İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜVENLİ BİR MOBİL TİCARET PROTOKOLÜ VE UYGULAMASI

YÜKSEK LİSANS TEZİ

Müh. H. Güpse GÜNEŞ

Anabilim Dalı : BİLGİSAYAR MÜHENDİSLİĞİ

Programı : BİLGİSAYAR MÜHENDİSLİĞİ

EYLÜL 2005

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

GÜVENLİ BİR MOBİL TİCARET PROTOKOLÜ VE UYGULAMASI

YÜKSEK LİSANS TEZİ

Müh. Hayriye Güpse GÜNEŞ

504021514

Sunulduğu Tarih : 28 Ağustos 2005

Savunulduğu Tarih: 19 Eylül 2005

Danışman: Prof. Dr. Coşkun SÖNMEZ

Jüri Üyeleri: Prof.Dr. Bülent ÖRENCİK (İTÜ.)

Prof.Dr. Oya KALIPSIZ (YTÜ.)

EYLÜL 2005

ÖNSÖZ

Günümüzde donanım olarak oldukça gelişen mobil cihazları telefon dışında bir çok amaç için kullanmak mümkün hale geldi. Bunlardan biri de mobil cihazların ödeme sistemlerinde kullanılmasıdır. Ancak bu işlem sırasında kablosuz iletişim de kullanılacağından güvenlik oldukça önemli hale gelmektedir. Çalışmamda güvenli bir şekilde kişisel mobil cihazların nasıl ödeme sistemleriyle birlikte kullanılabileceğini araştırdım ve güvenlik açıklarını en aza indiren bir protokol geliştirdim. Öncelikle bana bu çalışmada yol gösteren ve her zaman destek olan hocam Prof. Dr. Coşkun SÖNMEZ'e, bu çalışmayı yapabilecek bilgi birikimine sahip olmamı sağlayan tüm bölüm hocalarıma, kaynak bulmama yardımcı olan sevgili arkadaşlarım Zekeriye ERKİN ve Orhan BIYIKLIOĞLU'na ve bana her zaman sabır gösteren aileme teşekkürü bir borç bilirim.

Eylül 2005 **Hayriye Güpse GÜNEŞ**

İÇİNDEKİLER

KISALTMALAR iv
TABLO LİSTESİ v
ŞEKİL LİSTESİ vı
SEMBOL LİSTESİ vıı
ÖZET vııı
SUMMARY ıx

1. GİRİŞ 1
1.1. Giriş ve Çalışmanın Amacı 1

2. ŞİFRELEME TEKNİKLERİ VE ALGORİTMALAR 2
2.1. Şifreleme Algoritmaları Hakkında Genel Bilgiler 3
2.2. Simetrik Şifreleme 4
2.2.1. DES ve TDES Şifreleme Algoritmaları 4
2.2.1.1. DES Algoritması 4
2.2.1.2. TDES Algoritması 9
2.2.2. Hash Fonksiyonları 9
2.2.2.1. MD5 Algoritması 9
2.2.3. MAC Hesaplama 14
2.3. Asimetrik Şifreleme 14
2.3.1. RSA Algoritması 15
2.3.1.1. RSA Şifreleme İşlemi 15
2.3.1.2. RSA Şifre Çözme İşlemi 15
2.3.2. Sayısal İmza 16
2.3.3. Sertifika ve Açık Anahtar Yönetimi 16
2.3.3.1. Sertifikalar 16
2.3.3.2. PKI Yapısı 16

3. MOBİL TİCARET 21
3.1. Mobil Ticaret Kavramı 21
3.1.1. Mobil Cihazlar 21
3.1.2. Mobil Cihazların Ödeme işleminde Sağlayacağı Güvenlik ve Kolaylık 21
3.2. Kullanımda Olan Mobil Ödeme Uygulamaları 21
3.2.1. I-Mode 21
3.2.2. İşyeri Numarasını SMS Mesajı İle Gönderme 22
3.2.3. Araç Üzerindeki Akıllı Kartlar ile Otoyol ve Park Ücreti Ödeme 22
3.2.4. PayBox 22

3.2.5. IPIN 22
3.3. Kullanılmakta Olan Mobil Ödeme Protokolleri 24
3.3.1. SSL/TLS Protokolleri 24
3.3.1.1. SSL Oturumu 25
3.3.1.2. SSL Bağlantısı 26
3.3.1.3. SSL Kayıt Protokolü 27
3.3.1.4. SSL Şifreleme Algoritması Değiştirme Protokolü 28
3.3.1.5. SSL Alarm Protokolü 28
3.3.1.6. SSL El Sıkışma Protokolü 29
3.3.2. SET 33
3.3.2.1. Kart Sahibi CA Kaydı 33
3.3.2.2. İşyeri Sahibi CA Kaydı 34
3.3.2.3. Alışveriş İşlemi 35
3.3.2.4. Otorizasyon İşlemi 36
3.3.2.5. Ödeme İşlemi 38
3.3.2.6. SET Protokolü Analizi 39
3.3.3. 3D SECURE Protokolü 40
3.3.3.1. 3D Secure Özellikleri 40
3.3.3.2. Üçlü Alan Yapısı 40
3.3.3.3. 3D SECURE'un Temel Fonksiyonları 40
3.3.3.4. 3D SECUREYapısında Kullanılan Elemanlar 41
3.3.3.5. Ödeme İşlem Adımları 43
3.3.3.6. 3D SECURE Protokolünde Güvenlik 43
3.3.4. WAP Protokolü Kullanılarak Mobil Ödeme 44

3. GÜVENLİ BİR MOBİL TİCARET PROTOKOLÜ VE UYGULAMASI 46
4.1. Geliştirilen Protokol Genel Tanımlar 46
4.1.1. Protokolde Kullanılan Yapılar 46
4.1.2. Finansal Kuruluşun Görevleri 46
4.1.3. Müşterini Görevleri 46
4.1.4. POS Cihazında Bulunması Gerekn Özellikler 47
4.1.5. İşlem Akışı 47
4.2. Uygulama Genel Yapısı 49
4.2.1. Mobil Cihaz Üzerinde RSA Algoritması Gerçekleme 50
4.2.2. Açık Anahtar Dosya Formatı 50
4.2.3. Gizli Anahtar Dosya Formatı 50
4.2.4. Finansal Kurum Açık Anahtarının MPA ve POS'a Yüklenmesi 51
4.2.5. Mobil Cihaz Açık Anahtarının Finansal Kuruma Gönderilmesi 51
4.2.6. Hesap Tanımlama İşlemi 52
4.2.7. Alışveriş İşlemi 53

4.2.8. POS Üzerinde Gerçekleşen İşlemler 54
4.3. Geliştirilen Protokolün analizi 55
4.4. Yapılmış Diğer Çalışmalar 56

5. SONUÇLAR 60

KAYNAKLAR 61

ÖZGEÇMİŞ 63

KISALTMALAR

POS	:Point Of Sevice
FI	:Finansal Kuruluş
MPA	:Mobil Ödeme Uygulaması
MD	:Mobil Cihaz
FIP	:Finansal Kuruluş Açık Anahtarı
FIS	:Finansal Kuruluş Gizli Anahtarı
MDP	:Mobil Cihaz Açık Anahtarı
MDS	:Mobil Cihaz Gizli Anahtarı
POS_RN	:POS Referans Numarası
MPA_RN	:MPA Referans Numarası
A	:Tutar
MID	:İşyeri ID'si
RC	:Cevap Kodu
PC	:Provizyon Kodu
EMV	:Europay Mastercard Visa Akıllı kart Ödeme Protokolü
PIN	:Pesonal Identification Number
DES	:Data Encryption Standart Algoritması
MAC	:Message Authentication Code
RSA	:Rivest , Shamir , Adleman 'ın bulduğu asimetrik algoritma
PKI	:Açık Anahtar Yönetim Sistemi (Public Key Infrastructure)
PAA	:Policy Approval Authority
PCA	:Policy Certification Authority
CA	:Certificate Authority
ORA	:Organizational Registration Authority
SSL	: Secure Sockets Layer
SET	:Secure Electronic Transaction
3D SECURE	:3 Domain Secure Model
ACS	:Access Control Server

TABLO LİSTESİ

Tablo 2.1	**:** PC1 Tablosu	4
Tablo 2.2	**:** LS Tablosu	5
Tablo 2.3	**:** PC2 Tablosu	5
Tablo 2.4	**:** Başlangıç Permütasyon Tablosu	6
Tablo 2.5	**:** E Fonksiyon Tablosu	6
Tablo 2.6	**:** S Kutuları Tablosu	7
Tablo 2.7	**:** P Permütasyon Tablosu	7
Tablo 2.8	**:** F, G, H, I Fonksiyonları Doğruluk Tablosu	10
Tablo 2.9	**:** T[i] Değerler Tablosu	11
Tablo 2.10	**:** X509 v1 Sertifika Formatı	21
Tablo 2.11	**:** X509 v2 Sertifika Formatı	21
Tablo 2.12	**:** X509 v3 Sertifika Formatı	21
Tablo 3.1	**:** Kart Sahibi CA Kaydı İçin Yapılan İşlemler ve Gönderilen Mesajlar	34
Tablo 3.2	**:** İşyeri CA Kaydı İçin Yapılan İşlemler ve Gönderilen Mesajlar	36
Tablo 3.3	**:** Alışveriş İşlemi	37
Tablo 3.4	**:** Otoroizasyon İşlemi	38
Tablo 3.5	**:** Ödeme İşlemi	40
Tablo 3.6	**:** SET Protokolü İşlem Analiz Bilgileri	43
Tablo 3.7	**:** 3D SECURE Ödeme İşlemi Adımları	47
Tablo 4.1	**:** Açık Anahtar Dosya Yapısı	53
Tablo 4.2	**:** Gizli Anahtar Dosya Yapısı	54
Tablo 4.3	**:** Protokol Analizi	59
Tablo 4.4	**:** Kungpisdan Protokolü Analizi	61

ŞEKİL LİSTESİ

Şekil 2.1 **:** DES Anahtar Oluşturma 5
Şekil 2.2 **:** DES Şifreleme Adımları 8
Şekil 2.3 **:** Dijital İmzalama Yöntemi 17
Şekil 2.4 **:** Sertifika Doğrulama Zinciri 20
Şekil 3.1 **:** Paybox Yapısı 24
Şekil 3.2 **:** iPIN Yapısı 24
Şekil 3.3 **:** SSL Protokol Katmanları 25
Şekil 3.4 **:** SSL Fonksiyonları 26
Şekil 3.5 **:** SSL Mesaj Yapısı 29
Şekil 3.6 **:** SSL İletişim Adımları 33
Şekil 3.7 **:** WAP Modeli 48
Şekil 4.1 **:** İşlem Akışı 51
Şekil 4.2 **:** Epoc Ekranı 52
Şekil 4.3 **:** FIP Yükleme 55
Şekil 4.4 **:** MDP Gönderme 55
Şekil 4.5 **:** Kart Numarası ve CVV Tanımlama 56
Şekil 4.6 **:** Kullanıcı Şifresinin Girilmesi 57
Şekil 4.7 **:** POS Uygulaması 57

SEMBOL LİSTESİ

p, q	: Asal Sayılar
m	: Şifrelenecek veri
c	: Şifrelenmiş veri
e,n	: Açık anahtar
d	: Gizli anahtar
ki	: Simetrik anahtarın i. Biti
$\oplus$	: XOR işlemi
H[X]	: X'in hash değeri
Ipad, opad	: Padleme çeşitleri
M	: Orjinal Mesaj
S	: Dijital İmza
MAC[X]	: X değerinin MAC değeri
Khi	: Hash zinciri
σ	: Tutar

Üniversitesi : **İstanbul Teknik Üniversitesi**
Enstitüsü : **Fen Bilimleri**
Anabilim Dalı : **Bilgisayar Mühendisliği**
Programı : **Bilgisayar Mühendisliği**
Tez Danışmanı : **Prof. Dr. Coşkun SÖNMEZ**
Tez Türü ve Tarihi : **Yüksek Lisans – Eylül 2005**

ÖZET

GÜVENLİ BİR MOBİL TİCARET PROTOKOLÜ VE UYGULAMASI

Hayriye Güpse GÜNEŞ

Bu çalışmada asimetrik şifreleme algoritmaları ve dijital imza yöntemleri kullanılarak, mobil cihazların ödeme sistemlerinde güvenli bir şekilde kullanılabilmesini sağlayan bir protokol geliştirilmiştir. Bu protokolde, mobil cihaz dışında finansal kuruluşun işyerlerinde kullandığı POS'lar ve finansal kuruluşun merkezinde POS'larla iletişimde olan sunucu programının sahip olması gereken özellikler de tanımlanmıştır. Mobil cihaz ve POS arasındaki iletişimde Bluetooth , POS ve finansal kuruluş sunucu proramı arasındaki iletişim ise TCP/IP ile sağlanmıştır. Protokol, alışveriş işlemini gerçekleştirirken kablosus ve kablolu ortamda aktarılan tüm verilerin , veri bütünlüğünü , gönderenin kimliğini ve verinin gerçekliğini kontrol edebilecek şekilde tasarlanmıştır. Günümüzde performansı oldukça artan mobil cihazlar , asimetrik algoritma anahtar üretimi, şifreleme ve şifre çözme işlemlerini de yapabilecek kapasiteye gelmişlerdir. Çalışmanın devamında protokolün gerçek ortama aktarılması ve yapılan uygulama üzerinde performans değerlendirmeleri yapılmış ve gerçek bir mobil cihaz üzerinde tasarlanan protokol gerçeklenmiştir.

Anahtar Kelimeler: Asimetrik algoritmalar, Mobil cihaz, Bluetooth, TCP/IP, Mobil Ticaret Protokolü, Dijital imza

University	**:**	**İstanbul Technical University**
Institute	**:**	**Institute of Science and Technology**
Science Programme	**:**	**Computer Engineering**
Programme	**:**	**Computer Engineering**
Supervisor	**:**	**Prof. Dr. Coşkun SÖNMEZ**
Degree Awarded and Date	**:**	**Ms – Sep 2005**

ABSTRACT

A SECURE MOBILE COMMERCE PROTOCOL AND ITS APPLICATION

Hayriye Güpse GÜNEŞ

In this study a new secure mobile commerce protocol is developed by using asymmetric algorithms and digital signatures for security. This protocol defines the attributes of financial institutions POS devices in merchants and the attributes of the host application of financial institution that is connected to POS devices and authorises the transacation as well as the attributes of the mobile devices application that connects to POS and sends the required ciphered data to POS. Mobile device connects to POS via Bluetooth, and POS connects to host application via TCP/IP. This protocol ensures that the integrity , authenticity, confidentiality and non-repudiation by using asymmetric algorithms and digital signatures. New mobile devices performance features are adequate for asymmetric algorithm key generation, encryption, decryption and digital signature analysis. The newly designed protocol is simulated in a real mobile device.

Keywords: Asymmetric Algorithms, Mobile Devices, Bluetooth, TCP/IP, Mobile Commerce Protocol, Ditgital Signature

GİRİŞ

1.1 Giriş ve Çalışmanın Amacı

Günümüzde günlük hayatta yoğun olarak kullandığımız mobil cihazlar, donanımlarının da gelişmesiyle bir çok farklı amaçlar için de kullanılmaya başlanmıştır. Her an yanımızda taşıdığımız mobil cihazları , MP3 player, radyo veya fotoğraf makinası yerine kullanabiliyoruz. Ayrıca mobil cihazların ödeme sistemlerinde kredi kartı gibi kullanıldığı bir çok uygulama hali hazırda bir çok ülkede hayata geçirilmiş durumdadır.

Bu çalışmada, mobil cihazların, işyeri POS'larının, finansal kuruluş sunucu uygulamasının ve bu uygulamanın bağlı olacağı güvenlik cihazlarının özellikleri belirlenmiştir. Ayrıca mobil cihazda bulunması gereken ve bankanın sunucu uygulamasında bulunması gereken özellikler de saptanmıştır. Mobil cihazdan işyeri POS'una, işyeri POS'undan finansal kuruluşun sunucu programına gidecek olan ödeme işlemi mesajının içeriği ve nasıl şifreleneceği, ve tersi yönde finansal kuruluştan işyeri POS'una ve oradan da mobil cihaza iletilecek olan finansal kuruluşun işlem cevabında hangi bilgilerin bulunacağı ve nasıl şifrelenip imzalanacağı belirlenmiştir. Mobil cihazların artan donanım özellikleri ve performansları ile birlikte, asimetrik şifreleme algoritmalarını da rahatlıkla hesaplayabilecek duruma gelmeleri ile simetrik algoritmalara göre çok daha güvenli olan asimetrik algoritmalar ve sayısal imzalama teknikleri kullanılarak bir protokol geliştirilmiştir.

Mobil cihaz ile işyeri POS'u arasında yaygınlığı sebebiyle Bluetooth ile iletişim sağlanmış, işyeri POS'u ile finansal kuruluşun sunucu uygulaması arasında ise şu anda altyapısı mevcut olan TCP/IP haberleşmesi kullanılmıştır. Böylece şu anda kullanılmakta olan uygulamalarda yapıldığı gibi GSM servis sağlayacılar kullanılmamış, bu sayede de işlem süresi azaltılmış ve bir performans artışı sağlanmaya çalışılmıştır.

2. ŞİFRELEME TEKNİKLERİ VE ALGORİTMALAR

Şifreleme teknikleri genel olarak gizli(simetrik) ve açık(asimetrik) anahtar teknikleri olmak üzere iki gruba ayrılmaktadır. Gizli (simetrik) şifreleme tekniğinde şifreleme ve şifre çözme işlemini yapacak kurumlar aynı anahtarı paylaşmaktadır. Bu şekilde mesajı şifreleyen ve veri bütünlüğünü sağlayan gönderici taraf ile; mesajı alan, deşifre eden ve doğrulayan alıcı taraf aynı anahtarları kullanmış olur. Bu tip şifreleme teknikleri 1970'lere kadar kullanılan klasik şifreleme tekniğiydi.

Ancak 1970'lerde Diffie ve Hellman, açık anahtar sistemini keşfettiler [7]. Bu yeni sistemde anahtar açık ve gizli olmak üzere iki parçadan oluşmaktadır. Açık anahtar, herkes tarafından bilinecek şekilde yayınlanır. Gizli anahtar ise yanlızca anahtarı oluşturan kişice bilinmektedir. Açık anahtar şifreleme, gizli anahtar ise şifre çözme işlemi için kullanılmaktadır. Açık anahtarı alan kurum ya da kişiler bu anahtar ile istedikleri bilgiyi şifreleyerek anahtarı yayınlayan kuruma gönderirler, bu veri de ancak gizli anahtar ile çözülebilmektedir. Böylece şifreleme ve şifre çözme işlemleri gerçekleştirilir.

Simetrik şifrelemede, şifreli haberleşme gerçekleşmeden önce her iki tarafın da gizli bir biçimde kullanılacak anahtarı paylaşması gerekmektedir. Örneğin cep telefonlarında bu anahtar SIM kartın içine gömülür. Burada haberleşecek kurum sayısı arttıkça anahtar yönetimi de oldukça karmaşık hale gelmektedir. "n" tane kurum birbiri ile simetrik şifreleme ile haberleşmek istiyorlarsa (2.1) formülüne göre kullanılması gereken anahtar sayısı belirlenebilir[8].

$$n * (n-1) / 2 \qquad (2.1)$$

Anahtar yönetiminin zor olmasına karşın simetrik şifreleme asimetrik şifrelemeye göre oldukça hızlı bir şekilde gerçekleşmektedir.

Asimetrik şifrelemede ise açık anahtar rahatlıkla yayınlanabildiğinden herhangi bir anahtar yönetim sistemine de gerek kalmaz. Kendisine şifreli mesaj gönderilmesini isteyen her kurum bir açık anahtar bir de gizli anahtar oluşturur. Gizli anahtarı güvenli bir şekilde kendisinde saklar. Açık anahtar ise diğer kurumlara gönderilir. Her kurum kime şifreli mesaj gönderecekse, o kurumun açık anahtarını bilmelidir. Anahtar yönetimine gerek olmayan asimetrik şifrelemede, şifreleme ve anahtar oluşturma işlemleri simetrik şifreleme yöntemlerine göre oldukça yavaş

gerçekleşmektedir. “n” tane kurum birbiriyle gizli bir şekilde haberleşmek istiyorsa “n” tane anahtar bu işlem için yeterli olmaktadır . Ayrıca anahtarların da gizlice paylaşılması gerekmediğinden bu işlemi gerçekleştirmek için gereken süre ve maliyet azalmaktadır.

