T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

HİBRİT ŞİFRELEME ALGORİTMASI

HAKAN GENÇOĞLU

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: Yrd. Doç. Dr. Tarık YERLİKAYA

EDİRNE-2017

T.Ü. Fen Bilimleri Enstitüsü onayı

Prof. Dr. Murat YURTCAN
Fen Bilimleri Enstitüsü Müdürü

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

Doç. Dr. Muharrem Tolga SAKALLI
Anabilim Dalı Başkanı

Bu tez tarafımca okunmuş, kapsamı ve niteliği açısından bir Doktora tezi olarak kabul edilmiştir.

Yrd. Doç. Dr. Tarık YERLİKAYA
Tez Danışmanı

Bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından Bilgisayar Mühendisliği Anabilim Dalında bir Doktora tezi olarak oy birliği ile kabul edilmiştir.

Jüri Üyeleri	İmza
Prof. Dr. İsmail TOK	
Yrd. Doç. Dr. Tarık YERLİKAYA	
Yrd. Doç. Dr. Deniz TAŞKIN	
Yrd. Doç. Dr. Gökhan ERDEMİR	
Yrd. Doç. Dr. Cenk MISIRLI	

Tarih: 10/01/2018

T.Ü. FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ DOKTORA PROGRAMI

DOĞRULUK BEYANI

İlgili tezin akademik ve etik kurallara uygun olarak yazıldığını ve kullanılan tüm literatür bilgilerinin kaynak gösterilerek ilgili tezde yer aldığını beyan ederim.

10/01/2018

Hakan GENÇOĞLU

Doktora Tezi
Hibrit Şifreleme Algoritması
T.Ü. Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Bilgi teknolojilerinde veri güvenliği için kullanılan simetrik ve asimetrik algoritmaların kendilerine özgü avantajları ve dezavantajları bulunmaktadır. Avantajlar kullanılarak ortadan kaldırılmaya çalışılan dezavantajlar ortaya hibrit mekanizmanın ortaya çıkmasını sağlamıştır.

Hibrit mekanizmalar üzerine yapılan çalışmalar KEM – DEM mimarileri üzerine yoğunlaşmıştır. DEM – Data Encapsulation Mechanism mesajın kendisini şifrelemekte, KEM - KEY Encapsulation Mechanism verinin şifrelendiği anahtarın şifrelenmesi işlemini gerçekleştirerek güvenlik altına almaktadır.

Çalışmamızda klasik KEM-DEM yapısından farklı olarak mimari sadece veri ve verinin şifrelendiği anahtar olmaktan çıkarıp mimariye bir de güvenlik paketi eklenmiştir. Güvenlik paketinde doğrulama işlemleri ile ilgili veriler bulunmakta güvenlik paketi de şifrelenerek güvenlik altına alınmıştır.

Gerek oturum açma, gerek şifreli iletişim sırasında özellikle anahtarlar olmak üzere kişisel bilgiler önemlidir. Kişisel bilgilerin detaylarını ve şifreleme anahtarlarını akılda tutmak zor olabilir. Bu bilgiler doğrulama ve imzalama işlemleri için kullanılacaktır. Bu nokta düşünülerek bu bilgilerin, güvenliği sağlanmış elektronik sertifikalarda tutulması önerilmiştir. Çalışmada elektronik sertifikaların hukuki karşılığından bahsedilmiş ve mimari içinde kullanım biçimi anlatılmıştır.

Ayrıca mimarinin testleri .NET platformunda yazılan uygulamalarla test edilmiş. Asimetrik algoritma (RSA) için mobil platform ile de test gerçekleştirilmiş ve performans testi yapılmıştır.

Yıl : 2018
Sayfa Sayısı : 115
Anahtar Kelimeler : Hibrit şifreleme, Hibrit Algoritma, Elektronik İmza

Doctoral Thesis

Hybrit Encryption

Trakya University Institute of Natural Sciences

Computer Science

ABSTRACT

Asymmetric and symmetric algorithms used at information technologies for information security have some advantages and also disadvantages. Hybrid encryption mechanism came up to remove the disadvantages.Most of the studies on hybrid mechanism are about KEM-DEM. DEM – Data Encapsulation Mechanism encrypts data and KEM – Key Encapsulation Mechanism encrypts keys to secure the data.

In this study, we offer a new architecture different from KEM-DEM mechanism. Data encryption like DEM, key Encryption like KEM and also data security package are the parts of our mechanism. Data security package consists of the data related to the verification processes. Data security package is also encrypted.

Keys and personal information are vital when to login to the system and also in the process of transmission. Details of personal data and the keys can be difficult to keep in mind. These data are going to be used for verification and signing. For that reason, it has been offered for the data to be stored in the secure electronic certificates. We stated the legal status of electronic certificates, and expressed the way of the use in our architecture.

Personal information, especially encryption-decryption keys, is need during encrypted communication or login to a system. Details of keys or personal information can be hard to remember. This information is used for signing and verification the user because of that they are needed. For that reason personal information certificates offered to store this information. A software developed in .NET platform to test the architecture, and also an application developed for the test of asymmetric algorithm RSA in mobil devices.

Year : 2018

Number of Pages : 115

Keywords : Hybrid Encryption, Hybrid Algorithm, Electronic Signature

ÖNSÖZ

Bilgisayar teknolojilerinin gelişmesi, internetin hızlanması, mobil cihazların yaygınlaşması teknolojinin hayatımızın bir parçası olmasını sağlamıştır. Güvensiz bir ortam olan internette güvenli bir şekilde dolaşmak için veri güvenliği daha da önemli hale gelmiştir. Şifreleme tekniklerini kullanarak bu güvenlik sağlanabilir. Fakat şifreleme algoritmaları olduğu kadar bu algoritmaların kullanıldığı güvenlik mimarisi de en az algoritmalar kadar önemlidir.

Veri iletişimi sırasında sadece veriyi şifrelemek birçok durumda yeterli olmayabilir. Aynı zamanda iletişim kuran kişiler doğrulanmalı, iletilen verinin değişmediği ve sahibinin iddia edilen kişi olduğu ispat edilmesi de gerekebilir. İşte bu gibi detaylar mimariye önem kazandırır.

Tezimizde asimetrik ve simetrik şifreleme algoritmaları birlikte kullanılarak hibrit bir güvenlik mimarisi kurulmuştur. Simetrik ve asimetrik algoritmaların kendilerine özgü avantajları kullanılarak dezavantajları ortadan kaldırılmış, tüm platformlarda performanslı çalışacak bir mimari oluşturulmuştur.

Bu tez konusunu belirleyen ve çalışmalarım sırasında beni yönlendiren değerli danışmanın Yrd. Doç. Dr. Tarık YERLİKAYA'ya teşekkürlerimi sunarım. Ayrıca tez izleme komitesinde bulunan sayın hocam Yrd. Doç. Dr. Deniz TAŞKIN hocama yaptığı yapıcı eleştiriler ve tezin şekillenmesindeki katkıları için teşekkür ederim.

Ayrıca bu zorlu süreç içinde beni sabırla destekleyen aileme şükranlarımı sunarım.

İçindekiler

SİMGELER

$\oplus$ = Bit Düzey XOR

$\wedge$ = Bit Düzey AND

$\vee$ = Bit Düzey OR

$\neg$ = Bit Düzey Tamamlama

$+$ = Mod 2^{32} Toplama

R^n = N bit sağa öteleme

S^n = N bit sağ öteleme

$\delta_{K1} = K1$ Anahtarını kullanan Simetrik şifreleme algoritması

$\delta_{K2} = K2$ Anahtarını kullanan Simetrik şifreleme algoritması

$\delta_{K2} = K2$ Anahtarını kullanan, güvenlik paketi öncesi, simetrik şifreleme algoritması

$z =$ Karıştırılmış Şifreli Metin

$q =$ Katsayı Dizisi

$y =$ Şifrelenmiş Metin (Mesaj Paketi)

$SG_{SK} = SK$ Anahtarını kullanan imza Algoritması (Cihaz-Kişi)

$H =$ Özetleme (Hash) Algoritması

SKC = Cihaz Gizli Anahtarı (Asimetrik Algoritma)

SKK = Kullanıcının Gizli Anahtarı (Asimetrik Algoritma)

PK = Kullanıcının Açık Anahtarı (Asimetrik Algoritma)

I_{mC} =Cihaz tarafından oluşturulan mesaja ait imza

I_{mK} =Kişi tarafından oluşturulan mesaja ait cihaz imzasını imzalama

I_{zC} =Cihaz tarafından oluşturulan karıştırılmış şifreli mesaja ait imza

I_{zK} =Kişi tarafından oluşturulan karıştırılmış şifreli mesaja ait cihaz imzasını imzalama

$m =$ Açık Mesaj

$h_m =$ Mesaj Özeti

$h_z =$ Karıştırılmış şifreli metin özeti

q =Katsayı Dizisi

$\tau =$ Güvenlik Paketi

γ= Asimetrik şifreleme algoritması

$K1 =$ Mesaj Şifreleme Anahtarı (Simetrik Algoritma)

$K2$ = Güvenlik Paketi Şifreleme Anahtarı (Simetrik Algoritma)

$Ap =$ Anahtar paketi

$Gp =$ Güvenlik paketi

$Mp =$ Mesaj paketi

I_C =Cihaz tarafından oluşturulan mesaja ait imza

I_K =Kişi tarafından oluşturulan cihaz imzasını imzalama

$Sh =$ Karıştırma Algoritması

ŞEKİLLER LİSTESİ

TABLOLAR LİSTESİ

1. GİRİŞ

Geçmişten günümüze iletişim gizliliği önemli olmuştur. Geçmiş zamanlarda iletişim gizliliği askeri iletişim türleri için önemliyken, günümüzde günlük hayatın önemli bir parçası haline gelmiştir.

Eski zamanlarda uzak mesafeler arasında haberleşmeler haberciler ile yapılmaktaydı. Bu durum gönderilen veriyi güvensiz hale getiriyordu. Habercinin veya taşıdığı bilginin ele geçirilmesi haberin düşmanların eline geçmesi ve planların bozulması anlamına gelmekteydi. Bu güvenlik zafiyetleri habercinin başına bir şey gelse bile bilginin güvenliğin sağlamak zorunluluğunu doğurmuştur. Bilgi düşmanın eline geçse bile düşmanın bunu anlayamaması gerekir. İşte bu yaklaşım kriptografinin temellerini atmıştır.

Eski çağlarda son derece ilkel metodlar kullanılmıştır. Bu metotlar basit öteleme ve yer değiştirme işlemlerini içermektedir. O zamanlar için son derece etkili olan metotlar zamanla güvensiz olmaya başlamış ve sürekli yenileri geliştirilmiştir. Geçmişte kullanılan şifreleme algoritmalarına aşağıdaki örnekler verilebilir:

Ebced Hesabı: Her harfe karşılık gelen sayısal değerler kullanılarak metinlerin sayısal karşılığını hesaplama ve bu sonuçtan anlamlar çıkarma işlemidir. Büyük oranda semitik alfabeler de geçerlidir. Kelimelerin sayısal değerlerinin bir takım şifreler içerdiğine inanılır ve bu şifrelere ait anlamlara erişilmeye çalışılır.[1]

Sezar Şifresi: Sezar şifresi bu metotların içinde en bilinenidir. Bir yer değiştirme şifresi olan sezar şifresi M.Ö. 60-50 yıllarında Roma imparatoru Julius Sezar tarafından geliştirilmiş ve kullanılmıştır. Şifreleme işlemi için alfabe 3 harf sola kaydırılır ve açık metindeki harfler yeni alfabe de yerlerine gelen harflerle değiştirilerek şifreleme işlemi yapılır. [2,3]

Skytale: Yunanlılardan tarafından M.Ö. 5 yy da askeri amaçlı kullanılan bir tekniktir. Aynı zamanda ilk kriptografik cihaz olarak da bilinir. Uzun bir şerit silindir biçimindeki bir sopanın etrafına sarılır. Mesaj, boylamasına doğru sopa üzerine her bir satıra bir harf gelecek şekilde yazılır. Sopa üzerine sarılan şerit açılınca yazılan mesajın anlamsız harfler dizisi oluşturduğu görülür. Şifreli mesajın çözülmesi de aynı şekildedir. Aynı uzunlukta ve çapta bir sopanın kullanılması gerekir. [2]

Polybius'un Dama Tahtası: Yunanlılar tarafından M.Ö. 205-123 yılları arasında geliştirilmiş ve kullanılmıştır. Elemanları harfler olan 5x5 (Yunan alfabesine uygun olarak) tablo oluşturulur. Harfler sırasıyla bu tablonun satırlarına yazılır. Her harfin bulunduğu satır numarası ve sütun numarası o harf için belirlenen adrestir. Harflerin yerine bu adresler yazılarak şifreleme yapılır. [4]

Vigenere Şifresi:1586 yılında Fransız diplomat Blaise Vigenere tarafından duyurulan şifredir. Uzun süre kırılamayan şifre olarak anıldı. Tek alfabeli kaydırma şifrelerinin çok alfabeli olarak düzenlenmiş halidir. Hangi alfabenin kullanılacağı seçilen bir anahtar ile belirlenir. 1854-1963 yılları arasındaki çalışmaları sonucunda İngiliz matematikçi Charles Babbage ve Avusturya ordusunda görevli kriptografi uzmanı Friedrich Kasiski tarafından kırılmıştır.[5]

Affine Şifreleme: Birinci dereceden bir bilinmeyenli lineer bir fonksiyondur.

$y = ax + b \bmod N$(1.1)

Bu fonksiyon K uzayında tanımlı olmalıdır.

Fonksiyonda a ve b değerleri daha önce belirlenmiş sabit değerlerdir ve N değerinden küçük seçilmelidir. x değeri ise şifreleme işlemine tabi tutulacak değerlerdir. Her bir x değeri için farklı bir y değeri bulunarak şifreleme işlemi gerçekleştirilmiş olur. N değeri ise şifrelemek için x yerine yazılacak olan karakterlerinin sayısal değerlerinin bulunduğu uzayın eleman sayısıdır. Örneğin bizim alfabemizdeki harf sayısı.

Örneğin şifrelenecek kelime 'HAKAN' ve şifreleme fonksiyonu $y = 3x + 5$ olsun. Şifrelenecek kelimenin her bir harfinin sayısal değeri x yerine yazılarak y değerleri bulunur.

x	$y = 3x + 5 \bmod 29$
H=10	$y = 3 * 10 + 5 \bmod 29 = 6$=E
A=1	$y = 3 * 1 + 5 \bmod 29 = 8$=G
K=14	$y = 3 * 14 + 5 \bmod 29 = 18$=O
A=1	$y = 3 * 1 + 5 \bmod 29 = 8$=G
N=17	$y = 3 * 17 + 5 \bmod 29 = 27$=V

Böylece 'HAKAN' kelimesi 'EGOGN' şeklinde şifrelenmiş olur.

Şifreyi çözmek için de

$$x = \frac{y-b}{a} \bmod N \quad (1.2)$$

Fonksiyonu kullanılır. Örneğimize göre bu fonksiyon

$$x = \frac{y-5}{3} \bmod 29 \text{ dur}$$

x	$x = \frac{y-5}{3} \bmod 29$
E=6	$x = \frac{6-5}{3} \bmod 29 = 10 = H$
G=8	$x = \frac{8-5}{3} \bmod 29 = 1 = A$
O=18	$x = \frac{18-5}{3} \bmod 29 = 14 = K$
G=8	$x = \frac{8-5}{3} \bmod 29 = 1 = A$
V=27	$x = \frac{27-5}{3} \bmod 29 = 17 = N$

Aslında şifre çözmek için yapılan işlem (1.1) deki fonksiyonun tersini alıp (1.2) deki fonksiyonu elde etmek ve yine aynı uzayda modunu almaktan ibarettir. Sonra yapışan işlem şifreli metindeki harflerin sayısal karşılıklarını y yerine yazarak asıl x değerlerini bulmaktır. Böylece 'EGOGN' şifreli metni 'HAKAN' olarak çözülmüş olur.

Kerckhoffs Prensipleri: Hollandalı dilbilimci Auguste Kerckhoffs tarafından 1883 yılında yayınlanan "La Cryptographie Militaire" adlı makalede şifreleme sisteminin güvenliğinin sadece anahtara bağlı olması gerektiği, algoritma tamamen bilinse bile şifrenin kırılamaması gerektiği açıklanmıştır.

- Sistem pratik olmalı ve matematiksel bir altyapıya dayanmalıdır.
- Sistem gizli olmamalıdır. Sistem herkes tarafından bilinebilir.
- Sistem anahtarları kullanıcılar arasında kolayca değiştirilebilmelidir. Üçüncü şahıslar bu duruma müdahale edememelidir.
- Sistem telgraf uygulamasında kullanılabilmelidir.
- Sistemin uygulanabilmesi için çok fazla sayıda kişiye ihtiyaç duyulmamalıdır.

- Sistemin uygulanması ve anlaşılması kolay olmalıdır. Şifreleme sisteminin güvenliği şifreleme algoritmasını gizliliğine dayanmamalı, yalnızca anahtar gizliliğine dayanmalıdır.[6]

İnternet veri transferi için kullanılan güvensiz bir ortamdır. Mesajlaşma, e-posta gönderip alma, bankacılık ve finans işlemleri vb. günlük işlerin birçoğu internet altyapısı kullanılarak gerçekleştirilir. Bu işlemler gerçekleştirilirken gizli kalması gereken bazı kişisel bilgilerin kullanılması gerekebilir. Bu şekilde kullanılan bilgiler internet ortamında açık bir şekilde iletilir. Basit donanımlar ve yeterli bilgi ile iletişim hatlarını dinleyen herkes bu bilgilere ulaşabilir. Özellikle bankacılık, e-devlet gibi işlemlerde ortaya çıkan bu problem verilerin şifrelenerek iletilmesi ile çözüme kavuşmuştur. Günümüzde şifreleme algoritmaları haberleşme, dosya ve veri güvenliği, e-ticaret, sayısal imza, e-posta, güvenlik protokolleri vb. bir çok alanda kullanılmaktadır.

İletişim teknolojilerinin gelişmesi ve hızlı işlem yapabilme yeteneğinin artması daha modern ve güçlü algoritmaların gelişmesine sebebiyet vermiştir. Günümüzde çok daha karmaşık şifreleme sistemleri mevcuttur. Simetrik ve asimetrik olarak gruplandırılan bu algoritmalar karmaşık matematiksel yapılar üzerine kurulmuştur. Bu algoritmalardan ilerleyen konularda bahsedilecektir.

2. VERİ GÜVENLİĞİ

İletişim teknolojilerinin hızlı bir şekilde gelişmesi günlük hayatta kullanımını artırmıştır. Önce evlerimize giren telefonlar daha sonra ceplerimize girmiş ve daha sonra akıllı telefonların gelişmesiyle günlük hayatın vazgeçilmezi olmuştur. Eş zamanlarda gelişen internet teknolojileri de bilgiye ulaşma, yönetme, depolama gibi konularda kolaylık sağladığı gibi kamu ve özel sektöre ait sağlık, finans, eğitim, bankacılık gibi işlemleri daha hızlı ve kolayca yapabilmemize olanak sağlamıştır. Akıllı telefonlar ve internetteki bu gelişim sadece ihtiyaç için değil aynı zamanda eğlence, sohbet, sosyal medya gibi alanların gelişmesini sağlamış, bu teknolojilerin keyif içinde kullanılabilmesinin önünü açmıştır.

Veri iletişim teknolojileri güvensiz ortamlardır. Yani bilgiler bilgisayarların veya akıllı cihazların anlayabileceği elektrik sinyallerine çevrilerek iletilirler. Yeterli donanıma ve bilgiye sahip herkes bu bilgileri ele geçirebilir, değiştirebilir kısacası kendi çıkarları için kullanabilir. İletilen veriler gizli kalması gereken veriler olması durumunda bu durum çok önemli bir problem teşkil eder.

Örneğin e-ticaret sistemlerini düşünelim. Güvenliği sağlanmamış bir sistemde alışveriş yapmak için kullandığımız kredi kartı numarası, son kullanma tarihi, cvc numarası gibi bilgiler üçüncü şahısların eline geçebilir. Bu durum hem alıcının ekonomik olarak zarara sokar hem de satıcının itibar ve müşteri kaybetmesine sebebiyet verir.

Türkiye Cumhuriyeti Kalkınma Bakanlığı'nın hazırladığı 2015-2018 Bilgi Toplumu Stratejisi Eylem Planı'nda bilgi işlem teknolojilerinin yaygınlaşması ve geliştirilmesi sırasında bilgi güvenliğinin öneminden bahsedilmiştir. Planda kullanıcıların bilgi işlem teknolojilerini kullanırken kişisel verilerin güvenliğinin sağlanması ve alınması gereken tedbirler açıklanmıştır. Rapora göre " e-Sağlık, akıllı şebekeler, akıllı ulaştırma gibi BİT uygulamalarının kullanımının yaygınlaşması ile bilgi güvenliği günlük yaşamın sürdürülmesi için vazgeçilmez birnoktaya gelmektedir. Diğer yandan, nesnelerin interneti, büyük veri, mobil teknolojiler, yakınsama, bulut bilişim, dış kaynak kullanımı gibi teknolojik gelişme ve eğilimlerin de etkisiyle BİT güvenliği ve kullanıcı güveninin sağlanması daha da zorlaşmaktadır. Bu nedenle

hizmetlerin kesintisiz ve kaliteli sunumu ve hizmetten faydalananların veri mahremiyetinin temini için ülkelerin güçlü bir hukuki, kurumsal ve teknik altyapıya sahip olmalarının önemi artmaktadır.” denilmektedir. [7]

Aynı rapora göre internet üzerinden yapılan işlemlerin yükselmesi ve ekonomik hacmin artması, güvensiz bir ortam olan internetin yapısı nedeniyle suç çeşitlerinin ve miktarının artması öngörülmüştür. [7]

2.1 Bilgi GüvenliğiİlkeleriBilgi Güvenliği İlkeleri

İnternet ortamındaki hizmet çeşitliliklerinin artması karşılaşılan problemlerinde çeşitliliğinin artmasını sağlamıştır. Bilgi güvenliği sadece bilgilerin ele geçirilmesinin önüne geçmek yaklaşımı olmaktan çıkmıştır. Artık verilerin değiştirilmesi, kaybolması, vs. sorunlarda söz konusudur. Bu problemlerin tamamına, şifreleme algoritmaları kullanılarak çözüm bulmak mümkündür. Bilgi sistemleri için güvenlik çözümleri geliştirilirken sağlanması gereken bazı standartlar şunlardır:

2.1.1 Gizlilik

Verinin içeriğinin, iletişim sırasında, depolama alanlarında vs. yetkisiz kişilerce görülmemesi gerekir[8,9,10,11,12].

2.1.2 Bütünlük

Verinin işlenmesi işlemi sonucunda, başlangıçtaki hali korunmalıdır. Örneğin transfer sırasında veri değişime uğramamalı asıl hali korunmalıdır. Korunma altına alınmamış olan veri iletişim sırasında yetkisiz ve kötü niyetli kişiler tarafından ele geçirilip alıcıya verinin değiştirilmiş hali gönderilebilir. Bu ve benzeri durumların önüne geçmek için veri bütünlüğü kontrol edilmelidir. Bu kontrol için çeşitli metotlar bulunmaktadır. Bu metotlar ulaşan verinin bütünlüğü ve doğruluğunu kontrol ederek ulaşan verinin gönderilen veriyle olan durumunu garanti eder[8,9,10,11,12].

2.1.3 Doğrulama

Mesaj gönderenin kimliği doğrulanabilmelidir. Bir mesajın en önemli unsurlarından biri göndericisidir. İçeriğinin tutarlılığı açısından mesajı gönderen kişinin kimliği önemlidir. Göndericinin doğrulanması mesajın güvenli olduğu anlamına gelir[8,9,10,11,12].

2.1.4 İnkâr Edememe

Verinin gönderici tarafından inkâr edilmemesi gerekir. Özellikle finans ile ilgili işlemlerde oluşabilecek anlaşmazlıkları önlemek için kullanılır[8,9,10,11,12].

2.1.5 Kimlik denetimi

Göndericinin ve alıcının karşılıklı olarak birbirlerine ait kimlikleri doğrulayabilmesidir[8,9,10,11,12].

2.1.6 Erişilebilirlik

Yetkilileringerektiğinde gerekli bilgiye ulaşabilmesidir[8,9,10,11,12].

Bu ilkeler verinin, alıcı tarafından, güvenli bir şekilde değerlendirilmesini sağlayacaktır. Bu özelliklerin tamamı kriptografi algoritmaları ile sağlanabilir.

3. KRİPTOLOJİNİN TEMEL KAVRAMLARI

Teknolojinin gelişmesi, bilgisayarların işlem hızının artması kriptolojinin gelişmesinde önemli katkıda bulunmuştur. Gerek veri şifrelemek için kullanılan kriptograflar, gerekse şifrelenmiş verileri açmak için çalışan kriptanalistler kafalarındaki yapıyı deneme hamallığını bilgisayarlara bırakmışlar sadece teorik çözümler üzerine çalışma fırsatını yakalamışlardır. Bu fayda bilim adamlarının çalışma sürecini hızlandırmış ve daha güçlü yapılar ortaya çıkmaya başlamıştır. Kriptoloji ile ilgili bazı terimlerden bahsedelim.

Kriptoloji: Kriptografi ve kriptoanaliz disiplinlerinin genel adıdır.Matematik tabanlıdır.Kriptografi, Kriptoanaliz, steganografi gibi alt dallara ayrılır. [12]

Kriptografi: Açık verileri gizleme üzerine çalışmalar yapan bilim dalıdır. [12]

Kriptoanaliz: Gizlenmiş verileri çözmeye, ve veri gizleme algoritmalarını analiz etmeye çalışan bilim dalıdır.[12]

Anahtar: Kriptografi algoritmalarının açılmasını sağlayan araçtır. Anahtar büyüklüğü güvenliği artırır. Büyük anahtarlar yüksek güvenlik sağlarken küçük anahtarlar zaman kazandırır.

Plaintext (Açık Metin): Açık metin veya orjinal metin

Ciphertext (Şifreli Metin): Şifreli metin veya gizli metin

Encryption (Şifreleme) : Açık metni şifreli metine çevirme süreci.

Decryption (Şifre Çözme) : Şifreli metni açık metine çevirme süreci.

Kriptoanalist (Saldırgan): Anahtarı bilmeden şifreli metni, açık metne çevirmek için analiz eden kişi.

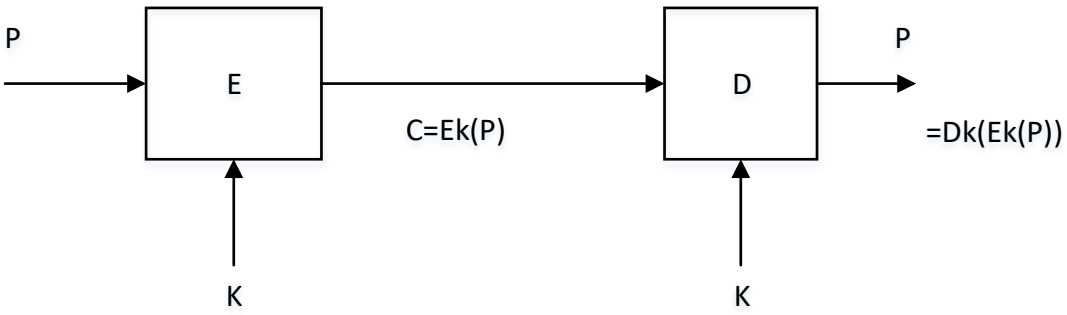
4. KRİPTOGRAFİ

Kriptografik algoritmalar çoğunlukla matematiksel fonksiyonlar üzerine kuruludur. Bu algoritmalar herkese açık ve içeriği bilinen algoritmalardır. Zaten Kercshoff prensiplerine göre algoritma gizli olamamalı, veri güvenliği anahtara bağlı olmalıdır. Bu algoritmaların çalışabilmesi için anahtar önemlidir. Kısacası veri güvenliği iki noktaya bağlıdır:

- Algoritmanın gücü
- Anahtarın gizli olması

Algoritmanın yapısı herkes tarafından bilinmesine rağmen güçlü olabilir.Algoritma ne kadar güçlü olursa olsun, gizli veriye ulaşmak için gerekli olan şifre çözme anahtarı sadece alıcı tarafından biliniyor olması gerekir. Algoritma sabit olduğundan anahtar değiştikçe aynı metine ait farklı şifreli metinler elde edilebilir. Anahtar seçilirken dikkatli davranılmalıdır. Kolay tahmin edilemeyen ve uzun bir anahtar seçilmesi gerekir. Anahtar uzunluğu arttıkça tahmin edilmesi ve algoritmanın gücüne de bağlı olarak veri güvenliği artacaktır.

Kripto algoritmaları tasarlanırken dikkat edilmesi gereken en önemli noktalardan biri de anahtar uzayı kavramıdır. Yani şifreleme işlemi için kullanılacak anahtarın seçileceği küme-uzay yeterince büyük olmalıdır. Aksi takdirde olası tüm anahtarlar denenerek şifreli metin çözülebilir.



Şekil 4.1. Şifreleme ve şifre çözme işlemi

- P: Şifrelenmemiş bilgi ya da metin.
- K: Anahtar, şifrelenmemiş bilgiye eklenir.
- C: Şifrelenmiş bilgi ya da metin

- E: Şifreleme bloğu ya da tekniği
- D: Şifreyi çözme bloğuya da tekniği

Kriptanalizin en önemli hedeflerinden biri de anahtarın bulunmasıdır. Eğer anahtar bulunursa sadece şifreli veri çözülmez aynı zamanda daha sonraki mesajlaşmalarda da şifreli metinlerin çözülmesi mümkün olur.

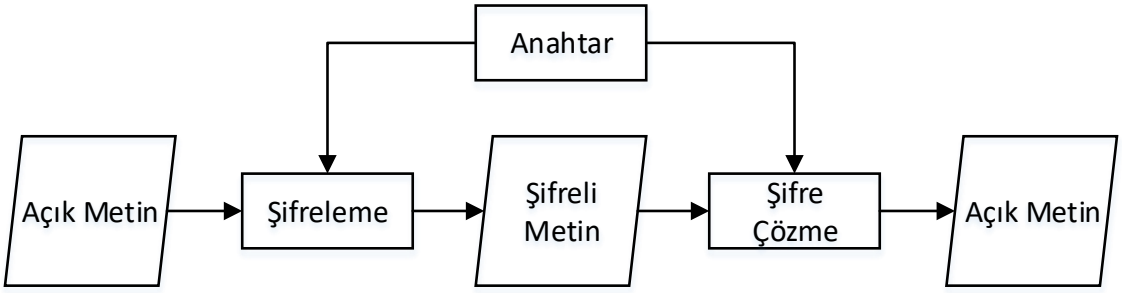
5. KRİPTOLOJİ ALGORİTMALARI

Şifreleme algoritmaları genel anlamda aşağıdaki gibi gruplandırılabilir. [13,14]

- Simetrik Algoritmalar (Tek Anahtarlı Sistemler)Blok Şifreler
  - Akış Şifreler
- Asimetrik Algoritmalar (Çift Anahtarlı Sistemler)
- Hash Algoritmaları (Özetleme Algoritmaları)

5.1 Simetrik Algoritmalar

Simetrik algoritmalar şifrelemek ve şifre çözmek için tek bir anahtar (Secret Key) kullanır. Kerchoff prensipleri temelinde bu anahtar gizli olarak tutulmalıdır. Bu anahtar hem gönderici hem de alıcı tarafından bilinmelidir. Gizli anahtar, iletişim kurulmadan önce, gönderici ve alıcı arasında güvenli bir yol kullanılarak paylaşılmıştır. Tek anahtarlı bu sistemler, çift anahtarlı sistemlere göre çok hızlı çalışılırlar. Bununla beraber, asimetrik algoritmalara göre saldırıya karşı daha az dirençlidirler. Makine diline uygun olarak tasarlanırlar. Basit öteleme, yer değiştirme, xor işlemi gerçekleştirmek gibi işlemler yaparak şifreleme işlemini gerçekleştirirler.



