T.C.

TRAKYA ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

NESNELERİN İNTERNETİ UYGULAMALARININ GÜVENLİĞİ İÇİN HAFİF SIKLET KRİPTOGRAFİK ALGORİTMALARIN ANALİZİ VE GÜVENLİ AKILLI BİR PLATFORM UYGULAMASI

FÜSUN YAVUZER ASLAN

DOKTORA TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

Tez Danışmanı: PROF. DR. M. TOLGA SAKALLI

EDİRNE-2020

Füsun YAVUZER ASLAN'ın hazırladığı "**NESNELERİN İNTERNETİ UYGULAMALARININ GÜVENLİĞİ İÇİN HAFİF SIKLET KRİPTOGRAFİK ALGORİTMALARIN ANALİZİ VE GÜVENLİ AKILLI BİR PLATFORM UYGULAMASI**" başlıklı bu tez, tarafımızca okunmuş, kapsam ve niteliği açısından **Bilgisayar Mühendisliği Anabilim Dalında** bir **Doktora tezi** olarak kabul edilmiştir.

Jüri Üyeleri	**İmza**
Prof. Dr. M. Tolga SAKALLI	……..
Prof. Dr. Yılmaz ÇAN	……..
Doç. Dr. Erdinç UZUN	……..
Doç. Dr. Sedat AKLEYLEK	……..
Dr. Öğr. Üyesi Deniz TAŞKIN	……..

Tez Savunma Tarihi: 15/09/2020

Bu tezin Doktora tezi olarak gerekli şartları sağladığını onaylarım.

İmza

Prof. Dr. M. Tolga SAKALLI
Tez Danışmanı ……

Trakya Üniversitesi Fen Bilimleri Enstitüsü onayı

Doç.Dr. H.R. Ferhat KARABULUT

Fen Bilimleri Enstitüsü Müdürü

T.Ü.FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI DOKTORA PROGRAMI
DOĞRULUK BEYANI

Trakya Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında, tüm verilerin bilimsel ve akademik kurallar çerçevesinde elde edildiğini, kullanılan verilerde tahrifat yapılmadığını, tezin akademik ve etik kurallara uygun olarak yazıldığını, kullanılan tüm literatür bilgilerinin bilimsel normlara uygun bir şekilde kaynak gösterilerek ilgili tezde yer aldığını ve bu tezin tamamı ya da herhangi bir bölümünün daha önceden Trakya Üniversitesi ya da farklı bir üniversitede tez çalışması olarak sunulmadığını beyan ederim.

15 / 09 / 2020

Füsun YAVUZER ASLAN

Doktora Tezi
Nesnelerin İnterneti Uygulamalarının Güvenliği için Hafif Sıklet Kriptografik Algoritmaların Analizi ve Güvenli Akıllı Bir Platform Uygulaması
T.Ü. Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

ÖZET

Son on yılda, nesnelerin interneti (IoT) kapsamında geliştirilen uygulamalar sensör ve iletişim teknolojilerinin de ilerlemesiyle sessiz ve kademeli olarak hayatımızda yer almaya başlamıştır. Küresel ölçüde, cihazlar ve bağlantılar olarak nitelendirilebilen nesnelerin sayısı insan nüfusundan daha hızlı artmaktadır. Akıllı sayaçlar, sağlık bakımı izleme, taşıma ve paketleme veya varlık izleme gibi artan sayıda M2M uygulaması, cihazların ve bağlantıların büyümesine önemli katkıda bulunmaktadır. Bu kadar hızlı büyüyen nesnelerin interneti ekosistemi ciddi güvenlik açıklarına sahiptir.

Tezin amacı nesnelerin interneti uygulamalarının güvenliğini artırmak için kullanılabilecek hafif sıklet kriptografik algoritmalarının analiz edilmesi ve güvenli bir platform oluşturulmasıdır. Nesnelerin interneti uygulamalarında kullanılan mimari yapılar ve protokoller incelenerek bu bağlamda tehditler belirlenmiştir. Seçilen hafif sıklet kriptografik algoritmalar ile güvenli bir test ortamı oluşturulmuş ve saldırılara karşı test edilmiştir.

Tez 10 bölümden oluşmaktadır. 1. Bölümde, nesnelerin interneti kavramına giriş yapılmıştır. 2. Bölümde, literatürde yer alan nesnelerin interneti mimarileri irdelenmiştir. 3. Bölümde, nesnelerin interneti uygulamalarında kullanılan teknoloji ve protokoller ele alınmıştır. 4. Bölümde, nesnelerin interneti kapsamında meydana gelen siber güvenlik olayları kronolojik olarak listelenmiş ve mevcut tehditler mimariye göre sınıflandırılarak açıklanmıştır. 5. Bölümde, tez içinde kullanılacak iletişim protokolleri anlatılmıştır. 6.

Bölümde, nesnelerin interneti uygulamalarında güvenliği sağlamak için kullanılabilecek hafif sıklet kriptografik algoritmalar analiz edilmiştir. 7. Bölümde, seçilen algoritmalar yazılımsal ve donanımsal olarak gerçeklenmiştir. Ayrıca algoritmalar belirlenen senaryoya göre çalıştırılıp enerji ve güç tüketimleri analiz edilmiştir. 8. Bölümde, kriptografik algoritmalar kurulan iletişim test ortamında test edilmiştir. 9. Bölümde, test ortamına sık kullanılan saldırılardan olan ortadaki adam saldırısı düzenlenerek sistemin güvenliği analiz edilmiştir. 10. Bölümde ise sonuçlara ve gelecekte yapılması planlanan çalışmalara yer verilmiştir.

Yıl : 2020

Sayfa Sayısı : 197

Anahtar Kelimeler : Nesnelerin İnterneti, Güvenlik, Hafif Sıklet Kriptografi

Doctoral Thesis
Analysis of Lightweight Cryptographic Algorithms for IoT Security and Implementation of a Secure Smart Platform
Trakya University Institute of Natural Sciences
Department of Computer Engineering

ABSTRACT

In the last decade, the Internet of Things (IoT) has begun to take place in our lives quietly and gradually due to presence of wireless communication systems. On a global scale, the number of objects that can be described as devices and connections is growing faster than the human population. Every year, new devices in different forms with increasing talent and intelligence are introduced and adopted. An increasing number of M2M applications, such as smart meters, healthcare monitoring, transportation and packaging, or asset tracking, make a significant contribution to the growth of devices and connections. The rapidly growing IoT ecosystem has serious vulnerabilities.

The aim of the thesis is to analyze the lightweight cryptographic algorithms that can be used to increase IoT security and to create a secure platform. The architectural structures and protocols used in IoT have been examined and threats in this context have been identified. A secure test environment with the chosen lightweight cryptographic algorithms has been development and tested against attacks.

The thesis composed of 10 chapters. In chapter 1, IoT concept is introduced. In chapter 2, IoT architectures in the literature are examined. In Chapter 3, the technologies and protocols used in IoT applications are discussed. In Chapter 4, IoT cyber security events are listed chronologically and current threats are classified according to architecture. The Communication protocols to be used in the thesis are explained in Chapter 5. In Chapter 6, lightweight cryptographic algorithms that can be used to provide IoT security have been analyzed. In Chapter 7, chosen algorithms are implemented to software and hardware. In addition, algorithms are testing according to the determined scenario and energy and power consumption are analyzed. In Chapter 8, cryptographic

algorithms are tested in the established communication IoT test environment. In Chapter 9, the security of the system was analyzed by attacking the test environment with the "man-in-the-middle attack" which is one of the common attacks. In Chapter 10, the results and future studies are discussed.

Year : 2020

Number of Pages : 197

Keywords : Internet of Things, Security, Lightweight Cryptography

TEŞEKKÜR

Tez çalışmam sırasında bana destek olan ve yardımlarını esirgemeyen çok değerli hocam ve danışmanım Prof. Dr. M. Tolga SAKALLI'ya teşekkür ederim.

Tezimin izleme komitesinde yer alan Prof. Dr. Yılmaz ÇAN'a ve Dr. Öğr. Üyesi Deniz TAŞKIN'a değerli katkılarından dolayı teşekkürlerimi sunarım. Ayrıca tez jürimde yer alarak yaptıkları bilimsel katkılardan dolayı Doç. Dr. Erdinç UZUN'a ve Doç. Dr. Sedat AKLEYLEK'e teşekkür ederim.

Son olarak, doktora sürecinde beni sürekli motive eden, attığım her adımda desteği ve emeği ile yanımda olan sevgili eşim Dr. Öğr. Üyesi Bora ASLAN'a sonsuz teşekkür ederim.

İÇİNDEKİLER

ÖZET iv
ABSTRACT vi
TEŞEKKÜR viii
İÇİNDEKİLER ix
KISALTMALAR DİZİNİ xiv
ÇİZELGELER DİZİNİ xvii
ŞEKİLLER DİZİNİ xix
ALGORİTMALAR DİZİNİ xxiii
BÖLÜM 1 1
1. Giriş 1
1.1. Nesnelerin İnterneti 1
1.2. Nesnelerin İnterneti Uygulama Alanları 3
1.3. Tezin Kapsamı 8
BÖLÜM 2 9
2. Nesnelerin İnterneti Mimarisi 9
2.1. Giriş 9
2.2. Üç Katmanlı Mimari 10
2.2.1. Algılama Katmanı 10
2.2.2. Ağ Katmanı 10
2.2.3. Uygulama Katmanı 11
2.3. Dört Katmanlı Mimari 11
2.3.1. Algılama Katmanı 11
2.3.2. Ağ Katmanı 11
2.3.3. Destek Katmanı 11
2.3.4. Uygulama Katmanı 12

2.4. Beş Katmanlı Mimari 12
2.4.1. Algılama Katmanı 12
2.4.2. Ağ Katmanı 13
2.4.3. İşleme Katmanı 13
2.4.4. Uygulama Katmanı 13
2.4.5. İş Katmanı 13
2.5. Nesnelerin İnterneti Mimarisinin OSI ve TCP/IP ile Karşılaştırılması 14
2.6. Katmanlarda Kullanılan Teknolojiler 15
BÖLÜM 3 19
3. Nesnelerin İnterneti Uygulamalarında Kullanılan Teknolojiler ve Protokoller 19
3.1. Giriş 19
3.2. IPv6 19
3.3. IEEE 802.15.4 21
3.4. 6LoWPAN 23
3.5. ZigBee 25
3.6. Bluetooth 28
3.7. LoRaWAN 32
3.8. Z-Wave 34
3.9. NFC 37
3.10. Sigfox 40
3.11. Sonuçlar 42
BÖLÜM 4 44
4. Nesnelerin İnterneti Uygulamalarındaki Güvenlik İhlalleri ve Tehditler 44
4.1. Giriş 44
4.2. Nesnelerin İnterneti Cihazlarının Hedef Olarak Seçilme Nedenleri 44
4.3. Nesnelerin İnterneti Cihazlarına Yapılan Bazı Saldırılar 46
4.4. Nesnelerin İnterneti Mimarisi Üzerindeki Tehditler 50
4.4.1. Algılama Katmanı Tehditleri 51
4.4.2. Ağ Katmanı Tehditleri 52
4.4.3. Destek Katmanı Tehditleri 53
4.4.4. Uygulama Katmanı Tehditleri 53

BÖLÜM 5 55
5. Tez Kapsamında İncelenen İletişim Protokolleri 55
5.1. Giriş 55
5.2. Message Queueing Telemetry Transport (MQTT) 55
5.2.1. Konu 57
5.2.2. Broker 58
5.2.3. Abone-Broker Bağlantısı 59
5.2.4. Yayın Mesajı 62
5.2.5. Abone İşlemi 64
5.2.6. Hizmet Kalitesi (QoS) seviyesi 69
5.2.7. MQTT ve Güvenlik 71
5.3. The Constrained Application Protocol (CoAP) 72
5.3.1. CoAP URI Adresleri 74
5.3.2. CoAP Mesaj Tipleri 75
5.3.3. CoAP Mesaj Yöntemleri 77
5.3.4. CoAP Çalışma Modları 81
5.3.5. CoAP ve Güvenlik 83
BÖLÜM 6 85
6. Nesnelerin İnterneti Uygulamalarının Güvenliği için Kriptografik Çözümler 85
6.1. Giriş 85
6.2. Kriptografi 85
6.3. Gelişmiş Şifreleme Standardı - AES 86
6.4. Hafif Sıklet Kriptografi 88
6.4.1. CLEFIA Blok Şifreleme Algoritması 91
6.4.2. PRESENT Blok Şifreleme Algoritması 99
6.4.3. PICCOLO Blok Şifreleme Algoritması 103
6.4.4. PRINCE Blok Şifreleme Algoritması 109
6.4.5. LBLOCK Blok Şifreleme Algoritması 111
BÖLÜM 7 116
7. Seçilen Hafif Sıklet Şifreleme Algoritmalarının Performans Analizi 116
7.1. Giriş 116

7.2. Test Ortamı 116
7.3. Test Senaryosu 117
7.4. CLEFIA Blok Şifreleme Algoritması 118
7.4.1. 128 Bit Şifreleme 118
7.4.2. 192 Bit Şifreleme 120
7.4.3. 256 Bit Şifreleme 121
7.5. PRESENT Blok Şifreleme Algoritması 123
7.6. PICCOLO Blok Şifreleme Algoritması 125
7.6.1. 80 Bit Şifreleme 125
7.6.2. 128 Bit Şifreleme 127
7.7. PRINCE Blok Şifreleme Algoritması 129
7.8. LBLOCK Blok Şifreleme Algoritması 131
7.9. Test Sonuçları 133
BÖLÜM 8 139
8. Güvenli İletişim İçin Test Ortamının Kurulması Ve Test Edilmesi 139
8.1. Giriş 139
8.2. MQTT Test Ortamı 139
8.3. CoAP Test Ortamı 142
8.4. Test Senaryoları 144
8.4.1. Test Senaryosu 1 145
8.4.2. Test Senaryosu 2 145
8.4.3. Test Senaryosu 3 146
8.4.4. Test Senaryosu 4 147
8.5. Test Sonuçları 147
BÖLÜM 9 148
9. Test Ortamına Saldırı Senaryosunun Uygulanması 148
9.1. Giriş 148
9.2. Ortadaki Adam Saldırısı (Man-In-The-Middle MITM) 148
9.3. ARP Zehirlenmesi 149
9.4. Kali Linux ile MITM Saldırısı 152
9.5. WebSploit 152

9.6. MQTT iletişim ortamına MITM Saldırı Senaryosu 156
9.7. CoAP iletişim ortamına MITM Saldırı Senaryosu 158
9.8. Test Sonuçları 161
BÖLÜM 10 162
10. SONUÇLAR ve TARTIŞMA 162
KAYNAKLAR 166
ÖZGEÇMİŞ 173
TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER 174

KISALTMALAR DİZİNİ

AES	Advanced Encryption Standard
AM	Active Mode
AMQP	Advanced Message Queuing Protocol
ARP	Address Resolution Protocol
ATT	Attribute Protokol
BLE	Bluetooth Low Energy
BPSK	Binary Phase-Shift Keying
BR/EDR	Bluetooth Basic Rate/Enhanced Data Rate
CCS	Code Composer Studio
CCTV	Close Circuit Television
CRC	Cyclic Redundancy Check
CoAP	Constrained Application Protocol
CSMA-CD	Carrier Sense Multiple Access With Collision Detection
DDoS	Distributed Denial of Service
DDS	Data Distribution Service
DES	Data Encryption Standart
DNS	Domain Name Server
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
EEG	Elektroensefalogram
EKG	Electrocardiography
FFD	Full-Function Device
FRAM	Ferroelectric Random Access Memory
GAP	Generic Access Profile

GATT	Generic Attribute Protocol
GE	Gate Equivalent
GFSK	Gaussian Frequency-Shift Keying
GPS	Global Positioning System
HCI	Host Controller Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
ISM	Industrial Scientific Medical Band
ISO	International Organization for Standardization
ITU	International Telecommunication Union
L2CAP	Logical Link Control and Adaptation Protocol
LAN	Local Area Network
LoRaWAN	Long Range Wide Area Network
LPM	Low Power Mod
LR-WPAN	Low Rate Wireless Personal Area Network
MAC	Media Access Control
MIC	Message Integrity Code
MITM	Man-in-the-middle Attack
MQTT	Message Queuing Telemetry Transport
M2M	Machine to Machine
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NOOBS	New Out Of the Box Software
NWK	ZigBee Network Layer
OEM	Original Equipment Manufacturer

OSI	Open Systems Interconnection
PAN	Personal Area Network
PLC	Programmable Logic Controllers
SSL	Secure Socket Layer
QoS	Quality of Service
RFD	Reduced-Function Device
RFID	Radio Frequency Identification
SCADA	Supervisory Control and Data Acquisition
SNMP	Simple Network Management Protocol
SPN	Substitution-Permutation Network
SSH	Secure Shell
TCP/IP	Transmission Control Protocol/Internet Protocol
TI	Texas Instruments
TLS	Transport Layer Security
UDP	User Datagram Protocol
UMTS	Universal Mobile Telecommunications System
UNB	Ultra Narrow Ban
URI	Uniform Resource Identifier
URL	Uniform Resource Loader
UuID	Universally Unique Identifier
VPN	Virtual Private Network
WAN	Wide Area Network
6LowPAN	IPv6 over Low Power Wireless Personal Area Networks

ÇİZELGELER DİZİNİ

Çizelge 3.1. Bluetooth sınıfları 29
Çizelge 3.2. Protokollerin karşılaştırılması 43
Çizelge 5.1. MQTT konu örnekleri 58
Çizelge 5.2. MQTT bağlantı mesaj yapısı 60
Çizelge 5.3. MQTT bağlantı cevap mesajı yapısı 61
Çizelge 5.4. MQTT yayın mesajı yapısı 63
Çizelge 5.5. MQTT abone mesaj yapısı 65
Çizelge 5.6. MQTT abone cevap mesaj yapısı 66
Çizelge 5.7. MQTT abonelikten çıkma mesaj yapısı 67
Çizelge 5.8. MQTT abonelikten çıkma cevap mesaj yapısı 68
Çizelge 5.9. CoAP mesaj tipleri 75
Çizelge 5.10. CoAP yöntemleri yanıt kodları 79
Çizelge 5.11. CoAP CODE alanı değerleri 81
Çizelge 6.1. Bazı hafif sıklet şifreleme algoritmalarının temel yapıları 90
Çizelge 6.2. CLEFIA S_0 ve S_1 S-Kutuları 95
Çizelge 6.3. Sabit sayılar için gereken değerler 97
Çizelge 6.4. PRESENT S-Kutusu 101
Çizelge 6.5. PRESENT permütasyon tablosu 101
Çizelge 6.6. PICCOLO S-Kutusu 106
Çizelge 6.7. PRINCE S-Kutusu ve tersi. 110
Çizelge 6.8. PRINCE döngü sabitleri 111
Çizelge 6.9. LBLOCK S-Kutuları 114
Çizelge 7.1. İncelenen algoritmaların enerji tüketimleri 134
Çizelge 7.2. İncelenen algoritmaların güç tüketimleri 135
Çizelge 7.3. İncelenen algoritmaların akım ölçümleri 136
Çizelge 7.4. İncelenen algoritmalar için tahmini pil ömürleri 137

Çizelge 7.5. İncelenen algoritmalar için aktif mod oranları.....138
Çizelge 8.1. Test senaryoları.....144

ŞEKİLLER DİZİNİ

Şekil 1.1. 2020 yılındaki nesnelerin interneti projelerinin sektörel dağılımı 4
Şekil 2.1. Literatürdeki nesnelerin interneti mimarileri 9
Şekil 2.2. OSI, TCP/IP ve beş katmanlı mimarinin karşılaştırılması 15
Şekil 2.3. Algılama katmanında kullanılabilen teknolojiler. 16
Şekil 2.4. Ağ katmanında kullanılabilen teknolojiler. 17
Şekil 2.5. Uygulama katmanında kullanılabilen teknolojiler. 18
Şekil 3.1. IPv6 yapısı. 20
Şekil 3.2. IPv6'ya geçiş oranı. 21
Şekil 3.3. 802.15.4 ağ topolojileri. 22
Şekil 3.4. IEEE 802.15.4 protokol yığını. 23
Şekil 3.5. 6LoWPAN ağ yapısı 24
Şekil 3.6. OSI, TCP/IP ve 6LoWPAN protokol yığınının karşılaştırılması 25
Şekil 3.7. ZigBee topolojileri. 27
Şekil 3.8. OSI, TCP/IP ve ZigBee protokol yığınının karşılaştırılması. 28
Şekil 3.9. Bluetooth Piconet ve Scatternet ağları. 30
Şekil 3.10. OSI, TCP/IP ve Bluetooth LE protokol yığınının karşılaştırılması. 31
Şekil 3.11. LoRaWAN ağ yapısı. 33
Şekil 3.12. OSI, TCP/IP ve LoRaWAN protokol yığınının karşılaştırılması 33
Şekil 3.13. Z-Wave ağ yapısı 36
Şekil 3.14. OSI, TCP/IP ve Z-Wave protokol yığınının karşılaştırılması 36
Şekil 3.15. NFC protokol yığını 39
Şekil 3.16. Sigfox ağ yapısı. 40
Şekil 3.17. Sigfox katmanları 41
Şekil 4.1. Nesnelerin interneti mimarisi üzerindeki tehditler 51
Şekil 5.1. MQTT mimarisi 56
Şekil 5.2. Yayın-Abone yapısı 56

Şekil 5.3. Bir ulaşım hattı için hiyerarşik yapı 57
Şekil 5.4. MQTT Bağlantısı 59
Şekil 5.5. MQTT yayın işlemi 64
Şekil 5.6. MQTT abonelik işlemi 65
Şekil 5.7. MQTT abonelik iptali işlemi 67
Şekil 5.8. Örnek bir MQTT bağlantısı 68
Şekil 5.9. MQTT QoS 0 iletimi 69
Şekil 5.10. MQTT QoS 1 iletimi 70
Şekil 5.11. MQTT QoS 2 iletimi 71
Şekil 5.12. CoAP Mimarisi 74
Şekil 5.13. CoAP CON iletişimi 76
Şekil 5.14. CoAP NON iletişimi 77
Şekil 5.15. CoAP mesaj formatı 80
Şekil 5.16. CoAP gözlemci modu 82
Şekil 6.1. AES blok şifresindeki tek döngülük SPN mimarisi 87
Şekil 6.2. CLEFIA şifreleme döngü yapısı 93
Şekil 6.3. F_0 ve F_1 fonksiyonları 93
Şekil 6.4. DoubleSwap işlemi 96
Şekil 6.5. PRESENT şifreleme algoritması diyagramı 100
Şekil 6.6. PRESENT şifreleme döngü yapısı 100
Şekil 6.7. 80 bit anahtar planlama 102
Şekil 6.8.128 bit anahtar planlama 103
Şekil 6.9. PICCOLO şifreleme döngü yapısı 104
Şekil 6.10. F fonksiyonu 105
Şekil 6.11. RP permütasyon işlemi 107
Şekil 6.12. PRINCE anahtar beyazlatma işlemi 109
Şekil 6.13. PRINCE Şifreleme algoritması diyagramı 110
Şekil 6.14. LBLOCK şifreleme döngü yapısı 112
Şekil 6.15. LBLOCK F fonksiyonu 113
Şekil 7.1. MSP-EXP430FR5994 LaunchPad Development Kit 117
Şekil 7.2. CLEFIA 128 bit şifreleme güç tüketimi 118
Şekil 7.3. CLEFIA 128 bit şifre çözme güç tüketimi 119

Şekil 7.4. CLEFIA 128 bit şifreleme ve şifre çözme için AM-LPM geçişleri.....119
Şekil 7.5. CLEFIA 192 bit şifreleme güç tüketimi 120
Şekil 7.6. CLEFIA 192 bit şifre çözme güç tüketimi.....120
Şekil 7.7. CLEFIA 192 bit şifreleme ve şifre çözme için AM-LPM geçişleri.....121
Şekil 7.8. CLEFIA 256 bit şifreleme güç tüketimi 122
Şekil 7.9. CLEFIA 256 bit şifre çözme güç tüketimi.....122
Şekil 7.10. CLEFIA 256 bit şifreleme ve şifre çözme için AM-LPM geçişleri.....123
Şekil 7.11. PRESENT şifreleme güç tüketimi 124
Şekil 7.12. PRESENT şifre çözme güç tüketimi.....124
Şekil 7.13. PRESENT şifreleme ve şifre çözme için AM-LPM geçişleri 125
Şekil 7.14. PICCOLO 80 bit şifreleme güç tüketimi 126
Şekil 7.15. PICCOLO 80 bit şifre çözme güç tüketimi.....126
Şekil 7.16. PICCOLO 80 bit şifreleme ve şifre çözme için AM-LPM geçişleri 127
Şekil 7.17. PICCOLO 128 bit şifreleme güç tüketimi 128
Şekil 7.18. PICCOLO 128 bit şifre çözme güç tüketimi.....128
Şekil 7.19. PICCOLO 128 bit şifreleme ve şifre çözme için AM-LPM geçişleri 129
Şekil 7.20. PRINCE şifreleme güç tüketimi 130
Şekil 7.21. PRINCE şifre çözme güç tüketimi.....130
Şekil 7.22. PRINCE şifreleme ve şifre çözme için AM-LPM geçişleri.....131
Şekil 7.23. LBLOCK şifreleme güç tüketimi.....132
Şekil 7.24. LBLOCK şifre çözme güç tüketimi.....132
Şekil 7.25. LBLOCK şifreleme ve şifre çözme için AM-LPM geçişleri.....133
Şekil 8.1. MQTT test ortamı 140
Şekil 8.2. MQTT uçtan uca şifreleme 141
Şekil 8.3. MQTT istemci-broker şifreleme 142
Şekil 8.4. CoAP test ortamı.....143
Şekil 8.5. CoAP istemci-sunucu-istemci şifreleme.....144
Şekil 8.6. Wireshark ile yakalanan MQTT paketleri (*Test Senaryosu1*) 145
Şekil 8.7. Yakalanan MQTT yayın paketinin içeriği (*Test Senaryosu1*) 145
Şekil 8.8. Wireshark ile yakalanan CoAP paketi (*Test Senaryosu2*) 146
Şekil 8.9. Yakalanan CoAP paketinin içeriği (*Test Senaryosu2*) 146
Şekil 8.10. Wireshark ile yakalanan MQTT paketleri (*Test Senaryosu3*) 146

Şekil 8.11. Yakalanan MQTT yayın paketinin içeriği (*Test Senaryosu3*) 146
Şekil 8.12. Wireshark ile yakalanan CoAP paketi (*Test Senaryosu4*) 147
Şekil 8.13. Yakalanan CoAP paketinin içeriği (*Test Senaryosu4*) 147
Şekil 9.1. ARP zehirlenmesi için örnek senaryo 150
Şekil 9.2. ARP zehirlenmesi öncesi veri iletimi 150
Şekil 9.3. ARP zehirlenmesi sonrası veri iletimi 151
Şekil 9.4. WebSploit yazılımının çalıştırılması 152
Şekil 9.5. WebSploit içerisindeki saldırı araçları 153
Şekil 9.6. WebSploit içerisindeki arp_spoof aracı 153
Şekil 9.7. WebSploit ARP zehirlenmesi saldırısı için bilgilerin ayarlanması 154
Şekil 9.8. ARP zehirlenme saldırısının başlatılması 154
Şekil 9.9. Hedef bilgisayarın ARP önbellek tablosu 155
Şekil 9.10. ARP zehirlenmesi saldırısı esnasındaki ARP paketleri 156
Şekil 9.11. Tez çalışmasında MQTT iletişim için kullanılan ortam 156
Şekil 9.12. MQTT iletişimi için MITM saldırısı 157
Şekil 9.13. Saldırganın yakaladığı şifrelenmemiş MQTT paketleri 158
Şekil 9.14. Saldırganın yakaladığı şifrelenmiş MQTT paketleri 158
Şekil 9.15. CoAP iletişimi için kullanılan ortam 159
Şekil 9.16. CoAP iletişimi için MITM saldırısı 159
Şekil 9.17. Saldırganın yakaladığı şifrelenmemiş CoAP paketleri 160
Şekil 9.18. Saldırganın yakaladığı şifrelenmiş CoAP paketleri 161

ALGORİTMALAR DİZİNİ

Algoritma 6.1. $GFN_{4,r}$ işleminin algoritması 91
Algoritma 6.2. $GFN_{8,r}$ işleminin algoritması 92
Algoritma 6.3. F_0 fonksiyonun algoritması 94
Algoritma 6.4. F_1 fonksiyonun algoritması 94
Algoritma 6.5. $CON_i^{(k)}$ sabit değerlerinin hesaplanma algoritması 97
Algoritma 6.6. 128-bit anahtar için anahtar planlama aşaması 98
Algoritma 6.7. 192-bit ve 256-bit anahtar için anahtar planlama aşaması 99
Algoritma 6.8. 80 bitlik anahtar için güncelleme algoritması 102
Algoritma 6.9. 128 bitlik anahtar için güncelleme algoritması 103
Algoritma 6.10. PICCOLO G_r şifreleme algoritması 105
Algoritma 6.11. PICCOLO G_r^{-1} şifre çözme algoritması 107
Algoritma 6.12. PICCOLO KS_r^{80} anahtar planlama algoritması 108
Algoritma 6.13. PICCOLO KS_r^{128} anahtar planlama algoritması 109
Algoritma 6.14. LBLOCK şifreleme algoritması 112
Algoritma 6.15. LBLOCK şifre çözme algoritması 115
Algoritma 6.16. LBLOCK anahtar güncelleme algoritması 115

BÖLÜM 1

1. GİRİŞ

Başlangıçta sadece birkaç sistemin bağlı olması fikrinden yola çıkarak 1969'da oluşturulan ARPANET, bugün milyarlarca bilgisayarın ve sistemin bir araya geldiği uçsuz bucaksız ayrı bir dünya halini almıştır. İnternet hızı, kapasitesi ve trafiği üssel olarak artarak günümüze gelmiş ve geleceğe uzanmaktadır. Günümüzde neredeyse herkesin cebinde taşıdığı mobil cihazlar 20 yıl öncesinin süper bilgisayarlarından daha üstün yeteneklere sahip olmuşlardır. Artık insanoğlu mikrosensörler ve akıllı çipler sayesinde neredeyse tüm cihazları akıllı hale getirebilmektedir. Akıllı telefonlar, arabalar, ısıtma sistemleri rahatlıkla kontrol edilebilir ve programlanabilir hale gelmiştir.

1.1. Nesnelerin İnterneti

Son on yılda, nesnelerin interneti (IoT) kapsamında geliştirilen uygulamalar sensör ve iletişim teknolojilerinin de ilerlemesiyle sessiz ve kademeli olarak hayatımızda yer almaya başlamıştır. Küresel ölçüde, cihazlar ve bağlantılar olarak nitelendirilebilen nesnelerin sayısı insan nüfusundan daha hızlı artmaktadır. Dolayısıyla bu durum hane ve kişi başına düşen ortalama cihaz ve bağlantı sayısındaki artışı hızlandırmaktadır. Her yıl, artan yetenek ve zekaya sahip, farklı formlarda yeni cihazlar piyasaya sürülmekte ve benimsenmektedir. Akıllı sayaçlar, sağlık bakımı izleme, taşıma ve paketleme veya varlık izleme gibi artan sayıda M2M (Makineden Makineye) uygulaması, cihazların ve bağlantıların büyümesine önemli bir katkıda bulunmaktadır.

Bu bağlamda nesnelerin interneti kavramı hakkında literatürde kullanılan çeşitli tanımlara aşağıda yer verilmiştir:

- Çeşitli haberleşme protokolleri sayesinde birbirlerine bağlanarak bilgi paylaşan ve ağ oluşturmuş cihazlar sistemidir (Yetimler, 2017).
- Benzersiz bir şekilde tanımlanabilir algılama ve programlama yeteneklerine sahip nesneleri internete bağlayan bir ağdır (Minerva, Biru, & Rotondi, 2015).
- Birbirleri ile entegre olmuş farklı teknolojilerin sosyal hayatta kullanıldığı uygulama alanıdır (Büyüktanır & Özer, 2017).
- Bilgi ve iletişim teknolojilerini kullanarak bir yerleşim yeri ile ilgili kritik altyapı ve eğitim, sağlık, güvenlik, ulaşım gibi hizmetlerin daha etkili ve etkileşimli kullanılabilmesine olanak sağlayan sistemdir (Bélissent, 2010).
- Çeşitli iletişim protokollerini kullanarak birbirleri ile haberleşen, bilgi üreten, oluşturdukları ağ sayesinde çevresiyle bilgi alış verişi yapabilen akıllı cihazların oluşturduğu bir topluluk ve pazardır (Erdem, 2015).

Nesnelerin interneti kavramından ilk defa Procter & Gamble şirketi için 1999'da Kevin Ashton tarafından hazırlanan bir sunumda bahsedilmiştir. Aslında Ashton sunumunda RFID teknolojisinin kullanımı ile şirketin sağlayacağı faydaları sıralamaktaydı. Fakat Ashton, belki tüm cihazların birbirlerine bağlanma fikrini ortaya atarak günümüzde dikkatleri üstüne çeken nesnelerin interneti kavramına ve bu doğrultuda birçok ürüne de önderlik etmiş oldu.