Asimetrik şifrelemenin başka bir yararı da sayısal imzalama tekniklerinin asimetrik şifreleme teknikleriyle gerçekleniyor olmasıdır. Sayısal imza ile mesajı gönderenin kim olduğu ispat edilebilmekte ve mesajı alan taraf mesajın istediği kişiden geldiğine emin olabilmektedir.

2.1 Şifreleme Algoritmaları Hakkında Genel Bilgiler

Şifreleme sisteminin iyi bir sistem olması için dört özelliği olması gerekmektedir.

Belgelenebilir Olması, Doğrulama Yapılabilmesi (Authentication): belgelenebilir olma özelliği iki farklı durum için düşünülebilir. Bunlardan bir tanesi iki kurumun haberleşmesi sırasında bir kurumun diğer kurumun kim olduğunu kesin olarak bilmesidir. Bir diğer durumda ise gelen verinin kimden geldiğinin belgelenebilir olması gerekmektedir.

Veri Bütünlüğü ve Güvenliği (Integrity): Kullanılan şifreleme algoritmasının, deşifre edildiğinde veriyi bozmayacak şekilde güvenilir bir algoritma olması gerekmektedir. Aynı zamanda şifrelenmiş verinin karşı tarafa iletilmesi sırasında kötü niyetli biri tarafından herhangi bir alanı bozulursa, şifreli veriyi alan kullanıcının verinin yolda değiştirildiğini anlayabilmesini sağlayacak bir algoritma olmalıdır.

Gizlilik (Confidentiality): Bu özellik de verinin yetkili olmayan bir kurumca okunamayacak şekilde şifrelenmiş olması anlamına gelmektedir.

İnkar Edilemezlik (Non-repudiation): Şifreleme algoritması öyle bir algoritma olmalıdır ki; şifreli mesajı alan taraf , şifreyi gönderenden emin olabilmelidir. Mesajı kimin gönderdiği alan tarafın itirazı durumunda ispat edilebilir olmalıdır.

2.2 Simetrik Şifreleme

2.2.1 DES ve TDES Şifreleme Algoritmaları

2.2.1.1 DES Algoritması

15 Ocak 1977 tarihinde DES (Data Encryption Standart) ilk defa yayınlanmıştır[7]. Bu protokol 64 bitlik veri bloklarını 56 bitlik anahtar ile şifreler ve sonuçta yine 64 bitlik şifrelenmiş veri oluşturur. Orjinal blok 16 kez S box adı verilen permütasyonlardan geçirilir. Deşifre işleminde de aynı anahtar ve aynı permütasyonlar kullanılır.

Kullanılan anahtar: DES için kullanılan anahtar 64 bit olarak üretilir ancak bunun yanlızca 56 biti kullanılır. 64 bitten 56 bite düşürmek için de anahtarın 8. bitleri çıkarılır (k8, k16, k24, k32, k40 , k48, k56, k64) ve Tablo (2.1)'deki gibi bir sıralama ile 56 bitlik yeni anahtar oluşur. Oluşturulan 56 bitlik anahtar 28 bitlik parçalara ayrılır. Bu parçalar daha sonra Tablo (2.2)'ye göre sola kaydırılır.

Tablo 2.1: PC1 Tablosu

57	49	41	33	25	17	9	1	58	50	42	34	26	18
10	2	59	51	43	35	27	19	11	3	60	52	44	36
63	55	47	39	31	23	15	7	62	54	46	38	30	22
14	6	61	53	45	37	29	21	13	5	28	20	12	4

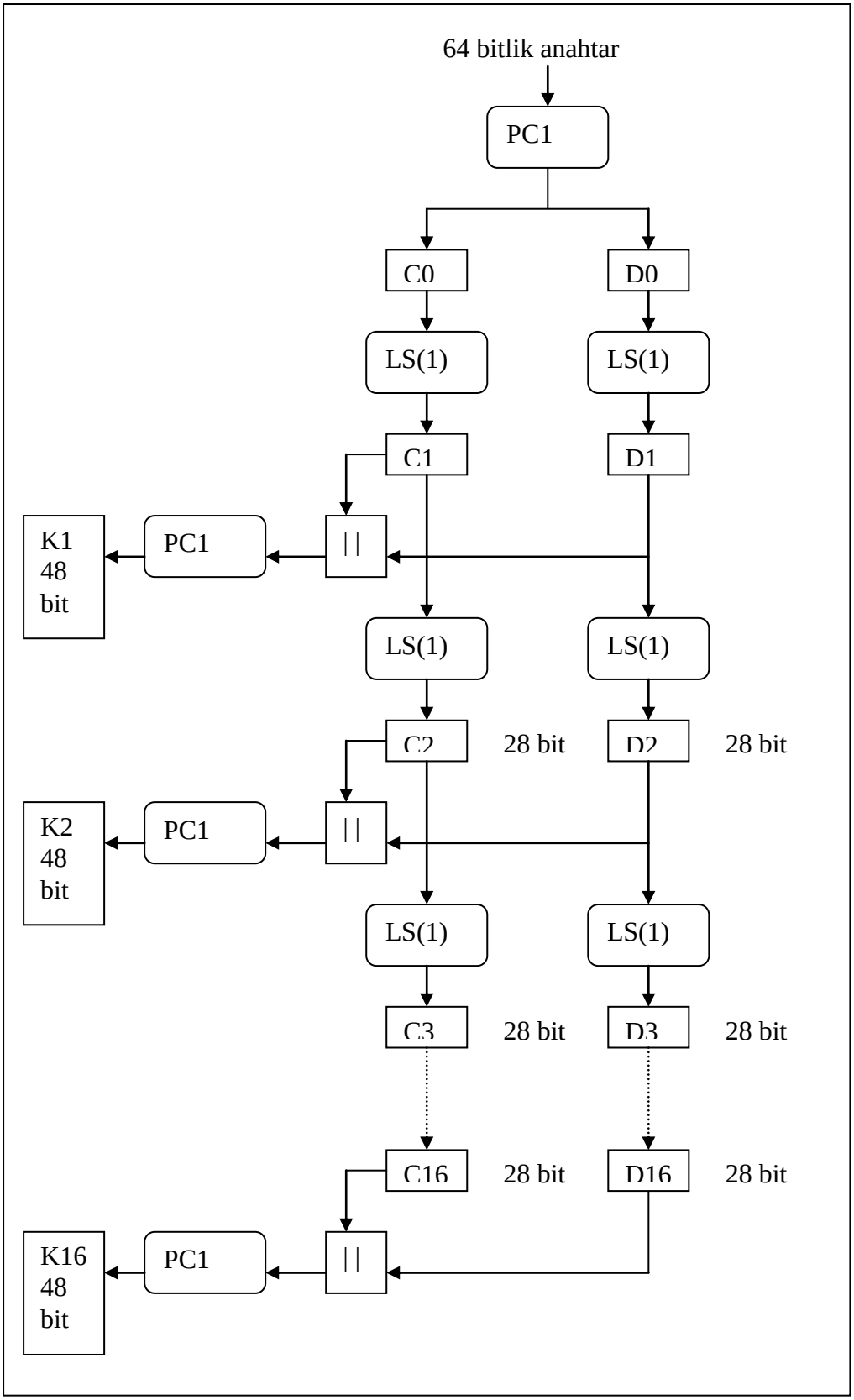
Tablo 2.2: LS Tablosu

Sıra	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Sola kaydırma sayısı	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Kaydırmalardan sonra oluşan 2 ayrı 28 bitlik anahtar birleştirilir ve Tablo (2.3)'deki bitleri seçilerek 48 bitlik anahtar oluşturulur. Anahtar oluşturulurken gerçekleştirilen işlem adımları Şekil (2.1)'deki gibidir.

Tablo 2.3: PC2 Tablosu

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32



Şekil 2.1: DES Anahtar Oluşturma

64 bitlik şifrelenecek veri bloğu önce Tablo(2.4)'deki başlangıç permütasyonundan geçirilir böylece Li ve Ri olmak üzere 2 tane 32 bitlik parçaya ayrılır.

Tablo 2.4: Başlangıç Permütasyonu Tablosu

Li	58	50	42	34	26	18	10	2
	60	52	44	36	28	20	12	4
	62	54	46	38	30	22	14	6
	64	56	48	40	32	24	16	8
Ri	57	49	41	33	25	17	9	1
	59	51	43	35	27	19	11	3
	61	53	45	37	29	21	13	5
	63	55	47	39	31	23	15	7

Ri daha sonra Tablo (2.5)'deki E fonksiyonundan geçirilir. Bu fonksiyon ile 32 bitlik veri 48 bite çıkarılır.

Tablo 2.5: E fonksiyon Tablosu

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Oluşan E(Ri) bloğu Ki ile XOR lanır ve böylece oluşan bloğa Γi denir. Γi 6 bitlik bloklara bölünerek Tablo (2.6)'daki S kutularından geçirilir. Örneğin 101110 bitlerinin ilk ve sonuncu biti satır numarasını , ortadaki 4 biti de sütun numarasını verir ve bu satır ve sütundaki değer S kutusundan çıkan değerdir.

8 farklı S kutusundan çıkan 4 bitlik sayılar yanyana gelerek Ωi bloğunu oluşturur. Bu blok daha sonra Tablo (2.7)'deki P permütasyonundan geçirilir.

Oluşan P(Ωi) bloğu da Li ile XOR lanır ve yeni Ri değerini oluşturur. Daha sonra Li ile Ri yer değiştirerek aynı işlemler tekrarlanır. En sonunda 16. işlemde oluşan R16 ve L16 blokları birleştirilir ve IP-1 permütasyonundan geçirilerek şifrelenmiş blok oluşturulur.

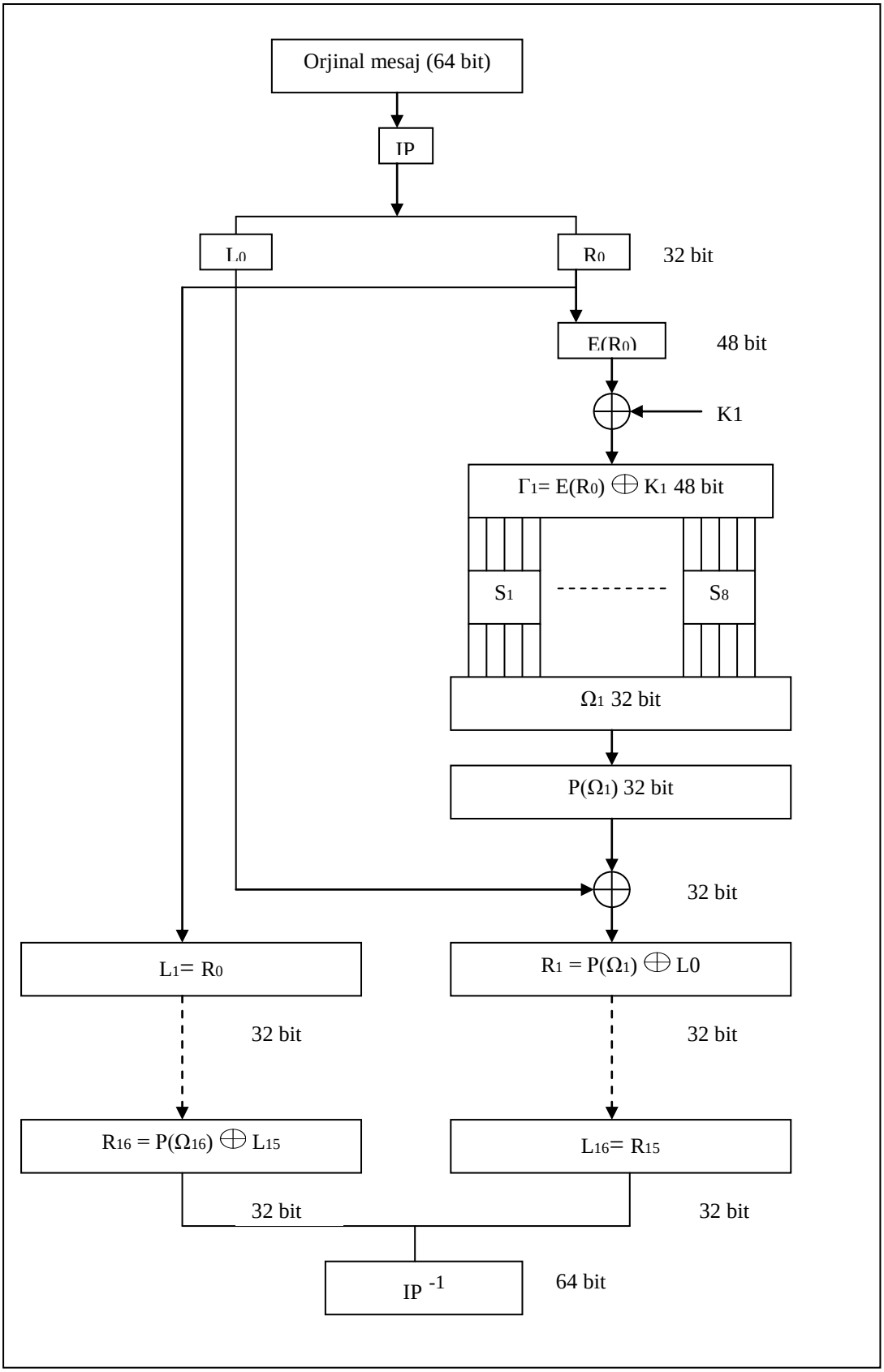
Daha sonra da Şekil(2.2)'deki yol izlenerek şifreleme işlemi gerçekleştirilir.

Tablo 2.6: S Kutuları Tablosu

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S1	0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
	1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
	2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
	3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
S2	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S3	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S4	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S5	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S6	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S7	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S8	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	1	2	5	6	11	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Tablo 2.7: P Permütasyon Tablosu

17	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25



Şekil 2.2: DES Şifreleme Adımları

2.2.1.2 Triple DES Algoritması

Bu şifreleme yönteminde 3 tane anahtar kullanılır. Ancak K1 ve K3 anahtarları aynı olduğundan aslında 2 anahtar kullanılmaktadır. Orjinal blok K1 ile şifrelenir, K2 ile deşifre edilir ve K3 ile yeniden şifrelenir. Bu şekilde şifrelenmiş blok oluşturulur. Deşifre etme işleminde de şifreli blok K3 ile deşifre edilir, K2 ile şifrelenir ve tekrar K1 ile deşifre edilerek orjinal blok elde edilir.

2.2.2 Hash Fonksiyonları

Bu fonksiyonlar farklı uzunluktaki orjinal veriyi alarak sabit bir uzunluğa getiren fonksiyonlardır. Bunu yaparken de farklı verilerin hash sonucunun farklı olmasını garanti eden bir algoritma kullanılmalıdır. Bu fonksiyonlar, daha çok sayısal imzalama işlemlerinde, imzalanacak verinin boyunun küçültülmesini sağlayarak işlemlerin performansını arttırmak için kullanılmaktadır.

2.2.2.3 MD5 Algoritması

Bu algoritmada gelen orjinal veririn uzunluğu ne olursa olsun, oluşacak hash değeri 128 bitlik bir veridir. MD5 algoritmasında orjinal veri 512 bitlik bloklara ayrılarak hash fonksiyonu uygulanır. Bu bloklar da 16 tane 32 bitlik bloğa ayrılır. M[k], 0<=k <= 15 olmak üzere orjinal mesajın bloklarını ifade eder.

Mesaj boyu (448 mod 512) değerine kadar önce bir tane 1 ve geri kalanı da 0 ile doldurulur. Geri kalan 64 bite de orjinal mesajın uzunluğu yazılır. A, B, C, D kütükleri (2.2)'deki gibi daha önceden belirlenmiş verilerle doldurulur. Daha sonra orjinal veri (2.3)'deki yardımcı fonksiyonları kullanan fonksiyonlardan geçirilerek hash fonksiyonu gerçeklenmiş olur.
Örnek kütük içerikleri:

A=	01	23	45	67	= AA
B =	89	AB	CD	EF	= BB
C=	FE	DC	BA	98	= CC
D=	76	54	32	10	= DD

(2.2)

Dört yardımcı fonksiyon (2.3)'de belirtildiği gibi formülleştirilmiştir.

$$F(X, Y, Z) = (X.Y) + (X'.Z)$$

$$G(X, Y, Z) = (X.Z) + (Y.\ Z')$$

$$H(X, Y, Z) = X \oplus Y \oplus Z$$

$$I(X, Y, Z) = Y \oplus (X + Z) \qquad (2.3)$$

(2.3)'deki fonksiyonların doğruluk tablosu Tablo (2.8)'deki gibidir:

Tablo 2.8: F, G, H, I Fonksiyonları Doğruluk Tablosu

X	Y	Z	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	0	0	0
1	1	1	1	1	1	0

Sola öteleme işlemi (2.4)'teki gibi gösteriliyorken ve

$$<<<s \qquad (2.4)$$

T[i] değeri (2.5)'teki gibi formüle edilmişken

$$T[i] = (int)\ (2^{32} * \sin(i)) \qquad (2.5)$$

$$FF(a, b, c, d, M[k], s, i): \quad a = b + ((a + F(b, c, d) + M[k] + T[i] <<< s)$$

$$GG(a, b, c, d, M[k], s, i): \quad a = b + ((a + G(b, c, d) + M[k] + T[i] <<< s)$$

$$HH(a, b, c, d, M[k], s, i): \quad a = b + ((a + H(b, c, d) + M[k] + T[i] <<< s)$$

$$II(a, b, c, d, M[k], s, i): \quad a = b + ((a + I(b, c, d) + M[k] + T[i] <<< s)$$

(2.6)

Hash değerinin bulunması için gereken formüller (2.6)'da gösterildiği gibidir.

T[i] değerinin 1 <= i <= 64 için hesaplanmış değerleri Tablo (2.9) ‘da gösterildiği gibidir.

Tablo 2.9: T[i] Değer Tablosu

T[1] = d76aa478	T[17] = f61e2562	T[33] = fffa3942	T[49] = f4292244
T[2] = e8c7b756	T[18] = c040b340	T[34] = 8771f681	T[50] = 432aff97
T[3] = 242070db	T[19] = 265e5a51	T[35] = 69d96122	T[51] = ab9423a7
T[4] = c1bdceee	T[20] = e9b6c7aa	T[36] = fde5380c	T[52] = fc93a039
T[5] = f57c0faf	T[21] = d62f105d	T[37] = a4beea44	T[53] = 655b59c3
T[6] = 4787c62a	T[22] = 02441453	T[38] = 4bdecfa9	T[54] = 8f0ccc92
T[7] = a8304613	T[23] = d8a1e681	T[39] = f6bb4b60	T[55] = ffeff47d
T[8] = fd469501	T[24] = e7d3fbc8	T[40] = bebfbc70	T[56] = 85845dd1
T[9] = 698098d8	T[25] = 21e1cde6	T[41] = 289b7ec6	T[57] = 6fa87e4f
T[10] = 8b44f7af	T[26] = c33707d6	T[42] = eaa127fa	T[58] = fe2ce6e0
T[11] = ffff5bb1	T[27] = f4d50d87	T[43] = d4ef3085	T[59] = a3014314
T[12] = 895cd7be	T[28] = 455a14ed	T[44] = 04881d05	T[60] = 4e0811a1
T[13] = 6b901122	T[29] = a9e3e905	T[45] = d9d4d039	T[61] = f7537e82
T[14] = fd987193	T[30] = fcefa3f8	T[46] = e6db99e5	T[62] = bd3af235
T[15] = a679438e	T[31] = 676f02d9	T[47] = 1fa27cf8	T[63] = 2ad7d2bb
T[16] = 49b40821	T[32] = 8d2a4c8a	T[48] = c4ac5665	T[64] = eb86d391

Tüm bu fonksiyonlar ve formüller MD5 algoritmasının 4 adımında sırayla kullanılmaktadır.

