MARMARAÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

CAdES E-İMZA STANDARDINA UYUMLU GÜVENLİ KONUM DAMGASI MODELİ

OKAN KONUR

527322604

YÜKSEK LİSANS TEZİ

Bilgisayar Mühendisliği Anabilim Dalı
Bilgi Güvenliği Mühendisliği Programı

DANIŞMAN
Doç. Dr. Bahattin YALÇINKAYA

İSTANBUL, 2023

MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

SECURE LOCATION STAMP MODEL COMPATIBLE WITH CAdES DIGITAL SIGNATURE STANDARD

OKAN KONUR

527322604

MASTER'S THESIS

Department of Computer Engineering

Information Security Engineering

Thesis Supervisor

Assoc. Prof. Bahattin YALÇINKAYA

ISTANBUL, 2023

MARMARA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Öğrencisi Okan KONUR'un "**CAdES E-İMZA STANDARDINA UYUMLU GÜVENLİ KONUM DAMGASI MODELİ**" başlıklı tez çalışması, …… tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Doç. Dr. Bahattin YALÇINKAYA (Danışman)
Marmara Üniversitesi ..... (İMZA).....

Dr. Öğr. Üyesi Aydın ERDEN (Üye)
Marmara Üniversitesi ..... (İMZA).....

Prof. Dr. Niyazi ÇİÇEK (Üye)
İstanbul Üniversitesi ..... (İMZA).....

ONAY

Marmara Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun …… tarih ve …….. sayılı kararı ile Okan KONUR'un Bilgisayar Mühendisliği Anabilim Dalı Bilgi Güvenliği Mühendisliği Programında Yüksek Lisans derecesi alması onanmıştır.

Fen Bilimleri Enstitüsü Müdürü

MARMARA UNIVERSITY INSTITUTE FOR GRADUATE STUDIES IN PURE AND APPLIED SCIENCES

Okan KONUR, a Master of Science student of Marmara University Institute for Graduate Studies in Pure and Applied Sciences, defended his thesis entitled "**SECURE LOCATION STAMP INFRASTRUCTURE COMPATIBLE WITH CAdES DIGITAL SIGNATURE STANDARD**", on …….. and has been found to be satisfactory by the jury members.

Jury Members

Assoc. Prof. Bahattin YALÇINKAYA (Advisor)
Marmara University ..... (SIGN) .....

Asst. Prof. Aydın ERDEN (Jury Member)
Marmara University ..... (SIGN) .....

Prof. Dr. Niyazi ÇİÇEK (Jury Member)
İstanbul University..... (SIGN) .....

APPROVAL

Marmara University Institute for Graduate Studies in Pure and Applied Sciences Executive Committee approves that Okan KONUR be granted the degree of Master of Science in Department of Computer Engineering, Information Security Engineering Program on ……. (Resolution no: ……..).

Director of the Institute

Teşekkür

Bu tezin oluşum sürecinde desteğini esirgemeyen birçok kişiye teşekkür etmek istiyorum. İlk olarak, danıışmanım Doç. Dr. Bahattin YALÇINKAYA'ya tez çalışmam boyunca gösterdiği sabır, anlayış ve yardımseverlik için teşekkür ederim.

Dr. Öğr.Üyesi Hüseyin YÜCE'ye tez sürecine başlama döneminde beni yönlendirdiği için teşekkür ederim. Kıymetli değerlendirmeleri ve iyileştirme önerileri için Prof. Dr. Niyazi ÇİÇEK ve Dr.Öğr.Üyesi Aydın ERDEN hocalarıma da teşekkür etmek isterim.

Ayrıca, bu süreçte her zaman yanımda olan arkadaşlarıma ve aileme destekleri için minettarım. Teknik sorunlarımda her zaman yardımcı olan Ahmet AYDIN'a teşekkür ederim.

Ek olarak, GALILEO altyapısı konusunda sorduğum sorulara, gönderdiğim e-postalara detaylı cevaplar vererek zaman harcayan Daniel ESTÉVEZ'e çok teşekkür ederim.

Kasım 2023 **Okan KONUR**

İçindekiler

TEŞEKKÜR ..... **İ**

ÖZET ..... **V**

ABSTRACT ..... **Vİ**

KISALTMALAR / ABBREVIATIONS ..... **Vİİ**

TANIMLAR / DEFINITIONS ..... **Vİİİ**

ŞEKİLLER / FIGURES ..... **İX**

TABLOLAR / TABLES ..... **İX**

1. GİRİŞ **11**

1.1 İLGİLİ ÇALIŞMALAR **13**

2. DİJİTAL İMZALAR VE AÇIK ANAHTAR ALTYAPISI **17**

2.1 DİJİTAL İMZALAR **17**

2.2 ÖZET FONKSİYONLARI **18**

2.3 AÇIK ANAHTAR ALTYAPISI (PKI) **19**

2.4 STANDARTLAR **20**

2.5 CADES **21**

2.6 ABSTRACT SYNTAX NOTATİON 1 (ASN.1) **21**

2.7 ITU-T NESNE TANIM AĞACI (OBJECT IDENTİFİER TREE) **22**

2.8 CADES İMZALANAN ÖZELLİKLER (SİGNED ATTRİBUTES) **23**

2.9 ZAMAN DAMGASI **23**

2.10 CADES-T VE CADES-XLONG **24**

2.11 UZUN DÖNEM SAKLAMA **25**

3. KÜRESEL KONUM BELİRLEME SİSTEMLERİ **27**

3.1 KONUM BELİRLEME SİSTEMLERİNİN GEÇMİŞİ **27**

3.1.1 UZAY TABANLI SİSTEMLER 28

3.1.1.1 Optik teknikler *28*

3.1.1.2 Radyo teknikler *28*

3.2 UYDU TABANLI KONUMLAMA PRENSİPLERİ VE ÖLÇÜM TİPLERİ **29**

3.2.1 UYDU MENZİL BİLGİLERİ İLE KONUM BELİRLEME 31
3.2.2 ALICI SAATİNİN SAPMASI 32

3.3 GNSS BÖLÜMLERİ 32

3.3.1 UZAY BÖLÜMÜ 33
3.2.2 KONTROL BÖLÜMÜ 33
3.2.3 KULLANICI BÖLÜMÜ 33

3.4 GNSS SİNYAL GÜVENLİĞİ 34

3.4.1 TELSİZ ENGELLEME (JAMMİNG) SALDIRILARI 34
3.4.2 ENGELLEME SALDIRILARI ÖRNEKLERİ 35
3.4.3 SAHTECİLİK (SPOOFİNG) SALDIRILARI 35
3.4.3.1 Sahte iletim (forwarding spoofing, meaconing) saldırısı 36
3.4.3.2 Sahte üretim (generating spoofing) saldırısı 37
3.4.3.3 Sahtecilik saldırılarına örnekler 37

3.5 GNSS SAHTE SİNYAL SALDIRILARININ ENGELLENMESİ 38

3.5.1 YAYIN DOĞRULAMA YÖNTEMLERİ 39
3.5.2 MESAJ DOĞRULAMA KODLARI 40
3.5.3 TESLA PROTOKOLÜ 41
3.5.4 TEK YÖNLÜ ZİNCİRLER 41
3.5.5 TESLA ZAMAN SENKRONİZASYONU 42
3.5.6 GALILEO OS 42
3.5.7 GALILEO OSNMA 43
3.5.8 GALILEO OSNMA MESAJI YAPISI VE TESLA ZİNCİRİ ENTEGRASYONU 46
3.5.8.1 HKROOT 48
3.5.8.2 NMA başlık alanı (NMA header) 49
3.5.8.3 Zincir ID (Chain ID, CID) 50
3.5.8.4 Zincir ve açık anahtar durumu (CPKS) 50
3.5.8.5 DSM (Digital Signature Message) 50
3.5.8.6 DSM-KROOT 51
3.5.8.7 KROOT 51
3.5.8.8 DS 52
3.5.8.9 MACK 52
3.5.8.10 Tag-Info 52
3.5.8.11 ADKD 53

4. NAVİGASYON MESAJININ E-İMZAYA EKLENMESİ 55

4.1 GEREKLİ CİHAZLAR VE YAZILIMLAR 55

4.2 ÖNERİLEN YÖNTEM 55

4.3 SLOW MAC **57**

4.4 ZAMAN SENKRONİZASYONU KOŞULU **57**

4.5 OSNMA SİGNAL-İN-SPACE KAYDETME **59**

4.6 MERKLE KÖKÜ VE AÇIK ANAHTAR **60**

4.7 OSNMA DOĞRULAMA AKIŞ BAŞLATMA TİPLERİ **61**

4.7.1 SOĞUK BAŞLANGIÇ (COLD START) 61

4.7.2 ILIK BAŞLANGIÇ (WARM START) 61

4.7.3 SICAK BAŞLANGIÇ (HOT START) 61

4.8 KRİPTOGRAFİK DOĞRULAMA ADIMLARI **62**

4.8.1 GEREKEN FONKSİYONLAR 62

4.8.2 DSM-KROOT DOĞRULAMA 63

4.8.3 TESLA ANAHTAR DOĞRULAMA 63

4.8.4 TAG DOĞRULAMA 64

4.9 KONUM DAMGASININ OLUŞTURULMASI **64**

4.9.1 DOĞRULANAN OSNMA MESAJLARI İÇİN ZAMAN DAMGASI ALINMASI 64

4.9.2 KONUM DAMGASI FORMATI 65

4.9.3 UNREGİSTERED OID TANIMLAMA (UUID) 68

4.9.4 DOĞRULAMA 69

4.10 CİHAZ VE UYGULAMA GÜVENLİĞİ **69**

5. SONUÇ **71**

5.1 GELECEKTEKİ ÇALIŞMALAR **72**

KAYNAKLAR **73**

EKLER **81**

ÖZGEÇMİŞ **95**

ÖZET

CAdES E-İMZA STANDARDINA UYUMLU GÜVENLİ KONUM DAMGASI MODELİ

Dijital bir verinin oluşturulma anındaki konumunun kanıtlanma ihtiyacı günümüzde resmi, ticari ve askeri çeşitli alanlarda ortaya çıkmaktadır. Bu çalışmayla, Küresel seyrüsefer uydu sistemlerinden alınan konum verilerini e-imza paketine entegre edilmesi amaçlanmaktadır. Bu tasarımın CAdES imza paketi standardını destekleyen programlarla geriye dönük uyumlu olması, bağımsız kurum veya kişilerin imza sırasındaki cihazın veya kullanıcının konumunu sonradan doğrulayabilmesi amaçlanmaktadır.

Bu yöntemle doğrulanacak konum verisi; taşınabilir IoT cihaz ve hareketli araçlarda, kriminal davalarda, ticari lojistik filoların koordinasyonunda, askeri harekatlardaki iletişim ve insansız cihazların uzaktan kontrolünde ek bir güvenlik katmanı daha sunması açısından faydalı olacaktır.

Çalışmamızda öncelikle odaklandığımız problem, uydudan aldığımız sinyallerin yanıltma (spoofing) saldırılarına maruz kalmış olma ihtimalidir. Bu zafiyeti engellemek için güncel kriptografik çözümlerden yaygın kullanıma en hazır olan OSNMA protokolü incelenmiştir. OSNMA protokolü GALILEO uydu sisteminde sinyal doğrulaması için TESLA adında bir simetrik anahtar paylaşım algoritması kullanmaktadır. Tek yönlü yayın için çok uygun olan bu algoritmada anahtar dağıtımının zamana bağlı olması, ihtiyacımız olan asimetrik özellikleri kazandırırken sistem performansını düşürmemektedir.

Çözüm sunmaya çalışacağımız ikinci nokta ise, oluşturulan konum doğrulama verisinin e-imza standardına uyumunun değerlendirilmesidir. Bağımsız kurum veya kişilerin bu bilgiyi okuması için verinin standartlara uygun bir yapıda paketlenmesi gerekmektedir. Bu amaçla CAdES formatında imzaya dahil olan bir özellik alanı tanımlanmaktadır.

Bu iki altyapıyı birleştiren bu çalışma ile bağımsız taraflar tarafından doğrulanabilir güvenli bir konum damgası modeli önerilmektedir.

ABSTRACT

SECURE LOCATION STAMP MODEL COMPATIBLE WITH CAdES DIGITAL SIGNATURE STANDARD

The need to prove the location where a digital data has been created arises in various governmental, commercial and military fields. This study aims to integrate location data retrieved from global navigation satellite systems into the e-signature package. This design is intended to be backward compatible with programs that support the CAdES package standard, and independent parties can subsequently verify the location of the device or user at the time of signature.

The location data to be verified with this method will be useful in portable IoT devices and mobile vehicles, criminal cases, coordination of commercial logistics fleets, communication in military operations and remote control of unmanned devices.

The problem we primarily focus on in our study is the possibility that the signals we receive from the satellite may be subject to spoofing attacks. In order to prevent this vulnerability, the OSNMA protocol, which is the most ready option for widespread use among current cryptographic solutions, is analyzed. The OSNMA protocol uses a symmetric key sharing algorithm called TESLA for signal authentication in the GALILEO satellite system. This algorithm is well suited for one-way broadcasting, and the time-dependent key distribution gives the asymmetric features we need, while not degrading system performance.

The second point we will try to solve is to evaluate the compliance of the generated location verification data with the e-signature standard. In order for independent institutions or individuals to read this information, the data must be packaged in a structure that complies with the standards. For this purpose, a new signed attribute is defined in CAdES.

By combining these two infrastructures, this study proposes a secure location stamping model that can be proved by independent verifiers in long term.

KISALTMALAR / ABBREVIATIONS

AGC: Automatic Gain Control (otomatik kazanç kontrolü)

ASN.1: Abstract Syntax Notation One

BES: Basic Electronic Signature (Temel Elektronik İmza)

CA: Certificate Authority (Sertifika Otoritesi veya ESHS)

CAdES: CMS Advanced Electronic Signatures

CHIMERA: Chips Message Robust Authentication

CMS: Cryptographic Message Syntax

CRL: Certificate Revocation List (Sertifika İptal Listesi - SİL)

ÇİSDUP: Çevrimiçi Sertifika Durum Protokolü (Online Certificate Status Protocol - OCSP)

DSM: Digital Signature Message (Dijital İmza Mesajı)

ECDSA: Elliptic Curve Digital Signature Algorithm

ES: Electronic Signature (Elektronik İmza)

ESHS: Elektronik Sertifika Hizmet Sağlayıcısı

GNSS: Global Navigation Satellite Systems (Küresel Uydu Seyrüsefer Sistemi)

GPS: Global Positioning System (Kürsel Konumlama Sistemi)

GST: GALILEO System Time (GALILEO Sistem Saati)

ICD: Interface Control Document (Arayüz Konrol Dokümanı)

IoT: Internet of Things (Nesnelerin Interneti)

ITU: International Telecommuncation Union (Uluslararası Telekomünikasyon Birliği)

MitM: Man-in-the-Middle (Mesaj iletiminde araya giren kötü amaçlı kişi)

MSB: Most Significant Bits (İkili tabandaki sayının en solundaki bitler)

NMA: Navigation Message Authentication (Navigasyon Mesajı Doğrulama)

NTP: Ağ Zaman Protokolü (Network Time Protocol)

NTS: Ağ Zaman Güvenliği (Network Time Security)

OID: Object Identifier (Nesne Tanıtıcısı)

OSNMA: Open Service Navigation Message Authentication (Açık Hizmet Konum Mesajı Doğrulması)

PKI: Public Key Infrastructure (Açık Anahtar Altyapısı)

PVT: Position, velocity, time (konum, hız, zaman)

RSA: Rivest–Shamir–Adleman

SCA: Spreading Code Authentication (Yayılım Kodu Doğrulama)

SCER: Security Code Estimation and Replay (Güvenlik Kodu Tahmini ve Yeniden Oynatma)

SDR: Yazılım Tanımlı Radyo (Software Defined Radio)

SIS: Signal-in-Space (Uydu Sinyali, Uzaydaki Sinyal)

SLR: Satellite Laser Ranging (Uydu Lazer Menzilleme)

SoC: System on a Chip (Yongada Sistem)

TESLA: Timed Efficient Stream Loss-Tolerant Authentication (Zamanlanmış Verimli Kayıp Toleranslı Akış Doğrulama)

TOA: Time of Arrival (Varış Zamanı)

UTC: Universal Time Coordinated (Eşgüdümlü Evrensel Zaman)

TANIMLAR / DEFINITIONS

Spoofing: Yanıltma. Yazılım ve bilgi güvenliğinde saldırganın gerçek olmayan, mesajı veya sinyali alan tarafı kandırma amaçlı saldırıları için kullanılır.

Jamming: Radyo sinyallerinin çeşitli yöntemlerle karıştırılması veya engellenmesi.

Authentication: Kriptografik kimlik doğrulama.

ŞEKİLLER / FIGURES

ŞEKİL 2.1 CADES ZAMAN DAMGASI YAPISI 25
ŞEKİL 2.2 CADES-A YAPISI 26
ŞEKİL 3.1 MENZİL BELİRLEME HESAPLAMASI 30
ŞEKİL 3.2 UYDU UZAKLIK HESAPLAMASI 31
ŞEKİL 3.3 3 UYDU İLE NİRENGİ (TRIANGULATION) 32
ŞEKİL 3.4 ENGELLEME VE YANILTMA SALDIRILARI 35
ŞEKİL 3.5 TEK YÖNLÜ ZİNCİRLER 41
ŞEKİL 3.6 TESLA İKİ TARAFLI ZAMAN SENKRONİZASYONU 42
ŞEKİL 3.7 GALILEO OSNMA İŞLEM SÜRECİ 45
ŞEKİL 3.8 GALİLEO PAGE-SUBFRAME İLİŞKİSİ 46
ŞEKİL 3.9 PAGE VE FRAME ZAMANLARI 47
ŞEKİL 3.10 E1-B I/NAV PAGE İÇERİĞİNDEKİ OSNMA MESAJI BÖLÜMLERİ 48
ŞEKİL 3.11 HKROOT MESAJI YAPISI 49
ŞEKİL 3.12 NMA BAŞLIK ALANI 49
ŞEKİL 3.13 MACK MESAJI 52
ŞEKİL 4.1 KONUM DAMGASI MODELİ 57
ŞEKİL 4.2 CONTENTINFO ASN.1 OBJESİ 66
ŞEKİL 4.3 KOORDİNAT BİLGİLERİ ASN.1 OBJELERİ 66
ŞEKİL 4.4 UYDU SİNYALLERİ İÇİN ASN.1 OBJELERİ 67
ŞEKİL 4.5 GNSS DOĞRULAMA VERİSİ ASN.1 OBJESİ 68
ŞEKİL 4.6 KONUM DAMGASI ASN.1 OBJESİ 68

TABLOLAR / TABLES

TABLO 3.1 GALILEO OSNMA ALICI GEREKLİLİKLERİ 46
TABLO 3.2 NMAS ALANI AÇIKLAMALARI 50
TABLO 3.3 ADKD TABLOSU 53

1. GİRİŞ

Yazılı bir mesajın, bir iradenin, veya herhangi bir dijital verinin beyan edildiği zamanda, beyan eden kişi tarafından hazırlandığının ve beyan edildiği orijinal hali sonradan bozulmamış halde saklandığının, bu işleme tanık olmamış bağımsız taraflar tarafından da inkâr edilemeyecek şekilde kanıtlanması için e-imza teknolojisi ve PKI kullanılmaktadır. Yapacağımız çalışmadaki amacımız e-imza paketine, imzanın gerçekleştiği konum bilgisini de ekleyerek, imza anındaki konumun sonradan kanıtlanabilmesini sağlamaktır. Bu bilginin pratik bir şekilde kanıtlanabilmesi, pek çok farklı sektörde GNSS uyumlu cihazların sunduğu verilerin doğrulanmasında büyük bir iyileştirme sağlayacaktır.

Hedefimiz bunun için bir altyapı önerisi sunmaktır. Bu çalışma kapsamında CAdES e-imza formatını genişletirken, zaman damgası benzeri bir konum damgası hizmeti tasarlamak amaçlanmıştır. Bu hedefi gerçekleştirmek için çözülmesi gereken iki ana probleme odaklanacağız. İlki, cihazların aldığı uydu sinyallerinin doğrulanmasıdır. İkincisi ise doğrulanan bu verinin, bağımsız kişiler tarafından da üstünden zaman geçse bile doğrulamaya devam edebilmesi için mevcut e-imza programlarıyla uyumlu olacak şekilde paketlenmesidir.

İlk problemin çözümü olarak küresel konumlama amaçlı uydu sistemlerinin devreye almayı planladığı mesaj doğrulama altyapılarını temel alacağız. Bu konuda mevcut şartlar altında mesaj doğrulama testlerini yapmakta olan iki GNSS mevcuttur. Avrupa Birliği tarafından kurulan ve Avrupa Uzay Ajansı (ESA) tarafından işletilen GALILEO ve Amerika Birleşik Devletleri'nin sahibi olduğu GPS. GALILEO, uydularında OSNMA destekli sinyalleri 2023 yılında hizmete sunmayı planlıyor. GPS ise aynı yıl CHIMERA adlı aynı amaçlı sisteminin ilk deneysel sinyallerini yayınlamaya başlayacak. Bu iki sistem de kullanıcı cihazlarının aldığı navigasyon sinyallerini doğrulayabilmesini ve yanıltma (spoofing) saldırılarının önüne geçmesini sağlayacak. Operasyonel açıdan GALILEO sisteminin OSNMA protokolünün 2023 yılı içinde herkes için açık hizmete sunulacak olması ve tasarımının daha olgunlaşmış olması bu çalışmamızda bizim için daha önemli olduğundan, bu sisteme odaklanılmıştır.

İkinci problemin çözümü için cihazın o an için sadece kendisinin doğrulayabildiği veriyi kaydetmesi, ve bu kaydedilen verinin sonradan başkalarının doğrulayabilmesi için uyumlu şekilde paketlenmesi gerekmektedir. Bunun için CAdES standardının izin verdiği şekilde ASN.1 formatında yeni bir imzalı özellik (signed attribute) tanımlanması öngörülmektedir. Bu yeni tanımlanacak alanda konum bilgisi ve konum bilgisini oluşturmak için gerekli doğrulanmış uydu sinyali detaylarının saklanabileceği bir model tasarlanacaktır. Konum bilgisinin bu alana yazılmadan önce zaman damgası ile anlık olarak ilgili sertifika otoritesinin imzasıyla damgalanması planlanmaktadır. Sonuçta oluşan CAdES paketi, konum doğrulama desteği vermeyen mevcuttaki programlar tarafından da okunabilecektir. Konum doğrulama desteği veren programlar ise ilgili alanı dikkate alacaklardır. Bu alandaki veri, imza paketinde olduğu için, standart imza doğrulama süreciyle her iki durumda da doğrulanmış olacaktır.

Bu çalışmada, sistemin detaylarının, kullanılacak protokollerin ve standartların açıklanması, sunulacak yeni mimari ögelerin tanımlanması teorik olarak sistem işleyişi tasarlanması hedeflenmektedir. İlgili yapının baştan sona simülasyon implementasyonu hedeflenmemektedir.

Bu tez çalışması ana hatlarıyla beş bölümden oluşmaktadır. İlk bölüm olan giriş bölümünde yapılacak çalışmanın içeriğiyle alakalı ön bilgiyle beraber kapsamı, amacı ve daha önce literatürde yayınlanan benzer yayın ve makaleler sunulmaktadır.

İkinci bölümde, e-imza teknolojileri ve PKI hakkında temel bilgiler verilerek, kalan bölümlerde anlatılan tasarım kararlarının neden ve nasıl alındığı konusunda bir dayanak oluşturması planlanmıştır.

Üçüncü bölümde, güvenilir konum bilgisinin imzalayan tarafından temin edilmesi için gereken uydu mesajlarını sağlayan sistemlerin yapısı ve teknik özellikleri detaylandırılmıştır.

Dördüncü bölümde önerilen konum damgası modeli açıklanmıştır. İkinci ve üçüncü bölümlerde temellendirilen sistemlerin birbirine nasıl entegre edileceği bu bölümde tasarlanmaktadır.

Beşinci bölüm ise sonuç bölümüdür. Yapılan çalışmada ulaşılan çıkarımlar ve gelecek araştırmalarda odaklanılması gereken noktalar belirtilmektedir.

Bu tez kapsamında teknik olarak bazı terimlerin Türkçe'ye çevrilmeden özel isim kullanılması uygun görülmüştür. İlgili tanımlar tek seferlik olmak üzere TANIMLAR başlığında verilmiştir.

1.1 İlgili Çalışmalar

Her ne kadar açık anahtar altyapısı ve e-imza işlemi sayesinde beyan edilen mesajın bütünlüğü ve beyan edilen kişi tarafından imzalandığı kanıtlansa da beyan anında mesajın içeriğine eklenen verilerin, hesaplama anındaki doğruluğu kanıtlanamayabiliyor. Örnek olarak Basit Elektronik İmza (BES) tipinde atılan imzalarda zaman damgası yoktur. Bu imzalarda, imzayı atan cihazın o anda kaydettiği zaman imzada beyan edilen zamandır. Fakat cihazın gerçekten o andaki doğru zamanı ölçüp ölçmediğini bilemeyiz. Bu yüzden zaman damgası hizmeti ortaya çıkmıştır. İmza atan cihazdan bağımsız bir sunucu, istek olması durumunda güvenilir bir merkezden (genellikle ulusal resmi saat hizmeti sunan bir otorite) zaman bilgisini çekerek, ESHS'nin sertifikasıyla istenen veri parçasını imzalar ve bu imza zaman damgası olarak kullanıcının imzalamak istediği mesaj paketine eklenir. Bu işlem ilgili veri parçasının, zaman damgasından önce var olduğunu kanıtlamaktadır. Bu süreç ikinci bölümde daha detaylı açıklanacaktır.

Bu tez kapsamında yapacağımız çalışmada sunacağımız mimari de zaman damgası işleyişine benzer olacak şekilde tasarlanmıştır. Fakat önerilen konum damgasında, imza işlemini gerçekleştirecek cihazdan bağımsız bir sunucudan konum verisini almamız mümkün değildir. Konum, zaman gibi bağımsız olamadığından, istemci cihazın konum bilgisini kendisinin oluşturup kaydetmesi gerekmektedir. Bu gereklilik güvenlik açısından bir zafiyet oluşturmaktadır. Bu zafiyet iki aşamada ortaya çıkmaktadır. Birincisi uydudan gelen konumlama sinyalinin güvenilir olup olmadığının kanıtlanması gerekliliğidir. Konumlama sinyali radyo frekansında yayılır ve kötü amaçlı saldırganlar tarafından yapay olarak oluşturulması günümüzde ucuz cihazlarla bile mümkün olmaktadır.[1]

İkinci aşama ise uygulama güvenliğidir. Konum sinyallerinin gerçek GNSS uydularından geldiği doğrulansa bile, koordinatları hesaplayan taraf uygulama yazılımı olmaktadır. Bu koordinat hesaplama işlemi eğer yanlış yapılırsa, imzaya eklenecek konum bilgisine güvenilemez. Ancak bu zafiyetin benzeri günümüzdeki tüm e-imza uygulamalarında bulunmaktadır. İmzalayan taraf, kendi sertifikasıyla imzalanan verinin gerçekten imzalamak istediği veri olduğundan emin olamaz. Kötü amaçlı bir yazılım, arayüzde göstermediği bir veriyi kullanıcıya imzalatabilir. Bu yüzden ilgili uygulamalar güvenilir otoriteler tarafından uyumluluk testlerinden geçirilmektedirler. Dolayısıyla bu tez kapsamında bu zafiyete çözüm bulunmamaktadır.

