerin kullanımı [39]. Antik Çin' de meyve sepetinin kullanımı meyve sepetindeki her meyvenin birbirine göre pozisyonu farklı bir anlam ifade etmektedir [40].
1640	Gaspar Schott' un müzik notaları ile bilgileri kodlaması [39].
1920' ye kadar	Görünmez mürekkeplerin kullanımı ilk olarak I. dünya savaşında kullanılmıştır. I. ve II. dünya savaşları sırasında Semagram' ların kullanımı [40].
1860-1950	I. ve II. dünya savaşları sırasında microdot' ların kullanımı [41]
Günümüz	Dijital çağda,sayısal (dijital) nesneler üzerinde steganografi uygulamaları yapılmaktadır ve gelişen teknoloji nedeniyle, verilerimizi korumak amacıyla son yıllarda sıklıkla kullanılmaya başlanmıştır. Gizli veri, yine masum içeriğe sahip olan bir dizi dosyanın içinde saklanabilmektedir.
Güçlendirilmiş Steganografi	Projemiz steganografik yapı ile beraber kriptoloji ve blok zincirini de kullanarak benzersiz güvenlikte bir steganografik bir yapı oluşturmaktadır.

Günümüzde, steganografik tekniklerle ilgili literatür incelendiğinde birçok dijital steganografik yöntemin kullanıldığı gözlemlenmektedir [53-57]. Yaygın tekniklerden biri, kapak görüntüsünün en az anlamlı bitlerini (LSB' leri) doğrudan mesaj bitleriyle değiştirerek değişiklik yapmaktır. Bu projede de steganografik yapımızı oluşturmak için LSB tekniğini kullandık.

2.1.2. Bir Resmin Sayısal Yapısı

Sayısal (dijital) resim N satır ve M sütunluk bir dizi ile temsil edilir. Genellikle satır ve sütun indeksleri y ve x veya r ve c olarak gösterilebilir. Bir resim dizisinin elemanlarına piksel denir. En basit durumda pikseller 0 veya 1 değerini alırlar. Bu piksellerden oluşan resimlere ikili (binary) resim denir. 1 ve 0 değerleri sırasıyla aydınlık ve karanlık bölgeleri veya nesne ve zemini (nesnenin önünde veya üzerinde bulunduğu çevre zemini) temsil ederler [58]. Sayısal (dijital) görüntü dosyaları renkli olarak genellikle 24 bit ya da 8 bit değerindedirler (gri-seviye görüntüler 1-2-4- 6 ya da 8 bit olabilirler) . Gizlenmek istenen veri satır ve sütunları verilen resmin herhangi bir satır ve sütunun kesiştiği matris bölgesine gömülür yani kodlanır. Tablo 2. 1' de N satır ve M sütunlu (N x M) bir resmin herhangi bir satır ve sütunun kesiştiği nokta olan y noktası gösterilmiştir. Bu y noktası bir bilginin gizleneceği nokta olarak belirlenebilir.

Tablo 2.1. Sayısal bir resmin temel yapısı

	STUNLAR								
SATIRLAR	1	2	3	.	.	.	.	.	N
	2								
	3								
	.								
	.								
	.								
	.								
	.						y		
	.								
	M								

2.1.3. Görüntü Steganografi Yöntemleri

Steganografi uygulamaları, dışarıdan alınan bir anahtar eklenerek etkili bir gizleme algoritması kullanılarak istenilen verinin taşıyıcı bir nesnenin içine gömülmesini amaçlar. Bu taşıyıcı nesnenin türü ve kalitesi büyük önem taşır. Veri gizleme işlemi, taşıyıcı nesnenin türüne bağlı olarak genellikle üç farklı tekniğe dayanarak gerçekleştirilir, bu teknikler:

I. Uzamsal/Geçici Alan Teknikleri

a. Least Significant Bit (Düşük anlamlı Bit, LSB) kodlama

b. Parity (Eşlik) kodlama

c. Echo (Yankı) kodlama

II. Dönüşüm Alanı Teknikler

a. Spread Spectrum (Tayf Yayılması)

b. Faz Kodlama

c. Ayrık Dalga Dönüşümü (Discrete Wavelet Transform, DWT)

d. Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform , DCT)

e. Ayrık Fourier Dönüşümü (Discrete Fourier Transform, DFT)

f. Gürültü/Ton Ekleme

g. Genlik Kodlama

III. Sıkıştırılmış Alan Teknikleri

a. Vektör Miktarı (Vector Quantization)

b. Fraktal Sıkıştırma (Fractal Compression)

Bu teknikler, her birinin taşıyıcı nesnenin türüne göre uygulama ve etkililik açısından farklılık gösterdiği çeşitli yöntemleri içerir [59]. Biz bu projede uzamsal teknikler arasında yer alan En Önemsiz Bite Ekleme (Least Significant Bit, LSB) yöntemini kullandık.

2.1.3.1. En Önemsiz Bite Ekleme (LSB) Yöntemi

En önemsiz bite ekleme yöntemi (Least Significant Bit Insertion Methods) yaygın olarak kullanılan ve uygulaması basit bir yöntemdir. Fakat yöntemin dikkatsizce uygulanması durumunda veri kayıpları ortaya çıkmaktadır. Bu yöntemde; resmi oluşturan her pikselin her biti' nin en önemsiz biti olan son biti değiştirilerek o bitin yerine gizlenmesini istediğimiz verinin bitleri sırasıyla verinin başlangıcından itibaren birer birer yerleştirilmektedir. Burada her sekiz bitin en fazla bir biti değişikliğe uğratıldığından ve eğer değişiklik olmuşsa da değişiklik yapılan bitin en az anlamlı biti olmasından dolayı, ortaya çıkan steganogramdaki (örtü verisi + gömülü veri) değişimler insan tarafından algılanamaz boyutta olmaktadır.

Bazı steganografik sistemler bazı gizli anahtarlar da kullanabilmektedir. Bu anahtarlar da ikiye ayrılırlar: 1. Anahtar, steganografik anahtarlar; mesajı resmin içine

gizleme ve tekrar elde etme işlemini kontrol etmek için kullanılırlar. 2. Anahtar, Kriptografik anahtarlar; Mesajın resmin içine gizlenmeden önce şifrelenmesi ve daha sonra şifrenin çözümlenmesinde kullanılırlar [60].

2.1.3.1.1. Gri Seviye Resimler ve LSB Yönteminin Uygulanması

Gri seviye resimlerde her piksel, 0 (siyah) ile 255 (beyaz) arasında tam sayı değeri alabilen 1 bit ile temsil edilmektedirler. 0-255 arasındaki değerler gridir ve bundan dolayı bir resme ait tam sayı "gri ton seviye" (gray level) olarak isimlendirilmektedir. Örneğin, renk değeri 182 olan bir pikselin içine ikilik sayı sistemindeki 1 değeri saklandığın da oluşan piksel ve renk değeri aşağıda gösterilmektedir.

Çizelge 2.2. .Piksel eklenmiş bir resimdeki farklılık detayının gösterilmesi

	Renk değeri	İkili sistemdeki karşılığı	Rengi
Orijinal piksel	182	10110110	
Bilgi saklanmış pikseli	183	10110111	

Çizelge 2. 2' deki renklerden de görüleceği gibi iki renk arasında gözle fark edilemeyecek kadar az bir değişim vardır. Son bitin 1 ya da 0 olması gözle görülebilir bir fark yaratmamaktadır [61].

2.1.3.1.2. 8 Bit Renkli Resimler ve LSB Yönteminin Uygulanması

8 bitlik görüntülerde piksel başına 1 bit kullanılır. 8 bitlik görüntüler renk sınırlaması yüzünden çok iyi bir sonuç vermemektedir. Saklanacak bilgi, saklama ortamını çok fazla değiştirmeyecek şekilde dikkatlice seçilmelidir. Orijinal görüntüde son bite ekleme işlemi yapıldığında, renk girişi göstergeleri değişmektedir. 8 bitlik görüntülerde 4 basit renk (WRBG) kullanılmaktadır. Bunlar; beyaz (White-W), kırmızı (Red-R), mavi (Blue-B) ve yeşildir (Green-G). Bu renklerin renk paletinde karşılık gelen girişleri ise sırasıyla 0 (00), 1 (01), 2 (10), 3 (11) şeklindedir. Örnek olarak verilen orijinal görüntü pikselleri "Beyaz, beyaz, mavi, mavi" (00 00 10 10) ise 10 sayısının ikilik (binary) tabandaki karşılığı olan 1010 değeri bu piksellere gizlendiğinde, yapılan değişiklikler sonucunda görüntünün yeni piksel değerleri aşağıdaki gibi elde edilmektedir. 01 00 11 10 Bu değerler de renk paletinde sırasıyla kırmızı, beyaz, yeşil ve mavi değerlerine karşılık gelmektedir [62].

Piksellerin renk değerleri oldukça değiştiğinden, gözle fark edilebilecektir ve bu kabul edilemez bir durumdur. Veri gizleme uzmanları bu nedenle 8 bitlik renkli görüntüler yerine gri-seviye görüntülerin kullanılmasını daha uygun bulmaktadırlar [62].

2.1.3.1.3. 24 Bit Renkli Resimler ve LSB Yönteminin Uygulanması

24 bitlik resimler, her piksel başına 3 bit kullanarak kırmızı (red), yeşil (green) ve mavi (blue) olmak üzere üç ana renkten oluşur. Bu pikselin RGB değeri olarak adlandırılır. Her bir bite bir pikselde 3 bitlik bilgi saklanabilir. Örneğin, 24 bit derinliğindeki, 1024x768 piksel boyutundaki bir resim, bilgi saklamak için 2.359.296 bit (294.912 byte) kapasiteye sahiptir. Gizlenmek istenen mesaj sıkıştırılmadan önce, resim içine saklanabilir; bu sayede daha fazla bilgi gizlenebilir. Ancak, en iyi sonuçları elde etmek için resim sıkıştırılmadan işlem yapılması gereklidir.
Aşağıdaki örnekte, 3 pikselin içine '101101101' bilgisi gizlendiğinde, oluşan yeni piksel değerleri şu şekilde değişir:
- Özgün Pikseller: (149,13,201), (150,15,202), (159,16,234)
- Gizlenen Bilgi Eklenmiş Pikseller: (149,12,201), (151,14,203), (159,16,234)

Bu yöntemle, sadece 4 bitten az bir değişiklikle bilgi gizlenebilmektedir. Ancak, gizlenecek bilgi az ise hangi bitlerin ihmal edileceğini belirlemek oldukça önemlidir. 24 bitlik resimlerde gizlenen bilginin varlığı neredeyse algılanamaz. Resim 2. 1' de, uygulamadan önceki 3 farklı resmin görüntüsü verilmiştir. Resim 2. 2' de ise resimlerin uygulamadan sonraki halleri gösterilmiştir. Her iki durumda da LSB yöntemi kullanılarak oluşturulan 24 bitlik resimler, görselde belirgin bir farklılık göstermez [62]. Bu projede, gizlenecek mesajların sadece 24 bitlik görsellerde kullanılması önerilmiştir.

Resim 2.1. LSB yöntemi uygulanmadan önce resimlerin görünümü

Resim 2.2. LSB uygulandıktan sonra 24 bit değerinde resimlerin görünümü.

2.1.3.2. PSNR ve MSE

Steganografi yöntemlerinin etkinliği, gizli bilgiyi taşıyıcı veriye ne kadar etkili bir şekilde gizleyebildiği ve steganaliz yöntemlerine karşı direnciyle değerlendirilir. Steganaliz yöntemleri, ilk olarak bir taşıyıcı veride bilgi gizlenip gizlenmediğini tespit etmeye çalışır; bu adım sezme (detection) olarak adlandırılır. Daha sonra, var olduğu tespit edilen gizli bilgiyi çıkarmak için çıkarma (extraction) süreci gerçekleştirilir. Bu iki süreç, steganografi ile steganaliz arasındaki sürekli bir mücadelede rol oynar ve steganografi yöntemlerinin güvenilirliği ve etkinliği bu denge üzerine kurulur. LSB yöntemiyle gizleme işlemi yapıldığında, sezme işleminin başarısız olabilmesi için kullanılan resimler üzerinde gerçekleşen işlemlerin en az düzeyde tutulması önemlidir. Bu sebeple, örtü verisi ile gizli veriden sonra elde edilen stego-imajı veya şifrelenmiş veriyi değerlendirmek için ortalama karesel hata (Mean Squared Error, MSE) ve Peak Signal to Noise Ratio (PSNR) hesaplamaları yapılabilir. Ortalama karesel hata, gizli bilginin ne kadar başarılı bir şekilde gizlendiğini belirtir; yani ne kadar düşükse, elde edilen bilginin güvenliği o kadar yüksektir. PSNR ise, MSE' ye bağlı bir değişkendir ve daha yüksek bir PSNR değeri, kullanılan görüntünün yapısal sağlamlığının arttığını gösterir. “Denklem 2. 1” de MSE denkleminin genel tanımı verilmiştir [63].

$$\boldsymbol{MSE = \frac{1}{MN} \sum \; [A_1(m,n) - A_2\,(m,n)]^2} \qquad \textbf{(2.1)}$$

"Denklem 2. 1", A_1 ve A_2 ile örtü imgesi (resmi) ve stego yani bilginin saklandığı imgeyi (resmi) sırasıyla temsil etmektedir. Burada M ve N, imgenin boyutlarını ifade etmektedir. m ve n ise birim genin sahip olduğu sayısal yapıdaki satır ve sütunu temsil