İlk adımda (2.7) formülü kullanılarak (2.8)’deki sıraya göre a,b,c,d değerleri hesaplanır. a,b,c,d değerleri fonksiyona girmeden önce A, B, C, D kütükleri ile eşitlenir.

FF[a, b, c, d, M[k], s, i] → a = b + ((a + F(b, c, d) +M[k] + T[i]) <<< s) (2.7)

FF[a, b, c, d, M[0], 7, 1],

FF[d, a, b, c, M[1], 12, 2],

FF[c, d, a, b, M[2], 17, 3],

FF[b, c, d, a, M[3], 22, 4],

FF[a, b, c, d, M[4], 7, 5],

FF[d, a, b, c, M[5], 12, 6],

FF[c, d, a, b, M[6], 17, 7],

FF[b, c, d, a, M[7], 22, 8],

FF[a, b, c, d, M[8], 7, 9],

FF[d, a, b, c, M[9], 12, 10],

FF[c, d, a, b, M[10], 17, 11],

FF[b, c, d, a, M[11], 22, 12],

FF[a, b, c, d, M[12], 7, 13],

FF[d, a, b, c, M[13], 12, 14],

FF[c, d, a, b, M[14], 17, 15],

FF[b, c, d, a, M[15], 22, 16] (2.8)

İkinci adımda ilk adımdan elde edilen a, b, c, d değerleri kullanılarak (2.9) formülü ile (2.10)'daki sıraya göre yeni a, b, c, d değerleri hesaplanır.

GG[a, b, c, d, M[k], s, i] → a = b + ((a + G(b, c, d) +M[k] + T[i]) <<< s) (2.9)

GG[a, b, c, d, M[1], 5, 17],

GG[d, a, b, c, M[6], 9, 18],

GG[c, d, a, b, M[11], 14, 19],

GG[b, c, d, a, M[0], 20, 20],

GG[a, b, c, d, M[5], 5, 21],

GG[d, a, b, c, M[10], 9, 22],

GG[c, d, a, b, M[15], 14, 23],

GG[b, c, d, a, M[4], 20, 24],

GG[a, b, c, d, M[9], 5, 25],

GG[d, a, b, c, M[14], 9, 26],

GG[c, d, a, b, M[3], 14, 27],

GG[b, c, d, a, M[8], 20, 28],

GG[a, b, c, d, M[13], 5, 29],

GG[d, a, b, c, M[2], 9, 30],

GG[c, d, a, b, M[7], 14, 31],

GG[b, c, d, a, M[12], 20, 32] (2.10)

Üçüncü adımda ikinci adımdan elde edilen a, b, c, d değerleri kullanılarak (2.11) formülü ile (2.12)'daki sıraya göre yeni a, b, c, d değerleri hesaplanır.

HH[a, b, c, d, M[k], s, i] → a = b + ((a + H(b, c, d) +M[k] + T[i]) <<< s) (2.11)

HH[a, b, c, d, M[5], 4, 33],

HH[d, a, b, c, M[8], 11, 34],

HH[c, d, a, b, M[11], 16, 35],

HH[b, c, d, a, M[14], 23, 36],

HH[a, b, c, d, M[1], 4, 37],

HH[d, a, b, c, M[4], 11, 38],

HH[c, d, a, b, M[7], 16, 39],

HH[b, c, d, a, M[10], 23, 40],

HH[a, b, c, d, M[13], 4, 41],

HH[d, a, b, c, M[0], 11, 42],

HH[c, d, a, b, M[3], 16, 43],

HH[b, c, d, a, M[6], 23, 44],

HH[a, b, c, d, M[9], 4, 45],

HH[d, a, b, c, M[12], 11, 46],

HH[c, d, a, b, M[15], 16, 47],

HH[b, c, d, a, M[2], 23, 48] (2.12)

Dördüncü adımda üçüncü adımdan elde edilen a, b, c, d değerleri kullanılarak (2.13) formülü ile (2.14)'deki sıraya göre yeni a, b, c, d değerleri hesaplanır.

II[a, b, c, d, M[k], s, i] → a = b + ((a + I(b, c, d) +M[k] + T[i]) <<< s) (2.13)

II[a, b, c, d, M[0], 6, 49],

II[d, a, b, c, M[7], 10, 50],

II[c, d, a, b, M[14], 15, 51],

II[b, c, d, a, M[5], 21, 52],

II[a, b, c, d, M[12], 6, 53],

II[d, a, b, c, M[3], 10, 54],

II[c, d, a, b, M[10], 15, 55],

II[b, c, d, a, M[1], 21, 56],

II[a, b, c, d, M[8], 6, 57],

II[d, a, b, c, M[15], 10, 58],

II[c, d, a, b, M[6], 15, 59],

II[b, c, d, a, M[13], 21, 60],

II[a, b, c, d, M[4], 6, 61],

II[d, a, b, c, M[11], 10, 62],

II[c, d, a, b, M[2], 15, 63],

II[b, c, d, a, M[9], 21, 64] (2.14)

Son adımda en son elde edilen a,b,c,d değerleri ile bu değişkenlerin ilk değerleri toplanır ve oluşan a,b,c,d değerleri birleştirilerek 128 bitlik veri oluşturulur. Bu şekilde hash fonksiyonundan değişken uzunluktaki veriler sabit bir uzunluğa indirgenebilir. Hash fonksiyonları tek yönlü fonksiyonlardır. Bu yüzden orjinal veriden hash sonucunu bulmak kolay bir işlemken , hash sonucundan orjinal veriyi bulmak çok zor olmaktadır.

2.2.3 MAC Hesaplama

MAC değeri hesaplanırken yine hash fonksiyonları kullanılır ancak bu defa bir de anahtar kullanılmalıdır.

İpad ve opad değerleri farklı padleme tekniklerini göstermek üzere HMAC değeri (2.15) formülüne göre hesaplanmaktadır.

$$HMAC = H[\ K \oplus opad \ ||\ H\ [K \oplus ipad \ ||\ M\ ]] \quad (2.15)$$

2.3 Asimetrik Şifreleme

Asimetrik algoritmalar RSA, Schnorr, ElGamal, ECC ve benzeri algoritmalardır. Yalnızca sayısal imzalama için uygun olan bir algoritma da DSA algoritmasıdır. Bu tip algoritmalarda anahtar, biri açık biri de gizli olmak üzere iki ayrı bileşenden

oluşur. Açık anahtar herkes tarafından bilinebilir ve şifreleme işleminde kullanılır. Gizli anahtar ise yanlızca anahtar üretimi yapan kurumca bilinir ve şifre çözme işlemi için kullanılır.

2.3.4 RSA Algoritması

RSA algoritmasının adı Rivest, Shamir, Adleman isimlerinin baş harflerinden oluşmaktadır[7]. Büyük sayıların çarpanlarına ayırmanın zorluğuna dayanan bir algoritmadır. "e" açık anahtarı, "n" modulusu, "d" gizli anahtarı değerleri "n" sayısını çarpanlarına ayırarak bulunabilir.

RSA algoritmasında "p" ve "q" olmak üzere iki tane büyük asal sayı seçilir.

Modulus "n" değeri (2.16)'daki formüle göre hesaplanır. Ø(n) değeri de (2.17) formülüne göre hesaplanır.

$$n = p.q \tag{2.16}$$

$$Ø(n) = (p - 1).(q - 1) \tag{2.17}$$

Ø(n) ile aralarında asal bir "e" sayısı seçilir. Daha sonra (2.18) formülüne uygun bir "d" sayısı hesaplanır. "d" ve modulus "n" sayıları da aralarında asal sayılardır.

$$e.d = 1 \bmod Ø(n) \tag{2.18}$$

Bu sayılardan

Açık anahtar : e, n

Gizli anahtar: d

şeklinde tanımlanır.

2.3.4.4 RSA Şifreleme İşlemi

"c" şifrelenmiş veri ve "m" şifrelenecek veri olmak üzere şifreleme işlemi (2.19)'daki gibi formüle edilmektedir.

$$c = m^e \pmod n \tag{2.19}$$

2.3.4.5 RSA Şifre Çözme İşlemi

Şifre çözme işlemi (2.20) formülüne göre yapılmaktadır.

$$m = c^d \pmod n \tag{2.20}$$

2.3.5 Sayısal İmza

Sayısal imza, belgelenebilirlik ve inkar edilemezlik özelliklerini sağlayan ve asimetrik algoritmalarla oluşturulan bir imzalama tekniğidir. Bu yöntemde gizli alıcıya gönderilecek verinin imzalanması için gönderici bir açık bir de gizli anahtar çifti oluşturur. Açık anahtarı alıcıya gönderir, gönderilecek mesajı da gizli anahtar ile şifrelenmiş hali ile birlikte alıcıya gönderir. Alıcı, açık anahtarla şifreli mesajın şifresini çözer ve oluşan mesajla gelen orjinal mesajı karşılaştırır. Eğer mesajlar uyumlu ise gönderici tarafın açık anahtarı gönderen kişi olduğu kesinleşmiş olur. Böylece hattı dinleyen birinin orjinal mesajı bozması halinde ya da mesajın farklı bir gönderici tarafından kötü niyetli olarak gönderilmesi halinde, alıcı taraf bu durumu sezebilir.

RSA hem şifreleme hem de sayısal imza için kullanılabilen bir algoritmadır. A kullanıcısının ürettiği anahtarlar e_A, n_A ve d_A olsun. A kullanıcısı göndereceği veriyi gizli anahtarı ile şifreler ve açık veri ile birlikte oluşan şifrelenmiş veriyi B kullanıcısına gönderir. Bundan önce de açık anahtarlarını yayınlayarak B kullanıcısının açık anahtara ulaşmasını sağlar. İmzalı veriyi alan B kullanıcısı açık anahtar ile şifreli mesajı çözer ve orjinal mesaj ile karşılaştırır. Eğer iki veri birbiri ile uyumlu ise o zaman mesajın A'dan geldiği anlaşılmış olur. Sayısal imzalama tekniği Şekil (2.3)'te detaylı olarak anlatılmaktadır.

2.3.6 Sertifika ve Açık Anahtar Yönetimi (PKI)

2.3.6.6 Sertifikalar

Sertifikalar bir açık anahtar ile, açık anahtarın amacı ve anahtarı yayınlayan kurumun kimliğini eşleştiren yapılardır.

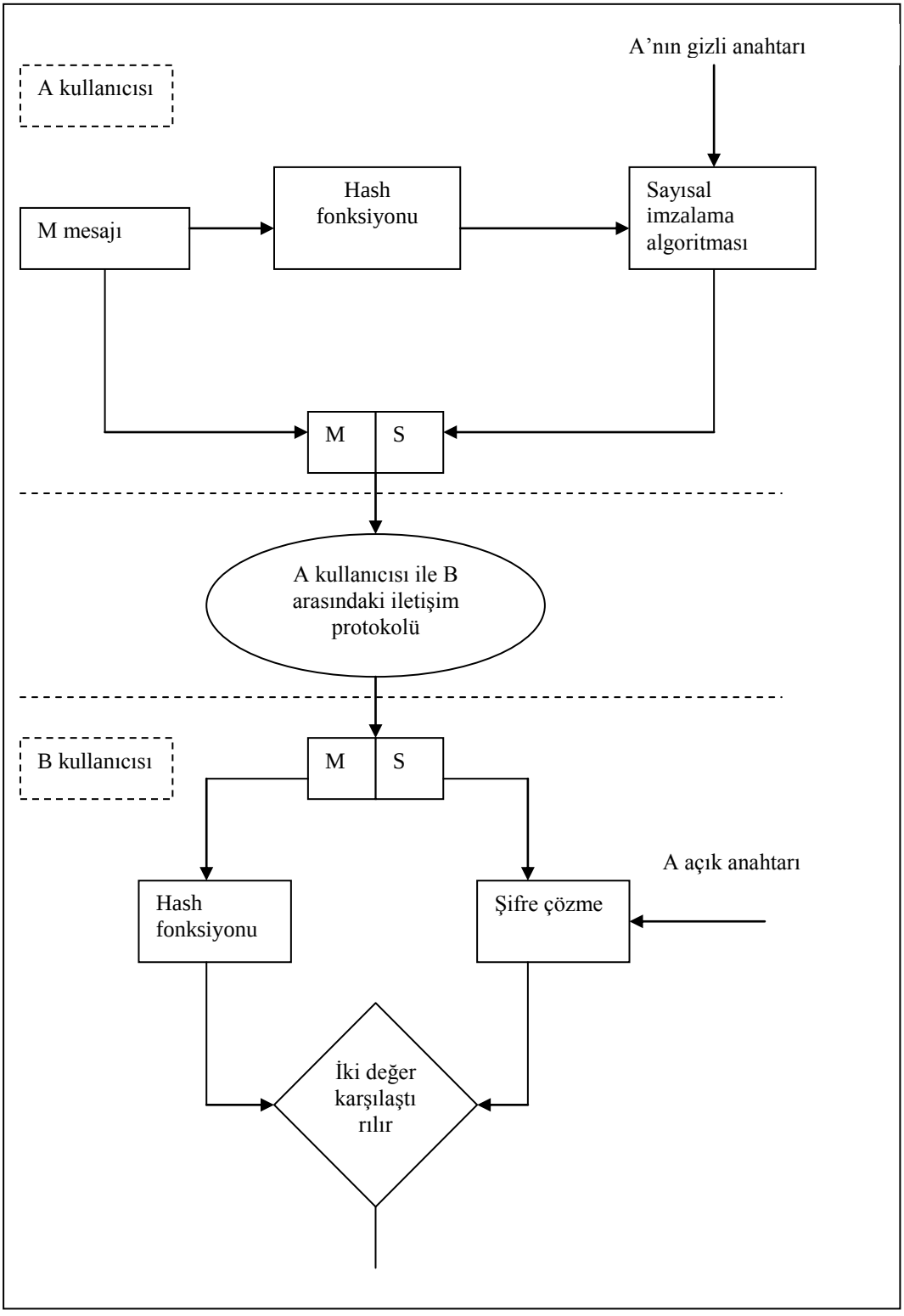
2.3.6.7 PKI Yapısı

Sertifikaların geçerli oldukları tarih boyunca yönetilebilmesi için PKI (Public Key Infrastructure) yapısı kullanılmaktadır. PKI açık anahtar sertifikalarını yöneten bir sistemdir. Bir çok sistemin birbirlerini doğrulayarak bilgi alışverişi yapmalarını sağlayan sertifikaların yönetimi bu altyapı ile sağlanmaktadır.

PKI Yapısında Görev Alan Kurumlar ve Fonksiyonları:

PAA :

PAA sertifika yönetimi sisteminin başlangıç noktasıdır. PKI yapısına dahil olan tüm yapıların uyması gereken kuralları PAA belirler ve PKI'ya dahil olmak isteyen yeni kurumların uyması gereken kurallar ve prosedürler de PAA tarafından belirlenir.



Şekil 2.3: Sayısal İmzalama Yöntemi

PAA'nın görevleri:

- Kendi açık anahtarını yayınlar.
- Yapıya dahil olan diğer kurumların uyacağı kuralları ve prosedürleri belirler.

- Yapıya yeni dahil olacak diğer kurumların uyması gereken kural ve prosedürleri belirler.
- Yapıya dahil olan diğer kurumların tanımlamasını ve doğrulamasını yapar.
- Kendine bağlı olan kurumlar için sertifika oluşturur.
- Kendine bağlı kurumların sertifika iptal isteğini karşılar.
- Kendi ürettiği iptal edilmiş sertifika listesini yayınlar.

PCA :

- Kendi kimlik, adres ve telefon numarası gibi bilgileri son kullanıcılara yayınlar.
- Kendine bağlı olan sertifika otorite bilgilerini yayınlar.
- Kime hizmet etmeyi planladığını yayınlar.
- Güvenlik kurallarını yayınlar. Bu kurallar aşağıdaki gibidir.
    - Anahtar üretimini kimin yapacağı
    - Anahtar boyutu
    - Kendine bağlı kurumların kimlik belirlemesi ve doğrulaması için gerekli bilgiler
    - Sertifika üreten ve bu sertifikaları iptal eden kurumlardaki güvenlik kontrolleri
    - Her kullanıcının gizli anahtarı için güvenlik kontrolleri
    - ORA sistemindeki güvenlik kontolleri
    - İptal edilmiş sertifika listesinin yayınlanma sıklığı
- Kendine bağlı tüm yapıların kimlik belirleme ve doğrulama işlemleri
- Kendine bağlı CA'ların sertifika üretimi ve yönetimi
- Kendi açık anahtarının kendine bağlı kurumlara yayınlanması
- Kendi ürettiği sertifikaların iptali ve bunların listelerinin yayınlanması

CA:

- Kendine bağlı kurumların kimlik tesbiti ve doğrulamasını yapar.
- Kendine bağlı kurumlar için sertifika üretir ve bu sertifikaların yönetimini yapar.
- Kendi ve üst kurumlarının açık anahtarlarını taşır.

- ORA'ların sertifika isteklerini karşılar.
- Sertifika iptali ve iptal listelerini yayınlar.

ORA:

ORA'lar kullanıcılar ve CA'lar arasında köprü görevi görür. Başvuran kurumun kimlik tesbitini ve doğrulamasını yapar. CA tarafından üretilen sertifikayı da kuruma iletir.

Bir kullanıcının CA tarafından üretilmiş bir sertifikası varsa bu sertifika CA tarafından imzalanmıştır. Kullanıc CA'nın açık anahtarını biliyorsa sertifikanın CA tarafından üretilip üretilmediğini imzayı doğrulayarak öğrenebilir.

X bir CA ve A bir kullanıcı olmak üzere , (2.21) şeklinde bir gösterim A kullanıcısının sertifikası X CA'sı tarafından yayınlanmıştır anlamını taşımaktadır.

X <<A>> **(2.21)**

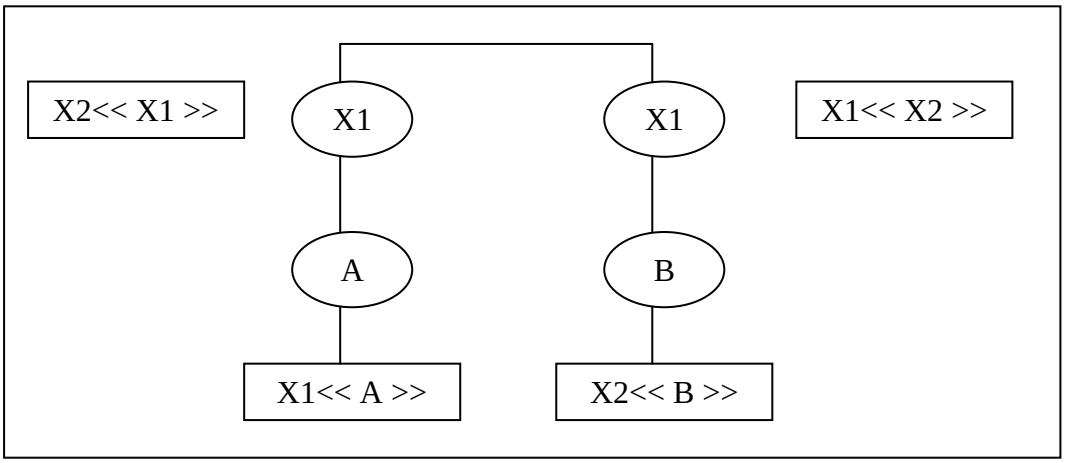
(2.22) ise B kullanıcısının sertifikasının X2 tarafından , X2'nin sertifikasının da X1 tarafından yayınlandığını gösterir.

X1 << X2 >> X2 << B >> ifadesi **(2.22)**

Aynı şekilde (2.24) ifadesi de A'nın sertifikasının X1 tarafından, X1 sertifikasının da X2 tarafından yayınlandığını gösterir.

X2 << X1 >> X1 << A >> **(2.23)**

A ve B kullanıcıları Şekil (2.4)'teki yol üzerinden birbirlerinin sertifikalarına ulaşabilirler.



Şekil 2.4: Sertifika Doğrulama Zinciri

X509 Sertifika Formatı

X509 versiyon1 sertifika formatı Tablo (2.10)'da, versiyon 2 formatı da Tablo (2.11)'de belirtilmektedir. Versiyon 3 formatı ise Tablo (2.12)'de belirtilmiştir.

