ESOGÜ Müh Mim Fak Derg. 2021, 29(1), 28-42 J ESOGU Engin Arch Fac. 2021, 29(1), 28-42

GÖRÜNTÜ DOSYALARININ ŞİFRELENEREK GÜVENLİ ŞEKİLDE SAKLANMASI

Merve CEYHAN[1*], Esra N. YOLAÇAN[2]

[1] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir
ORCID No : https://orcid.org/0000-0003-0733-3652
[2] Eskişehir Osmangazi Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği Bölümü, Eskişehir
ORCID No : https://orcid.org/0000-0002-0008-1037

Anahtar Kelimeler

Şifreleme, Görüntü Şifreleme, Güvenlik, Depolama, Bulut Depolama.

Öz

İnternet kullanımının artması ve teknolojik gelişmelerle beraber insanlar her gün yüzlerce fotoğraf çekmektedir. Gün geçtikçe artan fotoğraflar telefon hafızasında fazlasıyla yer kaplamaktadır. Taşınabilir bellek, bilgisayar ve telefonlarda bu fotoğrafların saklanması, kaybolma ve bozulma gibi durumlar nedeniyle çok güvenilir olmamaktadır. Bu durum telefon dışında yeni depolama alanları ihtiyacını ortaya çıkarmıştır. Bu noktada alternatif olarak bulut depolama sistemleri kullanılmaktadır. Bulut depolama alanları sisteme erişim ve saldırılara karşı güvenlik tedbirleri içermektedir fakat sisteme yüklenen resimler üzerinde doğrudan bir işlem yapmamaktadır. Bulut hesaplar bazen kötü niyetli kişiler tarafından ele geçirilmekte ve kişilerin fotoğrafları çalınabilmektedir. Bu durumu önlemek için depolama işleminden önce görüntü üzerinde şifreleme algoritmaları kullanan bir sistem önerilmiştir. Sistemde CodeIgniter (PHP Framework) altındaki çeşitli simetrik şifreleme algoritmaları karşılaştırılarak incelenmiş ve uygulama sonuçları sunulmuştur. Şifrelenecek görüntü dosyalarının kendi uzantılarını kullanmak yerine görüntüler şifrelendikten sonra farklı bir dosya uzantısı ile Amazon Web Servis ortamında ya da yerel ortamda saklanması sağlanmıştır. Görüntüleri ele geçirmek isteyen kötü niyetli kişiler farklı uzantılı bir dosya ile karşılaştıklarında bunun bir görüntü dosyası olduğunu doğrudan tespit edemeyecektir. Bu çalışmada önerilen sistem ile görüntülerin tutuldukları ortamdan bağımsız olarak daha güvenli bir depolama imkanı sağlanması hedeflenmiştir.

SAFE STORAGE OF IMAGE FILES BY ENCRYPTING

Keywords

Encryption, Image Encryption, Security, Storage, Cloud Storage.

Abstract

With the increase in internet usage and technological developments, people take hundreds of photos every day. Photos increasing day by day take up a lot of space in the phone memory. Storing these photos in flash memory, computers and phones is not very reliable due to situations such as loss and corruption. This situation has created the need for new storage areas other than the phone. Cloud storage systems are used as an alternative at this point. Cloud storage areas contain security measures against system access and attacks, but they do not directly work on the images uploaded to the system. Cloud accounts are sometimes hijacked by malicious people and photos of people can be stolen. In order to prevent this situation, a system that uses encryption algorithms on the image before storage has been proposed. Various symmetric encryption algorithms

* Sorumlu yazar; e-posta : mceyhan@ogu.edu.tr

under CodeIgniter (PHP Framework) have been compared in the system and the application results are presented. Instead of using their own extensions of the image files to be encrypted, it is ensured that the images are stored with a different file extension in the Amazon Web Service environment or local environment after they are encrypted. When malicious people who want to capture images encounter a file with a different extension, they will not be able to directly determine that it is an image file. With the system proposed in this study, it is aimed to provide a safer storage opportunity regardless of the environment in which the images are stored.

Araştırma Makalesi		Research Article	
Başvuru Tarihi	: 13.11.2020	Submission Date	: 13.11.2020
Kabul Tarihi	: 08.01.2021	Accepted Date	: 08.01.2021

1. Giriş

Sosyal medya ile birlikte günümüzde fotoğraf çekme oldukça popüler hale gelmiştir. Artan fotoğraf sayısı büyük boyutlarda depolama alanları gerektirmektedir. Günümüzde gelişen teknolojiyle birlikte artık depolama işlemleri bulut depolama alanları ile yapılabilmektedir. Bulut depolama, farklı uzantılı verilerin sanal ortamda saklandığı bir alandır. Bulut depolama özellikle artan fotoğraf, veri, video, dosya gibi ihtiyaçların sanal ortamda muhafaza edilmesi için büyük bir ihtiyacı karşılamaktadır (Liu ve Dong, 2012). Gelişmiş bulut sistemleri dağıtık depolama sistemleri sayesinde sağladıkları büyük kapasite imkanları ile tüm dünyada hem kurumsal hem de bireysel kullanıcı kitlesine sahiptir. Bulut sistemlerinin kullanıcı sayısının trilyonlara ulaşmasının diğer sebepleri de yüksek performans ve yedekleme ile çok kullanıcılı kurumlar için sağladığı imkanlardır. Ancak bulut depolama alanları fotoğraf gibi içerikleri depolarken çoğunlukla bu fotoğrafların gizliliği ve güvenliğini sağlamaya yönelik fotoğraf üzerinde herhangi bir şifreleme işlemi gerçekleştirmemektedir.

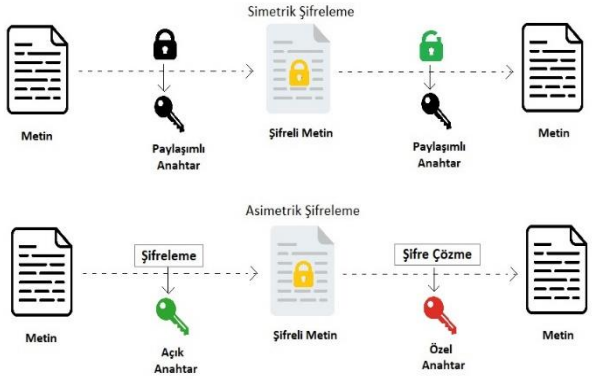
Şifreleme internet kullanımının artmasıyla beraber günümüzde önem kazanan çalışma alanlarından bir tanesidir. Kullanıcılar için önemli olan verilerin kaydedilmesinde ve iletilmesinde şifreleme işlemlerine ihtiyaç duyulmaktadır. Şifreleme, verileri okunamayan bir biçime dönüştürerek koruma biçimidir. Böylece verilere sadece yetkisi olan kişiler erişebilmekte ve verilerin güvenliği ve gizliliği sağlanmaktadır. Veriye erişim yetkisi olan kişiler, şifre çözme yöntemleri ile şifrelenmiş verilere kolaylıkla ulaşabilmektedir. Günümüzde araştırmacıların karşılaştığı zorluklardan birisi çoklu ortam (multimedya) verilerinin dijital ağlar üzerinden iletiminde iletim yolunun nasıl korunması gerektiği ile ilgilidir (Kumari, 2017). Veriye bağlı sistemlerin ve şirketlerin, artan güvenlik ihtiyacı insanları kriptoloji alanında çalışma yapmaya yönlendirmiştir. Verilerin güvenli şekilde saklanması ve aktarılması bu sistemlerin temel amaçlarındandır.

Tıbbi, askeri ve uzaktan algılama gibi uygulamalarda bilgiler ve görüntüler değerlidir. Bu veriler hem gizli hem de kişisel bilgiler içerir, bu nedenle verilerin güvenliğini ve bütünlüğünü sağlamak, güvenlik saldırıları ve bilgi kaybını önlemek için şifreleme oldukça önemlidir. Şifreleme işlemi için birbirinden farklı çalışma prensiplerine sahip yöntemler bulunmaktadır. Bu yöntemler hız, yük, karmaşıklık, bellek gereksinimi, maliyet, bilgi kaybı, saldırılara karşı duyarlılık durumlarına göre farklılık göstermektedir. Şifreleme işlemlerinde şifreleme algoritmaları, anahtarlama ve çözümleme algoritmaları kullanılmaktadır.

Şifreleme algoritmaları ile şifrelenmek istenen bilgi üzerinde bazı işlemler gerçekleştirilir ve şifrelenmiş veri elde edilir. Şifre çözme algoritmaları ile şifrelenmiş veri anlaşılır ve anlamlı hale getirilmeye çalışılır. Şifreleme yaklaşımları anahtar tipi ve kullanım şekline göre simetrik ve asimetrik şifreleme olarak ikiye ayrılmaktadır (Şekil 1).

Simetrik şifreleme algoritmaları verinin şifrelenmesi ve şifresinin çözülmesi için ortak bir anahtar kullanmaktadır. Bu anahtar veriyi gönderen ve alacak taraflar arasında gizli tutulur. Şifrelenmiş veri ile anahtar da karşı tarafa iletilir ve bu anahtar ile şifre çözme yapılır. Simetrik ve asimetrik şifreleme algoritmalarının çalışma hızı karşılaştırıldığında simetrik şifreleme algoritmaları daha hızlı çalışmaktadır. Çalışma prensibi olarak da asimetrik şifreleme algoritmalarına göre uygulanması daha kolaydır. Şifrelemede kullanılan anahtar boyu da asimetrik şifreleme anahtarına göre daha kısadır. Simetrik şifreleme işlemlerinde karşılaşılan en önemli sorun anahtar dağıtımıdır. Simetrik şifreleme algoritması kullanan bir sistemde aynı anahtarın birden fazla kullanıcıya dağıtılması güvenlik açısından

sorunlara sebep olabilmektedir. Bu sorunun çözümü olarak asimetrik şifreleme algoritmaları önerilmiştir.