Konum damgası fikri ilk kez ortaya atılmamaktadır. İmzalanan veride kanıtlanabilir koordinat verisinin bulunması amacıyla çeşitli çözümler sunulmuştur. Bu çözümleri ve bu tez çalışmasına göre farklarını aşağıdaki gibi inceleyebiliriz:

2011 yılında hazırlanan "Konum Damgası Sistemi Önerisi" adlı yüksek lisans tezinde Mustafa Özlü tarafından konum bilgisini metin imzasının içine gömülmesi önerilmiştir.[2] Önerilen sistem istemci ve sunucu uygulamalardan oluşmaktadır. Kullanıcının daha önceden belirlediği bir konum dışında imza atması engellenmektedir. Bu sistem için sunucu uygulamasında bir hesap açmaya ve veritabanı altyapısına ihtiyaç vardır. Ayrıca konum bilgisi metne eklendiği için ilgili çalışma kapsamında sadece metin imza tipindeki e-imzalarla uyumludur. Çalışmada iki eksiklik göze çarpmaktadır. Konum bilgisinin hangi formatta ve standartta mesaja eklendiği belirtilmemektedir. Metin dosyasının sonunda mı olacağı, konum verisinin hangi uzunlukta ve hangi verileri kapsayacağı gibi sorunlar açıklığı kavuşturulmamıştır. Bunlar tasarımın sağlıklı çalışmasını ve yaygınlaşmasını engelleyecektir. İkinci eksiklik ise alınan uydu sinyallerinin doğru kabul edilmesi ve spoofing saldırılarına karşı herhangi bir önlem alınmamasıdır.

2013'te yapılan bir diğer çalışma ile, Ádámkó ve Pethő tarafından mobil cihazların kullandığı GPS koordinatları için konum damgası önerilmiştir.[3] Bu çalışma iki adet protokol önermektedir. Biri güvenli bir donanım ile GPS verisinin alındığı anda imzalanmasını içermektedir, diğeri ise bu donanıma bağlı olmadan direkt mobil cihaz üzerindeki yazılım ile çalışmaktadır. Bu çalışmadaki imzalama protokolü akışı adımlarıyla detaylı bir şekilde tanımlanmıştır. Fakat e-imza formatları ve ilgili verinin

nasıl ekleneceği konusunda detaylandırma yapılmamıştır. Bu çalışmada da gelen sinyallerin spoofing saldırısı sonucunda alınmadığına güvenilmektedir. Spoofing saldırılarını engelleyen güncel uydu sistemleri çalışmanın yapıldığı zamanda henüz geliştirilmemişlerdir.

M. Kabatnik ve A. Zugenmaier'in 2001'de yaptığı çalışmada ise konum bilgisi küresel uydu sistemlerinden değil, GSM operatör sinyallerinden bulunmaktadır.[4] Kabatnik ve Zugenmaier bu sistemin GPS'e göre daha az kesinlik sağladığını fakat kandırılmasının daha zor olduğunu savunmaktadır. 2001 yılında GPS teknolojisinde sinyal doğrulama altyapısı mevcut olmadığından, bu çalışma da gelişen teknoloji ile güncelliğini yitirmiştir.

Literatürde görülen bu eksiklikleri tamamlama adına tez kapsamında iki ana bölüme sahip işlem akışı oluşturulmaya çalışılacak, ilk aşamada uydu sinyallerinin doğrulanması ve ikinci aşamada doğrulanan konum verisinin CAdES formatında ASN.1 uyumlu olarak yeni tanımlanan bir imza alanına eklenmesi yer alacaktır. Sinyal doğrulama altyapısı için GALILEO sistemindeki OSNMA protokolünün temel alınması planlanmıştır.

2. DİJİTAL İMZALAR VE AÇIK ANAHTAR ALTYAPISI

2.1 Dijital İmzalar

Dijital imza veya e-imza, bir aracı tarafından bir dosya, mesaj veya başka bir veri bloğu ile ilişkili olarak üretilen, ilişkili olduğu veriye mantıksal veya kriptografik olarak bağlı, bağımsız kişilerce doğrulanabilir bir bit örüntüsüdür.[5, 6] Başka bir aracı, veri bloğuna ve ilişkili imzasına erişebilir, veri bloğunun (inkâr edilemeyecek şekilde) beyan edilen imzalayan tarafından imzalandığını ve veri bloğunun imzadan bu yana değiştirilmediğini doğrulayabilir. Dijital imzaların kullanıcılar için sağladığı özellikler kısaca şunlardır[7]:

- **İnkâr Edilemezlik (**Non-repudiation**):** Bir veri parçasının gerçekten beklenen taraf tarafından imzalandığını bağımsız kişilerin de güvenebileceği şekilde kanıtlanmasıdır. **Doğrulama** (Authentication) özelliğini kapsar. Örnek vermek gerekirse, iki kişi arasındaki güvenli iletişimi düşünülebilir. B kişisi A kişisinden gelen bir mesajın gerçekten A tarafından gönderildiğine güvenebiliyorsa, bu mesaj doğrulanmış demektir. Fakat bu mesajın doğrulama kanıtı aynı zamanda A ve B'yi tanımayan ve güvenmeyen C kişisine gösterildiğinde, C de bu kanıta güvenebiliyorsa bu mesaj inkâr edilemezlik özelliğini sağlayabilmiş demektir.
- **Bütünlük** (Integrity): Mesajın oluşturulduktan sonra içeriği üzerinde değişiklik yapılmadığını kanıtlar.

Bu özellikleri mümkün kılan altyapılar sonraki bölümlerde anlatılmaktadır. Bilgi güvenliği açısından sağladığı bu özellikler sayesinde elektronik imza, imzalayan ile doğrulayan arasında daha sonra, hatta yıllar sonra ortaya çıkabilecek bir anlaşmazlık durumunda hakim kararları için kullanılabilir. [5]

Dijital imzalar tipik olarak asimetrik kriptografi algoritmaları kullanılarak oluşturulur. FIPS PUB 186-3'te NIST, üç dijital imza şeması belirtir: dijital imza algoritması (DSA), eliptik eğri DSA (ECDSA) ve RSA tabanlı dijital imza şemalarının birkaç versiyonu. Bu algoritmalar ANSI, ISO, IEEE ve IET dahil olmak üzere diğer kuruluşlar tarafından da benimsenmiştir. [6]

Asimetrik kriptografi açık ve gizli anahtar olmak üzere bir anahtar çifti kullanır. Bu iki anahtarla şifreleme ve şifre çözme yapılabilir. Bu anahtar çiftleri kontrol edilmesi kolay

ama hesaplanması zor olan bazı matematiksel problemlere dayanarak oluşturulur. Örnek olarak RSA temelde tam sayıların asal çarpanlara ayrılmasının algoritmik zorluğuna dayanır. Oluşturulan bir gizli anahtara karşılık gelen açık anahtar kötü niyetli kişi tarafından bilinse bile kısa sürede veya pratik yöntemlerle hesaplanamaz. Böylece gizli anahtar ile şifrelenmiş bir veriyi sadece o gizli anahtara karşılık gelen açık anahtar ile açabilirsiniz. Açık anahtarla şifresi çözülebilen verinin, ilgili gizli anahtarla şifrelendiğine kesin olarak güvenilebilir.

Dijital imzalarda performans ve bazı güvenlik zafiyetlerinden dolayı imzalama algoritmaları asimetrik kriptografi ile beraber özet fonksiyonlarını kullanır. Özet fonksiyonları kullanılmadığı durumlarda oluşacak zafiyeti kullanan saldırıya örnek olarak bir saldırganın anlamsız bir veri de olsa kurbanın imzalamadığı bir veriyi imzalamış gibi gösterebilmesini sağlayan varoluşsal sahtecilik saldırısı (existential forgery) verilebilir.[8]

2.2 Özet Fonksiyonları

Özet fonksiyonu, herhangi bir uzunluktaki mesajı girdi olarak alarak sabit uzunlukta bir değere dönüştüren tek yönlü bir fonksiyondur. Gizli anahtar kullanılmaz. Özet fonksiyonlarının birincil özelliği tek yönlü olmalarıdır, bir diğer deyişle, herhangi bir veriye uygulanmasında hesaplama açısından ucuz maliyetli olmasına rağmen, oluşan çıktı değerinden geriye dönerek orijinal veri hakkında herhangi bir bilgiye erişmenin çok zor olmasıdır. [9]

Özet fonksiyonlarının iki önemli güvenlik özelliği vardır:

- Fonksiyon tek yönlüdür, H(x) verildiğinde x'i bulmak zordur. (Pre-Image Resistance)
- Özet değerleri benzersizidir, x ve y birbiriden farklı verileri temsil ediyorsa, özetleri, H(x) ve H(y) de farklı olmak zorundadır. (Collision Resistence, Second Pre-Image Resistance) [10]

Bu güvenlik özellikleri ve sabit uzunlukta çıktı vermesi nedeniyle, imzalanacak veri direkt olarak imzalanmaz, bunun yerine önce verinin özeti alınır. Özet değeri üzerinde işlem yapılır. İmza doğrulama adımında ise yine özet değerleri karşılaştırılır.

2.3 Açık Anahtar Altyapısı (PKI)

Açık ve gizli anahtarlarla yapılan şifrelemeler matematiksel olarak kanıtlanabilir olsa da pratikte bu işlemleri yapan aracıların (kurum, kişi veya cihazların) ilgili gizli anahtara sahip tek güvenilir taraf olduğunun kanıtlanması için merkezi bir otoriteye ihtiyaç duyulmaktadır. Bu amaçla anahtar çiftlerinin sertifikalandırılması ve bu sertifikaların da doğrulanabilmesi için bir altyapı tasarlanmıştır. Asimetrik kriptografi algoritmalarında kullanılan açık anahtarları sertifikalandıran bu yapıya Açık Anahtar Altyapısı denir.

Sertifika Makamı (Certificate Authority - CA) olarak tanımlanan merkezi ve güvenilir bir kurum, kullanıcılar için anahtar çiftlerini üreterek (örnek: dijital imzalar için) veya kullanıcının kendisinin güvenli şekilde ürettiği gizli anahtara karşılık gelen açık anahtarı paylaşmasıyla beraber (örnek: SSL sertifika isteği) açık anahtar değerini bir sertifika dosyası ile paketler. Bu sertifika dosyasını ise kendi gizli anahtarı ile imzalar. Kendi gizli anahtarına karşılık gelen açık anahtarı ise yine kendisinin ürettiği bir sertifika ile onaylanabilir ve güvenilir kılar. [11]

Bu onaylama mekanizması için bir doğrulama ağacı kurulmuştur. Genelde pratikte şu şekilde işler: Kişinin sertifikası yine CA kontrolündeki alt kök denen ara bir sertifika sahibi katman tarafından imzalanarak yayınlanır. Alt kök sertifikası ise kök sertifikası tarafından imzalanır.[12] Bu ağaç yapısının amacı, sertifika makamı tarafından kritik köklerin anahtar yönetimini kolaylaştırmak ve daha güvenilir hale getirmektir.

En kökte yer alan sertifika, yine bu kök sertifikada bulunan açık anahtara karşılık gelen gizli anahtarla imzalanır (self signed root certificate).

Bu kök sertifikalar kullanıcıların cihazlarına varsayılan güvenilir sertifikalar olarak gömülecek şekilde dağıtılır. Böylece bu kökler ile başlayan zincirlerin herhangi bir katmanı tarafından imzalanan sertifikaların güvenilir olması sağlanmış olur. Kök sertifikaları yaklaşık 10 senede bir yenilenirler.[13]

PKI, diğer güvenlik hizmetleri için bir temel sağlar. PKI'ın birincil işlevi, genel anahtarların ve sertifikaların güvenlik ve bütünlük içinde dağıtılmasına ve kullanılmasına izin vermektir. PKI, diğer uygulamaların ve ağ güvenliği bileşenlerinin üzerine inşa edildiği bir temeldir. Genellikle PKI tabanlı güvenlik mekanizmaları gerektiren sistemler

arasında e-posta, çeşitli çipli kart uygulamaları, e-ticaret alışverişi (ör. banka ve kredi kartları), mobil bankacılık yer alır. PKI'ın birçok kullanımı ve uygulaması vardır. [14]

PKI aşağıdakiler gibi çeşitli sistemler için temel güvenlik hizmetlerinin temelindedir:

- İletişim ve işlem güvenliği için SSL, IPsec ve HTTPS
- E-posta güvenliği için S/MIME ve PGP

PKI'ın ve asimetrik şifreleme kullanımının e-ticaret ve diğer kuruluşlar için sunduğu bazı önemli avantajlar şunlardır:

- Güvenlik altyapısı geliştirme masraflarını azaltır.
- Güvenlik sistemlerinin karmaşıklığını azaltır.
- Zaafiyet risklerini azaltır ve bölümlere ayırır.
- Sistemlerin ve ağların verimliliğini ve performansını artırır.

Ek olarak, aşağıdakiler gibi sistemler de açık anahtar kriptografisinin temellerine dayanır:

- e-Oylama Sistemleri
- Kimlik (pasaportlar ve sürücü belgeleri)
- Noter onayı (sözleşme, e-posta vb.)
- Yazılım Dağıtımı
- Asimetrik anahtar yönetimi

2.4 Standartlar

Dijital imzaları ve çeşitli kriptografik verileri karmaşaya neden olmayacak şekilde ortak bir yapıda işlemek amacıyla bazı standartlar geliştirilmiştir.

ETSI bir Avrupa Standartları Organizasyonudur (ESO). Telekomünikasyon, yayın ve diğer elektronik iletişim ağları ve hizmetleriyle ilgili bölgesel standartları tanımlar. PKI konusunda tanıdığı ve tanımladığı bazı standartlar aşağıdaki gibi listelenebilir:

X.509: Açık anahtar sertifikalarının biçimini tanımlar.

CMS (Cryptographic Message Syntax): Bu standart, dijital imzalar ve diğer kriptografik işlemlerin biçimini tanımlar.

PKCS (Public-Key Cryptography Standards): RSA Laboratories tarafından geliştirilen bu standartlar serisi, anahtar üretimi, sertifika yönetimi ve sertifika iptali gibi

PKI'ın çeşitli yönlerini uygulamak için rehberlik sağlar. PKCS1-15 arası numaralandırılmışlardır. Örnek olarak CMS aynı zamanda PKCS#7 ile tanımlanır.

CAdES (CMS Advanced Electronic Signatures), CMS (Cryptographic Message Syntax) bazlı olarak geliştirilmiş bir dijital imza standardıdır. CAdES, CMS uyumlu imza verisinin saklanması için profiller tanımlar. CAdES, ETSI (European Telecommunications Standards Institute) tarafından yayınlanmıştır ve birçok Avrupa ülkesinde yasal olarak kabul edilmektedir.[15]

PAdES (PDF Advanced Electronic Signatures): Bu standart, görsel olarak oluşturulan ve PDF dosyalarına gömülebilen elektronik imzalar oluşturmanıza olanak tanır. [16]

XAdES (XML Advanced Electronic Signatures): Bu standart, dijital imzaların XML (eXtensible Markup Language) biçiminde saklanmasını ve taşınmasını sağlar. [17]

2.5 CAdES

CAdES (CMS Advanced Electronic Signature), ETSI TS 101 733 teknik dokümanında, ETSI EN 319 122-1 ve ETSI EN 319 122-2 standart dokümanlarında ve RFC 5126 kapsamında tanımlanmış, imzalama işleminin üstünden uzun zaman geçse de, doğrulamasının yapılabilmesini sağlayabilen, CMS tabanlı imza formatıdır.

Herhangi bir veri tipini imzalamak için kullanılabilir. PAdES ve XAdES gibi pdf ve xml doküman formatlarının içine gömülü olarak paketlenmez. İmzalanan verinin etrafında bir paket oluşturur. Bu nedenle CAdES imza paketlerini görüntülemek ve doğrulamak için ayrı programlar gerekmektedir.

2.6 Abstract Syntax Notation 1 (ASN.1)

ASN.1, ister karmaşık ister çok basit olsun, uygulama fark etmeksizin, programlama dilinden ve verilerin fiziksel temsilinden bağımsız olarak telekomünikasyon protokolleri tarafından iletilen verileri açıklamak için kullanılan resmi bir notasyondur.

ASN.1, 1984'ten beri kullanılmaktadır, ancak yeni talepleri karşılamak için sürekli olarak güncellenmektedir. 1988'de X.509 dijital sertifikalarını, 1995'te ise bant genişliği ve CPU açısından kısıtlı cihazlarını ve 2002'de XML'i destekleyecek şekilde geliştirildi. ASN.1

bugün bir milyardan fazla bilgisayarda ve gömülü sistem cihazında çok çeşitli uygulamalarda kullanılmaktadır. [18]

CAdES format kodlaması (encoding) ASN.1 (Abstract Syntax Notation 1)[19] yöntemi ile yapılmıştır. Telekomünikasyon ve uluslararası sistemlerin veri alışverişini gerektiren sistemlerde çok yaygın olarak kullanılan ASN.1, CAdES formatında paketlenmiş bir verinin bağımsız taraflar tarafından güvenilir şekilde işlenmesinde ortak düzene sahip bir altyapı sağlar. [15] CAdES imza dosyaları oluşturulurken, ASN.1 veri yapılarını ikili (binary) biçime dönüştürmek için BER veya DER kodlaması kullanılır.

BER, çeşitli kodlama seçeneklerine izin veren daha esnek bir kodlama şemasıdır, DER ise BER'in daha katı kurallar uygulayan bir alt kümesidir ve her ASN.1 veri yapısı için benzersiz bir kodlama sağlar. DER, tutarlı kodlama ve kod çözme sağlamak için dijital imzalar gibi benzersiz bir temsil gerektiren uygulamalarda sıklıkla kullanılır.[20]

2.7 ITU-T Nesne Tanım Ağacı (Object Identifier Tree)

Uluslararası Telekomünikasyon Birliği (International Telecommuncation Union - ITU), Birleşmiş Milletler'in bilgi ve iletişim teknolojileri (ICT'ler) konusunda uzmanlaşmış kuruluşudur. Ağların ve teknolojilerin sorunsuz bir şekilde birbirine bağlanmasını sağlayan teknik standartları geliştirir.

ITU-T ise ITU'nun Telekomünikasyon Standardizasyon Birimi'dir. ITU-T Tavsiyeleri (ITU-T Recommandations) olarak bilinen uluslararası standartları geliştirmek için dünyanın dört bir yanından uzmanları bir araya getirir.

Telekomünikasyon ve bilgi teknolojilerinde tanımlanan standartlarda genellikle bir "nesneye" benzersiz ve evrensel bir şekilde atıfta bulunulması gerekir. Bu nesne, özel bir varlıktan (örn: belirli tek bir dosya) ziyade, bir bilgi sınıfıdır (örn: bir dosya formatı).[21]

Kesinlik gerekteren tanımlamalar standardı kullanacak ilgili tarafların erişimine açık şekilde bir nesneye bir nesne tanıtıcısının (OID) atanmasıyla elde edilir. Bu atamalar bir kayıt otoritesi tarafından gerçekleştirilir.

ITU tanımlama yapısı, kayıt otoritesi veya tescil edilebilecek nesne sayısı sınırı olmadan uluslararası veya yerel olarak nesne tanımlamaya izin veren bir ağaç yapısıdır. Bu hiyerarşik adlandırma yapısı, web'deki alan ve alt alan adlarına benzer. Her düğüm, bir

ad (küçük harfle başlayan bir kelime) ve bir numara ile ilişkilendirilir. Bu düğümlerin veya nesne tanımlayıcı bileşenlerin (arc) sıralı bir listesi ise nesne id'sini, yani OID'yi oluşturur.

Örnek:

```
{joint-iso-itu-t(2) ds(5) attributeType(4)
        distinguishedName(49)}
```

Kayıt ağacı tamamen merkezi olmayan bir şekilde yönetilir. Bir düğüm, çocukları üzerinde tam yetkiye sahiptir. Alt düğümleri kendisi tanımlayabilir. OID Projesi, birçok OID hakkında (herhangi bir ITU-T Tavsiyesinde tanımlananların tümü dahil) bilgi toplayan bir havuz oluşturmuştur, ancak tahsis edilen tüm OID'leri (özellikle dünya çapında veya şirket özellerindeki) bu havuzda toplayabilmek mümkün değildir. Bu örnek çalışma kapsamında tanımlanması gereken OID, sonraki bölümlerde detaylandırılmıştır. (Bkz. 4.10.3 Unregistered OID tanımlama)

2.8 CAdES İmzalanan Özellikler (Signed Attributes)

Çalışmamızda yer alacak olan imzalanması gereken konum verisi için ilk aşamada en esnek yöntem CAdES formatına uyumluluğunu sağlamaktır. Bu uyumluluk için CAdES standardının bize sağladığı "İmzalanan Özellikler" (Signed Attributes) alanlarının kullanılması planlanmıştır. Bu alan, imza paketindeki imza değerinin oluşturulmasında hangi alanların kullanıldığını belirtmek için tanımlanmaktadır.

2.9 Zaman Damgası

Herhangi bir veri parçasına eklenebilen zaman damgasının amacı, ilgili verinin belirtilen zamandan önce var olduğunu kanıtlamaktır.[22] Zaman damgası, damgalanacak verinin özetini içeren bir dijital imzadır, güvenilir bir otorite tarafından (genellikle bir ESHS) oluşturulur. Zaman damgasında kullanılan imzanın ESHS sertifikası ve politikası zaman damgası operasyonuna özel olmalıdır. Zaman damgasını veren kurum, özet verisinin güvenilir bir algoritmayla oluşturulup oluşturulmadığı gibi bilgileri de damga isteğindeki OID bilgisiyle kontrol eder. Zaman damgasında bulunan zaman değeri UTC formatındadır. Saniye bazlıdır ve "Zulu" (Z) zamanı olarak da bilinir.

2.10 CAdES-T ve CAdES-XLONG

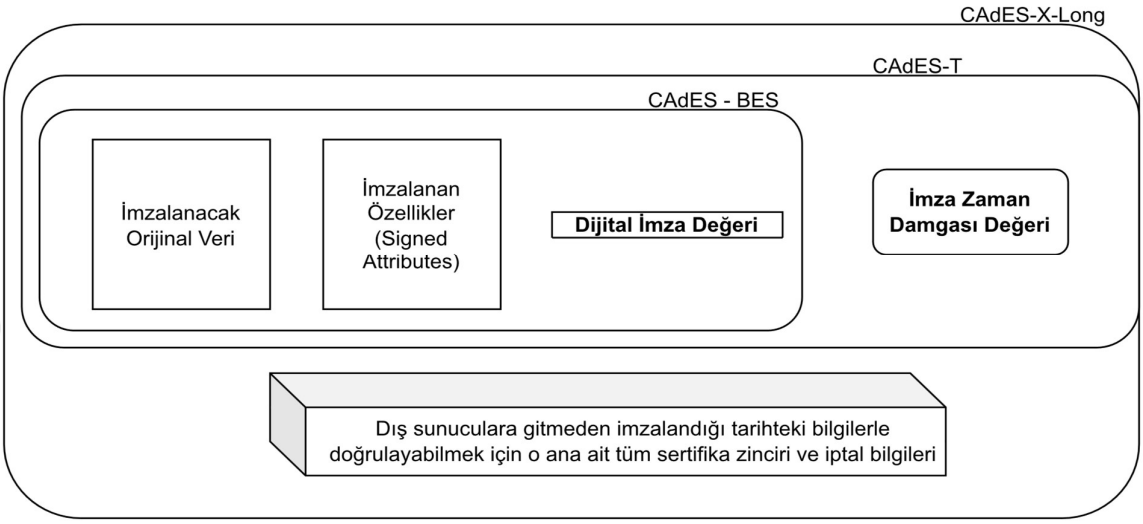
ES-T (Electronic Signature with Time) imza, BES imza temelinin üstüne tasarlanmıştır. BES imzadan farklı olarak beyan edilen zaman imza atan kişinin cihazından değil, bağımsız güvenilir bir otorite sunucusundan sağlanır. Bu sağlanan zaman damgası ile imzanın oluşturulma zamanı kanuni olarak güvence altına alınır.[23] İmza oluşturulma zamanı güvence altına alındığı için imza oluşturulduğu anda sertifika ve imza geçerliyse, sertifika iptal olduktan ya da süresi dolduktan sonra da geçerlidir. [15]

Zaman damgası alma süreci istemci ve CA'ya bağlı bir zaman damgası sunucusu arasında şu şekilde işler: [24]

1. İstemci zaman damgası sunucusuna bir istek göndererek zaman damgası token'ı ister.
2. İstemci, yanıtta (genellikle bir TimeStampToken (TST) içeren TimeStampResp) hata yoksa, zaman damgası belirtecindeki çeşitli alanları ve dijital imzanın geçerliliğini doğrular. Özellikle, zaman damgası atılan verinin, zaman damgası atılması istenen veriyle eşleştiğini doğrulamalıdır.
3. İstemci, zaman damgası belirtecinin doğru zaman damgası sertifikası, doğru veri özetini ve doğru özet algoritma OID'sini içerdiğini doğrulamalıdır.
4. Ardından, yanıtın zamanını, yerel zaman referansına karşı yanıtta yer alan zamanı doğrulayarak veya yanıtta yer alan tek seferlik değeri istekte yer alan değere karşı doğrulamalıdır. Yukarıdaki doğrulamalardan herhangi biri başarısız olursa, zaman damgası token'ı reddedilmelidir.
5. Ardından, zaman damgasının sertifikası iptal edilmiş olabileceğinden, sertifikanın durumunun hâlâ geçerli olup olmadığı kontrol edilmelidir.
6. Sonra, istemci uygulaması, token'ın verildiği imza politikasının uygulama için kabul edilebilir olup olmadığını belirlemelidir.
7. Zaman damgası sunucusu zaman damgası token'ını sadece bu amaç için ayrılmış bir anahtarla imzalamak zorundadır.

Bir imzanın uzun dönemli doğrulanabilmesi için zaman damgası ile güvence altına alınması yeterli değildir. İmzanın doğrulanabilmesi adına ilgili andaki iptal kontrol verileri ve sertifika zincirindeki tüm sertifikalara ihtiyaç duyulmaktadır. Bu sebeple ES-T imzanın daha kapsamlı hali olan ES-XLONG imza atılması önerilmektedir. Bu

çalışmada önereceğimiz modelde de oluşturduğumuz nihai imza paketini XLONG olarak kaydedilmesi süreç için önem taşımaktadır.[15]



Şekil 2.1 CAdES Zaman Damgası Yapısı

Bahsedilen iptal kontrol verileri, SİL (Sertifika İptal Listesi) dosyası ve ÇİSDUP (Çevrimiçi Sertifika Durum Protokolü) servis cevaplarıdır. Bu bilgiler için ulaşılması gereken URL bilgileri X.509 uyumlu sertifikalar içinde yer almaktadır.[25]

Zaman damgasını bağımsız taraflar aşağıdaki adımlarla doğrular:

- Zaman damgası sunucusundan gelen cevabın imzası kontrol edilir.
- Zaman damgası sertifikasının doğrulama işlemi yapılır.
- Sunucuya gönderilen özet bilgi ile gelen cevap içindeki özet bilgi karşılaştırması yapılır.