Kevin Ashton'dan önce 1992 yılında Cambridge Üniversitesindeki Quentin Stafford-Fraser ve Paul Jardetzky adındaki akademisyenler ofislerinin bulunduğu binada başka akademisyenler ile paylaştıkları kahve makinesinin takibi için makinenin dakikada üç adet görselini yakalayan ve bilgisayara aktaran xcoffee adında bir sistem tasarladılar. Bu basit sistem sayesinde kahve makinesinde bulunan kahvenin miktarı rahatlıkla takip edilebilmekteydi. Basit bir kamera, görüntü yakalama ve aktarma için geliştirilen bir yazılımdan oluşan bu sistem her ne kadar basit gibi görünse de nesnelerin birbirleri ile bağlanabileceği ve iletişime girilebileceğini göstermesi açısından nesnelerin interneti ekolünün ilk ürünü olarak kabul edilmektedir (Presser, 2016).

2005 yılında yayınlanan ITU (Uluslararası Telekomünikasyon Birliği) raporunda ilk defa nesnelerin interneti kavramından bahsedilmiştir. Bu raporda nesnelerin interneti

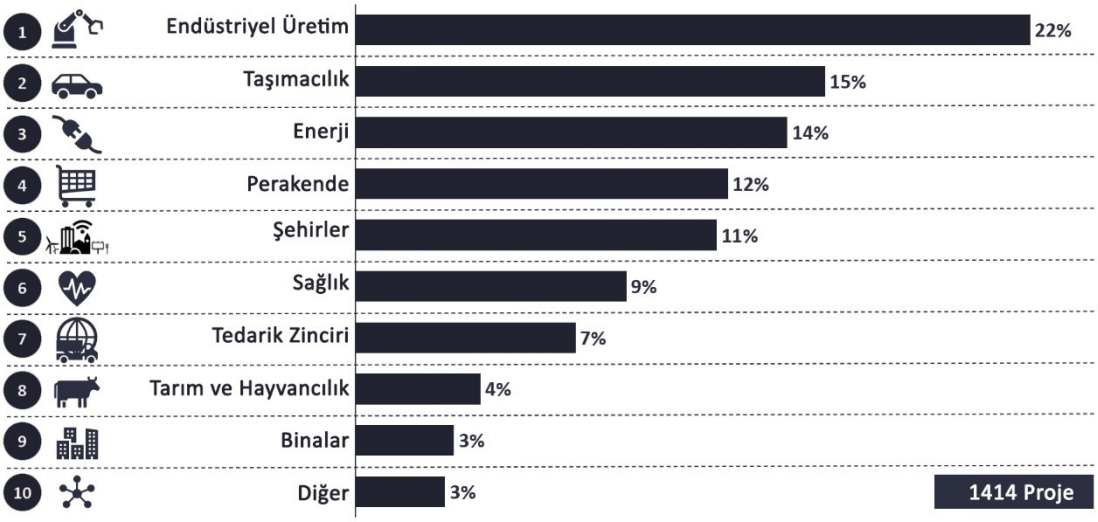
ile ilgili temel kavramlara, RFID uygulamalarına, sensör teknolojilerine, sektördeki fırsatlara, güvenlik ve mahremiyet kavramlarına değinilmiştir (EU-IoT, 2009).

Avrupa birliği komisyonu tarafından "Nesnelerin İnterneti – Avrupa için Eylem Planı" başlıklı rapor 2009 yılında yayınlanmıştır (EU-IoT, 2009). Benzer şekilde 2013 yılında yayınlanan Avrupa birliği raporunda, 2012 yılında yapılan kamuoyu araştırması ile nesnelerin interneti teknolojilerinin bireylerin hayatlarını sağlık, sosyal, ulaşım, çevre ve enerji gibi alanlarda kolaylaştıracağını gösteren bulguların elde edildiği vurgulanmıştır (EC, 2013).

İnternet dünyasında her geçen gün internete bağlı cihaz sayısı ve buna bağlı olarak da internet trafiği artmaktadır. Cisco'nun raporuna göre 2017 yılındaki trafik 100 EB iken 2020 yılında 212 EB olmuştur ve 2022'de 333 EB olacağı tahmin edilmektedir. 2023 yılında nesnelerin interneti uygulamalarında kullanılan cihazların tüm cihazlara oranının % 50'ye yaklaşacağı ve böylelikle 2018'de ağa bağlı cihaz sayısının 18.4 milyardan 29.3 milyara yükseleceği öngörülmektedir (Cisco, 2018).

1.2. Nesnelerin İnterneti Uygulama Alanları

Nesnelerin interneti uygulamaları hayatın tüm alanına yayılmaya devam etmektedir. 2020 yılında pazara sunulan 1414 nesnelerin interneti projesinin sektörlere dağılımı Şekil 1.1'de verilmiştir. Buna göre özellikle endüstriyel üretim, taşımacılık, enerji, perakende ve şehir uygulamaları yeni ürünlerin büyük çoğunluğunu oluşturmaktadır (Scully, 2020).



Şekil 1.1. 2020 yılındaki nesnelerin interneti projelerinin sektörel dağılımı

Üretim konusunda nesnelerin interneti uygulamalarının önderliğinde ciddi bir dönüşüm söz konusudur. Bu konuyu Microsoft yöneticilerinden Satya Nadella "Endüstriyel nesnelerin interneti uygulamaları, üretim kurallarını dönüştürüyor, bulut ve diğer yenilikleri besliyor, dijital fabrikaların gelişimini hızlandırıyor ve operasyonel performansı artırıyor." şeklinde değerlendirmektedir (Forbes, 2018). Benzer bir şekilde sektörün önemli temsilcilerinden Rockwell Otomasyon yöneticisi Blake Moret bu konuda "Üreticiler ve endüstriyel aktörler, nesnelerin interneti teknolojilerini operasyonlarına uygulamak için pratik yollar keşfediyorlar ve bunun sonucunda ölçülebilir iş değeri elde ediyorlar. Nesnelerin interneti teknolojisini özellikle belirli endüstriyel uygulamalar ile birleştirmek daha iyi işbirliği, daha hızlı problem çözme ve daha fazla üretkenlik sağlar." ifadelerini kullanmıştır (Business Wire, 2017).

Taşımacılık sektöründe özellikle nesnelerin interneti teknolojileri ile donatılmış otonom araçların öne çıktığı söylenebilir. Ayrıca bu konu içerisindeki tipik uygulamalar, akü izleme, lastik basıncı izleme, sürücü izleme ve filo yönetimi gibi çözümlerdir. Volvo yöneticilerinden Martin Lundstedt "Nesnelerin interneti uygulamaları ile sunulan bağlantılı çözümler sürücüler, operatörler ve diğer kullanıcılar için daha iyi güvenlik ve daha az karbondioksit emisyonu getiriyor. Volvo'daki ilk milyon bağlantılı varlık sadece başlangıç, bu alanda lider olmaya kararlıyız." ifadesi ile taşımacılık sektöründeki

geleceğinin nesnelerin interneti uygulamalarından geçtiğini belirtmiştir (TU Automotive, 2019).

Enerji alanındaki projeler akıllı enerji dağıtımı, şebeke optimizasyonu, uzaktan varlık izleme ve yönetimi, bakım tahmini ve müşteriler için daha fazla şeffaflık yaratmaya odaklanmıştır. Bu konuda Shell yöneticilerinden Jay Crotts "Nesnelerin interneti uygulamaları aracılığıyla, Shell'in operasyonlarında daha fazla ekonomik değer yaratmak için gelişmiş analiz yeteneklerimizin verimliliğini ve kapsamını önemli ölçüde artırmak istiyoruz. Nesnelerin interneti ve yapay zeka tabanlı uygulamaları birleşerek yatırımlarımızı optimize etmemizi sağlıyor, böylece müşterilerimize daha hızlı ve verimli hizmet verebiliyoruz." demektedir (Business, 2018).

Perakendecilik sektöründeki nesnelerin interneti uygulamaları çözümleri, mağaza içi dijital tabelalar, müşteri takibi ve geri bildirimleri, ürün izleme ve envanter yönetimi, akıllı otomatlar, akıllı raflar ve kasiyersiz marketleri içermektedir. Dünyaca ünlü Walmart market zincirinin yöneticilerinden Doug McMillon "Veri toplama ve daha etkin bir şekilde kullanma potansiyeli heyecan vericidir. Nesnelerin interneti uygulamalarının farklı şekilde çalışmamıza nasıl yardımcı olabileceğini öğreniyoruz. Süreçlerimizin çoğunu iyileştiriyoruz ve çalışma arkadaşlarımızı daha iyi araç ve teknoloji ile güçlendiriyoruz." demektedir (Diginomica, 2018).

Akıllı şehirler, milyonlarca insanın yaşamını doğrudan etkilediği için nesnelerin interneti uygulamaları için heyecan verici alanlardan biridir. Bu alandaki uygulamalar akıllı park, trafik yönetimi, akıllı atık toplama sistemleri, akıllı aydınlatma, kamu güvenliği sistemleri ve hava kirliliği takip sistemleri gibidir. İnsanların yaşadıkları şehirlerin ekonomik ve teknolojik yönlerini insancıl boyutlarla dengelemek algılarına odaklanan IMD Akıllı Şehir Endeksi 2019 raporunda akıllı şehirler listelenmektedir. Buna göre listenin ilk 10'u sırasıyla Singapur, Zürih, Oslo Cenevre, Kopenhag, Auckland, Taipei, Helsinki, Bilbao ve Düsseldorf şehirleridir (Bris, 2019).

Sağlık alanında nesnelerin interneti uygulamaları son zamanlarda yaygınlaşmaya başlamıştır. Bu alanda uzaktan sağlık takibi, dijital teşhis, uzaktan izleme ve robot yardımı gibi belirli sağlık uygulamalarına olan talep artmaktadır. Ayrıca spor ve kişisel bakım gibi bazı alanlarda da sağlık ürünü olarak nitelendirilebilecek son kullanıcıya yönelik ürünler ortaya çıkmaktadır. Genellikle bu ürünler hayat kalitesini artırmayı

amaçlamaktadır. Sağlık konusunda yeni yönelimlerden biri de COVID-19 pandemisi sonucunda nesnelerin interneti uygulamaları ile salgın süreçlerinin takibidir. Bu konuda General Electric Sağlık şirketinin yöneticisi Amit Phadnis "Mevcut salgının üstesinden gelmek ve uzaktan sağlık yoluyla sağlık çözümlerinin iyileştirilmesine yardımcı olmak için teknoloji, veri ve iletişim yöntemlerinin büyük bir uygulanabilirliği vardır. Dünya çapında talep ortaya çıkıyor, küresel olarak uzaktan tedavi gören hasta sayısı giderek artıyor, örneğin, birçok sağlık sisteminde hâlihazırda dünya çapında çevrimiçi danışma sayısı 50 ila yüz kat arttı. Bu çözümler mevcut krizden sonra bile kalacaklar " diyerek COVID-19 sürecine nesnelerin interneti uygulamalarının katkısını özetlemiştir (World Medical Innovation Forum, 2020).

Tedarik zincirlerinde lojistik sağlayıcılar karmaşıklığın üstesinden gelmek için birbirleri ile bağlantılı dijital çözümleri giderek daha fazla kullanmaktadır. Nesnelerin interneti uygulamaların ilk örnekleri RFID gibi sistemler ile ürün takibi olarak geliştirilmeye başlandığından tedarik zinciri ve lojistik sektörü nesnelerin interneti uygulamaları açısından günümüzde oldukça zengindir. Bu alandaki projeler arasında varlık izleme, soğuk zincir takibi gibi durum izleme, envanter ve depolama yönetimi, otonom araçlar gibi konular bulunur.

Dünya nüfusu arttıkça insanlığın daha fazla yiyeceğe ihtiyacı olacağı düşünüldüğünde akıllı tarım teknolojilerinin geliştirilmesi oldukça önemlidir. Nesnelerin interneti uygulamaları, gübre ve böcek ilacı kullanımını azaltarak daha yüksek verim ile daha kaliteli ve tasarruflu üretim sağlamak için çiftçilerin kararlar almalarına yardımcı olabilir. Aynı senaryo hayvancılıkta da düşünülebilir. Dünyaca ünlü John Deere firmasının yöneticilerinden John C. May konunun önemini "Nesnelerin interneti uygulamalarının kullanılmasının akıllı, gelişen ve daha verimli bir çiftlik için çiftçiliğin geleceği açısından temel oluşturduğuna inanıyoruz." şeklinde ifade etmektedir (I-CIO, 2018). Akıllı tarım projeleri arasında hassas tarım, hayvancılık izleme, sulama yönetimi, akıllı haritalama, akıllı ilaçlama sistemleri, insansız hava araçları kullanımı yer almaktadır.

Akıllı binalar ve ev otomasyonları nesnelerin interneti uygulamalarının bir süredir mevcut olan ve sıklıkla kullanıldığı alanlardan biridir. Tipik bina veya ev projeleri arasında akıllı aydınlatmalar, asansör takibi, akıllı duman ve yangın söndürücüler, bina

güvenliği için tesis otomasyonu ve bina izleme sistemleri yer almaktadır. Özellikle Google, Amazon, Microsoft gibi büyük firmaların tüketicilere yönelik çıkartılan akıllı ev kontrol sistemleri piyasanın önemli ürünleri arasındadır.

Bu kadar büyük ve kontrol edilemez ekosistem içinde nesnelerin interneti uygulamaları çeşitli yeni sorunlar ortaya çıkarmaktadır. Yeni bilişim dünyasının başa çıkmak zorunda olduğu bu sorunlar aşağıda listelenmiş ve kısaca açıklanmıştır (Kim & Solomon, 2016).

- Güvenlik sorunları: İnsanların, işletmelerin ve ülkelerin nesnelerin interneti uygulamalarına bağlılıkları arttıkça bilgisayar korsanlarının ve kötü niyetli kişilerin de verilere erişme ve çalma isteği artmaktadır. Bu nedenle, güvenlik nesnelerin interneti teknolojilerinin üstesinden gelmesi gereken en büyük sorundur.
- Mahremiyet sorunları: Nesnelerin interneti uygulamalarının çoğu insanların günlük yaşamlarını kolaylaştırmak için bilgiler toplamakta ve işlemektedir. Bu verilerin çoğu kişisel veri olarak nitelendirilebileceği için mahremiyet sorunları ortaya çıkmaktadır. Bu gibi sorular, özellikle yasal bakış açısıyla dikkatli bir analiz ve risk azaltıcı çözümler gerektirmektedir.
- Birlikte çalışabilirlik ve standartlara bağlı sorunlar: Her ne kadar internet TCP/IP altyapısı ve sunucu/istemci mimarisi ile çalışsa da nesnelerin interneti uygulamalarında düşük işlem kapasitelerine sahip nesnelerin daha iyi iletişim kurabilmeleri ve veri transferlerinin etkili bir şekilde çalışabilmesi için birçok standart olmayan protokoller geliştirilmiştir. Bu çeşitlilik birlikte çalışabilirlik sorunlarını ortaya çıkarmaktadır.
- Yasal sorunlar: Nesnelerin interneti uygulamalarının topladığı verilerin sahipliği ile ilgili yaşanan sıkıntıların çözümü için yasal mevzuatlar yeni yeni gündeme gelmektedir. Bu konu ile ilgili ülkelerin ulusal düzeyde yaptıkları çalışmalar küresel nesnelerin interneti uygulamaları için yetersiz kalabilmektedir.
- Ekonomik kalkınma sorunları: Nesnelerin interneti uygulamaları ve geliştirilen teknolojiler ekonomiyi ciddi oranda değiştirmektedir. Karanlık fabrikalar, insansız ulaşım ve taşıma araçlarının ciddi kalkınma sorunlarına neden olabileceği düşünülmektedir. Bu teknolojiler insan gücüne dayanan işin

azalmasını yani işsizliğin artmasını veya yeni iş alanlarının açılmasını mümkün kılmaktadır. Gelişmiş ülkelerin bu sorunları kısmen aşabileceği tartışılmakta fakat gelişmekte olan veya geri kalmış ülkelerin ciddi krizler yaşayacağı öngörülmektedir.

1.3. Tezin Kapsamı

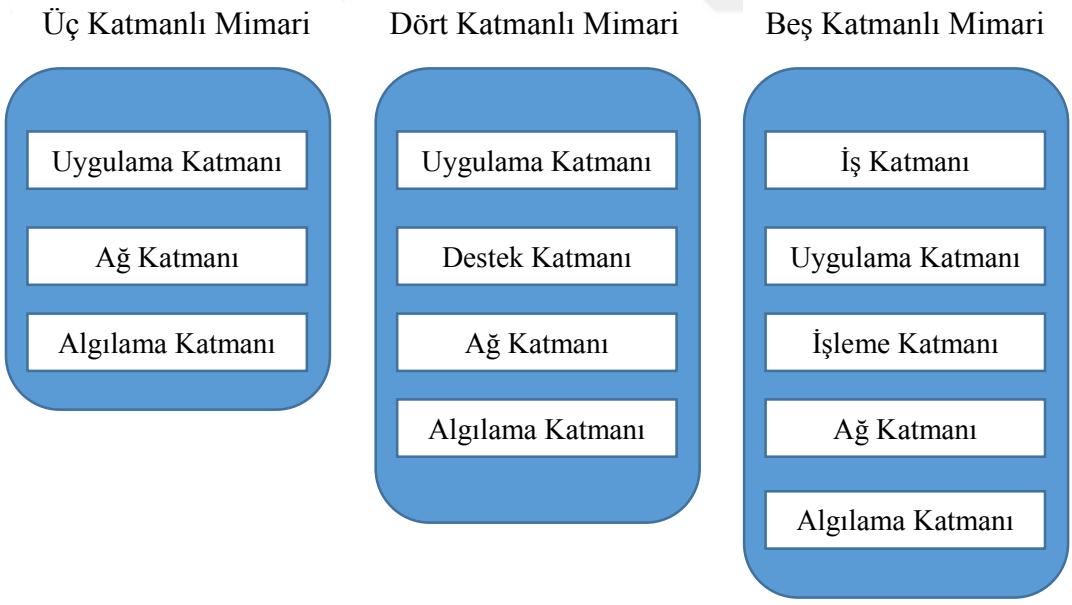
Tez kapsamında nesnelerin interneti mimarisi ve mimaride kullanılan iletişim protokolleri detaylı olarak incelenmiştir. Nesnelerin interneti cihazlarının hedef olarak seçilme nedenleri araştırılmış ve günümüze kadar meydana gelen önemli siber güvenlik saldırıları kronolojik olarak verilmiştir. Nesnelerin interneti uygulamalarındaki güvenliği artırabilmek için kullanılabilecek hafif sıklet şifreleme algoritmaları detaylı olarak incelenmiştir. Bu algoritmalar yazılımsal ve donanımsal olarak gerçeklenerek performans analizleri yapılmıştır. İncelenen algoritmalar kurulan test ortamında güvenli iletişim için test edilmiştir. Ayrıca kurulan test ortamına ortadaki adam saldırısı düzenlenerek sistemin güvenliği test edilmiştir. Performans ve test sonuçlarının ışığında nesnelerin interneti uygulamalarının güvenliğini kriptografik olarak artırmak için çeşitli öneriler sunulmuştur.

BÖLÜM 2

2. NESNELERİN İNTERNETİ MİMARİSİ

2.1. Giriş

Literatür incelendiğinde nesnelerin interneti konusu ile ilgili TCP/IP veya OSI referans modeli gibi evrensel olarak kabul görmüş bir mimari bulunmamaktadır. Farklı araştırmacılar farklı mimari modelleri önermiştir (Sethi & Sarangi, 2017). Buna göre önerilen mimari modeller Şekil 2.1'de verilmiştir.



Şekil 2.1. Literatürdeki nesnelerin interneti mimarileri

Şekil 2.1 incelendiğinde tüm yaklaşımların en alt katmanında algılama ve ağ katmanı ve üst katmanlarında ise uygulama katmanı olduğu görülebilir. Bu katmanların genel olarak özellikleri ve görevleri benzerdir. Dört katmanlı mimari yapısında uygulama ve ağ katmanı arasında destek katmanı önerilmiştir. Beş katmanlı mimari yapıda ise ağ ve uygulama katmanı arasına işleme katmanı ve üst katman olarak iş katmanı eklenmiştir. Tezin bu bölümünde nesnelerin interneti mimarileri ve bu mimarilerin içyapıları anlatılmıştır.

2.2. Üç Katmanlı Mimari

Üç Katman Mimarisi, temel nesnelerin interneti çözümlerinin geliştirilebileceği basit bir modeldir. Veri toplamak, iletmek ve işlemek için algılama, ağ ve uygulama katmanlarına sahiptir. Bununla birlikte, bu basit model sağlık gibi bazı sektörlerde gerekli olan güvenilirlik ve güvenliği sağlama yeteneğinden yoksundur. Bu mimari özellikle nesnelerin interneti araştırmanın erken aşamalarında ortaya çıkarılmıştır. Algılama, ağ ve uygulama katmanları olmak üzere üç katmanı vardır (M. Wu, Lu, Ling, Sun, & Du, 2010).

2.2.1. Algılama Katmanı

Algılama katmanı, çevre hakkında bilgileri algılamak ve toplamak için sensörlere sahip olan fiziksel katmandır. Bu sebeple bazı kaynaklarda aygıt katmanı olarak da nitelendirilmektedir. Ana görevi nesneyi tanımlamak ve bilgi toplamaktır. Algılama Katmanı, 2 boyutlu barkod etiketleri, karekodlar ve bunların okuyucuları, RFID etiketleri ve okuyucu/yazıcıları, kamera, GPS, sensörler, terminaller ve sensör ağlarını içerir. Sensör tipine ve uygulamaya göre toplanan veriler nesne bilgisi, konum, nem, sıcaklık, hareket, ivme, titreşim, yön, havadaki değişimler vb. olabilir. Toplanan bilgiler daha sonra işlenmek için ağ katmanına aktarılır. Algılama katmanı Şekil 2.1'de verilen 3 model içinde temelde aynı işlevi yapmaktadır.

2.2.2. Ağ Katmanı

Ağ katmanının ana işlevi algılama katmanından elde edilen bilgileri iletmek ve işlemektir. Bu sebeple iletim katmanı olarak da adlandırılır. Bu katmanda bilgi iletimi internet, mobil iletişim ağı, uydu ağları, kablosuz ağ gibi bir altyapı kullanılarak ilgili protokoller tarafından gerçekleştirilir. Ayrıca bu katman diğer nesnelere, ağ cihazlarına

ve sunuculara bağlanmaktan sorumludur. Veri aktarımı yapılırken veri üzerinde işlemler de yapılabilir. İletim ortamı kablolu veya kablosuz olabilir. Ayrıca sensör yapılarına bağlı olarak 4G, UMTS, Wi-Fi, Bluetooth, Kızılötesi, ZigBee vb. iletişim ortamları kullanılabilir. Ağ katmanı, mesajı paketlere bölmek ve IPv4/IPv6 adresleme mekanizmalarını kullanarak paketleri kaynaktan hedefe yönlendirmek için kullanılır. Ağ katmanı Şekil 2.1'de verilen 3 model içinde temelde aynı işlevi yapmaktadır.

2.2.3. Uygulama Katmanı

Uygulama katmanı, nesnelerin interneti uygulamasına özel hizmetlerin verilmesinden sorumludur. Her uygulamaya göre farklı protokollerin kullanımı gerekebilir. Bu sebeple geliştirilen ve standart olmayan protokoller, uygulama katmanı üzerinde uygulamaya özel olarak çalışır.

2.3. Dört Katmanlı Mimari

Bu mimari yaklaşımındaki katmanlar algılama, ağ, destek ve uygulama katmanlarıdır. Üç katmanlı mimariye göre ağ ve uygulama katmanı arasına bulut bilişim teknolojilerine olanak sağlamak amacıyla destek katmanı eklenmiştir. Ayrıca bu katman ile verilerin işlenebilmesi ve kısmen güvenliğin sağlanabilmesi amaçlanmıştır.

2.3.1. Algılama Katmanı

Algılama katmanı, bir önceki mimaride olduğu gibi fiziksel nesneler ve sensör cihazlarından oluşur. Algılama katmanı Şekil 2.1'de verilen 3 model için de temelde aynı işlevi yapmaktadır.

2.3.2. Ağ Katmanı

Ağ katmanının ana işlevi algılama katmanından elde edilen bilgileri iletmek ve işlemektir. Bu katman Şekil 2.1'de verilen 3 model için de temelde aynı işlevi yapmaktadır.

2.3.3. Destek Katmanı

Destek katmanı, uygulama ve ağ katmanı arasında daha sağlıklı ve güvenilir veri

iletimi için kullanılır. Bu aşamada nesneler genellikle bulut ve kenar bilişim teknolojileri ile desteklenir. İletilecek olan veri işlenerek alt veya üst katmana gönderilir. Nesnelerin interneti uygulamalarındaki cihazlar iletişim boyunca UDP (Kullanıcı Datagram Protokolü) kullanır. UDP, TCP'ye göre daha hızlı fakat güvenilir olmayan bir protokoldür. Destek katmanında genellikle DTLS (Datagram Aktarım Katmanı Güvenliği) kullanan bir güvenlik mekanizması bulunur.

2.3.4. Uygulama Katmanı

Uygulama katmanı, nesnelerin interneti uygulamasına özel hizmetlerin verilmesinden sorumludur. Her uygulamaya göre farklı protokollerin kullanımı gerekebilir. Bu sebeple geliştirilen ve standart olmayan bu protokolleri uygulama katmanı üzerinde uygulamaya özel olarak çalışır.

2.4. Beş Katmanlı Mimari

Bu mimari yaklaşımındaki katmanlar algılama, ağ, işleme, uygulama ve iş katmanlarıdır. Bu yaklaşımda ağ ve uygulama katmanları arasında eklenen işleme katmanı ile sistemin veri işleme yükü uygulama katmanından işleme katmanına aktarılmıştır. Bu şekilde verinin ağ üzerinde iletiminden önce veya sonra işleme alınması sağlanabilir. Tezin sonraki bölümlerinde önerilmiş olan güvenlik mekanizmalarının nesnelerin interneti mimarisi içindeki işleme katmanına eklenmiştir.

Beş katmanlı mimarinin en üst katmanına iş katmanı eklenmiştir. Bu katman ile nesnelerin interneti uygulamasının ölçeklenebilirliği, birlikte çalışabilirliği, veri akış kontrolü ve verilerin analiz edilerek anlamlar çıkarmak gibi işlemler yapılabilir. Bu şekilde nesnelerin interneti cihazlarının ürettiği büyük veri yönetilebilir.

2.4.1. Algılama Katmanı

Algılama katmanı, bir önceki mimaride olduğu gibi fiziksel nesneler ve sensör cihazlarından oluşur. Algılama katmanı Şekil 2.1'de verilen 3 model için de temelde aynı işlevi yapmaktadır.

2.4.2. Ağ Katmanı

Ağ katmanının ana işlevi algılama katmanından elde edilen bilgileri iletmek ve işlemektir. Bu katman Şekil 2.1'de verilen 3 model için de temelde aynı işlevi yapmaktadır.

2.4.3. İşleme Katmanı

İşleme katmanı aynı zamanda ara katman olarak da bilinir. Nesnelerin interneti uygulamalarında kullanılan cihazlar farklı hizmet türleri için çalışırlar. Bazı durumlarda cihazlar yalnızca aynı hizmet türünü sunan diğer cihazlar ile bağlantı kurar ve iletişim sağlarlar. İşleme katmanında geliştirilen teknolojiler ile taşıma katmanından gelen büyük miktarda veri depolanabilir ve/veya analiz edilebilir. Ayrıca çeşitli hizmetler bu katmanda yönetebilir ve sağlayabilir. Veritabanları, bulut bilişim ve büyük veri işleme modülleri gibi birçok teknoloji bu katmanda kullanılabilir. Bu katmanın amacı üst katmanlara anlamlı veriler göndermek ve üst katmanlardan gelen verileri alt katmanlara gönderebilecek halde düzenlemek olarak özetlenebilir.

2.4.4. Uygulama Katmanı

Uygulama katmanı, nesnelerin interneti uygulamasına özel hizmetlerin verilmesinden sorumludur. Her uygulamaya göre farklı protokollerin kullanımı gerekebilir. Bu sebeple geliştirilen ve standart olmayan bu protokolleri uygulama katmanı üzerinde uygulamaya özel olarak çalışır.

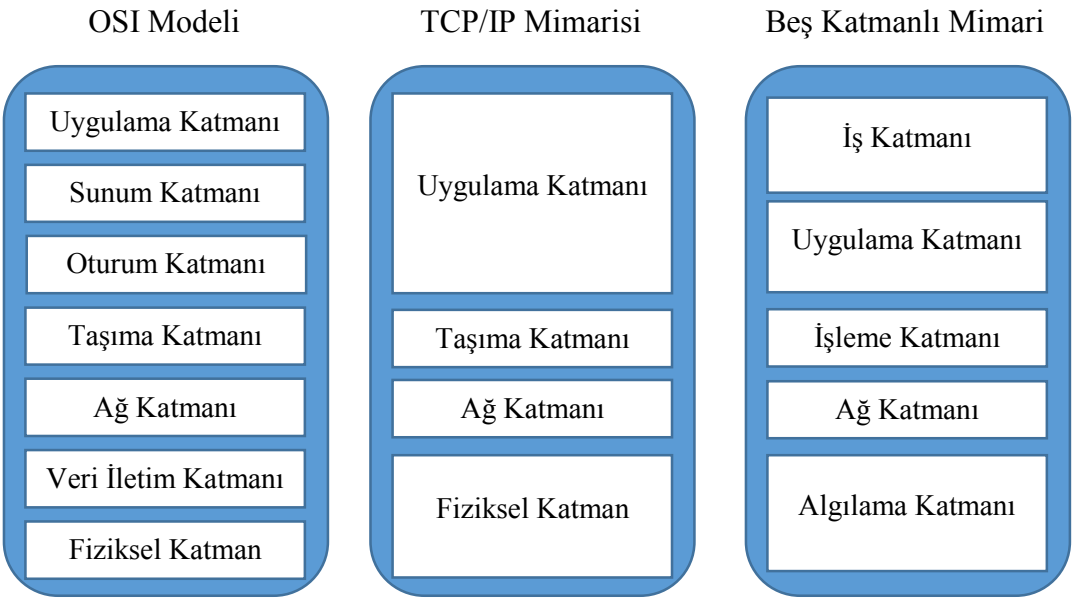
2.4.5. İş Katmanı

Bu katman genel olarak nesnelerin interneti sisteminin yönetiminden sorumludur. Alt katmanlardan alınan verilere göre planlamaların yapılarak çeşitli iş modelleri, grafikler, akış şemaları vb. doğrudan uygulamanın yönetimine yönelik tanımlar burada yapılır. Nesnelerin interneti teknolojisinin gerçek başarısı, protokoller ve teknolojilerin yanında iyi iş modellerine de bağlıdır. Bu katmanda kurulacak sistemler ile veriler ve sonuçlar analiz edilerek gelecekteki eylemlerin ve iş stratejilerinin belirlenmesi sağlanabilir.

2.5. Nesnelerin İnterneti Mimarisinin OSI ve TCP/IP ile Karşılaştırılması

OSI referans modeli 1970'lerin sonlarında çıkmış ve 1984'te kabul görerek standart olarak kullanılmaya başlanmıştır. TCP/IP mimarisi ise 1973 yılında geliştirilmiş 1983'te ARPANET'in standart protokolü olarak belirlenmiştir. Bu iki mimari günümüzde hala internetin temel protokolü olarak görülmektedir. Nesnelerin interneti uygulamalarının yayılması ile internete ya da ağa bağlanan cihaz sayısının artması ile bu alanda mimari arayışları devam etmekte ve bir standart yakalanamamış durumdadır. Özellikle nesnelerin interneti ekosisteminin büyümesi, üretilen verilerin büyük veri olarak düşünülmesi ile beş katmanlı mimari kabul görmüştür. Şekil 2.2'de OSI modeli, TCP/IP mimarisi ve beş katmanlı mimarinin karşılaştırılması verilmiştir. Buna göre aşağıdaki ifadeler söylenebilir:

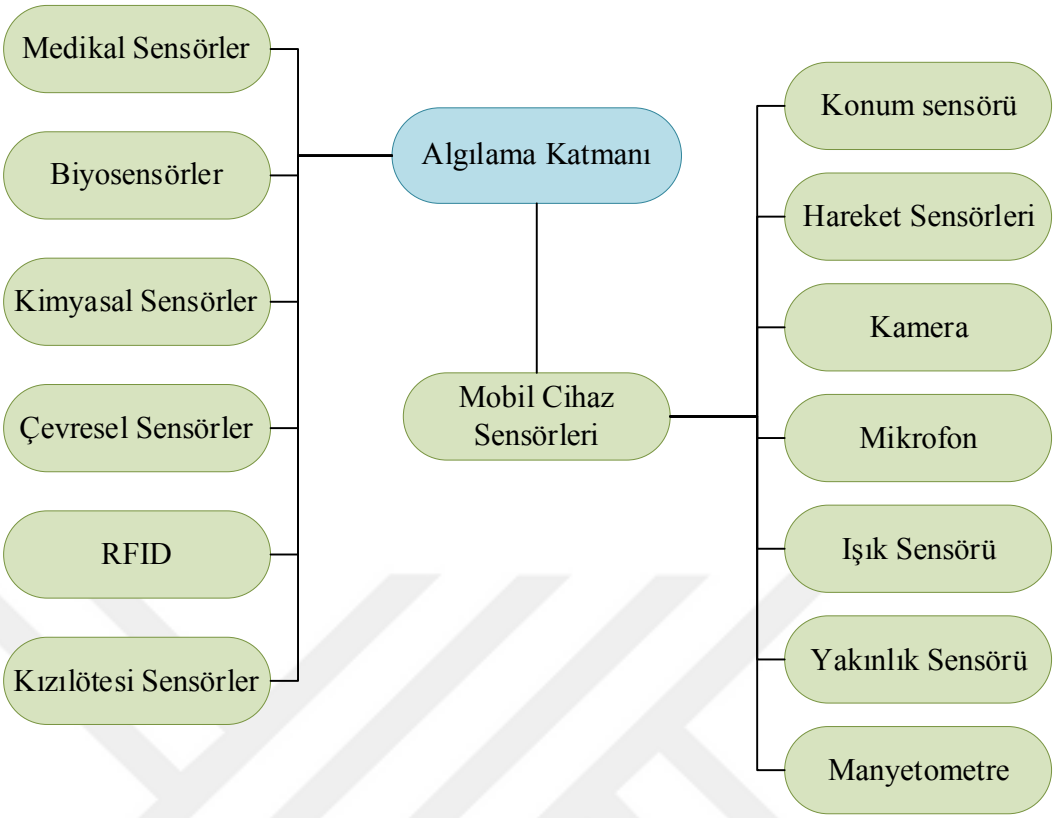
- TCP/IP'deki fiziksel katman nesnelerin interneti mimarisinde benzer görevleri üstlenen algılama katmanı ile eşleşmektedir,
- Ağ katmanı genel özellikleri açısından tüm modellerde benzer görevleri gerçekleştirmektedir,
- Her ne kadar taşıma katmanı işleme katmanı ile eşleşse de bazı ayrılıklar söz konusudur. Buradaki en önemli ayrım nesnelerin interneti uygulamalarında özellikle veri iletiminin erken safhasında verinin işlenebilmesi ve bu şekilde iletilmesidir,
- TCP/IP'de olduğu gibi 5 katmanlı nesnelerin interneti mimarisinde de TCP ve UDP paketlerinin güvenliği DTLS ve TLS (Aktarım Katmanı Güvenliği) protokolleri ile sağlanmaktadır,
- TCP/IP'nin üst katmanında yer alan uygulama katmanına ek olarak iş katmanı getirilmiştir.