Şekil 1. Simetrik ve Asimetrik Şifreleme Algoritmalarının Anahtar Kullanımı

Asimetrik şifreleme algoritmaları simetrik şifreleme algoritmalarından farklı olarak şifreleme için açık anahtar ve şifre çözme için özel anahtar kullanmaktadır. Özel anahtar sadece şifreyi çözecek kullanıcıda bulunurken açık anahtar özel anahtar gibi gizli değildir. Asimetrik şifreleme algoritmaları simetrik şifreleme algoritmaları ile karşılaştırıldığında güvenlik olarak daha başarılıdır. Asimetrik şifreleme algoritmaları, simetrik şifreleme algoritmalarında ortaya çıkan anahtar fazlalığı problemine de çözüm getirir. Asimetrik şifreleme algoritması güvenliği sağlayabilmek için büyük asal sayılar kullanır, bu nedenle simetrik şifreleme algoritmalarına göre daha yavaştır.

Şifreleme algoritmaları simetrik ve asimetrik olarak ayrılsa da sistemde kullanılan şifreleme algoritmasına yardımcı olmak amacıyla anahtar kullanmayan algoritmalar da bulunmaktadır.

Yapılan çalışmada bulut depolama alanlarına veya yerel depolama alanına yükleme yapmadan önce görüntüler üzerinde şifreleme işlemleri gerçekleştirilmektedir. Bu da depolama alanlarına kötü kişiler tarafından yapılacak olan herhangi bir saldırıda görüntülere erişimi ve paylaşımı engellemektedir. Çalışmada kişisel veri güvenliğini sağlamaya yönelik olarak görüntüler bulut depolama alanına veya yerel depolama alanına yüklenmeden önce şifreleme işleminden geçirilerek sadece fotoğraf sahibinin erişebileceği bir ortam geliştirilmiştir. Algoritmaların değerlendirilmesi için oluşturulan sistemde gerekli güvenlik önlemleri alınarak fotoğrafların güvenli şekilde depolanması sağlanmıştır. Bu sayede kişiler fotoğraflarını güvenli şekilde depolama imkanı bularak kişisel veri güvenliği sağlanmıştır. Üzerinde herhangi bir şifreleme işlemi uygulanmadan yeterli güvenlik önlemi alınmamış ortamlarda saklanan fotoğraflar, doğrudan kötü niyetli kişilerin hedefi olmaktadır. Bu kötü niyetli kişiler ele geçirdikleri verileri, kişilerin itibarını zedeleme ya da kazanç elde etme amaçlarıyla kullanabilmektedir. Yapılan çalışma ile veri hırsızlığının önüne geçmek ve kişilere güvenilir bir depolama alanı sunmak için şifreleme algoritmaları kullanılarak depolama işlemlerinin güvenliğini artırmaya yönelik bir yaklaşım üzerinde durulmuştur.

Bölüm 2'de görüntü şifreleme alanında yapılmış olan çalışmalar incelenerek kullanılan şifreleme yöntemlerine değinilmiş, çalışmaların karşılaştırılması ve yorumlamasına yer verilmiştir. Bölüm 3'te bu çalışmada kullanılan metodolojiye değinilmiş olup kullanılan yöntem, teknikler ve araçlara yer verilmiştir. Bölüm 4'te çalışmadan elde edilen sonuçların yanında kullanılan yöntem ve tekniklerin değerlendirmesi yer almaktadır. Bölüm 5'te elde edilen bilgi ve deneyimler, gelecekte yapılabilecek çalışmalara yönelik fikirler sunulmuştur.

2. Literatür Taraması

Görüntü şifreleme alanında şifreleme algoritmaları, şifre çözme işlemleri ve bu algoritmaların performansı üzerine yapılmış pek çok çalışma bulunmaktadır. Çalışmalarda farklı şifreleme algoritmaları incelenmiş ve şifre çözme işleminde yaşanan veri kayıplarından bahsedilmiştir. Görüntü dosyalarının şifrelenmesi amacıyla ilk olarak metin şifreleme işlemlerinde kullanılan algoritmalar tercih edilmiştir, ancak geleneksel şifreleme yöntemleri görüntü şifrelemede kullanıldığında önemli iki durum ortaya çıkmaktadır. İlk durum, görüntü verileri yazı verilerinden daha büyüktür. Bu yüzden görüntü şifrelemek, metin şifreleme işlemine göre daha yavaş kalmaktadır. İkinci durum, şifre çözme işlemlerinde metin üzerinde kayıplar daha az olurken görüntü üzerinde bit ve renk gibi görüntünün yapısını bozan veri kayıpları yaşanabilmektedir. Şifre çözüldükten sonra yaşanan veri kaybı bazen görüntülerde fark edilemeyecek kadar küçük boyutlarda olabilirken bakıldığında anlaşılabilecek boyutlarda da olabilmektedir. Bu veri kayıplarını önlemek amacıyla Chang, Hwang ve Chen (2001) tarafından karmaşık şifreleme algoritması geliştirilerek herhangi bir veri kaybı olmaması sağlanmıştır. Önerilen algoritma ile görüntüdeki piksel değerleri yerine pikselin bulunduğu yerin değiştirilmesi işlemi yapılmaktadır (Chang, Hwang, ve Chen, 2001).

Piksel değerlerinin yerini değiştirmeye yönelik bir diğer çalışmada Knutt/Durstenfeld Shuffle algoritması kullanılarak yapılmıştır (Guvenoglu ve Esin, 2009). Sunulan çalışmada görüntü şifreleme için görüntüyü oluşturan pikselleri temsil eden sayısal değerlerin yerlerinin değiştirilmesine dayanan yeni bir yaklaşım önerilmiş ve kullanılan algoritmanın detaylarına yer verilmiştir. Görüntü piksellerinin yerlerini değiştirmeye dayanan Knutt/Durstenfeld Shuffle algoritmasının iyileştirilmesi için yeni yöntemler denemiş ve bu yöntemler üzerine çalışmalar da yapılmıştır (Guvenoglu ve Tuysuz, 2015). Şifreleme işlemindeki veri kayıplarını önlemek amacıyla yapılan bir benzer çalışma Yen ve Guo (1999) tarafından yapılmıştır. Önerdikleri ayna yansıması algoritması piksellerin permütasyon yöntemiyle karıştırılması mantığına dayanmaktadır (Yen ve Guo, 1999). Chang ve diğ. (2001) tarafından yapılan çalışmadaki gibi sunulan çalışmada da pikseller üzerinde değer değişimi olmadığı için görüntülerde bozulma olmamaktadır.

Görüntü şifrelemede kullanılan algoritmalardan birisi de Brie algoritmasıdır. Görüntüdeki piksel yerlerinin değiştirilmesi mantığına dayanan Brie algoritması J. C. Yen ve J. I. Guo tarafından yeni bir görüntü şifreleme algoritması olarak önerilmiştir. Fakat bu algoritmanın saldırılara karşı yeteri kadar güvenli olmadığı belirlenmiştir ve güvenliğin ön planda olduğu uygulamalarda kullanılmaması önerilmiştir (Shujun ve Xuan, 2002). Yapılan bir başka çalışmada S-Box olarak adlandırılan yer değiştirme kutularına yer verilmiştir. Günümüzde kullanılmakta olan pek çok algoritmanın arka planında yer değiştirme kutuları bulunmaktadır. Yer değiştirme kutuları yer değiştirme işlemi ile şifreleme algoritmalarını daha güçlü yapmak amacıyla kullanılmaktadır. Yapılan bir çalışmada AES (Advanced Encryption Standard) şifreleme algoritmasında kullanılan yer değiştirme kutularının benzeri dinamik olarak üretilmiştir ve dinamik yaklaşımın sonuçları değerlendirilmiştir (Guvenoglu, 2016).

Sakal ve Yıldırım (2016) tarafından yapılan çalışmada, görüntü güvenliğinin sağlanması amacıyla melez bir görüntü şifreleme çalışmasından bahsedilmiştir. Görüntülerin şifrelenmesinde, görüntüye gürültü ekleme kullanılan yöntemlerden birisidir. Bu çalışmada şifreleme için gürültü ekleme yöntemi kullanılmıştır. Resme gürültü eklendikten sonra oluşan yeni görüntüye tarama yöntemi uygulanarak görüntünün daha da bozulması sağlanmıştır. Şifre çözme işleminde, şifreleme işlemi için yapılan uygulamaların tersi işlemler uygulanarak şifresiz görüntü elde edilir. Çalışmada melez tekniklerle görüntü daha çok bozularak sadece gürültü ekleme işlemine göre daha güçlü bir şifreleme işlemi gerçekleştirilmiştir (Sakal ve Yıldırım, 2016).