2.11 Uzun Dönem Saklama

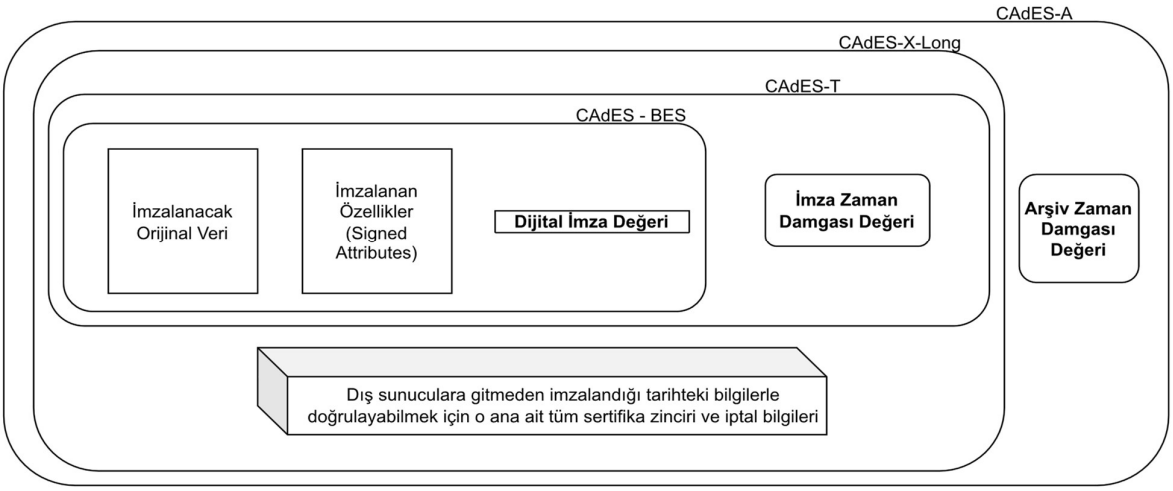
Dijital ortamda belli bir donanıma ve yazılıma göre oluşturulan verilerin yaşam süresini uzun tutabilmek; gelişen teknoloji, değişen ihtiyaçlar ve standartlar, çevrimdışı cihazların periyodik bakım ihtiyacı, çevrimiçi sistemlerin hizmet ömrü gibi birçok faktöre bağlı olarak daha ilkel yöntemlerle karşılaştırıldığında zorlaşmaktadır. Erişim kolaylığı ve güvenlik sebebiyle giderek daha çok kritik iş süreci, belge yönetim sistemlerine entegre edilmektedir. Belge yönetim sistemlerinde kullanılan e-imza protokolleri ve standartlarının, sadece günlük süreçleri işletmek için çalışması yeterli değildir. Çoğu

zaman doğrulanmasının üstünden yıllar geçmiş bir verinin tekrar kontrol edilmesi ihtiyacı ortaya çıkmaktadır. Bu tür durumlarda verinin imza paketi hala doğrulanabilir olmalıdır.

Burada genel olarak dikkat edilmesi gereken faktörler şunlardır[26]:

- Saklama ortamının yaşam süresi
- Anahtar ve sertifikaların yaşam süresi
- İmzalama yöntemi, anahtar uzunluğu ve kriptografik algoritmaların yaşam süresi
- Sertifika, imza ve belge formatlarının yaşam süresi
- Doğrulama için güvenilmesi gereken makamların hizmet süresi

Oluşturulan imza paketinin uzun dönem saklanması gerekmekteyse, periyodik aralıklarla arşiv zaman damgası eklenerek CAdES-A (Archival Form) tipinde yükseltilmesi gerekmektedir. Çünkü imza paketi ilk aşamada X-LONG tipinde oluşturulmuş olsa da zamanla alınan zaman damgalarının sertifikalarının süresi geçecek veya kriptografik yöntemlerin güvenilirliği zayıflayacaktır. Tüm paketi kapsayacak şekilde tekrar güncel zaman damgası almak, paketin teorik olarak süresiz şekilde saklanmasını sağlayacaktır.



Şekil 2.2 CAdES-A Yapısı

3. KÜRESEL KONUM BELİRLEME SİSTEMLERİ

Güvenilir bir konum damgasından bahsedebilmek için konum verisinin nasıl hesaplandığını ve hesaplamayı yapan modern cihazların hangi altyapılara ihtiyacı olduğunu incelemek gerekmektedir.

3.1 Konum Belirleme Sistemlerinin Geçmişi

Tarih boyunca insanoğlunun yeryüzündeki aktivitelerinin kapsadığı alan genişledikçe ve karmaşıklaştıkça, konum belirleme ihtiyacı doğmuş ve önem kazanmıştır. Tüm konumlama çözümlerinin temelinde uzaktaki cisimlerin gözlemlerinden yararlanılarak yeryüzündeki gözlemcinin kendi bulunduğu bölgeyi hesaplaması yatar.

Bu hesaplamalar iki kategoride ele alınabilir. Navigasyon ve haritalama ölçümleri (surveying). Haritalama yöntemleri yeryüzündeki sabit noktaların birbirine göre konumlarının hesaplanmasıyla oluşan modelleri kullanır. Navigasyon ise hareketli bir cismin konumunun belirlenerek belirli bir hedefe yönlendirilmesi bilimidir. İkisi de benzer alanlardır ancak zaman kavramı navigasyonda daha önemli bir yer tutar.

Kronolojik olarak tarihte yeryüzü haritalama yöntemlerine örnek olarak; Eski Mısırlıların taşan Nil nehrinin yok ettiği yapıların restorasyonunda ve tamirinde uzak kontrol noktalarını kullandıkları hesaplamalar, Eratosthenes'in dünya çapını hesaplamak için kullandığı açısal hesaplama ve daha sonradan Yunan ve Romalıların şehirleşmede kullandıkları yöntemler, ardından Hollandalı Snell van Royen'in uzak bölgelerdeki noktaların koordinatlarını hesaplamak için ilk kez birbirine bağlı üçgenlerin iç açılarını kullanması, daha büyük ölçekli nirengi çalışmalarında ise 18. yüzyılda Fransız Picard ve Cassini'nin Dunkirk'ten Collioure'yi uzanan hattın belirlenmesi verilebilir.

Nirengi (triangulation) yöntemi daha sonradan kıtasal ölçekteki koordinatların belirlenmesinde ana yöntem olmuştur. [27]

Navigasyon tarihinde ise milattan önce 3000-1100 arasında Akdeniz'de Girit adasında yaşayan Minoslular gibi antik denizciler yönlerini bulabilmek için takımyıldızları kullandıklarını belirten kayıtlar bırakmışlardır. Ortaçağ'da ve coğrafi keşiflerde de aylarca uçsuz bucaksız okyanuslarda seyreden gemiler için de ufuk çizgisi ile bir yıldız veya

gezegen arasındaki açıyı ölçmeye yarayan sekstant gibi araçlar çok kritik öneme sahip olmuştur.

3.1.1 Uzay tabanlı sistemler

Öte yandan günümüzün küresel ölçekli, yüksek hassaslık gerektiren konumlandırma ve navigasyon ihtiyaçlarını karşılayabilecek çözüm ancak uzay çağıyla gelmiştir. Gezegenimizin yörüngesine oturtabildiğimiz yapay elektronik uydular sayesinde daha önce eşi benzeri olmayan ölçüm olasılıkları ortaya çıkmıştır.

Uydu tabanlı bu sistemler ikiye ayrılabilir: Optik teknikler ve radyo teknikleri.

3.1.1.1 Optik teknikler

Elektromanyetik spektrumun gözle görülebilir kısmında kalan kısmını kullanan bu yöntemler, ortaçağdaki kullanıma benzer şekilde yine bir sekstant benzeri araç veya uydu lazer menzilleme (SLR) yöntemi yardımıyla yörüngedeki uydunun astronomik pozisyonunun hesaplanmasını içerir. SLR yönteminde bir yansıtıcıya sahip olan uydulara, yeryüzünden gönderilen bir lazer ışınının gidiş-geliş süresi ölçülür. Günümüzde bu yöntemin yerini başkaları almıştır. Fakat hala uydu yörüngelerinin takip edilmesi, jeodezik konumlama ve gök mekaniği konularında önemli yer tutmaktadır.[28]

3.1.1.2 Radyo teknikler

Günümüzde en yaygın olarak, uydular ve yeryüzü alıcıları arasındaki iletişimi radyo sinyalleriyle sağlayan uydu ağları kullanılmaktadır. GPS'ten önceki en başarılı uydu tabanlı sistem Transit idi. [29] Küresel olarak hizmet veren ilk uydu tabanlı konumlama sistemi olan Transit, Sovyetler Birliği'nin Sputnik 1 uydusunu takip etmeye çalışan ABD'nin, uydunun yaydığı sinyallerin doppler frekans kaymasının bilinen bir konumda hesaplanmasıyla uydunun yörüngesinin bulunabileceğini fark etmesiyle doğdu. 1959'da Transit sisteminin ilk uydusu fırlatıldı. Başta gizli olsa da, 1967'de sivillerin hizmetine de açıldı. 1996'da daha yüksek performansa sahip GPS'in devreye alınarak projenin kapatılmasına kadar hizmet vermeyi sürdürdü.

Sovyetler Birliği de bu dönemde Transit ile eş zamanlı olarak Tsikada ve askeri versiyonu olan Parus projeleri geliştirmişlerdi.[30] Bu sistemlerin de küresel çapta tam hizmet

verebilen ikinci GNSS (Küresel Navigasyon Uydu Sistemleri) olan GLONASS (Global'naya Navigatsionnaya Sputnikova Sistema) devreye alındığında hizmetleri son buldu.

Bu yeni sistemler, Doppler gözlemleri yerine menzil ölçümleri kavramına dayanıyordu ve sürekli kapsama alanı sunan farklı bir uydu ağı tasarımı kullanıyordu. Bu yeni kavramlar, dünya çapında anlık konumlandırmanın yanı sıra doğrulukta dikkate değer bir artış sağladı. [28]

3.2 Uydu Tabanlı Konumlama Prensipleri ve Ölçüm Tipleri

GNSS sinyalleri ışık hızında yol alan elektromanyetik dalgalardır. Bu sinyaller için radyo spektrumundaki 1.2-1.6 GHz bandı (L bandı) seçilmiştir. Bunun nedeni, bu frekansın yeterli hassaslıktaki hesaplamaların yapılmasını sağlarken aynı zamanda atmosferdeki hava koşullarından daha az etkilenmeleri ve daha basit kullanıcı ekipmanıyla işlem görebilmeleridir. [28]

GNSS, kullanıcı konumunu belirlemek için değişen varış zamanı (TOA) kavramını kullanır.

Bu hesaplama, bilinen bir yerde bir verici tarafından iletilen bir sinyalin bir alıcıya ulaşması için geçen sürenin ölçülmesini gerektirir.

Sinyal yayılma süresi olarak anılan bu zaman aralığı, vericiden alıcıya olan mesafeyi elde etmek için sinyal yayılma hızıyla (ışık hızı) çarpılır. Alıcı, bilinen konumlardaki çoklu vericilerden (yani navigasyon uydularından) yayınlanan sinyallerin yayılma süresini ölçerek konumunu belirleyebilir. İkiden fazla uyduyla yapılan ölçümlerde üç boyutlu pozisyonlama sağlanabilir. [31]

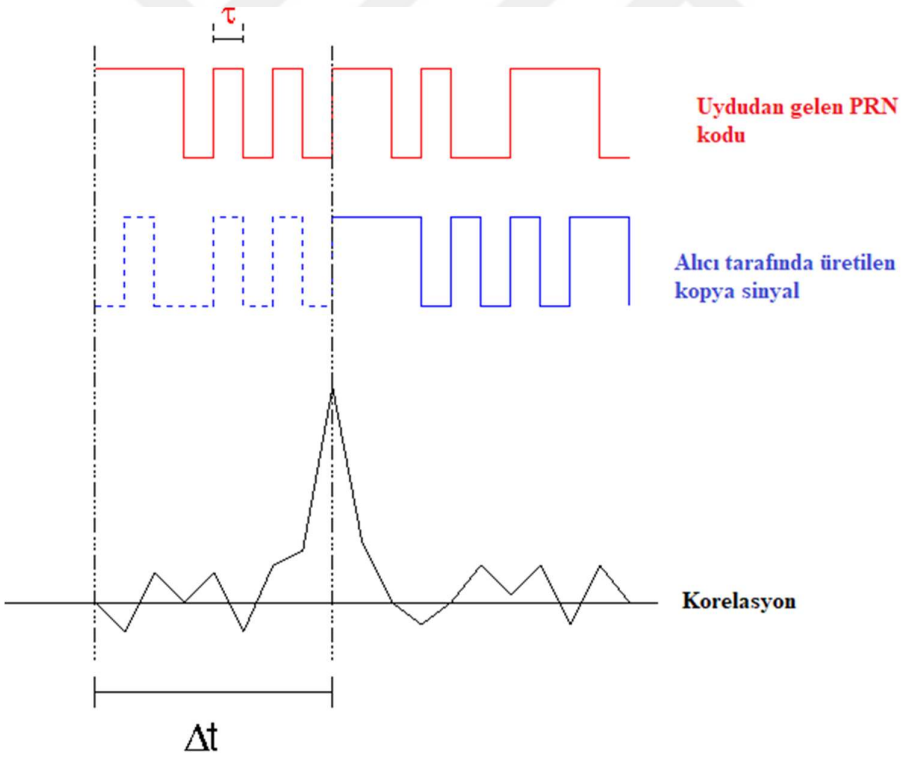
Uyduların yaydığı sinyallerde bulunan ana ögeler şöyledir:

- **Taşıyıcı (Carrier):** Belli bir frekansta sinüzoidal sinyal. (Örnek: GPS için L1 - 1575.42 Mhz)
- **Menzilleme Kodu**: Rastgele bit dizileri içeren ve uydunun sürekli tekrar ettiği sinyal. Bu rastgele bit dizisine aynı zamanda "Pseudo-Random Noise (PRN)" ismiyle PRN kodu da denir. PRN kodları bir yayılım kodu (spreading code) tipidir.

- **Navigasyon verisi**: Uydu ile ilgili bilgilerin (efemeris verisinin) ikili tabanda kodlanmış halde paylaşıldığı sinyal. İçeriğinde: uydu saati, uydu sağlık durumu, almanak (diğer uydular), uydu konumu ve hızı gibi bilgiler bulunur.

Bu sinyallerle alıcı tarafında aşağıdaki hesaplamalar yapılır:

- **Menzil belirleme (pseudorange)**: Sinyal alımındaki alıcı saati ile sinyal iletimindeki uydu saati arasındaki süre farkının (Δt) ışık hızıyla çarpılması ile elde edilir. Δt değeri uydudan gelen sinyal ile alıcının oluşturduğu PRN sinyali kopyasının senkron olması için gereken zaman bazında kaydırma miktarıdır. İki saatin senkron olamaması ve diğer bazı gecikmeler dışında, uydu-alıcı arasındaki mesafe dm hassaslığında ölçülebilir. [28]



Şekil 3.1 Menzil Belirleme Hesaplaması

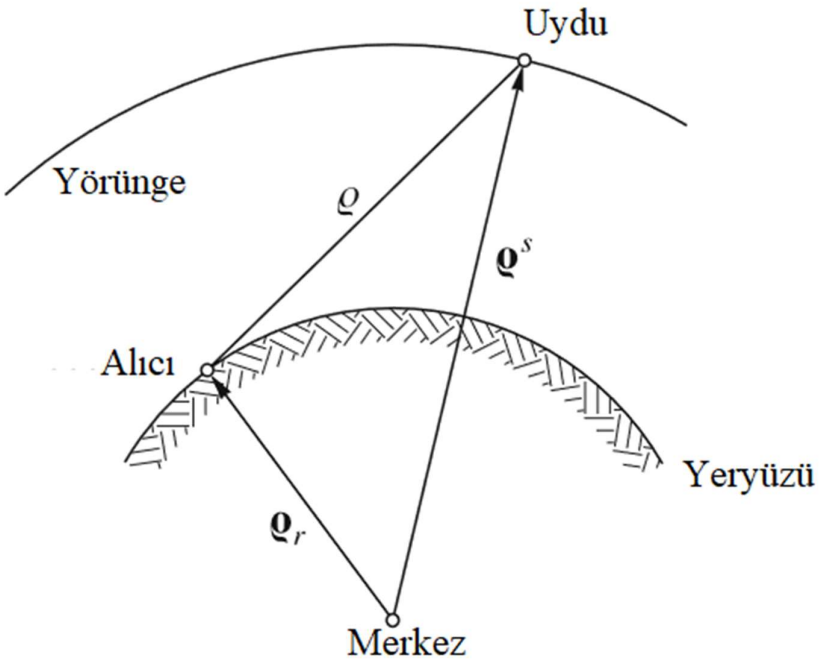
- **Taşıyıcı faz:** PNR kodunun yanı sıra, uydu ile alıcı arasındaki mesafeyi elde etmek için taşıyıcı fazın kendisi de kullanılır. Bu taşıyıcı faz ölçümleri, PNR kod ile yapılan menzil ölçümlerinden çok daha kesindir (tipik olarak 100 kat

daha kesindir), ancak bu hesaplamada belirsizlik ihtimali vardır. Bu belirsizlik, alıcı sinyali her kaybettiğinde, atlamalara veya aralık süreksizliklerine neden olarak keyfi olarak değişir. Dalga boyunun bilinmeyen bir tamsayı katıyla (λN) aynı sonuçlar verebilir.

- **Doppler kayması**: Doppler efekti nedeniyle, alıcının ölçtüğü sinyaldeki değişim, alıcının hızının ölçümünde kullanılır.

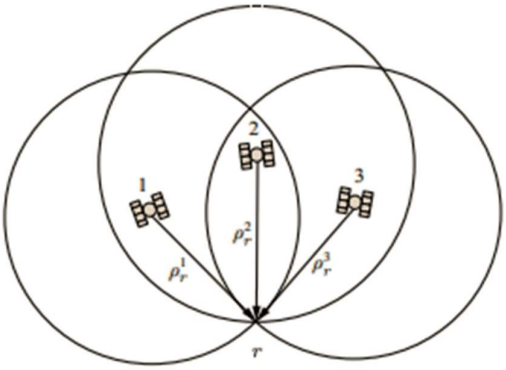
3.2.1 Uydu menzil bilgileri ile konum belirleme

En ideal durumda uydu menzilleri (pseudorange) yardımıyla pozisyon belirlemek için 3 ayrı uydu gerekmektedir. Bunu açıklamak için uydu sayılarını giderek artırarak konumun nasıl kesinleştiğini görebiliriz. Tek bir uydudan sinyal aldığımızı varsayalım. Uydudan gelen sinyal bize uydunun konumu hakkındaki bilgileri vermektedir. Aynı zamanda sinyalin bize ne kadar sürede ulaştığını da hesaplayabilmekteyiz. Bu iki bilgiyle beraber merkezi uydu olan bir menzil küresi çizebiliriz.



Şekil 3.2 Uydu Uzaklık Hesaplaması

Alıcı bu kürenin sınırında herhangi bir yerde yer almak zorundadır. İkinci bir uydu eklediğimizde, aynı şekilde bir küre daha eklemiş oluruz. Bu iki kürenin kesişimi bize bir daire verir. Fakat alıcının 3 boyuttaki kesin konumu için bir uyduya daha ihtiyaç vardır. 3. uyduyla gelen 3. menzil küresi, önceki daireyle kesiştiğinde elimizde iki nokta oluşur. Bu noktalardan biri dünya yüzeyinde olamayacağından görmezden gelinebilir. Diğer nokta ise alıcı konumudur.



Şekil 3.3 3 uydu ile nirengi (triangulation)

3.2.2 Alıcı saatinin sapması

Gerçekte ise alıcıların saati, GNSS uydularının saatleri ile çoğu zaman milisaniye düzeyinde senkronize değildir. Uydu konumlama sistemlerinde zaman çok önemlidir. Işık hızının saniyede yaklaşık 300.000 km olduğu göz önüne alındığında 1 milisaniyelik bir saat sapması, konumun 300 km hatalı çıkmasına yol açacaktır. Bu nedenle gerçek şartlarda her bir bilinmeyen ve hata toleransı için olabildiğince fazla uydu sinyaline erişim gerekmektedir. Örnek olarak bu saat sapmasından gelen zaman farkının alıcı tarafında hesaplanarak düzeltilebilmesi için 1 uyduya daha gerek vardır, böylece minimum 4 uydu sinyaline ihtiyaç oluşmaktadır.[27]

3.3 GNSS Bölümleri

Sahada amaçlandığı gibi çalışan bir uydu konumlama sistemi günümüzde üç parçadan oluşuyor. Uyduları içeren uzay bölümü, sistemi kontrol eden ve yön veren organizasyon merkezini içeren kontrol bölümü ve uydu sinyallerinden yararlanan alıcıların bulunduğu kullanıcı bölümü.

3.3.1 Uzay bölümü

Sürekli bir küresel konumlandırma hizmeti verebilmek için her bir GNSS organizasyonu yeterli sayıda uydu geliştirip, yörüngeye doğru optimizasyonlarla oturtmalıdır. Bunlardan en temel gereklilik ise, herhangi bir zamanda Dünya'nın herhangi bir yerinde en az 4 uydu sinyalinin alınabiliyor olmasıdır.

Bu optimizasyonların tasarım kriterleri; kullanıcı pozisyon kesinliği, uydu erişilebilirliği, hizmet kapsamı, uydu geometrisidir. Ayrıca uydunun oturduğu yörüngenin Dünya'ya olan uzaklığı, uydunun sağlaması gereken sinyal enerjisini de etkileyecektir.

GNSS uydularında, temel olarak atom saatleri, radyo alıcı-vericileri, bilgisayarlar ve uydu yörüngesini gerektiğinde düzeltebilmek için kullanılan çeşitli yardımcı ekipmanlar bulunur.[31]

3.2.2 Kontrol bölümü

Kontrol bölümü (yer bölümü olarak da adlandırılır) tüm sistemin yönlendirilmesinden sorumludur. Görevleri, sistemin devreye alınmasını ve bakımını, yörünge ve saat parametrelerinin belirlenmesi ve uyduların izlenmesini, yardımcı verilerin (örneğin, iyonosfer parametreleri) izlenmesini ve uydulara veri yüklenmesini içerir.

Kontrol bölümü ayrıca verilerin şifrelenmesinden ve hizmetlerin yetkisiz kullanıcılara karşı korunmasından da sorumludur. Genel olarak kontrol bölümü, tüm faaliyetleri koordine eden bir ana kontrol istasyonu, izleme ağını oluşturan izleme istasyonları ve uydulara iletişim bağlantısı olan yer antenlerini içerir.[28]

3.2.3 Kullanıcı bölümü

Kullanıcı kategorileri, askeri ve sivil kullanıcılara ek olarak yetkili ve yetkisiz kullanıcılar olarak ayrılmıştır. Sivil ve yetkisiz kullanıcılar, GNSS'in tüm sinyallerine veya hizmetlerine erişemez. Bugün piyasada çok çeşitli alıcı türleri bulunmaktadır. Alıcılar takip edebildikleri frekans aralıklarına ve sayısına göre fark gösterir.

3.4 GNSS Sinyal Güvenliği

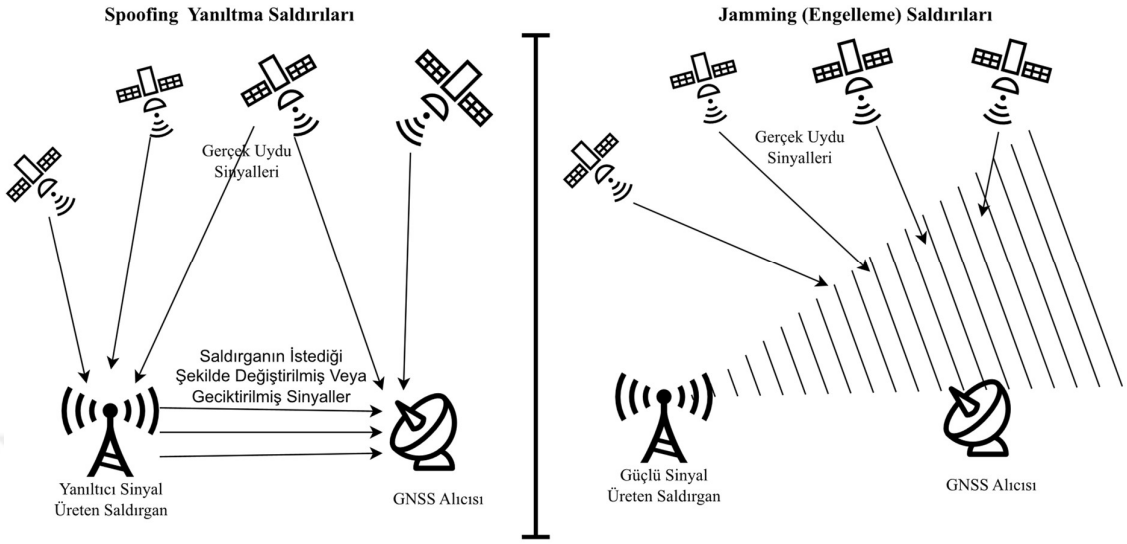
Uydu tabanlı küresel konumlama sistemleri modern toplumun günlük yaşantısında, şirketlerin büyük çaplı projelerinde, askeri harekatlarda çok temel bir yere oturmuştur. Dolayısıyla alıcıların yorumladığı sinyallerin güvenilirliği kritik önem taşımaktadır.

GNSS sinyallerinin sivil kullanıma izin verilen kısmıyla alakalı mesaj formatı ve modülasyon gibi teknik bilgiler, alıcıların sağlıklı işlev görebilecek şekilde üretilmeleri ve kullanılabilmeleri için halkın erişimine açıktır.[32] Sivil GNSS'in Arayüz Kontrol Dokümanı (ICD) sivil uydu navigasyon sinyalleri için taşıyıcı frekansı, modülasyon modu, navigasyon mesajı gibi ilgili parametreler hakkında detaylı açıklamalar ve hesaplamalar içermektedir. [33]

3.4.1 Telsiz engelleme (jamming) saldırıları

Yeryüzüne ulaşan GNSS sinyallerinin gücü, -163dbW ile -152dbW arasında değişir. Bu, 16,000 kilometre uzaktaki 25W ışık lambasının gücüne benzetilmiştir[34]. Gerçek uydu sinyalleri bu denli zayıf olduğu için, yerel olarak yayınlanan yeterince güçlü bir sinyal, uydu sinyallerini kolayca bastırabilir. Dört uydu ağının frekansları, 1164 MHz (Galileo E5a) ile 1602 MHz (Glonass L1) arasındaki 15 bandta yer almaktadır. Bu aralıkta sinyal üretebilen basit GPS engelleyiciler internette 15 dolara kadar satın alınabilir.[35] Engelleyiciler, otoyol ücretlerinden kaçınmak veya şirket araçlarının gerçek rotasını işverenlerden gizlemek için kullanılır.

