GÜVENLİ BİR UZAKTAN KONUM TAKİP SİSTEMİ UYGULAMASI

A SECURE REMOTE POSITION TRACKING SYSTEM APPLICATION

YUSUF İSTEMİ BEŞEL

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

ELEKTRİK ve ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü

YÜKSEK LİSANS TEZİ

olarak hazırlanmıştır.

2010

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **ELEKTRİK ve ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan	:….. Prof. Dr. Abdullah ÇAVUŞOĞLU
Üye (Danışman)	:….. Yrd. Doç. Dr. Mehmet DEMİRER
Üye	:….. Doç. Dr. Ali Ziya ALKAR
Üye	:….. Yrd. Doç. Dr. Derya ALTUNAY
Üye	:….. Yrd. Doç. Dr. Umut SEZEN

ONAY

Bu tez ...../...../2010 tarihinde Enstitü Yönetim Kurulunca kabul edilmiştir.

Prof. Dr. Adil DENİZLİ
Fen Bilimleri Enstitüsü Müdürü

Oğlum Demir’e,

GÜVENLİ BİR UZAKTAN KONUM TAKİP SİSTEMİ UYGULAMASI

Yusuf İstemi Beşel

ÖZ

Bu tez kapsamında bir uzaktan konum takip sistemi uygulaması gerçekleştirilmiştir. Araca yerleştirilen bir PDA cihazında çalışmakta olan yazılım, PDA ile bağlantılı GPS alıcısından alınan konum bilgilerini, GPRS bağlantısı üzerinden merkez üniteye aktarmakta ve merkez'de konum takibi yapılmaktadır.

Konum koordinat bilgileri, sunucuya GPRS üzerinden şifrelenerek gönderilmekte ve PDA cihazın merkez ünite ile bağlantısı sırasında verinin çalınması, ya da merkez ünitede sunucunun sahte veriler ile aldatılması engellenmiştir. Veri şifrelemek için AES Rijndael algoritması kullanılmış ve her oturum için değişik şifreleme anahtarı kullanılabilmesi amacıyla Diffie-Hellman anahtar değişimi kullanılmıştır.

Yazılım dili olarak C# ve yazılım geliştirme ortamı olarak Visual Studio .NET kullanılmış, Sunucu tarafında çok sayıda PDA cihazı ile iletişim yapılabilmesi için IIIS 6.0 arkasında çalışan bir yazılım geliştirilmiştir. Ağ duvarı arkasında intranet'te çalışacak harita izleme yazılımında, harita olarak Google'dan indirilen Google maps haritaları kullanılmış ve geliştirilen yazılım ile PDA cihazlardan gelen GPS konum bilgilerinin izlenmesi gerçekleştirilmiştir.

Anahtar Kelimeler: GPS, Konum takip sistemi, GPRS, Visual Studio .NET, C#.

Danışman: Yrd. Doç. Dr. Mehmet DEMİRER, Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü

A SECURE REMOTE POSITION TRACKING SYSTEM APPLICATION

Yusuf İstemi Beşel

ABSTRACT

In this thesis, a GPS based remote location tracking system application has been designed. A client application running on Windows mobile device with GPS receiver and GPRS connection is used on vehicle. This application sends GPS position coordinates over GPRS connection to a server, and position tracking is done on server side.

Encrypted GPS position coordinates are sent to server. During the communication on PDA client application with server application, eavesdropping of sent data and fake data which could be send to server by an intruder are prevented. For data encryption, AES Rijndael algorithm has been used in order to supply different symmetric key for each session of each PDA connection Diffie- Hellman key Exchange is used.

C# and Visual Studio .NET are used; a server application running behind IIS 6.0 is used. A map tracking application is used behind Firewall on intranet. Google maps images are used to track PDA devices' location.

Keywords: GPS, Location Tracking System, GPRS, Visual Studio .NET, C#.

Advisor: Asst. Prof. Mehmet DEMİRER, Hacettepe University, Department of Electrical and Electronics Engineering

TEŞEKKÜR

Yazar, bu çalışmanın gerçekleşmesinde katkılarından dolayı, aşağıda adı geçen kişi ve kuruluşlara içtenlikle teşekkür eder.

Sayın Yrd. Doç. Dr. Mehmet DEMİRER, tez çalışmasının gerçekleştirilmesi için gerekli ortamı hazırlamıştır ve karşılaşılan güçlüklerin aşılmasında yol gösterici olmuştur.

Yazar, ailesi ve arkadaşlarına, bu çalışmanın gerçekleşmesi esnasında gösterdikleri maddi, manevi destekler için müteşekkirdir.

İÇİNDEKİLER DİZİNİ

Sayfa

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

Sayfa

SİMGELER ve KISALTMALAR DİZİNİ

λ	Dalga boyu
AES	Geliştirilmiş Şifreleme Standardı (Advanced Encryption Standard)
API	Uygulama Programı Arabirimi (Application Programming Interface)
CBC	Kapalı Metin Zincirleme (Counter Block Chaining)
D-H	Diffie- Hellman
GPRS	Genel Paket Radyo Hizmetleri (General Packet Radio Services)
GPS	Genel Paket Radyo Hizmetleri (General Packet Radio Services)
GSM	Global Special Mobile
GUI	Grafiksel Kullanıcı Arabirimi (Graphical User Interface)
HMAC	Özet Mesajı Doğrulama Kodu (Hash Message Authentication Code)
HTTP	Hipermetin Aktarma İletişim Kuralı (Hypertext Transfer Protocol)
ETSI	Avrupa Telekomünikasyon Standartlar Komitesi (European Telecommunications Standards Institute)
IP	İnternet Protokolü (Internet Protocol)
IDS	İzinsiz Giriş Tespit Sistemi (Intrusion Detection System)
IIS	Internet Information Services (Internet Bilgi Hizmetleri)
IPS	İzinsiz Giriş Engelleme sistemi (Intrusion Prevention System)
IV	Başlangıç Vektörü (Initialization Vector)
LAN	Yerel Alan Ağı (Local Area Network)
MAC	Mesaj Doğrulama Kodu (Message Authentication Code)
MD5	Mesaj Özeti 5 (Message Digest 5)
NIST	Ulusal Standartlar ve Teknoloji Enstitüsü (National Institute of Standards and Technology)
PDA	Kişisel Sayısal Asistanı (Personal Digital Assistant)
PNG	Taşınabilir Ağ Trafiği (Portable Network Graphics)
PFS	Mükemmel İletme Gizliliği (Perfect Forward Secrecy)

PPS	Duyarlı Konum Belirleme Sistemi (Precise Positioning Service)
PRN	Sözde rassal gürültü (Pseudo Random Noise)
SHA	Güvenli Özetleme Algoritması (Secure Hash Algorithm)
SPS	Standart Konum Belirleme Sistemi (Standard Positioning Service)
SQL	Yapılandırılmış Sorgu Dili (Structured Query Language)
TCP	İletim Kontrol Protokolü (Transmission Control Protocol)
VPN	Sanal Özel Ağ (Virtual Private Network)
XML	Genişletilebilir İşaretleme Dili (Extensible Markup Language)
XOR	Dışlamalı Veya (Exclusive OR)

EKLER DİZİNİ

Sayfa

1. GİRİŞ

Küresel konumlama sistemi (GPS), dünya çapında kullanıma açık, her türlü hava şartlarında, gündüz ve gece 24 saat kesintisiz ve güvenilir konumlama ve zaman bilgisi hizmeti veren bir navigasyon sistemidir.

Amerika Birleşik Devletleri Savunma Bakanlığı tarafından askeri uygulamalar için geliştirilmiş olan GPS Sistemi, 1983 yılından itibaren sivil kullanıma da hizmet verir hale gelmiştir [1].

GPS alıcısı ile elde edilen konum ve zaman bilgisi, mobil iletişim sistemleri ile uzaktaki bir merkezi üniteye uzak izleme amacı ile iletilebilmektedir. Kamu kurumları, bankalar, özel sektörde filo takibi için coğrafi konum takip sistemleri kullanmaktadırlar. GPS alıcısının elde ettiği konum ve zaman bilgisi, merkezi üniteye SMS yöntemi ile ya da GRPS bağlantısı kurularak internet üzerinden ilave bir şifreleme mekanizması kullanılmadan şifresiz olarak iletilmekte ve bu durum konum bilgilerini atakla ele geçirebilecek saldırganın kötü amaçlı kullanımına yol açabilmektedir. (Örneğin bir banka para nakil aracının yol gidiş güzergahının öğrenilmesi)

Bu tez kapsamında PDA cihazlarda çalışacak bir istemci yazılımı, GPS alıcısından edindiği konum bilgilerini şifreleyerek, merkez ünite'de bulunan sunucuya GPRS bağlantısı ile iletmekte, sunucuda şifre çözülme işlemi yapılarak her alıcıdan alınan bilgiler veritabanı sunucusunda işlenmekte ve bir harita izleme yazılımı ile her cihaz son anlık konum bilgileri, geçmiş konum bilgileri, cihaz pasif ya da cihaz çalıntı bilgileri gözlenebilmektedir.

Bu tez kapsamında PDA cihazlarda çalışacak istemci yazılımı ve merkez ünite'de bulunan, IIS arkasında çalışacak sunucu yazılımı için, güvenli haberleşmeyi gerçekleştirebilmek amacıyla bir tasarım gerçekleştirilmiş, olası ağ saldırılarına karşı gerekli önlemler alınarak iletişimin devamlılığının korunması sağlanmıştır.

Küresel konumlama sistemi (GPS) ile ilgili temel bilgiler 2. bölümde sunulmuştur. 3. bölümde GPRS yapısına değinilmiştir. 4. bölümde uzaktan konum takip sistemi yazılımı özellikleri ve tasarım bilgileri sunulmuştur. 5. bölümde bu tez kapsamı ile ilgili sonuç ve öneriler yer almaktadır.

2. KÜRESEL KONUMLAMA SİSTEMİ (GPS)

2.1. GPS'in Tarihi

Bir nesnenin yerini, yönünü belirlemek ve yönlendirmek için insanlar yön güdüm sistemlerine ihtiyaç duymuşlardır. Tarih boyunca yönlerini belirlemek için başta gökyüzündeki yıldızları kullanmak üzere birçok yöntem kullanmışlardır. Uzay tabanlı yön güdüm sistemleri dünya yörüngesinde bulunan uydulardan gelen sinyalleri kullanarak kullanıcılara konum ve hız bilgisi sağlarlar. İlk uydu yön güdüm sistemleri ABD tarafından geliştirilen Transit ve SSCB tarafından geliştirilen Tsikada sistemleridir [2].

İlk örnek Transit uydusu 1961 yılında yörüngeye yerleştirilmiştir. Transit sistemi yeryüzünden yaklaşık 1100 km uzaklıkta konuşlandırılmış 6 uydudan oluşmakta idi. Sistem 1967 yılında sivil kullanıma açılmıştır. [9]. Sistemin yalnızca 6 uydudan oluşması, genellikle düzeltme bilgilerinin alınabileceği tek uydunun görünürde olmasına yol açıyordu. Sistemin amacının yüksek hızda, gerçek zamanlı konum bilgisi belirleme olmasına karşın, son geliştirmeleri ile sistem yaklaşık 200 metre hassasiyet ile konum bilgisi sağlayabilmekte idi.

1970'li yılların başlarında ABD Savunma Bakanlığı, konum bilgisi elde edebilmek için yeni bir uydu navigasyon sistemi geliştirmeye başlamıştır [2]. Eski nesil yön güdüm sistemlerinde hareketli kullanıcılar için başarım oranı istenilen seviyede değildi. Uydulardan düzeltme bilgileri kullanıcıya gelse de, bilgilerin işlenmesi süresinin uzun olması, yüksek hızlı hareket eden kullanıcılar için sorun oluşturmakta idi. 1989 ve 1994 yılları arasında 24 adet GPS uydusu başarı ile uzaya gönderilmiş olmuştur. GPS sistemi 1995 yılında bütün fonksiyonlarıyla çalışır duruma gelmiştir.

GPS Sistemi, halen aktif olarak çalışmakta olup, tüm dünyada kullanıcılara hizmet vermeyi sürdürmektedir.

Çizelge 2.1: Uydu Özeti

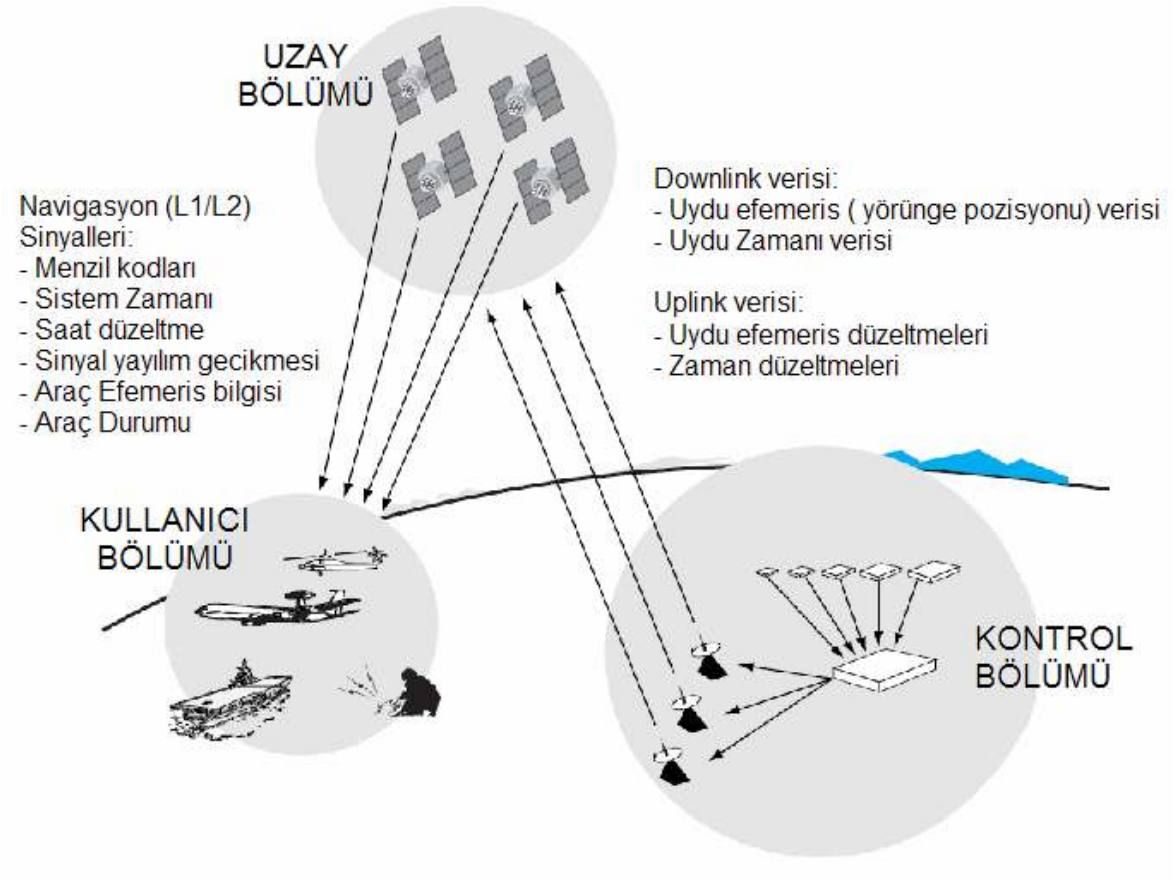
Blok	Fırlatılma Periyodu	Fırlatılan Uydular				Halen yörüngede bulunan ve hizmet veren
		Başarılı	Başarısız	Hazırlanıyor	Planlama aşamasında	
I	1978–1985	10	1	0	0	0
II	1989–1990	9	0	0	0	0
IIA	1990–1997	19	0	0	0	Fırlatılanların 11 adeti
IIR	1997–2004	12	1	0	0	Fırlatılanların 12 adeti
IIR-M	2005–2009	8	0	0	0	Fırlatılanların 7 adeti
IIF	2010–2011	0	0	10	0	0
IIIA	2014–?	0	0	0	12	0
IIIB		0	0	0	8	0
IIIC		0	0	0	16	0
TOPLAM		58	2	10	36	30

Son güncelleme 24.11.2009

Çizelge 2.1'de görüldüğü üzere farklı yıllar arasında bloklar halinde uyduların fırlatılması gerçekleştirilmiş ve bu fırlatmaların büyük çoğunluğu başarıya ulaşmıştır [3]. Geçen zaman içinde kullanım ömürleri dolan uyduların yerini alacak yeni uyduların fırlatılması planlanmaktadır.

2.2. Genel Bilgiler

GPS sistemi, Şekil 2.1'de gösterildiği üzere uzay bölümü, kontrol bölümü ve kullanıcı bölümünden oluşmaktadır.



Şekil 2.1. GPS Sistemi

GPS sisteminde konum belirleme ve navigasyon hizmetleri PPS (Duyarlı Konum Belirleme Hizmeti) ve SPS (Standart Konum Belirleme Hizmeti) olarak iki farklı seviyede verilmektedir [4].

PPS, yüksek doğruluklu konum, hız ve zaman belirleme hizmeti olup yalnızca Amerikan Savunma Bakanlığı tarafından yetkilendirilmiş askeri kullanıcılara açıktır. SPS ise PPS'ye göre daha düşük doğruluklu konum, hız ve zaman belirleme hizmeti olup sivil, asker tüm kullanıcılara açıktır [5].

GPS ölçmelerinde, elektromanyetik dalgalar kullanılarak uydulardan kullanıcılara veri akışı sağlanmaktadır. Her GPS uydusu konum belirleme amaçlı olarak L1(Link1) ve L2(Link2) olmak üzere iki temel frekansta radyo sinyalleri yayınlamaktadırlar. . L1 ve L2 frekansları 10.23 MHz olan temel frekansın 154 ve 120 tam katları alınarak elde edilmiş olup L1 frekansı 1575.42 Mhz ve L2 frekansı

1227.60 Mhz'dir [12]. GPS sisteminin tasarımı aşamasında birçok taşıyıcı frekans incelenmiştir. İnceleme sonucunda, frekans tahsisindeki kolaylıklar ve iyonosferik etkilerin diğer bantlara göre çok daha küçük olması nedeniyle L-bandı kullanımı tercih edilmiştir.

GPS sisteminde çift frekans kullanılmasının amaçları; L1 frekansının herhangi bir nedenle kesilmesi ya da elektronik karıştırmaya maruz kalması durumunda L2 frekansının yedek frekans görevi görmesi ve çift frekans özelliğinden yararlanarak iyonosferik düzeltme olanağı sağlaması olarak sıralanabilir [6].

L1 ve L2 taşıyıcı frekansları iki tip kod (Pseudo Random Noise- PRN) ve navigasyon mesajı verileri ile kiplenmiştir. L1 taşıyıcı frekansı üzerine C/A(Coarse/Acquisition; Clear/Access) kod ve P (Precise/Protected Code) kod ile navigasyon mesajı verileri kiplenmiştir. L2 taşıyıcı frekansı ise yalnızca P kod ve navigasyon mesajı verileri ile kiplenmiştir. P kodun sadece askeri kullanıcılara açık olması nedeniyle sivil kullanıcıların tek frekans (L1-C/A kod) kullanabilmekte ve bu durumda iyonosferik düzeltme olanağı sağlayan çift frekans özelliğinden yararlanamamaktadırlar. Sivil kullanıcıların da çift frekans üstünlüklerinden yararlanabilmeleri amacıyla 2003 yılından itibaren Block IIR-M uyduları aracılığıyla L2 frekansı üzerinden C/A kod yayınlanmasına karar verilmiştir. Ayrıca üçüncü ve yeni bir sivil frekans tahsisi söz konusudur. L5 (Link5) adı verilen bu sinyalin frekansı 1176.45 MHz'dir. Bu sinyalin 2012 yılına kadar 18 uydudan oluşması planlanmıştır [12]. Çizelge 2.2'de GPS uydu sinyali bileşenleri görülmektedir [5].

Çizelge 2.2. GPS Uydu Sinyali Bileşenleri

Uydu sinyali bileşeni	Frekansı (MHz)	Dalga Boyu (λ)
Temel Frekans	$f_0 = 10.23$	---
L1 Taşıyıcı	$f_0*154 = 1575.42$	~19.0 cm
L2 Taşıyıcı	$f_0*120 = 1527.60$	~24.4 cm
P-kod	$f_0 = 10.23$	29.3 cm
C/A Kod	$f_0/10 = 1.023$	293 m
Navigasyon Mesajı	$f_0/204600 = 50.10^{-6}$	---

Çizelge 2.2'de görüldüğü gibi C/A kod 1 MHz'lik bir kod olup milisaniyede bir tekrar etmektedir. C/A kod periyodunun çok kısa seçilmesinin amacı GPS alıcılarının

uydulara en kısa sürede kilitlenmesini sağlamaktır. C/A kod tüm kullanıcılara açıktır ve özellikle SPS için temel oluşturmaktadır [6].

P Kod, L1 ve L2 taşıyıcılarının her ikisinde de kiplenmiştir. Uzun periyotlu bir koddur. Elektronik karıştırmaya ve aldatmaya karşı korunmak için bu kod AS (Anti-Spoofing) özelliği kullanılarak kriptolanmıştır. W-kod olarak bilinen kripto kodu sayesinde askeri amaçlı GPS alıcılarının doğrudan çözebileceği bir kod ortaya çıkmıştır [7].

2.2.1. Uzay Bölümü

Uzay Bölümü, Orta Yörünge'de (Medium Earth) dünya yüzeyinin 20,183 km üzerindeki yörüngede bulunan 24 adet uydudan oluşmaktadır. Uydular oldukça geniş bir görüş alanına sahiptirler ve dünya üzerindeki bir GPS alıcısının her zaman en az 4 adet uyduyu görebileceği şekilde yerleştirilmişlerdir. GPS alıcısının düzgün konum hesaplayabilmesi için aynı anda 4 adet uydudan sinyal alabilmesi gerekmektedir.

24 GPS uydusu 6 farklı yörünge düzlemi üzerine, her düzlemde 4 uydu yer alacak şekilde yerleştirilmiştir. Her düzlem ekvator boyunca eşit konumlandırılmıştır ve 55 derece eğimlidir. Uydular saatte 7.000 mil hızla hareket ederler ve yaklaşık 12 saatte (11 saat 58 dk), dünya çevresinde bir tur atarlar. Uydular güneş enerjisi ile çalışırlar ve güneş enerjisi kesintilerine karşı (güneş tutulması vs.) yedek bataryaları ve yörünge düzeltmeleri için de küçük ateşleyici roketleri bulunmaktadır.

2.2.2. Kontrol Bölümü

Kontrol bölümü,

- Ana kontrol istasyonu
- Yer antenleri
- İzleme istasyonları

kısımlarından oluşmaktadır. Şekil 2.2'de konumları belirtilmiş olan, dünya üzerinde uygun olarak konuşlandırılmış sabit izleme istasyonlarından GPS uyduların takibi yapılmaktadır.

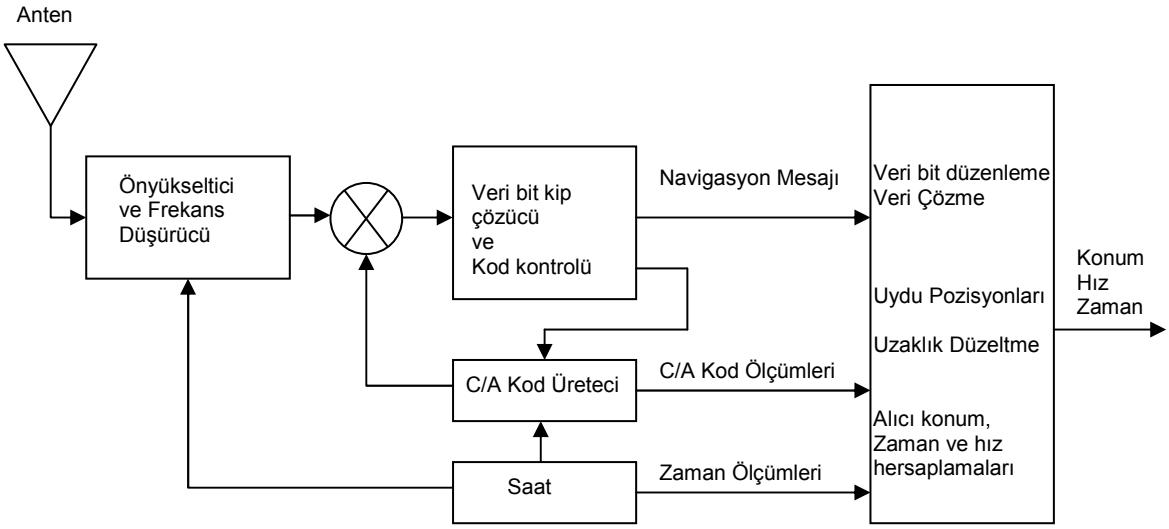
Şekil 2.2. GPS Ana kontrol İstasyonu ve İzleme İstasyonları

Ana kontrol istasyonu, izleme istasyonlarından da alınan bilgileri değerlendirerek, GPS uydularının doğru yörünge ve zaman bilgilerini sağlar. Uydu manevralarının kontrolü, yedek uydu ekipmanlarının konfigürasyonları, uydu seyir mesajlarının güncelleştirilmesinin yönetilmesi gibi fonksiyonları üstlenmektedir.

İzleme istasyonları, görüş alanlarına giren tüm uyduları izleyerek, uydu verilerini toplayıp ana kontrol merkezine iletmektedirler. Ana kontrol istasyonu, hesapladığı yörünge zaman bilgilerini yer anten istasyonları vasıtası ile uydulara iletişim kanalı (S bandı 1783.74 MHz ve 2227.5MHz) üzerinden yükler. GPS uydularına bu verilerin yüklenme sıklığı değişmekle birlikte günde 1-3 kez olabilmektedir.

2.2.3. Kullanıcı Bölümü

Kullanıcı bölümü GPS uydularından yayınlanan sinyalleri alıp çözümlemek üzere özel olarak tasarlanmış alıcılardan oluşur. Alacıların tasarımı, uygulama alanı ve kullanım amaçlarına göre farklılıklar gösterir. Genel GPS alıcı yapısı Şekil 2.3'deki gibidir.



Şekil 2.3. Temel GPS alıcısı blok şeması

GPS alıcısından alıcının konumu enlem, boylam, yükseklik bilgisi olarak elde edilir. Alıcı zamanı ve hızı, alıcının görüntülediği uydu sayısı, hassasiyet bilgisi gibi bilgiler de elde edilebilmektedir. GPS alıcıları tek yönlü çalışırlar, sinyal aldıkları uydulara geri bilgi iletmezler

GPS alıcıları sivil ve askeri uygulamalar için havacılık, denizcilik, kara araçları gibi birçok alanda kullanılmaktadır.

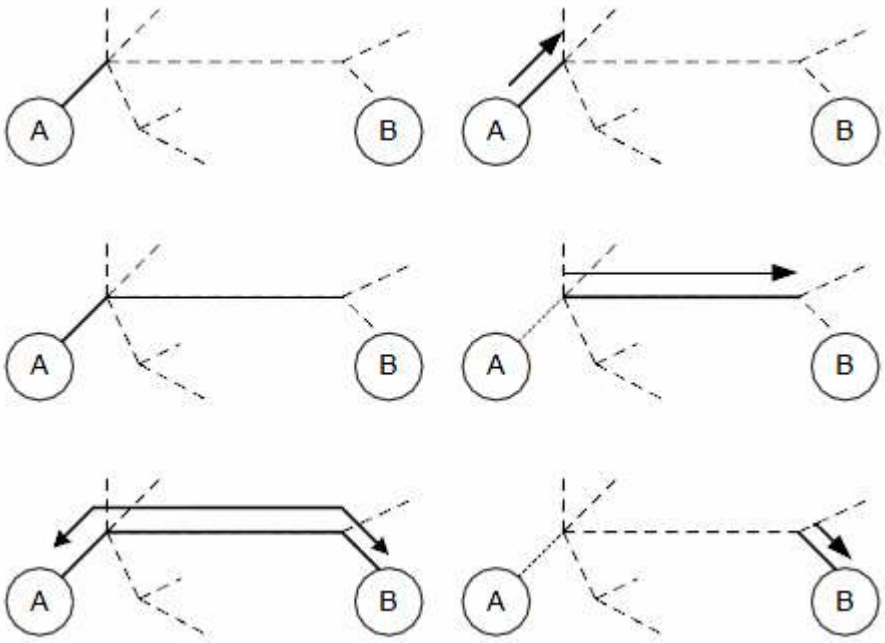
Yüksek doğruluk hassasiyeti gerektiren GPS uygulamalarında hata oranını azaltmak için Diferansiyel GPS gibi sistemlerden de faydalanılmaktadır. Bu teknikte sabit bir noktaya konuşlandırılmış bir GPS alıcısı kullanılır. Diğer uzak kullanıcılar, sabit alıcının gözlem yaptığı uydulara ait uydu ile alıcı arasındaki uzaklıkları, gerçek uzaklıklarla karşılaştırıp oluşturduğu düzeltme bilgilerini telsiz, yer istasyonu ya da uydular aracılığı ile elde ederler ve bu düzeltme bilgilerini kullanarak daha doğru konum bilgisi hesaplayabilirler.

3. GPRS

GPRS, verilerin mevcut GSM şebekeleri üzerinden 28.8 kbps'den 115 kbps'ye kadar varabilen hızlarda iletilmesine imkan veren, cep telefonu, diz üstü bilgisayar, PDA ve diğer mobil cihaz kullanıcılarına kesintisiz İnternet bağlantısı sunan, paket radyo prensibine dayalı mobil iletişim servisidir [8] .

GPRS, ETSI tarafından 1995 yılında oluşturulmaya başlanmış ve 1998 yılında GSM spesifikasyonu Release 97 (sürüm 97)'ye dahil edilmiştir [8] [9].

GPRS veri iletişiminde geleneksel devre anahtarlamalı veri iletişimindeki gibi hatta bağlı olunan süre başına ücretlenlendirmenin aksine transfer edilen veri paket büyüklüğü (kilo bayt) ile ücretlendirme yapılmaktadır. Dolayısıyla GPRS kullanıcısı sisteme bağlı olup veri transferi gerçekleştirmediği süreler için ücret ödemez.



Şekil 3.1. Paket anahtarlama ve Devre anahtarlama

Şekil 3.1'de görüldüğü gibi devre anahtarlamalı iletişimde A ve B uçları arasındaki iletişim için uçtan uca bir bağlantı kurulmuştur ve bu bağlantı süresince kullanılan hat diğer kullanıcılara hizmet verememektedir. Ancak Paket anahtarlamalı iletişimde A ve B uçları arasında uçtan uca bir bağlantının kurulmadığı ve eş zamanlı olarak birden fazla paketin taşınabilirliği görülmektedir.

Donanım cihazları GPRS veri iletişimi yönünden 3 sınıfta değerlendirilmektedir.

- A sınıfı: Devre anahtarlamalı ve paket anahtarlamalı bağlantıları herhangi bir kesinti olmaksızın aynı anda desteklemektedir (GSM ve GPRS).

- B sınıfı: Aynı anda iki hizmetten sadece birisini desteklemektedir. Bir GSM telefon çağrısı geldiğinde GPRS iletişimi kesilir, telefon çağrısı sona erdiğinde ise GPRS iletişimi kaldığı yerden devam eder.

- C sınıfı: Aynı anda iki hizmetten sadece birisini desteklemektedir. Hem GPRS hem de GSM hizmetlerini aynı anda kullanım ve kayıt mümkün değildir. Sadece SMS mesajları aynı zamanda alınabilmekte ve gönderilebilmektedir.

GPRS hizmeti alabilmek için GPRS uyumlu bir mobil terminal, GPRS hizmetini destekleyen bir GSM şebekesi aboneliği ve terminalde ilgili abonelik ile ilgili bağlantı ayarlarının yapılmış olması gerekmektedir.

GPRS hizmetinde kullanıcı veri transfer hızı aynı baz istasyonundan aynı anda hizmet alan kullanıcı sayısına bağlıdır. Bant genişliği paylaşımı söz konusudur. GSM evrimi için geliştirilmiş ver hızları anlamına gelen EDGE (Enhanced Data Rates for GSM Evolution) teknolojisinin 2003 yılında kullanılmaya başlaması ile, 384 kbps'ye kadar varan hızda iletişimi hızlarına çıkılması mümkün olmuştur [10].

Bu tez kapsamında GSM desteği olan PDA mobil cihazın merkez ile veri iletişiminde GPRS bağlantısı ile internet erişimi sağlanarak gerçekleştirilmiştir. GPS alıcısından geçerli konum bilgileri PDA cihazında oluşturulduğunda Bölüm 4.2'de ayrıntılı olarak açıklandığı üzere oluşturulan paketler güvenli bir şekilde GPRS bağlantısı ile merkeze gönderilmektedir.

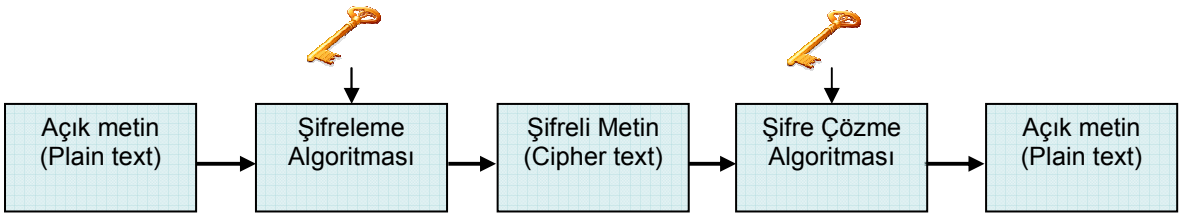
4. UZAKTAN KONUM TAKİP SİSTEMİ YAZILIMI

4.1. Genel Özellikler

Bu bölümde güvenli iletişim için kullanılan temel algoritmalar anlatılmıştır. Genel olarak asimetrik ve simetrik anahtar kullanarak şifreleme, AES- CBC modu, Diffie –Hellman Anahtar Değişimi (D-H), H-MAC Mesaj doğrulama algoritması gibi kavramlar hakkında genel bilgi verilmiştir.

4.1.1. Kriptografi

Veri iletişim ağlarında iletilen verilerin güvenliğinin sağlanması için veri şifreleme yöntemleri kullanılmaktadır. Kriptografi biliminde verinin şifrelenmesi (encryption) ve şifre çözme (decryption) işlemlerinde karmaşık işlemlerden oluşan matematiksel fonksiyonlar olan algoritmalar kullanılmaktadır. Şifreleme, açık metni (plain text) anlaşılamayacak bir forma dönüştürme (cipher text) işlemidir. Bu işlem bir matematiksel fonksiyon ve bir anahtar ya da anahtar çiftinin biri kullanılarak yapılır. Şifre çözme (deşifreleme) ise, şifrelenmiş mesajı (cipher text) , şifrelemede kullanılan fonksiyonun tersini ve bir anahtar veya anahtar çiftinin diğerini kullanarak açık metne dönüştürme işlemi olarak tarif edilebilir. Temel kriptografi mekanizması Şekil 4.1'deki gibidir.



Şekil 4.1. Temel Kriptografi Mekanizması

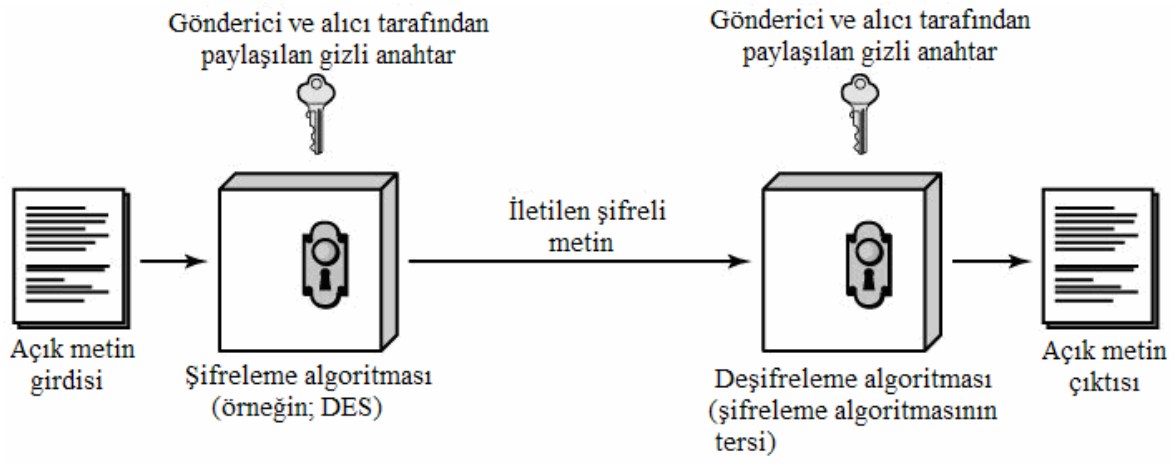
Şifreleme ve şifre çözme işlemini gerçekleştirmek amacıyla kullanılan *anahtar* ise, "0" ve "1"lerden oluşan uzun bir bit dizisidir. Anahtar uzunluğu kriptografi için kullanılan algoritmaya bağlıdır. Bir kripto algoritmasının olası tüm anahtar olasılıklarının oluşturduğu topluluk anahtar uzayı olarak adlandırılır. Bu anahtar uzayının bütün elemanlarının ilgili algoritmada teker teker denenmesi yöntemi kaba kuvvet saldırısı (brute force) olarak adlandırılır dolayısıyla kullanılan anahtar

uzunluğu arttıkça, olası anahtar sayısı arttığından saldırganın şifreyi çözmesi güçleşir, ama aynı zamanda da şifreleme ve şifre çözme hızı yavaşlar.

Kripto sistemler temelde, anahtar kullanma yöntemlerine göre, *gizli anahtarlı(simetrik) sistemler* ve *açık anahtarlı (asimetrik) sistemler* olmak üzere ikiye ayrılırlar.

- Gizli anahtarlı (simetrik) sistemler

Simetrik sistemlerde, şifreleme ve şifre çözme algoritmalarında aynı gizli anahtar (secret key) kullanılır. Bu gizli anahtar, yalnızca birbiri ile şifreli haberleşmek isteyen taraflarca (gönderici ve alıcı) bilinmelidir ve iletişimin güvenliği için anahtarın gizliliğinin korunması gerekmektedir. Simetrik algoritmalar asimetrik algoritmalara nazaran daha hızlı çalışırlar. Simetrik algoritmalara örnek olarak AES, DES, 3DES, Blowfish, IDEA, RC4 ve SAFER verilebilir [11].