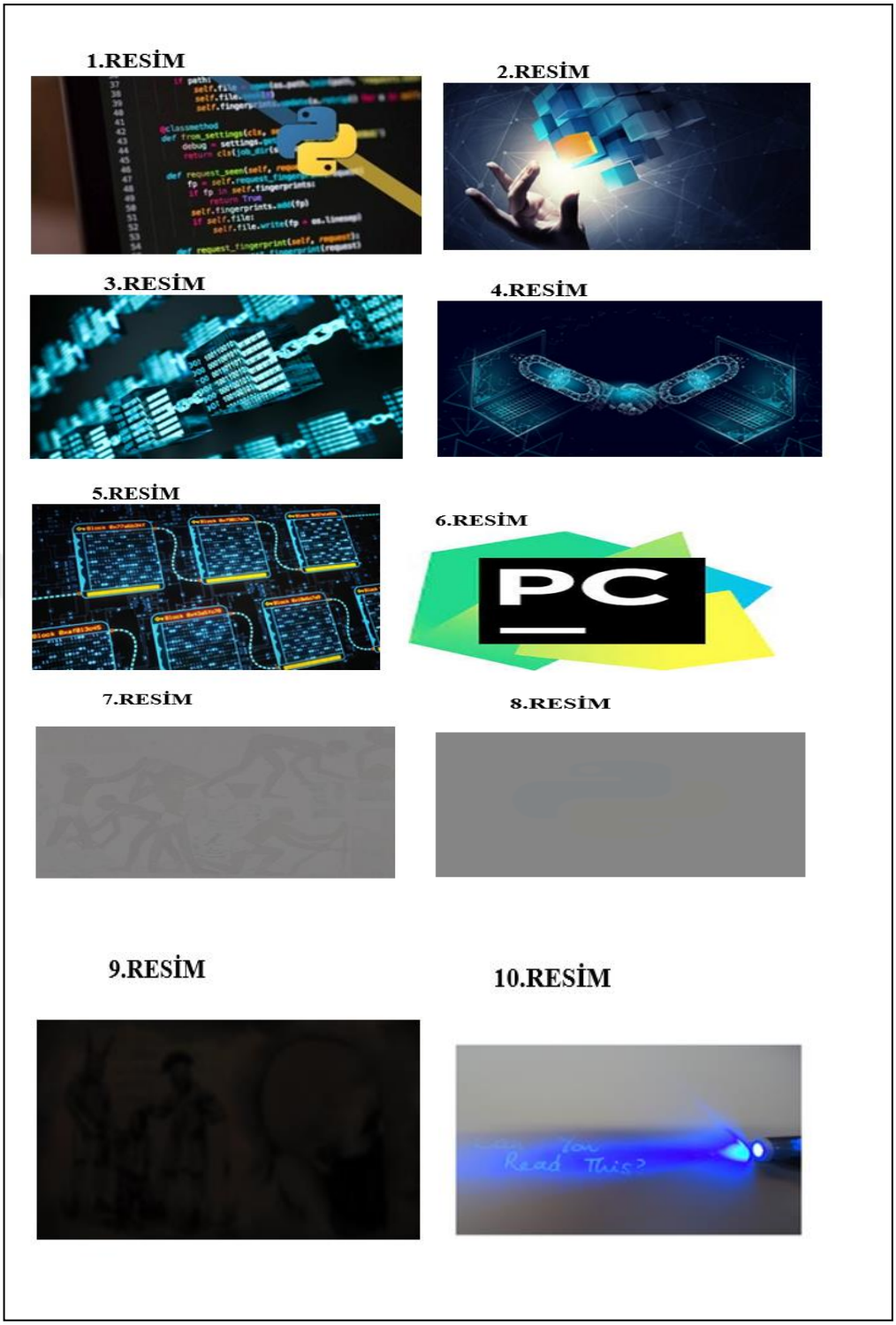
etmektedir. R ise bir pikselin alabileceği maksimum değeri göstermektedir. MSE değeri, steganografik yapıda kullanılan yöntemin güvenirliği arttıkça düşer. Genellikle MSE değeri 0' a yakın olmalıdır. Belirlenen eşik değerinden yüksek bir MSE değeri elde edilmesi durumunda, yapıların güvenilirliği sorgulanabilir hale gelir. "Denklem 2. 2" de PSNR' nin genel denklemini vermektedir [63].

$$\boldsymbol{PSNR = 10.log_{10}\frac{R^2}{MSE}} \qquad \textit{(2.2)}$$

Steganalizde, sezme işlemi sırasında kullanılan yöntemlerden PSNR değeri, gizli verinin iyi gizlendiği veya değişikliklerin insan gözü tarafından fark edilemeyecek kadar küçük olduğu anlamına gelir. MSE değerleri düşük (0' a yakın) ve PSNR değerleri yüksek çıkıyorsa, bu durum gizli verinin iyi saklandığı ve değişikliklerin görüntüde göze çarpmayacak kadar küçük olduğunu gösterir. Bu, kullanılan steganografik yöntemin güvenilirliği ve gizliliği hakkında olumlu bir gösterge olur [64]. Ancak, belirtmek gerekir ki, yüksek PSNR ve düşük MSE değerleri her zaman güvenliği garanti etmez. Bazı steganaliz teknikleri, insan gözü için fark edilemeyecek değişiklikler yapabilir ve bu da yüksek PSNR değerlerine yol açabilir. Bu yüzden, sadece MSE ve PSNR değerlerine dayanarak bir steganografik yöntemin güvenliği hakkında kesin sonuçlar çıkarmak doğru olmayabilir.

Çizelge 2.3. İnşa edilen yapıda kullanılan (256px) x (135px) boyutların da 10 farklı resmin MSE ve PSNR değerleri

Resim Numarası	MSE değeri	PSNR değeri
1	0.000813	79.0279
2	0.000198	85,1668
3	0.00046	81,4856
4	0.000321	85,0532
5	0.000135	86,8071
6	0.000228	84,5665
7	0.000219	84,7174
8	0.000197	85,1688
9	0.000811	79,0386
10	0.000183	85,4689



Resim 2.3. MSE ve PSNR değerlerini göstermek üzere kullanılan resimler

2.1.4. Python

Python; web uygulamaları, yazılım geliştirme, veri bilimi ve makine öğreniminde Machine Learning (ML) yaygın olarak kullanılan bir programlama dilidir. Geliştiriciler, etkili ve öğrenmesi kolay olduğu ve birçok farklı platformda çalıştırılabildiği için Python'

ı kullanır. Python yazılım dili ücretsiz olarak indirilebilir, her türlü sistemle iyi bütünleşme sağlar ve geliştirme hızını artırır [65].

2.1.4.1. Python Yazılım Dilinin Avantajları

1. Çeşitli uygulama alanları: Python, web geliştirme, veri analizi, yapay zeka, makine öğrenimi, bilgi güvenliği ve daha birçok alanda kullanılabilir. Bu, geliştiricilere geniş bir uygulama yelpazesi sunar.
2. Dinamik tür sistemi: Python, dinamik bir tür sistemine sahiptir, bu da değişken türlerinin çalışma zamanında belirlendiği anlamına gelir. Bu, geliştiricilere daha fazla esneklik sağlar ve tip tanımlamalarını daha az karmaşık hale getirir.
3. Topluluk ve dokümantasyon: Python, büyük bir topluluğa sahiptir ve bu topluluk sürekli olarak dilin gelişimine katkıda bulunur. Ayrıca, Python' ın kapsamlı bir dokümantasyonu bulunmaktadır, bu da geliştiricilerin sorunları çözme ve yeni konseptleri öğrenme sürecini kolaylaştırır.
4. Yüksek seviyeli dil: Python, yüksek seviyeli bir dil olduğu için geliştiricilere karmaşık ayrıntılarla uğraşma zorunluluğunu azaltır. Bu, hızlı prototip oluşturma ve hızlı geliştirme için uygundur.
5. Geniş kütüphane desteği: Python, birçok önceden yazılmış kütüphaneye sahiptir. Bu kütüphaneler, geliştiricilere işlevselliği hızlı bir şekilde ekleyebilmeleri için araçlar sağlar.
6. Geniş platform desteği: Python, farklı platformlarda çalışabilir olmasıyla bilinir. Bu, geliştiricilerin farklı işletim sistemlerinde aynı kodu kullanmalarını ve taşınabilir uygulamalar oluşturmalarını sağlar.
7. Hızlı prototip ve geliştirme: Python, hızlı prototipler oluşturmak ve uygulamaları hızlı bir şekilde geliştirmek için idealdir. Bu, iş süreçlerini hızlandırabilir ve ürünlerin daha hızlı piyasaya sürülmesine olanak tanır.

Python' ın bu avantajları, dilin geniş bir kullanıcı kitlesi ve endüstri tarafından benimsenmesine katkıda bulunmuştur [66]. Python, çoğunlukla kullanmamızın nedeni, hem ücretsiz bir yazılım dili olması hem de binlerce ücretsiz yazılım kütüphanesine sahip olmasıdır.

2.1.5. HTML

(Hyper Text Markup Language), yani Hiper Metin İşaretleme Dili, internet sayfalarının oluşturulmasında kullanılan bir metin işaretleme dilidir. Genellikle bir internet sitesinin iskeleti olarak nitelendirilen HTML, son sürümü HTML5 olan bir dildir.

Ancak dinamik işlevsellik oluşturamadığı için doğrudan bir programlama dili olarak tanımlanmaz. HTML, içeriğin tarayıcıda nasıl görüntüleneceğine dair bilgileri sağlayan bir dizi etiketten oluşur. Aynı zamanda, resmi bir web standardı olarak kabul edilir ve World Wide Web Consortium (W3C), HTML için düzenli güncellemeler sunarak HTML özelliklerini geliştirir ve korur [67]. HTML sayfaları, içerdikleri etiketler aracılığıyla tarayıcılar (örneğin; Google Chrome, Opera, Firefox gibi) tarafından görüntülenir. Tarayıcılar, HTML kodlarını insanların anlayabileceği bir forma çevirerek görüntülerler. Bu projede, steganografik yapının kullanıcı dostu arayüzü için ücretsiz ve güvenli HTML etiketleri kullanılarak, kullanıcıların erişimini kolaylaştırmak amaçlanmıştır.

2.1.6. Java Script (JS)

Java Script, geliştiricilerin etkileşimli web sayfaları oluşturmak için kullandığı bir programlama dilidir. Bu dil, sosyal medya akışlarını güncellemekten, animasyonlar ve etkileşimli haritalar göstermeye kadar birçok işlevi yerine getirerek web sitesi kullanıcılarının deneyimini geliştirebilir. İstemci tarafında çalışan bir betik dili olan JavaScript, World Wide Web' in temel teknolojilerinden biridir. Örneğin, internet üzerinde gezinirken bir görsel döngü, tıklanabilir açılır menü veya bir web sayfasında dinamik olarak değişen renkler gibi özellikleri gördüğünüzde, bu özellikler genellikle JavaScript' in etkilerini yansıtır [67]. Projemizde, renk akışları, sunucu desteği veya metin boyutlandırma gibi birçok bölümde, ücretsiz bir yazılım dili olan JS' nin desteğini kullandık.

2.1.7. Blok Zinciri Yapısı ve Bu Projede ki Önemi

Blok zinciri teknolojisinin günümüzdeki önemi ve bu projedeki kulanım amaç ve gerekliliği başlıklar şeklinde açıklanmıştır.

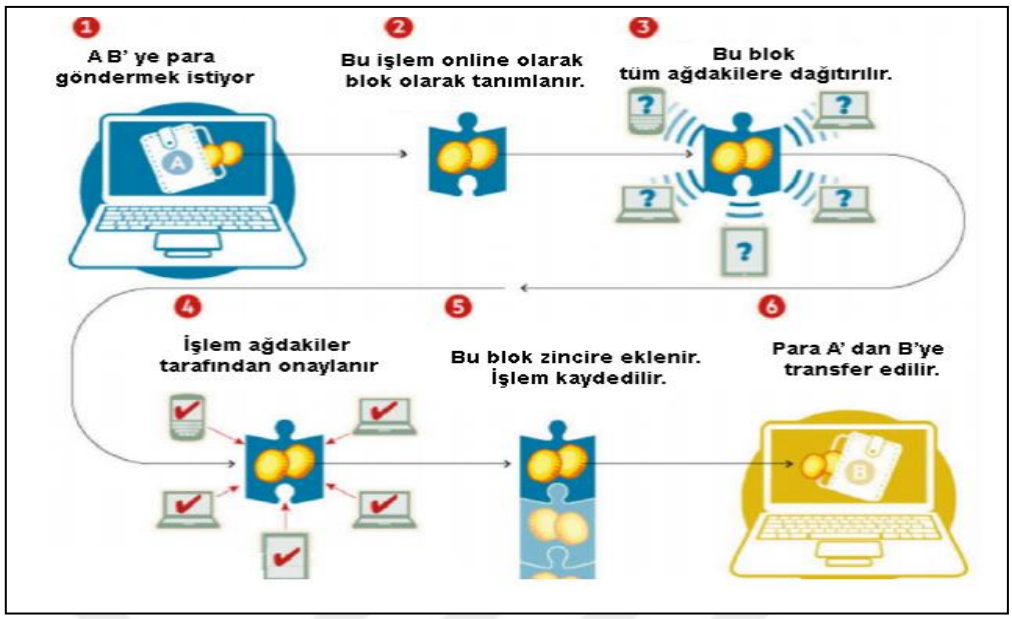
2.1.7.1. Bitcoin

Bitcoin, 2009 yılında piyasaya sürülen bir dijital para birimidir. Merkezi olmayan bir yapıya sahip olması, taraflar arasında değer aktarımını kolaylaştırmak için üçüncü taraf aracılara gerek duymamasıyla öne çıkar. Kendi değer birimine sahip olan Bitcoin, işlemlerin depolandığı ve kaydedildiği bir düğüm ağı olan blok zinciri teknolojisi üzerine kuruludur [57]. Blok zinciri, her Bitcoin' in geçmişini kaydeder ve işlemler bloklar halinde eklenir. Bu işlem, 'madencilik' olarak adlandırılan ve Nakamoto Konsensüsü adı verilen bir iş kanıtıyla gerçekleştirilir. Bu süreç, merkezi bir otoriteye ihtiyaç duymadan işlemleri doğrulayan ve herkesin erişebildiği açık bir ödeme sistemi sağlar. Bitcoin ağının

dayanıklılığı, merkezi bir arıza noktasının bulunmamasını ve Nakamoto Konsensüsü sayesinde tüm düğümlerin durumu konusunda anlaşmaya varmasını sağlar. Sonuç olarak, Bitcoin izin gerektirmeyen, kriptografik olarak güvenli ve merkezi olmayan bir ağdır [57].