Şekil 5.1. Simetrik anahtar şifrelemesi

5.1.1 Blok Şifreler

Blok şifreler veriyi sabit uzunluklu bloklara ayırarak işler. Şifreleme ve şifre çözmek için aynı anahtarı kullanır. Blok Şifreler yer değiştirme ve doğrusal dönüşüm işlemlerini kullanır. Şifreli metin ve açık metin arasındaki ilişki yer değiştirme ile gizlenir, doğrusal dönüşüm işlemi ise açık metindeki izlerin şifreli metinde sezilmemesini sağlar. Blok şifreler Feistel veya Permutasyon (SPN) ağları mimarisini kullanır.

Blok şifrelerin gücünü sağlayan faktörler şunlardır[6]:

- Anahtar Büyüklüğü
- S-Kutuları (Yer Değiştirme Kutuları)
- Doğrusal Dönüşümler

Anahtar büyüklüğü saldırılara karşı güçlü olmalıdır.

Tablo 5.2. Bazı Blok Şifreler ve Anahtar Uzunlukları[15]

Blok Şifreler	**Anahtar Uzunluğu**
DES	56 bit
IDEA	128 bit
AES	128,192, 256 bit
Camellia	128, 192, 256 bit
ARIA	128 bit
Khazad	128 bit

S-Kutuları blok şifreler için çok büyük önem arzeder. Karıştırma işlevini görürler. Doğrusal olmadıklarından dolayı iyi tasarlanmış s-kutusu algoritmanın gücünü doğrudan etkiler. S-kutularının amacı, bit bloklarının s-kutusu içindeki karşılıkları ile yer değiştirilmesidir[15].

Doğrusal Dönüşümler ise sabit uzunluktaki bir giriş bloğunu doğrusal olarak karıştırarak aynı uzunlukta bir çıkış bloğu elde etmek için gerçekleştirilir.AES-RijnDael örnek olarak verilebilir[15].

5.1.1.1 Blok Şifreleme Örneği AES-Rijndael

AES algoritması 2001 yılında ABD Ulusal Standart Enstitüsü tarafından standart olarak kabul edilmiştir. AES (Advanced Encryption Standart) için yapılan yarışmada 15 algoritma yarışmış ve Rijndael algoritması yarışmayı kazanmıştır. Algoritmalar güvenlik ve performans açısından değerlendirilmiştir. [16]

AES (Advanced Encryption Standart) olarak bilinen Rijndael algoritması Vincent Rijmen ve Joan Daemen tarafından geliştirilmiştir. Rijndael, ismini geliştiricilerinden almıştır: RIJmen aNd DAEmen.

AES-Rijndael:[7]

AES, Değiştirme-Karıştırma (Substutition-Permutation) üzerine kuruludur. 128-bit girdi bloğu, 128,192 ve 256 bit anahtar uzunlukları ile şifreleme yapar.

AES, 4x4 lük durum (state) matrisi üzerinde çalışır. Matristeki işlemler de özel bir sonlu cisim (finite field) üzerinde yapılmaktadır.

AES algoritması 128-192-256 bit anahtarla şifreleme işlemleri için sırasıyla 10-12-14 döngüden oluşur.

Her bir döngü (son döngü hariç) dört adımdan oluşur.

İlk Adım

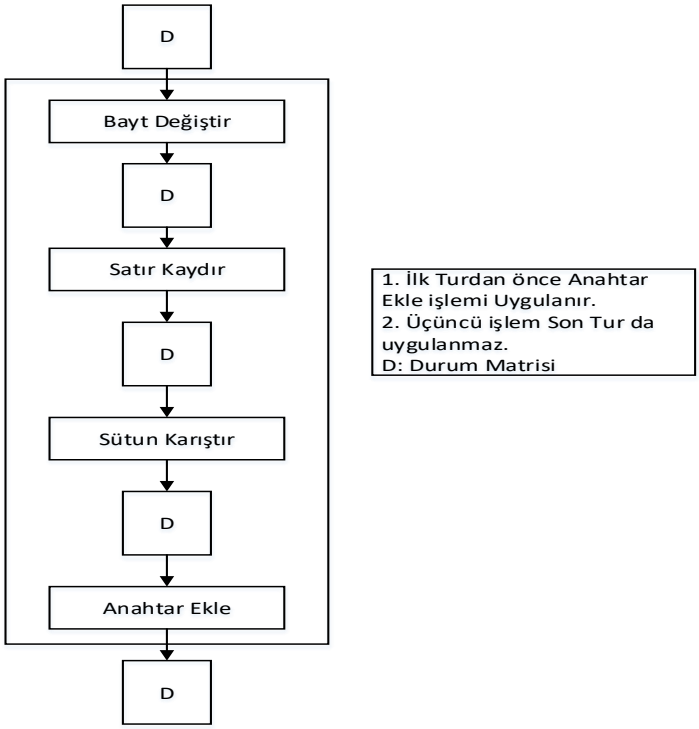
1. Anahtar Ekle (AddRoundKey)

Döngüler (9 tekrar)

1. Bayt Değiştir (SubBytes)
2. Satır Kaydır (ShiftRows)
3. Sütun Karıştır (MixColumn)
4. Anahtar Ekle (AddRoundKey)

Son Döngü

1. Bayt Değiştir (SubBytes)
2. Satır Kaydır (ShiftRows)
3. Anahtar Ekle (AddRoundKey)



Şekil 5.3. AES Bir Döngü Yapısı

Bayt Değiştir (SubBytes) Adımı

Durum matrisindeki her değer s-kutusu içinde kendisine karşılık gelen değer ile değiştirilir. Bu adımın amacı doğrusallığı bozmaktır. S-kutusu GF(2^8) üzerinde ters alma işlemi ile oluşturulmuştur.

Satır Kaydır (ShiftRows) Adımı

Matrisin her satırında kaydırma işlemi yapılmasıdır. 1. satır da herhangi bir değişiklik yapılmaz. 2. satırda 1, 3. satırda 2, 4. satırda 3 byte sola doğru kaydırılır. Rijndael algoritmasında ise 256-bit için bu değerler, ilk satır sabit kalacak şekilde, 1, 3 ve 4 bayttır.

Sütun Karıştır (MixColumn) Adımı

Sütun Karıştır işlemi, her sütunun sabit bir matrisle çarpılması işleminden oluşur. Bu sabit matris aşağıda verilmiştir:

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

Matris çarpması işlemi sonlu cisim GF (2^8) üzerinde yapılmaktadır. Her bayt bu sonlu cisim üzerinde bir polinom tanımlayacak şekilde, mod x^4+1'de

$$c(x) = 0x03 \cdot x^3 + x^2 + x + 0x02 \quad (5.1)$$

polinomu ile çarpılır. SütunKarıştır adımındaki sabit matris bir MDS matristir.

Anahtar Ekle (AddRoundKey) Adımı:

Bu adımda anahtar planlamasından gelen döngü anahtarı durum matrisi ile XOR işlemine tabi tutulur.

AES Şifresinde Anahtar Planlama

AES-Rijndael de her döngüde, anahtar planlamadan gelen, farklı bir anahtar kullanılır.

Anahtar genişletme evresi, şifreleme anahtarını her döngü için bir anahtar haline getirir. Eğer döngü sayısı Nr ise anahtar genişletme işleminden Nr +1 anahtar elde edilir. İlk anahtar döngü başlamadan önce kullanılırken geri kalan döngü anahtarları her döngünün sonundaki son dönüşüm olarak kullanılır. 128 bit döngü anahtarını tek bir 128 bit şifre anahtarından elde eder.

Anahtar genişletme işlemi döngü anahtarlarını kelime kelime (32bit 32 bit) oluşturur. Bu evre aşağıdaki gibi tanımlanan 4×(Nr +1) kelime yaratır.

$$w_0, w_1, w_2, \ldots, w_{4(Nr+1)-1} \quad (5.2)$$

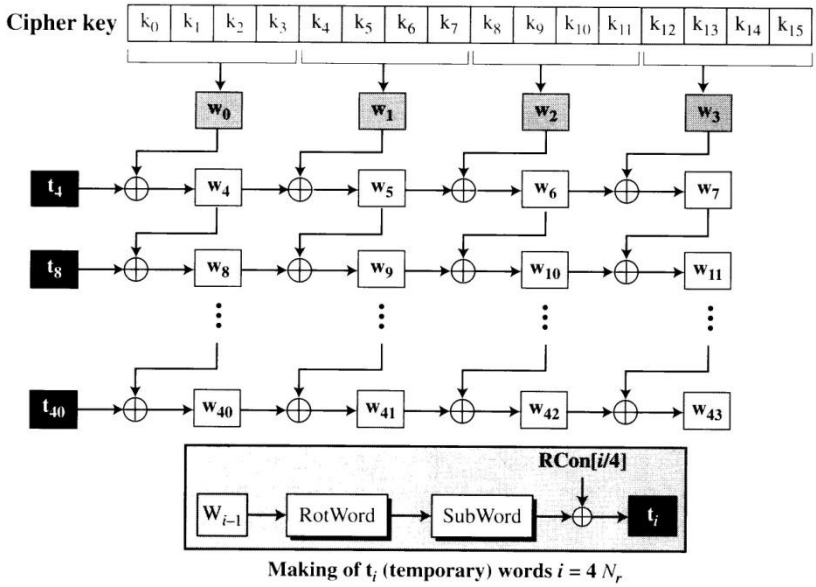
AES-128 şifresinde 44 kelime, AES-192 şifresinde 52 kelime ve AES-256 şifresinde 60 kelime vardır. Aşağıdaki tabloda hangi döngüde hangi kelimelerin kullanıldığı gösterilmiştir:

Tablo 5.4. Döngülerde Kullanılan Kelimeler

Döngü (Loop)	Kelime (Words)
Başlangıç (Pre-Round)	$w_0 w_1 w_3 w_3$
1	$w_4 w_5 w_6 w_7$
2	$w_8 w_9 w_{10} w_{11}$
…	…
N_r	$w_{4N_r} w_{4N_r+1} w_{4N_r+2} w_{4N_r+3}$

AES-128 de Anahtar Genişletme

Aşağıdaki şekil orijinal anahtardan 44 kelimenin nasıl oluşturulduğu gösterilmektedir.



Şekil 5.5. Anahtar Algoritması

Anahtar genişletme işlemi aşağıdaki gibidir:

1- İlk 4 kelime (w_0, w_1, w_2, w_3) şifre anahtarından elde edilir. Şifre anahtarı k_0 dan k_{15}'e kadar 16 byte bir dizi olarak düşünülür. İlk 4 byte (k_0 dan k_3'e) w_0, ikinci 4 byte (k_4'ten k_7'ye) $w_1$ve bu şekilde diğer kelimeler w_2 ve w_3'te şifre anahtarının kelimeler halindeartardayerleştirilmesi ile meydana getirilir.

2- Sonraki kelimeler (w_i i=4 den 43'e kadar) şöylemeydana getirilir:

   a- Eğer i (mod 4) ≠ 0 ise $w_i = w_{i-1} \oplus w_{i-4}$ şeklinde tablodan da görüldüğü gibi soldan ve üstten bir değerden elde edilir.

b- Eğer i (mod 4) = 0 ise $w_i = t \oplus w_{i-4}$ işlemiyleoluşturulur. Burada t geçici bir bellek ve SubWord ve RotWord rutinlerinin w_{i-1} üzerindeki uygulama sonucudur. t'nin elde edilme süreci bir döngü sabiti RCon ile XOR lama işlemi ile sona erer. Yani;

$$t = SubWord(RotWord(w_{i-1})) \oplus RCon_{i/4} \quad (5.3)$$

RotWord

Sadece 1. satırda bir word'ü 4 byte'lık bir dizi olarak ele alır ve her byte'ı sola öteler.

SubWord

Sadece 4 byte'a uygulanır, kelimenin her byte değerini diğer bir byte ile yerdeğiştirir.

Döngü Sabitleri (Round Constants)

Her döngü sabiti, RCon,en sağdaki 3 byte'ı 0 olan 4 byte değerinde bir değerdir. Aşağıdaki tablo AES-128 için (10 döngü) değerleri göstermektedir.

Tablo 5.6. Döngü Sabitleri

Döngü	Sabit (RCon)	Döngü	Sabit (RCon)
1	$(\mathbf{01}\ 00\ 00\ 00)_{16}$	6	$(\mathbf{20}\ 00\ 00\ 00)_{16}$
2	$(\mathbf{02}\ 00\ 00\ 00)_{16}$	7	$(\mathbf{40}\ 00\ 00\ 00)_{16}$
3	$(\mathbf{04}\ 00\ 00\ 00)_{16}$	8	$(\mathbf{80}\ 00\ 00\ 00)_{16}$
4	$(\mathbf{08}\ 00\ 00\ 00)_{16}$	9	$(\mathbf{1B}\ 00\ 00\ 00)_{16}$
5	$(\mathbf{10}\ 00\ 00\ 00)_{16}$	10	$(\mathbf{36}\ 00\ 00\ 00)_{16}$

Anahtar genişletme evresi kelime değerlerini hesaplarken ya yukarıdaki tabloyu ya da en soldaki byte'ı dinamik olarak hesaplamak için GF(2^8) cismini kullanır.

Tablo 5.7. GF(2^8) cismi

RC_1	x^{1-1}	x^0	mod prime	x^0	00000001	01_{16}
RC_2	x^{2-1}	x^1	mod prime	x^1	00000010	02_{16}
RC_3	x^{3-1}	x^2	mod prime	x^2	00000100	04_{16}
RC_4	x^{4-1}	x^3	mod prime	x^3	00001000	08_{16}
RC_5	x^{5-1}	x^4	mod prime	x^4	00010000	10_{16}
RC_6	x^{6-1}	x^5	mod prime	x^5	00100000	20_{16}
RC_7	x^{7-1}	x^6	mod prime	x^6	01000000	40_{16}
RC_8	x^{8-1}	x^7	mod prime	x^7	10000000	80_{16}
RC_9	x^{9-1}	x^8	mod prime	$x^4 + x^3 + x + 1$	00011011	$1B_{16}$
RC_{10}	x^{10-1}	x^9	mod prime	$x^5 + x^4 + x^2 + x$	00110110	36_{16}

En soldaki byte, RCi, x^{i-1} dir ve i değeri döngü sayısıdır. AES,

$x^8 + x^4 + x^3 + x + 1$ (5.4)

indirgenemez polinomunu kullanır[17]

5.1.2 Akış Şifreler

Akış şifreleme algoritmalarında açık metnin her bir karakteri ayrı ayrı şifrelenir. Belli bir kaynak değer kullanan anahtar üretme fonksiyonu, şifreleme işleminde kullanılmak üzere, mümkün olduğunca uzun – yüksek güvenlik için orijinal mesaj uzunluğunda – rastlantısal olarak gözüken anahtar dizisi üretir. 1949 yılında Shannon One – Time – Pad (Tek Kullanımlık Şerit) anahtarın rastlantısal olma ve bir defa kullanılma şartlarını sağladığında kusursuz bir güvenlik sağladığını göstermiştir. O zamandan bugüne akış şifrelerin en önemli kriteri haline gelmiştir. Örnek olarak RC4 verilebilir. [18]

Simetrik algoritmalar çok hızlı çalışmalarının yanında, anahtar dağıtım problemlerini de beraberinde taşır. Bu çalışma içinde bu probleme bir çözüm oluşturulmaya çalışılmıştır.

5.2 Bir Anahtar Dağıtım Mimarisi – Diffie-Hellman Algoritması

Kriptografi bilimi, 1976 yılında Whitfield Diffie ve Martin Hellman tarafından "New Directions in Cryptography" isimli makale ile yeni bir yaklaşımla tanıştı.[19]

Bu yaklaşım simetrik algoritmalarda kullanılan anahtarı güvenli bir şekilde paylaşma problemini ortadan kaldırıyordu. Diffie-Hellman algoritması iki kişinin ortak bir anahtar üretme prensibine dayanıyordu.

Diffie-Hellman anahtar üretme protokolü şöyledir:

Alice ve Bob birbirleriyle gizlice haberleşmek isteyen iki kişi olsun. Alice ve Bob hesaplamalar için p ve g sayılarına ihtiyaç duyarlar. p asal sayı ve g mod p ye göre primitif kök (Primitif kök mod p'de sıfır olmayan tüm sayıların üslerini yaratabilen bir g sayısıdır.) olmak durumundadır. Bu sayılar açıktır yani herkes tarafından bilinebilir. Her ikisi de sadece kendilerinin bileceği a ve b sayılarını belirler ve protokolü işletmeye başlarlar.

Tablo 5.10. Diffie-Hellman Algoritması

Alice	Bob
p,g (Açık) a(gizli)	p,g(açık) b(gizli)
A=g^a modp	B=g^bmodp
A değeri Bob'a gönderilir	B değeri Alice'e gönderilir
s=B^a modp	s = A^b modp

Bu protokolün işlemesi sırasında p,g,A ve B değerleri açıktır ve herkes tarafından bilinebilir. Alice'in kendisi için belirlediği a, Bob'un kendisi için belirlediği b ve işlemlerin sonunda hesaplanan s (Anahtar) değerleri gizlidir. Diffie-Hellman anahtar protokolü ayrık logaritma probleminin zorluğu üzerine kurulmuştur. A=g^a modp işleminde A, g, p biliniyor olmasına rağmen $\log_g A = a$ değerini hesaplamak çok zordur. Ve bu a değeri s=B^a modp eşitliğinde s anahtarını hesaplamak için kilit önemdedir.[19]

5.3 Asimetrik Algoritmalar

Şifrelemek ve şifre çözmek için farklı anahtarlar kullanılır. Şifrelemek için kullanılan anahtar açıktır ve herkes tarafından bilinebilir, şifre çözme anahtarı ise sadece alıcı tarafından bilinmelidir. Bu anahtarlar matematiksel olarak birbirine bağlıdır fakat birinden diğerini hesaplamak mümkün değildir.

Simetrik algoritmaların en büyük problemi olan anahtar dağıtım problemini ortadan kaldırır. Fakat çalışma performansları oldukça düşüktür. Çözülmesi zor veya çözümü olmayan matematiksel problemler üzerine kurulmuşlardır. Bu da çok fazla işlem yapılmasına sebebiyet verir. İşlem sayısı veya işlem yapılacak sayılar büyüdükçe performans problemleri doğmaya başlar.

Sadece şifreleme amaçlı olarak değil, aynı zamanda doğrulama amaçlı olarak da elektronik imzalarda kullanılabilir. Örnek olarak RSA algoritması verilebilir. [20]

Şekil 5.8. Asimetrik Şifreleme

5.3.1 Yardımcı Algoritmalar

5.3.1.1 Öklit Algoritması

Öklit algoritması a ve b gibi iki tamsayının en büyük ortak bölenini bulmak için kullanılır. Öklit Algoritmasını açıklamadan önce gerekli olan altyapıyı kuralım.

Tanımalamalar:

a ve b sıfırdan farklı tam sayılar olsun.

EBOB(a,b) (a ve b tam sayılarının en büyük ortak böleni) aşağıdaki özellikleri taşıyan d sayısıdır:

1. d sayısı a ve sayılarının her ikisini de böler. Yani d|a and d|b dir.
2. a ve b sayılarının her ikisini de bölen c sayısı ya d den küçüktür ya da d ye eşittir. Başka bir deyişle c|a ve c|b, dolayısıyla c≤d.

Algoritmaya geçmeden önce bir lemma açıklayalım:

Lemma:

a ve b sıfırdan farklı tamsayılar olmak üzere, q ve r a=bq+r eşitliğini sağlayan tamsayılar ise

gcd(a,b) =gcd(b,r)(5.5)

İspat:

ebob(a, b)≤ebob(b, r) ve ebob(b, r)≤ebob(a, b) olacak şekilde iki aşamada yapılacaktır.

1. ebob(a, b)≤ebob(b, r) olduğunu gösterelim
   a. a ve b nin her ikisini de bölen bir sayının b ve r nin de ortak böleni olduğunu gösterelim.

A ve b sıfırdan farklı tam sayılar ve c sayısı a ve b nin ortak bir böleni olsun. Bu durumda c|a ve c|b ve bölmenin tanımına göre a =nc ve b =mc, n ve m tamsayıları kullanılarak yazılabilir. a ve b sayılarının yerlerine değerlerini yazalım,

$$a = bq + r \quad (5.6)$$

ve

$$nc = (mc)q + r \quad (5.7)$$

olur. r sayısını yalnız bırakırsak $r = nc - (mc)q = (n - mq)c$ elde ederiz. $n - mq$ bir tam sayı olduğundan c sayısı r sayısını böler ve $c|r$ yazılabilir. c sayısı b sayısını da böldüğünden yani c|b olduğundan , c sayısının b ve r sayılarının ortak böleni olduğu sonucuna varırız.

Buna göre a ve b nin tüm ortak bölenleri b ve r nin de ortak bölenidir. a ve b sıfırdan farklı olduğundan ebob(a,b) nin var olduğu ve bu sayının b ve r nin bir ortak böleni olduğu anlamına gelir.

O halde ebob(a,b), b ve r nin ortak böleni olduğundan ebob(b,r) den küçük veya eşittir.

$$ebob(a, b) \leq ebob(b, r) \quad (5.8)$$

2. ebob(b, r)≤ebob(a, b): İspat birinci adımda olduğu gibi yapılır.

Öklit Algoritması şöyledir:

1. A ve B, $A > B \geq 0$ olacak şekilde, sıfırdan farklı tam sayılar olsun.
2. Eğer B =0 ise gcd(A, B) = A dir, değilse q bölüm r kalan olmak üzere bölme teoremine göre

$$A = Bq + r \ , \ 0 \leq r < B \quad (5.9)$$

yazılabilir.

Lemma1 e göre ebob(A, B) =ebob(B,r). Dolayısıyla problem A ve B sayılarının eboblarını bulmak yerine B ve r sayılarının eboblarını bulmaya indirgenir ve

$$B = rq+r' \ , \ 0 \leq r' < r \quad (5.10)$$

3. Bu işlemler r=0 sonucu elde edilene kadar, A yerine B ve B yerine r değerleri konarak, 2. adımdan itibaren tekrarlanır. [21]

5.3.1.2 Çin kalan Teoremi

Seçilen bir m doğal sayısına göre, a,b ∈ Z olmak üzere m|(a-b) ise a veb sayıları m moduna göre kongrüdür ve a≡b mod m şeklinde yazılır. Buna m mod una göre kongrüans bağıntısı denir ve bir denklik bağıntısıdır.

Çin kalan teoremi bazı kongrüans sistemlerini çözmekte kullanılır.

$m_1, m_2, m_3, \dots, m_r$pozitif tamsayılar ve her $i \neq j$ için ebob(m_i, m_j)=1 olsun. $a_1, a_2, a_3, \dots, a_r$tamsayılarına göre

$$\begin{gathered} x = a_1 \ mod \ m_1 \\ x = a_2 \ mod \ m_2 \\ x = a_3 \ mod \ m_3 \\ \vdots \\ x = a_r \ mod \ m_r \end{gathered} \quad (5.11)$$

Kongrüans sisteminin Z deki çözümü $M = m_1 m_2 m_3 \dots m_r$modülüne göre

$$x \equiv a_1 b_1 \frac{M}{m_1} + \cdots + a_r b_r \frac{M}{m_r} \ modM \quad (5.12)$$

ve

$$b_i \frac{M}{m_i} \equiv 1 \ mod \ m_i \quad (5.13)$$

Hesaplanır. [22]

5.3.2 Asimetrik Şifreleme Örneği RSA

RSA algoritması Ron Rivest, Adi Shamir, Leonard Adleman tarafından 1977 yılında duyurulmuştur. Asimetrik şifreleme algoritmasıdır. Çift anahtarla çalışır. Hem anahtar planlaması hem de şifreleme işlemlerini içerir. Ayrık logaritma problemi üzerine kuruludur [23].

RSA algoritmasına gücünü veren noktalardan biri de asal sayılardır. Asal sayılar hiçbir matematiksel yapıya uymaz. Hiçbir çarpanları yoktur (sadece 1 ve p). Hiçbir şekilde formülize edilemezler. Ve bir sayının asal olup olmadığını anlayabileceğimiz stabil bir test de bulunmamaktadır.

RSA algoritmasını açıklayalım.

Anahtar yönetimi;

1. Rastgele p ve q asal sayıları seçilir.
2. Bu iki sayının çarpımı ile mod alma işlemi yapılacak olan n sayısı hesaplanır.

$$n=p*q \quad (5.14)$$

3. Bu iki sayının 1 eksikliklerinin çarpımı ile $\varphi(n)$sayısı hesaplanır.

$$\varphi(n)=(p-1)*(q-1) \quad (5.15)$$

4. $\varphi(n)$sayısı açık anahtarı tespit etmede kullanılır. $1<e<\varphi(n)$ve $gcd(e, \varphi(n))=1$ olacak şekilde e sayıları tespit edilir ve içlerinden rastgele biri açık anahtar olarak seçilir ve şifreleme işlemi için kullanılır.
5. $e*d=1 \bmod \varphi(n)$olacak şekilde hesaplanan d sayısı da gizli anahtar olacaktır ve şifre çözme işlemi için kullanılır.
6. (e,n) açık anahtar değerleridir ve herkes tarafından bilinebilir, (d,n) gizli anahtar değerleridir ve sadece şifreyi çözmek isteyen kişi tarafından bilinmesi gerekir.

Şifreleme ve şifre çözme;

7. $m^e=c \bmod n$ işlemi ile m açık metni (e,n) açık anahtar çiftiyle şifrelenir ve alıcıya gönderilir.
8. $c^d=m \bmod n$ işlemi ile de c şifreli metni çözülür.

5.3.2.1 RSA ÜzerineBir Analiz

Şifreleme işleminin $m^e=c \bmod n$ işlemi ile yapıldığını söylemiştik. e açık anahtar olduğundan herkes tarafından bilinmektedir. Açık metin m ye ulaşmanın yollarından biri ters işlem uygulamaktır. Yani eşitliğin her iki tarafının e dereceden kökü alınmalıdır. Eğer e sayısı yeterince büyük seçilirse kök alma işlemi çok fazla işlem ve süre gerektirir.

Diğer bir yol d gizli anahtarına ulaşmaktır. Algoritma gereği d gizli anahtarının e açık anahtarının çarpma işlemine göre tersi olduğu bilinmektedir. Bu durumda d=1/e

mod $\varphi(n)$işlemine göre d sayısı hesaplanmaya çalışılır. Fakat bir diğer problem de $\varphi(n)$sayısının bilinmemesidir. Bu işlemin yapılabilmesi için $\varphi(n)$sayısı bilinmelidir ki bu da başka bir problem doğurur: seçilen p ve q asal sayılarının tespiti.

Yeterince büyük ve rastgele seçilmiş olan p ve q asal sayıları bilinmediğinden bu iki asal sayının tespiti için tek yol p*q=n eşitliğine göre bilinen n sayısının çarpanlara ayrılmasıdır. Çarpanlara ayırma işlemi için bilinen bir algoritma yoktur. Şu an bilinen tek yapı n sayısının karekköküne kadar olan asal sayıların denenmesidir. Eğer p ve q sayısı yeterince büyük seçilirse bu çok fazla sayının deneneceği anlamına gelir ki bu da çok fazla işlem ve süre gerektirir.

Aşağıda RSA algoritmasının çalışmasına yönelik örnekler gösterilmiştir.

Örnek: p=13 ve q=17 olacak şekilde iki asal sayı seçelim. Bu durumda n sayısı p*q=13*17=221 dir

$\varphi(n)$sayısı (p-1)*(q-1)=12*16=192 dir.

1<e<192 ve gcd(e,192)=1 olan sayılar tespit edilir.

5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49, 53, 55, 59, 61, 65, 67, 71, 73, 77, 79, 83, 85, 89, 91, 95, 97, 101, 103, 107, 109, 113, 115, 119, 121, 125, 127, 131, 133, 137, 139, 143, 145, 149, 151, 155, 157, 161, 163, 167, 169, 173, 175, 179, 181, 185, 187, 191

e=37 olsun. 37*d=1 mod 192 olacak şekilde d=109 hesaplanır.

Bu durumda şifreleme için kullanılacak açık anahtar ikilisi (37, 221), şifre çözmek için kullanılacak gizli anahtar ikilisi (109,221) olarak belirlenir.

Şifrelenecek metin “HAKAN” olsun. “HAKAN” kelimesine ait harflerin ASCII tablosundaki karşılıkları ile şifreleme işlemi yapılır. Bu durumda teker teker harflerin şifrelenmiş hali aşağıdaki gibidir.

Tablo 5.9. Şifrelenecek metne ait ASCII değerlerinin şifreli değerleri

H=72	72^37 mod 221	72
A=65	65^37 mod 221	182
K=75	75^37 mod 221	62
A=65	65^37 mod 221	182
N=78	78^37 mod 221	91

Bu durumda şifreli metin 72,182,62,182,91 olacaktır.

5.4 Özet (Hash) Algoritmaları

Özet fonksiyonları, bir mesajın, belirli uzunlukta bir sayısal değerinin – özet değeri -hesaplanması işlemini için kullanılır. Oluşan değere mesaj özeti denir. Bu özet değeri çoğunlukla orijinal mesajdan çok küçüktür. Tek yönlü fonksiyonlar kullanılır. Yani özet değerden orijinal mesajı elde etmek mümkün değildir. Orijinal mesajdaki bir bit değişikliği bile özet değeri değiştirir. Bu özelliği, özet değerin mesaja özel olmasını sağlar.

Özellikle elektronik imzalarda, belge yönetim sistemlerinde mesajın değişmediğini garanti altına almak için kullanılır. SHA – Secure Hash Algorithm örnek olarak verilebilir.[24]

5.4.1 SHA-256 Özetleme Algoritması

SHA-256 özet algoritması 512 bitlik mesaj blokları üzerinde çalışan 256 bitlik orta düzey özet algoritmasıdır. İncelenmesi gereken iki ana unsur vardır:

1. SHA-256 özet algoritması
2. SHA-256 mesaj algoritması

Bu anlatım da aşağıdaki notasyonlar kullanılacaktır:

Tablo 5.10. Kullanılan Notasyonlar

$\oplus$	Bit Düzey XOR
$\wedge$	Bit Düzey AND
$\vee$	Bit Düzey OR
$\neg$	Bit Düzey Tamamlama
$+$	Mod 2^{32} Toplama
R^n	N bit sağa öteleme
S^n	N bit sağ öteleme

Tüm bu işlemler 32 bitlik kelimeler üzerinde yapılacaktır.

Başlangıç özet değeri H(0) aşağıdaki sıradaki gibidir. Bu değerler ilk sekiz asal sayının kare köklerinin kesirli kısımlarından elde edilmiştir.

Tablo 5.11. H^0 ın 32 bitlik 8 parçadan oluşan başlangıç değerleri

$H_1^{(0)} = 6a09e667$
$H_2^{(0)} = bb67ae85$
$H_3^{(0)} = 3c6ef372$
$H_4^{(0)} = a54ff53a$
$H_5^{(0)} = 510e527f$
$H_6^{(0)} = 9b05688c$
$H_7^{(0)} = 1f83d9ab$
$H_8^{(0)} = 5be0cd19$

5.4.1.1 Ön hazırlık

Öncelikle özet değeri alınacak mesaj üzerinde hazırlık yapılır:

1. Mesaj genişletilir. Mesaj 512 bitlik bloklar halinde işleneceğinden, 512 bit ve katlarına uygun bir uzunlukta olmalıdır. Eğer bu uzunluktan kısa ise mesajın sonuna 1 ve ardından yeteri kadar 0 eklenerek gerekli uzunluğa gelmesi sağlanır. Mesaj uzunluğu l ve k denklem çözümü için en küçük negatif olmayan değer olmak üzere l+1+k=448 mod 512 olmalıdır. $l$uzunluğu bit olaraksona eklenir. Örneğin mesaj "abc" olsun. Uzunluğu 8*3=24 bittir.(ASCII tablosunda karşılık) sonuna 1 eklendiği düşünülürse 448-(24+1)=423 adet sıfır ve ardından "abc" metninin uzunluğunun bit karşılığı eklenmelidir

01100001	01100010	01100011	1	00…0	00…011000
				423 adet	64 adet
				(basamak)	(basamak)

2. Mesaj N adet 512 bitlik bloklara ayrılır. M(1), M(2), … ,M(N). Bu 512 bitlik bloklar da 32 bitlik bloklar halinde 16 kelimeye bölünerek M blokları doldurulur. Yani ilk 32 bitlik blok M0(i), sonraki M1(i), ve en son M15(i) olacak şekilde kelimeler oluşturulur. Her 32 bitlik kelimede, Big-Endian yaklaşımı kullanılacağından (en önemli byte'in solda olduğu sıralamaya **big-endian** denir) en solda en önemli bit bulunacaktır.