Şekil 2.2. OSI, TCP/IP ve beş katmanlı mimarinin karşılaştırılması

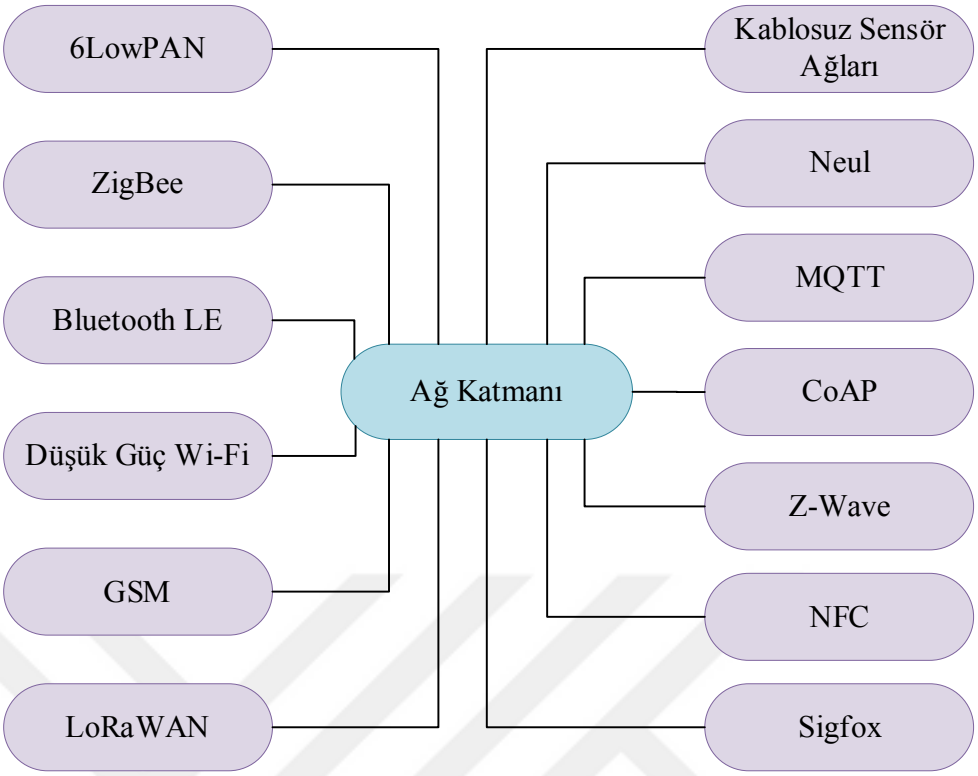
2.6. Katmanlarda Kullanılan Teknolojiler

Nesnelerin interneti mimarileri konusunda her ne kadar standart olmasa da geliştirilecek uygulamaya göre her katmanda çeşitli teknolojiler ve protokoller kullanılmaktadır. Genellikle tüm mimari önerilerinde algılama katmanı donanıma yakın olan kısımdır ve sensörlerden oluşur. Burada kullanılabilecek sensör tipleri Şekil 2.3'te verilmiştir ve her geçen gün artmaktadır. Medikal sensörler, nabız ve kalp atış sensörü, EKG/EEG sensörü, solunum hızı ölçüm sensörü, diyaliz makinelerinde kullanılan kuvvet sensörü, kandaki toplam hemoglobin sayısına göre oksijenle doymuş hemoglobin fraksiyonunu ölçen oksimetre sensörü gibi yüzlerce farklı olanaklara sahiptir. Biyosensörler ve kimyasal sensörler pH sensörü, glikoz sensörü gibi birçok farklı ölçümü yapabilecek şekilde planlanmıştır ve özellikle kimya ve biyoloji alanında kullanılmaktadır. Çevresel sensörler, ses, ivme, kuvvet, ışık, basınç, nem, gaz ölçümleri için sıklıkla kullanılır. Ayrıca mobil cihazlarda bulunan kamera, mikrofon ve ışık yakınlık, hareket gibi sensörler de bu katmana dahildir.



Şekil 2.3. Algılama katmanında kullanılabilen teknolojiler.

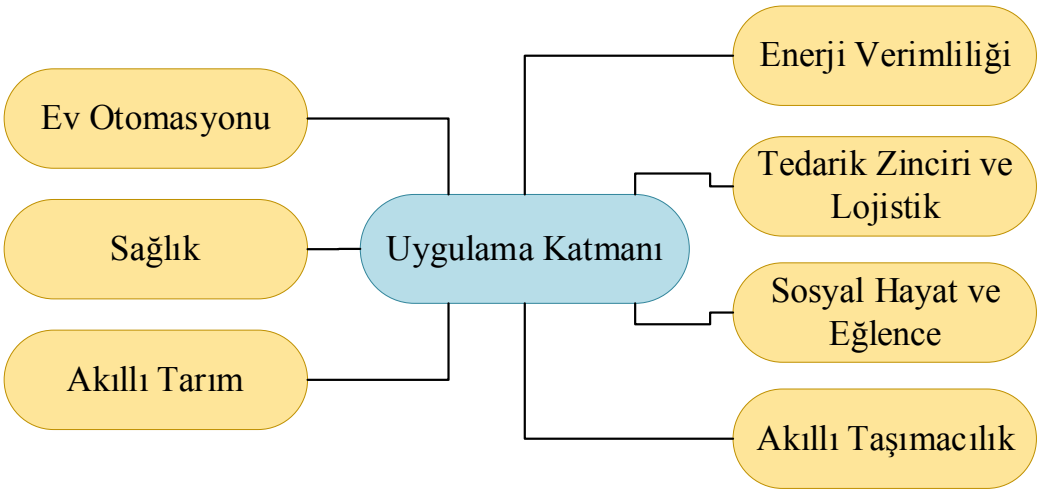
Ağ katmanında verinin dış ortama iletimi için birçok protokol ve teknoloji tanımlıdır. Bunlar Şekil 2.4'te verilmiştir ve sıklıkla kullanılanlar tezin 3. bölümünde açıklanmıştır.



Şekil 2.4. Ağ katmanında kullanılabilen teknolojiler.

Uygulama katmanı Şekil 2.5'te gösterildiği gibi nesnelerin interneti uygulamasının ne amaçla geliştirildiği ile ilgilidir. Bu konuda dünyada geliştirilmiş ve geliştirilmeye devam eden birçok proje vardır. Nesnelerin interneti uygulamalarının bazı kullanımları ev otomasyonu, kondisyon takibi, sağlık izleme, çevre koruma, akıllı şehirler ve endüstriyel üretimdir. Tüm bunlar uygulama katmanında tanımlıdır.

Beş katmanlı mimaride ise uygulamaların veri analitiği, karar verme mekanizmaları gibi bazı görevleri iş katmanına aktarılarak daha dinamik bir yapı oluşturulmuştur.



Şekil 2.5. Uygulama katmanında kullanılabilen teknolojiler.

BÖLÜM 3

3. NESNELERİN İNTERNETİ UYGULAMALARINDA KULLANILAN TEKNOLOJİLER VE PROTOKOLLER

3.1. Giriş

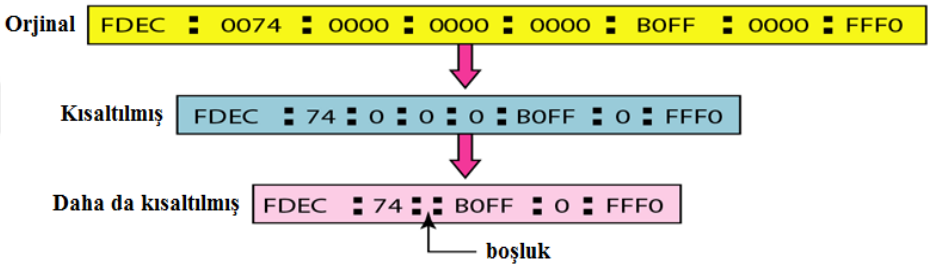
Nesnelerin interneti uygulamalarında cihazların birbirleri ile iletişimi için IPv6, 6LowPAN, ZigBee, Bluetooth, Wi-Fi, LoRaWAN, Z-Wave, NFC ve Sigfox gibi farklı teknolojiler kullanılmaktadır. Bu bölümde açıklanan teknolojilerin çoğunluğu tezin 2. bölümünde anlatılan mimarilerin ağ katmanı bölümünde yer almaktadır. Aslında doğrudan bilgisayar ağları ile ilgili olan bazı teknolojiler doğal olarak nesnelerin interneti uygulamalarında da gündeme gelmektedir. Fakat bu teknolojiler nesnelerin interneti bileşenleri için yetersiz de kalabilmektedir. Bu durum özellikle 5 yıllık bir süre içerisinde nesnelerin internetine uygun olan farklı protokoller ve teknolojiler doğmasına ve standartlaşmasına neden olacaktır. Tezin bu bölümünde genel bir altyapı oluşturması açısından nesnelerin interneti uygulamalarında kullanılan protokol ve kavramlara yer verilmiştir.

3.2. IPv6

IP (İnternet Protokolü), ağdaki cihazların birbirleri ile iletişim kurabilmek için kullanıldıkları ortak kurallardır. Bu protokol sayesinde ağdaki her cihaza birbirleri ile iletişim kurmaları için tekil bir adres verilir ve günümüzde çoğunlukla kullanılan sürümü IPv4'tür. IPv4 adresleri, 95.13.68.224 şeklinde aralarında nokta ile ayrılan ve onluklar şeklinde ifade edilen her biri 8 bitlik 4 oktetli bir yapıdadır. 32-bit yapıya sahip IPv4 yaklaşık olarak 4,3 milyar adet (2^{32}) IP adresi sağlar fakat bu adreslerin sadece yaklaşık

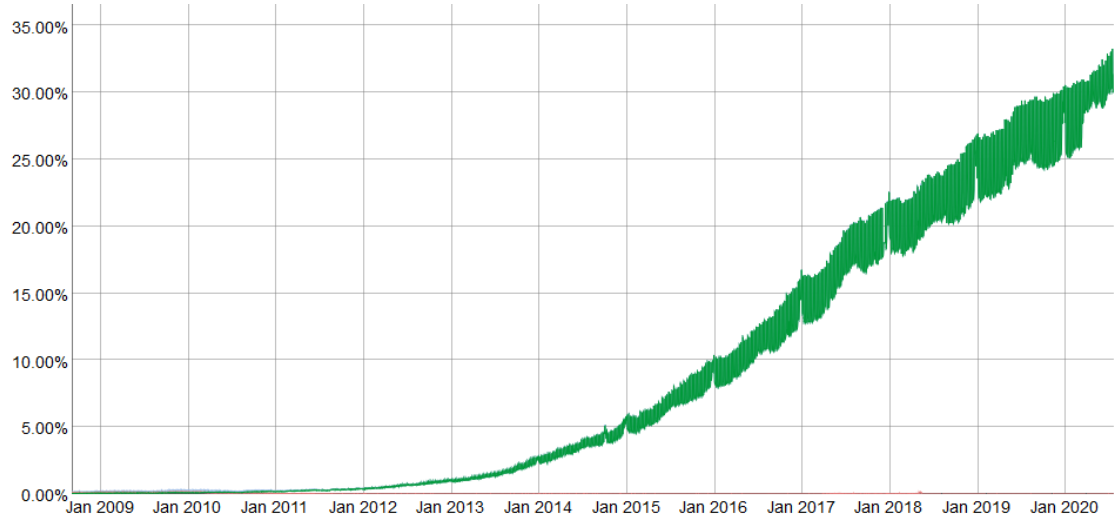
3,7 milyar adedi cihazlara verilebilir. Bu rakam günümüzdeki akıllı cihaz sayının çok altındadır.

IpV6, 128-bit yapıya sahiptir dolayısı ile 2^{128} adet IP adresi mevcuttur. Tezin 1. bölümde bahsedildiği üzere dünyadaki internete bağlanacak cihaz sayısı sadece önümüzdeki birkaç yıl içerisinde 50 milyara yaklaşacağı ve geçeceği düşünüldüğünde IPv4'ten IPv6'ya geçilmesi artık bir zorunluluk haline gelmiştir. IPv6'da adresleri Şekil 3.1'de gösterildiği gibi iki nokta üst üste (:) ile birbirlerinden ayrılan, her biri 16 bitlik 8 oktetten oluşan ve hexadecimal olarak belirtilen sayılardan oluşur (Deering & Hinden, 1998).



Şekil 3.1. IPv6 yapısı.

IPv6, internete bağlı noktaların artışıyla beraber ortaya çıkmış ve ciddi oranda IP çakışmalarının önüne geçmiştir. Şekil 3.2 incelendiğinde dünyada IPv6'ya geçişin 2015 yılında %5 civarında iken günümüzde %32'ye yaklaştığı görülebilir. Kullanım oranında ilk sırada %54.41 oranıyla Belçika yer almakla birlikte ülkemiz %0.01 oranındadır (Google, 2020). Elbette önümüzdeki 10 yıl içerisinde bu rakamlarda ciddi farklılıklar gözlenecektir. Dünyanın IPv6'ya geçişi ile beraber tüm cihazlar IP adresi alarak yönetilebilir ve kullanılabilir olabilecektir.



Şekil 3.2. IPv6'ya geçiş oranı.

3.3. IEEE 802.15.4

IEEE 802.15.4, LR-WPAN (Düşük Hızlı Kablosuz Kişisel Alan Ağı) için oluşturulan standart bir protokoldür. LR-WPAN, nesnelerin interneti uygulamalarındaki cihazların birbirleri ile olan veri iletişiminde kullanılan ağ çeşididir. LAN (yerel alan ağı) ve WAN (geniş alan ağı) içinde kullanılan diğer teknolojilerden ayrılan en önemli özelliği kablosuz olarak düşük güçte ve veri hızında kısa mesafeli iletişimde çalışmasıdır (Chen, Wang, German, & Dressler, 2010).

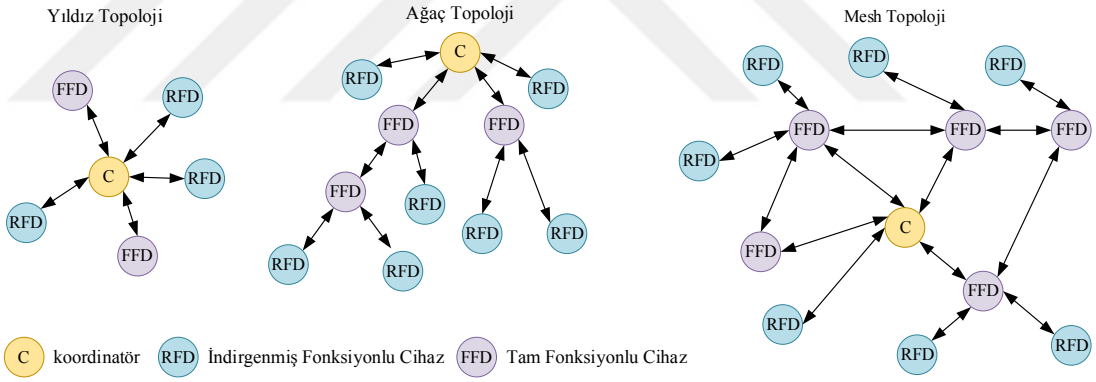
IEEE 802.15.4, düşük güçlü ve düşük hızlı aygıtlar için düşük maliyetli bir iletişim ağı sunmakla beraber basit ve kolay kurulum, güvenilir veri aktarımı, düşük maliyet ve makul bir pil ömrü sağlar. IEEE 802.15.4 protokolü, 2.4 GHz ISM[1] bandında 16 kanal ile 250 kbps hız ile iletime imkan verir.

IEEE 802.15.4 ağına iki farklı türde aygıt bağlanabilir. Bu aygıtlara FFD (Tam Fonksiyonlu Cihaz) ve RFD (İndirgenmiş Fonksiyonlu Cihaz) adı verilir. FFD aygıtları, ağda PAN (Kişisel Alan Ağı) koordinatörü olarak işlev görebilirler. PAN koordinatörü, ağın tüm sorumluluğunu üstlenen ana koordinatördür. Bu cihazlar diğer aygıtlarla iletişim kurar ve mesajları ağ üzerinden iletir. Öte yandan RFD'ler, minimum kaynak ve hafızaya

[1] ISM (Sınai, bilimsel ve tıbbi cihaz bandı), birçok ülkede telsiz iletişimi için sertifika veya lisansa gerek olmadan belirli bir çıkış gücü sınırlamasına uyarak, üzerinden yayın yapılabilen banttır.

sahip olan büyük miktarda veri göndermek zorunda olmayan uç cihazlardır. FFD aygıtları ise ağ içinde yönlendirmeleri yapan cihazlardır.

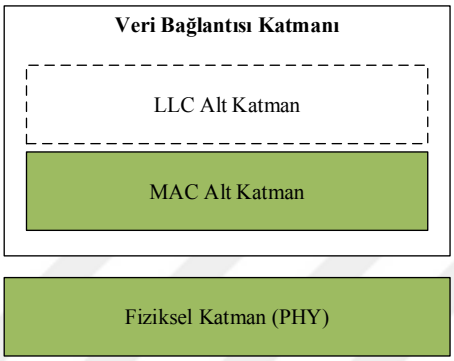
IEEE 802.15.4 ile çeşitli ağ topolojileri mümkündür. Bunlar Şekil 3.3'te gösterildiği gibi yıldız, ağaç ve mesh topolojidir. Yıldız topolojisinde, ağın tüm düğümleri merkezi bir PAN koordinatöründen oluşur. Bu düğümlerin her biri yalnızca PAN Koordinatörüne bağlıdır. Bu nedenle, bir düğümden diğerine mesaj göndermek için mesajın koordinatör aracılığıyla gönderilmesi gerekir. Ağaç topolojisinde, ağ içinde ebeveyn-çocuk ilişkilerine dayanan bir yapı mevcuttur. PAN koordinatörü hariç her düğümün bir üst öğesi vardır. Her düğüm yalnızca ebeveyni ve varsa çocuklarıyla iletişim kurabilir. Ebeveyn olan herhangi bir düğüm, çocukları için yerel bir koordinatör görevi görür. Ayrıca ağaç yapısı daha fonksiyonel bir ağ için kümelere ayrılabilir. Mesh topolojisinde, düğümler doğrudan birbirleri ile iletişim kurabilir. Tüm düğümler birbirinin menzili içinde olmayabilir, ancak bir mesaj son hedefine ulaşıncaya kadar bir düğümden diğerine geçerek hedefe iletilir.



Şekil 3.3. 802.15.4 ağ topolojileri.

Şekil 3.4'te IEEE 802.15.4 protokol yığını gösterilmektedir. Tezin bu bölümünde anlatılan birçok protokolün fiziksel katmanı IEEE 802.15.4 üzerine kurulmuştur. IEEE 802.15.4 fiziksel ve veri bağlantısı katmanı olmak üzere 2 katmanlı bir yapıdadır. Fiziksel (PHY) katman, sinyallerin MAC alt katmanına uygun bir şekilde aktarılması ile ilgilenir. Bu aşamada kablosuz sinyallerin kanal ayarlamaları ve bit modülasyonu, paket senkronizasyonu gibi bit seviyesindeki işlemler bu katmanda gerçekleşir. Cihazların ağ

ile ilişkilendirilmesi ve ilişkilerinin kesilmesi, paylaşılan kanallara erişimin kontrol edilmesi, bir üst katmana ve alt katmana uygun bir biçimde veri aktarılması MAC alt katmanının ana sorumlulukları arasındadır. MAC alt katmanı ve LLC alt katmanı birleşerek veri bağlantı katmanı olarak adlandırılır. LLC, tüm IEEE 802 standartları için ortaktır. Fakat IEEE 802.15.4 tabanlı uygulamaların geliştirilmesinde LLC göz ardı edilebilir.

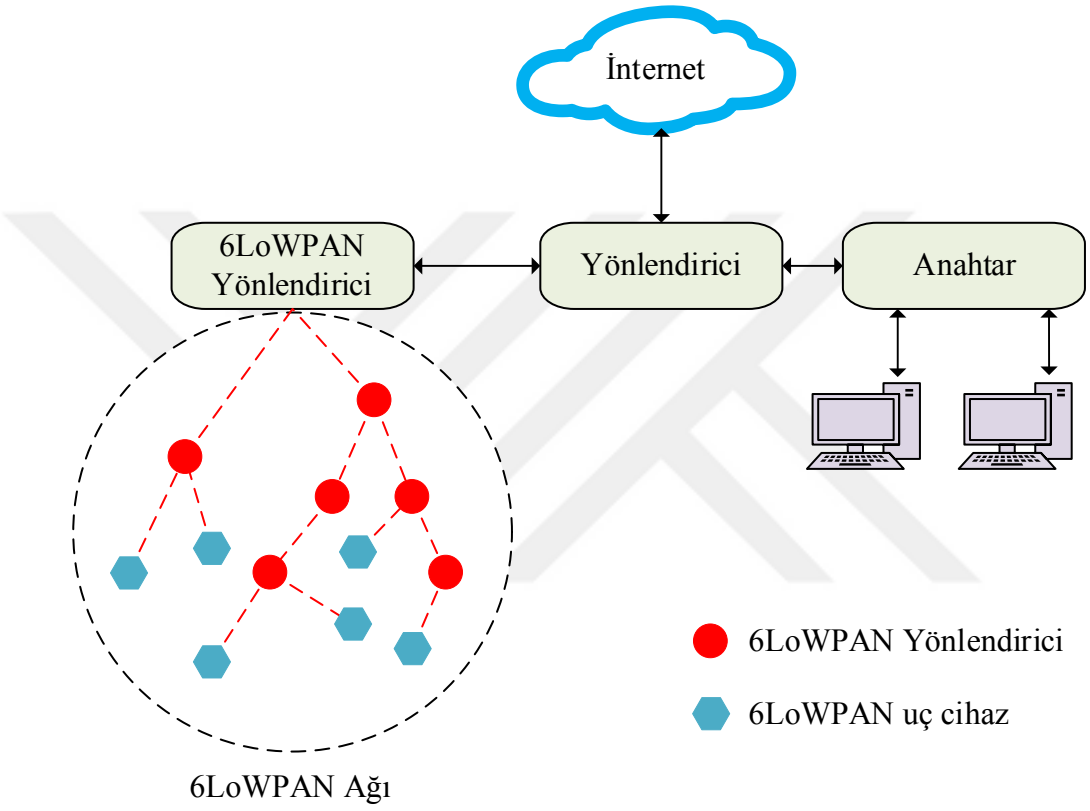


Şekil 3.4. IEEE 802.15.4 protokol yığını.

3.4. 6LoWPAN

6LoWPAN (IPv6 Kullanan Düşük Güçlü Kablosuz Kişisel Alan Ağı), IETF (İnternet Mühendisliği Görev Grubu) tarafından üretilmiş IPv6 destekli cihazları LR-WPAN üzerinden veri iletilmesini amaçlamayan açık kaynaklı bir protokoldür. Genel olarak üretim amacı sınırlı işlem kapasitesine sahip, düşük güç tüketen küçük cihazların internet ağına bağlanabilmesidir (Shelby & Bormann, 2011). 6LoWPAN destekli bir cihaz doğrudan internete bağlanabilmektedir. 6LoWPAN protokolünde ağdaki her cihaz Ipv6 adresine sahip olmalıdır. Düşük veri hızlarında kablosuz internet bağlantısı gerekli olan uygulamalar için tercih edilir. 6LoWPAN, 868 MHz ile 2,4 GHz frekans aralığında ve 250 kbps veri iletim hızı ile çalışmaktadır. Bu şekilde 10 ile 100 metre aralığında kapsama alanı sağlar. Veri iletimi optimize bir şekilde yönlendiriciler kullanılarak düğümler üzerinden sağlanır. 6LoWPAN, diğer WAN ve LAN ağları ile entegre çalışabilir. 6LoWPAN sektörde oldukça yenidir. Özellikle ev otomasyonu, sokak lambası izleme ve kontrolü, konut aydınlatması, akıllı ölçüm ve internet bağlantılı cihazlarla genel nesnelerin interneti uygulamaları gibi teknolojiler için idealdir (Mulligan, 2007).

Şekil 3.5'te içerisinde 6LoWPAN protokolü ile iletişim kuran cihazların bulunduğu bir ağ yapısı gösterilmiştir. Buna göre sistemdeki bilgisayarlar yönlendiriciler aracılığı ile internete ulaşırken 6LoWPAN ağı içindeki cihazlar ise 6LoWPAN yönlendiricisi tarafından internete ulaşmaktadır. Yönlendiricilerin görevi ağı oluşturmak, iç ve dış ağıdaki veri iletişimini sağlamak ve ağın bakımını yapmaktır.

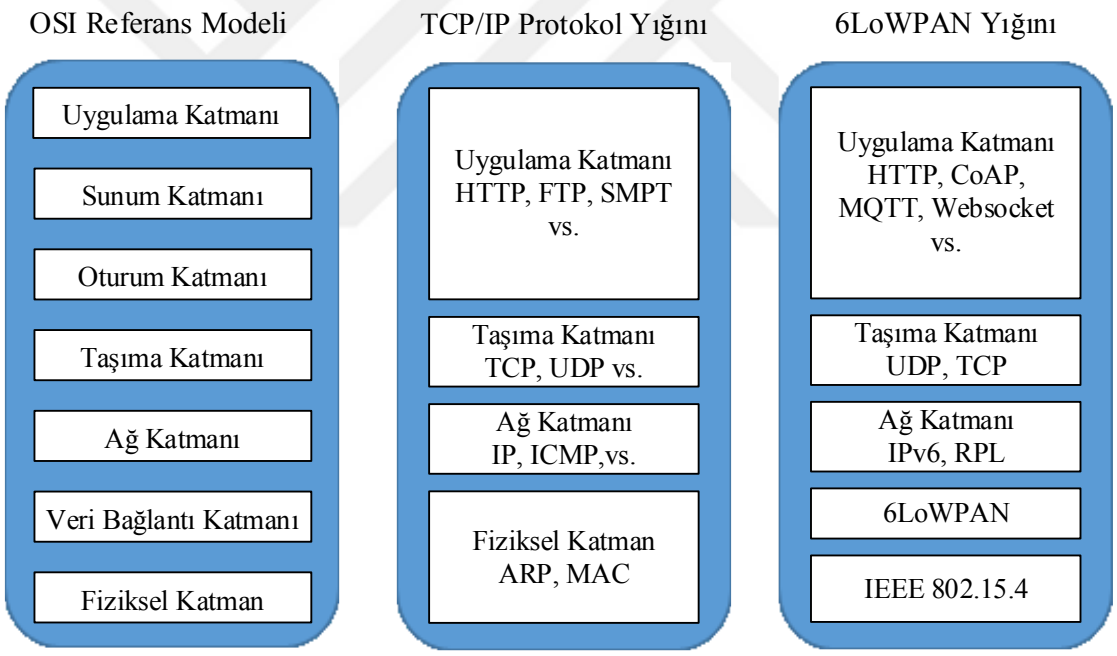


Şekil 3.5. 6LoWPAN ağ yapısı

Tipik bir 6LoWPAN ağına yönlendirici ve uç cihaz olmak üzere iki cihaz türü mevcuttur. Yönlendiriciler, adından da anlaşılacağı gibi, 6LoWPAN ağındaki başka bir düğüme giden verileri en kısa yoldan yönlendirirler. Uç cihazlar ise sensörlerden topladıkları verileri talep edildiği zaman bağlı olduğu yönlendirici cihaza iletir. Bu cihazlar sadece talep geldiği zaman uyku konumundan çıkıp aktif moda geçerler. Bu sebeple ağ ne kadar büyük olursa olsun enerji tüketimi düşük olur.

OSI referans modeli, TCP/IP protokol yığını ile 6LoWPAN protokol yığınının katman yapısı Şekil 3.6'da gösterilmiştir. Buna göre 6LoWPAN protokol yığını fiziksel

katmanında IEEE 802.15.4 kullanılarak veri bitlerini kablosuz sinyallere dönüştürür. Veri bağlantısı katmanı, veri iletişimi sırasında fiziksel katmanda oluşabilecek hataları tespit edip düzelterek sağlıklı bir iletişimin sağlanmasını amaçlar. OSI ve TCP/IP'de kullanılan MAC, CSMA-CD gibi protokoller 6LoWPAN için IEEE 802.15.4 tarafından sağlanır. Ayrıca ağ içindeki cihazların kontrolü, verilerin yönlendirilmesi gibi işlemler için 6LoWPAN uyarlaması 6LoWPAN adındaki protokol ve katman ile yapılmıştır. Ağ ve taşıma katmanları temelde aynı işlevi görmektedir. Fakat 6LoWPAN için veri taşıma esnasında genelde UDP protokolü kullanılır. Böylece daha hızlı bir iletişim kurulur. Uygulama katmanında TCP/IP iletişiminde genellikle HTTP protokolü kullanılır. Her ne kadar 6LoWPAN için HTTP protokolü mevcut olsa da uygulama düzeyinde iletişim için nesnelerin interneti cihazlarına daha uygun olan CoAP (Kısıtlı Uygulama Protokolü) veya MQTT (Mesaj Kuyruğu Telemetri Aktarımı) protokolleri tercih edilir.



Şekil 3.6. OSI, TCP/IP ve 6LoWPAN protokol yığınının karşılaştırılması

3.5. ZigBee

ZigBee, nesnelerin interneti uygulamalarındaki cihazların birbirleri ile iletişimi için tasarlanmış birlikte çalışabilirliği yüksek ve standartlara dayalı bir kablosuz ağ teknolojisidir. IEEE 802.15.4 standardını temel alan ZigBee, düşük veri hızlarını, düşük

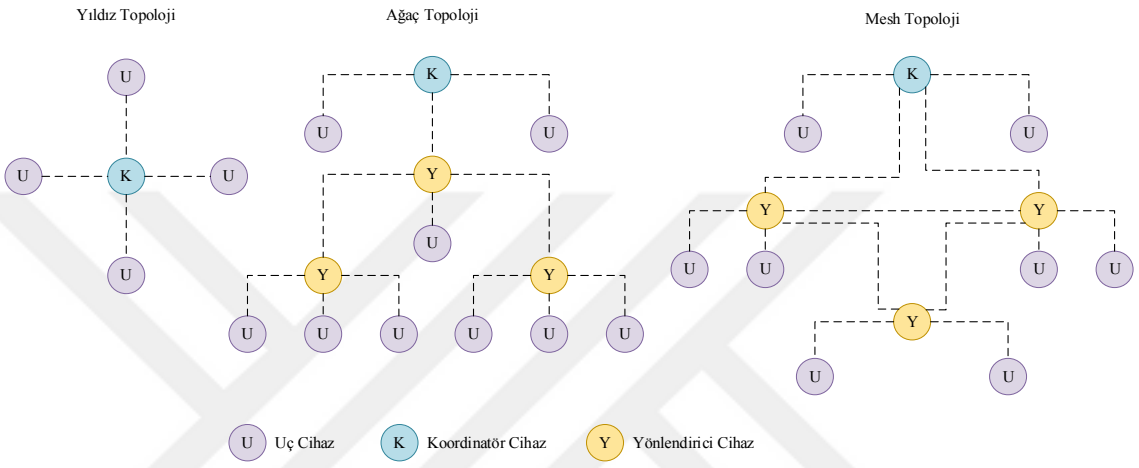
güç tüketimini, güvenliği ve güvenilirliği destekleyen uygun maliyetli kablosuz ağ çözümü ihtiyacını karşılamak için oluşturulmuştur. ZigBee, kendi kendini optimize eden ağ yapısına sahiptir. Bunu; bir düğümün veri iletim yolunun başarısız olması durumunda diğer düğümlerin ağ genelinde yeni yollar bulmasını sağlayarak gerçekleştirir. Merkezi olmayan bir ağ yapısı olduğu için nesnelerin interneti sektöründe güçlü bir kablosuz çözüm olarak kullanılmaktadır (Farahani, 2011).

