T.C.
BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

Yüksek Lisans Tezi

MOBİL CİHAZLARDA GÜVENLİ HABERLEŞME HAKKINDA ŞİFRELEME ALGORİTMALARININ KIYASLANMASI

Sercan BEDİR

Tez Danışmanı
Doç. Dr. Orhan ER

Yozgat 2019

T.C.
BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK ELEKTRONİK MÜHENDİSLİĞİ
ANABİLİM DALI

Yüksek Lisans Tezi

MOBİL CİHAZLARDA GÜVENLİ HABERLEŞME HAKKINDA ŞİFRELEME ALGORİTMALARININ KIYASLANMASI

Sercan BEDİR

Tez Danışmanı
Doç.Dr. Orhan ER

Yozgat 2019

YOZGAT BOZOK ÜNİVERSİTESİ

TEZ ONAY FORMU

T.C.
YOZGAT BOZOK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Enstitümüzün Elektrik-Elektronik Mühendisliği Anabilim Dalı Tezli Yüksek Lisans/Doktora Programı 70111516007 numaralı öğrencisi Sercan BEDİR'in hazırladığı "Mobil Cihazlarda Güvenli Haberleşme Hakkında Şifreleme Algoritmalarının Kıyaslanması" başlıklı tezi ile ilgili tez savunma sınavı, Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği'nin ilgili maddeleri gereğince 08/10/2019 Salı günü saat 11:00'da yapılmış, tezin onayına oy birliği/oy çokluğu ile karar verilmiştir.

Başkan : Dr.Öğr.Ü.Fehim KÖYLÜ

Jüri Üyesi (Danışman) : Doç.Dr.Orhan ER

Jüri Üyesi : Dr.Öğr.Ü.Ahmet Sertol KÖKSAL

ONAY:

Bu tezin kabulü, Enstitü Yönetim Kurulu'nun 17.../10.../19. tarih ve 49. sayılı Enstitü Yönetim Kurulu Kararı ile onaylanmıştır.

17.../10.../2019

Prof. Dr. Mustafa SAÇMACI
Müdür

KYT-FRM-110/00

İÇİNDEKİLER

MOBİL CİHAZLARDA GÜVENLİ HABERLEŞME HAKKINDA ŞİFRELEME ALGORİTMALARININ KIYASLANMASI

Sercan BEDİR

Bozok Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

2018; Sayfa: 60

Tez Danışmanı: Doç.Dr. Orhan ER

ÖZET

Son yıllarda teknolojinin hızla ilerlemesi iletişim alanında da büyük önem taşımakta, getirdiği birçok kolaylığın yanı sıra dezavantajları da taşıyarak büyük tehdit oluşturmaktadır. Bu dezavantajların en önemli konusu haberleşme esnasında iletilen bilginin güvenliğidir. Özellikle mobil cihazların da bu teknolojinin içerisinde yer almasıyla beraber bilgisayarlar tarafından yapılan her işlemin mobil cihazlar üzerinde yapılabilmesi ile bu güvenlik zafiyeti bu ortamlar için de son derece önemli bir durum haline gelmiştir.

Bu çalışmada mobil uygulama üzerinden güvenli bir veri haberleşme ortamı sağlamak amaçlanmıştır. Bu güvenli iletim hattını oluşturmak için bilgi güvenliğinin önemli kavramlarından biri olan şifreleme bilimi kullanılacaktır. Şu ana kadar ortaya çıkan bir şifreleme metodu kullanılarak kullanıcının amacına uygun bir iletim hattı oluşturulacaktır. Ayrıca özgün bir şifreleme metodu ile başka bir mesajlaşma ortamı tasarlanıp, güvenlik, hız gibi kullanıcının kullanım amacına uygun bir iletim hattı ortaya çıkarılacaktır. Oluşturulacak olan bu metotlar birbirleri ile karşılaştırılacak ve güvenli iletişim hattı oluşturabilmek için avantaj ve dezavantajları ortaya konulacaktır. Böylece şifreleme sistemlerinin bilgi güvenliği açısından önemi bu çalışmada desteklenecektir.

Çalışma içerisinde geliştirilen kodlar Java programlama dili ile geliştirilmiştir. Geliştirilen algoritmalar neticesinde bu çalışma ileri ki dönemlerde şifreleme adına yapılacak çalışmalara ışık kaynağı olacaktır.

Anahtar Kelimeler: Bilgi güvenliği, mobil cihazlar, şifreleme, güvenli haberleşme

MOBİL CİHAZLARDA GÜVENLİ HABERLEŞME HAKKINDA ŞİFRELEME ALGORİTMALARININ KIYASLANMASI

Sercan BEDİR

Bozok Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı
Yüksek Lisans Tezi

2019; Sayfa: 60

Tez Danışmanı: Doç.Dr. Orhan ER

ABSTRACT

In recent years, the rapid progress of technology is also of great importance in the field of communication and poses a great threat by carrying the disadvantages as well as many other convenience. The most important issue of these disadvantages is the safety of information transmitted during communication. Especially when mobile devices are included in this technology, this security weakness has become a very important situation for every operation made by computers by mobile devices.

In this study it is aimed to provide a secure data communication environment through mobile application. Cryptography will be used as one of the important concepts of information security to create this secure transmission line. Using a cryptographic method so far, a transmission line suitable for the user's purpose will be created. In addition, another messaging environment will be designed with a unique encryption method and a transmission line suitable for the user's purpose, such as security and speed, will be created. These methods will be compared with each other and the advantages and disadvantages will be determined in order to create a secure communication line. Thus, the importance of encryption systems in terms of information security will be supported in this study.

The codes developed in the study were developed with Java programming language. As a result of the developed algorithms, this study will be a source of light for the works that will be done in the name of encryption.

Keywords: Information security, mobile devices, encryption, secure communication

TEŞEKKÜR

Bu tezin hazırlanma aşamasında her konuda bana yardımcı olan, beni cesaretlendiren değerli danışmanım Doç. Dr. Orhan ER'e çok teşekkürlerimi sunuyorum. Ayrıca eğitim hayatım boyunca her zaman arkamda duran, her türlü desteğini esirgemeyen, maddi manevi her zaman yanımda olan aileme çok teşekkür ediyor, saygılarımı sunuyorum.

Sercan BEDİR

TABLOLAR LİSTESİ

ŞEKİLLER LİSTESİ

KISALTMALAR LİSTESİ

IBM	**:**	International Business Machines
DES	**:**	Data Encryption Standart
NSA	**:**	National Security Agency
RSA	**:**	Rives,Shamir Aldeman
GF	**:**	Galois Field
MD5	**:**	Message Digest
SHA	**:**	Secure Hash Algorithm
SMS	**:**	Short Message Services
ECDH	**:**	Elliptic Curve Diffie Hellman
IOS	**:**	iPhone OS

1. GİRİŞ

Bilgi güvenliği, elektronik ortamlarda verilerin veya bilgilerin saklanması ve taşınması esnasında bilgilerin bütünlüğü bozulmadan, izinsiz erişimlerden korunması için, güvenli bir bilgi işleme platformu oluşturma çabalarının tümüdür[1]. Bunun gerçekleşebilmesi için, uygun güvenlik politikalarının belirlenmesi ve uygulanmaya konulması gerekmektedir.

Bilgi güvenliği, "bilginin bir varlık olarak hasarlardan korunması, doğru teknolojinin, doğru amaçla ve doğru şekilde kullanılarak bilginin her türlü ortamda, istenmeyen kişiler tarafından elde edilmesini önleme olarak" tanımlanır. Bilgisayar teknolojilerinde güvenliğin amacı ise "kişi ve kurumların bu teknolojileri kullanırken karşılaşabilecekleri tehdit ve tehlikelerin analizlerinin yapılarak gerekli önlemlerin alınmasıdır" [2].

Günümüzde bilişim teknolojilerinin yaygınlaşması ve günlük hayatımızda yapmış olduğumuz iş ve işlemlerin elektronik ortamlarda hızla yapılmaya başlanması, bilgi güvenliğinin sağlanmasını zorunlu hale getirmektedir.

Bilgi güvenliği, bilginin gizliliğinin, bütünlüğünün ve erişilebilirliğinin sağlanması anlamına gelmektedir. Gizlilik, bütünlük ve erişilebilirlik bilgi güvenliğinin temel unsurları olarak değerlendirilebilir [4].

Gizlilik (Confidentiality): Bilginin yetkisiz kişilerce erişilememesidir. Günümüzde mobil iletişim, internet ve diğer iletişim türlerinde gönderilen her bilginin önlem alınmadığı takdirde üçüncü şahıslar tarafından okunabileceği bilinir. Güvenlik bir şekilde sağlandığı takdirde bu şahısların gönderilen veya alınan mesajlara ulaşabilme ihtimali çok zayıftır.

Bütünlük (Integrity): Bilginin doğruluğunun ve eksiksiz olmasının sağlanmasıdır. Bilginin içeriğinin değiştirilmemiş ve hiçbir bölümünün silinmemiş ya da yok edilmemiş olmasıdır. Bu anlamda gönderilen ve alınan mesajların karşılaştırılması gerekir.

Erişilebilirlik (Availability): Bilginin, bilgiye erişim yetkisi olanlar tarafından istenildiği anda ulaşılabilir, kullanılabilir olmasıdır. Erişilebilirlik durumu kullanılabilirlik ile doğru orantılıdır. Erişim sıkıntısı yaşanıldığında bilginin kullanımı mümkün olmamaktadır. Kullanılabilirlik ilkesine bakıldığı üzere her kullanıcı erişim hakkının bulunduğu bilgi kaynağına, yetkili olduğu zaman dilimi içerisinde kesin olarak erişebilmelidir. Veri depolama alanları ile kullanıcılar her an güncel bilgiye erişebilmektedir, ayrıca bu sistemlere hizmet veren cihazların altyapı yedek sistemleri ve yeterliliği, düzenli yedekleme yönetimi ve gerektiğinde en kısa sürede hizmete erişim bu unsurun önemsediği teknik alanlardır.

Bu üç temel unsur birbirinden bağımsız olarak düşünülememektedir. Bilginin gizliliğinin sağlanması o bilginin erişilebilirliğini engellememelidir. Aynı zamanda erişilebilen bilginin bütünlüğünün de sağlanması önemlidir. Eğer bir bilgi için sadece gizlilik sağlatılıyor ve bilgiye erişim engelleniyor ise kullanılamaz durumda olan bu bilgi bir değer ifade etmeyecektir. Eğer erişimi sağlanıyor ancak bütünlüğü sağlanmıyor ise kurumlar ve kişiler için yanlış veya eksik bilgi söz konusu olacak ve olumsuz sonuçlara neden olabilecektir. Dolayısıyla bilgi güvenliği kavramı temel olarak bu üç unsurun bir arada sağlanması demektir [5].

McCumber[6] modeline göre bilgi güvenliğinin üç unsuru bir sistem üzerinde incelenmiştir. Bu modele göre gizlilik alanı için bilgi gizliliğinin sağlanması amacıyla, bilginin transferi ve depolanması süreçlerinde kripto kullanımı önerilmektedir.

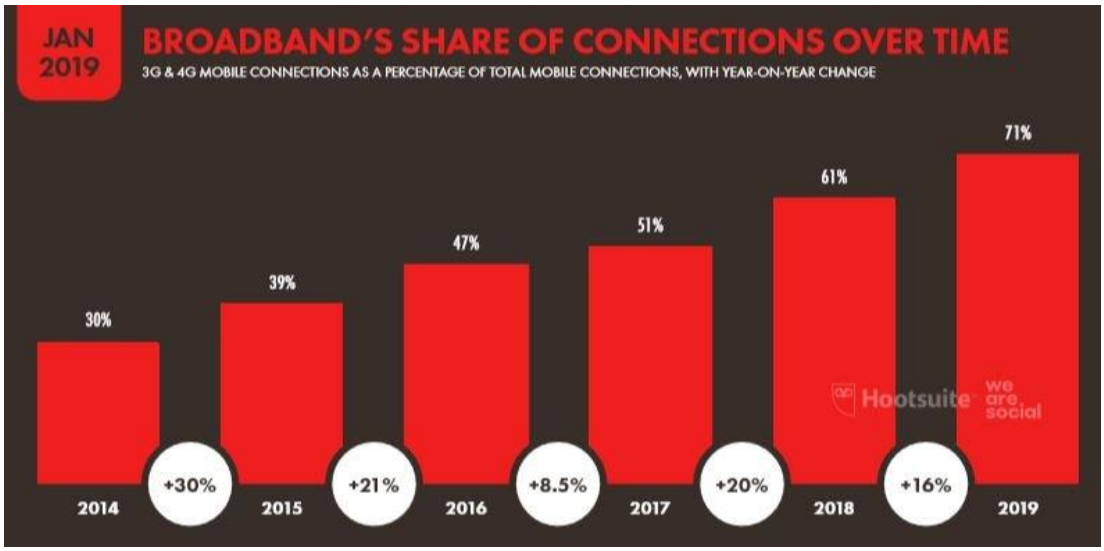
Bütünlük güvenlik sorunları için kısaca kazara veya kasıtlı olarak bilginin orijinal halinden ayrılıp farklı bir veri haline gelmesi tanımı yapılabilir. McCumber'e göre bilginin bütünlüğü; kripto çözümleri, karşılaştırmalı analiz, inkâr edememe, kimlik doğrulama ve erişim kontrolü süreçlerini de içine almaktadır.

Teknolojinin hızla gelişmesiyle beraber her geçen yıl bilgi güvenliği konusunda tehditler artmakta ve aynı zamanda bu tehditlere gösterilecek savunma hassasiyeti de artmıştır. Bu sayede her geçen dönemde yeni savunma yöntemleri çıkmaktadır. Bu sayede bilginin koruması amaçlanmıştır. Günümüzde hızla devam eden bilgisayar ve iletişim teknolojilerindeki gelişme eğitimden ticarete, devlet sektöründen özel

sektöre, eğlenceden alışverişe kadar birçok alanda kalıplaşmış anlayışları değiştirmiş ve insanlara yeni bir bakış açısı kazandırmıştır.

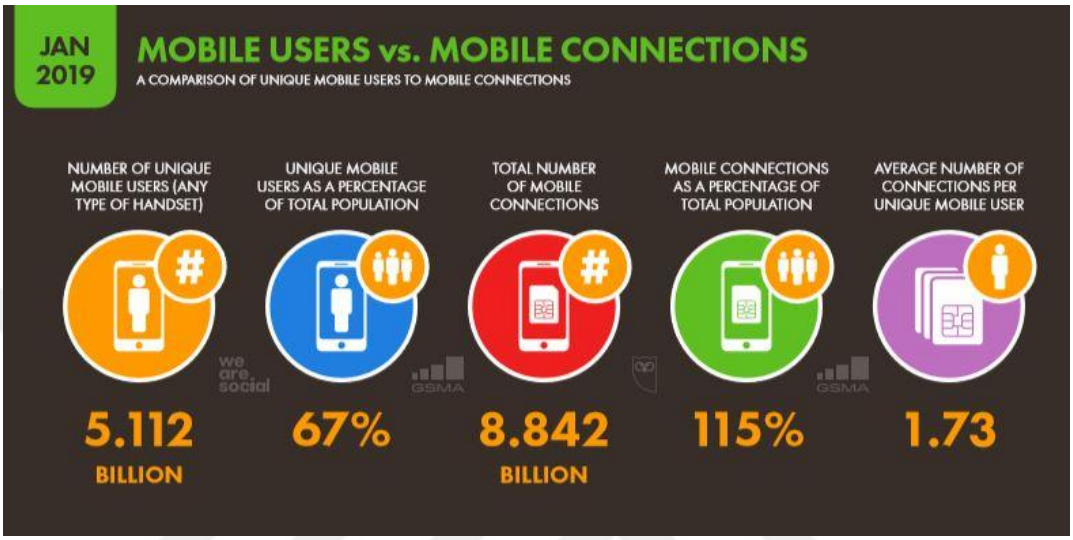
Bilişim dünyasının hızla gelişimi son yıllarda mobil cihazlara da yansımıştır ve mobil cihazlar da artık hayatımızın vazgeçilmez parçalarından biri haline gelmiştir. Mobil cihazlar o kadar hızlı gelişmiştir ki günümüzde birçok işlem aynı anda mobil cihazlar sayesinde yapılabilmektedir. Bilgisayarlarla karşılaştırıldığında mobil cihazların yokluğu toplumu daha zararlı bir şekilde etkilemeye yeterdir. Hal böyle olunca mobil cihazlarda tutulan bilginin ve verinin güvenliği daha önemli bir hale gelmektedir. Veri güvenliği artık yavaş yavaş yerini iletişim güvenliğine bırakmaya başlamıştır. Dikkat edecek olursak bir diğer hususta veri boyutunun artması ve cihazların aldığı yeni görünümün arasındaki ilişkidir. Veri boyutları her geçen yıl artmaya devam etmektedir. Buna bağlı olarak gelişen teknolojinin de etkisiyle cihazların görünümü gittikçe daha küçük boyutlar haline gelmeye başlamıştır.

Mobil cihazlar son on yılda boyut olarak daha küçük ancak kullanım kapasitesinde daha güçlü bir hale gelmişlerdir. Şu anda geliştirilen uygulamalar ile çok fazla özelliği gerçekleştirebilmektedirler. Dolayısıyla, cep telefonları basit bir bilgisayarın yapacağı işleri rahatlıkla yapabildikleri için bilgisayarların küçültülmüş hali olarak değerlendirilebilmektedir [7].



Şekil 1.1. Yıllara göre mobil kullanıcı trafiği[8]

Şekil 1,1'de yıllara göre mobil kullanıcı trafiği verileri paylaşılmıştır. 2019 yılı Ocak ayı itibariyle öne çıkan verilere göre her yıl mobil kullanıcı sayısı önemli bir şekilde artış sağlamıştır. Her geçen yıl büyüyen mobil kullanıcı sayısı yıllar geçtikçe büyümeye devam edecektir.



Şekil 1.2. Mobil Kullanıcı İstatistikleri[9]

Şekil 1,2'de dünya genelinde mobil kullanıcı istatistikleri gösterilmektedir. Bu istatistiklere göre dünya genelinde mobil kullanıcı sayısı 5,1 milyar seviyesindedir. Bu, toplam dünya nüfusunun % 67'sine tekabül etmektedir. Ayrıca dünya genelinde 8,8 milyar mobil cihaz bulunmaktadır. Bu ise toplam nüfus oranını aşmaktadır. Yani %115 seviyesinde bir orana denk gelmektedir. Her bir kişiye ortalama 1.73 cihaz düşmektedir. Bu çok yüksek bir orandır.