2.1.7.2. Ethereum

Ethereum, açık bir blok zinciri platformudur ve merkezi olmayan uygulamaların geliştirilmesine imkan tanır. Bu platform, eşler arasında koordinasyonu ve otomatikleştirilmiş uygulamaları destekler. Bitcoin gibi tamamen özelleştirilebilir bir ödeme mantığına sahiptir ve üçüncü taraflara bağımlı olmadan ödeme sistemlerinin oluşturulmasına olanak sağlar. Ethereum' un dikkat çeken özellikleri arasında hızlı geliştirme süreci, uygulama güvenliği ve farklı uygulamalar arasında etkileşim yeteneği bulunur. Ethereum, geliştiricilere esneklik sağlayan ve farklı programlama dillerinde çalışabilen bir "Turing complete" yani "Turing tamamlanmış" programlama dilini kullanır [57]. Platform, yüksek düzeyde güvenlik sağlamak amacıyla inşa edilmiştir ve hizmet reddi saldırılarına karşı korunma ve iş kanıtı madenciliğine dayanır. Tüm işlemler Ethereum ağı üzerinde gerçekleşir ve akıllı sözleşme hesaplamaları için Sanal Makine (Ethereum Virtual Machine, EVM) kullanılır. Bu işlem Ethereum' un her düğümünde hesaplamaların fikir birliği ile doğrulanmasını sağlar. Fikir birliği kullanılmasının amacı da oluşabilecek saldırıların önüne geçmektir. Kötü amaçlı yazılım veya kötü amaçlı bireyler sisteme sızsalar bile Ethereum’ un merkeziyetsizliğinden ötürü saldırılardan etkilenme en asgari düzeye indirgenir. Blok zincirinde depolanan veriler istenildiği durumlarda kişisel bilgiler hariç ortak erişime açık olması gereken her şey şeffaf olarak tüm düğümler (node) arasında paylaşılır. Blok zinciri verilerin güvenilir bir şekilde paylaşılması ve istenildiğinde keyfi olarak tek merkez tarafından yönetilmesinin önüne geçilmesi için merkeziyetsiz (decentralization) bir yapıya sahiptir. Geliştirdiğimiz bu uygulama, güvenilirliği artırmak amacıyla blok zinciri teknolojisi ile desteklenmiştir. Şekil 2. 5’ de Blok zinciri ağı ile bir verinin transferi şematize edilmiştir.

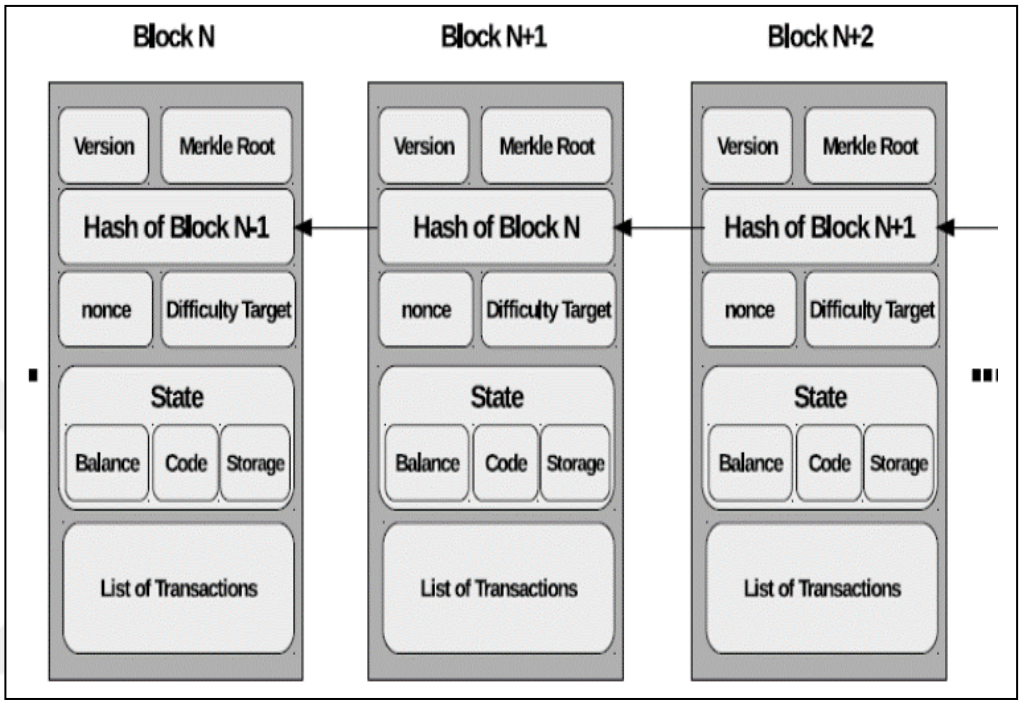


Şekil 2.2. Eşler Arası (Peer to Peer, P2P) veri transferinin blok zinciri ile dağıtımı

2.1.7.2.1. Blok Zincirinin Çalışma Mantığı

1. Veri depolama: Her blok, belirli miktarda veriyi depolar. Bu veriler, finansal işlemler, metinler, sözleşmeler veya herhangi bir dijital bilgi olabilir.
2. Özütleme (hashing): Her blok, içindeki verilerin benzersiz bir özetini oluşturur. Bu işlem kriptografik bir hash fonksiyonu kullanılarak yapılır. Bu özet, bloğun veri bütünlüğünü korur ve bir sonraki bloğa referans sağlar.
3. Zincirleme: Her blok, içerisinde bir önceki bloğun özetini (hash' ini) içerir. Bu, blokların birbirine bağlanmasını ve sıralı bir zincir oluşturulmasını sağlar. Her bir blok, kendisinden önceki bloğun veri yapısına referans yapar.
4. Dağıtık ve dağınık olarak saklama: Blok zinciri verileri, ağdaki birden fazla bilgisayarda (düğümde) saklanır. Bu dağıtık yapı, verilerin merkezi olmayan bir şekilde saklanmasını ve doğrulanmasını sağlar.
5.Konsensüs mekanizması: Yeni blokların eklenmesi, ağdaki kullanıcıların veya düğümlerin bir anlaşmaya varması gereken bir süreçtir. Bu süreç, farklı konsensüs mekanizmalarıyla (Proof of Work, Proof of Stake gibi) gerçekleştirilir ve yeni blokların eklenebilmesi için genellikle matematiksel veya kriptografik problemlerin çözülmesini gerektirir.
6.Değişmezlik ve güvenlik: Blokların birbirine bağlanması ve her bloğun önceki bloğun özetini içermesi, verilerin değiştirilmesini zorlaştırır. Bir bloğun içeriği değiştirilirse, bu

bloğun özeti değişir ve zincirin geri kalanındaki blokların geçerliliği kaybolur. Bu durum, veri değişikliklerini tespit etmeyi kolaylaştırır ve blok zincirin güvenliğini artırır. Bu adımlar, Blok zinciri' nin temel çalışma mantığını oluşturur. Bu teknoloji, güvenli ve şeffaf bir şekilde veri saklama, transfer etme ve doğrulama için kullanılabilir [68]. Resim 2. 4' de Temel Blok zinciri yapısı anlatılmıştır [69].



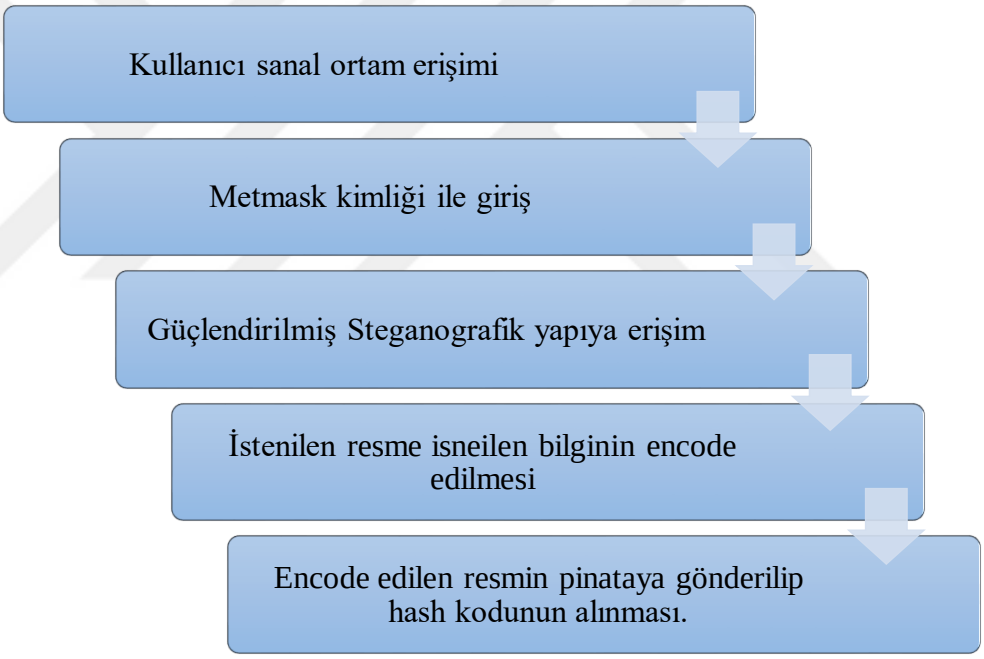
Resim 2.4. Temel blok zinciri yapısı

2.1.8. Pinata ve Ipfs' nin Birlikte Çalışma Mantığı ve Bu Projedeki Önemi

Merkezi olmayan depolama, blok zinciri teknolojisinin gücünden yararlanarak veri depolama için dönüştürücü bir çözüm olarak ortaya çıkmıştır [70]. Merkezi olmayan depolama, verileri tek bir merkezi varlığa dayanmak yerine bir düğüm ağı üzerinden dağıtarak geleneksel merkezi depolama çözümlerine bir alternatif sağlar. Bu yaklaşım, gelişmiş güvenlik, değişmezlik ve şeffaflık gibi avantajlar sunar. Merkezi olmayan depolamayı gerçekleştirmek için Pinata, çeşitli blok zincir ağlarında merkezi olmayan dosya depolama ve alma hizmetleri sunan, yaygın olarak kullanılan bir platformdur. Pinata, kullanıcıları IPFS (Inter planetary File System, Gezegenler arası Dosya Sistemi) gibi merkezi olmayan ağlara bağlayarak ve verilerin bu ağlarda dosya olarak depolanmasını kolaylaştırarak çalışır. Kullanıcılar bir hesap oluşturarak, Pinata' nın API' si ile etkileşim kurabilir, verilerini güvenli bir şekilde saklayabilirler. Kullanıcılar Pinata ağına bağlanmak için metamask cüzdan ile giriş yapılabilir. Metamask cüzdanı Ethereum

blok zincirindeki merkezi olmayan depolama bağlamında çok önemli bir rol oynar. Metamask, merkeziyetsizliğinden dolayı alternatif güvenlik ağlarına oranla daha gelişmiş bir güvenlik ağı oluşturur.

Projeyi kurarken, bireylerin işlemlerini daha güvenli bir ortamda gerçekleştirebilmeleri için çeşitli adımlar attık. Bu adımlar arasında, metamask kimlik girişi ile güvenli kimlik doğrulaması, Pinata aracılığıyla verilerin güvenli bir şekilde depolanması ve kullanıcıların sakladıkları bilgileri güvenli bir hash ortamında dağıtabilmeleri yer alıyor. Bu projede Pinata'nın merkeziyetsiz yapısından ve güvenilir hash kodu oluşturma ortamından faydalandık. Şekil 2. 5' te gösterilen süreçte, steganografik yapıya erişen bir kullanıcının sahip olduğu verinin Pinata ortamına dağıtılması ve Pinata ortamına yüklenen verinin kodlanarak kullanıcıya geri iletilmesi işlemi gösterilmektedir.



Resim 2.5. Kullanıcının hash (CID) koduna erişimi

2.1.9. Streamlit

Streamlit, Python tabanlı bir kütüphane olarak veri bilimcileri ve geliştiriciler için etkileşimli web uygulamaları oluşturmayı oldukça basit hale getirir. Bu kütüphane, hızlı prototipleme imkanı sunarak veri analizi, görselleştirme ve sonuçları paylaşma sürecini hızlandırır. Kullanıcı dostu arayüz bileşenleri sayesinde, karmaşık veri işleme işlemlerini Python betikleri üzerinden kolayca gerçekleştirmeyi sağlar. Python odaklı yapısıyla, Streamlit, veri bilimi ve makine öğrenimi gibi alanlarda kullanılan Python kütüphaneleriyle uyumlu çalışır. Ayrıca, etkileşimli arayüzler oluşturmayı destekleyerek

kullanıcıların veri analizi ve sonuçları interaktif bir şekilde incelemelerine olanak tanır. Ücretsiz ve açık kaynaklı olması da streamliti geniş bir kullanıcı kitlesi tarafından tercih edilir kılar [71]. Etkileşimli ve kolay uygulanabilir olmasından dolayı streamlit ortamını bu projede tercih ettik.

2.2.1. Steganografik Yapının Python Ortamında Geliştirlmesi, Yazılan Kodlar ve Kodların İşlevleri Hakkında Temel Bazı Bilgiler ve Bazı Komutlara Verilen Örnekler

```
try:
from distutils.command.install_egg_info import safe_name
import streamlit.components.v1 as components
import json
import os
from io import BytesIO
from PIL import Image
import streamlit as st
import numpy as np
import.base64



except Exception as e:
print(e)
```

Şekil 2.3. Steganografik yapının python ortamında geliştirlmesi,

yazılan kodlar ve kodların işlevleri hakkında temel bazı bilgiler

distutils.command.install_egg_info modülü: Python paketlerinin kurulumu ve dağıtımı sırasında egg-info dosyalarını işlemek için kullanılır. Bu modülde bulunan “safe_name” fonksiyonu, genellikle paket adlarını normalize etmek için kullanılır. Örneğin, boşlukları alt çizgi (_) karakteriyle değiştirerek veya büyük harfleri küçük harflere dönüştürerek paket adlarını kullanılabilir hâle getirir. Ancak bu fonksiyon genellikle doğrudan kullanıcılar tarafından çağrılmaz, genellikle arka planda paket kurulum işlemleri tarafından otomatik olarak kullanılır [72].

streamlit.components.v1 modülü: Streamlite dış kaynaklı bileşenleri (örneğin, özel JavaScript veya CSS dosyalarını) kullanarak özelleştirilmiş veya gelişmiş

görselleştirmeler eklemek için kullanılır. Bu modül, streamlit uygulamasına harici kaynakları entegre etme ve özelleştirilmiş içerik oluşturma yeteneği sunar. Örneğin, haritalar, interaktif grafikler veya özel veri görselleştirmeleri gibi bileşenleri eklemek için kullanılabilir [72].