ZigBee özellikle kablosuz sensör ağlarını hedeflemektedir. Kablosuz bağlantı gerektiren herhangi bir izleme ve kontrol uygulamasında kullanılabilir. Fakat daha çok akıllı ev ve bina otomasyonu, ev güvenlik sistemleri, akıllı şehirler, lojistik ve varlık yönetim ve takibi, hasta izleme gibi sistemlerde kullanılmaktadır.

ZigBee 868 MHz frekansında 20 kbps, 915 MHz frekansında 40 kbps ve 2,4 GHz frekansında 250 kbps hızla çalışmaktadır. Temel olarak düşük veri hızı, düşük maliyet ve uzun pil ömrünün temel gereksinimler olduğu pille çalışan cihazların dahil olduğu uygulamalar için uygundur. Birçok ZigBee uygulamasında, kablosuz aygıtın veri alma veya aktarma gibi herhangi bir etkinlik türüne katıldığı toplam süre çok sınırlıdır; cihaz zamanının çoğunu uyku modu olarak da bilinen güç tasarrufu modunda geçirir. Bu şekilde cihazlar pillerinin değiştirilmesi gerekmeden birkaç yıl boyunca çalışabilir.

ZigBee, Şekil 3.7'de gösterildiği gibi yıldız, ağaç ve mesh topolojilerini destekler. Yıldız topolojisinde, tüm ağ koordinatör adındaki tek bir cihaz tarafından kontrol edilir. Koordinatör ağdaki cihazların başlatılmasından ve korunmasından sorumludur. Uç cihazlar olarak bilinen diğer tüm cihazlar kendi aralarında iletişim kuramazlar, bu cihazlar doğrudan ZigBee koordinatörü ile iletişim kurarlar. Bu topolojinin dezavantajı, ağın işleyişinin sadece koordinatöre bağlı olmasıdır. Cihazlar arasındaki tüm iletişimin koordinatör üzerinden geçmesi gerektiğinden, ağın büyüklüğüne bağlı olarak koordinatör bazı isteklere cevap veremeyebilir ve tıkanabilir. Ayrıca, kaynaktan hedefe doğru alternatif bir yol yoktur. Bu basit topoloji özellikle küçük ağlarda tercih edilebilir. Ağaç topolojisinde koordinatöre ek olarak yönlendiriciler bulunmaktadır. Yönlendiriciler verileri hiyerarşik bir yönlendirme stratejisi kullanarak ağ üzerinden taşır ve kontrol ederler. Ağ ZigBee yönlendiricileri kullanılarak genişletilebilir. Ağaç topolojisinde sadece koordinatöre veya yönlendiriciye uç cihaz bağlanabilir. Gerektiğinde küme ağaçlar oluşturulabilir. Bu topoloji yıldız topolojiye göre daha yönetilebilir bir sistem

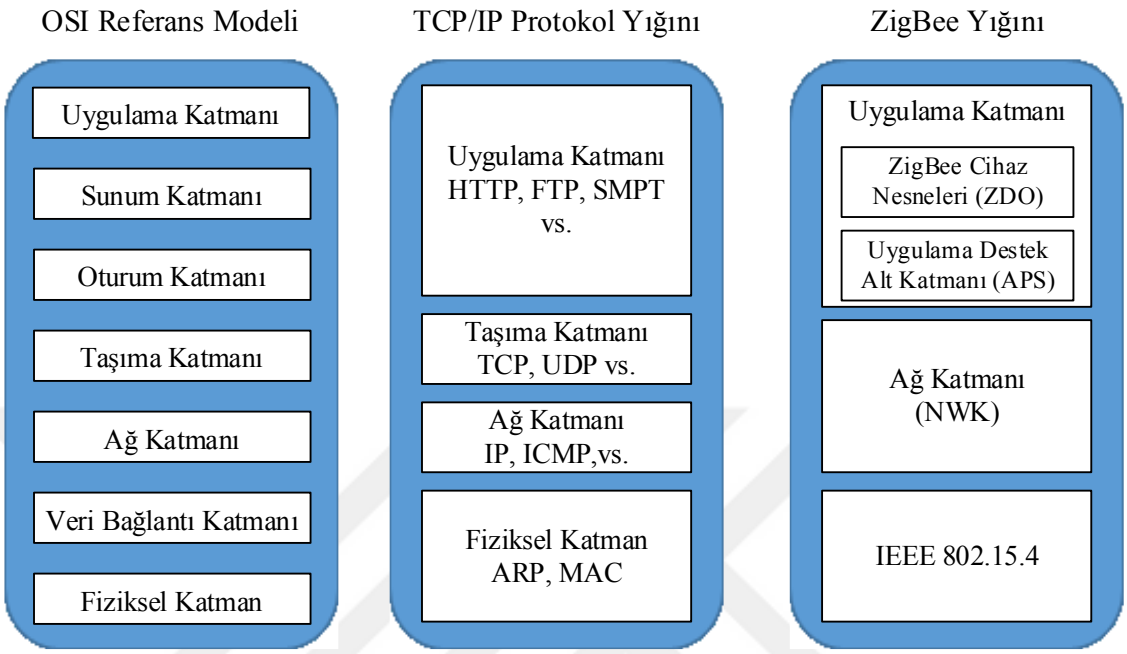
sunmaktadır. Fakat herhangi bir sebeple yönlendiricilerden biri devre dışı kaldığında ona bağlı olan tüm düşümlerin ağ ile olan iletişimi kesilir. Eşler arası ağ olarak da adlandırılan mesh topolojisi, bir koordinatör, birkaç yönlendirici ve uç aygıttan oluşur. Koordinatörler birbirleri ile iletişimde olduğundan dolayı kaynaktan hedefe olan iletişim için birden fazla yol mevcuttur. Bu topolojide ağ kolaylıkla ölçeklenebilir. Fakat daha karışık bir yönetim söz konusudur.



Şekil 3.7. ZigBee topolojileri.

OSI referans modeli, TCP/IP protokol yığını ile ZigBee protokol yığınının katman yapısı Şekil 3.8'de gösterilmiştir. Buna göre fiziksel katmandaki tüm süreçler IEEE 802.15.4 protokolüne göre yürütülür. ZigBee ağ katmanı (NWK) esas olarak yeni bir ağ oluşturma sorumluluğunu üstlenir. Ağa katılma ve ağdan ayrılma, yeni bir cihaz ağa katıldığında çalışacak şekilde ağı yapılandırma, ağa katılan cihaza adres atama, tüm koordinatör ve yönlendirici işlemlerini organize etme işleri bu katman ile sağlanır. Dolayısıyla TCP/IP kümesinin ağ ve taşıma katmanlarının görevlerinin bazıları bu katmanda tanımlanmıştır. Uygulama Katmanı, Uygulama Destek alt katmanı (APS) ve ZigBee Cihaz Nesneleri alt katmanından (ZDO) oluşur. Üretici tarafından tanımlanan uygulamaları içerir. APS alt katmanı, keşif ve bağlayıcı hizmetlerden sorumludur. Keşif, ağ üzerinde çalışan cihazları tespit etmek için; bağlama ise iki veya daha fazla cihazı eşleştirmek ve bağlı cihazlar arasında mesaj iletmek için kullanılır. ZigBee Cihaz Nesneleri alt katmanı (ZDO), ağın yerel ve kablosuz yönetimini yapar. Uygulama

katmanında her biri 1'den 240'a kadar bir uç nokta adresiyle tanımlanan 240'a kadar farklı uygulama nesnesi tanımlanabilir.



Şekil 3.8. OSI, TCP/IP ve ZigBee protokol yığınının karşılaştırılması.

3.6. Bluetooth

Bluetooth teknolojisi ilk olarak Ericsson Mobile firması tarafından 1994 yılında geliştirilmiştir. Daha sonrasında 1998 yılında kurulan Bluetooth Special Interest Group Bluetooth teknolojisinin geliştirilmesine devam etmiştir. Birçok üründe kullanılan Bluetooth IEEE 802.15.1 ile standartlaşmıştır. Bluetooth teknolojisi, kısa mesafeli, hareketli ve/veya sabit cihazların birbirleriyle iletişimini kablo bağlantısı olmadan gerçekleştirmek için geliştirilmiştir. Düşük maliyet, düşük güç tüketimi ve kararlı bir iletişime olanak sağlaması önemli avantajları arasındadır (Miller, Bisdikian, & Foreword By-Siep, 2000).

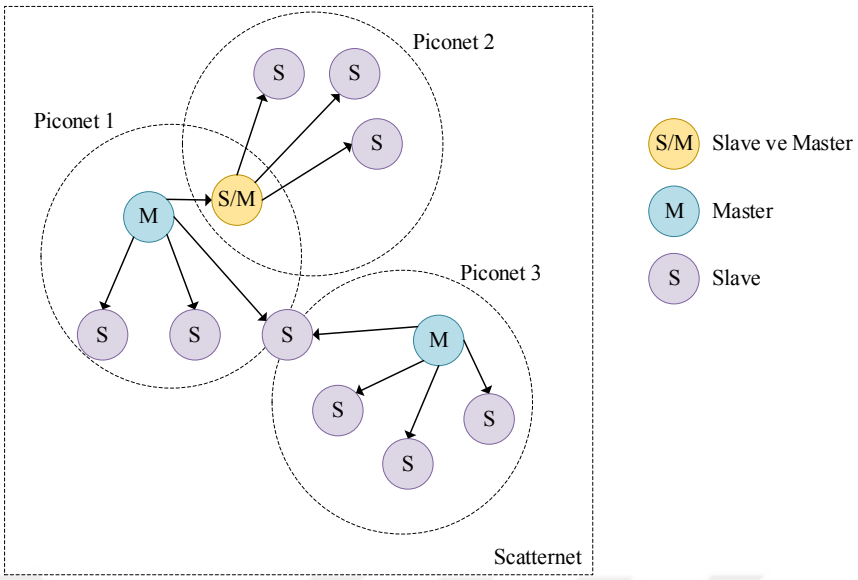
Bluetooth, 2.4 GHz lisansız ISM bandı üzerinde sayısal olarak ses ve veri aktarımı sağlayan kablosuz teknolojidir. Radyo dalgalarıyla çalıştığından cihazların birlerini doğrudan görmelerine ihtiyaç yoktur. Bluetooth 2.400 - 2.483 GHz arasında 1'er MHz'lik bantlar ile çalışır. Anten çıkış gücüne göre cihazların kapmasa mesafeleri değişmektedir

ve bu doğrultuda Bluetooth cihazlar üç sınıfa ayrılabilir. Bu sınıflar Çizelge 3.1'de verilmiştir.

Çizelge 3.1. Bluetooth sınıfları

Sınıf	**Maksimum Güç**	**Maksimum Mesafe**
1.Sınıf	100mW (20dBm)	100 metre
2.Sınıf	2.5mW (4dBm)	10 metre
3.Sınıf	1mW (0dBm)	1 metre

Bluetooth cihazlar ile oluşturulan ağlar Piconet ve Scatternet olarak isimlendirilmektedir. Piconet ağlarda en fazla 8 adet master-slave (efendi-köle) ilişkisinde bulunan cihaz bulunabilir. Ağa ilk bağlanan cihaz master olarak ifade edilirken diğer cihazlara slave adı verilir. Master cihaz ağı kontrol eder ve böylelikle kablosuz ağ zinciri kurulur. Her Piconet içerisinde sadece bir master cihaz bulunur. Slave cihazlar master cihaz ile iletişime geçtiği zaman aktif diğer durumlarda pasif konumda olurlar. Bir Piconet içerisinde en fazla 7 adet aktif slave bulunabilirken, 255 adet pasif slave olabilir. Piconet ağındaki tüm cihazlar aynı frekans kanalını kullanırlar. Piconet ağının aralığı, Bluetooth cihazının sınıfına göre değişir. Veri aktarım hızı yaklaşık 200 kbps ila 2100 kbps arasında farklılık gösterir. Ağı genişletmek için Piconet ağlar birbirine bağlanarak Scatternet ağlar oluşturulabilir. Bu durumda her Piconet farklı bir atlama kanalı kullanılır. Şekil 3.9'da Piconet ve Scatternet ağları için bir örnek verilmiştir. Buna göre 3 adet Piconet ağı birleşerek Scatternet ağını oluşturmuştur. 1. Piconet ağındaki slave cihaz 2. Piconet ağının master cihazıdır ve 2 ağın kapsama alanı içindedir. Ayrıca 1. Piconet içindeki slave cihaz 3. Piconet içinde yine slave olarak yer almıştır.



Şekil 3.9. Bluetooth Piconet ve Scatternet ağları.

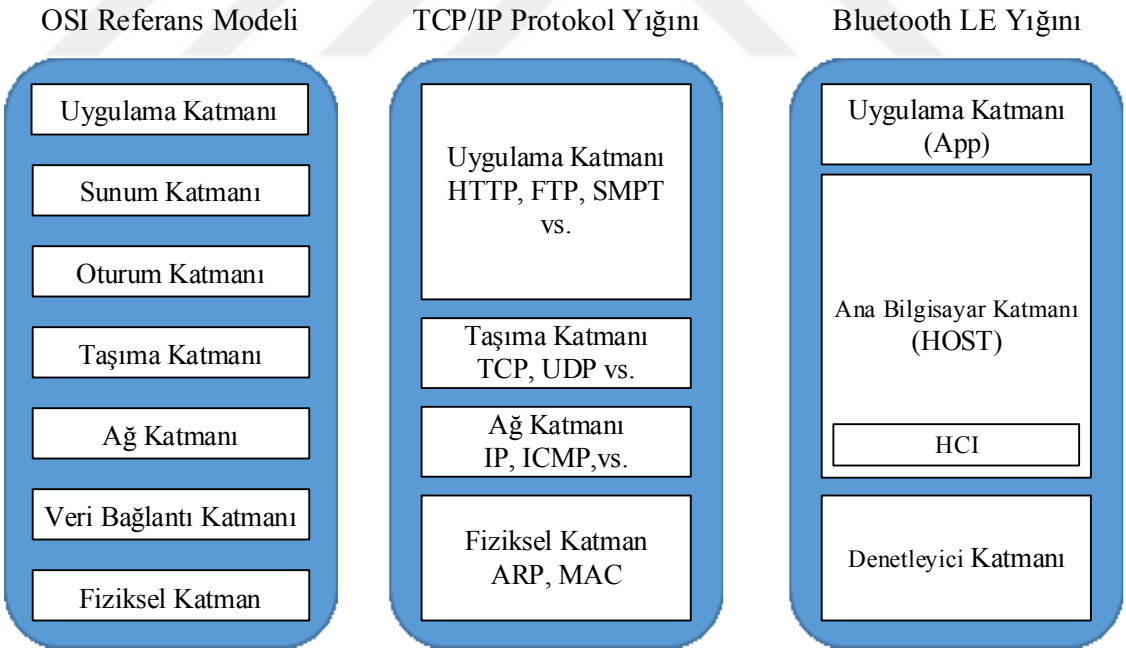
Bluetooth cihazlar, BR/EDR ve BLE (Bluetooth düşük enerji) teknolojisini destekleyen çift mod veya yalnızca Bluetooth düşük enerji teknolojisini destekleyen tek mod olabilir. BLE'nin ciddi anlamda enerji verimliliğinin yüksek olması nesnelerin interneti uygulamalarında oldukça fazla bluetooth cihazların kullanılmasını sağlamaktadır.

Şekil 3.10'da gösterildiği gibi tek modlu BLE aygıtı üç kısma ayrılmıştır: denetleyici, ana makine ve uygulama. Protokol yığında bu temel yapı taşlarının her biri, çalışması için gereken işlevselliği sağlayan çeşitli katmanlara da ayrılmıştır.

- Uygulama Katmanı: Diğer tüm sistem türlerinde olduğu gibi uygulama katmanı, kullanım durumu ile ilgili her şeyin mantığını, kullanıcı arabirimini ve veri işlemeyi içeren en üst katmandır.
- Ana Bilgisayar Katmanı: Genel Erişim Profili (GAP), Genel Öznitelik Profili (GATT) Mantıksal Bağlantı Kontrol ve Uyarlama Protokolü (L2CAP), Öznitelik Protokolü (ATT), Güvenlik Yöneticisi (SMP), Ana Bilgisayar Denetleyicisi Arabirimi (HCI) katmanlarını içerir. Ana bilgisayar kontrol arabirimi (HCI) protokol yığınının bir parçası değildir, ancak Bluetooth cihazların donanım özelliklerine erişmek için arabirim yöntemi sağlar. HCI ürün yazılımı, temel bant komutları, bağlantı yöneticisi komutları, donanım

durum kayıtları, kontrol kayıtları ve olay kayıtları gibi bilgilere erişerek Bluetooth donanımının istenilen düzeyde çalışmasını sağlar.

- Denetleyici: Ana Bilgisayar Denetleyicisi Arabirimi (HCI), Bağlantı Katmanı (LL) ve Fiziksel Katmanı (PHY) içerir.
    - Fiziksel katman, analog sinyalleri modüle edip demodüle edebilen ve bunları dijital sembollere dönüştüren analog iletişim devrelerini içeren kısımdır.
    - Bağlantı Katmanı, doğrudan PHY ile arabirim oluşturan kısımdır ve genellikle, özel donanım ve yazılımın birleşimi olarak uygulanmaktadır. Ayrıca, tüm protokol yığınının gerçek zamanlı katmanıdır, çünkü kurallar tarafından tanımlanan zamanlama gereksinimlerinin hepsine uymaktan sorumludur. Veri girişi, erişim adresi ve sinyal protokolü çerçevelemesi, CRC (Döngüsel Artıklık Denetimi) üretimi ve doğrulaması, Veri beyazlatma, Rasgele sayı üretme, AES (Gelişmiş Şifreleme Standardı) şifreleme gibi işlemler bu katmanda tanımlıdır.



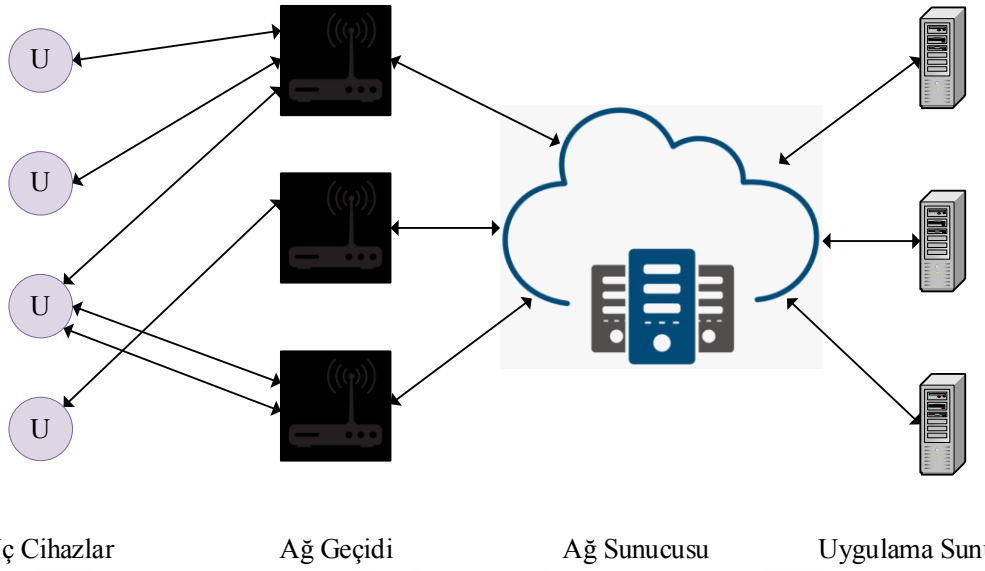
Şekil 3.10. OSI, TCP/IP ve Bluetooth LE protokol yığınının karşılaştırılması.

3.7. LoRaWAN

LoRaWAN (Uzun Menzilli Geniş Alan Ağı), geniş alan ağları için düşük güç kullanımına sahip açık kaynaklı bir teknolojidir. Nesnelerin interneti uygulamaları için bir düğümün pil ömrü; ağın kapasitesi, hizmet kalitesi ve güvenliği açısından önemlidir. LoRaWAN ağları, ağ geçitlerinin uç cihazlar ve arka uçtaki merkezi ağ sunucusu arasındaki mesajları ilettiği bir yıldız topolojisinde düzenlenir. Uygulanan ortama bağlı olarak cihazlar arası 0,3 kbps ile 50 kbps arasında veri iletim hızı sunar. LoRaWAN ile iletişim şehir içinde 2 ila 5 km arasında, şehir dışında ise 15 km'ye kadar mesafeye sahiptir. Akıllı şehir uygulamaları, akıllı tarım, akıllı fabrikalar gibi birçok alanda LoRaWAN uygun bir çözüm olarak düşünülebilir. Akıllı şehirlere örnek olarak Amsterdam'da LoRaWAN ile tam bir kapsama sağlamak için, 10 adet ağ geçidi kullanarak nesnelerin interneti veri ağı oluşturulmuştur. Bu derece kapsamlı bir LoRaWAN uygulaması olarak Amsterdam ilk örnektir (Adelantado vd., 2017).

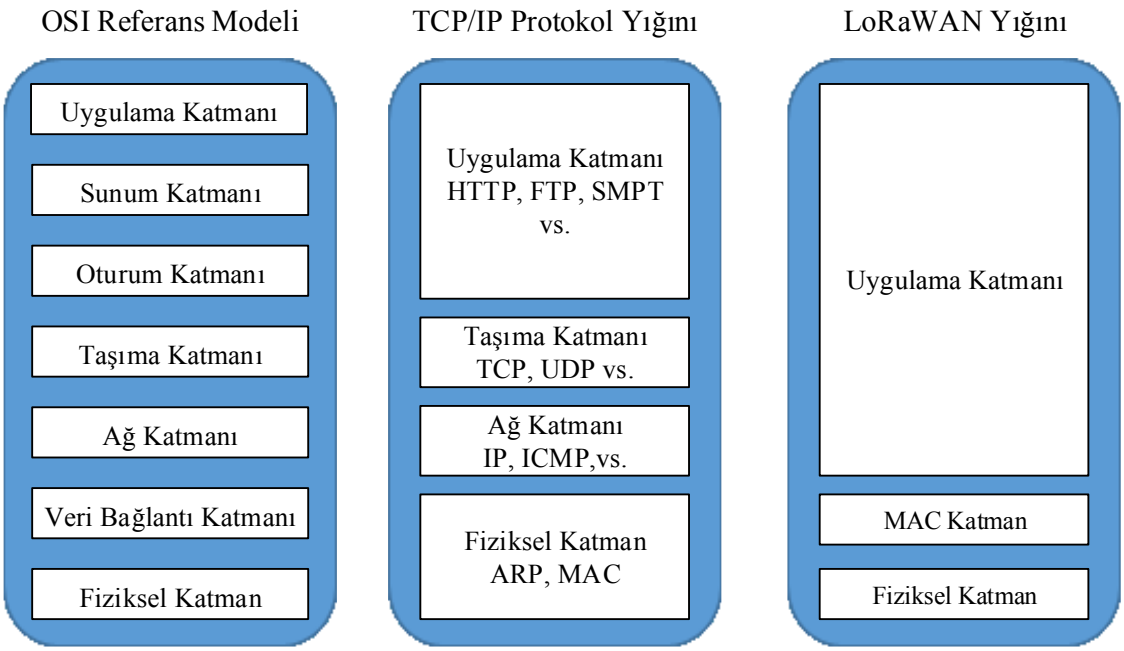
LoRaWAN lisanssız radyo spektrumunda çalışır. Böylelikle kurulacak uygulamada herhangi bir lisans ücreti ödemeden radyo frekansları rahatlıkla kullanılabilir. Wi-Fi'ye benzer olarak dünya genelinde 2.4 GHz ve 5 GHz ISM bantlarını kullanan sistem, herkesin bir lisans veya izne gerek kalmaksızın Wi-Fi yönlendiricileri kurmasına ve Wi-Fi sinyallerini gönderip almasına izin verir. LoRaWAN, ABD'de 902-928 MHz, AB'de 863-870 MHz ve Çin'de 779-787 MHz aralığını destekler ve genellikle Avrupa ülkelerinde 868 MHz, Amerika'da ise 900 MHz bandı kullanarak yayın yapar.

LoRaWAN ağı çeşitli unsurlardan oluşur. Bunlar uç cihazlar, ağ geçidi, ağ sunucusu ve uygulama sunucularıdır. Uç cihazlar, algılama veya kontrolün yapıldığı LoRaWAN ağının unsurlarıdır. Ağ geçidi, LoRaWAN uç cihazlarından verileri alır ve daha sonra ana taşıyıcı sisteme aktarır. LoRaWAN ağının bu kısmı Ethernet, GSM veya Wi-Fi gibi herhangi bir iletişim bağlantısı ile yapılabilir. Ağ geçitleri ağ sunucusuna standart IP ağları üzerinden bağlanır. Bu sebeple sistem diğer iletim ortamları ile entegre olarak çalışabilir. Ağ sunucusu ağı yönetir. Yinelenen paketleri ortadan kaldırmak, ağı planlamak ve veri hızlarını uyarlamak için çalışır. Uygulama sunucuları uç noktalara dışarıdan bağlanarak cihazların eylemlerini kontrol edebilir veya veri toplayabilir. Şekil 3.11'de LoRaWAN ağ yapısı gösterilmiştir.



Şekil 3.11. LoRaWAN ağ yapısı.

LoRaWAN, OSI modelinin ikinci ve üçüncü katmanına eşlenebilir. LoRaWAN protokolleri LoRa Alliance adlı bir grup tarafından tanımlanır. LoRaWAN yığın katmanları Şekil 3.12'de gösterilmektedir.



Şekil 3.12. OSI, TCP/IP ve LoRaWAN protokol yığınının karşılaştırılması

LoRaWAN, üç farklı 128-bit güvenlik anahtarını kullanır. Uygulama anahtarı (AppKey) yalnızca cihaz ve uygulama tarafından bilinir. Bir aygıt ağa katıldığında, uygulama oturum anahtarı (AppSKey) ve bir ağ oturum anahtarı (NwkSKey) oluşturulur. AppSKey özel tutulurken (private key), NwkSKey ağ ile paylaşılır (public key). Oturum anahtarları, oturum süresi boyunca yani cihaz ağa bağlı olduğu sürece kullanılmaktadır.

AppSKey, ağ paketinin uçtan uca şifrelenmesi için kullanılır. Bunun için kullanılan algoritma IEEE 802.15.4 standardında kullanılan algoritmaya benzer şekilde AES-128'dir. NwkSKey ağ ve aygıt tarafından bilinir ve ileti bütünlüğünü MIC (İleti Bütünlüğü Kodu) ile doğrulamak için kullanılır. MIC, iletinin dışarıdan değiştirilmesinin önlenmesini ve iletinin doğru olarak iletildiğinin kontrolü için kullanılır. Bunun için LoRaWAN AES-CMAC kullanmaktadır.

LoRaWAN yeni ve uygun bir teknoloji olmasının yanında bazı dezavantajları da barındırır. Sistem, düşük veri hızı gerektiren ve ortalama 27 kbps'ye kadar olan uygulamalar için kullanılabilir. Ayrıca özellikle düşük gecikme gereksinimine sahip olan gerçek zamanlı uygulamalarda kullanımı uygun değildir.

3.8. Z-Wave

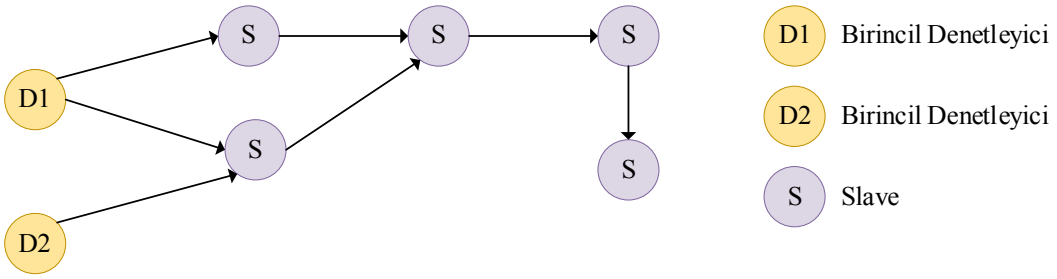
Z-Wave özellikle akıllı ev sistemleri için tasarlanmış düşük güç tüketimine sahip bir MAC protokolüdür. Yaklaşık 30 metrelik noktadan noktaya iletişimi kapsar ve nesnelerin interneti uygulamalarında, ışık kontrolü, enerji kontrolü, giyilebilir sağlık kontrolü gibi küçük uygulamalar için uygundur. Z-Wave protokolü, ITU tarafından geliştirilmiştir, günümüzde ise Z-Wave Alliance adındaki grup içindeki 300'den fazla uluslararası şirket tarafından desteklenmektedir. Birçok Z-Wave cihazı doğrudan elektrik enerjisine ihtiyaç duymadan pil ile çalışmaktadır. Z-Wave teknolojisini kullanan cihazlar sadece veri iletimi sırasında aktif oldukları için düşük güç ve enerji tüketimi sunarlar. Dünyada 35 milyonun üzerinde Z-Wave ürünü olduğu bilinmektedir (Paetz, 2018).

Z-Wave teknolojisi 1 GHz bandının altında ülkelere göre farklı frekanslarda çalışır. Örneğin Avrupa'da 868.4 MHz ve 869.85 MHz frekans bandında çalışırken, ABD'de 908.4 MHz ve 916 MHz frekans bandında çalışır. Bu frekanslarda yayın yaptığı için 2.4 GHz bandında yayın yapan Wi-Fi gibi protokoller ile çakışmaz. Ayrıca düşük bir frekansta yayın yapıldığından dolayı yayılan sinyaller duvar ve benzeri engellerden daha

az güç kaybederek geçerler. Z-Wave protokolü 100 kbps'e varan hızlarda haberleşme sağlar. Bunun yanında IPv6 ile uyumlu çalışabilmesi ve çoklu kanal desteği önemli avantajları arasındadır.

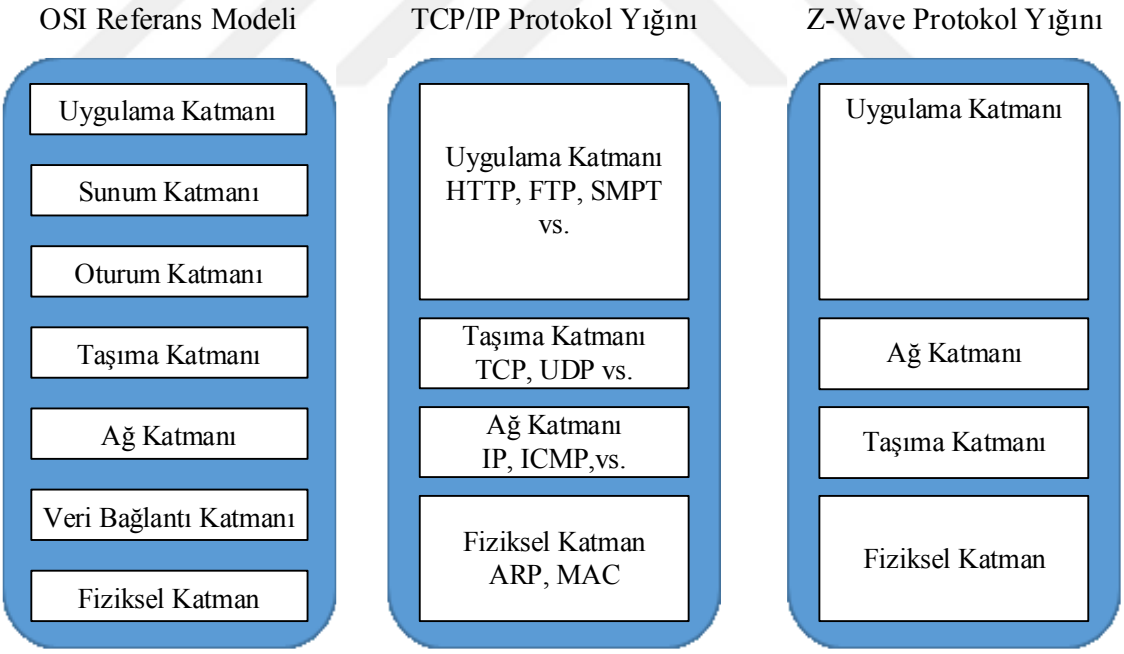
Z-Wave Topoloji olarak mesh ağı kullanmaktadır. Bu topoloji sayesinde kapsama alanı dışındaki iki düğüm birbirleri ile iletişime geçecekse, diğer düğümler üzerinden veri iletilerek hedef düğüm ile haberleşilir. Z-Wave ağı, denetleyicilerden ve slave cihazlardan oluşur. Basit bir Z-Wave ağ yapısı Şekil 3.13'te gösterilmiştir.