Tablo 2.10: X509 V1 Sertifika Formatı

Sertifika alanı	**Açıklama**
Versiyon	Sertifika versiyonu
Seri numarası	Sertifika seri numarası
Sayısal imza algoritması	Sertifikayı yayınlayan kurumun sayısal algoritması
Yayıncı	Sertifikayı yayınlayan CA
Geçerlilik süresi	Sertifikanın geçerli olduğu başlangıç ve bitiş tarihleri
Konusu	Sertifikanın hangi amaç için kullanılacağı
Açık anahtar	Kullanılan algoritma bilgisi ve açık anahtar değeri
Yayıncı imzası	CA'nın sayısal imzası

Tablo 2.11: X509 V2 Sertifika Formatı

Sertifika alanı	**Açıklama**
Yayıncı Id	Aynı yayıncı isminin ilerde farklı kurumlarca kullanılması durumunda sertifikanın yayıncısını ayırt edebilmek için eklenmiştir.
Konu Id	Aynı konu isminin ilerde farklı kurumlarca kullanılması durumunda sertifikanın konusunun ayırt edebilmek için eklenmiştir.

Tablo 2.12: X509 V2 Sertifika Formatı

Sertifika alanı	**Açıklama**
Anahtar ve tanımlanmış kurallar bilgisi	Sertifika kullanımında ve anahtar kullanımında uyulması gereken kurallar bilgisi.
Konu ve yayıncı özellikleri	Sertifikanın yayın konusu ve yayınlayan kurumun özellikleri.
Sertifika rotası	Sertifikanın son kullanıcıya gelene kadar izlediği yollar.
İptal edilmiş sertifika listesine ait bilgiler	

3. MOBİL TİCARET

3.1 Mobil Ticaret Kavramı

Mobil ticaret, herhangi bir mobil cihaz kullanılarak yapılan herhangi bir elektronik işlem ya da bu amaç için yapılan bilgi alışverişidir. Mobil ticaret, mobil ödeme sistemlerini de kapsar. Mobil ödeme sistemi ise herhangi bir alışveriş işlemi sırasında finansal bilgi aktarımı için mobil cihazların kullanılmasıdır [1].

3.1.1 Mobil Cihazlar

Mobil cihazlar cep telefonları, PDA'lar, kablosuz tabletler ve mobil bilgisayarları kapsayan cihazlardır[1].

3.1.2 Mobil Cihazların Ödemede Sağlayacakları Güvenlik ve Kolaylık

Geliştirilen protokolde tüm finansal işlemler , daha önceden finansal kuruluşun güvenilir bir kanalı ile müşteri tarafından alınmış mobil ödeme şifresi (PIN) kullanılarak yapılacağından, mobil cihaz çalınsa bile PIN bilgisine sahip olunmadan bu cihaz ile mobil ödeme gerçekleştirilemez. Ayrıca her an yanımızda taşıdığımız cep telefonlarımızı kredi kartı gibi kullanarak, ayrıca kredi kartı ve cüzdan taşıma zorunluluğumuz kalkar.

Mobil cihazlardaki gelişen ekran çözünürlükleri ve donanım özellikleri ile, eskiye ait yapılan harcamalar saklanıp listelenebilir. Ödeme işlemleri analizleri yapılabilir.

3.2 Kullanılmakta Olan Mobil Ödeme Uygulamaları

3.2.1 I-Mode

WAP tabanlı bir uygulamadır. Japonya'da NTT DoCoMo şirketi tarafından kullanılmaktadır ve oldukça yaygınlaşmıştır. Mobil cihazlar veri aktarımında cHTML yapısını kullanırlar. Bu yapı herhangi bir web sayfası yapımında da kullanılabilen bir yapıdır[2]. İşlemler cep telefonu faturasına yansıtılır ve yapılan harcamalar bu şekilde tahsil edilir.

3.2.2 İşyeri Numarasını SMS Mesajı İle Gönderme

Bu uygulamada müşteri alışveriş yaptığı işyerinin ID'sini işyerinden öğrenir ve daha önceden belirlenmiş bir numaraya SMS mesajı ile bu ID'yi gönderir. Müşterinin bağlı olduğu finansal kuruluş bu mesajla gelen ID'yi alır ve bir kriptogram oluşturarak müşteriye bu kriptogramı tekrar SMS ile gönderir. Müşteri gelen kriptogramı işyerine bildirir ve işyeri POS'una bu kriptogram girilerek işlem tekrar finansal kuruluşa onay için gönderilir. [3]

3.2.3 Araç Üzerindeki Akıllı Kartlar İle Otoyol ve Park Ücreti Ödeme

Dedicated Short Range Communications kullanılarak oluşturulmuştur. 5.8 GHz bandında çalışır. Japonya'da ITS(Intelligent Transport System Services) tarafından yayınlanmış bir uygulamadır. Her araç içerisine yerleştirilen akıllı kartlar ile , bu kartları okuyabilecek kart okuyucuların otoparklar, otoyol gişeleri, arabalı vapurlar ve benzeri yerlerde kullanılması ile ödemenin yapıldığı sistemdir. Bu sistemde EMV standartları[4] kullanılmaktadır. Mart 2001 tarihinde Japonya'da uygulanmaya başlanmıştır[5].

3.2.4 Paybox

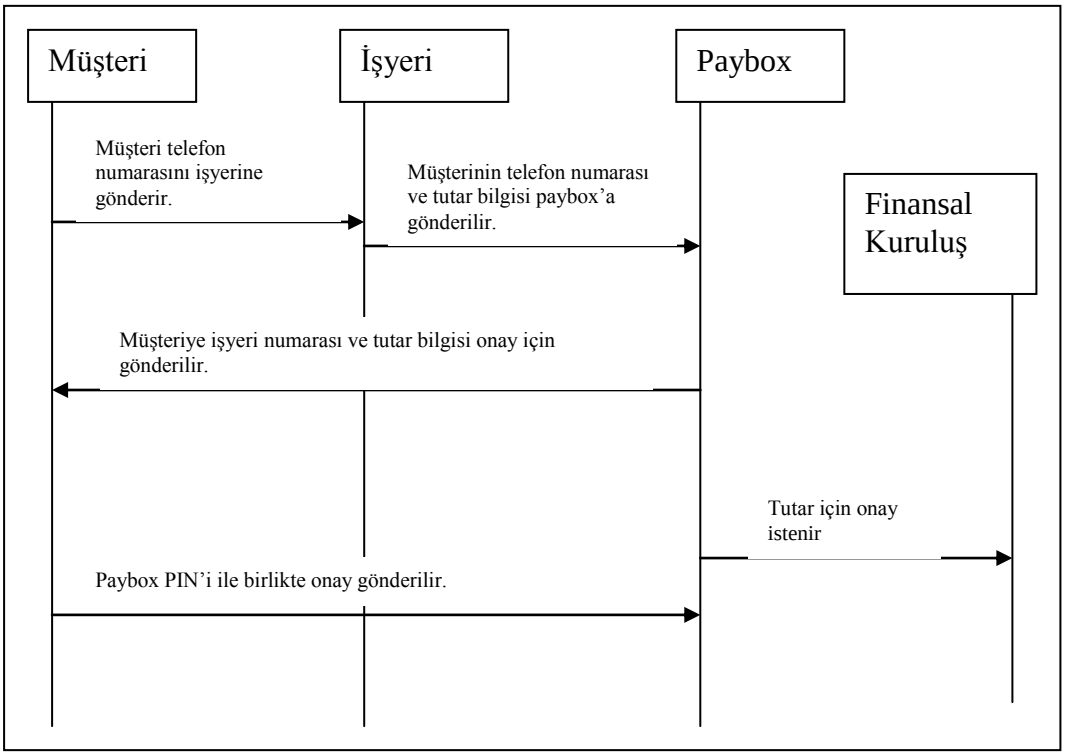
Mayıs 2000 tarihinde Almanya'da uygulamaya açılmış bir sistemdir. Müşteri telefon numarasını işyerine iletir. İşyeri bu numarayı Paybox sistemine iletir. Paybox müşteriye tutar bilgisini göndererek onay ister. Müşteri Paybox PIN'ini girerek işlemi onaylar. Paybox sistemi tutarı müşteri hesabından , işyeri hesabına aktarır. Paybox ile yapılan veri akışı GSM üzerinden Şekil (3.1)'deki yapıya uygun olarak gerçekleştirilir [2].

3.2.5 iPIN

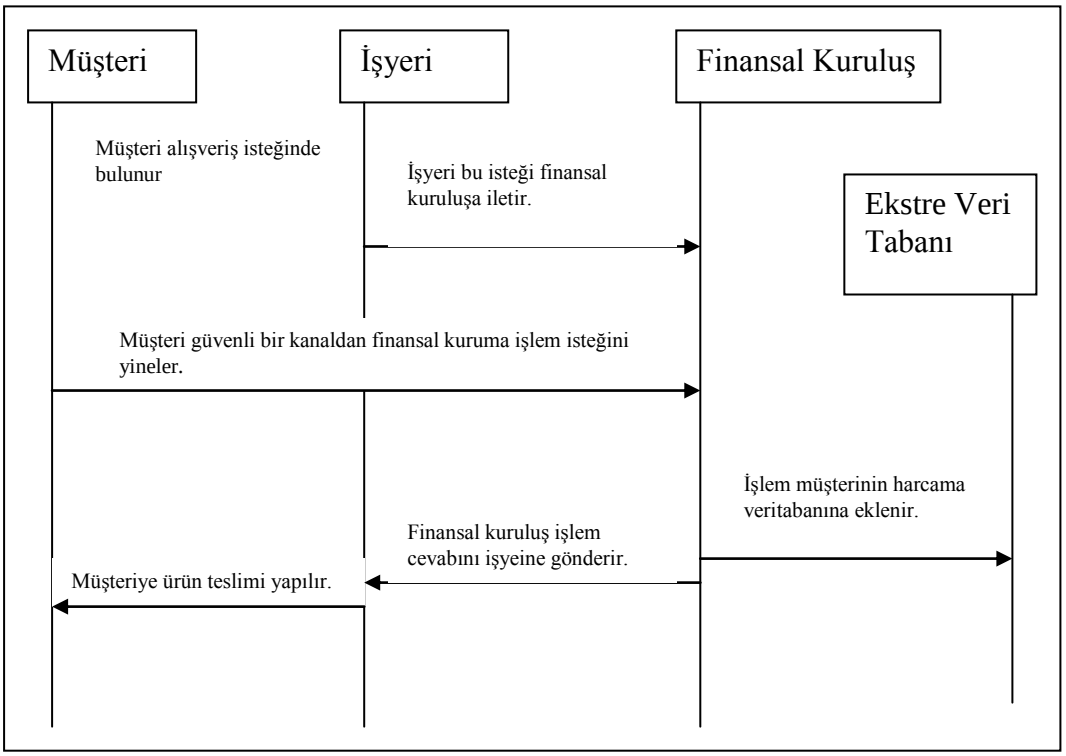
Bu uygulamada müşteri işeyerine işlem isteğinde bulunur. İşyeri bu isteği finansal kuruluşa iletir. Daha sonra müşteri ve finansal kuruluş arasında kurulan güvenli bir bağlantı üzerinden müşteri bu isteği yineler. Finansal kuruluş işlemi onayladığına dair işyerine bilgi döner. Şekil (3.2)'de iPIN yapısı gösterilmektedir. Ancak müşteri ve finansal kuruluş arasındaki güvenli bağlantıyı kurmak için WAP kullanılır bu yüzden de GSM operatörleri devreye girer.

Bunlar dışında sadece işlemin tamamen WAP üzerinde gerçekleştirildiği WPP protokolü , Vodafone m-PayBill uygulaması gibi bir çok örnek mevcuttur. Ancak bu örneklerin hepsinde GSM operatörleri bir şekilde devreye girmektedir ve bu

durumda da sistemin kararlılığı ve servis kalitesi en fazla GSM operatörünün hizmetlerindeki kararlılık ve servis kalitesi kadar iyi olabilir.



Şekil 3.1: Paybox Yapısı



Şekil 3.2: iPIN Yapısı

3.3 Kullanılmakta Olan Mobil Ödeme Protokolleri

3.3.1 SSL/TLS Protokolleri

SSL ve TLS protokolleri yanlızca mobil ödeme işlemlerinde kullanılmaz, genel olarak güvenli veri iletişimi için oluşturulmuş ve TCP/IP katmanı üzerinde yer alan bir protokoldür. SSL ve TLS protokolleri birbirlerine çok benzediklerinden yanlızca SSL hakkında ayrıntılı bilgi verilecektir.

Bu protokol 1995 yılında Netscape tarafından yayınlanmıştır. Kayıt (Record) Protokolü ve Elsıkışma (Handshake) protokolü olmak üzere ikiye ayrılmaktadır. Protokol katmanları Şekil (3.3)'te gösterilmektedir.

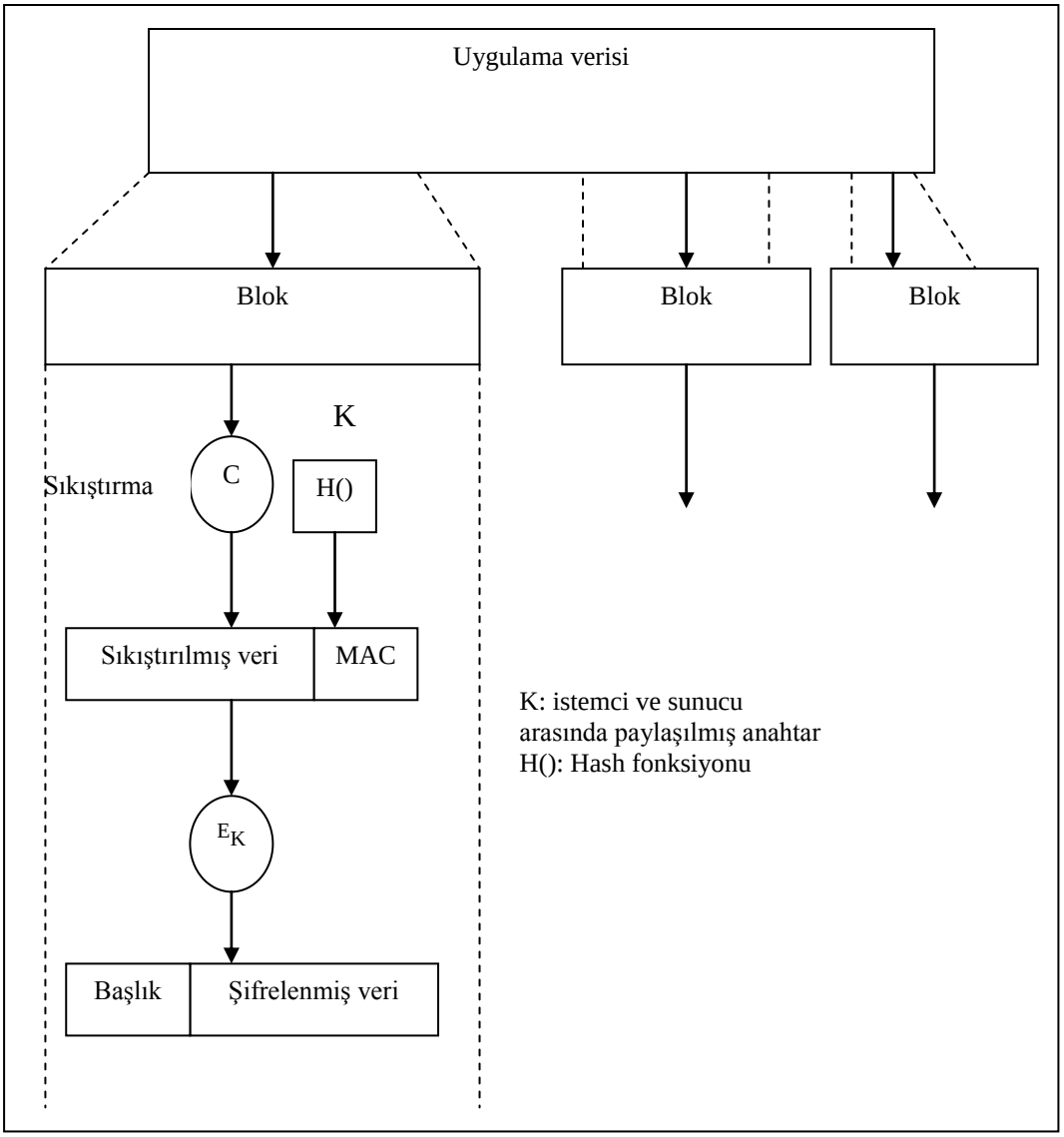
SSL katmanlı bir yapıya sahiptir ve 2 protokolün birleşmesi ile oluşmaktadır. En altta SSL kayıt protokolü yer almaktadır. Bu protokol TCP ya da daha farklı güvenilir iletişim protokollerinin üzerine yerleşmiştir. HTTP gibi protokoller de SSL üzerinde yer alabilecek şekilde çalışabilirler.

<table>
<tr><td>SSL
El skışma
Protokolü
(SSL Handshake
Protocol)</td><td>SSL
Şifreleme Türü
Değiştirme
Protokolü
(SSL Change
Cipher Spec
Protocol)</td><td>SSL
Alarm Verme
Protokolü
(SSL Alert
Protocol)</td><td>HTTP</td></tr>
<tr><td colspan="4">SSL Kayıt Protokolü
(SSL Record Protocol)</td></tr>
<tr><td colspan="4">TCP</td></tr>
<tr><td colspan="4">IP</td></tr>
</table>

Şekil 3.3: SSL Protokol Katmanları

SSL Kayıt protokolü üst seviyedeki protokollerden gelen mesajları bloklara ayırır, sıkıştırır, MAC değerini hesaplar, şifreler, bir başlık ekler ve oluşan blokları TCP protokolüne iletir. Şekil (3.4)'te SSL protokolünün işleyişi gösterilmektedir.

TCP'den gelen mesajların da şifresini çözer, sıkıştırılmış mesajı eski haline çevirir, birleştirir ve üst seviye protokollerine iletir.



Şekil 3.4: SSL Fonksiyonları

3.3.1.1. SSL Oturumu

SSL oturumu istemci ve sunucu arasında ortaklaşa yapılan bir çalışmadır ve el sıkışma protokolü tarafından oluşturulur. Her bağlantı için oldukça maliyetli olan güvenlik parametrelerinin belirlenmesi işleminin maliyetini azaltan bir yapıdır. SSL oturumu istemci ve sunucu durumlarının koordinasyonunu sağlar. İstemci ya da sunucu "çalışmakta olan durum" ve "askıda olan durum" şeklinde 2 farklı değer alabilmektedir. İstemci ya da sunucu "şifreleme tipi değişikliği" mesajı aldığında askıda olan durumunu okuma durumunaa getirir. İstemci ya da sunucu "şifreleme tipi değişikliği" mesajı gönderdiğinde de askıda olan yazma durumunu , çalışır duruma getirir. El sıkışma işlemi bittiğinde istemci ve sunucu "şifreleme tipi değişikliği" mesajlarını birbirlerine gönderirler ve geri kalan iletişimlerini de bu yeni yapıya göre sürdürürler.

Kullanılan Yapılar:

- **Oturum Id'si:** Sunucu tarafından oluşturulan oturum id'sidir.
- **Sertifika:** Bu sertifika X509 v3 tipinde bir sertifikadır.
- **Sıkıştırma Algoritması:** Şifrelenmeden önce verinin hangi algoritma ile sıkıştırılacağını belirleyen elemandır.
- **Şifreleme Algoritması:** Verilerin hangi algoritma ile şifreleneceğini , hangi hash algoritmasının kullanılacağını ve hash boyutunu belirleyen elemandır.
- **En Yüksek Dereceli Gizli Bilgi (Master secret):** 48 byte lık bir veri olup istemci ve sunucu arasında paylaşılır. Bu veri ile anahtar oluşturma gibi işlemler yapılır.
- **Devam Edilebilirlik Bayrağı:** Bu bayrak ile oturumun yeni bağlantı başlatmak için kullanılıp kullanılamayacağını belirler.

3.3.1.2. SSL Bağlantısı

Bağlantılar 2 bilgisayar arasında kurulur ve kısa sürelidirler. Her bağlantı bir oturum ile birlikte çalışır. Bağlantı durumları alttaki yapılarla belirlenir.