Mobil cihazlardaki gelişmeler ve bu cihazların kullanımının gittikçe yaygınlaşması, kullanıcıların yapacağı işleri mobil ortamlara yönlendirmektedir. Mobil ortamlarda bilgi güvenliğini sağlayabilmek için her gün yeni yöntemler ve yeni yaklaşımlar geliştirilmektedir [10].

Mobil cihazlarda şu an hâlihazırda bulunan güvenlik duvarları, test uygulamaları gibi güvenliği sağlayacak olan segmentler yeterli olamayabilmektedir. Bunun sebebi mobil kullanımın hızla büyümesi ve kullanım oranının ciddi bir şekilde artmasıdır. Buna bağlı olarak veri akış trafiği de aynı oranda artmaktadır.

Mobil teknolojilerin gelişim hızı ile beraber cep telefonu kullanımının yüksek seviyelere ulaştığı günümüzde, e-devlet uygulamaları ve diğer kurumsal uygulamalar mobil cihazlar üzerinden verilmeye başlanmış, mobil ortamlardan faydalanma oranı gerek bireysel gerekse kurumsal açıdan bakıldığında hızla artmaktadır. Kullanılan uygulamalar zamanla daha çok fonksiyonellik içermekte, kullanıcılara çok daha çeşitli ve kaliteli hizmet vermekte, bununla beraber, teknik açıdan bakıldığında uygulama boyutu büyümekte ve karmaşıklık artmaktadır [11].

Sistemler arası bağlantılarda ya da herhangi iki nokta arasındaki haberleşmede verinin güvenli bir şekilde gittiğinden emin olmak gerekir. Bu yüzden günümüze kadar birçok şekilde güvenli bir iletişimin sağlanması için protokoller geliştirilmiş, sistemler tasarlanmıştır. Ancak son yıllarda gelişen mobil devrim bu uygulamaların ve ya protokollerin çok daha hızlısını ve iyisini mobil cihazlar üzerinde çalıştırılmasını şart koşmuştur. Bunun sebebi daha önce belirtildiği gibi mobil cihazların artık hayatımızda daha fazla yer kaplamasıdır.

Bilgi güvenliğini sağlamanın bir aracı şifrelemedir. Veriler birbirleriyle haberleşirken, kullanıcılar iletişimini sağlarken ortama sunulan bilginin şifrelenmesi ile bilginin güvenliği sağlanabilir. Bugüne kadar birçok şifreleme algoritmaları bulunmuştur ve kullanılmıştır. Bu şifreleme algoritmalarının hepsinin performansları farklıdır ve kullanıldığı ortama göre değişkenlik gösterip kullanıcıya güvenlik imkânı sunmuştur.

Şifreleme gönderilen mesajların güvenliğini sağlayan, gönderici ile alıcı arasına üçüncü kişilerin girmesini engellemeye çalışan ve iletilerin güvenliğini sağlayan araçtır. Şifrelemeyi inceleyen bilim dalına kriptoloji denilmiştir. Günümüzde şifreleme bilimi o kadar önem arz etmektedir ki artık devlet ve ya özel sektör kurum ve kuruluşları ayrı bir alan olarak sadece şifrelemem bilimine yön verip, çalışmalarını bu alanda devam ettirmektedir.

Şifreleme biliminin tarihi incelendiğinde öne çıkan isim şüphesiz Alan Turing olacaktır. 1912 yılında Londra'da doğan Alan Turing bilime ve özellikle şifrelemeye olan katkılarıyla ön plana çıkmaktadır. Modern bilgisayar bilimlerinin kurucusu olarak kabul edilen Turing 2.Dünya Savaşı esnasında çözülemez olarak nitelendirilen

ENİGMA adlı kod makinesini çözmüş, savaşın seyrini değiştirmiş ve savaşın kısalmasına etki etmiştir. ENİGMA adlı cihaz döner disklerden ve daktilo şeklindeki tuş biçiminden oluşan bir şifreleme makinasıdır. Her tuş basıldığında döner diskler çalışır ve daktilodan bir tuş basılmaktadır. Disklerin her dönüşünde farklı bir harf üretilirdi. Disk turları tamamladıktan sonra şifreli metin ortaya çıkmaktaydı. O dönemlerde henüz bilgisayar bile keşfedilmemişken bu şifreli metinlerin çözümü imkânsız hale gelmişti. Alan Turing 1936 yılında bir makale yayınlayarak matematiksel işlemlerin makineler üzerinde nasıl işlem yapabileceğini ön plana attı. Bu fikrini "Bombe" adı verilen şifre kırıcı cihazı geliştirme aşamasında kullanmıştır. O makine icat edilene kadar ENİGMA şifreleri elle çözülüyor ve bu işlem de saatler sürüyordu. Turing'in bu adımı şifreleme bilimlerinin gelişmesine ve daha da önemlisi bilgisayarın icat edilmesine ön ayak olmuştur.

ENİGMA makinesinin temel çalışma prensibi aslında yer değiştirme şifrelemesinde olduğu gibi bir kurala dayanır. Makine 3 temel parçadan oluşmaktadır. Bunlar

- Rotor
- Yansıtıcı Birimi
- Fiş paneli

olmak üzere ayrılmaktadır. Rotor denilen cihazda harflerin üretilmesi ve birbirleriyle yer değiştirilip karıştırılması mümkün olmaktadır. 26 alfabeli bir çalışma sistemi üzerine kurulmuş bu yapı 1/26 oranında tur yapmakta ve harfleri üretmektedir. 3 rotor bulunduğu için 6 kat daha güvenli harf üretimi sağlanır. Yansıtıcı birimi ise birbirlerine simetri şifre oluşturma işlemini gerçekleştirmektedir. Fiş paneli ise rotorlar sonucu üretilen harflerin, şifreleme işlemi yapılmadan önce, yer değiştirme şifrelemesi işleminin yapılmasını sağlamaktadır. Makinenin ilerleyen zamanlarda 5 rotorlusu üretilmiştir.

Kriptoloji bir başka deyişle anlamlı halde olan bir iletinin bir şifreleme aracı kullanılarak anlamsız hale getirilmesi manasını da taşımaktadır. İletişim temelde mesajı yayınlayan kaynak ile mesajı alması planlanan alıcı kaynak arasında gerçekleşen bir süreçtir. Bu süreç içerisinde mesaj bir noktadan diğer noktaya gidene

kadar birçok adıma maruz kalmaktadır. Mesajın bir bütün halde, hızlı bir şekilde ve üçüncül şahısların ulaşamayacağı halde gitmesi önem arz etmektedir.

Kriptoloji basit manada iki uç kaynağın birbiri ile haberleşmesinde güvenliği sağlamasının yanında modern olarak düşünüldüğünde birçok ihtiyacın bu bilim dalı sayesinde karşılandığı tespit edilmektedir. Her ne kadar gizliliğin, bütünlüğün, hızın ve buna benzer unsurların tam anlamı ile sağladığı bir iletişim ortamını tasarlamak zor da olsa kriptoloji sayesinde bazı tedbirler almak mümkündür.

İletişim sistemlerinin günlük hayatımızın vazgeçilmez parçaları haline gelmesiyle, hızlı ve hatasız sayısal bilgi iletimi gerekliliği de önemli bir konu olarak karşımıza çıkmaktadır. Haberleşme sistemlerinde bilginin iletimi için vericiden gönderilen mesaj işareti, işaretin gönderildiği kanaldan ve aynı zamanda verici ve alıcı devrelerinden kaynaklanan işareti bozucu etkilerden dolayı, bozulmaya uğrar ve alıcıda gönderilenden farklı bir bilgi alınabilir. Bu durum haberleşme sisteminin hatalı mesaj göndermesi olarak adlandırılır. Kanal kodlama, sayısal haberleşme sisteminde, alıcıda alınan bilgi bitlerinde meydana gelebilecek bu hataları ortadan kaldırmayı amaçlayan geniş bir araştırma alanıdır [12].

Kriptoloji ve kodlama olayının bilgi iletişiminde amaçları ayrıdır. Kriptoloji gizlilik, bütünlük, erişilebilirlik esaslarını alarak mesajın iletilmesini sağlar. Fakat kodlama olayına bakıldığı zaman o bilginin iletilmesindeki veri kaybını, kayıp oranını ve bu aşamaları inceler.

Kodlama olayı bir verinin uçtan uca başka bir noktaya iletilmesi olayıdır. Bu iletim hattında sayısal sinyal ve işaretlerden yararlanılır. Bu haberleşmenin sağlanmasında 3 temel unsur ön plana çıkmaktadır. Bunlar kaynak, hedef ve iletişim ortamıdır. ‘ şekilde iletişim sağlanabilir. Bunlar seri ve paralel iletişim yöntemleridir.

Seri iletişim ortamında uzun mesafelerde tek bir iletişim ortamı üzerinden gönderilmek istenen bitlerin zaman ön planda tutularak art arda gönderilmesidir. Paralel iletişim ortamında ise kısa mesafeli noktalarda oluşturulan verinin ayrı yollar üzerinden gönderilmesidir. Paralel iletişim ortamı için bilgisayar içindeki bir

haberleşme örnek verilebilirken, seri iletişim için bilgisayarlar arasında oluşan haberleşme örnek verilebilir.

Kodlama işlemi analog ve sayısal şekilde yapılmaktadır. Gönderilecek olan mesaj sinyal olarak gönderilmektedir. Bu sinyal işlemi gönderilme aşamasında veriye kodlanarak çevrilmektedir. Sayısal veya analog veri iki şekilde gönderilir: Sayısal işaret ve analog işaret. Analog iletim verinin içeriğinden bağımsız gönderilmektedir. Uzun mesafelerde zayıflamalar görülebilir. Gerektiğinde veri gönderimi esnasında tekrarlayıcı kullanılabilir. Sayısal iletim ise verinin içeriğine bağlı gönderilir. Yani herhangi bir bozulma veya zayıflamada tekrar düzeltilerek yeniden veri gönderilmelidir. Uzun mesafelerde gerektiğinde düşük hat kullanılarak veri gönderilme işlemi yapılmaktadır. Veri kaybına ve güvenliğine hassasiyetlidir. Bu yüzden sayısal iletim hattını kullanmak haberleşme için avantajlıdır.

Bu tez çalışmasında hedef iki nokta arasında sağlanan iletişimin güvenliği konusunda kullanılan şifreleme türlerinin test edilmesidir. Yapılacak çalışmalar neticesinde bilginin mobil ortamda iletiminin, güvenliği gerektirecek unsurlar doğrultusunda, bir noktadan diğer bir noktaya iletilmesi sağlanacaktır. Bu sağlanırken daha önce birçok yerde kullanılan ve tanımlı olan olan bir şifreleme türü ile bir yöntem geliştirilerek ortaya çıkarılacak olan yeni bir şifreleme türünün arasındaki performans ve bilginin iletimi konusunda göstereceği başarı oranı hesaplanacaktır.

Çalışmada önce bu zamana kadar kullanılmış ve halen kullanılıyor olan birtakım şifreleme algoritmalarının ve performanslarının tanıtılması sağlanacaktır. Ardından haberleşmenin sağlanacağı uygun materyaller tanımlanacak olup bunun için bir geliştirme ortamı hazırlanacaktır.

Uygulamanın yapım aşaması ve performans analizleri çalışmamızda paylaşıldıktan sonra sonuç kısmına gelip sonuçlar paylaşılacaktır ve ardından bu sonuçları literatürdeki diğer çalışma sonuçları ile beraber inceleyip tartışılacak konular ele alınacaktır.

2. ŞİFRELEME ALGORİTMALARI

Şifrelemede kullanılacak olan anahtarların özelliklerine göre farklı algoritmalar geliştirilmiştir. Bu algoritmalar ve anahtarlama türlerine göre iki farklı şekilde kategorize edilmiştir. Bunlar;

1.Simetrik Şifreleme Algoritmaları

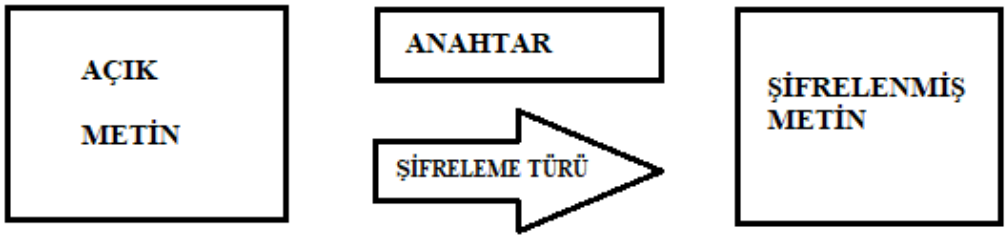
2.Asimetrik Şifreleme Algoritmaları

2.1. Simetrik Şifreleme Algoritmaları

Simetrik şifreleme türünde şifreleme işlemleri genel olarak anahtarlama işlemi ile yapılır. Anahtarlama işleminde bilginin gönderilme aşamasında güvenliği sağlayacak olan kavramı göndericinin ve alıcının bilmesi sağlanmaktadır. Bir tarafın anahtar ile şifrelediği bilgi aynı anahtar tarafından diğeri bir tarafın bilgiyi çözmesi işlemi gerçekleşmektedir.

Bazı şifreleme sistemlerinde bir anahtar kullanılırken bazı sistemlerde ise şifreleme karmaşık hesaplamalar sonucu elde edilmektedir. Bunun sebebi genel olarak anahtar işleminde iki kişi tarafından bilinen anahtarın üçüncü bir kişiye verilme ihtimalinin fazla olmasıdır. Neticede iki kişinin bildiği anahtarı üçüncü bir kişi de öğrenebilir ve bunun sonrasında güvenlik zafiyeti oluşur. Bu güvenlik zafiyetinin ortadan kaldırılmasını sağlayan güvenli anahtar algoritmaları bulunmaktadır. Bu algoritmalar sayesinde anahtar üçüncü bir tarafın eline geçse dahi güvenli haberleşme sağlanabilir. Eğer iletişim değil de kişisel bilgilerin güvenliği için şifreleme sağlanıyorsa kişinin kendine ait bir şifreleme metodu oluşturması kaçınılmaz olacaktır. Bu sayede oluşturulacak metot ile kendinden başka herhangi bir tarafın bu gizliliğe ulaşması mümkün olmayacaktır.

Simetrik şifreleme algoritmalarının en büyük avantajı diğer algoritmalara göre hızlı olmasıdır. Bunun yanı sıra gizlilik konusunda daha başarılıdır ve anahtar için kullanılan bit sayısı küçük olduğundan daha az yer kaplar. Kullanılan basit işlerden dolayı elektronik olarak kullanılabilirliği ve uygulanabilirliği daha kolaydır



Şekil 2.1 : Şifreleme Şeması

Şekil 2.1'de şifreleme aşamasını gösteren şema verilmiştir. Bir anahtar ile şifreleme algoritması kullanılarak metin şifrelenmektedir.

Şekil 2.2: Şifrelenmiş metnin çözülmesi

Şekil 2.2'de ise şifrelenmiş metnin çözülme şeması verilmiştir. Burada aynı anahtar ve şifreleme türünün kullanıldığına dikkat edilmelidir. Sonuç olarak bir anahtar ve bir şifreleme türü kullanılarak metin şifrelenmiş ve tekrar çözülmüştür.

Başlıca simetrik şifreleme algoritmaları aşağıda verilmiştir.

- DES
- AES
- Blowfish

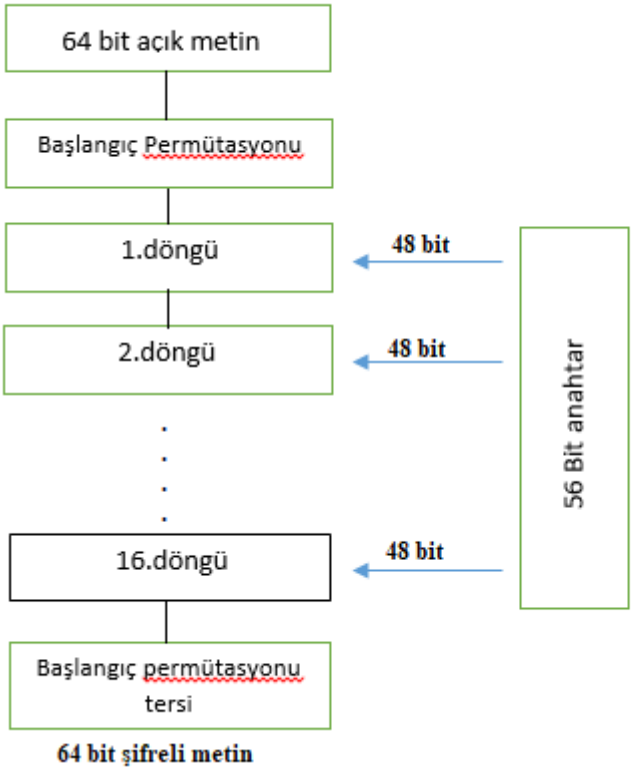
2.1.1 Veri Şifreleme Standardı (Data Encryption Standart- DES)

Verileri şifrelemek ve bu verilerin şifrelerinin çözülmesi amacıyla 1971 yılında IBM tarafından geliştirilen şifreleme sistemidir. Horst Feistel başkanlığında IBM ekibi önce Lucifer adı verilen şifrelemeyi bulmuş ve bu sistemin geliştirilmesi ile DES ortaya çıkmıştır.128 bitlik anahtar uzunluğuna sahip Lucifer sistemi NSA uzmanları tarafından 56 bite kadar düşürüldükten sonra Data Encryption Standart adını almıştır. 1999'da son olarak yapılan çalışmalar neticesinde 3-DES olarak yeni bir standart haline gelmiştir.

DES hem yazılımsal hem donanımsal olarak geliştirilmiştir fakat yazılımsal yöntemi oldukça yavaş olduğu için kullanılmamaktadır.

DES bir blok şifreleme metodudur. Yani şifrelenecek olan veri bloklara ayrılır ve bloklar halinde şifrelenmektedir. Bu veri çözülürken aynı şekilde bloklar halinde çözülerek aynı işlemler tekrarlanarak eski haline getirilmektedir.