Json modülü: JSON (JavaScript Object Notation), verileri depolamak, aktarmak ve paylaşmak için kullanılan bir veri formatıdır. İnsanlar tarafından okunabilir bir metin formatıdır ve birçok programlama dilinde kullanılabilir. Anahtar-değer çiftlerini kullanarak verileri organize eder ve sıklıkla web servislerinde, API' lerde ve veri depolama alanlarında kullanılır. Basit, hafif ve çok amaçlıdır [72].

os (Operating System): Python' da işletim sistemiyle etkileşim kurmak için kullanılır. Bu modül, dosya işlemleri yapmak, dizinleri kontrol etmek, sistem bilgilerini almak ve çalışma dizinini değiştirmek gibi işlemleri gerçekleştirmenize olanak tanır. Özetle, "os" modülü, Python programlarınızın işletim sistemine erişmesini ve sistemle etkileşimde bulunmasını sağlar [72].

BytesIO:, Python' da baytlarla çalışmak için kullanılan bir araçtır. Bu, bellekte bayt verilerini saklamak ve işlemek için kullanılır. Gerçek bir dosyaya ihtiyaç duymadan bayt verileri üzerinde dosya benzeri işlemler yapmamızı sağlar. Örneğin, bayt verilerini okuma, yazma veya işleme gibi işlemleri gerçekleştirebilirsiniz. Bu, baytlarla çalışırken esneklik ve kolaylık sağlar[72].

PIL (Python Imaging Library): Python' da resim işleme için kullanılan bir kütüphanedir. Görüntüleri açma, kaydetme, döndürme, yeniden boyutlandırma gibi işlemleri gerçekleştirmemizi sağlar. Ancak PIL' nin gelişimi durdurulmuştur ve yerine geliştirilmeye devam edilen "pillow" adında bir kütüphane kullanılıyor. Pillow, PIL' in özelliklerini devralmış ve geniş bir görüntü işleme işlevselliği sunmaktadır [72]. Bu projede görsel verileri rahat işleyebilmek için pillow kütüphanesini kullandık.

numpy kütüphanesi: Python' da çok boyutlu dizilerle ve matematiksel işlemlerle çalışmak için kullanılan bir kütüphanedir. Hızlı, etkili ve güçlüdür. Veri analizi, bilimsel hesaplamalar ve matris işlemleri gibi birçok alanda yaygın olarak kullanılır [72]. Bu projede resim yapılarını kullanmak için matris yapılarını işleyip kullanabilmek için bu kütüphaneyi kullandık.

base64 modülü: Verileri ASCII karakter setinde temsil etmek için kullanılan bir kodlama yöntemidir. Binary verileri (örneğin, resimler veya ses dosyaları gibi) okunabilir metin

formatına dönüştürmek veya iletim sırasında metin formatında taşımak için kullanılır. Python' daki "base64" modülü, base64 kodlaması ve kod çözme işlemleri için kullanılır. Örneğin, metni base64 ile kodlamak ve çözmek için bu modülü kullanabilirsiniz[72].

try stunu: try ve except blokları, Python' da hata giderme yönetimi için kullanılır.

try bloğu, hata oluşma olasılığı olan kodların bulunduğu bölümdür. Eğer bu bölümde bir hata meydana gelirse, program çalışmasını durdurmadan except bloğuna geçer.

except bloğu: try bloğunda oluşan hatayı ele alır. Bu blok, hata türüne bağlı olarak özelleştirilmiş işlemler yapmanıza veya hatayı görmezden gelmenize olanak sağlar [72]. Bu projede try- except bloğunu kullanma amacımız oluşabilecek hataları görüp çözüm oluşturabilmektir.

```
STYLE = """
<style>
img {
max-width: 100%;
}
</style>
"""
astyle = """
display: inline;
width: 200px;
height: 40px;
background: #F63366;
padding: 9px;
margin: 8px;
text-align: center;
vertical-align: center;
border-radius: 5px;
color: black;
line-height: 25px;
text-decoration: none;
"""
st.set_page_config(
page_title="Steganografi Bilimine Yeni Bir Boyut",)
```

Şekil 2.4. (devam).

asytpe: Veri analizinde kullanılan bir işlemdir asynic kütüphanesinden çekilir.

shape : Bir numphy fonksiyonudur ve eleman ekleme işi yapar

```
tabs = ["Bilgi Gizleme veya Gizlenen Bilgiyi Elde Etme"]
page = st.sidebar.radio("Sekmeler", tabs)
def mse(imageA, imageB):
err = np.sum((imageA.astype("float") - imageB.astype("float"))
** 2)
err /= float(imageA.shape[0] * imageA.shape[1])
    return err
def get_image_download_link(filename, img):
    buffered = BytesIO()
    img.save(buffered, format="PNG")
    img_str = base64.b64encode(buffered.getvalue()).decode()
    href = '<a href="data:file/png;base64,' +img_str+"'
download='+filename +' style="' + astyle + '" target="_blank">
    return href0
def get_key_download_link(filename, key):
    buffered = BytesIO()
    key.dump(buffered)
    key_str = base64.b64encode(buffered.getvalue()).decode()
    href = '<a href="data:file/pkl;base64,' + key_str+"'
download=' + filename + ' style="' + astyle + '"
target="_blank">Download Key</a>'
    return href
```

Şekil 2.5. (devam).

BytesIO: Değişkenlerle yaptığımız gibi, io modülünün ByteIO işlemlerini kullandığımızda veriler bir bellek içi arabellekte bayt olarak tutulabilir.

dump: Bir değer döndürmez ama verilen değeri istenilen konuma gönderir.

```
def modPix(pix, data):
    datalist = [format(ord(i), '08b') for i in data]
    lendata = len(datalist)
    imdata = iter(pix)
    for i in range(lendata):
        pix =[value for value in imdata.__next__()[:3] +
imdata.__next__()[:3] + imdata.__next__()[:3]]
        for j in range(0, 8):
            if (datalist[i][j] == '0'):
                pix[j] &= ~(1 << 0)
            elif (datalist[i][j] == '1'):
                pix[j] |= (1 << 0)
                 if (i == lendata - 1):
            pix[-1] |= (1 << 0)
              else:
            pix[-1] &= ~(1 << 0)
        pix = tuple(pix)
        yield pix[0:3]  # pixel 1
        yield pix[3:6]  # pixel 2
        yield pix[6:9]  # pixel 3
def encode_enc(newimg, data):
    w = newimg.size[0]
    (x, y) = (0, 0)
    for pixel in modPix(newimg.getdata(), data):
```

Şekil 2.6. (devam).

yield: return ile aynı mantık ile çalışır. Yield fonksiyonu çağrıldığında işlenen değer geri eski yerine gönderilir ve kaldığı yerden çalışmasına devam eder return fonksiyonu çağrıldığında da yield fonkksiyonunun işlevi biter.

```
“Değiştirilmiş pikselleri yeni görüntüye yerleştirme”
newimg.putpixel((x, y), pixel)
        if (x == w - 1) x = 0  y += 1
        else:
            x += 1
“Verileri görüntüye kodlar”
def encode(filename, image, bytes):
    global c1, c2
    data = c1.text_area("Kodlanacak veriyi giriniz",
max_chars=bytes)
    if (c1.button('Encode', key="1")):
        if (len(data) == 0):
            c1.error("Veri boş")
        else:                c2.markdown('#')
            result = "İstenilen Mesaj, Örtü verisine
kodlanmıştır."
            c2.success(result)
            c2.markdown('####')
            c2.markdown("#### Kodlanmış resim")
            c2.markdown('######')
            newimg = image.copy()
            encode_enc(newimg, data)
            c2.image(newimg, channels="BGR")
            filename = 'encoded_' + filename
            image_np = np.array(image)
            newimg_np = np.array(newimg)
            MSE = mse(image_np, newimg_np) msg = "MSE: "
+ str(MSE)
            c2.warning(msg)
            c2.markdown("#")
            c2.markdown(get_image_download_link(filename,
newimg), unsafe_allow_html=True)
```

Şekil 2.7. (devam).

```
def decode(image):
    data = ''
    imgdata = iter(image.getdata())
    while (True):
        pixels = [value for value in imgdata.__next__()[:3] +
imgdata.__next__()[:3] + imgdata.__next__()[:3]]
        binstr = ''
        for i in pixels[:8]:
            if (i % 2 == 0):
                binstr += '0'
            else:
                binstr += '1'
        data += chr(int(binstr, 2))
        if (pixels[-1] % 2 != 0):
            return data
def main():
    global c1, c2, d1, d2
    if page == "Bilgi Gizleme veya Gizlenen Bilgiyi Elde
Etme":
        if st.checkbox("Resim Gizleme ve Gizli Resmi
Çözme"):
            st.sidebar.title("Dijital Görüntü İşleme Projesi")
            md = ("")
            st.sidebar.markdown(md,
unsafe_allow_html=True)
```

Şekil 2.8. (devam).

Verilen resimlerin içine bilgi gizlemek için encode fonksiyonu kuruldu. Encode edilen kodların çözülmesi için decode fonksiyonu kuruldu.

sidebar: Yazılmak istenen değer ve fonksiyonları streamlit sayfanın sol tarafına yazar.

md: Kullanılan href ile istenilen resme ulaşılır

```
        info = """
– Resim Steganografisi """
        fileTypes = ["png", "jpg"]
        fileTypes1 = ["pkl"]
        choice = st.radio('Seçim', ["Encode", "Decode"])
        if (choice == "Encode"):
            c1, c2 = st.columns(2)
            file = c1.file_uploader("Kapak Resmini Yükle",
type=fileTypes, key="fu1")
            show_file = c1.empty()
            if not file:
                show_file.info("Lütfen bir dosya türü yükleyin: " + ",
".join(["png", "jpg"]))
                return
            im = Image.open(BytesIO(file.read()))
            filename = file.name
            w, h = im.size
            bytes = (w * h) // 3
            c1.info("maksimum veri: " + str(bytes) + " Bytes")
            encode(filename, im, bytes)
            content = file.getvalue()
            if isinstance(file, BytesIO):
                show_file.image(file)
            file.close()
```

Şekil 2.9. (devam).

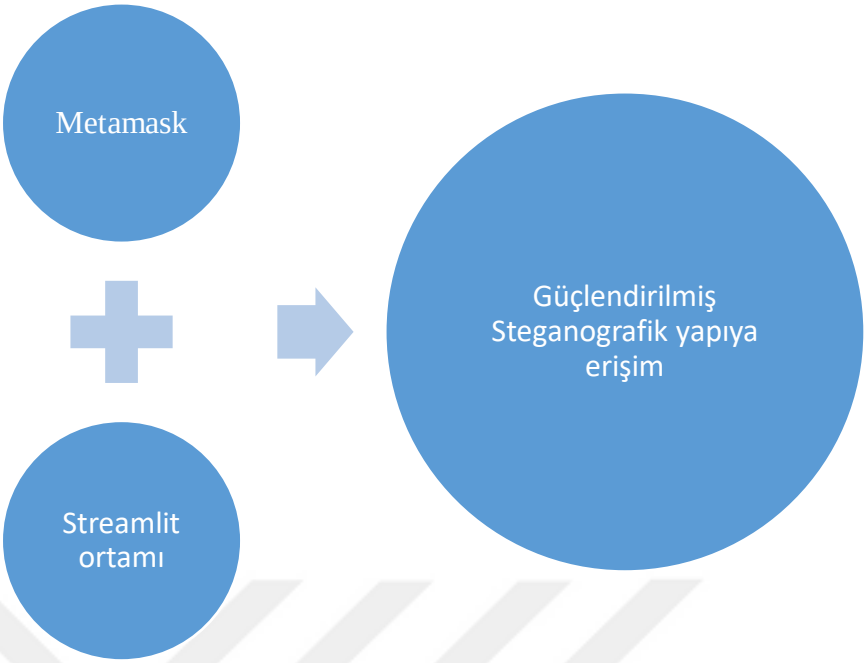
```
    elif (choice == "Decode"):
            file = st.file_uploader("Kodlanmış Resmi Yükle",
type=fileTypes, key="fu2")
            show_file = st.empty()
            if not file:
              show_file.info("Lütfen bir dosya türü yükleyin:
" + ", ".join(["png", "jpg"]))
              return
    im = Image.open(BytesIO(file.read()))
            data = decode(im)
            if (st.button('Decode', key="4")):
              st.subheader("kodu çözülmüş metin")
              st.write(data)
            content = file.getvalue()
            if isinstance(file, BytesIO):
              show_file.image(file)
            file.close()
    if __name__ == "__main__":
main()
```

Şekil 2.10. (devam).

Streamlit üzerinde istenilen butonlarda seçim yapabilmek için "choice" komutu kullanıldı. Streamlit ortamı için geliştirilen python kodlarının kusursuz çalışabilmesi için "main()" fonksiyonu kullanıldı. Yazılan kodların tamamı ücretsiz ve açık kaynaklı bir yazılım dili olan python ortamında yazılmıştır.

2.2.2. Steganografik Ortamın Metamask Dijital Kimlik Doğrulama Ağı ile Etkileşimi

Steganografik yapının streamlit ortamına aktarılmasının ardından, HTML ile bir Metamask kimlik doğrulama giriş sayfası geliştirildi. Geliştirilen metamask doğrulama kimliği, arka plan (backend: kod kısmı) olarak oluşturuldu ve ardından sistem içerisine frontend (sunucunun gördüğü ön yüz) olarak entegre edildi. Metmask ile kimlik doğrulama aşaması şekil 2.11' de şematize edilmiştir.