- **İstemci Sunucu Rastgele Sayıları:** Bu sayılar her bağlantı için istemci ve sunucunun seçtiği byte serileridir.
- **Sunucu MAC anahtarı:** sunucunun veri gönderirken MAC işlemi sırasında kullandığı anahtardır.
- **İstemci MAC anahtarı:** istemcinin veri gönderirken MAC işlemi sırasında kullandığı anahtardır.
- **Sunucu yazma anahtarı:** sunucu tarafında şifrelemede, istemci tarafında da şifre çözme işlemi sırasında kullanılan anahtardır.
- **İstemci yazma anahtarı:** istemci tarafında sunucuya gönderilecek verilerin şifrelenmesinde, sunucu tarafında da istemciden gelen verileri şifrelerinin çözülmesinde kullanılan anahtardır.
- **Başlangıç vektörleri:** CBC modunda şifreleme yapılığında kullanılır.
- **Sıra numarası:** istemci ya da sunucular gönderilen ve alınan mesajların herbiri için bir sıra numarası vermelidirler. Şifreleme algoritması değiştirildikten sonra gönderilen ilk mesajın sıra numarası 0 olarak belirlenir.

3.3.1.3. SSL Kayıt Protokolü

Kayıt protokolü üst katmanlardan gelen mesajı bloklara böler , tercihe bağlı olarak blokları sıkıştırır , MAC değerlerini hesaplar , şifreler , mesaja bir başlık ekler ve oluşan yeni mesajı TCP protokolüne gönderir. Aynı protokol alıcı bilgisayarda TCP'den gelen mesajı ilk haline çevirip üst katmanlara iletir.

- **Mesajın bloklara ayrılması:** Üst katmanlardan gelen mesaj 2^{14} byte ya da daha küçük boyutlara ayrılır. Bu yapıya SSLPlaintext yapısı adı verilir.
- **Sıkıştırma ve Sıkıştırılmış Veriyi İlk Haline Getirme:** Çalışmakta olan durumda belirtilen sıkıştırma algoritması kullanılarak veriler sıkıştırılır. Sıkıştırılmış mesajın yapısı SSLCompressed yapısı adını alır.
- **MAC:** Şifreleme işleminden önce mesajın MAC değeri hesaplanır. MAC hesaplama işlemi hash algoritmaları ile yapılmaktadır.
- **Şifreleme:** Şifreleme işlemi asıl mesaja eklenmiş MAC verisi üzerinden DES, TripleDES, IDEA, RC5 ya da Fortezza algoritmaları kullanarak yapılmaktadır.
- **Mesaja başlık eklenmesi:** Başlığa eklenecek veriler ve uzunlukları alttaki gibidir:
  - **İçerik Tipi 8 Bit:** Mesaj hangi üst protokolden geldi ise mesajın gönderildiği bilgisayarda da o protokol tarafından alınıp işlenmelidir. Bu veri sayesinde mesajın hangi protokol tarafından işleneceği belirlenir.
  - **En Yüksek SSL Versiyonu 8 Bit:** Kullanılan en yüksek SSL protokolü versiyonunu belirtir.
  - **En Düşük SSL Versiyonu 8 Bit:** Kullanılan en düşük SSL protokolü versiyonunu belirtir.
  - **Blok Boyutu 16 bit:** Eğer veri sıkıştırması kullanılmamışsa blok boyutu, sıkıştırma uygulanmışsa sıkıştırılmış verinin boyutunu içerir Şekil(3.5).

<table>
<tr><td>İçerik tipi</td><td>En yüksek SSL versiyonu</td><td>En düşük SSL versiyonu</td><td>Blok boyutu</td><td></td></tr>
<tr><td colspan="4">Sıkıştırılmış ya da orjinal veri</td><td rowspan="2">Şifrelenecek mesaj alanları</td></tr>
<tr><td colspan="4">MAC değeri(MD5 ya da SHA-1 algoritması ile oluşturulmuş)</td></tr>
</table>

Şekil 3.5: SSL Mesaj Yapısı

3.3.1.4. SSL Şifreleme Algoritması Değiştirme Protokolü

SSL protokolleri arasında en basit olan protokoldür. Bir tane mesajdan oluşur ve bu mesaj da sıkıştırıldıktan sonra çalışmakta olan şifreleme algoritması ile şifrelenir. Bu mesaj hem istemci hem de sunucu tarafından gönderilebilir ve mesajı alan taraf bundan sonra gelecek olan mesajların algoritma değişikliği mesajı ile gelen algoritma ile şifreleneceğini anlamalıdır. Beklemede olan durum , çalışmakta olan durumun yerine geçer ve bağlantının şifreleme tipini değiştirir.

3.3.1.5. SSL Alarm Protokolü

Alarm mesajlarını oluşturan protokoldür. İki byte'tan oluşurlar. İlk byte'da alarma sebep olan hatanın ölümcül bir hata mı yoksa sadece uyarı hatası mı olduğunu yani hatanın ciddiyetini belirleyen değer bulunur. Eğe hata ölümcül ise SSL bağlantıyı hemen keser. İkinci byte'ta da hata kodu bulunmaktadır. Hata mesajları da diğer SSL mesajları gibi şifreli olarak iletilmektedirler.

SSL'de ölümcül hatalara sebep olan faktörler:

- Beklenmeyen mesaj
- Hatalı MAC değeri
- Sıkıştırılmış dosyanın eski haline getirilmesi sırasında hatalı bir veri ile karşılaşılması
- Sertifika isteği yapıldığı halde cevabın gelmemesi ya da sertifikanın olmaması
- Gelen sertifikanın doğrulanması sırasında hata oluşması
- Sertifika tipinin desteklenmemesi
- Sertifikayı yayınlayan kurumca geri alınmış olması
- Sertifikanın geçerlilik tarihinin dolması
- Yayıncısı bilinmeyen sertifika
- El sıkışma sırasında hatalı bilgi gelmesi
- Kapatma bilgilendirme mesajı gelmesi , bu mesaj mesaj gönderen ya da alan tarafın bu bağlantı üzerinden herhangi bir veri aktarmayacığını belirtmek için gönderdiği bir mesajdır.

3.3.1.6. SSL El Sıkışma Protokolü

Bu protokol SSL protokolünün en önemli parçasıdır. Şekil (3.6)'da protokolün gerçekleştirdiği işlemler görülebilir.

3 ayrı görevi yerine getirmektedir.

- İstemci ve sunucunun bir protokol versiyonu üzerinde anlaşmasını sağlar.
- MAC oluşturarak istemci ve sunucunun birbirlerini doğrulamasını sağlar.
- Bir şifreleme algoritması ve bu algoritmada kullanılacak anahtar üzerinde istemci ve sunucunun birbirleriyle anlaşmasını sağlar.

SSL El Sıkışması İçin Gereken Mesajlar:

1. Faz: Fiziksel Bağlantı İçin "Merhaba Mesajları"

İstemci bağlanmak istediği sunucuya bir "merhaba" mesajı gönderir ve sunucudan da "merhaba" cevabı bekler. Merhaba mesajları içerisinde

- İstemci ve sunucunun rastgele ürettiği birer sayı
- Oturum ID'si
- Şifreleme algoritması
- SSL protokol versiyonu
- Sıkıştırma algritması

Bilgileri bulunmaktadır. Böylece merhaba mesajları ile sadece fiziksel bağlantı kurulmaz, aynı zamanda bundan sonra gidecek olan mesajların yapıları ve şifreleme algoritmaları da belirlenmiş olur.

- **Merhaba İsteği:** Sunucu tarafından herhangi bir zamanda istemciye gönderilebilir. Eğer sunucu ve istemci zaten merhaba mesajları ile bağlantı kurmaya çalışıyorlarsa istemci bu mesajı reddeder.
- **İstemci Merhaba Mesajı:** İstemci sunucudan gelen merhaba mesajını aldıktan sonra üzerinden bağlantı kurulacak oturum ID'sini merhaba mesajı ile gönderir. Bu oturum daha önceden istemci ve sunucu arasında açılmış olmalıdır. Sunucu oturum ID'sini kontrol eder ve eğer kendisi ile istemci arasında daha önceden böyle bir oturum açılmış olduğuna karar verirse aynı oturum ID'sini içeren başka bir merhaba mesajını istemciye gönderir. İstemci merhaba mesajında alttaki parametreler bulunur
  - İstemci SSL versiyonu

  - 28 byte'lık rastgele üretilmiş sayı
  - Oturum ID'si
  - İstemcinin destekleyeceği şifreleme algoritmaları
  - İstemcinin desteklediği sıkıştırma metodları
- **Sunucu Merhaba Mesajı:** Sunucu istemciden gelen merhaba mesajındaki parametreleri onaylar ise istemciye bir merhaba mesajı gönderir. Bu mesajda aşağıdaki bilgiler bulunur.
  - Server SSL versiyonu
  - İstemci rastgele sayısından farklı bir rastgele sayı
  - Oturum ID'si
  - Seçilen şifreleme algoritması
  - Seçilen sıkıştırma metodu

2. Faz: Sunucu Doğrulama ve Anahtar Değişimi:

Eğer sunucunun doğrulanması gerekiyorsa , merhaba mesajlarından sonra sunucu sertifikasını istemciye gönderir.

- **Sunucu Sertifikası:** Sunucu merhaba mesajından hemen sonra gönderilir. İstemcinin ve sunucunun anlaştığı şifreleme algoritmalarına uygun bir sertifka gönderilmiş olmalıdır.
- **Sunucu Anahtar Değişim Mesajı:** Sadece gerektiğinde gönderilir. Eğer sunucu sertifikası Diffi-Hellman parametreleri içeriyorsa ya da RSA anahtar değişimi uygulanmaya karar verilmişse bu mesaj gönderilmez.
  - **Parametreler:** Sunucunun anahtar değişim parametreleri
  - **İmzalanmış Parametreler:** Hash değerinin imzalanması ile oluşan değerin parametrelere eklenmesidir.
- **Sertifika İsteği Mesajı:** Sertifika tipi ve otoritesini içeren bir mesajdır.
  - **Sertifika Tipi:** Sunucunun destekleyebileceği sertifika tipleri listesidir.
  - **Sertifika Otoritesi:** Sunucunun tercih ettiği sertifika otoritelerinin listesidir.
- **Sunucu Merhaba Mesajlarının Sonu Mesajı:** Bu mesaj ile sunucu tüm merhaba mesajlarının sonuçlandığını istemciye göndermiş olur.

3. Faz : İstemci Doğrulama ve Anahtar Değişimi:

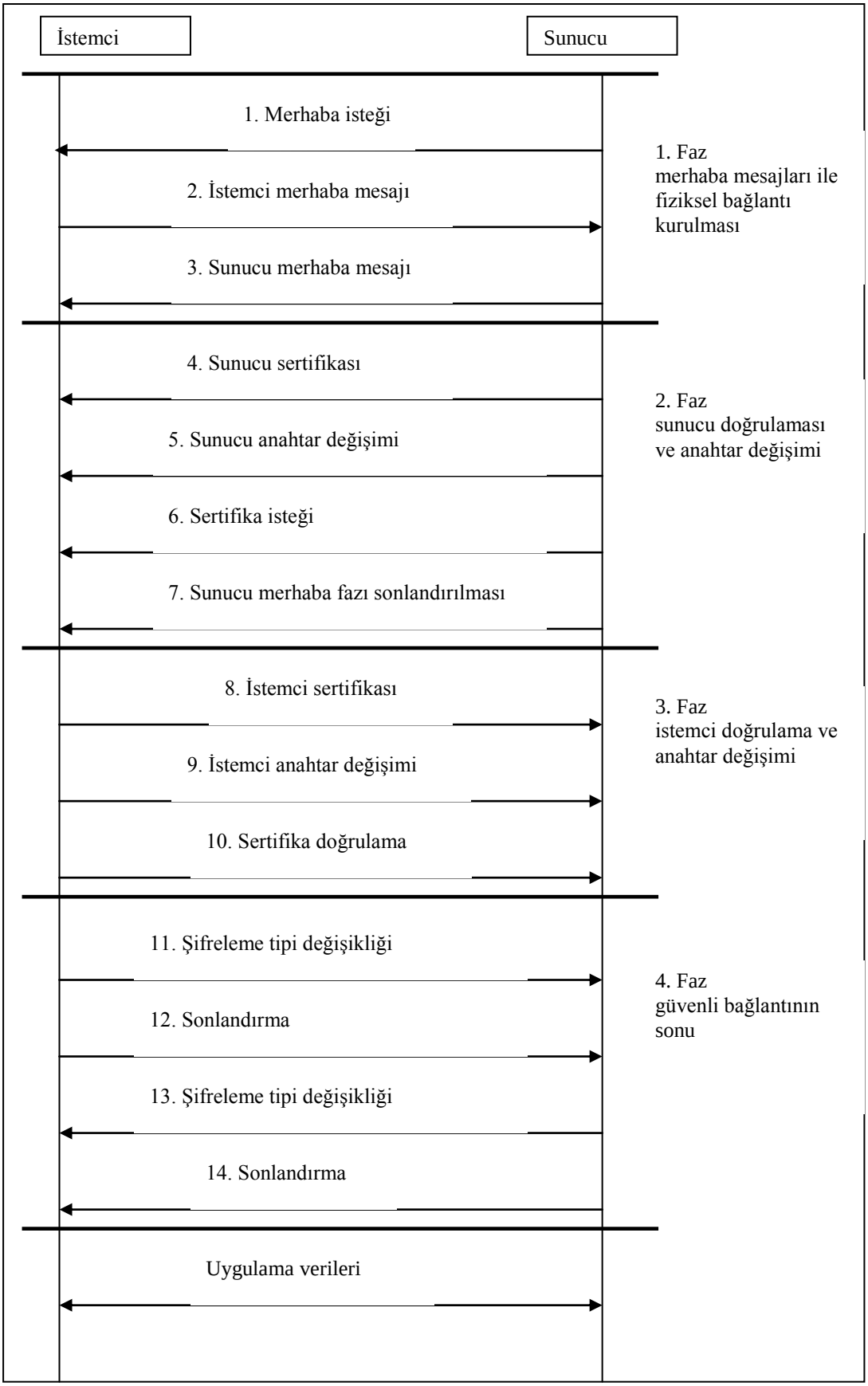
Eğer sunucu bir sertifika isteği yapmışsa, istemci sunucuya sertifika göndermek zorundadır. Daha sonra istemci anahtar değişim mesajı gönderir.

- **İstemci Sertifikası:** Sunucu merhaba mesajları sonu mesajını gönderdikten sona istemci kendi sertifikasını gönderebilir. Eğer sunucudan gelen sertifika tipi listesi ya da sertifika otoritesi listesindeki parametreler ile kendi sertifikaları uyumlu ise sertifika gönderilir.
- **İstemci Anahtar Değişimi:** Bu mesaj ile premaster(muhtemelen simetrik bir anahtar) anahtarı asimetrik bir algoritma ile şifrelenerek sunucuya gönderilir. Bu şekilde premaster anahtarı hem sunucuda hem de istemcide saklanır.
- **Sertifika Doğrulama Mesajı:** İstemci sertifikasının imzalanması ile elde edilir.

4. Faz: Güvenli Bağlantının Sonu:

Bu mesajı gönderdikten sonra istemci beklemede olan şifreleme metodunu çalışır duruma getirir.

- **Şifreleme Metodu Değiştirme Mesajı:** İstemci şifreleme metodu değişim mesajı gönderir ve gönderdiği mesajdaki metodu kullanmaya başlar.
- **Sonlandırma:** Bu mesaj ile doğrulama mesajı anahtar değişim ve doğrulama işlemlerinin istemci tarafında başarı ile bitirildiğini belirtmek üzere gönderilir.



Şekil 3.6: SSL İletişim Adımları

3.3.2 SET (Güvenli Elektronik İşlem) Protokolü

Bu protokolde işlem beş adımda gerçekleşir.

3.3.2.1 Kart Sahibi CA Kaydı

Bu işlem kart sahibi ve CA arasında yürütülen bir işlemdir. SET protokolünden yararlanmak isteyen bir kullanıcı önce bir CA seçmeli ve bu CA'ya kayıt olmalıdır. Ya da bağlı olduğu finansal kurumun tercih ettiği bir CA'ya kayıt olabilir. Kayıt işlemi Tablo (3.1)'de ayrıntılı olarak açıklanmıştır

Tablo 3.1: Kart Sahibi CA Kaydı İçin Yapılan İşlemler ve Gönderilen Mesajlar

Kart Sahibi		**CA**
1. Sertifika isteği başlangıç mesajı	→	
	←	2. CA anahtar değişim sertifikası CA sayısal imza sertifikası 3. CA tarafından imzalanmış cevap mesajı
4. Gelen sertifikalar doğrulanır 5. Sayısal imza ile gelen cevap mesajı doğrulanır 6. Müşteri bilgileri sisteme kaydedilir 7. Başvuru formu isteği mesajı oluşturulur 8. 1# nolu rastgele simetrik bir anahtar oluşturulur. 9. Bu anahtar ile başvuru formu isteği şifrelenir. 10. 1# anahtarı ve müşteri bilgisi CA anahtar değişim açık anahtarı ile şifrelenir.		
11. CA anahtar değişim açık anahtarı ile şifrelenmiş 1# anahtar + müşteri bilgileri 12. 1# anahtar ile şifrelenmiş başvuru formu isteği	→	
		13. 1# anahtarı ile müşteri bilgileri deşifre edilir. 14. 1# anahtarı ile başvuru formu isteği mesajı deşifre edilir. 15. Uygun bir başvuru formu oluşturulur ve bu form sayısal imzalanır.
	←	16. Sayısal imzalanmış başvuru formu 17. CA sayısal imza sertifikası
18. CA sayısal imza sertifikası doğrulanır. 19. Başvuru formunun doğruluğu kontrol edilir. 20. Sayısal imzada kullanmak üzere bir açık bir de gizli anahtar oluşturulur. 21. Başvuru formu doldurulur. 22. Başvuru formundaki bilgilerle birlikte bir sertifika isteği oluşturulur. 23. #2 nolu simetrik anahtar rastgele oluşturulur. 24. Sertifika isteği + #2 nolu anahtar + sayısal imza açık anahtarı , kart sahibi sayısal imza gizli anahtarı ile imzalanır.		

25. #3 simetrik anahtarı rastgele oluşturulur. 26. Kart sahibi sayısal imza gizli anahtarı ile imzalanmış sertifika isteği + #2 nolu anahtar + sayısal imza açık anahtarı #3 anahtarı ile şifrelenir.		
27. Kart sahibi sayısal imza gizli anahtarı ile imzalanmış ve #3 anahtarı ile şifrelenmiş (sertifika isteği + #2 nolu anahtar + sayısal imza açık anahtarı) 28. CA anahtar değişim açık anahtarı ile şifrelenmiş müşteri bilgisi + 3# anahtarı.	→	
		29. CA #3 anahtarını ve müşteri bilgisini deşifre eder. 30. #3 anahtarı ile sertifika isteği + #2 anahtarı ve kart sahibi sayısal imza açık anahtarı deşifre edilir ve kart sahibi sayısal imza açık anahtarı ile kart sahibinin sayısal imzası doğrulanır. 31. Kayıt formu ve müşteri bilgileri kontrol edilerek sertifika isteği onaylanır. 32. Kart sahibi sertifikası kart sahibi sayısal imza açık anahtarı bilgisi CA sayısal imza gizli anahtarı ile imzalanarak oluşturulur. 33. Sertifika cevabı oluşturulur ve sayısal imzalanır.
	←	34. #2 anahtarı ile şifrelenmiş sayısal imzalı sertifika cevabı 35. Kart sahibi sayısal imza sertifikası 36. CA sayısal imza sertifikası
37. Kart sahibi sayısal imza sertifikası doğrulanır 38. Sertifika cevabı #2 anahtarı ile deşifre edilerek kontrol edilir. 39. CA sayısal imza sertifikası doğrulanır. 40. Kart sahibi sertifikası ve sertifika cevabı saklanır.		

3.3.2.2 İşyeri CA Kaydı

Bu adımda işyeri , kendi açık anahtarını bir sertifika içinde yayınlamak istediğinden, açık anahtarı sertifika haline getirecek kurum olan CA'ya kayıt olur. Kayıt işlemi Tablo(3.2) 'deki sıra ile gerçekleşir.