DES algoritması 64 bitlik anahtar uzunluğuna sahip olmasına rağmen 56 bit uzunluğunda simetrik kriptolama tekniği kullanan bir sistemdir. Her kullanımında o kullanıma özel yeni bir anahtar yaratması DES'in güçlü yanı olup, günümüz teknolojisi için algoritmasının yavaş ve 56-bit'lik anahtar uzunluğunun yetersiz kalması DES'in zayıf yönleridir. 2000'li yılların başında kırılmasıyla günümüz teknolojisi için yetersiz kaldığı görülmüştür ve itibarını kaybetmiştir. DES'in algoritmasından kaynaklanan bu sorunlar "Triple DES" ya da "DES-3" olarak bilinen yeni bir algoritma ile düzeltilmiştir. SSH gibi günümüzde kullanılan çoğu uygulama 3DES'i kullanmaktadır. 3DES algoritması DES şifrelemesinin 3 kere art arda yapılması şeklinde çalışır. Bu yüzden DES'e göre 3 kat daha yavaştır. Bununla birlikte 3DES şifreleme yapmak için uzunluğu 24 bayt olan bir anahtar kullanılır. Her bayt için 1 eşlik biti vardır. Dolayısıyla anahtarın uzunluğu 168 bittir [13].



Şekil 2.3: DES çalışma prensibi

Şekil 2.3'de DES çalışma prensibi verilmiştir.64 bitlik açık metin önce başlangıç permütasyonuna tabi tutulur. Ardından 32 bitlik iki parçaya ayrılır. Bu 32 bitlik parçaya bir fonksiyon yardımı ile alt anahtarlama işlemi uygulanır.56 bitlik anahtar sayesinde indirgenmiş bir şekilde 48 bitlik uygulanan bu anahtarlama işlemi 16 defa tekrarlanır.16 döngü sonucunda iki parça yer değiştirilir. Son olarak 64 bitlik bu metine başlangıç permütasyonunun tersi uygulanarak şifreleme işlemi gerçekleştirilir. DES protokolünün kabataslak hali bu şekildedir. Aslında anahtar 64 bittir fakat 8 biti parity için ayrılacağı için 56 bit anahtarlama için kullanılır. Her döngüye geçişte 48 bit anahtar üretilmesinin amacı her döngüde farklı bir anahtarın üretilmesinin sağlanmasıdır.

Zaman geçtikçe bilgisayarların gelişmesi ve daha da hızlanmasına paralel olarak DES önemini yitirmeye başlamış ve saldırılara açık hale gelmeye başlamıştır. Buna bağlı olarak 3-DES geliştirilmiştir.

3-DES normalde kullanılan DES'in 3 katı daha hızlıdır. Bu DES in 3 kez kullanılmasına denk gelmektedir. Hızlı olmasının mantığı 128 bit anahtar uzunluğunu kullanmasına dayanmaktadır. Triple DES olarak da adlandırılan bu standardın güvenlik seviyesi DES'e göre 2 kat daha fazladır.

DES algoritmasının en büyük zaafı 2^{56} olan anahtar uzayı genişliğidir. Bu gerçekten de güçlü bir şifreleme algoritması için oldukça küçük bir anahtar uzayı miktarıdır. 1970'lerde dahi bilinen açık metin saldırısının bu şifreleme algoritmasına karşı uygulanabileceği ve geniş anahtar arama saldırısı ile anahtarın elde edilebileceği fikri ortaya atılmıştır. Daha sonra DES algoritmasını kırabilecek çeşitli özel anahtar arama cihazları geliştirilmiştir. Bunlardan 1998 de RSA laboratuvarı tarafından geliştirilmiş olan "DES Challenge II-2" 56 saat içerisinde bir DES anahtarını başarı ile bulmuştur[14].

2.1.2 Gelişmiş Şifreleme Standardı(Advanced Encryption Standart-AES)

Günümüzde karşılaşılması muhtemel daha popüler ve yaygın olarak benimsenen simetrik şifreleme algoritması, Gelişmiş Şifreleme Standardıdır (AES). AES üçlü DES'den en az altı kat daha hızlı bulunmuştur. Anahtar boyutu çok küçük olduğu için DES için bir yedek gerekmekteydi. Artan bilgisayar gücü ile kapsamlı anahtar arama saldırısına karşı savunmasız kabul edilmiştir. Üçlü DES bu dezavantajın üstesinden gelmek için tasarlandı, ancak yine de yavaş bulunmuştur[15].

AES, Feistel şifresinden ziyade iteratiftir. 'Yerine koyma-permütasyon ağı' üzerine kurulmuştur. Bir kısmı, girdileri belirli çıktılarla değiştirme (değiştirme) ve diğerleri bitleri karıştırma (permütasyonlar) içeren bir dizi bağlı işlemden oluşmaktadır. İlginç bir şekilde, AES tüm hesaplamalarını gerçekleştirmektedir. DES gibi, AES simetrik bir blok şifrelemedir. Bu, hem şifreleme hem de şifre çözme için aynı anahtarı kullandığı anlamına gelmektedir. Bununla birlikte, AES, DES ile birçok açıdan oldukça farklıdır. Rijndael algoritması, DES ve kilit boyutlarının 64 ve 56 bitleri değil, çeşitli blok ve anahtar boyutlarına izin vermektedir. Blok ve anahtar aslında 128, 160, 192, 224, 256 bit arasından bağımsız olarak seçilebilir ve aynı olması gerekmez. Bununla birlikte, AES standardı, algoritmanın 128 bitlik bir blok boyutunu ve 128, 192, 256 bitlik üç tuştan birini seçebileceğini belirtmektedir. Hangi

sürümün kullanıldığına bağlı olarak, standardın adı sırasıyla AES-128, AES-192 veya AES-256 olarak değiştirilmektedir. Bir feistel yapıda veri bloğunun yarısının veri bloğunun diğer yarısını değiştirmek için kullanıldığını ve bunların yarılarının değiştirildiğini hatırlayın. Bu durumda tüm veri bloğu ikame ve permütasyon kullanarak her tur sırasında paralel olarak işlenmektedir[16].

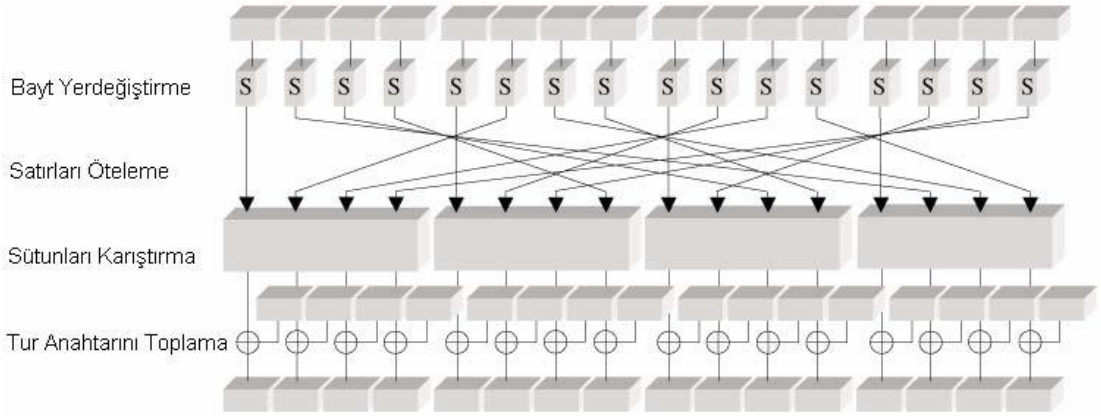
AES algoritması döngü anahtarı ekleme aşaması ile başlar, her biri dört aşamadan oluşan dokuz döngü ile devam eder ve üç aşamadan oluşan onuncu döngü ile son bulmaktadır. Bu uygulama şifreleme ve şifre çözme işleminde sadece şu yönden değişir; şifre çözme işleminde, döngünün her aşamasında şifreleme algoritmasının karşılığının tersi alınmalıdır. şifreleme algoritmasının ilk dokuz döngüsünün her biri aşağıdaki aşamaları kapsamaktadır[17]:

1. SubByte (Byte Yerdeğiştirme)

2. Shift Rows (Satırları Öteleme)

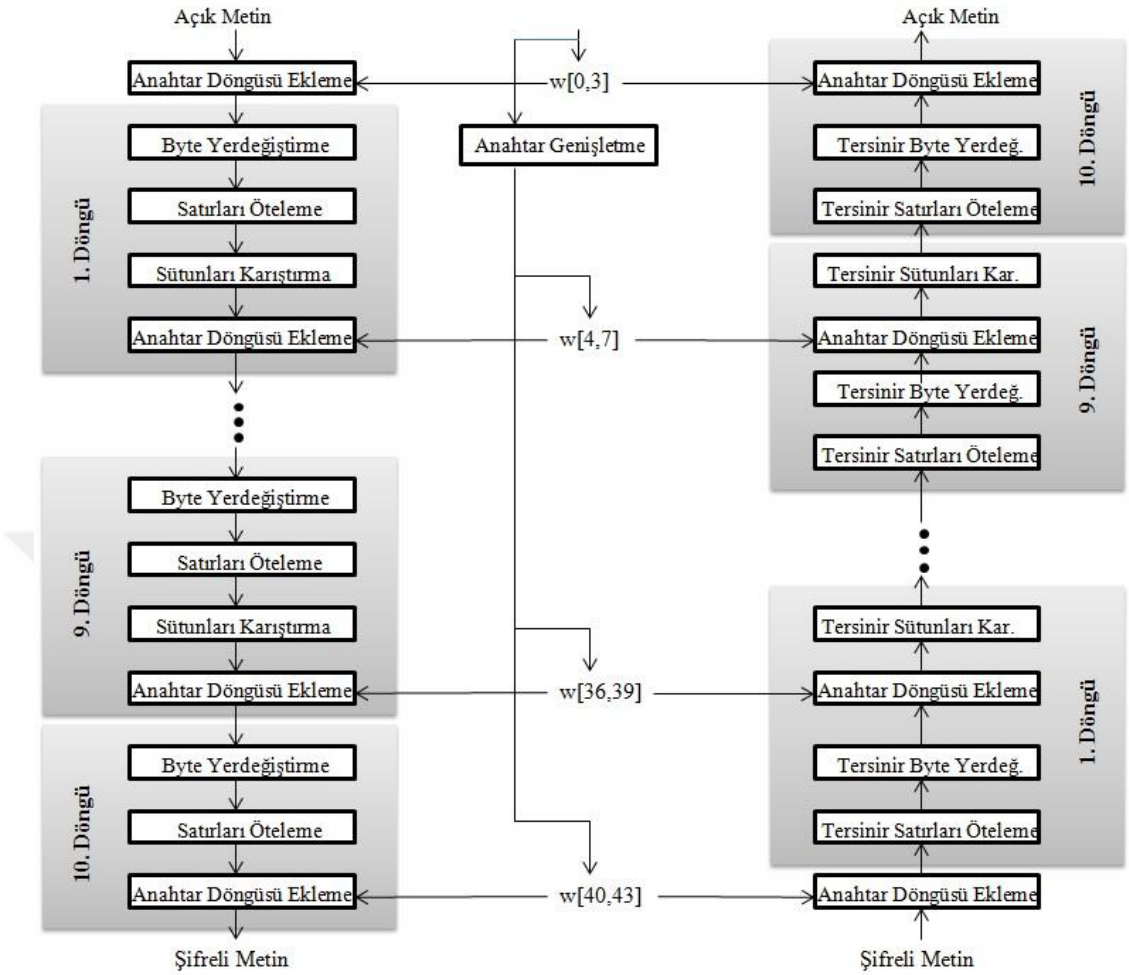
3. Mix Columns (Sütunları Karıştırma)

4. Add Round Key (Döngü Anahtarı Ekleme)



Şekil 2.4: AES tur dönüşümünün katmanları ve işleyişi [18].

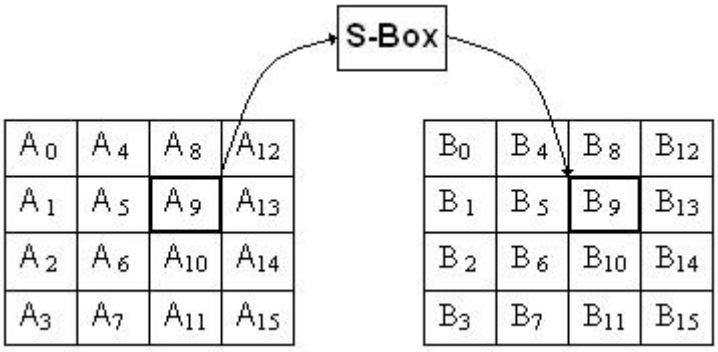
Şekil 2.4'te klasik bir AES'in tur dönüşümündeki katmanları ve işleyişi verilmiştir.



Şekil 2.5: AES algoritma yapısı [19].

2.1.2.1 Bayt Değiştir (SubBytes) Adımı

Bayt yer değiştirme, algoritma içerisindeki doğrusal olmayan tek dönüşüm olarak ele alınmaktadır. Bir s-box adı verilen 16x16 baytlık bir matrisini kullanarak bir tablo oluşturulur. Bu matris, 8 bitlik bir dizinin (2^8 =256) olası tüm birleşimlerinden oluşmaktadır. Burada s-box bir yer değiştirme çizelgesi olarak düşünülebilir. Bu çizelge ile her bir bayt başka bir bayta dönüşmektedir.

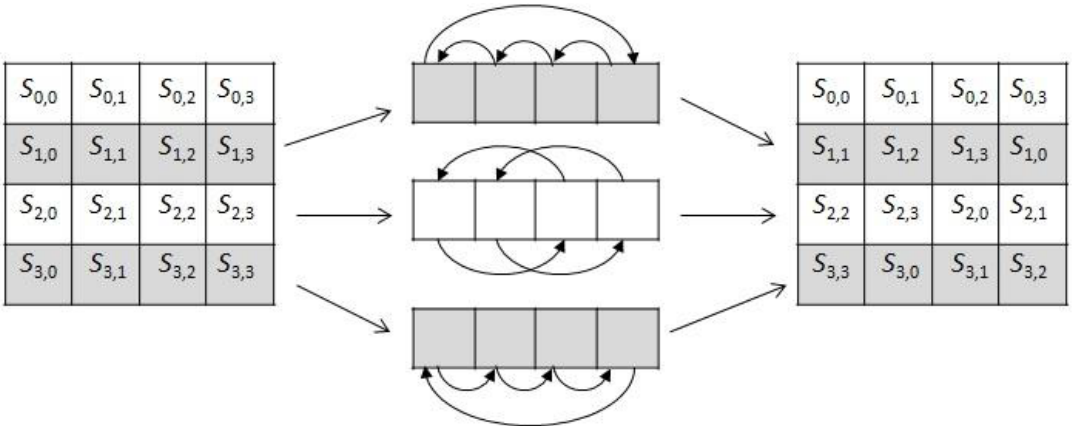


Şekil 2.6: Bayt değiştir katmanı [20].

2.1.2.2 Satırları Öteleme (ShiftRows) Adımı

AES şifresinin döngü fonksiyonunda yer alan diğer bir yapı ise satırları öteleme işlemidir. Bu işlem bir permütasyon işlemi olup matris üzerinde bir ayarlama ile gerçekleşir. Durum matrisin birinci satırında bir değişiklik meydana gelmez. İkinci satırda ki baytlar bir kez sol tarafa kaydırılır. Üçüncü satırdaki baytlar iki kez sol tarafa kaydırılır. Dördüncü satırdaki baytlar üç kez sol tarafa kaydırılır. Bu işlemin yapılma amacı durum matristeki baytların 4 sütuna dağıtılmasını sağlamaktır.

Şifre çözme işlemi için, ters satırları öteleme dönüşümü kullanılır. Ters dönüşüm de benzer bir yapıdadır. Sadece öteleme işlemi ters yönde, soldan sağa doğru, yapılır.[21]

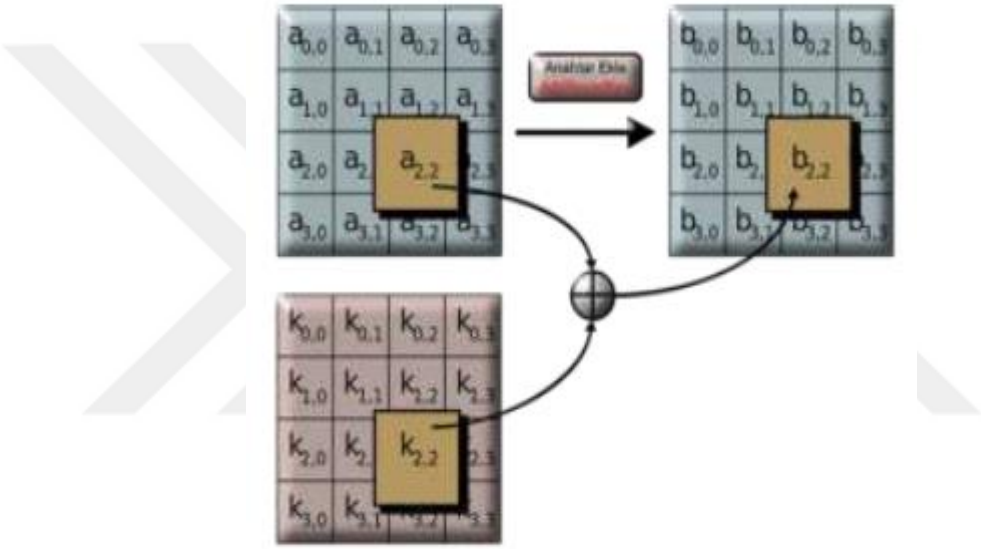


Şekil 2.7: Satırları öteleme yapısı [22].

2.1.2.3 Sütun Karıştır (MixColumn) Adımı

Bu aşama (MixColumn olarak bilinir) temelde bir ikame ancak GF aritmetiği (2^8) kullanır. Her sütun ayrı ayrı işletilmektedir. Bir sütunun her baytı, sütundaki dört baytın tümünün bir fonksiyonu olan yeni bir değere eşlenir. Dönüşüm, aşağıdaki durumdaki matris çarpımı ile belirlenebilir. Sütun Karıştır işlemi, her sütunun sabit bir matrisle çarpılması işleminden oluşmaktadır[23].

2.1.2.4 Anahtar Ekle (Add RoundKey) Adımı



Şekil 2.8: Anahtar ekleme yapısı [24].

Anahtar, anahtar döngüsünden türetilir ve onu yalnızca bir permütasyon olmaktan ziyade algoritmayı bir blok şifre (bir anahtarlı permütasyon) yapar. Yuvarlak tuş hiçbir turda eklenmediyse, blok şifre çıktısı anahtara bağlı olmaz ve anahtarsız bir permütasyon olurdu. Başlangıç ve bitiş yerine her turda yuvarlak anahtarı eklemek kadar güvenliği artıran bir tasarım seçeneğidir. XOR-encrypt-XOR, bir blok şifreyi kolaylıkla değiştirmek için kullanılabilir ve XOR-permute XOR'un temel ilkesi, bir blok şifre oluşturmak için güçlü bir sahte rasgele permütasyon ile kullanılabilir [25].