Şekil 2.11. Steganografik yapıya erişimi sağlayan temel iki öğe

Python ortamında geliştirilen steganografik yapının streamlit üzerinden web ortamına entegre edilmesi, GitHub aracılığıyla gerçekleştirildi. Streamlit ortamındaki erişim linkine ulaşabilmek için Ipfs 'nin masaüstü uygulaması, merkezi olmayan bir CID (Content Identifier) üreterek link erişimi için bir aracı görevi gördü. Sisteme giriş yapmak isteyen kullanıcılar, kullanıcı dostu arayüz üzerinden erişim sağladıktan sonra, yine merkezi olmayan bir yapı olan metamask kimlik doğrulamasını gerçekleştirerek istenilen yapıya erişebileceklerdir. Bu adımlar, sisteme katılımı ve güvenliği sağlamak amacıyla özenle tasarlanmıştır, böylece kullanıcılar istedikleri yapıya güvenli ve doğrulanmış bir şekilde erişebilirler.

2.2.3. HTML Ortamında Geliştirilen Metasmask Hesap Doğrulama Kodu ve Bazı Önemli Kodların Açıklanması

```
!DOCTYPE html>  <html>  <head> <style>
 body {background-color: black;  text-align: center;} #ana {
  background-color: white; padding: 15px; font-weight:bold; }
</style>  <meta  charset="UTF-8"/>  <link  rel="stylesheet"
type="text/html" href="styles.css"/> </head>
  <title> Metamask Giriş Sayfası </title>
 <body >
<button  id="connect-button">  metamaska  Bağlan  <img
src="metamaskicon.png"> </button>
<script >
document.getElementById('connect-
button').addEventListener('click',event=>
 let account;
 ethereum.request({method:
'eth_requestAccounts'}).then(accounts => { account = accounts[0];
 console.log(account);                ethereum.request({method:
'eth_getBalance' ,params: [account, 'latest']}).then(rsult => {
console.log(result);
let wei = parseInt(result,16);
let balance = wei / (10**18);
console.log(balance + "ETH"); });
window.location.href="streamlitapp.com/";});});
</script> <br> </body>
```

Şekil 2.12. Metamask hesap doğrulama kodlarının yazımı

Kullanıcılar giriş yaptıktan sonra ilgili siteye, yani steganografik yapının bulunduğu alana yönlendirileceklerdir. Metamask doğrulaması yapmak isteyen bir kullanıcının, bu işlemi gerçekleştirmek için kullandığı ortamda kesinlikle metamask hesabının yüklü olması gerekmektedir. Bu şekilde bir yol izlememizin temel amacı,

bireylerin güvenliklerini sağlamak ve kullandıkları ortamı güvenli bir şekilde kullanabilmelerini sağlamaktır. Metamask ile bağlantı kurmak için yazılan HTML kodları, ücretsiz ve güvenilir Ethereum dokümantasyonundan elde edilen bilgilerle oluşturulmuştur. Bu adımlar, kullanıcıların güvenli bir şekilde doğrulama yapmasını ve steganografik yapıya güvenle erişmesini sağlamak amacıyla özenle tasarlanmıştır.

2.2.4. Steganografik Yapıda Bir Resmin İçine Gizli Bilginin İşlenmesi ve Şifreli Bilginin Gizlendiği Resmin İstenen Adreslere İletilebilmesi İçin Kullanılan Ipfs Kodları ve Bu Kodlar İçin Bazı Önemli Noktaların Açıklanması

```
        Def decode_from_pinata(ipfs_hash):
            image      =fetch_from_pinata(ipfs_hash)
            if,image:
                data=decode(image)
                st.subheader("kodu_çözülmüş_metin”.)
                st.write(data)
                st.image(image,channels="BGR")
            else:
                st.error("Hatalı Hash Kodu Girdiniz Lütfen Doğru Kodu
        Girdiğinizden Emin Olun.")
def upload_to_pinata(image):
    img_buffer = BytesIO()
    image.save(img_buffer, format="PNG")
    img_bytes = img_buffer.getvalue()
    files = {'file': ('image.png', img_bytes)}
    headers = {
        'pinata_api_key': PINATA_API_KEY,
        'pinata_secret_api_key': PINATA_SECRET_KEY}
    response = requests.post(PINATA_ENDPOINT, files=files,
headers=headers)
    if response.status_code == 200:
        pinata_response = response.json()
        return pinata_response['IpfsHash']
    else: return None
def fetch_from_pinata(ipfs_hash):
    ipfs_url = f"{PINATA_GATEWAY_PREFIX}{ipfs_hash}"
    response = requests.get(ipfs_url)
    if response.status_code == 200:
        img_buffer = BytesIO(response.content)
        image = Image.open(img_buffer)
        return image
    else: return None
```

Şekil 2.13. Pinata' ya erişim için inşa edilen kodların yazımı

Oluşturulan projede streamlit ortamında görünen steganografik yapının içinde encode edilen bilgi bir hash fonksiyonu sayesinde özütlenir. Hash kodu 46 karakterli anlamsız görünen bir yapıya sahiptir. Özütlenen veri pinata ortamında güvenli bir şekilde saklanır ve kullanıcı istediği adrese bu hash kodunu kopyalayarak gönderebilir. Hash kodunun en büyük avantajı günlük hayatta her kesin görmesi fakat içeriğini asla görememesidir. Hash koduna sahip olan bir kullanıcı gizli bilgiye erişmek için yine aynı şekilde sisteme metamsk hesabı ile giriş yaptıktan sonra sisteme hash kodunu yükler ve istediği bilgiye sahip olur. Hash kodunun anlamsız görünmesi ve dikkat çekmeme özelliğinden dolayı günlük dijital iletişimde sıkça kullanılabilir bir yöntem halini alabilir.

3. BULGULAR VE TARTIŞMA

Hazırlamış olduğumuz bu tez çalışmasının sağlıklı bir şekilde işleyebilmesi için üç aşama üzerine kurulu bir entegrasyon kurduk. Her bir aşama, kendi işlevselliği ve özellikleriyle bu yapıya önemli katkılar sağlar. Bu aşamaların entegrasyonu, sistemde güvenilirlik, bütünlük ve etkinlik sağlayarak steganografik yapının başarılı bir şekilde çalışmasını temin eder. Bahsi geçen aşamalar başlıklar halinde anlatılmıştır.

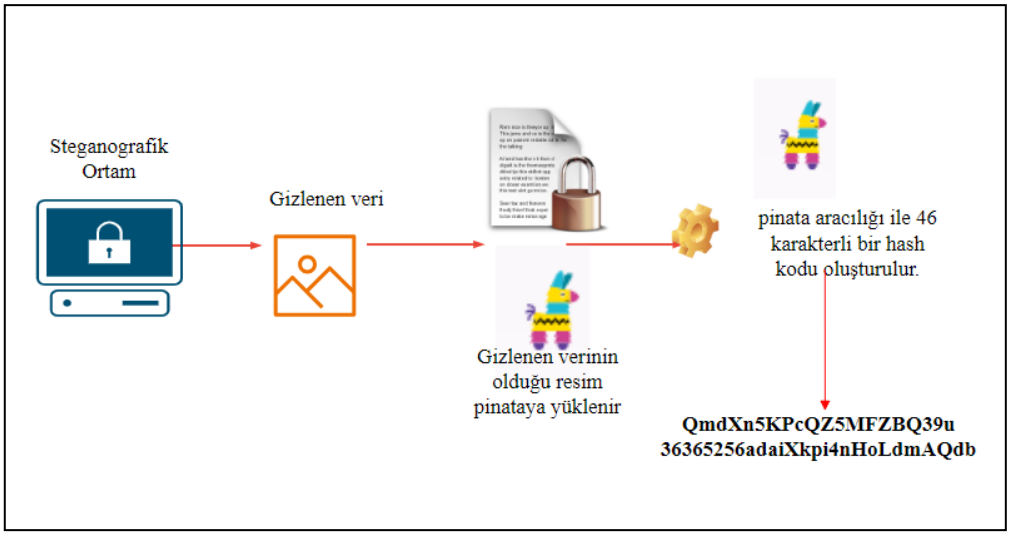
1. Aşama: Python ile LSB ve Diğer Algoritmaların Kullanımı

Bu çalışmada, Python programlama dilinin LSB ve diğer gizli mesaj gömme algoritmalarının uygulanmasında nasıl kullanılabileceğini araştırmıştık. LSB, steganografi alanında en yaygın kullanılan tekniklerden biridir ve verileri resim dosyalarının en az önemli bitlerine gömmeyi amaçlar. Python programlama dili ile bu algoritmanın uygulanması oldukça kolaydır. Gerekli kütüphanelerin kullanılmasıyla görüntü dosyalarının piksellerine erişebilir ve istenilen mesajı en az önemli bitlere gömebiliriz. Bu yöntem, resim dosyalarının değişmeden kalmasını sağlarken, gizli mesajın da içine yerleştirilmesini mümkün kılar.

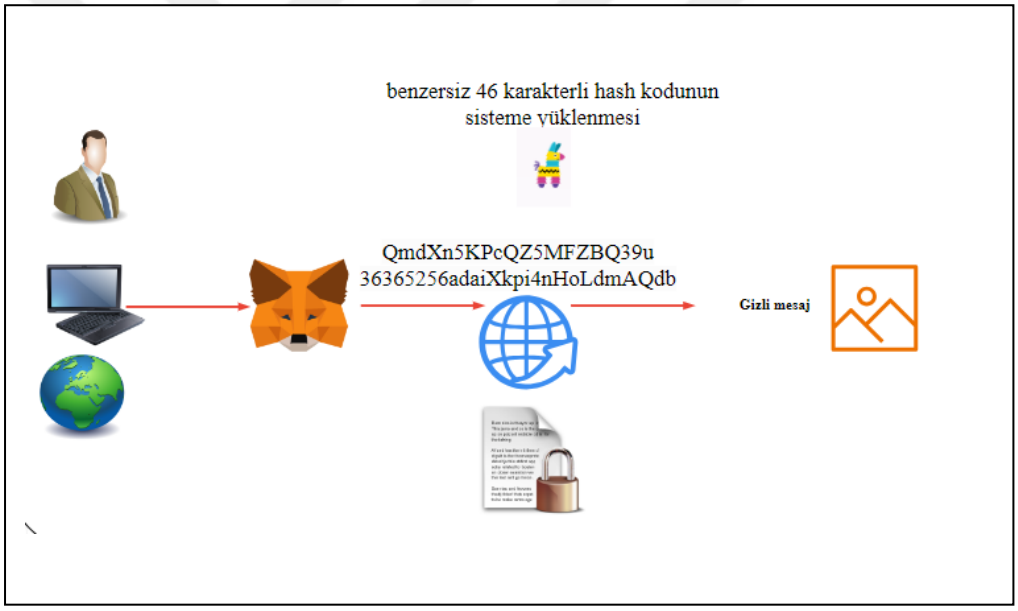
Diğer algoritmaların kullanımı: MSE, PSNR gibi algoritmalar, gizli mesajın gömülmesi ve çıkarılması sırasında veri bütünlüğünü ve kalitesini değerlendirmede kullanılır. Python dilinde bu algoritmaların uygulanması, gömülü mesajın kalitesini analiz etmekte ve çıkarılan mesajın orijinal mesaja olan benzerliğini değerlendirmede oldukça faydalı olduğundan bu projede bunlara yer verilmiştir.

2. Aşama: Blok Zinciri, Ipfs, Pinata ve Metamask Entegrasyonu

Geliştirdiğimiz proje, blok zinciri, Ipfs, Pinata ve metamask gibi farklı teknolojilerin entegrasyonunu içermektedir. Bu teknolojilerin bir araya gelmesi, veri güvenliği ve bütünlüğü açısından yeni olanaklar sunmaktadır. Blok zinciri, Ipfs, Pinata ve metamask gibi teknolojilerin birlikte kullanımıyla verilerin güvenliği ve gizliliği artarken, kullanıcıların veri depolama ve paylaşım süreçlerinde daha kolay ve güvenli bir deneyim elde etmelerine yardımcı olur. Şekil 3. 3' te, steganografik yapıya erişen bir kullanıcının veriyi Pinata' ya gönderme süreci şematik olarak gösterilmiştir. Şekil 3. 4' te ise Pinata ortamında saklanan bir verinin hash kodu olarak sisteme dahil edilmesi ve sistemde istenilen gizli verinin ilgili resimden elde edilme süreci şematik olarak gösterilmiştir.