Tablo 3.2: İşyeri CA Kaydı İçin Yapılan İşlemler ve Gönderilen Mesajlar

İşyeri		**CA**
41. Sertifika isteği başlangıç mesajı	→	
	←	42. CA tarafından imzalanmış başvuru formu 43. CA sayısal imza sertifikası 44. CA anahtar değişim sertifikası
45. CA sertifikaları doğrulanır 46. Başvuru formu sayısal imzası doğrulanır. 47. Bir tanesi sayısal imza bir tanesi de anahtar değişiminde kullanılmak üzere 2 farklı açık - gizli anahtar çifti oluşturulur. 48. Başvuru formu doldurulur		

49. Sertifika isteği mesajı oluşturulur. 50. 1# simetrik anahtarı rastgele bir şekilde üretilir. 51. İşyeri sayısal imza gizli anahtarı ile imzalanmış sertifika isteği mesajı + sayısal imza açık anahtarı + anahtar değişim açık anahtarı bilgisi 1# anahtarı ile şifrelenir.		
52. CA anahtar değişim açık anahtarı ile şifrelenmiş 1# anahtarı + işyeri bilgileri 53. 1# anahtarı ile şifrelenmiş işyeri sayısal imza gizli anahtarı ile imzalanmış (sertifika isteği mesajı + işyeri sayısal imza açık anahtarı + işyeri anahtar değişim açık anahtarı) bilgisi.	→	
		54. #1 anahtarı deşifre edilir. 55. 1# anahtarı ile (sertifika isteği mesajı + işyeri sayısal imza açık anahtarı + işyeri anahtar değişim açık anahtarı) bilgileri deşifre edilir. 56. Bu bilgilerin işyeri sayısal imza gizli anahtarı ile oluşturulmuş sayısal imzası doğrulanır. 57. İşyeri bilgileri kontrol edilerek sertifika isteği onaylanır. 58. işyeri sayısal imza açık anahtarı + işyeri anahtar değişim açık anahtarı bilgilerini CA sayısal imza gizli anahtarı ile imzalayarak işyeri sertifikalarını oluşturur. 59. Sertifika cevabı da aynı şekilde imzalanır.
	←	60. İmzalanmış sertifika cevabı 61. İşyeri anahtar değişim sertifikası 62. İşyeri sayısal imza sertifikası 63. CA sayısal imza sertifikası
64. Sertifikalar doğrulanır. 65. Sertifika cevabının sayısal imza kontrolü yapılır. 66. Sertifikalar sonraki işlemlerde kullanılmak üzere saklanır.		

3.3.2.3 Alışveriş İşlemi

Bu adım kart sahibi ve işyeri arasında gerçekleşir. Tablo (3.3)'te alışveriş işleminin ayrıntıları yer almaktadır.

Tablo 3.3: Alışveriş İşlemi İçin Yapılan Hesaplamalar ve Gönderilen Mesajlar

Kart Sahibi		**İşyeri**
67. İşlem başlangıç mesajı	→	
		68. İşyeri kart sahibine cevap mesajı oluşturur ve bu mesajı sayısal imza gizli anahtarı ile imzalar.
	←	69. Sayısal imzalı cevap mesajı 70. İşyeri sayısal imza sertifikası

		71. Ödeme sistemi anahtar değişim sertifikası
72. Sertifikalar doğrulanır. 73. Cevap mesajının imzası doğrulanır 74. Sipariş bilgisi (OI) hazırlanır. (Bu bilgi içerisinde hangi maldan ne kadar fiyatta alındığı bilgisi yoktur. Bu bilgiler ilk set mesajı gönderilmeden önce kart sahibi ve işyeri tarafından belirlenmiş olmalıdır.) 75. Ödeme bilgisi (PI) hazırlanır. 76. OI ve PI ayrı ayrı kart sahibi tarafından iki defa imzalanır. 77. İmzalanmış PI rastgele oluşturulan #1 anahtarı ile şifrelenir.		
78. #1 anahtarı ile şifrelenmiş imzalı PI 79. İmzalı OI 80. Ödeme sistemi anahtar değişim açık anahtarı ile şifrelenmiş 1# ve müşteri bilgisi 81. Kart sahibi sayısal imza sertifikası	→	
		82. Kart sahibi sertifikası doğrulanır. 83. OI 'nın çifte imzası doğrulanır. 84. 1# anahtarı ve müşteri bilgisi deşifre edilir. 85. 1#anahtarı ile PI imzalı verisi deşifre edilir ve imzası doğrulanır. 86. PI bilgisi ödeme sistemine otorizasyon alması için gönderilir. 87. İşlem cevabı oluşturulur ve bu cevap sayısal imzalanır.
	←	88. İmzalı işlem cevabı 89. İşyeri sayısal imza sertifikası
90. İşyeri sertifikası doğrulanır. 91. İşlem cevabının imzası kontrol edilir ve işlem cevabı saklanır.		

3.3.2.4 Otorizasyon İşlemi

Bu adım işyeri ve ödeme sistemi arasında gerçekleşir. Tablo (3.4)'te otorizasyon işleminin ayrıntıları yer almaktadır.

Tablo 3.4: Alışveriş İşlemi İçin Yapılan Hesaplamalar ve Gönderilen Mesajlar

İşyeri		**Ödeme Sistemi**
92. Otorizasyon isteği oluşturulur. 93. Bu istek işyeri sayısal imza gizli anahtarı ile imzalanır. 94. 2# anahtarı oluşturulur. 95. 2# anahtarı ile imzalı otorizasyon isteği şifrelenir. 96. 2# anahtarı ödeme sistemi anahtar değişim açık anahtarı ile şifrelenir.		
97. Ödeme sistemi anahtar değişim açık anahtarı ile şifrelenmiş 2# anahtarı. 98. İşyeri tarafından imzalanmış ve 2# anahtarı ile şifrelenmiş otorizasyon isteği mesajı 99. Ödeme sistemi anahtar değişim açık anahtarı ile şifrelenmiş 1# anahtarı +	→	

müşteri bilgisi 100. Kart sahibi tarafından çifte imzalanmış ve 1# ile şifrelenmiş PI bilgisi 101. Kart sahibi sayısal imza sertifikası 102. İşyeri sayısal imza sertifikası 103. İşyeri anahtar değişim sertifikası		
		104. İşyeri sertifikaları doğrulanır 105. 2# anahtarı deşifre edilir ve 2# anahtarı ile otorizasyon isteği mesajı deşifre edilir. 106. Otorizasyon mesajındaki işyeri imzası doğrulanır. 107. Kart sahibi sertifikası doğrulanır. 108. 1# anahtarı ile müşteri bilgisi deşifre edilir. 109. 1# anahtarı ile de çifte imzalı PI bilgisi deşifre edilir. 110. Çifte imzalı PI bilgisinin imzaları doğrulanır. 111. Ödeme sistemi PI bilgisi ile işyeri otorizasyon isteğinin uyumluluğunu doğrular. 112. Otorizasyon isteği finansal bir ağ üzerinden kart sahibinin bağlı olduğu finansal kuruma gönderilir. 113. Finansal kurumdan gelen cevaba göre ödeme sistemi bir cevap mesajı oluşturur ve bu cevabı ödeme sistemi sayısal imza gizli anahtarı ile imzalar. 114. 3# anahtarını oluşturur 115. 3# anahtarı ile müşteri bilgisi verileri **işyeri anahtar değişim açık anahtarı** ile şifrelenir. 116. Bir capture token oluşturulur ve bu tokenı sayısal imzalanır. 117. 4# anahtarı ile bu token şifrelenir. 118. 4# anahtarı ve müşteri bilgisi **ödeme sistemi anahtar değişim açık anahtarı** ile şifrelenir.
	←	119. İşyeri anahtar değişim açık anahtarı ile şifrelenmiş 3# anahtarı + müşteri bilgisi 120. 3# anahtarı ile şifrelenmiş ödeme sistemi gizli anahtarı ile imzalanmış otorizasyon cevabı 121. İşyeri anahtar değişim açık anahtarı ile şifrelenmiş 4# anahtarı + müşteri bilgisi 122. 4# anahtarı ile şifrelenmiş ve ödeme sistemi gizli anahtarı ile imzalanmış ÖDEME FİŞİ 123. Ödeme sistemi sayısal imza sertifikası.
124. Sertifika doğrulaması yapılır. 125. İşyeri anahtar değişim açık anahtarı ile şifrelenmiş 3# anahtarı ve müşteri bilgileri işyeri anahtar değişim gizli anahtarı ile deşifre edilir. 126. 3# şifresi ile de otorizasyon cevabı deşifre edilir. 127. Deşifre edilen otorizasyon cevabının ödeme sistemi tarafından atılan sayısal imzası doğrulanır. 128. Şifreli ÖDEME FİŞİ daha sonra		

kullanılmak üzere saklanır. 129.İşyeri alışveriş işlemini tamamlar.		

3.3.2.5 Ödeme İşlemi

Bu adım işyeri ve ödeme sistemi arasında gerçekleşir. Tablo (3.5)'te otorizasyon işleminin ayrıntıları yer almaktadır.

Tablo 3.5: Ödeme İşlemi İçin Yapılan Hesaplamalar ve Gönderilen Mesajlar

İşyeri		**Ödeme Sistemi**
130.Ödeme isteği oluşturulur. 131.Kendi sertifikalarını da bu mesaja ekleyen işyeri , bu veriyi sayısal imzalar. 132.5# anahtarını oluşturur ve sayısal imzaladığı ödeme isteğini bu anahtar ile şifreler. 133.5# anahtarını daha sonra ödeme sistemi anahtar değişim açık anahtarı ile şifreler.		
134.5# anahtarı ile şifrelenmiş ödeme isteği 135.Ödeme sistemi anahtar değişim açık anahtarı ile şifrelenmiş 5# anahtarı. 136.4# ile şifrelenmiş ÖDEME FİŞİ 137.Ödeme sistemi anahtar değişim açık anahtarı ile şifrelenmiş 4# anahtarı ve müşteri bilgisi 138.İşyeri anahtar değişim sertifikası 139.İşyeri sayısal imza sertifikası	→	
		140.İşyeri sertifikaları doğrulanır. 141.Ödeme sistemi anahtar değişim gizli anahtarı ile 5# deşifre edilir ve 5# ile de ödeme isteği mesajı deşifre edilir. 142.Ödeme isteğinin işyerince atılan sayısal imzası doğrulanır. 143.Ödeme sistemi anahtar değişim gizli anahtarı ile 4# deşifre edilir ve 4# ile de ÖDEME FİŞİ deşifre edilir. 144.Ödeme isteği ve ÖDEME FİŞİ'nin uyumluluğu doğrulanır. 145.Ödeme sistemi ödeme isteği mesajını kart sahibinin finansal kurumuna finansal bir ağ üzerinden gönderir. 146.Ödeme sistemi ödeme cevabı oluşturur . 147.Ödeme cevabı ile ödeme sistemi sayısal imza sertifikasını sayısal imzalar. 148.6# anahtarı ile sayısal imzalı ödeme cevabını şifreler. 149.6# anahtarını da işyeri anahtar değişim açık anahtarı ile şifreler.
	←	150.İşyeri anahtar değişim açık anahtarı ile şifrelenmiş 6# 151.6# ile şifrelenmiş ödeme sistemi sayısal imzalı ödeme cevabı. 152.Ödeme sistemi sayısal imza sertifikası
153.Ödeme sistemi sertifikası doğrulanır. 154.6# anahtarını işyeri anahtar değişim gizli anahtarı ile deşifre eder. 155.Ödeme cevabı mesajını 6# ile deşifre		

eder. 156.Ödeme sistemi imzası doğrulanır.		

3.3.2.6 SET Protokolü Analizi

SET protokolü ile işlem yapabilmek için her adımda yapılması gereken şifreleme işlemlerinin sayısı Tablo (3.6)'da belirtilmektedir.

SET protokolünde işlemler PIN'siz yapıldığından , müşteri bilgilerini bilen herhangi birisi kolaylıkla müşteri gibi davranarak alışveriş yapabilmektedir. Bunun dışında SET protokolünde anahtar değişimi güvenli bir şekilde yapılsa da, bir çok mesaj simetrik anahtarlar altında şifrelenip gönderilmektedir. Simetrik şifreler asimetrik şifrelere göre daha kolay kırılabilmektedir. Şifrelenmiş verinin tahmin edilebilir olması da şifrenin kırılmasını kolaylaştırmaktadır.

SET protokolünde çok fazla şifreleme, anahtar oluşturma gibi işlemler yapıldığı gibi bir de oldukça fazla mesaj trafiği yaşanmaktadır. Ayrıca PKI sistemi kullanıldığından, bir kullanıcı (işyeri ya da müşteri) işlem yapmadan önce mutlaka bir CA'den sertifika alması gerekmektedir. Bu da SET protokolü ile işlem yapmak için gereken mesaj trafiğini çoğaltmaktadır.

Tablo (3.6) SET Protokolü İşlem Analiz Bilgileri

	1. Adım	**2. Adım**	**3. Adım**	**4. Adım**	**5. Adım**	**Toplam**
Sertifika Oluşturma	3	2	1	0	0	**6**
Hash, Sayısal imzalama , sayısal imzakontrolü	10	7	10	8	7	**42**
Simetrik şifreleme, şifre çözme	6	2	1	5	5	**19**
Asimetrik anahtar oluşturma	1	2	0	2	0	**5**
Simetrik anahtar oluşturma	3	1	1	0	2	**7**
Asimetrik şifreleme ,şifre çözme	4	2	1	5	5	**17**

3.3.3 3D SECURE Protokolü

3D SECURE Protokolü internet üzerinden alışveriş yapılabilmesi için VISA tarafından geliştirtirilmiş bir protokoldür. İnternet üzerinden yapılan işlemlerde Türkiye'de de kullanılmaktadır.

3.3.3.1 3D SECURE Özellikleri

1. Uzaktan yapılan ödemeler için global bir iskelet oluşturur[10].
2. İzinsiz yapılan işlemlerdeki geri ödemeleri azaltarak operasyonel yükü azaltır[10].
3. Kart sahibinin bu işe özel bir donanıma ya da yazılıma sahip olması gerekmez[10].
4. Müşteri yönetimi ve güvenlik için issuer tarafından geliştirme yapılması acquirer tarafını etkilemez[10].
5. Mobil telefonlar, PDA'lar ve sayısal TV'lerle birlikte kullanılabilecek bir yapısı vardır[10].
6. Anlaşmazlık çıkan noktalarda sorunu kolayca bulabilmek için merkezi bir arşive sahiptir[10].

3.3.3.2 Üçlü Alan Yapısı

Issuer Alanı: Issuer sistemleri, fonksiyonları ve kart sahipleri

Acquirer Alanı: Acquirer sistemleri, fonlsiyonları ve müşteriler

Interoperability Alanı: Dünyanın her yerindeki issuerlar ile acquirerların uyumlu bir şekilde birbirleri ile çalışmalarını sağlayan sistemler, fonksiyonlar ve mesajlar [10].

VISA'nın oluşturduğu bu modelde Issuer alanında kart sahibi ve kullandığı bilgisayar, PDA, cep telefonu ve issuer finansal kurum yer almaktadır. Interoperability alanında ise VISA kurumu bulunur. Acquirer alanında da işyeri ve acquirer bulunmaktadır.

3.3.3.3 3D Secure'un Temel Fonksiyonları

1. Issuer ve acquirer arasında işlem izni mesaj isteği ve işlem izni mesajlarının interoperability alanında iletilmesi.
2. Issuer ile kart sahibi arasında kart sahibi doğrulaması için issuer alanında mesajların iletilmesi.

3. İşyeri ve acquirer arasında doğrulama ve ödeme işlemi mesajlarının acquirer alanında iletilmesi
4. Acquirer ve issuer arasında doğrulama ve ödeme işlem mesajlarının VisaNet üzerinden iletilmesi.

3.3.3.4 3D Secure Yapısında Kullanılan Elemanlar

Issuer Alanı Dahilindeki Elemanlar:

1. **Kart sahibi:** Kart sahibi online alışveriş yaparken, müşteri ismi, kart numarası, vade tarihi bilgilerini kullandığı internet browser'a ya da cep telefonundaki elektronik cüzdan yazılımına bilgileri girer. Satış işlemi doğrulama sayfasında da doğrulama için gerekli bilgileri girer.
2. **Kart Sahibi Browser'ı:** Browser işyeri server yazılımı ile acquierer alanında, ve işlem izin kontrol server'ı ile de issuer alanında mesajlaşan bir katman olarak görev yapmaktadır.
3. **Kart Sahibi Ek Cihazlar ve Yazılımları:** Chip kart okuyucu, ya da browser için gerekli olan yazılımlar gibi donanım ve yazılımlar da bu gruba girmektedir.
4. **Issuer:** Issuer VISA üyesi bir finansalkurumdur. Kart sahibinin bu kurumun issue ettiği en az bir tane VISA markalı kartı olmalıdır. Kart sahibinin 3D secure servislerini kullanıp kullanamayacağına karar verecek mercidir. Hangi kartların 3D secure yapısında kullanılabileceğini belirler. 3D secure yapısında kullanılabilecek kartları VISA 'ya bildirir. Kart sahiplerinin her kart ya da hesap için 3D secure yapısına kayıt olmasını sağlar.
5. **Erişim Kontrol Sunucusu (ACS):**

   Bir kartın 3D secure sistemine erişip erişemeyeceğine karar verir. Kartın yaptığı işlem için izni olup olmadığını kontrol eder.

Acquirer Alanı Dahilindeki Elemanlar

1. **İşyeri:** Kart sahibi bilgilerini ve işlem için gerekli bilgileri alıp işlem onayı için bunu gerekli yerlere gönderen yapıdır.
2. **İşyeri Server Yazılımı:** İşlem onay mesajları oluşturur. Aynı zamanda otorizasyon cevabı içindeki sayısal imzayı da doğrular.
3. **Acquirer:** Acquirer bir VISA üyesi finansal kurumdur.
   1. VISA kartların kendi üzerinden işlem yapmasının sağlanmasını isteyen işyerleri ile anlaşmalar yapar.

2. İşyerinin 3D secure sistemini kullanacak yapıda olup olmadığına karar verir, hangi işyerlerinin buna uygun hangilerinin uygun olmadığıı belirler.
3. İşyerinden otorizasyon isteği alır.
4. Bu mesajları VisaNet gibi otorizasyon sistemine yönlendirir
5. İşyerine otorizasyon cevabı gönderir.
6. VisaNet gibi ödeme sistemlerine işlemin sonucunu gönderir.

Interoperability Alanı Dahilindeki Elemanlar

1. **Visa Dizin Sunucusu:**
    1. İşyerinden gelen kart numarası sorgu mesajlarını karşılar.
    2. Kart numarasının 3D secure sisteminde kullanılıp kullanılmayacağına karar verir.
    3. Kart sahibi doğrulaması için mesajı Erişim Kontrol Sunucusuna yönlendirir.
    4. ACS'den (Erişim Kontrol Sunucus) doğrulama mesajının cevabını alır.
    5. Cevabı işyerine iletir.
2. **Ticari Sertifika Otoritesi :**3D secure sisteminde kullanılmak üzere SSL ve TLS sertifikası üretir.
3. **VISA Sertifika Otoritesi:** 3D secure sisteminde kullanılmak üzere sayısal imza sertifikası ve Visa kök sertifikalarını üretir.
4. **Geçmiş Doğrulama Bilgileri Sunucusu:** Her doğrulama mesajı için bu server'a ACS'den mesaj gelir.

   Tüm mesajlar daha sonra kontrol edilebilcek şekilde saklanır.
5. **VisaNet:**
    1. Acquirer'dan gelen otorizasyon isteği mesajını karşılar.
    2. Bu mesajı issuera iletir.
    3. İssuerdan gelen cevabı acquirera iletir.
    4. İşlem iptali ya da onaylanmış bir işlemin hesaplaşma(settlement) mesajlarını acquirer ve issuera iletir.

3.3.3.5 Ödeme İşlemi Adımları

Ödeme işleminin adımları Tablo (3.7)'de açıklanmaktadır.