Telsiz engelleme saldırısı alınan sinyallerdeki gürültü oranını oldukça artırır ve GNSS alıcısının sağlıklı çalışma yeteneğini azaltır veya tamamen kaybetmesine neden olur. Telsiz engelleme saldırısına maruz kalan GNSS alıcısı, rutin olarak takip edilmesi gereken uydu sinyallerini alamaz. Fakat alıcı, antenlerini ayarlayarak bloke eden sinyali, alınan sinyalin tamamından çıkararak, sinyal-gürültü oranını artırabilir. Bu nedenle, telsiz engelleme saldırısı alıcılar tarafından tespit edilebilir ve karşı önlemler ile direnç sağlanabilir.



Şekil 3.4 Engelleme ve Yanıltma Saldırıları

3.4.2 Engelleme saldırıları örnekleri

En yaygın engelleme tekniği, bilinen GNSS frekanslarında, özellikle L1'de sürekli veya 'cıvıldama' (testere dişi örüntüsü) şeklinde bir gürültü sinyali yayınlamaktır [36] Alternatif olarak, GNSS PRN kod sinyalleri de, veri olmadan, anti-engelleme filtrelerinden geçebilir şekilde gürültü olarak kullanılabilir [37].

Engelleme saldırıları, uçakların yön belirleme sistemleri için de tehlike oluşturur. ABD ve İngiltere gibi ülkelerde yasadışı GPS engelleyicilerini tespit etmek için ulusal ölçekte ağlar ve izleme sistemleri kurulmaktadır. İngiltere'nin Sentinel sistemi 2013 yılında Ekim-Aralık dönemi boyunca bir havaalanında yüzlerce engelleme saldırısı olayını bildirmiştir.[35] Benzer şekilde Tayvan'daki Kaohsiung Uluslararası Havaalanı'nda günde yüzlerce GPS/Galileo girişim (interference) olayı bildirilmiştir.[38] Bu tür girişim olayları ABD'de birkaç havaalanında [39] ve Londra finans bölgesinde tespit edilmiştir.

Özet olarak engelleme saldırıları, düşük maliyetli teknik ekipmanla yapılabilecek en etkili saldırıdır. Fakat alıcı antenlerinin ayarlanmasıyla kolayca tespit edilip engellenebilir.

3.4.3 Sahtecilik (spoofing) saldırıları

Sahtecilik saldırıları, engelleme saldırılarından daha inceliklidir ve bir zamanlayıcıyı veya navigasyon alıcısını gerçek sinyalden kendine yönlendirmek için doğru güçle sahte

bir sinyalin üretilmesine dayanır. Sadece 1,1'lik bir güç oranı (sahte / orijinal oranı), alıcıyı sahte sinyallere yükseltmek için yeterlidir.[40]

Gerçek uydu sinyalleri ile sahte uydu sinyallerini ayırmak için sinyallerin belli özelliklerindeki tutarsızlıklara dikkat edilmelidir. Bunlar; zamanlama, sinyal yönü, güç, doppler kayması ve sinyal-gürültü oranıdır.

Modern alıcıların çoğu bu farklılıkları algılayacak donanıma sahip değildir. Otomatik kazanç kontrolünün (AGC) GNSS alıcılarında yaygın olarak kullanılması, dalgalanan sinyal gücünü telafi ederken aynı zamanda alıcıları yanıltmaya karşı daha savunmasız hale getirir. [41], [42] Ayrıca, bir alıcının sinyalin yerel bir yanıltma kaynağından mı yoksa gerçek bir uydudan mı geldiğini söylemesinin hiçbir yolu yoktur. [1]

Bu nedenlerden ötürü eğer alıcı, aldığı sinyallere uygulayabileceği bir kimlik doğrulama altyapısına sahip değilse, bir yanıltma saldırısına maruz kalıp kalmadığının güvenilir bir kanıtını oluşturamaz. Bu kanıt oluşturulamadığı için, alıcının hesapladığı pozisyonlama ve zamanlama bilgilerinin yanlış olabileceği her zaman iddia edilebilir.

Çoğu güncel sinyal alıcısı sadece uydulardan navigasyon mesajı alabilmektedir, sistemin kontrol merkezine bağlanamamaktadır. [31] Bu nedenle operasyonel bir kimlik doğrulama sisteminin sadece uydu sinyallerini temel alabilmesi gerekmektedir.

3.4.3.1 Sahte iletim (forwarding spoofing, meaconing) saldırısı

Sahte iletim saldırısında, gerçek uydu sinyalinin alıcıya doğru zamanda ulaşması saldırgan tarafından engellenir. Saldırgan alıcıya gerçek sinyali iletir fakat iletmeden önce bir süre geciktirir. Bu gecikme süresi yüzünden alıcı, sinyal yayılma süresini ve dolayısıyla menzili (pseudorange) hesaplarken yanlış sonuca ulaşır. [43] Gecikme süresi dışında sinyal üzerinde bir değişiklik yapılmamıştır. Birkaç milisaniyelik gecikme binlerce kilometre farklı konum sonucuna neden olabilir.

Bu saldırı sivil yayında daha olasıdır. Şifreli askeri sinyal durumunda zordur, çünkü yayın daha uzun bir PRN dizisi üzerine modüle edilmiştir.

SCER veya "seçici gecikme" olarak adlandırılan bir sahte iletim saldırısı çeşidi ise saldırganın sinyali geciktirirken aynı zamanda yalnızca konum bilgisini seçerek manipüle etmesini ve böylece bir saldırı başlatırken zaman sıçramasından kaçınmasını sağlar. [44], [45]

3.4.3.2 Sahte üretim (generating spoofing) saldırısı

Sahte sinyal üretimi saldırısı gerçekleştirmek isteyen saldırgan, navigasyon uydusunun mesaj formatı konusunda kapsamlı bilgiye sahip olmalı ve aynı zamanda uydu sinyal yapısına da hakim olmalıdır. Saldırgan kendisinin ürettiği sinyalleri, uydu sinyaliymiş gibi alıcıya iletir. Amacı alıcının sahte sinyali bilinçsizce yakalaması ve gerçek sinyal olarak izlemesidir. Bu tür bir saldırıda, yanıltma sinyalini kademeli olarak farklılaştırıldığında alıcının fark etmeden yanlış hız ve konum sonucuna varmasına neden olur. [46]

3.4.3.3 Sahtecilik saldırılarına örnekler

İran, Aralık 2011'de ABD'ye ait RQ-170 "Sentinel" insansız hava aracının GPS ekipmanına, insansız hava aracının inişini kontrol etmek için yanıltma saldırısı kullanarak müdahale etmiştir. İran ordusu, ABD insansız keşif uçağının yanlış talimat almasını sağlamak için yanıltıcı saldırılara başvurmuştur. Kontrol bölümünden alıcıya iletilen mesaj bilgisini başarıyla değiştirerek insansız keşif uçağının orijinal operasyon rotasından sapmasına neden olmuştur. [47]

Haziran 2017'de ABD Okyanus İdaresi, Rusya'nın Yeni Rossik limanı yakınlarındaki en az 20 geminin yanlış GPS konumlandırmasına sahip olduğunu tespit etmiş ve Rus tarafının muhtemelen GPS sahtekarlığı için yeni teknolojileri test ettiğini belirtmiştir. [48]

Texas Üniversitesi'nden Profesör Todd Humphreys liderliğindeki araştırma ekibi, 2013 yılında yalnızca bir dizüstü bilgisayar, küçük bir anten ve ucuz bir GPS sahte sinyal üretici kullanarak bir süper yatın navigasyon sistemini başarıyla kontrol etmiştir.[49]

2017'de Stanford Üniversitesi, Texas Üniversitesi ve Cornell Üniversitesi gibi araştırma birimleri, GNSS alıcılarının sahtekarlığa karşı oldukça savunmasız olduğunu ve kullanıcıların sahtekarlık saldırılarının varlığını tespit etmesinin zor olduğunu deneylerle göstermiştir. [50]

3.5 GNSS Sahte Sinyal Saldırılarının Engellenmesi

Bu çalışmada olduğu gibi kullanıcının bulunduğu konumun kanıtlanmasının kritik olduğu durumlarda, GNSS sinyalinin gerçek uydu sisteminden doğru zamanda geldiği ve üstünde oynanmadığının şüpheye yer bırakmayacak kesinlikte kaydedilmesi büyük önem taşımaktadır. Bu amaçla alıcının alabileceği bazı önlemler bulunmaktadır: [1]

- Sinyal işleme savunmaları
- Kriptografik savunmalar
- Diğer zamanlama kaynaklarıyla korelasyon
- Radyo spektrumu ve anten savunmaları

Çalışmamızda kriptografik önlemlere odaklanacağız. Çünkü diğer çözümler sadece alıcı cihazın kendi donanımında ve o an için geçerlidir. Alınan sinyalin, hesaplanan konum verisinin, alıcıdan bağımsız kişilerce doğruluğunun kanıtlanmasını sağlayamazlar.

Günümüzde navigasyon mesajı doğrulama için çoğu GNSS kendi organizasyonlarıyla çözümler üretmeye ve devreye almaya başlamışlardır. Örnek olarak GPS üzerinde devreye alınması planlanan CHIMERA sistemi ve GALILEO bünyesinde sahadaki test sürecini bitirmek üzere olan OSNMA hizmeti bulunmaktadır.

CHIMERA, GPS için tasarlanmış, navigasyon mesajı doğrulamasını (NMA) ve yayılım kodu doğrulamasını (SCA) sağlayan bir GNSS kimlik doğrulama şemasıdır. Navigasyon mesajını dijital imza kullanarak doğrularken, yayılım kodları veya PRN'lere tahmin edilemez filigranlar eklenmesini sağlar.[51] Sistem, alınan sinyallerin meşru bir uydudan geldiğini garantileyerek sahtekarlık saldırılarına karşı koruma sağlamak üzere tasarlanmıştır.

CHIMERA, mevcut GNSS sistemleri ile geriye dönük uyumluluğu desteklerken ek güvenlik özellikleri sağlar. Ancak bu sistem henüz operasyonel olarak devreye alınmamıştır. Sivil alıcılar için testlerin, donanımı güncel uyduların 2023 sonunda fırlatılmasıyla başlayacağı planlanıyor. [52]

Küresel olarak kullanıma açılma tarihi daha yakın olan ve yaygınlaşması muhtemel olan, testleri tamamlanmış GALILEO OSNMA protokolü üzerinde duracağız.[53] Bu protokolün mesaj yapısı, kriptografik tasarımı ve GNSS sinyaline olan entegrasyonu incelenecek. Ardından önerdiğimiz e-imza standardı bu temele oturtulacak.

3.5.1 Yayın doğrulama yöntemleri

İletişim ağlarında veri iletimi için dört temel tek yönlü yayın modeli vardır. [54]

- **Tekli yayın (Unicast):** Bir kaynaktan, belirli tek bir alıcıya veri gönderimi. 1-1 iletim şeklidir.
- **Yayın (Broadcast):** Bir kaynaktan, hedef seçimi yapmadan, belirsiz birden çok alıcıya veri iletimi.
- **Çoklu Yayın (Multicast):** Bir kaynaktan, seçilen, önceden belirlenmiş birden çok alıcıya veri iletimi.
- **Herhangi Noktaya Yayın (Anycast):** Bir kaynaktan, potansiyel alıcılar arasında en yakın veya en uygun alıcıya veri iletimi.

GNSS sinyallerinin veri aktarım modeli, broadcast yayın tipindedir. Yörüngedeki uydular oluşturdukları sinyalleri tek yönlü olarak yeryüzüne gönderirler. Herhangi bir alıcı seçimi veya sayı limiti söz konusu değildir. Sinyalleri almak için gereken donanıma sahip olan tüm kullanıcılar verilere sahip olabilir.

Bu nedenle GNSS sinyal güvenliğinin, daha genel bir sorun olan "yayın verisinin gerçekten de istenen kaynaktan geldiğine nasıl emin olabiliriz?" ile kesiştiği birçok nokta vardır.

Yayın iletiminde, bir cihaz bilgi iletirken (kaynak), alıcı taraf gönderenin menzilindeki tüm cihazlardır. Alıcılar sonsuz olabileceğinden, alıcıların her birinin gönderenle birebir kimlik doğrulama yapması mümkün değildir. Bu nedenle, cihazların merkezi bir sunucuyla kimlik doğrulama yapmasını sağlayan kimlik doğrulama altyapıları bu senaryo için uygun değildir. Yukarıda belirtilen nedenlerden dolayı, yayın kimlik doğrulaması, mesaj kaynağının özgünlüğünü doğrulamak için cihaz kimliğinin doğrulanmasını değil, mesajın doğrulamasını (message authentication) tercih eder.

Fakat maalesef, mesaj doğrulama yapmak için paylaşılan bir gizli anahtar kullanılarak oluşturulan her pakete bir mesaj kimlik doğrulama kodu (MAC) eklemek güvenli yayın kimlik doğrulaması için yeterli olmayacaktır. Çünkü, MAC yöntemi simetrik şifrelemeye dayandığından ortak anahtara sahip herhangi bir alıcı verileri almak yerine, oluşturmayı de seçebilir ve gönderenin kimliğine bürünüp, göndereni taklit edebilir.

Asimetrik kriptografi çözümünde ise ilk akla gelen yöntem gönderilen her paketi dijital olarak imzalamaktır. Bu yöntemdeki sorun ise asimetrik işlemlerin hesaplama açısından pahalı olmasıdır. Hem imzalamak hem de doğrulamak, gereken süre ve bant genişliği açısından yüksek ek yüke sahiptir. Birkaç pakete tek bir imza atarak bu yükü azaltmayı deneyen birkaç şema önerilmiştir.[55]–[57] Fakat bu şemalar, bant genişliği ek yükü, işlem süresi, ölçeklenebilirlik, hizmet reddi (DoS) saldırılarına ve paket kaybına karşı dayanıklılık açısından yeterli değildir. Bir saldırganın alıcıya imza içerdiği varsayılan sahte paketler göndererek ciddi bir DoS saldırısı yapması genellikle mümkündür. İmza doğrulama genellikle hesaplama açısından pahalı olduğundan, alıcı sahte imzaları doğrulamaya çalışırken aşırı yüklenecektir.

Yayın kimlik doğrulaması sağlamaya yönelik başka bir yaklaşım, mesaj doğrulama kodlarını (MAC) kullanır. Bu yaklaşım simetrik kriptografiyi kullanır ancak gönderici tarafından anahtarların gecikmeli olarak açıklanması kritik öneme sahiptir.

3.5.2 Mesaj doğrulama kodları

Yayın sinyallerinin doğrulanmasında kullanılan protokolleri detaylı incelemeden önce bu protokollerin kullandığı ve sinyallerle iletilen dijital paketlere yerleştirilen MAC etiketleri (tag) konusunda bilgi sahibi olunması faydalı olacaktır.

Mesaj Doğrulama Kodu (MAC), bir mesajın bütünlüğünü ve doğruluğunu doğrulamak için kullanılan kısa bir bilgi parçasıdır. MAC, bir mesajın değiştirilmemiş olduğunu ve güvenilen bir kaynaktan geldiğini garanti eder. MAC'ler, genellikle mesaj bütünlüğü ve kaynak doğrulamanın önemli olduğu iletişim protokollerinde kullanılır.

Bir MAC algoritması, gizli bir simetrik anahtar ve bir girdi mesajını alır ve MAC değeri veya etiketi olarak adlandırılan sabit boyutlu bir çıktı üretir. Gönderici ve alıcı gizli anahtarı paylaşır ve MAC algoritması her iki taraf tarafından bilinir. Gönderici, bir mesajın MAC değerini hesaplar ve mesaja ekler. Alıcı, mesajı aldıktan sonra, aynı gizli anahtar kullanarak MAC değerini yeniden hesaplar ve bunu mesaja eklenen MAC değeriyle karşılaştırır. Hesaplanan ve alınan MAC değerleri eşleşirse, mesajın gerçek ve değiştirilmemiş olduğu kabul edilir. [58] En yaygın kullanılan MAC algoritmasına örnek olarak HMAC (Hash-tabanlı Mesaj Doğrulama Kodu) verilebilir. Etiketi oluşturmadan önce mesaj içeriği bir özet fonksiyonundan (örn: SHA-256) geçirilir.

3.5.3 TESLA protokolü

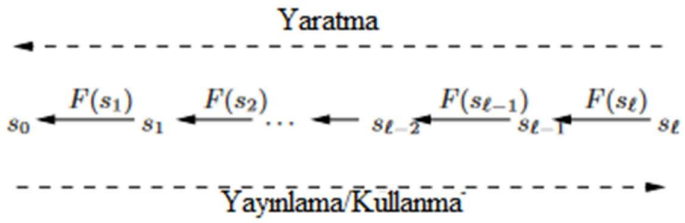
TESLA'nın ana fikri, göndericinin her pakete yalnızca kendisi tarafından bilinen bir *k* anahtarıyla hesaplanan bir MAC eklemesidir. Alıcı, alınan paketi kimliğini doğrulamadan hafızaya alarak bekletir. Kısa bir süre sonra, gönderici *k*'yi sonraki paketlerin içeriğinde ifşa eder ve alıcı paketin kimliğini doğrulayabilir.[59] Sonuç olarak, alıcı ve göndericinin saatlerinin senkronize olması koşuluyla, paket başına tek bir MAC yeterli olur. Çünkü k anahtarları ifşa edildiğinde gelen yeni paketler artık henüz ifşa edilmemiş $k_{(i+1)}$ anahtarlarıyla şifrelenmiş olmaktadır. Böylece *k* anahtarına sahip olan bir alıcının kendini gönderen gibi göstermek için zamanı olmaz.

Çalışmamızda sunacağımız kriptografik şemada ve GALILEO uydularının mesaj doğrulamasında bu protokol kullanıldığından, adımları daha detaylı incelemek yardımcı olacaktır.

3.5.4 Tek yönlü zincirler

Tek yönlü zincirler, bir değere arka arkaya tek yönlü bir fonksiyon uygulanarak oluşturulur. TESLA protokolü bu işlem için bir özet fonksiyonu kullanır. [59] TESLA protokolündeki tek yönlü özet zincirinin her bir elemanı paketlerin MAC değerini hesaplarken kullanılacak *k* anahtarlarıdır.

l uzunluğunda bir zincir oluşturmak için önce rastgele şekilde zincirin son elemanı olacak bir s_1 değeri seçilir. Bu değere art arda özet fonksiyonu F uygulanır. En sonunda s_0 değeri, tüm zincirin taahhüdüdür (commitment). Zincirdeki herhangi bir eleman s_0 üzerinden doğrulanabilir. $F^i(s_i)=s_0$ eşitliğinin kontrolü yapılır. Zincirin elementleri oluşturuldukları sıranın tersi yönde ifşa edilir: $S_0, s_1, \ldots. S_{l-1}, s_l$

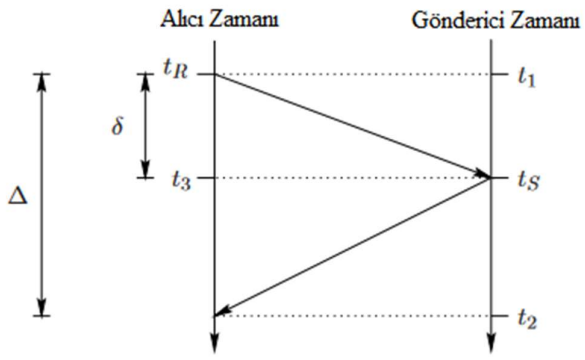


Şekil 3.5 Tek Yönlü Zincirler

3.5.5 TESLA zaman senkronizasyonu

TESLA protokolü, MAC anahtarlarının güvenli şekilde ifşa edilebilmesi için alıcı ile verici arasında bir zaman senkronizasyonuna ihtiyaç duyar. Öte yandan bu zaman senkronizasyonunun çok hassas olması gerekmez. Alıcının bilmesi gereken tek şey, vericinin yerel zamanı için bir üst sınırdır. Diğer bir deyişle alıcı saatinin, verici saatinden en fazla ne kadar geride olabileceğinin bilinmesi gerekmektedir. Bu üst sınır aynı zamanda "maksimum zaman senkronizasyon hatası" olarak bilinir.

TESLA protokolü önerisinde bir zaman eşitleme çözümü sunulmuştur. [59] Fakat bu çözüm vericiye bir zaman senkronizasyonu isteği gönderilmesini içerdiğinden GNSS koşullarına uygun değildir. GNSS sadece yayın yapar, alıcılardan sinyal alamaz. Bu nedenle Anderson ve arkadaşları tarafından 2022'de GNSS ortamı için uyarlanmış bir çözüm daha önerilmiştir. [60] Bu çözüm için ise bağımsız ve zaman ölçümüne güvenilir bir sunucu varsayılmaktadır. Bu tip bir sunucu sertifika otoritelerinin sahip olduğu zaman damgası sunucularıyla aynı sunucu olabilir.



Şekil 3.6 TESLA İki Taraflı Zaman Senkronizasyonu

3.5.6 GALILEO OS

GALILEO Open Service (OS, Açık Hizmet), ücretsiz olarak sunulan bir küresel menzilleme, konumlama ve zamanlama hizmetidir. Her Galileo uydusu 3 frekans bandında navigasyon sinyalleri (SIS) yayınlar. Galileo OS sinyali bu 3 bandın 2'sinde

yayınlanır ve uyumlu alıcısı olan taraflara menzilleme, konumlama ve zamanlama hizmeti sunar.

3.5.7 GALILEO OSNMA

GNSS kimlik doğrulaması, yayın sinyallerine kötü niyetli aktörler tarafından tahmin veya taklit edilemeyen kriptografik özelliklerin dahil edilmesiyle sağlanır. Kimlik doğrulama için uyumlu olan bir alıcı, gerçek sinyalleri taklitlerden ayırt etmek için bu özellikleri yorumlayabilir. Bu, birbirini tamamlayan iki seviyede yapılabilir: yayın navigasyon mesajlarının kimliğini doğrulamak için veri seviyesinde ve uydulara olan uzaklık ölçümünü doğrulamak için menzilleme seviyesinde. Hem veri hem de menzilleme kimlik doğrulamasının birleşimi sayesinde, kimliği doğrulanmış bir konum, hız, zaman (PVT) çözümüne ulaşılır.

OSNMA, dünya çapındaki Galileo OS kullanıcıları için, herkes tarafından ücretsiz olarak erişilebilen ve OS sinyali kullanıcıları ve performansı üzerinde hiçbir etkisi olmayan bir navigasyon mesajı seviyesinde kimlik doğrulama işlevidir. [61]

OSNMA hizmeti, alınan Galileo navigasyon mesajının uydudan iletim sırasında değiştirilmediğine dair güvence vererek, yanıltma (spoofing) saldırılarını tespit etme olasılığını artırır.

OSNMA, kimlik doğrulama desteğini E1-B sinyal bandında yayınlanan navigasyon mesajı (I/NAV) üzerinden hizmet vererek sağlar. E1-B I/NAV mesaj standardında önceden ayrılmış alanlarda doğrulamaya özel veriler iletilir.[62]

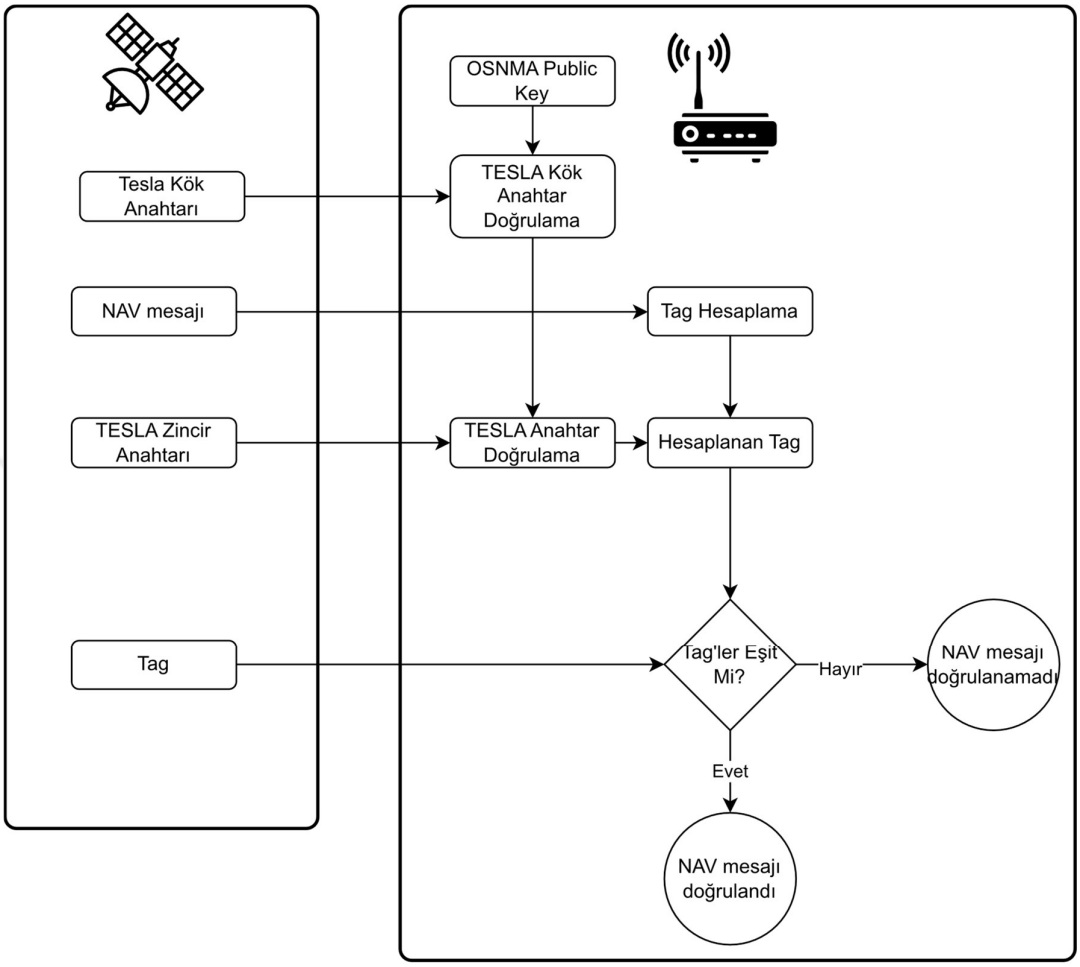
OSNMA, Galileo aracılığıyla en verimli iletim için TESLA protokolünü uyarlar. TESLA protokolünde kullanılan tek yönlü zincir GALILEO uyduları tarafından paylaşılır. Ayrıca bu doğrulama verisi hizmeti, "çapraz kimlik doğrulama" (cross-authentication) olarak adlandırılan, OSNMA verilerini iletmeyen uyduları OSNMA ileten uydulardan alınan verilerle doğrulama özelliğini de içerir. Yani GPS uydularından gelen navigasyon mesajları da GALILEO üzerinden OSNMA ile doğrulanabilir. OSNMA protokolü ile sadece navigasyon mesajları doğrulanır. PRN doğrulaması için ek bir hizmet olan CAS (Commercial Authentication Service) desteği planlanmaktadır. Bu kısım incelenmeyecektir.