Yapılan bir diğer çalışmada Arnold Cat Mapping kullanarak renkli görüntülerin piksel konumlarını karıştırma işlemi ile şifreleme yapılmaya çalışılmıştır. Arnold Cat Mapping, dijital renkli görüntüleri şifreleyen ve şifresini çözen klasik yöntemleri kullanarak görüntü piksellerinin değerlerini değiştirip gri renkli karıştırma görüntüsü elde eder. Karıştırma görüntü üzerinden şifreleme için difüzyon mekanizmasıyla birleştirme - karıştırma mekanizması kullanır. Önerilen bu yöntem, Veginner Substitution Cipher metodu ve Hill Cipher metodu gibi klasik şifreleme yapılarını da içermektedir. Bu yöntem renkli görüntüler üzerinde denendiğinde, Hill Cipher metodunun diğer yöntemlerden daha güvenli ve yüksek hızda olduğu görülmüştür (Hariyanto ve Rahim, 2016). Bir başka çalışmada görüntü şifreleme teknikleri ve yazı ile görüntü şifreleme arasındaki farklar üzerinde durulmuştur. Bu farkların en aza indirilmesi için görüntü üzerinde pozisyon permütasyonu teknikleri, değer dönüşümü teknikleri ve bu tekniklerin kombinasyonları gibi yöntemlerin uygulanması gerektiğinden bahsedilmektedir. Son yıllarda bu alanda Kaotik Haritalar (Chaotic Maps) yapısına dayalı görüntü şifrelemesinde kullanılan permütasyon ve difüzyon işlemleri de kullanılmıştır (Sharma, Godara, Singh, Tech, ve Sabo, 2012).

Görüntü şifrelemede AES gibi algoritmalar da kullanılabilir. Ghoradkar ve Shinde (2015) yaptıkları çalışmada görüntü şifrelemede AES kullanmışlardır. Çalışmalarında AES şifreleme algoritmasının arka planındaki çalışma aşamalarından bahsetmişlerdir (Ghoradkar ve Shinde, 2015). Alanda yapılan bir diğer çalışmada AES ile görüntü şifreleme işlemine yer verilmiştir. Kullanılan yöntemde görüntü dizi (array) formatına dönüştürülür ve AES şifreleme algoritma aşamaları ile şifreleme işlemi gerçekleştirilir. AES uygulanarak görüntü şifrelenebilir ve daha iyi sonuçlar için başka şifreleme algoritmalarıyla entegre hale getirilebilir. Uygulanan şifreleme yöntemi sırasında gizli bir görüntü de oluşturulur. Şifre çözme için, oluşturulan gizli görüntü kullanılır (Upadhyaya, Shokeen, ve Srivastava, 2015). RSA (Ron Rivest, Adi Shamir ve Leonard Adleman) da görüntü şifrelemede kullanılabilir. Algoritmanın isimlendirmesi, geliştiricilerinin soyadlarının baş harflerine göre yapılmıştır. RSA, şifreleme ve kimlik doğrulama sistemi sağlayan bir algoritmadır. Böyle bir şifreleme sisteminde şifreleme anahtarı herkese açık bir anahtardır ve şifre çözme anahtarı bundan farklı ve gizlidir (Anandakumar, 2015). RSA şifreleme anahtarı görüntüyü şifreler, böylece şifreli metin biçimine dönüştürülür ve metin dosyası

olarak depolanır. Ters şifreleme yöntemi, RSA algoritmasının bir başka şifre çözme anahtarı ile hesaplanır ve deşifre (şifre çözme işlemi) teknikleri ile yeniden görüntü elde edilir (Chepuri, 2017). Maniccam ve Bourbakis (2001) yaptıkları çalışmada hem kayıpsız sıkıştırma hem de etkin şifreleme yapan bir yöntemden bahsetmiştir. Mevcut şifreleme yöntemleri algoritmanın gizliliğine değil, anahtar gizliliğine dayalıdır. Bu nedenle çalışmada hem kayıpsız hem de etkin bir şifreleme yöntemine ihtiyaç olduğu öngörülmüştür. Önerilen sıkıştırma şifreleme yönteminde, görüntünün her pikseline bir kez erişilerek elde edilen bir tarama yolu bulunur. Tarama yolu ve tarama yolu boyunca bit dizisi kodlanarak ikili görüntü sıkıştırılır. Şifreleme işleminde tarama yolu gizli tutulmaktadır. Sıkıştırma yönteminde kodlanmış tarama yolunu ve tarama yolu boyunca kodlanmış bit dizisini temsil etmek için gereken toplam bit sayısını en aza indiren en uygun algoritma üzerinde durulmuştur (Maniccam ve Bourbakis, 2001).

Kester (2013) tarafından yapılan bir başka çalışmada görüntünün RGB (Red, Green, Blue) piksel değerlerini karıştırarak m*n boyutunda görüntüyü şifrelemek için görüntü tabanlı bir şifreleme algoritması önerilmiştir. Önerilen algoritma, RGB piksel değerlerine dayalı olarak görüntünün şifrelenmesini ve şifresinin çözülmesini mümkün kılar (Kester, 2013). Benzer şekilde görüntünün RGB değerlerini düzenleyerek şifreleme yapan başka çalışmalar da yapılmıştır (Goel ve Chandra, 2012).

Askar, Karawia ve Alshamrani (2015) yaptıkları çalışmada, görüntü şifreleme algoritması olarak kaotik ekonomik modeli önermişlerdir. Görüntü verilerini şifrelemek kaotik ekonomik model haritası ile kaotik bir dizi oluşturmaya bağlıdır. Görüntü piksel değerlerini içerirken, kaotik dizideki değerler ondalık yapıdadır. Bu nedenle ondalıklı sayıları tam sayıya aktarmak için bazı işlemler yapılmaktadır. Görüntü, elde edilen tamsayı dizi ile şifrelenir. Bu çalışmada algoritmanın çalışma prensibi ve aşamalarına yer verilmiştir (Askar, Karawia, ve Alshamrani, 2015). Kumar ve Mathew (2020) yaptıkları çalışmada görüntü şifrelemede kullanılan yaygın metot ve alternatif metotlara odaklanmış ve bu metotların karşılaştırmasına yer vermiştir. Bunun için farklı makaleler üzerinde inceleme yapılmış ve güvenlik saldırıları, hız, bilgi gibi konulara değinilmiştir (Kumar ve Mathew, 2020). Xingbin Liu, Dia Xiao ve Yanping Xiang (2018) yaptıkları çalışmada bit seviyesinde permütasyon stratejisi kullanarak kuantum görüntü şifrelemesi yapan bir yöntem önermişlerdir. Bu yöntemde görüntü, yeni geliştirilen kuantum temsil modeliyle şifrelenir, ardından bitler üzerinde XOR (Exclusive Or) ile permütasyon gerçekleştirilir. Çalışmada gerçekleştirilen aşamalara ait detaylara yer verilmiştir (X. Liu, Xiao, ve Xiang, 2018). Bir diğer çalışmada görüntü şifreleme algoritmaları çeşitli yönleriyle incelenmiş ve algoritmaların saldırılara karşı direnç hassasiyetleri, karmaşıklıkları, bilgi kayıpları gibi konulara odaklanılmıştır. Algoritmalar parametreleriyle ele alınıp değerlendirme ve karşılaştırmalar yapılmıştır (Kumar ve Mathew, 2020).

Görüntü şifreleme alanında yapılmış olan tüm bu yöntemler incelendiğinde her biri farklı avantaj ve dezavantajlara sahiptir. Karmaşık görüntü şifreleme algoritmaları yaygın olarak kullanılan yöntemlerden birisidir. Bu yöntemde piksel değerleri yer değiştirilerek karmaşıklık arttırılır. Bu karıştırma işlemi ne kadar karışık hale getirilirse kişilerin görüntünün orijinal halini anlaması zorlaşacaktır. Ayna tabanlı şifreleme algoritmaları piksel değerlerinin yerlerinin değiştirilmesi prensibine bağlı çalışmaktadır. Algoritmanın yapısı bilindiğinde bu yöntemde güvenlik sorunları ile karşılaşılmaktadır. Görüntü şifrelemede, metin şifrelemede kullanılan klasik şifreleme algoritmaları da kullanılabilir. Fakat bu algoritmalar metin şifreleme işlemine göre daha yavaş kalmakta ve veri kayıplarına neden olabilmektedir. Şifrelenen görüntünün boyutları ve görüntünün renk durumu da şifreleme sonuçlarında farklı sonuçlar ortaya koymaktadır. Çözünürlüğü düşük görüntülerde şifre çözme sırasında veri kayıpları yaşadığında gözle görülebilirken, bu durum çözünürlüğü yüksek görüntülerde anlaşılmamaktadır. Bu çalışmada, şifrelenen görüntülerin deşifre edilmiş hallerinde bozulma olması istenmediğinden, görüntü şifrelerken piksel değerleri üzerinde işlem yapmayan yöntemler tercih edilmiştir.

Yapılan literatür taramasında görüntüleri şifrelemek amacıyla çeşitli algoritmalar ve yöntemlerin kullanıldığı görülmüştür. Bu yöntemlerde şifreleme hızı, şifre çözme sonrası elde edilen performansların birbirinden farklı olduğu gözlenmiştir. Bu çalışmada, görüntü dosyalarının güvenliği için çeşitli simetrik anahtar şifreleme algoritmalarının kullanılması ile elde edilen sonuçlar sunularak, algoritmaların karşılaştırmalı değerlendirilmesine olanak sağlanmıştır. Görüntü şifreleme için yapılacak yeni çalışmalar için de uygulamalı karşılaştırma kılavuzu niteliğinde olması hedeflenmiştir.

3. Metodoloji

Bu bölümde, çalışmada kullanılan yöntem ve teknikler açıklandıktan sonra kullanılan şifreleme algoritmalarına yer verilmiş ve alınan güvenlik

önlemlerine değinilmiştir. Bu çalışmada araştırma ve yayın etiğine uyulmuştur.