Şekil 3.1. Streamlit ortamında bilginin bir görselin içine gizlenmesi, bu görselin Pinata ortamına taşınması ve hash kodunun oluşturulması

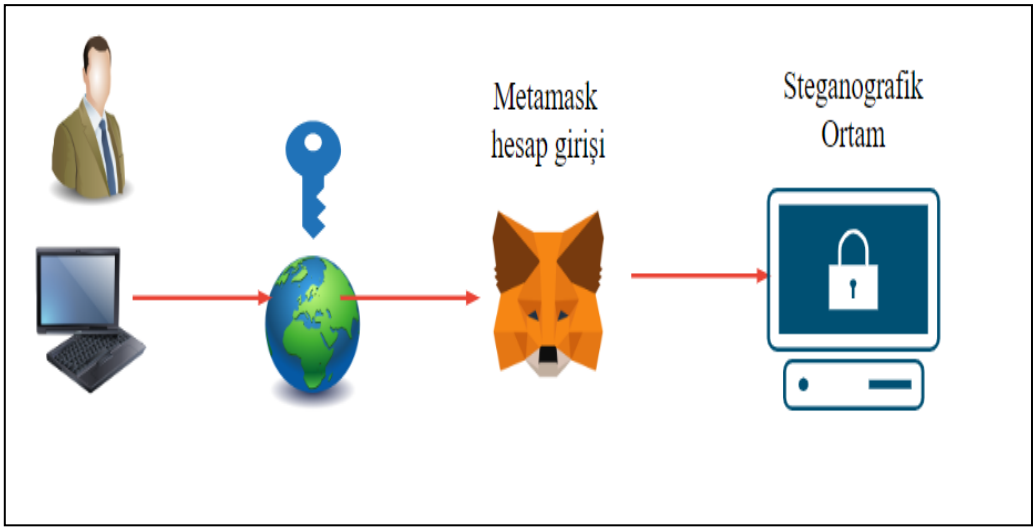


Şekil 3.2. Pinata ortamından sisteme dahil edilen bir hash kodunun ve hash kodundan elde edilen gizli verinin kullanıcıya gösterilmesi

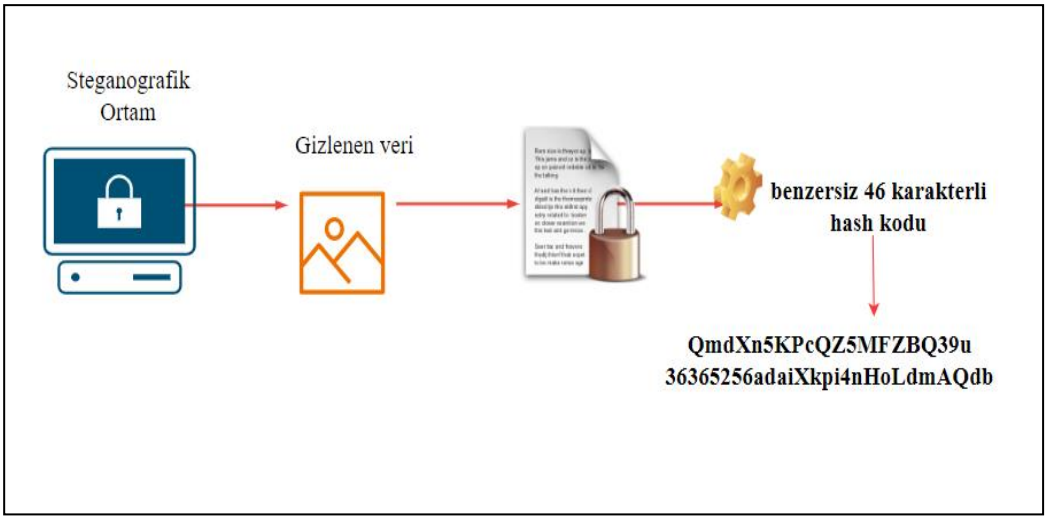
3. Aşama: Entegrasyonun Veri Güvenliği ve Bütünlüğü Üzerindeki Etkileri

Bu tez çalışmasında, blok zinciri, Ipfs, Pinata ve metamask gibi teknolojilerin entegrasyonunun veri güvenliği ve bütünlüğü üzerindeki etkilerini incelemiştik. Blok zinciri, dağıtık ve değiştirilemez yapısıyla verilerin güvenli bir şekilde saklanmasını sağlar. Verilerin bloklar halinde birleştirilmesi ve şifrelenmesi, veri bütünlüğünü korurken, blok zincirindeki dağıtık doğası veri güvenliğini artırır. Özellikle finansal işlemler veya kişisel veriler gibi hassas verilerin blok zinciri üzerinde saklanması, veri

güvenliği açısından büyük bir avantaj sağlar. Ipfs, merkezi olmayan bir dosya paylaşım protokolüdür ve benzersiz içerik karmaları oluşturarak veri parçalarını dağıtık bir şekilde saklar. Bu dağıtık yapısı sayesinde, verilerin bütünlüğü sağlanır ve verilerin değiştirilmesi veya bozulması durumunda bile orijinal veriye erişim mümkün olur. Ipfs' nin bu özelliği, veri bütünlüğünü korurken, verilerin güvenli bir şekilde saklanmasını sağlar. Bu teknolojilerin entegrasyonu, gelecekte veri güvenliği ve bütünlüğü üzerinde daha fazla etki sağlayabilir. Birçok sektörde bu teknolojilerin kullanımı, veri güvenliği standartlarını artırır, veri manipülasyonu gibi riskleri minimize eder. Örneğin, Ipfs üzerinde saklanan bir verinin blok zinciri üzerindeki kaydı, verinin değişikliğe uğramadığını ve bütünlüğünü koruduğunu doğrulayabilir. Bu noktada, steganografinin güvenliği üzerindeki etkileri, gizli mesajların güvenliği ve bütünlüğünün sağlanmasında kritik bir rol oynamaktadır. Kullanılan algoritmaların güvenlik açıkları, veri güvenliği risklerini artırabilir. Bu nedenle, steganografi uygulamalarının geliştirilmesi ve değerlendirilmesi, veri güvenliği standartlarının yükseltilmesine yardımcı olur. steganografik yapının güvenirliğini ve başarılı bir şekilde çalışmasını sağlar. Veri gizleme, erişim ve güvenlik kontrolleri, bu yapı içinde önemli başlıklar oluşturur. Bizde bu projede olabildiğince steganografik yapının en güvenli hale gelmesini sağlamaya çalıştık. Şekil 3. 1' de ve Şekil 3. 2' de bir resmin içine bilgi gizleme ve gizlenen bilgiyi elde etme süreci şematize edilmiştir.



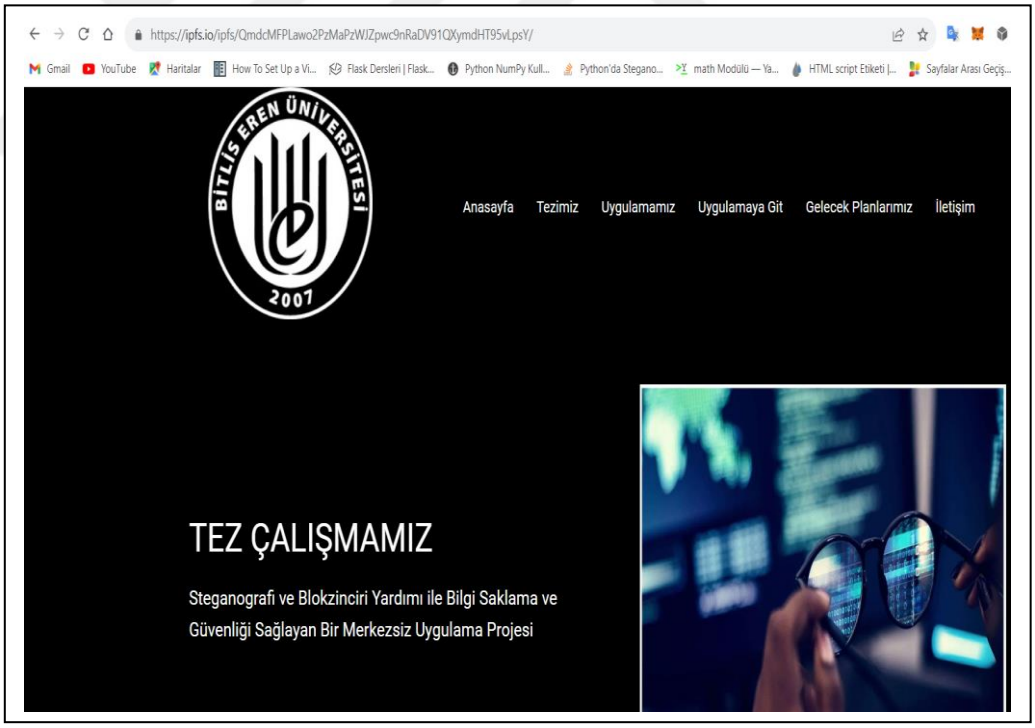
Şekil 3.3. Kullanıcılar internet erişimi olan bir ortamda sisteme dahil olmak için metamask kimlik doğrulaması yaparak ilerler.



Şekil 3.4. Steganografik ortama giriş sağlandıktan sonra benzersiz hash kodunun elde edilmesi

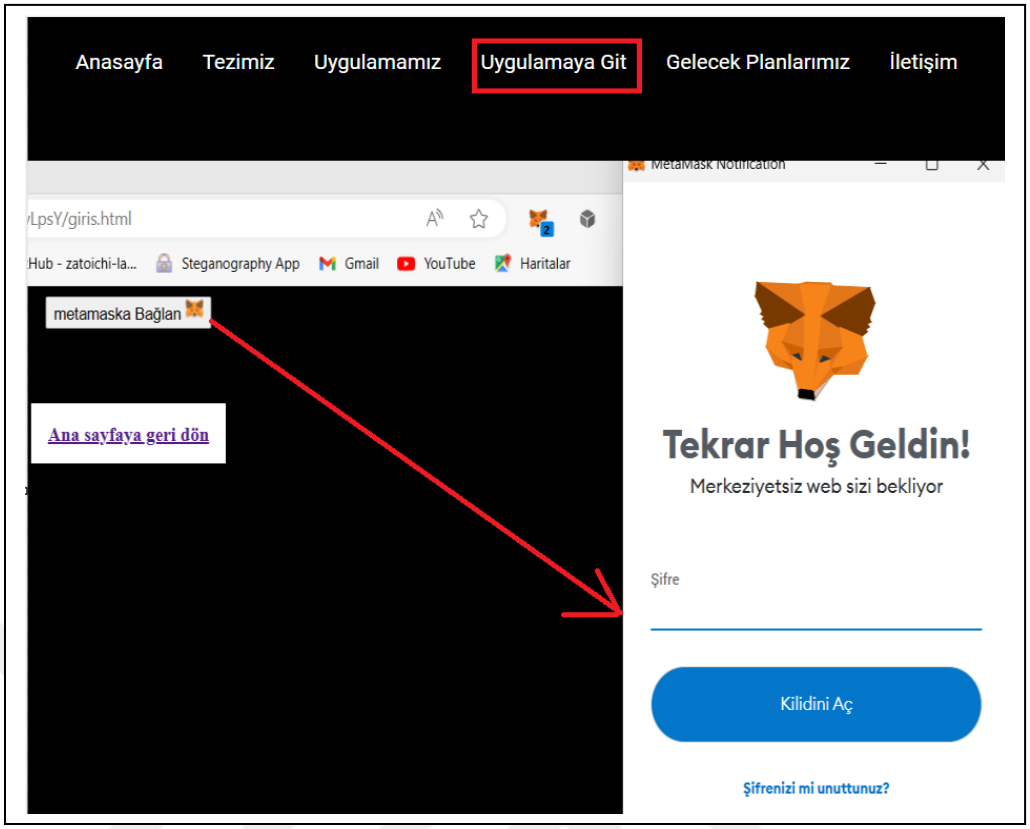
3.2. Örnek Çalışmalar

Bu bölümde geliştirilen sisteme resimler yüklenip sistemin işlenişi anlatılmıştır.



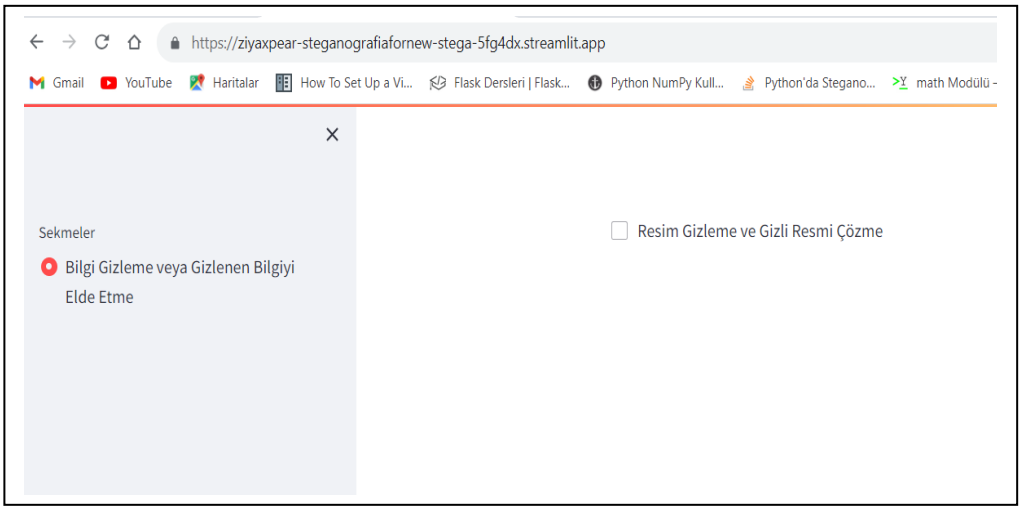
Resim 3.1. Kullanıcı dostu arayüzün kullanıcılara görünen ön yüzü (frontend kısmı)

Resim 3. 1' de görüldüğü gibi, arama çubuğunda Ipfs yardımıyla oluşturulan kullanıcı dostu arayüzün linki, tamamen CID yapısına sahip olan merkeziyetsiz bir hash kodundan oluşmaktadır.



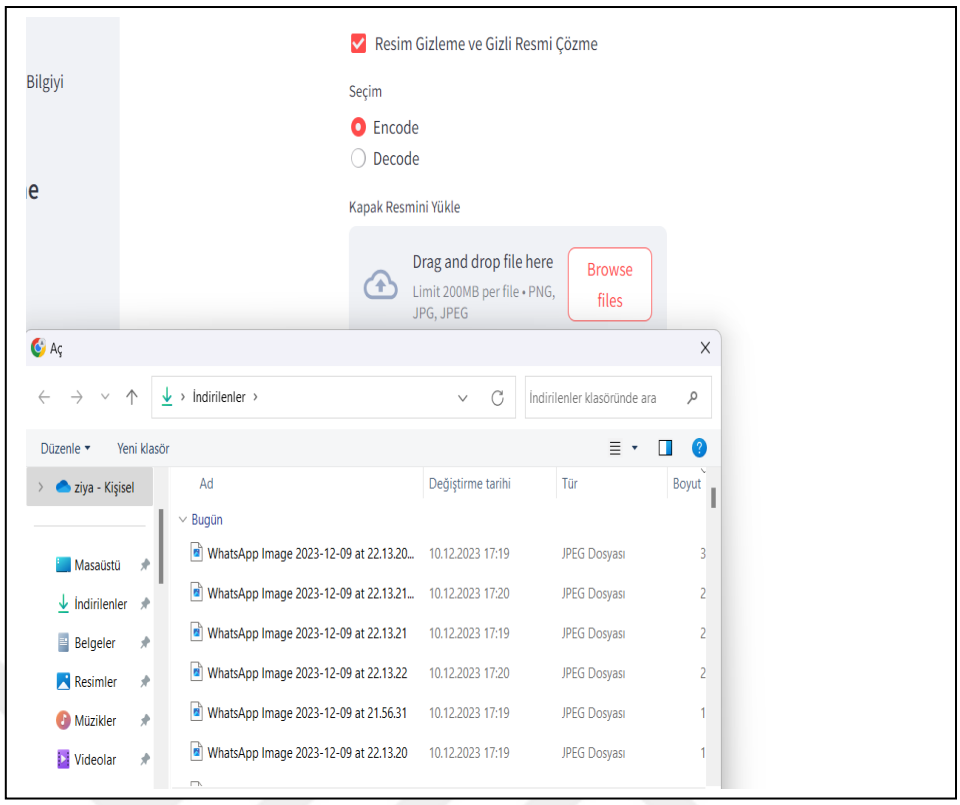
Resim 3.2. Anasayfa üzerinden uygulama giriş isteği

Resim 3. 2' de uygulamaya git butonuna basıldıktan sonra sistem otomatik olarak lokal (yerel) olarak çalışabilen metamask cüzdanına bağlanır ve kullanıcılar kişisel şifreleri ile sisteme giriş yaparlar.