5.4.1.2 Ana Döngü

Özet hesaplama süreci şöyledir:

For i = 1 to N (N = Genişletilmiş mesaj bloklarının sayısı)

{

a,b,c,d,e,f,g,h değişkenlerine (registers) (i-1) numaralı özet değerleri ile doldur (değişkenlerin değerleri = (i = 1) olduğunda başlangıç özet değerleri)

$$a \leftarrow H_1^{i-1}$$

$$b \leftarrow H_2^{i-1}$$

.

.

.

$$h \leftarrow H_8^{i-1} (5.16)$$

SHA-256 fonksiyonun sıkıştırma algoritması a,b,c,…,h değişkenlerini güncellemek için uygulanır.

For j=0 to 63

{

Ch(e,f,g), Maj(a,b,c), $\sum_0(a)$, $\sum_1(e)$ ve W_j fonksiyonu hesaplanır. Bu fonksiyonlar şöyledir:

$T_1 \leftarrow h + \Sigma_1(e) + Ch(e,f,g) + K_j + W_j$ (5.17)

$T_2 \leftarrow \Sigma_0(a) + Maj(a,b,c)$ (5.18)

$h \leftarrow g$ (5.19)

$g \leftarrow f$ (5.20)

$f \leftarrow e$ (5.21)

$e \leftarrow d + T_1$ (5.22)

$d \leftarrow c$ (5.23)

$c \leftarrow b$ (5.24)

$b \leftarrow a$ (5.25)

$a \leftarrow T_1 + T_2$ (5.26)

}

i numaralı ara değer H^i ara özet değeri hesaplanır.

$H_1^i \leftarrow a + H_1^{i-1}$

$H_2^i \leftarrow b + H_2^{i-1}$

.

.

.

$H_8^i \leftarrow h + H_8^{i-1}$ (5.27)

}

$H^{(N)} = H_1^{(N)}, H_2^{(N)}, \dots, H_8^{(N)}$değeri M mesajının özet (Hash) değeridir.[25]

5.4.1.3 Tanımlamalar

SHA-256 altı adet mantıksal fonksiyon kullanır. Bu fonksiyonların tamamı 32 bitlik kelimelerle (blok) ile çalışır ve 32 bitlik çıktı üretir. Fonksiyonlar aşağıda açıklanmıştır:

$Ch(x, y, z) = (x\Lambda y) \oplus (\neg x\Lambda z)$ (5.28)

$Maj(x, y, z) = (x\Lambda y) \oplus (x\Lambda z) \oplus (y\Lambda z)$ (5.29)

$\Sigma_0(x) = S^2(x) \oplus S^{13}(x) \oplus S^{22}(x)$ (5.30)

$\Sigma_1(x) = S^1(x) \oplus S^{11}(x) \oplus S^{25}(x)$ (5.31)

$\sigma_0(x) = S^7(x) \oplus S^{18}(x) \oplus R^3(x)$ (5.32)

$\sigma_1(x) = S^{17}(x) \oplus S^{19}(x) \oplus R^{10}(x)$ (5.33)

Genişletilmiş mesaj blokları W_1,W_2,…,W_{63} SHA-256 mesaj planlama algoritması kullanılarak aşağıdaki gibi hesaplanır:

$W_j = M_j^{(i)}$ for j=0,1,2,…,15 ve

For j=16,…,63

{

$W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}$ (5.34)

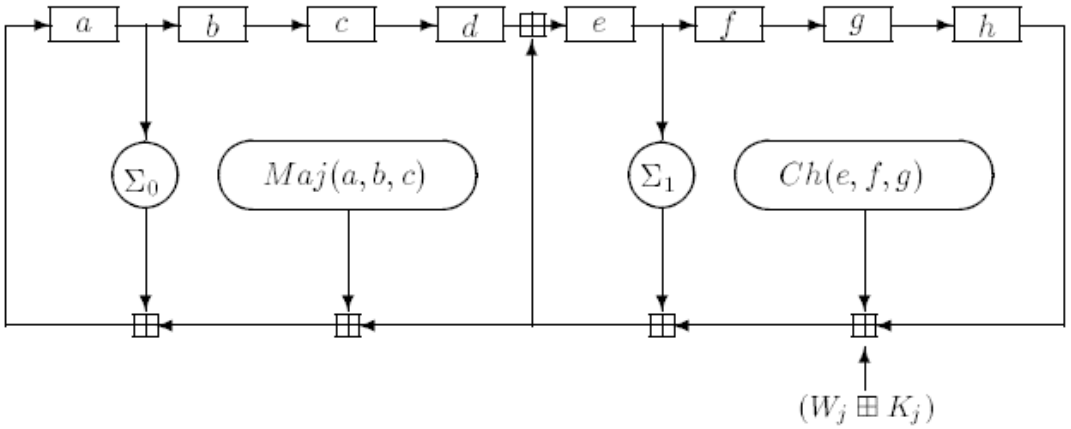
}

SHA-256 da kullanılan sabit $K_0, \dots, K_{63}$ kelimeleri aşağıda verilmiştir. Bu değerler ,ilk 64 asal sayının küp kökünün ilk otuz iki bitidir.

Tablo 5.12. ilk 64 asal sayının küp köklerinin ilk 32 bitlik ondalık kısımları

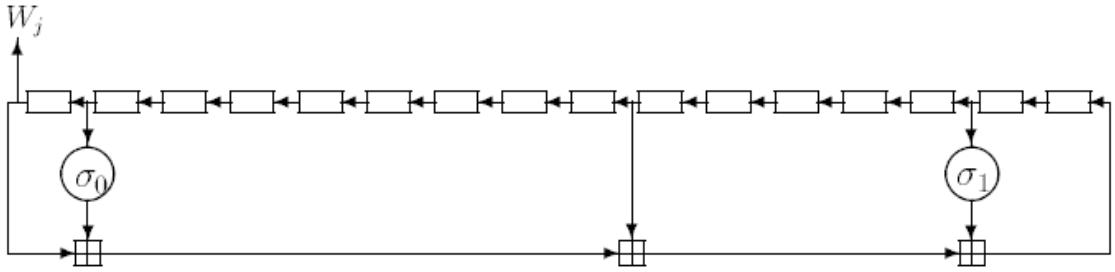
428a2f98	71374491	b5c0fbcf	e9b5dba5	3956c25b	59f111f1	923f82a4	ab1c5ed5
d807aa980	12835b01	243185be	550c7dc3	72be5d74	80deb1fe	9bdc06a7	c19bf174
e49b69c1	efbe4786	0fc19dc6	240ca1cc	2de92c6f	4a7484aa	5cb0a9dc	76f988da
983e5152	a831c66d	b00327c8	bf597fc7	c6e00bf3	d5a79147	06ca6351	14292967
27b70a85	2e1b2138	4d2c6dfc	53380d13	650a7354	766a0abb	81c2c92e	92722c85
a2bfe8a1	a81a664b	c24b8b70	c76c51a3	d192e819	d6990624	f40e3585	106aa070
19a4c116	1e376c08	2748774c	34b0bcb5	391c0cb3	4ed8aa4a	5b9cca4f	682e6ff3
748f82ee	78a5636f	84c87814	8cc70208	90befffa	a4506ceb	bef9a3f7	c67178f2

5.4.1.4 Diyagramlar



Şekil 5.13. SHA-256 Sıkıştırma Fonksiyonu Diyagramı

Şekildeki $\boxplus$ sembolü mod 2^{32} deki toplama işlemini ifade etmektedir.



Şekil 5.14. SHA-256 Mesaj planlaması

Değişkenler (registers) $W_0, \ldots, W_{15}$ ile doldurulmuştur.

5.5 Karıştırma Algoritmaları

Karıştırma Algoritmaları veri bloklarını karıştırma amacıyla kullanılır. Veri blokları sayı dizisi olarak düşünülürse algoritmaların yaptığı iş; bu sayı dizisine ait elemanların kendilerine dokunmadan yerlerini değiştirmek şeklindedir. Bu işlemi rastgele (random) sayı üreterek yaparlar. Fisher/Yates algoritması ve KNUTT/DURNSTENFELD algoritması örnek olarak verilebilir [26,27].

Karıştırma algoritmaları eğlence sektöründe sıklıkla karşımıza çıkmaktadır. Örnek olarak kart oyunları verilebilir. Rastgele sayılar üreterek verilerin yeri değiştiğinden dolayı karıştırılmış veri bloğunun geri döndürülmesi mümkün değildir. Bazı eklemelerle orijinal veri bloğunu geri getirmek mümkün olur. Karıştırma algoritmaları sadece karıştırma işlemi için değil veri şifreleme amaçlı olarak da kullanılmıştır[26,27]

"Knutt / Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması" isimli çalışmada resim pikselleri diziye dönüştürülmüş, dizi üzerinde karıştırma işlemi uygulanmıştır. Bununla birlikte başka bir dizide resmin orijinal sıradaki piksel değerleri ile yeni yeri bir eşleşme dizisinde tutulmuştur. Şifrelenmiş resmin geri dönüşümü işlemi de eşleşme dizisi kullanılarak piksellerin eski yerlerine gelmesi sağlanarak başarılmıştır. [26]

Benzer bir çalışma olan "File Encryption using Fisher-Yates Shuffle" isimli çalışmada dosya şifreleme işlemi gerçekleştirilmiştir. [27]

Karıştırma algoritmaları yapıları gereği geri dönüşü olmayan algoritmalardır. Yani rastgele üretilen değerler ile yerleri değiştirilen değerler daha sonra eski yerlerine dönemezler. Karıştırma işleminde başarı sağlansa da bir problemin doğmasına neden olur: Eğer karıştırma işleminden geçmiş bir veri bütünü (Dosya, resim, ses, video vb.) tekrar eski haline getirilmek istenirse ne yapılacaktır? Eğer geri dönüş istenirse, yerleri değiştirilen değerlerin eski yerleri de kayıt altında tutulmalıdır. Bu da oluşturulan sisteme ek işlem yükü, ek depolama alanı ve veri transferinde ek trafik anlamına gelir.

5.5.1 FISHER/YATES Algoritması

Ronald Fisher ve Frank Yates tarafından 1938 yılında yayınlanmıştır[26]. Sonlu kümelerde elemanların yerlerini değiştirerek karıştırmak için kullanılır. N elemanlı sonlu bir dizi için algoritma şu şekilde çalışır:

1. Dizinin elemanlarını 1 den N e kadar sırala
2. 1≤k≤N olacak şekilde rastgele k sayısı seçilir.
3. k numaralı eleman dizinin en sonundan itibaren öne doğru dizilir.
4. 2 ve 3 numaralı adımlar dizinin yeri değiştirilmeyen elemanı kalmayana dek devam eder.

Fisher/Yates Karıştırma Algoritmasının Uygulamasının bir örneği Tablo 5.16 da gösterilmiştir.[26,29]

Tablo 5.15. Fisher/Yates Karıştırma Algoritmasının Uygulaması

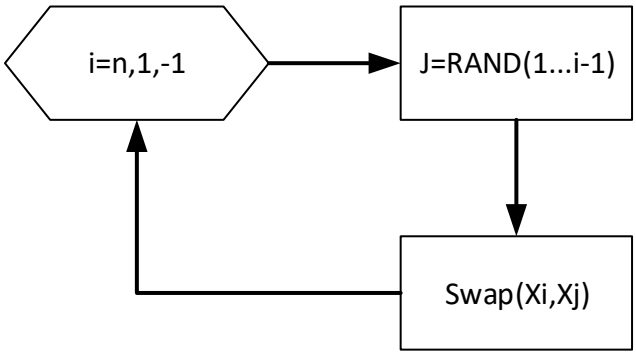
Orijinal Dizi	K numaralı elemanın Yeri – Elemanın Kendisi	Yeni Dizi
1-2-3-4-5-6-7-8-9	3 – 3	
1-2-4-5-6-7-8-9	5 – 6	3
1-2-4-5-7-8-9	4 – 5	6-3
1-2-4-7-8-9	2 – 2	5-6-3
1-4-7-8-9	5 – 9	2-5-6-3
1-4-7-8	1 – 1	9-2-5-6-3
4-7-8	3 – 8	1-9-2-5-6-3
4-7	1 – 4	8-1-9-2-5-6-3
7	0 – 7	4-8-1-9-2-5-6-3
		7-4-8-1-9-2-5-6-3

5.5.2 **KNUTT/DURNSTENFELD Algoritması**

1964 yılında Richard Durnstenfeld tarafından yayınlanmıştır. FİSHER/YATES algoritmasının düzenlenmiş halidir [15]. KNUTT/DURNSTENFELD Shuffle Algorithm (K/DSA) bir dizinin elemanlarının kendi içinde yer değiştirilmesi ile karıştırılması işlemini gerçekleştirir. K/DSA şöyledir:

Bir A dizisi n elemanlı olsun.

1. 1≤k≤n olacak şekilde rastgele bir k numarası seçilir
2. Dizinin k numaralı elemanı ile n numaralı elemanı yer değiştirilir. ($A_k \leftrightarrow A_n$)
3. 1. Adım 1≤k≤n-1 olacak şekilde tekrarlanır, 2 numaralı adım ($A_k \leftrightarrow A_{n-1}$) olacak şekilde tekrarlanır.
4. 1 ve 2 numaralı adımlar 3. Numaralı adımda olduğu gibi sürekli en son işleme sokulan elemandan bir önceki eleman işleme sokularak tekrarlanır.
5. 1 ve 2 numaralı işlemler 3 ve 4 de olduğu gibi $A_1 \leftrightarrow A_2$ olana kadar devam eder.



Şekil 5.16. KNUTT/DURNSTENFELD Algoritmasının akış şeması

K/DSA kullanılarak karıştırılmış bir dizinin karıştırılma işlemine ait bir kayıt tutulmadığı için geri dönüşü olmayan tek yönlü bir algoritmadır. Seçilen k değerleri rastgele olduğundan ve algoritmanın doğası gereği, algoritma her uygulandığında farklı bir sonuç elde edilecektir.

5.5.3 Karıştırma Amaçlı Doğrusal Üreteç Tasarımı

Geri dönüşüm gerektiren işlemlerde karıştırma algoritmaları yerine doğrusal üreteçler (Lineer kongrüans üreteçler - LKÜ) tercih edilebilir. Bu üreteçler aşağıdaki gibi tanımlanır:

(m > 0) bir doğal sayı olmak

üzere, $X_i \in$ {1,2,..., m-1} baslangıç değerini seçip,

$X_{i+1} = aX_i + cmodm$ algoritmasına göre X_1, X_2, ...sayılarını ve bu sayılar yardımıyla,

$u_1 = X_1/m$, $u_2 = X_2/m$, $u_3 = X_3/m \in$ (0,1) sayılarını üretmektedir.

Böyle bir fonksiyon eğer a ve c katsayıları biliniyorsa geri dönüşümü olan bir karıştırma işlemi sağlayabilir. Eğer geri dönüşüm olması isteniyorsa a ve c katsayıları, fonksiyon mod m e göre bijektif olacak şekilde seçilmelidir. Bu şekilde tersi alınabilir bir fonksiyon olur ki geri dönüş sağlanabilir.

5.5.4 Asal Katsayılarla Düzenlenmiş Doğrusal Üreteç - AKDÜ

Karıştırma işlemi:

A[n]=$\{a_1, a_2, a_3, \ldots, a_n\}$ ve 1 ≤ x ≤ n olacak şekilde A dizisinin her x numaralı elemanını başka bir numara ile değiştiren algoritma şöyledir:

1. 2 ≤ p ≤ n ve 2 ≤ q ≤ n olacak şekilde p,q asal sayıları seçilir.

2. Sh:[1,n] → [1,n]
   Sh(x)=px+q karıştırma fonksiyonu oluşturulur.

3. ($T[x] = A[Sh(x)]$)$_{x=1}^{n}$ A dizisinin, karıştırma fonksiyonu ile belirlenmiş sıradaki elemanı, T dizisine sırayla yerleştirilir.

4. ($A[x] = T[x]$)$_{x=1}^{n}$ Geçici T dizisinin elemanları A dizisine aktarılarak A dizisinin elemanları yer değiştirilmiş olur.

Buradaki asıl soru Sh:[1,n] → [1,n] , Sh(x)=px+q fonksiyonunun 1 → 1 bir fonksiyon olup olmadığıdır. Yani farklı indislerdeki elemanları aynı indise atayıp atamayacağıdır.

İspat: $x_1 \neq x_2$ olmak üzere iki farklı indis numaralarına sahip eleman seçelim.

px_1+ q=px_2 + q mod n (5.35)

olsun.

px_1+ q=px_2 + q + kn (5.36)

px_1=px_2 + kn (5.37)

px_1 − px_2 = kn (5.38)

p(x_1 − x_2) = kn (5.39)

bulunur.

i. k=0 ise p ≠ 0 olduğundan $x_1 - x_2 = 0$ olacağından $x_1 = x_2$ bulunur ki bu baştaki kabulümüze aykırıdır.

ii. $x_1 - x_2 = \frac{nk}{p}$ise ebob(p,n)=1 olduğundan p∤n dir. Yani p, n yi bölmez. p|k olduğunu varsayalım: $\frac{k}{p} = k'$ olsun. Bu durumda $x_1 - x_2 = nk' \Rightarrow x_1 = nk' + x_2$ bulunur. Bu sonuç $x_1 > n$ demektir ki bu durum $x_1 \leq n$ olma durumu ile çelişir.

Dolayısıyla Sh(x) = px+q fonksiyonu mod n e göre 1 → 1 bir fonksiyondur. Yani her $x_1 \neq x_2$ için $Sh(x_1) \neq Sh(x_2)$ olur.

Geri Dönüşüm İşlemi:

Karıştırma işlemi sonucunda elde edilen verinin eski haline gelebilmesi için Sh(x)=px+q fonksiyonunun tersi ile işlem yapılmalıdır.

Sh(x)=px+q mod n fonksiyonun tersi $Sh' = \frac{x-q}{p}$ şeklindedir. Bu doğrusal fonksiyon bir LKÜ olduğundan p ve q değerlerinin mod n e göre tersleri alınarak işlem

yapılabilir. $p'veq', pveq$değerlerinin mod n e göre tersi olmak üzere $p' = \frac{1}{p} modn$ ve $q' = -qmodn$ değerleri hesaplanırsa $Sh' = (x + q')p'modn$ halini alır.

Karıştırma işlemi için, dizinin her bir elemanı $Sh(x) = px + qmodn$ fonksiyonunda işleme sokulup yeni yeri belirlenir. Karışık verinin geri dönüşümü için ise $Sh'(x) = (x + q')p'modn$ fonksiyonu kullanılmalıdır. Karışık haldeki verinin tüm elemanları sırasıyla $Sh'(x)$ fonksiyonunda işleme sokulursa orijinal dizi elde edilmiş olur.

Şifreleme işlemlerinin çeşitli aşamalarında kullanılabilecek olan AKDÜ'nün çalışabilmesi için sadece p,q ve n yeterlidir. Yer değiştirme ve geri dönüşüm işlemleri için fazladan dizinin oluşturulmasına, saklanmasına, iletilmesine gerek yoktur. Bu durum özellikle cihazlar arasında iletişim sırasında, veri boyutunu son derece azaltacağından daha hızlı bir iletişim söz konusu olacaktır.

6. ÇOK BÜYÜK SAYILARLA İŞLEM YAPMA

Asimetrik algoritmalardan en çok tercih edileni olan RSA'nın güvenliğini sağlamak için çok büyük asal sayılarla çalışılmalıdır. RSA Security firması 1024 bitlik RSA anahtarının 2006 – 2010 yılları arasında kırılabileceğini bildirmiş, Ayrıca 2048 bitlik anahtarla yapılan şifrelemelerin 2030 a kadar kırılamayacağını, eğer 2030 sonrasına uzanan bir güvenlik isteniyorsa 3072 bitlik anahtar kullanılması gerektiğini açıklamıştır. [30]

Bu açıklama seçilen sayıların 500 – 1000 basamaklı sayılar olması gerektiği anlamına gelir. Bu büyüklükteki sayılarla çalışmak iki problemi doğurur:

1. Şifreleme işlemi sırasında sayıların bilgisayarın hafızasında depolanabilmesi
2. İşlem hızı

6.1 Modüler Üs Alma Algoritmaları

Asimetrik algoritmalardan RSA, şifreleme ve şifre çözme işlemleri sırasında, açık veya gizli metnin,açık veya gizli anahtara (e,d) göre üs değerinin moduna göre hesaplayarak işlem yapar. Anahtarların ve mod değerinin (e,d,n) çok büyük seçilmesi durumunda ciddi anlamda işlem yükü getirir ve bu işlemler çok uzun sürer. RSA nın güvenliğinden taviz vermemek adına büyük sayılarla çalışılması da kaçınılmazdır.

İşlem süresini kısaltmak için modüler üs alma algoritmaları kullanılabilir. Modüler üs alma işlemi, üssün bit karşılığı bulunup bu bitlerdeki 0 ve 1 değerlerine göre farklı işlemler yaparak sonuca ulaşılması şeklindedir.

Örnek olarak Binary Metot (İkili Üs Alma) algoritması şöyledir:

x^ndeğeri şöyle hesaplanır

1. Başlangıç değerleri $x_0 = x$, $y_{-1} = 1$, $n_0 = n$ olarak alınır
2. x_i, y_iven_i değerleri tekrarlı olarak şöyle hesaplanır:

   $n = \frac{n_i - 1}{2}$ (6.1)

   $x_i = {x_{i-1}}^2$ (6.2)

   $y_i = \begin{cases} x_i y_{i-1}, n_i \, çift \; ise \\ y_{i-1}, n_i tek \; ise \end{cases}$ (6.3)
3. $n_i = 0$ise işlemi durdur.
4. $y_k = x^n$ dir.

Örnek: x^{10} değerini hesaplayalım.

i	n_i	x_i	r_i	y_i
0	10	x	0	1
1	5	x^2	1	x^2
2	2	x^4	0	x^2
3	1	x^8	1	x^{10}

Şekil 6.1. x^{10} değerinin hesaplama tablosu

Ayrıca Bir Seferde 2 Bit (Quaternary Method) ve Bir Seferde 3 Bit (The Octal Method) metotları da benzer şekilde üs alma işlemlerini hızlandıran metotlardır. Bu metotlar Bir Seferde 1 Bit (Binary Method) dan daha hızlıdır. Farkları işlem yapılacak bitlerin gruplandırarak daha hızlı çözüme ulaşılmasını sağlamaktır. Bu metotların arasında en performanslısı Bir Seferde 3 Bit metodudur.[31]

6.2 Biginteger Sınıfı

Bilgisayarlarda sayılarla işlem yaparken değişken tanımlanarak, bilgisayarın hafızasında yer ayrılması ve sayıların bu alanlarda depolanması ve gerektiğinde bu alanlardan çağrılarak işlem yapılması sağlanır. Fakat sayıların depolanabilmesi için kullanılan veri tiplerinin hiçbiri bu büyüklükte sayıları depolayabilmek için uygun değildir.

Bu sorun BigInteger sınıfı ile çözülmüştür. Nesne tabanlı programlama dilleri tarafından desteklenen bu sınıftan nesne türetilerek hafızaya yerleştirilir. Bu nesne içinde kullanılan çok büyük asal sayımız bloklar halinde parçalanarak bir diziye atılır ve bu dizi içinde işlem yapılması sağlanır. Parçalanarak diziye atılan sayıların depolanması çok daha kolaydır. Peki, bu şekildeki sayılarla nasıl hızlı bir çarpma işlemi gerçekleştirilecektir? Bu sorunun cevabı da Toom Cook Algoritmasında yatmaktadır.

6.3 Toom Cook Algoritması

Toom Cook algoritması büyük sayılarda hızlı çarpma işlemi yapmayı sağlayan bir algoritmadır. Sayılar parçalanarak her bir parça, bir polinomun katsayıları gibi

düşünülür ve matrisler üzerinden işlem yapılır. Toom Cook Algoritması bir örnekle açıklanacaktır.

Algoritma tamsayıyı birden fazla parçaya ayırır. Genel olarak Toom-K algoritması eşit uzunluktaki k parça oluşturur ve sonuç polinomu 2k-1 parçalı (katsayılı) olur.

Algoritma 5 ana adımdan oluşmaktadır:

1. Parçalama - Splitting
2. Değerlendirme - Evaluation
3. Noktasal çarpma - Pointwise multiplication
4. İnterpolasyon - Interpolation
5. Birleştirme - Recomposition

Belli bir tamsayı b tabanında dizi olarak temsil edilir. Bu örnekte b=10000 olarak alınacaktır.

m = 34 1278 5609 3412 7856 0934

n = 8 3547 1260 9835 4712 6098

6.3.1 Parçalama - Splitting

İlk adım B = bi tabanını belirlemektir. örneğin m and n sayılarının basamak sayısı B tabanında en fazla k olur (Örneğin Toom-3 de k=3 dür.). i aşağıdaki gibi belirlenebilir:

$$i = \max\left\{\left[\frac{[\log_b m]}{k}\right], \left[\frac{[\log_b n]}{k}\right]\right\} + 1 \quad (6.4)$$

Örneğimizde B = b2 = 108 alınacaktır. Dolayısıyla m ve n B tabanına göre mi, ni parçalarına ayrılır.

$m_2 = 341278$

$m_1 = 56093412$

$m_0 = 78560934$

$n_2 = 83547$

$n_1 = 12609835$

$n_0 = 47126098$

Daha sonra bu elemanları k-1. dereceden p(B)=m ve q(B)=n polinomlarının katsayıları olarak kullanacağız.

$$p(x) = m_2x^2 + m_1x + m_0 = 341278x^2 + 56093412x + 78560934$$

$$q(x) = n_2x^2 + n_1x + n_0 = 83547x^2 + 12609835x + 47126098$$

Bu iki polinomun çarpımı r(x) = p(x)q(x), olacağından r(B) = m×n olur.

Basamak sayısı farklı olan sayılarda, m ve n tamsayıları için farklı k değerleri kullanılması yararlı olur. (km ve kn) Örneğin Toom-2.5 de km = 3 ve kn = 2 dir. Bu durumda B = bi deki i değeri aşağıdaki gibi seçilir:

$$i = \max\left\{\left[\frac{[\log_b m]}{k_m}\right], \left[\frac{[\log_b n]}{k_n}\right]\right\} + 1 \quad (6.5)$$

6.3.2 Değer Biçme – Evaluation

Toom Cook, p ve q polinomlarının çarpımı ile elde edilen r nin katsayılarını hesaplamak fikrine dayanır.

Derecesi d olan bir polinom d+1 nokta ile tanımlanır(Örneğin derecesi 1 olan polinom iki nokta ile tanımlanır.). deg(pq) = deg(p) + deg(q), olduğundan deg(p) + deg(q) + 1 = km + kn – 1 ile r polinomunun derecesi hesaplanabilir. Örneğimizde $k_m = k_n = 3$ olduğundan deg(p) + deg(q) + 1 = 3 + 3 – 1, deg(p) + deg(q) =4 bulunur. Oluşan polinomlar için rastgele değerler seçilerek (örneğin 0,1,-1,-2 ve en büyük dereceli terimin katsayısı için ∞ değeri gibi) r polinomunun katsayıları hesaplanmaya çalışılır.

$p(0) = m_0 + m_1(0) + m_2(0)^2 = m_0$

$p(1) = m_0 + m_1(1) + m_2(1)^2 = m_0 + m_1 + m_2$

$p(-1) = m_0 + m_1(-1) + m_2(-1)^2 = m_0 - m_1 + m_2$

$p(-2) = m_0 + m_1(-2) + m_2(-2)^2 = m_0 - 2m_1 + 4m_2$

$p(\infty) = m_2$

p ve q polinomları için aşağıdaki değerler elde edilir.

$p(0)$	$= m_0$	= 78560934	= 78560934
$p(1)$	$= m_0 + m_1 + m_2$	= 78560934 + 56093412 + 341278	= 134995624

$p(-1) = m_0 - m_1 + m_2$ = 78560934 -56093412 + 341278 = 22808800

$p(-2) = m_0 - 2m_1 + 4m_2$ = 78560934 − 2×56093412 + 4×341278 = −322260778

$p(\infty) = m_2$ = 341278 = 341278

$q(0) = n_0$ = 47126098 = 47126098

$q(1) = n_0 + n_1 + n_2$ = 47126098+ 12609835+ 83547 = 59819480

$q(-1) = n_0 - n_1 + n_2$ = 47126098− 12609835+ 83547 = 34599810

$q(-2) = n_0 - 2n_1 + 4n_2$ = 47126098− 2×12609835+ 4×83547 = 22240616

$q(\infty) = n_2$ = 83547 = 83547

Aşağıda polinom kuvvetleri ve katsayılar iki ayrı matris halinde gösterilmiştir:

$$\begin{pmatrix} p(0) \\ p(1) \\ p(-1) \\ p(-2) \\ p(\infty) \end{pmatrix} = \begin{pmatrix} 0^0 & 0^1 & 0^2 \\ 1^0 & 1^1 & 0^2 \\ (-1)^0 & (-1)^1 & (-1)^2 \\ (-2)^0 & (-2)^1 & (-2)^2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ m_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & -2 & 4 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} m_0 \\ m_1 \\ m_2 \end{pmatrix}$$

Matrisin derecesi d, p polinomu için km ve q polinomu için kn dir. p(∞) ve q(∞) için son sütun her zaman 1 olacaktır.

6.3.3 Noktasal Çarpma - Pointwise multiplication

p ve q polinomlarının çarpılması işlemi özel bir a değeri için p(a) * q(a) işleminin sonucunu da içerir. Bu noktada yukarıda belirlenmiş özel değerler için r sonuç polinomun o değerlerdeki karşılıkları bulunabilir.

$r(0) = p(0)q(0)$ = 78560934× 47126098 = 3702270274655532

$r(1) = p(1)q(1)$ = 134995624× 59819480 = 8075368029955520

$r(-1) = p(-1)q(-1)$ = 22808800× 34599810 = 789180146328000

$r(-2) = p(-2)q(-2)$ = −322260778× 22240616 = -717499575359248

$r(\infty) = p(\infty)q(\infty)$ = 341278×83547 = 28512753066.

6.3.4 İnterpolasyon - Interpolation

En karmaşık adımdır. Değerlendirme adımının tersidir. Bulunan değerler için r matrisi oluşturulur, ve r polinomu elde edilir.

$$\begin{pmatrix} r(0) \\ r(1) \\ r(-1) \\ r(-2) \\ r(\infty) \end{pmatrix} = \begin{pmatrix} 1^0 & 0^1 & 0^2 & 0^3 & 0^4 \\ 1^0 & 1^1 & 0^2 & 1^3 & 1^4 \\ (-1)^0 & (-1)^1 & (-1)^2 & (-1)^3 & (-1)^4 \\ (-2)^0 & (-2)^1 & (-2)^2 & (-2)^3 & (-2)^4 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & -2 & 4 & -8 & 16 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix}$$

Yukarıdaki matris Gauss metodu ile çözülerek r0,r1,r2,r3,r4 değerleri hesaplanabilir fakat çok uzun sürer. Bunun yerine eşitliğin her iki tarafı katsayılar matrisine bölünerek (yada katsayılar matrisinin tersi ile çarpılarak) noktasal çarpım dan elde edilen değerler kullanılır ve istenen katsayılar bulunur.

$$\begin{pmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \\ r_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & -2 & 4 & -8 & 16 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r(0) \\ r(1) \\ r(-1) \\ r(-2) \\ r(\infty) \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1/2 & 1/3 & -1 & 1/6 & -2 \\ -1 & 1/2 & 1/2 & 0 & -1 \\ -1/2 & 1/6 & 1/2 & -1/6 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r(0) \\ r(1) \\ r(-1) \\ r(-2) \\ r(\infty) \end{pmatrix}$$

Matris kesirler içermesine rağmen sonuçlar tamsayı çıkar.

r polinomu aşağıdaki gibi bulunur.

$$r(x) = 3702270274655532 + 3634104046252266\text{x} + 729975300733162x^2 + 8989895561494x^3 + 28512753066x^4$$

6.3.5 Birleştirme - Recomposition

Son olarak r(B) değerini (çarpma işlemin sonucunu) bulmak kısmı kaldı. Örneğimizde B = b2 = 108 olduğundan, r polinomunun tüm katsayıları, 8 basamak kaydırılarak, alt alta yazılarak toplanır.