Tablo 3.7: 3D Secure Ödeme İşlemi Adımları

Adımlar	Açıklama
1.	Müşteri işyerinin internet sitesinden alacağı ürünleri ve miktarlarını seçer. Bu aşamada hem müşteri hem de işyeri ürün miktarını ve fiyatını öğrenmiş olur.
2.	İşyeri kart numarası ve kullanıcının internet sitesine erişim cihaz bilgisini Visa Directory Server'a gönderir.
3.	Visa Dizin Sunucusu uygun ACS'ye kart numarası ve cihaz bilgisini gönderir. Böylece ACS'den kullanılan kart numarasının 3D secure sisteminde işlem yapma izni olup olmadığını ve izni varsa bile kullandığı ciaz ile işlem yapmaya izni olup olmadığı kontrolünü yapmasını ister. Eğer kart için uygun bir ACS yok ise o zaman Visa Dizin Sunucusu işlemi reddeder ve bu cevabı işyerine bildirir.
4.	ACS kartın işlem onay bilgisini Visa Dizin Sunucusu'na gönderir.
5.	Visa Dizin Sunucusu ACS'den gelen ya da kendi ürettiği cevabı işyerine iletir.
6.	İşyeri kart sahibinin işlem yaptığı cihaz üzerinden ACS'ye işlem yapan kişiyi doğrulama isteği (PAReq) gönderir.
7.	ACS bu isteği alır.
8.	ACS kart numarası, chip ve PIN doğrulama metodları ile müşteri doğrulaması yapar ve uygun değerlerle bir PARes mesajı oluşturur ve bu mesajı sayısal imzalar.
9.	ACS PARes mesajını müşterinin cihazı üzerinden işyerine gönderir. Bu işlemin kayıt altına alınması için de Geçmiş Doğrulama Bilgileri Sunucusuna gönderir.
10.	İşyeri PARes mesajını alır.
11.	PARes mesajındaki imzanın doğrulamasını yapar.
12.	Eğer işlem işyeri için de uygunsa işlem acquirera işlemin gönderilmesi ile devam eder.
13.	Acquirer gelen otorizasyon isteğini alır , kontrol eder ve işyerine onay ya da red döner.

3.3.3.6 3D Secure Protokolünde Güvenlik

Güvenlik çözümleri:

- **Kart sahibi işlem gerçekleştirilmeden önce doğrulanmalıdır:** Kart sahibi kaydı sırasında bu kontrol yapılmaktadır.
- **Kart sahibi ve issuer arasında güvenli ve herkese açık olmayan bir iletişim kanalı kurulmuş olmalıdır:** SSL/ TLS yapısı kullanılır.

- **Kart ile işlem yapan kişinin o kart ile işlem yapmaya yetklisi olması gerekmektedir:** Kart sahibi ve issuer arasındaki doğrulama işlemleri ile bu sağlanmaktadır.
- **Kart sahibi olmayan tüm elemanarla da güçlü ve güvenli iletişim kanalları oluşturmak:** SSL/TLS yapısı kullanılır.
- **Doğrulama isteği mesajlarının bütünlüğünün sağlanması ve doğrulama cevabının issuerdan başka bir yapıdan gelmediğinin kesinleştirilmesi sağlanmalıdır:** Doğrulama cevabında kullanılan sayısal imza ile sağlanır.

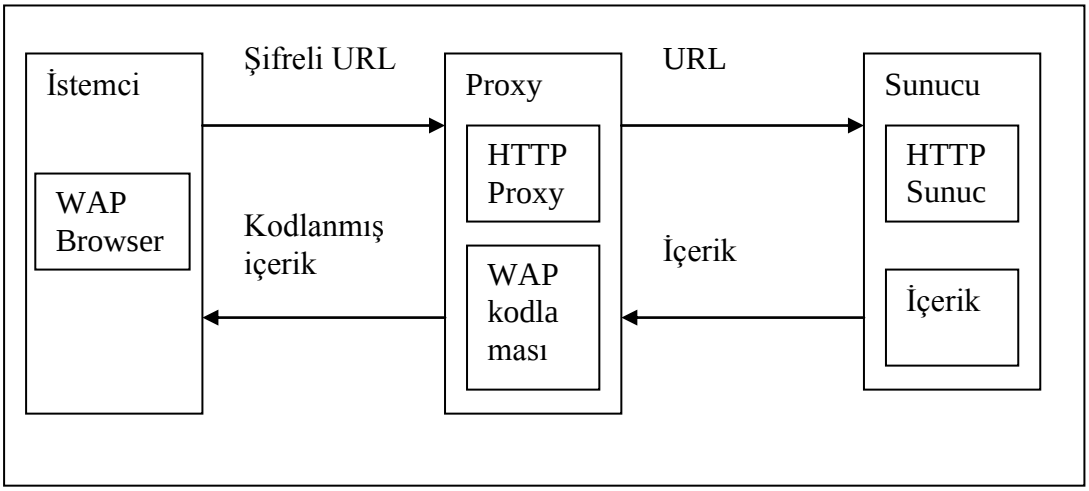
Bunların dışında

- Id ve password kullanılır
- Asimetrik şifreleme algoritmaları ve sayısal sertifikalar.
- Donanımsal güvenlik modülleri
- Visa tarafından yayınlanan prosedürler
- Sayısal imza
- SSL/TLS gibi iletişim sırasında güvenliği sağlayan protokoller

3D secure yapısında güvenliği sağlamak için kullanılan yapılardır.

3.3.4 WAP Protokolü Kullanılarak Mobil Ödeme

Müşteri alışveriş yapmak için bir işyerine gittiğinde WAP üzerinden alışveriş yapması yerine tüm işyerlerinde hazır olarak bulunan kablolu banka bağlantısını otorizasyon için kullanmaları performans açısından oldukça faydalıdır.



Şekil 3.7: WAP Modeli

WAP serverlarına ulaşmak için GPRS gibi mobil operatörlerin sağladığı iletişim protokollerini kullanmak gerekmektedir. Bu da her zaman sağlıklı ve sürekli olmamaktadır. Şekil (3.7)'de WAP Protokolü ile haberleşmek için müşterinin hangi yollardan finansal kurumun sistemine ulaştığı görülmektedir.

4. GÜVENLİ BİR MOBİL TİCARET PROTOKOLÜ VE UYGULAMASI

4.1 Geliştirilen Protokol Genel Tanımlar

4.1.1 Protokolde Kullanılan Yapılar

- Mobil cihaz, cep telefonu, PDA vs.
- İşyerinde Bluetooth özelliği olan bir POS cihazı.
- İşyeri ile finansal kuruluş arasında TCP/IP bağlantısı.
- Finansal kuruluşta işleme onay verecek bir uygulama.
- Mobil cihaza yüklenmiş bir uygulama.

4.1.2 Finansal Kuruluşun Görevleri

Finansal kuruluş kendine ait açık (FIP) ve gizli(FIS) anahtarları üretmeli, gizli anahtarı kendi üzerinde saklamalı, açık anahtarı ise işyeri POS'larına ve mobil cihazlara iletilemek üzere yayınlamalıdır. İşyeri POS'larından gelecek şifrelenmiş verileri düzgün olarak ayırıp şifresini çözecek ve işlemin onaylanıp onaylanmayacağına karar verecek bir otorizasyon mekanizması bulunmalıdır. Finansal kuruluştaki uygulama işyeri POS'larıyla daima bağlantıda kalacak ve mesaj alıp gönderebilecek yetenekte olmalıdır.

4.1.3 Müşteri Tarafından Yapılması Gerekenler

Müşteri bağlı olduğu finansal kuruluşun güvenli bir kanalından mobil cihazı için bir uygulama almalı ve cihazına yüklemelidir. Uygulama cihaza yüklendikten sonra önce kendi gizli(MDS) ve açık(MDP) anahtarlarını üretir daha sonra finansal kuruluşun sunucu uygulaması ile bağlantı kurar ve kendi açık anahtarını bankaya, bankanın açık anahtarını da mobil cihaza yükler. Bunun dışında müşteri bağlı olduğu finansal kuruluştan mobil ödeme PIN'i istemelidir.

4.1.4 POS Cihazında Bulunması Gereken Özellikler

POS cihazı RSA algoritması kullanarak şifreleme ve sayısal imza doğrulama işlemlerini yapabilecek kapasitede olmalıdır. Aynı zamanda Bluetooth ve TCP/IP haberleşmelerini kullanabilecek yapıda olmalıdır. Açık anahtarların güncellenmeleri zaman zaman gerekli olduğundan POS cihazının bu güncellemeyi yapabilecek yapıda olması gerekmektedir. POS cihazı mobil cihaz uygulamasından gelen verileri düzgün bir şekilde ayırabilmelidir.

4.1.5 İşlem Akışı

Müşteri almaya karar verdiği ürünlerin fiyatını öğrenerek cep telefonuna daha önceden yüklediği mobil ödeme uygulamasını açar ve Bluetooth üzerinden işyeri POS cihazına bağlanır.

POS cihazı işyeri numarası (MID) ve işlem tutarını (A) mobil ödeme uygulamasına gönderir ve bu sırada kendisi üzerinde her işlem için tekil olan bir sayı üretir(POS_RN).

Mobil cihaz üzerindeki mobil ödeme uygulaması MID ve A değerlerini alır ve müşterinin mobil ödeme PIN'inin girmesini bekler. Bu işlemden sonra mobil ödeme uygulaması (MPA) kendi içinde her işlem için tekil olan bir sayı üretir(MPA_RN).

İşlem tutarı (A), işyeri ID'si (MID), işlem zamanı (TD), hesap bilgisi (AI), mobil ödeme uygulaması referans numarası (MPA_RN) ve PIN bilgileri finansal kuruluşun açık anahtarı(FIP) altında şifrelenir. Daha sonra bu verileri hepsi daha önceden kararlaştırılmış bir sırayla işyeri POS cihazına gönderilir.

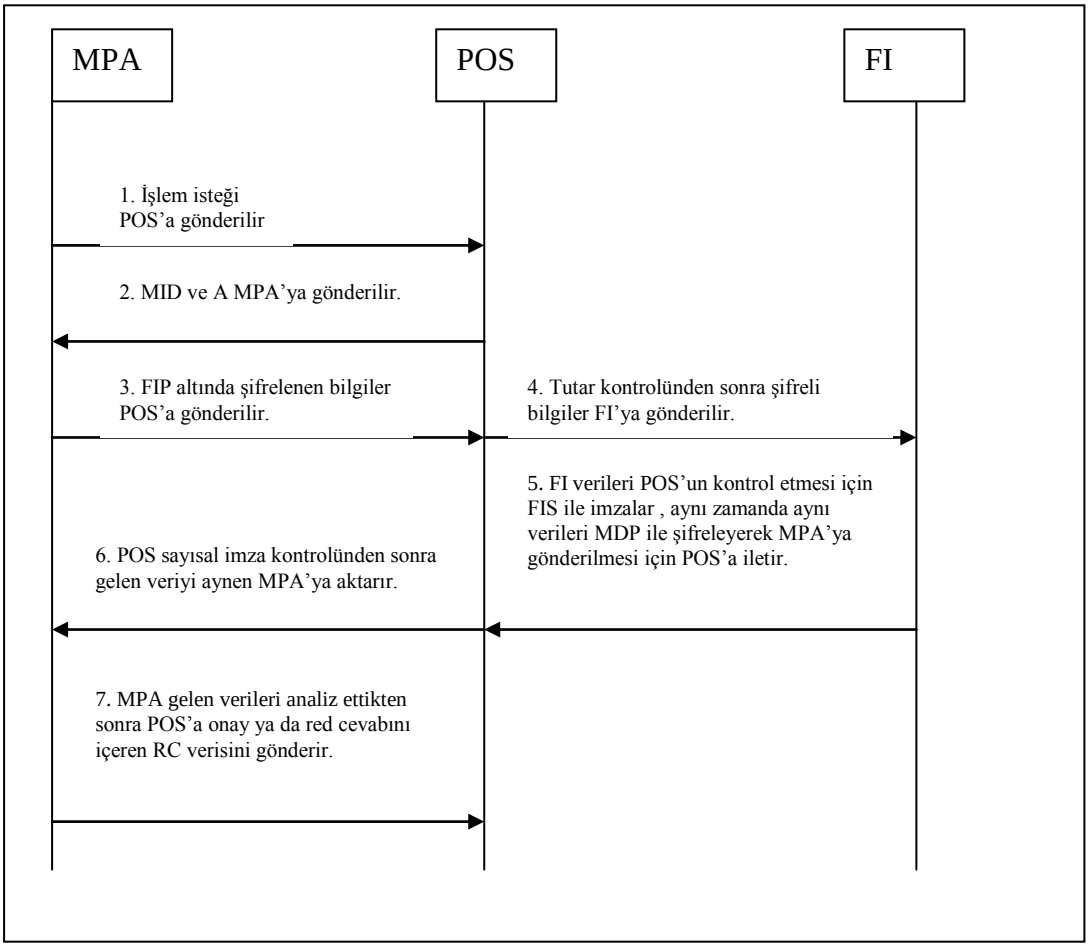
POS cihazı kendi üzerinde tuttuğu FIP ile MID ve A değerlerini şifreler ve MPA'dan gelen değer ile karşılaştırır. Böylece MPA'nın belirlenenden daha az bir tutar bilgisini işyerinden habersiz bir şekilde göndermesi engellenir. Eğer elde edilen değerler farklı ise POS cihazı işlemi sonlandırır, aksi durumda POS_RN bilgisini FIP altında şifreleyerek, MPA'dan gelen şifrelenmiş verilere ekleyerek finansal kuruluşa gönderir.

Finansal kuruluş gönderilen verileri FIS ile deşifre ederek asıl verileri elde eder. PIN doğrulamasını yapar ve limit bilgilerini kontrol ederek işleme onay ya da red döner. Finansal kuruluşun Cevap Kodu (RC) + Provizyon Kodu (PC) + POS_RN + AI değerleri, FIS ile şifrelenerek imzalanmış olur. RC + PC + MPA_RN değerleri de MDP altında şifrelenir ve bu iki değer işyeri POS'una gönderilir.

POS gelen açık RC + PC + POS_RN değerlerinin sayısal imzasını kontrol ederek bu işlemin finansal kuruluştan geldiğini ve farklı bir işlemin cevap kodu olmadığını

doğrular. Böylece gelen cevabın hangi işlemin cevabı olduğu kesinleşmiş olur. Eğer gelen verilerde uyuşmazlık varsa POS hem FI'ya hem de MPA'ya red gönderir ve işlem sonlandırılır. Eğer elde edilen değerler uyumlu ise o zaman MDP altında şifrelenmiş RC + PC + MPA_RN ve açık RC + PC değerlerini MPA'ya gönderir. MPA MDS ile şifreli değerlerin şifresini çözer ve elde ettiği RC + PC değerlerini POS'tan gelen açık RC + PC değerleri ile karşılaştırır. Böylece POS'a farklı MPA'ya farklı cevap gönderilmesi engellenmiş olur. Aynı şekilde MPA kendi ürettiği MPA_RN ile bankadan gelen şifreli MPA_RN değerlerini karşılaştırır ve RC + PC değerlerinin o işlem için geldiğini doğrulamış olur. Eğer bütün kontroller başarılı ise MPA POS'a işlemin başarılı olduğunu belirten bir RC gönderir. Bu şekilde işlem gerçekleştirilmiş olur. Eğer elde edilen değerlerde uyuşmazlık varsa MPA POS'a işlemin reddedildiğine dair bir RC gönderir. Böylece hem MPA tarafında hem de POS tarafında işlem başarısız olur.

FI'nın(banka) işlemi reddetmesi durumunda POS aynı şekilde verileri MPA'ya gönderir ve MPA'nın da işlemin başarısız olduğundan haberdar olmasını sağlar.



Şekil 4.1: İşlem Akışı

4.2 Uygulama Genel Yapısı

Bu uygulamada mobil cihaz olarak Bluetooth özelliği olan bir cep telefonu kullanılmıştır. Uygulama ismi "SelfPay" olarak belirlenmiştir. Cep telefonu üzerinde SymbianOS işletim sistemi çalışmaktadır ve SymbianOS işletim sisteminde C/C++ ile geliştirme yapılmıştır. SymbianOS'un C++ ile yazılmış olması ve C++ dili ile bluetooth gibi devrelere erişim ve devreler üzerinde program geliştirmek JAVA diline göre daha kolay olduğundan bu dil tercih edildi. Cep telefonu üzerine yazılım yüklenmeden kodun test edilebilmesi için Nokia'nın Epoc emülatöründen faydalanıldı. SymbianOS SDK'sı ile derlenen kodlar Visual C++ ile çalıştırılıp Epoc emülatörü kullanılarak gerçek bir cihaz kullanmadan programın test edilmesi sağlanabilmektedir.



Şekil 4.2: Epoc Ekranı

Cep telefonunda 8 MB'lik dahili, 64 MB'lik de harici bellek bulunmaktadır.

POS cihazı olarak Bluetooth özelliği olan bir PC kullanılmıştır ve POS üzerindeki uygulama da C/C++ dili kullanılarak yazılmıştır. POS ekranlarını simüle eden bir kullanıcı arayüzü de Visual Basic ile geliştirilmiştir. Uygulamada hem mobil cihaz ve POS arasındaki mesaj alışverişi ve işlem akışı hem de FI ile POS arasındaki mesajlaşmalar ve kontroller gerçeklenmiştir.

4.2.1 Mobil Cihaz Üzerinde RSA Algoritması Gerçekleme

RSA algoritması için RSA Laboratories'in Mart 1994 tarihli örnek kodları kullanılmış ancak SymbianOS üzerinde çalışabilmesi için yeniden düzenlenmiştir. Aynı kodlar POS üzerinde de kullanılmıştır. Bu kod örneğinde

- RSA şifreleme ve anahtar üretme
- MD2 ve MD5 hash algoritmaları
- CBC modda DES Algoritması
- Diffie-Hellman anahtar değişim algoritması
- Triple-DES algoritması

desteklenmektedir. Uygulamada RSA anahtar üretimi, RSA şifrelemesi ve MD5 algoritmaları kullanılmıştır.

4.2.2 Açık Anahtar Dosya Formatı

Açık anahta dosya yapısı Tablo (4.1)'deki gibidir. Mobil cihaz, finansal kurum ve POS cihazında aynı dosya yapısı kullanılmaktadır.

Tablo 4.1: Açık Anahtar Dosya Yapısı

Alan	**Açıklama**
Bits (unsigned int)	2 byte
Modulus (unsigned char)	128 byte
Exponent (unsigned char)	128 byte

4.2.3 Gizli Anahtar Dosya Formatı

Gizli anahta dosya yapısı Tablo (4.2)'deki gibidir. Mobil cihaz, finansal kurum ve POS cihazında aynı dosya yapısı kullanılamkatdır.

Tablo 4.2: Gizli Anahtar Dosya Yapısı

Alan	**Açıklama**
Bits (unsigned int)	2 byte
Modulus (unsigned char)	128 byte
Açık Anahtar Exponent (unsigned char)	128 byte
Exponent (unsigned char)	128 byte
Prime 1 (unsigned char)	64 byte
Prime 2 (unsigned char)	64 byte
Prime Exponent1 (unsigned char)	64 byte
Prime Exponent2 (unsigned char)	64 byte
Coefficient (unsigned char)	64 byte

4.2.4 Finansal Kurum Açık Anahtarlarının MPA ve POS'a Yüklenmesi

POS'a Issuer Banka Açık Anahtarı (FIP) zaten hali hazırda mevcut olan kablolu ya da kablosuz bağlantısı üzerinden iletilir. Mobil cihaza ise SMS ya da internet üzerinden dosya yükleyerek FIP yüklenebilir. İnternet üzerinden yüklenmiş herhangi bir dosyanın aktive edilmesi için SelfPay uygulamasından Şekil (4.3)'deki gibi "FIP Yükleme" menüsü seçilmelidir.



Şekil 4.3: FIP Yükleme

Menüden seçim yapıldıktan sonra daha önceden internetten yüklenmiş ve formatı belli olan dosya belleğe alınır ve FIP ile şifreleme işlemleri bu şekilde gerçekleştirilir.