3.1. Yöntem ve Teknikler

Yapılan çalışmada şifreleme algoritmalarını değerlendirmek amacıyla geliştirilen sistem açık kaynak kodlu PHP (Hypertext Preprocessor) dilinde CodeIgniter ile oluşturulmuştur. PHP günümüzde hem açık kaynak topluluğunda hem de endüstride web odaklı büyük uygulamalar ve uygulama çerçeveleri oluşturmak için yaygın olarak kullanılan en popüler programlama dillerinden biridir (Siame ve Kunda, 2017). PHP ortamında geliştirme yapılırken entegresi kolay olan MySQL (My Structured Query Language) gibi bir veritabanları seçilmektedir. Yönetilebilir web siteleri için çoğunlukla MySQL veritabanı tercih edilmektedir. PHP ve MySQL bağlantısı doğru sağlandığında veritabanına erişim hızlı şekilde yapılabilmektedir. Bu durum geliştirilen ortamın erişim performansı olarak değerlendirilir. Geliştiriciler iyi çalışan uygulamalar yazmak ve bunu olabildiğince basit ve kolay bir şekilde yapmak ister. CodeIgniter, PHP kullanımını kolaylaştıran bir araçtır. CodeIgniter ücretsiz ve kurulumu basittir, geliştiricilerin işlerini kolaylaştırır (Upton, 2007). Hata ayıklama ve kodun testini kolaylaştırır ve uygulamadaki kodların daha kolay optimize edilmesini sağlar. CodeIgniter en sık ihtiyaç duyulan kütüphaneleri içerisinde bulundurduğu için kullanım kolaylığı sunmaktadır.

Algoritmaların karşılaştırılması amacıyla geliştirilen sistemde geliştirme ortamı olarak JetBrains PHPStorm tercih edilmiştir. PHPStorm, PHP'de uygulamaların geliştirilmesini kolaylaştırmak için özel olarak tasarlanmış bir IDE (Integrated Development Environment)'dir (Gajda, 2013). Web programlama için en iyi IDE'lerden birisi olan PHPStorm kod yazarken işleri kolaylaştırmanın yanında hız da sağlamaktadır. PHPStorm ücretli bir yazılımdır ancak ücretsiz versiyonu da bulunmaktadır. PHPStorm ortamında PHP yanında HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), JavaScript ile de geliştirmeler yapılabilmektedir. Bu özellikleri nedeniyle geliştirme ortamı olarak tercih edilmiştir. MySQL, çok kullanışlı bir ilişkisel istemci / sunucu veritabanı sistemidir. Pek çok uygulama için güvenli ve kararlıdır, yüksek maliyet / fayda oranı sunar (DuBois, 2008). Sorgu oluşturmada sağladığı kolaylıklar ve güçlü karakter seti desteği, PhpMyAdmin arayüzü gibi özellikleri ve kullanım kolaylığı nedeniyle veritabanı olarak MySQL tercih edilmiştir.

Yapılan çalışmada görüntülerin şifrelenmiş hallerinin saklanması için depolama alanı olarak AWS (Amazon Web Servis) kullanılmıştır. Bulut teknolojisi (Cloud Computing) yazılım, veritabanı, depolama gibi önemli hizmetlerin internet aracıyla erişimini sağlayan bir modeldir. İnternet yoluyla sanal sunuculara istenilen zamanda her yerden ulaşılabilmektedir. Dosyalar bilgisayar, taşınabilir bellek gibi alanlar yerine bulut depolama teknoloji ile çevrimiçi olarak saklanabilmektedir. Bulut depolama teknolojisinden yararlanan kullanıcılar telefon, tablet, bilgisayar gibi internet bağlantısı olan herhangi bir cihaz aracılığı ile resim, dosya gibi verileri internet ortamına yükleyebiliyor ve yine aynı şekilde internet ortamından dosya indirme işlemini gerçekleştirebilir. Bulut depolama alanlarına erişmek için özel yazılımlara veya teknik donanımlara gereksinim duyulmamaktadır. Kullanıcılar bulut depolama alanlarına kullanıcı adı ve şifre bilgileri ile kolaylıkla erişebilmektedir. Bulut depolama alanlarının kullanımı özel gereksinimler içermemesi nedeniyle pek çok kullanıcı tarafından tercih edilmektedir. Bulut depolama sistemleri kullanıcılarına günün her saatinde kesintisiz hizmet vermektedir. Bulut teknolojisi kullanıcıların internet aracılığı ile düşük maliyetlerle depolama alanlarına erişmesi prensibine dayanmaktadır.

Günlük hayatta internet üzerinden gerçekleştiğimiz pek çok işlemin alt yapısı bulut teknoloji ile sağlanmaktadır. Kullanımı yaygınlaşmaya başlayan bulut teknolojileri profesyonel amaçlarla pek çok şirket ve kurum tarafından da tercih edilmektedir. Bulut teknolojisini kullanan şirket ve kurumların veri merkezi oluşturma ihtiyacı bulunmamaktadır. Bulut sistemleri ölçeklenebilir bir yapıdadır ve yüksek erişim hızları sayesinde kullanıcılarına çeviklik sağlar. Bulut teknolojinin bu avantajları tercih edilmesinde etkili rol oynamaktadır. Bu durumda farklı özelliklere sahip bulut depolama sistemleri geliştirilmiştir. En bilinen depolama alanları AWS, Dropbox, Google Drive, Microsoft OneDrive şeklinde sıralanabilir. AWS, diğerlerine göre hızlı entegrasyon, akıllı senkronizasyon, kullandığın kadar ödeme mantığı, ölçeklenebilir oluşu, geniş güvenlik önlemlerini içerisinde barındırması nedeniyle bu çalışmada tercih edilmiştir (Mukherjee, 2019).

Veriler, kurumlar için büyük önem taşımaktadır. Veri sızıntıları büyük kayıplara neden olabilmektedir. Bu nedenle her kurum hassas verilerini korumak için veri gizliliğine önem verir. Veri koruma talimatlarına uyulmaması şirketin fikri mülkiyetinin çalınmasına, kuruluşun itibarının zedelenmesine, sistemin bilgisayar

korsanlığı veya kötü amaçlı yazılım açıklarına karşı tehlikeye atılmasına neden olabilir.

AWS, cihaz ve sunucuda veri şifreleme işlemini gerçekleştirir ve hassas veriler şifrelenerek saklanır. Hassas verileri şifrelemek için NIST (National Institute of Standards and Technology) onaylı şifreleme standardı algoritmalarını kullanır. Şifrelemede kullanılan anahtarlar herhangi bir depolama alanında saklanmaz, kullanımı sırasında gerçek zamanlı olarak üretilir ve yok edilir. Önbellekte ya da log kayıtlarında hassas bilgiler tutulmaz (Mukherjee, 2019).

Diğer depolama alanları hızlı entegrasyon, akıllı senkronizasyon ve sınırlı ücretsiz depolama alanı sunmaktadır fakat bilgi şifreleme özelliklerinde ve dosya paylaşım güvenliğinde zayıflıklar bulunmaktadır.

AWS, geleneksel depolama alanlarından farklı olarak esneklik, uygun maliyet, ölçeklenebilirlik, güvenlik gibi pek çok avantaja sahiptir. AWS, kuruluşların hali hazırda kullandıkları programlama modellerini, işletim sistemlerini, veri tabanlarını ve mimarileri kullanmasına olanak tanır. Bu sistemde kullanıcılar ve kuruluşlar uzun vadeli taahhütler yerine sadece kullandıkları sistem kadar ödeme yapmaktadır. Kuruluşlar müşteri talebini karşılamak ve maliyetleri yönetmek için AWS kaynaklarını uygulamalarına hızla ekleyip çıkarabilir. AWS, uçtan uca güvenlik ve gizlilik sağlamak için en iyi güvenlik uygulamalarına uygun olarak hizmetler oluşturup bu hizmetlerdeki uygun güvenlik özelliklerini sağlamaktadır (Varia ve Mathew, 2014). AWS kendi yapısı içinde doksandan fazla güvenlik standardını karşılamaktadır ve veri şifreleme imkanı da sunmaktadır. Ölçeklenebilir yapısı, kullanıcı kitlesinin çok oluşu ve güvenlik önlemleri nedeniyle diğer bulut depolama sistemlerine göre AWS kullanımı avantajlı görülmüştür. AWS dışında farklı bulut depolama alanları da tercih edilebilir.