3702 2702 7465 5532

3634 1040 4625 2266

729 9753 0073 3162

8 9898 9556 1494

\+ 285 1275 3066

285 1284 2965 0286 1247 3707 4202 8327 4968 7465 5532

Bulunan sonuç m=3412785609341278560934ve n=8354712609835471260 98sayılarının çarpımıdır.[31,32,33,34,35]

7. E-İMZA

İnternet ortamında veri güvenliğin gereklerinden biri de verinin aslının tespit edilmesidir. İletişim sırasında veya sonrasında üçüncü şahıslar orijinal veriye ulaşabilir ve veri de değişiklikler yapabilir. Bu durum iletişim güvenliği için çok büyük problem teşkil eder.

Gönderilen verinin alıcıya ulaştığı anda herhangi bir değişikliğe uğramadığının garanti edilmesi önem arz eder. Verinin göndericisinin değişip değişmediği, içeriğinde herhangi bir değişiklik olup olmadığı kontrol edilmelidir. İşte bu parametrelerin değişmediği gönderilen verinin orijinal olduğu e-imza ile garanti altına alınabilir.

Elektronik imza kullanılan yerlerde veride kimseye fark ettirmeden işlem yapamazsınız. Orijinal veri de yapılan en küçük değişiklik bile fark edilir.

Elektronik imza bazı parametrelerin bir araya gelmesiyle oluşturulur. Bu parametreler e-imza sahibinin ayırt edici özellikleri ve veriye ait bilgilerdir. Bu özellikler elektronik sertifikalar içinde tutulur. Elektronik sertifikalar devlet tarafından verilen kimlik belgeleri gibidir. Sertifikalar gerekli yazılımlar kullanılarak verilere eklenir ve veri imzalanmış olur.

Başka bir tanımla e-imza, bir dokümandan özetleme fonksiyonları ile oluşturulan özet değerinin, göndericinin gizli anahtarı kullanılarak şifrelenmesi ile oluşan veri parçasıdır. Yani e-imza hem göndericinin hem de metnin kendine has verilerini taşır. Bu veriler kullanılarak kimlik tespiti yapılabileceği gibi mesaj bütünlüğünün korunması adına da oldukça etkili ve güvenli bir yoldur.

7.1 Güvenli Elektronik İmza

T.B.M.M. Genel Kurulunda 15.01.2004 tarihinde görüşülerek kabul edilen 5070 sayılı Elektronik İmza Kanunu 23.01.2004 tarihli 25355 sayılı resmi gazetede yayınlanarak yürürlüğe girmiştir. Telekomünikasyon kurumu tarafından yönetmeliklerin hazırlanması 6 ay sürmüş ve Resmi Gazete'de yayınlanarak yürürlüğe girmiştir [36].

Yasada, güvenli elektronik imza tanımı şöyledir;

a) Münhasıran imza sahibine bağlı olan

b) Sadece imza sahibinin tasarrufunda bulunan güvenli elektronik imza oluşturma aracı ile oluşturulan

c) Nitelikli elektronik sertifikaya dayanarak imza sahibinin kimliğini tespitini sağlayan

d) İmzalanmış elektronik veride sonradan herhangi bir değişiklik yapılıp yapılmadığını sağlayan elektronik imza.

Kanuna göre resmi şekle veya özel törenle gerçekleştirilen hukuki işlemler ve teminat sözleşmeleri güvenli elektronik imza kullanım alanının dışında tutulmuştur. Yine yasaya göre sadece gerçek kişiler güvenli elektronik imza sahibi olabilir. Yasanın gerekçesinde, belirtilmesi koşuluyla,gerçek kişilerin tüzel kişilerin adına elektronik imza kullanabileceği belirtilmiştir.

Yasada tanımlanmış güvenli elektronik imzalar, elle atılmış ıslak imza ile aynı hukuki sonucu doğurmaktadır. Fakat bu sonucu doğurmaları için yasada belirtildiği gibi Nitelikli Elektronik Sertifika (NES) ve elektronik imza oluşturma aracı kullanılarak oluşturulmaları gerekir.

Nitelikli Elektronik Sertifika'nın özellikleri ise yasanın 9. Maddesinde belirtilmiştir. Nitelikli Elektronik Sertifika (NES), hukuk nezdinde, ıslak imza ile eşdeğer kabul edilen "Güvenli Elektronik İmza" nın en önemli bileşenlerinden biridir. Yasaya göre NES ler, yurtiçinde kanunlara tabi bir hizmet sağlayıcıdan veya yurt dışındaki bir hizmet sağlayıcıdan temin edilebilecektir.

Nitelikli Elektronik Sertifikalarla ilgili detaylı bilgi 8. bölümde verilmiştir.

7.2 E-imza Algoritmaları

İmzalama işlemi için RSA başta olmak üzere El-Gamal, DSA, E-Sign algoritmaları kullanılmaktadır.

RSA algoritması kullanılarak imzalama işlemi aşamaları şöyledir [37]:

1- m = R (m) hesaplanır, burada aralık değeri [0 , n – 1] olur.

2- s = m^d (mod n) formülünden s hesaplanır.

3- Mesaj için imza değeri s'dir.

RSA algoritması kullanılarak imza doğrulama işlemi aşamaları şöyledir [37]:

1- Genel anahtar olan (n, e) değerleri elde edilir.

2- m = s^e (mod n) hesaplanır.

3- m ‘in MR elemanı olduğunu doğrulanır, değilse reddedilir.

4- m(message) = R-1 (m) elde edilir. [38,39]

7.3 Zaman Damgası

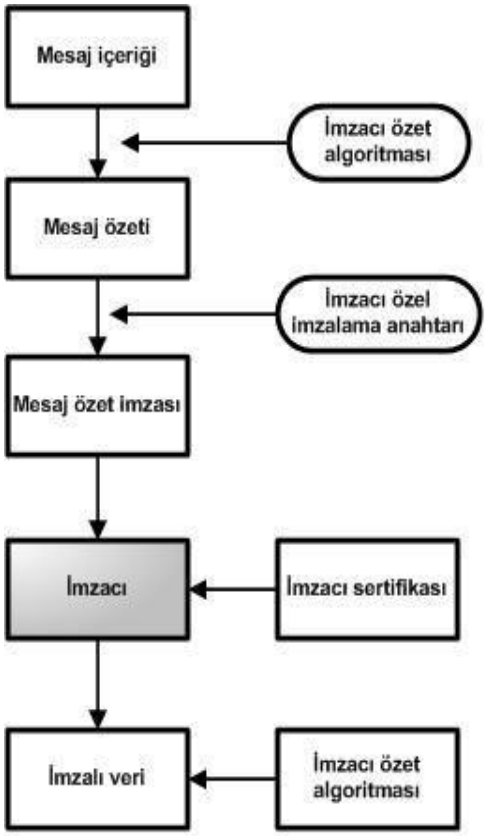
5070 sayılı Elektronik İmza Kanunu’na göre zaman damgası “Bir elektronik verinin, üretildiği, değiştirildiği, gönderildiği, alındığı ve / veya kaydedildiği zamanın tespit edilmesi amacıyla, elektronik sertifika hizmet sağlayıcısı tarafından elektronik imzayla doğrulanan kayıt”tır.

7.4 Elektronik İmza Formatları

Elektronik imza formatları “Basit Elektronik İmza (BES)” ve “Belirlenmiş Politika Temelli Elektronik İmza (EPES)” olarak tanımlanmıştır. Doğrulama verili elektronik imza formatları ise “Zamanlı Elektronik İmza(ES-T)”, “Doğrulama Verisi Referanslı Elektronik İmza(ES-C)”, “Genişletilmiş Elektronik imza” ve “Arşiv Elektronik İmza(ES-A)” olacak şekilde dört adettir. Bu elektronik imza türleri ETSI TS 101 733 standardına göre oluşturulmuştur.

7.4.1 Basit Elektronik İmza (Basic Electronic Signature)

Elektornik imzalar içinde en basit olanıdır. Özellikleri minimumdur. Diğer formatlara temel teşkil eder. Elektronik imza yapısının iskeleti burada oluşturulur. Basit E-imza da “İçerik Tipi (Conten-Type)” , “Mesaj Özeti (message-digest)” ve “İmzalama Sertifikası (signing certificate)” bulunması gereken özelliklerdir. ETSI standartına aie CMS imza formatının “Basit elektronik İmza” olabilmesi için taşıması gereken minimum özellikler bunlardır.



Şekil 7.1. İmzalı Veri Oluşturmak İçin Gerekli Adımlar

7.4.2 Belirlenmiş Politika Temelli Elektronik İmza (Explicit Policy-based Electronic Signature)

Bu imza türü imzalama politikasi ile oluşturulmalıdır. Politika bilgisi BES imzadaki özelliklere ek olarak "İmza Politika Belirleyicisi İmza Niteliği" ile imzaya eklenir. İmzalama politikası, e-imza oluşturma ve doğrulanmasınaait kuralları içerir ve bir özellik olarak imzaya eklenir. EPES tanımlanmış olan politikaya göre oluşturulur ve yine bu politikalar eşliğinde doğrulanma işlemi gerçekleştirilir.

7.4.3 Zamanlı Elektronik İmza (Electronic Signature Time-stamped)

Bu format, BES veya EPES den birine zaman damgası bilgisinin eklenmesiyle oluşur. Zaman damgası bilgisi, güvenliği sağlanmış bir zaman damgası servisi tarafından sağlanır ve imzaya bilgisine imzasız olarak eklenir.

7.4.4 Doğrulama Verisi Referanslı Elektronik İmza (ES with Complete validation reference data)

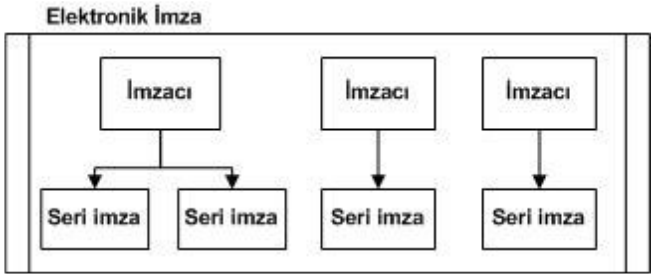
Bu imza türü "Zamanlı Elektronik İmza" ya sertifika ve iptal referans bilgilirenin eklenerek oluşturulur. Bu bilgiler imzasız olarak, imza yapısına eklenir. Tüm sertifika referansları, imzanın doğrulanabilmesi için kullanılan sertifika yolu üzerindeki tük sertifikaları içerir. Tüm iptal referansları ise imzanın doğrulanması için kullanılan tüm sertifika iptal listelerini ve OCSP cevaplarını içerir.

7.4.5 Genişletilmiş Elektronik İmza (ES with Extended validation data)

Doğrulama Verisi Referanslı Elektronik imzaya, imzalanmamışdeğerler eklenerek genişletilebilir.

7.4.6 Arşiv Elektronik İmza (ES with Archive validation data)

Uzun zamanlı imzaların arşivlenmesi için kullanılan imza formatıdır. "Genişletilmiş Elektronik İmza" veya "Zaman Damgalı Gelişletilmiş Elektronik İmza" ya "Arşiv Zaman Damgası" eklenerek oluşturulur.



Şekil 7.2. Elektronik İmza Yapısı

7.5 Elektronik İmza Uygulama Alanları

E-imza yardımıyla çok çeşitli güvenlik tedbirleri almak mümkündür. Kimlik veya veri doğrulama, veri paylaşımı sırasında kişi doğrulama gibi işlemler ile güvenlik sağlanılabilir. Güvenlik seviyesi nasıl olursa olsun e-imzadan faydalanılabilinir. Askeri seviyede kimlik tanılama, e-ticaret de kimlik ve kurum tanılama, e-posta veya anlık mesajlaşma işlemlerinde yine kişi ve/veya veri tanılama gibi daha bir çok alanda e-imzadan faydalanmak mümkündür.

E-imzanın ortaya çıkış sebeplerinden biri de resmi yazışmaların hızlandırılmasıdır. Resmi yazışmaların hazırlanma, ıslak imza atılması ve alıcıya

gönderilip işleme sokulması çok uzun zaman alabilirken, bir belgenin elektronik ortamda hazırlanması ve e-imza kullanılarak imzalanması ve gönderilmesi saniyeler sürebilir. Bu durum hem zaman kazandıracağı gibi bazı sarf malzemelerinden kar edilmesi anlamına gelir. Böyle bir hız ve güvenilirlik e-imzanın hukuki geçerliliği olması ile gerçekleşebilirdi. Çıkarılan yasa ve yapılan altyapı çalışmalarıyla bu statü e-imzaya kazandırıldı. Artık ülkemizde özel sektör veya kamuya ait bir çok kurum e-imza ile yazışmalar gerçekleştirmektedir. Hala yaygınlaşmakta olan e-imzanın şu hizmetlerde de yaygınlaşması beklenmektedir:

- Her türlü başvurular (ÖSS, KPSS, pasaport, ikametgah, taşınma vb.),
- Elektronik belgeler: Bireylerin devletten almak zorunda olduğu ve devletin tuttuğu kayıtlar (nüfus kâğıdı, pasaport, tapu ve kadastro, nüfus, adli kayıt ve sicil, askerlik, ithalat/ihracat vb.),
- Veri aktarımı (Devlet içi yazışmaların, belge ve veri aktarımların yapılması),
- Yerel yönetim uygulamaları,
- Kurum içi ve kurumlararası resmi yazışmalar,
- Kurumlar arası işlemler (Emniyet/nüfus ve vatandaşlık işleri),
- Sosyal güvenlik uygulamaları (Emekli sandığı, SSK, Bağkur),
- Sağlık uygulamaları (Sağlık personeli – hastaneler - eczaneler, Sağlık veri bankaları),
- Yargı hizmetleri (Yargı sistemindeki etkileşimlerin kağıt ortamından elektronik ortama taşınması),
- Ödemeler (Vergi, harç vb. ödemeler),
- Elektronik oy verme işlemleri vb.

Türkiye'de özel sektöre yönelik e-imza uygulamalarının ise aşağıda verilen alanlarda olması beklenilmektedir:

- Tüm ilk anlaşmalar,
- Kağıt tasarrufu,
- İnternet bankacılığı
- İnternet üzerinden gerçekleştirilen borsa işlemleri,
- Sigortacılık işlemleri, menkul kıymetler,
- E-ticaret,

- E-sipariş,
- E-sözleşme,
- E-fatura,
- E-kütüphane vb.

Gerek kamuya veya gerekse özel sektöre ait internet tabanlı hizmetler geliştikçe listeye yeni alanlar eklenebilir. Eklenecek alanlarda amaç hızlı ve güvenilir bir internet hizmeti vermek ve işlemleri hızlanmaktır.

Hizmetler e-imza kullanımı, hizmetler için oluşturulmuş yazılımın bir özelliği olmalıdır. Böyle bir özelliğin kullanılması için yönetmeliklerde o sektörle ilgili iş akışlarının tanımlı olması gerekir. Tanımlama işleminden sonra gerekli olan altyapı (donanım, iletişim vb.) kurularak yazılıma özellik olarak eklenir.[39,40,41,42.43,44]

8. ELEKTRONİK SERTİFİKA

Elektronik sertifika, bir belgenin imzalama-doğrulama sürecinde imzalayanın kimliğini güvenilir üçüncü taraflar tarafından onaylanması amacıyla kullanılır. 5070 sayılı kanunda "İmza sahibinin imza doğrulama verisini ve kimlik bilgilerini birbirine bağlayan elektronik kayıt" olarak tanımlanmıştır.

E-imzanın, ıslak imzanın yerini tutması, resmi yazışmalarda kullanılabilmesi, e-devlet uygulamalarında yeri olması, hukuki olarak tanımlanmasıyla olabilir. TBMM nin 15.01.2004 tarihli toplantısında 5070 sayılı Elektronik İmza Kanunu kabul edilmiş, 23.01.2004 tarihli ve 25355 sayılı resmi gazetede yayınlanarak, 23.07.2007 tarihinde yürürlüğe girmiştir.

Günlük hayatta kullanılan kanunla belirlenmiş nitelikli elektronik sertifikalar (NES) kanun tarafından tanımlanan ve devlet tarafından yetkilendirilen kurumlar vasıtasıyla üretilir ve dağıtılır. Bu kurumlar kanunda Elektronik Sertifika Hizmet Sağlayıcı (ESHS) olarak tanımlanmıştır. Kanunda elektronik sertifika hizmet sağlayıcıları "elektronik sertifika, zaman damgası ve elektronik imzalarla ilgili hizmetleri sağlayan kamu kurum ve kuruluşları ile gerçek veya özel hukuk tüzel kişilerdir" şeklinde tanımlanmıştır.

Elektronik sertifika hizmet sağlayıcıların bazı yükümlülükleri şöyle sıralanabilir:

Nitelikli sertifika verdiği kişilerin kimliğini resmî belgelere göre güvenilir bir biçimde tespit etmek: Bir kişiyi karışıklığa neden olmadan, başka bir kişiden ayırmak için dört bilgi yeterlidir. Bu bilgiler ad ve soyadı, doğum yeri, doğum tarihi ve kimlik numarasıdır. Bu bilgiler devlet tarafından verilmiş resmi belgelerden elde edilmelidir. Bu belgeler kimlik, pasaport veya bunlara eşdeğer nitelikteki bir resmi belgeolabilir.

Sertifika sahibinin diğer bir kişi adına hareket edebilme yetkisi, meslekî veya diğer kişisel bilgilerinin sertifikada bulunması durumunda, bu bilgileri de resmî belgelere dayandırarak güvenilir bir biçimde belirlemek yükümlülükler kapsamına girer.

Sertifika hizmet sağlayıcısı sadece güvenliğini sağlaması gereken unsurlardan biri de sertifika üretme sürecidir. İmza oluşturma verisi eğer sertifika hizmet sağlayıcısı tarafındanyada sertifika talep eden kişi tarafından hizmet sağlayıcıya ait yerlerde üretiliyorsa nu işlemin güvenliğini sağlamakla yükümlüdür. Eğer hizmet sağlayıcının

sağladığı araçlarla üretilmesi durumunda bu işleyişin güvenliği yine hizmet sağlayıcının yükümlülüğü içindedir.

Hizmet sağlayıcı sertifikayı kendi araçlarıyla üreterek sertifika sahibine teslim eder veya sertifika sahibi kendi araçlarıyla üretir ve hizmet sağlayıcı tarafından onaylanır. Her iki durumda da hizmet sağlayıcı bu işlemin gizliliğini sağlamak ve kopyalamayı engellemekle yükümlüdür. İlk durumda bu işlemin güvenliğini sağlamak için gerekli tedbirleri almalı, ikinci durumda da sertifika sahibini tedbirleri alması konusundayönlendirmelidir.

Sertifika, ıslak imza ile eşdeğer e-imza oluşturmak için kullanılacağından dolayı kişiye özgüdür. Dolayısıyla sahibinden başkası tarafından kullanılmamalıdır. Bu konuda hizmet sağlayıcı sertifika sahibini yazılı olarak bilgilendirmelidir. Ayrıca sertifikanın kullanımı ile ilgili de bilgilendirme yapılmalıdır. Bu bilgilendirmeden sonra, yanlış kullanımlar sonucu ortaya çıkan sonuçlardan hizmet sağlayıcı sorumlu değildir.

Hizmet sağlayıcı sertifikanın kopyasını alamaz veya oluşan veriyi saklayamaz[44].

Yabancı Elektronik Sertifikalar

5070 sayılı Elektronik İmza Kanunu'nun 14. maddesine göre yabancı bir ülkede yerleşik olan elektronik sertifika hizmet sağlayıcıdan temin edilmiş sertifikaların Türkiye'de Elektronik İmza Kanunu kapsamında hukuki değer alması iki yolla olabilir:

i) Uluslararası antlaşmalar

ii) Eğer böyle bir anlaşma yoksa ülke içinde yerleşik olan bir hizmet sağlayıcı tarafından kullanıma sunulmuş olması veya bu hizmet sağlayıcı tarafından garanti veren sıfatıyla taahhüt edilmesi.

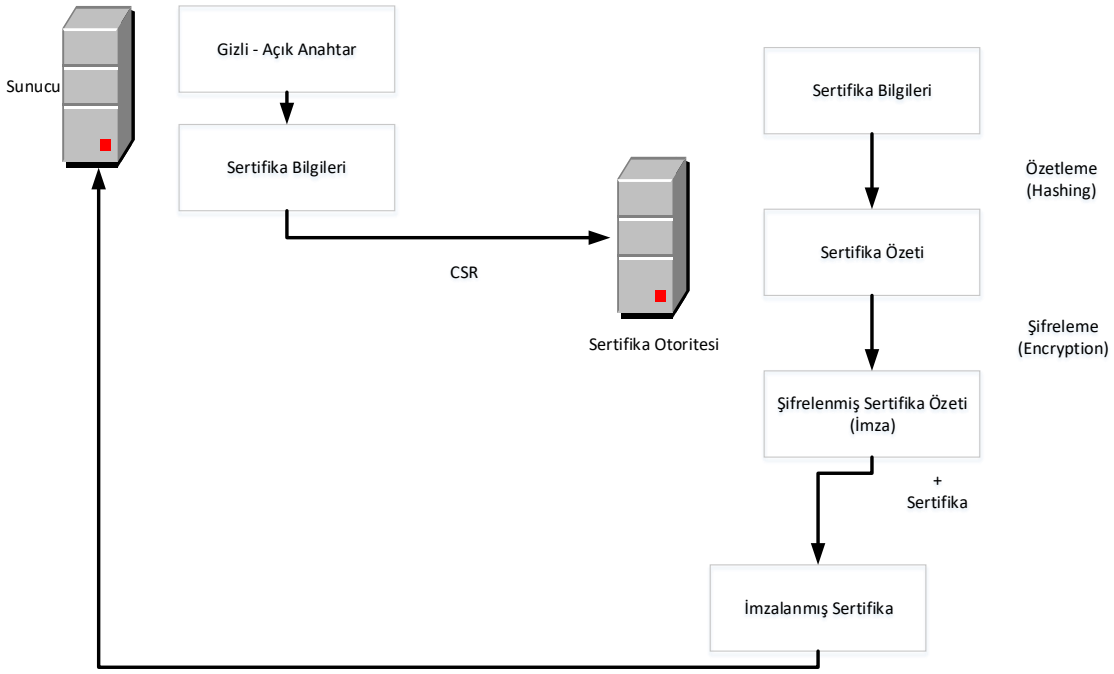
Bu şartlar sağlandığında yabancı NES ler Türkiye'de geçerli ve hukuki statüsü olan elektronik sertifikalar olacaklardır. Yabancı NES lerin kendi ülkelerindeki statüsü önemli değildir. Dikkat edilmesi gereken nokta 5070 sayılı kanunda açıklanmış olan NES ler ile aynı hukuki statü ve sonuçlara sahip olması durumudur. Ortaya çıkacak sonuçlardan garanti veren hizmet sağlayıcı da sorumlu olacağından garanti etme yöntemi hem hizmet sağlayıcıyı hem de tüketiciyi koruması gerekmektedir. Burada çapraz sertifikasyon (cross- certification) yöntemininтercih edilerek çapraz

sertifikasyon yapacak olan elektronik sertifika hizmet sağlayıcıları arasındaki kural ve koşulların yönetmelik içinde detaylandırılması doğru bir yaklaşım olacaktır.[41]

8.1 SSL – Secure Socket Layer Nedir?

Otorite tarafından üretilen sertifika, sertifika sahibinin iddia ettiği kişi olduğunu ispatlamak için kullanılır. web ortamında bu görevi SSL üstlenmiştir.

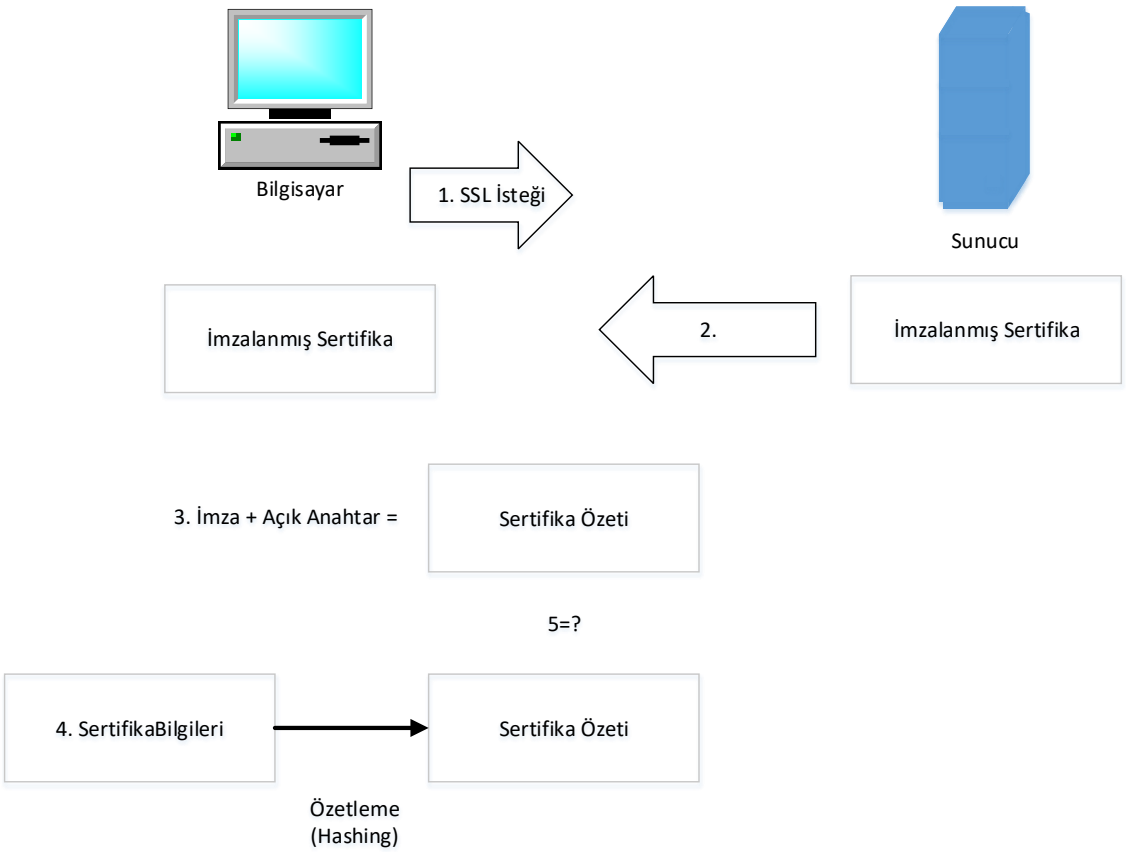
Web sitesi üzerinden sertifika ile imzalama ve veriyi şifreleyerek gönderme işlemi sunucular üzerinden yapılır. SSL sertifikası web sitesinin bulunduğu sunucuya eklenir. Bir sunucuya sertifika alabilmek için sertifika otoritesine sunucunun gizli anahtarı (private key) ile oluşturulmuş bir csr (certificate signing request – sertifika imzalama isteği) dosyası gönderilmelidir. Bu isteğe karşı yapılan kontroller sonucunda, sertifika otoritesi kendisi tarafından imzalanmış bir sertifika oluşturur ve isteği yapan firmaya gönderir. Otoritenin sertifikayı imzalama işlemi şöyledir: sertifika içeriğinden bir özet (hash) değer oluşturulur ve bu özet değer otoritenin kendi gizli anahtarı kullanılarak imzalama işlemi gerçekleştirilir. Bu imza değeri de sertifikaya dahil edilir. Otoriteden gelen sertifika web sitesinin bulunduğu sunucuya yüklenir ve gerektiği hallerde sunucu sertifikayı kullanarak imzalama ve şifreleme işlemlerini gerçekleştirilir.



Şekil 8.1. Sertifika otoritesinin sertifika oluşturma ve imzalama süreci

İstemci web tarayıcı aracılığıyla sunucudan kimliğini doğrulaması ister. Bu işlem sertifika aracılığıyla gerçekleşir. Sunucuya doğrulama talebi geldiğinde sertifika istemciye gönderilir. Sertifika firma hakkındaki bilgileri, sertifika seri numarasını, sertifikanın son kullanma tarihini, sertifika otoritesinin (sertifikanın alındığı kurum) imzasını ve sunucunun açık anahtarını (Public Key) içerir.

Sunucunun iddia ettiği sunucu olduğunu doğrulama işlemi sertifika üzerindeki değerler ile yapılır. Sertifika değerlerinden birinin otoritenin imzası olduğunu belirtmiştik. Bu imza sertifika ile gelen sunucunun açık anahtarı ile çözülür ve bir özet değeri (Hash) elde edilir. Bu özet değeri sertifka ile gelen bilgiler kullanılarak oluşturulan özet değeri (Hash) ile aynı ise sertifika ile sunucu, yani firma doğrulanmış olur. [45]



Şekil 8.2. Sunucunun sertifika kullanarak kendini tanıtması

SSL, http protokolüne SSL inde eklenmesiyle HTTPS (Güvenli Hiper Metin Aktarım Protokolü) protokolü üzerinden 443 numaralı portu kullanarak iletişim sağlar. http protokolünde ağı dinleyen kişi verilere düz metin olarak ulaşır ve istediği verileri

çekebilir, https de ise veriler paraçalara ayrılmış ve bu parçalar rastgele olarak dizilmiş ve şifrelenmiş bir düzende olduğundan ağı dinleyen kişinin işi zorlaştırılmıştır. [46]

Sertifika sistemlerinin ve açık anahtar altyapılarının sunduğu çözümlerle birlikte bir takım problemleri de vardır.

Mahtremiyet problemleri: SSL bağlantıları esnasında bazı sunucular istemci sertifikasını istemekte ve bu çağrı internet tarayıcı programlar tarafından cevaplanmaktadır. Bu durumda kullanıcı bilgileri internet ortamında kontrolsüz bir şekilde dolaşıma açılmış olur.

Güven Sorunu: Sertifika oluşturmak ve dağıtmak çok kolaydır. Gerekli yazılımları bulup herkese ücretsiz sertifika dağıtmak mümkündür. Fakat bu tip sertifikaların kök sertifikası olmayacağından uygulamalar veya tarayıcılar tarafından güvensiz olarak işaretlenecektir. [47]

Web siteleri kullanılarak yapılan e-ticaret süreci basitçe şöyledir:

Adım 1. E-ticaret sitelerinde seçilen ürün alışveriş sepetine eklendikten sonra ödeme sayfasına geçilir. Bu sayfa da müşterinin kredi kartı bilgileri alınır ve siteye iletilir

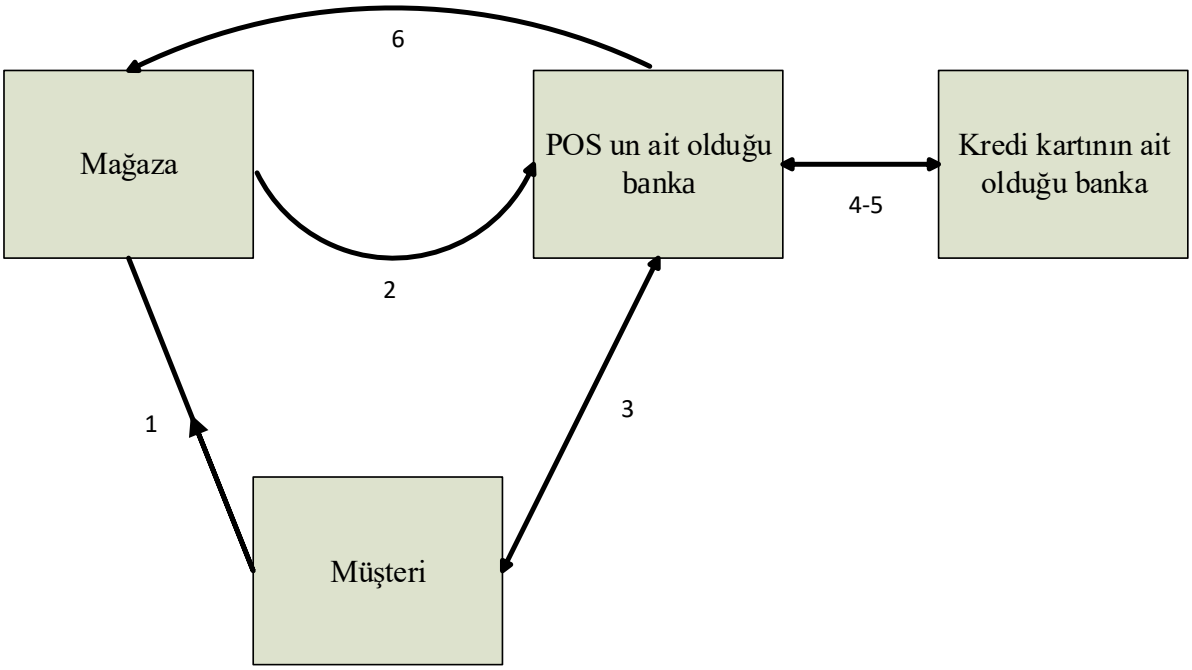
Adım 2. Müşterinin kredi kartı bilgileri, sitenin POS sisteminin olduğu bankaya iletilir.