Denetleyici aygıtları, Z-Wave ağındaki kontrol komutlarını başlatan düğümlerdir ve komutları diğer düğümlere gönderirler. Slave cihazlar, alınan komuta dayanarak bu komutları yanıtlayan ve çalıştıran düğümlerdir. Slave düğümleri ayrıca komutları ağdaki diğer düğümlere iletir. Bu durum kapsam dışına veri iletilmesini de mümkün kılar. Denetleyici aygıtlar ağdaki yönlendirme tablosuna sahiptirler diğer bir değişle düğümlerin bağlantılarını ve koordinatlarını saklarlar. Dolayısı ile denetletici cihazlar ağın tüm düğümleri ile iletişim kurabilirler. Z-Wave ağını ilk olarak oluşturan denetleyici birincil denetleyici olur ve her ağda sadece bir tane birincil denetleyici mevcuttur. Bu aygıtların ağa düğüm dahil etme veya çıkarma gibi özellikleri vardır. Ayrıca birincil denetleyici ağdaki cihazların kimliklerini (ID) tahsis eder. Birincil denetleyici kullanılarak ağa dahil edilen diğer denetleyiciler ikincil denetleyici olarak adlandırılır. Bu aygıtların en önemli özelliği birincil denetleyicinin sahip olduğu yönlendirme tablosunun kopyasını barındırmasıdır. Z-Wave ağındaki slave cihazlar kendilerine gelen komutları alır ve bu komutları yerine getirirler. Eğer aksi bir komut gelmediyse doğrudan bağlı olduğu düğümlere komutu iletmezler. Bu düğümler yönlendirme tablolarını hesaplayamazlar sadece komutu diğer düğümlere iletirken kullanmak üzere yönlendirme tablolarını saklarlar.



Şekil 3.13. Z-Wave ağ yapısı

Z-Wave protokolü ağları birbirinden ayırmak için Home ID kullanır. Home ID, veri denetleyici cihazlar tarafından önceden bilinen 32-bitlik benzersiz bir değerdir. Ağdaki tüm cihazlar Home ID verisini bilmek zorundadır ve bu veri denetleyici cihaz tarafından tüm cihazlara iletilir. Ayrıca her düğüme ağ içinde benzersiz 8-bitlik Node ID verisi denetleyici düğüm tarafından verilir.



Şekil 3.14. OSI, TCP/IP ve Z-Wave protokol yığınının karşılaştırılması

Şekil 3.14'te verildiği gibi, Z-Wave protokol yığını fiziksel katman (PHY/MAC), taşıma katmanı, ağ katmanı ve uygulama katmanı olmak üzere 4 katmandan oluşur.

Fiziksel katman, modülasyon ve RF kanal atamasının yanı sıra vericide başlık (header) eklemeyi ve başlığı kullanan alıcıda senkronizasyonu üstlenirken aynı yerde tanımlı MAC katmanı HomeID ve NodeID'e bakar, çarpışma önleme algoritmasına ve geri çekme algoritmasına dayalı düğümler arasındaki ortamı kontrol eder. Taşıma katmanı, çerçevelerin iletimini ve alımını üstlenir, yeniden iletimi, ACK paketi iletimini ve doğrulama kontrolünü uygular. Ağ katmanı, paket yönlendirme, topoloji tarama ve yönlendirme tablosu güncellemeleri ile ilgilenir. Uygulama katmanı, alınan veya iletilen paketlerdeki uygulamaya yönelik görevlerden sorumludur.

Yeni model tüm Z-Wave cihazları 128 bit-AES şifreleme özelliğine sahiptir. Buna göre anahtar uzunluğu 128-bit ve ileti tazeliği (message freshness) 64-bit nonce[2] olarak belirlenmiştir. Şifreleme modu AES-OFB[3] ve doğrulama modu ise AES- CBC MAC'tir. Z-Wave veri kaynaklı kimlik doğrulama, şifre bloğu zincirleme mesaj doğrulama koduna, yani CBC-MAC tekniğine dayanır. Bu teknik, AES gibi bir blok şifreleme algoritmasından gelen bir mesaj doğrulama kodunu, yani MAC değerini hesaplar.

3.9. NFC

NFC (Yakın Alan İletişimi), telefon, tablet, dizüstü bilgisayarlar ve diğer cihazların NFC donanımlı diğer cihazlarla kısa mesafede kolayca veri paylaşmasına olanak tanır. RFID teknolojisi üzerinden geliştirilmiştir. İletişim mesafesi 10 cm'ye kadardır. NFC 13.56 MHz ISM bandında çalışmaktadır ve 106 kbs, 212 kbs ve 424 kbs gibi farklı veri iletim hızlarını destekler (Coskun, Ok, & Ozdenizci, 2012). NFC protokolünü kullanan ürünler kolay kullanım biçimi sebebiyle aşağıdaki gibi birçok uygulamada kullanılmaktadır:

- Mobil ödeme: Biletler veya taksi yolculukları için NFC telefonlarıyla ödeme yapma, temassız POS cihazlarında NFC destekli telefonlarıyla ödeme yapma, NFC telefonlarda fiş kaydetme gibi.
- Kimlik doğrulama, erişim kontrolü: NFC telefonlarda elektronik anahtarlar saklama, güvenli bina erişimi, güvenli PC oturum açma, araba kapısının

[2] Kriptografide sadece bir kerelik kullanılan sayı anlamına gelir (number used once) ve replay saldırılarından kaçınmak için kullanılır.
[3] Output Feedback Mod

kilidini açma, evdeki bazı akıllı cihazları NFC destekli telefon üzerinden ayarlama gibi.

- Akıllı telefonlar, dijital kameralar, dizüstü bilgisayarlar gibi NFC birimleri arasındaki veri aktarımı: elektronik kartvizit alışverişi, fotoğraf makinesini yazıcıya yakın tutarak fotoğrafları basabilme, başka bir cihazı ya da servisi açma veya kapatma gibi.
- Dijital bilgilere erişim: akıllı posterdeki programları NFC destekli telefona okutma, NFC telefonuna haritaları indirme gibi.

NFC iletişim sisteminde okuyucu veya yazıcı bir cihaz ve etiket bulunmaktadır. Etiket, anten ve az miktarda bellek içeren manyetik alanla çalışan pasif bir cihazdır. Etiket belleği salt okunur, yeniden yazılabilir veya bir kez yazılabilir şekilde olabilir. Okuyucu, etiket ile iletişim kurmak için radyo sinyalleri üreten aktif bir cihazdır. Okuyucu, pasif iletişim modu durumunda pasif cihaza güç sağlar ve verinin okunabilmesini sağlar.

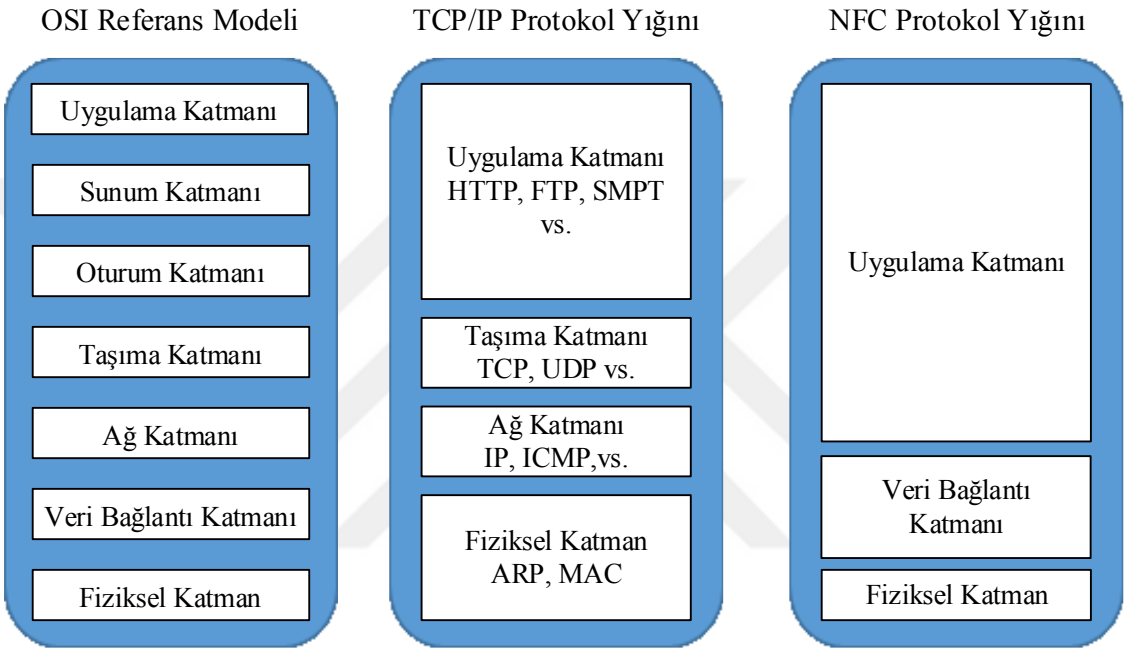
NFC cihazları aktif ve pasif olmak üzere iki iletişim modelini destekler. Aktif modda, hedef ve başlatıcı cihazların güç kaynakları vardır ve alternatif sinyal iletimi ile birbirleriyle iletişim kurabilirler. Pasif modda ise, başlatıcı cihaz radyo sinyalleri üretir ve hedef cihaz bu elektromanyetik alandan güç alır. Hedef cihaz, mevcut elektromanyetik alanı modüle ederek başlatıcıya yanıt verir.

NFC cihazları, ISO/IEC 18092, ISO/IEC 14443 ve NFC IP-1 temassız kart standartlarına göre üç farklı modda çalışabilir. Okuma/yazma modunda, NFC özellikli telefon, standart NFC veri biçiminde desteklenen etiket türlerinden herhangi birini okuyabilir veya bunlara veri yazabilir. Bire bir modunda, NFC özellikli iki cihaz veri alışverişi yapabilir. Kart emülasyon modunda, okuyucu bir cihaz mevcut okuyucular için etiket veya temassız kart görevi görebilir (Want, 2011).

NFC teknolojisi içinde 4 etiket tipi mevcuttur ve özellikleri aşağıda verilmiştir:

- Tip 1: Bu etiket tipi maliyet açısından oldukça uygundur ve birçok NFC uygulaması için idealdir. Okuma ve yeniden yazma özelliği olan etiketler istenildiği takdirde salt okunur olarak ayarlanabilir. Genellikle 96 bayt belleğe sahiptirler ve 2 KB'ye kadar genişletilebilir. İletişim hızı 106 kbps olmakla beraber veri çalışma koruması bulunmamaktadır.

- Tip 2: Tip 1 etiketleri ile aynı özelliklere sahiptir. En önemli farkı veri çakışma korumasının bulunmasıdır.
- Tip 3: Tip 1 ve 2 etiketlerinin genel özelliklerine sahiptir. Farklı olarak tip 3 etiketler 1 MB veri saklayabilir ve 212 veya 424 kbps iletişim hızını destekler.
- Tip 4: Tip 1 ve 2 etiketlerinin genel özelliklerine sahiptir. Farklı olarak 32 KB belleği mevcuttur ve 106, 212 veya 424 kbps hızlarında çalışabilir.



Şekil 3.15. NFC protokol yığını

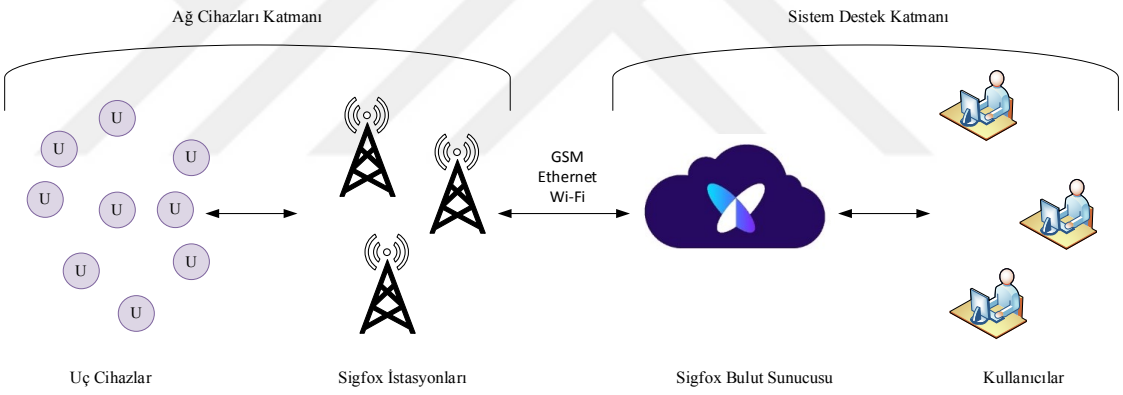
Şekil 3.15'de P2P modunda basit NFC aygıtı protokol yığınının OSI referans modeli ve TCP/IP protokol yığını ile eşleşimi gösterilmektedir. Görüleceği gibi NFC protokol yığını uygulama katmanı, veri bağlantı katmanı ve fiziksel katmandan oluşur. Uygulama katmanı, NFC cihazları arasında veya NFC cihazı ile etiketler arasında değiştirilecek verilerin formatını belirler. Veri bağlantı katmanı farklı çalışma modları ve çakışma önleme mekanizması ile ilgilenirken, fiziksel katman, frekans ve güç gibi modülasyon, kodlama ve RF ile ilgili parametrelerle ilgilenir.

NFC protokolüne sahip cihazlar AES 128-bit şifreleme üzerinden veri aktarımı yapılması önerilmektedir. Fakat belirgin bir ağ olmadığından dolayı saldırılara açık bir

ortam meydana gelir. Güvensiz olarak tasarlanmış NFC teknolojisinde veri bozulması, dinleme, araya girme saldırıları ve hırsızlık gibi saldırılar yapılabilmektedir.

3.10. Sigfox

Sigfox, 2009 yılında kurulan ve elektrik sayaçları, akıllı saatler ve çamaşır makineleri gibi düşük enerjili nesneleri sürekli olarak açık tutmak ve az miktarda veri yaymak için kablosuz ağlar kuran bir Fransız şirketidir. Şirketin ürettiği kendi adını taşıyan sistemi, uzaktaki cihazların UNB (Ultra Dar Bant) ile erişim noktasına bağlanmasını sağlayan hücresel sistemin bir varyasyonudur. Sigfox, nesnelerin interneti ağının gereksinimlerini karşılayacak şekilde tasarlamıştır. Uzun pil ömrü, düşük cihaz maliyeti, düşük bağlantı ücreti, yüksek ağ kapasitesi ve uzun menzil ile özellikle akıllı tarım ve akıllı şehir gibi geniş alan ağları şeklinde tasarlanacak olan uygulamaların performans ve verimliliğini artırmak için ideal bir çözümdür (Vejlgaard vd., 2017).

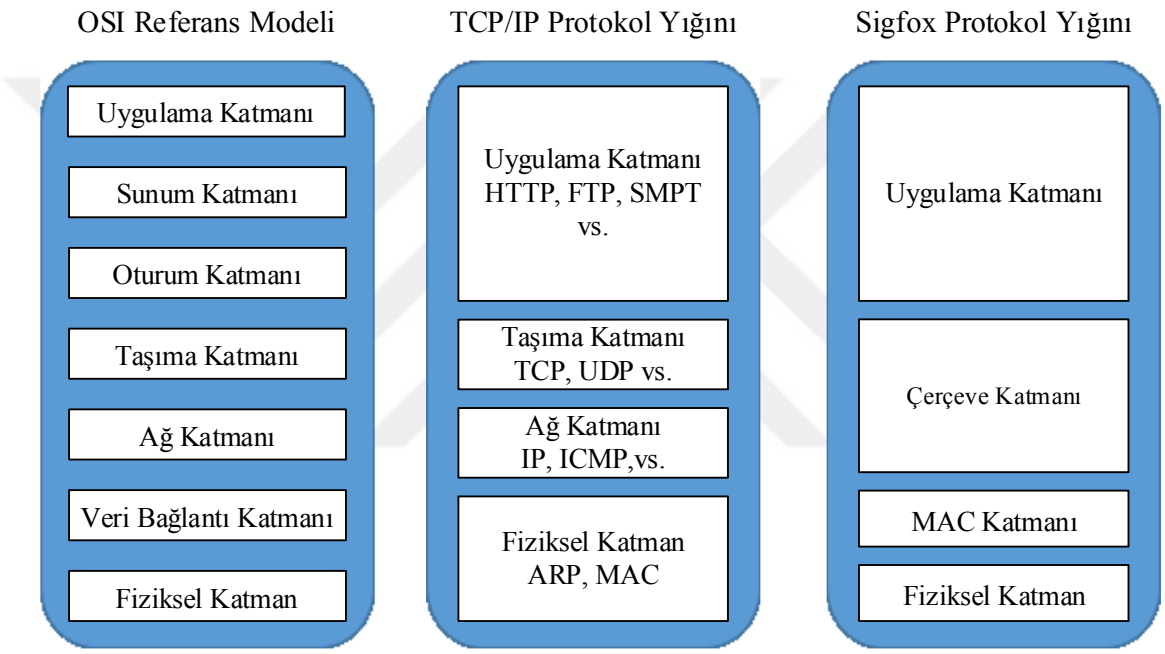


Şekil 3.16. Sigfox ağ yapısı

Şekil 3.16'da Sigfox ağ yapısı gösterilmiştir. Sigfox ağı, ağ cihazları katmanı ve sistem destek katmanı olmak üzere iki katmandan oluşur. Bu iki katman arasında GSM, Wi-Fi veya Ethernet gibi standart bir ağ kullanılır. Ağ cihazları katmanındaki uç cihazlar sensör verilerini Sigfox istasyonlarına iletir. İstasyonlar internet ağı kullanarak verileri Sigfox bulut sunucularına iletir. Bu bölüme sistem destek katmanı denir. Bu aşamadan sonra işlenen ve anlamlandırılan veriler kullanıcılara iletilir.

Sigfox, 868 MHz frekans ISM bandında çalışır ve spektrum 100 Hz'lik 400 kanala bölünmüştür. Frekans verimi, iletim sırasında çok fazla enerji tüketmemesine ve daha

fazla menzil elde etmesine imkan verir. Sigfox yıldız topolojisine dayalıdır ve oluşturulan trafiği taşımak için bir mobil operatör gerektirir. Her düğüm cihaz, 100 bps'e kadar bir veri hızında 12 kutuplu yük taşıma kapasitesi ile günde 140 mesaj gönderebilir. Sigfox ağı, teorik olarak her erişim noktasının kırsal alanlarda 30-50 km ve kentsel alanlarda 3-10 km kapsama alanında bir milyon son cihaza kadar hizmet verecek şekilde planlanmıştır. Sigfox frekans bandının çalışma döngüsü maksimum % 1'dir ve cihazının her saatte sadece 36 saniye veri iletmesine izin verir. Bu da Sigfox paketleri için 4, 8 veya 12 baytlık büyüklük ile saatte maksimum 6 adet iletim anlamına gelir.



Şekil 3.17. Sigfox katmanları

Şekil 3.17'de gösterildiği gibi Sigfox, fiziksel katman, MAC katmanı, çerçeve katmanı ve uygulama katmanından oluşmaktadır. Bu katmanların her birinin fonksiyonları aşağıda belirtilmiştir.

- Fiziksel katman, Sigfox sinyallerinin nasıl modüle edileceğini belirler. Bu katmanda başlık ekleme (gönderim sonunda) ve kaldırma (alma sonunda) işlemlerini gerçekleştirir. Yukarı bağlantı için BPSK[4] modülasyonunu, aşağı

[4] BPSK: Lojik işaret 0 ise sinüsün fazını 180 derece kaydıran, 1 ise aynen gönderen sayısal işaret gönderim anahtarlama yöntemi.

bağlantı için ise GFSK modülasyonunu kullanır. Sistemin kurulduğu bölgeye bağlı olarak bit hızı 100 bps veya 600 bps'dir. Aktarım gücü, Avrupa ülkelerinde maksimum 14 dBm, Kuzey ve Güney Amerika'da ise 22 dBm'ye ulaşır. Çalışma frekansları da bu katman tarafından yönetilir.

- MAC katmanı, iletilecek veriye cihazın tanımlanması için gerekli alanların yanı sıra hata algılama kodu gibi diğer standart parametreleri ekler. Paketleri, protokolde tanımlanan biçimde gönderi ve alım biçimlerine göre hazırlar.
- Çerçeve katmanı, uygulama katmanı tarafından gönderilen veriler alır ve TCP/IP protokol kümesindeki taşıma ve ağ katmanlarındaki göreve benzer bir şekilde bir sıra numarası ekleyerek sistematik olarak alt katmana gönderir.
- Uygulama katmanı, Sigfox teknolojisinde farklı uygulamaları desteklemektedir. WAN ile sunucular arasında iletimi desteklemek için SNMP, HTTP, MQTT, IPv6 gibi çeşitli arabirimler/protokoller bu katmanda tanımlıdır.

Güvenlik, Sigfox için son derece önemlidir. Bağlı cihazlar ve Sigfox bulutu arasında gizli anahtar temelli uçtan uca kimlik doğrulama yöntemi bulunur. Bu gizli anahtar, salt okunur bir bellekte saklanır. Gizli anahtar ile bir imza üretilir. Bu imza kullanılarak gelen mesajın doğru kaynaktan gelip gelmediği test edilir. Veriler iletilirken her mesaj 3 farklı frekansta 3 kez ve rasgele gönderildiğinden, radyo paketleri koklama (sniffing) saldırısı karşısında korunur çünkü çalışma bandında cihazın nereye veri ileteceğini bilmek mümkün değildir. Sigfox baz istasyonları, şifrelenmiş bir VPN (Sanal Özel Ağ) kullanarak noktadan noktaya bağlantı yoluyla Sigfox bulutu ile bağlanır. Baz istasyonları ile Sigfox bulutu arasındaki iletişim AES 128-bit kullanılarak yapılır. Ağın sonunda müşteriler, IT platformları, web sitesi, API'ler haberleşmek için HTTPS şifreli ara yüzleri kullanarak Sigfox bulutuna bağlanır.

3.11. Sonuçlar

Nesnelerin interneti uygulamalarında kullanılmak üzere birçok protokol üretilmiştir. Çizelge 3.2'de tez kapsamında araştırılan protokoller frekans, mesafe ve veri transfer hızlarına göre özetlenmiştir. Uygulama tipine, nesnelerin aralarındaki mesafeye ve iletim gücüne bağlı olarak protokol seçimi yapılmalıdır. Kriptografik güvenlik

açısından bakıldığında protokollerde genellikle güvenliğin AES algoritması kullanılarak sağlandığı gözlemlenmiştir.

Çizelge 3.2. Protokollerin karşılaştırılması

Protokol	**Frekans**	**Mesafe**	**Veri Transfer Hızı**
Bluetooth	2.4 GHz	50-150 m	1 mbps
ZigBee	2.4 GHz	10-100 m	250 kbps
Z-Wave	900 MHz	30 m	9.6/40/100 kbps
6LoWPAN	2.4 GHz	10-100 m	250 kbps
NFC	13.56 MHz	10 cm	100-420 kbps
Sigfox	900 MHz	30-50 km kırsalda, 3-10 km şehir içi	10-1000 bps
LoRaWAN	Değişken	2-5 km, 15 km	0,3-50 kbps

BÖLÜM 4

4. NESNELERİN İNTERNETİ UYGULAMALARINDAKİ GÜVENLİK İHLALLERİ VE TEHDİTLER

4.1. Giriş

Günümüzde nesnelerin interneti teknolojileri ile hayatın hemen hemen her alanında kullanılan birçok ürün geliştirilmiştir. Fakat geliştirilen ürünlerin büyük çoğunluğunda veri güvenliği ön planda değildir. Unit 42 grubunun hazırladığı rapora göre tüm dünyada nesnelerin interneti cihaz trafiğinin %98'i şifrelenmemiş olarak ağda iletilmektedir. Ayrıca bu cihazlarının %57'si orta veya yüksek şiddetli saldırılara karşı savunmasızdır (Unit42, 2020). Bu durum saldırganların dikkatini nesnelerin interneti cihazlarına ve uygulamalarına yönlendirmektedir.

Çalışmanın bu bölümünde nesnelerin interneti cihazlarının hedef olarak seçilme nedenleri tartışılmış ve günümüze kadar meydana gelen önemli siber güvenlik saldırıları verilmiştir. Bu saldırılar nesnelerin interneti mimarilerine göre incelenerek her katmanda meydana gelebilecek olan tehditler sınıflandırılmıştır.

4.2. Nesnelerin İnterneti Cihazlarının Hedef Olarak Seçilme Nedenleri

Nesnelerin interneti uygulamaları için üretilen cihazlar birçok avantaj sağlamasının yanı sıra doğaları gereği birçok dezavantajı da barındırır. Bu dezavantajlar özellikle saldırganların ilgisini çekmektedir. Aşağıda bu cihazların saldırganlar tarafından hedef olarak seçilmesinin bazı nedenlerine değinilmiştir.

- Cihaz sayısının fazlalığı: Çalışmanın başında da bahsedildiği gibi nesnelerin interneti alanında kullanılan cihazların sayısı dizüstü veya masaüstü bilgisayar

sayısının oldukça üstündedir. Saldırı yüzey alanının büyük olması saldırganlar açısından ilgi çekmektedir. Çünkü saldırgan açısından bakıldığında ne kadar fazla cihaz var ise o kadar giriş noktası yakalanabilir.

- Kaynak kullanımının kısıtlı olması: Nesnelerin interneti ortamı içinde kullanılan cihazlar sadece görevlerini gerçekleştirecek ölçüde kaynak kullanmak için üretilmişlerdir. Dolayısı ile bilgisayarlara uygulanan güvenlik duvarı, anti-virüs yazılımı gibi çeşitli güvenlik tedbirlerinin bu tip cihazlara uygulanması mümkün değildir. Bu durum saldırganlar tarafından bu cihazların kolay hedef olabileceği öngörüsünü doğurmaktadır.
- Üreticilerin çalışabilirliği güvenliğin önüne alması: Nesnelerin interneti kapsamındaki cihazları üreten şirketler öncelikli olarak sistemin sağlıklı bir şekilde çalışmasını amaçlamaktadırlar. Bu üreticiler genellikle güvenlik sorunları çıktıkça yeni ürünler ortaya çıkarmaya çalışmaktadırlar. Genellikle üretilen ilk ürünler piyasadaki ilk ve en çok kullanılan ürün olma hedefinden dolayı zayıf güvenlik tedbirleri ile piyasaya sunulmaktadır.
- Kişisel bilgilerin toplanması: Özellikle son kullanıcıya yönelik olan nesnelerin interneti ürünleri büyük oranda kişisel bilgi toplamakta ve kaydetmektedir. Örneğin sağlık uygulamaları, ev otomasyonları bu kategoride sayılabilir. Elbette kişilerin mahrem bilgilerini bir sistemde saklamak saldırganların ilgisini çekmekte ve bu konudaki saldırıların artmasına neden olmaktadır.
- Yazılım güncellemelerinin zamanında verilmemesi: Üreticiler sadece bir sorun olduğunda veya üst düzey bir yenilik olduğunda güncelleme yayınlamaktadırlar. Fakat bu güncellemelerin tüm sistemlere uygulanamaması veya hiç uygulanmayıp bunun yerine yeni ürün almaya teşvik etmek gibi durumlarda yaşanan gecikmeler özellikle güvenlik açısından sıkıntılar yaratmaktadır.
- Üretici firmanın arka kapı bırakması: Üretici firmalar cihazlara uzaktan müdahale etmek için sadece kendileri tarafından bilinen arka kapı denilen çeşitli erişim yöntemleri kullanmaktadır. Bu yöntemin saldırganlar tarafından tespit edilmesi durumunda tüm cihazlar savunmasız kalabilmektedir.
- Varsayılan kullanıcı adları ve şifreleri: Çoğu cihaz, yönetici ara yüzüne bağlanmak için ilk kurulum esnasında fabrikada verilen varsayılan kullanıcı

adı ve şifre ile piyasaya sürülmektedir. Bu şekilde varsayılan ayarları değiştirmeden kullanılan cihazlar rahatlıkla ele geçirilebilmektedir.

4.3. Nesnelerin İnterneti Cihazlarına Yapılan Bazı Saldırılar

Yukarıda bahsedilen nedenlerden dolayı nesnelerin interneti uygulamalarında kullanılan cihazlar saldırganlar açısından kolay lokma olarak görülmekte ve birçok saldırıya maruz kalmaktadır. Nesnelerin interneti alanı içerisinde meydana gelen her bir güvenlik ihlalinin bilinmesi mümkün değildir. Çoğu üretici güvenlik ihlallerini saklamakta veya hiçbir şekilde tespit edememektedir. Aşağıda nesnelerin interneti ile ilgili bazı güvenlik ihlalleri tarihsel olarak sıralanmıştır.

- Avusturalya'da 2000 yılında Vitek Boden adındaki kişi iş başvurusu reddedildiği için eyalet kanalizasyon arıtma tesisine uzaktan siber saldırı gerçekleştirmiştir. Boden, saldırı sonucunda 800.000 litrelik sıvı atığın şehir parklarına, nehirlere ve lüks bir otelin bahçesine akmasına neden olmuştur (Abrams & Weiss, 2008).
- Kuzey Amerika'da faaliyet gösteren CSX demiryolu şirketinin bilgisayarlarına "Sobig" adlı bir virüs tren yolunun sinyalizasyon sistemlerini devre dışı bıraktığı için tren seferlerinde ciddi gecikmelerin yaşanmasına sebep olmuştur (Niland, 2003).
- 2008 yılında ABD'de bir dizi internet solucanı, Daimler Chrysler'in 13 otomobil üretim tesisini yaklaşık 1 saat boyunca çevrimdışı kalmasını sağlamış ve 50.000'den fazla montaj hattı kesintiye uğratmıştır (Roberts, 2008).
- Yine 2008 yılında Polonya'nın Lodz kentinde 14 yaşındaki bir çocuk ev yapımı bir verici sistemi ile şehrin tramvay sistemine saldırması sonucunda bazı trenlerin kontrolünü ele alarak 12 kişinin yaralanmasına sebep olmuştur (Baker, 2008).
- 2010 yılında, 15'den fazla İran enerji tesisi Stuxnet adlı internet solucanı tarafından saldırıya uğramış ve tahmini olarak 984 uranyum zenginleştirici santrifüj tahrip edilmiştir. Uzmanlar bu saldırının uranyum zenginleştirme verimliliğinde %30'luk bir düşüş oluşturduğunu tahmin etmektedirler. Saldırının bir işçinin USB sürücüsü tarafından başlatıldığına inanılmaktadır.

PLC (Programlanabilir Mantıksal Denetleyici) cihazlara saldıran Stuxnet'in amacının nükleer santrali sabote etmek olduğu söylenmiştir. Bu saldırının bilişim savaşı tarihinde bir dönüm noktası olduğu bilinmektedir (Farwell & Rohozinski, 2011).

- 2011 yılında ABD'de bir su dağıtım şirketinin SCADA[5] (Merkezi Denetim ve Veri Toplama) sistemleri Rusya kökenli bir internet servis sağlayıcı üzerinden saldırıya uğramıştır. Saldırganların şirketin veritabanına yetkisiz erişim sağladığı ve su ilçe sistemleri de dahil olmak üzere çeşitli SCADA sistemlerinin kullanıcı adlarını ve şifrelerini ele geçirdiği bilinmektedir (Keszthelyi, 2014).
- 2012 yılında ABD elektrik dağıtım şirketine yapılan saldırılarda milyonlarca dolar zarara neden olduğu belirlenmiştir (Cárdenas, Amin, & Sastry, 2008).
- 2015 yılında saldırganlar iklimlendirme alanında çalışan büyük bir şirkete nüfuz etmek için kötü amaçlı yazılım kullanmış ve 40 milyondan fazla müşterinin kredi kartı ve bankamatik kartı bilgileri de dahil olmak üzere 70 milyondan fazla müşteriye ait kişisel verileri elde etmişlerdir (Krebs, 2014).
- 2015 yılında ünlü bir araba üreticisi 1.4 milyon aracını geri çağırmıştır. Yapılan açıklamada araçların otomatik kontrol yazılımında bir açık bulunduğunu ve yapılan örnek bir senaryoda uzaktan bağlantı ile kontrolün ele geçirebileceğinin kanıtlandığı belirtilmiştir (Greenberg, 2015).
- 2016 yılının Haziran ayında, güvenlik alanında çalışan Sucuri firmasının güvenlik uzmanları, DDoS[6] (Dağıtık Servis Dışı Bırakma) saldırıları başlatmak için kullanılabilecek 25.000'den fazla CCTV[7] (Kapalı Devre Televizyon) cihazından oluşan bir botnet keşfettiklerini bildirmişlerdir. Sucuri, kötü amaçlı botnet'in IP adreslerini dünyanın 105 ülkesine kayıtlı olduğunu belirlemiştir (Bawany, Shamsi, & Salah, 2017).

[5] Kapsamlı ve entegre bir veri tabanlı kontrol ve izleme sistemi olan SCADA ile bir tesise veya işletmeye ait tüm ekipmanların kontrolünden üretim planlamasına, çevre kontrol ünitelerinden yardımcı işletmelere kadar tüm birimlerin otomatik kontrolü, gözetlenmesi ve sonuçların raporlanması sağlanabilir.

[6] Farklı yerlerden aynı anda gerçekleştirilen DoS saldırılarıdır.

[7] Televizyon sistemleri, belirli bir merkezden alınan ses ve görüntü sinyalinin çeşitli alıcı çeşitleri ile birçok kişiye aktarılmasıdır. CCTV ise izleme yapılacak olan bölgelere yerleştirilen kameralardan alınan görüntülerin lokal bir bölgede, herkes tarafından erişimin olmadığı sadece yetkili kişilerin izlemesine izin veren bir sistemdir.