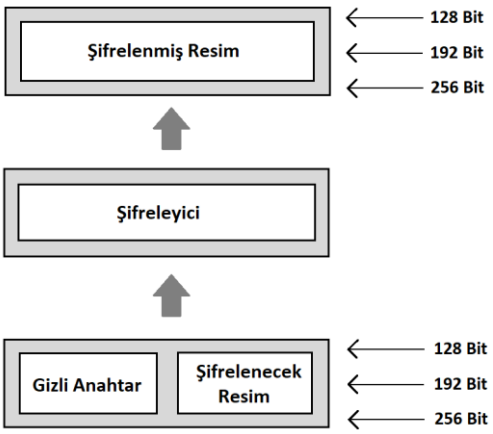
3.2. Şifreleme Yöntemleri

Algoritmaların karşılaştırılması için geliştirilen sistemde saklanacak olan veriler görüntülerdir. Görüntü dosyalarını, veritabanında doğrudan saklanmak yerine görüntülerin şifrelenmiş halleri saklanmaktadır. Görüntülerin şifrelenerek saklanması kötü niyetli kişilerin veritabanına erişip kullanıcıların görüntülerine doğrudan ulaşmasını engellemektedir. Görüntü üzerinde şifreleme işlemleri çeşitli şekillerde yapılabilmektedir. Çalışmada kullanılan yöntemlerden bazıları, piksel değerlerinin yerlerini değiştirme, resmin piksel değerlerini değiştirme ya da metin şifrelemede kullanılan şifreleme yöntemlerinin uygulanması şeklindedir. Kullanılan algoritmalar şifrelemede farklı performanslar göstermiştir. Bu performans farkları hız ve görüntülerin şifresinin çözülmesi sırasında yaşanan veri kayıpları şeklinde görülmektedir. Görüntüleri şifrelemek için CodeIgniter içerisinde bulunan şifreleme ve şifre çözme kütüphanelerinden yararlanılmıştır. Bu kütüphane içerisinde AES-128, AES-192, AES-256, DES (Data Encryption Standart), TripleDES (Triple Data Encryption Standart), Twofish, Blowfish, Rijndael-128, Rijndael-192, Rijndael-256, Camellia-128, Camellia-192, Camellia-256, CAST-128 (Stafford Taveres ve Carlisle Adams adlarının ilk harflerini temsil eder), CAST-256, Loki97, GOST (Government Standard), XTEA (Tiny Encryption Algorithm), RC2 (Rivest's Cipher), SaferPlus (Secure and Fast Encryption Routine Plus), Serpent, Seed şifreleme algoritmaları bulunmaktadır. Bu şifreleme kütüphanesi varsayılan olarak AES-128 şifresini SHA512 (Secure Hash Algorithm) HMAC (Hash Message Authentication Code) kimlik doğrulaması ile kullanır. HMAC tabanlı anahtar türetme fonksiyonudur. Bu fonksiyonda iki ayrı anahtar türetilir, biri şifreleme için diğeri kimlik doğrulama için kullanılmaktadır. Bunun yanında görüntü farklı uzantı ile kaydedilerek erişim de kısıtlanmaktadır. Uygulama sırasında kullanılan temel şifreleme süreci Şekil 2 ile gösterilmiştir.

AES şifreleme algoritması şifrelemek ve şifre çözme işlemlerinde 128, 192 ve 256 bit uzunluğunda anahtarlar kullanmaktadır. AES-128, bir mesaj bloğunu şifrelemek ve şifresini çözmek için 128 bitlik bir anahtar uzunluğu kullanırken, AES-192 mesajları şifrelemek ve şifresini çözmek için 192 bit anahtar uzunluğu ve AES-256 256 bit anahtar uzunluğu kullanır. Her bir şifre, sırasıyla 128, 192 ve 256 bitlik kriptografik anahtarlar kullanarak verileri 128 bitlik bloklar halinde şifreler ve şifresini çözer (Selent, 2010).

DES, NIST tarafından yayınlanan ilk şifreleme standartıdır. DES, 56 bitlik bir anahtar kullanır ve 64 bitlik giriş bloğunu 64 bitlik bir çıkış bloğuna eşler. DES'in zayıflıklarından yararlanan birçok saldırı ve yöntem bulunmaktadır (Thakur ve Kumar, 2011).

TripleDES olarak bilinen Üçlü Veri Şifreleme Standardı, simetrik bir anahtar blok şifresidir. Adından da anlaşılacağı gibi algoritma DES algoritmasını şifreleme, şifre çözme ve anahtar oluşturma süreçlerinde üç kez kullanır. TDES, şifreleme ve şifre çözme işlemleri için 56 bitlik üç anahtar kullanır. Üç kere tekrarlanması nedeniyle DES algoritmasına göre daha yavaştır (M. Kumari, Gupta, ve Sardana, 2017).



Şekil 2. Şifreleme Süreci

Rijndael algoritması 128, 192 veya 256 bitlik simetrik anahtarlar kullanarak sırasıyla 128, 192 veya 256 bitlik blokları şifreleyen 10, 12, 14 turluk bir blok şifresidir. Rijndael, çok çeşitli ortamlarda hem donanım hem de yazılım platformlarında tutarlı bir şekilde iyi performans göstermektedir. Rijndael içyapısında S-Box'ları da kullanmaktadır (Verma ve Singh, 2012).

Twofish, 128 bitlik simetrik bir anahtar blok şifresidir. Twofish, 256 bite kadar değişken uzunluklu bir anahtarı kabul eder. Twofish tasarımında anahtara bağlı S kutuları, maksimum ayrılabilir mesafe (MDS) matrisi ve sözde Hadamard dönüşümü (PHT) bulunur (Verma ve Singh, 2012).

CAST-256 şifresi, 128 bitlik blok boyutuna sahip simetrik blok şifresidir ve AES için alternatif olarak sunulmuştur. CAST-256'nın tasarımı, 64 bitlik bir blok şifreleme olan CAST-128 şifresinden türetilmiştir ve bu önceki şifrenin analizinin sonuçlarından yararlanılarak ortaya konmuştur. AES'in büyük blok boyutu gereksinimi nedeniyle, CAST-256'nın mimarisini CAST-128'de kullanılan klasik Feistel yapısından değiştirilmiştir (Adams, Heys, Tavares, ve Wiener, 1999).

128-bit blok boyutuna sahip blok şifresi Camellia, sırasıyla Camellia-128, Camellia-192 ve Camellia-256 olarak adlandırılan 128, 192, 256 değişken anahtar uzunluklarına sahiptir. Güvenlik açısından ele alındığında, Camellia, yüksek dereceli diferansiyel saldırı başta olmak üzere çeşitli saldırı çeşitlerine karşı kriptanalistlerden ilgi görmüştür (Dong, Li, Jia, ve Wang, 2015).

Tablo 1

Şifreleme Algoritmalarının Karşılaştırılması

Algoritma Adı	Anahtar Uzunluğu	Blok Boyutu	Devir	Çoklu okuma özelliğine sahip 4 çekirdekli bir makinede 1024 MB / sn'ye, saniyede 2^{30} bayta eşittir. Böyle bir bilgisayar saniyede 2^{x} blok şifreleyebilir. Bu, saniyede 2x farklı şifreleme anahtarını deneyebileceği anlamına gelir. X Değerleri:	Çok çekirdekli ve yüksek RAM hızına sahip bir bilgisayarın bir yılda arayabileceği anahtar sayısı 31,557,600 * 2^{key} (Bir yıl = 31,557,600 saniye)	Yıllarla ölçüm birimiyle Brute Force saldırısını gerçekleştirmek için geçen süre: $2^{(anahtar\ uzunluğu-1)}$/anahtar sayısı
AES	128-192-256	128	10--12--14	128 = 2^7 --> 30-7 =23	31557600* 2^{23}	6.42711E+23
DES	56	64	16	64 = 2^6 --> 30-6 =24	31557600*2^{24}	68.04965042
TripleDES	168	64	48	64 = 2^6 --> 30-6 =24	31557600*2^{24}	3.53334E+35
BlowFish	32 -448	64	16	64 = 2^6 --> 30-6 =24	31557600*2^{24}	4.05608E-06
CAST	128	64	12--16--20	64 = 2^6 --> 30-6 =24	31557600*2^{24}	3.21355E+23
Rijndael	128-192-256	128	10--12--14	128 = 2^7 --> 30-7 =23	31557600* 2^{23}	6.42711E+23
GOST	256	64	32	64 = 2^6 --> 30-6 =24	31557600*2^{24}	1.09352E+62
Twofish	128-192-256	128	16	128 = 2^7 --> 30-7 =23	31557600* 2^{23}	6.42711E+23
CAST	40 -128	64	12--16	64 = 2^6 --> 30-6 =24	31557600*2^{24}	0.001038355
Loki97	128-192-256	128	16	128 = 2^7 --> 30-7 =23	31557600* 2^{23}	6.42711E+23
SaferPlus	64	128	6--8	128 = 2^7 --> 30-6 =23	31557600* 2^{23}	34841
Serpent	128-192-256	128	32	128 = 2^7 --> 30-6 =23	31557600* 2^{23}	6.42711E+23
XTEA	128	64	64	64 = 2^6 --> 30-6 =24	31557600*2^{24}	3.21355E+23
RC2	8-128	64	16	64 = 2^6 --> 30-6 =24	31557600*2^{24}	2.41761E-13
Camellia	128-192-256	128	18--24	128 = 2^7 --> 30-6 =23	31557600* 2^{23}	6.42711E+23
Seed	128	128	16	128 = 2^7 --> 30-6 =23	31557600* 2^{23}	6.42711E+23

Blowfish 64 bitlik blok şifreleme kullanan simetrik anahtar şifreleme algoritmasıdır, 32 ve 448 uzunluğunda anahtarlar kullanır. Yapılan araştırmalarda Blowfish algoritmasının kullanılan diğer yaygın şifreleme algoritmalarından daha iyi performansa sahip olduğu görülmüştür. Algoritma donanım uygulamalarında da kullanılmaktadır (Nie ve Zhang, 2009).

GOST şifreleme algoritması, Rusya Federasyonu'nda bir devlet şifreleme standardıdır. Rusya Federal Güvenlik Servisi tarafından ticari sırlar, kişisel veriler gibi sınırlı ve güvenli dağıtılması gereken veriler için kriptografik koruma sistemleri olarak önerilmiştir. Bu algoritma, Feistel şemasına uyan simetrik bir blok şifresidir. 64 bitlik veri blokları girişe gönderilir ve 256 bitlik anahtarla 64 bitlik şifrelenmiş veri bloklarına dönüştürülür (Babenko, Ishchukova, ve Maro, 2013).