Adım 3. Veri transferleri şifrelenerek yapılır.

Adım 4-5. POS siteminin bulunduğu banka ile müşteri kredi kartının bilgilerinin bulunduğu banka arasında iletişim sağlanarak bilgiler kontrol edilir ve onaylanır.

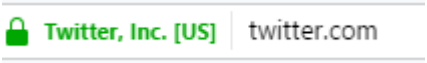
Adım 6. POS sisteminin bulunduğu banka ile e-ticaret sitesi arasında iletişim sağlanarak kredi kartının onaylandığı ödemenin alındığı gibi bilgiler iletilir ve alışveriş tamamlanır. [39]

Bu süreçteki veri iletim işlemlerinin tamamı SSL aracılığıyla şifrelenerek yapılır.



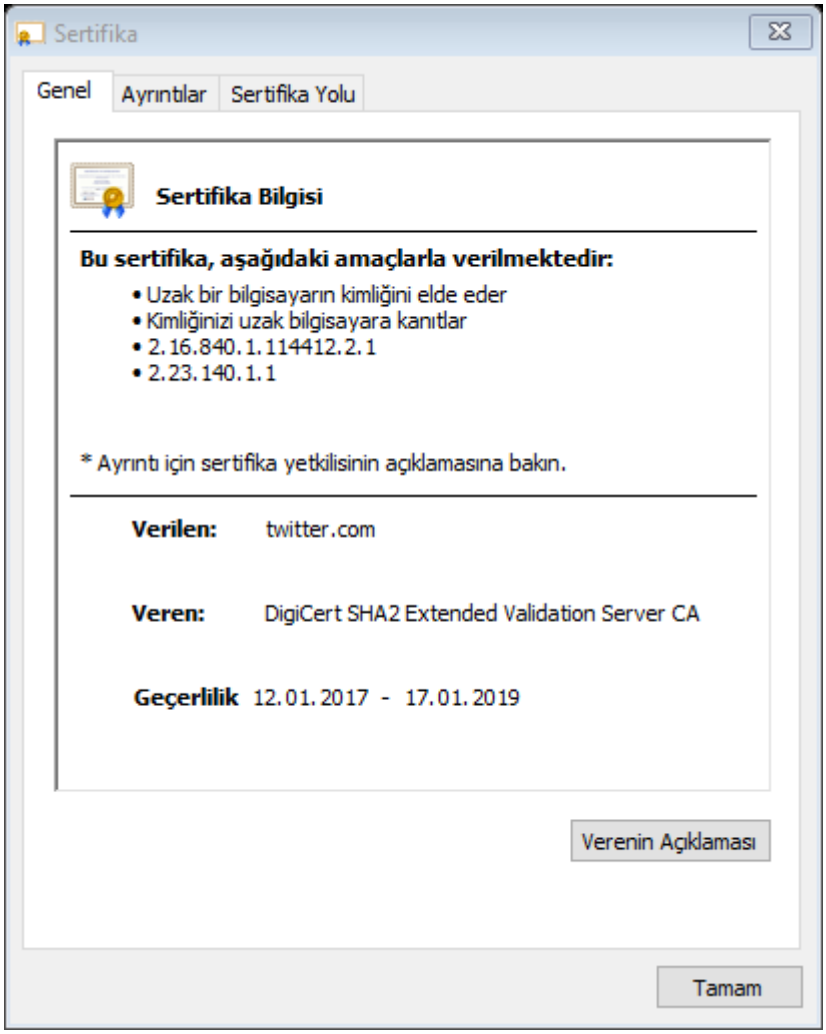
Şekil 8.3. E-ticaret süreci

Herhangi bir web sitesinin SSL kullanıp kullanmadığı, web tarayıcı adres çubuğunun yanındaki kilit işaretinden anlaşılabilir. Eğer kilit varsa SSL ve sertifika vardır. [48]



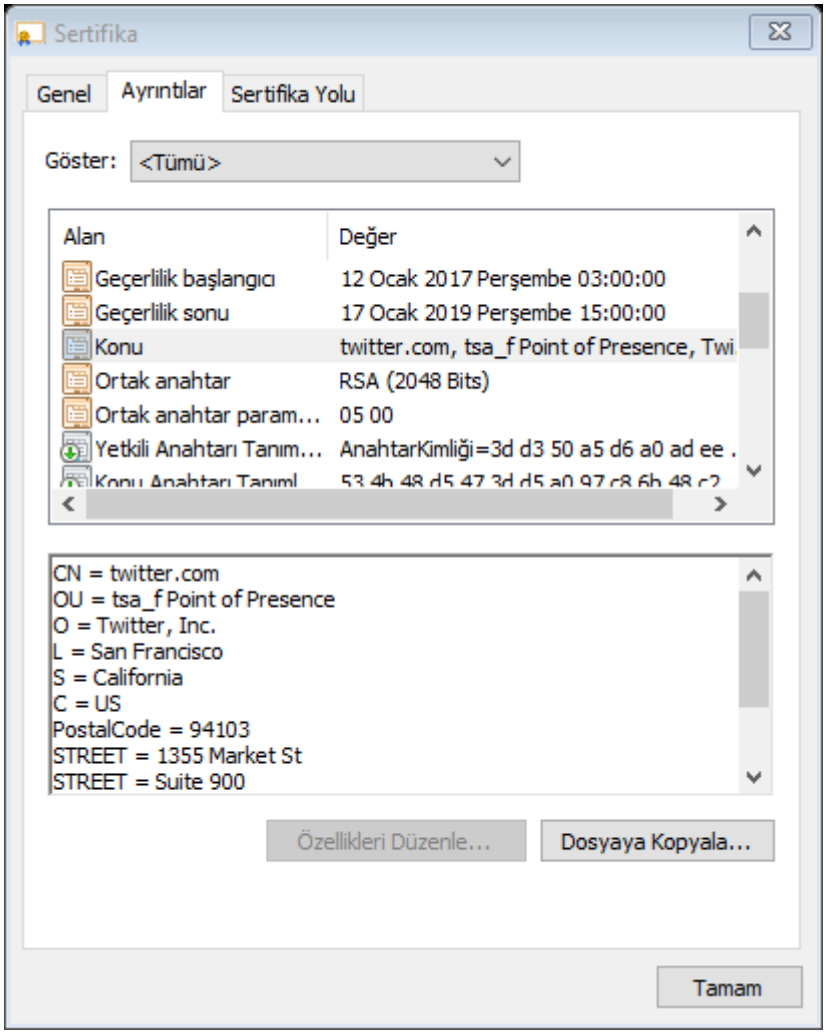
Şekil 8.4. Sertifika kullanan sitelerin adres çubuğundaki görünümü

Kilite tıklandığında o siteye ait sertifikaya ulaşılabilir. Sertifika da sertifikayı alan ve veren firmaların bilgilerine ulaşılabilir.



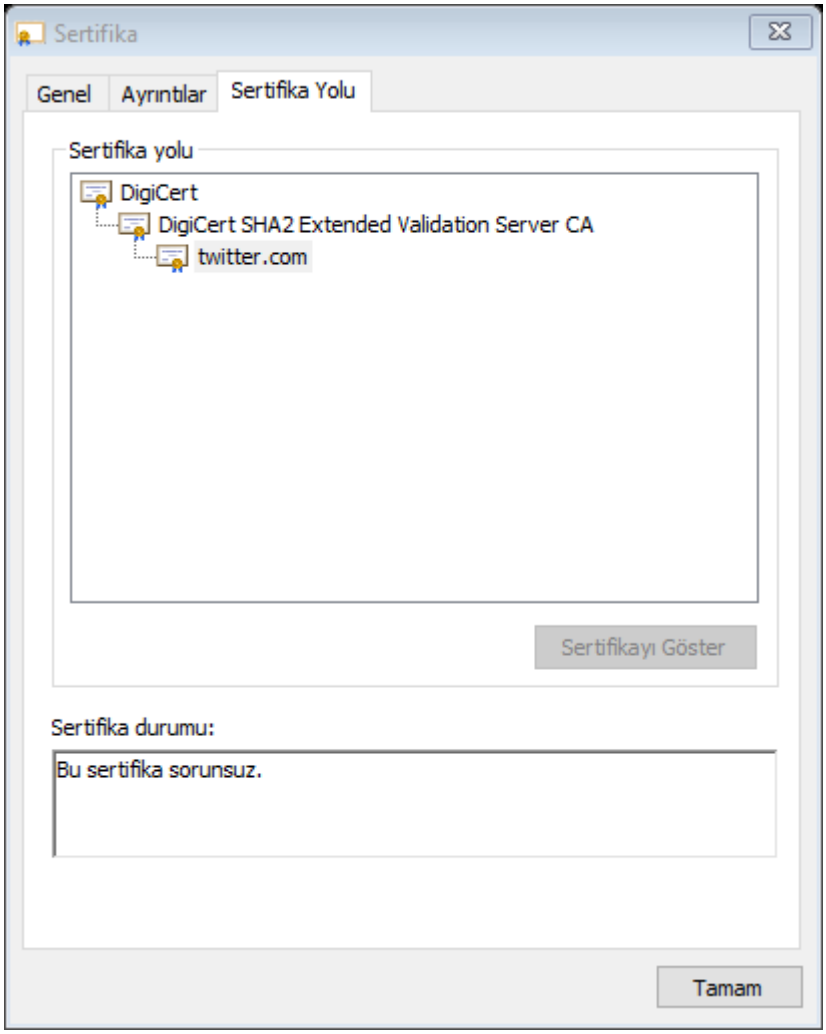
Şekil 8.5. Sertifika genel bilgileri

Bununla birlikte özet değerleri, açık anahtarlar, sertifika başlangıç ve bitiş tarihleri vb. bilgileri görüntülemek mümkündür.



Şekil 8.6. Sertifika ayrıntıları

Ayrıca sertifikanın kök sertifika ile olan ilişkisi de görüntülenebilir.



Şekil 8.7. Sertifika kök sertifika ilişkisi

9. HİBRİT MİMARİ YAPISI

Hibrit mimari ilke olarak simetrik ve asimetrik algoritmaların eksikliklerini gidermek için yine simetrik ve asimetrik algoritmalarınkullanılmasıyla oluşan bir çerçevedir. Kısacası kendisi bir algoritma olmaktan ziyade algoritmaları birleştiren bir mimaridir. Hibrit mimari yapısına geçmeden önce bu eksiklikleri inceleyelim.

Simetrik algoritmalar işlemcilere ek yük getirmeyecek şekilde tasarlanırlar. Basit öteleme, yer değiştirme işlemleri yapar. Bu yapıları en güçlü yanlarıdır ve çok hızlı çalışmalarını sağlar. Tasarlanan en güçlü algoritma, şimdilik, AES-Rijndael algoritmasıdır. 128-192-256 bit anahtar kullanarak şifreleme yapar. Anahtar boyutunun yüksek olması güvenliği artırır. AES-Rijndael algoritmasının kırılmasına yönelik ispatlanmış bir çalışma yoktur. Fakat simetrik algoritmalara karşı sık yapılan çalışmalar Bilinen Metin Saldırısı, Seçilen Şifre Saldırısıdır. Bu iki metot ile şifreleme anahtarının bulunması amaçlanır. Büyük boyutlu anahtar kullanan algoritmalara karşı Brute Force – Deneme Yanılma metodu çok fazla işlemci gücü ve zaman gerektirdiğinden ekonomik değildir. Simetrik algoritmalar şifreleme ve şifre çözme işlemleri için tek anahtar kullandığından Anahtarın ele geçirilmesi veya hesaplanması algoritmanın tüm gücünü kaybetmesi anlamına gelir. Ayrıca simetrik algoritmaları kullananlar içinde anahtarı alıcıya güvenli bir şekilde ulaştırmak da başka bir problemdir. Kısacası simetrik algoritmaların en zayıf tarafı anahtar yönetimidir.

Asimetrik algoritmalar çözülmesi zor matematik problemler üzerine kuruludur. Örneğin RSA algoritması, bilinen bir çarpanlara ayırma algoritması olmaması ve çok büyük asal sayıların çarpımlarından oluşan sayıların çarpanlara ayrılmaktaki zorluğuna dayanır. Bu tarz problemler çok fazla işlemci gücü ve zaman gerektirdiğinden ekonomik değildir. Dolayısıyla Büyük verilerin şifrelenmesinde tercih edilmezler. Asimetrik algoritmalar şifrelemek için ayrı şifre çözmek için ayrı anahtarlar kullanır. Şifreleme anahtarı açık anahtardır ve herkes tarafından bilinebilir, şifre çözme anahtarı ise gizli anahtardır sadece şifreyi çözmesi gereken kişinin bilmesi gerekir. Açık anahtar ve gizli anahtar matematiksel olarak birbirine bağlıdır fakat açık anahtardan gizli anahtarı elde etmek mümkün değildir. Şifreleme işlemleri ne kadar yavaş olsa da anahtar yönetimi bir o kadar başarılıdır.

Hibrit mimari bu zayıflıkları gidermek, güçlü taraflarını kullanmak yaklaşımı ile ortaya çıkmıştır. Yani aslında şifrelemede önemli olan noktalardan biri de algoritmaların gücü kadar, kullanım zamanları ve yönetimleridir.

Bu çalışmamızda hibrit mekanizmalar incelenmiş ve yeni bir mekanizma ortaya konmuştur.

9.1 Hibrit Mimari ile ilgili Literatür Özeti

Bu konudaki önemli çalışmalardan biri Masayuki Abe, Rosario Gennaro, Kaoru Kurosawa ve Victor Shoup tarafından yapılmıştır. Bu çalışmada Hibrit mekanizma iki modülden oluşturulmaktadır. Hibrit mekanizmanın verinin kendisini şifreleme mekanizmasına DEM Data Encapsulation Mechanism, şifreleme anahtarlarını yöneten mekanizmasına KEM-Key Encapsulation Mechanism denmiştir. İlke olarak verilerin DEM ile şifrelendiği anahtarlar ise KEM ile şifrelenerek güvenlik sağlanmıştır.[49]

Tag-KEM/DEM mimarisinde KEM oluşturulurken önce random bir anahtar seçilir sonra bu anahtar bir tag değeri ile şifrelenir. Fakat imzalama işlemleri bu mimaride bulunmamaktadır.

Fujisaki-Okamoto's KEM-DEM mimarisi "Tag-KEM/DEM" mimarisinin iyileştirilmiş halidir. Fakat bu mimaride de imzalama işlemleri bulunmamaktadır.

Diğer bir çalışma da Kerim YILDIRIM ve H. Engin DEMİRAY tarafından yapılan bir çalışmadır. "SİMETRİK VE ASİMETRİK ŞİFRELEME YÖNTEMLERİNE METOTLAR: ÇIRPILMIŞ VE BİRLEŞİK AKM-VKM" isimli çalışmalarında daha önce yapılmış olan hibrit mekanizmalarına iyileştirme yapılmıştır. [50]

Bu çalışmada 3 algoritmadan bahsedilmiştir. "Scrabbled KEM-DEM"mimarisinde KEM ve DEM yapıları alıcının açık anahtarı kullanılarak karıştırılmış ve tek lok halinde kullanıcıya gönderilmiştir. Bu karıştırma işlemindeki amaç saldırganın KEM veya DEM ile çalıştığının fark etmemesi için yapıldığı açıklanmıştır. Diğer algoritma olan cascaded KEM-DEM mimarisinde ise karıştırma algoritmasının alıcının açık anahtarı yerine random anahtarla yapmanın daha güvenli olduğu düşünülmüştür. Halbuki her iki durumda da DEM anahatarlarının KEM içinde şifrelenerek saklandığı düşünülürse aralarında bir fark olmadığı görülecektir. "Combined KEM-DEM" yapısında ise önce

sunucu ile oturum açma yoluna gidilmiş ve bu oturum açma işleminde "Scrabbled KEM-DEM" mimarisi ile güvenli iletişim sağlanmıştır. Oturum açma işleminde sonucu tarafından belirlenen oturum açma anahtarı şifreleme işlemi diğer iki sistemde olduğu gibi gerçekleştirilmiştir.

Bu çalışma güvenlik önlemlerini artırmasına rağmen imzalama ve doğrulama işlemleri eksik kalmıştır. Sistem içinde açık veya şifreli metin imzalanmamış sadece metnin şifrelenmesi için kullanılan anahtarı üreten anahtar imzalanmıştır. Halbuki anahtarın yerine metnin özetinin imzalanması daha anlamlı olacaktır. Ayrıca sistem içinde önce bir string ifade ile bir anahtar üretilmiş daha sonra bu anahtar kullanılarak şifreleme anahtarı üretilmiş ve şifreleme işlemi gerçekleştirilmiştir. Bu süreç sistemdeki iş yükünü artırıp sistemin çalışma süresini artıracaktır. Eğer şifreleme anahtarının göndericiden de bağımsız olması isteniyorsa herhangi bir string girişe gerek olmadan sistemin anahtar üretmesi sağlanabilir ve şifreleme gerçekleştirilebilir.

Her iki çalışmada veri güvenliği açık mesajın şifrelenmesi ve değişmediğinin garanti edilmesi için özetinin alınmasıyla sağlanmıştır. Günümüzde saldırı tekniklerinin çok çeşitli olduğu düşünülürse, mesajı gönderenin doğrulanması amacıyla gönderici tarafından imzalanması da önemli bir güvenlik parametresidir ve doğrulama amaçlı da kullanılmalıdır.

Bir başka çalışma olan "A Novel Idea on Multimedia Encryption using Hybrid Crypto Approach" isimli çalışmada hibrit algoritma mimarisinin dışına çıkılmadan multimedya dosyalarının güvenliğinde kullanılabileceği anlatılmıştır [51].

"Dik eşleştirme arayış yöntemi ile hibrit veri sıkıştırma ve optiksel kriptografi" isimli çalışmada işlenen sinyal hibrit mekanizma ile güvenlik altına alınarak iletilme işlemi önerilmiştir [52].

"A Password-Protected Secret Sharing Based on Kurosawa-Desmedt Hybrid Encryption" isimli çalışmada hibrit mekanizma, gizli paylaşım şemalarındaki kusurları gidermek için kullanılmıştır [53].

Başka bir çalışmada hibrit mekanizma bulut depolama sistemlerine yönelik olarak uygulama çözümü sunulmuştur. "The hybrid encryption algorithm of lightweight data in cloud storage" isimli çalışmada asimetrik algoritma olarak RSA, simetrik algoritma olarak AES kullanılmıştır. [54]

“Implementing a hybrid crypto-coding algorithm for an image on FPGA” isimli çalışmada hibrit mimarinin FPGA üzerinde çalışmasına yönelik bir fikir önerilmiştir. [55]

Görüldüğü üzere hibrit şifreleme algoritmalarına ait yapılan çalışmaların çok büyük çoğunluğu mimarinin geliştirilmesinden ziyade mimarinin uygulanması veya performans artışının sağlanması üzerinedir.

9.2 Güvenlik sertifikaları

Veri iletimi ve doğrulama için çeşitli sertifikasyon metotları mevcuttur. E-posta için PGP, S/MIME, http protokolü için SSL gibi seçenekler bulunmaktadır.

9.2.1 PGP (Pretty Good Privacy)

Bağımsız kullanıcılar tarafından geliştirilen PGP karmaşık bir güven mekanizması üzerine kurulmuştur. Kullanıcılar kendi güvenlik çemberlerini belirleyebilirler ve açık anahtarları onaylayan bir otorite olabilirler. Kullanıcıların açık anahtarları PGP açık anahtar sunucularında tutulur ve istem bazında bu sunucular tarafından dağıtılır.

PGP simetrik ve asimetrik sistemlerin avantajlarını birleştiren bir yöntemdir. Öncelikle veriyi sıkıştırarak küçültür. Bu işlem hem verinin iletimi hem de saklanması sırasında avantaj sağlar. Bu işlem ayrıca güvenliği de artırır. Bazı şifre çözme metotları şifreli veri üzerinde paket analizi yaparak exploit şablonlarının bulunması ilkesine dayanır. Sıkıştırılmış verilerde bu bölgeler azaldığından direnç artırılmış olur. Daha sonra oturum anahtarı ile oluşturulur. Bu oturum anahtarı simetrik algoritmanın gizli anahtarı görevini üstlenir. Gizli anahtar asimetrik anahtar kullanılarak şifrelenir ve alıcıya gönderilir. Şifre çözme işlemi sürecinde bu işlemler ters sırayla yapılarak açık metne ulaşılır. PGP içinde oturum anahtarı fare ve klavye hareketleriyle oluşturulan rastgele sayılardan oluşan bir değerdir.

9.2.2 S/MIME (Secure / Multipurpose Internet Mail Extensions)

S/MIME e-posta iletimi için, e-posta programları tarafından kullanılan protokoldür. Sertifika Otoriteleri (Certification Authority – CA) üzerine kuruludur. Sertifika Otoriteleri kişilerin açık anahtarlarını doğrulayan, kendileri de devlet tarafından doğrulanmış ticari kurumlardır. Kişileri doğrulamak için kullanıcı bilgileri ile

beraber kişilerin açık anahtarlarını da içeren sertifikalar dağıtırlar. Bu sertifikalar sertifika otoritesi tarafından imzalanarak güven altına alınır. İletişim sırasında kişiyi doğrulamak ve verinin internete kişinin açık anahtarı kullanılarak şifreli olarak gitmesini sağlamak için kişi sertifikası sertifika otoritesi tarafından doğrulanmış olması gerekir. Bunun için de sertifika otoritesinin kök sertifikası bilgisayarda bulunması gerekir. Oldukça basit bir sistem olan bu güvenlik hiyerarşisi için sadece gönderenin değil tüm alıcıların da kök sertifikayı kurması gerekir. [56]

9.2.3 SSL (Secure Socket Layer)

İnternet tarayıcılar tarafından kullanılan alt yapıdır. S/MIME de olduğu gibi sertifika otoriteleri tarafından sağlanan sertifikalar ile güvenli iletişim sağlar. Çoğunlukla bankalar olmak üzere e-ticaret yapan siteler tarafından kullanılır. 128 bit şifreleme ile sunucu – istemci arasında bilgi transferinde güvenlik iletişim ortamı sağlar. [57]

10.İletişim Güvenliği Mimarisi

10.1 Amaç

İletişim güvenliği sadece şifreleme olarak düşünülmemesi gerekir. iletişim sırasında transfer edilen verinin şifrelenmesi kadar iletişim halindeki kişilerin kimliklerinin doğrulanması, transfer edilen verinin değişmediği aynı zamanda gönderici tarafından gönderildiğinin ispatlanması gibi parametreler de iletişim güvenliği içinde değerlendirilmesi gerekir. Bu parametreler düşünülerek yeni bir mimari oluşturulmuştur. Oluşturulan mimari gerek sisteme dahil olma (Login olma) sırasında ve veri iletişimi sırasında güvenliği sağlayacak şekilde tasarlanmıştır.

Yeni oluşturulan mimari, diğer çalışmalardan farklı olarak, uygulamaya yönelik düşünülmüş ve daha programlanabilir bir mimari oluşturulmuştur. İletişim sağlanabilmesi için önce kişilerin ve iletişim sağlanacak cihazların doğrulanarak oturum açılması, ardından iletişimin sağlanması gerekir.

Kişi ve cihaz doğrulanması aşaması için kişiler ve cihazlar için hizmet sağlayıcı (Otorite) tarafından sertifikalar tanımlanacak ve bu sertifikalar kullanılarak doğrulama yapılıp oturum açılması sağlanacaktır. Tanımlanan sertifikalar kişiler ve cihazlar için bir nevi kimlik belgesi olacak ve kişi-cihaz a yönelik tanımlayıcı özellikte tekil bilgiler veya bilgilerin bir araya gelmesiyle tekillik oluşturan bilgiler içerecektir.

Oturum onayı alındıktan sonra iletişim aşamasına geçilecek ve hibrit mekanizma kullanılarak veri güvenliği sağlanacaktır. Önceki çalışmalardan farklı olarak hibrit mekanizma veri paketleri iki değil üç ayrı paket olarak düşünülmüştür. Bunun temel sebebi veri paketlerinin yönetimin daha kolay olması sağlamaktır. Bir diğer sebepte, veri paketlerinden biri olan güvenlik paketi de şifrelenerek alıcıya gönderilmesidir. Güvenlik paketinin oluşturulması ve şifrelenmesi sırasında mesajı gönderenin ve mesajın değişmediğinin doğrulanması işlemi için gerekli olan detaylar arttıkça güvenlik paketi büyüyeceğinden, asimetrik şifreleme algoritmalarından kaçarak simetrik şifreleme algoritmalarının tercih edilmesinin, performans açısından daha doğru olduğunun gözlenmesidir. Dolayısıyla asimetrik şifreleme algoritmasına sadece anahtar şifreleme işlemi yaptırılacak ve asimetrik algoritmanın veri paketinin küçülmesi sağlanıp hem güvenlikten hem de performanstan taviz verilmeyecektir.

10.2 Aşama1: Kişi-Cihaz Sertifikalandırılması ve Oturum Açılması

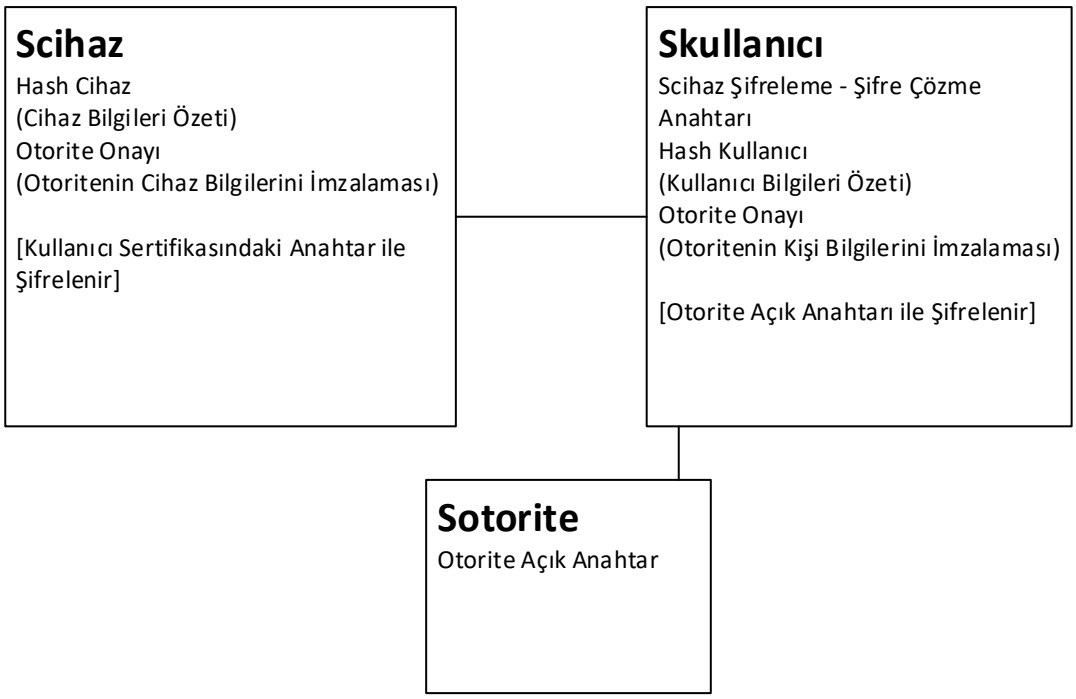
İletişime başlamadan önce güvenilir kişilerle, güvenilir cihazlarla iletişim kurulduğunun sağlanması gerekir.

Bu mimaride güvenilir cihaz olarak mobil cihazlar düşünülmüştür. Mobil cihazların güvenilir olduklarının göstergesi olarak cihaz sertifikalandırılacak ve bu sertifika her oturum başlangıcında kontrol edilecektir. Sertifikanın cihazın içinde saklandığı, kopyalanabileceği, değiştirilebileceği düşünülecek olursa sertifikayı güvende tutmanın yolu şifreli bir dosya halinde cihazın içinde saklamaktan geçer.

Cihaz güvenliğini kontrol etmenin amaçlarından biride cihazın iletişim kurmayı amaçlayan kullanıcıya ait olduğunun ispat edilmesidir. Bu durumda kullanıcı için de bir sertifika tanımlanmalı ve bu sertifika cihaz sertifikası ile ilişkilendirilmelidir. Ayrıca kullanıcı sertifikası içinde cihaz sertifikasının deşifrelenmesini sağlayan şifre anahtarı da bulunmalıdır. Bu kontrol cihaz sertifikasının dolayısıyla cihazın kullanıcıyla ilişkili olduğu ve güvenilir olacağı anlamına gelir.

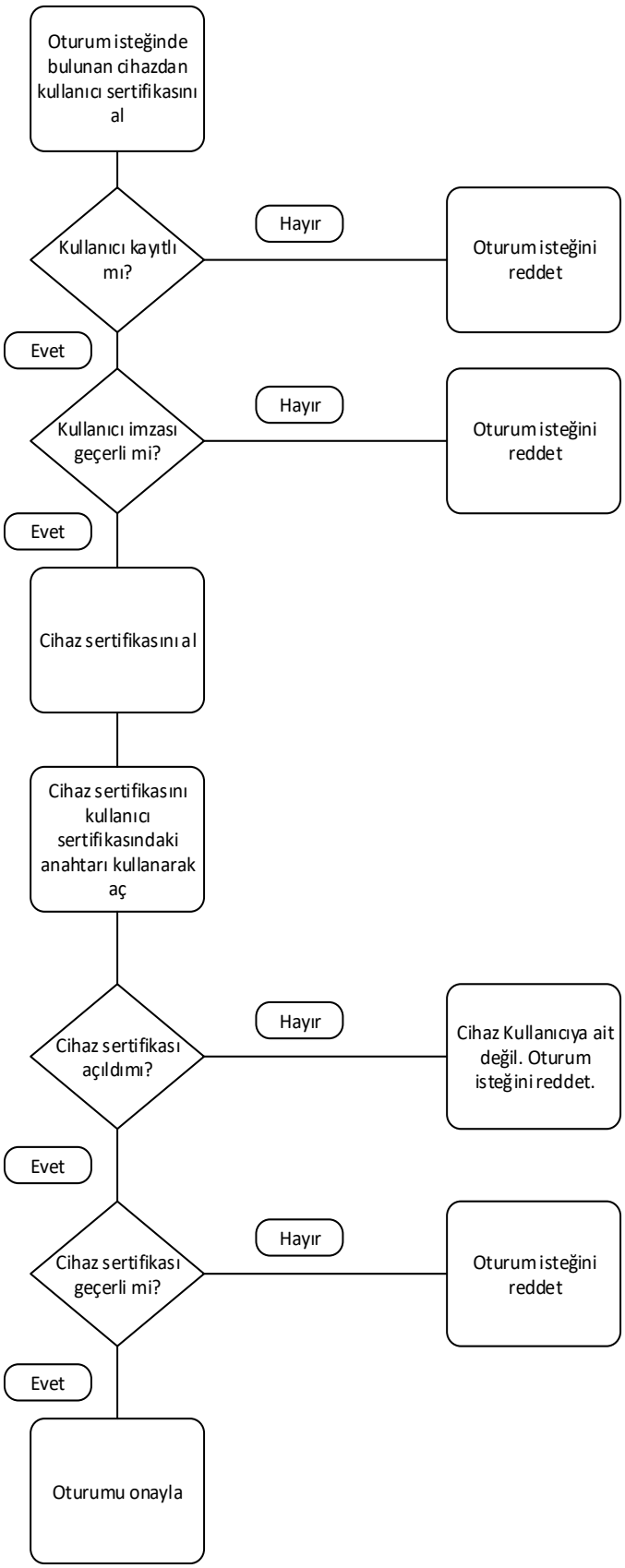
Kullanıcı sertifikası cihazın çalınması, cihazdaki dosyaların kopyalanması vb. tehlikelere karşı cihazın içinde tutulmamalıdır. Harici bir donanım içinde olabileceği gibi yine hizmet otoritesinin gizli anahtarı ile şifrelenmiş bir biçimde QR code biçiminde yazılı olarak da bulundurulabilir.