OSNMA, hesaplama ve iletişim ek yükünü azaltırken hizmet kullanılabilirliğini ve veri kaybına karşı sağlamlığı artırır. I/NAV mesajının önceden ayrılmış alanlarını kullanan OSNMA, sisteme herhangi bir katman eklemediği için, cihazların navigasyon mesajları işleme performansı etkilenmez. Ayrıca, bu tasarım tam geriye dönük uyumluluk sağlar ve standart alıcılar, OSNMA'ya özel I/NAV alanlarını göz ardı edebilir. [62] Yalnızca OSNMA'ya uyumlu alıcılar bu alanların kodunu çözecek ve GALILEO navigasyon verilerini doğrulayabilecektir. [61]

Bir alıcı perspektifinden, OSNMA verilerinin işlemi aşağıdaki adımlarla basitçe açıklanabilir:

a) Alıcı, navigasyon verilerini ve karşılık gelen OSNMA verilerini alır:
   i) Etiket (mac tag), TESLA zincir anahtarı ve TESLA kök anahtarı.
b) Tag, kendisinden önce iletilen ve ilişkili TESLA zincir anahtarından önce alınan navigasyon verilerinin kimliğini doğrular.
c) TESLA kök anahtarı, alıcıda mevcut olacak bir açık anahtar kullanılarak Galileo'nun dijital imzası aracılığıyla doğrulanır. (Bkz: 4.7.2 Ilık Başlangıç)
d) Alıcı, TESLA zincir anahtarının kimliğini TESLA kök anahtarıyla veya TESLA zincirine ait önceden doğrulanmış bir anahtarla doğrular.
e) Alıcı, doğrulanmış TESLA zincir anahtarı ve verilerle tag'i yerel olarak yeniden oluşturur ve alınan tag ile aynı olup olmadığını kontrol eder.

Tüm bu adımların sonucu başarılı olursa, kullanıcı navigasyon verilerini gerçek olarak kabul edecektir.



Şekil 3.7 GALILEO OSNMA İşlem Süreci

Bu adımları daha iyi anlamak için OSNMA mesaj içeriğini ve TESLA zincirinin bu mesaj içeriğinde nasıl yer aldığını daha detaylı incelemek gerekmektedir.

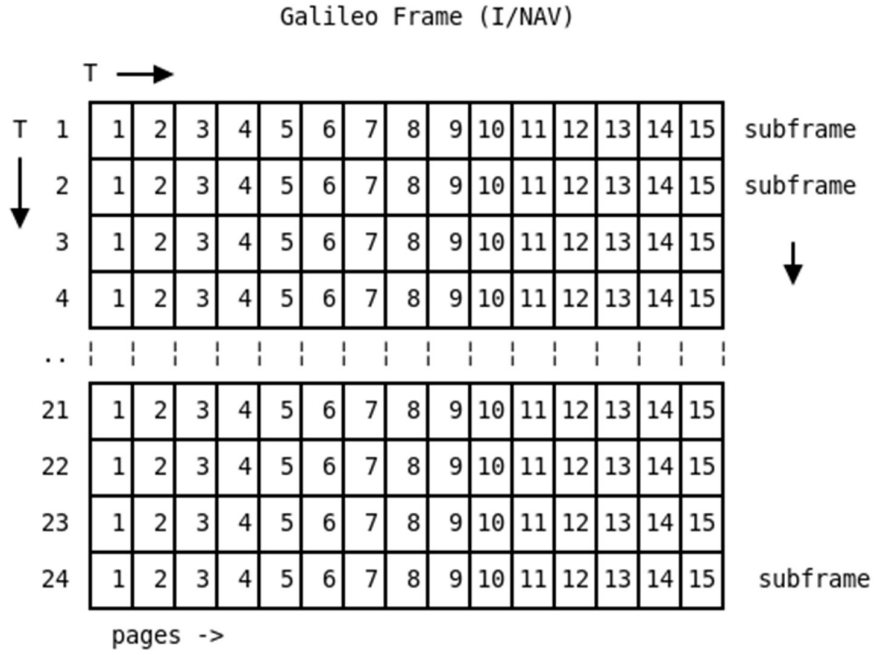
Özellik	OSNMA
GNSS Alıcı Minimal Yeterlilik	Tek frekans E1
Doğrulam Objesi	NAV verisi (E1B I/NAV ve E5b I/NAV)
Gerekli Bileşenler	E1B

Doğrulama	Saat ve Ephemeris Verisi (CED) ve zamanlama parametreleri
İlk Doğrulama İçin Geçen Süre	En fazla birkaç dakika
Doğrulama Erişilebilirliği	Yüksek, %95 üzeri
Kurcalamaya Karşı Dayanıklılık	Hafif, alıcı sadece public keyi saklıyor.
Diğer Gereklilikler	Zaman senkronizasyonu
Alıcının Uydu Sinyalini Kaydetme Gerekliliği	OSNMA protokolü için yok fakat bu tez çalışmasındaki imzalı alan için opsiyonel olarak eklenmektedir.

Tablo 3.1 GALILEO OSNMA Alıcı Gereklilikleri

3.5.8 GALILEO OSNMA mesajı yapısı ve TESLA zinciri entegrasyonu

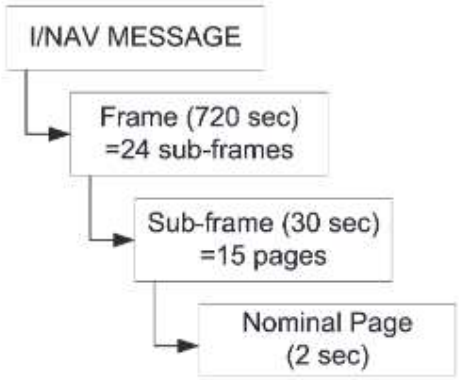
Navigasyon mesajı verileri, bir çerçeve (frame) dizisi olarak iletilir. Bir frame birkaç alt çerçeveden (sub-frame) oluşur ve bir sub-frame de birkaç sayfadan (page) oluşur. Page, navigasyon mesajını oluşturmak için temel yapıdır. [33, 63]



Şekil 3.8 Galileo Page-Subframe İlişkisi

Navigasyon mesajlarında bu frame ve page yapısını belirtecek herhangi bir düzenleme bilgisi yer almaz. Alıcılar verilerin bu yapıda aktığını varsaymak zorundadırlar. [33]

Mutlak Galileo Sistem Saatini (GST) belirlemek için yayın yapan uydu tarafından düzenli aralıklarla navigasyon mesajına zaman verisi eklenir.



Şekil 3.9 Page ve Frame zamanları

GST, aşağıdaki gibi iki parametreden oluşur:

- **Hafta Numarası (WN)**, GST başlangıç döneminden itibaren sıralı hafta sayısını veren bir tamsayı sayacıdır. Bu parametre 4096 haftayı (yaklaşık 78 yıl) kapsayan 12 bit ile temsil edilir. Daha sonra sayaç, modulo 4096 ek periyodunu kapsayacak şekilde sıfırlanır.
- **Haftanın Zamanı (TOW)**, önceki haftadan bitişinden bu yana geçen saniye sayısı olarak tanımlanır. TOW, 0 ila 604799 saniye arasında tüm bir haftayı kapsar ve her haftanın sonunda sıfırlanır.

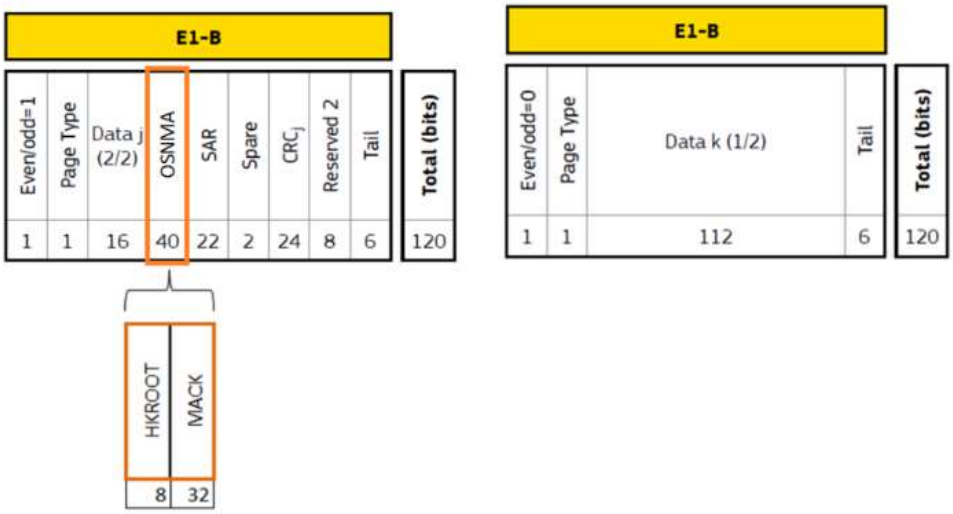
OSNMA iki ana veri türü taşır. HKROOT, GALILEO'nun gizli anahtarı ile imzalanan kısımdır ve güvenilir bir TESLA anahtarı (tek yönlü anahtar zinciri taahhüdü) içerir. Ayrıca, verilerin geri kalanını yorumlamak için ihtiyacımız olan tüm parametrelere sahiptir. HKROOT, "Başlıklar ve Kök Anahtar (Headers and Root Key)" anlamına gelir.

İkinci kısım MACK olarak adlandırılır ve MAC etiketleri içerir, ancak aynı zamanda yeni ifşa edilmemiş anahtarlar ve ayrıca MAC etiketlerinin hangi veriden üretildiğine dair açıklama da taşır (ADKD).

Bir I/NAV mesajındaki (page) HKROOT ve MACK bitlerinden oluşan bu 40 bitlik özel kısım, subframe ve frame boyunca devam eder ve en sonunda alıcı bu page'lerdeki bitleri birleştirerek hedeflenen doğrulama verisine ulaşır. OSNMA alanı, her 30 saniyelik E1-B I/NAV subframe'inde 15 kez iletilir. Diğer bir deyişle OSNMA mesajı 2 saniyede 1 gönderilmektedir. [33]

HKROOT bitleri birleşerek DSM (Digital Signature Message - Dijital İmza Mesajı) oluşturur. Bu dijital imza ECDSA ile oluşturulmuştur ve TESLA protokolünde kullanılan tek yönlü zincir anahtarlarının kök anahtarını doğrulamak için kritiktir. [33]

MACK bitleri ise birleşerek navigasyon mesajının MAC etiketini ve gelecekte kullanılacak TESLA anahtarlarını oluşturur.

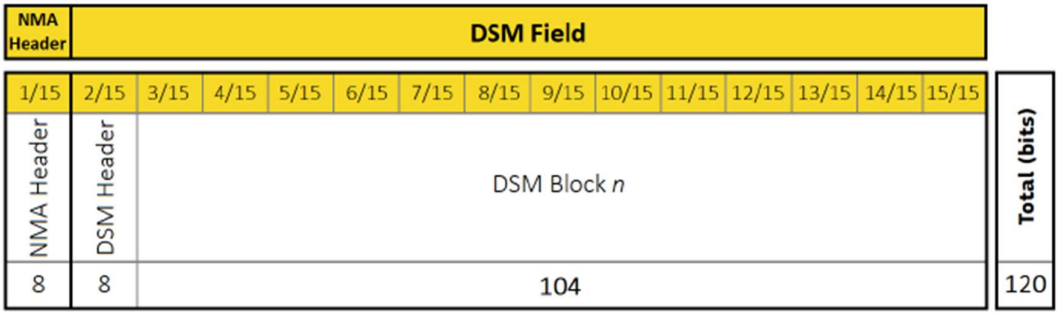


Şekil 3.10 E1-B I/NAV Page İçeriğindeki OSNMA Mesajı Bölümleri

3.5.8.1 HKROOT

Her bir E1-B I/NAV page'i içeriğindeki OSNMA için ayrılmış alanda HKROOT ve MACK olmak üzere iki farklı veri dizisinin parçaları iletilir. OSNMA alanında HKROOT için ayrılmış uzunluk 8 bittir. Subframe tamamlandığında (yani 15 adet I/NAV page yayınlandığında) HKROOT bilgisinin 120 bit olan tamamı iletilmiş olur.

Tamamlanmış bir HKROOT mesajının yapısı aşağıdaki gibidir. 8 bitlik bir NMA başlık alanı ve DSM başlık alanını takip eden DSM bloğundan oluşan 112 bitlik bir Dijital İmza Mesajı (DSM) alanı gelir.

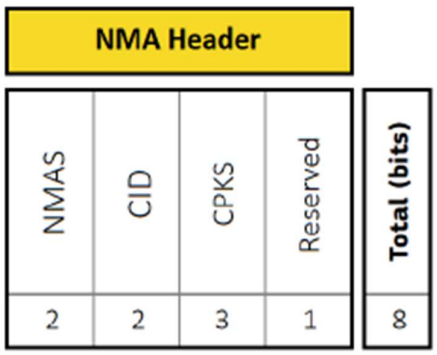


Şekil 3.11 HKROOT Mesajı Yapısı

Farklı uydular, yayınlanan bir subframe'de aynı DSM'in farklı bloklarını iletiyor olabilirler.[33] Bu yüzden eğer zamanlama doğru ayarlanabilirse 30 saniyeden daha önce de HKROOT içindeki DSM'e ulaşılabilir.

3.5.8.2 NMA başlık alanı (NMA header)

Bu alan ile NMA (Navigation Message Authentication) hizmetinin durumu tanımlanır.



Şekil 3.12 NMA Başlık Alanı

NMAS alanı 2 bittir, OSNMA'nın operasyonel durumunu gösterir. NMAS, 0 ila 3 arasındaki değerleri alabilir ve aşağıdaki tabloda her bir değer için karşılık gelen anlam belirtilmiştir.

NMAS	Kategori	Açıklama
0	Ayrılmış	
1	Test	OSNMA hizmeti sunulmaktadır fakat herhangi bir operasyonel garanti verilmemektedir.
2	Operasyonel	OSNMA hizmeti yayınlanan doküman ve şartnamelere uygun olarak verilmektedir.
3	Kullanmayınız	Navigasyon mesajları sağlanan OSNMA verisiyle doğrulanmamalıdır

Tablo 3.2 NMAS alanı açıklamaları

3.5.8.3 Zincir ID (Chain ID, CID)

2 bitlik CID alanı, yürürlükteki anahtar zincirinin kimliğini temsil eder. Değeri yeni zincir yürürlüğe girdiğinde artar. Desteklenen değerler 0 ile 3 arasındadır ve bu değerlerden sonra CID tekrar sıfırlanır.

3.5.8.4 Zincir ve açık anahtar durumu (CPKS)

3 bitlik CPKS alanı, o sırada kullanımda olan TESLA anahtar zincirinin ve açık anahtarın durumunu belirtir.

Bu değerler aşağıdaki durumları açıklamakta kullanılır:

- Mevcut zincirin sonlanmakta olup olmadığı
- Açık anahtar ve merkle ağacının yenilenmekte olup olmadığı

3.5.8.5 DSM (Digital Signature Message)

Her HKROOT mesajında bir DSM Başlığı (8 bit) ve bir DSM bloğu (104 bit) iletilir. DSM blok dizileri bir DSM oluşturur. DSM'in uzunluğu DSM türüne ve kullanılan şifreleme parametrelerine bağlıdır.

İki farklı DSM türü vardır:

- **DSM-PKR**, TESLA zincirinin kök anahtarının doğrulanması için açık anahtar sağlar. (Aynı zamanda OSNMA Web sunucusundan da indirilebilir. Bu çalışmamızda bu alanın SIS ile alınıp doğrulanması incelenmeyecektir.)
- **DSM-KROOT**, bir TESLA zinciri için dijital olarak imzalanmış bir KROOT sağlar.

3.5.8.6 DSM-KROOT

DSM-KROOT, mevcut zincirin veya bir sonraki zincirin TESLA kök anahtarını sağlar. TESLA kök anahtarı ve zincirle ilgili diğer sabit parametrelerin, dijital imza mesajında olmaları sayesinde güvenilirlikleri sağlanmış olur. Bu parametreler aşağıdaki gibidir[33]:

- DSM-KROOT blok sayısı
- Açık Anahtar ID'si (PKID)
- KROOT Zincir ID'si (CIDKR)
- Özet Fonksiyonu (HF)
- MAC Fonksiyonu (MF)
- Anahtar Uzunluğu (KS)
- Tag Uzunluğu (TS)
- MAC Tablosu (MACLT)
- KROOT Hafta Numarası ve Hafta Zamanı (WN*k* ve TOWH*k*)
- Rastgele α örüntüsü (özet alma işleminde kullanılır)
- KROOT
- Dijital İmza (DS)
- DSM-KROOT padding (doldurma) bitleri

3.5.8.7 KROOT

KROOT, DSM-KROOT'ta imzalanan ve zincir bilgisiyle birlikte WN*k* ve TOWH*k* ile ilişkili olan TESLA zinciri kök anahtardır. KROOT'un uzunluğu sabittir.

- WN*k*: KROOT Hafta Numarası (12-bit)
- TOWH*k*: Saat cinsinden haftada geçen zaman. (8-bit)

3.5.8.8 DS

Bu alan DSM-KROOT alanıyla oluşturulmuş ECDSA dijital imzayı içerir.

Dijital imzanın doğrulaması, PKID tarafından tanımlanan açık anahtarla ilişkili dijital imza fonksiyonu kullanılarak gerçekleştirilir.

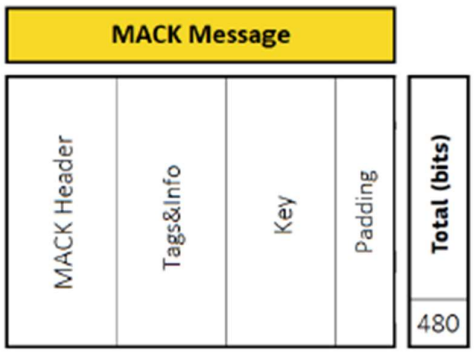
3.5.8.9 MACK

MACK mesajının tamamı 480 bit uzunluğundadır ve her E1-B I/NAV subframe'inde bir tane iletilir. Her 40-bitlik OSNMA mesajının 32 biti MACK için ayrılmıştır.

MACK mesajı içeriği:

- MACK Başlığı
- Kısaltılmış MAC verisi. (tag)
- Tag ile ilişkili veriler (tag-info)
- TESLA Anahtarı

Tag'ler, Tag-Info alanındaki bilgileri dikkate alarak oluşuturulan MAC'lerdir ve ardından bu MAC'ler, zincirdeki DSM-KROOT'un Tag Uzunluğu (TS) alanında tanımlanan uzunluğa kadar (MSB'den başlayarak) kısaltılır. MACK mesajının yapısı aşağıdaki şekilde gösterilmiştir.



Şekil 3.13 MACK Mesajı

3.5.8.10 Tag-Info

Tag-Info bölümü, bir tag oluşturmak ve ilgili doğrulanmış verileri tanımlamak için gerekli bilgileri içeren 16 bitlik bir bölümdür.

Tag-Info alanındaki önemli alanlar:

- **PRNd**: Navigasyon mesajını yayınlayan uyduyu tanımlar. OSNMA protokolü GALILEO dışındaki uyduların sinyallerini de doğrulamayı destekleyecek şekilde tasarlanmıştır. Bu yüzden bu alandaki veriye göre bu fark anlaşılabilir.
- **PRNa**: 8-bitlik işaretli tam sayı, kimlik doğrulama bilgisini ileten uyduyu tanımlar ve her zaman bilgiyi ileten Galileo uydu SVID'inin değerini alır.

3.5.8.11 ADKD

Doğrulama Verisi ve Anahtar Gecikmesi (Authentication Data and Key Delay) alanı 4 bittir, ilgili etiketi (tag) oluşturmak ve karşılık gelen doğrulanmış veriyi tanımlamak için gereken doğrulanmış navigasyon verilerini açıklar. ADKD, 0 ila 15 arasındaki değerleri alabilir ve her bir değer için Galileo PRN'leri için aşağıdaki tabloda gösterilmiştir.

ADKD	Doğrulanan Veri	PRN_D	Zaman Aralığı
0	**I/NAV Efemeris Verisi, Saat ve Durum.** Tag bir önceki subframe'de iletilen I/NAV verisini doğrular.	Doğrulanacak veriyi gönderen uydu no: SV_{ID}	Subframe
1-3	Ayrılmış		
4	**I/NAV Zamanlama Parametreleri.** GST-UTC çevirme parametreleri ve GST-GPS çevirme parametreleri. Bir veya iki subframe önceki I/NAV verisi doğrulanır.	ADKD = 0 ile aynı	Subframe
5-11	Ayrılmış		
12	**Slow MAC, I/NAV Efemeris, Saat ve Durum.** ADKD=0 ile farkı anahtarların 10 subframe gecikme ile paylaşılması.	ADKD=0 ile aynı	ADKD=0 ile aynı
13-15	Ayrılmış		

Tablo 3.3 *ADKD tablosu*

4. NAVİGASYON MESAJININ E-İMZAYA EKLENMESİ

Uydudan alınan navigasyon mesajının, oluşturulacak e-imza paketine doğrulanabilir halde eklenmesi için, GNSS ve e-imza bölümlerinde açıklanan sistemlerin uyumlu şekilde kullanılması gerekmektedir.

4.1 Gerekli Cihazlar ve Yazılımlar

Önerdiğimiz model kapsamında donanım ve yazılım ihtiyaçları şu şekildedir.

- E-imza için güvenilir bir ESHS'den alınmış e-imza sertifikası.
- GALILEO uydu sinyallerini işleyebilen bir alıcı.
  - ❖ ublox çiplere sahip bağımsız bir usb modulü veya imzalamayı yapan sisteme entegre bir SoC olabilir.
- Güvenilir bir ESHS tarafından uyumluluk testlerinden geçmiş konum damgası uyumlu yazılım.
  - ❖ Önerdiğimiz bu modelin implementasyonunu yapmış olan programın, GNSS sinyallerini hesaplayan kısmının kaynak kodunun güvenilir olması varsayılmaktadır. Mevcut durumda, sahada e-imza atılmasını sağlayan yazılımlar, uyum belgesi almak zorundadırlar.
  - ❖ Önerilen modelde gereken yazılımsal yükü karşılayabilecek donanıma sahip bilgisayar/devre. (PC veya raspberry pi, arduino gibi modüller.)

Önerilen temel modelde, kötü niyetli bir imzalama programıyla, alınan sinyalin alınma zamanını bilerek değiştirerek menzil hesaplamalarının yanlış oluşmasına neden olan bir imzalayıcı engellenmemiştir. Konum damgası önerisi haricinde, bu olasılık standart e-imzalarda da bulunmaktadır. Kötü niyetli bir program kullanıcının haberi olmadan başka verileri imzalatabilir. Bunun önüne geçilebilmesi için önerilen yöntemlere ise 4.10 Cihaz ve Uygulama Güvenliği başlığında değinilmiştir.

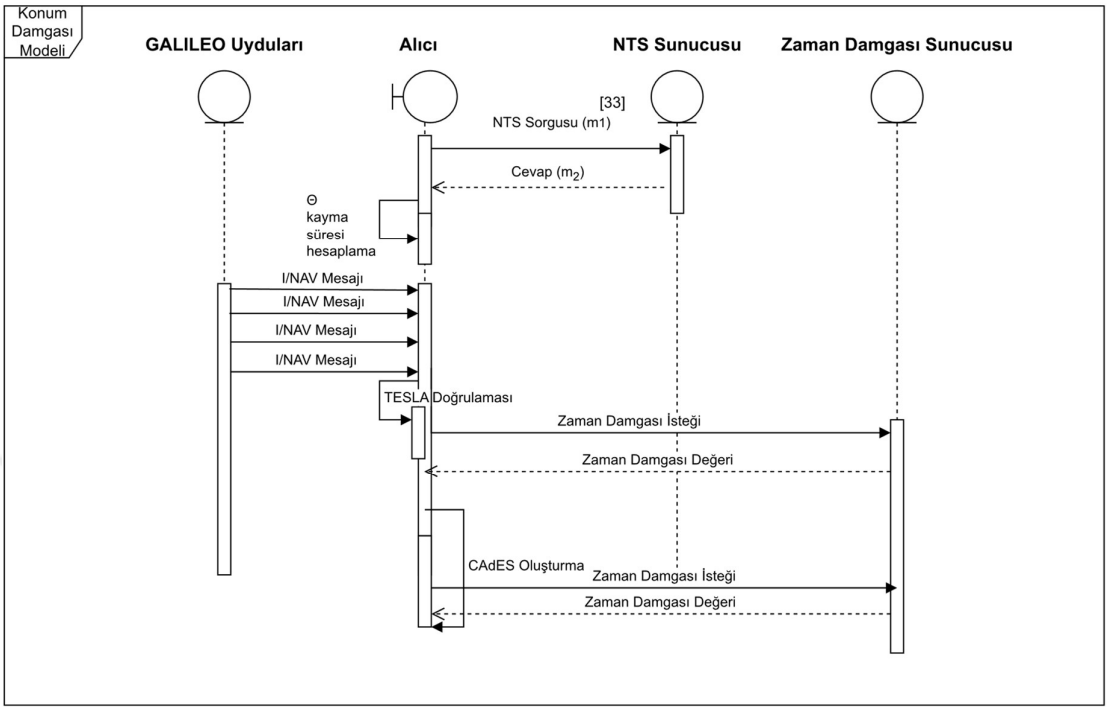
4.2 Önerilen Yöntem

Önerilen süreç akışında kullanılan yazılım, GNSS uydu alıcısıyla entegre çalışan bir e-imza uygulamasıdır. Uygulamanın GNSS sinyallerine donanıma yakın seviyede tam erişimi olduğu varsayılmıştır. Yapılan işlem kronolojik olarak iki ana parçada işlenebilir.

İlk adım mesaj doğrulama altyapısı için gerekli başlangıç parametrelerinin kurulması ve uygulama tarafından teyit edilmesi. GALILEO OSNMA protokolünün sağlıklı çalıştığının kontrolünden sonra başlayacak olan ikinci adım ise işlem sırasında gerekli aralıklarla zaman damgası temin edilen bir dijital imza işlemidir. Sürecin akış diyagramı EK-1'de belirtilmiştir.