Whatsapp başta olmak üzere bazı büyük şirketler AES şifreleme sistemini kullanmaktadır.

Whatsapp uygulaması son yıllarda gönderilen mesajların, sağlanan iletişimin şifrelendiğini duyurmuştur. Burada amaç kullanıcıların güvenliğini sağlamasıdır. Whatsapp şifreleme konusunda AES256 şifreleme yöntemini temel almıştır, bunun yanında SHA gibi fonksiyonlar kullanılmıştır. Uçtan uca şifreleme amacıyla yola çıkılmıştır. Whatsapp ile iletişim sağlanırken alıcıya gönderilen mesaj, karşı taraf çevrimdışı olsa bile gönderilmektedir. Mesaj sunucularda şifreli olarak tutulmaktadır. Bu projede iki tarafın mesajlaşabilmesi için çevrimiçi olmaları şu anlık zorunludur.

Whatsapp depolama aracı olarak Microsoft'un teknoloji dünyasına kazandırdığı, büyük verilerin depolanmasına yardımcı olan blob depolama sistemini kullanmaktadır. Blob depolama sisteminde görüntüler veya belgeler doğrudan bir tarayıcıya aktarılmaktadır. Video ve ses akışları ile yedekleme, geri yükleme gibi işlemlerin de kolaylıkla yapılabilmesini sağlamaktadır. Gönderici ile alıcı arasındaki şifreleme güvenliği Curve25519, AES-GCM ve SHA256 protokolleri ile kontrol altına alınmaktadır. Curve25519, Diffie-Helman anahtar değişim sistemini kullanarak 128 bit mesajın güvenliğini sağlayabilen özel olarak tasarlanmış eliptik bir eğridir.

2.1.3 BLOWFİSH Algoritması

DES'in eksik kalmaya başlamasından sonra onun yerini alması amacıyla 1993 yılında Bruce Schneier tarafından tasarlanmıştır.64 bitlik bir blok boyutuna sahiptir. 32 bitten 448 bite kadar uzanan bir anahtarlama özelliğine sahiptir. Bu algoritma en az 4 kb'lik bir ram ihtiyacı duymaktadır. Bu yüzden akıllı kartlar gibi küçük yapılarda kullanılamaz.

Büyük bloklarda şifreleme olarak etkili bir algoritmadır. E-posta gibi sistemlerde kullanıldığında yüksek başarılı bir performans sergilemektedir. Blowfish algoritması çıktığı dönemde çok başarılı olmuş ve yaygın olarak kullanılmıştır. Bunun nedeni o dönemde kullanılan diğer şifreleme algoritmalarının lisanslı ve paralı olması, bunun aksine Blowfish'in lisanssız ve ücretsiz olmasından kaynaklanır.

Algoritma bir haberleşme sistemi veya otomatik dosya şifreleyici gibi, anahtarın sık sık değişmediği yapılar için daha uygundur.

Blowfish piyasada kullanılan en hızlı blok şifreleme sistemlerinden birisidir. İçerdiği karmaşık anahtarlama yapısı da çözülmesini zor kılmıştır.

2.2. Asimetrik Şifreleme Algoritmaları

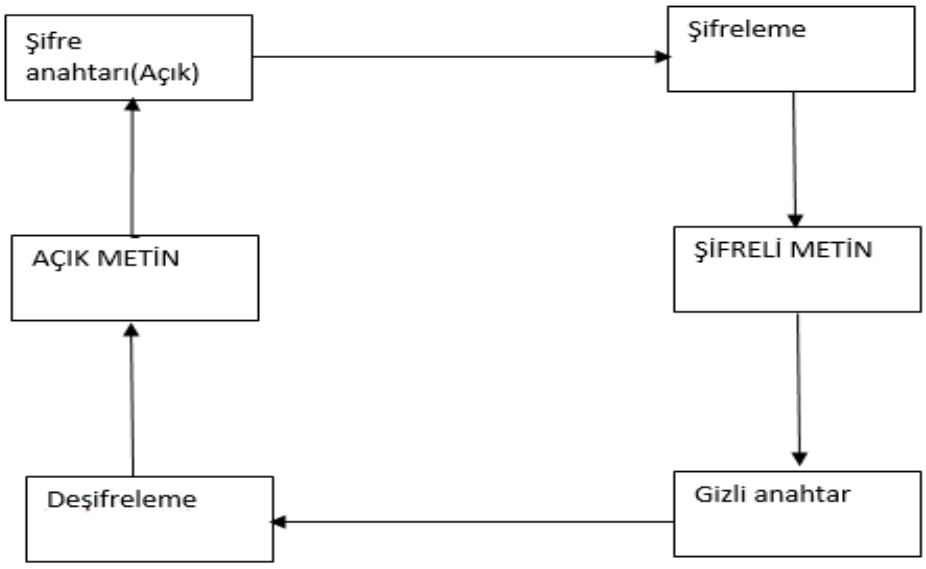
Simetrik şifreleme algoritmalarında açığa çıkan en büyük problem anahtarın dağıtımıdır. Birden fazla kullanıcılı bir sistemde aynı anahtar farklı kişilere dağıtılacağından güvenlik problemi açığa çıkar. Her kullanıcıya farklı anahtar temin edilmesi de sistemi oldukça zorlamaktadır.

Güvenlik problemlerini ortadan kaldırmak amacıyla asimetrik şifreleme algoritmaları geliştirilmiştir. Bu algoritmalarda şifreleyen anahtar ile şifreyi çözen anahtar birbirinden farklıdır. Şifreleme yapan anahtar açık anahtar olarak adlandırılır. Açık anahtar tüm kullanıcılara dağıtılabilir. Ancak hangi anahtarın hangi kullanıcıya ait olduğu bilinmelidir. Şifreyi çözen anahtara ise özel anahtar adı verilir. Özel anahtar sadece şifreyi çözecek olan tarafta bulunmalıdır. Bu sayede asimetrik şifreleme çalışmaları simetrik şifreleme çalışmalarına göre daha başarılı bulunmuştur. Az sayıda anahtar kullanan simetrik şifreleme algoritmaları anahtar fazlalığı durumunu engellediği için hız ve zaman konusunda asimetrik şifreleme çalışmaları biraz geri planda kalmaktadır. Asimetrik algoritmaların güvenliğini sağlamak için çok büyük asal sayıların kullanılması gerekir. Bu da çalışmalara zaman kaybı olarak geri döner. Bu sayıların kullanılmasından dolayı da donanımsal yapılara uyum sağlaması zorlaşmaktadır.

Asimetrik yaklaşımı ile sadece gizli anahtar gizli muhafaza edilmelidir ve bu gizliliği sadece bir tarafın saklaması gereklidir. Özellikle de taraflar birbirleriyle önceden hiç iletişim sağlamadıysa asimetrik yöntem kullanmak bir avantajdır. Özel anahtarın sadece bir taraf tarafından saklanması gerektiği için hiçbir zaman tehlikeye atılmış ağlar aracılığıyla iletilmesine gerek yoktur. Bu nedenle, birçok durumda asimetrik bir anahtar çifti birçok yapıda veya birkaç yılda değişim görmeden kullanılabilir.

Günümüzde simetrik ve asimetrik şifreleme yöntemlerini birlikte kullanarak hem yüksek derecede güvenlik hem de yüksek hızlı sistemler şifrelenebilmektedir. Bu gibi sistemlere karma şifreleme sistemleri adı verilir. Anahtar şifreleme, anahtar

anlaşma ve sayısal imza işlemleri genellikle asimetrik şifrelemeyle, yığın veri işlemleri ve imzasız veri bütünlüğü korumaysa simetriklerle gerçekleştirilir.



Şekil 2.9: Asimetrik Şifreleme Şeması

Asimetrik şifreleme algoritmaları günümüzde de hala güncel olarak kullanılmaktadır. Bu algoritmaların birçok çeşidi bulunmaktadır. Her birinin kullanım alanı farklıdır fakat kullanım amacı olarak aynıdır. Kullanım amacı olarak düşünüldüğünde her algoritmanın güvenliği sağlamayı amaç edindiğini görebiliriz. Biz tez çalışmasında en önemli iki asimetrik şifreleme algoritmasını ele alacağız. Bunlar:

- Diffie-Helman Anahtar Değişimi Sistemi
- RSA Şifreleme'dir

2.2.1 Diffie-Helman Anahtar Değişimi Sistemi

Asimetrik şifreleme algoritmalarını önemli kılan en büyük etmenlerden biri şifreleme için kullanılan anahtarın ve şifreyi çözmek için kullanılan anahtarın birbirinden farklı olmasıdır. Bu simetrik algoritmaların açtığı güvenlik açığını ciddi derecede kapatır.

1976 yılında Diffie ve Helman tarafından bir makale yayınlanmıştır. Yayınlanan bu makalede açık anahtarlı şifrelemenin temelleri atılmıştır. Geliştirilen yapıda sistem temelde bir anahtar değişim algoritmasıdır.

Şifreleme düzeninin amacı iki kullanıcı arasında gizli anahtarın nasıl güvenli bir şekilde iletilebileceğini açıklamaktır. Temeli bir matematik problemine ve asal sayıların kullanımına bağlıdır. Diffie-Helman anahtar değişimi sayesinde iki kullanıcı aslında güvenli bir şekilde ortak bir şifrede karar kılmaya çalışmaktadır. Bu algoritma aynı zamanda sadece gizli anahtarın ortak bir şekilde belirlenmesini sağlar. Fakat kullanıldığı zaman da simetrik algoritmaların anahtar dağılımında yaşadığı güvenlik zafiyetini çözer.

Sistemin çalışma mantığı şu şekildedir:

$$g^a \pmod{p} \equiv x \qquad (2.1)$$

Verilen (2.1) numaralı denklemde anahtar belirlenecektir. Burada a,p,x bilinse dahi g sayısını bulmak oldukça zor bir durumdur. Ancak deneme yanılma yöntemi ile çözülebilir. Burada x'in çözümü nispeten kolay olsa da ters işlem olarak x'den g sayısının bulunması oldukça zordur. Bu tek yönlü şifreleme olarak nitelendirilebilir.

İşlemler yapılırken sayıların asal olarak seçilmesi şifrelemenin gücünü ortaya koyabilir. İki büyük asal sayının çarpımı ve bu çarpımdan doğacak sonucun çarpanlara ayrılma işlemi bir bilgisayarın performansını oldukça zorlayacaktır.

2.2.2 RSA Şifreleme

Açık anahtarlama sisteminin asıl prensiplerini oluşturan şifreleme türüdür. Dünyada en yaygın kullanılan, en önemli şifreleme sistemidir.1977 yılında Ron Rivest,Adi Shamir ve Leonard Adleman bu şifreleme türünün kurallarını oluşturan yapıyı ortaya çıkarmışlardır. Şifrelemenin ismi de zaten bu üç mucidin baş harflerinden gelmektedir.

Sistem tam sayılarda çarpanlara ayırma problemine dayanmaktadır. Hem şifreleme için hem de dijital imza için kullanılabilir. Matematiksel altyapısı modüler aritmetik yapısına dayanmaktadır.

Şifreleme için kullanılacak sayılar iki büyük sayıdan oluşmalıdır ki algoritmanın güvenliği konusunda büyük sayı üretme problemi ortaya çıksın.

İlk olarak, rasgele bir kamu ve özel anahtar çifti oluşturulmaktadır. Kriptografide her zaman olduğu gibi, en rastgele ve dolayısıyla öngörülemeyen şekilde anahtarlar üretmek çok önemlidir. Ardından, veriler RSA algoritması kullanılarak genel anahtar ile şifrelenmektedir. Son olarak, şifreli verileri özel anahtar ve sonucu orijinal verilerle karşılaştırarak çalıştığı doğrulanmaktadır. Veriler genel anahtarla şifrelenmekte ve özel anahtarla şifresi çözülmektedir [26].

Ortak ve gizli anahtar seçimini oluşturmak için şu adımlar takip edilebilir:

- p ve q asal sayılar olsun. Bu asal sayılar en az 1024 bit ve büyük sayılardan seçilmelidir.
- n=p*q. (n burada açık modülü temsil etmektedir.)
- Z= (p-1)*(q-1).Anahtar üretimi için bir ara değer hesaplanır.Bu ara değer z olarak burada kabul edilir.
- Açık anahtar hesaplanır. Açık anahtarı e olarak kabul edersek e<z ve gcd(z,e)=1 hesabına göre birden fazla e değerlerinden biri seçilir.
- Ardından gizli anahtar hesaplamaları yapılır. Gizli anahtar olarak d'yi seçtiğimizi varsayalım. (d*e) mod z =1 işleminden gizli anahtar hesabı yapılır.

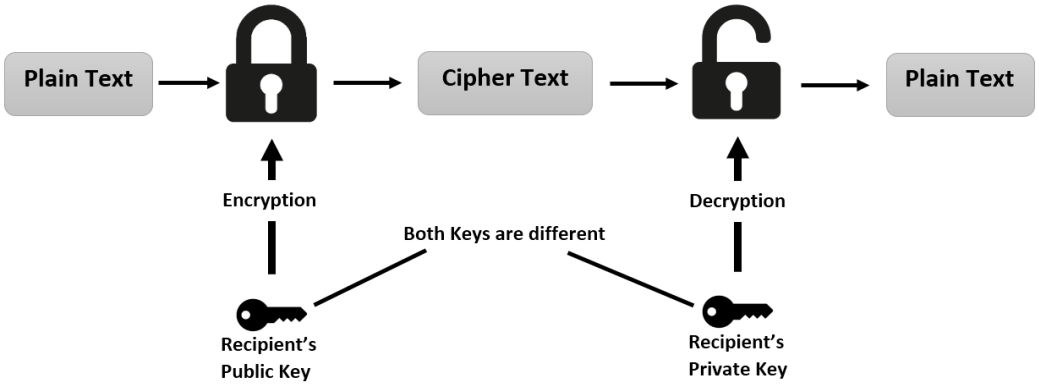
Basit bir anahtar elde etme denklemleri yukarıda belirtilmiştir. Ayrıca x'in açık metin, y'nin ise şifreli metin olduğunu kabul edersek,

$$y= x^e \bmod n \qquad (2.2)$$

$$x= y^d \bmod n \qquad (2.3)$$

(2.2) ve (2.3) denklemlerinden şifreleme ve şifre çözme denklemlerini sonuçlandırabiliriz.

RSA sistemlerinde güvenlik büyük p-q sayılarının oluşturulmasıyla sağlanır. Çarpanlarına ayırma işlemi uzun sürdüğünden ve zaman aldığından dolayı n sayısının da sonuca bağlı olarak çarpanlarına ayrılma süresi zaman almaktadır. Böylece RSA sistemlerinde çözüm işlemi zor ve zahmetli olacaktır. Asal sayılardan herhangi birinin küçük olması durumunda n sayısının çözülmesi de daha kolaylaşacaktır.



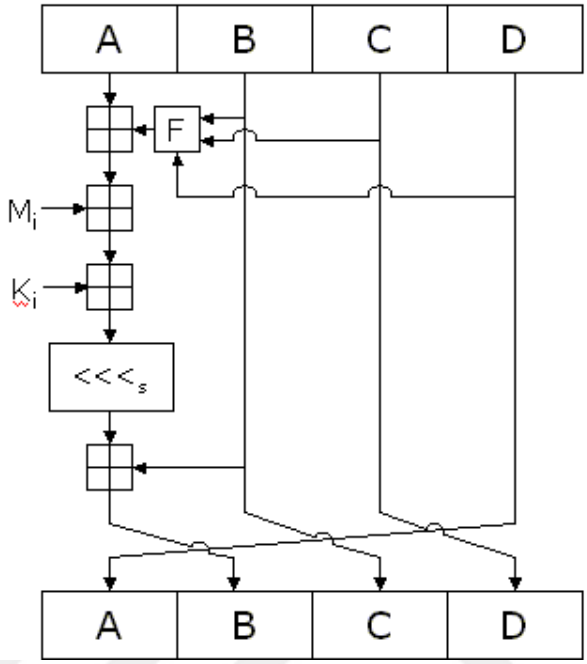
Şekil 2.10: RSA Çalışma Prensibi [27].

Simetrik ve asimetrik şifreleme algoritmalarının yanında herhangi bir anahtar kullanmadan şifreleme yapabilen algoritmalar da mevcuttur. Genel olarak yapı itibariyle tek başlarına kullanılmayan bu algoritmalar simetrik ve asimetrik yapılara yardımcı olabilmek için tasarlanmıştır. Şifre saklama işlemlerinde ve bütünlük yapıların oldukça sık kullanılan bu yapılar sayısal imza işlemlerinde yavaşlığa yol açmaktadır.

En çok kullanılan anahtarsız algoritma türü özet fonksiyonlarıdır. Uzun bir girdiyi alarak çıkışta daha kısa bir alandan o girdinin çıktı olarak üretilmesini sağlar. Verileri sınıflandırabilir ve verinin güvenliği açısından ilk hali ilke son halini karşılaştırarak farklı olanları tespit etmeye çalışır. En çok kullanılan türler MD5 ve SHA-1 algoritmalarıdır.

MD5 128 bitlik bir hash fonksiyonu olup yapısal olarak çok güvenli değildir fakat internetin yoğun kullanıldığı günümüzde değerini her zaman korumuş ve kullanılmaya devam etmektedir. Algoritmaya giriş yapılan boyutu fark etmeyen

herhangi bir verinin çıkışı olarak 128-bit uzunluğunda 32 karakterli 16'lık sayı sisteminde bir dizi elde edilir. Tek yönlü çalışma yapar. Yani şifrelemeyi yapar ama çözümünü üretmez.