Her iki sertifika da hizmet alınan otorite tarafından imzalanmalıdır. Böylece sistem tarafından yetkilendirilmiş cihaz ve kişi olduğu garanti altına alınmış olur. Şekilde Üçlü Sertifika mimarisi ve birbirleri ile olan ilişkisi görülmektedir.



Şekil 10.1. Üçlü Sertifika Mimarisi

Oturum başlatma işlemi sunucu tarafından yönetilecektir. Uygulama sunucuya bağlanmak istediğinde ilk olarak kullanıcı sertifikası sunucu tarafından istenecek ve kullanıcının sertifikadaki bilgileri sunucudaki bilgilerle kontrol edilecektir. Onay alındıktan sonra cihaz sertifikası istenecek, kullanıcı sertifikasındaki cihaz çözme anahtarı ile cihaz sertifikası çözülecek ve bu sertifika ile cihaz bilgileri kontrol edilecektir. Cihaz bilgileri, sertifikadaki bilgiler ile eşleşme onayı alındıktan sonra oturum izni verilecektir.

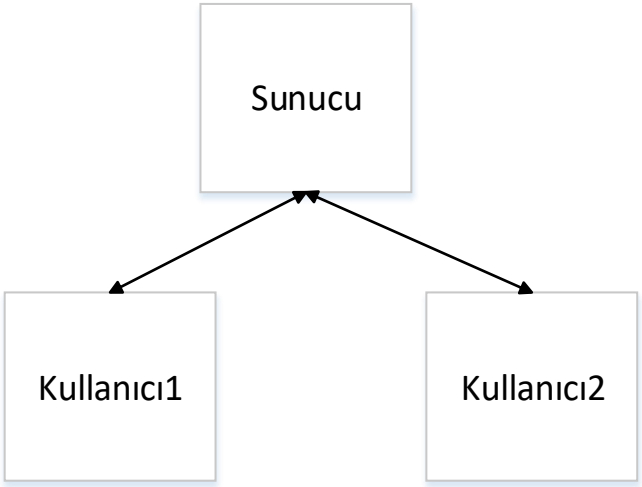


Şekil 10.2. Kullanıcı oturum isteği onaylama süreci

Kullanıcılar onaylandıktan sonra karşılıklı iletişim için, kullanıcılara ait bir takım bilgiler paylaşılmalıdır. Cihaz imzası, kullanıcı imzası ve kullanıcının asimetrik algoritma açık anahtarı sunucu tarafından kullanıcılara gönderilir. İmzalar doğrulama işlemleri için, açık anahtar da aşağıda anlatılmış olan paket3 ü açmak için kullanılacaktır.

Bu noktada şu soru akla gelebilir: Sadece imza kullanıcı doğrulama işlemi için yeterli olur mu?

Kullanıcı doğrulama işlemi sunucu tarafından yapılacağından mesaja eklenmiş olan imza, kullanıcının doğrulanması için değil, gönderilen mesajın sunucu tarafından onaylanmış kullanıcı tarafından gönderildiğini, yani başka bir deyişle güvenli kullanıcı olduğunu ispatlamak için kullanılır.



Şekil 10.3. Kullanıcı onayından sonra karşılıklı sertifika paylaşımı

10.3 Aşama2: Şifreli İletişim

Oturum güvenliği sağlandıktan sonra iletişim başlar. İletişimde yine sertifikalar önemlidir. Oturum açmak için kullanılan sertifikaların haricinde yeni sertifikalar kullanılır. Bu sertifikalar oturum onayı alındıktan sonra sunucu tarafından üretilerek kullanıcıya gönderilir. İlk olarak sunucunun, iletişim kurulacak kişinin ve cihazın açık anahtarlarının bulunduğu sertifika kullanıcıya gönderilir. Otorite ile iletişim kurulması gereken zamanlarda bu sertifikadaki açık anahtar kullanılarak veri şifrelenecektir. Kişi ile kurulan iletişimde ise kullanıcı ve cihaz açık anahtarları kullanılacaktır.

Bu sertifikalar sunucudan kullanıcıya gönderilirken kullanıcı tarafından anlık olarak üretilen bir değer ile şifrelenerek iletilir. Şifreli olarak otoriteden gelen sertifikalar, iletişim sırasında kullanılmak üzere açılmak için kullanıcı tarafından üretilen anahtar kullanılır. İletişim sırasında Hizmet sağlayıcının sertifika değerlerine ihtiyaç duyulmaz. Sadece, eğer yazışmalar sunucuda tutulacaksa paketlere otorite imzası da eklenerek 3 lü güven mekanizması (Kullanıcı1, Kullanıcı2, Otorite) devam ettirilir.

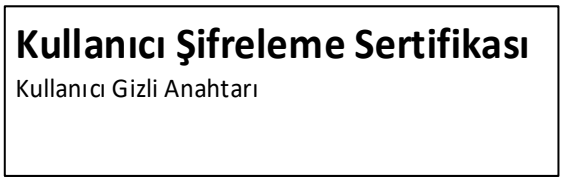
İletişim sırasında internet ortamına 3 parçadan oluşan iletişim paketi gönderilir.

10.3.1 Açık Anahtarlar Sertifikası

Açık Anahtarlar şifreleme için kullanılacağından otorite tarafından kullanıcılara karşılıklı olarak gönderilir. Herkes tarafından bilinebilir.

10.3.2 Kullanıcı Sertifikası

Kullanıcı sertifikası içinde sadece kullanıcı gizli anahtarı bulunacaktır. Bu anahtar imzalama ve gelen şifreli mesajları açmak için kullanılacaktır.



10.3.3 Cihaz Sertifikası

Cihaz sertifikası içinde sadece cihaz gizli anahtarı bulunacaktır. Bu anahtar imzalamak için kullanılacaktır.

10.3.4 Paket1 – Mesaj Paketi

Açık mesaj iki farklı işleme gider:

- Şifreleme işlemi için Simetrik Algoritma
- Mesajın gönderici tarafından imzalanabilmesi için İmza Algoritması

$\delta_{K1}(m) = y$ (10.1)

$Sh(y) = (z, q)$ (10.2)

$q \rightarrow \delta_{K2}$ (10.3)

$z \rightarrow Mesaj\ Paketi$ (10.4)

$\delta_{K1} =$ $K1$ Anahtarını kullanan Simetrik şifreleme algoritması

$\delta_{K2} =$ $K2$ Anahtarını kullanan Simetrik şifreleme algoritması

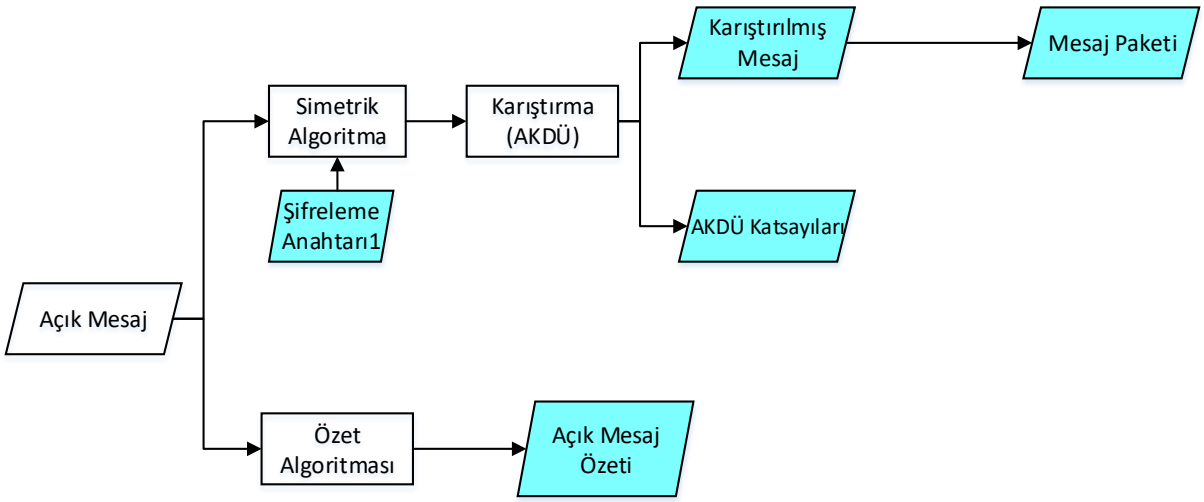
$Sh =$ Karıştırma Algoritması

$K1 =$ Mesaj Şifreleme Anahtarı (Simetrik Algoritma)

$z =$ Karıştırılmış Şifreli Metin

$q =$ Katsayı Dizisi

$y =$ Şifrelenmiş Metin (Mesaj Paketi)



Şekil 10.4. Mesaj Şifreleme ve Mesaj Paketi Oluşturma Akış Şeması

10.3.4.1 Şifreleme işlemi için Simetrik Algoritma

Açık mesaj, random üretilmiş anahtar ile simetrik algoritma kullanılarak şifrelenir. Şifrelenmiş mesaj karıştırma algoritmasına girer ve iki çıktı üretilir:

1. Karıştırılmış şifreli mesaj

2. Karıştırma (Haritalama) işlemi için katsayılar

Şifrelenmiş mesajın tekrar karıştırılmasının amacı simetrik algoritmalara karşı uygulanan bilinen metin saldırısı tarzında saldırılara karşı tahmini zorlaştırmaktır. Bu işlemin anlamlı hale gelmesi için katsayılar alıcı gönderilmeli ve haritalama tersine çevrilerek veri bloklarının orijinal yerlerinin tespiti sağlanmalıdır. Güvenlik paketinin girdilerinden biri de bu katsayılardır.

10.3.4.2 Mesajın gönderici tarafından imzalanabilmesi için İmza Algoritması:

Açık mesaj özet algoritmasına girerek özet değeri hesaplanır. Bu değer göndericinin gizli anahtarı kullanılarak imzalanır ve mesajın kullanıcı tarafından onaylanması anlamına gelen mesaj imzası oluşturulur. Mesaj imzası da güvenlik paketinin girdilerinden biridir.

10.3.5 Paket2 - Güvenlik Paketi:

Güvenlik paketi mesaj güvenliğini sağlamak için oluşturulur.

Açık mesaj özeti, karıştırılmış mesaj özeti bir veri paketi halinde önce cihaz gizli anahtarı sonra kullanıcı gizli anahtarı ile imzalanarak imzalı veri oluşturulur. Bu imzalı veriye karıştırma algoritması katsayıları da eklenerek oluşan veri paketi simetrik algoritma ile şifrelenir.

Şifreleme işlemi için random üretilmiş bir anahtar kullanılır, daha sonra bu anahtar paketine gider.

$H(m) = h_m$ (10.5)

$H(z) = h_z$ (10.6)

$SG_{SKC}(h_m) = I_{mC}$ (10.7)

$SG_{SKK}(I_{mC}) = I_{mK}$ (10.8)

$SG_{SKC}(h_z) = I_{zC}$ (10.9)

$SG_{SKK}(I_{zC}) = I_{zK}$ (10.10)

$(I_{mC}, I_{mK}) = I_m$ (10.11)

$I_m \rightarrow \delta_{K2}$ (10.12)

$q \rightarrow \delta_{K2}$ (10.13)

$\delta_{K2}(I_m, q) = \tau$ (10.14)

$\delta_{K2} = K2$ Anahtarını kullanan, güvenlik paketi öncesi, simetrik şifreleme algoritması

$SG_{SK} = SK$ Anahtarını kullanan imza Algoritması (Cihaz-Kişi)

$H =$ Özetleme (Hash) Algoritması

SKC = Cihaz Gizli Anahtarı (Asimetrik Algoritma)

SKK = Kullanıcının Gizli Anahtarı (Asimetrik Algoritma)

I_{mC} =Cihaz tarafından oluşturulan mesaja ait imza

I_{mK} =Kişi tarafından oluşturulan mesaja ait cihaz imzasını imzalama

I_{zC} =Cihaz tarafından oluşturulan karıştırılmış şifreli mesaja ait imza

I_{zK} =Kişi tarafından oluşturulan karıştırılmış şifreli mesaja ait cihaz imzasını imzalama

$K2$ = Güvenlik Paketi Şifreleme Anahtarı (Simetrik Algoritma)

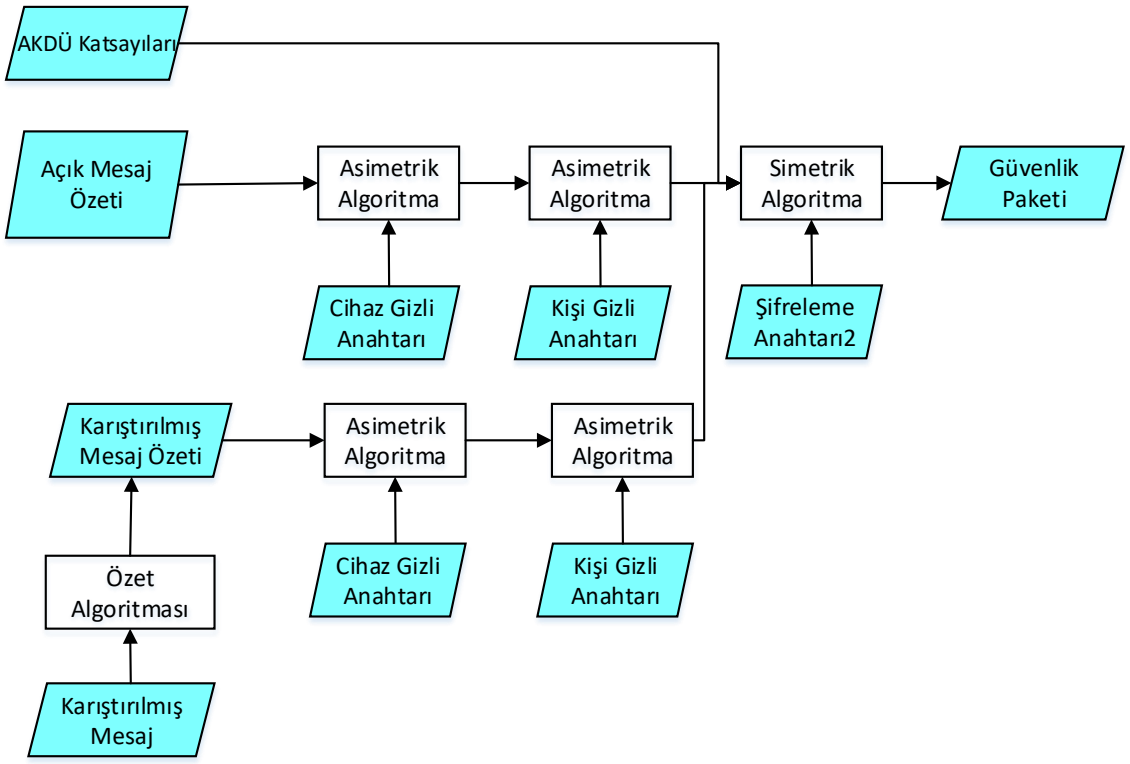
$m =$ Açık Mesaj

$h_m =$ Mesaj Özeti

$h_z =$ Karıştırılmış şifreli metin özeti

q =Katsayı Dizisi

$\tau =$ Güvenlik Paketi



Şekil 10.5. Güvenlik Paketi girdilerinin cihaz ve kişi tarafından imzalanarak şifrelenmesi ve güvenlik paketinin oluşturulması

10.3.6 Paket3 - Anahtar Paketi:

Son pakette Asimetrik algoritma kullanılır. Şifrelenmiş haldeki ilk iki paketin şifreleme anahtarlarını taşır. Mesaj paketinde ve Güvenlik paketinde kullanılan simetrik algoritmalara ait şifreleme anahtarların Asimetrik Algoritma ile şifrelenerek Paket3 oluşturulur.

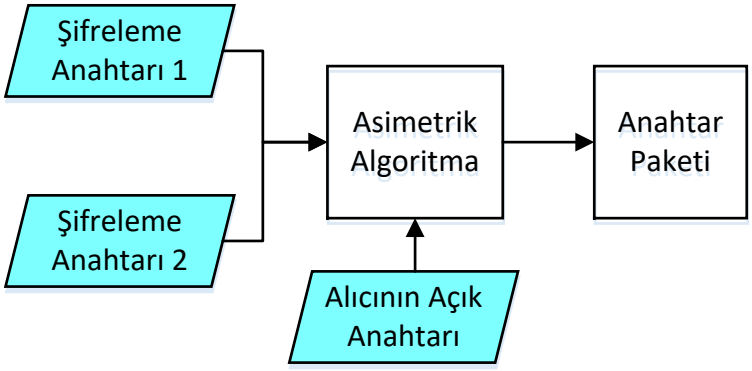
$\gamma_{PK}(K1, K2) = Ap$ (10.15)

γ= Asimetrik şifreleme algoritması

$K1$ = Mesaj Şifreleme Anahtarı (Simetrik Algoritma)

$K2$ = Güvenlik Paketi Şifreleme Anahtarı (Simetrik Algoritma)

PK = Kullanıcının Açık Anahtarı (Asimetrik Algoritma)



Şekil 10.6. Anahtar Paketi Oluşturma Süreci

$Vp\ (Mp, Gp, Ap)$

$Ap =$ Anahtar paketi

$Gp =$ Güvenlik paketi

$Mp =$ Mesaj paketi

Veri paketi, mesaj paketi güvenlik paketi ve şifre paketinin birleşmesiyle oluşur.

Şekil 10.7. Şifre Paketi ve Paketlerin Birleştirilmiş Hali

Alıcı mesajı aldıktan sonra mesaj içeriğini görüntüleyebilmesi için gerekli anahtarlar şifre paketi içinde şifreli olarak bulunmaktadır. Bu durumda şifreli paket sadece alıcı tarafından açılabilmesi gerekir. Bunun için şifreli paketin alıcının açık anahtarı ile şifrelenmesi gerekir. Şifreleme işleminin gerçekleşebilmesi için açık anahtarların karşılıklı olarak paylaşılması gerekir. Bu görev sunucu tarafından gerçekleştirilir. Kişilerin onaylanması süreci içinde imza kontrollerinin yapılabilmesi için kullanıcıların açık anahtarlarının zaten sunucuda bulunacağı düşünülecek olursa oturum onayı verildikten sonra açık anahtarlar karşılıklı olarak kullanıcı tarafından paylaştırılır.

Her iki kullanıcının da imza kontrolü yapabilmesi için açık anahtarla birlikte imza değerinin de kullanıcılara paylaştırılması gerekir. Bu paylaştırma işlemi sunucu tarafından alıcıların açık anahtarları kullanılarak asimetrik algoritma ile şifrelenerek alıcılara gönderilmesi gerekir.

10.4 Şifre Çözme

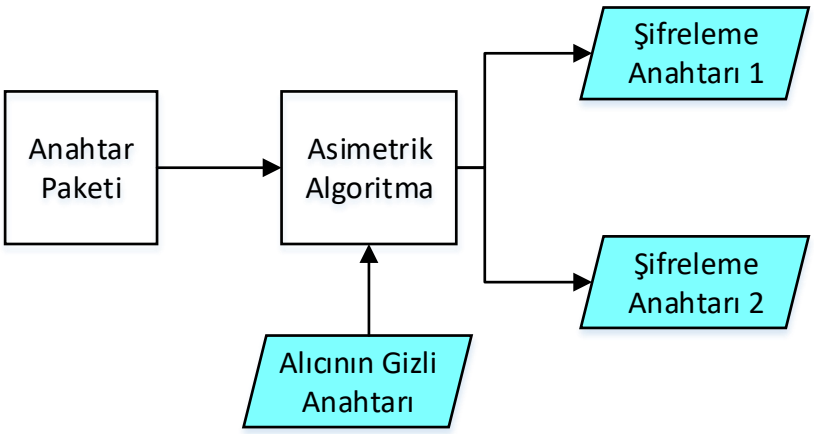
Veri paketi alıcıya ulaştığında açık mesaja, imza bilgilerine ulaşılabilmesi için işlemlerin sırasının tersine çevrilmesi yeterlidir. Simetrik algoritmada aynı anahtar, asimetrik algoritmada ise ikinci anahtarların kullanılması gerekir.

İlk olarak paketler birbirinden ayrılır. Her paket, mimari içinde kendi işlem sırasını takip eder. Önce Anahtar Paketi açılarak şifreleme anahtarları elde edilir. Ardından Güvenlik paketi açılarak karıştırma katsayıları ve imzalar elde edilir, imzalar kontrol edilerek özet değerlerine ulaşılır. Son olarak Mesaj Paketi açılır ve açık mesaj elde edilir. Açık mesaj için özet algoritmaları çalıştırılarak güvenlik paketinden elde edilen özet değerleri ile karşılaştırılır. Eğer bu karşılaştırma sonucunda elde edilen değerler aynı ise mesajın değiştirilmediği garanti edilmiş olur.

Şekil 10.8. Şifre Paketi ve Paketlerin Birleştirilmiş Hali

10.4.1 Aşama 1. Anahtar Paketinin Açılması

Anahtar paketi alıcının açık anahtarı ile şifrelenmiştir. Alıcının gizli anahtarı ile paket çözülerek mesaj şifreleme ve güvenlik paketi şifreleme için kullanılan simetrik şifreleme algoritmasına ait anahtarlar elde edilir.



Şekil 10.9. Anahtar Paketinin Açılması ve Şifreleme Anahtarlarının Elde Edilmesi

$\gamma_{SKK}(Ap) = (K1, K2)$ (10.16)

γ= Asimetrik şifreleme algoritması

$K1$ = Mesaj Şifreleme Anahtarı (Simetrik Algoritma)

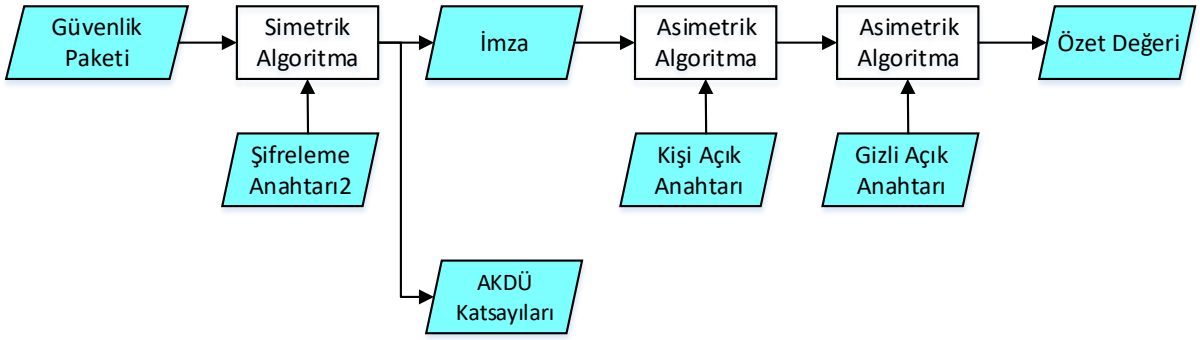
$K2$ = Güvenlik Paketi Şifreleme Anahtarı (Simetrik Algoritma)

SKK = Kullanıcının Gizli Anahtarı (Asimetrik Algoritma)

Şifreleme Anahtarı2 Güvenlik paketinin çözülmesinde, Şifreleme Anahtarı1 açık mesajı elde etmede kullanılacaktır.

10.4.2 Aşama 2: Güvenlik Paketinin Açılması

Veri paketinden elde edilen güvenlik paketi, anahtar paketinin çözülmesiyle elde edilen Şifreleme Anahtarı2 kullanılarak çözülür. Elde edilen veri, mesajın kullanıcı ve cihaz tarafından imzalanmış özetidir ve şifrelenmiş mesajın karıştırılması işleminde kullanılan katsayılardır. Bu özet bir sonraki aşamada elde edilen mesajın özeti alınarak karşılaştırılacak ve mesajın gönderildiği cihaz ve kişinin doğrulanması sağlanacaktır.



Şekil 10.10. Güvenlik Paketinin Açılması ve Özet değeri ile AKDÜ katsayılarının Elde Edilmesi

$\delta_{K2}(\tau) = (I_K, q)$ (10.17)

$SG_{SKK}(I_K) = I_C$ (10.18)

$SG_{SKC}(I_C) = (h_m, h_z)$ (10.19)

$H(m) = h_m$ (10.20)

$H(z) = h_z$ (10.21)

$I_K \rightarrow \delta_{K2}$ (10.22)

$q \rightarrow \delta_{K2}$ (10.23)

$\delta_{K2} = K2$ Anahtarını kullanan, simetrik şifreleme algoritması

$SG_{SK} = SK$ Anahtarını kullanan imza Algoritması (Cihaz-Kişi)

$H =$ Özetleme (Hash) Algoritması

$I_C =$Cihaz tarafından oluşturulan mesaja ait imza

$I_K =$Kişi tarafından oluşturulan cihaz imzasını imzalama

$K2 =$ Güvenlik Paketi Şifreleme Anahtarı (Simetrik Algoritma)

$m =$ Açık Mesaj

$h_m =$ Mesaj Özeti

$h_z =$ Karıştırılmış şifreli metin özeti

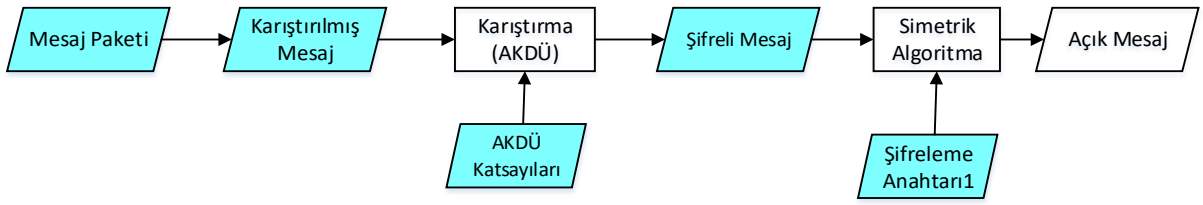
$q =$Katsayı Dizisi

$\tau =$ Güvenlik Paketi

İmza, mesaj verilerinin özet değerinin önce cihaz ardından kullanıcının gizli anahtarı ile peş peşe asimetrik algoritma ile işleme sokulmasıyla elde edilir. İmza doğrulama işlemi de önce göndericinin sonra cihazın açık anahtarı kullanılarak asimetrik algoritmada işleme sokularak yapılır. Bu işlemin sonucuna imza verisinin özeti elde edilmelidir.

10.4.3 Aşama 3: Mesaj Paketinin Açılması

Mesaj paketi şifrelendikten sonra karıştırma işleminde geçirilmiş olan veridir. Dolayısıyla önce karıştırma işlemi tersine alınarak şifreli mesaj elde edilmeli, elde edilen şifreli mesaj simetrik algoritma ile anahtar paketinden gelen Şifreleme Anahtarı1 ile çözülmelidir. Karıştırma işleminin geri alınması için işlenecek olan katsayılar güvenlik paketinden gelmektedir.



Şekil 10.11. Mesaj Paketinin Açılması ve Açık Mesajın Elde Edilmesi

$Sh(z)_q = y$ (10.24)

$\delta_{K1}(y) = m$ (10.25)

$\delta_{K1} =$ $K1$ Anahtarını kullanan Simetrik şifreleme algoritması

$Sh =$ Karıştırma Algoritması

$K1 =$ Mesaj Şifreleme Anahtarı (Simetrik Algoritma)

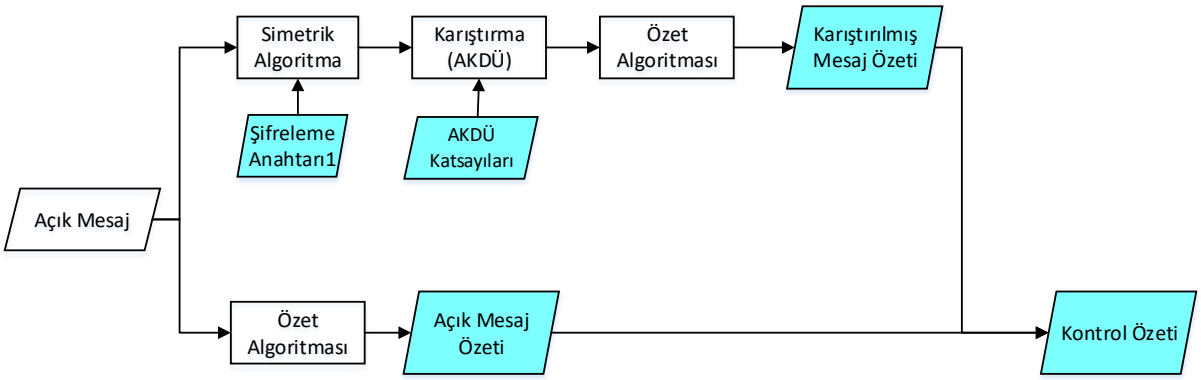
$z =$ Karıştırılmış Şifreli Metin

$q =$ Katsayı Dizisi

$y =$ Şifrelenmiş Metin (Mesaj Paketi)

10.4.4 Aşama 4: Doğrulama

İkinci aşamanın sonunda özet değeri elde edilmişti. Bu özet değeri açık mesaj özeti ile karıştırılmış mesaj özetinden oluşuyordu. Özet değerinden orijinal verileri elde etmek mümkün olmadığından dolayı açık mesaj yine karıştırma ve özet alma işlemlerinden geçirilerek özet değeri elde edilir ve veri paketi ile gelen özet değeri ile karşılaştırılarak doğrulama yapılır.



Şekil 10.12. Güvenlik Paketi ile Gelen Değerler ile Açık Mesajdan Elde Edilen Değerlerin Karşılaştırılarak Doğrulanması

Kontrol özeti veri paketinden gelen özetle karşılaştırılır. Eğer elde edilen sonuçlar doğru ise cihaz ve gönderici doğrulanmış olur.

11.UYGULAMA

İnşa etmiş olduğumuz iletişim güvenliği mimarisini Windows platformunda yazılmış olan bir anlık mesajlaşma uygulaması ile test ettik. Uygulama sunucu istemci mimarisi üzerine socket programlama kullanılarak gerçekleştirildi. Sunucu kısmı login olma sürecini kontrol edip mesajlaşma sürecinde sadece dağıtıcı olarak programlandı. Dolayısıyla login olma süreci haricinde sunucu üzerinde herhangi bir şifreleme, şifre çözme işlemi gerçekleştirilmedi. İmzalama, şifreleme, şifre çözme, imza kontrolü gibi işlemlerin tamamı istemci tarafında gerçekleştirildi.

Uygulamada simetrik algoritma olarak AES-256, asimetrik algoritma olarak RSA kullanıldı. AES-256 mimari de olduğu gibi mesaj paketini ve güvenlik paketini şifrelemek için kullanıldı. Anahtar büyüklüğü 256 bittir.