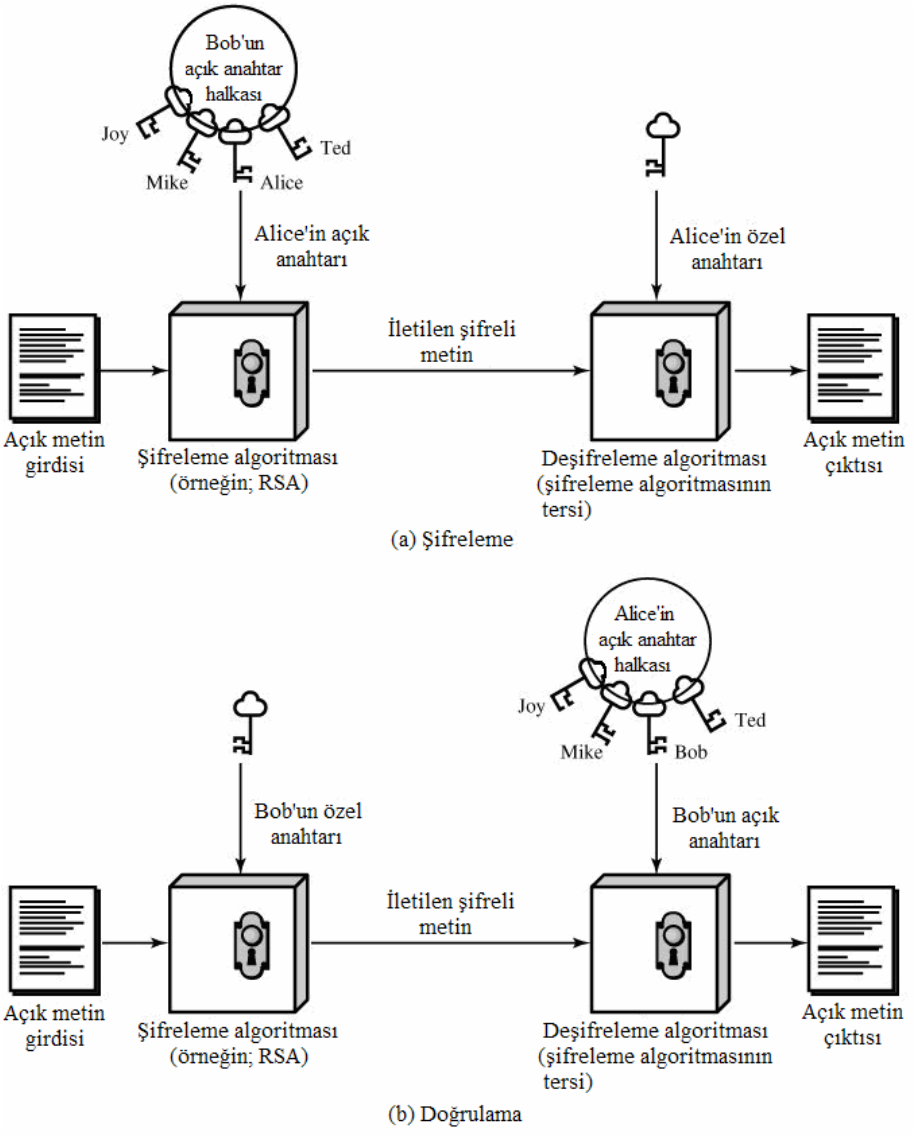


Şekil 4.2. Simetrik Anahtar kriptografi Mekanizması

- Açık anahtarlı (asimetrik) sistemler

Şifreleme ve şifre çözme algoritmalarında farklı anahtarlar kullanılır. Bu anahtarlar *açık anahtar (public key)* ve *özel anahtar (private key)* olarak adlandırılır. Kullanılacak bu iki anahtar birlikte üretilirler ve aralarında mantıksal bir bağ vardır. Anahtar üretme algoritmaları, bu anahtarlardan herhangi birine sahip olan bir şahsın, bu anahtarı kullanarak diğer anahtarı üretmesini matematiksel olarak oldukça güçleştirecek şekilde dizayn edilmiştir.

Asimetrik algoritmaların başarımları (performans) simetrik algoritmalara göre daha düşüktür. Asimetrik algoritmalarda her kullanıcının bir anahtar çifti vardır. Bir kullanıcının özel anahtarı, yalnızca kendi kullanımı içindir ve başkalarının eline geçmemesi gerekir. Kullanıcının açık anahtarı ise, o kullanıcıya mesaj göndermek isteyen herhangi başka bir kullanıcı tarafından kullanılabilir. Gönderici mesajı, alıcının açık anahtarı ile şifreler. Alıcı, gelen mesajı kendi özel anahtarı ile açar. Asimetrik algoritmalara örnek olarak RSA, ECC, Diffie-Hellman ve El Gamal verilebilir.



Şekil 4.3. Açık Anahtar (Asimetrik) kriptografi Mekanizması

4.1.2. Simetrik Şifreleme ve AES

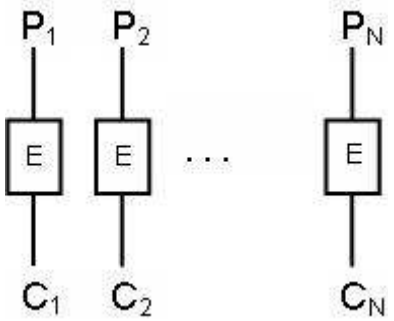
Ocak 1997'de NIST, yeni bir şifreleme standardının geliştirilmesi için bir çalışma başlatmıştır [4, 8]. Geliştirilecek yeni şifreleme standardının mevcut standart olan DES'in yerini alması düşünülmüştür. Çünkü DES'in 64 bitlik anahtar uzayı, gelişen teknoloji ve artan işlemci hızları karşısında güvenilirliğini yitirmeye başlamıştı.

NIST tarafından, yeni şifreleme standardını belirlemek amacıyla bir yarışma düzenlenmiş ve Eylül 1997'de algoritmalar için resmi çağrıda bulunulmuştur. Dört yıl boyunca süren değerlendirme ve eleme süreci sonrasında, Ekim 2000'de sonuç açıklanmış ve NIST, Joan Daemen ve Vincent Rijmen tarafından tasarlanan, Rijndael algoritmasının Gelişmiş Şifreleme Standardı (AES) olarak kullanılacağını ilan etmiştir [12], [13].

Ekim 2000'de, NIST, Gelişmiş Şifreleme Standardı'nın (AES) geliştirilmesi konulu bir rapor yayınlamış Rijndael algoritmasının 'AES' olarak adlandırılması tavsiye edilmiştir.

AES, Rijndael'in yeteneklerinin bir bölümünü gerçekleştirebilmektedir. Rijndael algoritmasında blok ve anahtar uzunlukları 128 bitten 256 bite kadar 32 bit araklıklarla birbirinden bağımsız olarak değişebildiği halde; AES algoritması, 128 bit blok uzunluğu ve 128, 192 ve 256 bit anahtar uzunluklarıyla kullanılmaktadır [12], [13].

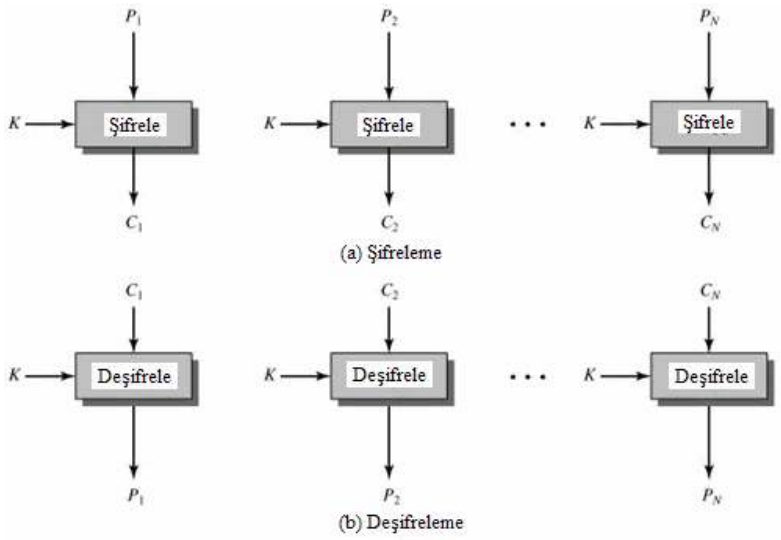
Blok şifrelemede, açık metin kullanılan blok uzunluğu (128 bit) kadar bitişik bloklara bölünür, her blok şifrelenerek şifreli metin çıkışları oluşturulur.



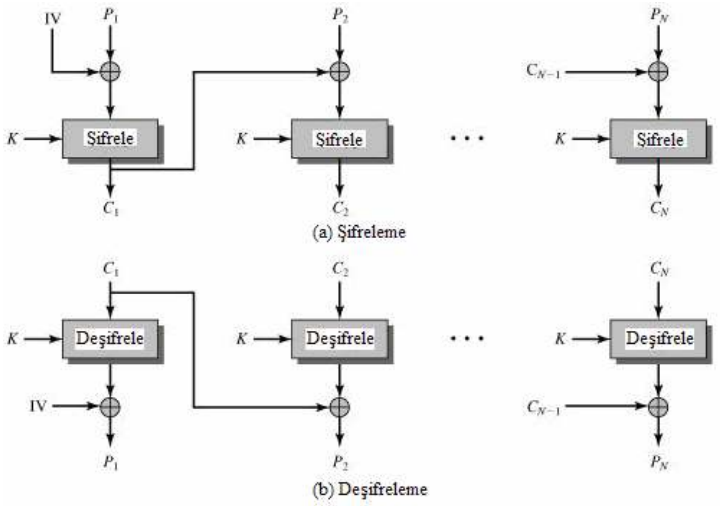
Şekil 4.4. Blok Şifreleme

P_1, P_2, …, P_N açık metnin blokları, yani her biri b bitten oluşan ardışık parçaları, C_1, C_2, …, C_N bu bloklara karşılık gelen şifrelenmiş metinler ve E ise şifreleme işlemini ifade etmektedir.

Bu tez kapsamında AES çalışma modu olarak NIST tarafından tanımlanmış çalışma modlarından CBC modu (Kapalı metin zincirleme) tercih edilmiştir. Bu mod ECB (Elektronik kod kitabı modeli) modu kullanıldığı durumda metin içinde aynı açık metin bloklarının olması durumunda üretilen şifreli blokların aynı olması nedeniyle güvenlik açığı yaratması nedeniyle tercih edilmiştir. Şekil 4.5 ve Şekil 4.6'da ECB ve CBC modları gösterilmiştir.



Şekil 4.5. Elektronik kod kitabı (ECB) modeli



Şekil 4.6. Kapalı metin zincirleme (CBC) modeli

CBC modunda şifreleme algoritmasının girişi, o anki açık metin bloğu ve bir önceki şifreli metin bloğunun XOR işlemine sokulmasından elde edilen mesajdan meydana gelir ve her blok için aynı anahtar kullanılır. Her bir açık metin bloğu için, şifreleme fonksiyonunun girişinin, açık metin bloğu ile doğrudan bir bağlantısı bulunmamaktadır, dolayısıyla ECB modunda ortaya çıkabilecek tekrarlayan b-bitlik (blok sayısı kadar) kalıpların gözlenmesi mümkün değildir.☒

☒
Deşifrelemede, her bir şifre bloğu deşifreleme algoritmasına sokulur ve sonuç, açık metin bloğunu üretmek için, bir önceki şifreli metin bloğuyla XOR'lanır.

$$\text{Şifreleme: } Cj = E(K, [Cj\text{-}1 \oplus Pj])$$

$$\text{Deşifreleme: } Pj = Cj\text{-}1 \oplus D(K, Cj)$$

Şifreli metnin ilk bloğunu oluşturmak için, bir başlangıç vektörü (IV) ile açık metnin ilk bloğu XOR'lanır. Deşifrelemede ise, açık metnin ilk bloğunu elde etmek için, IV, deşifreleme algoritmasının çıktısıyla XOR'lanır.

AES algoritması sabit 128 bit blok yapısı kullanan 128/192/256 bit anahtar boyutlarına ve 10/12/14 döngü sayısında işlem yapılan bir yapıya sahiptir.

Çizelge 4.1: AES Tur sayısı ve anahtar uzunlukları

	Anahtar Uzunluğu (N_k kelime)	**Blok Uzunluğu (N_b kelime)**	**Tur Sayısı (N_r)**
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Çizelge 4.1'de belirtilen 1 kelime 4 adet bayt'tan oluşmaktadır. AES' de 4x4 bayt dizileri ile işlem yapılmaktadır. AES'in şifreleme işlemleri aşağıda özetlenmiştir:

- İlk Tur

  1.Tur anahtarı ekle (AddRoundKey)

- Sonraki Turlar

1.Bayt yer değiştirme (Substitute Bytes)

2. Satırları öteleme (ShiftRows)

3. Sütunları Karıştır (MixColumns)

4. Tur anahtarı ekle(AddRoundKey)

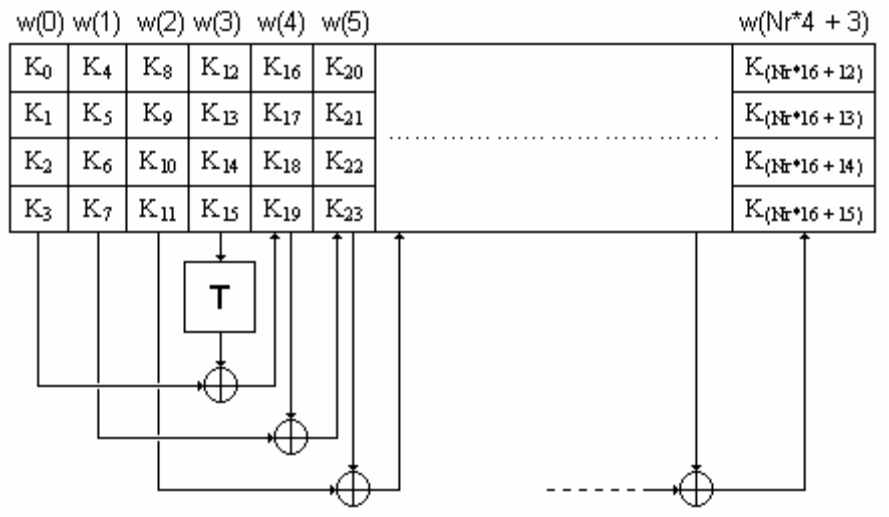
- Son Tur

1. Bayt yer değiştirme (Substitute Bytes)

2. Satırları öteleme (ShiftRows)

3. Tur anahtarı ekle (AddRoundKey)

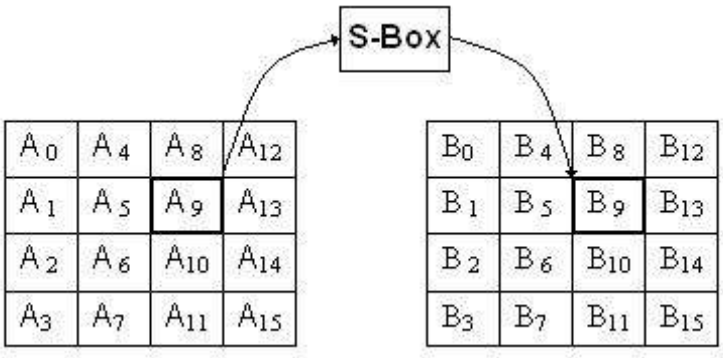
İlk turda yapılan ilk işlem, 16 bayttan oluşan (4x4 bayt matris) düz metini 16 baytlık döngü anahtarı ile XOR'lamak işlemidir. Bu aşamada kullanılan tur anahtarı, 16 baytlık gizli anahtarın kendisidir. Sonraki turlarda kullanılmak üzere Rijndael anahtar üreteci (Key Expansion) kullanılarak 4x (N_r+1) boyutunda bir matris elde edilir [5].



Şekil 4.7. Nr = 10, Nk = 4 için Anahtar Üreteci

Bayt yer değiştirme işlemi, AES algoritması içinde doğrusal olmayan tek dönüşümdür. Bayt yer değiştirme adımında, girişindeki durumun her bir bayt'ını,

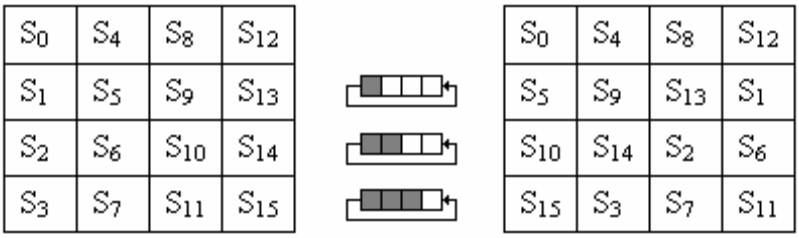
‘S kutusu’(S-Box) adı verilen bir değiştirme çizelgesi kullanarak, başka bir bayta dönüştürür [12], [13].



Şekil 4.8. Bayt yer değiştirme

Şifreleme işlemi sırasında satırları öteleme aşamasında İleri S kutusu (Forward S-box), Şifre çözümleme yapılacağı zaman Ters S kutusu (Reverse S-box) kullanılmaktadır [12], [13].

Satırları öteleme işlemi, 4x4 boyutundaki matrisin satırları (son üç satır) ile yapılmaktadır. İlk satırda öteleme yapılmaz. İkinci satır sağdan sola doğru bir pozisyon değiştirecek şekilde, dairesel olarak ötelenmektedir. Dairesel öteleme nedeniyle, 1. sütunda bulunan eleman ötelendiğinde 4. sütuna geçer. Üçüncü satır iki pozisyon, dördüncü satır da üç pozisyon sola doğru, dairesel olarak ötelenir. Satırları öteleme dönüşümünün durum baytlarının yer değiştirmesi Şekil 4.9’da gösterilmektedir.

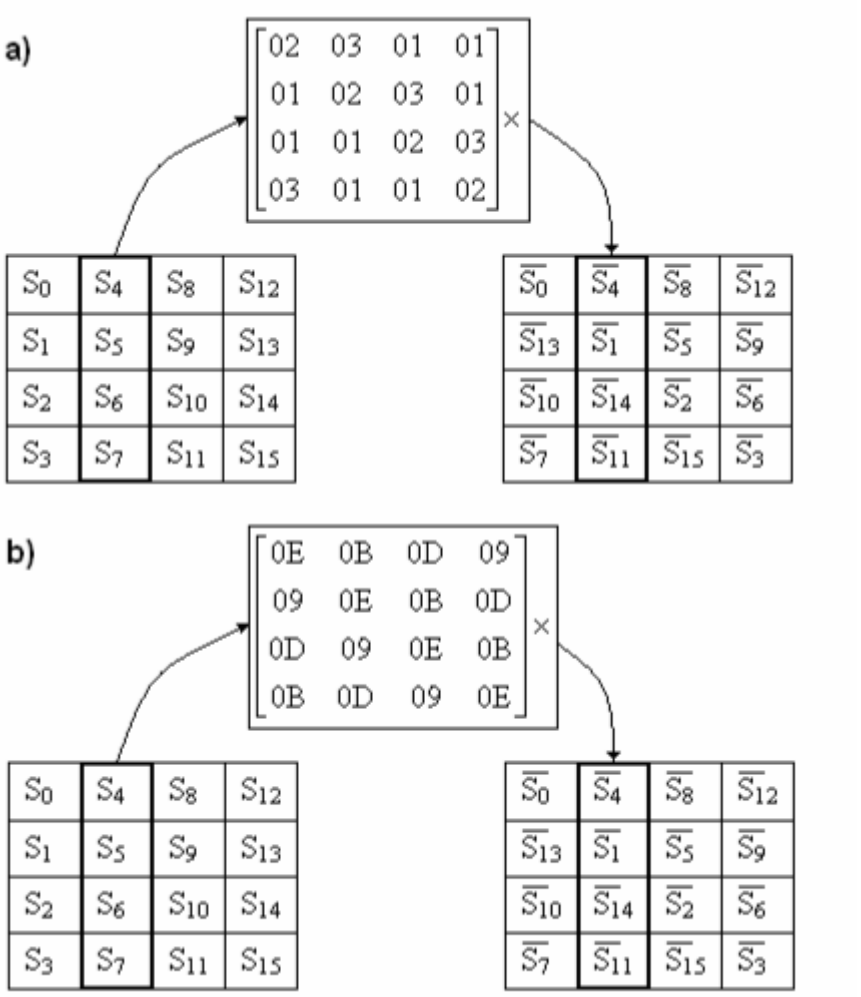


Şekil 4.9. Satırları öteleme

Şifre çözümleme yapılacağı zaman, öteleme işlemi aynı prosedür ile, ancak bu kez soldan sağa doğru öteleme olacak şekilde yapılacaktır.

Sütunları karıştırma aşamasında, 4x4 durum matrisinde sütunlar üzerinde işlem yapılmaktadır. Giriş durum matrisinin her sütunu, sütunları karıştırma dönüşümünden geçirilerek, çıkış durum matrisi elde edilir.

Sütunları karıştırma dönüşümü için yapılmaktadır. Giriş durum matrisinin her sütunu, sütunları karıştırma için sabit bir matris çarpımı şeklinde olmaktadır. Şifreleme ve şifre çözümleme aşamalarında kullanılan sabit matrisler ve yapı Şekil 4.10'da gösterilmiştir.

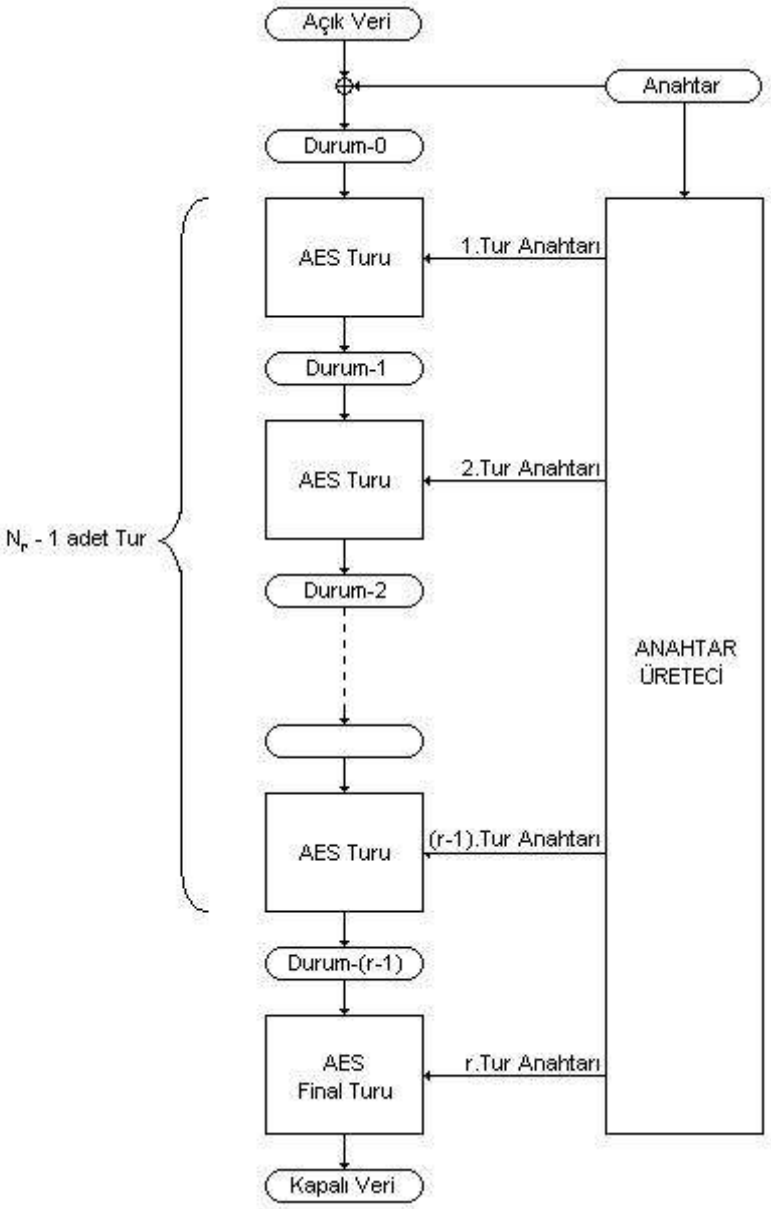


Şekil 4.10. Sütunları karıştırma a) Şifreleme

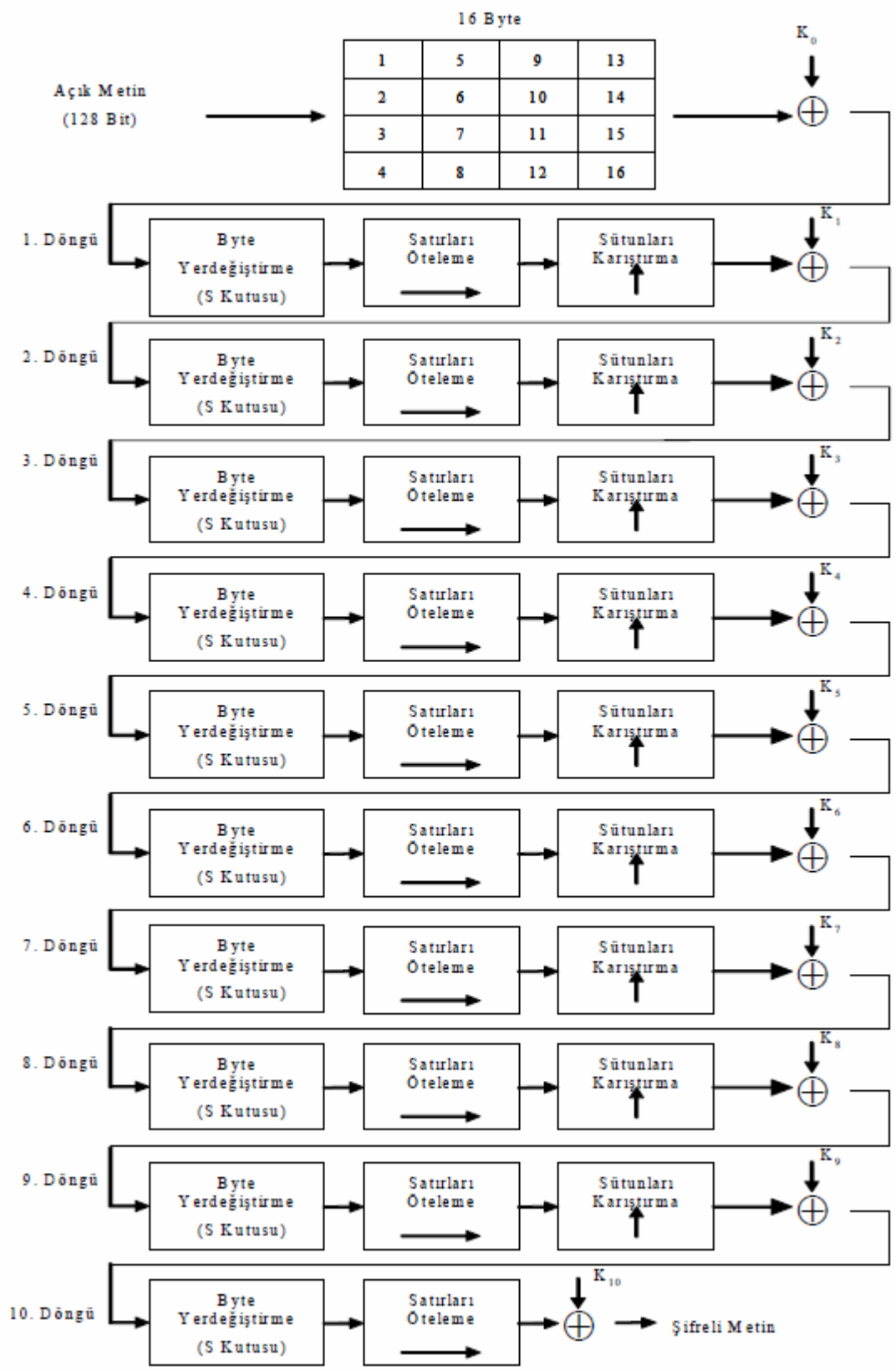
b) Şifre Çözme

He turun sonunda tur anahtarı ile toplama işlemi gerçekleştirilmektedir. Her bir tur sonunda elde edilen matris, Anahtar Üreteci tarafından ilgili tur için üretilmiş, tur anahtarı ile karşılıklı XOR işlemine tabi tutulmaktadır. Şifreleme ve şifre çözme işlemleri için aynı işlem uygulanmaktadır.

AES yapısında anahtar ile XOR' lama işlemi giriş verisine anahtar bilgiler eklemede, S kutuları lineer olmayan bir yapıyı sağlamada, bayt karıştırma ve sütunları karıştırma işlemleri de difüzyon sağlayarak yapının güvenirliğini arttırmaktadır [12], [13].



Şekil 4.11. AES Şifreleme Bloğu



Şekil 4.12. 128 bit anahtar için AES şifreleme akış şeması

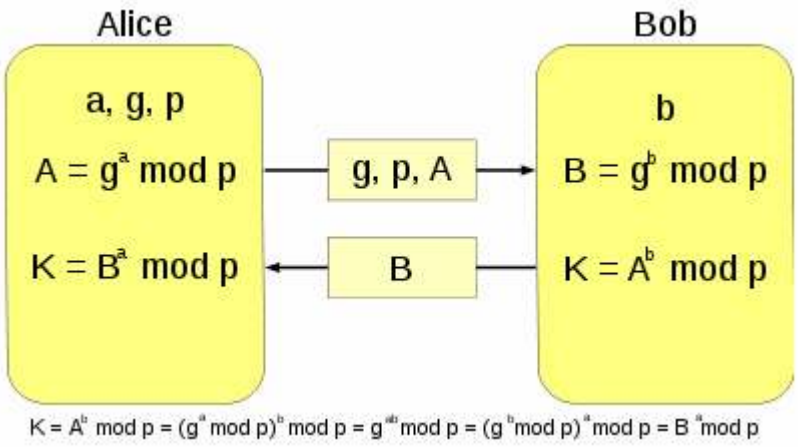
Şekil 4.11 ve Şekil 4.12'de AES şifreleme bloğu ve 128 bitlik gizli anahtar için 10 turlu AES şifreleme akış şeması gösterilmiştir.

Tez kapsamında AES şifreleme için Microsoft. System.Security.Cryptography ad alanında yer alan RijndaelManaged sınıfı kullanılarak gerçekleme yapılmıştır. Blok uzunluğu 16 bayt, gizli anahtar uzunluğu 128 bit ve CBC modu kullanılmıştır.

4.1.3. Diffie-Hellman Anahtar Değişimi

Diffie-Hellman Anahtar Değişimi (D-H) , güvenli haberleşmek isteyen tarafların, güvenli olmayan bir iletişim kanalında, güvenli iletişim için kullanacakları ortak anahtarı, beraber oluşturdukları (güvensiz iletişim kanalında haberleşerek) bir protokoldür [11]. Oluşturulan ortak anahtar vasıtası ile simetrik anahtar şifrelemesi yapılarak tarafların güvenli iletişimi sağlanabilmektedir.

Diffie–Hellman anahtar anlaşması Whitfield Diffie ve Martin Hellman'ın ortak çalışması sonucu 1976 yılında yayımlandı. Bu algoritma, güvenli olmayan bir iletişim kanalında paylaşılan bir ortak anahtar oluşturulabilmesinin ilk pratik metodu olmuştur. Birçok uygulamada D-H kullanılmaktadır.



Şekil 4.13. Diffie-Hellman Anahtar Değişimi

D-H, ayrık logaritma problemi üzerine kurulmuş ve güvenirliği çok büyük asal sayılar seçmeye dayanmaktadır [14], [15].

p yeteri kadar büyük bir asal sayı olsun öyle ki Zp de ayrık logaritma problemini çözmek mümkün olmasın. g'de Zp de primitif bir kök (primitive root) olsun. p ve g herkes tarafından bilinsin.

A ve B kişileri aşağıdaki yolu izleyerek ortak bir anahtar yaratabilirler:

- Alice, $0 \leq a \leq p-2$ eşitsizliğini sağlayan ve tesadüfi olan bir a sayısı seçer.

$$A = g^a \mod p$$

Hesaplar ve Bob'a gönderir.

- Bob, $0 \leq b \leq p-2$ eşitsizliğini sağlayan ve tesadüfi olan bir b sayısı seçer.

$$B = g^b \mod p$$

Hesaplar ve Alice' e gönderir.

- Alice , ortak anahtar K'yı şu şekilde hesaplar:

$$K = A^b \mod p = (g^a \mod p)^b \mod p = g^{ab} \mod p$$

- Bob, ortak anahtar K'yı şu şekilde hesaplar:

$$K = B^a \mod p = (g^b \mod p)^a \mod p = g^{ab} \mod p$$

Ortak anahtar K, oluşturulduktan sonra, Bu anahtar kullanılarak taraflar simetrik anahtar şifrelemesi ile güvenli iletişim yapabilirler.

a ve b gizli tamsayıları, her oturum sonunda yenilenmektedir. Bu nedenle Diffie-Hellman anahtar değişimi PFS' ye ulaşmaktadır. PFS' ye göre gizli ve ortak anahtar çiftleri kullanılarak elde edilen oturum anahtarları her oturumda değişmektedir. Böylece tarafların oturum anahtarı elde etmek için kullandığı gizli anahtara ulaşan bir saldırgan, tarafların geçmişteki ve gelecekteki oturumlarını çözümleyemeyecektir.

Diffie-Hellman anahtar değişimi orijinal tanımında, iletişimde bulunan tarafların birbirinin kimliğini doğrulaması için kimlik doğrulama sağlamamaktadır. Bu nedenle ortadaki adam saldırılarına açıktır. Ortadaki adam, birbirinden ayrık şekilde iki Diffie-Hellman anahtar değişimi gerçekleştirebilir. Birinci değişim için taraf A ile konuşarak taraf B taklidi yapar. İkinci değişimde ise taraf B ile konuşarak taraf A taklidi yapmaktadır. Dolayısı ile mesajların şifresini çözerek ve mesajları taraflara

yeniden şifreleyerek yollayabilir. Bu tür atakların engellenebilmesi için bir kimlik doğrulama mekanizması kullanılması gereklidir.

Taraflar Diffie-Hellman anahtar değişimi esnasında birbirlerine yolladıkları g^a ve g^b değerlerini MQV ya da IPsec protokol kümesinin IKE bileşeninde olduğu üzere imzalayarak ya da önceden paylaştıkları bir ortak anahtar ile değerleri birbirlerine şifreleyerek göndererek kimlik doğrulama işlemini gerçekleştirebilirler.

Diffie Hellman anahtar değişimi için tasarımda şu kapsamda bir yöntem izlenmiştir.

D-H gerçeklemesi için Mono.Security.Cryptography.DiffieHellmanManaged sınıfı kullanılmıştır [16]. D-H için taraflardan birinin belirleyeceği (sunucu tarafı) p değeri için Oakley grupları olarak adlandırılan ve IKE protokolünde de kullanılan 768 bit ve 1024 bit asal olduğu ispatlanmış ve daha önceden hesaplanmış sayılar kullanılmıştır [17] [18]. Böylece yazılım gerçeklemesi daha hızlı gerçekleşmektedir.

Çizelge 4.2. 1024 bit asal sayı

Group 2: 1024 bit asal sayı
Asal sayı: 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }

Tasarımda, Diffie-Hellman anahtar değişimi için gönderilen mesajlar (sunucu tarafı için p, g, g^a modp ve istemci tarafı için g^b modp) , önceden paylaşılan bir ortak simetrik anahtar kullanılarak Rijndael algoritması ile şifrelenerek gönderilmektedir. Böylece ortadaki adam saldırısı engellenmiştir. D-H ile elde edilen yeni anahtarlar, yeni oturum boyunca mesajların simetrik anahtar ile şifrelenmesi için kullanılmaktadır.

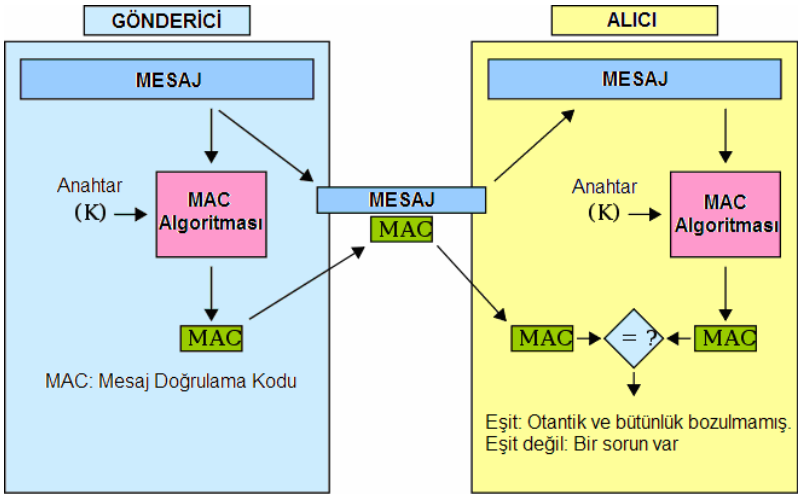
Tarafların D-H anahtar değişimi sonra hesaplayarak oluşturdukları veri, anahtar değişimi işlemini gerçekleştiren tarafların birinin belirlediği bir bilgi (party info) verinin sonuna eklenerek tüm verinin bir özet algoritmasından geçirilmesi (SHA1) ve özetleme algoritmasının fonksiyon çıkışı boyutunun simetrik şifreleme fonksiyonun gizli anahtar boyutu kadar kısmının oturum gizli anahtarı olarak kullanılması gerçekleştirilmiştir [19].

$$K = B^a \bmod p = (g^b \bmod p)^a \bmod p = g^{ab} \bmod p \quad \text{(Şekil 4.13)}$$

(K // party info) mesajı MD5 özetleme algoritmasından geçirilerek elde edilen 16 bayt değer Rijndael şifrelemesi için oturum gizli anahtarı olarak belirlenir. Oturum H-MAC gizli anahtarı ise K değerinin 20 bayt kadar kısmı kullanılarak belirlenir. Bölüm 4.1.4'de H-MAC algoritması ve özet algoritmaları anlatılmaktadır.

4.1.4. H-MAC Mesaj Doğrulama Algoritması

Şifrelemede, MAC (Message Authentication Code) algoritması, gizli bir anahtar ile doğrulanması istenilen belirli uzunluktaki mesajın birlikte kullanılarak, sabit uzunlukta bir MAC oluşturulması için kullanılmaktadır. Şekil 4.14'de görüldüğü üzere oluşturulan MAC, mesaj sonuna eklenerek, alıcıya ulaştırılmaktadır. Gizli anahtara sahip olan alıcı, mesaj içeriğinin değiştirilip değiştirilmediğinin kontrolünü yapabilmektedir [20].



Şekil 4.14. MAC Kullanımı

HMAC (Hash-based Message Authentication Code) ise, kriptografik bir hash (özet) fonksiyonu ile gizli anahtarın birlikte kullanılarak bir MAC hesaplamasının hesaplandığı yapıdır.