Sürecin adımları aşağıdaki gibi detaylandırılabilir:

1. GALILEO ile zaman senkronizasyonu kurulur. [60] (zaman senkronizasyonu çok düşükse slow mac seçilebilir, 4. adımdaki zaman damgası zamanını etkiler.)
2. GALILEO'nun bize sağladığı OSNMA hizmeti kapsamında uyduların gönderdiği doğrulanabilir verilerin alıcı tarafından alındığı haliyle (SIS) kaydedilir.
3. Koordinat hesaplama sırasında gereken parametreler kaydedilir. (Alıcı yerel zamanı ve PRN shift. Bu işlem, imzlayan program tarafından yapılır.)
4. Koordinat hesaplanana kadar belirli aralıklarla (t < 15 sn.) 2 ve 3. adımlardaki veriler birleştirilerek oluşan veri için zaman damgaları alınarak kaydedilir.
5. 2. adımdaki navigasyon mesajı, yerel parametreler ve zaman damgalarından oluşan paketler birleştirilerek tüm işlemi kapsayan bir CAdES-XLONG oluşturulur.
6. Doğrulamak isteyen kişi, çatı e-imza paketinde konum damgası için ayrılmış olan imzalı alanı (signed attribute) işleyerek, 4. adımdaki navigasyon mesajıyla ilişkili OSNMA verisini ve TESLA zincirindeki anahtarların alınan zaman damgasından önce kaydedildiğini doğrular.



Şekil 4.1 16 Konum Damgası Modeli

4.3 Slow MAC

Alıcının zaman senkronizasyonunu tam olarak sağlamakta zorlandığı durumlar için tag'lerin anahtarlarının daha yavaş yayınlandığı alternatif bir süreç daha sunulmuştur. ADKD değerinin 12 olduğu özel durumda, sistem tarafından iletilen tag, ek olarak 10 subframe gecikmesi ile yayınlanan bir anahtarla oluşturulur. Anahtarları bu şekilde paylaşılan sürece "Slow MAC" (Yavaş MAC) denir. Bu zincirdeki tag'ler için kullanılacak anahtarın seçilmesinde fazladan 10 subframe gecikmesi dikkate alınmalıdır.[33] Bu kasıtlı geciktirme sayesinde performansı veya zaman sunucularına olan bağlantıları ideal olmayan alıcılar için bir çözüm amaçlanmıştır.

4.4 Zaman Senkronizasyonu Koşulu

Alıcının OSNMA verisini işleyebilmesi için bir zaman senkronizasyon limit değeri belirlenmiştir. Bu senkronizasyon limit süresi 30 saniyedir.[64] Alıcının OSNMA verisini işleyebilmesi için ise alıcı zamanının (4.1) eşitliğini sağlaması gerekmektedir. Yoksa OSNMA verisini işleyemez.

$$\left|\mathrm{GST}^{\mathrm{SIS}} - \mathrm{GST}^{\mathrm{Rx}}\right| < \frac{\mathrm{T_L}}{2} \tag{4.1}$$

$\mathrm{GST}^{\mathrm{SIS}}$: I/NAV mesajından alınan sistem zamanı.

$\mathrm{GST}^{\mathrm{RX}}$: Alıcı zamanı

Bu durumda eğer alıcı zamanı, uydu sinyalindeki sistem zamanından 15 saniyeden fazla gecikmişse, OSNMA doğrulaması yapılamaz.

SLOW MAC tipinde sinyal işleme yapılıyorsa, (4.2) koşulunun sağlanması yeterlidir:

$$\left|\mathrm{GST}^{\mathrm{SIS}} - \mathrm{GST}^{\mathrm{Rx}}\right| < \frac{T_L + 300}{2} \tag{4.2}$$

TESLA protokol doğrulamaları için gerekli olacak subframe başlangıç zamanı ise (4.3) denklemiyle hesaplanır:

$$\mathrm{GST_{SF}} = \mathrm{GST_0} + 30 \cdot \left[\frac{\mathrm{GST}^{\mathrm{SIS}} - \mathrm{GST_0}}{30}\right] \tag{4.3}$$

$\mathrm{GST_0}$: TESLA zincir başlangıç zamanıdır.

Alıcının OSNMA protokolünü kullanabilmesi için alıcı üzerindeki saatin önceki bölümdeki bahsedilen senkronizasyon limitinden daha fazla gecikmemesi kritiktir. Dolayısıyla işlemlere başlamadan önce alıcı üzerindeki saatin doğru ayarlanması odaklanılması gereken ayrı bir noktadır.

Alıcı saatinin doğru olmadığı bir durumda standartlara göre işletilen bir OSNMA başlangıç sürecinde alıcı-uydu zaman farkı GST alanındaki zamanla karşılaştırılarak anlaşılmaktadır. Fakat OSNMA doğrulama süreci başlamadan önce GNSS sinyal paketlerine güvenilemeyeceği için burada süreç kitlenmektedir.

Zaman senkronizasyonu sırasında olası temel tehdit, alıcı saatinin sağlayıcı saatinden belirli bir eşik değerinden (Θ) daha fazla geride olduğu durumlarda TESLA güvenliğinin bozulması ve sahte mesajların oluşturulabilmesidir. Kötü niyetli kişiler zamanlama hesaplamaları sırasında gerçek GNSS sinyallerini bozarak alıcının yanlış zaman hesaplamasına neden olabilirler. Bunu engellemek amacıyla alıcı tarafında ani zaman değişikliklerini tespit edebilen önlemler alınması mümkündür. Öte yandan daha ileri

seviye saldırılarda, saldırganın, zaman farkını tespit edilmesi zor şekilde kısa aralıklarla artırdığı, daha uzun süren bir senaryo başarılı sonuçlanabilir. [60]

Bu nedenle güvenilir bir zaman bilgisi hizmeti veren bağımsız bir otoriteye ihtiyaç duyulmaktadır. Bunun için en yaygın yöntem ağ üzerinde bağlı iki saat arasında senkronizasyon sağlayan NTP'dir [65] İlk olarak 1985'te yayınlanan ve şu anda 4.versiyonu kullanımda olan bu protokolde spoofing, replay ve MitM saldırılarına karşı bazı zafiyetler mevcuttur. Bu zafiyetlerin giderilmesi için NTP'ye eklenen bir güvenlik mekanizması olan NTS önerilmektedir. [66]

NTS şu özellikleri sağlar:

Kimlik Doğrulaması: Zaman senkronizasyon sürecine katılan tarafların birbirlerini karşılıklı olarak doğrulayabileceği şekilde sahte saldırıları önlemek için güvenlik sağlar.

Bütünlük Koruma: İstemci ve sunucu arasında iletilen zaman bilgilerinin iletim sırasında değiştirilmemesini sağlamak için kriptografik mesaj bütünlük kontrolleri sunar.

Replay Saldırılarına Karşı Koruma: Benzersiz tanımlayıcılar ve sıra numaraları kullanarak, NTS, zaman senkronizasyon sürecini bozmak için bir saldırganın daha önce geçerli olan bir mesajı yeniden göndermesini önleyen tekrar saldırılarına karşı koruma sağlar.

Gizlilik (isteğe bağlı): NTP mesajının belirli bölümleri için gizlilik sağlayarak, dinlemeye karşı koruma sunabilir.

Çalışmamızda önerilen konum damgası modelinde ilk adım olarak da Anderson ve arkadaşlarının önerdiği şekilde bir zaman senkronizasyonu adımı olmalıdır. [60] Bu adımda işletilen algoritma bağımsız bir NTS sunucusuna atılan NTS sorgularıyla mümkün olmaktadır.

4.5 OSNMA Signal-in-Space Kaydetme

Signal-in-Space (SIS), uydu sinyallerinin uzaydaki halidir. Alıcı cihaz yorumlamadan önceki formatıdır. Bu verilerin doğrulanabilmesi için, cihaza gelmeden önceki bu halinin kaydedilmesi gerekmektedir. SIS formatı ve içerdiği mesaj yapısı önceki bölümde detaylı

bir şekilde açıklanmıştır. Bu bölümde bu mesajın içeriğinin kriptografik doğrulama için nasıl kullanılacağı üstünde durulacaktır.

E-imza paketine eklenecek olan navigasyon mesajının, kriptografik olarak doğrulamaya yeterli olacak kadar OSNMA verisi içermesi gerekmektedir. Bunun için gerekli olan OSNMA verileri [64]:

- Merkle Kökü
- Açık Anahtar
- TESLA kök anahtarı
- TESLA zincir anahtarları
- Tag verileri ve ilgili navigasyon mesajları

Sürekli olmayan veri alımı durumunda kullanıcı, SIS'te sağlanan protokolün farklı unsurlarını (örneğin etiketler, anahtarlar, DSM bloğunun bir kısmı) OSNMA alanları içinde iletilme şekillerini kullanarak ayrı ayrı alabilir. Örneğin, birden fazla uydu tarafından alınan page'lerin birleştirilmesiyle anahtar alıcı cihaz tarafında oluşturulabilir.

4.6 Merkle Kökü ve Açık Anahtar

Merkle ağaçlarının temel amacı, tüm veri kümesini depolamadan veya iletmeden büyük miktardaki verinin bütünlüğünü verimli bir şekilde doğrulamaktır.

Merkle ağacının kökü, ayrıntılı veri kümesinin elemanlarından oluşturulan tek bir özet değeri olan Merkle köküdür.

Merkle ağacı, *l* seviyeli bir ikili ağaç üzerine kurulmuştur. h1, ağacın kökünü temsil eder. Her h_i düğümü, sol alt düğümü h_{2i} ve sağ alt düğümü h_{2i+1} olan bir ağacın düğümüdür. Merkle ağacı, alt düğümlerin özetleri ile ana düğümün hesaplanması ile tanımlanır:[67]

$$h_i = hash(h_{2i} \mid h_{2i+1}) \tag{4.4}$$

GALILEO navigasyon mesajlarının imzalanmasında kullanılan açık anahtar, GALILEO web sitesinden veya kontrol merkezinden güvenilir yollarla temin edilebileceği gibi, aynı zamanda SIS içindeki DSM-PKR alanından temin edilebilir. Uydu sinyallerinden temin edilen açık anahtarın doğrulanması için alıcının hafızasında kayıtlı olan Merkle ağacı kökünden (Merkle Tree Root) faydalanılır. Merkle ağacı kökünün nadiren yenilenmesi

planlanmaktadır. [64] Alıcı Merkle ağacı kökünü OSNMA web sunucusundan önceden temin etmiş olur.

Bu çalışmamızda açık anahtarı merkle kökünden doğrulamak yerine direkt olarak OSNMA web sunucusundan alacağımız varsayılmıştır. Bu yüzden DSM-PKR alanından açık anahtar doğrulaması incelenmeyecektir.

4.7 OSNMA Doğrulama Akış Başlatma Tipleri

Alıcının OSNMA paketlerini doğrulamaya başlayabilmesi için bazı anahtarlara sahip olması gerekmektedir. Bu anahtarların tipine, hangi anahtarın eksik olduğuna ve temini için gereken hazırlıklara bağlı olarak süreç farklılaşmaktadır ve bu farklı başlangıç yolları aşağıdaki gibi ayrılmıştır:

4.7.1 Soğuk başlangıç (Cold Start)

Eğer GALILEO Açık Anahtarı ve TESLA kök anahtarı alıcı tarafında yok ise, bu soğuk başlangıç ihtiyacını doğurmaktadır. Bu durumda alıcı, DSM-PKR alanı üzerinden Açık Anahtarı temin etmek zorundadır. Açık Anahtar temin edildikten ve doğrulandıktan sonra TESLA kök anahtarı doğrulanabilir hale gelecektir. TESLA kök anahtarı bir kere edinildikten ve doğrulandıktan sonra gelecekteki MACK doğrulamaları için kaydedilir ve yeni TESLA anahtarları bu kök ile doğrulanır.

4.7.2 Ilık başlangıç (Warm Start)

Eğer alıcı önceden doğrulanmış bir Açık Anahtara sahipse, DSM-KROOT içindeki Açık Anahtar ID'si (PKID) ile kendi elindeki Açık Anahtarı karşılaştırır. Eğer ID eşleşiyorsa bu anahtar güvenilir bir Açık Anahtar olarak kullanılabilir demektir. Direkt TESLA kök doğrulamasından başlayabilir.

4.7.3 Sıcak başlangıç (Hot Start)

Eğer alıcı hem doğrulanmış bir Açık Anahtara hem de TESLA kök anahtarı ve zincir parametrelerine sahipse ve eğer alıcının SIS ile aldığı TESLA anahtarları, elindekilerle uyuşuyorsa doğrulamaya direkt başlayabilir.

Çalışmamızda açık anahtarın direkt OSNMA web sunucusundan temin edildiğini varsayarsak, en kötü ihtimalde Warm Start (Ilık Başlangıç) ile başlanacağı tasarlanmıştır.

4.8 Kriptografik Doğrulama Adımları

Alıcı tarafından SIS formatında kaydedilmiş navigasyon mesajlarının, bağımsız bir kişi tarafından doğrulanabilmesi için atılacak adımlar, alıcının anlık olarak doğrulama yapması için gereken işlemlerle aynıdır.

Bu bölümde OSNMA mesajının alanlarının kriptografik olarak doğrulanabilmesi için verinin nasıl işlendiği açıklanacaktır.

4.8.1 Gereken fonksiyonlar

Kullanılacak temel şifreleme işlevleri aşağıda listelenmiştir:

- **hash$H256(m)$,** giriş mesajı m'ye uygulanan SHA-256 özet işlemidir.
- **hashchain(m),** TESLA zinciri için belirtilen özel özet işlevini ifade eder.
- **mac(K,m),** TESLA zinciri için kullanılan anahtardan üretilen MAC oluşturma için kullanılan fonksiyondur. İlgili zincir, DSM-KROOT'un MF alanında belirtilen zincir olup K anahtarını içerir. m, giriş mesajını ifade eder.

Yukarıdaki üç işlev için giriş mesajı m'nin bir tam sayı byte'ı olarak uyması gerektiğine dikkat edilmelidir. Bu durum gerçekleşmiyorsa, sıfır ile doldurulmalıdır (zero-padding).

signature$(x,$PK$)$: Protokol için seçilen özel dijital imza algoritması ECDSA olarak belirlenmiştir. Bu algoritma, farklı imza uzunluklarını ve Açık Anahtar (PK) uzunluklarını destekler. Açık anahtarın kendisi ve fonksiyonla ilgili parametreler DSM-PKR mesajının NPKT (Açık Anahtar Tipi) alanında belirtilmiştir, aynı zamanda OSNMA'nın güvenilir web sunucusundan da temin edilebilir.

$trunc(L,I)$: girdi I'nin en önemli L bitini (MSB) koruyan bir kırpma işlevidir.

$(X||Y)$**:** bit kümesi X'i Y'ye bit düzeyinde ekleme işlemidir. X, MSB'de olacak şekilde eklenir.

4.8.2 DSM-KROOT doğrulama

$$M = (\text{NMA Header} \parallel \text{CIDKR} \parallel \text{Reserved1} \parallel \text{HF} \parallel \text{MF} \parallel \text{KS} \parallel \text{TS} \parallel$$

$$\text{MACLT} \parallel \text{Reserved2} \parallel WN_K \parallel TOWH_K \parallel \alpha \parallel KROOT)$$

İlgili alanlar için 3.5.8 başlığına bakabilirsiniz.

$$\text{DS} = \text{signature}(M \parallel P, \text{PK})$$

P: sıfır dolgu bitleri (zero-padding)

PK: Açık Anahtar

4.8.3 TESLA anahtar doğrulama

TESLA anahtarları, yalnızca OSNMA sağlayıcısı tarafından bilinen rastgele bir tohum (seed) anahtar K_N ile başlayan bir zincire aittir ve DSM-KROOT aracılığıyla güvenilir bir açık anahtara sahip dijital imza ile imzalanan kök anahtar K_0 ile sonlanır. TESLA anahtarları simetriktir.

Seed anahtarı K_N ve kök anahtarı K_0, aşağıdaki fonksiyon F aracılığıyla birbirleriyle ilişkilidir:

$$K_0 = F^N(K_N)$$

Burada, FN, işlevinin N kez tekrarlanması anlamına gelir, böylece zincirin her bir öğesi, önceki öğeye F uygulanarak oluşturulabilir, şu şekilde:

$$K_I = F(K_{I+1}) = trunc(l_k, hashchain(K_{l+1} \parallel GST_{SF_l} \parallel \alpha))$$

Burada:

- I, zincirdeki anahtarın yeridir;
- lk, anahtar boyutudur;
- α, rastgele zincir örüntü değeridir (özet fonksiyonunda tuzlama benzeri bir yapı için kullanılır);
- GST_{SFl}, K_I anahtarının iletildiği 30 saniyelik subframe başlangıcındaki Galileo Sistem Zamanını temsil eder ve 32 bitlik işaretli tamsayı (signed integer) olarak gösterilir. Bu subframe başlangıç zamanının 1 saniye öncesidir.

$$\mathrm{GST}_{SF,I} = \mathrm{GST}_0 - 30 \text{ [seconds]}$$

GST_0: ilgili zincirin uygulanmaya başlama zamanıdır.

Bir TESLA anahtarı K_I, $F^I(K_I)$ hesaplanarak ve sonucun K_0 ile karşılaştırılarak K_0 ile doğrulanabilir. K_I'nin K_0 ile karşılaştırılması için yapılacak özet işlemi sayısı güncel subframe ile zincirin başladığı subframe arasındaki zaman farkının 30'a bölünerek 1 eklenmesiyle elde edilir.

4.8.4 Tag doğrulama

MACK mesajlarında iletilen tag'ler (MAC etiketleri) aşağıdaki şekilde oluşturulur:

$$\mathrm{tag} = \mathrm{trunc}(l_t, \mathrm{mac}(K, m))$$

l_t : Tag uzunluğu

K: Tesla anahtarı

m: Girdi mesajı

m şöyle hesaplanır:

$$m = (\mathrm{PRN_D} \| \mathrm{PRN_A} \| \mathrm{GST}_{SF} \parallel \mathrm{CTR} \parallel \mathrm{NMAS} \parallel \text{ navdata } \parallel \mathrm{P})$$

GST_{SF}: tag'in iletildiği 30 saniyelik subframe'in başlangıcını belirten bir değerdir.

CTR: MACK mesajındaki tag'in konumunu belirleyen 8 bitlik işaretsiz tamsayıdır; tag dizisindeki ilk etiket için '1' değerini alır ve mesajın her bir sonraki tag'i için birer birer artar. **m** mesajı her tag için benzersizdir.

4.9 Konum Damgasının Oluşturulması

4.9.1 Doğrulanan OSNMA mesajları için zaman damgası alınması

Yukarıdaki adımlarla doğrulanan OSNMA navigasyon mesajları için, TESLA anahtar zincirindeki anahtarların ifşa edilmeden önce alıcıya geldiğinin kanıtlanması amacıyla güvenilir bir ESHS'den zaman damgası alınmak zorundadır.

Zaman damgası isteği için oluşturulacak özet değeri, E1-B I/NAV sinyali üzerinden gönderilen 40 bitlik OSNMA paketlerinin birleştirilmesiyle oluşan byte dizisinin SHA256 veya daha modern özet algoritmalarıyla özetinin alınmasıyla oluşturulur.

Zaman damgası alma periyodu, alıcının OSNMA sinyallerini doğrulamaya başlamasından itibaren geçen her bir subframe başlangıç zamanı (GSTsf) öncesi olmalıdır. Bu periyot OSNMA protokolüne göre değişiklik gösterebilir. Eğer koordinasyon hesaplaması gelecek subframe öncesi tamamlandıysa, yeni subframe başlangıcı beklemeden zaman damgası alınabilir.

Eğer zaman damgası alma işleminin ağdaki veya sistemdeki gecikmelerden ötürü 30 saniyeden (güncel protokoldeki subframe süresi) uzun sürdüğü bir durumun gerçekleştiği tespit edilirse, doğrulama SLOW MAC sinyalleri üzerinden yapılacak şekilde değiştirilebilir.

4.9.2 Konum damgası formatı

Mevcut CAdES standardında, tasarladığımız bu yeni model için düzenlenmiş bir attribute tanımı bulunmamaktadır. Konum ile ilgili en yakın anlama sahip alan [5] dokümanı 5.11.2 başlığında gösterilen "*signer-location*" isimli attribute. Fakat bu alan DirectoryString tipinde bir adres alanı tutmakta, "*countryName*", "*localityName*", "*postalAddress*" gibi açık adres bilgilerine ayrılmaktadır.

Bu nedenle CAdES formatında kullanılan ASN.1 standardıyla uyumlu olacak, CAdES uyumlu yazılımların imza dosyalarını işlemesinde sorun çıkarmayacak, aynı zamanda tanımladığımız bu alanı tanıyan güncel uygulamalarda aynı dosyanın konum damgası doğrulaması yapabilmesine izin verecek şekilde yeni bir attribute tanımlamak gerekmektedir. CAdES standardı istediğimiz tipte yeni attribute alanları tanımlamamıza izin vermektedir. Bu attribute alanları ASN.1 formatına uygun olarak tasarlanmalıdır.

İmza paketini işleyen uygulamaların alıcı cihazın hesapladığı koordinat bilgisine kolayca erişmesi için aşağıdaki objeler tanımlanır:

CMS standardını tanımlayan [68]'a göre kendi attribute tanımımızı aşağıdaki ContentInfo üzerine temellendirebiliriz:

```
ContentInfo ::= SEQUENCE {
     contentType ContentType,
     content EXPLICIT ANY DEFINED BY contentType
}
```

```
ContentType ::= OBJECT IDENTIFIER
```

Şekil 4.2 ContentInfo ASN.1 Objesi

contentType, object identifier tipinde bir değerdir. Kendi oluşturacağımız obje tanımı için bu değeri benzersiz olacak şekilde bir sonraki bölümde tanımlayacağız.

content, ilişkili veridir. Bu verinin nasıl işleneceği contentType alanındaki object identifier bilgisine göre belirlenir.

Zaman damgası standardında olduğu gibi biz de bu değerleri şu şekilde kullanacağız:

contentType: `id-signedData OBJECT IDENTIFIER ::= { iso(1) member-body(2) us(840) rsadsi(113549) pkcs(1) pkcs7(7) 2 }`

Koordinat bilgilerini tutmak için tasarlanan sınıflar:

```
LatitudeSign ::= ENUMERATED {
   north,
   south
}
DegreesLatitude ::= INTEGER
DegreesLongitude ::= INTEGER
GeographicalCoordinates ::= SEQUENCE {
   latitudeSign LatitudeSign,
   degreesLatitude DegreesLatitude,
   degreesLongitude DegreesLongitude
}
```

Şekil 4.3 Koordinat Bilgileri ASN.1 Objeleri

LatitudeSign: Enlemin yarımküre bilgisi.

DegreesLatitude: Enlem derecesi.

DegreesLongitude: Boylam derecesi.

Uydu sinyallerini tutmak ve gerekirse doğrulayan kişiye fazladan bir değerlendirme şansı sunmak amacıyla, Signal-in-Space hakkında bilgi:

```
PRNShift ::= SEQUENCE {
    localPRN   EXPLICIT OCTET STRING
    svID       INTEGER
    svPRN      EXPLICIT OCTET STRING
}
RangingLog ::= SEQUENCE OF PRNShift
```

Şekil 4.4 9. Uydu sinyalleri için ASN.1 Objeleri

localPRN, alıcı tarafında oluşturulmuş zamana bağlı PRN sinyal grafiğidir.

svID, sinyali gönderen uydu ID'sidir.

svPRN, uydudan gelen zamana bağlı PRN sinyal grafiğidir.

Koordinat hesaplanmasında kullanılan PRN kodlarında uydudan gelen sinyal ve alıcının sinyali arasındaki senkronizasyon farkının hesaplanması uyduya olan uzaklığın hesaplanmasında temel alınmaktadır. Bu sinyallerin alıcı cihaz tarafından zamana bağlı grafiklerinin bir byte dizisi olarak tutulması anlık olarak yapılan koordinat hesaplama işlemi için bir kaynak sunmaktadır.

Subframe bazında zaman damgasına sahip her bir OSNMA paketinin oluşturulması ve doğrulanması için gerekli tanımlamalar:

Doğrulanan navigasyon mesajları için almak istediğimiz zaman damgası yapısı:

```
TimeStampToken ::= ContentInfo
GNSSAuthenticationData ::= SEQUENCE {
      gnssTime        DATE-TIME
      iod             BIT STRING
      nmaSIS          EXPLICIT OCTET STRING
```

```
        ts                  TimeStampToken
    }
```

Şekil 4.5 GNSS Doğrulama Verisi ASN.1 Objesi

gnssTime, uydu sisteminin sinyallerinde belirtilen zamandır. Galileo için: GSTsf. Subframe başlangıç zamanı.

iod, issue of data, uydu sinyalinin versiyonunu belirtir.

nmaSIS, "navigation message authentication Signal-in-Space", navigasyon mesaj sinyalinin uydudan geldiği halinin zamana bağlı grafiğidir.

ts, nmaSIS byte dizisinin zaman damgası değeridir.

Galileo sisteminde örnek vermek gerekirse, *GNSSAuthenticationData* objesi, bir subframe iletim zamanını geçmeyecek şekilde (en geç 15 saniyede 1) oluşturulur.

Oluşturulan bu değerler, bir dizi olarak *LocationStampToken* alanına eklenir. Bu alan nihai adımda, CAdES formatında imzalı bir özellik (signed attribute) olarak eklenecektir.

```
LocationStampInfo ::= SEQUENCE {
    rngLog              RangingLog OPTIONAL
    coordinates         GeographicalCoordinates
    gnssAuth            SEQUENCE OF GNSSAuthenticationData
}
LocationStampToken ::= ContentInfo
```

Şekil 4.6 Konum Damgası ASN.1 Objesi

ContentInfo tipindeki *LocationStampToken* için tanımlamamız gereken *contentType*, *LocationStampInfo* için tanımlayacağımız bir OID'dir.

4.9.3 Unregistered OID tanımlama (UUID)

Önceki bölünmde tasarladığımız obje tanımlarının PKCS#7 uyumlu işlenmesi için ASN.1 standardında en az bir OID tanımlamamız gerekmektedir. Standartlara ekleme talebi uzun

bir süreç olduğundan, pratik CAdES implementasyon çalışmaları için kayıt dışı (unregistered) OID tanımlanabilir. [69]

```
id-locationStampInfo OBJECT IDENTIFIER ::= {joint-iso-
ccitt(2) uuid(25) id-locationStampInfo(None)}
```

Örnek OID: 2.25.218271439484934529412311957794017043830

4.9.4 Doğrulama

Konum damgasına sahip olacak şekilde CAdES formatında oluşturulan imza paketindeki uydu sinyallerinin bağımsız üçüncü taraflar tarafından "*GNSSAuthenticationData*" üzerindeki *nmaSIS* üzerinden doğrulanma süreci, alıcı cihazın yapması gereken kriptografik adımlarla tamamen aynıdır.

Bu süreçte yapılması gereken ek kontrol, MACK doğrulaması yapılırken kullanılan TESLA anahtarlarının ifşa edilmeden önce alındığının kanıtlanması amacıyla, "*GNSSAuthenticationData*" altında "*ts*" alanındaki zaman damgasındaki zaman değeriyle, anahtarın paylaşıldığı subframe zamanı başlangıcı (GST_{sf}) ile kontrol edilmelidir. Alıcı cihazın OSNMA doğrulaması yapması için zaman limiti 15 saniyedir. Bu nedenle alınan zaman damgasının da bu limit dahilinde olması gerekmektedir.