LOKI97, 128-bit veri bloklarını 128, 192 veya 256-bit anahtar kullanarak şifreleyen özel bir anahtar blok şifresidir. 128 bitlik düz metin giriş değerini iki 64 bit kelimeye bölerek şifreleme işlemini başlatır (Brown ve Pieprzyk, 1998).

SaferPlus algoritması, Safer K-64, Safer K-128, Safer SK-128 şifrelerini içeren mevcut Safer şifreleme ailesine dayanmaktadır. Tüm algoritmalar bayt odaklı blok şifreleme algoritmalarıdır. İlk olarak, istenen difüzyon için doğrusal dönüşümler ve zayıf anahtarlardan kaçınmak için eğilim (bias) vektörlerini kullanırlar (Sharmila ve Neelaveni, 2009).

XTEA algoritması, 128-bit anahtar uzunluğunda 64-bitlik şifreleme algoritmasıdır ve TEA algoritmasındaki zayıflıkları gidermek için geliştirilen bir blok şifreleme algoritmasıdır (Ciflikli ve Aba, 2018).

RC2, Ron Rivest tarafından RSA DataSecurity isimli şirket için 1989 yılında tasarlanmış bir blok şifreleme algoritmasıdır. RC2, 64 bitlik bir blok boyutuyla DES'in yerine geçmesi için tasarlanmıştır. RC2'nin önemli bir özelliği, etkin anahtar boyutunun kullanıcıya sunduğu

esnekliktir. Bu artık birçok blok şifreleme önerilerinin ortak bir özelliği haline gelmiştir ve ticari uygulamalarda önemli olduğu kanıtlanmış bir özelliktir (Knudsen, Rijmen, Rivest, ve Robshaw, 1998).

Bu çalışmada kullanılan algoritmaların özellikleri ve kaba kuvvet (Brute Force) saldırısı karşısında kırılma süreleri Tablo 1'de gösterilmiştir. Farklı özelliklere sahip bu algoritmaların Brute Force saldırılarına karşı dayanaklılık süreleri değişmektedir. Çok çekirdekli ve yüksek RAM hızına sahip bir bilgisayarda tek bir çekirdekte yaklaşık 120 MB / sn'de şifre çözülebilmektedir (ScramBox, 2016). Hızlar bilgisayarlar ve şifreleme uygulamalarına göre değişebilmektedir. Hesaplama kolaylığı için rakamı en yakın iki güce yuvarlanarak çekirdek başına 128 MB / sn olarak kabul edilmiştir. Çoklu okuma (Hyperthreading) özelliğine sahip 4 çekirdekli bir makinede (8 eşzamanlı iş parçacığı), 1024 MB / sn'ye, saniyede 2^{30} bayta eşittir. Böyle bir bilgisayar saniyede 2^{x} blok şifreleyebilir. Bu, saniyede 2^{x} farklı şifreleme anahtarını da deneyebileceği anlamına gelmektedir (ScramBox, 2016). Örneğin, AES 128 baytlık bir blok boyutu kullanır ve 128 bayt 2^{7} 'e eşittir. Bu nedenle ortalama olarak çok çekirdekli ve yüksek RAM hızına sahip performanslı bir bilgisayar saniyede $2^{(30-7)} = 2^{23}$ blok şifreleyebilir. Bu, saniyede 2^{23} farklı şifreleme anahtarını da deneyebileceği anlamına gelmektedir. Bir yıl 31,557,600 (60(saniye)* 60(dakika)* 24(saat)* 365,25(gün)) saniyedir. Çalışma performansı yüksek bir bilgisayar 31,557,660 * blok sayısı kadar şifre denemesi yapabilmektedir. Brute Force saldırısında şifrenin kırılmasını hesaplamak için anahtar uzunluğundan yararlanılmaktadır. Hesaplama sonuçları Tablo 1'de detaylandırılmıştır.

3.3. Güvenlik Önlemleri

Geliştirme sırasında CSRF (Cross Site Request Forgery), XSS (Cross Site Scripting), SQL enjeksiyon (Structured Query Language Injection), temel yol (Basepath / Current Directory) fonksiyonu, karma şifre (Hash Password) gibi güvenlik önlemlerine de dikkat edilmiştir. CSRF, kötü amaçlı sayfaların oluşturduğu HTTP (Hyper Text Transfer Protocol) isteklerine oturum tanımlama bilgilerinin eklenmesiyle etkinleştirilir.

CSRF'ye karşı olası bir savunma, güvenliğe duyarlı her HTTP isteğinin yönlendiren (Referrer) başlığının içeriğini kontrol etmek ile gerçekleştirilir (Calzavara, 2020). Bu başlık, isteği gönderen sayfanın URL (Uniform Resource Loader)'sini içerir.

Siteler Arası Komut Dosyası saldırısı olarak bilinen XSS, bugün web dünyasında tanımlanan en önemli güvenlik açığıdır. XSS saldırıları, bir saldırganın kötü niyetli kodu ve bozuk kaynağı kullanıcı tarayıcısında çalıştırmasına izin verir ve bu da çerezlerin çalınması, şifre saldırısı, kredi / banka kartı numaralarının ele geçirilmesi gibi durumlara neden olabilir. Dünya çapında XSS araştırmacı uzmanları ve sektör uzmanları tarafından en yaygın web uygulaması güvenlik açığı olarak kabul edilir (Vijayalakshmi ve Syed Mohamed, 2020). Güvenlik tedbirlerine dikkat edilmeden geliştirilen web uygulamaları genellikle XSS saldırısına ve diğer güvenlik açıklarına maruz kalır. Buna karşılık, bilinen web uygulamaları güvenlik açıklarını azaltmak için on yıldan fazla bir süredir çeşitli araçlar geliştirmekte ve güvenlik alanındaki araştırmalarına devam etmektedir.

SQL enjeksiyon güvenlik açıkları web uygulamaları için en ciddi tehditlerden birisi olarak tanımlanmaktadır. SQL enjeksiyonu kullanıcı tarafından sağlanan verilerin SQL kodu olarak değerlendirileceği şekilde bir SQL sorgusuna dahil edildiği bir kod yerleştirme saldırı biçimidir. SQL enjeksiyonuna karşı savunmasız olan uygulamalar bir saldırganın veritabanlarına tam erişim sağlamasına izin verebilir. Bu veritabanları genellikle hassas kullanıcı bilgileri içerdiğinden ortaya çıkan güvenlik ihlalleri arasında kimlik hırsızlığı, gizli bilgilerin kaybı ve dolandırıcılık yer almaktadır. Bazı durumlarda saldırganlar web uygulamasını barındıran sistemin denetimini ele geçirmek ve sistemi bozmak için bir SQL enjeksiyon kullanmaktadır (Halfond, Viegas, ve Orso, 2006).

4. Bulgular

Sistemin geliştirilmesinde PHP dili kullanılmıştır. PHP Windows, MAC OS, Unix gibi platformlarda çalışabilmektedir. Hızlı çalışması nedeniyle performans açısından da avantaj sağlamaktadır. MySQL veritabanı ile PHP yüksek performans ve güvenli bir şekilde çalışmaktadır. PHP diğer veritabanları ile de çalışabilmektedir, çalışmada MySQL tercih edilmiştir. MySQL, dünyanın en popüler açık kaynak veritabanlarından biridir. Güvenliği ve kullanım kolaylığı nedeniyle Facebook, Twitter, Youtube gibi yüksek profilli web siteleri tarafından tercih edilmektedir. MySQL içerisinde bulunan özellikleri ile geliştiricilere kolaylık sağlamaktadır. Çalışmada şifrelenen görüntüler doğrudan JPEG, JPG, PNG vb. gibi uzantılar ile saklanmamaktadır. Şifrelenmiş dosyaların uzantısı biliniyorsa dosya yapısına uygun saldırılar yapılabilmektedir. Bu sebeple şifrelenecek görüntü dosyalarının kendi uzantılarını kullanmak yerine bu

görüntüler şifrelendikten sonra farklı bir dosya uzantısı (.mdv) ile kaydedilmiştir. Görüntüleri ele geçirmek isteyen kötü niyetli kişiler ".mdv" uzantılı bir dosya ile karşılaştıklarında bunun bir görüntü dosyası olduğunu doğrudan tespit edemeyecektir. Sistemde şifrelenmiş görüntüler farklı bir uzantı ile saklanarak görüntülerin açılması da sistem tarafından engellenmiştir.

Yapılan çalışmada web arayüzünün tasarımında Bootstrap ve HTML-CSS kullanılmıştır. Bootstrap, geliştiricilere önceden hazırlanmış ve web sitesinde doğrudan kullanılabilir durumda kodlar içeren kütüphaneler sunmaktadır. Bu nedenle Bootstrap kullanımı geliştirme sırasında zaman tasarrufu sağlamıştır. Detaylı dokümanı Bootstrap kullanacaklara tüm detayları sunmaktadır. Bootstrap kodlarının özelleştirilmesi için de HTML-CSS kullanılmıştır. Geliştirme sırasında tercih edilen programlama dilleri, ortamı ve kütüphaneler kullanım kolaylığı ve zaman tasarrufu sağlamaları nedeniyle tercih edilerek kullanılmışlardır.

Web tabanlı sitelerde gözlenen güvenlik açıklarına bakıldığında bu açıklar çoğunlukla XSS, CSRF, SQL enjeksiyon şeklindedir. Bu güvenlik açıkları kötü niyetli kişiler tarafından veri hırsızlığı ve veri kaybı gibi sorunlara neden olabilmektedir. Günümüzde pek çok site bu güvenlik açıklarına karşı korumasız durumdadır. Yapılan çalışma ile bu güvenlik açıklarına karşı gerekli önlemler alınmıştır. CodeIgniter içinde yer alan güvenlik önlemleri aktifleştirilerek gerekli tedbirler alınmıştır. Bu sayede belirtilen güvenlik açıklarına karşı algoritmaların değerlendirilmesi için güvenli ve güvenilir bir sistem geliştirilmiştir.