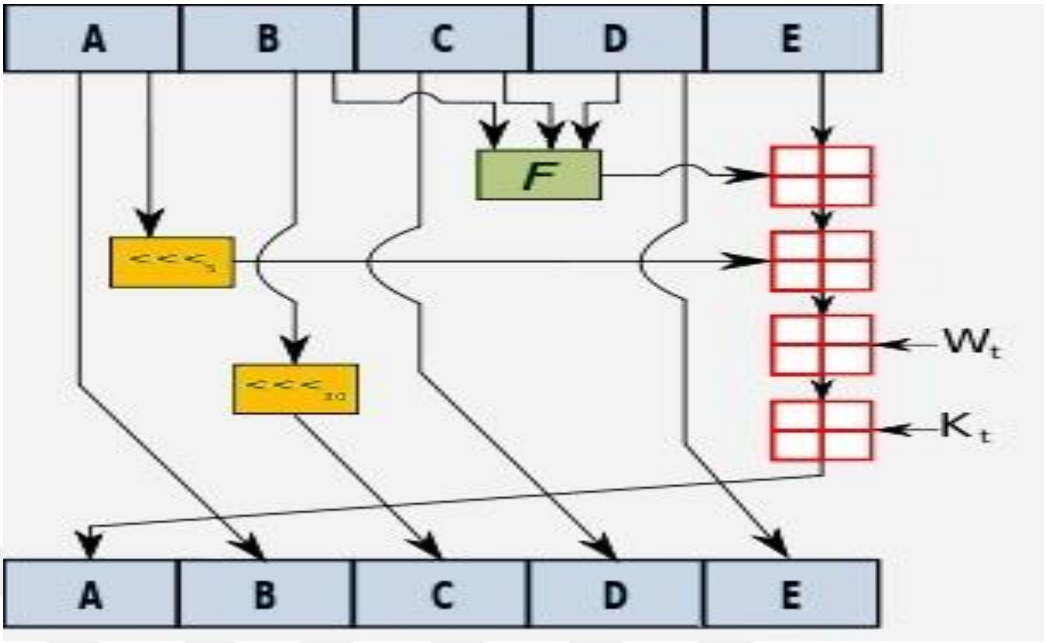


Şekil 2.11: MD5 Çalışma Prensibi

Bir diğer çok kullanılan özet fonksiyonu türü de SHA-1 yapısıdır. Aslında birden fazla SHA şifreleme türü vardır ama en fazla kullanılan SHA-1 türüdür. Hash fonksiyonlarına dayanan veri tabanlarına ulaşım imkânı sağlamaktadır.

Bu algoritma ile sadece şifreleme yapılmaktadır, şifrenin çözümünü üretmez. Algoritma ile 160 bitlik özet bloklar oluşturulmaktadır. Zaten MD5 ile arasındaki en büyük fark oluşturdukları özet boyutundadır.

SHA-1 algoritması e-posta şifreleme uygulamaları, güvenli uzaktan erişim uygulamaları, bilgisayar ağları gibi birçok alanda kullanılabilir. Güvenlik konusunda hala açığı bulunmaktadır. MD-5 ile birlikte kullanıldığında daha fazla güvenli olmaktadır.



Şekil 2.12: SHA Çalışma Prensibi

3. YAZILIM GELİŞTİRME ORTAMLARI

Bu bölümde tez çalışmasında kullanılan uygulamaların hazırlandığı ortamlar hakkında bilgi verilecektir. Uygulamanın oluşturulduğu bu alanların detaylı incelenmesi yapılacaktır.

Tez çalışmasında veri iletişimini sağlayacak olan yani haberleşme için kullanılacak uygulama React Native teknolojisi kullanılarak Java programlama dili ile yazılmıştır. Java programlama dili ile beraber uyum içinde çalışacak olan Javascript eklentilerine yer verilmiştir. Veri tabanı olarak PostgreSQL kullanılmıştır.

React Native 2015 yılında geliştirilmiş Javascript ile mobil uygulama geliştirme olanağı sunan bir sistemdir. Facebook tarafından geliştirilen bu yapı ile uygulama geliştirme hızı artmaktadır. Ayrıca uzaktan uygulama güncelleme fırsatı sunmaktadır. Native bir yapısı olduğu için kullanımı kolaydır. Bir kütüphane görevi de gören React yapısı ile çift yönlü ve asenkron bir şekilde haberleşme sağlanabilmektedir.

Tez çalışmasında sunulan uygulama içeriğinde oldukça sık kullanılan React yapısına, Java ortamında geliştirilen kod yapılarının mobil ara yüzü oluşturabilmek için ihtiyaç duyulmuştur. Çalışmada önce mesaj gönderme işlemi yapılmıştır. Mesaj iletimi sonrasında resim gönderme işleminde de React ortamından faydalanılmıştır.

PostgreSQL ise ücretsiz ve açık kaynak kodlu nesne yönelimli bir veri tabanı yönetim sistemidir. SQL dilinin güvenlik, depolanabilirlik ve ölçeklendirilebilme özelliklerinden faydalanan PostgreSQL, birçok alanda veri tabanı yöneticisi olarak da kullanılmaktadır. Veri yönetimine ve depolanmasına yardımcı olmaktadır. 1986 yılında ilk olarak piyasaya sürülen bu sistem hala gelişmeye devam etmektedir. Birçok programlama dili ile uyum içerisinde çalışabilmektedir. PostgreSQL'i dğer veri tabanları ile kıyaslandığında en güçlü kılan özelliği transaction, subselect, trigger, view, foreign key referential integrity ve sophisticated locking gibi (user-defined types), rules, inheritance ve lock cakışmalarını düşürmek için multi-version uyumluluk özelliklerine sahip olmasıdır. Performans açısından ise PostgreSQL diğer ticari veya açık kaynak kodlu veri tabanlarıyla kıyaslandığında başarılı metrik elde

etmektedir. Kimi veri tabanı sistemleri karşısında bazı açılardan hızlıyken bazı açıdan ise yavaştır.

3.1 Jetbrains Uygulaması

JetBrains (resmi adı IntelliJ) yazılım geliştirme ve proje yönetimi ile ilgili araçlar geliştiren bir yazılım şirketidir [28].

2015 yılı itibarıyla dünya genelindeki beş ofisinde 500'ün üzerinde çalışanı vardır. Ofisleri Prag, Sankt Peterburg, Moskova, Münih ve Boston şehirlerindedir [29].

Şirketin ilk uygulaması Intellij Renamer olarak kabul edilmektedir. Bu uygulamanın amacı java kodlarını yeniden düzenlemektir.

Şirket kullanıcılar için birçok programlama dilini IDE'si sunmuştur. Bunlardan bazıları Javascript için Webstorm,Java için IntelliJ Idea,PHP için PHPStorm,Ruby için RubyMine,Python için PyCharm,C ve C++ için CLion,veritabanı işlemleri için DataGrip vs.dir.2011 yılında ise kotlin adında bir programlama dili geliştirilmiş ve tanıtılmıştır.

2010 yılında Android geliştirme desteği başlamıştır. Google 2012 yılında EclipseADT'nin yanı sıra IntelliJ IDEA'ya da destek vermeye başlamış ve Android studio çıkarılmıştır.2015 yılında EclipseADT üzerinde destek çekilince Android konusunda Android studio programı daha işlevsel hale gelmiştir.

Halen çalışmalarına devam eden yazılım şirketi çeşitli anketler ile de yazılım dünyasına ışık tutmaktadır.Teknolojinin hızla ilerlemesiyle ortaya çıkan yeni ürünlere de çabuk adapte olması ile kullanıcılara büyük kolaylık sağlamaktadır.

3.2 Intellij IDEA

Java programlama dilinde bilgisayar yazılımı geliştirmek için 2016 yılında sunulan platformdur. Ticari ve kişisel kullanımları mevcuttur. Yeni teknolojinin ve güncellemelerin sunulması kullanımı açısından kullanıcıyı rahatlatmaktadır. İlk

kullanımı Ocak 2001'dir fakat bu kullanımda sadece Java kodlarının düzenlenmesi özelliğini içermektedir.

IntelliJ 2010'da InfoWorld'ün Eclipse, IntelliJ IDEA, NetBeans ve JDeveloper IDE'leri arasında yaptığı test sonucunda en yüksek sonucu almıştır[30].

3.3 Webstorm

Webstorm bir front-end geliştirme aracıdır. Javascript ile geliştirme imkanı sunan bu platform en iyi web editörlerinden biridir. Yeni standartlara uyumluluğu ile en fazla kullanılan platform haline gelmiştir.

Webstorm kod tamamlama özelliği ile yarıda bırakılan Javascript kodlarının tekrar girildiğinde kaldığı yerden devam etmesini sağlar. Tek bir lisans ile farklı işletim sistemlerinde çalışma imkânı sunmaktadır. Hata ayıklama özelliği en önemli özelliklerinden biridir. Kullandığımız programlama dillerine göre kod biçimlendirme yapmaktadır. En yeni teknolojilerle uyumlu yapıda çalışarak birleştirme gücü sağlamaktadır.

3.4 Sunucu

Bilgisayar ağlarında, diğer ağ bileşenlerinin (kullanıcıların) erişebileceği, kullanımına ve/veya paylaşımına açık kaynakları barındıran bilgisayar birimidir[31].

Shared Hosting (paylaşımlı barındırma), Co-Location, Reseller ve Dedicated (atanmış) Hosting olmak üzere 3 çeşidi bulunmaktadır. Shared Hosting bir sunucuda birden fazla alan bulundurma özelliği taşır. Co-Location, kendi sunucunuzu özel hazırlanmış veri merkezlerinde yüksek hızda hizmet bağlantısı ile çalıştırılmasını ve sunulmasını sağlamaktadır. Dedicated Hosting'de, sunucu donanımı ve bağlantısı dâhil tüm hizmetlerin sunucuyu kullanan kullanıcı veya şirket tarafından karşılanmaktadır. Reseller Hosting de belli sayıda alan, disk ve bant genişliği sınırları ile kullanıcıya sunulur.

Sunucu yapısı kullanımlarına göre farklılık göstermektedir. Örneğin web sayfalarını ve içeriklerini saklayan bir web sunucusu vardır. Sunucular kullanacakları alana göre yazılımlarla da desteklenebilir. Bazen yeri gelmekte bilgisayarlar sunucu olarak da kullanılmaktadır. Bunlara sunucu bilgisayar denilmekte olup bir bilgisayar ait donanımsal parçaların oluşturduğu bir yapı olduğunu söylemek gerekir.

Tez çalışmasında kullanılacak sunucu için firewall kullanılmaktadır. Firewall kullanmanın amacı haberleşme esnasında oluşacak verilerin güvenliğini sağlamaktır. Olası ddos saldırıları için güvenlik duvarı sunucular için önemli yer tutmaktadır.

Firewall gelişmiş filtreleme seçenekleri ile alınan ve gönderilen veri trafiğinde kontrolü sağlayarak veri geçişine izin verir veya verinin geçişini engellemektedir. Genelde iki tür güvenlik duvarı kullanılmaktadır. Bunlardan birincisi veri akışını engellemektedir. Diğer tür ise belirlenen protokollere göre veri akışını sınırlamakta ve bunu düzenlemektedir.

Bir sunucu ile ağ arasındaki router'lara yerleştirilen güvenlik duvarlarına donanımsal firewall adı verilmektedir. Bu sistem saldırı önleme yöntemi, anti virüs özelliği ile ağda hızı ve güvenliği ön plana çıkarmaktadır. Birçok protokolde(FTP,http gibi) önemli rol oynamaktadır. Bunun haricinde yazılımsal güvenlik duvarları da mevcuttur. Yazılımsal güvenlik duvarları, tek tek sunuculara yüklenmektedir ve her bir bağlantı isteğini keserek talebin geçerli olup olmadığını belirlemektedir. Performans açısından zayıf kalmaktadır. Fakat kullanış kolaylığı ve güvenlik göz önüne alındığında önemli yer tutmaktadır.

Çalışmada kullanılan veri tabanının PostgreSql olmasının bazı sebepleri vardır. Bu sebepler şu şekilde listelenebilir.

- SQL'in bütün özelliklerini desteklemektedir.

- Master-Standby replikasyonunu destekler ve standby sunucular için hemen hemen gerçek zamanlı replikasyon ve hot standby yetenekleri ile sonuçlanan yüksek hızlı WAL işlemi üreten kayda değer geliştirmeler getirmiştir.

- PostgreSQL, okuma ve yazma hızlarının çok önemli olduğu ve verilerin doğrulanması gereken büyük sistemlerde yaygın olarak kullanılmaktadır. Ayrıca, yalnızca coğrafi veri desteği, okuma kilidi olmayan eşzamanlılık ve benzeri (örneğin Oracle, SQL Server) gibi ticari çözümlerde kullanılabilen çeşitli performans optimizasyonlarını desteklemektedir.

- PostgreSQL performansı, karmaşık sorguların yürütülmesini gerektiren sistemlerde en iyi şekilde kullanılmaktadır.

- PostgreSQL, izinleri ayarlamak ve sürdürmek için devralınmış rollere sahiptir. PostgreSQL, istemci / sunucu iletişimini şifrelemek için bağlantılar için yerel SSL desteğine sahiptir. Ayrıca Satır Seviyesi Güvenliği de vardır.

Buna ek olarak, PostgreSQL SELinux güvenlik politikasına dayalı ek erişim kontrolleri sağlayan SE-PostgreSQL olarak adlandırılan dâhili bir geliştirme ile birlikte gelir.

Çalışmada kullanılacak olan algoritma yapısında Base64 tekniği de yer almaktadır. Base64 tekniği veri sıkıştırma ve güvenlik işlemlerinde sık kullanılan yapılardan biridir. Bu teknik veri güvenliği için kodlama algoritması olarak değerlendirilebilir. Gönderilmek istenen mesajın farklı semboller kullanılarak gösterilmesidir. Bu semboller alfabenin büyük, küçük harflerinden ve sayılardan oluşmaktadır. Genel olarak gönderilen mesaj text ise bu yöntem fazla kullanılmaktadır.

Bu teknik ile veri kaybı engellenebilir. Örneğin 8 bitlik bir veriyi 64 bite çevirerek farklı katmanlar oluşturmaktadır. Farklı katmanlar ile veri iletişimi sağlanarak veri kaybı önlenmektedir. Bu açıdan Base64 tekniği oldukça yaygın kullanılan şifreleme yapılarından birisidir ve bu çalışmada da RSA algoritması içinde yer alacaktır.

Veri tabanı için program olarak Data Grip kullanılmıştır. Data Grip, Jetbrains firmasının kullanıcılarına sunduğu veri tabanı yönetim sistemlerini kontrol etmek için tasarlanmıştır.

4. DENEYSEL ÇALIŞMA

Bu bölümde bilgi güvenliğini pekiştirmek amacıyla tezimizin amacında da belirtildiği üzere şifreleme metotları ile beraber gizlilik ilkesini temel alarak geliştirilen bir uygulama incelenecektir.

Bilgi güvenliğinin temelinde daha önce de belirtildiği üzere 4 temel olgu yatıyordu. Bunlardan biri de gizlilik esası idi. Gizlilik esasının kullanıldığı en önemli güvenlik önlemlerinden biri şifreleme algoritmalarıdır.

Daha önceki çalışmalarda bilgi güvenliği adına önemli çalışmalar yapılmıştır. Mobil platformların hızla yükselmesinden dolayı bilgi güvenliği kavramı da mobil ortamlarda çok daha önemli hal almaktadır.

Mobil ortamlarda güvenliği sağlamak için birçok adım bulunmaktadır. Bu zamana kadar yapılan çalışmalarda verilerin güvenliğini esas alan makale ve incelemeler bulunmaktadır [32-34]. Bu çalışmalar çeşitli güvenlik önlemlerini ön plana çıkarmaktadır. Kötücül yazılımlar, anti virüs yapıları, cep telefonu için kullanılan güvenlik uygulamaları bu çalışmalara konu olmuştur. Veri iletişimi sağlanırken şifreli mesaj kullanımı ön plana çıkarılmıştır.

Bu çalışmada verilerin güvenli bir şekilde iletilmesi için kullanıcının oluşturduğu bir metot ile RSA algoritması ile yapılan bir metodun birbirleri ile karşılaştırılması, iletişim halinde verilerin şifrelenmesi, güvenli iletişimi sağlamak için ortaya sunulan metotların avantajları ve dezavantajları değerlendirilecektir. Şu ana kadar şifreleme algoritmalarının bilhassa mobil platformlar için yapılan çalışmaları benzer bir şekilde bulunmaktadır. Yapılan bir incelemede RSA algoritması kullanılarak SMS ile güvenli iletişimi sağlamayı amaçlayan çalışma bulunmaktadır[35].Çalışmada Android işletim sistemine sahip cihazlarda SMS kanalı kullanılarak haberleşme amaçlanmıştır. Çalışmada yapılan uygulamada başarı elde edilmiştir. Açık anahtar her bir mesaj gönderilecek kişi için üretilmiştir. Açık anahtarlar oluşturulduktan sonra iletişim sağlanmaya başlamıştır. Bu çalışmada dezavantaj SMS gönderimi sırasında mesajın merkez yapılarından ve yönlendirme aşamasından geçmesi ve buna

bağlı olarak güvenlik konusunda açık oluşmasına olanak sağlamasıdır. Yapılan başka bir çalışmada ise asimetrik şifreleme algoritmalarının birbirleri ile karşılaştırılması yapılmış ve bu çalışmada RSA ve ECDH algoritmaları kullanılmıştır[36].Yine bir diğer çalışmada RSA algoritmasının mobil cihazlardaki optimizasyonu incelenmiştir [37]. Bu çalışmada RSA algoritması kullanılarak mesaj şifreleme işlemi yapılmaktadır. Bu olay gerçekleşirken bir form üzerinde, bir de emülatör kullanılarak mesaj şifreleme işlemleri yapılmaktadır. Belli sayıda basamak ve bit kullanılarak şifreleme süreleri karşılaştırılmaktadır. Bu mesajlaşma esnasında küçük ve büyük bitler şifrelenmiştir. Uygulamada C# programlama dili kullanılmış ve emülatör kullanıldıktan sonra uygulama gerçek tablet üzerinde denenmiştir. Bu çalışmada big integer sınıfı kullanılmıştır.

Bu tarz çalışmalar sonucu asimetrik şifreleme ve RSA algoritmanın güvenliği her defasında hız ve güvenlik açısından uygun olduğu da anlaşılmaktadır.

Çalışmada Android tabanlı bir cihaz üzerinden iletişim sağlanacaktır. Bir uygulamanın içine iki metot gömülecek ve kullanıcı hangi metodu kullanmak istiyorsa o metot üzerinden haberleşmesi sağlanacaktır. Bu iki metot da Java ortamında geliştirilmiş ve Android uyarlaması da aynı şekilde yapılmıştır. Gönderilecek verinin alıcı ile verici arasında sorunsuz gidebilmesi yani tam bir iletişimin sağlanması gerçekleşecektir. Bu iletişim sırasında mesajı gönderecek olan kullanıcının her mesajı kullanılan metot ile şifrelenecek ve veri tabanında şifreli bir şekilde tutulacaktır. Mesajın iletildiği kullanıcı ise veri tabanından çekilecek olan kendisine gelen şifreli mesajı çözülmüş olarak görecektir.

Kullanıcıların iletişimi sırasında birbirlerine ilettikleri her mesaj uygulamada kullanılan bir sunucu tarafında depolanacaktır. Veri tabanında tutulan her şifreli mesajın gönderim süresi hesaplanmaktadır. Bu sürenin kullanılma amacı başta da belirtildiği üzere hangi metodun daha hızlı ve güvenli bir metot olduğuna karar vermektir. Avantajlarını ve dezavantajlarını belirlemektir.