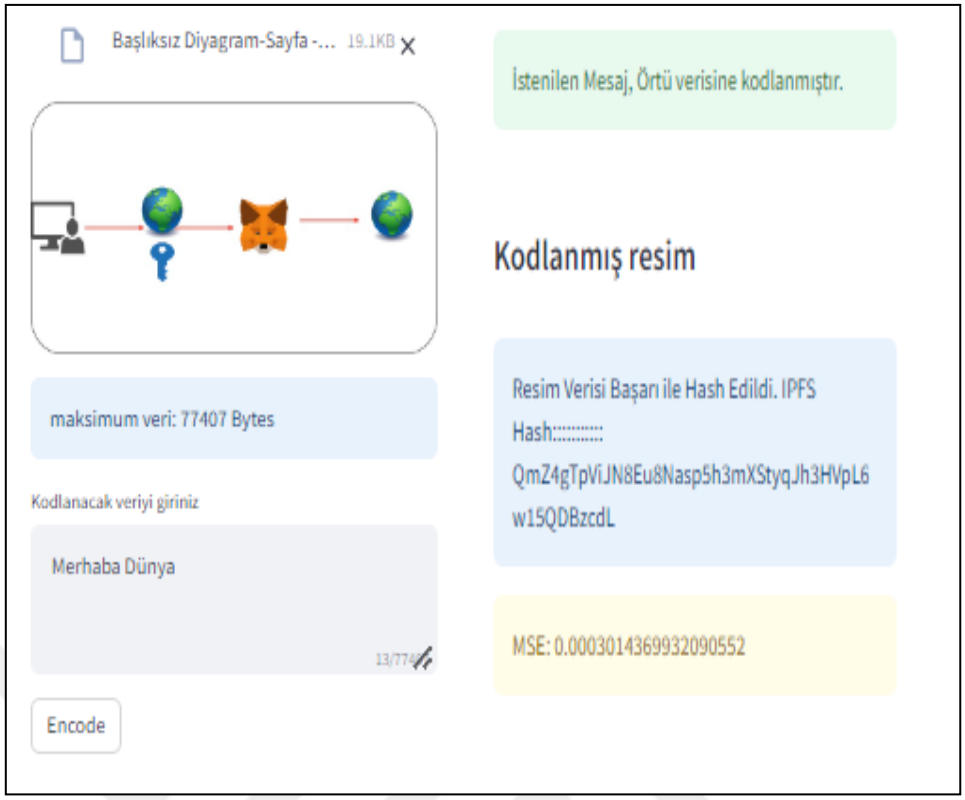
Resim 3.3. Güçlendirilmiş Steganografik ortamın anasayfası

Resim 3. 3' te, streamlit ortamında geliştirilen yapının web ortamına uyumlu linki, arama çubuğunda bulunmaktadır.



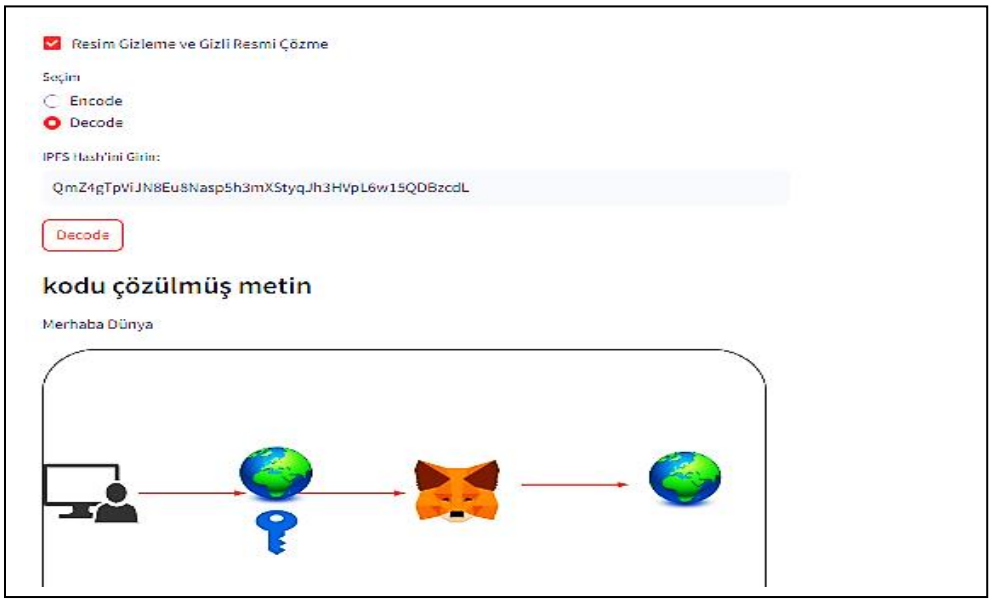
Resim 3.4. Steganografik yapıda bir resmin içine bilgi gizleme basamağı

Resim 3. 4' te, steganografik yapının içinde bulunan kullanıcılar, "Encode" butonuna bastıklarında sistem otomatik olarak kullanıcıları yerel dosyalarından bir resim seçmeye yönlendirir. Bu sayede kişiler, güvendikleri bir resmi seçerek kolaylıkla işlemlerini gerçekleştirebilirler.



Resim 3.5. IPFS hash' inin elde edilmesi

Resim 3. 5' te, "Encode" butonuna basıldığında "Merhaba Dünya" yazısı resmin içine kodlandı ve resim bir Ipfs hash' i haline getirildi.



Resim 3.6. IPFS hash' inden gizli bilginin elde edilmesi.

Resim 3. 6' da görüldüğü gibi, kullanıcılar bilgiyi gizledikleri resmin Ipfs hash' ini "Decode" kısmına yükleyip "Decode" butonuna tıkladıklarında, hem gizli mesaj

ekrana yansıtılacak hem de güvenlik teyidi için bilginin yüklü olduğu resim görünecektir.

Görüldüğü üzere sisteme erişmek hem kolay hem de güvenlidir. Biz bu projeyi geliştirirken herhangi bir ticari amaç gütmedik. Projenin oluşturulma amacı tamamı ile insanlık adına hizmet etmek insanların yararına bir güvenlik protokolü geliştirmek olmuştur. Sistemi detaylı inceleyip geliştirmek isteyen bireyler için istenildiği takdirde sistemin kaynak kodları da paylaşılacaktır.

4. SONUÇ VE ÖNERİLER

Bu çalışma, blok zinciri, Ipfs, Pinata, Metamask gibi teknolojilerin entegrasyonunu ve Python dilinin kullanımını içeren bir dizi kapsamlı analiz gerçekleştirmiştir. Elde edilen sonuçlar ve bu teknolojilerin gelecekteki potansiyeli üzerine yapılan değerlendirmeler şu şekildedir:

i. Metamask ve Blok Zinciri Entegrasyonu

Metamask, kullanıcıların blok zinciri tabanlı uygulamalarla etkileşim kurmasını sağlayan bir cüzdan aracıdır. Bu entegrasyon, güvenli bir şekilde Blok zinciri üzerinde işlem yapmayı ve akıllı sözleşmeler gibi özellikleri kullanmayı mümkün kılar.

ii. Python ve Veri Analizi Algoritmaları

Python, veri analizi ve işleme için geniş bir kullanım alanına sahip bir dil olarak steganografi, veri gizleme algoritmaları, görüntü işleme ve veri bütünlüğü gibi konularda etkili bir şekilde kullanılmaktadır. Özellikle LSB, PSNR, MSE gibi algoritmalar veri gizleme, veri bütünlüğü ölçümleri ve gizli mesajların saklanması konusunda önemli araçlardır.

iii. Hash Fonksiyonları ve Güvenlik

Hash fonksiyonları, verinin bütünlüğünü korumak, gizli mesajları güvenli bir şekilde saklamak ve veri doğrulaması için kritik öneme sahiptir. Bu çalışma, veri bütünlüğünü sağlamak ve gizli mesajların güvenliğini artırmak için hash fonksiyonlarının kullanımını vurgulamaktadır.

iv. Steganografi ve Gizli Mesajların Güvenliği

Steganografi, gizli mesajların gömülmesi ve saklanmasını sağlayan bir tekniktir. Bu çalışma, steganografi tekniklerinin gizli mesajların güvenli bir şekilde saklanması ve iletilmesi için önemli olduğunu ortaya koymaktadır. Bu teknikler, veri gizliliği ve güvenliği açısından büyük öneme sahiptir.

4.1. Öneriler ve Gelecek Çalışmalar

Bu çalışmadan elde edilen sonuçlar, gelecek araştırmalar ve uygulamalar için bazı önerilerde bulunmaktadır. Öncelikle, Python ve benzeri programlama dillerinin steganografi ve veri güvenliği alanında daha fazla kullanılabilir olmasını sağlamak önemlidir. Kullanıcı dostu araçların ve eğitim materyallerinin geliştirilmesi, bu

teknolojilere daha geniş bir erişimi mümkün kılabilir. Hash fonksiyonlarının ve steganografik algoritmalarının daha da geliştirilmesi gerekmektedir. Veri güvenliği ve bütünlüğünü artırmak için daha etkili ve güvenilir algoritmaların geliştirilmesi, dijital dünyadaki veri güvenliği açısından kritik öneme sahiptir.

Son olarak, blok zinciri, Ipfs ve steganografi entegrasyonunun daha geniş alanlarda uygulanması için daha fazla çalışma yapılmalıdır. Bu teknolojilerin sağladığı güvenlik avantajlarının ve veri bütünlüğünün daha geniş bir kullanım alanına yayılması, dijital dünyada daha güvenilir ve güvenli bir iletişim ortamının oluşturulmasına yardımcı olabilir.

Bu öneriler, steganografi, blok zinciri ve veri güvenliği alanlarında gelecek çalışmalar için önemli bir yol gösterici olabilir ve bu teknolojilerin daha etkili, güvenilir ve yaygın bir şekilde kullanılmasına katkıda bulunabilir.

KAYNAKLAR

[1] Agecon Search, "This Document is Discoverable and Free to Researchers Across the Globe Due to the Work of Agecon Search." in *The World' s Largest Open Access Agricultural & Applied Economics Digital Library*, 2013-2016.

[2] "Internet of Things (Nesnelerin İnterneti) Iot Nedir? Cihazların Etkileşi Trendleri,"https://www.karel.com.tr/blog/internet-things-nesnelerin-interneti-nedir-cihazlarin-etkilesim-trendleri,(accessed November 15, 2022).

[3] "How Many Iot Devices are There in 2022?," Https://Techjury.Net/Blog/How-Many-İot-Devices-Are-There/#Gref(accessed November 25, 2023).

[4] Y. Yiğit and M.Karabatak, "A Stenography Application for Hiding Student Information in to an Image," in *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. Ieee, pp. 1-4, 2019.

[5] B. Demirci, "Görüntü Steganografi Medotları ve Performanslarının Karşılaştırılması", Yüksek Lisans Tezi, Fen Bil. Ens, Selçuk Üniv., Konya, Türkiye, 2016.

[6] F. Neil, S. Johnson and C. Katzenbeisser, "A Survey of Steganographic Techniques," in *Information hiding techniques for steganography and digital watermarking*, 2007, pp. 43-78.

[7] S. M. Darwish, S. K. Guirguis and A. Wesam, "An Enhanced Steganographic System for Data Hiding in True Color Images," *Significance.*, vol. 2, no. 3, pp. 73-84, 2013.

[8] M. Khan, A.Jamil, F. Haleem and Z.Muhammad, "A Novel Image Steganographic Approach for Hiding Text in Color Images Using Hsi Color Model," Department of Computer Science, Islamia College Peshawar, Pakistan, 2015.

[9] C. Khalil and F. Hikmat, "Combining Steganography and Cryptography: New Directions," *International Journal on New Computer Architectures and Their Applications(Ijncaa*), vol. 1, no. 1, pp. 199-208, 2011.

[10] W. Huaiqing and W. Huozhong, "Cyber Warfare: Steganography vs. Steganalysis," *Communıcatıons of The Acm.*, vol. 47, no. 10, pp. 76-82, 2010.

[11] S. Bhallamudi, “Image Steganography,” Final project, *Wrıght State Univ.*, USA, pp. 1- 7, 2015.

[12] Ç. Ay, “Kablosuz Ağlarda Yeni Bir Anahtar Dağıtım Yöntemi,” *Düzce Üniversitesi Bilim ve Teknoloji Dergisi.*, cilt. 5, sayı. 1, ss. 306-313, 2017.

[13] M. Kalkan,“Kriptografi, Steganografi ve Kodlama Teorisinin Grup Teorisi ile İlişkisinin İncelenmesi ve Bazı Uygulamaların Geliştirilmesi” Doktora tezi, Fen Bil. Ens., Nevşehir Hacı Bektaş Veli Üniv., Nevşehir, Türkiye, 2021.

[14] R. Dippesh and B. Vijaya, “A Steganography Technique for Hiding Image in an Image Using Lsb Method for 24 Bit Color Image,” *International Journal of Computer Applications.*, vol. 64, no. 20, pp. 15-19, 2013.