Bulut depolama alanı olarak AWS tercih edilmiştir. AWS kendi içerisinde gerekli güvenlik önlemleri bulundurması, esnek kullanımı, büyük ölçekli kullanıcı kitlesine sahip olması, veri taşınması sırasında da şifreleme işlemlerini gerçekleştirmesi nedeniyle sisteme entegre edilmiştir. Kullanıcı bir AWS hesabına sahipse geliştirilen sistem üzerinden bulut depolama yapabilmektedir. Kullanıcı AWS hesabına ait anahtarla geliştirilen sistem üzerinden güvenli bir şekilde bulut depolama yapabilmektedir.

Önerilen sistem ile kullanıcılar görüntülerini şifreleyerek görüntülerini güvenli şekilde saklama imkanı bulmaktadır. Görüntülerin şifrelenmesinde CodeIgniter şifreleme ve şifre çözme kütüphaneleri içinde yer alan AES-128, AES-192, AES-256, DES, TripleDES, Twofish, Blowfish, Rijndael-128, Rijndael-192, Rijndael-256, Camellia-128, Camellia-192, Camellia-256, CAST-128, CAST-256, Loki97, GOST, XTEA, RC2, SaferPlus, Serpent, Seed şifreleme algoritmaları kullanılmıştır. Bu algoritmaların kullanımına bakıldığında her birinin şifreleme yapısı farklıdır. Farklı boyutta ve çözünürlükteki görüntülerde algoritmalar farklı performanslar göstermektedir. Algoritmalar OpenSSL (Open Secure Socket Layer) ve MCrypt PHP kütüphaneleriyle çalışmaktadır. Şifrelemenin sorunsuz bir şekilde sağlanması için kullanılan PHP sürümünün bu iki kütüphaneye sahip olması gerekmektedir. Eski PHP sürümlerinde algoritmaların bazıları çalışmamaktadır. Bu durum karşılaştırma işlemlerinde sorunlara neden olabilmektedir. Performansları süre açısından test etmek için yedi yüz görüntü üzerinde sistemdeki şifreleme ve şifre çözme algoritmaları için ölçüm yapılmıştır. Ölçüm sonuçlarına Tablo 2'de yer verilmiştir. Şifreleme algoritmalarının ortalama şifreleme sürelerini hesaplamak için veriseti araştırması yapılmıştır. Fakat bulunan veri setleri çok küçük boyutlarda ve çözünürlükte oldukları için tercih edilmemiştir. Görüntülerden oluşan test veriseti internet üzerinden görüntü toplanarak oluşturulmuştur. Bu görüntü test setinde yedi yüz görüntü yer almaktadır. Görüntülerin en küçük olanı 562 KB, en büyük olanı 6,8 MB büyüklüğündedir (Tablo 3).

Sistemde görüntü şifreleme süre hesaplaması her bir algoritma için ayrı ayrı gerçekleştirilmiştir. Şifreleme işleminde yedi yüz resim kullanılmıştır. Kullanılan resimlerin özellikleri Tablo3'te yer almaktadır. Kullanılan algoritmaya göre şifreleme süreleri farklılık göstermektedir. Tablo 2'de algoritmalara göre bir görüntünün şifrelenmesi için harcanan maksimum - minimum süreye ve yedi yüz resmin şifrelenmesi için geçen toplam süreye milisaniye cinsinde yer verilmiştir.

Tablo 2

Şifreleme Algoritmalarının Görüntü Şifreleme Sürelerinin Milisaniye Cinsinden Karşılaştırılması

Sıralama Değeri	Algoritma Adı	Maksimum Değer (ms)	Minimum Değer (ms)	Toplam Değer (ms)
1	AES-128	0.0705	0.0023	17.3441
2	Twofish	0.1211	0.0037	31.0069
3	Rijndael-192	0.1383	0.0044	35.9238
4	CATS	01320	0.0042	35.9659
5	Blowfish	0.1323	0.0043	35.0257
6	Rijndael-256	0.1588	0.0052	39.1585
7	Rijndael-128	0.1508	0.0049	39.9340
8	CAST-256	0.1510	0.0047	41.2490
9	CAST-128	0.1791	0.0048	41.6192
10	DES	0.2033	0.0056	47.7055
11	Serpent	0.1804	0.0055	49.1346
12	RC2	0.2921	0.0073	61.3038
13	GOST	0.2595	0.0087	69.1320
14	Loki97	0.2584	0.0083	72.5070
15	SaferPlus	0.2842	0.0087	79.1928
16	TripleDES	0.3863	0.0119	100.032

Tablo 3
Şifrelemede Kullanılan Görüntülerin Özellikleri

Görüntü Sayısı	Maksimum Boyut	Minimum Boyut	Renkli/Gri
700	6,8 MB	562 KB	Renkli

5. Sonuçlar

Yapılan çalışmada öncelikli olarak güvenlik alanında yaşanan mağduriyetler incelenmiştir. Yapılan araştırmalarda son zamanlarda hesap hırsızlığı, kişilerin görüntülerinin çalınması gibi durumların oldukça çok olduğu görülmüştür. Bu saldırıların çoğunlukla sitelerdeki güvenlik açıkları, yeterli güvenlik önleminin olmadığı ortamlarda kötü niyetli kişiler tarafından gerçekleştirildiği gözlemlenmiştir. Bu durum güvenilir ve güvenli sistemlere olan ihtiyacı artırmıştır. İnsanların fotoğraflarını, videolarını ve benzeri bilgilerini saklamak için kullandıkları depolama alanları incelendiğinde, bu depolama alanlarında sistem ile ilgili güvenlik önlemleri varken içerikler üzerinde herhangi bir şifreleme işleminin uygulanmadığı gözlemlenmiştir. Depolama alanlarına yüklenen her içerik için doğrudan şifreleme yapmak her zaman sağlıklı bir yöntem olmayabilir. Çünkü şifrelenmiş içeriğin şifresinin çözülmesi aşamasında veri kayıpları yaşanabilir. Bu durum görüntüde bozulmalara neden olacağı için istenilmeyen bir durumdur. Bu durumda şifrelenecek içeriğin yapısına uygun şifreleme ve şifre çözme algoritmaları tercih edilmelidir. Bu içerik bir metin, görüntü, video, gif vb. şekilde ayrıştırılarak uygun şifreleme işlemleri ile saklanmalıdır.

Çalışmada kullanılan simetrik algoritmalar şifreleme ve şifre çözme süreleri açısından incelendiğinde birbirlerinden farklı performans sergilemişlerdir. Tablo 2 ile bu değerler gösterilmiştir. Uygulanan algoritmaların her biri aynı performansı göstermemiştir. Algoritmaların beklenen şekilde çalışması için yeni PHP sürümlerine şifreleme kütüphanesinin eklenmesi gerekmektedir. Yapılan çalışmada kullanılan şifreleme algoritmalarının görüntü üzerindeki etki durumunu kontrol etmek için görüntülerin şifrelenmiş hali ve şifre çözüm işlemlerinden sonraki halleri kodlama ile kontrol edilmiştir. Kontrol işlemleri resmin piksel değerleri, çözünürlük, boyut bakımından değerleri üzerine gerçekleştirilmiştir. Yapılan kontrolde şifreleme ve şifre çözme işlemlerinin görüntü üzerinde herhangi bir

bozma, piksel değerlerinde değişiklik yapmadığı belirlenmiştir. Böylece kullanıcılar görüntülerinde bozulma ve değişiklik olmadan güvenli şekilde depolama yapabilmektedirler.

Yapılan çalışma kapsamında görüntüler çeşitli simetrik şifreleme algoritmaları ile şifrelenmiştir. İleride yapılacak çalışmalarda, yeni algoritmaların da karşılaştırmaya dahil edilerek incelenmesi hedeflenmiştir. Gelecekte yapılması hedeflenen bir diğer bir çalışma, görüntünün şifrelenmeden önce analiz edilerek boyutu ve çözünürlüğüne uygun bir şifreleme algoritmasının otomatik olarak seçimidir.

Araştırmacıların Katkısı

Bu araştırmada; Merve CEYHAN makalenin oluşturulması, bilimsel yayın araştırması, yöntemin belirlenmesi, uygulanması, grafik ve tabloların oluşturulması ve makale sonuçlarının hazırlanması; Esra N. YOLAÇAN makalenin oluşturulması, sonuçların yorumlanması ve makalenin genel kontrolünün yapılması konularında katkı sağlamışlardır.

Çıkar Çatışması

Yazarlar tarafından herhangi bir çıkar çatışması beyan edilmemiştir.