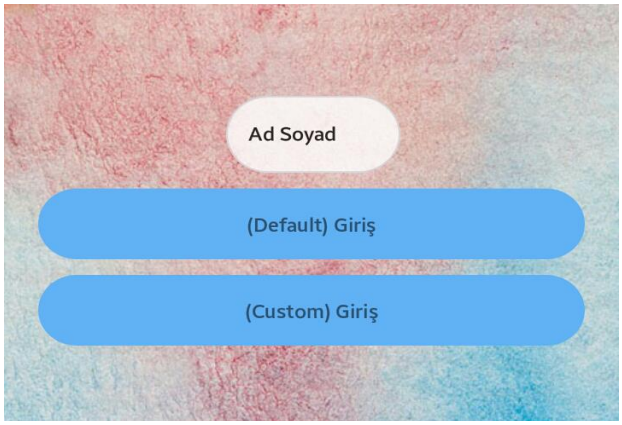
Çalışmada daha önce veri tabanı işlemlerinde PostgreSQL kullanılacağı anlatılmıştır, bunun nedenleri de birlikte verilmiştir. Veri tabanında uygulama için 9 adet alan oluşturulmuştur. Bu alanlar “id”, “date”, “file_name”, “file_type”, “message”, “message2”, “name”, “pic”, “type” olmak üzere oluşturulmuştur. İletişim halinde sunulacak olan mesaj, türü ne olursa olsun şifrelenmiş bir şekilde veri tabanında gizli olarak tutulacaktır. Burada oluşturulan “date” alanı ile mesajın gönderildiği tarih ve süre tutulacaktır. “message” alanında ise mesajın olacağı aşikârdır. ”name” alanına bakıldığı takdirde giriş yapılmadan önce belirlenen isim tutulmaktadır. “type” alanında ise gelecek sayfalarda anlatılacak olan 2 farklı şifreleme türünden hangisinin seçildiğini gösteren sayı tutulmaktadır. “file_type” alanında da mesaj türünün gizlendiği kısım yer almaktadır. Mesaj türü eğer resim olacaksa images/jpeg şeklinde bir örnek veri söz konusu olacaktır. Gerekli alanların kodlama aşamaları bir sonraki şifrelemelerin belirtildiği sayfalarda açıklanmaktadır.

4.1. Geliştirilen Uygulamanın Kullanım Arayüzleri

Bu kısımda uygulamanın arayüzü tanıtılacaktır. Uygulama giriş kısmında kişinin bilgilerini isteyen bir yapı bulunmaktadır. Ardından iki metot birden seçenek olarak verilmiş kullanıcı hangi metot ile giriş yaparsa o metot ile iletişimini sağlayabilecektir.

Uygulama geliştirilebilirlik açısından ucu açık tutulmuştur. Şu seviyede hedeflenen nokta farklı olduğu için tasarım bu şekilde uygun görülmüştür. İlerleyen zamanlarda yapılacak çalışmalarda gelişmeye açık bir şekilde uygundur.

Giriş kısmında belirtile ad, soyad bilgileri ve seçilen şifreleme türü bilgisi veri tabanında saklanmaktadır. Seçilen şifreleme türünün ardından mesaj gönderme ekranı görülecek ve ya text olarak ya images olarak iletişim sağlanabilecektir

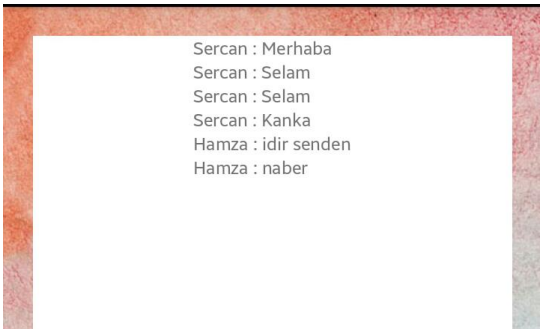


Şekil 4.1. Uygulama Giriş Bölümü

Yukarıda da belirtildiği gibi Şekil 4.1'de verilen görüntü uygulamanın giriş kısmını temsil etmektedir. Ad, soyad bilgileri girildikten sonra kullanıcıdan hangi metodu kullanacağı beklenmektedir.

"Default" olarak belirlenen kısım RSA algoritması ile mesajların şifrelendiği ve iletişimin sağlanacağını belirten giriş bölümüdür.

"Custom" olarak belirlenen giriş kısmında ise belirlediğimiz bir metot ile mesaj şifrelenecek ve kullanıcıların iletişimi sağlanacaktır. İki giriş kısmınında içeriğindeki görsel tasarım aynı şekildedir.



Şekil 4.2. Default Giriş Bölümü

Yukarıdaki Şekil 4.2'de "Default" olarak tanımlanan RSA algoritması ile şifrelenen iletim hattının tasarımı verilmiştir. Görüldüğü üzere mesajlar şifreli olarak değil normal bir şekilde ekrana yansımaktadır.

"Custom" yani özel olarak hazırlanan şifreleme metodu kullanılarak yapılan güvenli iletim hattı verilmiştir. İki giriş bölümünün de tasarımı aynı şekildedir.

İletim ortamına bakıldığı üzere iki kullanıcının birbiri ile haberleşmesinde iki farklı girişlerin seçilebilmesi mümkündür. Bu uygulamada haberleşme ortamının kullanılabilmesi için bu mümkün kılınmıştır. İki farklı ortamı da seçerek kullanıcılar birbiri ile haberleşebilmektedir. Asıl amaç haberleşme esnasında gönderilecek olan mesajın güvenliği ve şifrelenmesidir. Şifreli mesaj ile haberleşme sağlandıktan sonra karşılaştırma işlemi yapılacaktır. Hangi metodun ön plana çıkacağı belli olacaktır. Bu seçeneğin ortaya çıkmasında farklı kriterler söz konusudur. Bu kriterlerden en önemlisi güvenlik ve hız durumudur.

Koşullara göz atıldığında mesajlaşma içerisinde mesajların güvenliğinden veri tabanı ,sunucu gibi kullanılan yöntemlerin tamamının senkron bir şekilde çalışması gerekmektedir. Yani veri tabanının da sunucunun da güvenli olması gerekmektedir ki tam manasıyla bir iletim ortamı oluşmalıdır.

4.2. Yöntemler

4.2.1 RSA Algoritması kullanılarak oluşturulan metot

Uygulamada "Default" giriş bölümünün altında yatan kısmı RSA algoritması ile oluşturulan şifreleme metodu oluşturmaktadır. RSA algoritması daha önce de belirtildiği gibi asal sayıların çarpımı kullanılarak oluşturulan bir şifreleme problemidir. Birçok alanda hala kullanılmaya devam edilmektedir.

Bir uygulama hazırlanırken güvenli bir iletim hattının oluşturulması birden fazla platform için uygundur. Mobil-mobil, mobil-bilgisayar ve ya bilgisayar-mobil şeklinde üç adet verici-alıcı ortamı oluşturulabilir. Bu uygulamada kullanılacak olan esas mobil-mobil iletişimi sağlamaktır.

RSA algoritması esas alınarak “128,256,512” bit ve daha fazla veri şifrelenmesi sağlanabilir. Teknolojinin ve bilgisayar bilimlerinin hızla yükselmesi, veri güvenliğinin sürekli ön planda olması bu alanda da gelişimin sağlanmasına aracı olmuştur. Uygulamada kullanacağımız olan iletim için 512 bit esas alınmıştır. Bu seçimdeki sebep günümüzde artık çok boyutlu verilerin şifrelenebilir olmasıdır. Daha önce yapılan çalışmalarda SMS gönderimi için 512 bitlik bir durum 0-64 karakter mesaja izin verilmiştir [38].

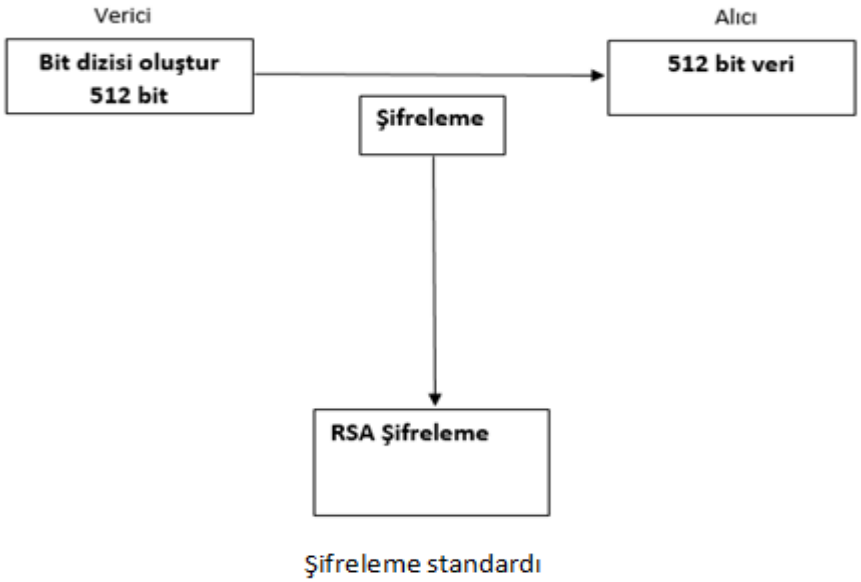
Uygulama Java programlama dili ile geliştirilmiştir. Şifreleme yapılırken Java geliştirme araçlarının kullanıcılarına hazır olarak sunduğu kriptolama kütüphaneleri kullanılmıştır. Şifreleme yapılırken yine kripto kütüphanelerine bağlı olan fonksiyonlar kullanılarak veriler şifrelenmiş ve çözülme sağlanmıştır. Bu kütüphaneler özetleme fonksiyonlarından blok şifreleme algoritmalarına, açık anahtarlama yapısından dijital imza sertifika yapısına kadar bütün şifreleme işlemlerinde hazır taslak olarak kullanıcılarına kolaylık sağlar.

Kullanıcıdan girilen karakterleri dizi şeklinde alan uygulama önce gerekli fonksiyon ile bu mesajı şifrelemektedir. Gerekli frameworkler ve servisler sağlandıktan sonra veri tabanına gelen mesajı şifreli olarak kaydetmektedir. Şifreleme süresi uygulama ekranında gözükmemektedir. Veri tabanına kayıt işlemi tamamlandıktan sonra mesajın şifreleme süresi veri tabanından işlem olarak alınmaktadır.

Uygulama yapım aşamasında React Native teknolojisi kullanılmıştır. React native teknolojisi Facebook tarafından geliştirilen IOS ve Android uygulamalar oluşturmak için kullanıcılara kolaylık sağlayan Javascripti kendine temel alan açık kaynaklı bir platformdur [39] .

Bu teknoloji ile Java ile geliştirilen yazılım kolaylıkla Android platforma dönüştürülebilmektedir.

RSA algoritma yapısının kullanılabilmesi için anahtar uzunluğunun en az 512 bit olması gerekmektedir. Uygulamada bu dikkate alınmıştır ve anahtar boyutu 512 bit belirlenmiştir.



Şekil 4.3. RSA şifreleme şeması

Şekil 4.3'te verilen yapıda RSA şifreleme yapısı şemalandırılmıştır. Güvenli iletim hattının oluşturulabilmesi adına çalışma 512 bite kadar desteklenmektedir. Görüldüğü üzere mesaj iletilirken şifrelenir, alıcı tarafından görüldüğü zaman şifre çözülür.

Bilindiği üzere bilgisayar sistemleri ikili sayı sitemlerinin oluşturduğu yapıdır. Bu sistemde her bir karakter bir baytı temsil etmektedir. Bir bayt ise 8 bitten oluşmaktadır. Buna bağlı olarak tasarlanan yapımızda 512 bite kadar mesaj içeriği kabul edileceğinden dolayı 64 bayt veriye kadar şifreleme yapılmaktadır.

RSA yöntemi ile şifreleme yapılırken Base64 metodu hazır olarak kullanılmıştır. Java kütüphanesinde kullanıma hazır olarak bulunan Base64 yapısı şifrelemeyi bir standart haline getirir. Bu standarda göre kaç karakter şifrelenirse şifreli metin de o kadar karakter sayısına sahip olacaktır.

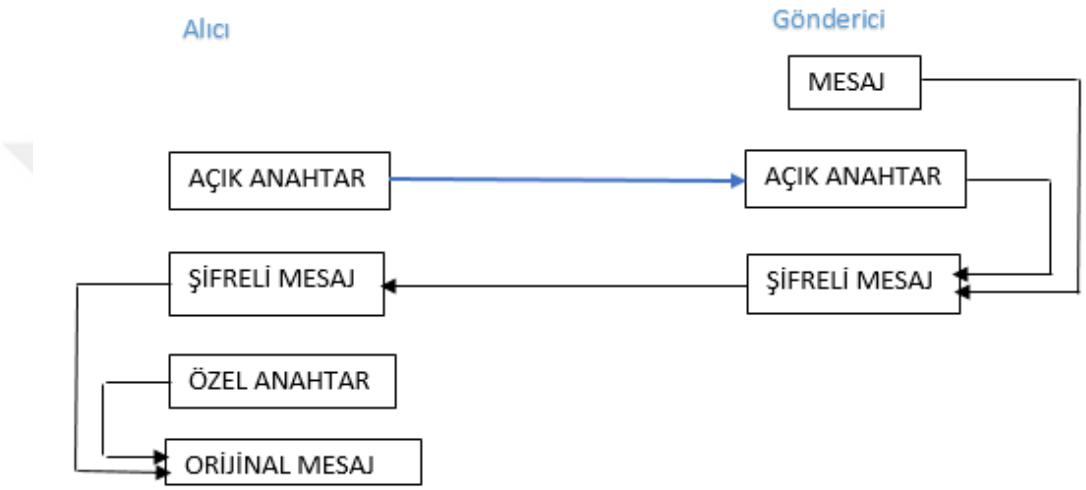
Base64 şifreleme ve şifre çözme tekniklerinde oldukça önemli rol oynamaktadır. Base64 şifreleme tekniği basit bir kodlama yapısına dayanmaktadır. Temel olarak farklı semboller halinde verinin gösterilmesi işlemidir. 8 bitlik verileri 64 bit veriye çevirmektedir.

```
import org.apache.tomcat.util.codec.binary.Base64;

import java.security.*;

import javax.crypto.Cipher;                                    (4.1)
```

4.1'de verilen kod parçacığı temel olarak şifreleme ve Base64 kütüphanesinin yapıya eklenmesini sağlar. Bunun yanında birçok web servisi de kod parçacığındaki yerini almıştır.



Şekil 4.4. RSA algoritma yapısı

Şekil 4,4'te RSA algoritma yapısı verilmiştir. Bu algoritmaya göre anahtarlar oluşturularak ve kodlama aşamasından geçirilerek işlem gerçekleşmeye devam eder. RSA algoritması çoklu iletim ortamlarında kullanıcılara güvenli bir ortam sunmaktadır.

```
public static KeyPair buildKeyPair() throws NoSuchAlgorithmException {
    final int keySize = 512;
    KeyPairGenerator keyPairGenerator = KeyPairGenerator.getInstance("RSA");
    keyPairGenerator.initialize(keySize);
    return keyPairGenerator.genKeyPair();                              }
```

(4.2)

4.2'de verilen kod parçacığında Şekil 4.4'te belirtilen anahtar yapısı oluşturulmuştur. Anahtar oluşturulurken dikkat edilirse anahtar boyutu 512 seçilmiştir. Anahtar değeri

seçildikten sonra metodun RSA olduğu belirtilmiş ve işlem yapılması için gerekli fonksiyona gönderilmiştir.

Bu işlemden sonra şifreleme ve deşifreleme işlemleri başlar.Base64 yapısı ile ortak yürütülen şifreleme durumu ile şifreli mesaj üretilir

```
public static byte[] encrypt(PrivateKey privateKey, String message) throws Exception {
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.ENCRYPT_MODE, privateKey);
    return cipher.doFinal(Base64.encodeBase64(message.getBytes()));
}
```
(4.3)

```
public static byte[] decrypt(PublicKey publicKey, byte[] encrypted) throws Exception {
    Cipher cipher = Cipher.getInstance("RSA");
    cipher.init(Cipher.DECRYPT_MODE, publicKey);
    return cipher.doFinal(encrypted);}
```
(4.4)

4.3'te verilen kod parçacığında RSA şifreleme yöntemine göre şifreleme yapılması beklenmektedir.Base64 yapısı ile beraber kullanılarak verilerin belli bir ortamda iletilmesine ve saklanmasına izin verilmesi sağlanmıştır.4.3'te verilen kod parçacığı sayesinde özel anahtar oluşturma işlemleri yapılmaktadır. Bu özel anahtar şifreleme işleminde alıcının mesajı deşifre edebilmesine yaramaktadır. Alıcıdan başka herhangi bir kişi anahtara sahip olamayacağı için şifreyi çözememektedir.

4.4'te verilen kod parçasında ise şifreleme işleminin tam tersi olan mesajın çözülmesi ile ilgili yapı bulunmaktadır. Bu kod paçasında verilen kodlara göre iletim ortamdaki tüm kullanıcıların kullanabileceği anahtarlama işlemi yapılmaktadır. Burada bulunan fonksiyonlar Java'nın sunduğu hazır kripto kütüphanelerinin de içinde yer alan fonksiyonlardır.

```
componentDidMount(): void {
    fetch('http://134.209.86.189:8080/decryptMessage')
      .then((response) => response.json())
      .then((responseJson) => {
        this.setState({data: responseJson});
        this.setState({isLoading: false});
      });}
```
(4.5)

4.5 kısmında verilen kod parçacığı ise Şekil 4.4'te verilen yapıda çözülmüş metnin karşılığına denk gelmektedir. Bu kısımda sunucu ile iletişim halinde olan yapı veri tabanına gelen mesajı çözmek ile görevlendirilmektedir.

Gerekli işlemler yapıldıktan sonra sistem sonucunu verecektir. İletim hattı oluşturulacaktır ve gönderici ile alıcı arasında rahatlıkla mesajlaşma işlemi başlayacaktır.

Uygulamada 4 farklı test yapılmıştır. Bu testler farklı boyutlarda oluşturulan mesajların karşı tarafa iletilmesinde geçen süreyi ele almaktadır. Bu süreler karşılaştırılacak ve belirlenen unsurlar içinde değerlendirmeler yapılacaktır. Test sürecinden önce Tablo 1.1'de uzunluk ve karakter eşleşmesi verilmiştir. Bu tablodaki verilere göre karakterlerin uzunluğunun ne kadar byte veriye denk geleceği gösterilmiştir.