Tekrarlı bir özet fonksiyonu, mesajı sabit uzunlukta bloklara böler ve bir sıkıştırma fonksiyonu kullanarak tüm bloklara tekrarlayarak uygular. Örneğin, MD5 ve SHA-1 algoritmaları 512 bit uzunluğunda bloklar ile işlem görmektedir. MD5 fonksiyonunun çıkış verisi 128 bit, SHA-1 fonksiyonunun ise 160 bit uzunluğundadır. Aynı şekilde HMAC algoritmasının çıkış verisi boyutu da

beraberinde kullanılan özetleme fonksiyonu ile aynıdır. (MD-5 için 128 bit, SHA-1 için 160 bit).

HMAC mimarisinin tanımlaması ve analizi Mihir Bellare, Ran Canetti, ve Hugo Krawczyk tarafından ilk kez 1996 yılında yayımlanmıştır [21]. Ayrıca FIPS PUB 198'de HMAC kullanımı genelleştirilmesi ve standardı yayımlanmıştır [20]. HMAC-SHA-1 ve HMAC-MD5 yapıları IPsec ve TLS protokollerinde kullanılmaktadır.

Öyle ki:

- H(·) bir kriptografik özet fonksiyonu olsun.
- Kullanılacak gizli anahtar K, kullanılacak özet fonksiyonunun blok uzunluğu kadar sağa ilave sıfır eklenerek uzunluğu blok uzunluğuna eşitlenmiş olsun.
- m, doğrulanması istenilen mesaj olsun.
- ||sembolü birbirine bağlamayı ifade etsin
- ⊕sembolü XOR işlemini ifade etsin.
- Opad işlemi dışarıya eklemeyi ifade etsin. (0x5c5c5c…5c5c değerlerini bir blok uzunluğunda onaltılık sabit olarak eklemek)
- ipad işlemi içeriye eklemeyi ifade etsin. (0x363636…3636 değerlerini bir blok uzunluğunda onaltılık sabit olarak eklemek)

Bu durumda,

HMAC(K,m) işlemi matematiksel olarak,

HMAC(K,m) = H((K ⊕ opad) ||H((K ⊕ ipad) ||m)) olarak tanımlanır [21], [22].

HMAC'nin nasıl uygulanabileceğinin sözde kodu ise aşağıdaki gibidir.

function hmac (key, message)

opad = [0x5c * block size] *// blok uzunluğu kullanılan hash fonksiyonuna bağlıdır.*

ipad = [0x36 * block size]

if (length(key) > block size) **then**

key = hash(key) *// blok boyutundan uzun anahtarlar kısaltılır* **end if**

for i **from** 0 **to** length(key) - 1 **step** 1

ipad[i] = ipad[i] ⊕ key[i] *// Burada ⊕ (XOR)anlamındadır.*

opad[i] = opad[i] ⊕ key[i]

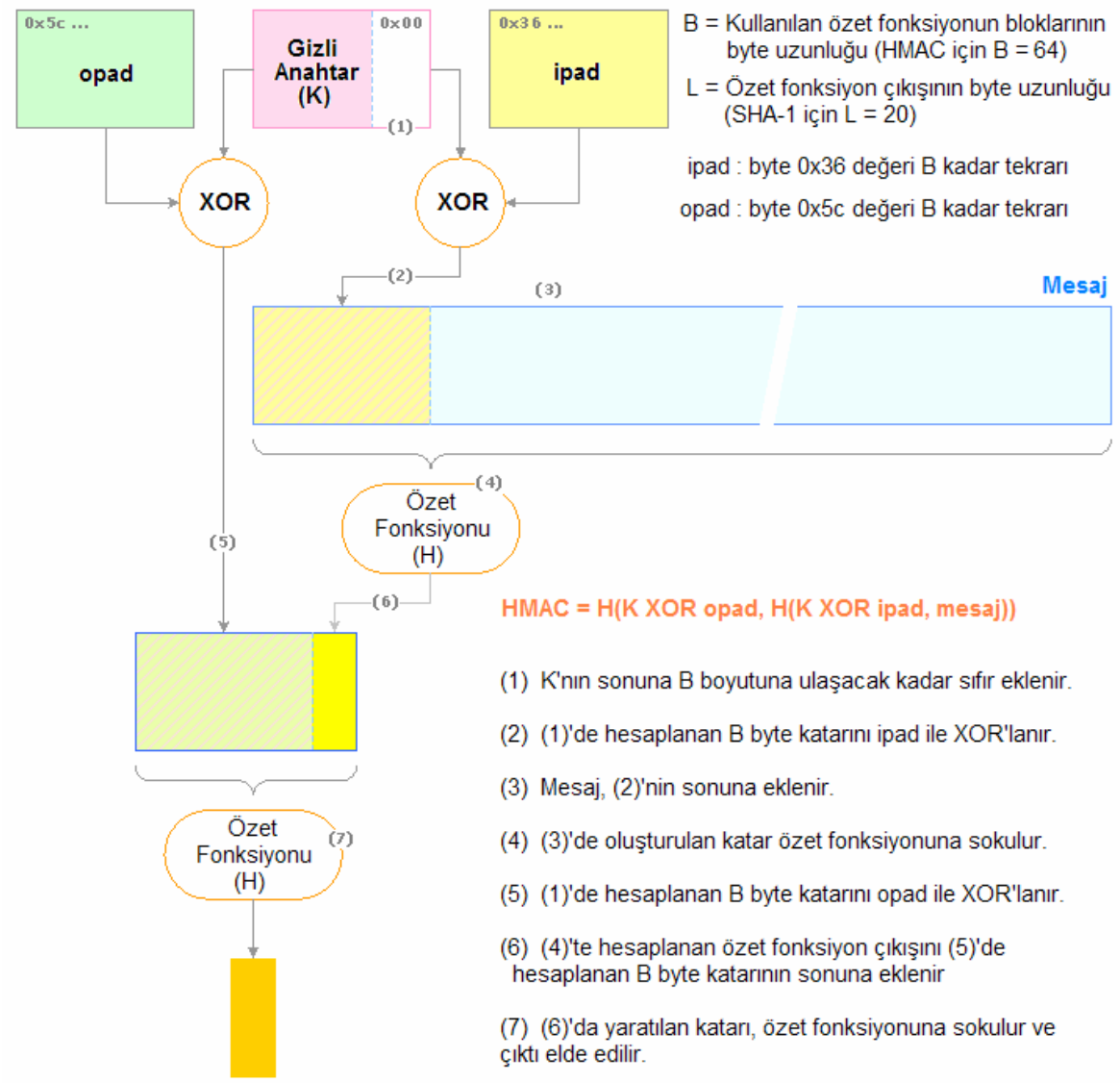
end for

return hash(opad || hash(ipad || message)) *// Burada || ardı ardına eklemedir*

end function

H-MAC tanımlamasının tasarımında, bir özet fonksiyonun gizli bir anahtar ile kullanıldığı durumda karşılaşılan saldırılar göz önüne alınmıştır. Eklenti saldırılarında, birçok özet fonksiyonu için gizli anahtar bilinmese dahi, mesajın içeriğine veri ekleyerek geçerli bir MAC elde etmenin mümkün olduğu belirtilmektedir. [5] [6]. MAC = H(anahtar || mesaj) olarak kullanımda bu açık oluşmaktadır. Alternatif olarak MAC = H(mesaj || anahtar) olarak kullanımda ise, kullanılan özet fonksiyonun gizli anahtar kullanılmayan halinde bir çakışma bulan saldırgan, MAC kullanımı için de bir çakışma bulabilir.

MAC = H(anahtar || mesaj || anahtar) kullanımı daha olumludur, ancak çeşitli kaynaklarda, bu yaklaşımda kullanılan anahtarların farklı seçilmesi halinde dahi bazı güvenlik açıklarının oluştuğu savunulmaktadır [23] [24] [25].



Şekil 4.15. H-MAC Mimarisi

Mevcut H-MAC tanımlamasındaki H(anahtar1 ||☒ H(anahtar2 ||☒ mesaj)) kullanımında bilinen bir eklenti saldırısı bulunmamaktadır [23]. Tanımlamada kullanılan ipad ve opad değerleri algoritmanın güvenliği için kritik değildir, ancak aralarında büyük bir Hamming mesafesi olacak şekilde belirlenmişlerdir. Hamming mesafesi, eşit uzunluktaki iki dizi arasında, karşılıklı pozisyonlara karşılık gelen sembol değerlerinin birbirlerinden farklı olduğu sayı ile belirlenir. (Örneğin “101101” ile “001100” arasında karşılıklı olarak bakıldığında ilk ve son bit değerleri birbirinden farklıdır ve Hamming mesafesi 2’dir.) Böylece içte ve dışta kullanılan gizli anahtarların daha az ortak bit değerleri olacaktır [23] [24].

H-MAC algoritmasının kriptografik gücü, kullanılan gizli anahtarın uzunluğu ile ilişkilidir. HMAC için Kullanılan gizli anahtarın uzunluğu, kullanılan özet fonksiyonun çıkış uzunluğunun (L) yarısından büyük ya da eşit olmalıdır [8]. Örneğin SHA-1 özet fonksiyonunun çıkışı 20 bayt (160 bit) olduğuna göre, anahtar uzunluğu 10 bayt'tan büyük seçilmelidir. Özet fonksiyonun çıkış uzunluğundan (L) daha büyük boyutta seçilen anahtarlar, fonksiyon gücünde anlamlı bir artış sağlamamaktadır [20]. Özet fonksiyonun blok uzunluğundan (B) daha fazla bayt içeren anahtar kullanılması durumunda, anahtar önce özet fonksiyondan geçirilerek L uzunluğunda bayt katarı elde edilir ve sonuç H-MAC anahtarı olarak kullanılır. Kullanılan anahtarlar rassal olmalı ve belirli aralıklar ile yenilenmelidir.

Uygulamada H-MAC gerçeklemesi için Microsoft.NET "System.Security.-Cryptography.HMAC" sınıfı kullanılmıştır [26]. Özet fonksiyonu olarak SHA-1 algoritması kullanılmıştır. Kullanılan gizli anahtar boyutu 160 bit uzunluğunda seçilmiştir.

4.1.5. RSA Asimetrik Şifreleme Algoritması

RSA Algoritması Ron Rivest, Adi Shamir, ve Leonard Adleman tarafından 1977 yılında yaratılmıştır. Hem mesaj şifreleme, hem de mesaj imzalama amacıyla kullanılabilmektedir. Güvenliği tam sayıları çarpanlara ayırmanın algoritmik zorluğuna dayanmaktadır [11].

Anahtarlar şu şekilde üretilir:

1. İki adet birbirinden farklı asal sayı (n, p) seçilir. $n = p \cdot q$ hesaplanır.
2. Bu sayıların totienti olan $\Phi(n) = (p - 1) \cdot (q - 1)$ hesaplanır.
3. $1 < e < \Phi(n)$ ve $\Phi(n)$ ile aralarında asal olan bir tamsayı (e) üretilir.
4. $d \cdot e \equiv 1 \pmod{\Phi(n)}$ olacak şekilde bir d sayısı üretilir.

Ortak anahtar şu verileri içerir:

- n, yani modülüs
- e, yani ortak üs (bazen şifreleme üssü de denir)

Özel anahtar şu verileri içerir:

- n, yani modülüs
- d, yani özel üs (bazen şifre çözme üssü veya deşifre üssü de denir)

Hesaplama işlemlerini kolaylaştırmak için aşağıdaki değerler özel anahtar olarak saklanır.

- p ve q (anahtar üretiminde kullanılan asal sayılar)
- d mod(p – 1) ve d mod(q – 1) (dmpl ve dmql olarak adlandırılırlar)
- (1/q) mod(p) (iqmp olarak adlandırılır)

Örnek olarak:

p = 61 ve q = 53	Birbirinden farklı seçilmiş asal sayılar (Gizli)
n = p.q = 3233	Modülüs (Paylaşılabilir)
e = 17	Ortak üs (Paylaşılabilir)
d = 2753	Özel üs (Gizli)

Ortak anahtar (e, n) ve Gizli anahtar (d, n) dir.

Şifreleme fonksiyonu, Şifrele(veri) = $veri^e$ mod(n)

Şifre çözme fonksiyonu için Çöz(şifre) = $şifre^d$ mod(n) olarak kullanılır.

Şifrelenecek veri 123 sayısı olsun:

Şifreli veri = 123^e mod(n) = 123^{17} mod(3233) = 855 olarak hesaplanır.

Dolayısıyla 123 verisinin yukarıdaki anahtar ile şifrelenmiş hali 855 değeridir. Bu şifreli değeri deşifre etmek için:

Çöz(855) = 855^{2753} mod(3233) = 123 olarak elde edilir.

Uygulamada RSA gerçeklemesi için .NET System.Security.Cryptography alan adında yer alan RSA sınıfı kullanılmıştır [25]. Anahtar boyutu olarak 1024 bit uzunluğunda anahtarlar kullanılmıştır.

```
<RSAKeyValue>
       <Modulus>

              x1flia6JuI2KRJaHPjuE/Pfgv8xhwi6cS48M3MIGSq8t+Pzrrf
              eduCRtnVmcqDzoGVqDib3j7hoyMJXDxZioOOcQ7sLoJ8nB8Jpi
              06BZxzvsTZEJxzv+UaBkRsmZAW6AEwxHP5FGvWKclmYbIQ0j4R
              U69D33uFb/9DcVU7ibeAM=
        </Modulus>
        <Exponent>
              AQAB
        </Exponent>
        <P>
              5Xb/HKZB744BjW6PG7TNshQv7bK32UWo9K0QLULc1afKAO+Cf1
              fXb01LlVFSQgIqj4sxgYCV+y2GACQrTkjQLw==
        </P>
        <Q>
              3mUxfWkCVUK+1DsZg1DXoG4rDt/7nvyBRxSuism+eGQXkxa+Xc
              yTIU0u+zYUB6tw9Nd4qiQK6tNotWfWn+lsbQ==
        </Q>
        <DP>
              IW4wxK/EM25w83YEC3lP7n3abbr35gOsFdmQ3cc7/TSwzWoXjQ
              3zKRgNxvdOEKfYI4mX8NIXB70eJK8gUAtzqQ==
        </DP>
        <DQ>
              2dg99h6ajnHnA3UwtJTH4vqHava0PzQQXlQ/9OtRjBRfk1FAAB
              mYNZKpZqG+VRT2fJ9kLyVHHGchtjCU5+d6sQ==
        </DQ>
        <InverseQ>
              Lv0rzyYQbR3Ux5umZZknEKjLip1faaVJk7TRwX5PFDh602NvtZ
              Xyv/KEWT5hWVLEQLlnRvZ1teomU5xDgBFbdg==
        </InverseQ>
        <D>
              SXdKYBCULxDy6U40r6nN3FEaYnVXffIhpfMlgWJ8YR/oqH/F+A
              pu9QNv5nK1Ooef3mAuZCA4lXLQFnnLedJGBHQUJxT0wf0nymbN
              T10g/ZKBemZe8EjN2hLLl6wn+ydwyx6PjY8jqzQ8lLeEx2rr9r
              1oCPxx5Qy+3++ahSs81yk=
        </D>
</RSAKeyValue>
```

Şekil 4.16. RSA anahtarı

RSA sınıfı kullanılarak üretilen anahtar çiftleri XML dosyası olarak kaydedilir. Şekil 4.16 ‘da üretilmiş RSA sınıfı kullanılarak üretilen RSA anahtar çiftinin XML gösterimi verilmiştir.

4.2 Tasarım Bilgileri

Mevcut araç takip ve benzeri sistemlerde, GPRS üzerinden gönderilen veriler açık gitmektedir. GPRS üzerinden yapılacak bir atak sonucu cihazların gönderdikleri verilerin taklidi ile sistemin çalışmaz hale gelmesini sağlamak mümkündür. İkinci bir problem bunların dinlenmesi neticesinde cihazların yerlerinin tespiti de söz konusudur. Bu ise başka bir takım güvenlik problemlerine yol açabilir.

Verilerin güvenli bir protokol ile transfer edilmesi durumunda bu problemler olmayacaktır. Ancak güvenli iletişim için kullanılan protokollerde de arada kullanılan anahtarların zaman içinde değiştirilmesi, cihazların bu verileri güvenli bir şekilde saklaması, cihazların içindeki programın çalınarak başka cihaza kopyalanması durumunda sistemin kararlılığının bozulmamasının sağlanması gibi ek gereksinimler ortaya çıkmaktadır. Ayrıca yine paket tekrarı veya uygulama kodunun çözülerek network protokolünün anlaşılması durumunda bile sisteme izinsiz girişin mümkün olmaması gerekmektedir.

1. Bu gereksinimler doğrultusunda yapılması gereken birinci aşamada cihazın sisteme girebilmesi için sisteme tanıtılması işlemi gerçekleştirilmelidir.

    a. Mobil uygulama ve sistem servisleri bir mekanizma ile güvenli iletişime geçebilmelidir. Bu noktada her oturum için kullanılan anahtarların oluşturulmasında D-H yöntemi kullanılmaktadır.

    b. Rastgele bir cihazın da bu sisteme girememesi için periyodik olarak (örnek: 5 dakikalık aralıklarla) değiştirilecek olan bir giriş anahtarı ile aygıtın kurulum için getirilen cihaz olduğu dışarıdan bir cihaz olmadığı, (AUTHENTIC) tespit edilebilir. Servis tarafında bir kullanıcı ara yüzü ile anahtar görüntülenir. Mobil uygulama kurulum aşamasında bu anahtar ile kendini sisteme tanıtır. Anahtarın üretilmesinde Simetrik şifreleme servislerinin anahtarlarını üretmekte de kullanılan ve System.Security.Cryptography ad alanında (namespace) yer alan RNGServiceProvider sınıfı Random Number Generator kullanılır [27]. Buradan 8 bayt değer onaltılık formata

çevrilerek 16 bayt Authentication key (doğrulama anahtarı) elde edilir.

c. “Unique ID” cep bilgisayarının donanımında cihaz üreticisi, modeli ve seri numarasına bağlı olarak belirlenmiş ve cihazı tarif eden bir değer olan ve cihaza ait eşsiz tekil numarasıdır [28]. “Unique ID” Hal_Get_UniqueId metodu kullanılarak elde edilir [29], [30]. Bu aşamada cihaz, Authentication key ile birlikte cihaz UNIQUE ID (tekil numara) değerini sunucuya gönderir, sunucudan geriye cihazın veritabanındaki kayıt anahtarı olan Unit ID bilgisini alır. Cihaz daha önceden kurulmuş olabilir. Ancak cihazdaki verilerinin kaybolması nedeniyle veya bir sebepten dolayı konfigürasyonun bozulması gibi durumlarda tekrar kurulması gerekebilir.

d. Cihaz geri dönüşte aldığı “Unit ID” değerini konfigürasyonla beraber saklayacaktır. Konfigürasyon verileri cihaz belleğinde cihazın UniqueID değeri Rijndael algoritmasına anahtar olacak şekilde şifrelenerek saklanır. Bu şekliyle verilerin başka bir cihaza kopyalanarak çalıştırılması engellenecektir.

e. Uygulama bu bilgiler alındıktan sonra veri göndermeye başlar.

2. Veri gönderme işleminde mümkün olan en az veri paketinin geri dönüşü sağlanarak GPRS data optimizasyonu sağlanacaktır.

a. Cihaz ilk aşamada sunucu üzerinde bir oturum açmak için D-H güvenli anahtar değişimi ile güvenli oturum açma işlemini başlatacaktır. Oturum açma işlemi neticesinde oturumu tarif eden bir SessionID değeri asp.net tarafından üretilir ve cookie olarak cevapla beraber gönderilir. Oturumu bu değer tarif eder. Bu değer cihazın ve sunucunun bildiği şifreleme anahtarları olmadan bir işe yaramayacaktır. SessionID değeri sunucu tarafında belirlenen bir değer ile tanımlı süre ile (sessionTimeout) geçerlidir. Bu süre geçtiğinde oturumda kapanmış olur. Cihaz bu süreden daha kısa süreler ile veri gönderir ise (hareket halindeyken) oturumun ömrü her veri göndermeyle beraber bir sessionTimeout süresi kadar daha

uzatılır. SessionID değerini içeren cookie ifadesi her http isteği ile birlikte gönderilir. İlk bağlantıda bu değer belli olmadığından gönderilmez.

b. Oturum açıldıktan sonra cihaz konum değişikliklerini bu SessionID ile birlikte D-H anahtarları kullanılarak şifrelenmiş ve imzalanmış olarak gönderecektir. Her cihaz veri göndermek için oturum açmak zorundadır. Sunucu tarafında açılan oturum belleği ancak SessionID ile erişilebilir hale gelmektedir. Şifreleme, çözme ve HMAC-SHA1 imzalama için gerekli D-H anahtarları oturum belleğinde saklanır. Eğer sunucu yeniden başlatılır (restart) ya da IIS tekrar başlatılırsa cihazların tekrar oturum açmaları gerekecektir. Oturum açıldıktan sonra işleme kaldığı yerden devam edebilir.

c. Cihaz iki durumda veri gönderecektir.

i. Cihazın GPS alıcısından konum konfigürasyon ayarlarında belirlenen hassasiyette alınabiliyorsa ve son gönderilen noktadan sonraki toplam konum değişimi konfigürasyon ayarlarında belirlenen bir mesafeyi (Örneğin 50mt) geçmiş ise,

ii. Son konum gönderme zamanından sonra hiçbir konum değişikliği olmasa bile belli peryotlar ile (Örneğin 2 dakikada bir).

d. Paketlere sıra numarası verilecektir. Bu şekliyle veri tekrarı yöntemiyle sistemin zehirlenmesi engellenecektir. Sıra numarası her gönderme denemesiyle bir arttırılacaktır. Bu şekliyle her gönderilen verideki sıra numarası bir öncekinden büyük olacaktır. Bu sıra numarası sayesinde paket zehirlemeleri ve aynı paketin tekrar gönderilerek sistemdeki verilerin kirletilmesi mümkün olamayacaktır. Olası bir durumda konfigürasyon dosyasında bozulma veya problem olması durumunda yeniden kurulumla sıra numaraları sunucu ve istemci tarafında sıfırlanır.

e. Paket geçerli bir paket ise veri işleme sokulacaktır. Paket geçerliliği şu şekilde kontrol edilecektir.

i. İstemciden geldiği, Unit Id değeriyle denetlenerek böyle bir cihaz olup olmadığı (sistemde kayıtlı olup olmadığı) kontrol edilecek.

ii. Cihaz'ın gönderdiği paketteki imzanın (HMAC-SHA1 ile alınmış özet) D-H anahtarları ile kontrol edildiğinde geçerli olması gerekmektedir. Bu veri bütünlüğü için de gereklidir.

iii. Sıra numarası, veri veritabanındaki son değerden büyük olacaktır. Bazı durumda GPRS bağlantısı açık olmadığında ve cihaz bu verileri gönderemeden kapanırsa veriler disk üzerinden tekrar yüklenir. Burada kontrol için sıra numarası geçmiş konum bilgilerinde var mı şeklinde kontrol yapılabilir. Bu geçmiş konum bilgilerinde yoksa geçmiş konum bilgilerine kaydedilir.

3. Cihaz servis ile (sunucu tarafı) bağlantı aşamasında oturum açma talebini gönderir. Burada D-H anahtarını üretilmesi işlemi sistemi yoran bir işlemdir. Bu noktada yapılacak bir paket tekrarı saldırısı ile sunucunun servis dışı kalması sağlanabilir. Bunu engellemek için gönderilen paketin içine sistem saati (time stamp) eklenmektedir. Saatler ilk aşamada senkronize edilir. Sonraki aşamada gönderilen veri içinde yer alan saat ile sunucu saati kontrol edilir. Bu şekliyle eğer paket tekrarlama saldırısı ile gönderilse bile belirlenebilecek bir süre sonunda sistem bu paketleri dikkate almayacağından sistemin durdurulması söz konusu olmayacaktır.

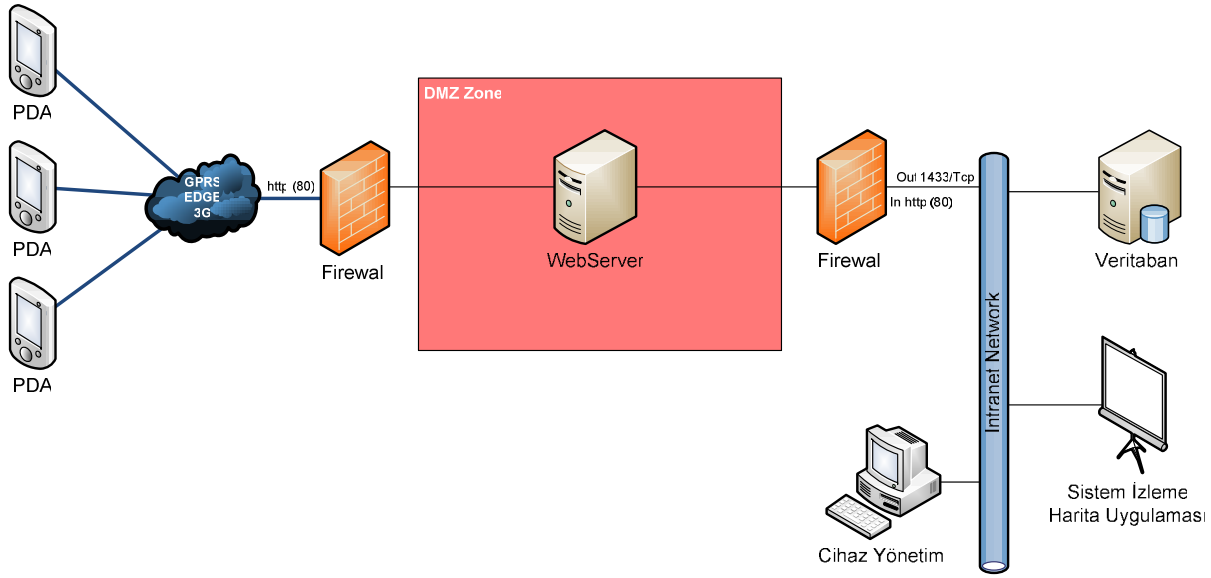
4. Servis gelen verileri veritabanına kaydedecektir. Cihazın bulunduğu en son güncel geçerli konum bilgisi cihaz kaydıyla beraber saklanacaktır. Bu şekliyle cihazın o anda nerede olduğu bilgisi cihaz kayıtları ile birlikte gelecektir. Eğer istenirse geçmiş konum bilgileri çekilerek cihazın hareketliliği görülebilir.

5. Masaüstü bilgisayarda çalışacak bir uygulama ile harita üzerinde cihazlar en son konumları itibariyle görüntülenecektir. Uygulamada, cihazların geçmiş konum bilgilerinin belirlenen tarih/saat aralıkları için gösterilmesi de gerçekleştirilecektir.

6. Cihaz çalındı ise cihaza hard reset (fabrika ayarlarına sıfırlama) atılması sağlanabilir. Bunun için oturum açma işleminde veya veri göndermede cihaza gönderilecek bir bilgi ile cihaz içindeki program korunabilir.

4.2.1. Donanım Mimarisi

Uygulamanın çalışacağı donanım mimarisi Şekil 4.17'deki gibidir.



Şekil 4.17. Donanım Mimarisi

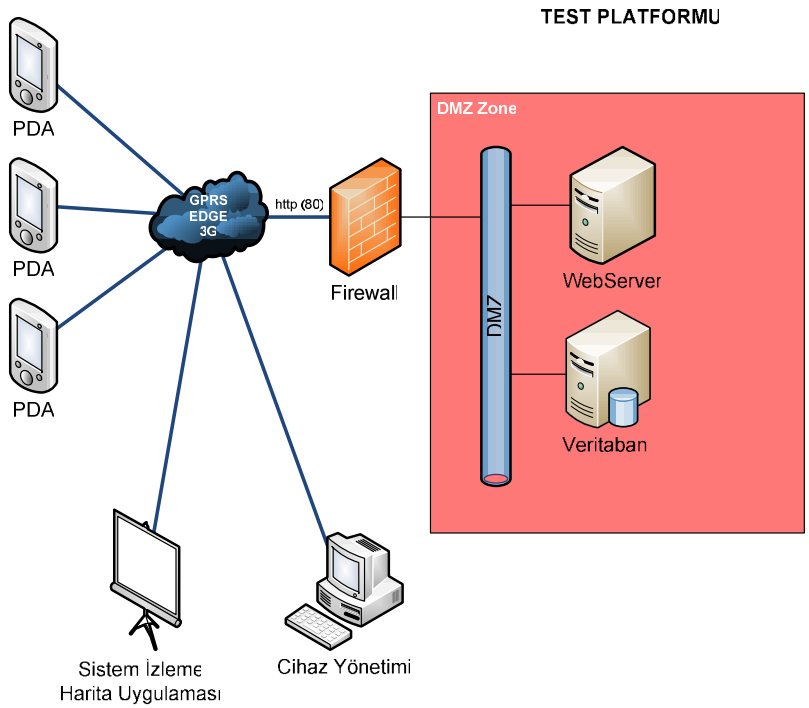
Donanım olarak ;

1. Windows Mobile 5.0 işletim sistemi yüklü mobil cihaz (PDA)
2. Firewall
3. Web sunucusu
4. Veritabanı sunucusu
5. Sistem izleme harita uygulamasının çalışacağı bilgisayar

6. Cihaz yönetiminin yapıldığı bilgisayar

ihtiyaçları bulunmaktadır.

Test Platformu için kullanılacak donanım mimarisi ise Şekil 4.18'deki gibidir.



Şekil 4.18. Test Platformu

Test platformu sırasında Web sunucu, veritabanı ve yönetim aynı bilgisayar kullanılarak sistem gerçekleştirilmiştir. Sistem izleme ve harita uygulaması ve cihaz yönetiminin çalışmasının gösterilebilmesi amacıyla GPRS bağlantısı olan bir bilgisayar kullanılarak internet bağlantısı üzerinden sunucu bilgisayara VPN uzak bağlantı yapılmıştır. Böylece test uygulamasında sunucu üzerindeki verilere uzaktan güvenli erişim yapılması gerçekleştirilmiştir.

4.2.2. Geliştirme Platformu

Uygulamanın geliştirilmesi için Microsoft.NET platformu kullanılmış olup, C# ve ASP kullanılmıştır. Veritabanı uygulaması için Microsoft SQL Sunucu kullanılmıştır.

Uygulamaların çalıştığı yerlere göre:

a. Mobil Cihaz

Cihazda ".NET Compact Framework" yüklüdür ve bir Windows uygulaması çalışmaktadır. Bu uygulama, ilk kurulumda verilerin aktarılması, GPS verilerinin alınması ve şifrelenerek gönderilmesi bu uygulama tarafından gerçekleştirilmektedir. Gerektiğinde "hard reset" atarak cihazdaki verileri tamamen silebilir.

b. Web Sunucu (Web Servis):

Buradaki uygulama mobil cihazlar ile iletişimi sağlayacaktır. Cihazların kurulması için gerekli authentication (kimlik doğrulama) anahtarının oluşturulması ve gelen konum bilgilerinin veritabanına aktarılmasından sorumludur.

c. İzleme ve Yönetim Uygulaması

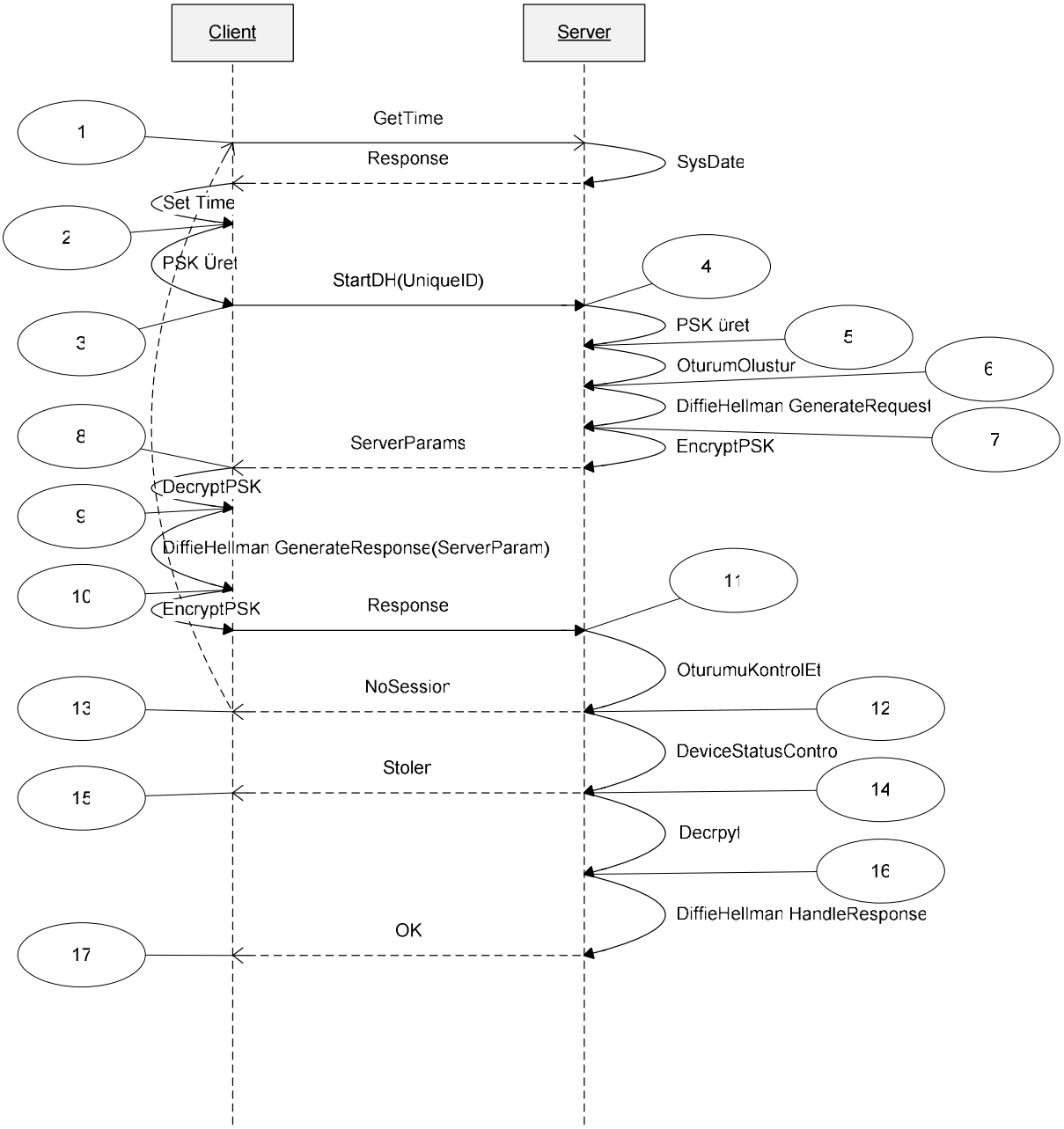
Cihazların harita üzerinden takibinin yapıldığı ve bu cihazların gerekirse sistemden kaldırılmasını yapacak uygulamadır.

4.3. Kod Yapısı

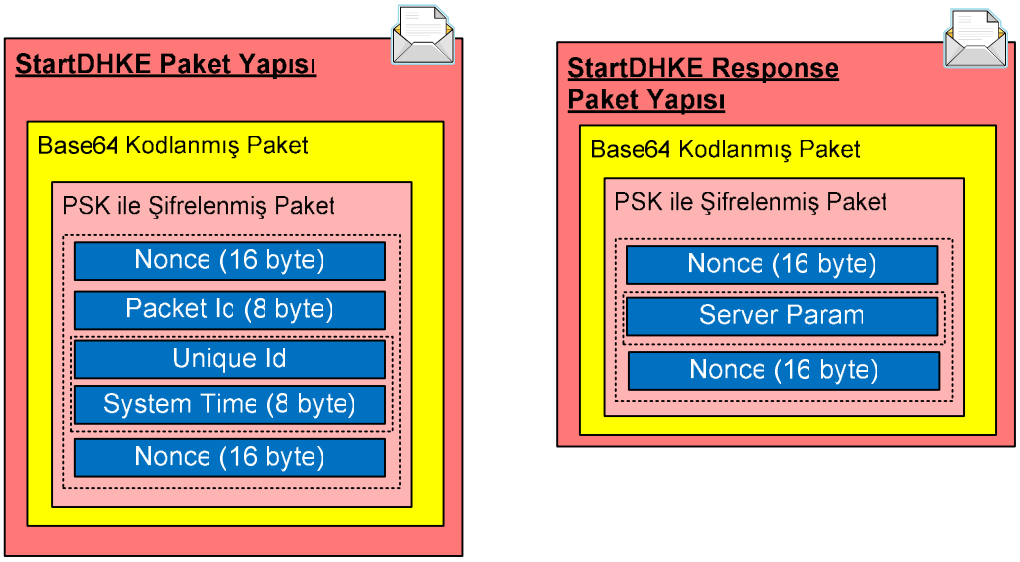
Mobil cihazda programın kurulup çalışır hale gelmesi, D-H anahtar değişimi işleminin gerçekleştirilmesi ve cihazın sistemde bir oturum açması safhalarında paketlerin şifrelenmesinde 2 tür senaryo yaratılmış ve uygulanmıştır. 1.senaryoda PSK anahtar kullanılarak simetrik AES şifreleme, 2. senaryoda ise sunucu tarafında üretilen anahtar çiftleri kullanılarak asimetrik RSA şifreleme yapılmıştır.

4.3.1. PSK Kullanımı

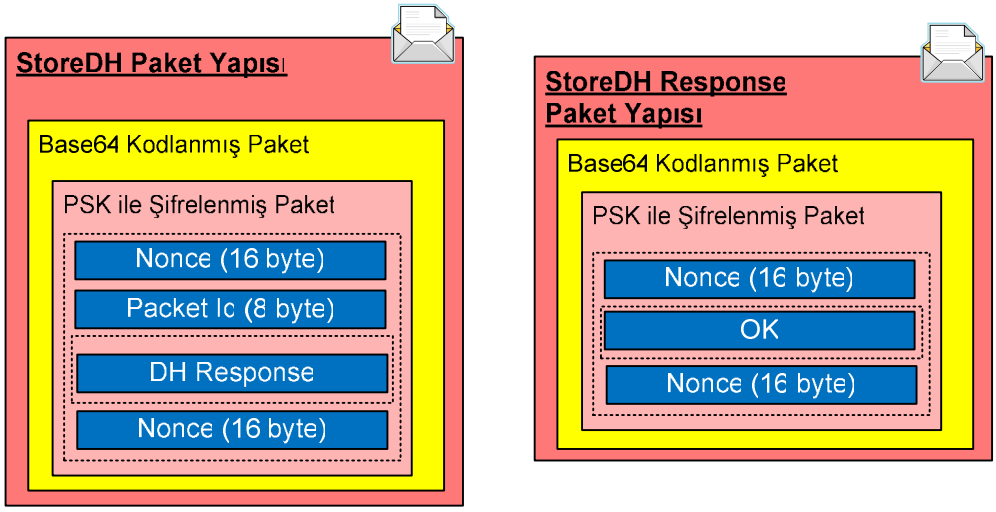
D-H anahtar değişimi işleminin gerçekleştirilmesi ve cihazın sistemde bir oturum açması ile ilgili sequence (düzen) diyagramı aşağıda gösterilmiştir:



Şekil 4.19. Oturum Açma Sequence (düzen) Şeması



Şekil 4.20. 3. ve 7. adım için istek ve cevap paket yapıları



Şekil 4.21. 10. ve 16. adım için istek ve cevap paket yapıları (Cevap için OK, Stolen, ya da No Session bilgileri olabilir)

1. Sistem saati güncellemesi için talepte bulunulur. Sistem saati bilgisi sunucudan döndürülür. Bu değer ile cihazın saati güncellenir.