[15] F. Khan and .A. A. Gutub, ” Message Concealment Techniques Using Image Based Steganography,” Department of Computer Engineering, King Fahd University of Petroleum & Minerals, Dhahran, Kingdom of Saudi Arabia, 2007.

[16] H. Aminou, M. Youssoufa, D. Georges, and O. Olle, “Robust Steganography Based on Matching Pixel Locations*,” International Journal of Computer Applications.*, vol. 168, no. 12, pp. 3-10, 2017.

[17] A. Abdelfatah A. Tamimi and M. A. Ayman, “An Audio Shuffle Encryption Algorithm,”in *The world congress on engineering and computer science*. 2014, pp. 409-412.

[18] E. A. Güzeldereli, “Veri Gizlemede Sıkıştırma Algoritması Kullanımı ve Uygulaması,” Yüksek Lisans tezi, Fen Bil. Ens., Sakarya Üniv., Sakarya, Türkiye, 2012.

[19] S. F. Katzenbeisser and A.P. Petitcolas, *Information Hiding Techniques for teganography and Digital Watermarking*, United Kingdom: Artech House, 2000.

[20] T. Valıbaylı, ”Web Güvenlik Dahilinde Steganografi Kullanımı ve Kullanıcı Giriş Sisteminde Uygulanması,” Yüksek Lisans tezi, Lisansüstü Eğitim Ens., stanbul Aydın Üniv., İstanbul, Türkiye, 2020.

[21] F. Nail, F. Johnson and J. Sushil, “Steganalysis of Images Created Using Current Steganography Software,” in *International Workshop on Information Hiding*. Berlin, Heidelberg, 1998, pp. 273-289.

[22] M. Hatim, H. Mathkour, S. Dokhekh, M. Morsi and G. Yassa, "Steganiys of Jpep İmages: an İmproved Approach For Breaking the F5 Algorithm," in *12 th Wseas International Conferance on Computers*, Department of Computer, 2008, pp. 923-25.

[23] J. Qin, Y. Luo, X. Xiang, Y. Tan and H. Huang,"Coverless Image Steganography: A Survey," *Ieee Access*, vol. 7, pp. 171372-171394, 2019.

[24] A. A. Abdula, H. Seleheva, S. A. Jasiom, "Improving Embedding Efficiency for Digital Steganography by Exploiting Similarities Between Secret and Cover Images," *Multimedia Tools and Applications*, vol. 78, no. 1, pp. 7799-17823, 2019.

[25] "Ipfs Nedir Matrıx- Ipfs işkisi",Https://Medium.Com/@Matrixaiturkey/İpfs-Nedir-Matrix-İpfs-İ%Cc%87li%C5%9fkisi-42153f3ca7a3(erişim tarihi Kasım 25 , 2022).

[26] "What is Web 3.0? Everything You Need to Know About Web 3.0," Https://Www.Simplilearn.Com/Tutorials/Blok Zinciri-Tutorial/What-İs- Web-3-0(accessed November 29, 2023).

[27] "What is Blockhain,"Https://Lisk.Com/What-İs-Blok Zinciri(accessed November 29, 2023).

[28] D. Volkhonskiy, L. Nazarov and E. Burnaev, "Steganographic Generative Adversarial Networks,"in *Twelfth international conference on machine vision (ICMV 2019)*, 2020, pp. 991-1005.

[29] A. A. Mamun, S. R. Hasan, S. Bhuiyan, M. S. Kaiser and M. A. Yousuf, " Secure and Transparent Kyc for Bankıng System Usıng Ipfs and Blockchaın Technology," in *2020 IEEE region 10 symposium (TENSYMP)*, 2020, pp. 348-351.

[30] C. Zhang, L. Zhu, C. Xu, Z. Zhang, and R. Lu, " Ebdl: Effective Blockchain-Based Covert Storage Channel With Dynamic Labels," *Journal of Network and Computer Applications*, vol. 210, no. 103541, pp. 1-14, 2023.

[31] A. Chaudhary, A. Sharma and N. Gupta, "Designing a Secured Framework for the Steganography Process Using Blockchain and Machine Learning Technology," *Intellıgent Systems and Applıcatıons in Engıneerıng*, vol. 11, no. 2, pp. 96-103, 2023.

[32] S. Athanere,and R. Thakur, "Blockchain Based Hierarchical Semi-Decentralized Approach Using Ipfs for Secure and Efficient Data Sharing," in *Journal of King Saud University – Computer , , 4th International Conference On Innovative Data Communication Echnology And Applicationand Information Sciences*, 2020, pp. 1523-1524.

[33] I. Noreen, M.S. Muneer and S. Gillani, "Deepfake Attack Prevention Usin Gsteganography," *Gans Peerj Computer Science*, no. 8, pp. 1-24, 2022.

[34] S. Badlani, T. Aditya, S. Maniar, and K. Devadkar, "Educrypto: Transforming Education Using Blockchain," in *6th Proceedings of the Sixth International Conference on Intelligent Computing and Control Systems*, 2022, pp. 829-836.

[35] M. Y. Jabarulla and H. N. Lee, "Blockchain-based distributed patient-centric image management system," *Applied sciences*, vol. 11, no. 196, pp. 196, 2020.

[36] Y. Chen, C. C. Chen, L. C. Tang and W. H. Chieng, " Nutext: a Novel Method for Music Encoding and Itspractical Application," *Research Square*, pp. 1-23, 2022, doi: Https://Doi.Org/10.21203/Rs.3.Rs-1969878/V1

[37] S. Zhao and D. O'Mahony, "BMCProtector: A blockchain and smart contract based application for music copyright protection," in *Proceedings of the 2018 International Conference on Blockchain Technology and Application*, 2018, pp.1-5

[38] Z. Qu, and H. Sun, "A Secure Information Transmission Protocol for Healthcare Cyber Based on Quantum Image Expansion and Grover Search Algorithm," *IEEE Transactions on Network Science and Engineering*, vol. 10, no. 5, pp. 2551-2563, 2022.

[39] H. A. Khan, R. Abdulls, S. K. Selvaperumal, and A. Bathich, " Iot Based on Secure Personal Healthcare Using Rfid Technology and Steganography," *International Journal of Electrical and Computer Engineering (Ijece*), vol. 11, no. 4, pp. 3300-3309, 2021.

[40] T. Manikandan, A. Murogandham, R. Babuji, V. Nandal and J. L. Mazher,"Secure e-Health Using Images Steganography," in *Journal of Physics: Conference Series*, 2021, pp. 12016, doi: 10.1088/1742-6596/1917/1/012016

[41] M. M. Hashim, S. H. Rhaif, A. A. Abdulrazzaq, A. H. Ali, and S. M. Taha,

"Based on Iot Healthcare Application for Medical Data Authentication: Towards a New Secure Framework Using Steganography," in *3rd International Conference on Sustainable Engineering Techniques*, 2020, vol. 881, no. 1, pp. 012120, doi:10.1088/1757-899x/881/1/012120

[42] A. O. Babatunde, A. J. Taiwo, and A. G. Dada, " Informatıon Securıty in Health Care Centre Usıng Cryptography and Steganography," Computer Science Department, University of Ilorin, Kwara State, Nigeria, 2018.

[43] M. Sajjad, M. Khan, S. W. Baik, S. Rho, Z. Jan, S. S. Yeo, and I. Mehmood, "Mobile-Cloudassisted Framework for Selective Encryption of Medical Images with Steganography for Resource-Constrained Devices," *Multimedia Tools and Applications*, vol. 76, pp. 3516-3539, 2016, doi:10.1007/S11042-016-3811-6

[44] M. S. Rahman, I. Khalil, and X. Yi, "Reversible Biosignal Steganography Approach for Authenticating Biosignals Using Extended Binary Golay Code." *Ieee Journal of Biomedical and Health İnformatics*, vol. 25, no. 1, pp. 35-46, 2020.

[45] R. Karakış, K. Gürkahraman, B. Çiğdem, İ. Öztoprak, and A. S. Topaktaş, "Evaluationof Segmented Brain regions for Medical Image Steganography," *Journal of the Faculty of Engineering and Architecture of Gazi University*, vol. 36, no. 4, pp. 2301-2314, 2021.

[46] A. Omotosho, O. Adegbola, and O. O. Mikail, "A Secure Electronic Prescription System Using Steganography with Encryption Key Implementation," *International Journal of omputer and Information Technology*, vol.3, no. 5, pp. 980-986, 2014.

[47] E. S. Bin Hureib, and P. A. Gutub, "Enhancing Medical Data Security via Combining Elliptic Curve," *IJCSNS International Journal of Computer Science and Network SecurityCryptography and Image Steganography* ,vol. 20, no. 8, pp.1-8, 2020.

[48] P. Mathivanan, and A. B. Ganesh, "ECG Steganography Based on Tunable Q-factor Wavelet Transform and Singular Value Decomposition." wileyonlinelibrary.com/journal/*ima*, vol. 31, no. 1, pp. 1-18, 2021, doi: 10.1002/ima.22477

[49] P. Mathivanan, S. E. Jero, and A. B. Ganesh, "QR Code-Based Highly Secure

ECG Steganography," in *International Conference on Intelligent Computing*, Singapure, 2019, pp. 171-178.

[50] A. K. Sahu, and G. Swain, "Reversible Image Steganography Using Dual-Layer LSB Matching," *Sensing and Imaging*, vol. 21, no. 1, pp. 1-21, 2019, doi:10.1007/s11220-019-0262-y

[51] F. Barkuş ve M. Koç, "Dijital Mahremiyet Kavramı ve İlgili Çalışmalar Üzerine Bir Derleme," *Bilim Eğitim Sanat ve Teknoloji Dergisi*, cilt. 3, sayı. 1, ss. 35-44, 2019.

[52] M. Nanohar and P. V. Kummar, "Data Encryption & Decryption Using Steganography," in *2020 4th international conference on intelligent computing and control systems (ICICCS)*, pp. 697-702, 2022.

[53] Ç. Dereli, " Dilbilimsel Steganografi Yöntemleri Üzerine Bir Araştırma," Yüksek Lisans tezi, Fen Bil. Ens., Ege Üniv., İzmir, Türkiye, 2010.

[54] Z. Erkin, "Steganografi Bilgisayar Ağlarında Güvenlik", Bitirme tezi, Fen Bil. Ens., İstanbul Teknik Üniv., İstanbul, Türkiye, 2005.

[55] L. Maryel and C. Retter, "Spread Spectrum Image Steganography," *Ieee Transactions on Image Processing*, vol. 8, no. 8, pp. 1075-1083, 1999.

[56] E. M. Esin ve E. Güvenoğlu, "Resim İçine Yazı Gizlenmesi Amacıyla Kullanılan Lsb Ekleme Yönteminin Shuffle Algoritmasıyla İyileşitirilmesi, " *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, cilt. 2, sayı. 1, ss. 1-6, 2007.

[57] Y. Yalman, İsmail. Ertürk, "Sayısal Ses İçerisinde Gizli Veri Transferinin Kablosuz Ortamda Gerçekleştirilmesi," *Politeknik Dergisi*, cilt. 11, sayı. 4, ss. 319-327, 2008.

[58] Ş. Sağırlıoğlu ve M. Tunçkanat, "Güvenli İnternet Haberleşmesi İçin Bir Yazılım," Yüksek Lisans tezi, Fen Bil. Ens., Erciyes Üniv., Kayseri, Türkiye, 2002.

[59] E. Duman, "Kayıpsız Veri Sıkıştırma ve Steganografi Tekniğine Dayalı Yeni Bir Yöntem," Yüksek Lisans tezi, Fen Bil. Ens., Sakarya Üniv., Sakarya, Türkiye, 2019.

[60] A. Westfeld, and A. Pfitzmann, " Attacks on Steganographic Systems, Breaking the Steganographic Utilities Ezstego, Jsteg Steganos and S-Tools and Some

Lessons Learned," in *International Workshop on Information Hiding*. Berlin, Heidelberg, 1999, pp. 61-76.

[61] R. Karakış ve İ. Güler, "Bulanık Mantık Tabanlı Görüntü Steganografi Uygulaması," Doktora tezi, Bilişim Ens., Gazi Üniv. Ankara, Türkiye, 2014.

[62] N. F. Johnson and S. Jajodia, "Steganalysis of Images Created Using Current Steganography Software," in *International Workshop on Information Hiding*, 1998, pp. 273-289.

[63] Ö. Kurtuldu ve N.Arıca," İmge Kareleri Kullanan Yeni Bir Steganografi Yöntemi," *Journal of Naval Science and Engineering*, vol. 5, no. 1, pp. 107-118, 2009.

[64] Y. İnan, "Assesment of the Image Distortion in Using Various Bit Lengths of Steganographic LSB," in *ITM Web of Conferences. EDP Sciences*, 2018, p. 01026.

[65] "Yazılım Dilleri ve Özellikleri ," https://aws.amazon.com/tr/what-is/javascript/(erişim tarihi 1 Kasım, 2023).

[66] "why and what python," https://github.blog/2023-03-02-why-python-keeps-growing-explained/(accessed November 11, 2023).

[67] "What is HTML," https://coderspace.io/sozluk/html(accessed November 15, 2023).

[68] "How does blockchain work?," https://www.simplilearn.com/tutorials/blockchain-tutorial/blockchain-technology(accessed November 15, 2023).

[69] A.Dorri, S. Salil, R. Jurdak and P. Gauravaram, "Blockchain for Iot Security and Privacy: the Case Study of a Smart Home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops*, 2017, pp. 618-623.

[70] A. Singh, H. V. Gupta, and V. Gupta, "Exploring the Cosmos of Data: Unleashing the Potential of Ipfs (Interplanetary File System) for Decentralized Storage," *Vidhyayana an International Multidisciplinary Peer Reviewed E-Journal*, vol. 8, no. 6, pp.1-19, 2023.

[71] "an aplication with streamlit," https://medium.com/machine-learning-

t%C3%BCrkiye/streamlit-ile-web-uygulamalar%C4%B1na-giri%C5%9F-7093e749973f (accessed November 16, 2023).

[72] "İmport Meta Data,- Python Dokumantasyon," Https://Docs.Python.Org/3/Library/İmportlib.Metadata.Html?Highlight=Egg_İnfo (accessed November 25, 2023).