Mesaj paketi ve güvenlik paketine ait anahtarların şifrelenmesi ve imzalama işlemeleri için de RSA algoritması kullanıldı. RSA algoritması içinde büyük asal sayılar kullanıldı, rastgele iki asal sayı p ve q 401(1334 bit) ve 402 (1331 bit) basamaklı olacak şekilde seçildi. [59]

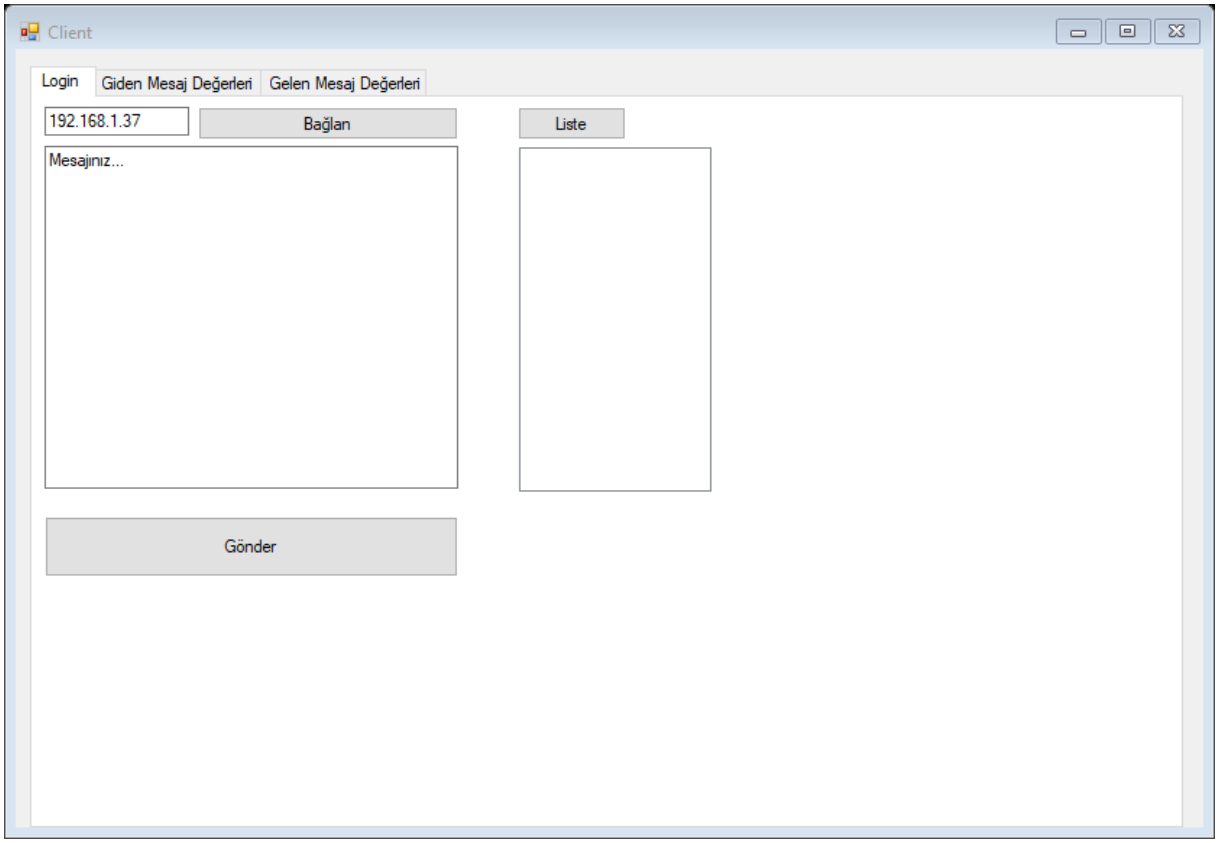
p ve q sayılarının çarpımı ile elde edilen N değeri 802 (2664 bit), p ve q sayılarının 1 eksiklerinin çarpımı ile elde edilen $\varphi(n)$ sayısı da yine 802 (2664 bit) basamaklıdır.

Uygulamada, mimariye uygun olarak, imzalama işlemi gerçekleştirilmiştir. RSA ile imzalama işlemi gerçekleştirebilmek için yine public - private (açık – gizli) anahtar ikililerine ihtiyaç duyulmuştur. Alıcıya gönderilen mesaj hem cihaz hem de gönderici tarafından imzalanacağından dolayı, 2 adet cihaz 2 adet kullanıcı tanımlanmıştır. Bu tanımlamalar içinde imza için 4 adet, imza kontrolü için 4 adet anahtara ihtiyaç duyulduğundan 8 adet anahtar tanımlanmıştır. Bu anahtarlar 801 – 802 (2659 bit - 2664 bit arası değişen) basamaklı olarak seçilmiştir.

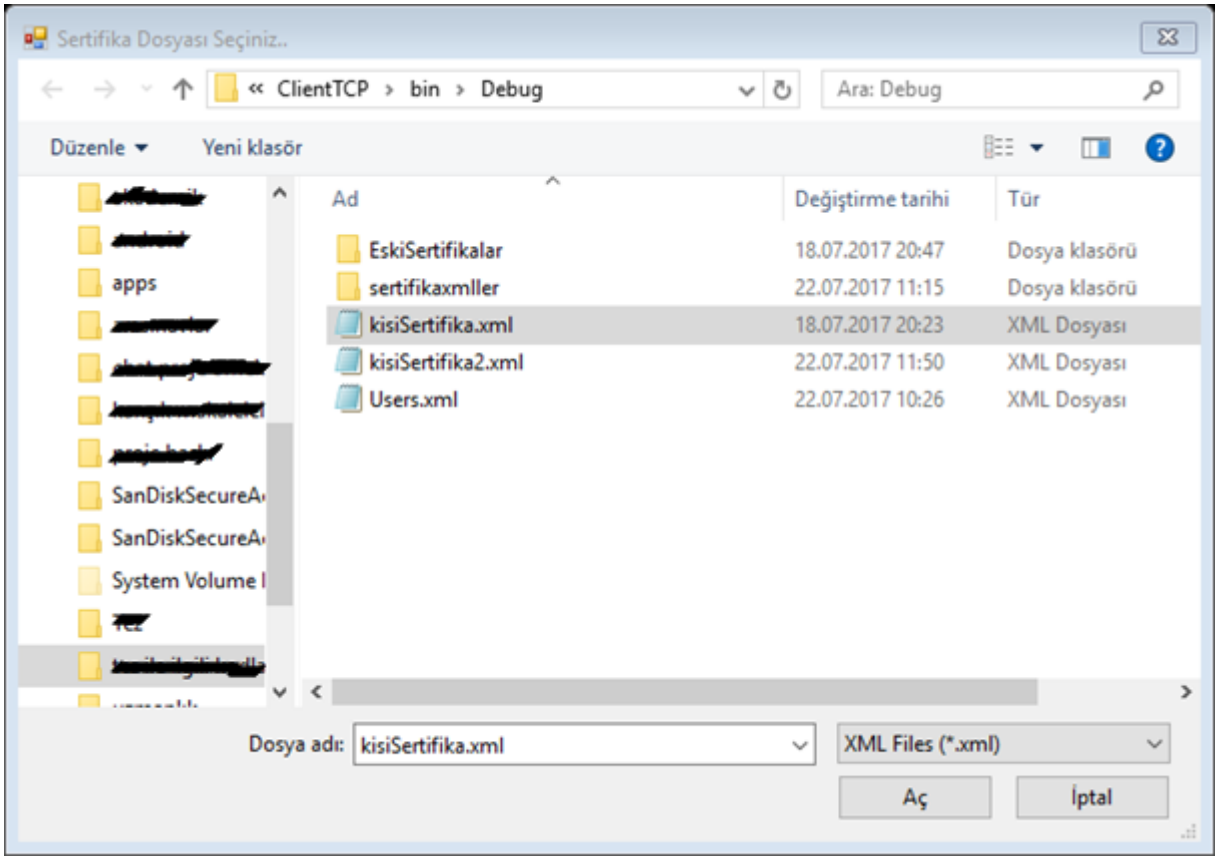
Algoritma gereği $\varphi(n)$ den küçük $\varphi(n)$ ile aralarında asal olan sayılar tespit edilerek aralarından rastgele biri e (şifreleme – imza kontrol anahtarı) olarak seçilir. Seçilen e sayısı da yine 802 basamaklıdır. e Şifreleme anahtarının Mod N e göre çarpma işlemine göre tersi olan d (Şifre çözme – imzalama anahtarı) sayısı hesaplanır ve bu sayıda yine 801 basamaklıdır.

Asimetrik algoritmalardan en çok tercih edileni olan RSA'nın güvenliğini sağlamak için çok büyük asal sayılarla çalışılmalıdır. RSA Security firması 1024 bitlik RSA anahtarının 2006 – 2010 yılları arasında kırılabileceğini bildirmiş, Ayrıca 2048 bitlik anahtarla yapılan şifrelemelerin 2030 a kadar kırılamayacağını, eğer 2030 sonrasına uzanan bir güvenlik isteniyorsa 3072 bitlik anahtar kullanılması gerektiğini açıklamıştır. [30]

Şekil 11.1. deki ekranda Sunucunun ip adresi yazıldıktan sonra Bağlan düğmesine tıklandığında şekil 11.2 de görünen kişi sertifikası seçme penceresi açılır ve sertifika seçilerek sunucuya gönderilir. Kullanıcı sertifikası üçlü sertifika mimari yapısına göre sunucunun açık anahtarı ile şifrelenerek sunucuya gönderilir ve sunucuda açılır.

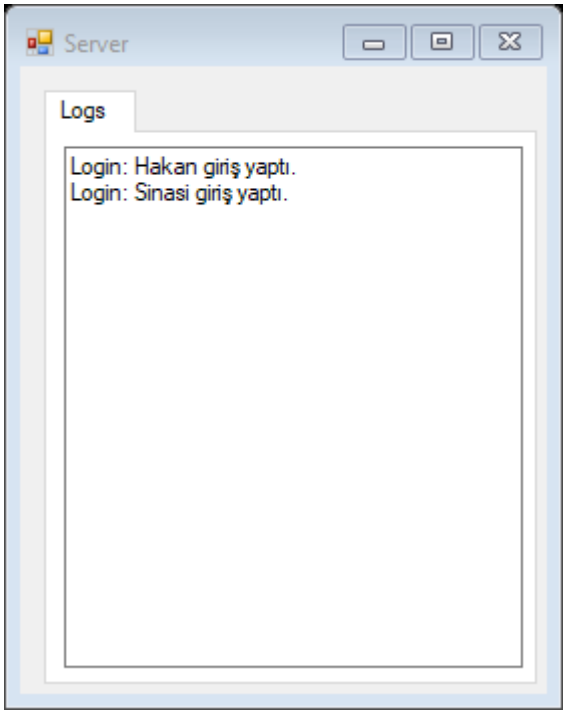


Şekil 11.1. Kullanıcı ekranı



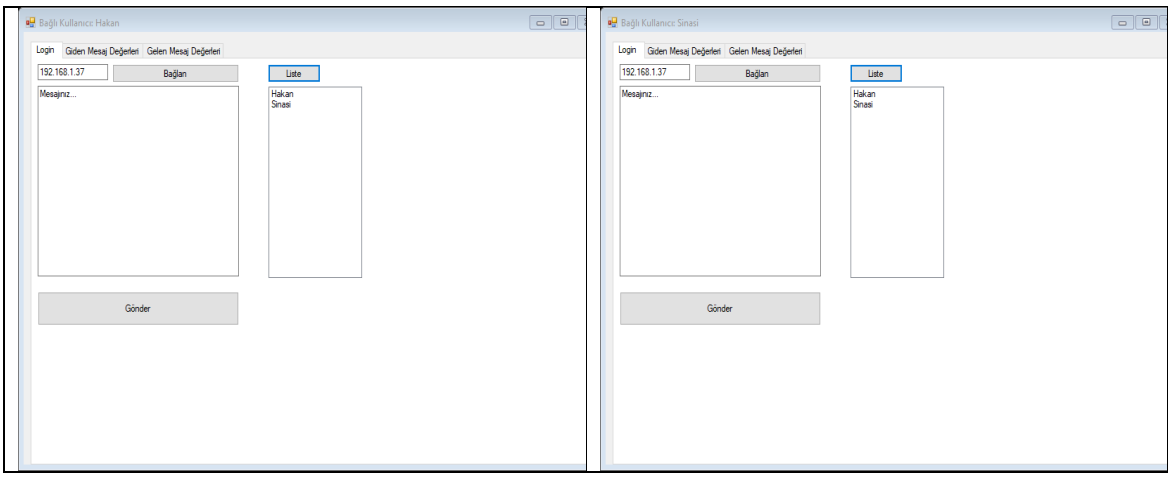
Şekil 11.2. Kullanıcı sertifikası seçme ekranı

Kişi sertifikasının onaylanmasından sonra kullanıcıdan herhangi bir işlem yapılması beklenmeksizin cihaz sertifikası istemciden çekilir. Bu sertifika kullanıcının açık anahtarı ile şifrelenmiştir ve bu anahtar az önce gönderilen kişi sertifikası içinde bulunmaktadır. Bu anahtar kullanılarak cihaz sertifikası açılır ve onaylanarak sisteme giriş sağlanır. Eğer kişi sertifikasındaki cihaz sertifikası anahtarı, cihaz sertifikasını açamıyorsa zaten cihaz ve kişi eşleşmeyeceğinden sisteme girişe izin verilmeyecektir.



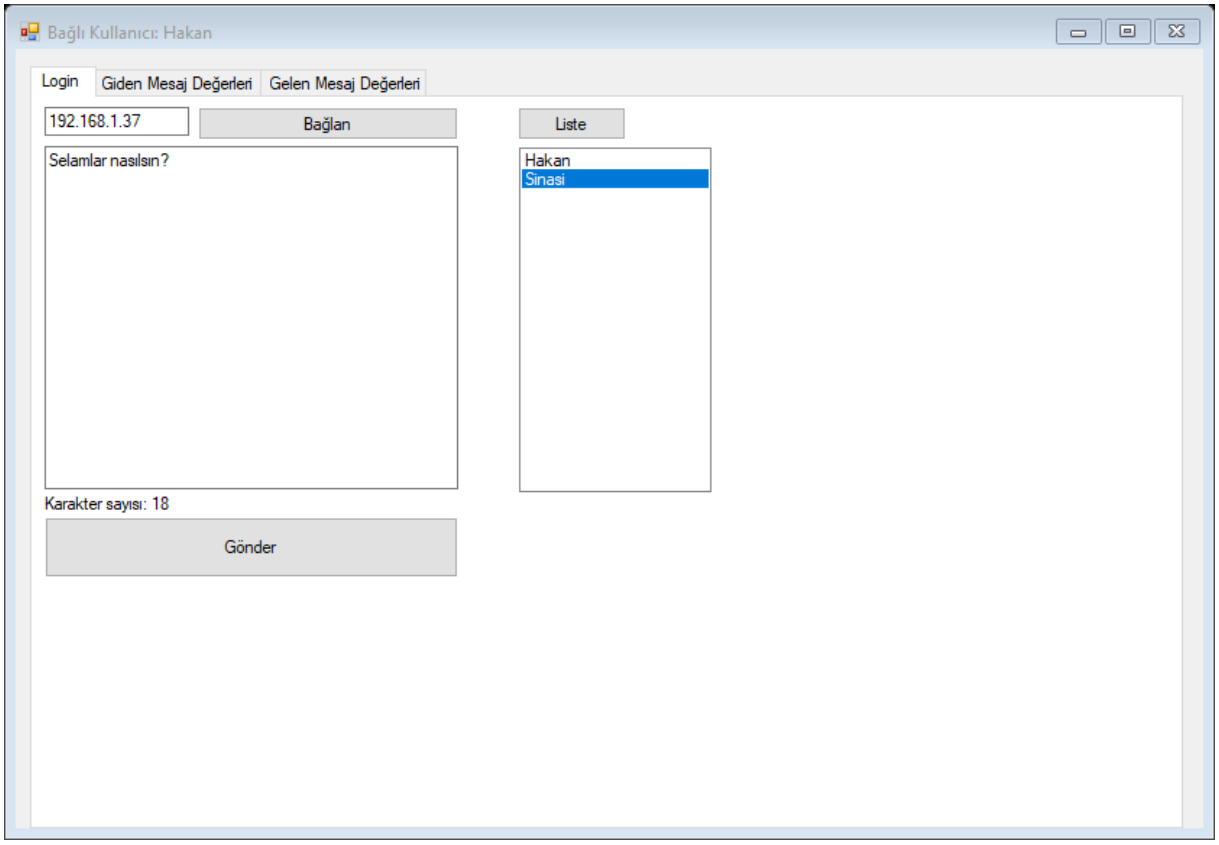
Şekil 11.3. Kullanıcı sertifikalı onaylandıktan sonra kullanıcıların sisteme giriş yapmış olma durumu

Liste düğmesine tıklanarak sistemde olan kişiler listelenir.



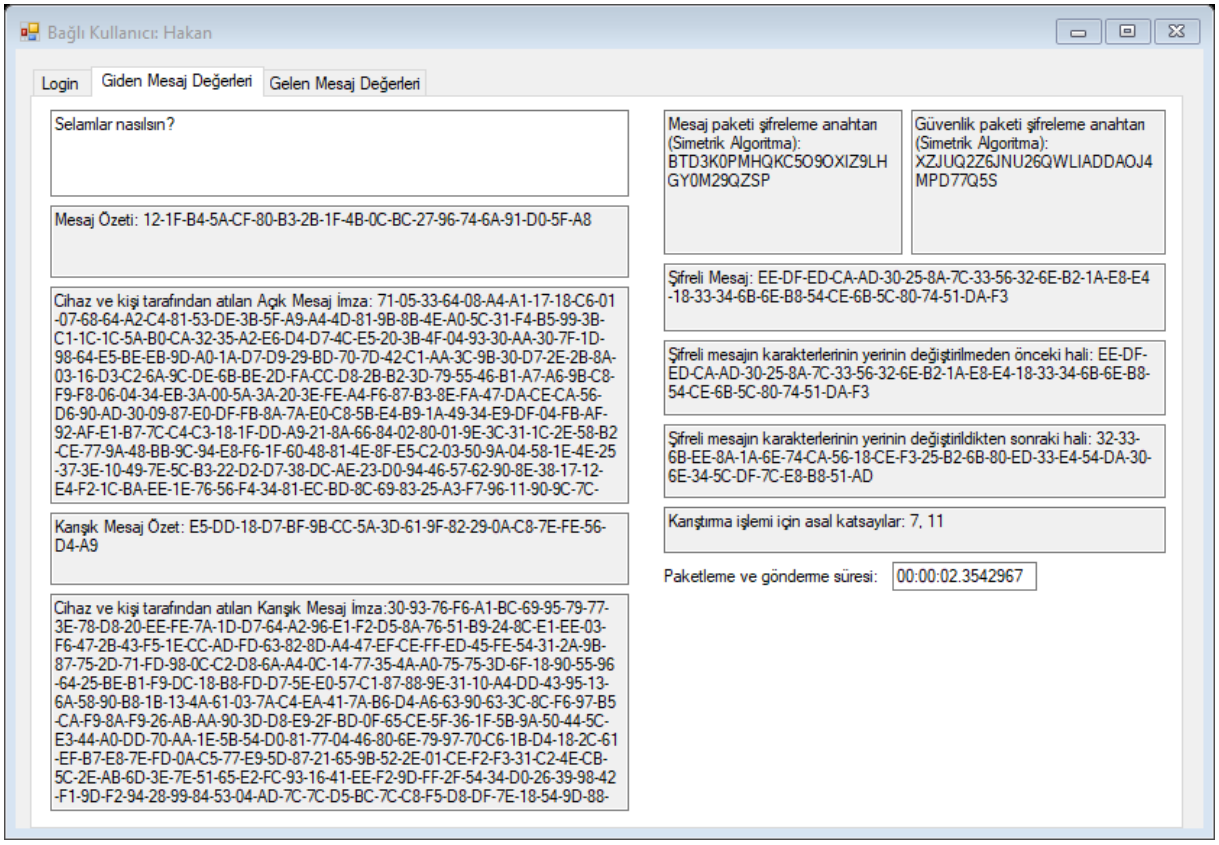
Şekil 11.4. Sistemde bulunan kullanıcıların listelenmesi

Mesaj yazılıp kişi seçilerek gönder tuşu ile mesaj gönderilir. Mesaj gönderilmeden önce hibrit mimari içindeki işlemlerin tamamı uygulanarak özet değerleri imzalar hesaplanır, paket oluşturulur ve alıcıya gönderilir.



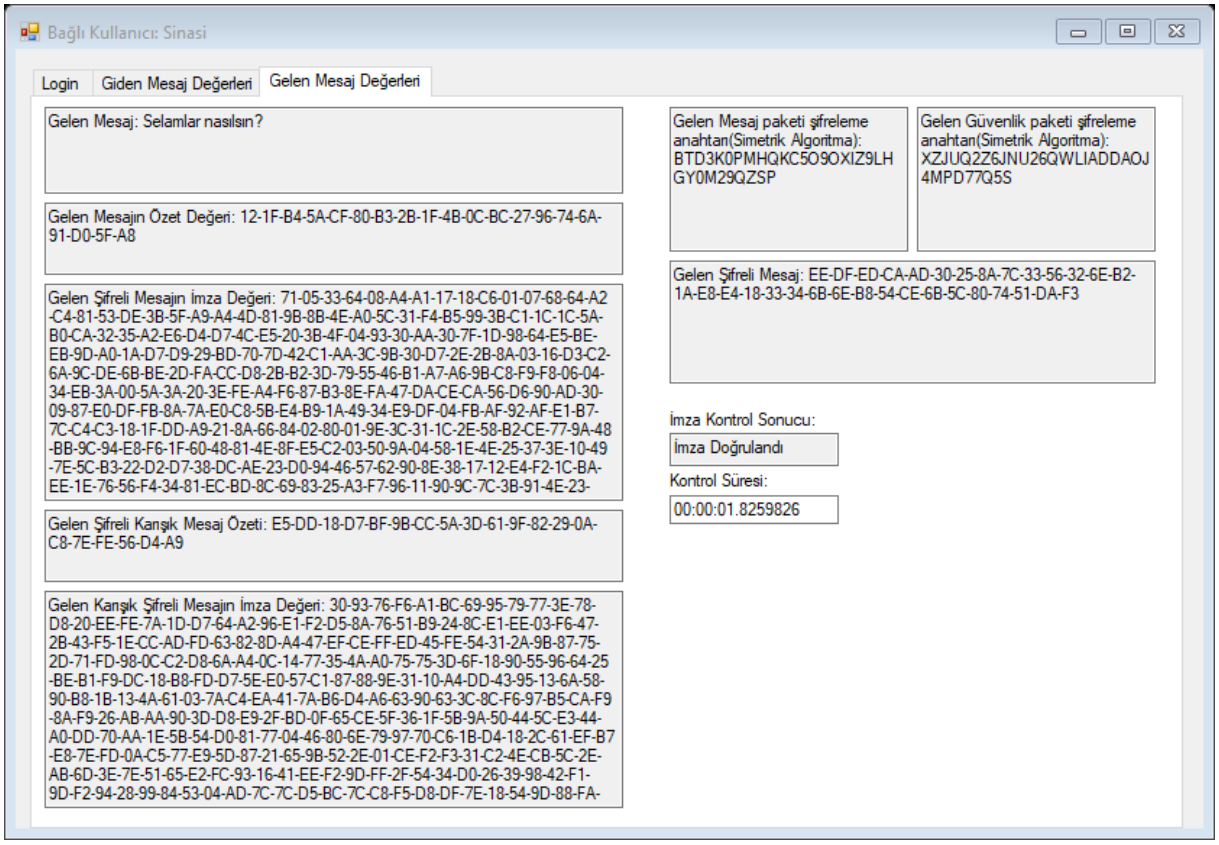
Şekil 11.5. Birinci kullanıcının ikinci kullanıcıya mesaj göndermesi

Gönderilen mesaja ait özet ve imza değerleri şekil 11.6. de olduğu gibi ‘Giden Mesaj Değerleri’ sekmesinde görülebilir.



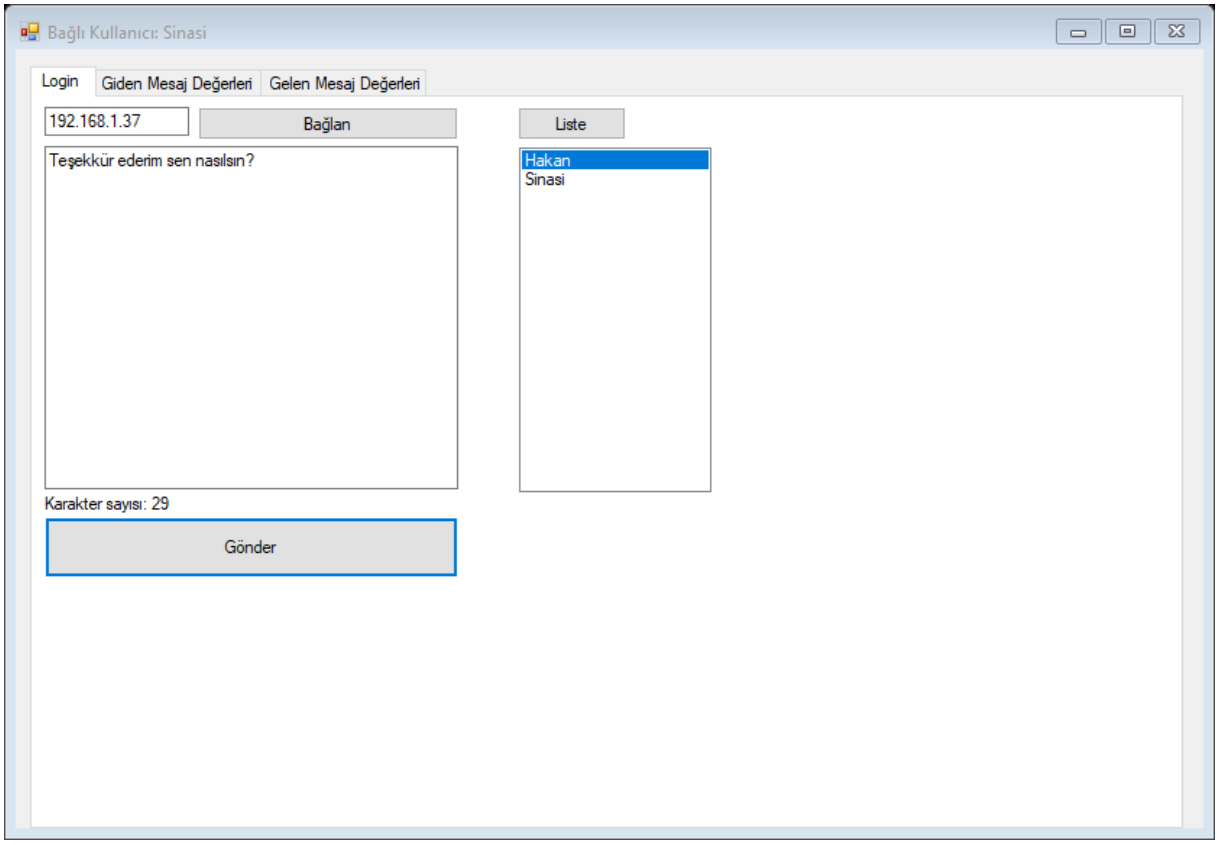
Şekil 11.6. Gönderilen mesajın imza-özet değerleri ve paketleme süresi

Mesaj alıcıya ulaşınca mimariye uygun olarak, paketlerin ayrılması şifreli mesajın çözülmesi, mesaja ait özet ve imza değerlerinin tekrar hesaplanarak gelen güvenlik paketi içindeki özet ve imza değerleri ile karşılaştırılarak doğrulama işlemi gerçekleştirilir. Bu hesaplamaların sonucu alıcının kullanıcı ekranında ‘Gelen Mesaj Değerleri’ sekmesinde şekil 8 de olduğu gibi görülebilir.

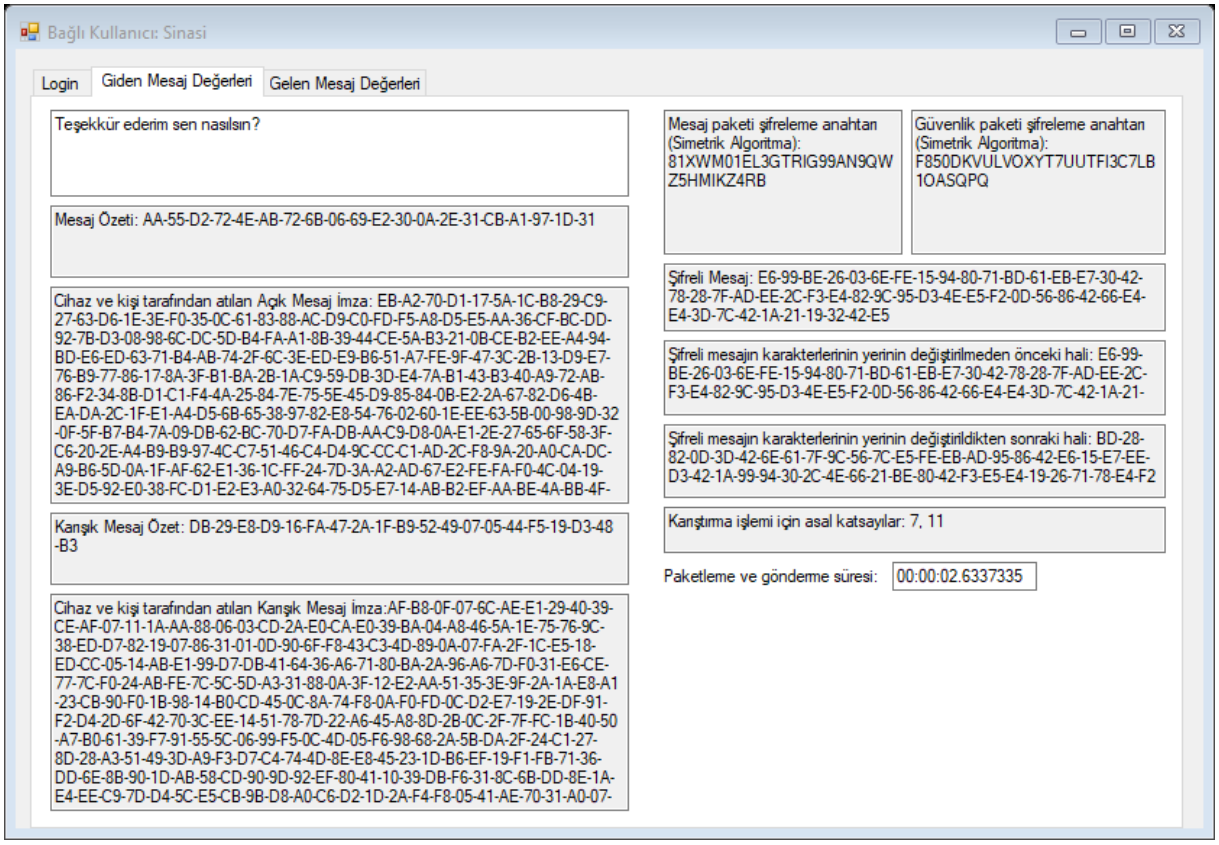


Şekil 11.7. İkinci kullanıcı tarafından alınan mesajın çözümlendikten sonraki imza-özet değerleri ve kontrol süresi

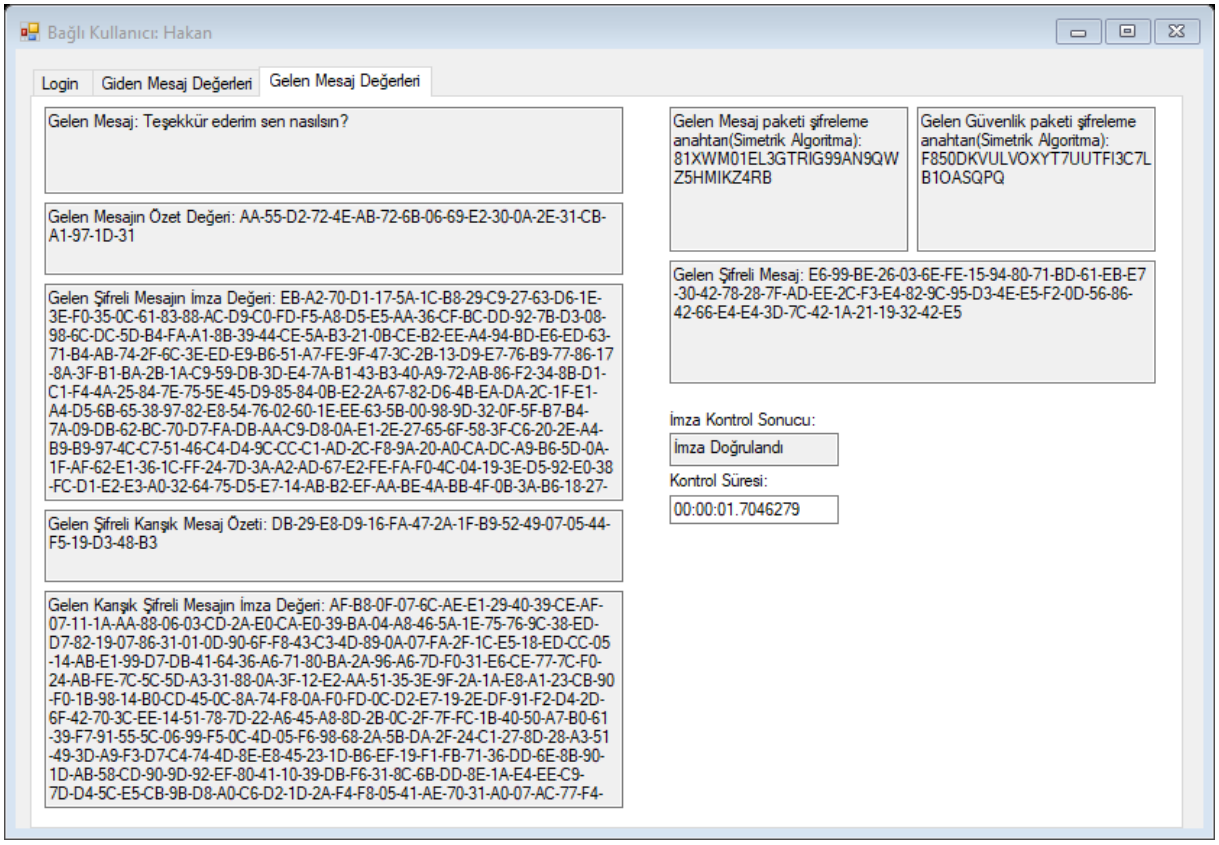
Benzer işlemler kullanıcıların durumları değişerek de gerçekleştirilir. (Şekil 11.8.- 11.9. -11.10)



Şekil 11.8. İkinci kullanıcı tarafından birinci kullanıcıya mesaj gönderilmesi



Şekil 11.9. İkinci kullanıcı tarafından gönderilen mesajın imza-özet değerleri ve paketleme süresi



Şekil 11.10. Birinci kullanıcı tarafından alınan mesajın çözümlendikten sonraki imza-özet değerleri ve kontrol süresi

Bu işlemler sırasında asimetrik algoritma anahtarları olarak 802 basamaklı asal sayılar kullanılmış ve ekranlarda gözüken süreler elde edilmiştir.