2. PSK, cihazda konfigürasyondan okunan bir metin PasswordDeriveBytes sınıfı kullanılarak elde edilir. Bu sınıfa istemci ve sunucu tarafında sabit olarak tanımlanmış 32 bayt bir sabit SALT değeri parametre olarak verilir. Bu sınıftan 16 + 16 = 32 bayt bir anahtar üretilir.

3. Cihaza ait Unique ID değeri, PSK Rijndael algoritması gizli anahtarı olacak şekilde şifrelenerek StartDH talebinde parametre olarak gönderilir.

4.Web.Config içinde tanımlanmış bir parametre kullanılarak PasswordDeriveBytes sınıfı aracılığı ile PSK üretilir.

5. StartDH paketi içindeki gelen veri bu PSK ile Rijndael algoritması kullanılarak çözülür. Unique ID değeri oturum ile birlikte kaydedilir.

6. D-H anahtar değişimi için gerekli parametreler üretilir.

7. Üretilen parametreler PSK ile şifrelenerek cihaza gönderilir.

8. Cihaz verileri PSK ile açmaya çalışır.

9. İstemci, aldığı sunucu parametrelerini kullanarak D-H özel anahtarını üretir. Bu anahtarı ürettikten sonra sunucuya aynı özel anahtarı üretmesi için gerekli parametre hazırlanır. Burada üretilen D-H özel anahtarı doğrudan doğruya kullanılmaz. İşlem aşamasında kullanılan Rijndael simetrik anahtarlama için gerekli olan IV, anahtar ve imzalama için gerekli olan anahtarlar farklı algoritmalar ile bu anahtar ve geçen parametrelerden üretilir. IV için sunucunun gönderdiği parametrenin bir bölümü kullanılmaktadır. Anahtar için özel anahtarın PSK ile birleştirilip MD5 ile özet değerinin üretilmesi ile kullanılır. İmzalama için kullanılan anahtar özel anahtarın bir bölümünden alınmaktadır. Aynı algoritma 16. aşamada da kullanılarak sunucu tarafında aynı değerler üretilir.

10. Cihaz paket sıra numarası ve Parametreyi yine PSK ile şifreler. Ve sunucuya gönderilir.

11. Cihazın oturumundaki bilgiler denetlenir. Eğer uyumsuzluk varsa veya zaman aşımı varsa geriye NoSession değeri PSK ile şifreli olarak gönderilir.

12. Cihazın eğer durumu pasif veya çalıntı ise durum bilgisi PSK ile şifrelenerek cihaza gönderilir.

13. Cihaza ait oturum kaydı bulunamadı ise D-H anahtar değişimi 1. Adımdan itibaren tekrar başlatılır.

14. Cihaza ait oturum kaydı geçerli ve cihaza ait veri uygun ise veri işleme alınır. Bu aşamada cihazdan gelen veri PSK ile çözülür. Paket sıra numarası kontrol edilir. Eğer sıra numarası uygun ise (Paket tekrarı denetimi) bir sonraki işleme geçilir.

15. Cihaz çalıntı ise cihaz HardReset işlemini gerçekleştirir.

16. Parametreler kullanılarak özel anahtar üretilir ve geriye işlemin başarılı olduğuna dair bilgi PSK ile şifreli olarak gönderilir.

17. PSK ile veri çözülür eğer sonuç başarılı ise oturum açma işleminin başarılı olduğu şeklinde bir bilgi ile bu yordam işlemini tamamlar.

Yukarıda D-H anahtar değişimi için oluşturan mesaj tiplerinden StartDH(UniqueID), ServerParams, Response, paketleri PSK ile şifreli olarak iletilir. Burada D-H anahtarlarının üretilmesi sistemi yoran ve zaman alan işlemlerdir. Paket tekrarı yöntemiyle sistemin yorulmasını engellemek için ilk aşamada parametreler oturum üzerinde saklanmakta ikinci veya tekrar eden çağrılarda buradan parametreler alınarak otomatik gönderilmektedir.

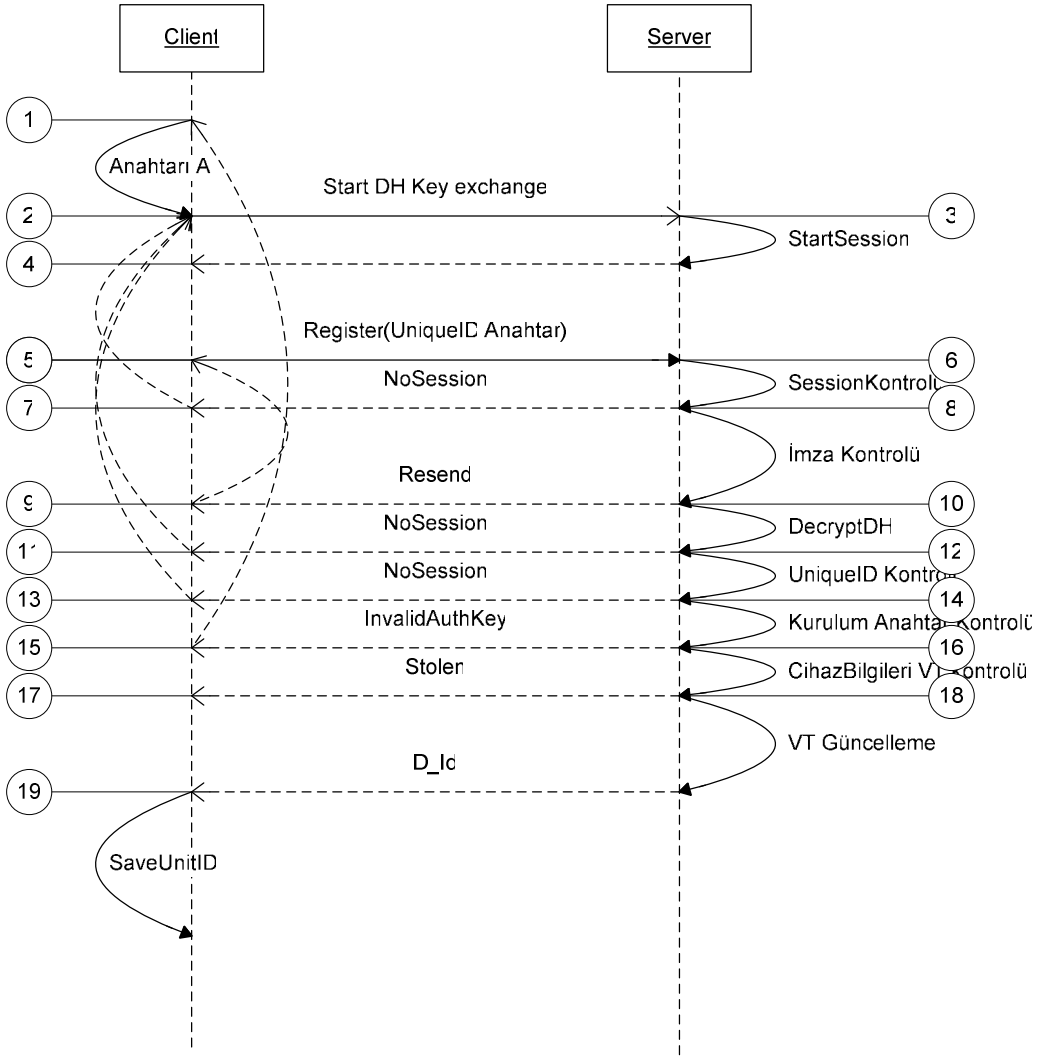
Aynı paketin tekrar gelmesi gerçekte bir saldırı olarak algılanmamalıdır. GPRS ağ yapısında bazen isteğin gönderilip cevabın alınamaması durumu olabilir. Bu istemci tarafta hata olarak karşımıza çıkacaktır. Kullanıcıya hata raporlandıktan sonra kullanıcının tekrar talebi doğrultusunda tekrar paket gönderilebilir. Bu durumda sunucu buna cevap vermelidir.

D-H anahtar değişimi sonrası istemci ve sunucu taraflarında oluşan 1024 bit veri (128 bayt) istemci tarafından gönderilen rastgele değer ile MD5 özetleme fonksiyonundan geçirilerek elde edilen 16 bayt veri oturum veri şifreleme gizli anahtarı (Rijndael anahtarı) olarak kullanılır. İstemcinin D-H anahtar değişiminde sunucu parametrelerine cevap olarak kendi D-H parametrelerini yolladığı StoreDH paketi içerisindeki istemci parametrelerinin ilk 16 baytı ise IV (başlangıç vektörü) olarak kullanılacaktır. D-H anahtar değişimi sonrası istemci ve sunucu tarafında oluşan 128 bayt verinin farklı bir kısmı da (20 bayt) HMAC-SHA1 özeti (imza) için kullanılacak ortak anahtar olarak belirlenir ve kullanılır.

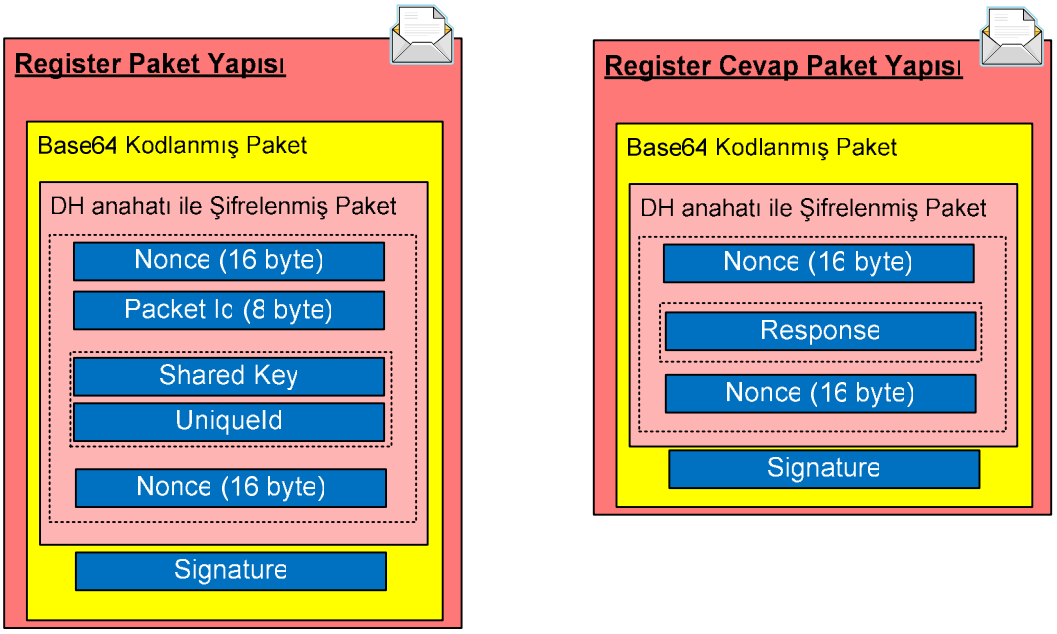
Oturum açma işleminde SessionID ve karşılıklı iletişim için kullanılan anahtar değerleri konfigürasyonda tutulur. Bu konfigürasyon diskte cihazın Unique ID değeri ile şifrelenerek saklanır. Bu konfigürasyon başka bir makineye taşınırsa işe yaramaz.

GetTime paketi, StartDH ve StoreDH talep ve cevap paketleri örnekleri EK1'de gösterilmiştir.

Mobil cihazda programın kurulup çalışır hale gelmesi aşağıdaki düzen diyagramında gösterilmiştir:



Şekil 4.22. İlk Kurulum (Cihazın sisteme tanıtılması) Düzen Şeması



Şekil 4.23. Register Paketi ve Register Cevap Paketi Yapıları

1. Arayüzden tanıtım anahtarı alınır.

2. Tanıtım işlemi öncesi güvenli oturum başlatılır. Bu aşamada Şekil 4.19'da yer alan işlem adımları gerçekleştirilir. Bu adımlar ve çağrı işlemi sembolik olarak tek adımda gösterilmiştir.

3. Oturum açılır.

4. Oturum bilgileri kaydedilir.

5. Tanıtım anahtarı ve cihaz Unique Id değeri D-H anahtar değişimi aşamasında belirlenmiş olan simetrik anahtar ile şifrelenir. Kayıt talep paketi oluşturularak gönderilir.

6. Eğer oturum zamanaşımına uğramış ise oturum anahtarları olmadığı için devam edilemez. Burada NoSession cevabı PSK ile şifreli olarak gönderilir.

7, 11 ve 13 aşamalarında, cihaz gelen cevabı önce D-H anahtarı ile çözmeye çalışır. Eğer çözemiyor ise PSK kullanarak çözmeyi dener. Cevap incelenerek işlemin ne olması gerektiğine karar verilir. Oturumun yeniden açılması, paketin tekrar gönderilmesi, hard reset, anahtarın tekrar istenmesi gibi işlemler gerçekleştirilir.

8. İmza kontrolünde eğer imza geçersiz ise cevap olarak Resend komutu gönderilir. Bu paketin yolda bozulduğu ve aynı paketin tekrar gönderilmesi gerekliliğini belirtir. İmzanın bozulmuş olduğu D-H ile şifrelenerek gönderilir.

9. Paket tekrar hazırlanarak gönderilmek üzere 5. Aşamaya döndürülür.

10. Verinin şifresi çözülerek paket açılır. Şifre çözmede bir problem olursa anahtarla ilgili bir sorun var demektir. NoSession PSK ile şifreli olarak gönderilir. Pakette bulunan paket sıra numarası eğer uygun değilse (paket tekrarı atağı) burada yine NoSession bilgisi PSK ile şifreli olarak geri gönderilir.

12. Bu aşamada eğer paketin içindeki UniqueID değeri ile oturumda kullanılan anahtar farklı ise D-H ile şifrelenmiş olarak NoSession cevabı gönderilir. Bu ilave bir kontroldür.

14. Tanıtım anahtarı ile gönderilen anahtar karşılaştırılır. Eğer farklılık varsa (kurulum anahtarı zaman aşımına uğramış ise) InvalidAuthKey cevabı D-H ile şifreli olarak geri döndürülür.

15. Tanıtım anahtarının tekrar alınması için arayüze dönülür.

16. Cihaz Unique Id ile veritabanı kontrolünde cihazın çalıntı olup olmama durumu kontrol edilir. Çalıntı ise cihaz HardReset mesajı D-H anahtarları ile şifreli olarak gönderilir.

17. Cevap hardreset ise cihaza HardReset komutu gönderilerek cihaz içindeki bilgiler silinir.

18. Veritabanında cihaz bilgileri güncellenir. Cevap olarak Cihazın veritabanı ID değeri gönderilir.

19. Cihazın ID değeri konfigürasyona kaydedilir. Ve tanıtım işlemi tamamlanmış olur.

Cihazın kurulumu tamamlandıktan sonra cihaz veri göndermeye başlayacaktır.

GPS üzerinden alınan verilerin gönderilmesinde aşağıdaki işlem adımları takip edilir:

- GPS eğer konum alamıyor ise GPS' ten gelen veri dikkate alınmaz.
- GPS' ten alınan konum verisi eğer konfigürasyonda tanımlanan rakamdan daha hassas değilse dikkate alınmaz.
- Eğer GPS' ten gelen veride enlem boylam bilgilerinden biri geçerli değilse dikkate alınmaz.

Bu veri elde edilen son geçerli konum bilgisi olarak ayrıca saklanır.

- Son konum bilgisi varsa ve son gönderilen konum ile gelen verideki konum arasında eğer konfigürasyonda belirtilen mesafeden fazla uzaklaşılmamış ise ve son gönderme zamanı ile verinin alındığı zaman arasında gönderme periyodundan daha az bir zaman geçmiş ise GPS verisi dikkate alınmaz.

Eğer bu koşulları geçerse bu veri gönderilir.

Bu durumda;

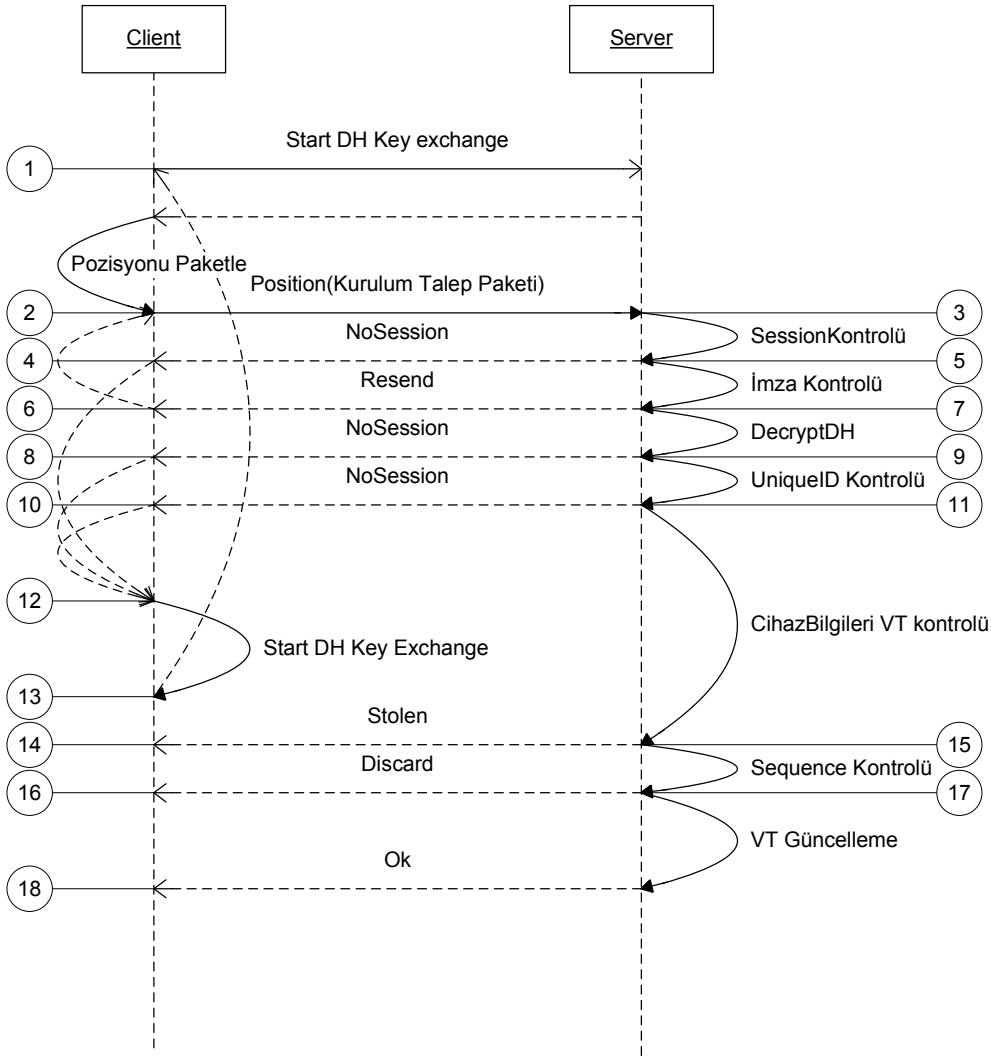
- Cihaz hareket halindeyse belli bir mesafe uzaklaştığında konum bilgisini gönderecektir. (Örneğin 50 mt)
- Cihaz durağan haldeyse belli bir periyot ile veri gönderecektir. (Örneğin 2 dakikada bir)

Gönderilecek veri ile beraber, son veri gönderme konumu ve saati kaydedilir. Veri aynı zamanda mobil cihazın hafızasına kaydedilir. Bu kayıt olası şarj problemi veya uygulamanın bir şekilde veri gönderemeden kapatılma olasılığına karşı alınmaktadır. Veri göndermesi başarılı olduğunda dosya silinir. Uygulama açıldığında konum bilgilerini bilgisayar hafızasından tekrar yükler ve gönderir.

Gönderilecek veriler bir kuyruğa atılarak belli periyotlar ile kontrol edilir. Kuyrukta veri varsa gönderilir. Eğer kuyrukta veri yoksa ve son gönderme zamanından o ana kadar herhangi bir GPS verisi alınamamış ise son geçerli konum bilgisi gönderilir. Bu şekliyle örneğin tünele giren araç nerden girdiyse giriş noktası bilgisi gönderilmiş olacaktır.

Register Paketi örneği be Register Cevap paketi örneği EK1'de gösterilmiştir.

Pozisyon bilgilerinin gönderilmesi işlemi aşağıdaki düzen diyagramında verilmiştir:



Şekil 4.24. Oturum güvenli haberleşme şeması

1. Konum bilgisi gönderme işleminden önce cihaz öncelikle sunucuda oturum açar.

2. GPS' ten gelen veriler paket haline getirilerek Cihaz Id, paket sıra numarası ile birlikte D-H anahtarı ile şifrelenir. Veri imzalanır ve pozisyon bilgisi olarak sunucuya gönderilir.

3. Eğer oturum zaman aşımına uğramış ise oturum anahtarları olmadığı için devam edilemez. Bir sonraki aşamaya geçmeden NoSession ile cevap verilerek geri dönülür. Burada NoSession cevabı PSK ile şifreli olarak gönderilir.

4. 8. ve 10. aşamalarda, cihaz gelen cevabı önce D-H anahtarı ile çözmeye çalışır. Eğer çözemiyor ise PSK kullanarak çözmeyi dener. Cevap incelenerek işlemin ne olması gerektiğine karar verilir. Cevabın değerlendirilmesi aşamasında oturumun yeniden açılması, paketin tekrar gönderilmesi, hard reset, anahtarın tekrar istenmesi gibi işlemler gerçekleştirilir.

5. İmza kontrolünde eğer imza geçersiz ise cevap (response) olarak Resend (paketi yeniden gönder) cevabı gönderilir. Bu paketin yolda bozulduğu ve aynı paketin tekrar gönderilmesi gerekliliğini belirtir. İmzanın bozulmuş olduğu bilgisi D-H ile şifrelenerek gönderilir.

6. Paket tekrar hazırlanarak gönderilmek üzere 5. Aşamaya döndürülür.

7. Verinin şifresi çözülerek paket açılır. Şifre çözmede bir problem olursa anahtar ile ilgili bir sorun var demektir. NoSession PSK ile şifreli olarak gönderilir. Pakette bulunan paket sıra numarası eğer uygun değilse (paket tekrarı saldırısı) burada yine NoSession bilgisi PSK ile şifreli olarak geri gönderilir.

9. Bu aşamada eğer paketin içindeki UniqueID değeri ile oturumda kullanılan anahtar farklı ise D-H ile şifrelenmiş olarak NoSession cevabı gönderilir. Bu ilave bir kontroldür.

11. Cihazla ilgili bilgiler veritabanından kontrol edilir. Eğer cihaz çalıntı ise DH ile şifrelenmiş olarak Stolen (çalıntı) bilgisi gönderilir.

12. ve 13. Cihazın gönderdiği veriler öncelikle DH anahtarları ile çözülmeye çalışılır. Eğer çözülemiyor ise PSK ile çözülmeye çalışılır. Çözülen veriye göre yeniden oturum açma, paketin tekrar gönderilmesi gibi işlemler gerçekleştirilir.

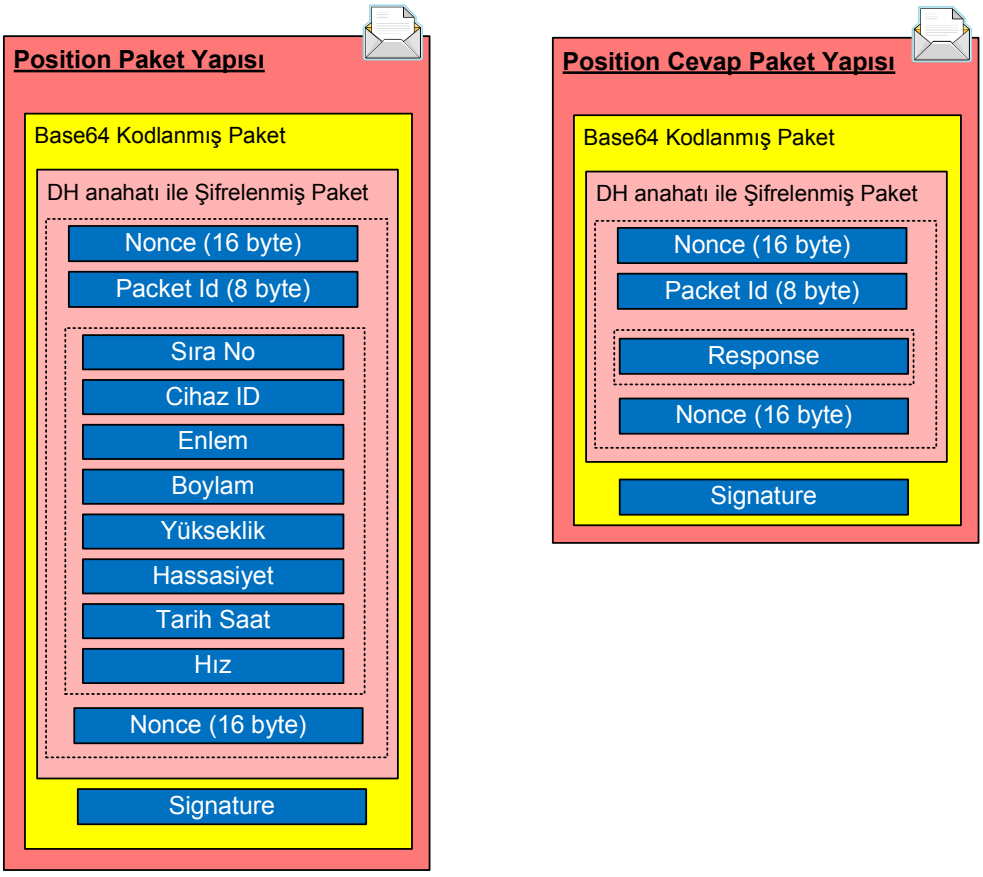
14. Cihaza gelen cevap eğer Stolen ise Cihaz hardreset ile içindeki bilgileri temizler.

15. Cihazdan gelen paket sıra numarası eğer son paket sıra numarasından küçük ise dikkate alınmaz. Ancak eğer bu paket cihazın daha önce gönderdikleri arasında yoksa konum geçmişi için kaydedilir.

16. Eğer paket sıra numarası geçerli değilse yeni “Discard” mesajı gelir. Bu bir yerde mesajın tekrar gönderildiği anlamına gelir. Cihaz açısından veri göndere başarılıdır.

17. Cihazın gönderdiği konum bilgisi veritabanında güncellenir. Ve OK bilgisi DH anahtarı ile şifrelenmiş olarak gönderilir.

18. İşlemin başarılı olduğu bilgisi alınır.



Şekil 4.25. Konum ve Konum cevap Paket Yapıları

Position (Konum) paketi örneği ve Position cevap paketi örneği EK1’de gösterilmiştir.

Çizelge 4.12’de haberleşmede kullanılan temel paketlerin bayt cinsinden boyut bilgileri gösterilmiştir. Değerler kullanılan mobil cihazın tekil numarası (Unique Id) boyutuna bağlı olarak bir miktar farklılık gösterebilir.

Çizelge 4.3. Paket Boyutları

İşlem	Request (İstek) (bayt)		Response (Cevap) (bayt)	
	Toplam	Body (Paket Gövdesi)	Toplam	Body (Paket Gövdesi)
GetTime	243	0	231	12
StartDH	373	128	796	576
StoreDH	481	236	283	64
Register	424	156	283	64
Position	424	156	311	92
NoSession Response			283	64

4.3.2. RSA anahtar kullanımı (2.Senaryo)

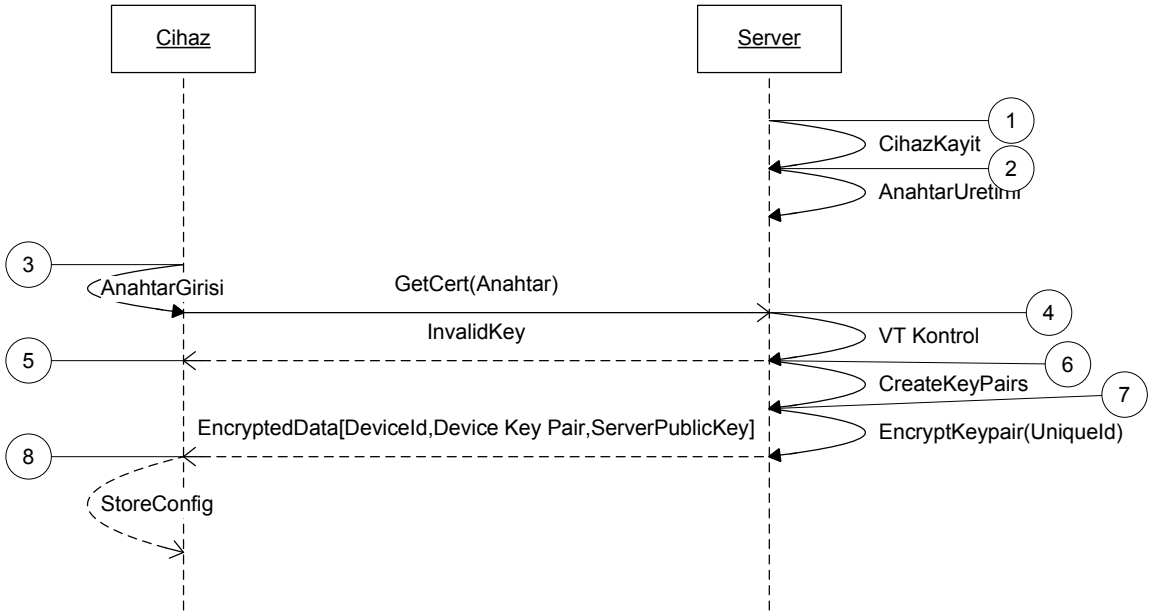
- RSA anahtar kullanımı ile sisteme tanıtma işlemi

RSA anahtarlar kullanarak güvenli iletişime geçmede cihaz arayüzünden okunan cihaz UniqueId değeri bilgisayardan arayüz aracılığı ile girilir. Buna karşılık bir kullanımlık zamanaşımı olan bir tanıtım anahtarı üretilerek, UniqueID + tanıtım anahtarı + son geçerli olduğu tarih bilgisi veritabanına kaydedilir. Tanıtım anahtarı Bilgisayar arayüzünden kullanıcıya gösterilir. Cihaz arayüzünden tanıtım anahtarı bilgisi girilir. Tanıtım anahtarının girilmesi ile birlikte cihaz sunucudan bu anahtar ile cevap beklemektedir. Sunucu tanıtım anahtarını alır bu anahtarla zaman aşımına uğramamış kayıt olup olmadığını kontrol eder. Yoksa boş dönecektir. Eğer bilgi varsa cihaza RSA anahtar çifti yaratır ve veritabanında cihazların tutulduğu tabloya bu cihaza ait bir kayıt oluşturur. Bununla birlikte tanıtım anahtarının olduğu tablodan ilgili kayıt silinir. Geriye cihaza ait kayıt numarası, cihaz anahtar çifti ve sunucu genel anahtarı cihazın UniqueID değeri ile şifrelenerek döndürülür. Süresi dolan kayıtlar da otomatik olarak silinir.

Olası tehdit senaryoları dikkate alındığında ilk olarak akla, geri dönen paketten verinin çözülebileceği düşünülebilir. Burada her cihaz için farklı bir şifre kullanılacağı ve şifrenin cihazın Unique Id değeri olduğu düşünülürse çözme imkanı olmadığı görülecektir.

Paket tekrarı atağında, paketin yeniden gelmesi sunucu için anlamlı olmayacaktır. Bu paket ikinci kez istendiğinde kayıt silinmiş olduğundan erişim imkanı olmayacaktır.

Denial of Service (servis durdurma) atağı için sunucuya saldırı söz konusu olduğunda atak veritabanına yük olarak yansıyacaktır. Burada da en fazla 5-10 kaydın bulunduğu bir tablodan sorgulama söz konusu olduğundan sistemin cevap verememe gibi bir durumu söz konusu olmayacaktır.



Şekil 4.26. RSA anahtar ile sisteme tanıtma akış şeması

1. Cihazdan okunan Unique ID değeri Monitör uygulaması aracılığı ile sisteme kaydedilir.

2. Bu cihaz için tek kullanımlık ve belli bir süre geçerli olan bir anahtar üretilir.

3. Bu anahtar cihaza arayüz aracılığıyla girilir. Cihaz bu anahtarı kullanarak talepte bulunur.

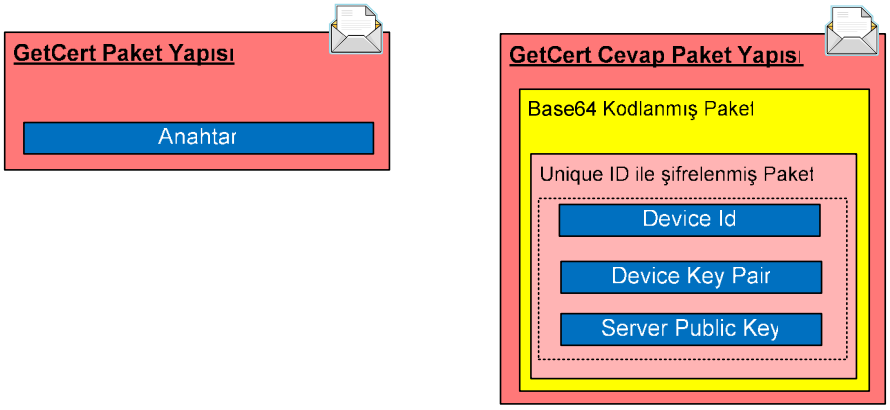
4. Gelen anahtar incelenerek geçerli olup olmadığına bakılır. Geçerli değilse geriye InvalidKey mesajı döndürülür.

5. Anahtarın geçersiz olduğu mesajı gelirse kullanıcı uyarılır.

6. Mesaj geçerli ise cihaz için asimetrik RSA anahtar çiftleri üretilir. Bu asimetrik anahtar çiftleri veritabanına kaydedilir.

7. Üretilen asimetrik anahtar çiftleri cihazın daha önce kaydedilen UniqueID değeriyle şifrelenir. Ve paketlenip cihaza cevap olarak gönderilir.

8. Cihazın Unique ID değeri network üzerinde taşınmadığından ve UniqueID değerinin kopyalanması söz konusu olmadığından, uygulamanın çalıştığı cihaz dışında bir yerden anahtar çözülemeyeceği için RSA sertifikalar güvenli olarak cihaza iletilmiş olacaktır. Bu aşamada gelen, cihaz veritabanı referans numarası (Device Id) bilgisi, Cihaz Anahtar Çifti (cihazın özel ve genel anahtarı) ve sunucunun Public Key (genel anahtar) değeri bilgileri cihazın konfigürasyonuna kaydedilir.



Şekil 4.27. GetCert ve GetCert Cevabı Paket yapıları

- RSA anahtar kullanımı ile güvenli iletişime geçme işlemi

Burada şifrelemede RSA kullanılmaktadır. Anahtarlar 1024 bit RSA anahtarlarıdır. Giden pakette şifreleme sunucu genel anahtarı ile imzalama cihazın özel anahtarı ile yapılmaktadır. İstek gönderilirken, cihaz ID değeri açık olarak gönderilmektedir. Aksi takdirde sunucunun imzayı denetlemesi söz konusu olmayacaktır. İmza cihaz ID bilgisini de kapsadığı için iteratif deneme söz konusu olamayacaktır. Paket ID değeri paket tekrarı ataklarından korunmak için her istek için arttırılan bir sayaç değeridir. Cevapta da gelen paketin bu isteğe gönderilen veri olduğunu garantilenecektir. Olası paket tekrarı durumunda sunucu tarafında paket iptal edilir. Cihaz tarafında gerçekte herhangi bir soket açık olarak beklememektedir. İstek yapılırken soket açılır cevapla birlikte kapatılır. İsteğin yapıldığı client port

numarası da her zaman aynı port numarası olmayıp değişebilmektedir. Bu nedenle olası tek durum bir istek yapıldığında önceki paketin cevabının gelmesi durumudur. Burada paket ID kontrolü ile bu problem ortadan kalmış olacaktır.

Anahtar değişimi iş akışı daha önceki PSK ile yapılan iş akışına göre farklılık göstermektedir. PSK ile yapılan işlemlerde PSK kullanımı cihazın kimliğini onaylamamaktadır. Bu nedenle ilk çağrıda güvenli iletişim tamamlanmadan cihazın kontrolü söz konusu değildir. Sertifika kullanımında bu durum ortadan kalkmaktadır.

Sertifika kullanılarak D-H anahtar değişimi yapılırken gönderilen paketin başında bir Cihaz ID bilgisi bulunacaktır. Bu bilgi olmadan ve geçerliliği denetlenmeden güvenli iletişime geçilmemektedir. Bu denetim sonucu çıkan bazı sonuçların cihaza bildirilmesinde dikkat edilmesi gereken durumlar ortaya çıkmaktadır. Örneğin cihazın kayıtlı olmaması durumunda geri bildirimin açık gitmesi gerekir.