- Eylül 2016 tarihinde varsayılan kullanıcı adı ve şifreye sahip nesnelerin interneti cihazlarını ele geçiren NyaDrop adındaki kötücül yazım tespit edildi. NyaDrop analiz edildiğinde saldırının Rus IP adreslerinden geldiği tespit edilmiştir. Yazılım cihaza bulaştığında uzak ağa ulaşmak için bir kapı açıp kontrol için gerekli diğer yazılımların yüklenmesini sağlamaktadır. NyaDrop, yalnızca DVR'ler, CCTV kameraları, yönlendiriciler ve diğer gömülü sistemler gibi cihazları hedef almıştır. Burada muhtemel amaç cihazların bir botnet'e katılması olduğu düşünülmektedir. (Beltov, 2016)
- 22 Eylül 2016 tarihinde güvenlik bilgilerinin yayınlandığı "Krebs on Security" sitesi DDoS saldırısına maruz kalarak hizmet veremez hale gelmiştir. Saniyede yaklaşık 620 gbps veri iletimi ile saldırının gerçekleştirildiği bildirilmiştir. Aynı gün Fransız internet sağlayıcısı OVH'ye de benzer bir DDoS saldırısı düzenlenmiştir. Saniyede 1 Tbps veri iletimi ile gerçekleşen saldırının bilinen en büyük DDoS saldırısı olduğu kabul edilmektedir. Saldırılarda 150.000 adetin üzerinde CCTV cihazın ve dijital video kaydedicisinin kullanıldığı bilinmektedir. Bu ve benzeri saldırılar ele geçirilen nesnelerin interneti cihazlarının ne kadar tehlikeli olabileceğinin örnekleridir (Krebs, 2016).
- 21 Ekim 2016 tarihinde tüm dünyada etkin olan DDoS saldırılar paypal, amazon, twitter, github, reddit gibi birçok web sistemine girişi zorlaştırmış veya engellemiştir. Bu saldırılarda kullanılan Mirai botnet'i tamamıyla nesnelerin interneti cihazlarından oluşmuştur. Japoncada gelecek anlamına gelen Mirai botnet'i saldırı esnasında saatte 4000 cihaza bulaşmıştır ve şuan 500.000'dan fazla cihazı kontrol ettiği tahmin edilmektedir. Mirai botnet, 148.000 nesnelerin interneti cihazıyla rekor kıran 1 Tbps DDoS saldırılarında kullanılmakla ünlüdür. Çoğunlukla içinde 164 farklı ülkeden CCTV kameraları, DVR cihazları ve ev yönlendiricilerini barındırmaktadır (Dobbins, 2016).
- Kaspersky güvenlik şirketinin yayınladığı rapora göre Ekim 2016 tarihinde 300.000 üzerinde cihaza sızmış Hajime botnet'i farkedildi. Kötü amaçlı bir yazılım sayesine özellikle başka botnet'lerdeki cihazları da hedef alarak kendi ağına katmaya çalışan Hajime için Mirai botnet ile cihazlarının kontrolü için

aralarında bir savaş olduğu söylenebilir. Her geçen gün büyüyen botnet'in amacı henüz bilinmemektedir. (Millman, 2017)

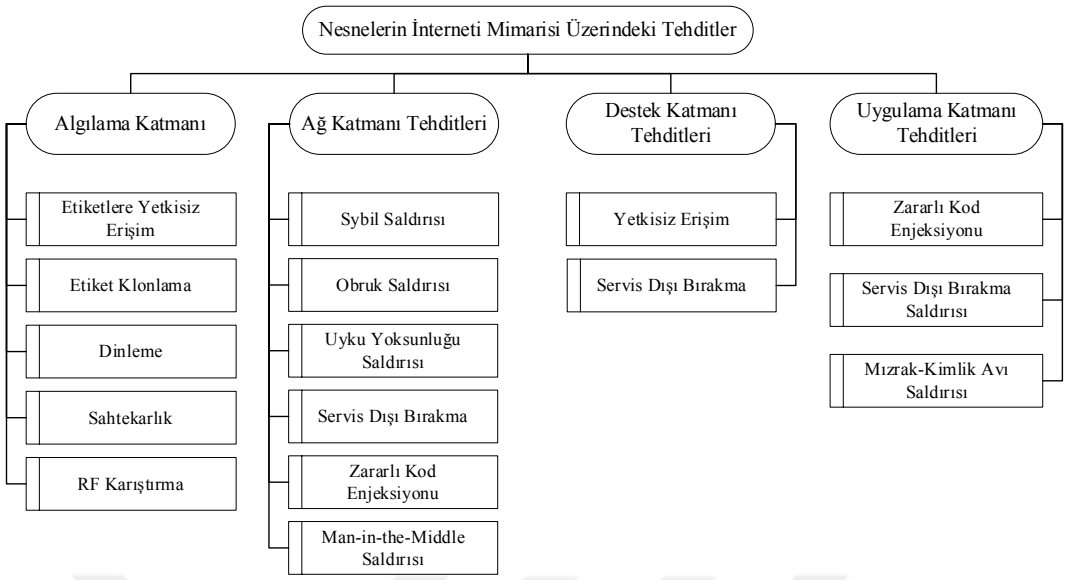
- 2017 Nisan ayında Unit 42 araştırmacıları Amnesia adında Tsunami botnet'inin yeni bir varyantını belirlediler. TVT Digital tarafından üretilen ve dünya çapında 70'den fazla satıcı tarafından markalanan DVR cihazlarda şirketin kullandığı ve kamuya açıklamadığı bir uzaktan bağlanmak için kullanılan güvenlik açığını hedeflemektedir. Amnesia botnet ağına tüm dünyada aralarında Tayvan, ABD, İsrail, Türkiye ve Hindistan'ın bulunduğu yaklaşık 227.000 cihaz katılmıştır. Günümüzde bu sayının çok fazla olduğu öngörülmektedir. (Xiao & Zheng, 2017)
- 2017 Mayıs ayında Persirai adında yeni bir botnet belirlendi. Botnet çeşitli Orijinal Ekipman Üreticisi (OEM) ürünlerine dayanan 1000'den fazla IP Kamera modelini hedef almaktadır. Trend Micro firması Persirai botnet ağına dahil olan 120.000 kamera tespit etmiştir (Yeh, Chiu, & Lu, 2017).
- 2017 Haziran ayında araştırmacılar sektörde önemli pay sahibi olan Fitbit adındaki akıllı bilekliklerden kişisel verilerin çalınabileceğini gösteren bir çalışma yayınladı. Bilekliğe ortadaki adam saldırısı düzenleyerek verilerin analiz için gönderildiği bulut sunuculara aktarılmadan yakalanmasını sağladılar. (Fereidooni vd., 2017)
- 2017 Ekim ayında Mirai botnet'den daha karışık olabileceği belirtilen Reaper botnet ağı tespit edildi. Mirai ağının tersine Reaper, güvenlik açığı tespit edilememiş popüler yönlendirici markalarını, IP kameraları ve ağa bağlı depolama cihazlarını hedef almaktadır (Greenberg, 2017).
- 2018 Mart ayında Suudi Arabistan Petrol Şirketinin nesnelerin interneti tabanlı endüstriyel güvenlik sistemlerine bilgisayar korsanları tarafından saldırı düzenlendi (Perlroth & Krauss, 2018).
- 2019 Mart ayında Amazon'un popüler Ring adlı video kapı zilinde, istenmeyen ev gözetimine olanak tanıyan bir güvenlik açığı tespit edildi. Bu açık ile bilgisayar korsanlarının şifrelenmemiş ses ve video kayıtlarına erişmelerinin mümkün olduğu gösterildi. (Halon, 2019)
- 2019 Haziran ayında güvenlik araştırmacıları, cihazların firma yazılımını hızla silerek yayılan Silex adlı yeni bir kötü amaçlı yazılımı fark ettiler. Yalnızca

bir gün çalışan kötü amaçlı yazılım, binlerce cihazı etkisiz getirmeyi başardı. Kötü amaçlı yazılımın hatalı kodlanması sonucunda yazılım daha fazla yayılamadı. Güvenlik araştırmacısı Ankit Anubhav'a göre, Silex bir Avrupa ülkesinden üç genç tarafından yaratıldı (TrendLabs, 2019).

- 2020 Şubat ayında Philips Hue akıllı ampullerinin diğer cihazlarla olan iletişimi esnasında meydana gelebilecek bir güvenlik açığı belirlendi. Bu güvenlik açığını kullanan saldırganın tüm ev ağı üzerinde kontrol sahibi olmasına izin verebileceği açıklandı. (Owen, 2020)
- 2020 Mayıs ayında Nozomi Networks Labs'deki güvenlik araştırmacıları, nesnelerin interneti cihazlarından oluşan Dark Nexus adlı botnet'in çeşitli varyantlarını tespit etti (Pinto, 2020).
- 2020 Mayıs ayında ortaya çıkan Kaiji botnet ağını keşfeden araştırmacılara göre, kötü amaçlı yazılım, botnet ağını oluşturmak amacıyla mümkün olduğunca çok sayıda nesnelerin interneti cihazına ve Linux sunucusuna bulaşacak şekilde tasarlanmıştır. SSH kaba kuvvet saldırıları üzerinden cihazları ele geçirmeye çalışan Kaiji özellikle internet üzerinden herkese açık olarak SSH bağlantı noktalarına sahip zayıf parolaları olan cihazları hedeflemektedir (SecureMac, 2020).

4.4. Nesnelerin İnterneti Mimarisi Üzerindeki Tehditler

Önceki bölümde nesnelerin interneti cihazlarına saldırganların ilgisi ve yapılan saldırılar anlatılmıştı. Bu kapsamda yapılan araştırmaya göre nesnelerin interneti uygulamalarına karşı olan tehditler mimarilere göre Şekil 4.1'deki gibi sınıflandırılmıştır. Meydana gelen veya gelebilecek olan saldırı çeşitleri katmanlara göre ayrılarak belirlenmiştir. Belirlenen saldırıların bazıları geleneksel bilgisayar ağlarına yapılan saldırılar olmasına rağmen işlevleri nesnelerin interneti uygulamalarına göre yorumlanmıştır.



Şekil 4.1. Nesnelerin interneti mimarisi üzerindeki tehditler

4.4.1. Algılama Katmanı Tehditleri

Algılama katmanı, RFID gibi farklı sensör teknolojilerinden oluşur. Bu teknolojiler aşağıda verilen birçok türde tehdide maruz kalır.

- Etiketlere Yetkisiz Erişim: Çok sayıda RFID sisteminde uygun kimlik doğrulama mekanizmasının bulunmaması nedeniyle, etiketlere yetkisiz kişiler tarafından erişilebilir. Saldırgan sadece veriyi okumakla kalmaz ayrıca verileri değiştirilebilir veya silebilir (Uttarkar & Kulkarni).
- Etiket Klonlama: Kolay ve basit bir şekilde okunabilen etiketlerin rahatlıkla kopyaları üretilebilir. Böylelikle orijinalinden ayırt edilemeyen bir kopya oluşturulabilir (Uttarkar & Kulkarni).
- Dinleme (Eavesdropping): RFID gibi sistemler kablosuz özellikleri nedeniyle saldırganların açık hedefi halindedir. Okuyucuya kablosuz olarak gönderilen veriler çoğunlukla şifrelenmemiştir. Bu veriler rahatlıkla saldırganlar tarafından dinleme yoluyla elde edilebilir (Khoo, 2011).
- Sahtekarlık (Spoofing): Bir saldırganın RFID sistemlerine sahte bilgi yayınlaması ve sanki doğru kaynaktan veri geliyormuş gibi sistemi aldatmasıdır. Bu şekilde saldırgan sisteme tam erişim ile girerek onu savunmasız hale getirebilir (Mitrokotsa, Rieback, & Tanenbaum, 2010).

- RF Karıştırma (RF Jamming): RFID etiketlerindeki RF sinyal iletişiminin fazla miktarda gürültü sinyali ile bozulmasıdır. Bir DoS (Servis Dışı Bırakma) atağı türü olarak nitelendirilebilir (Li, 2012).

4.4.2. Ağ Katmanı Tehditleri

Ağ katmanı, veriyi sensörden hedefe doğru güvenli bir şekilde ileten kablosuz algılayıcı ağından oluşur. Ağ katmanı ile ilgili güvenlik tehditleri aşağıda verilmiştir.

- Sybil Saldırısı: Sybil, saldırganın ilgili düğümü birden fazla kimlik bilgisi sunmak üzere manipüle ettiği bir saldırı türüdür. Genellikle ağ hakkında yanlış bilgi vermek ve ağdaki diğer düğümlerin düzenini karıştırmak için kullanılır (Douceur, 2002).
- Obruk (Sinkhole) Saldırısı: Ele geçirilen bir düğümü aktif hale getirilerek komşu düğümlerin bu düğüme bilgi gönderilmesinin zorlandığı saldırı tipidir. Bu şekilde ağdaki paket akışı tek bir yöne kaydırılarak tüm paketlere ulaşılabilir. Diğer düğümler, güvenli bir düğüme veri aktarıyormuş zannetseler de aslında ağ verileri saldırgana aktarılmış olur. Ele geçirilen düğüm sürekli aktif olduğundan dolayı düğümlerin dolayısı ile nesnelerin enerji tüketimi artacaktır (Ahmed, Kanhere, & Jha, 2005).
- Uyku Yoksunluğu (Sleep Deprivation) Saldırısı**:** Sensör düğümleri genellikle pil ile çalışan cihazlardır. Pil ömrünü uzatmak için ağ içindeki düğümler veri iletmediği zamanlarda uyku durumuna geçerler. Uyku yoksunluğu temelli saldırılarda düğümlerin uyku konumuna geçmesi engellenir. Bu durumun sonucunda düğümlerin pil ömürleri azalarak çalışmaz hale gelirler. Bu tip saldırılar daha çok ağın hizmet vermesini engellemek amacıyla DoS saldırıları olarak yorumlanabilir (Bhattasali, Chaki, & Sanyal, 2012).
- DoS Saldırısı: Ağın bir saldırgan tarafından işe yaramaz trafikle dolaştığı ve dolayısıyla ağın kullanıcılar tarafından kullanılamaz hale gelmesi nedeniyle hedeflenen sistemin tükendiği bir saldırı türüdür (Padmavathi & Shanmugapriya, 2009).
- Zararlı Kod Enjeksiyonu: Bu tip saldırılarda saldırgan bir düğümün içerisine zararlı bir kod yerleştirir. Zararlı kod ile en iyi ihtimal ile ağ devre dışı kalabilir

en kötü ihtimal ile tüm ağ saldırgan tarafından ele geçirilir (Fulare & Chavhan, 2011).

- Man-in-the-Middle Saldırısı: İki düğüm arasında sanki ağın bir parçasıymış gibi kontrolü saldırganda olan bir düğümün yerleştirildiği saldırı tipidir. Bu şekilde ağ verileri ister istemez saldırganın düğümü üzerinden geçer ve saldırgan tarafından ele geçirilir (Padhy, Patra, & Satapathy, 2011).

4.4.3. Destek Katmanı Tehditleri

Destek katmanı, bir sonraki uygulama katmanı için bulut bilişim gibi veri depolama teknolojilerinden oluşan bir platform sağlar. Bu katmanın güvenlik tehditleri aşağıda verilmiştir.

- Yetkisiz Erişim: Saldırgan, iyi bir şekilde güvenlik denetimi yapılmamış nesnelerin interneti servislerine yetkisiz erişim yaparak mevcut verilere ulaşabilir. Bu durum en kötü senaryodur. Saldırgan tüm sistemi silebilir veya kontrol edebilir.
- DoS Saldırısı: Diğer katmanlarda olduğu gibi destek katmanında gerçekleştirilen servis dışı bırakma saldırıları ile veri depolama veya işleme platformları devre dışı kalabilir. Bu şekilde ağ hizmet veremez hale gelir (Padmavathi & Shanmugapriya, 2009).

4.4.4. Uygulama Katmanı Tehditleri

Nesnelerin interneti uygulamasını ihtiyacına göre sunulan hizmetler dolayısı ile son kullanıcının kullanacağı uygulamalar bu katmanda tanımlıdır. Son kullanıcıya yönelik tüm saldırı tehditleri bu katmanı ilgilendirir. Belli başlı tehditler aşağıda verilmiştir.

- Zararlı Kod Enjeksiyonu: Nesnelerin interneti uygulamasını kullanan son kullanıcı üzerinden sisteme yapılan zararlı kod eklenmesi saldırısıdır. Bu şekilde son kullanıcının ve dolayısı ile sistemin verileri elde edilebilir (Farooq, Waseem, Khairi, & Mazhar, 2015).
- DoS Saldırısı: Diğer katmanlar da olduğu gibi nesnelerin interneti uygulamasına doğrudan yapılan saldırılar ile hizmet durdurulabilir. Özellikle

uygulama düzeyinde yapılan bu saldırılarda sadece nesnelerin interneti sistemi değil son kullanıcının şifrelenmemiş verileri de tehlike altındadır (Farooq vd., 2015).

- Mızrak-Kimlik Avı (Spear-Phishing) Saldırısı: Bu tip saldırılar son kullanıcıyı hedef almaktadır. Mızrak hırsızlığı, toplu saldırılarda hedeflenen milyonların aksine kurbanların belirli kişi ve veya kuruluşlar olarak seçilmesi prensibine dayanan sosyal mühendislik saldırısıdır. Kişinin bilgileri ele geçirildikten sonra nesnelerin interneti sistemi ile ilgili kullanım bilgileri üzerinden sisteme erişilmeye çalışılır (Farooq vd., 2015).

BÖLÜM 5

5. TEZ KAPSAMINDA İNCELENEN İLETİŞİM PROTOKOLLERİ

5.1. Giriş

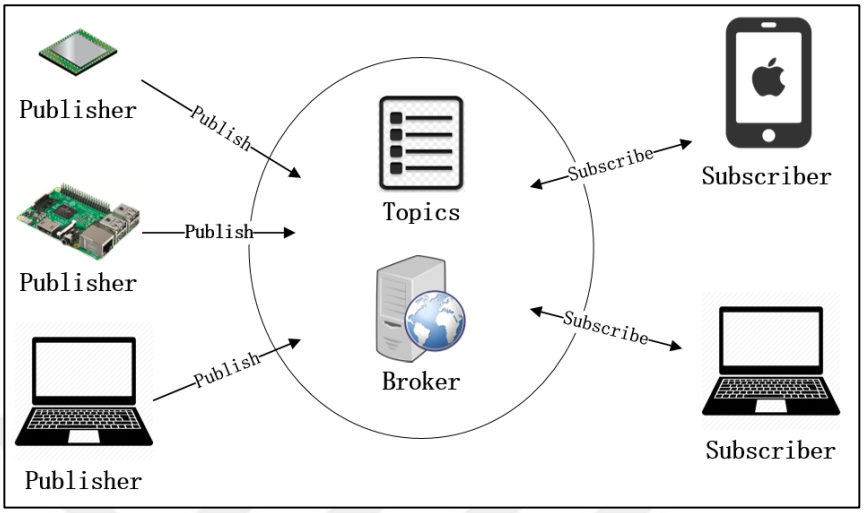
Nesnelerin interneti mimarisindeki uygulama katmanında çalışan ve nesnelerin birbirleri ile olan iletişimini sağlayan MQTT, CoAP, AMQP, XMPP, HTTP, DDS gibi birçok protokol bulunmaktadır (Dizdarević, Carpio, Jukan, & Masip-Bruin, 2019). Literatür incelendiğinde MQTT ve CoAP iletişim protokollerinin nesnelerin interneti uygulamalarında sıklıkla kullanıldığı görülmektedir. Bu bölümde detaylı olarak MQTT ve CoAP protokolleri ve altyapıları anlatılmıştır. Ayrıca tez kapmasında hazırlanan test ortamında bu iki iletişim protokolü kullanılmıştır.

5.2. Message Queueing Telemetry Transport (MQTT)

MQTT, TCP/IP'nin üstünde çalışan hafif sıklet, yayın-abone (publish-subscribe) tabanlı bir telemetri[8] mesajlaşma protokolüdür. MQTT 1999 yılında Dr. Andy Stanford-Clark ve Arlen Nipper tarafından geliştirilerek cihazların iletişimi için önerilmiştir. 2013 yılında IBM, MQTT v3.1'i OASIS[9] konsorsiyumuna sunmuştur ve o zamandan beri MQTT standartlaştırılmış bir protokol haline gelmiştir. Güncel versiyonu ise V5.0'dır (Banks, Briggs, Borgendale, & Gupta, 2019). Bu protokol cihazlar arasındaki iletişimin kurulmasını sağlamaktadır ve nesnelerin interneti alanında önemli bir yer kaplamaktadır. MQTT nesnelerin interneti uygulamalarında kullanılan cihazlar için kesintisiz, kayıpsız ve çift yönlü bağlantı sağlayan bir protokoldür. Protokol, bant genişliğini ve cihaz kaynak

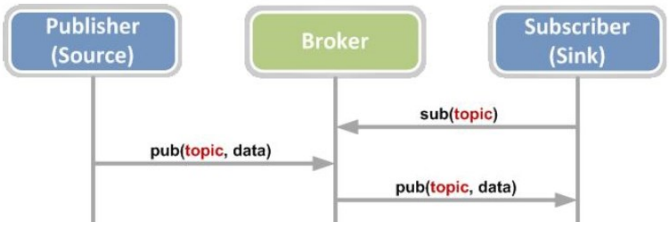
[8] Telemetri: Bir sistem ya da tesisin uzaktan kablo veya kablosuz olarak izlenmesi veya kontrol edilmesidir.
[9] OASIS nesnelerin interneti, güvenlik, bulut bilişimi, enerji, acil durum yönetimi ve diğer alanlar için dünya standartlarını üreten kar amacı gütmeyen bir konsorsiyumdur.

gereksinimlerini en aza indirirken, aynı zamanda iletim güvenilirliğini ve doğruluğunu sağlamaya çalışmaktadır.



Şekil 5.1. MQTT mimarisi

Şekil 5.1'de MQTT protokolünün mimarisi verilmiştir. Buna göre mesaj yayınına PUBLISH, aboneye SUBSCRIBE ve mesaj trafiğini kontrol eden yöneticiye BROKER denir. Mesaj alışverişi, yayıncı cihazdan (publisher) abone cihazlara (subscriber) doğru olmaktadır. Bu cihazlara genel olarak istemci denilebilir. MQTT sunucusu ya da diğer adı ile BROKER, abonelerin yayınladığı veya yönlendirdiği mesajları işleyebilir ya da doğrudan başka bir aboneye iletebilir. Broker, konuların (topic) listesini kaydederek istemcilerin abonelik veya abonelikten çıkma taleplerini yönetir.



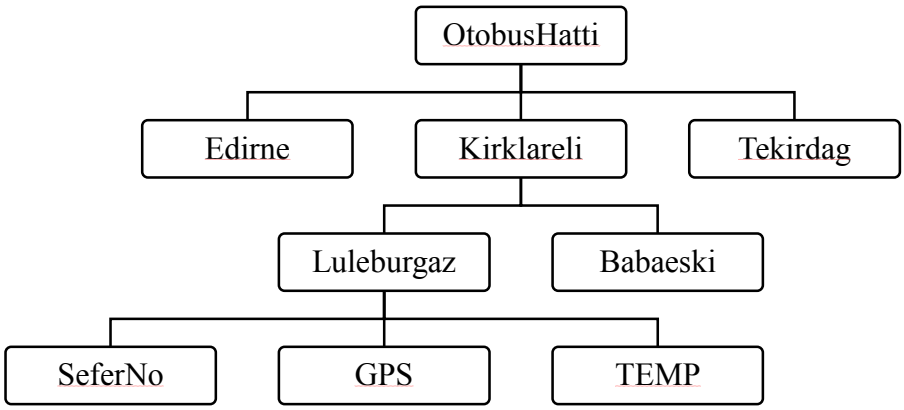
Şekil 5.2. Yayın-Abone yapısı

Geleneksel istemci-sunucu mimarisinde bulunan istek-yanıt (request-response) yapısının aksine MQTT protokolünde yayın-abone (publish-subscribe) yapısı kullanılır.

İstemciler yani yayıncı/abone cihazlar yayın/abonelik talebini bir veya daha fazla konu içeren bir konu ile Şekil 5.2'de gösterildiği gibi broker cihaza iletirler. Ayrıca istemci-sunucu mimarisinde, iletişim kurmak isteyen istemci ile hizmet veren sunucu birbirlerine direk olarak bağlı olmalıdır. Fakat yayın-abone yapısında istemciler, uygulamaya bağlı olarak yayıncı veya abone rolüne sahip olabilirler. Öte yandan mesajı gönderen bir yayıncının abonenin varlığını bilmesine gerek yoktur. Böylelikle yayıncı ve abonenin aynı anda çalışmaları gerekmediğinden veriler asenkron olarak yayınlanabilir. Fakat istemci-sunucu mimarisinde, istemci sunucu olmadan hareket edemez ve hizmet sunmak sadece sunucunun görevidir. Yayın-abone yapısında mesajların yayınlanması ve abone işlemleri, broker cihaz ile gerçekleştirilir.

5.2.1. Konu

MQTT, mesaj iletiminde mesajların konu (topic) tabanlı filtrelemesini kullanır. Konu, bir veya daha fazla konu seviyesinden oluşan metin tabanlı hiyerarşik bir yapıdır. Konu içindeki her bir hiyerarşiyi "/" işareti ayırır. Bir mesajın konusu Unicode karakter kümesinde bulunan karakterlerden oluşur. Ayrıca konu başlıkları büyük-küçük harfe duyarlıdır, örneğin gps ile GPS farklı konu başlıklarıdır.



Şekil 5.3. Bir ulaşım hattı için hiyerarşik yapı

Örneğin basit bir otobüs ulaşım hattı için hazırlanan hiyerarşik yapı Şekil 5.3'te verilmiştir. Buna göre OtobusHatti / Kirklareli / Luleburgaz / GPS ifadesi bir konudur.

Bu durumda OtobusHatti ilk konu başlığı, Kirklareli ikinci konu başlığı ve Luleburgaz üçüncü konu başlığı olarak abonelere açılmıştır. Belirtilen konu başlığına abone olan istemcilere Lüleburgaz'a giden ilgili otobüsün GPS koordinatları verisi Broker tarafından iletilir. Ayrıca istemciler özel karakterler (+, # gibi) ile farklı verilere de abone olabilirler. Şekil 5.3'teki yapıya göre bazı konular için abonelik ile ilgili örnekler Çizelge 5.1'de verilmiştir:

Çizelge 5.1. MQTT konu örnekleri

Örnek Konu	**Açıklama**
OtobusHatti / Kirklareli / Luleburgaz / +	Lüleburgaz'a giden otobüsteki tüm sensör verilerini ifade eder.
OtobusHatti / Kirklareli / + / TEMP	Kırklareli'ye giden tüm otobüslerdeki sıcaklık verilerini ifade eder.
OtobusHatti / Kirklareli / #	Kırklareli'ye giden tüm otobüslerdeki tüm sensör verilerini ifade eder.
OtobusHatti/#	Sistemdeki tüm otobüslerin tüm sensör verilerini ifade eder.
OtobusHatti / + / + / GPS	Sistemdeki tüm otobüslerin GPS verilerini ifade eder.

5.2.2. Broker

MQTT broker, yayın-abone mimarisinin merkezi bir parçasıdır. Broker, yayıncılardan gelen verileri filtreleyerek konulara göre depolar. Ayrıca konulara abone olan cihazlara ilgili verinin iletilmesini sağlar. Dolayısı ile sistemdeki sunucu ve yönetici görevini üstlenir. Filtreleme işlemi 3 farklı seçenekle gerçekleştirilebilir. Bunlar:

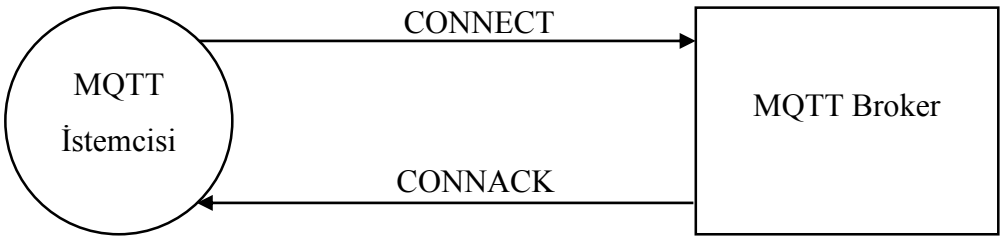
- Konu tabanlı filtreleme: Sadece konu başlığına göre yapılan filtrelemedir. Bu durumda konu başlığı her mesaja eklenmektedir ve ilgili konuya abone olan

istemcilere mesaj iletilir. Örneğin sıcaklık konu başlığına abone olan cihazlar bu konu ile ilgili tüm verileri alırlar.

- İçerik tabanlı filtreleme: Bu filtreleme yönteminde mesajın içeriğine göre işlem yapılır. Örneğin 25 dereceden daha yüksek sıcaklık verisine abone olan istemcilere sıcaklık konu başlığındaki verilerin içeriğine bakılarak sadece 25 derece üzerindeki verilerin iletimi yapılır.
- Tip tabanlı filtreleme: Bu tür filtrelemede birden fazla konu bir tip adı altında sınıflanır ve bu veriler ilgili abone istemcilere iletilir. Örneğin sıcaklık, karbondioksit oranı, nem gibi veriler hava adlı bir tip içerisinde sınıflandırılabilir. İstemci hava tipine abone olur ve mesaj içerisinde ilgili verileri alabilir.

5.2.3. Abone-Broker Bağlantısı

MQTT iletişim ortamında abone ve yayıncı cihazlar birbirlerine doğrudan bağlanamaz. Fakat bu cihazlar sadece broker ile doğrudan bağlıdır. Bunun için CONNECT mesajıyla yayıncı veya abone cihaz broker ile bağlantı kurar. Broker bu CONNECT mesajını CONNACK mesajı ile cevaplar. Şekil 5.4'te istemci ile broker arasındaki MQTT bağlantısının kurulma şekli gösterilmektedir. Broker ile iletişim kurulduktan sonra istemci broker içindeki bir konuya abone olabilir ya da bir konuya veri iletebilir.



Şekil 5.4. MQTT Bağlantısı

MQTT istemci CONNECT mesajının yapısı ve alanların açıklamaları Çizelge 5.2'de verilmiştir.

Çizelge 5.2. MQTT bağlantı mesaj yapısı

CONNECT Mesajı	
Alanlar	**Örnek Mesaj**
clientID	“istemciNo-1”
cleanSession	true
username (isteğe bağlı)	“Trakya”
password (isteğe bağlı)	“trak123”
lastWillTopic (isteğe bağlı)	“kampus / muh / kat1 / +”
lastWillQos (isteğe bağlı)	2
lastWillMessage (isteğe bağlı)	“basarisiz olundu”
lastWillRetain (isteğe bağlı)	false
keepAlive	60

- clientId: Her MQTT istemcisinin benzersiz bir istemci tanımlayıcısı vardır. Bir istemci kimliği 65535 karakter uzunluğunda olabilir. İstemci kimliği olarak, cihaza verilecek bir isim, UuID (Evrensel Tekil Tanımlayıcı), istemcinin ağ modülünün MAC adresi veya cihazın seri numarası gibi benzersiz bilgiler kullanılabilir.
- cleanSession: İstemcinin kalıcı bir oturum açmak isteyip istemediğinin belirtildiği bayraktır. Kalıcı bir oturum, broker cihazın ilgili istemci için abonelik ile ilgili iletileri saklayacağı anlamına gelir. Bunun için bayrak false konumuna alınmalıdır. Clean Session true değerine ayarlanırsa, broker istemci

için herhangi bir veri saklamaz ve aynı zamanda bir önceki kalıcı oturumdaki tüm bilgileri de temizler.

- username/password: MQTT, istemci kimliğinin doğrulanması ve yetkilendirilmesi için kullanıcı adı ve şifre göndermeye izin verir. Bununla birlikte eğer gönderilen şifre, uygulama tarafından şifrelenmediyse veya SSL gibi güvenli bir hat üzerinden iletilmediyse açıkça ağ üzerinde görünecektir. Bu sebeple kullanıcı adı ve şifrenin güvenli bir şekilde taşınması için önlemler alınması gerekmektedir.
- lastWill: Sistem içerisinde bir istemci etkisiz hale geldiğinde veya ağdan koptuğunda ağdaki diğer istemcilere gönderilen bazı mesajları içeren bölümdür. İstemciler bağlantısız bir şekilde koparsa broker bu mesajı istemci adına diğer istemcilere gönderir.
- keepAlive: İstemcinin ağda canlı kalabilmek için gereken zaman aralığıdır. Bu durumu istemciler broker cihaza normal PING mesajı ileterek taahhüt ederler. Broker tarafından üretilen PING cevabı ile istemcinin hala canlı ve ulaşılabilir olup olmadığı tespit etmesine izin verir.

MQTT broker CONNACK mesajının yapısı ve alanların açıklamaları Çizelge 5.3'te verilmiştir.

Çizelge 5.3. MQTT bağlantı cevap mesajı yapısı

CONNACK Mesajı	
Alanlar	**Örnek Mesaj**
sessionPresent	true
returnCode	0

- sessionPresent: Broker cihazın istemcinin önceki bağlantılarından kalıcı bir oturuma sahip olup olmadığını gösterir. Eğer bir istemci CONNECT mesajının CleanSession bölümünü true olarak ayarladıysa, Session Present her zaman false olur. Çünkü kullanılabilir bir oturum yoktur ya da var ise yeni bir

oturum talep edilmiştir. Diğer durumda istemci CONNECT mesajında CleanSession bölümünü false olarak ayarlamış ise broker istemcinin daha önce oturum bilgisi olup olmadığını inceler ve bu duruma göre CONNACK mesajının Session Present bölümü true veya false olarak ayarlanır.