Kaynaklar

Adams, C., Heys, H., Tavares, S., ve Wiener, M. (1999). *An analysis of the CAST-256 cipher.* Paper presented at the Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No. 99TH8411). doi:https://doi.org/10.1109/ccece.1999.807225

Anandakumar, S. (2015). Image Cryptography Using RSA Algorithm in Network Security. *International Journal of Computer Science ve Engineering Technology, 5*(9), 326-330. Erişim adresi:http://ijcset.net/docs/Volumes/volume5issue9/ijcset2015050902.pdf

Askar, S. S., Karawia, A., ve Alshamrani, A. M. (2015). Image Encryption algorithm based on chaotic economic model. *Mathematical Problems in Engineering, 2015*, 1-10. doi:https://doi.org/10.1155/2015/341729

Babenko, L., Ishchukova, E., ve Maro, E. (2013). GOST encryption algorithm and approaches to its analysis. In *Theory and Practice of Cryptography Solutions for Secure Information Systems* (pp. 34-61): IGI Global. doi:http://doi.org/10.4018/978-1-4666-4030-6.ch002

Brown, L., ve Pieprzyk, J. (1998). *Introducing the new LOKI97 block cipher.* Paper presented at the First AES Candidate Conference. Erişim adresi:http://madchat.fr/crypto/hash-lib-algo/loki97/loki97spec.pdf

Calzavara, S. (2020). Security II-CSRF ve XSSI. Erişim adresi:https://secgroup.dais.unive.it/wp-content/uploads/2020/02/csrf.pdf

Chang, C.-C., Hwang, M.-S., ve Chen, T.-S. (2001). A new encryption algorithm for image cryptosystems. *Journal of Systems and Software, 58*(2), 83-91. doi:https://doi.org/10.1016/S0164-1212(01)00029-2

Chepuri, S. (2017). An RGB image encryption using RSA algorithm. *International Journal of Current Trends in Engineering ve Research (IJCTER), 3*(3), 1-7. Erişim adresi:https://ijcter.com/papers/volume-3/issue-3/an-rgb-image-encryption-using-rsa-algorithm.pdf

Ciflikli, C., ve Aba, K. (2018). TEA ve XTEA Şifreleme algoritmaları için kaos tabanlı kaydırma dizisi oluşturulması ve uygulanması. *3*. Erişim adresi:http://www.set-science.com/manage/uploads/ISAS2018-Winter_0039/SETSCI_ISAS2018-Winter_0039_00187.pdf

Dong, X., Li, L., Jia, K., ve Wang, X. (2015). *Improved attacks on reduced-round Camellia-128/192/256.* Paper presented at the Cryptographers' Track at the RSA Conference. doi:https://doi.org/10.1007/978-3-319-16715-2_4

DuBois, P. (2008). *MySQL*: Pearson Education.

Gajda, W. (2013). *Instant PhpStorm Starter*: Packt Publishing Ltd.

Ghoradkar, S., ve Shinde, A. (2015). Review on image encryption and decryption using AES algorithm. *International Journal of Computer Applications, 975*, 8887. Erişim adresi:https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.740.4766verep=rep1&type=pdf

Goel, A., ve Chandra, N. (2012). A technique for image encryption with combination of pixel rearrangement scheme based on sorting group-wise of RGB Values and explosive inter-pixel displacement. *International Journal of Image, Graphics and Signal Processing, 4*, 16-22. doi:https://doi.org/10.5815/ijigsp.2012.02.03

Guvenoglu, E. (2016). Resim şifreleme amacıyla dinamik s kutusu tasarımı için bir yöntem. *El-Cezeri Journal of Science and Engineering, 3*(2). doi:https://doi.org/10.31202/ecjse.264182

Guvenoglu, E., ve Esin, E. M. (2009). *Knutt / Durstenfeld Shuffle Algoritmasinin Resim Sifreleme Amaciyla Kullanilmasi*. Erişim adresi:https://dergipark.org.tr/en/pub/politeknik/issue/33049/367816

Guvenoglu, E., ve Tuysuz, M. A. A. (2015). *An improvement for Knutt / Durstenfeld algorithm based image encryption.* Paper presented at the 2015 23nd Signal Processing and Communications Applications Conference (SIU). Erişim adresi:https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7130194

Halfond, W. G., Viegas, J., ve Orso, A. (2006). *A classification of SQL-injection attacks and countermeasures.* Paper presented at the Proceedings of the IEEE international symposium on secure software engineering. Erişim adresi:https://www.cc.gatech.edu/fac/Alex.Orso/papers/halfond.viegas.orso.ISSSE06.pdf

Hariyanto, E., ve Rahim, R. (2016). Arnold's cat map algorithm in digital image encryption. *International Journal of Science and Research (IJSR), 5*(10), 1363-1365. doi:https://doi.org/10.21275/art20162488

Kester, Q. A. (2013). Image encryption based on the RGB PIXEL transposition and shuffling. *International Journal of Computer Network ve Information Security,* 5(7). doi:https://doi.org/10.5815/ ijcnis.2013.07.05

Knudsen, L. R., Rijmen, V., Rivest, R. L., ve Robshaw, M. J. B. (1998). On the Design and Security of RC2. In (pp. 206-221): Springer Berlin Heidelberg. doi:https://doi.org/10.1007/3-540-69710-1_14

Kumar, R. R., ve Mathew, J. (2020). Image Encryption:Traditional Methods vs Alternative Methods. *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 1-7. doi:https://doi.org/10.1109/iccmc48092.2020.iccmc-000115

Kumari, M., Gupta, S., ve Sardana, P. (2017). A Survey of Image Encryption Algorithms. *3D Research, 8*, 1-35. doi:https://doi.org/10.1007/s13319-017-0148-5

Kumari, S. (2017). A research Paper on Cryptography encryption and compression techniques. *International Journal of Engineering and Computer Science, 6*. Erişim adresi:http://www.ijecs.in/index.php/ijecs/article/view/3630

Liu, K., ve Dong, L.-j. (2012). Research on cloud data storage technology and its architecture implementation. *Procedia Engineering, 29*, 133-137. doi:https://doi.org/10.1016/j.proeng.2011.12.682

Liu, X., Xiao, D., ve Xiang, Y. (2018). Quantum image encryption using intra and inter bit permutation based on logistic map. *IEEE Access, 7*, 6937-6946. doi:https://doi.org/10.1109/access.2018.2889896

Maniccam, S. S., ve Bourbakis, N. G. (2001). Lossless image compression and encryption using SCAN. *Pattern Recognition, 34*(6), 1229-1245. doi:https://doi.org/10.1016/S0031-3203(00)00062-5

Mukherjee, S. (2019). Benefits of AWS in Modern Cloud. *Available at SSRN 3415956*. doi:https://doi.org/10.2139/ssrn.3415956

Nie, T., ve Zhang, T. (2009). *A study of DES and Blowfish encryption algorithm.* Paper presented at the Tencon 2009-2009 IEEE Region 10 Conference. doi:https://doi.org/10.1109/tencon.2009.5396115

Sakal, H., ve Yıldırım, M. (2016). Görüntü şifreleme için scan paternlerini kullanan hibrit bir yöntem. *Selçuk-Teknik Dergisi, 15*(3), 264-283. Erişim adresi:http://sutod.selcuk.edu.tr/sutod/article/view/353

ScramBox. (2016). How long would it take to brute force AES-256?. Erişim adresi:https://scrambox.com/article/brute-force-aes/

Selent, D. (2010). Advanced encryption standard. *Rivier Academic Journal, 6*(2), 1-14. Erişim adresi:https://www2.rivier.edu/journal/roaj-fall-2010/j455-selent-aes.pdf

Sharma, P., Godara, M., Singh, R., Tech, S. M., ve Sabo, T. (2012). Digital Image encryption techniques: A Review. *International Journal of Computing ve Business Research*, 2229-6166. Erişim adresi:http://researchmanuscripts.com/isociety2012/46.pdf

Sharmila, D., ve Neelaveni, R. (2009). A Proposed SAFER plus security algorithm using Fast Walsh Hadamard transform for Bluetooth technology. *International Journal of Wireless & Mobile Networks (IJWMN), 1*(2), 80-88. Erişim adresi:http://airccse.org/journal/jwmn/1109s6.pdf

Shujun, L., ve Xuan, Z. (2002). *On the security of an image encryption method*. doi:https://doi.org/10.1109/icip.2002.1040103

Siame, A., ve Kunda, D. (2017). Evolution of PHP applications: A systematic literature review. *Int. J. Recent Contributions Eng. Sci. IT, 5*, 28-39. Erişim adresi:https://online-journals.org/index.php/i-jes/article/view/6437

Thakur, J., ve Kumar, N. (2011). DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis. *International journal of emerging technology and advanced engineering, 1*(2), 6-12. Erişim adresi:https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.366.831&rep=rep1&type=pdf

Upadhyaya, A., Shokeen, V., ve Srivastava, G. (2015). *Image encryption: using aes, feature extraction and random no. generation.* Paper presented at the 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions). doi:https://doi.org/10.1109/icrito.2015.7359286

Upton, D. (2007). *CodeIgniter for rapid php application development*: Packt Publishing Ltd.

Varia, J., ve Mathew, S. (2014). Overview of amazon web services. *Amazon Web Services*, 1-22. Erişim adresi:http://cabibbo.dia.uniroma3.it/asw-2014-2015/altrui/AWS_Overview.pdf

Verma, H. K., ve Singh, R. K. (2012). Performance analysis of RC6, Twofish and Rijndael block cipher algorithms. *International Journal of Computer Applications, 42*(16), 1-7. Erişim adresi:https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.4934&rep=rep1&type=pdf

Vijayalakshmi, K., ve Syed Mohamed, E. (2020). Case Study: Extenuation of XSS attacks through various detecting and defending techniques. *Journal of Applied Security Research*, 1-36. doi:https://doi.org/10.1080/19361610.2020.1735283

Yen, J.-C., ve Guo, J.-I. (1999). *A new image encryption algorithm and its VLSI architecture.* Paper presented at the 1999 IEEE Workshop on Signal Processing Systems. SiPS 99. Design and Implementation (Cat.No.99TH8461). doi:https://doi.org/10.1109/sips.1999.822348