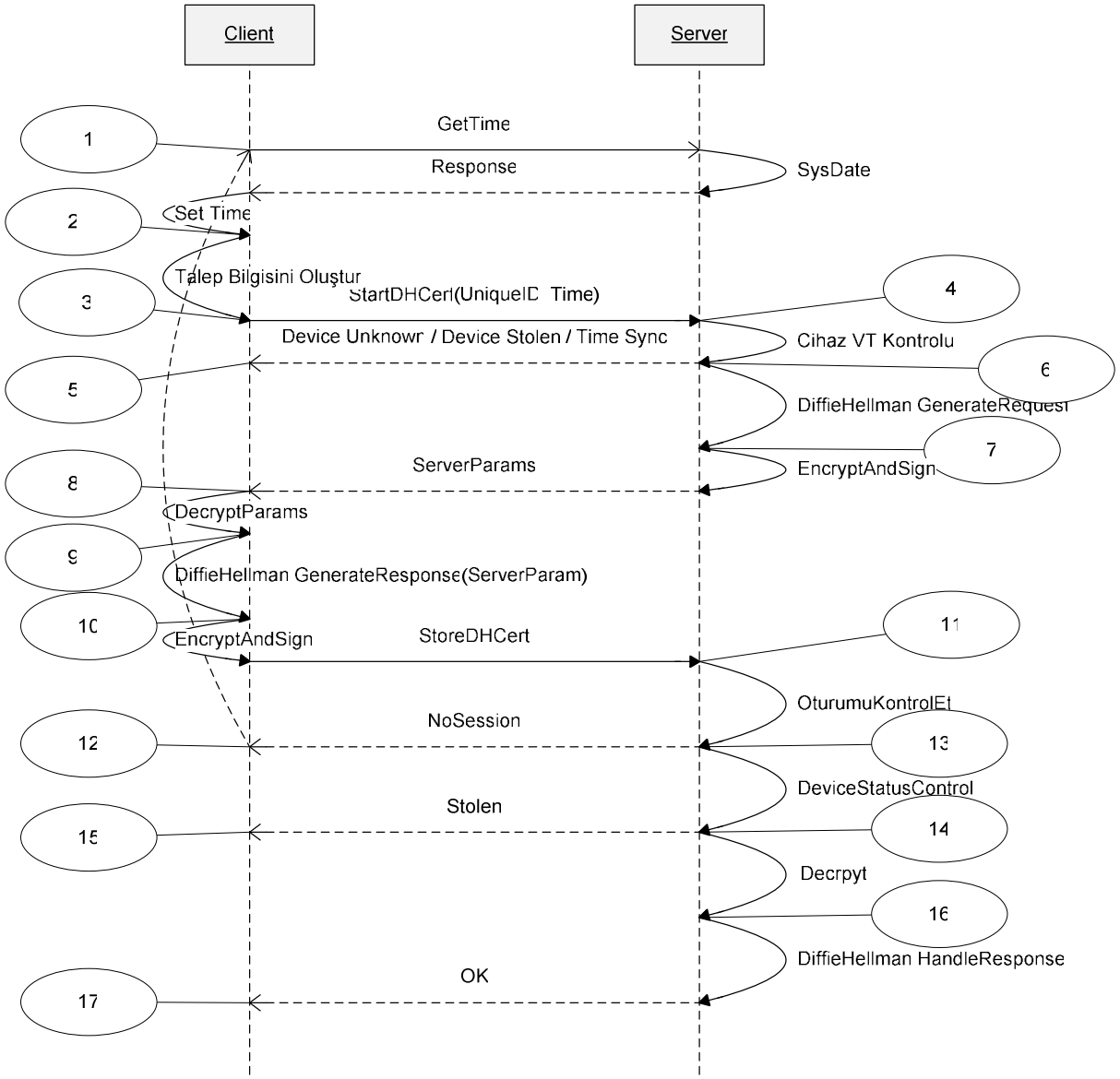
Bunun için paket tipleri tanımlanmıştır. Paket tipi, dönen verinin açık, sunucu özel anahtarı ile şifrelendiğini veya cihaz genel anahtarı ile şifrelendiğini belirtir.

Paket tipi “1” cihazın kayıtlı olmadığı, imzanın geçerli olmadığı veya imzanın bozuk olduğunu belirtmek için kullanılır. Bu durumda cihazdaki anahtarlar geçerli olmayabilir. Bu nedenle cevap açık olarak gönderilir. Cihaz tanınmıyor veya cihaz anahtarı ile üretilen imzası geçersiz ise tekrar tanıtma işleminin yapılması gerekir. Bu nedenle oturum başlatma devam edemez. İmza bozuk ise paket tekrar edilir.

Paket tipi “2” cihazın gönderdiği pakette paket sıra numarasında sorun olması, zaman aşımı söz konusu olması ve cihazın çalıntı olması durumu için veri gönderilmesi işlemlerinde kullanılır.

Paket tip “3” ise D-H parametrelerini içerir.

Cihazda D-H parametreleri alındıktan sonra önce sunucu tarafında da bu anahtarın üretilmesi için gerekli parametre ve peşinden D-H anahtarları üretilir. D-H anahtarları konfigürasyona kaydedilir. Parametre ise sunucuya yine StoreDHCert komutu ile gönderilir. Sonuç paket tipi paket tip 1 veya paket tip 2 olabilir. Paket tip içindeki cevap eğer Ok ise işlem tamamlanmış demektir.



Şekil 4.28. RSA anahtar ile güvenli iletişime geçme akış şeması

1. İlk işlem sunucu sistem saatinin okunmasıdır.

2. Sunucu sistem saati ile cihaz saati güncellenir.

3. Device Unique Id değeri sistem saati ile birlikte paket yapılır, bu paket sunucunun genel anahtarı ile şifrelenir. cihazın özel anahtarı ile imzalanır Pakete cihazın Device Id bilgisi (veritabanı kayıt no) eklenir. Bu şekilde talep yapılır.

4. Cihaz ID değeri okunur. Veritabanından bilgileri çekilir. Eğer cihaz tanınmıyor ise açık olarak DeviceUnknown mesajı döner. Eğer cihaz tanınıyor ise imza denetimi yapılır. Eğer imza geçerli değil ise geriye imza geçersiz uyarısı

dönecektir. İmza geçerli olmasına rağmen cihaz çalıntı olarak görünüyor ise cihaza Cihazın genel anahtarı ile şifreli, sunucunun özel anahtarı ile imzalı olarak Stolen bilgisi geri döndürülür. Eğer imza geçerli ancak paket zaman aşımına uğramış ise yine cihazın özel genel anahtarı ile şifreli ve sunucu özel anahtarı ile imzalı olarak SyncTime mesajı gönderilir.

5. Gelen paketin içeriği sunucu genel anahtarı kullanılarak imza kontrolü yapılır. Bundan sonraki aşamada mesajın başlığında eğer 1 varsa açık, 2 veya 3 ise cihaz genel anahtarı ile şifreli demektir. “1” durumu için dönebilecek değerler DeviceUnknown, InvalidSignature (geçersiz imza) ve SignatureDamaged (imza hasarlı) olabilir. DeviceUnknown ve InvalidSignature yeniden tanıtım gerektirir. SignatureDamaged gelmesi durumunda paket yeniden gönderilir. “2” durumu için Stolen, SyncTime komutlarına yönelik olarak ya cihaz’a hard reset atılır veya saat güncellemesi yapılır. Paket tip “3” ise cevap D-H parametreleridir.

6. D-H anahtar değişimi için gerekli parametreler üretilir.

7. Üretilen parametreler cihaz genel anahtarı ile şifrelenir. Sunucu özel anahtarı ile imzalanır ve cihaza gönderilir.

8. Cihaz İmzayı denetler, veriyi özel anahtarıyla açar.

9. Server parametrelerini kullanarak D-H özel anahtarını üretir. Bu anahtarı ürettikten sonra sunucuya aynı özel anahtarı üretmesi için gerekli parametre hazırlanır. Burada üretilen D-H özel anahtarı doğrudan doğruya kullanılmaz. İşlem aşamasında kullanılan Rijndael simetrik anahtarlama için gerekli olan IV, anahtar ve İmzalama için gerekli olan anahtarlar farklı algoritmalar ile bu anahtar ve geçen parametrelerden üretilir. IV için sunucunun gönderdiği parametrenin bir bölümü kullanılmaktadır. Anahtar için özel anahtarın PSK ile birleştirilip MD5 özetleme fonksiyonundan geçirilerek elde edilen değer kullanılır. İmzalama için kullanılan anahtar özel anahtarın bir bölümünden alınmaktadır. Aynı algoritma 16. Aşamada da kullanılarak sunucu tarafında aynı değerler üretilir.

10. Cihaz paket sıra numarası ve Parametreyi yine sunucu genel anahtarı ile şifreler ve cihaz özel anahtarı ile imzalar ve sunucuya gönderilir. Üretilen anahtarları konfigürasyona kaydeder.

11. Cihazın oturumundaki bilgiler denetlenir. Eğer uyumsuzluk varsa veya zaman aşımı varsa geriye NoSession değeri gönderilir.

12. Eğer Nosession değeri geliyor ise anahtar değişim işlemi tekrar başlatılır.

13. Cihaz eğer çalıntı ise geri bildirimde Stolen bilgisi gönderilir.

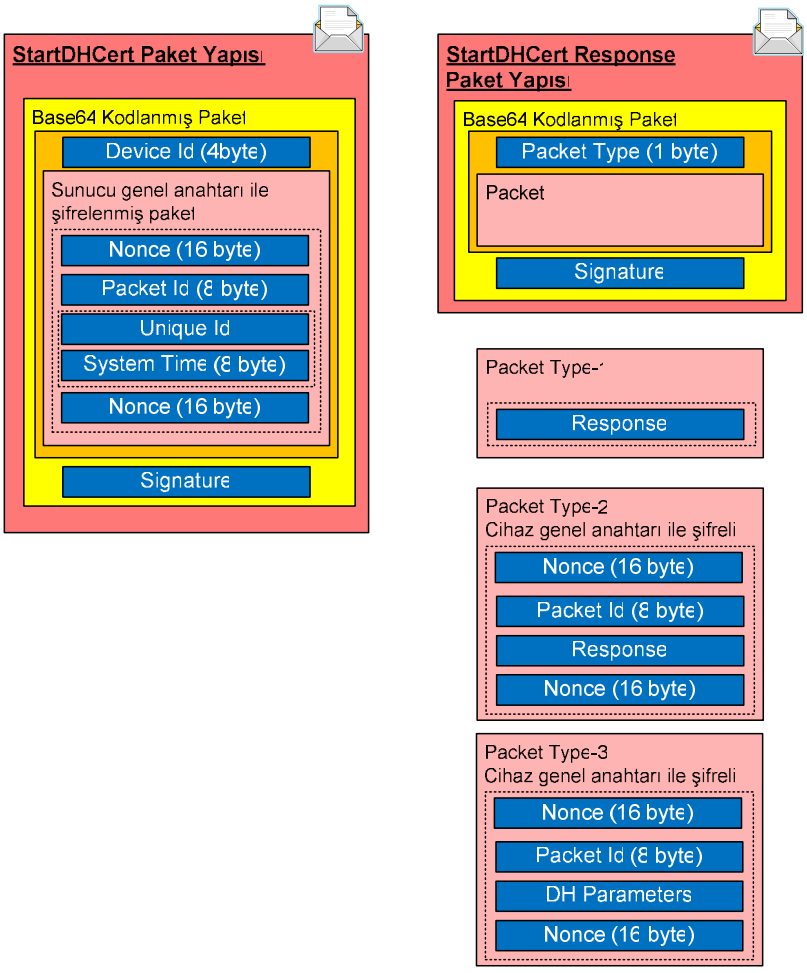
14. Parametreler sunucu özel anahtarı ile çözülür.

15. Cihaz çalıntı mesajı geldiğinde cihaz hardreset atarak verileri temizler.

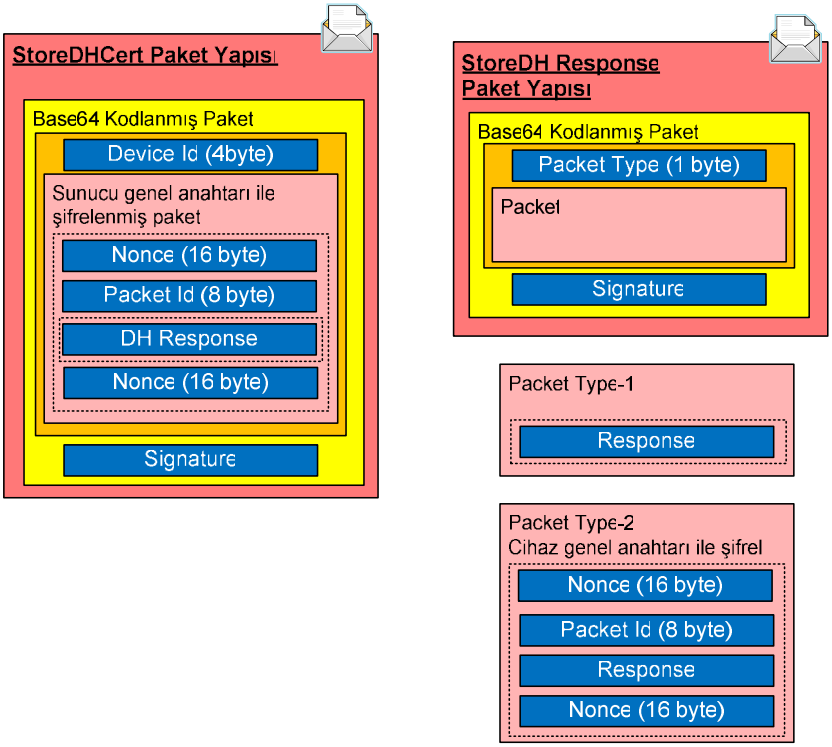
16. D-H özel anahtarı üretil oturum açma işleminin başarılı olduğu şeklinde bir bilgi ile bu prosedür işlemini tamamlar.

17. Özel anahtarın üretildiğine dair mesajla işlemin başarıyla tamamlanır.

Paket yapıları Şekil 4.29 ve Şekil 4.30'da gösterilmiştir.



Şekil 4.29. StartDHCert talep ve cevap paket yapıları



Şekil 4.30. StoreDHCert talep ve cevap paket yapıları

RSA algoritmasının kullanıldığı paket boyutları Çizelge 4.4.'de gösterilmiştir.

Çizelge 4.4. Paket Boyutları (RSA kullanılan paketler, 2.senaryo)

İşlem	Request (İstek) (bayt)		Response (Cevap) (bayt)	
	Toplam	Body	Toplam	Body
GetCert	290	24	1781	1560
StartDHCert	618	348	1076	856
StoreDHCert	791	520	564	344

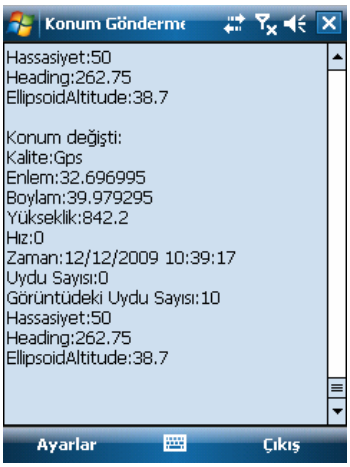
4.4. İstemci Yazılımı (Mobil cihaz)

PDA cihazda çalışmakta olan yazılım 3 ana ekrandan oluşmaktadır.

- Durum ekranı
- İlk Kurulum ekranı
- Konfigürasyon ekranı

4.4.1. Durum Ekranı

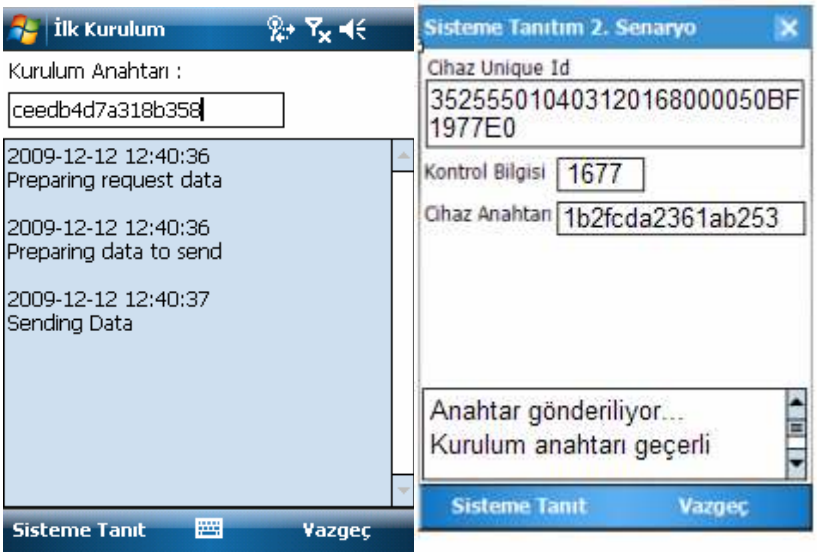
PDA cihazda çalışmakta olan yazılımda durum ekranı GPS bilgileri, veri gönderme durumu gibi bilgiler kayar şekilde gösterilmekte olduğu ana ekrandır.



Şekil 4.31. Mobil Cihaz Durum Ekranı

Bu ekrandan, ekranda sağ aşağıda bulunmakta olan "Çıkış" butonu ile çıkılmaktadır. "Ayarlar" kısmında iki alt menü bulunmaktadır. Birincisi "Sisteme Tanıtma" diğeri ise "Konfigürasyon" dur.

4.4.2. Sisteme Tanıtım Ekranı



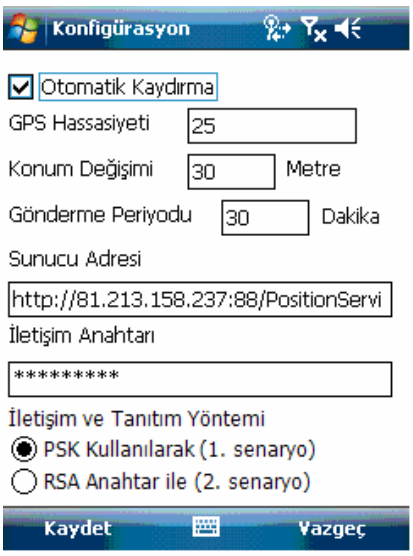
Şekil 4.32. Mobil cihaz sisteme tanıtım ekranları (PSK ve RSA anahtar ile)

Sisteme tanıtım ekranında, sunucu yazılımı tarafında oluşturulmuş olan özel kurulum anahtarı "Kurulum anahtarı" sekmesine doğru olarak girilmelidir. Ardından "Sisteme Tanıt" düğmesi tıklanarak Bölüm 4.3.1'de anlatılmış olan ilk kurulum işlemleri gerçekleştirilir. "Vazgeç" düğmesi ile işlem iptal edilebilir. RSA anahtar kullanılması durumunda ise, cihaz Unique Id değeri girilir. Kontrol bilgisi kısmında

cihaz Unique Id'sinin ondalık değerlerinin ayrı ondalık toplamı ekranda gözükmektedir. Bu aşamada sunucu tarafında da üretilen cihaz anahtarı, cihaz anahtarı parametresi olarak girilir ve bu değerin sunucuya gönderilmesi için "Sisteme Tanıt" düğmesi tıklanarak Bölüm 4.3.2'de anlatılmış olan sisteme tanıtım işlemleri gerçekleştirilir.

4.4.3. Düzenleme Ekranı

Bu ekranda cihazın gerekli düzenleme parametreleri girilebilmektedir. Şekil 4.30'da düzenleme ekranı gösterilmektedir.



Şekil 4.33. Mobil Cihaz Düzenleme Ekranı

- Otomatik kaydırma: Otomatik kaydırma, ana ekrana yeni veri geldikçe otomatik olarak en alt satıra kayma özelliğidir.

- GPS hassasiyeti: Eğer GPS hassasiyeti bu rakamdan daha büyük ise veri iptal edilir. Kötü noktalarda (GPS alıcısının uydular ile direk görüş sağlayamadığı durum) bu hassasiyet değeri artabilir. Bu durumda cihaz veri gönderemeyebilir.

- Konum Değişimi: Eğer en son gönderilen konum değeri ile yeni alınan konum arasında bu sekmede metre cinsinden girilmiş olan değerden daha fazla bir mesafe kat edilmiş ise veri gönderilecek demektir. Bu cihazın hareketli olduğunu gösterir. Ancak bu rakamı birkaç metre gibi

küçük değerlerde tutmamak gerekir. Bazen cihaz GPS hassasiyetindeki problemler nedeniyle hareketli olmasa bile hareketliymiş gibi görünebilir.

- Gönderme Periyodu: Eğer cihaz hareketli değilse, konum değiştirmiyorsa normalde cihazın veri göndermemesi gerekir. Ancak uzun süre hareketsiz kaldığında cihazın hali hazırda çalışır durumda olduğunu belirlemek için en son veri gönderme zamanından bu sekmede dakika cinsinden girilmiş olan süre boyunca hiç veri gönderilmemiş ise veri gönderir.

- Sunucu Adresi: Mobil cihazın verileri göndereceği sunucunun IP adresidir.

- İletişim Anahtarı: Cihaz ile sunucu arasındaki ilk iletişimi açmak için kullanılan önceden karşılıklı olarak paylaşılmış olan gizli anahtardır. Bu değer sunucu ile farklılık gösterdiğinde asla iletişim kurulamaz.

- İletişim ve tanıtım yöntemi: Cihazların ilk tanıtımı ve oturum açma safhalarında paketlerin şifrelenmesi için kullanılan metodun belirlendiği kısımdır.

4.4.4. İstemci Yazılımı Sınıfları

İstemci yazılımı için kullanılan temel sınıflar:

- **LogApi:** Mesajların program ekranına düşürülmesi için gerekli sınıftır.

- **GPSQueue:** Cihazın tespit ettiği ve göndermek üzere belirlediği konum bilgileri buraya kaydedilir. Böylelikle GPS modülüne ait thread (iş parçacığı) serbest kalabilir. Diğer tarafta veri gönderme fonksiyonu Queue (Kuyruk) içinde kayıt bulursa bu veriyi otomatik olarak göndermeye çalışır.

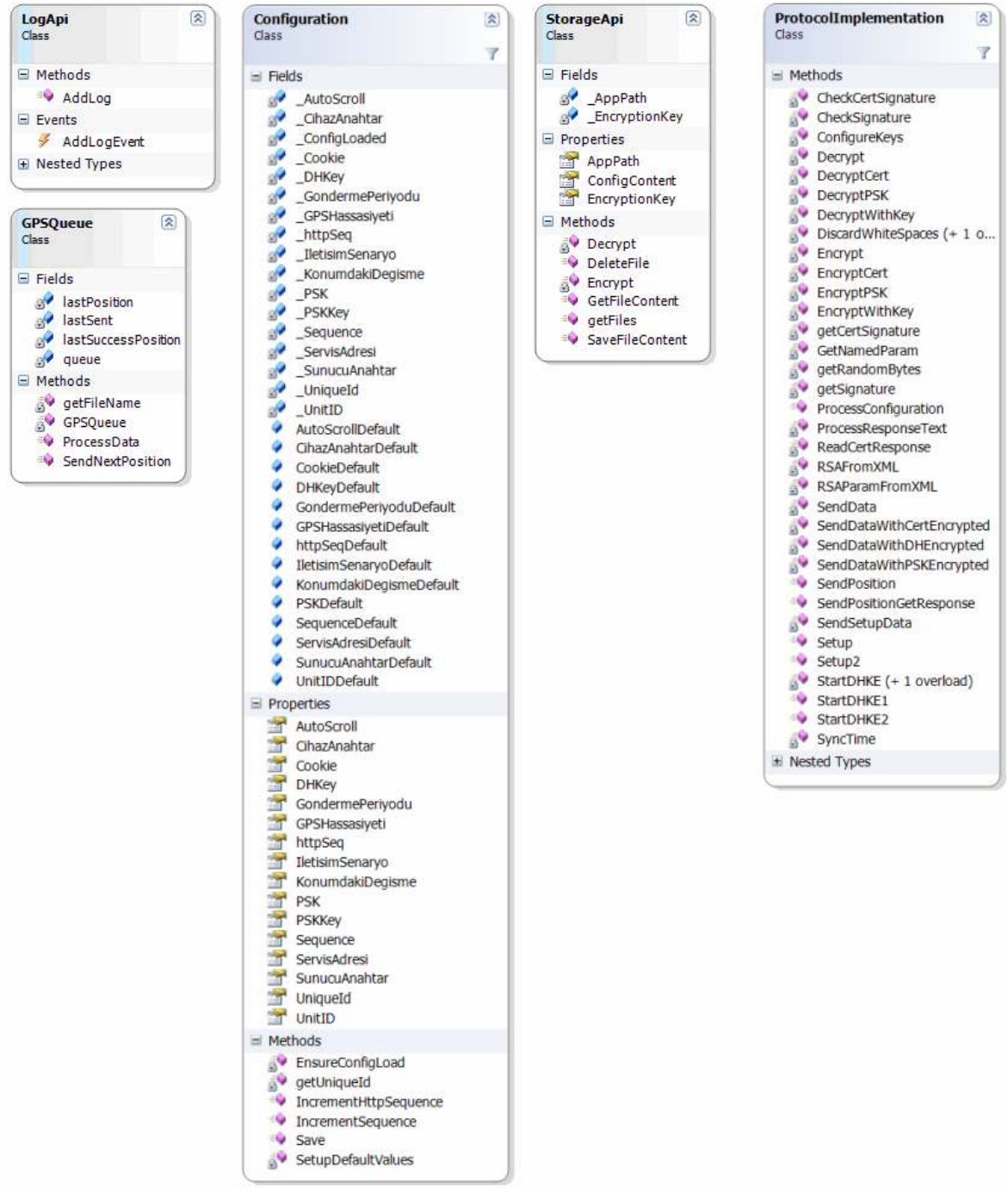
- **Configuration:** Cihaza ait konfigürasyon bilgileri burada tutulmaktadır. Konfigürasyon arayüzünden alınan veriler buraya yazılır. Diğer modüller buradan konfigürasyon bilgilerine erişirler.

- **Storage API:** Diskten okuma ve yazma işlemleri bu sınıf tarafından yönetilir. Diske yazma ve okuma işlemleri şifreli yapılır. Okuma yazma işlem

için bu sınıfın api fonksiyonları kullanılır.

- **ProtocolImplementation:** iletişimle ilgili her türlü işlem burada gerçekleştirilir. Kurulum, veri gönderme, oturum açma gibi işlemler bu sınıf tarafından yönetilir.

İstemci yazılımı için kullanılan temel sınıflar Şekil 4.34'te gösterilmiştir.

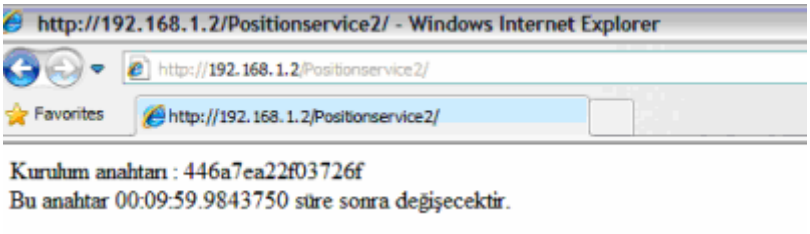


Şekil 4.34. İstemci Yazılımı Sınıfları

4.5. Web servis yazılımı (Sunucu)

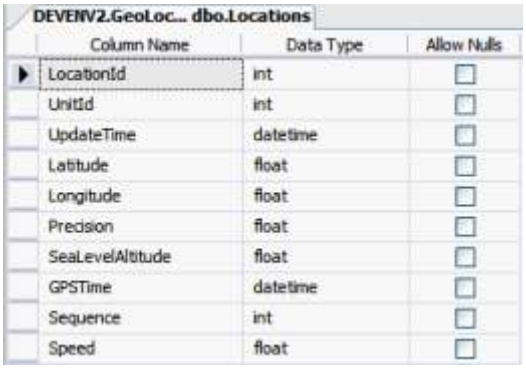
4.5.1. Ekran görüntüleri ve açıklamalar

Mobil cihazlar ile iletişimi gerçekleştiren yazılım, IIS arkasında çalışmaktadır. Cihazların kurulması için gerekli doğrulama anahtarının oluşturulması ve gelen konum bilgilerinin veritabanına aktarılmasından sorumludur.



Şekil 4.35. Sunucu tarafı kurulum anahtarı ekran gösterimi

Çizelge 4.5. Veritabanı konum kayıtları parametreleri



	LocationId	UnitId	UpdateTime	Latitude	Longitude	Precision	SeaLevelAltitude	GPSTime	Sequence	Speed
407	1012	2	2009-09-25 23:33:14	39,9595902166	32,79717265	4,19999	883	2009-09-25 20:33:09	7358	1,3999
408	1013	2	2009-09-25 23:33:27	39,95938775	32,796950283	3,90000	879,5	2009-09-25 20:33:22	7359	1,7999
409	1014	2	2009-09-25 23:33:40	39,9592723833	32,7969569	8,39999	880,5	2009-09-25 20:33:35	7360	1,7000
410	1015	2	2009-09-25 23:33:53	39,9591788833	32,79700745	4	884	2009-09-25 20:33:48	7361	1,7000
411	1016	2	2009-09-25 23:34:06	39,9590889	32,79705975	5,09999	884	2009-09-25 20:34:01	7362	1
412	1017	2	2009-09-25 23:34:19	39,9589395	32,796988583	4,80000	884	2009-09-25 20:34:14	7363	1,2999
413	1018	2	2009-09-25 23:34:33	39,9588332166	32,797018933	7,80000	885	2009-09-25 20:34:27	7364	1,2999
414	1019	2	2009-09-25 23:34:45	39,95876355	32,79709495	4,69999	882,5	2009-09-25 20:34:40	7365	1,8999
415	1020	2	2009-09-25 23:34:59	39,9586284333	32,797072316	4,90000	879	2009-09-25 20:34:53	7366	1,5
416	1021	2	2009-09-25 23:35:12	39,9585687166	32,797211716	10	878,5	2009-09-25 20:35:06	7367	2
417	1022	2	2009-09-25 23:35:22	39,9581969833	32,796559516	4,30000	869	2009-09-25 20:35:18	7368	1,2000
418	1023	2	2009-09-25 23:35:38	39,9581205333	32,7965477	5,40000	869	2009-09-25 20:35:26	7369	1,7000

Şekil 4.36. Veritabanı konum kayıtları ekranı

Çizelge 4.6. Veritabanı cihaz kayıtları parametreleri

DEVENV2.GeoLocations - dbo.Unit

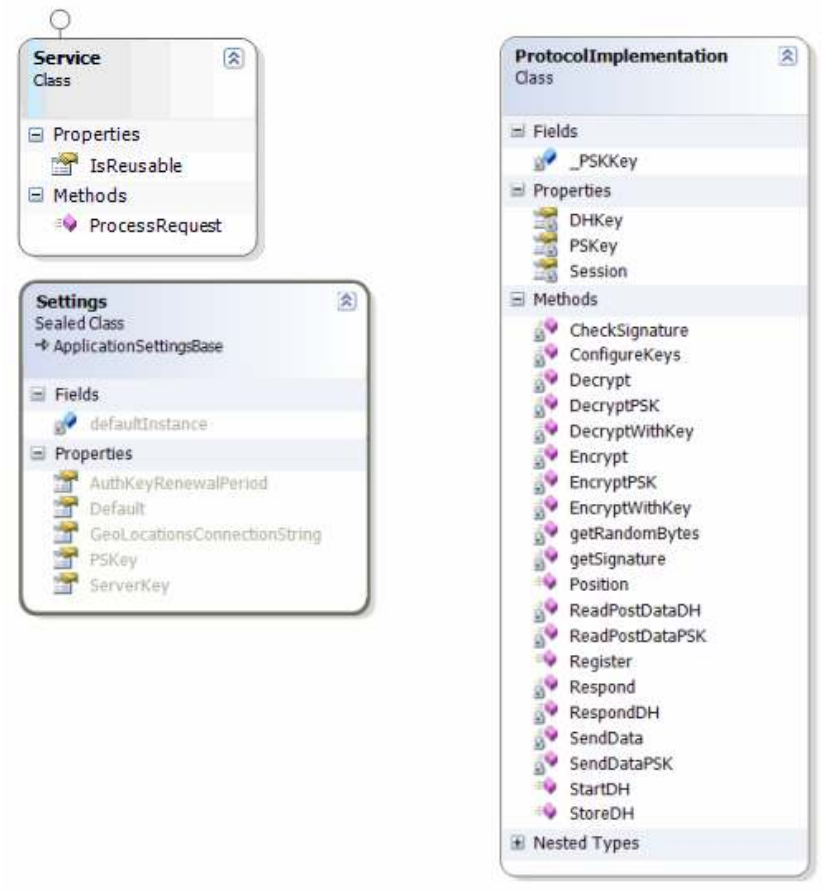
Column Name	Data Type	Allow Nulls
UnitId	int	☐
UniqueId	varchar(512)	☐
DeviceKeys	ntext	☑
UpdateTime	datetime	☑
Precision	float	☑
Latitude	float	☑
Longitude	float	☑
SeaLevelAltitude	float	☑
GPSTime	datetime	☑
Sequence	float	☑
DeviceStatus	smallint	☐
Speed	float	☑

UnitId	UniqueId	UpdateTime	Precision	Latitude	Longitude	SeaLevelAltitude	GPSTime	Sequence	DeviceStatus
1	50006F0063006B006500740050004 3...	2009-09-14 05:16:22.723	2,900000	40,536993133	33,608399316	732	2009-09-14 02:16:33	9	1
2	50006F0063006B006500740050004 3...	2009-09-30 16:11:11.970	8,5	39,909844466	32,804209516	913	2009-09-30 12:50:10	7506	1
3	3525550104031201680000 50BF1977	2009-12-20 15:36:55.627	1,200000	39,96963	32,613531666	796,200012207(	2009-12-20 13:35:22	853	0

Şekil 4.37. Veritabanı cihaz kayıtları

4.5.2. Web servis (sunucu) yazılımı sınıfları

İstemci yazılımı için kullanılan temel sınıflar Şekil 4.38'de gösterilmiştir.



Şekil 4.38. Web Servis (Server) Yazılımı Sınıfları

- Servis sınıfı: Bu sınıf sunucudan dışarı verilen servislerin sunulduğu sınıftır. Bir httpHandler'dan türemiştir.

- Settings: Bu sınıf gerekli konfigürasyon parametrelerinin tutulduğu sınıftır.

- ProtocolImplementation: Mobil cihazlar ile olan tüm iletişimin kontrol edildiği sınıftır.

4.6. Harita izleme ve Yönetim Yazılımı

Cihazların mevcut konumlarının harita üzerinden izlenebildiği, gerektiğinde cihazların sistem dışı bırakılabildiği bir uygulamadır. Kurulum ve diğer işlemler bu modül üzerinden yapılabilmektedir.

Sisteme tanıtılmış cihazların konumlarının takibi ve işlevselliği (aktif, pasif veya çalıntı) ile ilgili işlemlerin yapılabildiği bir Windows uygulaması gerçekleştirilmiştir. Bu uygulama Veritabanına yakın bir noktada (LAN) çalışmak zorundadır. Bu uygulama ile çevrimdışı harita üzerinde cihazlar görüntülenir. Cihazın hareketliliği neredeyse anlık olarak takip edilir. Ayrıca "GPS tarihçesi" nden cihazın geçmiş konum hareketleri takip edilebilir. Cihaz konumları ve "GPS tarihçesi" burada katman olarak tanımlanmıştır. Cihazların anlık olarak bulundukları yer bir harita katmanıdır. Bu katmanda yer alan objeler harita üzerinde bir ikon ile (yıldız, kare vb) ile işaretlenir. GPS tarihçesinde de cihaz bilgisi, tarih aralığı gibi bilgiler kullanılarak belli bir cihazın geçtiği yol yine çizgiler ve ikonlar ile işaretlenir. Katmanlar birden fazla kez yüklenebilir. Örneğin GPS konumu farklı cihazlar ile farklı tarihler ile üst üste bindirilebilir.

Burada önemli bir kısıt cihazların sembolleri üzerinde yapılan fare hareketlerinin yakalanması ve bununla ilgili çizim işlemlerinin doğru yere yapılması işlemidir.

Harita üzerindeki sembol (MapElement) sayısı 30, 40 olana kadar fark edilmeyecek olan performans problemidir. Element sayısı bu rakamları geçtiği andan itibaren ekranda görüntülemede performans kaybı yaşandığı gözlemlenmiştir. Bu çizim ve fare hareketlerinin elementlere yansıtılmasında kendini göstermektedir. Bu noktada R-tree denilen indeksleme yönteminin bir çeşidi olan Quad-tree metodundan faydalanılmıştır.

R-tree indekslemesi harita uygulamalarında sıkça kullanılmaktadır. Algoritma olarak B-Tree (binary tree – ikili ağaç) algoritmasının iki boyutlu olanı denilebilir. B-Tree yönteminde her node (düğüm)'e bağlı iki alt düğüm bulunmaktadır. Düğüm değerine göre istenilen değer küçük ise bir alt düğüme, büyük ise diğer alt düğüme dallanmaktadır. Bu şekliyle ağaç yapısı üzende çok hızlı şekilde arama yapılabilmektedir. R-Tree'de ise aynı yöntem izlenmekle beraber, N boyut sayısı olmakla üzere 2N kadar düğüm devreye girmektedir. İki boyutta yapılandırma şu şekilde olmaktadır.

Çizelge 4.7. Düğüm yapısı

1. Nod X:-;Y:+	2. Nod X:+;Y:+
3. Nod X:-;Y:-	4. Nod X:+;Y:-

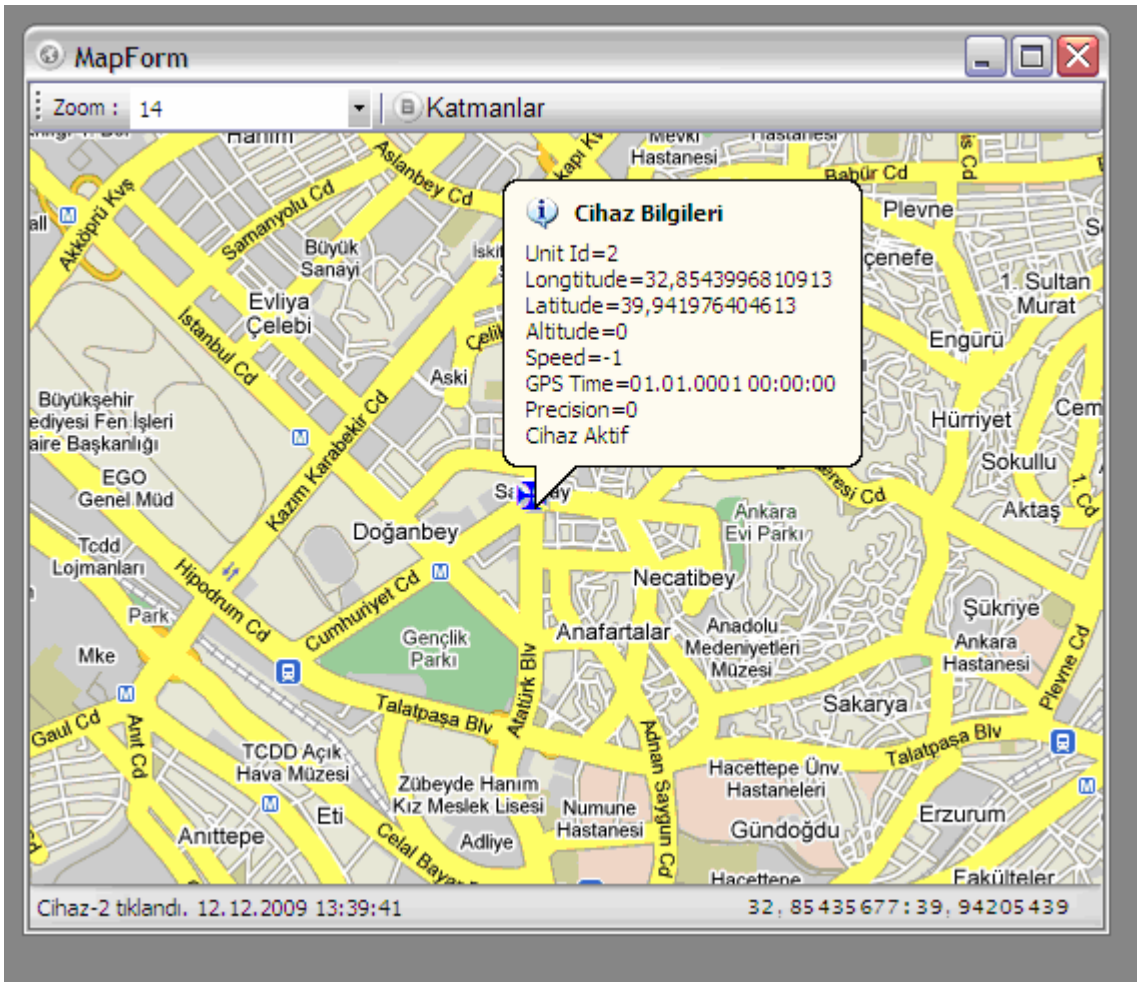
Karşılaştırma iki değerin büyüklük küçüklük durumlarına göre yapılmaktadır. Karşılaştırmaya göre hangi alt dala ineceği belirlenmektedir.