4.2.5 Mobil Cihaz Açık Anahtarlarının Finansal Kuruma Gönderilmesi

Finansal kuruma MDP (Mobil Cihaz Açık Anahtarı)'nı göndermek için SMS sistemi kullanılabilir. MD finansal kurumun sağladığı bir SMS sunucuya MDP'sini SMS ile

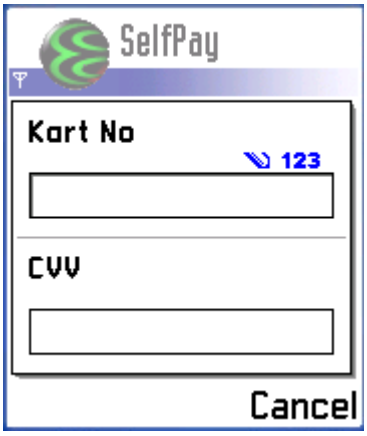
şifrelemeden gönderebilir. Gönderilen MDP'nin formatı Tablo (4.1) ile uyumluluk göstermelidir. "SelfPay" uygulamasında kullanıcı Şekil (4.4)'deki gibi "Anahtar Oluşturma" menüsüne girerek bu işlemi yapabilir. Ancak her anahtar üretiminde, üretilen anahtara belli bir işlem ya da zaman sınırı verilerek bu limit aşıldığında otomatik olarak anahtar üretmek de başka bir çözümdür.



Şekil 4.4: MDP Gönderme

4.2.6 Hesap Tanımlama İşlemi

"Hesap Tanımlama" işlemi Şekil (4.4)'deki "Hesap Tanimlama" menüsü seçilerek yapılmaktadır. Burada hesaptan kastedilen müşterinin bağlı olduğu finansal kurumun alışveriş işlemi sırasında müşterinin şifresi ile eşleştirebileceği bilgidir. Bu bilgi banka kartlarında hesap numarası, kredi kartlarında da kart numarası ve CVV bilgisi olabilir. Bu uygulamada kredi kartı ile ödeme örneği kullanıldığından kart numarası ve CVV bilgileri müşteri tarafından alışveriş işlemlerinden önce bir defaya mahsus olmak üzere girilmelidir. Daha sonra yapılacak tüm işlemlerde bu bilgiler kullanılacaktır. Ancak müşteri isterse aynı menüden hesap bilgilerini değiştirebilir. Kart numarası ve CVV bilgilerinin girildiği ekran Şekil (4.5)'te gösterildiği gibidir.



Şekil 4.5: Kart Numarası ve CVV Tanımlama

4.2.7 Alışveriş İşlemi

Alışveriş işlemi MD'den POS'a alışveriş isteği mesajı gönderilmesi ile başlar. Bu mesaj gönderildikten sonra POS cihazı tutar (A), işyeri no (MID) değerlerini MD'ye gönderir ve MD bu bilgiler ile hesap bilgileri , tarih bilgisi ve MPA_RN bilgilerini FIP ile şifreleyerek tekrar POS'a gönderir. POS gerekli kontrolleri yaparak işlemi FI'ya yönlendirir ve FI'da MDP ile cevabı şifreleyerek tekrar POS'a ve POS üzerinden de MD'ye iletir. Bu işlemler önce MID ve tutar bilgilerini alan "IsyeriID ve Tutar" menüsü ile başlatılmaktadır. Bu menü seçildiğinde önce kullanıcının FI şifresini girmesi istenmektedir Şekil (4.6). Şifre girildikten sonra POS'tan gelen MID ve A bilgileri okunur ve MPA_RN bilgisi rastgele olarak üretilir.

Daha sonra bu bilgiler FIP altında şifrelenir ve POS'a gönderilir. Bu işlemi yapmak için ise "Alisveris" menüsü seçilmelidir. Bu menü seçildiğinde önce MID ve A daha sonra da MPA_RN, tarih, saat , kart no ve CVV bilgileri FIP ile şifrelenerek POS'a gönderilir.

POS'tan gelen cevabı alacak menü ise "Otorizasyon Al" menüsüdür. Bu menü seçildiğinde POS'tan gelen cevap kontrol edilerek işlemin onay alıp almadığı MPA tarafından anlaşılmış olur.

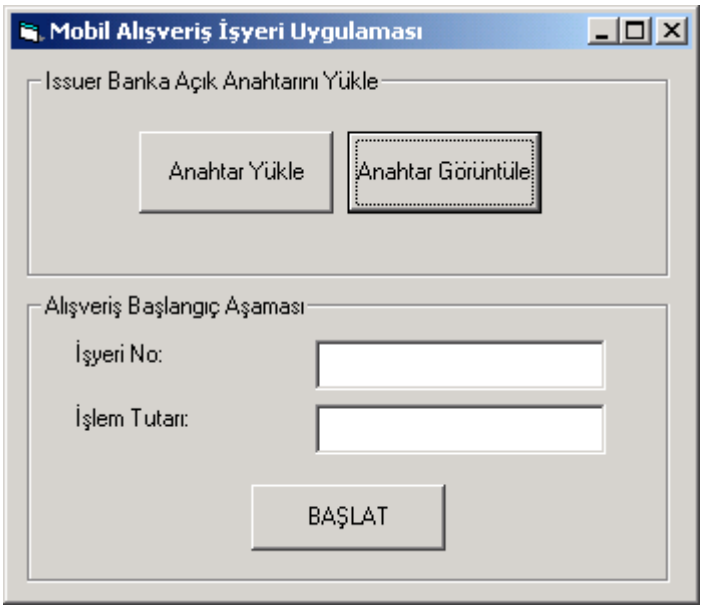
Bu şekilde işlem güvenli tamamlanmış olmaktadır.



Şekil 4.6: Kullanıcı Şifresinin Girilmesi

4.2.8 POS üzerinde Gerçekleşen İşlemler

Çalışmada POS üzerinde de bir uygulama geliştirilmiştir. POS olarak bir PC kullanılmıştır. Bu uygulamada Şekil (4.6) FIP anahtarının okunması ve işlemin başlatılması menüleri vardır.



Şekil 4.6: POS Uygulaması

"Anahtar Yükle" seçeneği seçildiğinde POS FI ile bağlantı kurarak FIP anahtarını alır. Işlem MPA tarafından başlatıldıktan sonra POS'a işyeri numarası ve tutar girilerek POS tarafından da işlem başlatılmış olur.

4.3 Geliştirilen Protokolün Analizi

MPA kendi ürettiği MPA_RN ile FI'dan şifreli şekilde gelmiş MPA_RN değerlerini karşılaştırarak gelen cevabın bağlı olduğu FI tarafından ve o anda gerçekleştirilmekte olan işlem için gönderildiğini doğrular.

POS MPA'dan gelen A ve MID bilgilerini kontrol ederek tutar bilgisinin kendi gönderdiği ile aynı olduğunu doğrular.

FI, PIN ve AI değerleri ile bir doğrulama yapar, ve işlemin gerçekten kendi müşterisi tarafında yapıldığını doğrular.

FI'nın oluşturduğu RC ve PC değerleri POS kontrolü için POS_RN ile birlikte imzalanarak POS'a gönderilir ve POS cevabın FI tararfından gönderildiğini doğrular.

FI MDP ile RC, PC ve MPA_RN değerlerini şifreler ve bunu da POS'a MPA'ya iletilmek üzere gönderir. MPA da bu değerleri kendi üzerinde tuttuğu MDS ile deşifre ederek cevabın o işlem için gönderilmiş olduğunu doğrular.

Bu kontroller sayesinde MPA – POS, POS – FI arasındaki mesajlar dinlense bile , müşterinin banka bilgilerine ulaşılamaz. Aynı zamanda mesajlar taklit edilerek herhangi bir işlem gerçekleştirilemez. Böylece güvenli bir protokolde erişilmeye çalışılan güvenilirlik, veri bütünlüğü, veri gerçekliği ve inkar edilemezlik bu protokolde sağlanmış olur.

Bu çalışmada sertifika kullanılmamış, böylece sertifika yönetimi için fazladan kaynak kullanılmamıştır. Bunun yerine MPA ve FI kendi anahtar çiftlerini kendileri oluştururlar. MDP'nin gerçek FI müşterisi tarafından oluşturulduğu kullanıcı şifresi ile banka tarafından kontrol edilebilmektedir.

İşlem POS ile MPA arasında ve POS ile FI arasında gerçekleşip herhangi bir GSM servis sağlayıcısı ya da WAP sunucusu kullanılmadığından GSM şebekesi olmayan yerlerde de kullanılabilir ve sistemin performansı şebeke performansından bağımsızlaşır. Ayrıca kullanıcı için de maliyetler düşer. Bunun dışında çok az sayıda şifreleme işlemi kullanıldığından diğer protokollere göre çok daha basit bir yapısı vardır

Protokol için kullanılan şifreleme sayıları Tablo (4.3)'te gösterilmektedir.

Tablo 4.3: Protokol Analizi

RSA Şifreleme	RSA Şifre Çözme	Sayısal İmza Kontrolü
3	1	1

4.4 Yapılmış Diğer Çalışmalar

Yapılan çalışmalarda RSA simetrik algoritmalara göre çok daha yavaş çalıştığından asimetrik algoritma kullanılmayan bazı algoritmalar geliştirilmeye çalışılmıştır [16]. Bu çalışmada da discrete logarithm probleminden faydalanılmış ve güvenli bir alışveriş protokolü oluşturulmaya çalışılmıştır. Bu protokole göre alışveriş işlemi şu şekilde gerçekleşir.

Bu yapıda 3 farklı etmen rol almaktadır. Bunlar "C" müşteri, "V" işyeri ve "B" bankadır.

Discrete logarithm problemi: Bu probleme dayanılarak asimetrik algoritmalar oluşturulmuştur. "p" asal bir sayı olmak üzere m ve g sayısı bilinen sayılar ise

$$g^X = m \text{ (mob p)} \quad \textbf{(4.1)}$$

(4.1) formülüne uyan bir x sayısının bulunması zor bir işlem olduğundan 1024 bitlik sayılarla birlikte bu problem bir şifreleme yöntemine dönüşmüştür.

Satış işleminde hash fonksiyonları ve discrete logarithm şifreleme yöntemi kullanılmıştır.

Müşteri 2 tane açık (y1,y2) ve 2 tane de gizli anahtar üretir (x1,x2). Aynı zamanda bankada müşteri hesabına bağlı bir rastgele k sayısı bulunur. Bu k sayısı da Gq uzayından seçilir.

"r"Є Gq olan rastgele bir "r" sayısı seçilir

Müşteriden işyerine giden mesajlar:

α, β ve ω değerleri işyeri uygulamasına gönderilmektedir. Bu değerler de (4.2), (4.3), (4.4), (4.5), (4.6) formüllerine göre hesaplanmaktadır

$$H(ID) = H(\ IDc \ || \ IDv \ || \ IDb) \quad \textbf{(4.2)}$$

$$Khi = H(k \ || \ Khi\text{-}1) \quad \textbf{(4.3)}$$

$$\alpha = X_1 + r \quad \textbf{(4.4)}$$

$$\beta = k + Khi \quad \textbf{(4.5)}$$

$$\omega = r^{-1} [\sigma X1 + X2 + Khi + H(ID)] \quad \textbf{(4.6)}$$

İşyeri de y1, y2 ve yk açık anahtarlarını alır ve gelen mesajdaki bilgileri doğrular.

x1 gizli, y1 açık anahtar olmak üzere x1 ve y1 (4.7) formülüne göre hesaplanır.

$y1= g^{x1}$ **(4.7)**

Khi ise hash zinciridir. İlk adımda "k" değeri hash fonksiyonuna sokulur ve oluşan sonuç ile birleştirilerek oluşan sonuç tekrar hash fonksiyonuna sokulur. Bu şekilde her işlemde bir önceki hash fonksiyon sonucu ve "k" değerinin birleştirilmiş hali tekrar hash fonksiyonundan geçirilmiş olduğundan oluşan sonuçlar hep farklı çıkacaktır.

Bu protokolde α, β ve ω ile IDc ve IDb değerleri işyerine gönderilir. Ancak bizim sistemimize uygulanması, bluetooth kullanıldığından çok güvenli olmamaktadır. Hattı dinleyen birisi bir kaç işlemden sonra kolaylıkla kullanıcının gizli anahtarlarını ele geçirip , mesaj üzerinde oynama yapabilir.

Bir diğer protokode de (Kungpisdan Protokolü) MAC değerleri ve hash fonksiyonları kullanılarak güvenlik sağlanmaya çalışılmıştır[14].

Bu protokole göre:

C: Müşteri

M : İşyeri

PG : Ödeme Sistemi

I: Issuer

A: Acquirer

TID: İşlem ID'si (bu veride işlem tarihi de bulunur)

OD : İşlem açıklaması

OI: İşlem Bilgisi = {TID, h(OD, Price)}

Price: Tutar ve kur bilgisi bulunur.

Yes/No: işlemin kabul ya da reddini gösterir.

TIDReq: İşlem isteği ID'si

$\mathbf{X_i}$, $1< i < n$ için **müşteri** ve **işyeri** arasında paylaşılmış bir anahtardır.

$\mathbf{Y_i}$, $1< i < n$ için **issuer** ve **müşteri** arasında paylaşılmış bir anahtardır.

$\mathbf{Z_i}$, $1< i < n$ için **ödeme sistemi** ve **işyeri** arasında paylaşılmış bir anahtardır.

Bu protokolde müşteri, işyeri ve ödeme sistemi arasındaki mesajlar aşağıda anlatılmaktadır.

Müşteri → işyeri: (Müşteriden işyerine giden mesaj olmak üzere)
IDc ,
i ,
TIDReq,
MIDReq

İşyeri → Müşteri
$\{TID, ID_M\}_{X_i}$
Müşteri → İşyeri:
$\{OI, Price, IDc, ID_I, MAC[(Price, h(OI), IDm), Yi]\}_{X_i,}$
$MAC[(OI, Price, IDc, ID_I), X_{i+1}]$

İşyeri → Ödeme sistemi
$\{MAC[(Price, h(OI), ID_M,), Yi], h(OI), i, TID, Price, IDc, ID_I\}_{Z_j}$,
j ,
IDm,
$MAC[(h(OI), i, TID, IDc, ID_I), Z_{j+1}]$

Ödeme sistemi → Issuer
MAC[(Price, h(OI), IDm) ,Yi],
h(OI),
i,
TID,
Price,
IDc,
IDm,
$h(Z_{j+1})$

Ödeme sistemi →Acquirer
Price, IDm

Issuer/acquirer → ödeme sistemi:
Yes/No, $\{h(OI), Yes/No, h(Z_{j+1})\}_{Y_i,}$ h(Yes/No, h(OI), h(Yi))

Ödeme sistemi → İşyeri:
$\{Yes/No, \{h(OI), Yesy/No, h(Z_{j+1})\}_{Yi}, h(Yes/No, h(OI), h(Yi))\}_{Z_{j+1}}$

İşyeri →Müşteri
$\{\{h(OI), Yes/No, h(Zj+1)\}_{Yi}\}_{Xi+1}$

Protokol için kullanılan şifreleme sayıları Tablo (4.4)'te gösterilmektedir.

Tablo 4.4: Kungpisdan Protokolü Analizi

MAC	Simetrik şifreleme	Hash fonksiyonu
5	8	13

Bu protokolde daha önce paylaşılmış simetrik anahtarlar eğer kötü niyetli kişilerce ele geçirilirse bütün sistemi kontrol altına alabilirler. Aynı zamanda ödeme sisteminden issuer'a giden mesajda müşteri ID'si IDc açık gitmektedir. Mesajı

dinleyen birisi mesajın IDc kısmını değiştirerek farklı bir müşteriden ödeme yapılmasını sağlayabilir, IDm ve Price bilgileri hash sonuçları ile gönderildiklerinden bunlar değiştirilemez. Ancak hattı dinleyen kişi hangi hash fonksiyonunun kullanıldığını biliyorsa o zaman değiştirdiği değerin hash değerini de oluşturarak mesajı değiştirebilir ve issuer'un yanılmasını sağlayabilir.

5. SONUÇLAR

Bu çalışmada geliştirilen protokol, mobil cihazlar kullanılarak yapılacak alışveriş işlemlerinde güvenlikli ve uygulaması kolay bir ortam sağlamaktadır. Her işlemde sayısal imza ve RSA şifrelemesi kullandığından, simetrik şifreleme kullanan diğer protokollere göre çok daha güvenlidir. Geliştirilen örnek uygulamada da mobil cihazların artık asimetrik şifreleme işlemlerini yeteri kadar hızlı yaptığı görülmüş, işlem oldukça kısa bir zaman almıştır. Bu çalışma günlük hayatta özellikle restoranlarda ve benzin istasyonlarında oturduğumuz yerden ödeme yapmamızı sağlayacağından büyük bir kolaylık sağlayacaktır, ayrıca kredi kartlarının kopyalanması gibi güvenlik açıklarını ve bankaların kart basım masraflarını ortadan kaldıracaktır. İşlemlerin hepsinde kullanıcı şifresi kontrolü yapıldığından mobil cihaz çalındığında da bu bilgilerin kullanılabilmesi mümkün olmayacaktır.

KAYNAKLAR

[1] **Hu, W., Lee, C., Kou, W.**, 2005. Advances In Security And Payment Methods For Mobile Commerce, Idea Group, Inc.

[2] **Peikari, C., Fogie, S.**, 2002. Maximum Wireless Security, Sams Publishing

[3] **http://www.avea.com.tr**

[4] **hhtp://www.emvco.com**

[5] **Hall, J., Kilbank, S., Barbeu, M., Kranakis, E.**, WPP: A Secure Payment Protocol for Supporting Credit and Debit Transactions Over Wireless Networks, 2001. *IEEE International Conference on Telecommunications,* Bükreş, Romanya, Haziran 4-7.

[6] **Swaminata, T. M., Elden, C. R.,** 2002. Wireless Security And Privacy: Best Practices and Design Techniques, Addison Wesley, Boston.

[7] **Rhee, M. Y.**, 2003., Internet Security: Cryptographic Principles, Algorithms and Protocols, John Wiley & Sons, Chichester.

[8] **Mitchell ,C. J.,**2003., Security for Mobility , John Wiley & Sons, Chichester.

[9] **Mastercard & VISA Publications,** 1997., SET Secure Electronic Transaction Specification Book 1: Business Description Version 1.0 .

[10] **VISA Publications,** 2002., 3D Secure Introduction:Version 1.0.2 .

[11] **Jipping, M. J.**,2003., Symbian OS Communications Programming, John Wiley & Sons, Chichester.

[12] **Edwards, L. Barker, R.,** ,2004., Developing Series 60 Applications , Addison Wesley, Boston.

[13] **Bao, F. Anantharaman, L. Deng, R.,** Design of Portable Mobile Devices Based E-Payment System and E-Ticketing System With Digital Signature , 2001. *Info-tech and Info-net, 2001. Proceedings. ICII 2001* ,Pekin , Çin.

[14] **Kungpistan, S. Srinivasan, B. Le, P. D.,** 2004. Accountability Logic for Mobile Payment Systems, *Coding and Computing (ITCC'04) Volume 1* , 40.

[15] **Capkun, S. Buttyán, L. Hubaux, J.,** 2002. Self-Organized Public Key Management for Mobile Ad Hoc Networks, *ACM International Workshop on Wireless Security*

[16] **Ham, W. Hyungki, C. Xie, Y. Lee, M. Kim, K.,** 2002. Secure One-way Mobile Payment System Keeping Low Computation in Mobile Devices , *Proc. of WISA2002, 287-301*

ÖZGEÇMİŞ

Bu çalışmayı yapan Hayriye Güpse GÜNEŞ, 1980 yılında Tokat'ta doğmuştur. Ortaokul ve lise öğrenimini Turhal Anadolu Lisesi'de tamamladıktan sonra, 1998 yılında İstanbul Teknik Üniversitesi Bilgisayar Mühendisliği bölümünde lisans eğitimine başlamıştır. 2002 yılında lisans eğitimini tamamladıktan sonra aynı yıl içinde İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bölümünde yüksek lisans eğitimine başlamıştır. 2002 yılında Banksoft Bankacılık Yazılım Sistemleri Ltd. Şti'de ATM ve Debit Kart yönetim sistemleri üzerinde uygulamalar geliştirmek üzere göreve başlayıp halen burada görevini sürdürmektedir.