Tablo 4.1. Gönderilebilecek mesaj boyutları

Şifrelenecek veri boyutu	Oluşturabilecek karakter sayısı
16 byte	0-16 karakter
32 byte	0-32 karakter
64 byte	0-64 karakter
128 byte	0-128 karakter

Tablo 4.1'de gönderilebilecek mesaj boyutları verilmiştir. Sadece bu kadar mesaj iletimi değil gerektiği takdirde 128 byte mesaj gönderimi de sağlanabilir. Uygulama çerçevesinde 4 boyut kullanılması uygun ve yeterli görülmüştür.

Tablo 4.2. Farklı boyutlardaki verilerin şifreleme süreleri

ŞİFRELENECEK VERİ BOYUTU	**ŞİFRELEME SÜRESİ**
16 BYTE	0,005786 sn
32 BYTE	0,013549 sn
64 BYTE	0,082537 sn
128 BYTE	0,405932 sn

RSA şifreleme algoritması kullanılarak yapılan bu şifreleme işlemlerinde ortaya çıkan süre sonuçları bu şekildedir. Bakıldığı zaman anahtar boyutunun artması ile birlikte şifreleme süreleri de hemen hemen aynı orantıda artmıştır. Şifreleme sürelerinin artmasının sebebi mesajı oluşturan karakter sayısının artması ile birlikte hesaplanacak olan matematiksel işlemlerin artmasından kaynaklanmaktadır.

Matematiksel işlemlerin artması ile birlikte işlem maliyetinin de uzaması süre konusunda dezavantaj getirmiştir. Tabi ki dezavantajı olduğu gibi avantajlı kısımlar da bulunmaktadır. Anahtar boyutunun artması ile beraber gönderilebilecek mesaj boyutunun da artması kullanıcıya kolaylık sağlamaktadır.

Anahtar boyutunun artması güvenlik seviyesinin de arttığının işaretçisidir. Bunun sebebi boyut arttıkça şifreleme esnasında yapılacak olan matematiksel işlemlerin fazlalaşmasıyla ilgilidir.

Bakıldığı zaman matematiksel işlemlerin artması hem fayda hem zarar getirebilmektedir. Güvenlik açısından incelememiz gerektiği için kullanıcının her zaman bu işlem işine yarayacaktır.

4.2.2 Özel olarak oluşturulan kişiye özel şifreleme metodu

RSA şifreleme algoritması kullanılarak yapılan uygulamaya karşılık hazırda kullanılan şifreleme algoritmalarının yerine özel oluşturulan bir metot kullanılacaktır. Bu metot oluşturulurken tamamen bağımsız şekilde anahtar oluşturulmuş ve kod kısmı bağımsız olarak özgün bir şekilde yazılmıştır.

Bu şifreleme metodunda kullanılan ana yöntem şifrelemede kullanılacak karakterlerin yer değiştirmesi ile açıklanabilir. İlkel yöntemlerden biri olan yer değiştirme algoritmasına benzer olan bu yöntemde anahtar boyutu daha önceden belirlenmiştir.

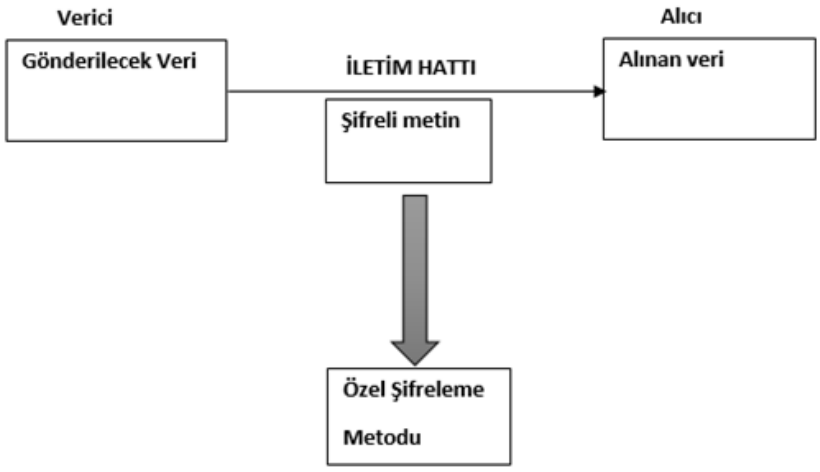
Yapı basit olarak sezar şifreleme türüne benzemektedir. Sezar şifrelemede amaç bir alfabede bulunan karakterlerin her birisinin yerine aynı alfabeden farklı bir karakter koyarak şifreleme yapmaktır. Buna göre bir tablo oluşturularak her karaktere karşılık gelen alternatif karakter tabloda tutulur. Mesajı şifrelemek isteyen kişi bu tablo yardımıyla her karakteri teker teker karşılık geldiği karaktere çevirir. Şifrelenmiş mesajı açmak isteyen kişi ise aynı tabloda tersi işlemi uygular [40]. Sezar şifresi bilinen ve en basit şifrelerden biridir. Düz metindeki her bir harfin alfabenin belirli bir yerinde belirli bir yere 'kaydırıldığı' bir ikame şifresidir [41].

Sezar şifresi, yer değiştirme ve harf değiştirme sistemleriyle çalışan bir algoritmadır. Alfabe belli sayıda ötelenmesi (yerinin değiştirilmesi) ve açık metindeki harfler ile ötelenmiş harflerin değiştirilmesiyle şifreleme işlemi gerçekleşir. Şifreleme işleminden tekrar açık metne ulaşmak için (deşifre) bu ötelemenin tersi uygulanır.[42]

Metot özel olarak oluşturulduğu için ve sadece kaydırma olarak karakterlerin yerleri değişeceği için güvenlik zafiyeti oluşacaktır. Şifreleme daha hızlı gözükmektedir. Çünkü hiçbir matematiksel işlem gerektirmez ve şifreleme esnasında zorlukla karşılaşmayacaktır.

İki liste halinde önce karakterler dizi olarak belirlenmiştir. Ardından anahtar boyutuna göre şifreleme işlemi tamamlandıktan sonra hangi karakterin hangi karakterin yerini alacağı belirlenecektir. Şifreleme işlemi tamamlandıktan sonra alıcıya iletilmesi esnasında yine aynı yöntemin tersi ile çözülmüş metin alıcıya gözükecektir.

Şifreleme sadece dizi halinde verilen karakterlerin birbiri ile etkileşim içinde olacağı için güvenli değildir. Çünkü el yordamı ile dahi bir süre sonra bu mesaj çözülebilir. Eğer 28 karakter dizi içerisinde belirtiliyorsa 28 denemede bir harfin yeri bulunabilir. Bu en kötü ihtimalle 28 deneme olarak düşünülmelidir.



Şekil 4.5. Özel şifreleme metot şeması

Yukarıdaki şekilde şifreleme süreci gösterilmiştir. İletim ortamında hem şifreleme işlemi hem de çözme işlemi başarılı bir şekilde gerçekleşecektir. Bu sayede iletişim kesintisiz olarak sağlanmış olacaktır.

Hazırlanan bu şifreleme metodunda herhangi bir kütüphane kullanılmamıştır. Buna bağlı olarak işlemlerde kolaylık sağlayacak fonksiyonlar da kullanılmamıştır. Sade bir yapıya sahiptir. Dizi halinde tanımlanan alfabe elemanlarına göre klasik programlama dili kodları sayesinde bir sistem tasarlanmaya çalışılmıştır.

```
public String encryptMessage2(@RequestParam("name") String name,
@RequestParam("message") String message) throws Exception {
    String signed = "";
     int a = 3;       int b = 5;       int c = 0;
     Character[] alfabe = {'a', 'b', 'c', 'ç', 'd', 'e', 'f', 'g', 'ğ', 'h', 'ı', 'i', 'j', 'k', 'l', 'm', 'n', 'o',
'ö', 'p', 'r', 's', 'ş', 't', 'u', 'ü', 'v', 'y', 'z'};
     Character[] alfabe2 = {'d', 'e', 'f', 'g', 'ğ', 'h', 'ı', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'ö', 'p', 'r',
's', 'ş', 't', 'u', 'ü', 'v', 'y', 'z','a','b', 'c', 'ç'};
     message = message.toLowerCase();
     for(int i = 0;i<message.length();i++) {          c = -1;
        for(int j = 0;j<alfabe.length;j++) {
           if(alfabe[j] == message.charAt(i)){
              c = j;} }
        if(c == -1) {
       signed += message.charAt(i);          }
        else          {
           signed += alfabe2[c];          }       }
     kripto k = new kripto(name, 2, signed);
     vtkripto.save(k);                                                    (4.6)
```

4.6’da verilen kod parçacığında bir alfabe karakter türünde belirlenmiş ve liste şeklinde tanımlanmıştır. Bu listedeki elemanların uzunluğu hesaplanarak bir döngü içerisine sokulmuştur. Verilen anahtarın uzunluğuna göre harfler kendi arasında yer değiştirerek veri tabanına başarılı bir şekilde kayıt işlemi tamamlanmaktadır.

```
List<kriptoReturn> returnList = new ArrayList<>();
 List<kripto> listk = vtkripto.findAllByOrderByIdDesc();
      String result = "";
      for (kripto k : listk) {
         if (k.getType() == 1) {
            byte[] bytes = decrypt(pubKey, k.getMessage());
            returnList.add(newkriptoReturn(k.getName(),new
String(Base64.decodeBase64(bytes))));} else {
            result = "";
            for(int i = 0;i<k.getMessage2().length();i++) {    c = -1;
               for(int j = 0;j<alfabe2.length;j++) {
                  if(alfabe2[j] == k.getMessage2().charAt(i)) { c = j; }}
               if(c == -1) {
               result += k.getMessage2().charAt(i); } else{ result += alfabe[c]; } }
              returnList.add(new kriptoReturn(k.getName(), result)); }        }
return returnList; }                                                                    (4.7)
```

4.7’de verilen kod parçacığında daha önce şifrelenmiş metin çekilmektedir. Çekilen bu metin bir liste haline getirilmektedir. Yine Base64 yapısı kullanarak ikili sisteme ve karakterlere çevrilmekte olan mesaj çözüldükten sonra yine bir liste halinde tutularak kullanıcıya sunulmaktadır.

Uygulamada RSA şifrelemede olduğu gibi yine aynı boyutlarda çalıştırılıp kontrol edilmesi sağlanacaktır. Şifrelemede anahtar oluşturma süreleri tutulacak olup karşılaştırılma işlemine geçilecektir.

Tablo 4.3. Farklı boyutlardaki verilerin özel metot ile şifrelenme süreleri

ŞİFRELENECEK VERİ BOYUTU	ŞİFRELEME SÜRESİ
16 BYTE	0,004593 sn
32 BYTE	0,010439 sn
64 BYTE	0,079635 sn
128 BYTE	0,399638 sn

Şifreleme sürelerine bakıldığı zaman oldukça hızlı bir metot olduğu görülmektedir. Algoritma açısından hızlıdır. Donanımsal yapılara bu sayede daha kolay entegre edilebilir.

Bilgi güvenliği açısından bir ilke olarak gizlilik kavramına uygun bir yapıda bulunmaktadır. Anahtarın boyutu kullanıcı tarafından seçildiği için ufak bir avantaj sağlamaktadır.

Bunun yanında daha önce belirtildiği üzere güvenli bir iletişim hattı oluşturması zordur. Bundan dolayı kimlik doğrulama ve bütünlük kavramlarını tam olarak yerine getirememektedir.

Kaba kuvvet saldırısı [43] ile rahatlıkla çözülme işlemi sağlanabilir. Bu sebeple güvenlik açığı uygulamada ne derece önemli olduğunu göstermektedir. Ne kadar hızlı olursa olsun bilgi güvenliğini sağlayan etmenlerin tümünün dikkate alınması gerekmektedir. Böylece bütünlük sağlanabilir ve güvenli iletişim yolunun önü açılmış olur.

Deneysel çalışmada yer alan iki metot ile mesajın şifreleme süreleri, bir başka deyişle anahtar oluşturma süreleri nanosaniye cinsinden ekrana yansıtılmaktadır. Ayrıca sadece karakter değil, aynı zamanda emoji gönderimi de sağlanmaktadır. Bu

emoji gönderimi esnasında ortaya çıkan şekil karaktere çevrilerek veri tabanında şifrelenmesi sağlanmaktadır.

```
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

byte[] signed = encrypt(privateKey, message);

kripto k = new kripto(name, 1, signed,format.format(new Date()));

k.setFileType(file.getContentType());

k.setPic(file.getBytes());

vtkripto.save(k);

return "kayıt edildi";                                                    (4.8)
```

4.8'de verilen kod parçacığına göre bir tarih formatı oluşturulmaktadır. Tarih ve süre kısmının nasıl tutulacağı "format" nesnesinde oluşturulmuştur. Yine kripto ile oluşturulan "k" nesnesi ile tarih ve süre, ikisi birlikte kaydedilecektir.

```
startTime = System.nanoTime();

...

endTime   = System.nanoTime();

returnList.add(newkriptoReturn(k.getName(),result,k.getDate(),

Long.toString(endTime - startTime)));                                     (4.9)
```

4.9’da verilen kod parçacığında yazılan komutlardan bir parça verilmiştir. Bu parçaya göre mesajın gönderileceği ve gittiği süre hesaplanmaktadır. Burada bu sürenin hesaplanmasındaki amaç şifrelenecek verinin ne kadar sürede şifrelendiğini ön plana çıkarmaktır.

Her mesajın kendine özgü şifreleme süresi vardır. Bu şifreleme süresi nano saniye olarak tutulmakta olup aynı zamanda veri tabanında şifreli olarak da saklanmaktadır.

```
@GetMapping("/image/{id}")

   public ResponseEntity<byte[]> encryptMessage2(@PathVariable("id") String id) throws Exception {

      kripto k = vtkripto.findByIdEquals(Integer.parseInt(id));

      HttpHeaders headers = new HttpHeaders();

      byte[] media = k.getPic();

      headers.setCacheControl(CacheControl.noCache().getHeaderValue());

      headers.setContentType(MediaType.IMAGE_JPEG);

      ResponseEntity<byte[]> responseEntity = new ResponseEntity<>(media, headers, HttpStatus.OK);

      return responseEntity;                                                    (4.10)
```

Web servisleri kodlama aşamasında önemli bir yer tutmaktadır. 4.10 numaralı kod bloğunda verilen GetMapping yapısı ile uygulamanın diğer aşamalarında öne çıkan RequestMapping yapısı web servisler üzerinde işlemlerin gerçekleşmesini sağlayan kolay yapılardır. Bu yapılar sayesinde servislere ulaşılmaktadır

Kod parçacığında verilen yapılara bakıldığında HttpHeader metotu web servislerinin tarayıcılarla iletişimini sağlamak, içeriği tarayıcılara göndermek için kullanılmaktadır. Security Header olarak da adlandırılan bu yapı ile bazı saldırılardan korunma işlemi başarı ile gerçekleşmektedir.

4.10'da verilen blokta ortaya çıkan diğer detaylardan biri de ön bellekleme yapılarıdır. Burada ön bellekleme için "setCacheControl" yapıları kullanılmaktadır. Bu yapı ile gönderilmek istenen mesaj Header yapılarında ön bellekte tutulmaktadır. Blok parçası kontrol edildiğinde "MediaType" metodu ile hangi türün ön belleklendiği gösterilmektedir. Resim türü bu yapıda kullanılmaktadır. Resim gönderimi esnasında bu parçada tarayıcı üzerinden iletişim sağlanmaktadır. Bunun mobil kısma dökülmesi React yapısı ile gerçekleşmektedir.

```
@Entity

public class kripto {    @Id

   @GeneratedValue(strategy=GenerationType.AUTO)

   private int id;

   private String name;

   private int type;

   private byte[] message;

   private String message2;

   private String date;                                    (4.11)
```

4.11'de verilen yapıya bakıldığında bir sınıf ortamı görülmektedir. Bu sınıfın ayrı bir önemi vardır. Veri tabanında kullanılan elemanlar ile sınıf yapılarının aynı olduğu görülmektedir. Bu sınıf yapısı ile mesaj, mesaj türü, tarih gibi elemanların tanımlanması görülmektedir. Burada sınıf içerisinde Entity yapısı kullanılmaktadır. Entity bir framework çeşididir. Nesne yönelimli programlamada sınıflandırma amacı ile kullanılır. Ayrıca bu framework yapısı ile anlık olarak veri tabanı ile haberleşme sağlanabilmektedir. En önemli avantajı bu özellik olarak görülmektedir. Çünkü veri tabanı ile haberleşme, bir programın yapım aşamasında güncel olarak sürekli çalışacağından programcı ile uygulama arasında hızı artırmaktadır.

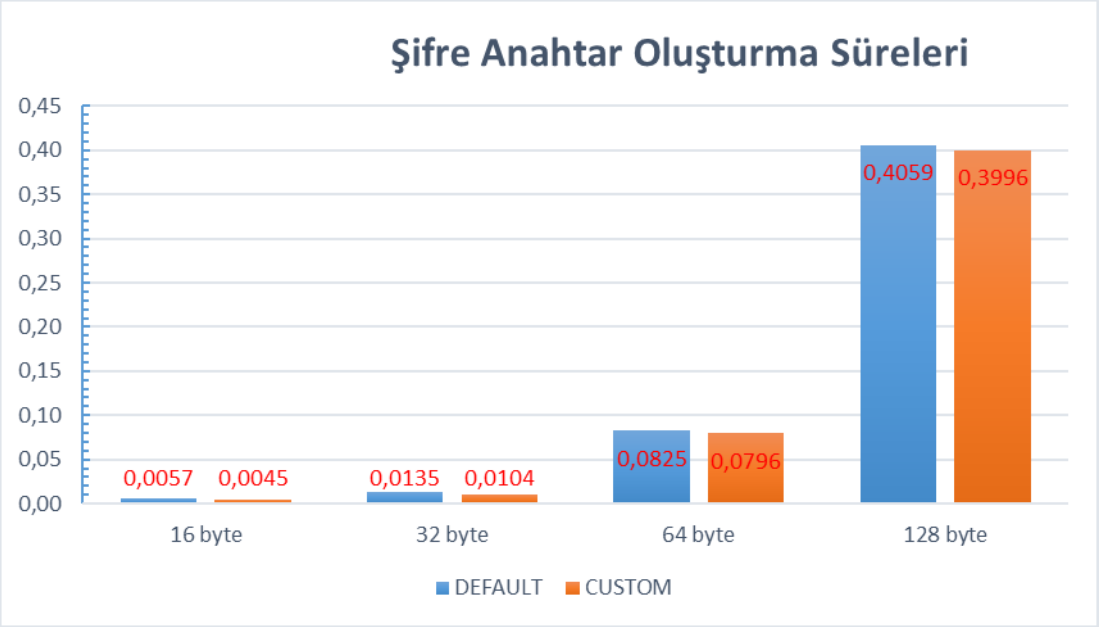
Şekil 4.6. Farklı boyutlar ile mesajlaşma örneği

Şekil 4.6'da verilen uygulama içi örnek mesajlaşma resminde metin mesajı, emoji ve resim gönderme formları bir arada gösterilmiştir. Aynı zamanda verilen tarihe ait iletişim esnasında gönderilen mesajların şifreleme süreleri de nanosaniye cinsinden ekrana yansıtılmıştır.