Bu uygulamada farklı olarak cihazlar hareketlidir. Bu şekilde bir yapı kurulduğu takdirde, cihaz her hareket ettiğinde tüm ağaç yapısının yeniden yapılanması söz konusudur. Bu nedenle R-tree'nin farklı bir gerçeklenmesi olan bir yöntem (Quadtree) seçilmiştir. Burada en üst düğüm olarak Türkiye'nin tümünü kapsayan tek bir düğüm vardır. Cihazlar mutlaka bu düğümün içinde olacaklardır. Eğer cihaz sayısı **8**'i geçmiyor ise bu cihazlar bu düğüm altında durmaktadırlar. Ancak sayı bu rakamı geçtiğinde alt dala cihazları yerleştirmek daha performanslı hale gelmektedir. Cihazların büyük bir bölümü örneğin İstanbul civarında olsun. X-Y+ düğümü otomatikman oluşturularak o düğüme bağlanır. Eğer X-Y+ düğümüne 8'den fazla cihaz yerleşirse burası da yine bir alt düğümünü oluşturacaktır. Böylelikle cihazlar en uygun ağaç yapısının altına yerleşmiş olacaklardır. Bu çözüm özellikle cihazların hareketli olması nedeniyle seçilmiştir. Cihaz bir bölgeden diğerine geçtiğinde bu düğüm altından çıkartılır, yine uygun bir düğüme yerleştirilir. İstenilen herhangi bir anda istenilen bölgeye düşen cihazlar ağaç üzerinden kolaylıkla bulunabilir.

Quadtree yöntemi cihazların harita üzerine hangi cihazların çizileceğinin elde edilmesi için ve fare ile etkileşimlerinde kullanılmaktadır.

MapForm üzerinde harita görüntüsü üzerinde yer alan cihaz sembolleri bulunmaktadır. Bu sembollerle cihazın anlık olarak konumu takip edilebilmektedir. Cihazın üzerine tıklanarak cihaz hakkında bilgi alınabildiği gibi, sağ buton aracılığı ile cihazın durumu değiştirilebilmektedir. Bu şekliyle pasif hale getirilebilir veya çalıntı olarak işaretlenip, tamamen sıfırlanması sağlanabilir.

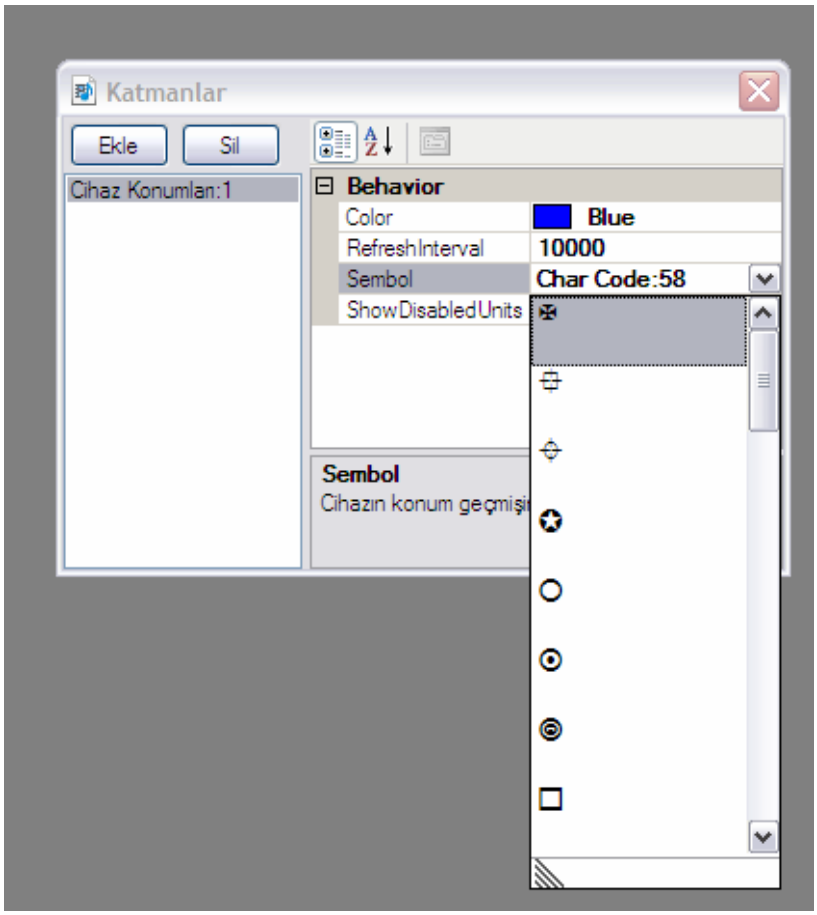
4.6.1. Ekran görünümleri ve açıklamalar



Şekil 4.39. Harita izleme cihaz bilgileri ekranı

Katmanlar menüsüne harita izleme ana ekranında “Katmanlar” düğmesine basılarak erişilebilmektedir.

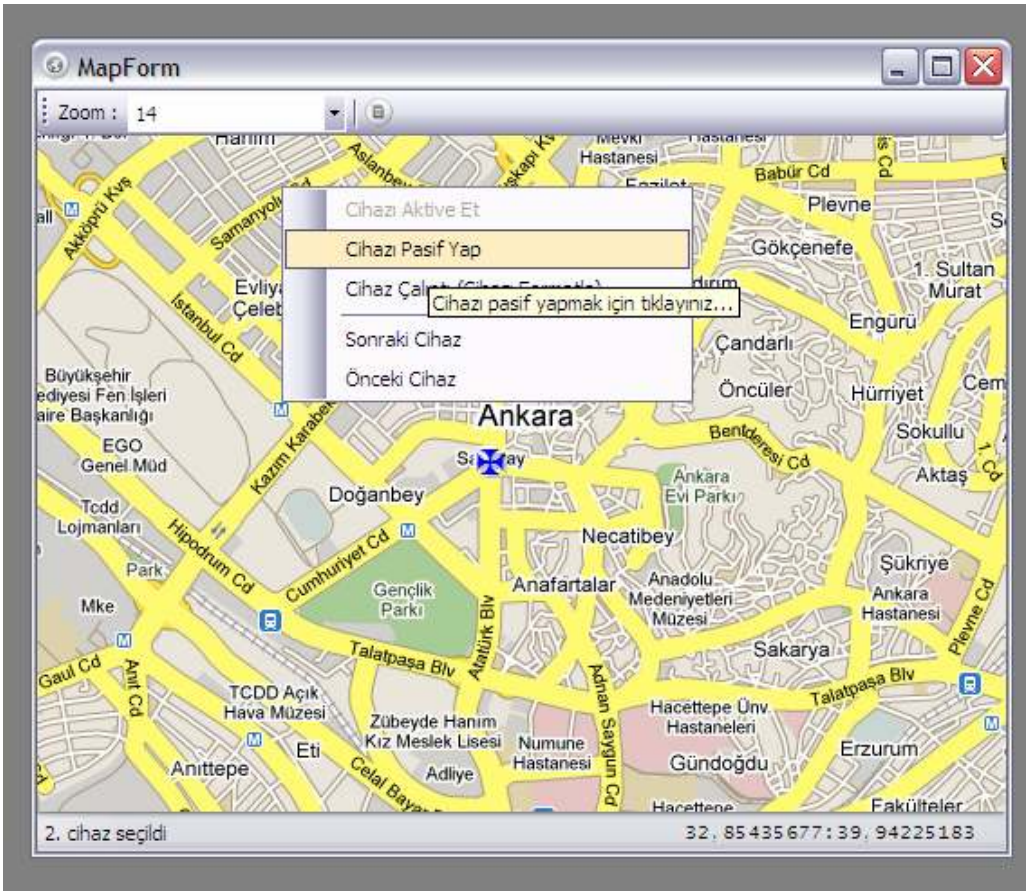
Cihaz konumları ve Konum Geçmişi olmak üzere 2 çeşit katman yapısı kullanılmaktadır. Katmanlar menüsünde yeni katmanlar eklenebilir, silinebilir, seçilen katmanın özellikleri değiştirilebilir.



Şekil 4.40. Cihaz konumları katmanı

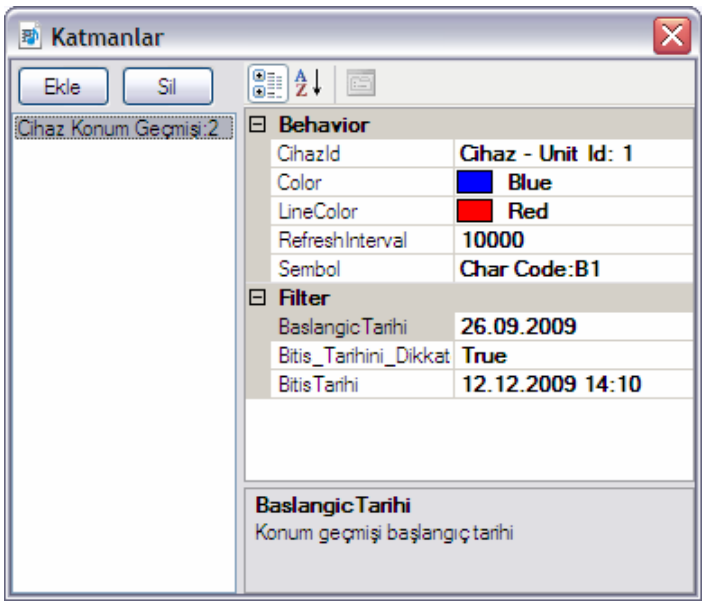
Cihaz konumları katmanı menüsünde:

- “Color” parametresi ile cihazın ikon rengi seçilir.
- “Sembol” parametresi ile cihaz konum katmanı sembolleri ayarlanabilmektedir.
- “RefreshInterval” parametresi ile canlı konumların ekranda tazelenme sıklığı milisaniye cinsinden belirlenir.
- “ShowDisabledUnits” parametresi ile de izleme anında aktif olmayan cihazların da ekranda gösterilmesi gibi işlemler yapılmaktadır.



Şekil 4.41. Cihaz konumları arası geçiş

Farklı cihazlar arası geçiş için imleç ekran üzerinde iken sağ fare tıklaması ile görüntülenmek istenilen cihazın seçilmesi yeterlidir.

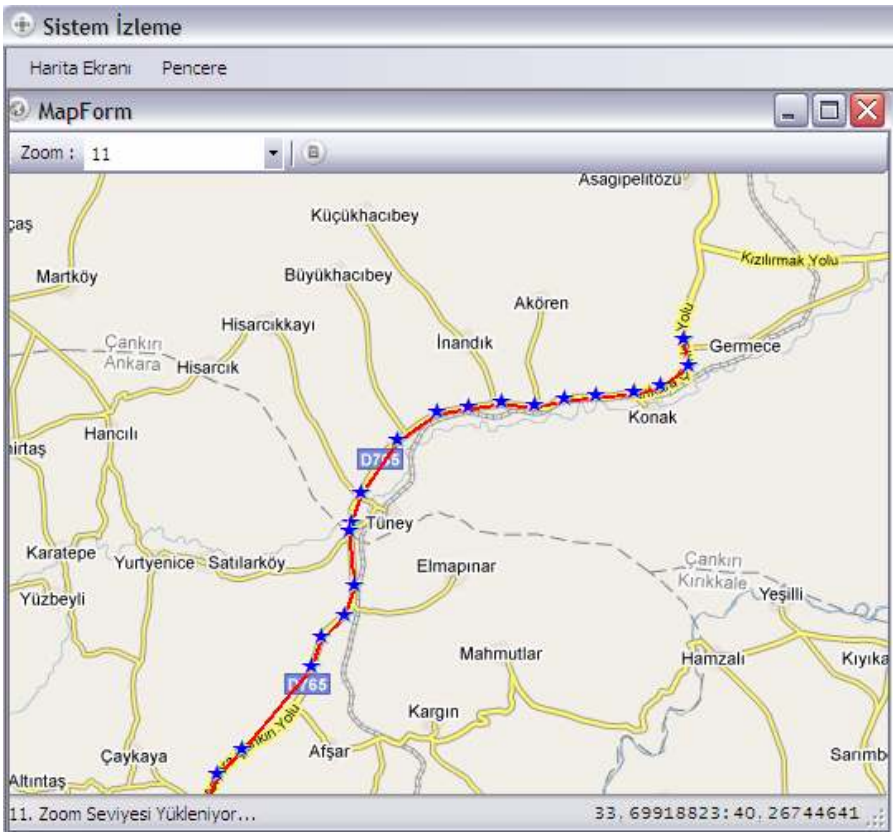


Şekil 4.42. Cihaz konum geçmişi katmanı

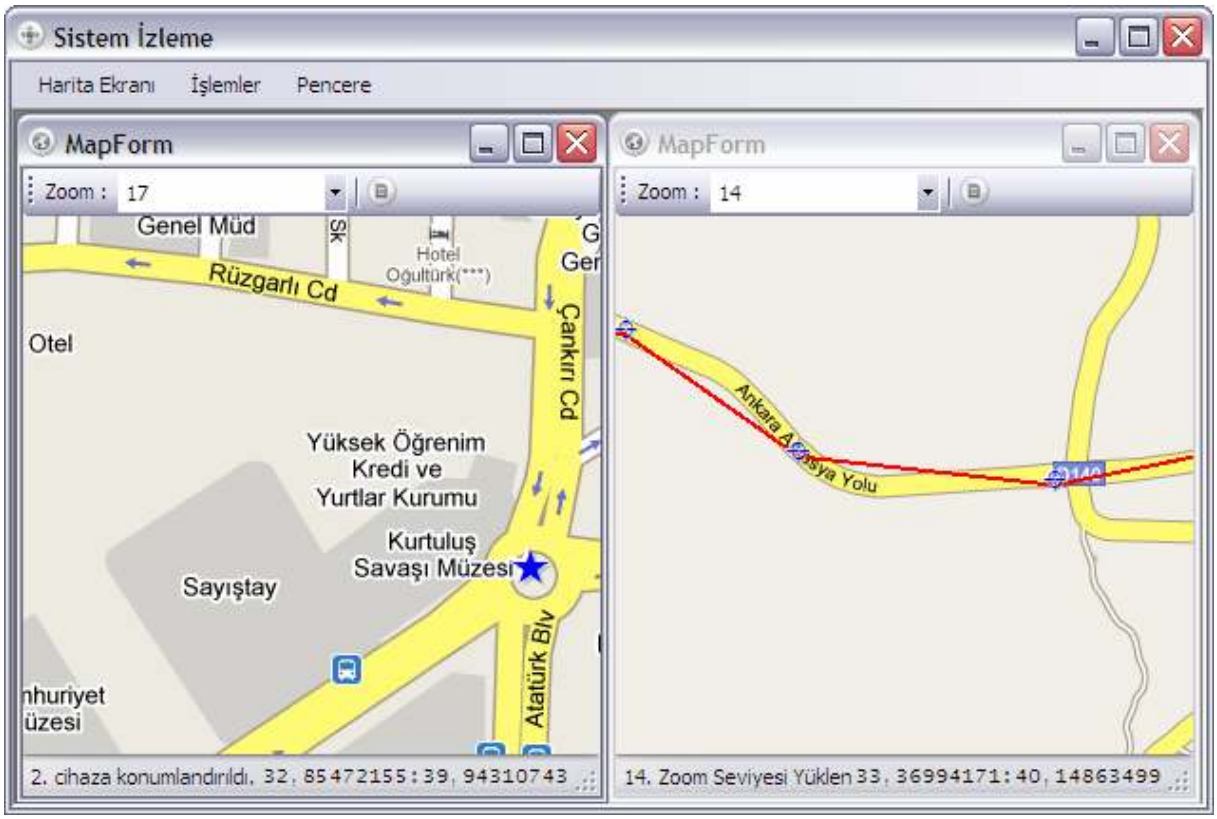
Cihaz konum geçmişi katmanı, sadece seçilecek cihaz için konum geçmişini göstermektedir. İkinci bir cihazın da konum geçmişi izlenmek istenildiğinde katmanlar menüsüne ikinci bir konum geçmişi katmanı eklenebilmektedir. Sadece farklı cihazların konum geçmişinin görüntülenmesi için değil, aynı cihazın farklı tarih aralıkları için de ayrı konum cihaz katmanları eklenebilmektedir.

Cihaz konum geçmişi menüsünde :

- “Cihaz Id” parametresi ile konum geçmişi görüntülenmek istenilen cihaz seçilir.
- “Color” parametresi ile görüntülenmek istenilen cihazın ikon rengi seçilir.
- “Line color” parametresi ile konumlar arası çizilecek birleştirme çizgilerinin rengi seçilir.
- “Sembol” parametresi ile cihaz geçmiş konum katmanı sembolleri ayarlanabilmektedir.
- “BaşlangıçTarihi” parametresi görüntülenmesi istenilen konum geçmişi bilgilerinin başlayacağı tarih ve saat seçilir.
- “Bitis Tarihi” parametresi görüntülenmesi istenilen konum geçmişi bilgilerinin bitiş tarihi ve saati seçilir.
- “Bitis_Tarihini_Dikkate_Alma” parametresi “True” olarak ayarlandığı taktirde “Bitis Tarihi” ayarları dikkate alınmaz. Seçilmiş olan cihaz aktif olduğu sürece ekranda gösterilecektir.

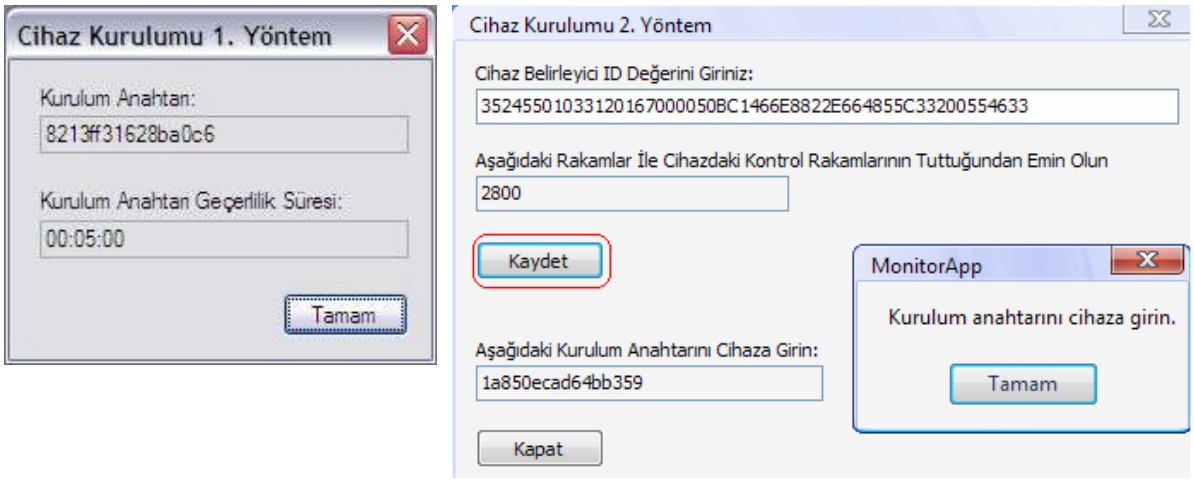


Şekil 4.43. Cihaz konum geçmişi ekranı



Şekil 4.44. Farklı pencereler ile izleme

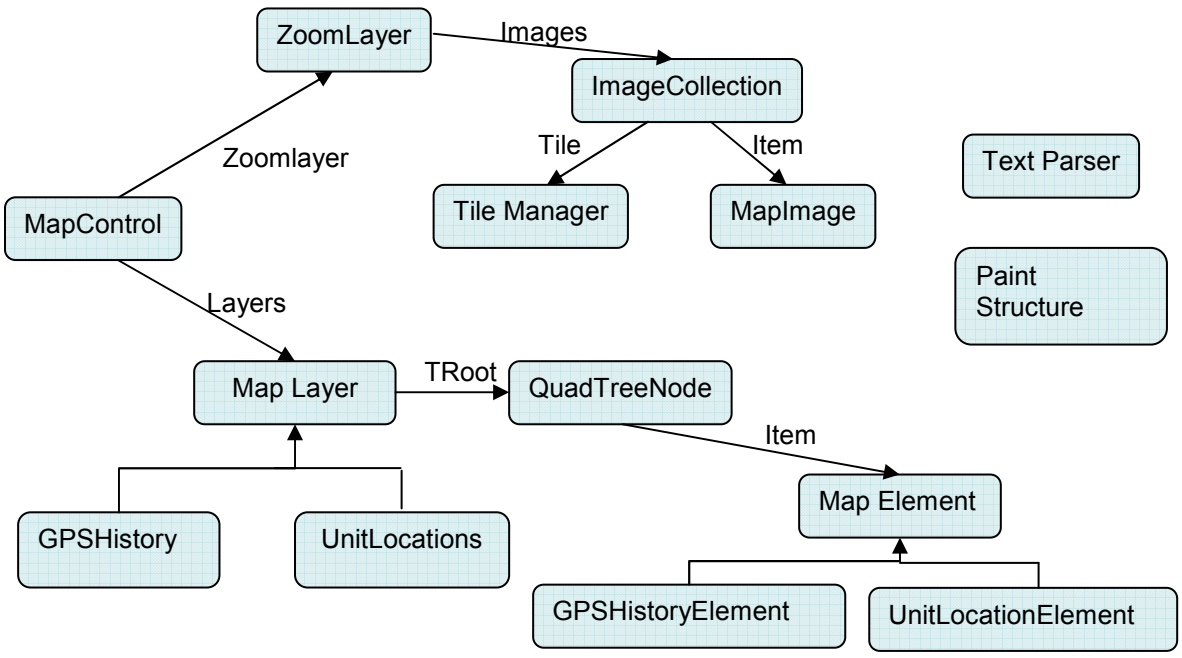
Harita ekranı menüsünden birden fazla pencere seçilebilmektedir. Böylece bir harita ekranında seçilmiş olan bir cihazın konum geçmişi izlenirken, bir diğer ekranda cihazların canlı konum bilgileri izlenebilmektedir.



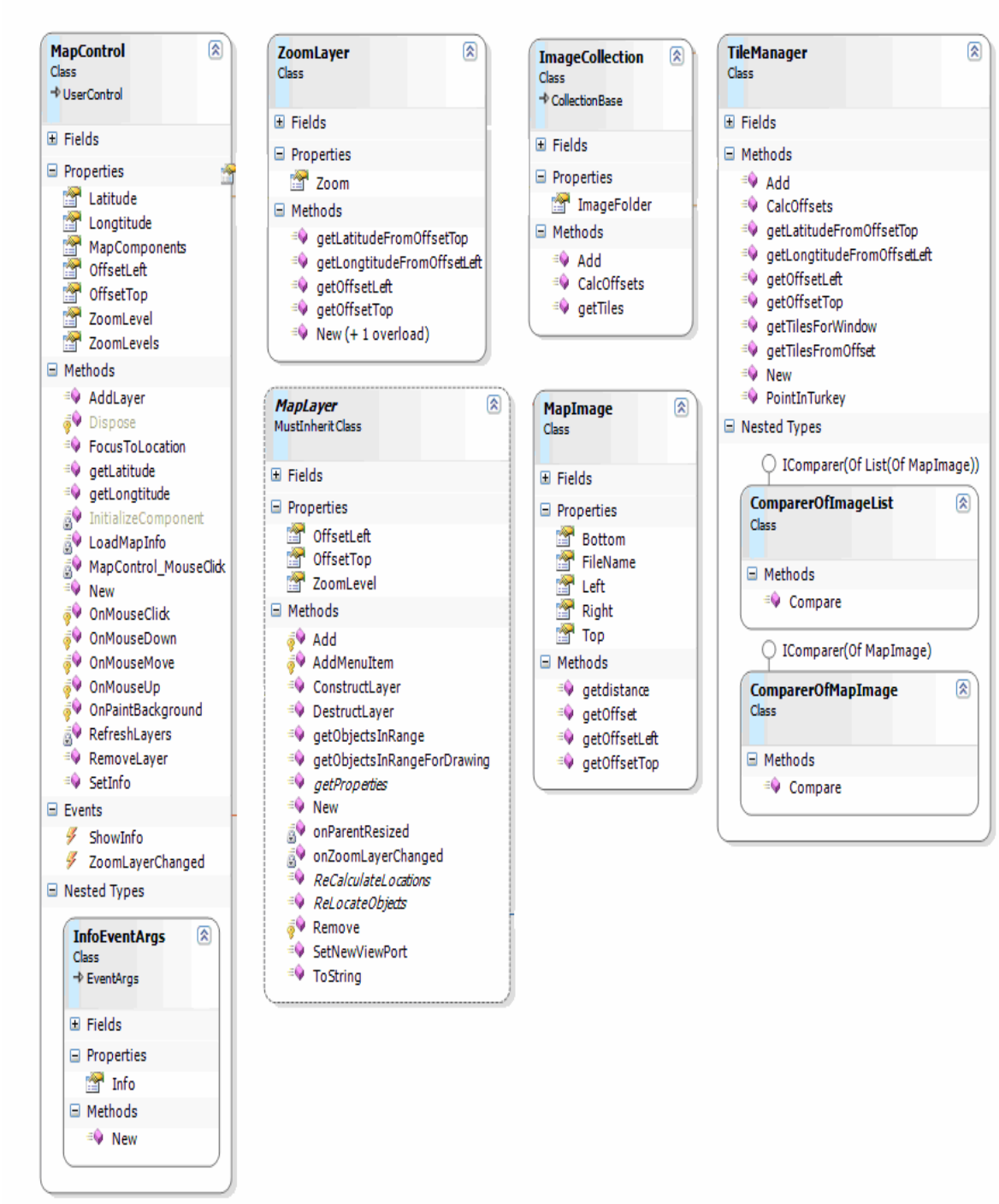
Şekil 4.45. Cihaz kurulum anahtarı gösterimi (1. ve 2. yöntem)

Program ana menüdeki İşlemler sekmesinden (Şekil 4.44.) cihaz kurulum işlemi için ilk aşamada kullanılan kurulum anahtarı da ekranda görüntülenebilmektedir.

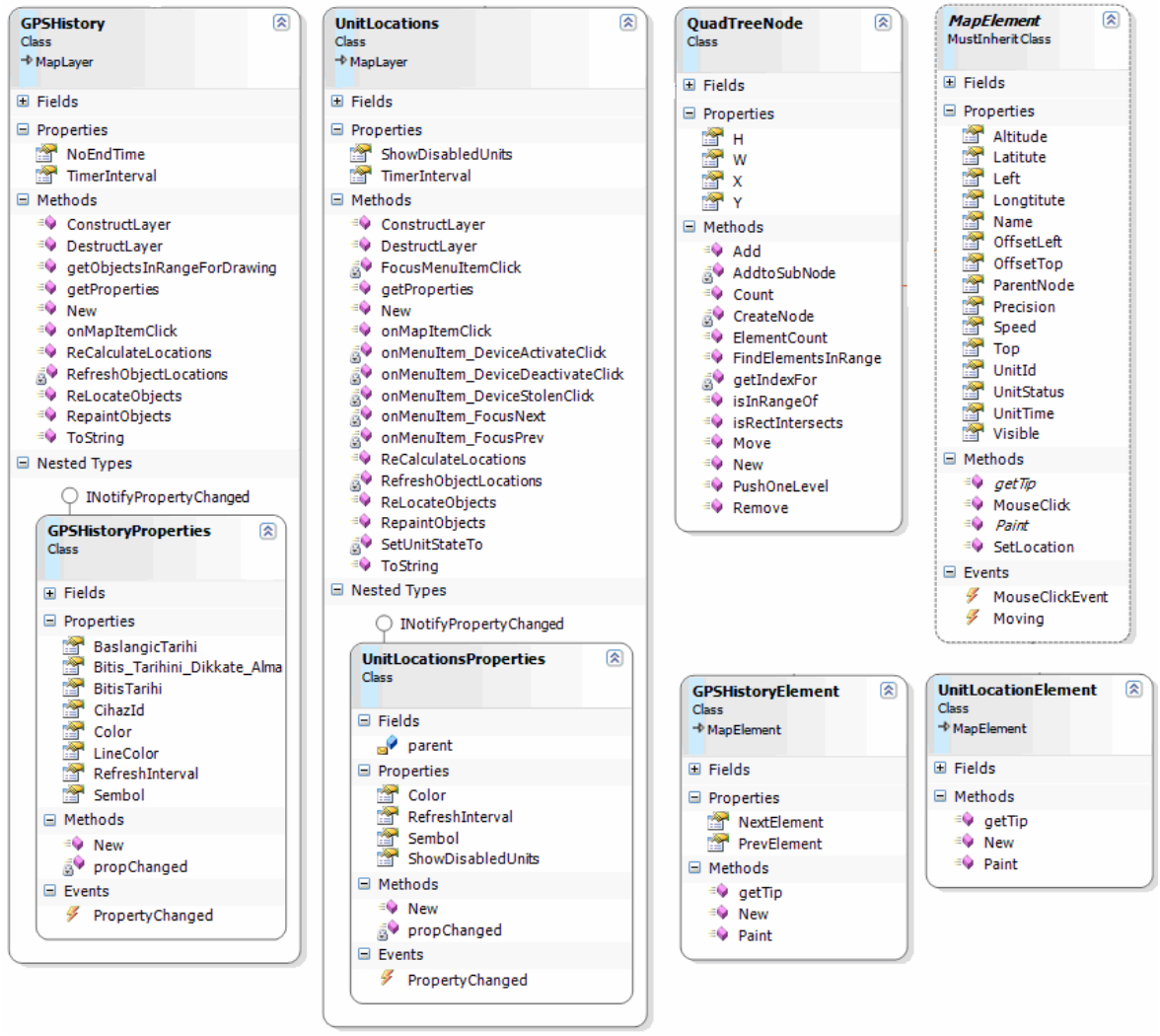
4.6.2. Harita izleme ve Yönetim Yazılımı Sınıfları



Şekil 4.46. Harita İzleme Yazılımı Sınıfları (bağlantılar)



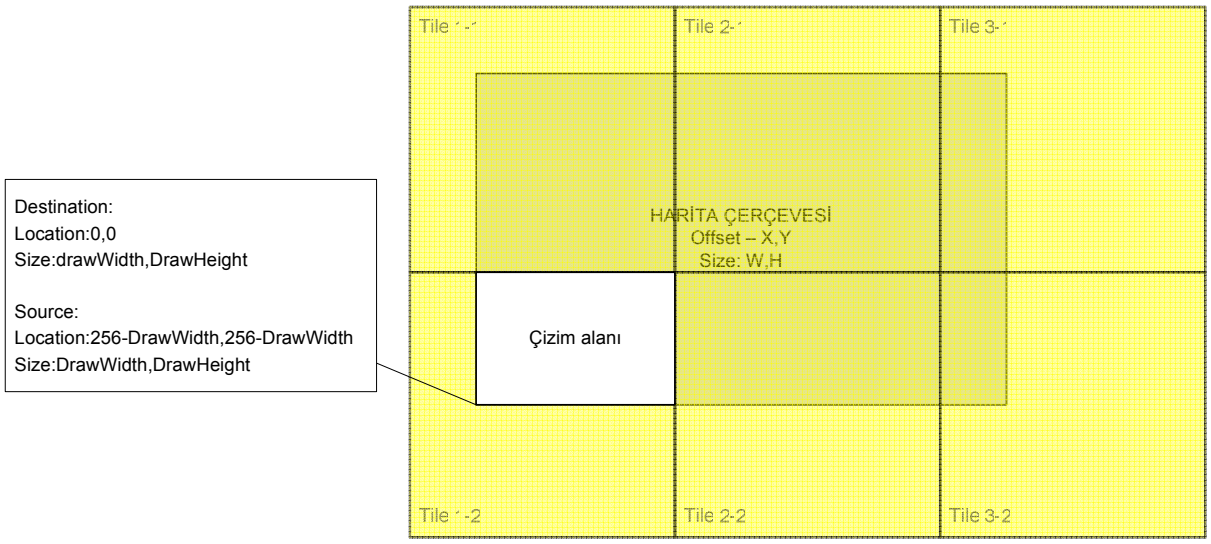
Şekil 4.47. Harita İzleme Yazılımı Sınıfları -1



Şekil 4.48. Harita İzleme Yazılımı Sınıfları -2

Harita görüntüsünün oluşturulması için MapControl isimli nesne kullanılmaktadır. Bu nesne bir System.Windows.Forms.UserControl sınıfından türemiş olup herhangi bir form üzerine yerleştirildiği andan itibaren harita görüntüsünü vermeye başlar. Harita görüntüsünü oluşturmak için Zoomlayer sınıfını kullanır. Bu sınıf yaratıldığında, zoom seviyesine göre liste dosyasından "textparser" sınıfından da faydalanarak haritayı oluşturmak için gerekli tile (döşeme) bilgilerini toplar. Bunu TileManager sınıfı üzerinde tutmaktadır. TileManager'da yer alan her bir tile aslında bir MapImage nesnesidir. Arka planda istenen X,Y koordinatı için hangi resimlerin çizilmesi gerektiğini TileManager belirler. Burada tile içerisinde nesneler Item[X][Y] şeklinde "Array of Array of MapImage" (MapImage matrisi) şeklinde saklar. Herhangi bir anda belli bir alanın içine düşen resimler istendiğinde ortalama bir hesap yaparak başlangıç ve bitiş indekslerini bulur. Bu aralıkta yer alan "Array of PaintStructure" (Paint Structure dizisi) şeklinde geriye döndürür. MapControl

objesine düşen sadece bu dizi üzerinde dolaşarak Çizeceği resim dosyalarını yükleyip uygun şekilde paint (çizdirme) işlemini gerçekleştirmek olmaktadır.



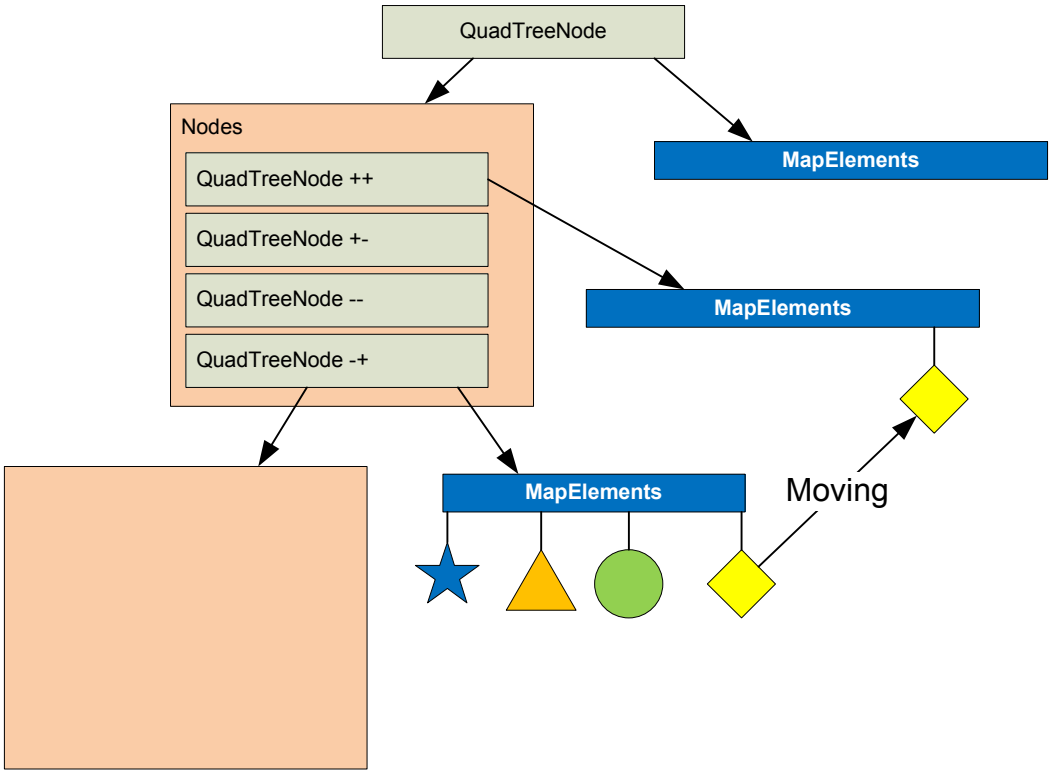
Şekil 4.49 Ekrana Harita Çizdirme

TileManager aynı zamanda, koordinat dönüşümlerini de gerçekleştirir. Örneğin ekranda farenin tıklandığı yerin hesaplanması ile bir cihazın enlem boylamlarından hangi X,Y değerlerinde gösterileceği de bu sınıf tarafından hesaplanır.

Harita kontrolü üzerinde bir cihazların takibinin yapılabilmesi için katmanlara ihtiyaç vardır. Bu katmanlar “Layers property”‘(katman özelliği) ile dışarı veriler MapLayer nesnesinden türemek zorundadır. MapLayer abstract (soyut) bir sınıf olup türetilmeden kullanılamaz. Bu sınıftan türeyen UnitLocations ve GPSHistory isimli iki sınıf mevcuttur. Bu sınıflar üzerinden işlemler gerçekleştirilir. Harita katmanı ekranın çizilmesi esnasında her bir layer objesi için ayrı ayrı ekrana çizilecek MapElement sınıfındaki nesneleri ister. Bunun için çerçevenin başlangıç ve bitiş koordinatlarını gönderir. Gelen nesnelerin Paint (çizdirme) metotları çağrılarak harita üzerine cihaz simgesinin basılması sağlanır.