- returnCode: Bağlantının başarılı bir şekilde onaylandığını veya başarısız bir durum var ise ilgili durumu geri iletir. Durum kodları aşağıda verilmiştir.

    0. Bağlantı kabul edildi.
    1. Bağlantı kabul edilmedi, protokol sürümü kabul edilemez.
    2. Bağlantı kabul edilmedi, istemci adı kabul edilmedi.
    3. Bağlantı kabul edilmedi, sunucu kullanılamıyor.
    4. Bağlantı kabul edilmedi, hatalı kullanıcı adı veya hatalı şifre.
    5. Bağlantı kabul edilmedi, yetkilendirilme sorunu.

5.2.4. Yayın Mesajı

MQTT istemcisi bir broker cihaza bağlandıktan sonra iletileri PUBLISH mesajı ile yayınlayabilir. Her yayın ve abone olma mesajı broker tarafından mesajların filtrelenmesi için kullanılacak bir konuyu ve mesajın kendisi olan yük (payload) bilgisini içerir. Yük bilgisi ikili veri olabileceği gibi metin verisi ve hatta JSON veya XML kodu dahi olabilir.

MQTT istemcisi tarafından yayınlanan PUBLISH mesajının yapısı ve alanların açıklamaları Çizelge 5.4'te verilmiştir.

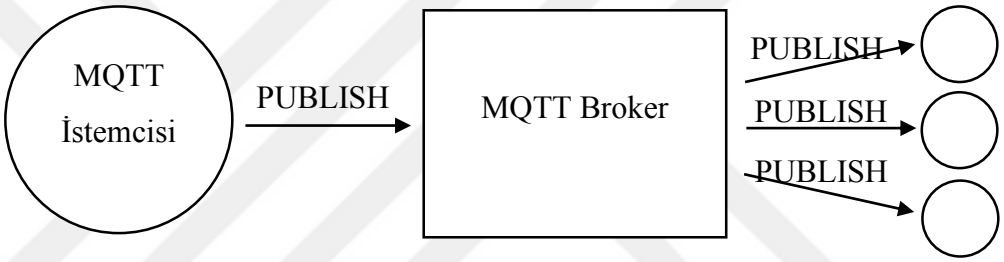
Çizelge 5.4. MQTT yayın mesajı yapısı

PUBLISH Mesajı	
Alanlar	**Örnek Mesaj**
packetId	417
topicName	“kampus / muh / kat1 / +”
QoS	1
retainFlag	false
payload	“sicaklik:25.5”
dupFlag	false

- packetId: Gönderilen mesajın paket numarasıdır.
- topicName: İstemcinin broker cihaza ileteceği verinin hangi konuya ait olduğu bu bölümde belirtilir.
- QoS: Mesajın hizmet kalite seviyesini belirtir. MQTT içerisinde 3 hizmet kalite seviyesi bulunmaktadır. Bunlar QoS 0, QoS 1, QoS 2’dir. Bu seviyeler mesajın ulaşma garantisini belirler.
- retainFlag: Bu bayrak, iletinin belirtilen konu için broker tarafından son bilinen iyi değer olarak kaydedilip kaydedilmeyeceğini belirler. Bu konuya abone olan diğer istemciler, abone olduktan hemen sonra bu konudaki en son saklanan mesajı alacaklardır.
- payload: İletilen yayının içerisindeki mesajdır. Bu bölümde herhangi bir kodlama, şifrelenmiş veya şifrelenmemiş metin veya görüntü iletmek mümkündür.

- dupFlag: Gönderilen iletinin daha önce gönderilen bir iletinin kopyası (dublicate) olduğunu belirtmek için ayarlanır. Bu durum sadece QoS 0 seviyesinde gerçekleşir.

Bir istemci MQTT broker cihazına bir yayın gönderdiğinde, broker yayını okur, gerekirse QoS seviyesine göre yayınlamayı kabul eder ve sonra işler. İşleme, söz konusu istemciye hangi istemcinin abone olduğunu belirlemek ve daha sonra bu konuya abone olan seçilmiş istemcilere mesajı göndermekle ilgilidir. Yayını yapan istemci sadece mesajı broker cihaza iletir. Konuya abone olanlara mesajın iletildiğini kontrol etmez. Bu durum sadece broker cihazın sorumluluğundadır. Böylelikle geri bildirimler minimuma indirilmiş olunur. Bu durum Şekil 5.5'te ifade edilmiştir.



Şekil 5.5. MQTT yayın işlemi

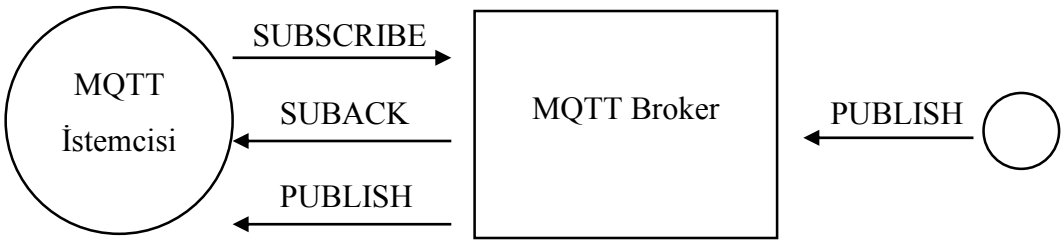
5.2.5. Abone İşlemi

Bir istemcinin ilgili mesajları almak için MQTT broker cihazına bir SUBSCRIBE mesajı göndermesi gerekir. Abonelik mesajı oldukça basittir, sadece benzersiz bir paket tanımlayıcısı ve bir abonelik listesi içerir. MQTT istemcisi tarafından yayınlanan SUBSCRIBE mesajının yapısı ve alanların açıklamaları Çizelge 5.5'te verilmiştir.

Çizelge 5.5. MQTT abone mesaj yapısı

SUBSCRIBE Mesajı	
Alanlar	**Örnek Mesaj**
packetId	1980
qos1	1
topic1	“kampus / muh / kat1 / +”
qos2	0
topic2	“kampus / fbe / +”
…	…

- packetId: Paket tanımlayıcı numarası, mesajı tanımlamak için istemci ve broker arasında benzersiz bir tanımlayıcıdır.
- Aboneliklerin listesi: SUBSCRIBE mesajı içerisinde abone olunacak konu ile ilgili bilgi gönderilmelidir. Bu talep konu adı ve QoS seviyesinden oluşur. Abone mesajındaki konu, belirli konu başlığına abone olmayı mümkün kılan joker karakterler de içerebilir. Bir istemci için çakışan abonelikler varsa, bu konu için en yüksek QoS seviyesi öncelik kazanır ve iletiyi iletmek için broker tarafından kullanılır.



Şekil 5.6. MQTT abonelik işlemi

İstemcinin SUBSCRIBE mesajı ile abonelik talebi sonrasında broker tarafından SUBACK mesajı ile onaylama işlemi gerçekleştirilir. Bu durum Şekil 5.6'da özetlenmiştir. SUBACK mesajı, orijinal abone mesajıyla ve geri dönüş kodlarının bir listesiyle aynı paket tanımlayıcı numarasını içerir.

MQTT broker tarafından yayınlanan SUBACK mesajının yapısı ve alanların açıklamaları Çizelge 5.6'da verilmiştir.

Çizelge 5.6. MQTT abone cevap mesaj yapısı

SUBACK Mesajı	
Alanlar	**Örnek Mesaj**
packetId	1981
returnCode1	1
returnCode2	0
…	…

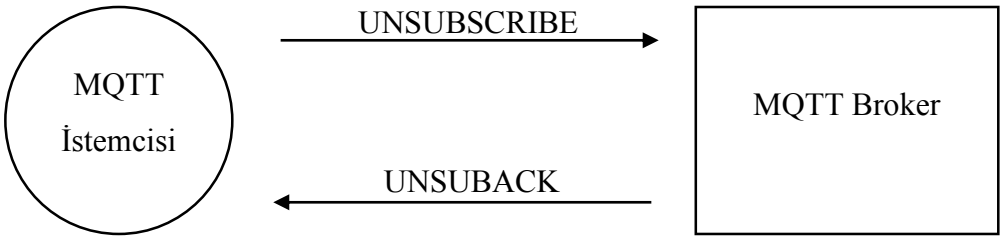
- packetId: Paket tanımlayıcı numarası, mesajı tanımlamak için istemci ve broker arasında benzersiz bir tanımlayıcıdır.
- returnCode: Broker, SUBSCRIBE iletisinde aldığı her konu ve QoS çifti için bir dönüş kodu gönderir. Dolayısıyla, SUBSCRIBE mesajında 5 abonelik talebi varsa, broker tarafından verilen QoS seviyesi ile her bir konuyu kabul etmek için 5 adet dönüş kodu olacaktır. Aboneliğin broker tarafından yetkisiz erişim, hatalı konu başlığı gibi nedenler ile reddedilmesi durumunda broker belirli bir konu için bir hata dönüş kodu ile yanıt verecektir. Dönüş kodları aşağıdaki gibidir:
    - 0 Başarı Maksimum QoS 0,
    - 1 Başarı Maksimum QoS 1,
    - 2 Başarı Maksimum QoS 2,
    - 128 Başarısız.

İstemcinin aboneliği sonlandırması için UNSUBSCRIBE mesajı yayınlaması gerekir. UNSUBSCRIBE mesajı QoS seviyesi olmayan SUBSCRIBE mesajına benzemektedir. Aboneliği sonlandırılacak konular benzersiz bir paket tanımlayıcı kodu ile beraber broker cihaza gönderilir.

MQTT istemci tarafından yayınlanan UNSUBSCRIBE mesajının yapısı ve alanların açıklamaları Çizelge 5.7'de verilmiştir.

Çizelge 5.7. MQTT abonelikten çıkma mesaj yapısı

UNSUBSCRIBE Mesajı	
Alanlar	**Örnek Mesaj**
packetId	2015
topic1	"kampus / muh / kat1 / +"
topic2	"kampus / fbe / +"
…	…



Şekil 5.7. MQTT abonelik iptali işlemi

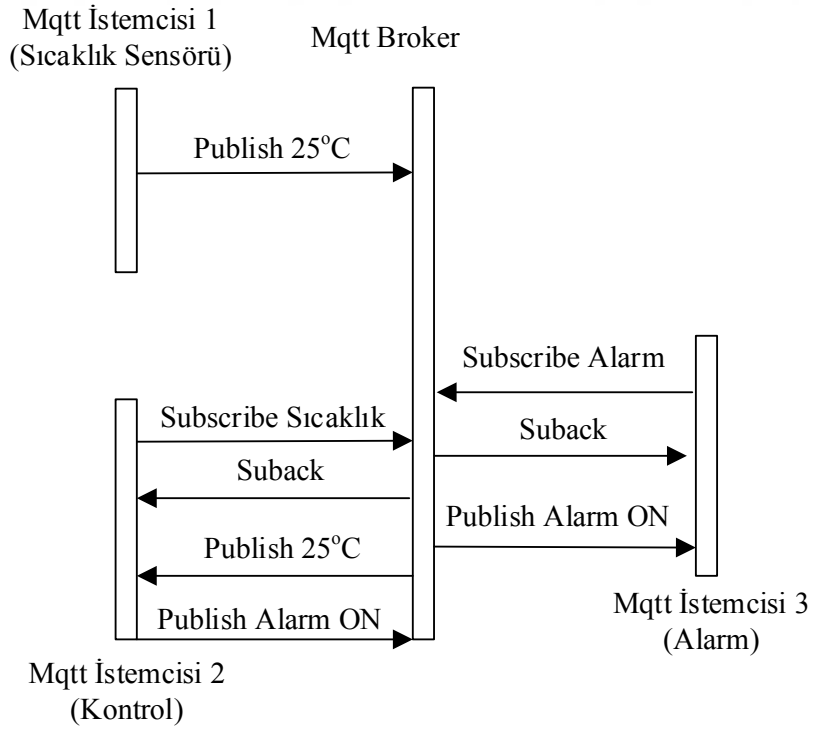
İstemcinin abonelik iptali için gönderdiği UNSUBSCRIBE mesajına karşılık broker UNSUBACK mesajını istemciye iletir. Bu mesaj sadece bir paket tanımlayıcı içerir. Broker cihazdan UNSUBACK mesajı alındıktan sonra istemci UNSUBSCRIBE mesajındaki talep edilen aboneliklerin sildiğini kabul eder. Bu durum Şekil 5.7'de özetlenmiştir.

MQTT istemci tarafından yayınlanan UNSUBACK mesajının yapısı Çizelge 5.8'de verilmiştir.

Çizelge 5.8. MQTT abonelikten çıkma cevap mesaj yapısı

UNSUBACK Mesajı	
Alanlar	**Örnek Mesaj**
packetId	2016

Şekil 5.8'de örnek bir MQTT bağlantısı görülmektedir. Sıcaklık sensörüne sahip MQTT istemcisi 1, broker cihaza veri göndermektedir. MQTT istemcisi 2, sıcaklık sensörünün ürettiği veriye ilgili başlık üzerinden abone olmuş ve sıcaklık verilerini almaya başlamıştır. MQTT istemcisi 2, sıcaklık verisi üzerinden broker cihazda listelenen alarm konusuna veri göndermiş ve bu işlemden tetiklenen MQTT istemcisi 3, harekete geçerek alarm çalmıştır.



Şekil 5.8. Örnek bir MQTT bağlantısı

5.2.6. Hizmet Kalitesi (QoS) seviyesi

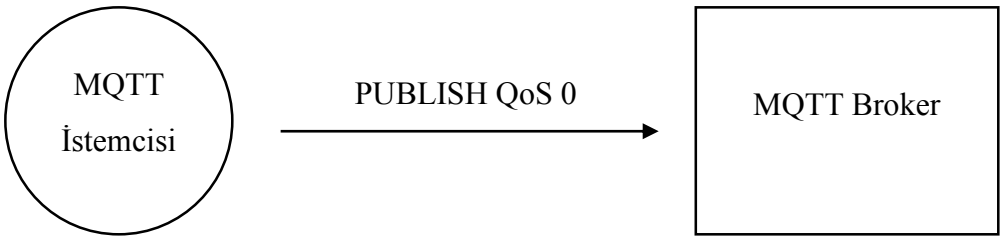
Hizmet Kalitesi (QoS: Quality of Service) seviyesi, bir mesajın iletilmesi garantisine ilişkin gönderici ve alıcı arasındaki anlaşmadır. MQTT'de 3 adet hizmet kalitesi seviyesi vardır.

- En fazla bir kere (QoS 0: At most once)
- En az bir kez (QoS 1: At least once)
- Tam olarak bir kez (QoS 2: Exactly once).

MQTT içerisinde mesajın iletimi, istemcinin broker cihaza mesaj iletimi ve broker cihazın abone istemciye mesaj iletimi şeklindedir. İstemciden broker cihaza mesaj iletmek için QoS seviyesi, istemcinin mesaj paketi içinde belirlediği QoS seviyesine bağlıdır. Broker cihazın istemciye mesaj iletimi durumunda ise broker, istemcinin daha önceden abone olurken ilettiği QoS seviyesini kullanır. QoS, MQTT'nin önemli bir özelliğidir. Güvenilir olmayan ağlarda iletişimi çok daha kolay hale getirir. Ayrıca ağ güvenilirliği ve uygulamanın mantığına göre istemciye QoS seviyesi seçme esnekliği sağlanmış olunur.

5.2.6.1 QoS 0

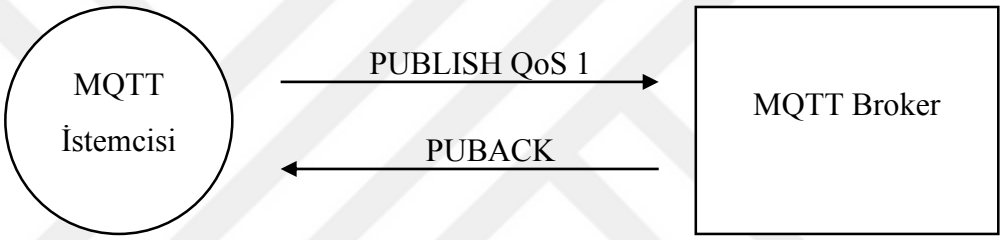
MQTT QoS seviyelerinde minimum seviye sıfırdır. Bu seviyede alıcı tarafından herhangi bir cevap gönderilmez ve gönderen tarafından tekrar deneme yapılmaz. Geri bildirim olmadığı için performans olarak en hızlı iletim şeklidir. Fakat iletinin ulaşım garantisi ağın sağlıklı çalışmasına bağlıdır. QoS 0 iletimi aşamasında gönderen, QoS = 0, DUP = 0 ile bir PUBLISH paketi hazırlamalıdır.



Şekil 5.9. MQTT QoS 0 iletimi

5.2.6.2 QoS 1

QoS seviye 1'i kullanırken, bir mesajın alıcıya en az bir kez iletilmesi garanti edilir ancak mesaj bir kereden fazla da teslim edilebilir. QoS 1 seviyesinde gönderilen PUBLISH paketinin iletimi içerisinde sadece benzersiz bir paket tanımlayıcısı bulunan PUBACK paketiyle onaylanmaktadır. Bu durum Şekil 5.10'da özetlenmiştir. QoS 1 iletimi aşamasında gönderen, QoS = 1, DUP = 0 ile bir PUBLISH paketi hazırlamalıdır. PUBACK paketi makul sürede alınmaz ise gönderici PUBLISH paketini bir daha gönderecektir. Bir PUBLISH paketinin yeniden iletilmesi durumunu belirleyen DUP bayrağı sadece dahili amaçlar içindir ve QoS 1'de broker veya istemci tarafından işlenmemektedir. Alıcı, DUP bayrağına bakılmaksızın bir PUBACK gönderecektir.

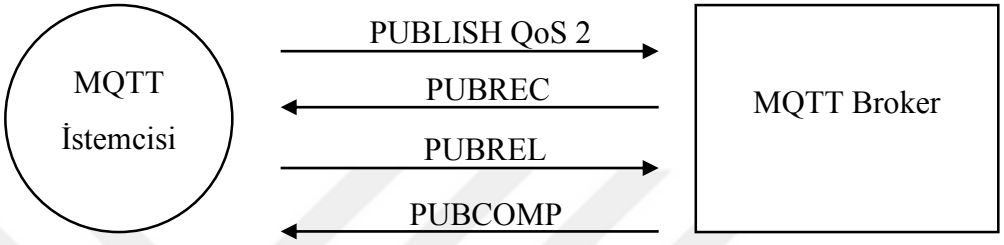


Şekil 5.10. MQTT QoS 1 iletimi

5.2.6.3 QoS 2

En yüksek seviye QoS 2'dir. Her mesajın ilk gönderimde karşı taraf tarafından alındığını garanti eder. En güvenilir ve en yavaş servis seviyesidir. Garanti, gönderen ve alıcı arasında iki akış sayesinde sağlanır. Alıcı QoS 2 seviyesinde PUBLISH paketi alırsa mesajını uygun şekilde işleyecek ve paketi aldığını belirtmek için içerisinde sadece benzersiz bir paket tanımlayıcısı bulunan PUBREC (Publish Received) mesajı ile göndericiye bildirecektir. Gönderici PUBREC paketini aldığında, ilk yayınlamayı güvenli bir şekilde silebilir çünkü alıcının gönderilen mesajı başarıyla aldığını bilir. Gönderici PUBREC mesajını depolayarak içerisinde sadece benzersiz bir paket tanımlayıcısı bulunan bir PUBREL (Publish Release) ile cevap verecektir. Gönderilen bu PUBREL paketinin tanımlayıcısı orijinal PUBLISH paketiyle aynı olmak durumundadır. Alıcı PUBREL paketini aldıktan sonra, depolanan her durumu silebilir ve içerisinde sadece benzersiz bir paket tanımlayıcısı bulunan bir PUBCOMP (Publish Complete) ile

yanıtlayabilir. Gönderen PUBCOMP paketini aldığında aynı şekilde tüm verileri silebilir. Akış tamamlandığında, her iki taraf da mesajın teslim edildiğinden emin olurlar. Bu durum Şekil 5.11'de özetlenmiştir Bir paket yolda kaybolduğunda, son mesajı makul bir süre sonra tekrar göndermek göndericinin sorumluluğundadır. Alıcı ise her komut mesajına uygun şekilde cevap verme sorumluluğuna sahiptir.



Şekil 5.11. MQTT QoS 2 iletimi

5.2.7. MQTT ve Güvenlik

MQTT protokolünün amacı, nesnelerin interneti cihazlarının iletişimi için hafif sıklet ve kullanımı kolay bir iletişim protokolü sağlamaktır. Bu sebeple protokol içinde doğrudan güvenlik amacıyla bir mekanizma bulunmamaktadır. Buna karşın protokolün uygulamalarında verilerin gizliliği için çoğunlukla TLS ve SSL (Güvenli Yuva Katmanı) gibi servisler kullanılmaktadır.

TLS ve SSL, sunucu ve yayıncı/abone cihaz arasında güvenli bir bağlantı oluşturmak için çeşitli parametrelerde anlaşmak üzere bir el sıkışma mekanizması kullanan güvenlik protokolleridir. El sıkışma tamamlandıktan sonra, taraflar arasında şifreli bir iletişim kurulur ve böylece saldırgan iletişimin herhangi bir bölümünü dinleyemez. Sunucular, genellikle istemcilerin sunucunun kimliğini doğrulamak için kullandıkları güvenilir bir otorite tarafından verilen bir X509[10] sertifikasını kullanırlar. MQTT varsayılan olarak TCP üzerinden şifrelenmemiş iletişim yolunu 1883 numaralı port üzerinden kullanmaktadır. Tüm MQTT iletişimini şifrelemek için, çoğu MQTT

[10] X.509, ortak anahtar sertifikalarının biçimini tanımlayan bir standarttır. Elektronik imzalar, HTTPS gibi birçok sistemde kullanılmaktadır.

broker düz TCP yerine TLS kullanımına izin vermektedir. 8883 numaralı port güvenli bir MQTT bağlantısı için kullanılmaktadır ve bu bağlantının IANA[11]'daki standart isim "secure-mqtt" dir. Uygulama seviyesinde MQTT protokolü, cihazların kimliğini doğrulamak için kullanıcı adı / şifre mekanizmasının kullanımını sağlar fakat kullanıcı adı ve şifre olmadan da bağlantının kurulması mümkündür. Protokolün sağladığı en önemli ve etkin güvenlik önlemi bu durumdur. MQTT protokolünde kimlik doğrulama söz konusu olduğunda bağlantı aşamasında gönderilen CONNECT mesajında kullanıcı adı ve şifre alanları doldurulabilir. Kullanıcı adı ve şifre UTF-8 ile kodlanmış en fazla 65535 bayt uzunluğunda bir metin olmalıdır. MQTT belirtim dokümanında en kısa şifrenin 12 karakter olması gerektiği bildirilmektedir.

Görüleceği gibi MQTT güvenlik odaklı bir protokol değildir. Bu sebeple üçüncü şahıslar tarafından erişilmemesi gereken bilgiler iletilecekse paketler oluşturulmadan önce MQTT istemcisi mutlaka şifreleme yaparak mesajları iletmelidir. Bunun için istemci-istemci veya istemci-broker şifreleme/şifre çözme (deşifreleme) yaklaşımları tercih edilebilir. Bu işlem Bölüm 8'de anlatılmıştır.

5.3. The Constrained Application Protocol (CoAP)

2010 yılında IETF[12] CoRE çalışma grubu, kısıtlı cihazların internet üzerinden iletişimi ve birbirleri arasında birlikte çalışabilirliği sağlayan bir protokol geliştirmeye başlamış ve CoAP ortaya çıkmıştır. CoAP'ın ana hedefi makineden makineye (M2M) olan uygulamaları dikkate alarak, nesnelerin interneti uygulamalarında kullanılan kısıtlı özel gereksinimleri olan cihazlar için HTTP tabanlı genel bir iletişim protokolü hizmeti sağlamaktır (Bormann, Castellani, & Shelby, 2012). Bunun için protokol multicast desteği ve eşzamanlı olmayan iletişim gibi M2M için birçok özellik sunmaktadır. CoAP aşağıdaki ana özelliklere sahiptir:

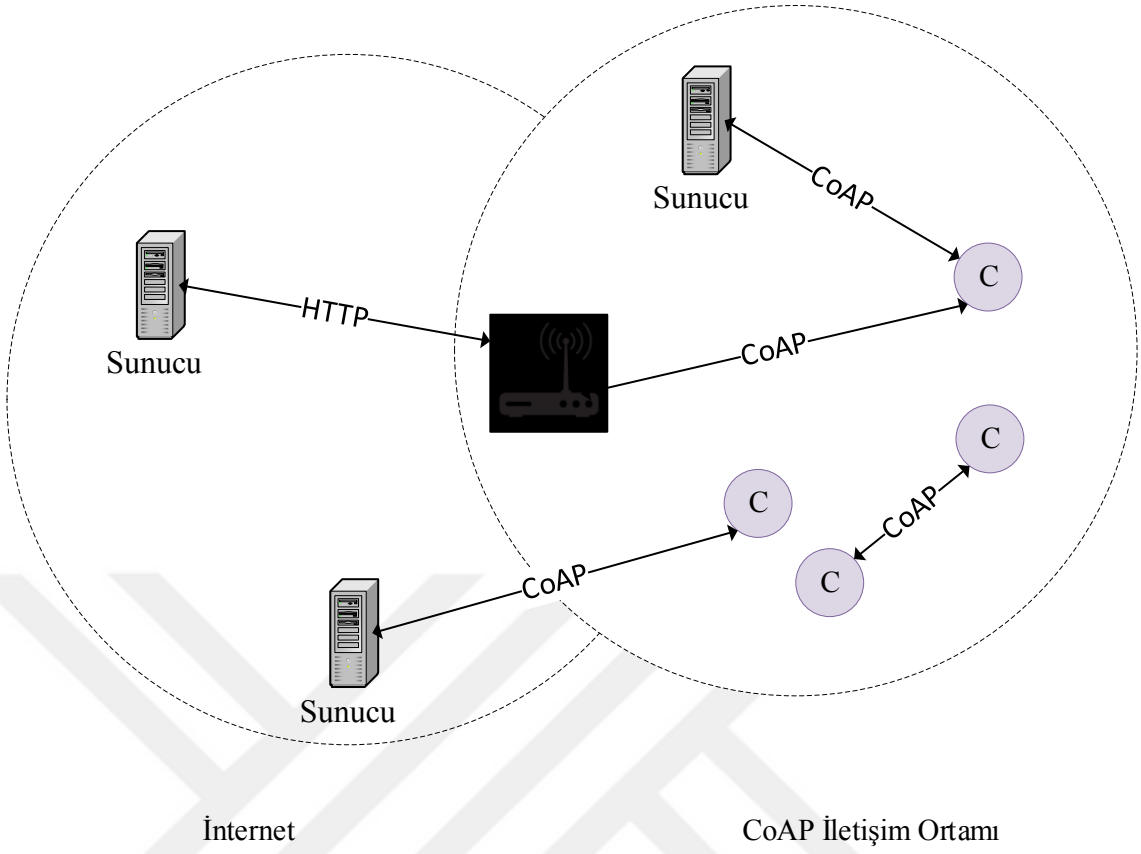
- Kısıtlı ortamlarda M2M gereksinimlerini karşılayan web protokolü,
- UDP üzerinden isteğe göre unicast veya multicast desteği,
- Asenkron mesaj alışverişi,
- Küçük paket yapısı,

[11] IANA (Internet Assigned Numbers Authority) internet alanında IP adresleri, alan adları gibi parametreleri belirlemek için kurulmuş yetkili kurumdur.
[12] Internet Engineering Task Force, internet protokollerini geliştiren ve standartlaştıran bir gruptur.

- URI[13] ve içerik tipi desteği,
- Basit proxy ve önbellekleme yeteneği.

Şekil 5.12'de verilen CoAP mimarisi, HTTP'nin istemci-sunucu mimarisine benzemektedir. Fakat burada istemci ve sunucu rolleri cihazın davranışına göre değişmektedir. HTTP iletim aşamasında TCP paketlerini kullanırken, CoAP hızlı iletişimin sağlanması için UDP paketlerini kullanmaktadır. İletişim kurulurken CoAP istemcisi, sunucu üzerinden veri alabilmek veya gönderebilmek için istek mesajı iletmektedir. CoAP sunucusu ise isteği işleyerek yanıt kodu ve veriyi içeren bir yanıt paketi gönderir. CoAP mesajlarında 4-byte sabit uzunluğunda başlık (header) bölümü bulunmaktadır. Her mesajda 16-bit uzunluğunda benzersiz bir kimlik numarası (MID), mesajın türü ve verinin URI bilgisi bulunmaktadır. CoAP'ın standart parametreleri ile uçtan uca saniyede 250 mesaj gönderilebilir (CoAP RFC 7252, 2014).

[13] Uniform Resource Identifier teriminin kısaltılmış halidir. İnternet'te bir kaynağın tam yerine işaret eden (belge veya resim gibi), standart bir formata uygun bir karakter dizgisidir. Bir URL'in altında bulunan bir kaynağın tam yoluna işaret eder.



Şekil 5.12. CoAP Mimarisi

5.3.1. CoAP URI Adresleri

CoAP, veri kaynaklarının yerini tanımlamak ve bu kaynaklara ulaşmak için "coap" ve "coaps" URI şemalarını kullanır. Kaynaklar hiyerarşik olarak düzenlenir ve belirli bir UDP portunda standart CoAP istekleri “coap”, DTLS ile güvenliği sağlanan CoAP istekleri ise “coaps” protokol etiketi ile belirlenir. URI adresinin geri kalanı CoAP protokolü tarafından tanımlanan yöntemlerle çalıştırılabilen bir kaynağı belirtir. Örnek bir URI adresi aşağıdaki gibi yazılabilir.

coap://	192.168.1.50	:5683	/ledler/kirmizi	?durum=acik
Protokol	IP, DNS veya ağ adı	Port	Alt adres	Sorgu

CoAP URI adresinde protokol mutlaka yazılmalıdır. DTLS üzerinden güvenli iletim için coaps tercih edilmelidir. Protokol önekinin devamında cihazın ip adresi

yazılabileceği gibi DNS adresi veya ağ adı da yazılabilir. Veri iletimi özel bir port üzerinden yapılacak ise cihazın port bilgisi mutlaka verilmelidir. Port bilgisi verilmez ise iletişim varsayılan CoAP portu olan 5683 numaralı UDP portu üzerinden yapılır. "/" işaretinden sonra verilerin bulunduğu hiyerarşik alt adres yazılır. "?" işaretinden sonra erişilmek istenilen veri etiketi belirtilerek sorgu bölümü hazırlanır. Sorgu bölümü genellikle "anahtar = değer" (key=value) eşleşmesi üzerinden yapılır. Birden fazla sorgu için "&" işareti kullanılır.

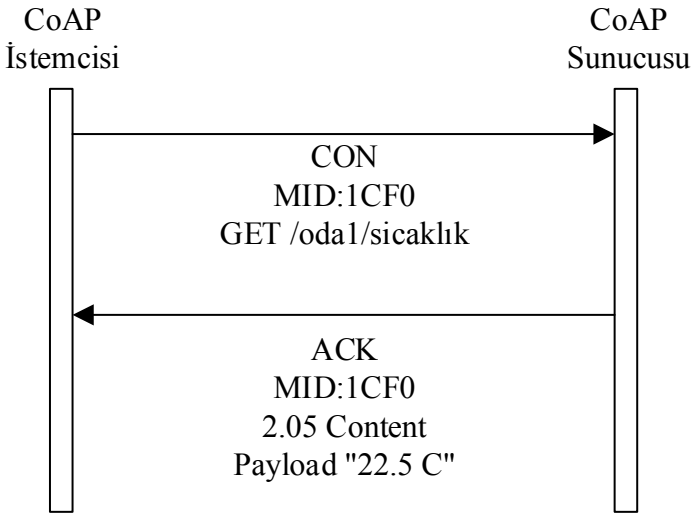
5.3.2. CoAP Mesaj Tipleri

CoAP içerisinde Çizelge 5.9'da verilen 4 tip mesaj tanımlanmıştır. Mesaj tipi başlıktaki 2 bitlik bir tam sayı ile gösterilir.

Çizelge 5.9. CoAP mesaj tipleri

Tip No	Mesaj	Kod
0	Confirmable (doğrulanabilir)	CON
1	Non-confirmable (doğrulanamaz)	NON
2	Acknowledgement (onaylandı)	ACK
3	Reset (sıfırla)	RST

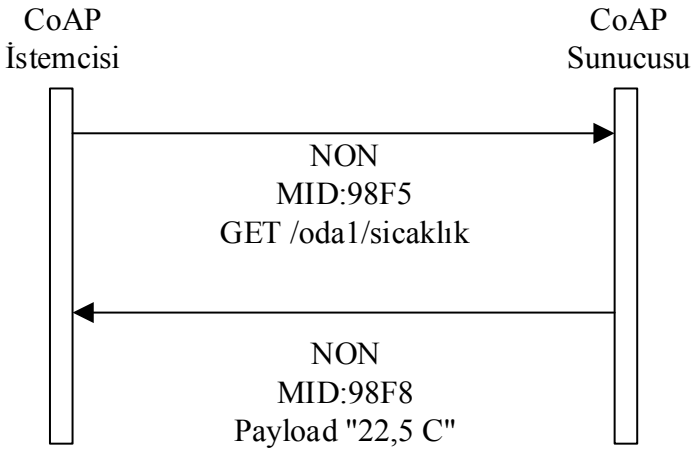
Gönderilen mesajın ulaşılmasının garanti edilebilmesi için mesaj tipi CON olarak işaretlenmelidir. Bunun karşılığında alıcı, talep edilen isteği mutlaka bir ACK tipindeki mesaj ile iletmek zorundadır. Alıcı tarafından gönderilen ACK mesajının MID bilgisi göndericinin ilettiği CON mesajındaki MID ile aynıdır. Böylelikle gönderici aldığı mesajın hangi istek doğrultusunda geldiğini bilir. Bu durum Şekil 5.13'te verilmiştir.