	RSA anahtar uzunluğu Şifreleme Anahtarı: 2662 bit Şifre Çözme Anahtarı: 2658 bit	
	Şifreleme Süresi	Şifre Çözme Süresi
128 bit 1410*8=11280	2.187 saniye	1.575 saniye
256 bit1426*8=11408	2.352 saniye	1.591 saniye
512 bit1458*8=11664	2.234 saniye	1.613 saniye

İletişim sırasında güvenlik paketi, mesaj paketi ve şifre paketi birlikte gönderilmektedir. Bu üç paket veri paketini oluşturur. Veri paketi alıcıya ulaştıktan sonra alıcı tarafı ilk iş olarak paketleri birbirinden ayırır. Sonra şifreyi çözerek orijinal mesajı elde eder ve anahtarları kullanarak imzaları kontrol eder. Yani gönderici tarafında imza işlemlerinin tamamı alıcı tarafında tekrar yapılarak kontrol sağlanır. Veri paketi içinde mesaj haricinde sabit büyüklükte bir sürü veri vardır. Kullanılan anahtarlara bağlı olarak bu sabit büyüklük değişecektir. Örneğimizde 11.152 bittir. Yani

128 bitlik bir mesaj için işlenen veri 128+11.152=11.280 bittir. Yani alıcı, 128 bit yerine 11.280 bitlik veri üzerinde çalışmaktadır.

P=199999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999
99999999999999999999999999999999999999999999999999999999999 – 402 basamak – 1334 bit[59]

Q=244674769216829134384881927892398754540525363525493778370176900
0376739214211560388139384104131071632781621361335444952091589464923694
0295705032203772143700063945600771074430370325781739206543222519232991
3856366128220847391636867576496273816819500585939328376720113657188244
4415341372938189320861369266594079428428246207933531214241183420041649
5641677380129898882762716949168399500016681107627423520001 – 401 basamak – 1331 bit[59]

N=489349538433658268769763855784797509081050727050987556740353800
0753478428423120776278768208262143265563242722670889904183178929847388
0591410064407544287400127891201542148860740651563478413086445038465982
7712732256441694783273735152992547633639001171878656753440227314376488
8830682745876378641722738533188158856856492415867062428482366840083299
1283354760259797765525433898336799000033362215254847040001755325230783
1708656151180721076012454594746364745062216298230999623260785788439611
8606158958689283672183786386645550479084105350763059704294967796227856
2999360543992289255696296742182607934567774807670086143633871779152608
3631324235037261831804994140606716232798863428117555584658627061810679
1386307334059205715717537920664687857588165799583504358322619870101117
237283050831600499983318892372576479999 – 802 basamak – 2664 bit

$\varphi(n)$=48934953843365826876976385578479750908105072705098755674035
38000753478428423120776278768208262143265563242722670889904183178929847

3880591410064407544287400127891201542148860740651563478413086445038465
9827712732256441694783273735152992547633639001171878656753440227314376
4888830682745876378641722738533188158856856492415867062428482366840083
2991283354760259797765525433898336799000033362215254847039999510650461
5663417312302361442152024909189492729490124432596461999246521571576879
2237212317917378567344367572773291100958168210701526119408589935592455
7125998721087984578511392593484365215869135549615340172287267743558305
2167262648470074523663609988281213432465597726856235111169317254123621
3582772614668118411431435075841329375715176331599167008716645239740202
2344745661016632009999666377847451529600000 – 802 basamak – 2664 bit

E(Şifreleme)=1803745922815875327103769727184895015999358106175112941906774207517300714284080866021027799762513551651076235817703911292488919120397089448592078613954300891517796522838811539268980166400241654624224056308763595415273026321242528462982309187328290719744204422251090949411698542672483029897365823054964075378398102272226100503390858206466244130840745962076103263038815789127390371523740365868565485512212102199299727971906744397564833607901102332974866804471354149167998502768687113495101811666836144081888184205645398514705713246426548791650622591246069258166195690535051058669471804633520057102291235313566722917474538930047495095106945855226769822776497955062358244930193716307035035791022154802560730099626250917573262787509393980942556555439839983282347767048311295644078111915586532104787982151973 – 802 basamak – 2662 bit

D(ŞifreÇözme)=1072403574649467268876096207194560859766050122885100104427495015046456558647192857581844707027689554621147174673823759352874393311993410285115709736639160171518307774550899911690805937903265055347546351303779822613437119467064947944935115302012960069744364790191683144382508231483068404537834172898820794658877895972466298506601863943281712637603557415342102398774559384991966998155326999424555683784638802529327660012914305533613912136411277407016085469393524173172787862328612586196460525477039941392035443876585485888051343231257274129263783158028820527518128177407850716456971600244453890066350987778129678054607252287297221190390067619364641836179693248481865341867239197999392

595246136115262490448330708305816797634100096974437693188065101674776310328842106121358251328948370687561246214639090 37 – 801 basamak – 2658 bit

E(Cihaz)=48687416738056686639297730344013178394071136251510718780892370067882239739332325415507551444338725419789885046441112040204848709357928864140624468711164534904147935015258646147681173169285597462842313857217809960923746394112915659391818688781226805043208192996609862603411140012223232906620558132069300114163052499788990623447788078469434505855376310358715052151715786313524459318744184001060625733807784851277112566324510289024956024304433227135907790772869801317740541946138861435311112436893617932740208733554903216617025683534192211260616891009838706729529697072020540697962392145670906600296317022037024495759674790963101284012843218688204409737784507868394432520178782355385684784484732095712639758139155114189445965357730154750016798392568173571410095273055048799727885920011348214211895282488 23 – 802 basamak – 2664 bit

D(Cihaz)=90877392771522695492916190552416094948171964065485611189460161410205214623269913337213389095136258927622108334593756843898934221955426632103002781606296209796409451551030423433083391980658750814027376039096871960951836144973081350876015266401230292391908051277418774140983626528722593053202421281685091753123940895233503741992194209727056881849111348333477335712720603387867775047029110462864247922284710880116394612163214645956764697387946569545345221947720702100308924851133827778894088982798360977257966478829264583113689831260171611540057034022490659415868630640948528654896460622140092804758411260805008196141978192596280569649078103166790827440716159156771312664340102389761914190568997353640917622665358896291407415541086132457359512787816040244376421388178500765491072113708311930166711521888 7 – 801 basamak – 2661 bit

E(Kişi)=643318275279071825518011162862332514844191595832235717531795165537640972182490469192753850205561927096561970266935906507055010653806179597008545374987145345542319051539105457009185303886425275833119080240413789869474254300802142299626675020712254108704016631774450329422651028968621758032479858617185927682789526201266138352136006621781813

037518810675772891396219182469422076424924165441753734626376451844041350183113872282387428346856090863353729513123367309640647201746205799814562567913327624198143048299048720429276488049675451570675238149743852969209487996914661593228395870632529324139089389272131937046709802183823328028924689835720413316662563961118761116366867575407411647813905568053791598255753521653108358386596933497083572907023683761539130265492065289634361029423655536234282658670790393 – 801 basamak – 2661 bit

D(Kişi)=16318638478963650659348602698339247182288624039337924626519404942857167699942122609482792284083311665152730298488999422532446386580476072636086897848959379048133625470458514396805462699464099316656944044244927437239423002383978439792806126273353274359883246232311756256929354411384754374850639554407834217153509471862572051791685760406691118194037357712021068835084011627803812291698637318439584508970321138415328399274651681484841464756756224074911089172197204479323837380793486618723864740212115303842250974024651298094854844061688570515953470589228830132882010293237989148218378382821274058806210201956282002605407137939816111561262379995809569156743458338791494405854713735130728821722472749755551591413152021866709732303108838009365909418960663243500295283182608000490945793532423650871961490532573 – 801 basamak – 2659 bit

E(cihaz2)=37388149527829236229988882972284730417602383019896948432394487225652437664221912723545397660189924192547911388847708381415300687293986719155829219538947337239876757121013064568463985802102355717975165959677792736299640137066892093091038028717431222337508246279794644134724504377017804856570320587025065699199572293662604393067419883788104406802987110455091669595724618764294940020572933645863702360166213417791728533357210293108399417269509273256820923824932268425691828249518332612943916359995389579381238236037675170097646738349453741587536831519422584338367189537277812728619989118369242899494825310791528791305479281388794543389849326146552834616770675291092768200994857827256118571873853848609002263304907768274599757901044878554205242401024108682487587096339343708020916903140143734021318750480612 – 802 basamak – 2663 bit

D(cihaz2)=290039068744205028884442912800325186742860440208001622004
873569716461638299737383348585963423061014681532903539085851049897772 3
946027641102031610658674854857644512473956775910506406339658132156682 5
553411282924722485040308445352192851065484914307956071875924638281388 0
386438056330525947502853719377844194184067956772222618367864760996394 8
550402016590799658038646072706813091162862162134656012229262813308875 6
278017592061081681692641944540572125292441231113332666224316486234880 6
614745149569843450720242048788386565213721075978090657849345702620707 8
821661763896599921624200531467040305096807108755940143273788244697830 6
209578209333985408726118179926032681223540044567554753753829989881379 5
839355102979859401812450067702461513502111860807217985935543832548699 3
072944271781423973950348006935192431520419541 – 802 basamak – 2663 bit

E(kisi2)=444128290978702035654959303570200401255008262541132981401 1
525060187881515652585509015650590311589582986891566270467019457504055 5
714450815513151631712982532381216514251411922748937450544008395922374 9
780958026122230207020748721456511346752184717670645133629743715964408 9
706922347101241511120514616831241478915030751077626728463383412512783 3
816998343115648959023132129097218339838601699491664196082991885523296 6
218321624898742809093154826466455631779199030260170056785984652437589 2
209739004280951672458444303203768185680892107925885426265820182794614 4
374595155068833490271062094560952270762177598471754092112942786988065 1
574848663585415132298347248056346836653521049530037925929734309916354 1
820807252357759315638088555823240194103995754762281573813667211536966 9
755855376113139903745580212221391686745073 59 – 802 basamak – 2664 bit

D(kisi2)=277713923483185483441304998868224937672669499807652130919 6
416362130051136506497826569737527532698301319041440269388091495973630 9
703251904958398601862591613393600013702441467775029271825127039103949 3
927897843998905908013198462472440135617549937450349898982635248429406 9
529660271922081711731252138329323044639542215887396170317538687931088 7
493523827063660388780983595910007353928179968119458932416658201139604 2
945555131629219221611404904359210786222400126367353706227330588045512 9
803242672064428839389320168599364674843023678657726575043706766232674 9

3148256406314368157063723083615984881148076158852684637850375783830180
9511290945756508009766552225393475958409968960986119552352957063390175
1808008568425839150007394562364526140036526221364261817149117906276817
561275904106988058423113292777110432934670 39 – 802 basamak – 2663 bit

12.TARTIŞMALAR

İletişim güvenliği artık açık verinin şifrelenmesinin ötesine geçmiştir. Verinin şifrelenmesi güvenlik için tabi ki vazgeçilmezdir. Bununla birlikte göndericinin doğrulanması, gelen verinin gönderilen veri ile aynı olduğu yani değiştirilmemiş olmasının garanti edilmesi de iletişim güvenliğinin en önemli parametreleri haline gelmiştir.

Bu çalışmamızda iletişim güvenliğinin farklı özellikleri için veri güvenliği mimarisi oluşturulmuş ve performans testleri yapılmıştır.

Veri güvenliği özelliği için simetrik veya asimetrik şifreleme algoritmaları kullanılabilir. Performans isteniyorsa simetrik algoritmalar, daha fazla güvenlik isteniyorsa asimetrik algoritmalar tercih edilir. Simetrik algoritmalar performans sunarken anahtar yönetimi noktasında sorunludur. Asimetrik algoritmalar simetrik algoritmalara göre daha yavaştır ve daha fazla işlem içerir fakat anahtar yönetimi başarılıdır. Asimetrik algoritmalar küçük veri bloklarının şifrelenmesi için tercih edilmelidir. İşte bu iki yapının birbirlerinin eksiklerini gidermek için bir araya gelmesiyle melez (hibrit) yapılar oluşur.

Melez mimarilerde açık verinin şifrelenmesi işlemini simetrik algoritmalar ile çözerken, simetrik algoritmanın kullandığı anahtar da asimetrik algoritmalar kullanılarak şifrelenir ve şifreli veri ile alıcıya gönderilir.

Metnin şifrelenmesinin yanı sıra doğrulanması da gereken durumlar olabilir. Doğrulamanın amacı, veri alıcıya iletilirken yolda herhangi bir değişikliğe uğramadığını göstermek için yapılır. Örneğin veri iletimi sırasında kötü niyetli bir kişi araya girip veriyi değiştirerek, alıcıya değiştirilmiş verinin iletilmesini sağlayabilir. Bu saldırı türüne ortadaki adam saldırısı denir. Bu gibi durumlar için verinin doğrulanması gerekebilir. Veri doğrulama için özet (Hash) algoritmaları kullanılır. Asıl verideki en küçük değişiklik, veri özetinde büyük değişikliklere sebebiyet verir. Tek yönlü fonksiyonlar kullanıldığından özet kullanılarak asıl veriye ulaşılamaz. Dolayısıyla alıcı asıl verinin özetini alarak kendisine gönderilen özet değeri ile karşılaştırmalıdır.

Kişi doğrulamanın en güvenilir yolu da e-imza kullanılmasıdır. E-imza asimetrik algoritmalar ile oluşturulur. Tekniği son derece basittir. Asimetrik algoritmalarda gizli-

açık anahtar ikilisi birbirine matematiksel olarak bağlıdır, dolayısıyla biri değiştiğinde diğeri de değişir. Asimetrik algoritmalarla imza oluşturmak için iletilecek olan veri göndericinin gizli anahtarı ile işleme sokularak imza değeri oluşturulur ve alıcıya bu imza değeri gönderilir. Alıcıda da zaten göndericinin açık anahtarı bulunduğundan bu anahtar kullanılarak algoritma çalıştırılır ve imza doğrulanır.

"Madem daha iyi anahtar yönetimi sağlıyor, neden asimetrik algoritmalar tercih edilmiyor?" sorusu akla gelebilir. Bu sorunun cevabı tabi "Asimetrik Algoritmalar" bölümünde verilmiştir. Fakat mimari içinde özellikle imzalama işlemi için kullanılması kaçınılmazdır. Tez çalışması sırasında asimetrik algoritmalardan RSA için bir takım testler gerçekleştirilmiştir. RSA algoritması Güvenlik için p,q ve anahtarlar değerlerinin büyük seçilmesi ile yüksek güvenlik sağlar. "Mobil Cihazlarda RSA Algoritmasının Performans Optimizasyonu" isimli çalışmamızda RSA algoritmasının mobil cihazlardaki performans testleri gerçekleştirilmiş ve sonuçlar elde edilmiştir. RSA algoritmasının en zorlu tarafı büyük sayılarla işlem yapmaktır. Çalışmamızda 420 basamaklı (1395 bit) sayılar kullanılmış ve bu sayılarla modüler üs alma işlemleri gerçekleştirilmiştir. Çok küçük donanımlı mobil cihazlarda bile çok kısa sürelerde işlem yapılabilmiştir. Sürenin kısa olması doğrulama işlemlerinin hızlıca yapılabileceği anlamına gelir. [58]

Mimari de simetrik ve asimetrik algoritmalara ek olarak, verinin şifrelenmesinin ardından karıştırma işlemi uygulanmıştır. Karıştırma işlemi için yaygın olarak Fisher/Yates ve Knutt/Durnstenfeld algoritmaları kullanılır. Bu algoritmalar "Karıştırma Algoritmaları" Başlığı altında detaylı olarak anlatılmıştır. Karıştırma algoritmalarındaki rastgelelik, karıştırılmış olan verilerin geri dönüşünün olmamasına yol açmaktadır. Bu mimarimizde doğrusal lineer haritalama işlemi karıştırma işlemi için algoritma geliştirilmiş ve böylece veri paketi küçültülmüştür. [60]

Teknolojideki gelişmeler cihazların hızlanmasını sağlamıştır. Artık büyük sayılarla da hızlı bir şekilde işlem yapılabilmektedir. Bu gelişmeler asimetrik algoritmaların rahatlıkla kullanılmasını sağlamaktadır. Ayrıca yazılım dünyasındaki çalışmalar şifreleme işlemleri için kullanılan algoritmaların, hazır kütüphane olarak yazılım dillerinde kullanılabilmesinin önünü açmıştır. Yine de asimetrik algoritmaların

daha performanslı çalışabilmesi için büyük sayılarla çarpma, üs alma gibi işlemlerle ilgili çalışmalara devam edilmektedir.

13.KAYNAKLAR

[1] https://tr.wikipedia.org/wiki/ebced_hesabı

[2]Yerlikaya, T., Yüksek Lisans Tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü 2002

[3] Yıldırım, H. M., Bilgi Güvenliği Ve Kriptoloji, *Uluslararası Adli Bilişim Sempozyumu, 31 Mayıs – 01 Haziran 2014*

[4] Kondo, T. S., Mselle, L. J., An Extended Version Of The Polybius Cipher, *International Journal Of Computer Applications (0975 – 8887) Volume 79 – No 13, October 2013*

[5]Klima, R., Sigmon, N.,Using Graphs To Break Vigenère Ciphers,*24.International Conference On Technology İn Collegiate Mathematics S094,March 22-25, 2012*

[6] Kerckhoffs, A.,La Cryptographie Militaire**,***University Microfilms, 1978.*

[7] 2015-2018 Bilgi Toplumu Stratejisi Ve Eylem Planı, Yüksek Planlama Kurulu'nun 24 Şubat 2015 Tarihli Kararı İle Kabul Edilmiş Ve 6 Mart 2015 Tarihli Resmi Gazete'nin Mükerrer Sayısında Yayımlanmıştır. Sf.29-30

[8] Bozkurt, F.,Elektronik Güvenlik, Şifreleme Teknikleri Ve Algoritması Açık Olan Şifreleme Teknikleri,Dokuz Eylül Üniversitesi http://courses.cs.deu.edu.tr/cse428/assignment1/ferhat bozkurt_2001510051.doc

[9] Sağıroğlu, Ş., Alkan, M., Bilgi Güvenliği Bilimi(Kriptoloji), *Her Yönüyle Elektronik İmza, Grafiker Yayınları, Ankara, 21-50, 2005.*

[10] Aslandağ, K., Bilgi Güvenliği Kavramı Ve Bilgi Güvenliği Yönetim Sistemleri İle Şirket Performansı İlişkisine Dair Bir Uygulama, *Yüksek Lisans Tezi, Gebze Yüksek Teknoloji Enstitüsü, Sosyal Bilimler Enstitüsü Strateji Anabilim Dalı, Gebze 2010.*

[11] Uğur, A., Uzaktan Erişimli Kriptografik Güvenli Haberleşme Protokolü, *Yüksek Lisans Tezi, Pamukkale Üniversitesi Fen Bilimleri Enstitüsü, Denizli 2005.*

[12]Coşkun, A., Ülker, Ü., Ulusal Bilgi Güvenliğine Yönelik Bir Kriptografi Algoritması Geliştirilmesi Ve Harf Frekans Analizine Karşı Güvenirlik Tespiti,*Bilişim Teknolojileri Dergisi, Cilt: 6, Sayı: 2, Mayıs 2013 31http://dergipark.gov.tr/download/article-file/75327*

[13] Schneier , B., Applied Cryptology, Second Edition: Protocols, Algorithms, And Source Code İn C , *Wiley Publishing, (1996)*

[14]Stalings , W., Crytography And Network Security Second Edition, *Prentice Hall, 1997*

[15]Aslan, F. Y., Sakallı , M. T., Aslan , B.,Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi, *XIV. Akademik Bilişim Konferansı Bildirileri 1 - 3 Şubat 2012*

[16] Announcing The Advanced Encryptıon Standard (AES), *Federal Information Processing Standards Publication 197, November 26, 2001*

[17] Daemen , J., Rıjmen ,V., AES Proposal: Rijndael (1998) http://citeseerx.ist.psu.edu/viewdoc/versions?doi=10.1.1.36.640

[18]Sakallı, M. T., Buluş ,E., Şahin, A., Büyüksaraçoğlu, F., Akış Şifrelerinde Tasarım Teknikleri Ve Güç İncelemesi, *IX. Akademik Bilişim Konferansı Bildirileri 31 Ocak - 2 Şubat 2007*

[19]Diffie, W., Hellman, M., New Directions İn Cryptography, *Ieee Transactıons On Informatıon Theory, Vol. It-22, No. 6, November 1976*

[20] Yerlikaya, T., Buluş, E., Buluş N., Asimetrik Şifreleme Algoritmalarında Anahtar Değişim Sistemleri, *Akademik Bilişim'06*

[21] A Note On Extended Euclid's Algorithm, https://arxiv.org/abs/1607.00106v1

[22] Arda, D., Bulus, E., Çin Kalan Teoremini Kullanan Bir Gizlilik Paylasım Şeması, *IV. İletişim Teknolojileri Ulusal Sempozyumu, 15 EKİM 2009*

[23] Rivest, R.L., Shamir, A., Adleman, L., A Method For Obtaining Digital Signatures And Public-Key Cryptosystems, *Communications Of The ACM, 1978. 21(2): P. 120-126.*

[24]Öztürk, V.,Elektronik Belge Yönetim Sistemlerinde Bilgi Güvenliği, *2.Ulusal Yönetim Bilişim Sistemleri Kongresi, 8-9-10 Ekim 2015, Erzurum*

[25]http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf(Ziyaret Tarihi:03.07.2017)

[26] Güvenoğlu, E., Esin, E. M., Knutt / Durstenfeld Shuffle Algoritmasının Resim Şifreleme Amacıyla Kullanılması, *Politeknik Dergisi Cilt:12 Sayı: 3 S.151-155, 2009*

[27] Hazra, T. K., Ghosh ,R., Kumar, S., Dutta, S., Chakraborty, Dr. A. K., File Encryption Using Fisher-Yates Shuffle, *Computing And Communication (IEMCON), 2015 International Conference And Workshop 2015*

[29] *Kids, Code & Computer Science.* Dec2016, Vol. 4 Issue 3, P22-24. 3p

[30] h*ttps://En.Wikipedia.Org/Wiki/Key_Size*

[31] Yerlikaya, T., Gençoğlu, H., Emir, M. K., Çankaya, M., Buluş, E., RSA Şifreleme Algoritması Ve Aritmetik Modül Uygulaması, *İstanbul Aydın Üniversitesi Dergisi, Yıl:3, Sayı:9, Sayfa: 95-104, 2011*

[32] Cook, S. A., On The Minimum Computation Time Of Functions, *Phd Thesis, Harvard University Department Of Mathematics. Http://Cr.Yp.To/Bib/1966/Cook.Html, 1966*

[33] Toom , A. L., The Complexity Of A Scheme Of Functional Elements Realizing The Multiplication Of Integers,*Soviet Math (Translations Of Dokl. Adad. Nauk. SSSR), Vol. 4, No. 3, 1963.*http://www.de.ufpe.br/~toom/articles/rusmat/multipli.pdf

[34] Knuth, D. E., The Art Of Computer Programming, Volume 2, Seminumerical Algorithms, 3rd Edition, *Addison-Wesley, 1998.*

[35] Yedugani, S. K., Toom-Cook 3 Way Method, http://cs.indstate.edu/~syedugani/ToomCook.pdf

[36] Erturgut, M., Elektronik İmza Kanunu Bakımından E-Belge Ve E-İmza, *Bankacılar Dergisi, Sayı 48, 2003*

[37]Çakar, M. A., Yiğit, T., Çoklu Algoritma Desteğine Dayalı E-İmza Uygulaması (E-Signat),*Akademik Bilişim'07, 31 Ocak - 2 Şubat 2007*

[38] www.e-guven.com

[39]Kırımlı, M., Erdem,O. A., Açık Anahtar Kriptografisi İle Sayısal İmza Tasarımı Ve Uygulaması, *Ulusal Elektronik İmza Sempozyumu Bildiriler Kitabı, sf. 264-268, 2006*

[40] Erol, H., Akcayol, M. A., Türkiye'de Elektronik İmza Uygulamalarında Durum Analizi Ve Öneriler,***Türk Hukuk Dünyası**, Cilt:3, S. 157-174, 2007. (1.Ulusal Elektronik İmza Sempozyumunda Seçilerek Türk Hukuk Dünyası Dergisinde Yayınlanmıştır.)http://docplayer.biz.tr/15258914-Turkiye-de-elektronik-imza-uygulamalarinda-durum-analizi-ve-oneriler.html*

[41] Elektronik İmza Ulusal Koordinasyon Kurulu Hukuk Çalışma Grubu İlerleme Ve Sonuç Raporu - Temmuz, 2004

[42] www.kamusm.gov.tr/tr/bilgideposu/belgeler/teknik/aaa/index.html

[43] Hasırcıoğlu, I., Elektronik İmza Oluşturma Ve Doğrulama- TÜBİTAK UEKAE Gebze/KOCAELİ,

[44] http://www.resmigazete.gov.tr/eskiler/2004/01/20040123.htm T.C. Resmi Gazete 23 Ocak 2004 Cuma Sayı:25355

[45] http://networkkampus.com/ssh-nedir-ne-ise-yarar/

[46] Yeşiltepe, M., Bozkurt, Ö. Ö., Servis Odaklı Mimari Güvenliğinde Güvenlik Tiplerinin Karşılaştırılması,XVII. *Akademik Bilişim Konferansı -- AB 2015 4-6 Şubat 2015*

[47] Levi, A., Özcan, M., Açık Anahtar Tabanlı Şifreleme Neden Zordur?, *Bilişim 2002, TBD 19. Bilişim Kurultayı, Sayfa 41 – 45, 3 - 6 Eylül 2002*

[48]İçli, G. E., Aslan, B., İnternette Ödeme Ve Güvenlik**,** *XIII. Türkiye'de İnternet Konferansı 22-23 Aralık 2008*

[49] Abe, M., Gennaro, R., Kurosawa, K., Shoup, V., Tag-KEM/DEM: A New Framework For Hybrid Encryption And New Analysis Of Kurosawa-Desmedt KEM, *Advances in Cryptology – Eurocrypt 2005, Lncs 3494, Pp. 128–146, 2005.*

[50]Yıldırım, K., Demiray, H. E., Simetrik Ve Asimetrik Şifreleme Yöntemlerine Metotlar: Çırpılmış Ve Birleşik Akm-Vkm, *Gazi Üniv. Müh. Mim. Fak. Der. Cilt 23, No 3, 539-548, 2008*

[51] Iyer, S. C., Sedamkar, R.R., Gupta, S., A Novel Idea On Multimedia Encryption Using Hybrid Crypto Approach, *7th International Conference On Communication, Computing And Virtualization 2016, Procedia Computer Science 79 (2016) 293 – 298*

[52] Atar, E., Ersoy, O. K., Özyılmaz, L., Dik Eşleştirme Arayış Yöntemi İle Hibrit Veri Sıkıştırma Ve Optiksel Kriptografi, *Journal Of The Faculty Of Engineering And Architecture Of Gazi University 32:1 (2017) 139-147*

[53] Arai, T., Obana, S., A Password-Protected Secret Sharing Based On Kurosawa-Desmedt Hybrid Encryption,*Fourth International Symposium On Computing And Networking (CANDAR) CANDAR Computing And Networking (CANDAR), 2016 Fourth International Symposium On. :597-603 Nov, 2016*

[54] Chengliang, L., Ning, Y., Malekian, R., Ruchuan, W., The Hybrid Encryption Algorithm Of Lightweight Data in Cloud Storage,*2nd International Symposium On Agent, Multi-Agent Systems And Robotics (ISAMSR) Agent, Multi-Agent Systems And Robotics (ISAMSR), 2016 2nd International Symposium On. :160-166 Aug, 2016*

[55] Srividya, B.V., Akhila, S., Implementing A Hybrid Crypto-Coding Algorithm For An Image On FPGA, *Information And Communication Technology For Intelligent Systems, ICTIS 2017. (Smart Innovation, Systems And Technologies, 2018, 84:72-84)*

[56] Levi, A., Nasıl Bir E-Posta Güvenligi?, *Bilişim Güvenlik, Mart/Nisan 2003, in Turkish, sayfa 38 - 40.*

[57] Barışık, S., Temel, H., İnternet Bankacılığı Kullanımında Güvenlik Unsurlarının Bilinilirliği (Anket Uygulamasına Dayalı Spss Çözümlemesi), *Karamanoğlu Mehmetbey Üniversitesi Sosyal Ve Ekonomik Araştırmalar Dergisi Sayı2, Sayfa: 136-160 ,2007*

[58] Yerlikaya, T., Gençoğlu, H., Mobil Cihazlarda RSA Algoritmasının Performans Optimizasyonu, *Trakya Üniversitesi Mühendislik Bilimleri Dergisi, Cilt 18, Sayı 1, Haziran 2017, Sayfa: 43-52*

[59] http://primes.utm.edu/curios/index.php?start=301&stop=1000

14.ÖZGEÇMİŞ

1978 yılında Bursa'da doğdum. İlk, Orta ve Lise öğrenimini Bursa'da tamamladıktan sonra lisans eğitimimi İstanbul Üniversitesi Fen Edebiyat Fakültesi Matematik Bölümünü, 2003 yılında, bitirdim. 2008 yılında bu yana farklı üniversitelerde tam zamanlı – yarı zamanlı olarak çalışmaktayım.