UnitLocations (Cihaz konum bilgisi) nesneleri ve GPS history (GPS geçmiş konum bilgisi) nesneleri bir QuadTreeNode yapısı içinde saklanır. uygulamada cihazların hareketi de göz önünde bulundurularak ağaç yapısı QuadTreeNode öğeleri ile oluşturulur. MapElement nesneleri ise bu düğümlere bağlanarak saklanır. Cihaz bir konumdan diğerine hareket ettiğinde düğümün sınırları dışına çıkmış ise bir

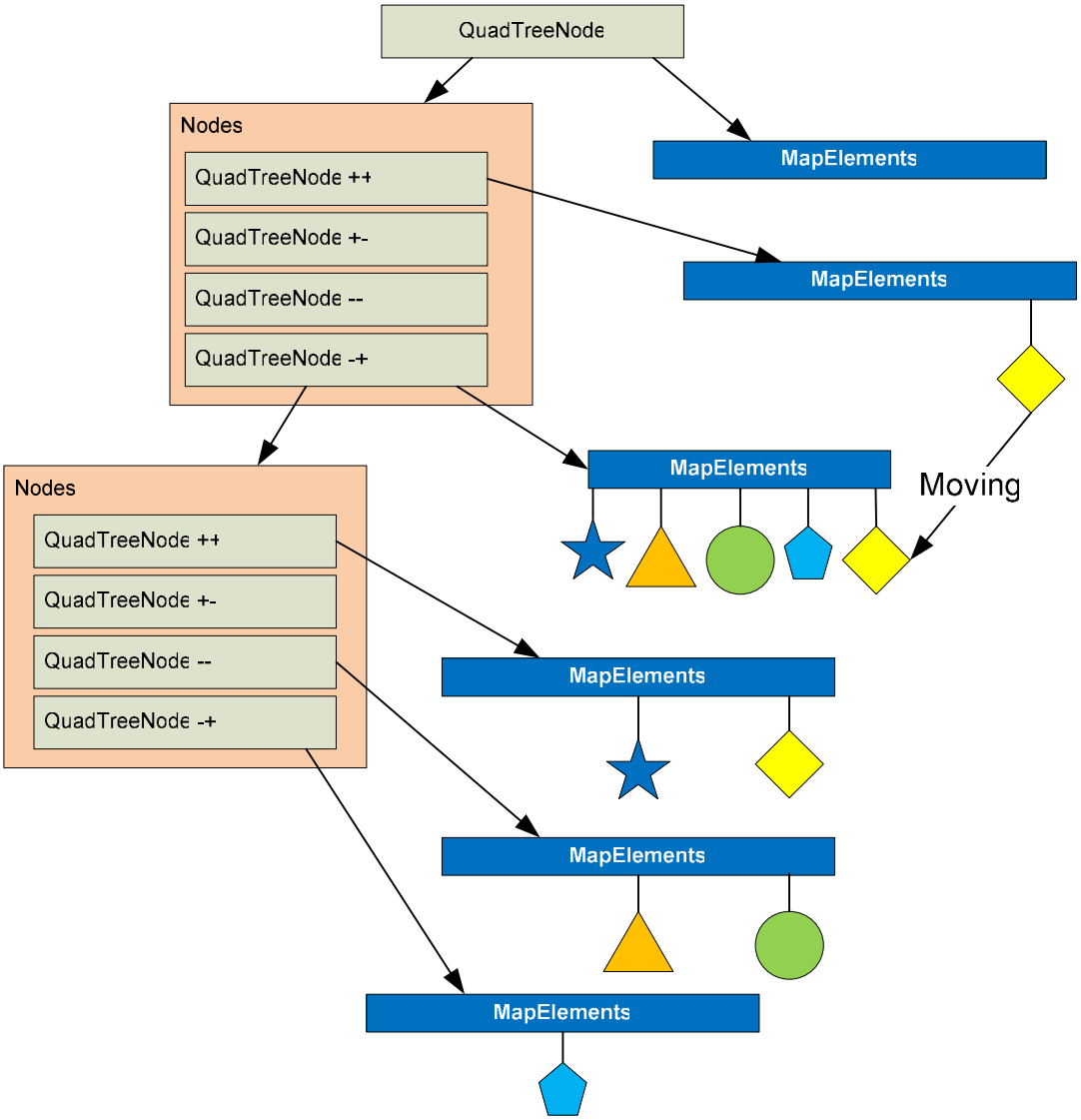
QuadTreeNode öğesinden diğerine taşınır. Tree üzerinde istenen çerçeve üzerindeki öğelere ulaşmak için öncelikle tile ile kesişen düğümler bulunur ve sadece bunlar altındaki Mapelement'lerden uygun olanlar alındığı için oldukça hızlı bir şekilde listeye erişilebilir. Bu işlem belki 10-20 element için çok bir performans farkı getirmez. Ancak GPShistory gibi bir katmanda binlerce kayıt bir anda ekrana gösterilmek istendiğinde sadece çizilmesi gerekenleri getirmek bile birkaç saniyeyi alabilir.



Şekil 4.50. Quadtree oluşturma

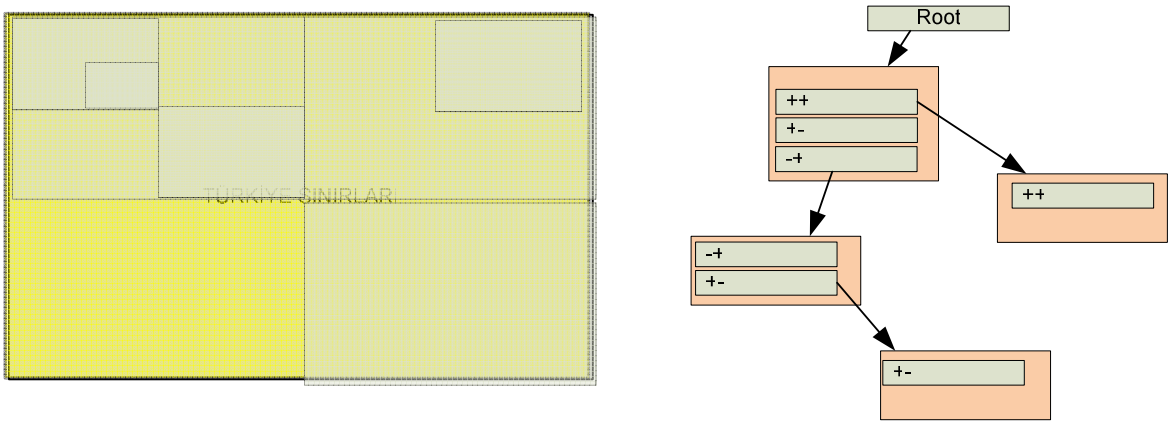
Bu sistemde işleyiş bu şekilde olmakla beraber işleyişi iyileştirmek için bir takım ek fonksiyonlar getirilmiştir. Örneğin düğümlerin oluşturulmasında şöyle bir yol izlenmektedir. Öncelikle tüm Türkiye'yi kapsayacak tek bir düğüm oluşturulmaktadır. Normalde tüm istekler bu düğüm ile kesişir. Ve tüm elementler de bu düğüme bağlanır. Ancak bağlı element sayısı belli bir sayıyı geçtiğinde for döngüsü içinde bunları tek tek incelemek yerine alt düğümlerin oluşturularak bunlardan kesişenlerin içinde for döngüsü kurmak arama süresini en az teorik olarak %50 kısalacaktır. Bir düğüm altında Element bulunup bulunmadığı bir IsLeaf bayrağı (flag) ile saklanır. Eğer altına düğüm açılırsa bu flag (bayrak FALSE (mantıksal yanlış) yapılarak alt dallara dallanması sağlanır. Bu flag TRUE

(mantıksal doğru) ise elementler sadece MapElements altındadır. Sembolik olarak bu yapı Şekil 4.50'de gösterilmiştir. Başka bir dalda yer alan MapElement bir düğüme aktarıldığında eğer o düğümde çok fazla element oluşursa (Uygulamamız için 8 değeri) otomatik olarak alt düğümler oluşturulur ve elementler bunlara dağıtılır.



Şekil 4.51. Quadtree elemanların düğümlere dağılımı

Görüldüğü üzere Kök'e (Root) bağlı -+ düğümüne eklenen elementten sonra bu element altındaki kayıtlar alt düğümlere dağıtılacaktır.



Şekil 4.52. Quadtree düğüm oluşturma

Düğümlerin oluşturulması ve bölümlenmesi Şekil 4.52'de gösterilmektedir. Her bir düğüm alt dörtte bir alana sahip 4 düğüme bölünmektedir. Eğer düğüm gerekli değil ise hiç yaratılmamaktadır.

Türkiye dışında element oluşması durumunda bir eklenti olarak root elementle aynı genişlik ve yükseklikte 8 ayrı düğüm ihtiyaç olması halinde yaratılabilmekte ve dağılım bu yan düğümlerden yapılmaktadır.

R-Tree düğüm olarak MapElement objeleri konmuş olsaydı, cihazın her hareketinde tüm alt dallanmanın yeniden oluşturulması gerekecekti. Bu durumda her hareketin yansıtılması tüm ağacın yeniden yapılandırılmasına neden olacağından en uygun yapı olan QuadTree yapısı tercih edilmiştir.

MapControl objesi bu ağaç yapısından iki noktada faydalanmaktadır. Çizim sırasında çerçeve içinde kalan mapelement objelerinin bulunmasında ve ekranda fare ile tıklanan noktada bulunan "Mapelement" 'in tespiti için. Fare tıklaması ile ilgili event (olay) oluştuğunda, X,Y değerinden bu noktada bulunan "Mapelement" ler tespit edilir ve bu elemente ait MouseEvent 'i çağrılır. Buradan da ilgili fonksiyona yönelik işlem gerçekleştirilir. Bu bir tooltip göstermek olabildiği gibi, menü açmak da olabilir.

MapControl üzerinde ekrandaki harita aynı fare ile tıklanarak sürüklenerek taşınabilmektedir. Bu durumda sadece Offset değerleri yeniden hesaplanıp ekranın yeniden çizilmesi sağlanır. Burada performans için elementlerin çizimi fare ile taşıma başladığında iptal edilir.

MapElement sınıfından iki sınıf türemektedir. UnitLocationElement ve GPSHistoryElement bunların çizim metotlarında da farklılıklar vardır. Unitlocation sadece belirlenen bir sembolü ekrana çizerken, GPSHistory bir önceki konum ile araya çizgi çizdirmektedir. Burada bir önceki konum bilgisi element olarak yüklenirken kendi aralarında da bir zincir oluşturacak şekilde de özellikleri tanımlanır.

MapLayer objesinden türeyen katmanlar için özelliklerin tanımlanacağı bir obje property (özellik) olarak istenmektedir. Bu obje Property Editor kontrolüne verilerek katmana ait özellikler arayüze yansıtılır. Yapılan değişikliklerin de katmana yansıması sağlanır.

UnitLocation katmanı sadece aktifler ve hem aktif hem de pasiflerin gösterilmesi için ayarlar içermektedir.

GPSHistory katmanı ise belli bir cihaza yönelik olarak geçmişi gösterebilmektedir. Birden fazla cihazın geçmişi gösterilmek istenirse GPSHistory katmanının ikinci kez yüklenebilir. Bu şekliyle birçok cihazın konum geçmişleri verilecek tarih aralığına bağlı olarak üst üste gösterilebilir. Özellikleri değiştirilerek her katman için farklı sembol ve renk seçilebilir.

4.7. Olası atak senaryoları ve alınan önlemler

Olası ağ saldırıları yöntemleri ve bunlara karşı alınan önlemler aşağıda listelenmiştir.

4.7.1. Man in the middle (Ortadaki adam)

Ortadaki adam ataklarında ağ trafiğini dinleyen saldırgan aşağıdaki yöntemlerden faydalanarak ağ saldırılarında bulunabilmektedir.

Yöntem : Dinleme

- Cihaz kurulumu yapılırken cihaza ait veriler çalınmaya çalışılabilir
    - Cihaz kurulum esnasında verileri D-H güvenli anahtar değişimi ile şifrelemiş olarak göndermektedir. http SessionID ile sunucu tarafında D-H anahtarlar saklanmakta ve cihaz tekrar geldiğinde aynı anahtar ile veri çözülmeye çalışılmaktadır. Bu aşamada anahtarlar kısa süreli

kullanımda olduğu için cihaz bilgilerinin yakalanması söz konusu değildir.

- Cihaz konumları çalınabilir.
    - Cihaz konumları cihaz bazında oturum açma zamanında belirlenen bir anahtar ile şifrelenmektedir. Bu anahtar belli sürelerde değiştirilmektedir.

Yöntem: Zehirleme

- Cihaz kurulumu ve konum bilgilerinin gönderilmesi esnasında sanki cihazdan veya sunucudan geliyor gibi paket gönderilip güvenli anahtar değişimi engellenebilir:
    - Bağlantı “http request” (http bağlantı isteği) süresince açık kalmaktadır. İstek ve cevap süresince gönderilecek paketler sistemi etkileyebilir. Bu noktada zehirleme ancak istek yollandıktan sonra cevap gönderme şeklinde olabilir. Cevap bilgisi olarak sunucudan gönderilen veri PSK ile şifrelidir. Bir şekilde daha önce başka istemciye gönderilen paketin tekrarlandığını ve istemciye yanlış cevap ulaştırıldığını varsayarsak, arada şifreleme için istemcide yanlış anahtar belirlenmiş olacaktır. Gönderilecek ilk paket ile birlikte paketin açılamadığı görülecek ve NoSession (oturum mevcut değil) cevabı ile istemcinin tekrar oturum açması sağlanacaktır.
    - Zehirleme paketinin hazırlanabilmesi için, istemci IP numaralarını bilmesi gerekir. Oysa istemci IP adresleri her yeni GPRS bağlantısında değişir.
- Cihazın güvenli anahtar değişimi sonrası kurulum aşamasında paket tekrarı:
    - Sunucu paketinin tekrar edilmesi istemci tarafı için bir problem oluşturmamaktadır. İlk paketi aldıktan sonra bağlantıyı kapatacaktır. Bu paketin sunucu taklidi yapan bir bilgisayardan gelmesi durumunda istemci paketi açmaya çalışacaktır. Artık D-H anahtarı ile

veri şifrelendiğinden, çözülememesi durumunda iletişim gerçekleşmemiş olacaktır. Kullanıcıya verilecek uyarı sonrasında kullanıcının tekrar kurulum başlatmasıyla işlem tekrarlanabilir.

- İstemci paketinin tekrar edilmesi durumunda sunucu tarafında buna özgü servis içinde gerekli önlemler alınmaktadır. Örneğin kurulum işleminde daha önce kurulmuş ise şeklinde denetim yapılmaktadır. Konum göndermede de sıra numarası kontrolü ile veri devre dışı bırakılabilmektedir. Burada kritik bir nokta istemci paketinin tekrarı bir saldırı olmayabilir. Cihaz da eğer GPRS üzerinden bağlıyken sunucu cevabını alamamış olabilir. Bu noktada isteği tekrarlayacaktır. Böyle bir işlem için alınacak tedbir cihazın veri göndermesini veya düzgün iletişim kurmasını engelleyebilir.

4.7.2. Denial of Service (Servis engelleme)

Bu tür ağ saldırılarında amaç bir ağ servisinin aşağıdaki yöntemler kullanılarak hizmetinin engellemesi ya da yavaşlatılmasına yöneliktir.

Yöntem: Servis Durdurma

- Sunucu yoğun oturum açma talepleri doğrultusunda bellek alanı doldurularak sistem susturulabilir.
  - ASP.NET belli bir miktar (%60) bellek kullanımından sonra IIS Worker Thread'ini kapatmakta ve yeniden başlatmaktadır [31]. Susma söz konusu değildir.
  - IIS günlük kayıtları nedeniyle bazı durumda işletim sistemine açılan disk alanı boyutu sınırlı ise disk alanı dolabilir. Bu tür durumda sunucu cevap veremeyebilir. Bu tür bir saldırı sonucu sistemin susması oldukça zaman alacaktır.

Yöntem: Sistemin Cevap Veremez Duruma Düşürülmesi

- Sunucuya yoğun oturum açma talebi gönderilerek sunucu cevap veremez duruma getirilebilir:

- D-H güvenli anahtar değişimi esnasında sunucunun ürettiği anahtarın oluşturulması zaman almaktadır. Yoğun talebin sistemi yoracağı kesindir. Gerçekte CPU kullanımı belli bir noktayı geçtiğinde ASP.NET tarafında yeni gelen talepler iş kuyruğuna atılmaktadır. Sistemin tamamen susturulması söz konusu değildir. Sadece geç cevap verebilir. Sunucu CPU kullanımı monitör edildiğinde, bir kaynaktan yoğun bağlantı talebi olduğu görüldüğünde saldırgan tespit edilmiş olacaktır.

- Gerçekte sunucu sistemlerin mevcut performansı ile çok yoğun talepler bile olsa sunucular büyük bir ölçüde bunu karşılayabilirler. Bu durumda problem sunucu tarafı iletişimi için kullanılan bant genişliği olacaktır. Bant genişliğini dolduracak kadar büyük talep gelmesi durumunda en büyük olasılık cihazların veri göndermesinde RequestTimeout (İstek zaman aşımı) hatası alınabilir. Bu tür saldırılar çoğunlukla uzun süreli yapılmaz. Uzun süreli saldırılar saldırganın tespitini kolaylaştırır.

4.7.3. Spoof Attack (Aldatma saldırısı)

Bu tür ağ saldırılarında amaç yetkili kullanıcı taklidi yapılarak sisteme erişim sağlamaya yöneliktir.

Yöntem: Kimlik denetiminden geçmeden işlem yapılması

- İstemcinin kullandığı SessionID değeri kullanılarak Identity (kimlik) çalınabilir ve sunucuya yanlış bilgiler gönderilebilir.

  - Veri iki tarafın bildiği bir anahtar ile simetrik olarak şifrelendiği için böyle bir durum söz konusu değildir.

  - Veri istemcide HMAC-SHA1 ile imzalandığı için imzasız veya geçersiz imzaya sahip veriler iptal edilir.

4.7.4 Flood Attack (Taşma saldırısı)

Yöntem: Sistemin cevap vermesinin engellenmesi

- Rastgele veri gönderilerek sunucu bant genişliği doldurulabilir. Böylelikle sistemin cevap vermesi engellenebilir.

- IPS / IDS sistemleri bu tür atakları tespit edebilir ve saldırgan tespit edilebilir.

5. SONUÇ ve TARTIŞMA

Bu tez kapsamında PDA cihazlarda çalışacak bir istemci yazılımı kullanılarak cihazlarda GPS alıcısından elde edilen konum bilgilerinin şifrelenmesi ve TCP/IP protokolü ile GPRS bağlantısı üzerinden merkez sunucu ile haberleşmesi gerçekleştirilmiştir. Sunucu yazılımı, cihazlardan gelen şifreli paketleri çözümleyerek bilgileri yeniden oluşturmakta ve veritabanında bu bilgilerin kaydedilmesi işlemini gerçekleştirmektedir. Böylece geçmiş tarihlere yönelik sorgulama da yapılabilmektedir. Google maps uygulamasından indirilen png uzantılı resim dosyaları kullanılarak bir harita izleme yazılımı geliştirilmiş ve görsel olarak cihaz konumlarının izlenebilmesi sağlanmıştır. Ayrıca cihazların geçmiş konum bilgilerinin ve aktif, pasif olma durumları ile çalıntı cihaz olması durumunda sistem dışı bırakılması gibi işlemler de yapılabilmektedir.

Sunucu tarafında multi threading (çoklu işleme) özelliği olan IIS servisi kullanılarak http protokolü üzerinden istemci – sunucu haberleşmesi işlemi gerçekleştirilmiştir. İstemci sunucu haberleşmesinde kullanılan TCP/IP paketleri http header (önek) leri dahil azami 0.5 kilo bayt civarında tutulmuştur. Böylece GPRS bağlantısı sırasında internet erişim hızının cihazın hareket hızına, cihazın GSM baz istasyonuna uzaklığına ve bağlı olunan baz istasyonu GPRS kullanıcılarının anlık sayısına bağlı olarak azaldığı durumlarda da iletişimin devamlılığının sağlanması amaçlanmıştır. Ayrıca PDA cihazın sunucu ile iletişiminde Thuroya (15 kbit upload hızı) ya da Globalstar uydu bağlantısı gibi yöntemler ile haberleşmesi durumunda da iletişim veri hızının düşük tutulması avantaj sağlayacaktır [32]. Bu nedenler ile IPSEC VPN ya da SSL VPN gibi oturum boyunca sürekli iletişimin kurulduğu yapılar yerine etkin ve yeterince güvenli olacak bir güvenli haberleşme metodu oluşturulmaya çalışılmıştır.

Piyasada paralı olarak satılan birçok vektörel tabanlı harita işleme yazılımı bulunmaktadır. Ayrıca bu yazılımlar ile olası değişikliklerin güncellendiği haritalar yıllık lisans ücretleri ile satılmaktadır. Bu tez kapsamında sunucu tarafında oluşturulan bilgiler basit bir arayüz programı ile verilerin rahatlıkla bahsedilen tipte olan ücretli harita işleme yazılımlarına aktarılması sağlanabilir. Tez kapsamında veritabanında işlenen verilerin görselliğinin oluşturulması için ücretsiz olan Google maps hizmetinin hazır api'leri kullanılarak önce çevrimiçi internet bağlantısı

gerektiren bir Windows uygulaması ile harita izleme yapısı tasarlanmıştır. Ancak güvenli bir kanal vasıtası ile elde edilen bu verilerin, çevrimiçi olarak Google maps hizmetine bağlanılarak gönderilmesi sonuçta bir güvenlik açığı yaratmaktadır. Bu nedenle Google maps haritaları internetten indirilerek çevrimdışı (internet bağlantısız) bir yapı tasarlanması gerçekleştirilmiştir. İstenilen zoom (mesafe) seviyelerinde (5-19) Türkiye'yi kapsayacak şekilde haritalar png resim formatında indirilip ilgili dizinlerde belirlenen formatta izleme yazılımında kullanılmak üzere kopyalanmıştır. Böylece internet bağlantısı kurulmadan da bir Harita izleme yazılımı gerçekleştirilmiştir. Aynı yöntem kullanılarak belirli aralıklar ile (6 ayda bir gibi) bu resim dosyalarının Google maps servisinden yeniden indirilmesi de belirli bir oranda harita bilgisi güncellenmesi için kullanılabilir.

Harita izleme yazılımının gerçek kullanım ortamında güvenli bir yapıda (Firewall arkasında, intranet yapısında) çalışan bir bilgisayarda çalışması gereklidir. Uzak internet kullanıcılarının da bu hizmeti kullanmak istemesi durumda bir kimlik doğrulama işlemi sonrası (Radius sunucu üzerinden gibi) uzaktan iç ağa bağlantı kurarak ve bir VPN (sanal özel ağ) uygulaması ile güvenli bir iletişim kanalını kurması gerekli olacaktır. Bu aşamada VPN kullanılması uygun olabilir. Zira uzak izleme kullanıcısının, mobil cihazlarda çalışan uygulamada olduğu gibi internet erişimi hız problemi genellikle olmayacağından rahatlıkla bu yapı kurulabilir. Test ortamında Uzak internet kullanıcısının bilgisayarına harita izleme uygulaması kurulmuş ve açık kaynak kodlu olan OpenVPN istemci yazılımı kullanılarak sunucu tarafında kurulmuş olan OpenVPN sunucusuna sanal özel ağ bağlantısı yapılarak sunucu veritabanından verilerin güvenli olarak uzak internet kullanıcısına iletilmesi gerçekleştirilmiştir [33].

İleriye yönelik yapılabilecek çalışmalarda istemci cihazlar ile sunucunun haberleşmesinde TCP/IP protokolü üzerinde http servisi kullanılması yerine belirlenmiş bir port (kapı) üzerinden servis verecek bir TCP sunucu uygulama yapısı geliştirilebilir. Http sunucu ile iletişimde gelmekte olan (azami 0.2KB) ilave paket boyutu minimize edecek şekilde bir TCP soket sunucu tasarlanabilir. Burada önemli parametre sunucu uygulama yazılır iken multi threading yapısının düzgün tasarlanabilmesi olacaktır. Bu sunucu uygulamasının çok sayıda istemci isteğine sorunsuzca cevap verebilecek yapıda modellenmesi gerekecektir.

İstemci cihazlarda kullanılan PSK ile D-H anahtar değişimi işinin başlatılması simetrik şifrelemenin uygulama kolaylığından dolayı etkin olmakla birlikte, kullanılan anahtarların belirli zamanlarda değiştirilmesi gerekmektedir. Bu aşamada istemci cihazlar, D-H anahtar değişimi ile oturum şifresini belirleme aşamasında, D-H parametrelerini içeren paketleri asimetrik şifreleme (RSA) kullanarak sunucu genel anahtarı ile şifreleyerek ve paketi istemci özel anahtarı ile imzalayarak gönderme yöntemi (2.senaryo) de tercih edilebilir. Bu noktada sıkıntı, 1024 bitlik asimetrik anahtar kullanılması durumunda 80 bitlik anahtar uzunluğuna sahip simetrik şifreleme gücüne ulaşılabildiği [34] ve asimetrik anahtar şifrelemenin gerçeklenmesinin simetrik anahtar şifrelemeye görece daha fazla işlemci gücü ve zamanına ihtiyaç duyacağıdır [35]. Bu aşamada tercih, tasarlanacak yapının büyüklüğü, kullanılacak istemci PDA sayısı, PSK anahtarı belirli aralıklar ile güncellemenin kolay olup olmadığı durumlara göre belirlenebilir.

KAYNAKLAR

[1] America.gov. United States updates Global Positioning System http://www.america.gov/xarchives/display.html?p=washfile-english&y=2006&m=February&x=20060203125928lcnirellep0.5061609 (22 Temmuz, 2009)

[2] E.D.Kaplan, 2006, Understanding GPS:Principles & Applications

[3] List of GPS Satellite Launches , Kasım, 2009 http://en.wikipedia.org/wiki/List_of_GPS_satellite_launches

[4] Misra, P., Enge, P., 2001, "Global Positioning System: Signals, Measurements, and Performance", Ganga-Jamuna Press, Lincoln.

[5] Kahveci, M., Yıldız, F., 2005, "GPS Global Konum Belirleme Sistemi Teori-Uygulama", Nobel, Ankara,1-2

[6] Özenç R.F., 2003, "Lokal Alan Diferansiyel GPS Hassas Yaklaşması ve GözlemSonrası Hesaplama Yöntemiyle Uygulaması", Yüksek Lisans, Gazi Fen Bilimleri Enstitüsü, Ankara

[7] Leick, A., 2004, "GPS Satellite Surveying",Wiley, USA, 72

[8] Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/General_Packet_Radio_Service (11 Temmuz, 2009)

[9] ETSI, http://www.etsi.org/WebSite/Technologies/gprs.aspx (12 Ağustos, 2009)

[10] Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Enhanced_Data_Rates_for_GSM_Evolution (11 Eylül, 2009)

[11] Tanenbaum, 2004, Computer Networks, A.S.

[12] J.Daemen, V.Rijmen, 2002, The Design of Rijndael:AES-the Advanced Encryption Standard,

[13] NIST, 2001, "Advanced Encryption Standard," FIPS PUB 197

[14] Diffie W., Hellman M.E., New directions in cryptography, IEEE Trans. IT-22 (1976), no. 6, 644–654.

[15] Schneider B., 1996, "Applied Cryptography, second edition

[16] RFC 2412, 1998, OAKLEY Key determination protocol http://www.ietf.org/rfc/rfc2412.txt

[17] RFC 2409, 1998, IKE protocol, http://www.ietf.org/rfc/rfc2409.txt

[18] FIPS PUB 198, March 2002, The Keyed-Hash Message Authentication Code (HMAC)

[19] Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/HMAC (9 Ağustos, 2009)

[20] RFC2104, February 1997, HMAC, http://tools.ietf.org/html/rfc2104 (2 Kasım, 2009)

[21] Bellare, Mihir; Canetti, Ran; Krawczyk, Hugo, 1996, Keying Hash Functions for Message Authentication

[22] Preneel, Bart; van Oorschot, Paul C., 1995, MDx-MAC and Building Fast MACs from Hash Functions

[23] Preneel, Bart; van Oorschot, Paul C., 1995, On the Security of Two MAC Algorithms, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.8908, retrieved 2009-08-28

[24] Microsoft Developer Network Documantation, http://msdn.microsoft.com/enus/library/systemsecurity.cryptography.hmac.aspx

[25] J.Liberty, D.Herwitz, 2005, Programming ASP.NET: Building web Application and Services with ASP.NET,

[26] http://www.patentstorm.us/patents/7594274.html (10 Mayıs, 2009)

[27] Microsoft Developer Network Documantation, http://msdn.microsoft.com/en-us/library/ms893522.aspx (9 Ekim, 2009)

[28] Microsoft Developer Network Documantation, http://blogs.msdn.com/windowsmobile/archive/2006/01/09/510997.aspx (21 Ağustos.2009)

[29] Microsoft Developer Network Documantation, http://msdn.microsoft.com/en-us/library/ms972959.aspx (24 Eylül, 2009)

[30] Wikipedia, The Free Encyclopedia, http://tr.wikipedia.org/wiki/Uydu_interneti

[31] OpenVPN , Open Source VPN, http://openvpn.net/ (Erişim: 15 Kasım, 2009)

[32] RFC 3766,2004, Strength for public keys

[33] Performance Evaluation of Symmetric Encryption Algorithms, IJCSNS Interational Journal of Computer Science and Network Security, VOL.8 No.12, December 2008

[34] The Mono Project, http://www.mono-project.com/Cryptography (12 Temmuz, 2009)

[35] RFC2631, June 1999, Diffie Hellman key aggrement method

EKLER

EK1- PSK kullanımı (1.senaryo) paket örnekleri

Çizelge EK1.1. Gettime işlemi için kullanılan http request bilgisi ve bu bilginin açık hali

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST /PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61    rvice2/service.a
00032: 73 68 78 3F 47 65 74 54 69 6D 65 20 48 54 54 50    shx?GetTime HTTP
00048: 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79    /1.1..Content-Ty
00064: 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F    pe: application/
00080: 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65 6E    x-www-form-urlen
00096: 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69 65 3A 20 41    coded..Cookie: A
00112: 53 50 2E 4E 45 54 5F 53 65 73 73 69 6F 6E 49 64    SP.NET_SessionId
00128: 3D 66 32 62 78 74 6D 75 65 34 63 61 66 68 74 35    =f2bxtmue4cafht5
00144: 35 68 79 33 33 77 75 34 35 3B 20 70 61 74 68 3D    5hy33wu45; path=
00160: 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D 0A 43 6F 6E    /; HttpOnly..Con
00176: 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 30 0D 0A    tent-Length: 0..
00192: 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65 70    Connection: Keep
00208: 2D 41 6C 69 76 65 0D 0A 48 6F 73 74 3A 20 31 39    -Alive..Host: 19
00224: 32 2E 31 36 38 2E 35 35 2E 31 30 30 3A 38 38 0D    2.168.55.100:88.
00240: 0A 0D 0A                                           ...
POST /PositionService2/service.ashx?GetTime HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=f2bxtmue4cafht55hy33wu45; path=/; HttpOnly
Content-Length: 0
Connection: Keep-Alive
Host: 192.168.55.100:88
```

Çizelge EK1.2 GetTime Cevap Paketi (http response) ve bu bilginin açık hali

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 54 68 75 2C 20 32 34 20 44    .Date: Thu, 24 D
00032: 65 63 20 32 30 30 39 20 31 33 3A 33 37 3A 32 37    ec 2009 13:37:27
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 31 32 0D 0A 0D 0A 30 59 37 50 49    gth: 12....0Y7PI
00224: 63 5A 53 7A 41 67 3D                               cZSzAg=
HTTP/1.1 200 OK
Date: Thu, 24 Dec 2009 13:37:27 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 12
0Y7PIcZSzAg=
```

Çizelge EK1.3. Start DHKE Talep Paketi Örneği (header ve body)

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST /PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61    rvice2/service.a
00032: 73 68 78 3F 53 74 61 72 74 44 48 20 48 54 54 50    shx?StartDH HTTP
00048: 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79    /1.1..Content-Ty
00064: 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F    pe: application/
00080: 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65 6E    x-www-form-urlen
00096: 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69 65 3A 20 41    coded..Cookie: A
00112: 53 50 2E 4E 45 54 5F 53 65 73 73 69 6F 6E 49 64    SP.NET_SessionId
00128: 3D 66 32 62 78 74 6D 75 65 34 63 61 66 68 74 35    =f2bxtmue4cafht5
00144: 35 68 79 33 33 77 75 34 35 3B 20 70 61 74 68 3D    5hy33wu45; path=
00160: 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D 0A 43 6F 6E    /; HttpOnly..Con
00176: 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 32 38    tent-Length: 128
00192: 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65    ..Connection: Ke
00208: 65 70 2D 41 6C 69 76 65 0D 0A 48 6F 73 74 3A 20    ep-Alive..Host:
00224: 31 39 32 2E 31 36 38 2E 35 35 2E 31 30 30 3A 38    192.168.55.100:8
00240: 38 0D 0A 0D 0A                                     8....

00000: 44 74 32 66 74 4D 79 6E 46 68 48 37 50 6B 4C 47    Dt2ftMynFhH7PkLG
00016: 79 72 33 6C 48 43 6E 35 2F 42 4A 62 2F 51 54 4B    yr3lHCn5/BJb/QTK
00032: 4C 70 46 56 4B 51 4E 43 6E 51 57 30 43 72 48 78    LpFVKQNCnQW0CrHx
00048: 6F 64 67 43 37 61 51 66 4F 6D 52 31 39 6C 4B 47    odgC7aQfOmR19lKG
00064: 65 58 38 48 45 47 79 6C 6E 6A 75 4B 37 46 63 6F    eX8HEGylnjuK7Fco
00080: 44 42 58 38 50 37 33 79 5A 4E 2F 32 39 57 54 41    DBX8P73yZN/29WTA
00096: 54 49 32 6E 6E 74 30 53 62 34 48 57 30 4E 2B 4A    TI2nnt0Sb4HW0N+J
00112: 36 76 58 63 6C 78 41 75 36 75 68 57 71 66 54 55    6vXclxAu6uhWqfTU
```

Çizelge EK1.4. Start DHKE Cevap Paketi Örneği

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 54 68 75 2C 20 32 34 20 44    .Date: Thu, 24 D
00032: 65 63 20 32 30 30 39 20 31 33 3A 33 37 3A 34 34    ec 2009 13:37:44
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 35 37 36 0D 0A 0D 0A 33 68 44 6F    gth: 576....3hDo
00224: 4F 30 79 6C 6B 35 33 6B 6E 61 62 6E 30 41 4C 79    O0ylk53knabn0ALy
00240: 46 6E 6D 7A 36 6F 4A 5A 4B 6E 63 4E 50 71 41 73    Fnmz6oJZKncNPqAs
00256: 33 4A 53 73 32 31 5A 55 43 6E 6E 48 79 53 4D 55    3JSs21ZUCnnHySMU
00272: 74 5A 65 79 44 44 59 68 77 2F 37 63 7A 6E 53 2F    tZeyDDYhw/7cznS/
00288: 6D 66 76 42 71 39 55 76 6E 6F 47 75 6F 72 33 56    mfvBq9UvnoGuor3V
00304: 56 37 34 41 38 71 59 6B 30 46 30 4E 56 30 66 76    V74A8qYk0F0NV0fv
00320: 47 59 4A 43 78 41 74 64 70 47 4A 5A 68 42 67 66    GYJCxAtdpGJZhBgf
00336: 79 31 6B 34 79 61 6B 6F 7A 59 6D 46 72 6C 6A 46    y1k4yakozYmFrljF
00352: 77 4B 51 30 58 73 32 69 52 4B 79 45 6D 6C 31 62    wKQ0Xs2iRKyEml1b
00368: 57 4A 77 65 4E 32 41 79 47 66 43 5A 58 35 33 43    WJweN2AyGfCZX53C
00384: 5A 4E 67 65 4A 55 77 75 63 34 59 6E 42 51 6C 39    ZNgeJUwuc4YnBQl9
00400: 38 63 79 30 62 79 31 7A 4E 44 76 6C 63 46 76 30    8cy0by1zNDvlcFv0
00416: 6A 4B 2F 42 4B 45 51 75 65 35 39 7A 6D 4C 36 36    jK/BKEQue59zmL66
```

```
00432: 78 41 47 64 5A 75 34 59 68 53 37 56 69 7A 6F 30    xAGdZu4YhS7Vizo0
00448: 2F 6D 32 6F 39 6F 32 6D 50 75 57 72 76 30 63 52    /m2o9o2mPuWrv0cR
00464: 7A 7A 42 59 41 74 63 78 6A 35 35 37 30 32 67 65    zzBYAtcxj55702ge
00480: 48 37 33 44 63 44 72 39 47 57 6D 59 41 55 62 62    H73DcDr9GWmYAUbb
00496: 4B 77 64 6D 54 30 6B 62 4E 33 6A 58 38 2F 61 4B    KwdmT0kbN3jX8/aK
00512: 41 56 48 69 7A 62 63 4B 7A 38 43 74 5A 59 56 4F    AVHizbcKz8CtZYVO
00528: 75 77 47 6A 68 43 4E 57 4A 35 74 62 51 64 6A 34    uwGjhCNWJ5tbQdj4
00544: 6D 6D 6D 76 65 52 63 33 46 31 54 36 32 48 32 50    mmmveRc3F1T62H2P
00560: 4F 78 32 67 41 4A 30 49 4E 51 4C 79 56 71 63 4D    Ox2gAJ0INQLyVqcM
00576: 2B 65 6A 43 75 54 31 45 48 76 53 6C 69 72 42 46    +ejCuT1EHvSlirBF
00592: 66 6B 61 34 4E 63 59 6E 50 32 73 31 77 47 59 75    fka4NcYnP2s1wGYu
00608: 6C 76 4E 79 68 47 37 68 55 4D 78 33 37 33 66 42    lvNyhG7hUMx373fB
00624: 67 4A 6C 34 2F 2B 57 74 75 33 2F 56 48 35 41 69    gJl4/+Wtu3/VH5Ai
00640: 4B 69 76 38 58 48 41 57 75 59 4B 71 77 2B 79 49    Kiv8XHAWuYKqw+yI
00656: 4B 6B 6D 62 30 6C 42 77 51 6C 31 2F 4D 57 53 52    Kkmb0lBwQl1/MWSR
00672: 73 56 4E 71 68 48 71 36 78 64 70 39 64 78 73 5A    sVNqhHq6xdp9dxsZ
00688: 36 51 2B 49 2B 47 56 5A 31 78 41 35 72 6C 77 43    6Q+I+GVZ1xA5rlwC
00704: 72 78 36 6A 63 75 78 2F 54 46 44 44 51 2B 4E 78    rx6jcux/TFDDQ+Nx
00720: 46 30 38 56 71 56 63 66 69 67 55 6B 65 58 4D 34    F08VqVcfigUkeXM4
00736: 43 56 70 65 4E 63 51 67 52 36 45 2F 72 78 56 65    CVpeNcQgR6E/rxVe
00752: 5A 6F 45 66 76 68 6B 55 35 50 37 32 36 71 44 50    ZoEfvhkU5P726qDP
00768: 55 77 71 67 6F 2B 77 45 69 4B 47 4E 76 56 4C 57    Uwqgo+wEiKGNvVLW
00784: 68 51 4F 71 2F 48 51 4A 50 42 5A 6E                hQOq/HQJPBZn
```