5. SONUÇ

Teknolojinin geldiği son noktaya bakıldığında veri iletişiminin, haberleşmenin son derece önemli olduğunu söylemek gerekiyor. Bu haberleşme aracılığıyla verinin miktarının hızla arttığı bir dönemde iletişim sağlanırken bazı faktörlere de artık dikkat etmek gerekiyor. Şüphesiz bu faktörlerden en önemlisi bilginin güvenliği ve verinin saklanmasıdır. Gerek bilgisayar kısmında gerekse mobil kısımda veri akış trafiği o kadar çok hızlanmıştır ki güvenlik konusu da bu denli önemli hale gelmiştir. Son yıllarda özellikle mobil platformlarda gelişen teknolojiye bağlı olarak güvenlik konusu bu ortamlarda daha çok önem taşımaktadır. Bugün insanlar mobil cihazlardan alışveriş yapabilmekte, banka işlemlerini halletmekte ve iletişimini farklı uygulamalar ile bu ortamlarda yapmaktadır. Bu işlemler yapılırken ortaya çıkan güvenlik sorunu detaylı bir şekilde ele alınmalıdır. Tabi bu güvenlik ortamının sağlanabilmesi için en önemli faktörlerden birisi de şifrelemedir. Tarih boyunca iletişim ortamlarında güvenlik ortamının sağlanması için türlü şifreleme metotları ortaya çıkarılmıştır. Bunlardan bazıları çok işe yararken bazıları yeteri kadar fayda sağlamamıştır. Günümüze doğru geldikçe veri ortamı teknolojiye bağlı olarak müthiş bir artış göstermiştir ve buna istinaden bu şifreleme türleri zaman zaman yetersiz kalabilmektedir. Bunun için her geçen gün şifreleme üzerine çalışmalar devam etmeli ve yeni yöntemler geliştirilmelidir. Bir platform oluşturulurken hangi şifreleme türünün kullanılacağına özen ve dikkat edilmelidir.

Biz de uygulamamızda güvenli bir iletim ortamı tasarlamaya çalıştık ve güvenliği sağlamak için şifreleme yöntemlerini kullanmaya karar verdik. Uygulamada farklı boyutlar ile veri göndermeye çalıştık. Şifreleme türü olarak RSA algoritmasını test ederek bu algoritmanın hızını, güvenilirliğini ele aldık. Aynı zamanda bir başka metot geliştirerek yine gönderilen verinin şifrelemesini sağladık ve bu şifreleme performansını RSA algoritmasından çıkan sonuç ile karşılaştırdık.



Grafik 5.1 Şifreleme Sürelerinin Karşılaştırılması

Yukarıda verilen grafikte uygulanan yöntem sonucu ortaya çıkan verilerin karşılaştırılması verilmiştir. Şifre anahtar oluşturma süreleri verilen bu grafik iki yöntem hakkında çıkarımlarda bulunmak için oluşturulmuştur.

Yapılan uygulamada elde edilen sonuçlara göre kullanıcı tarafından oluşturulan özel metoda göre verilerin şifrelendiği mesajlaşma alanı oldukça hızlıdır. Bu hızda basit bir yapısı olmasının büyük katkısı vardır. Kodlama aşamasındaki temel yapılar teknik olarak daha hızlı olmasına aracı olmuştur. Çünkü RSA algoritması işlem gerektirdiği için zaman kaybına yol açabilir. Fakat hızlı olmasına rağmen özel oluşturulan metodun dezavantajları RSA algoritması ile yapılan şifrelemeye göre oldukça fazladır. Bilgi güvenliği konusunda önemi belirtilen kavramlar göz önüne alındığında önemli kuralları ihlal etmektedir. İşlem sadece gizlilik kuralını yerine getirmektedir. Bundan dolayı mobil bir ortamda iki taraf arasındaki iletişimde bu metot önerilmemektedir.

Bir başka tarafta güvenlik unsuruna bakıldığı zaman matematiksel işlemlerin yoğun olması ve karmaşık bir şifreleme yapısı içerdiğinden RSA algoritması ile yapılan şifreleme ile iletişim sağlanabilir.

Güvenlik açısından özel olarak hazırlanan şifreleme metodu kullanılması önerilmemektedir. Bunun sebebi şifrenin üçüncü şahıslar tarafından görülmesinin olasılığının çok yüksek olmasıdır. Yüksek olmasının altında yatan sebep şifre kırıcı konusunda gelişmişlik seviyesine göre oluşturulan metodun daha basit yapı olmasından kaynaklanır.

Görüldüğü gibi şifreleme algoritmalarının kullanılma durumları amacına göre farklılık göstermektedir. Fakat ne olursa olsun öncelik güvenliğin sağlanmasıdır. Şifreleme tekniklerine bakıldığında hepsinin güçlü veya zayıf yönleri mevcuttur. Tezimizi mobil ortamlar üzerinden değerlendireceğimiz için mobil platformlarda sağlanacak iletişimde ortamın getireceği kavramlar üzerinde durmak gerekir.

Günümüzde mobil cihazların bilgisayarlar seviyesine geldiği göz önünde bulundurulursa RSA algoritma üzerine yapılan çalışma kullanıcının işine daha fazla yarayacaktır.

6. TARTIŞMA VE ÖNERİLER

Çalışma genel olarak değerlendirildiği takdirde mobil cihazlarda güvenli bir iletişimin sağlanabilmesi ve bilginin korunması için başvurulan yöntem şifrelemedir. Deneysel çalışma sonucu başarılı bir şekilde haberleşme sistemi kurulmuş, verinin gizliliği sağlanmıştır. Güvenlik, hız ve zaman konusunda değerlendirmelerde bulunulmuştur. Literatür çalışmalarına bakıldığı zaman tez çalışmasının içinde de belirtildiği üzere RSA algoritması ile çalışmalar mevcuttur. Bunlardan birini hatırlatmak gerekirse mobil platform üzerinde bir uygulama geliştirilerek form üzerinde şifreleme işlemi sağlanmış ve aynı ekranda bu şifrenin görüntüsü etiket olarak verilmiştir. Bu gibi çalışmalar RSA ve bunun gibi çeşitli algoritmaların mobil işletim sistemleri üzerinde kullanılabileceğini kanıtlamıştır.

Çalışmada RSA algoritması ile beraber, Sezar şifreleme tekniği referans alınarak programlanan özel bir metot ile haberleşme sistemi kurulmuştur. Haberleşme sistemi ile başarılı bir şekilde veri iletişimi sağlanmaktadır.

Çalışmada 4 farklı veri boyutu incelenmiştir. Bu veri boyutları arttıkça şifreleme süresi de artmıştır. Dikkat edilmesi gereken noktalardan birisi veri boyutu arttıkça haberleşme esnasında veri kaybının önlenmesidir. Veri kaybı ihtimaline karşı önlem alınmalıdır. Bu önlemlerden başlıcası veri kaybını önleme teknolojisi(Data Loss Prevetion) olabilir.[44] Bir başka çözüm RF teknolojisi kullanılarak geliştirilen kontrol algoritması uygulanarak sağlanabilir.[45] Bu uygulama verinin bir yerde toplanmasını sağlamaktadır. Telemetri adı verilen kablosuz haberleşme esnasında veri kaybının önlemeye yönelik geliştirilmiştir.

Çalışmada güvenlik ve hız kriterleri karşılaştırıldığı için deneysel çalışma sonucu elde edilen uygulamada hangi yöntemin kullanılacağı kişiden kişiye göre değişmektedir. Güvenlik ön plana çıkarıldığı için RSA algoritması ile yapılan şifreleme ortamı önerilmektedir.

Deneysel çalışma esnasında kullanılan algoritmalar ile ilgili olarak veri boyutunun artması ile gelen zaman kaybını önlemek için verilerin boyutlar indirgenebilir. Yani

veri sıkıştırma yöntemi ile kayıpsız ve boyutu daha az şekilde oluşturulan paketler halinde veri gönderimi sağlanabilir.

Veri iletişimi esnasında kullanılan veri tabanı ve sunucuların da güvenliğinin sağlanması çok önemlidir. Veri tabanına kimlerin ne zaman hangi metodla ve hangi içeriğe ulaştığı gibi önemli bilgileri tutan logların kaydedilmesi gerekmektedir. Veri tabanı güvenliğinin sağlanması için üçüncü parti bir yazılım ve ağ cihazı kullanılabilir. Sunucudan ayrı bir kurulumu olduğu için yük olmamaktadır. Veri tabanı güvenliği yine şifreleme algoritmaları ile de gerçekleşebilir. Sistemin güncel tutulması, gerektiğinde yamalar uygulanması ve denetlenmesi önem arz etmektedir.

Sunucu güvenliğinin sağlanması için anti virüs, firewall gibi yöntemler kullanılabilir. Saldırı tespit yöntemleri kullanılarak sunucu güvenli hale getirilebilir. SSH yöntemi kullanılarak port üzerinde gerekli değişiklikler yapılmaktadır. Ayrıca saldırı tespit yazılımları kullanılarak sunucu güven altına alınabilir.

Bu çalışma ile şifreleme algoritmalarının mobil işletim sistemleri üzerinde haberleşme imkânı sunabilmesi, bu iletişim esnasında güvenlik, hız gibi problemlerin karşılaştırılması sağlanmıştır. Şifreleme çalışmalarına bir yenisi daha eklenmiş olup, güvenlik ve hız konusunda farklı yöntemler üzerinde çalışılmıştır.

KAYNAKLAR

1. Canbek G. ,Sağıroğlu Ş., Bilgi, Bilgi Güvenliği ve Süreçleri Üzerine Bir İnceleme, Gazi Üniversitesi Politeknik Dergisi ,Cilt: 9 Sayı: 3,s. 165-174,2006.

2. Canbek, G., Sağıroğlu, Ş., "Şifre Bilimi Tarihine Genel Bakış - I", Türk Telekom Dergisi, Mayıs (Sayfa 34-42), 2005.

3. Canbek, G., Sağıroğlu, Ş., "Kötücül ve Casus Yazılımlar: Kapsamlı Bir Araştırma", Gazi Mühendislik Mimarlık Dergisi, Basımda, Temmuz 2006.

4. Fussell, R.S., Protecting Information Security Availability Via Self-adapting Intelligent Agents. Military Communications Conference, IEEE, 297s., 2005.

5. Mondal S, Bours P, A continuous combination of security & forensics for mobile devices, Journal of Information Security and Applications 40 63-77,2018

6. Maconachy V, W & Schou, Corey & Ragsdale, Daniel & Welch, Don,A Model for Information Assurance:An Integrated Approach. Proceedings of the 2001 IEEE Workshop on Information Assurance and Security.2001.

7. Karataş G.,Akbulut A.,Zaim H / Mobil Cihazlarda Güvenlik–Tehditler ve Temel Stratejiler, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 15(30), 2016.

8. Yıllara göre mobil kullanıcı trafiği, https://wearesocial.com/global-digital-report-2019

9. Dünya Mobil Kullanıcı İstatistikleri, https://wearesocial.com/global-digital-report-2019

10. Ju, H., et al., Implementation of a hardware security chip for mobile devices. IEEE Transactions on Consumer Electronics, 61(4), 500-506.2015.

11. Housman, E. M., "The Nature of Information", Bulletin of the American Society for Information Science, 26 (4): (April/May 2000).

12. Arda D.,Kodlama Teorisinin Kriptografik Açıdan İncelenmesi,Doktora Tezi, Trakya Üniversitesi ,EDİRNE,2011.

13. Şifreleme Yöntemleri, Eylül 2013, [Online] http://bidb.itu.edu.tr/seyir-defteri/blog/2013/09/07/şifreleme-yöntemleri. .

14. Harn L., Lin C., " Detection and identification of cheaters in (t,n) secret sharing scheme", Des. Codes Cryptography, 52 (1), 15-24. 2009

15. Aghayev M.,Kriptoloji ve Veri Şifreleme Teknikleri Üzerine, Yüksek Lisans Tezi, Ege Üniversitesi,İzmir,2017.

16. Wagstaff S. S. –(2002), Cryptanalysis of Number Theoretic Ciphers,Computational Mathematics, Chapman & Hall/Crc, ISBN 1-58488-153-4.

17. FIPS 197, Advanced Encryption Standard, Federal Information ProcessingStandard (FIPS), Publication 197, National Bureau of Standards, U.S. Department of Commerce,Washington D.C., November 26, 2001.

18. Başkök M.D.,AES Şifreleme Algoritmasının Modellenmesi, Yüksek Lisans Tezi, Gazi Üniversitesi,Ankara,2007 .

19. Vural H.,Simetrik Şifreleme Tekniklerinde Anahtar Planlama, Yüksek Lisans Tezi, Trakya Üniversitesi,Edirne,2014.

20. Başkök M.D.,AES Şifreleme Algoritmasının Modellenmesi, Yüksek Lisans Tezi, Gazi Üniversitesi,Ankara,2007.

21. Daemen, J., Rijmen, V., "The Design of Rijndael AES-The Advanced Encryption Standard", Springer, Germany, 10-17, 31-50 (2002).

22. Vural H.,Simetrik Şifreleme Tekniklerinde Anahtar Planlama, Yüksek Lisans Tezi, Trakya Üniversitesi,Edirne,2014.

23. Stinson D. R. Cryptography Theory and Practice, 2nd Ed., Chapman &Hall/Crc, ISBN 1-58488-206-9. – (2002),

24. Çimen C., Akleylek S., Akyıldız EŞifrelerin Matematiği Kriptografi,ODTÜ yayıncılık.,-(2011),

25. Koblitz N.-(1994)., A Course in Number Theory and Cryptography, 2nd Edition,Springer - Verlag, New York

26. Markus Jakobsson –(1998) A practical mix. InAdvances in Cryptology – EUROCRYPT, volume 1403 of Lecture Notes in Computer Science, pp. 448–461.SpringerVerlag, May/June 1998.

27. Java RSA Encryption and Decryption Example | ECB Mode + 4096 Bits + OAEPWITHSHA512ANDMGF1PADDING,https://www.javainterviewpoint.com/rsa-encryption-and-decryption/.

28. Jetbrains Developer Tools [Online] http://www.infoq.com/articles/jetbrains-developer-tools/.

29. The drive to develop [Online] https://www.jetbrains.com/company/.

30. InfoWorld review: Top Java programming tools[Online] http://www.infoworld.com/article/2683534/development-environments/infoworld-review--top-java-programming-tools.html.

31. Sunucu(Bilişim) [Online] https://www.wikizero.com/tr/Sunucu_(bilişim).

32. Karataş G.,Akbulut A.,Zaim H / Mobil Cihazlarda Güvenlik–Tehditler ve Temel Stratejiler, İstanbul Ticaret Üniversitesi Fen Bilimleri Dergisi, 15(30), 2016.

33. Souppaya M,Scarfone K, Guidelines for Managing the Security of Mobile Devices in the Enterprise, NIST Special Publication 800-124 Revision 1.

34. Mondal S,Bours P, A continuous combination of security & forensics for mobile devices Department of Information Security and Communication Technology, Norwegian University of Science and Technology, Norway ,Journal of Information Security and Applications 40 (2018) 63–77.

35. Kara R,Bodur H,Zavrak S, RSA Şifreleme Algoritması Kullanılarak SMS İle Güvenli Mesajlaşma Yöntemi,Düzce Üniversitesi,2016.

36. Neidhardt E, "Asymmetric Cryptography for Mobile Devices.", Service-centric Networking Telekom Innovation Laboratories and TU Berlin, Germany.

37. Gençoğlu H,Yerlikaya T, MOBİL CİHAZLARDA RSA ALGORİTMASININ PERFORMANS OPTİMİZASYONU,Trakya University Journal of Engineering Sciences, 18(1): 43-52, 2017 ISSN 2147–0308.

38. Kara R,Bodur H,Zavrak S, RSA Şifreleme Algoritması Kullanılarak SMS İle Güvenli Mesajlaşma Yöntemi,Düzce Üniversitesi,2016.

39. What is the React Native [online], https://www.oreilly.com/library/view/learning-react-native/9781491929049/ch01.html .

40. ŞEKER S.E, Yerine Koyma Şifrelemesi(Substitution Cipher),21.02.2008, http://bilgisayarkavramlari.sadievrenseker.com/2008/02/21/yerine-koyma-sifrelemesi-substitution-cipher/.

41. Caesar Cipher [Online],http://practicalcryptography.com/ciphers/caesar-cipher/.

42. AYDOĞAN Y. , Katlı Parçalı Sezar Şifreleme Metodu ve Uygulamaları , Gaziosmanpaşa Üniversitesi , Yüksek Lisans Tezi , 2014, Tokat.

43. What is the brute force?, https://resources.infosecinstitute.com/popular-tools-for-brute-force-attacks/#gref.

44. Data Loss Prevention , https://www.cybermagonline.com/veri-kaybi-onleme-data-loss-prevention-dlp-nedir.

45. Manır E, Çınar S.M, Tasarlanan akıllı telemetri kontrol algoritmasının incelenmesi, http://www.emo.org.tr/ekler/b8e8d37efc4c7ef_ek.pdf

ÖZGEÇMİŞ

1993 doğumlu Sercan BEDİR Kırıkkale'de dünyaya gelmiştir. İlk, orta ve lise eğitimini aynı şehirde tamamladıktan sonra 2011 yılında Kırıkkale Üniversitesi Bilgisayar Mühendisliğini kazanmış, 2015 yılında bu bölümü başarı ile tamamlamıştır.

2017 yılında yüksek lisans eğitimine Bozok Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği Anabilim Dalı'nda başlamıştır. Doç.Dr. Orhan ER danışmanlığında yüksek lisansına devam etmektedir. 2018 yılından bu yana özel bir eğitim kurumunda yazılım eğitmeni olarak görevine devam etmektedir.

İletişim bilgileri

Adres: Kurtuluş Mahallesi Şair Nedim Caddesi Merve Bozkurt Apt No:3/2,Merkez KIRIKKALE
Telefon: 05457337470
E-Posta: bedir-sercan@yandex.com