Diğer bir deyişle "Zaman senkronizasyon koşulu" başlığında detaylandırılan koşul kontrollerindeki GST_{rx} olarak tanımlanan alıcı zamanını, üçüncü taraf doğrulama yaparken zaman damgasındaki zaman değeri ile değiştirebilir.

4.10 Cihaz ve Uygulama Güvenliği

Önerilen temel yöntem, uydu sinyali güvenliğini sağlamasına rağmen, alınan uydu sinyallerinin kullanıldığı koordinat hesaplama işlemlerinin hatasız olacağını kanıtlamamaktadır. Atılan konum damgalı imzayı doğrulayan üçüncü taraf için konum verisinin doğru hesaplandığının da kanıtlanması iki faktöre bağlıdır:

- GNSS koordinat hesaplama modulünün donanımınn kötü amaçla değişitirilmemiş olması.
- İmzayı atan yazılımın, bir otoritenin (genellikle CA) uyum değerlendirme sürecinden geçmiş olması.

Donanım güvenliğinin sahada bağımsız taraflara kanıtlanabilmesi için kolay ve uyarlanabilir yöntemler sunulması bu çalışmanın kapsamında değildir.

Uygulamanın güvenilir olduğunun kanıtlanmasında ise yazılım tabanlı çözümler üretilebilir. Bir örnek çözüm olarak iki taraflı TLS doğrulamayı[70] temel alan bir çözüm öneriyoruz.

Konum damgasını ve içinde bulunduğu imza paketini oluşturan yazılımın, gerçekten uyum değerlendirmeden geçmiş ve güvenilir bir yazılım olduğunu kanıtlamak için, yazılımın kendisi de sertifikalandırılabilir. Yazılım, oluşturduğu imza paketlerini böylece son olarak kendisi imzalayabilir.

Bu çözümde dikkat edilmesi gereken nokta, yazılımın kendine özel bir gizli anahtarı kullanması gerektiğinden, bu gizli anahtarın güvenilir bir şekilde saklanması veya işlem anında temin edilmesi gerekmektedir. Bu amaçla sanal makine koşturma kapasitesine sahip bir cihazda, mevcut işletim sisteminin erişiminin çok kısıtlı olduğu sanal bir güvenli bölge (secure zone) kullanılabilir. Bu güvenli bölge ve işletim sistemi arasındaki iletişim, ve bir uzak sunucuyla iki taraflı TLS doğrulama yapma süreci [71] çalışmasında detaylandırılmıştır.

5. SONUÇ

Bu çalışmada, bir veri parçasının gerçekten beyan eden tarafından, beyan edilen zamanda oluşturulduğunu, onaylandığını veya değiştirildiğini kanıtlayan kriptografik altyapı ve yöntemlere uyumlu olacak şekilde yeni bir *beyan edilen konum* veya *konum damgası* özelliği eklenmesini ve bu konum bilgisinin gerçek uydu sinyalleriyle hesaplandığının üçüncü taraflara kanıtlanabilmesini sağlayan bir model önerisi öne sürülmüştür.

Konum damgası fikri yeni bir fikir olmamasına rağmen, daha önce bu konuda yayınlanan makaleler ve tez çalışmaları incelendiğinde, ilgili çalışmaların yapıldığı tarihteki teknolojik gelişmelerin ve uydu sistemi altyapısının henüz sinyal güvenliği için yeterli olmadığı görülmüştür.

Sinyal güvenliğinin sağlanması ve sağlandığının kanıtlanması amacıyla açık anahtar altyapısı, dijital imza formatları ve güncel küresel konumlama sistemlerinin sunduğu hizmetlerin kullanılması uygun görülmüştür.

Bu amaçla öncelikle dijital imzalar ve açık anahtar altyapısının kriptografik açıdan çalışma prensipleri açıklanmış, ardından sahada kullanılan yöntemlerden ve uluslararası standartlardan bahsedilmiştir. Bu standartlardaki dosya yapıları ve paket çözümleme (parsing) yöntemleri incelenmiştir.

Ardından konum hesaplama yöntemlerinin kısa tarihçesi aktarılmış ve günümüzde kullanılan uydu tabanlı küresel konumlama sistemlerinin çalışma prensipleri detaylandırılmıştır. Temelinde bir radyo yayını (broadcast) olan uydu sinyallerinin, radyo yayınlarıyla aynı zafiyetlerden zarar görme olasılığına sahip olduğu belirtilip, aldatma (spoofing) ve engelleme (jamming) saldırılarına örnekler verilmiştir.

Bu zafiyetlerin giderilmesine güncel bir çözüm olan GALILEO OSNMA protokolünün seçilme nedenleri, sağladığı faydalar ve temelinde kullandığı kriptografik doğrulama algoritması olan TESLA detaylı şekilde açıklanmıştır.

Dijital imzalar ve uydu sistemlerinin altayapısı açıklandıktan sonra iki sistemin bütünleşmiş şekilde nasıl çalışabileceği incelenmiştir. Alıcının aldığı konum sinyallerinin CAdES formatına entegre edilebilmesi için işlem adımları önerilmiş ve bu adımlarda

hesaplanması gereken değerlerle, oluşturulan veri paketi şablonu belirtilmiş, yeni bir imzalı özellik (signed attribute) tanımlanmıştır.

Sonuç olarak güvenli olduğu varsayılan yazılımın oluşturduğu pakette, bağımsız üçüncü tarafların, imza paketinde beyan edilen konum bilgisinin hesaplanmasında kullanılan uydu sinyallerinin gerçek olduğunu doğrulayabileceği bir model sunulmuştur.

Bu model, uydu verisi doğrulamasını üçüncü taraflara kanıtlayabildiği gibi aynı zamanda arşiv imza (CAdES- LTA) yapısıyla da uyumlu çalışabilmeyi desteklediği için uzun dönemli olarak saklanabilecektir.

5.1 Gelecekteki Çalışmalar

Önerilen modelin yazılım implementasyonu yapılmamıştır. GALILEO sisteminde OSNMA protokolü olgunlaştıkça değişiklikler olabilir. Bunları da göz önüne alarak mevcut OSNMA kütüphanelerinden destek alarak bir yazılım demosu yapılması ve sahada kullanılabilecek şekilde performans optimizasyonlarının yapılması gerekmektedir. [72]

Uygulamanın e-imza kanunlarına uygunluğu araştırılmalıdır. Örnek olarak Türkiye'deki e-imza kanunlarına göre imzalanan verinin kişiye gösterilme zorunluluğu vardır.[23] Burada ise kişinin bulunduğu konumun doğrulunun teyit edilebilmesi için kişiye gösterilme yapısı ve tipinin görsel olarak tasarlanması gerekebilir.

Uygulamanın zaman senkronizasyonu ve zaman damgası işlemleri için internet bağımlılığı bulunmaktadır. Sahada kullanılırken bu bağımlılıkları en aza indirmek için gerekli iyileştirmeler tasarlanması gerekmektedir.

KAYNAKLAR

[1] D. Schmidt, K. Radke, S. Camtepe, E. Foo, ve M. Ren, "A Survey and Analysis of the GNSS Spoofing Threat and Countermeasures", *ACM Comput. Surv.*, c. 48, sy 4, ss. 1-31, May. 2016, doi: 10.1145/2897166.

[2] M. Özlü, "E-İmza Güvenliğinin Artırılmasına Yönelik Konum Damgası Sistemi Önerisi ve Uygulaması, Yüksek Lisans Tezi", *TC Selçuk Üniversitesi Fen Bilim. Enstitüsü*, s. 78, Ocak 2011.

[3] E. Ádámkó ve A. Pethő, "Location-stamp for GPS coordinates", *Acta Univ. Sapientiae Inform.*, c. 5, sy 1, ss. 63-76, Tem. 2013, doi: 10.2478/ausi-2014-0005.

[4] M. Kabatnik ve A. Zugenmaier, "Location Stamps for Digital Signatures: A New Service for Mobile Telephone Networks", içinde *Networking — ICN 2001*, içinde Lecture Notes in Computer Science, vol. 2094. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, ss. 20-30. doi: 10.1007/3-540-47734-9_3.

[5] N. Pope, D. Pinkas, ve J. Ross, "CMS Advanced Electronic Signatures (CAdES)", Internet Engineering Task Force, Request for Comments RFC 5126, Mar. 2008. doi: 10.17487/RFC5126.

[6] W. Stallings, "Digital Signature Algorithms", *Cryptologia*, c. 37, sy 4, ss. 311-327, Eki. 2013, doi: 10.1080/01611194.2013.797044.

[7] C. Adams ve S. Lloyd, *Understanding PKI: concepts, standards, and deployment considerations*. Addison-Wesley Professional, 2003.

[8] H.-Y. Chien, "Forgery attacks on digital signature schemes without using one-way hash and message redundancy", *IEEE Commun. Lett.*, c. 10, sy 5, ss. 324-325, 2006, doi: 10.1109/LCOMM.2006.1633310.

[9] I. Damgård, "Collision-free hash functions and public-key signature schemes", *Adv. Cryptology—EUROCRYPT'87*, ss. 203-216, 1989.

[10] P. Rogaway ve T. Shrimpton, "Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance", *Fast Softw. Encryption*, c. 3017, ss. 371-388, 2004.

[11] S. Chokhani, W. Ford, R. Sabett, C. Merrill, ve S. Wu, “RFC 3647: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework”, RFC3647, Kas. 2003. doi: 10.17487/rfc3647.

[12] S. Boeyen, S. Santesson, T. Polk, R. Housley, S. Farrell, ve D. Cooper, “Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile”, Internet Engineering Task Force, Request for Comments RFC 5280, May. 2008. doi: 10.17487/RFC5280.

[13] Kamu Sertifikasyon Merkezi, “NİTELİKLİ ELEKTRONİK SERTİFİKA UYGULAMA ESASLARI, YON.01.01, Revizyon 16”. 20 Ekim 2022. [Çevrimiçi]. Erişim adresi: https://kamusm.bilgem.tubitak.gov.tr/BilgiDeposu/KSM_NES_SUE/KSM_NES_SUE.pdf

[14] J. Weise-Sunps, “Public Key Infrastructure Overview”, 2001.

[15] ETSI, “ETSI TS 101 733: Electronic Signatures and Infrastructures (ESI); CMS Advanced Electronic Signatures (CAdES)”. ETSI, 2013. [Çevrimiçi]. Erişim adresi: https://www.etsi.org/deliver/etsi_ts/101700_101799/101733/02.02.01_60/ts_101733v020201p.pdf

[16] E. T. S. ETSI, “ETSI TS 103 172: Electronic Signatures and Infrastructures (ESI); PAdES digital signatures; Part 2: Additional PAdES profiles”, ETSI, 2012. [Çevrimiçi]. Erişim adresi: https://www.etsi.org/deliver/etsi_ts/103100_103199/103172/01.01.01_60/ts_103172v010101p.pdf

[17] E. T. S. ETSI, “ETSI TS 101 903: Electronic Signatures and Infrastructures (ESI); XML Advanced Electronic Signatures (XAdES)”, ETSI, 2010. [Çevrimiçi]. Erişim adresi: https://www.etsi.org/deliver/etsi_ts/101900_101999/101903/01.04.01_60/ts_101903v010401p.pdf

[18] “Introduction to ASN.1”, *ITU*, 2023. https://www.itu.int:443/en/ITU-T/asn1/Pages/introduction.aspx (erişim 01 Şubat 2023).

[19] "ITU-T Recommendation X.690 (2002): 'Information technology ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)'", 2002.

[20] J. Jonsson ve B. Kaliski, "RFC 8017: PKCS #1: RSA Cryptography Specifications Version 2.2". 2016. [Çevrimiçi]. Erişim adresi: https://tools.ietf.org/html/rfc8017

[21] "OID project", *ITU*, 2023. https://www.itu.int:443/en/ITU-T/asn1/Pages/OID-project.aspx (erişim 01 Şubat 2023).

[22] R. Brandner, U. Pordesch, ve C. Wallace, "RFC 4810: Long-Term Archive Service Requirements". RFC Editor, Mart 2007. doi: 10.17487/RFC4810.

[23] *5070 Sayılı Elektronik İmza Kanunu*. 2004.

[24] R. Zuccherato, P. Cain, C. Adams, ve D. Pinkas, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)", Internet Engineering Task Force, Request for Comments RFC 3161, Ağu. 2001. doi: 10.17487/RFC3161.

[25] M. Koschuch ve R. Wagner, "Trust Revoked — Practical Evaluation of OCSP- and CRL-Checking Implementations", Cham: Springer International Publishing, 2015, ss. 26-33. doi: 10.1007/978-3-319-25915-4_2.

[26] J. Olnes ve A. Seip, "On long-term storage of digitally signed documents", 2003, ss. 115-130.

[27] B. Hofmann-Wellenhof, H. Lichtenegger, ve E. Wasle, *GNSS--global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Wien ; New York: Springer, 2008.

[28] P. J. G. Teunissen ve O. Montenbruck, Ed., *Springer Handbook of Global Navigation Satellite Systems*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-42928-1.

[29] T. A. Stansell, "The Navy Navigation Satellite System: Description and Status", *Navigation*, c. 15, sy 3, ss. 229-243, Eyl. 1968, doi: 10.1002/j.2161-4296.1968.tb01612.x.

[30] P. Daly ve G. E. Perry, "Recent developments with the Soviet Union's VHF satellite navigation system", *Space Commun. Broadcast.*, c. 4, ss. 51-61, Mar. 1986.

[31] E. D. Kaplan ve C. Hegarty, *Understanding GPS/GNSS : Principles and Applications*, c. Third edition. içinde GNSS Technology and Applications Series, vol. Third edition. Boston: Artech House, 2017.

[32] S. Gleason ve D. Gebre-Egziabher, *GNSS applications and methods*. Artech House, 2009.

[33] European GNSS Supervisory Authority., *Galileo Open Service Navigation Message Authentication (OSNMA): Signal in Space Interface Control Document (SIS ICD) : Issue 1.0, December 2022.* LU: Publications Office, 2022. Erişim: 07 Şubat 2023. [Çevrimiçi]. Erişim adresi: https://data.europa.eu/doi/10.2878/594840

[34] J. S. Warner ve R. G. Johnston, "GPS Spoofing Countermeasures", 2003.

[35] Chronos Technology, "Sentinel Project Report on GNSS Vulnerabilities", Şub. 2014, [Çevrimiçi]. Erişim adresi: https://rntfnd.org/wp-content/uploads/SENTINEL-Project-Report.pdf

[36] G. Kar *vd.*, "Detection of On-Road Vehicles Emanating GPS Interference", içinde *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, Scottsdale Arizona USA: ACM, Kas. 2014, ss. 621-632. doi: 10.1145/2660267.2660336.

[37] The Royal Academy of Engineering, "Global Navigation Space Systems: reliance and vulnerabilities. Technical Report.", Eyl. 2013, [Çevrimiçi]. Erişim adresi: https://rntfnd.org/wp-content/uploads/2013/09/Royal-Acad-of-Eng_Global_Navigation_Space-Systems_Report.pdf

[38] O. Isoz, D. Akos, T. Lindgren, C.-C. Sun, ve S.-S. Jan, "Assessment of GPS L1/Galileo E1 Interference Monitoring System for the Airport Environment".

[39] N. A. of Engineering, *Global Navigation Satellite Systems: Report of a Joint Workshop of the National Academy of Engineering and the Chinese Academy of Engineering*. Washington, DC: The National Academies Press, 2012. doi: 10.17226/13292.

[40] D. P. Shepard, T. E. Humphreys, ve A. A. Fansler, "Evaluation of the vulnerability of phasor measurement units to GPS spoofing attacks", *Int. J. Crit. Infrastruct. Prot.*, c. 5, sy 3-4, ss. 146-153, Ara. 2012, doi: 10.1016/j.ijcip.2012.09.003.

[41] H. Borowski, O. Isoz, F. M. Eklof, S. lo, ve D. Akos, "Detecting false signals: With automatic gain control", c. 23, ss. 38-43, Nis. 2012.

[42] A. Jafarnia Jahromi, "GNSS Signal Authenticity Verification in the Presence of Structural Interference", 2013, doi: 10.11575/PRISM/26310.

[43] L. Huang, Z.-C. Lv, ve F.-X. Wang, "Spoofing pattern research on GNSS receivers", c. 33, ss. 884-890, Tem. 2012, doi: 10.3873/j.issn.1000-1328.2012.07.005.

[44] M. G. Kuhn, "An Asymmetric Security Mechanism for Navigation Signals", içinde *Information Hiding*, J. Fridrich, Ed., içinde Lecture Notes in Computer Science, vol. 3200. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, ss. 239-252. doi: 10.1007/978-3-540-30114-1_17.

[45] O. Pozzobon, C. Wullems, ve K. Kubik, "Secure Tracking using Trusted GNSS Receivers and Galileo Authentication Services", *Positioning*, c. 1, Ara. 2004.

[46] K. Wesson, M. Rothlisberger, ve T. Humphreys, "Practical Cryptographic Civil GPS Signal Authentication", *Navigation*, c. 59, sy 3, ss. 177-193, Eyl. 2012, doi: 10.1002/navi.14.

[47] Y. Hu, S. Bian, B. Li, ve L. Zhou, "A Novel Array-Based Spoofing and Jamming Suppression Method for GNSS Receiver", *IEEE Sens. J.*, c. 18, sy 7, ss. 2952-2958, 2018, doi: 10.1109/JSEN.2018.2797309.

[48] "Mass GPS Spoofing Attack in Black Sea? The Maritime Executive.", 11 Temmuz 2017. [Çevrimiçi]. Erişim adresi: https://maritime-executive.com/editorials/mass-gps-spoofing-attack-in-black-sea

[49] "UT Austin Researchers Spoof Supery- acht at Sea. The University of Texas at Austin", *Jul. 29, 2013*. [Çevrimiçi]. Erişim adresi: https://news.utexas.edu/2013/07/29/ut-austin-researchers-successfully-spoof-an-80-million-yacht-at-sea/

[50] M. L. Psiaki ve T. E. Humphreys, "GNSS Spoofing and Detection", *Proc. IEEE*, c. 104, sy 6, ss. 1258-1270, 2016, doi: 10.1109/JPROC.2016.2526658.

[51] J. M. Anderson *vd.*, "Chips-Message Robust Authentication (Chimera) for GPS Civilian Signals", program adı: 30th International Technical Meeting of The Satellite

Division of the Institute of Navigation (ION GNSS+ 2017), Portland, Oregon, Kas. 2017, ss. 2388-2416. doi: 10.33012/2017.15206.

[52] G. Norris, "Lockheed Lays Out GPS IIIF Upgrade Roadmap | Aviation Week Network", *AviationWeek*, 08 Nisan 2022. https://aviationweek.com/shows-events/space-symposium/lockheed-lays-out-gps-iiif-upgrade-roadmap (erişim 19 Nisan 2023).

[53] European GNSS Service Centre, "Galileo Open Service Navigation Message Authentication (OSNMA)", 19 Nisan 2023. https://www.gsc-europa.eu/galileo/services/galileo-open-service-navigation-message-authentication-osnma (erişim 19 Nisan 2023).

[54] B. A. Forouzan, *Data Communications and Networking*, 4th bs. McGraw-Hill Higher Education, 2006.

[55] R. Gennaro ve P. Rohatgi, "How to Sign Digital Streams", *Inf. Comput.*, c. 165, sy 1, ss. 100-116, Şub. 2001, doi: 10.1006/inco.2000.2916.

[56] S. Miner ve J. Staddon, "Graph-based authentication of digital streams", içinde *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, 2001, ss. 232-246. doi: 10.1109/SECPRI.2001.924301.

[57] D. Song, D. Zuckerman, ve J. D. Tygar, "Expander graphs for digital stream authentication and robust overlay networks", içinde *Proceedings 2002 IEEE Symposium on Security and Privacy*, 2002, ss. 258-270. doi: 10.1109/SECPRI.2002.1004376.

[58] H. Krawczyk, M. Bellare, ve R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", *RFC 2104 Internet Eng. Task Force*, 1997, [Çevrimiçi]. Erişim adresi: https://tools.ietf.org/html/rfc2104

[59] A. Perrig, R. Canetti, D. Song, P. D. Tygar, ve B. Briscoe, "Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction", Internet Engineering Task Force, Request for Comments RFC 4082, Haz. 2005. doi: 10.17487/RFC4082.

[60] J. Anderson, S. Lo, ve T. Walter, "Time Synchronization for TESLA-based GNSS-enabled Systems", program adı: 35th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2022), Denver, Colorado, Eki. 2022, ss. 3408-3417. doi: 10.33012/2022.18442.

[61] European Union Agency for the Space Programme, “EUROPEAN GNSS (GALILEO) OPEN SERVICE - SERVICE DEFINITION DOCUMENT”, sy 1.2, 2021.

[62] European GNSS Supervisory Authority., *Galileo Open Service Navigation Message Authentication (OS-NMA): info note.* LU: Publications Office, 2021. Erişim: 05 Mayıs 2023. [Çevrimiçi]. Erişim adresi: https://data.europa.eu/doi/10.2878/49446

[63] “Galileo’s Proposed Authentication Algorithm: Part 2”, *Bert Hubert’s writings*, 26 Ağustos 2020. https://berthub.eu/articles/posts/galileos-authentication-algorithm-part-2/ (erişim 11 Mart 2023).

[64] European GNSS Supervisory Authority., *Galileo Open Service Navigation Message Authentication (OSNMA): receiver guidelines : Issue 1.0, December 2022.* LU: Publications Office, 2022. Erişim: 19 Şubat 2023. [Çevrimiçi]. Erişim adresi: https://data.europa.eu/doi/10.2878/256023

[65] J. Martin, J. Burbank, W. Kasch, ve P. D. L. Mills, “Network Time Protocol Version 4: Protocol and Algorithms Specification”, Internet Engineering Task Force, Request for Comments RFC 5905, Haz. 2010. doi: 10.17487/RFC5905.

[66] D. F. Franke, D. Sibold, K. Teichel, M. Dansarie, ve R. Sundblad, “Network Time Security for the Network Time Protocol”, Internet Engineering Task Force, Request for Comments RFC 8915, Eyl. 2020. doi: 10.17487/RFC8915.

[67] P. Zajac, “Ephemeral Keys Authenticated with Merkle Trees and Their Use in IoT Applications”, *Sensors*, c. 21, sy 6, s. 2036, Mar. 2021, doi: 10.3390/s21062036.

[68] R. Housley, “Cryptographic Message Syntax (CMS)”, Internet Engineering Task Force, Request for Comments RFC 5652, Eyl. 2009. doi: 10.17487/RFC5652.

[69] International Telecommunication Union (ITU), “ITU-T Recommendation X.667 (2004): "Information technology – Open Systems Interconnection – Procedures for the operation of OSI Registration Authorities: Generation and registration of Universally Unique Identifiers (UUIDs) and their use as ASN.1 object identifier components”, 2004.

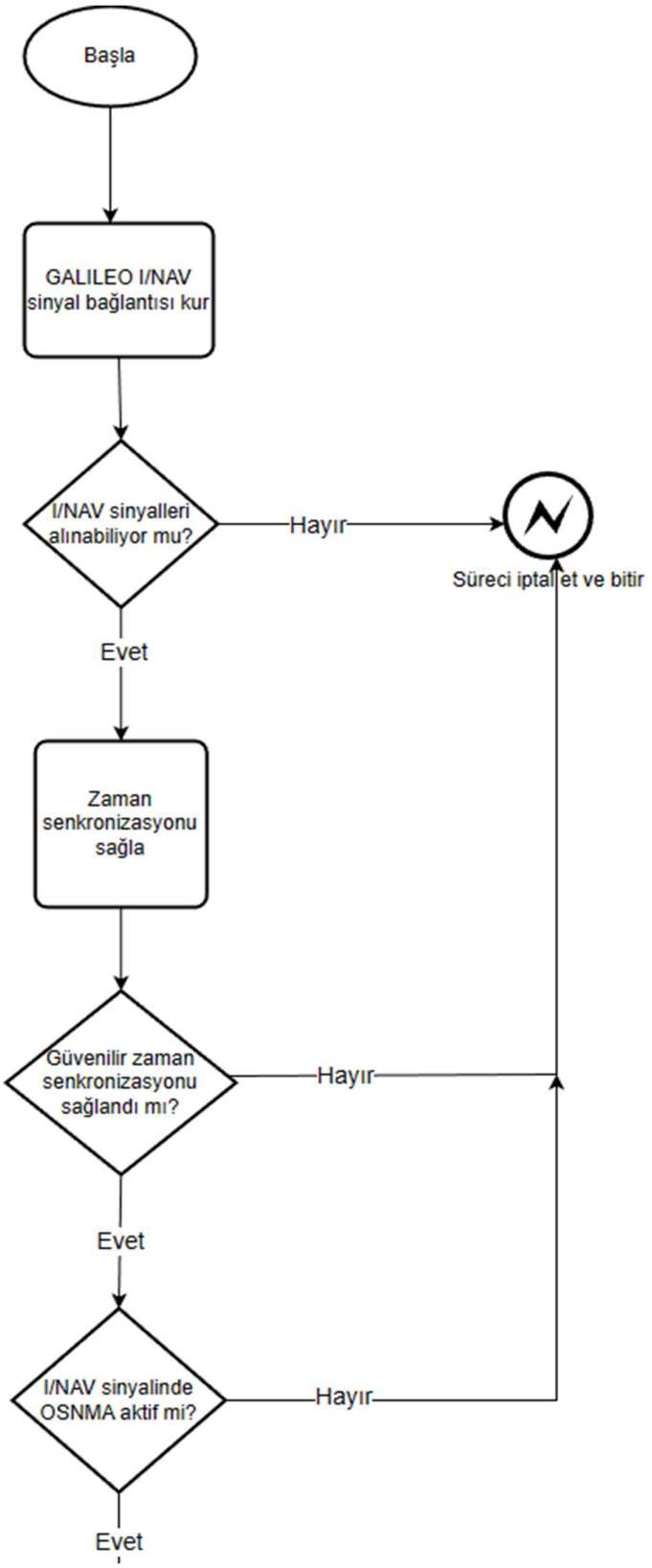
[70] E. Rescorla, “RFC 8446: The Transport Layer Security (TLS) Protocol Version 1.3”. RFC Editor, Ağustos 2018. doi: 10.17487/RFC8446.

[71] G. Kim, J. Lim, ve J. Kim, “Secure user authentication based on the trusted platform for mobile devices”, *EURASIP J. Wirel. Commun. Netw.*, c. 2016, sy 1, s. 233, Ara. 2016, doi: 10.1186/s13638-016-0729-7.