Çizelge EK1.5. Store DH Paketi Örneği

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST/PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61   rvice2/service.a
00032: 73 68 78 3F 53 74 6F 72 65 44 48 20 48 54 54 50    shx?StoreDHHTTP
00048: 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E 74 2D 54 79    /1.1..ContentTy
00064: 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E 2F    pe:application/
00080: 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65 6E    x-www-formurlen
00096: 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69 65 3A 20 41    coded..Cookie:A
00112: 53 50 2E 4E 45 54 5F 53 65 73 73 69 6F 6E 49 64   SP.NET_SessionId
00128: 3D 66 6E 6F 34 74 6D 35 35 67 63 62 76 63 34 65   =fno4tm55gcbvc4e
00144: 71 71 68 35 78 67 70 35 35 3B 20 70 61 74 68 3D    qqh5xgp55;path=
00160: 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D 0A 43 6F 6E    /;HttpOnly..Con
00176: 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 39 32    tent-Length:192
00192: 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65    ..Connection:Ke
00208: 65 70 2D 41 6C 69 76 65 0D 0A 45 78 70 65 63 74    epAlive..Expect
00224: 3A 20 31 30 30 2D 63 6F 6E 74 69 6E 75 65 0D 0A    : 100continue..
00240: 48 6F 73 74 3A 20 31 39 32 2E 31 36 38 2E 35 35    Host:192.168.55
00256: 2E 31 30 30 3A 38 38 0D 0A 0D 0A                   .100:88....
```

Çizelge EK1.6. Store DH Cevap Paketi Örneği

```
00000: 71 73 69 77 77 4A 30 4F 5A 5A 4A 78 67 59 4D 7A   qsiwwJ0OZZJxgYMz
00016: 68 36 70 2F 39 43 49 46 77 49 47 6E 30 53 61 4E   h6p/9CIFwIGn0SaN
00032: 2B 6F 76 37 35 4E 70 36 45 77 33 75 44 54 62 47   +ov75Np6Ew3uDTbG
00048: 2B 55 52 5A 49 34 45 73 74 77 76 50 39 63 41 76   +URZI4EstwvP9cAv
00064: 38 4A 71 6F 41 39 62 47 76 4B 6C 78 62 49 62 30   8JqoA9bGvKlxbIb0
00080: 6F 51 68 43 75 54 62 52 55 4F 69 66 55 48 66 52   oQhCuTbRUOifUHfR
00096: 44 66 39 47 59 78 68 61 63 66 39 61 4E 37 7A 75   Df9GYxhacf9aN7zu
00112: 78 72 31 58 68 48 4B 78 6B 44 50 76 62 6A 65 34   xr1XhHKxkDPvbje4
00128: 43 43 65 66 38 70 45 45 45 74 6E 6B 79 2B 39 30   CCef8pEEEtnky+90
00144: 67 68 4A 6C 71 74 56 66 6D 69 54 59 6F 59 58 33   ghJlqtVfmiTYoYX3
00160: 67 6F 46 6E 4F 74 54 41 77 50 48 6F 49 59 47 58   goFnOtTAwPHoIYGX
00176: 6E 39 37 63 58 46 4F 33 75 75 49 4D 44 6D 44 36   n97cXFO3uuIMDmD6
```

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200OK.
00016: 0A 44 61 74 65 3A 20 53 61 74 2C 20 31 32 20 44    .Date: Sat, 12D
00032: 65 63 20 32 30 30 39 20 30 39 3A 32 31 3A 35 30    ec 200909:21:50
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server:Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoftIIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By:A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..XAspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version:2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..CacheContr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol:private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type:text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html;charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..ContentLen
00208: 67 74 68 3A 20 36 34 0D 0A 0D 0A 79 75 77 68 4F    gth:64....yuwhO
00224: 42 2B 2B 6C 55 45 4B 7A 78 58 57 47 41 46 57 35   B++lUEKzxXWGAFW5
00240: 43 56 6A 44 48 35 74 74 79 56 51 44 74 5A 63 4C   CVjDH5ttyVQDtZcL
00256: 5A 76 45 6C 57 63 78 77 71 42 76 6A 4D 61 6A 5A   ZvElWcxwqBvjMajZ
00272: 6E 39 2B 4F 4F 6B 4D 5A 38 64 43                   n9+OOkMZ8dC
```

Çizelge EK1.7. Register Talep Paketi Örneği ve paketin açık hali

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST /PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61    rvice2/service.a
00032: 73 68 78 3F 52 65 67 69 73 74 65 72 20 48 54 54    shx?Register HTT
00048: 50 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E 74 2D 54    P/1.1..Content-T
00064: 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E    ype: application
00080: 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65    /x-www-form-urle
00096: 6E 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69 65 3A 20    ncoded..Cookie:
00112: 41 53 50 2E 4E 45 54 5F 53 65 73 73 69 6F 6E 49    ASP.NET_SessionI
00128: 64 3D 66 32 62 78 74 6D 75 65 34 63 61 66 68 74    d=f2bxtmue4cafht
00144: 35 35 68 79 33 33 77 75 34 35 3B 20 70 61 74 68    55hy33wu45; path
00160: 3D 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D 0A 43 6F    =/; HttpOnly..Co
00176: 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 35    ntent-Length: 15
00192: 36 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B    6..Connection: K
00208: 65 65 70 2D 41 6C 69 76 65 0D 0A 45 78 70 65 63    eep-Alive..Expec
00224: 74 3A 20 31 30 30 2D 63 6F 6E 74 69 6E 75 65 0D    t: 100-continue.
00240: 0A 48 6F 73 74 3A 20 31 39 32 2E 31 36 38 2E 35    .Host: 192.168.5
00256: 35 2E 31 30 30 3A 38 38 0D 0A 0D 0A                5.100:88....
00000: 42 63 35 75 57 6E 75 4B 62 65 65 76 4E 6D 4E 6B    Bc5uWnuKbeevNmNk
00016: 53 36 67 65 45 34 78 48 64 35 53 6B 72 4E 6D 6A    S6geE4xHd5SkrNmj
00032: 35 77 68 68 38 57 59 6A 54 6C 45 38 55 58 69 63    5whh8WYjTlE8UXic
00048: 64 74 6E 77 6D 34 68 36 69 37 6F 4D 76 69 70 6D    dtnwm4h6i7oMvipm
00064: 6C 35 44 67 49 50 44 71 47 44 47 4C 30 47 4A 54    l5DgIPDqGDGL0GJT
00080: 61 4B 4F 43 75 36 64 67 42 64 73 45 57 38 39 6A    aKOCu6dgBdsEW89j
00096: 57 56 5A 6F 68 46 69 6B 4F 69 70 51 52 2B 65 6D    WVZohFikOipQR+em
00112: 74 76 36 38 59 34 59 43 32 67 62 64 4E 74 35 47    tv68Y4YC2gbdNt5G
00128: 2F 57 41 5A 42 68 6D 48 59 59 61 77 71 4E 66 71    /WAZBhmHYYawqNfq
00144: 4A 74 73 4E 4E 68 4E 4B 62 4A 67 3D                JtsNNhNKbJg=
POST /PositionService2/service.ashx?Register HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=f2bxtmue4cafht55hy33wu45; path=/; HttpOnly
Content-Length: 156
Connection: Keep-Alive
Expect: 100-continue
Host: 192.168.55.100:88
Bc5uWnuKbeevNmNkS6geE4xHd5SkrNmj5whh8WYjTlE8UXicdtnwm4h6i7oMvipml5DgIPDqGDGL0GJ
TaKOCu6dgBdsEW89jWVZohFikOipQR+emtv68Y4YC2gbdNt5G/WAZBhmHYYawqNfqJtsNNhNKbJg=
```

Çizelge EK1.8. PSK ile şifrelenmiş No Session paket örneği ve paketin açık hali

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 46 72 69 2C 20 32 35 20 44    .Date: Fri, 25 D
00032: 65 63 20 32 30 30 39 20 30 36 3A 34 37 3A 31 37    ec 2009 06:47:17
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 36 34 0D 0A 0D 0A 4F 76 77 31 31    gth: 64....Ovw11
00224: 4D 2B 32 52 79 72 48 56 46 6D 61 37 64 55 34 67    M+2RyrHVFma7dU4g
00240: 4A 7A 51 67 4F 69 74 44 46 4A 4F 48 70 78 4F 6D    JzQgOitDFJOHpxOm
00256: 37 6F 53 73 32 73 36 76 77 75 57 7A 36 2F 70 6C    7oSs2s6vwuWz6/pl
00272: 2B 59 58 78 41 69 66 79 48 6B 39                   +YXxAifyHk9

HTTP/1.1 200 OK
Date: Fri, 25 Dec 2009 06:47:17 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 64

Ovw11M+2RyrHVFma7dU4gJzQgOitDFJOHpxOm7oSs2s6vwuWz6/pl+YXxAifyHk9
```

Çizelge EK1.9. Device Id'nin döndüğü paket ve paketin açık hali

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 46 72 69 2C 20 32 35 20 44    .Date: Fri, 25 D
00032: 65 63 20 32 30 30 39 20 30 36 3A 34 38 3A 30 36    ec 2009 06:48:06
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 39 32 0D 0A 0D 0A 44 46 62 54 61    gth: 92....DFbTa
00224: 52 50 71 4C 75 77 4A 49 7A 72 55 4A 39 69 69 74    RPqLuwJIzrUJ9iit
00240: 32 4D 6E 72 6D 66 66 78 59 51 63 32 42 2F 38 6D    2MnrmffxYQc2B/8m
00256: 65 6A 53 66 66 73 6A 34 4B 2F 46 73 32 54 4F 5A    ejSffsj4K/Fs2TOZ
00272: 62 4D 55 4F 72 79 38 51 51 46 64 33 4A 46 48 70    bMUOry8QQFd3JFHp
00288: 55 32 64 41 66 47 42 34 4A 69 49 50 38 74 32 78    U2dAfGB4JiIP8t2x
00304: 66 57 43 38 59 41 3D                               fWC8YA=
```

```
HTTP/1.1 200 OK
Date: Fri, 25 Dec 2009 06:48:06 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 92
DFbTaRPqLuwJIzrUJ9iit2MnrmffxYQc2B/8mejSffsj4K/Fs2TOZbMUOry8QQFd3JFHpU2dAfGB4Ji
IP8t2xfWC8YA=
```

Çizelge EK1.10. Konum Paketi Örneği ve paketin açık hali

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST /PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61    rvice2/service.a
00032: 73 68 78 3F 50 6F 73 69 74 69 6F 6E 20 48 54 54    shx?Position HTT
00048: 50 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E 74 2D 54    P/1.1..Content-T
00064: 79 70 65 3A 20 61 70 70 6C 69 63 61 74 69 6F 6E    ype: application
00080: 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75 72 6C 65    /x-www-form-urle
00096: 6E 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69 65 3A 20    ncoded..Cookie:
00112: 41 53 50 2E 4E 45 54 5F 53 65 73 73 69 6F 6E 49    ASP.NET_SessionI
00128: 64 3D 66 32 62 78 74 6D 75 65 34 63 61 66 68 74    d=f2bxtmue4cafht
00144: 35 35 68 79 33 33 77 75 34 35 3B 20 70 61 74 68    55hy33wu45; path
00160: 3D 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D 0A 43 6F    =/; HttpOnly..Co
00176: 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 31 35    ntent-Length: 15
00192: 36 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B    6..Connection: K
00208: 65 65 70 2D 41 6C 69 76 65 0D 0A 45 78 70 65 63    eep-Alive..Expec
00224: 74 3A 20 31 30 30 2D 63 6F 6E 74 69 6E 75 65 0D    t: 100-continue.
00240: 0A 48 6F 73 74 3A 20 31 39 32 2E 31 36 38 2E 35    .Host: 192.168.5
00256: 35 2E 31 30 30 3A 38 38 0D 0A 0D 0A                5.100:88....
00000: 39 63 6B 68 4E 49 79 52 33 6D 38 4D 46 73 41 39    9ckhNIyR3m8MFsA9
00016: 76 78 33 61 6A 45 38 59 35 35 2F 4B 6F 54 59 59    vx3ajE8Y55/KoTYY
00032: 73 6D 50 43 49 42 32 4B 34 76 6A 66 31 4D 43 6A    smPCIB2K4vjf1MCj
00048: 58 4A 79 6D 38 49 71 59 42 73 79 42 37 67 50 30    XJym8IqYBsyB7gP0
00064: 4E 48 4B 77 6A 32 51 71 79 50 64 6D 39 31 2F 2B    NHKwj2QqyPdm91/+
00080: 38 41 71 62 67 34 30 6C 43 70 74 30 5A 69 57 4B    8Aqbg40lCpt0ZiWK
00096: 32 36 7A 61 49 61 33 30 54 62 31 64 2B 42 2B 41    26zaIa30Tb1d+B+A
00112: 57 52 42 36 4C 48 71 7A 6C 72 42 31 53 69 36 30    WRB6LHqzlrB1Si60
00128: 33 53 6B 67 48 53 35 33 6C 43 75 6A 59 41 6A 45    3SkgHS53lCujYAjE
00144: 37 64 58 74 75 6E 42 33 7A 67 55 3D                7dXtunB3zgU=
POST /PositionService2/service.ashx?Position HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Cookie: ASP.NET_SessionId=f2bxtmue4cafht55hy33wu45; path=/; HttpOnly
Content-Length: 156
Connection: Keep-Alive
Expect: 100-continue
Host: 192.168.55.100:88
9ckhNIyR3m8MFsA9vx3ajE8Y55/KoTYYsmPCIB2K4vjf1MCjXJym8IqYBsyB7gP0NHKwj2QqyPdm91/
+8Aqbg40lCpt0ZiWK26zaIa30Tb1d+B+AWRB6LHqzlrB1Si603SkgHS53lCujYAjE7dXtunB3zgU=
```

Çizelge EK1.11. Konum Cevap Paketi Örneği ve paketin açık hali

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 54 68 75 2C 20 32 34 20 44    .Date: Thu, 24 D
00032: 65 63 20 32 30 30 39 20 31 33 3A 33 36 3A 30 32    ec 2009 13:36:02
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
```

```
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 39 32 0D 0A 0D 0A 67 65 53 4F 73    gth: 92....geSOs
00224: 43 4C 4F 74 77 49 31 56 47 6E 4A 79 73 57 66 56    CLOtwI1VGnJysWfV
00240: 55 47 69 67 64 36 4A 67 72 66 47 59 32 75 45 69    UGigd6JgrfGY2uEi
00256: 43 49 70 48 6F 45 51 6C 58 6D 52 5A 71 50 62 6A    CIpHoEQlXmRZqPbj
00272: 31 75 6D 34 63 4A 57 34 6B 4B 47 69 6B 70 47 53    1um4cJW4kKGikpGS
00288: 74 58 58 44 62 62 39 65 65 51 51 59 39 56 53 70    tXXDbb9eeQQY9VSp
00304: 77 4B 5A 74 61 51 3D                               wKZtaQ=
HTTP/1.1 200 OK
Date: Thu, 24 Dec 2009 13:36:02 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-AspNet-Version: 2.0.50727
Cache-Control: private
Content-Type: text/html; charset=utf-8
Content-Length: 92
```

EK2- RSA kullanımı (2.senaryo) paket örnekleri

Çizelge EK2.1. StartDHCert Request Paketi

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST /PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61    rvice2/service.a
00032: 73 68 78 3F 53 74 61 72 74 44 48 43 65 72 74 20    shx?StartDHCert
00048: 48 54 54 50 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E    HTTP/1.1..Conten
00064: 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74    t-Type: applicat
00080: 69 6F 6E 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75    ion/x-www-form-u
00096: 72 6C 65 6E 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69    rlencoded..Cooki
00112: 65 3A 20 41 53 50 2E 4E 45 54 5F 53 65 73 73 69    e: ASP.NET_Sessi
00128: 6F 6E 49 64 3D 66 32 62 78 74 6D 75 65 34 63 61    onId=f2bxtmue4ca
00144: 66 68 74 35 35 68 79 33 33 77 75 34 35 3B 20 70    fht55hy33wu45; p
00160: 61 74 68 3D 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D    ath=/; HttpOnly.
00176: 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A    .Content-Length:
00192: 20 33 34 38 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E     348..Connection
00208: 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 45 78    : Keep-Alive..Ex
00224: 70 65 63 74 3A 20 31 30 30 2D 63 6F 6E 74 69 6E    pect: 100-contin
00240: 75 65 0D 0A 48 6F 73 74 3A 20 31 39 32 2E 31 36    ue..Host: 192.16
00256: 38 2E 35 35 2E 31 30 30 3A 38 38 0D 0A 0D 0A       8.55.100:88....
00000: 41 77 41 41 41 42 46 68 37 6A 53 6F 72 67 47 72    AwAAABFh7jSorgGr
00016: 79 5A 69 6A 72 48 2B 36 63 35 4A 44 53 37 51 64    yZijrH+6c5JDS7Qd
00032: 68 42 70 66 63 53 75 38 32 69 79 61 7A 4D 62 36    hBpfcSu82iyazMb6
00048: 4D 63 46 68 59 33 79 62 56 4D 62 59 66 49 39 52    McFhY3ybVMbYfI9R
00064: 4F 49 4B 74 45 33 32 53 34 69 61 6E 7A 70 6F 53    OIKtE32S4ianzpoS
00080: 47 55 69 31 78 51 4C 41 53 45 76 68 74 77 66 54    GUi1xQLASEvhtwfT
00096: 7A 2F 35 62 75 68 51 34 4D 78 37 44 45 45 68 2B    z/5buhQ4Mx7DEEh+
00112: 63 63 31 75 7A 63 4F 74 66 4F 78 6D 59 76 73 42    cc1uzcOtfOxmYvsB
00128: 47 51 49 6A 39 72 62 50 56 4D 5A 2F 79 70 33 4E    GQIj9rbPVMZ/yp3N
00144: 52 54 38 4A 36 70 58 2B 77 6A 4D 36 6C 35 7A 31    RT8J6pX+wjM6l5z1
00160: 78 39 54 65 69 4E 32 42 59 5A 6C 61 4F 31 6E 45    x9TeiN2BYZlaO1nE
00176: 57 38 43 58 74 4A 47 38 54 72 47 50 73 63 69 67    W8CXtJG8TrGPscig
00192: 36 79 52 77 52 53 66 43 56 53 58 72 4B 70 72 43    6yRwRSfCVSXrKprC
00208: 6C 46 79 72 6D 49 56 65 71 59 64 6C 63 6A 4B 50    lFyrmIVeqYdlcjKP
00224: 79 59 35 42 64 4E 42 37 39 46 51 58 6B 73 39 37    yY5BdNB79FQXks97
00240: 70 46 71 50 4F 34 31 68 35 35 30 6A 52 4B 51 68    pFqPO41h550jRKQh
00256: 6A 4B 68 42 2B 69 39 39 68 43 70 6F 6E 57 49 7A    jKhB+i99hCponWIz
00272: 59 78 36 6B 66 6C 33 41 62 47 67 37 77 42 53 43    Yx6kfl3AbGg7wBSC
00288: 46 66 4B 34 58 4D 67 79 53 78 46 71 52 35 36 42    FfK4XMgySxFqR56B
00304: 4D 4C 4D 38 4D 55 4B 55 39 5A 53 67 6A 52 55 44    MLM8MUKU9ZSgjRUD
00320: 6F 45 78 43 4E 78 38 67 65 69 6E 66 49 44 43 34    oExCNx8geinfIDC4
00336: 75 66 44 4C 47 78 45 4F 6F 58 38 3D                ufDLGxEOoX8=
```

Çizelge EK2.2. StartDHCert Cevabı

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 4D 6F 6E 2C 20 30 34 20 4A    .Date: Mon, 04 J
00032: 61 6E 20 32 30 31 30 20 30 35 3A 34 32 3A 34 39    an 2010 05:42:49
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
```

```
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 38 35 36 0D 0A 0D 0A 41 79 35 4C    gth: 856....Ay5L
00224: 2B 6D 4C 73 75 58 61 6E 56 50 4E 2B 5A 63 41 39    +mLsuXanVPN+ZcA9
00240: 4C 53 45 2B 7A 36 36 5A 63 34 57 4A 59 34 37 45    LSE+z66Zc4WJY47E
00256: 4B 6F 48 51 73 70 47 63 36 2B 71 2F 45 4A 48 57    KoHQspGc6+q/EJHW
00272: 38 6F 4F 76 71 77 4A 49 37 55 7A 50 4A 46 73 4F    8oOvqwJI7UzPJFsO
00288: 59 66 65 6F 53 4A 76 55 6E 31 61 75 2B 73 38 68    YfeoSJvUn1au+s8h
00304: 57 4B 77 69 4D 57 69 74 55 56 32 54 71 43 4C 2B    WKwiMWitUV2TqCL+
00320: 52 56 50 55 79 5A 74 6A 68 78 30 66 4F 5A 36 2B    RVPUyZtjhx0fOZ6+
00336: 59 4A 67 2B 72 6D 31 77 38 78 54 4D 61 41 65 39    YJg+rm1w8xTMaAe9
00352: 57 47 73 70 2F 48 64 79 78 6C 64 45 68 6D 6F 79    WGsp/Hdyx1dEhmoy
00368: 75 6A 68 48 33 30 75 7A 5A 30 6E 4D 6D 69 6F 37    ujhH30uzZ0nMmio7
00384: 4A 44 59 68 72 6D 38 4F 6A 2B 52 6E 78 45 43 35    JDYhrm8Oj+RnxEC5
00400: 66 4E 79 54 37 53 72 70 5A 6D 59 31 33 4F 39 4A    fNyT7SrpZmY13O9J
00416: 77 4B 72 2B 58 35 58 49 65 65 53 38 35 77 45 62    wKr+X5XIeeS85wEb
00432: 6E 38 35 6E 67 48 43 4F 4B 4B 59 73 36 76 79 48    n85ngHCOKKYs6vyH
00448: 6D 66 76 7A 6B 77 64 38 6B 47 43 35 53 75 2B 64    mfvzkwd8kGC5Su+d
00464: 76 55 48 31 78 78 4E 42 43 61 75 46 57 4B 6A 72    vUH1xxNBCauFWKjr
00480: 6F 2B 66 47 4F 69 2B 33 35 76 49 6B 4F 74 76 46    o+fGOi+35vIkOtvF
00496: 4B 79 67 32 35 4A 79 4B 59 32 4A 38 53 53 6A 49    Kyg25JyKY2J8SSjI
00512: 5A 65 34 41 43 58 4E 62 77 52 2F 49 66 65 2B 73    Ze4ACXNbwR/Ife+s
00528: 2B 7A 69 62 68 33 58 46 54 4C 41 4B 54 6E 50 2B    +zibh3XFTLAKTnP+
00544: 56 4F 4C 6A 47 63 50 71 64 36 2B 6F 4E 31 55 56    VOLjGcPqd6+oN1UV
00560: 6E 32 78 6B 38 37 50 41 45 54 69 76 4C 41 48 4E    n2xk87PAETivLAHN
00576: 4F 37 63 31 75 43 38 4F 59 55 61 36 7A 5A 77 52    O7c1uC8OYUa6zZwR
00592: 4D 79 70 57 53 4C 4F 38 2B 51 2F 64 35 6D 57 31    MypWSLO8+Q/d5mW1
00608: 63 39 75 39 6F 4C 55 59 30 6A 6A 41 6D 54 38 34    c9u9oLUY0jjAmT84
00624: 57 66 6D 6C 4A 73 2F 63 65 4A 33 41 67 70 66 69    WfmlJs/ceJ3Agpfi
00640: 6F 63 30 44 72 41 34 5A 6C 72 4B 42 4F 56 72 30    oc0DrA4ZlrKBOVr0
00656: 73 76 6D 78 6A 65 6C 30 62 6D 31 59 4D 38 32 6A    svmxjel0bm1YM82j
00672: 67 45 66 54 79 53 31 6F 78 31 62 73 66 4F 54 43    gEfTyS1ox1bsfOTC
00688: 34 76 52 71 71 31 61 56 5A 6F 72 6F 78 4B 63 71    4vRqq1aVZoroxKcq
00704: 73 61 6B 36 6E 57 38 38 41 70 53 4E 52 37 78 6D    sak6nW88ApSNR7xm
00720: 61 42 70 7A 62 39 53 45 72 55 4B 66 74 6C 77 79    aBpzb9SErUKftlwy
00736: 36 46 54 74 6C 33 4E 2B 75 72 54 41 78 4F 63 6E    6FTtl3N+urTAxOcn
00752: 44 38 43 44 48 4E 63 61 48 64 63 63 71 42 49 48    D8CDHNcaHdccqBIH
00768: 58 56 7A 62 45 4A 50 76 66 6C 75 54 32 52 51 7A    XVzbEJPvfluT2RQz
00784: 2B 77 4D 32 61 7A 78 6D 32 6E 74 65 44 74 35 36    +wM2azxm2nteDt56
00800: 38 58 55 33 38 69 57 6B 30 35 6F 37 50 73 7A 6D    8XU38iWk05o7Pszm
00816: 59 5A 69 74 33 38 61 75 41 56 64 64 6C 79 62 58    YZit38auAVddlybX
00832: 57 43 6F 6F 44 63 6D 55 65 50 6D 33 4B 42 55 34    WCooDcmUePm3KBU4
00848: 45 4D 50 67 42 55 75 63 2B 4C 31 34 51 66 54 46    EMPgBUuc+L14QfTF
00864: 6C 2F 6F 77 49 58 66 30 42 79 76 4C 64 33 44 71    l/owIXf0ByvLd3Dq
00880: 55 6F 35 61 58 6B 70 49 68 69 76 35 34 34 4C 4F    Uo5aXkpIhiv544LO
00896: 47 72 4B 6B 37 59 57 46 57 62 59 6D 79 44 67 33    GrKk7YWFWbYmyDg3
00912: 4B 6D 53 44 62 6E 4D 61 70 36 6F 73 7A 6B 4A 32    KmSDbnMap6oszkJ2
00928: 50 65 7A 38 42 68 4D 71 6E 35 72 6F 2F 30 51 77    Pez8BhMqn5ro/0Qw
00944: 6F 4E 48 77 6F 6D 48 6B 6B 75 4F 52 48 75 64 49    oNHwomHkkuORHudI
00960: 74 64 74 70 46 65 2B 35 35 73 63 7A 67 4C 33 2B    tdtpFe+55sczgL3+
00976: 4D 4F 76 55 32 74 39 32 34 6F 48 6C 38 70 71 65    MOvU2t924oHl8pqe
00992: 55 39 6A 58 6E 31 30 46 30 76 36 6D 6B 59 4B 65    U9jXn10F0v6mkYKe
01008: 6F 64 36 64 4E 31 63 38 79 5A 6C 65 7A 6D 48 61    od6dN1c8yZlezmHa
01024: 55 4E 48 46 4C 70 2B 7A 74 49 33 50 62 70 37 4B    UNHFLp+ztI3Pbp7K
01040: 6F 44 6D 4C 66 2F 6B 4E 56 67 38 68 4F 68 4E 70    oDmLf/kNVg8hOhNp
01056: 41 4E 52 7A 37 33 5A 34 36 53 73 41 5A 52 52 75    ANRz73Z46SsAZRRu
01072: 4C 44 77 3D                                        LDw=
```

Çizelge EK2.3. StoreDHCert Request Paketi

```
00000: 50 4F 53 54 20 2F 50 6F 73 69 74 69 6F 6E 53 65    POST /PositionSe
00016: 72 76 69 63 65 32 2F 73 65 72 76 69 63 65 2E 61    rvice2/service.a
00032: 73 68 78 3F 53 74 6F 72 65 44 48 43 65 72 74 20    shx?StoreDHCert
00048: 48 54 54 50 2F 31 2E 31 0D 0A 43 6F 6E 74 65 6E    HTTP/1.1..Conten
00064: 74 2D 54 79 70 65 3A 20 61 70 70 6C 69 63 61 74    t-Type: applicat
00080: 69 6F 6E 2F 78 2D 77 77 77 2D 66 6F 72 6D 2D 75    ion/x-www-form-u
00096: 72 6C 65 6E 63 6F 64 65 64 0D 0A 43 6F 6F 6B 69    rlencoded..Cooki
00112: 65 3A 20 41 53 50 2E 4E 45 54 5F 53 65 73 73 69    e: ASP.NET_Sessi
00128: 6F 6E 49 64 3D 66 32 62 78 74 6D 75 65 34 63 61    onId=f2bxtmue4ca
00144: 66 68 74 35 35 68 79 33 33 77 75 34 35 3B 20 70    fht55hy33wu45; p
00160: 61 74 68 3D 2F 3B 20 48 74 74 70 4F 6E 6C 79 0D    ath=/; HttpOnly.
00176: 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A    .Content-Length:
00192: 20 35 32 30 0D 0A 43 6F 6E 6E 65 63 74 69 6F 6E     520..Connection
00208: 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 45 78    : Keep-Alive..Ex
00224: 70 65 63 74 3A 20 31 30 30 2D 63 6F 6E 74 69 6E    pect: 100-contin
00240: 75 65 0D 0A 48 6F 73 74 3A 20 31 39 32 2E 31 36    ue..Host: 192.16
00256: 38 2E 35 35 2E 31 30 30 3A 38 38 0D 0A 0D 0A       8.55.100:88....
00000: 41 77 41 41 41 45 38 66 7A 41 69 38 43 5A 4F 38    AwAAAE8fzAi8CZO8
00016: 2B 6D 45 7A 78 78 54 4F 31 70 68 34 36 61 6D 47    +mEzxxTO1ph46amG
00032: 51 4E 35 65 6B 67 31 31 46 69 78 47 75 44 61 37    QN5ekg11FixGuDa7
00048: 57 69 44 6D 46 47 57 43 31 34 62 37 50 48 48 70    WiDmFGWC14b7PHHp
00064: 66 56 6E 34 73 62 46 66 4C 42 6B 68 42 63 39 50    fVn4sbFfLBkhBc9P
00080: 61 4A 51 43 76 78 61 64 2B 34 46 44 64 72 68 79    aJQCvxad+4FDdrhy
00096: 72 4C 6A 41 77 33 68 59 58 70 70 75 45 38 64 32    rLjAw3hYXppuE8d2
00112: 63 45 69 70 59 70 30 72 43 73 50 52 4C 74 53 66    cEipYp0rCsPRLtSf
00128: 52 49 6C 75 7A 6D 55 55 72 74 33 6A 6D 79 44 72    RIluzmUUrt3jmyDr
00144: 50 49 73 47 44 4F 41 6F 75 7A 61 6B 44 73 6A 48    PIsGDOAouzakDsjH
00160: 47 68 68 70 67 79 43 73 39 52 50 69 6F 30 72 6E    GhhpgyCs9RPio0rn
00176: 46 4B 64 61 6F 78 65 6A 61 4B 6C 6D 5A 44 46 4F    FKdaoxejaKlmZDFO
00192: 6A 44 4C 2B 41 61 2F 7A 71 72 57 7A 30 6A 78 30    jDL+Aa/zqrWz0jx0
00208: 6D 78 6F 6A 75 50 34 52 59 75 33 37 38 48 36 6C    mxojuP4RYu378H6l
00224: 54 2B 76 61 46 6B 6D 62 49 78 2F 55 74 37 51 4C    T+vaFkmbIx/Ut7QL
00240: 79 66 4C 37 74 38 6E 68 66 74 73 52 4D 57 6B 63    yfL7t8nhftsRMWkc
00256: 30 51 47 47 4C 31 79 32 56 44 68 65 69 48 56 41    0QGGL1y2VDheiHVA
00272: 54 6E 79 65 6B 4B 41 43 55 36 64 63 64 42 33 59    TnyekKACU6dcdB3Y
00288: 48 57 38 31 6F 56 45 77 42 67 68 43 74 49 70 4F    HW81oVEwBghCtIpO
00304: 4D 6C 51 41 45 4B 7A 6B 37 73 35 49 36 4D 63 58    MlQAEKzk7s5I6McX
00320: 6F 58 33 62 65 61 46 34 33 70 34 52 56 57 45 73    oX3beaF43p4RVWEs
00336: 55 4B 66 64 68 4D 4D 31 33 34 35 74 47 48 4B 2F    UKfdhMM1345tGHK/
00352: 6A 79 2F 77 70 63 6E 38 57 7A 66 62 36 43 5A 54    jy/wpcn8Wzfb6CZT
00368: 58 6C 6D 63 37 44 4C 4A 4A 41 41 72 2F 67 76 68    Xlmc7DLJJAAr/gvh
00384: 75 31 61 58 61 44 37 31 36 43 52 6E 67 38 56 35    u1aXaD716CRng8V5
00400: 37 53 79 49 69 72 77 62 73 56 68 74 75 6C 38 62    7SyIirwbsVhtul8b
00416: 48 56 37 59 53 43 48 66 42 69 5A 79 2B 6B 56 67    HV7YSCHfBiZy+kVg
00432: 34 45 63 75 45 6E 65 48 38 44 75 73 76 32 6A 49    4EcuEneH8Dusv2jI
00448: 77 72 61 50 62 33 73 58 7A 70 48 37 41 67 68 37    wraPb3sXzpH7Agh7
00464: 4D 59 4C 49 78 69 4A 6F 48 35 54 38 43 48 69 2F    MYLIxiJoH5T8CHi/
00480: 4F 70 45 70 62 67 43 73 77 6D 65 45 61 37 6B 4D    OpEpbgCswmeEa7kM
00496: 55 41 58 50 66 51 74 56 4E 6E 4E 55 34 6F 56 30    UAXPfQtVNnNU4oV0
00512: 7A 7A 59 54 59 67 3D 3D                            zzYTYg==
```

Çizelge EK2.4. StoreDHCert Response Paketi

```
00000: 48 54 54 50 2F 31 2E 31 20 32 30 30 20 4F 4B 0D    HTTP/1.1 200 OK.
00016: 0A 44 61 74 65 3A 20 4D 6F 6E 2C 20 30 34 20 4A    .Date: Mon, 04 J
00032: 61 6E 20 32 30 31 30 20 30 36 3A 31 37 3A 32 30    an 2010 06:17:20
00048: 20 47 4D 54 0D 0A 53 65 72 76 65 72 3A 20 4D 69     GMT..Server: Mi
00064: 63 72 6F 73 6F 66 74 2D 49 49 53 2F 36 2E 30 0D    crosoft-IIS/6.0.
00080: 0A 58 2D 50 6F 77 65 72 65 64 2D 42 79 3A 20 41    .X-Powered-By: A
00096: 53 50 2E 4E 45 54 0D 0A 58 2D 41 73 70 4E 65 74    SP.NET..X-AspNet
00112: 2D 56 65 72 73 69 6F 6E 3A 20 32 2E 30 2E 35 30    -Version: 2.0.50
00128: 37 32 37 0D 0A 43 61 63 68 65 2D 43 6F 6E 74 72    727..Cache-Contr
00144: 6F 6C 3A 20 70 72 69 76 61 74 65 0D 0A 43 6F 6E    ol: private..Con
00160: 74 65 6E 74 2D 54 79 70 65 3A 20 74 65 78 74 2F    tent-Type: text/
00176: 68 74 6D 6C 3B 20 63 68 61 72 73 65 74 3D 75 74    html; charset=ut
00192: 66 2D 38 0D 0A 43 6F 6E 74 65 6E 74 2D 4C 65 6E    f-8..Content-Len
00208: 67 74 68 3A 20 33 34 34 0D 0A 0D 0A 41 6B 69 64    gth: 344....Akid
00224: 73 77 4F 70 57 74 7A 6E 74 53 77 4E 69 7A 76 6F    swOpWtztnSwNizvo
00240: 4B 66 4A 4A 42 62 57 6C 4E 54 58 59 44 45 6C 72    KfJJBbWlNTXYDElr
00256: 2B 45 6B 53 59 44 59 46 33 74 72 2F 6C 37 36 6E    +EkSYDYF3tr/l76n
00272: 78 44 6E 4D 6E 65 49 41 56 38 6A 64 35 30 37 52    xDnMneIAV8jd507R
00288: 6A 70 59 61 76 6C 4D 55 6B 51 67 53 2B 52 31 68    jpYavlMUkQgS+R1h
00304: 31 62 59 6D 79 69 4B 5A 6E 78 2F 30 63 35 6D 4F    1bYmyiKZnx/0c5mO
00320: 78 4D 6A 70 56 32 6C 44 4F 69 4E 36 50 50 62 4F    xMjpV2lDOiN6PPbO
00336: 4C 37 42 47 38 4D 71 30 2F 6B 34 4E 61 31 36 4D    L7BG8Mq0/k4Na16M
00352: 41 75 59 4E 6D 76 64 2F 77 7A 4C 2F 62 59 43 63    AuYNmvd/wzL/bYCc
00368: 4A 47 73 53 44 45 36 5A 38 44 72 55 51 39 43 38    JGsSDE6Z8DrUQ9C8
00384: 47 6A 4A 39 2B 43 52 53 58 4C 73 52 51 6A 31 56    GjJ9+CRSXLsRQj1V
00400: 52 50 58 58 37 51 36 2F 43 70 61 51 50 70 67 30    RPXX7Q6/CpaQPpg0
00416: 58 69 6A 42 63 78 2F 79 34 71 2F 79 64 57 2F 69    XijBcx/y4q/ydW/i
00432: 4B 38 76 42 64 76 4A 46 4C 38 45 54 59 55 4C 35    K8vBdvJFL8ETYUL5
00448: 52 35 61 58 33 6C 33 52 48 4D 48 48 31 69 53 66    R5aX3l3RHMHH1iSf
00464: 68 73 68 4F 70 33 64 33 73 76 4C 6B 37 44 63 6B    hshOp3d3svLk7Dck
00480: 4E 55 6C 77 63 33 52 6D 4F 6E 47 6C 54 33 64 30    NUlwc3RmOnGlT3d0
00496: 39 63 2B 35 36 53 53 52 63 6C 6D 2F 6E 63 72 56    9c+56SSRclm/ncrV
00512: 31 2F 78 2F 70 6B 39 32 35 44 2B 44 6A 36 78 5A    1/x/pk925D+Dj6xZ
00528: 54 4C 58 4F 51 6E 4F 77 75 67 36 39 70 54 6E 34    TLXOQnOwug69pTn4
00544: 67 53 56 7A 32 52 2F 32 76 66 4B 6C 37 2F 35 52    gSVz2R/2vfKl7/5R
00560: 32 6A 41 3D                                        2jA=
```

ÖZGEÇMİŞ

Adı Soyadı : Yusuf İstemi BEŞEL

Doğum Yeri : Zonguldak

Doğum Yılı : 1977

Medeni Hali : Evli ve bir çocuk babası

Eğitim ve Akademik Durumu:

Lise : 1991-1994 Zonguldak Atatürk Anadolu Lisesi

Lisans : 1994-1999 Gazi Üniversitesi

Elektrik-Elektronik Mühendisliği Bölümü

Yabancı Dil: İngilizce

İş Tecrübesi:

2000 – 2003 MHT

Proje Mühendisi

2004 - Başbakanlık

Proje Mühendisi