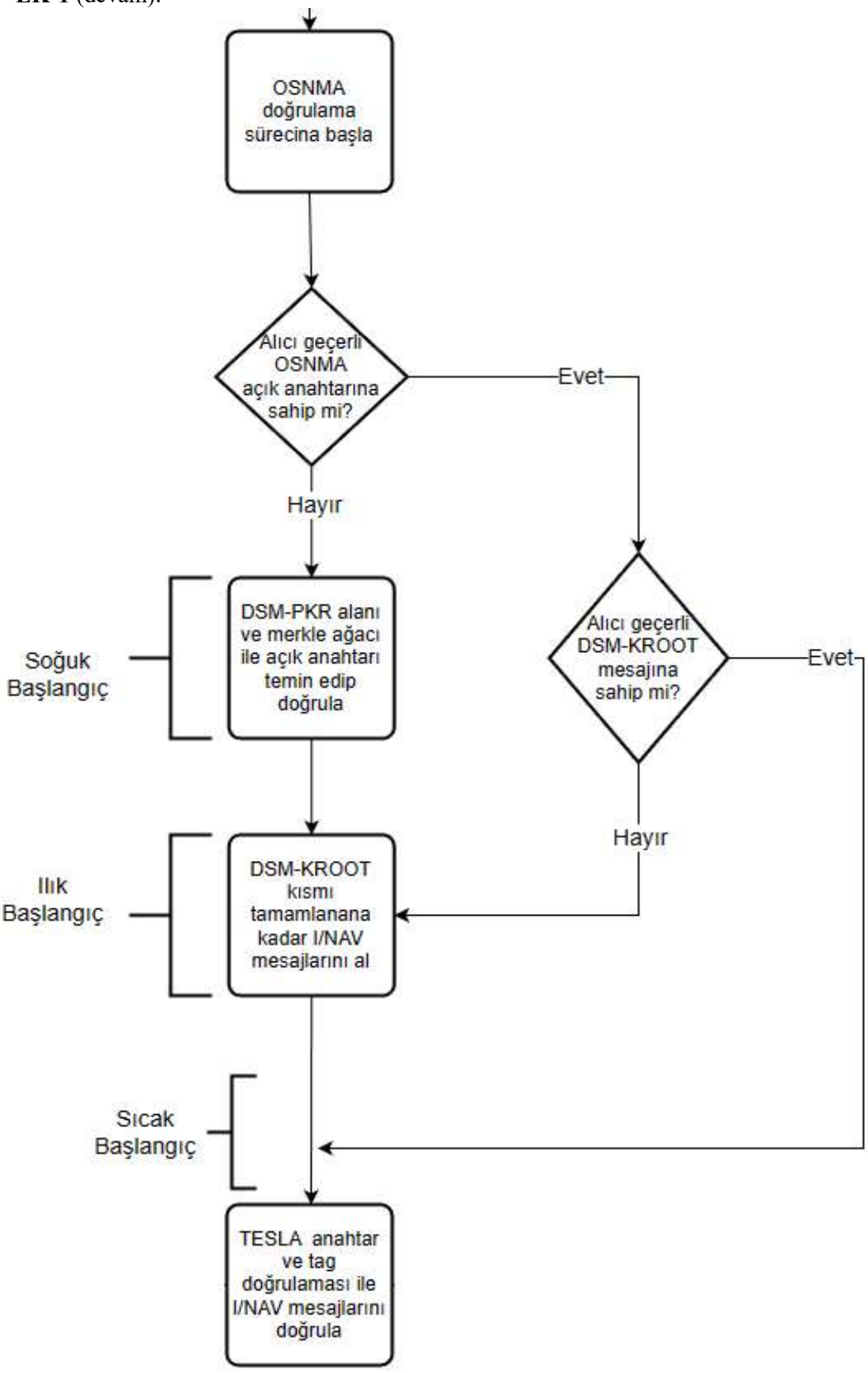
[72] A. Galan, I. Fernandez-Hernandez, L. Cucchi, ve G. Seco-Granados, “OSNMAlib: An Open Python Library for Galileo OSNMA”, içinde *2022 10th Workshop on Satellite Navigation Technology (NAVITEC)*, Noordwijk, Netherlands: IEEE, Nis. 2022, ss. 1-12. doi: 10.1109/NAVITEC53682.2022.9847548.

EKLER

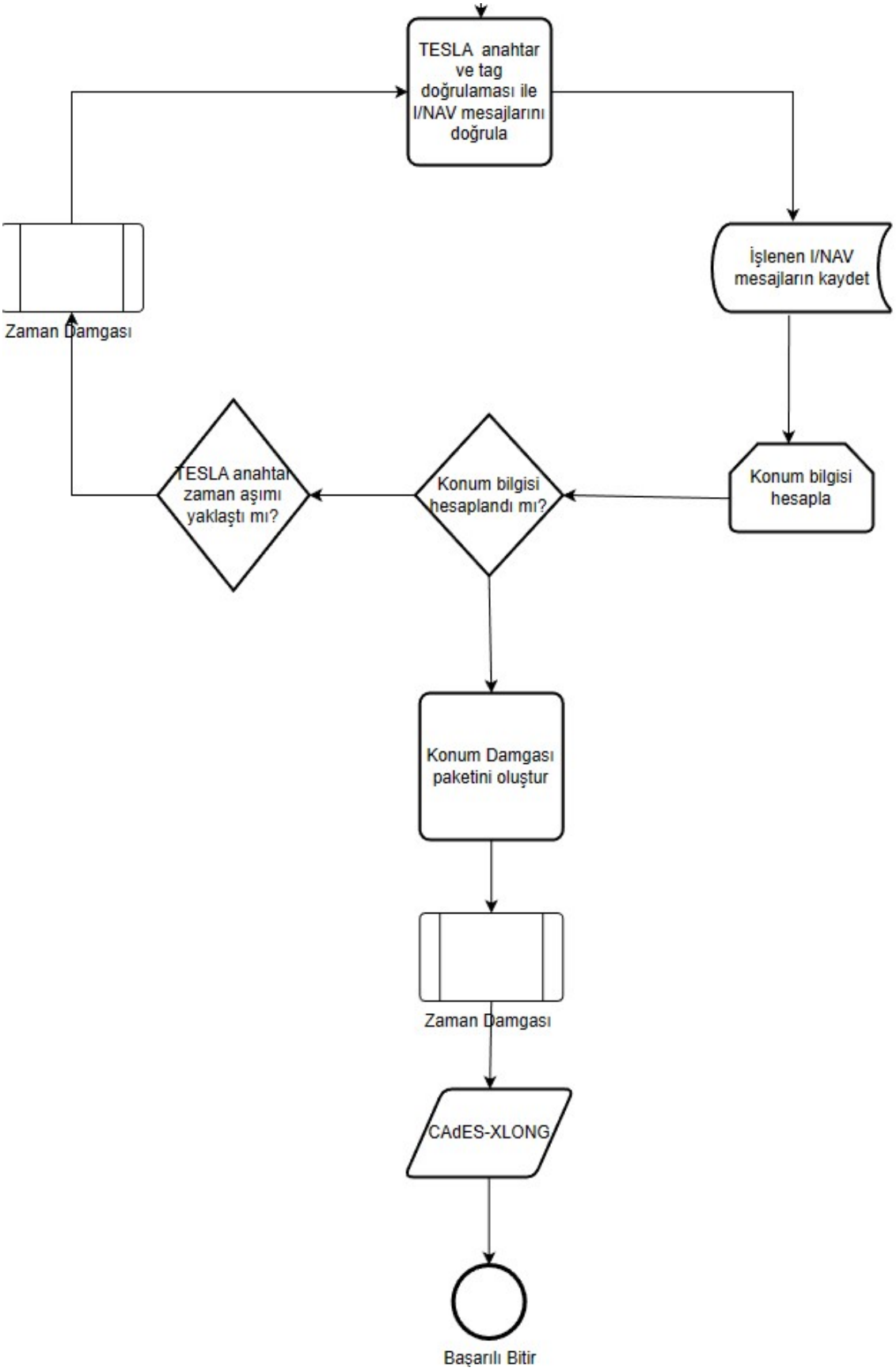
EK-1. Önerilen Konum Damgası İşlemi Süreci:



EK-1 (devam):


OSNMA doğrulama sürecina başla
Alıcı geçerli OSNMA açık anahtarına sahip mi?
Evet
Hayır
Soğuk Başlangıç
DSM-PKR alanı ve merkle ağacı ile açık anahtarı temin edip doğrula
Alıcı geçerli DSM-KROOT mesajına sahip mi?
Evet
Hayır
Ilık Başlangıç
DSM-KROOT kısmı tamamlanana kadar I/NAV mesajlarını al
Sıcak Başlangıç
TESLA anahtar ve tag doğrulaması ile I/NAV mesajlarını doğrula

EK-1 (devam):


TESLA anahtar ve tag doğrulaması ile I/NAV mesajlarını doğrula
İşlenen I/NAV mesajların kaydet
Zaman Damgası
Konum bilgisi hesapla
Konum bilgisi hesaplandı mı?
TESLA anahtar zaman aşımı yaklaştı mı?
Konum Damgası paketini oluştur
Zaman Damgası
CAdES-XLONG
Başarılı Bitir

EK-2. Önerilen Konum Damgası İçin Gereken Örnek ASN.1 Obje Tanımları (Java):

```
import com.objsys.asn1j.runtime.*;
import java.io.*;
import java.util.*;

// LatitudeSign
public class LatitudeSign extends Asn1Enumerated {

    public static final int NORTH = 0;
    public static final int SOUTH = 1;

    public LatitudeSign() {
        super();
    }

    public LatitudeSign(int value) {
        super(value);
    }

    protected void _decodeContents(InputStream in) throws IOException,
Asn1Exception {
        setEnumValue((int) Asn1Integer.decodeLong(in));
    }

    public void _encodeContents(OutputStream out) throws IOException,
Asn1Exception {
        Asn1Integer.encodeLong(getEnumValue(), out);
    }

    public String toString() {
        int value = getEnumValue();
        switch (value) {
            case NORTH:
                return "north";
            case SOUTH:
                return "south";
            default:
                return "unknown";
        }
    }
}

// DegreesLatitude
public class DegreesLatitude extends Asn1Integer {

    public DegreesLatitude() {
        super();
    }

    public DegreesLatitude(long value) {
        super(value);
    }
}
```

```
// DegreesLongitude
public class DegreesLongitude extends Asn1Integer {

    public DegreesLongitude() {
        super();
    }

    public DegreesLongitude(long value) {
        super(value);
    }
}

// GeographicalCoordinates
public class GeographicalCoordinates extends Asn1Sequence {

    public LatitudeSign latitudeSign;
    public DegreesLatitude degreesLatitude;
    public DegreesLongitude degreesLongitude;

    public GeographicalCoordinates() {
        super();
    }

    public GeographicalCoordinates(LatitudeSign latitudeSign,
DegreesLatitude degreesLatitude, DegreesLongitude degreesLongitude) {
        super();
        this.latitudeSign = latitudeSign;
        this.degreesLatitude = degreesLatitude;
        this.degreesLongitude = degreesLongitude;
    }

    protected void _decodeContents(InputStream in) throws IOException,
Asn1Exception {
        latitudeSign = new LatitudeSign();
        latitudeSign.decode(in);
        degreesLatitude = new DegreesLatitude();
        degreesLatitude.decode(in);
        degreesLongitude = new DegreesLongitude();
        degreesLongitude.decode(in);
    }

    public void _encodeContents(OutputStream out) throws IOException,
Asn1Exception {
        latitudeSign.encode(out);
        degreesLatitude.encode(out);
        degreesLongitude.encode(out);
    }

    public String toString() {
        return "GeographicalCoordinates{latitudeSign=" + latitudeSign
+ ", degreesLatitude=" + degreesLatitude + ", degreesLongitude=" +
degreesLongitude + "}";
    }
}
```

```
// PRNShift
public class PRNShift extends Asn1Sequence {

    public Asn1OctetString localPRN;
    public Asn1Integer svID;
    public Asn1OctetString svPRN;

    public PRNShift() {
        super();
    }

    public PRNShift(Asn1OctetString localPRN, Asn1Integer svID,
Asn1OctetString svPRN) {
        super();
        this.localPRN = localPRN;
        this.svID = svID;
        this.svPRN = svPRN;
    }

    protected void _decodeContents(InputStream in) throws IOException,
Asn1Exception {
        localPRN = new Asn1OctetString();
        localPRN.decode(in, true);
        svID = new Asn1Integer();
        svID.decode(in);
        svPRN = new Asn1OctetString();
        svPRN.decode(in, true);
    }

    public void _encodeContents(OutputStream out) throws IOException,
Asn1Exception {
        localPRN.encode(out, true);
        svID.encode(out);
        svPRN.encode(out, true);
    }

    public String toString() {
        return "PRNShift{localPRN=" + localPRN + ", svID=" + svID + ",
svPRN=" + svPRN + "}";
    }
}

// RangingLog
public class RangingLog extends Asn1SequenceOf<PRNShift> {

    public RangingLog() {
        super();
    }

    public RangingLog(List<PRNShift> elements) {
        super(elements);
    }

    public RangingLog(PRNShift... elements) {
```

```
        super(Arrays.asList(elements));
    }

    protected PRNShift _createElementType() {
        return new PRNShift();
    }

    protected void _decodeContents(InputStream in) throws IOException,
Asn1Exception {
        while (in.available() > 0) {
            PRNShift element = _createElementType();
            element.decode(in);
            add(element);
        }
    }

    public void _encodeContents(OutputStream out) throws IOException,
Asn1Exception {
        for (PRNShift element : this) {
            element.encode(out);
        }
    }

    public String toString() {
        StringBuilder sb = new StringBuilder();
        sb.append("RangingLog{");
        for (int i = 0; i < size(); i++) {
            if (i > 0) {
                sb.append(", ");
            }
            sb.append(get(i));
        }
        sb.append("}");
        return sb.toString();
    }
}

// GNSSAuthenticationData
public class GNSSAuthenticationData extends Asn1Sequence {

    public Asn1GeneralizedTime gnssTime;
    public Asn1BitString iod;
    public Asn1OctetString nmaSIS;
    public TimeStampToken ts;

    public GNSSAuthenticationData() {
        super();
    }

    public GNSSAuthenticationData(Asn1GeneralizedTime gnssTime,
Asn1BitString iod, Asn1OctetString nmaSIS, TimeStampToken ts) {
        super();
        this.gnssTime = gnssTime;
        this.iod = iod;
        this.nmaSIS = nmaSIS;
```

```
        this.ts = ts;
    }

    protected void _decodeContents(InputStream in) throws IOException,
Asn1Exception {
        gnssTime = new Asn1GeneralizedTime();
        gnssTime.decode(in);
        iod = new Asn1BitString();
        iod.decode(in);
        nmaSIS = new Asn1OctetString();
        nmaSIS.decode(in, true);
        ts = new TimeStampToken();
        ts.decode(in);
    }

    public void _encodeContents(OutputStream out) throws IOException,
Asn1Exception {
        gnssTime.encode(out);
        iod.encode(out);
        nmaSIS.encode(out, true);
        ts.encode(out);
    }

    public String toString() {
        return "GNSSAuthenticationData{gnssTime=" + gnssTime + ",
iod=" + iod + ", nmaSIS=" + nmaSIS + ", ts=" + ts + "}";
    }
}
// LocationStampInfo
public class LocationStampInfo extends Asn1Sequence {

    public RangingLog rngLog;
    public GeographicalCoordinates coordinates;
    public Asn1SequenceOf<GNSSAuthenticationData> gnssAuth;

    public LocationStampInfo() {
        super();
    }

    public LocationStampInfo(RangingLog rngLog,
GeographicalCoordinates coordinates,
Asn1SequenceOf<GNSSAuthenticationData> gnssAuth) {
        super();
        this.rngLog = rngLog;
        this.coordinates = coordinates;
        this.gnssAuth = gnssAuth;
    }

    protected void _decodeContents(InputStream in) throws IOException,
Asn1Exception {
        if (in.available() > 0 && Asn1Tag.peekTag(in) !=
Asn1Tag.SEQUENCE) {
            rngLog = new RangingLog();
            rngLog.decode(in);
        }
```

```
        coordinates = new GeographicalCoordinates();
        coordinates.decode(in);
        gnssAuth = new Asn1SequenceOf<>(GNSSAuthenticationData::new);
        gnssAuth.decode(in);
    }

    public void _encodeContents(OutputStream out) throws IOException,
Asn1Exception {
        if (rngLog != null) {
            rngLog.encode(out);
        }
        coordinates.encode(out);
        gnssAuth.encode(out);
    }

    public String toString() {
        return "LocationStampInfo{rngLog=" + rngLog + ", coordinates="
+ coordinates + ", gnssAuth=" + gnssAuth + "}";
    }
}

// LocationStampToken
public class LocationStampToken extends ContentInfo {

    public LocationStampToken() {
        super();
    }

    public LocationStampToken(String contentType, Asn1Type content) {
        super(contentType, content);
    }

    public LocationStampToken(Asn1ObjectIdentifier contentType,
Asn1Type content) {
        super(contentType, content);
    }
}
```

EK-3. Önerilen Konum Damgası İçin Örnek Attribute Ekleme (Java):

```
import com.objsys.asn1j.runtime.*;
import tr.gov.tubitak.uekae.esya.api.cmssignature.*;
import tr.gov.tubitak.uekae.esya.api.cmssignature.attribute.*;

import java.io.*;
import java.math.*;
import java.nio.charset.*;
import java.util.*;
```

```
public class TestLocationAttribute extends AttributeValue {

    @Override
    public Asn1ObjectIdentifier getAttributeOID() {

        String oid = "2.25.142312163956071652603888631318689442116";
        Asn1BerDecodeBuffer decodeBuffer = new
Asn1BerDecodeBuffer(oid.getBytes(StandardCharsets.UTF_8));
        Asn1ObjectIdentifier asn1ObjectIdentifier = new
Asn1ObjectIdentifier();

        try {
            asn1ObjectIdentifier.decode(decodeBuffer, false,
oid.length());
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

        return asn1ObjectIdentifier;
    }

    @Override
    public boolean isSigned() {
        return true;
    }

    @Override
    public void setValue() throws CMSSignatureException {
        this._setValue(new LocationStampToken());

    }
}
```

EK-4. Önerilen Konum Damgası Attribute İle Örnek İmzalama (Java):

```
import tr.gov.tubitak.uekae.esya.api.asn.x509.*;
import tr.gov.tubitak.uekae.esya.api.certificate.validation.policy.*;
import tr.gov.tubitak.uekae.esya.api.cmssignature.*;
import tr.gov.tubitak.uekae.esya.api.cmssignature.attribute.*;
import tr.gov.tubitak.uekae.esya.api.cmssignature.signature.*;
import tr.gov.tubitak.uekae.esya.api.common.*;
import tr.gov.tubitak.uekae.esya.api.common.crypto.*;
import tr.gov.tubitak.uekae.esya.api.common.util.*;
import tr.gov.tubitak.uekae.esya.api.common.util.bag.*;
import tr.gov.tubitak.uekae.esya.api.crypto.*;
import tr.gov.tubitak.uekae.esya.api.crypto.Signer;
import tr.gov.tubitak.uekae.esya.api.crypto.alg.*;
import tr.gov.tubitak.uekae.esya.api.crypto.util.*;
import tr.gov.tubitak.uekae.esya.api.smartcard.pkcs11.*;
import tr.gov.tubitak.uekae.esya.asn.util.*;
```

```
import java.io.*;
import java.nio.charset.*;
import java.nio.file.*;
import java.security.*;
import java.util.*;

public class SimpleSignature {


    private static final String PFX_FILE_PATH = "pfx/test_745418.pfx";
    private static final String PFX_PIN = "745418";


    public static ECertificate getCertificateFromPFX() throws
Exception {
        //Pfx okunuyor.
        InputStream is =
Thread.currentThread().getContextClassLoader().getResourceAsStream(PFX
_FILE_PATH);
        PfxParser pfxParser = new PfxParser(is,
PFX_PIN.toCharArray());
        List<Pair<ECertificate, PrivateKey>> entries =
pfxParser.getCertificatesAndKeys();
        return entries.get(0).getObject1();
    }

    public static Signer getSignerFromPFX() throws Exception {
        //Pfx okunuyor.
        InputStream is =
Thread.currentThread().getContextClassLoader().getResourceAsStream(PFX
_FILE_PATH);
        PfxParser pfxParser = new PfxParser(is,
PFX_PIN.toCharArray());
        List<Pair<ECertificate, PrivateKey>> entries =
pfxParser.getCertificatesAndKeys();

        Signer signer = Crypto.getSigner(SignatureAlg.RSA_SHA256);
        signer.init(entries.get(0).getObject2());
        return signer;
    }

    public static void main(String[] args) {

        try {

LicenseUtil.setLicenseXml(Thread.currentThread().getContextClassLoader
().getResourceAsStream("lisans/lisans.xml"));
        } catch (ESYAException e) {
            throw new RuntimeException(e);
        }

        BaseSignedData bs = new BaseSignedData();
        ISignable content = new SignableByteArray("test".getBytes());

        try {
```

```
            bs.addContent(content);
        } catch (CMSSignatureException e) {
            throw new RuntimeException(e);
        }

        HashMap<String, Object> params = new HashMap<String,
Object>();

        try {
            params.put(EParameters.P_CERT_VALIDATION_POLICY,
PolicyReader.readValidationPolicy(Thread.currentThread().getContextCla
ssLoader().getResourceAsStream("config/certval-policy.xml")));
        } catch (ESYAException e) {
            throw new RuntimeException(e);
        }

        BaseSigner signer = null;
        ECertificate cert = null;
        try {
            cert = getCertificateFromPFX();
            signer = getSignerFromPFX();

        } catch (ESYAException e) {
            throw new RuntimeException(e);
        } catch (Exception e) {
            throw new RuntimeException(e);
        }

        List<IAttribute> optionalAttributes = new ArrayList<>();
        optionalAttributes.add(new TestLocationAttribute());

        try {
            bs.addSigner(ESignatureType.TYPE_BES, cert, signer,
optionalAttributes, params);
        } catch (CMSSignatureException e) {
            throw new RuntimeException(e);
        }

        byte[] signedDocument = bs.getEncoded();

        try {
            Files.write(Paths.get("test-output/bes.p7s"),
signedDocument);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

    }
}
```

EK-5. EK 3 ve EK 4'teki örnek kodlarda kullanılan MA3 API Kütüphanesi Lisansı

MA3 API – KULLANICI LİSANS SÖZLEŞMESİ

İşbu Kullanıcı Lisans Sözleşmesi (bundan böyle "Sözleşme" olarak anılacaktır) TÜBİTAK BİLGEM ile KULLANICI arasındaki yasal anlaşmadır.

Sözleşme'de kullanılan kısaltmalar aşağıda tanımlanmıştır:

"TÜBİTAK BİLGEM", TÜBİTAK Bilişim ve Bilgi Güvenliği İleri Teknolojiler Araştırma Merkezi kamu tüzel kişiliğidir.

"MA3 API", TÜBİTAK BİLGEM tarafından geliştirilen elektronik imza yazılım kütüphanesi ve tüm bileşenleridir.

"KULLANICI", MA3 API'yi kullanarak yazılım geliştiren gerçek veya tüzel kişilerdir.

"SON KULLANICI", MA3 API'nin kullanıldığı yazılımı kullanan gerçek veya tüzel kişilerdir.

"UYUM DEĞERLENDİRME", elektronik imza atan ve/veya doğrulayan uygulamaların elektronik imza mevzuatında geçen uluslararası teknik standartlara uygunluğunun değerlendirildiği ve denetiminin yapıldığı yetkili mercilerdir.

KULLANICI, MA3 API'yi indirerek, kopyalayarak, bulundurarak veya kullanarak burada belirtilen tüm şart ve hükümlerin tamamına uymayı taahhüt ettiğini ve bu şart ve hükümler ile yasal olarak bağlı olduğunu kabul etmiş olur:

1) İşbu Sözleşme hükümleri, MA3 API'nin TÜBİTAK BİLGEM tarafından değiştirilmediği, yenilenmediği veya yeni sözleşme ile ortadan kaldırılmadığı sürece geçerlidir.

2) TÜBİTAK BİLGEM ve KULLANICI arasında farklı bir sözleşme olmadığı sürece işbu Sözleşme, MA3 API'ye ilişkin daha önceki tüm beyanları, görüşmeleri, yükümlülükleri, haberleşmeleri veya tanıtımları geçersiz kılar ve bunların yerine geçer.

3) TÜBİTAK BİLGEM, işbu Sözleşme'de değişiklik yapma hakkını saklı tutar.

4) TÜBİTAK BİLGEM, MA3 API'nin ücret, kullanım ve dağıtım politikalarında değişiklik yapma hakkını saklı tutar.

5) MA3 API veya MA3 API'den türetilmiş imzalama kütüphaneleri, TÜBİTAK BİLGEM'in yazılı onayı olmadan satılamaz, alt lisansı verilemez, kiralanamaz, ödünç verilemez veya ödünç alınamaz.

6) MA3 API'nin kullanıldığı uygulama veya yazılım bileşenleri için UYUM DEĞERLENDİRME onayı alınması KULLANICI'nın sorumluluğundadır.

7) MA3 API'nin kullanıldığı ürünlerdeki süreç, konfigürasyon, politika yönetimi ve kullanım ile ilgili tüm sorumluluk ve risk KULLANICI'ya aittir.

8) TÜBİTAK BİLGEM, KULLANICI ve/veya SON KULLANICI tarafından MA3 API kullanılarak icra edilecek görevlere/yürütülecek işlere yönelik olarak herhangi bir sorumluluk almamakta ve KULLANICI, SON KULLANICI ve/veya üçüncü taraflar nezdinde herhangi bir taahhüt altına girmemektedir. KULLANICI ve/veya SON KULLANICI MA3 API'nin kullanılması sonucu uğrayabileceği doğrudan veya dolaylı herhangi bir kayıp ve zarardan dolayı, TÜBİTAK ve TÜBİTAK BİLGEM'in sorumlu olmadığını, kabul ve taahhüt eder.

9) KULLANICI, işbu Sözleşme'nin hükümlerini ihlali sebebi ile, TÜBİTAK'ın ve TÜBİTAK BİLGEM'in doğabilecek tüm zararlarının tazmininden doğrudan sorumludur. KULLANICI'nın işbu Sözleşme hükümlerinden herhangi birini ihlali durumunda, TÜBİTAK'ın ve TÜBİTAK BİLGEM'in ihtiyati tedbir kararı aldırmak da dahil olmak üzere tüm yasal hakları saklıdır.

10) MA3 API için bir teknik destek taahhüdü bulunmamaktadır. Destek talepleri münferiden değerlendirilip, uygun görüldüğü takdirde belirli bir ücret karşılığında destek verilebilecektir. TÜBİTAK BİLGEM kendi takdirine bağlı olarak, sorunun çözülmesinin teknik desteğin kapsamının dışında olduğuna hükmetme, teknik destek sağlamayı reddetme, askıya alma veya sonlandırma hakkını saklı tutar.

EK 6. EK 2, EK 3, EK 4'teki Örnek Kodlarda Kullanılan ASN1J Kütüphanesi Lisansı

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.

Özgeçmiş

Bu alan 6689 sayılı KVKK hükümleri kapsamı dolayısıyla kaldırılmıştır. Orijinal içerikte bulunabilir.