Şekil 5.13. CoAP CON iletişimi

Varsayılan olarak, geri alma (back-off) yani dönüş bekleme süresi 2 sn ve 3 sn arasındadır. Bu süre içerisinde alıcıdan ACK mesajı gelmez ise gönderici CON mesajını tekrar iletir. Tekrar ACK mesajı gelmez ise geri alma süresi bir önceki gönderimindeki sürenin 2 katına çıkarılarak ACK mesajı beklenir. MAX_RETRANSMIT yani en fazla tekrar iletim sayısı varsayılan olarak 4'tür. İleti denemeleri sonunda MAX_RETRANSMIT değerinin üzerinde çıkıldığında hata mesajı üretilir. Eğer gönderilen mesajdaki veri işlenemez durumdaysa veya mesajda bir hata varsa göndericiye hatalı mesajın MID bilgisini içeren RST tipinde bir mesaj gönderir. RST mesajını alan gönderici aynı mesajı tekrar iletir.

Gönderilen mesajda alındı iletisi istenmediği durumlarda gönderilecek mesajın tipi NON olarak ayarlanmalıdır. Alıcı NON mesajını aldığına dair herhangi bir geri bildirimde bulunmaz. Sunucu gönderilen NON tipindeki istek mesajının yanıtını yine NON tipindeki bir yanıt mesajı ile cevaplar. Bu durum Şekil 5.14'te ifade edilmiştir. Bir sensör, sürekli sunucuya veri gönderiyorsa bu durumda her mesaj için ulaşım bilgisinin alınması ağ ortamında aşırı trafik yaratacaktır. Bu tip mesajlar NON etiketi ile gönderilmelidir. Buna ek olarak CoAP cihazları multicast mesajlarını da NON etiketi ile gönderirler.



Şekil 5.14. CoAP NON iletişimi

Eğer mesajdaki veri işlenemez durumdaysa veya mesajda bir hata varsa göndericiye hatalı mesajın MID bilgisini içeren RST tipinde bir mesaj gönderilir. RST mesajını alan gönderici aynı mesajı tekrar iletir.

5.3.3. CoAP Mesaj Yöntemleri

CoAP, HTTP benzeri istek/yanıt (request/response) modeli altında çalışır. CoAP istemcisi istek mesajlarını CoAP sunucusuna iletir ve CoAP sunucusu ise bu mesajların yanıtlarını CoAP istemcisine gönderir. Fakat CoAP'ta, HTTP'nin aksine, istek ve yanıtlar önceden kurulmuş bir bağlantı üzerinden gönderilmez.

CoAP istek veya yanıt mesajları, GET, PUT, PUSH, DELETE gibi çeşitli yöntemleri içermektedir. Bu komutlar istek veya yanıt tipini belirler.

- GET: URI adresinde bulunan kaynağın ürettiği verileri almak için kullanılan yöntemdir. GET yöntemi ile gönderilen isteğin yanıt mesajında bulunan 2.05 kodu içeriği (Content), 2.03 kodu ise verinin geçerli olduğunu belirtir.
- POST: POST yöntemi, gönderilen mesajın işlenmesini belirtir. Genellikle işlem yeni bir kaynağın oluşturulması, hedef kaynağın güncellenmesi ya da silinmesidir. Eğer sunucuda kaynak başarı ile oluşturuldu ise 2.01 oluşturuldu (Created) kodu, bir güncelleme talebi iletilip başarılı bir şekilde işlem

tamamlandıysa 2.04 değiştirildi (Changed) kodu veya bir silme talebi iletilip başarı ile işlem tamamlandıysa 2.02 silindi (Deleted) kodu geri döndürülür.

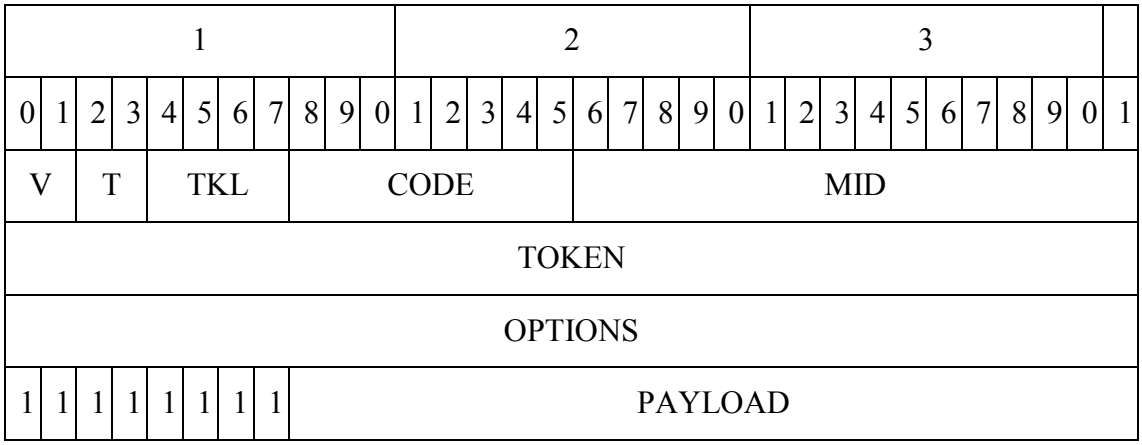
- PUT: Gönderilen istek içerisindeki URI adresi ile tanımlanan kaynağın güncellenmesini veya oluşturulmasını talep eden yöntemdir. Eğer değiştirme başarı ile tamamlanırsa 2.04 değiştirildi (Changed) kodu geri döndürülür. İstekte belirtilen URI adresi üzerinde bir kaynak yok ise yeni bir kaynak oluşturulur ve bu sefer 2.01 oluşturuldu (Created) kodu geri döndürülür. Eğer kaynak değiştirilemez veya oluşturulamaz ise HOULD mesajı geri döndürülür.
- DELETE: Gönderilen istek URI adresi üzerinde tanımlanan kaynağın silinmesini isteyen yöntemdir. İşlem tamamlandığında 2.02 silindi (Deleted) kodu geri döndürülür.

İstek mesajlarına göre 2 ile başlayan yanıt kodları işlemin başarılı, 4 ile başlayanlar istemci hatalarını, 5 ile başlayanlar ise sunucu hatalarını belirtir. Dönüş kodları HTTP kodları ile oldukça benzerlik gösterir. Buna göre bazı hatalar Çizelge 5.10'da verilmiştir.

Çizelge 5.10. CoAP yöntemleri yanıt kodları

Başarı İşlem		**İstemci Hataları**		**Sunucu Hataları**	
Kod	**Mesaj**	**Kod**	**Mesaj**	**Kod**	**Mesaj**
2.01	Created	4.00	Bad Request	5.00	Internal Server Error
2.02	Deleted	4.01	Unauthorized	5.01	Not Implemented
2.03	Valid	4.02	Bad Option	5.02	Bad Gateway
2.04	Changed	4.03	Forbidden	5.03	Service Unavailable
2.05	Content	4.04	Not Found	5.04	Gateway Timeout
		4.05	Method Not Allowed	5.05	Proxying Not Supported
		4.06	Not Acceptable		
		4.12	Precondition Failed		
		4.13	Request Entity Too Large		
		4.15	Unsupported Content-Format		

Gönderilen istek ile yanıt mesajlarını eşleştirmek için token adlı bir anahtar kullanılır. 0 ile 8 bayt arasında boyutu olan token, MID'den farklı bir değerdir ve mesajlarda bulunmalıdır. Seri olarak istek gönderilmesi gibi bazı özel durumlarda token boş bırakılabilir. Token istemci tarafından oluşturulur ve yanıt mesajı içerisinde aynen yer alır. Bu sebeple bazı kaynaklarda yanıt ID olarak belirtilir. İstemci bir sunucuya gerçekleştirdiği her isteğinde ayrı token kullanır. CoAP mesaj formatı Şekil 5.15' verilmiştir.



Şekil 5.15. CoAP mesaj formatı

- V: 2-bit uzunluğundaki versiyon bölümü CoAP sürüm numarasını gösterir. Standart versiyon "01" değerini alır. Henüz başka bir sürüm üretilmediğinden "01" dışındaki diğer değerler göz ardı edilmektedir.
- T: 2-bit uzunluğundaki tip bölümü CoAP mesajının tipini belirtir. Bu tipleri Çizelge 5.9'da verilen CON, NON, ACK, RST'dir.
- TKL: Token değerinin uzunluğu bu bölümde verilen 4-bitlik bir sayı ile ifade edilir.
- CODE: 8-bitlik bir alan olan kod bölümü "c.dd" şeklindeki bir yapıya sahiptir. c değeri 0-7 arasındaki sınıf değerini, dd değeri ise 00-31 arasında olabilen alt alan değerini tanımlar. Bu alandaki değerler Çizelge 5.11'de verilmiştir.
- MID: Her iletiye verilen benzersiz bir anahtardır.

Çizelge 5.11. CoAP CODE alanı değerleri

Kod	Açıklama
0.00	Boş bir mesajı gösterir.
0.01	GET yöntemini gösterir.
0.02	POST yöntemini gösterir.
0.03	PUT yöntemini gösterir.
0.04	DELETE yöntemini gösterir.
0.05-0.31	Gelecekte yeni versiyonlarda olabilecek istek kodları için ayrılmıştır.
1.00-1.31	Gelecekte olabilecek yeni durumlar için ayrılmıştır.
2.00-5.31	Çizelge 5.10'da verilen CoAP yanıt kodlarını gösterir.
6.00-7.31	Gelecekte olabilecek yeni durumlar için ayrılmıştır.

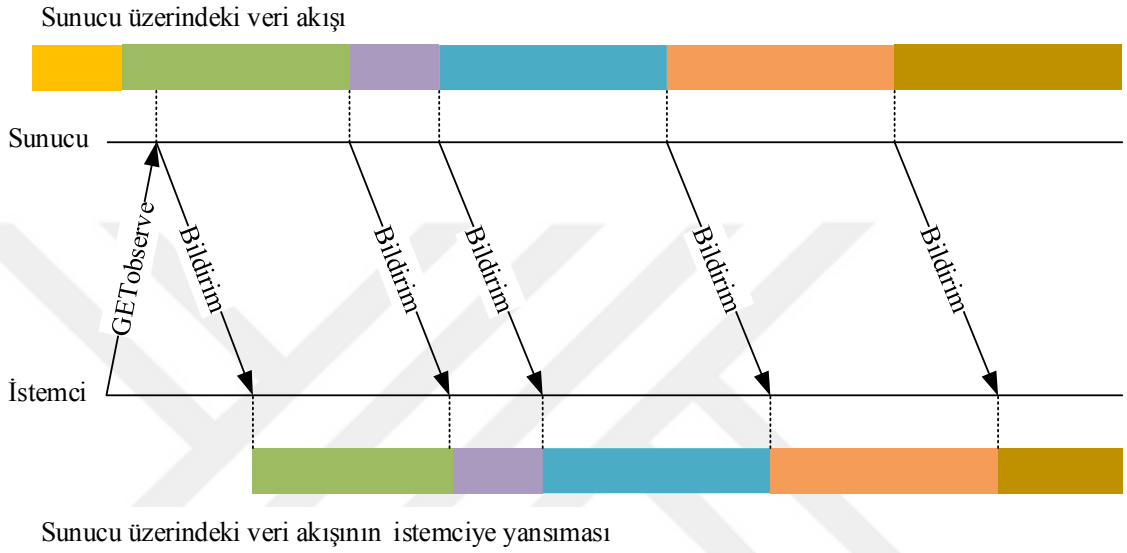
CoAP, tepki süresini azaltmak ve band genişliğini daha verimli kullanmak amacıyla önbellekleme (caching) tekniğini kullanır. Önbellekleme işleminde yeni bir isteğin yanıtının önceki yanıt mesajları ile verilmesi sağlanır. Önbelleğe alınan yanıt mesajı doğrulanmış ve güncel olmalıdır. Yanıtın güncel olup olmadığı sunucu tarafındaki Max-Age değişkeni ile belirlenir. Eğer önbellekleme yapılmak istenmiyorsa kaynak sağlayıcısı tarafından Max-Age değişkeni 0 olarak ayarlanır.

5.3.4. CoAP Çalışma Modları

CoAP kullanan nesnelerin interneti uygulamalarındaki kısıtlı cihazlar için farklı ortamlarda farklı gereksinimleri karşılayan, protokol performansını ve yeteneklerini artırabilecek çeşitli CoAP modları bulunmaktadır. Bu seçenekler aşağıda verilmiştir:

5.3.4.1 Gözlemci Modu

Gözlemci, CoAP istemcisinin kaynakları gözlemlemesine izin veren basit bir kural kümesidir. Bu şekilde CoAP istemcisi belirli bir kaynak için bir okuyucu olarak hareket edebilmektedir. Gözlemlenen kaynakta bir değişiklik olduğunda istemci gerekli bildirimleri sunucudan alır.



Şekil 5.16. CoAP gözlemci modu

Şekil 5.16'de gözlemci süreci ifade edilmektedir. Buna göre istemci bir GET isteği ile sunucudaki bir kaynağa gözlem isteğini göndermiş ve kendini kaydettirmiştir. Sunucu ise kaynaktaki değişikliği istemciye bildirmektedir. Genellikle bu bildirimler Max-Age değerinin sona ermesine kadar sunucudan istemciye iletilir. Max-Age değeri sona erdiğinde istemci aynı token değerini kullanarak tekrar kaynağa sunucu üzerinden kayıt olabilir ve kaynağı gözlemeye devam edebilir. Gözlemci modu, yayın/abone mimarisine benzemektedir. Fakat bildirimler tüm istemcilere değil sadece kaynağa kayıt olan istemcilere iletilir. Ayrıca kaynakta bir değişiklik yok ise istemcilere bildirim gönderilmez. Bu durum da kısıtlı ağlarda mesajların minimuma indirilmesini sağlar.

5.3.4.2 Multicast Grup İletişimi Modu

CoAP multicast grup iletişimi, uç noktalar arasında bire-çok ilişkidir yani diğer bir deyişle bir cihazın birden fazla cihazı tetiklemesi durumudur. Örneğin bir tesiste bulunan CoAP istemcisinin diğer tüm istemcilere yayın göndererek hepsini aktif etmesi durumunda multicast grup iletişimi kullanılabilir. CoAP grup iletişimi komutları UDP üzerinden IP multicast ile gönderilirken, yapılan işlemin yanıtları ise yine UDP üzerinden IP unicast ile alınır. DTSL multicast işlemini desteklemediğinden grup iletişimi güvensiz bir şekilde NoSec modunda gerçekleştirilir.

5.3.4.3 Blok Transfer Modu

CoAP mesajlarının çoğunlukla UDP paketlerine uygun, küçük boyutlara sahip olması beklense de bazı durumlarda daha büyük veri aktarımı yapılması gerekebilir. Bu durumda blok transfer modu tercih edilmelidir. Blok transfer modunda blok boyutu SZX değişkeninde, gönderilen bloğun sıra numarası NUM değişkeninde ve beklenecek bloğun bulunup bulunmadığı bilgisi ise M değişkeninde saklanır. Böylelikle gelen veriler istemcide M değişkenine bağlı olarak SZX blok boyutuna göre birleştirilerek işlenir.

5.3.5. CoAP ve Güvenlik

CoAP içerisindeki tek güvenlik mekanizması UDP den kaynaklı DTLS üzerinden veri aktarımıdır. DTLS, UDP protokolü altında bütünlük, kimlik doğrulama ve gizlilik gibi güvenlik hizmetlerini sağlayan SSL protokolünün bir türevidir. DTLS kullanan cihazların, ön hazırlık aşamasında sağlanan kontrol listelerine ve temel materyallere erişim bilgisine sahip olması gerekir. Bu cihazlar aşağıda verilen 4 güvenlik modu ile çalışabilir:

- NoSec: DTLS'in devre dışı bırakıldığı bu modun içerisinde hiçbir güvenlik önlemi yoktur. Bu durumda tüm güvenlik önlemleri alt katmanlara bırakılır.
- Önceden paylaşılan anahtarlar (Pre-shared keys (PSKs): Bu mod içerisinde tüm istemci ve sunucular önceden üzerinde anlaşılmış ve belirlenmiş anahtarları bilirler. Güvenli iletişim için DTLS aktif tutulur ve belirlenmiş anahtarlar üzerinden güvenli iletişim yolu açılarak veri aktarımı yapılır.

- Ortak açık anahtar (Raw Public Key): Bu modda DTLS aktiftir ve cihazın bir asimetrik anahtarı mevcuttur. Bu modda sertifika kontrolü yapılmamaktadır.
- Sertifika: Bu mod ortak açık anahtar moduna benzer fakat kontrol X509 sertifika sistemi üzerinden yapılmaktadır.

BÖLÜM 6

6. NESNELERİN İNTERNETİ UYGULAMALARININ GÜVENLİĞİ İÇİN KRİPTOGRAFİK ÇÖZÜMLER

6.1. Giriş

Bilişim dünyasında güvenlik, genel olarak üç ana ilkenin sağlanmasıdır. Bu üç ilke aşağıdaki gibidir:

- Gizlilik: Sadece yetkili kullanıcılar bilgiyi görebilmelidir.
- Bütünlük: Sadece yetkili kullanıcılar bilgiyi değiştirebilmelidir.
- Kullanılabilirlik: Yetkili kullanıcılar talep ettikleri her zaman bilgiye erişebilmelidir.

Veri transferi yapan sistemler için en önemli sorun gizliliktir. Bilişim dünyasında gizlilik, veriyi hakkı olanlar dışındaki herkesten korumak anlamına gelmektedir. Gizliliği sağlayabilmek için en önemli ve işlevsel teknik veriyi kriptografik algoritmalar kullanarak şifrelemektir. Bu bölümde genel olarak kriptografi kavramı ve günümüzde tüm sistemlerde standart olarak kabul edilen AES algoritması anlatılacaktır. Ardından nesnelerin interneti uygulamalarındaki gizliliği sağlayabilmek için kullanılabilecek hafif sıklet kriptografik algoritmalar hakkında bilgi verilecektir.

6.2. Kriptografi

Sözlüğe göre, kriptografi gizli karakterlerle yazım eylemi ya da sanatıdır, diğer yandan verilerin yalnızca belirli bireyler tarafından çözülebilmesi için kodlanması olarak özetlenebilir. Veriyi şifrelemek ve şifre çözmek için kullanılan algoritma sistemleri

kriptosistem olarak adlandırılır. Şifreleme algoritmaları, açık metin olarak adlandırılan şifrelenmemiş verileri alır ve açık metni bir gizli mesaja dönüştürmek için bir veya daha fazla anahtar kullanır. Anahtar, yalnızca gönderici ve/veya alıcı tarafından bilinen bir sayı veya karakter dizisidir. Elde edilen gizli mesaj şifreli metindir.

Temel olarak şifreleme algoritmaları simetrik, asimetrik ve hash algoritmaları olmak üzere üç gruba ayrılır. Bu şifreleme algoritmalarından simetrik algoritmalar, şifreleme ve şifre çözme işlemlerinde aynı anahtarı (gizli anahtarı) kullanır. Asimetrik şifreleme algoritmaları ise şifreleme için herkesin ulaşabileceği açık bir anahtar kullanılırken şifre çözme işlemi için gizli bir anahtar kullanır. Son olarak Hash algoritmaları, verinin özetini oluşturmak için kullanılırlar ve kimlik denetiminin sağlanmasında büyük rol oynarlar (Yavuzer Aslan, Sakallı, & Aslan, 2012). Simetrik şifreleme algoritmaları şifreleme ve şifre çözme işlemi için tek anahtar (gizli anahtar) kullanır. Açık metin gizli anahtar kullanılarak şifrelenir ve karşı tarafa iletilir. Şifreli mesaj tekrar aynı gizli anahtar kullanılarak açık metne dönüştürülür. Simetrik şifreleme algoritmaları blok şifreler ve akan şifreler olmak üzere 2 kategoride incelenebilir. Blok şifreleme algoritmaları açık metni sabit uzunluklu blok adı verilen bit grupları halinde işler. Bloklar bir anahtar aracılığı ile şifrelenerek şifreli metin ortaya çıkar. Şifre çözme işleminde yine aynı anahtar sayesinde şifreli metin açık metin haline getirilir (Aslan, 2013).

1990'lı yıllara gelindiğinde DES (Veri Şifreleme Standardı) (FIBS PUB 46-2, 1988) NIST (Ulusal Standartlar ve Teknoloji Enstitüsü) tarafında güvenli bir şifreleme algoritması olarak görülmekteydi. 1991 (Biham & Shamir, 1991) ve 1993 (Matsui, 1993) yıllarında DES'in kırılabileceği gösterildi. 2001 yılında Rijndael şifreleme algoritması AES (Gelişmiş Şifreleme Standardı) (FIBS PUB197, 2001) adıyla yeni şifreleme standardı olarak kabul gördü.

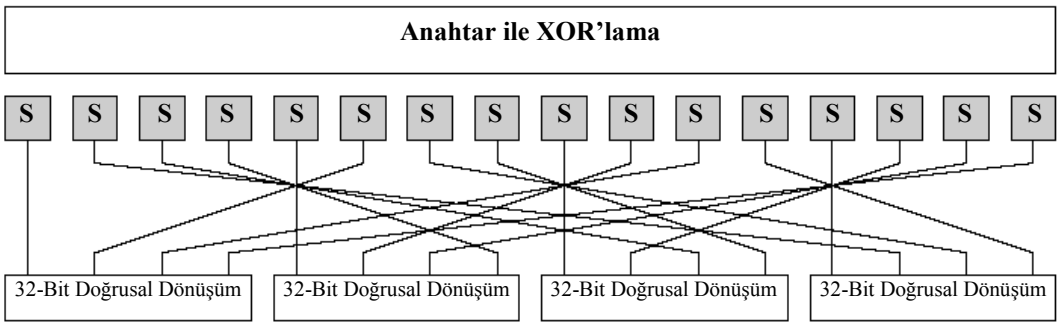
6.3. Gelişmiş Şifreleme Standardı - AES

AES 128 bit veri bloklarını 128, 192, 256 bit anahtar seçenekleri ile şifreleyen bir algoritmadır. SPN mimarisi tabanlıdır. Döngü sayısı anahtar uzunluğuna göre değişmektedir. 128 bit anahtar uzunluğu için AES şifreleme algoritması 10 döngüde şifreleme yaparken 192 ve 256 bit anahtar uzunlukları için sırasıyla 12 ve 14 döngüde şifreleme yapmaktadır (Yavuzer Aslan vd., 2012).

AES algoritmasında her döngü dört katmandan oluşur. Her döngü dört adım içerir:

- Byte Yer değiştirme (SubBytes),
- Satırları Öteleme (ShiftRows),
- Sütunları Karıştırma (MixColumns),
- Döngü Anahtarı Ekleme (AddRoundKey).

İlk olarak 128 bit veri 4×4 byte matrisine dönüştürülür. Bu matrise durum (state) matrisi denir. Her döngüde sırasıyla gerçekleştirilen yukarıdaki adımlardan byte yer değiştirme adımında 8 bit (byte) değerler farklı 8 bit (byte) değerler ile yer değiştirilir. Bu dönüşüm doğrusal olmayan bir dönüşümdür ve GF(2^8) sonlu cisminde ters haritalama tabanlıdır. Satırları öteleme adımında byte değerlerinin permütasyonu ile byte değerlerinin sırası değiştirilirken, sütunları karıştırma doğrusal dönüşümde 32 bit giriş değerlerinden sabit bir matris çarpımı yardımıyla 32 bit çıkış değerleri elde edilmektedir. Diğer yandan son adım olan döngü anahtarı ekleme evresinde 128 bit anahtar seçeneği ile şifreleme yapan AES şifresi için anahtar planlama evresinden gelen 128 bit anahtar değeri ile o anki blok XOR'lama işlemine alınır. Tek döngülük SPN mimarisine uygun AES algoritması Şekil 6.1'de gösterilmektedir (Aslan, 2013).



Şekil 6.1. AES blok şifresindeki tek döngülük SPN mimarisi

AES güvenli bir standart olmasına rağmen özellikle donanım olarak kısıtlı cihazlar için pek uygun görülmemektedir. Özellikle, düşük güç ile çalışması gereken uygulamalar için AES dışında yeni bir çözüme ihtiyaç olduğu bilinmektedir. Çünkü AES'in kabul edilişinden bu yana geçen 16 yılda teknolojik olarak birçok yenilik ortaya çıkmıştır. Diğer yandan nesnelerin interneti uygulamalarında kullanılan RFID ve benzeri

cihazlar için 2000 GE[14] (Eşdeğer Kapı) bütçesinin makul olduğu ve buna yönelik entegre ve devrelerin üretilmesi gerektiği konusunda uzlaşılmıştır (Juels & Weis, 2005). Fakat AES uygulamaları en iyi şartlarda 2400 GE bütçeye sahiptir (Moradi, Poschmann, Ling, Paar, & Wang, 2011).

AES algoritmasının sağladığı 128, 192 ve 256-bit yüksek güvenlik önlemleri özellikle nesnelerin interneti uygulamaları için her durumda gerekli değildir. Örneğin RFID kimlik doğrulama protokolü, yalnızca 64 bitlik miktarların şifrelenmesini isteyebilir ve bu durumda AES 128 bit kullanılır. Fakat geriye kalan 64-bitlik bölge ekleme algoritmaları (padding) ile doldurulur. Bu durum sonucunda gereksiz işlemler gerçekleşecektir.

6.4. Hafif Sıklet Kriptografi

Nesnelerin interneti içerisinde kullanılacak kriptografik algoritma çözümleri bu alanda kullanılan cihazların kısıtlı kaynaklarına göre uygun bir biçimde tasarlanmalı ve uygulanmalıdır. Bu zorunluluk küçük boyutlu, az enerji kullanan, az yer kaplayan (RFID etiketleri, sensörler, temassız akıllı kartlar vb.) sistemler için daha etkin olarak kullanılabilecek hafif sıklet (lightweight) kriptografi adı altında yeni bir şifreleme alanını doğurmuştur.

Hafif sıklet kriptografinin amacı, kısıtlı kaynaklar (alan, güç tüketimi, enerji tüketimi) kullanılarak bilgi güvenliğini sağlayan algoritmalar sunmaktır. Bu amacı gerçekleştirmek için birçok hafif sıklet algoritma üretilmiştir. Geliştirilen algoritmaların çoğunluğu blok şifreleme algoritmalarıdır. PRESENT (Bogdanov vd., 2007), kısıtlı donanım ortamları için önerilen ilk hafif sıklet şifreleme algoritmalarından biridir. Bunun dışında CLEFIA (Mouha vd., 2014), LBLOCK (W. Wu & Zhang, 2011), CHASKEY (Shirai, Shibutani, Akishita, Moriai, & Iwata, 2007), DESLX (Leander, Paar, Poschmann, & Schramm, 2007), HIGHT (Hong vd., 2006), HUMMINGBIRD (Engels, Saarinen, Schweitzer, & Smith, 2011), ITUBEE (Karakoç, Demirci, & Harmancı, 2013), KHUDRA (Kolay & Mukhopadhyay, 2014), KLEIN (Gong, Nikova, & Law, 2011), MCRYPTON (Lim & Korkishko, 2005), PICCOLO (Shibutani vd., 2011), PRINCE (Borghoff vd., 2012), ROADRUNNER (Baysal & Şahin, 2015), SPARX (Dinu vd.),

[14] Mantıksal devrelerin imalat teknolojisinden bağımsız olarak karmaşıklığının belirlenmesine izin veren bir ölçü birimdir.

TWINE(Suzaki, Minematsu, Morioka, & Kobayashi, 2011), FANTOMAS (Grosso, Leurent, Standaert, & Varıcı, 2014), MIBS (Izadi, Sadeghiyan, Sadeghian, & Khanooki, 2009) gibi birçok algoritma geliştirilmiştir. Adı geçen algoritmaların temel yapıları Çizelge 6.1'de verilmiştir.

Tez çalışması kapsamındaki test ortamlarındaki şifreleme ve şifre çözme işlemleri CLEFIA, PRESENT, PICCOLO, PRINCE ve LBLOCK algoritmaları ile yapılmıştır. Bu bölümde adı geçen algoritmaların yapıları hakkında bilgiler verilmiştir.

Çizelge 6.1. Bazı hafif sıklet şifreleme algoritmalarının temel yapıları

Şifreleme Algoritması	Blok Boyutu	Anahtar Boyutu	Döngü Sayısı	Mimari	Uygulama Alanı
PRESENT	64	80,128	31	SPN	Donanım
CLEFIA	128	128,192,256	18,22,26	Feistel	Yazılım, Donanım
PICCOLO	64	80,128	25,31	Feistel	Donanım
PRINCE	64	128	12	SPN	Donanım
LBLOCK	64	80	32	Feistel	Yazılım, Donanım
CHASKEY	128	128	8	ARX	Yazılım
DESLX	64	184	16	Feistel	Yazılım, Donanım
HIGHT	64	128	32	Feistel	Donanım
LED	64	80	48	Feistel	Yazılım, Donanım
ITUBEE	80	80	20	Feistel	Yazılım, Donanım
KLEIN	64	64,80,96	12,16,20	SPN	Donanım
MCRYPTON	64	64,96,128	12	SPN	Yazılım, Donanım
MIBS	64	64,80	31	Feistel	Donanım
ROADRUNNER	64	80,128	10,12	SPN	Donanım
SPARX	64,128	128,256	24,40	SPN	Yazılım, Donanım
TWINE	64	80	36	Feistel	Yazılım, Donanım
FANTOMAS	128	128	10	SPN	Yazılım
SIMON	64	96	42	Feistel	Yazılım, Donanım
SPECK	64	96	26	Feistel	Yazılım, Donanım

6.4.1. CLEFIA Blok Şifreleme Algoritması

CLEFIA Sony tarafından geliştirilmiş 128 bit veri bloklarını 128, 192, 256 bit anahtar seçenekleri ile şifreleyen bir algoritmadır. Feistel mimarisini kullanan algoritma 128 bit anahtar uzunluğu için 18 döngüde şifreleme yaparken 192 ve 256 bit anahtar uzunlukları için sırasıyla 22 ve 26 döngüde şifreleme yapmaktadır. Her bir döngü 4 veri yolundan ve iki adet 32-bitlik F fonksiyondan oluşur. Bu fonksiyonlar F_0 ve F_1 dir ve giriş/çıkışları ifade 6.1 gibidir.

$$F_0, F_1: \begin{cases} \{0,1\}^{32} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32} \\ (RK_{(32)}, x_{(32)}) \mapsto y_{(32)} \end{cases} \tag{6.1}$$

CLEFIA, d dal, r döngü olmak üzere Genelleştirilmiş Feistel mimarisini ($GFN_{d,r}$) kullanmaktadır. 32 bitlik d adet X_i giriş ve Y_i çıkış ve yine 32 bitlik $dr/2$ adet RK_i değerleri için $GFN_{d,r}$ ifadesi $d = 4$ ve $d = 8$ için sırası ile ifade 6.2 ve ifade 6.3'teki gibi tanımlanmaktadır.

$$GFN_{4,r}: \begin{cases} \{\{0,1\}^{32}\}^{2r} \times \{\{0,1\}^{32}\}^{4} \rightarrow \{\{0,1\}^{32}\}^{4} \\ \left(RK_{0(32)}, \dots, RK_{2r-1(32)}, X_{0(32)}, \dots, X_{3(32)}\right) \mapsto Y_{0(32)}, \dots, Y_{3(32)} \end{cases} \tag{6.2}$$

$$GFN_{8,r}: \begin{cases} \{\{0,1\}^{32}\}^{4r} \times \{\{0,1\}^{32}\}^{8} \rightarrow \{\{0,1\}^{32}\}^{8} \\ \left(RK_{0(32)}, \dots, RK_{4r-1(32)}, X_{0(32)}, \dots, X_{7(32)}\right) \mapsto Y_{0(32)}, \dots, Y_{7(32)} \end{cases} \tag{6.3}$$

$GFN_{4,r}$ ve $GFN_{8,r}$ işleminin algoritmaları sırası ile Algoritma 6.1 ve Algoritma 6.2'de verilmiştir.

Algoritma 6.1. $GFN_{4,r}$ işleminin algoritması

$Adım1$: $T_0|T_1|T_2|T_3 \leftarrow X_0|X_1|X_2|X_3$

$Adım2$: $for\ i = 0\ \ to\ r-1$

$Adım2.1$: $T_1 \leftarrow T_1 \oplus F_0(RK_{2i}, T_0)$,

$T_3 \leftarrow T_3 \oplus F_1(RK_{2i+1}, T_2)$,

$Adım2.2$: $T_0|T_1|T_2|T_3 \leftarrow T_1|T_2|T_3|T_0$

$Adım3$: $Y_0|Y_1|Y_2|Y_3 \leftarrow T_3|T_0|T_1|T_2$

Algoritma 6.2. $GFN_{8,r}$ işleminin algoritması

$Adım1: T_0|T_1|\ldots|T_7 \longleftarrow X_0|X_1|\ldots.|X_7$

$Adım2: for\ i = 0\ \ to\ r-1$

$Adım2.1: T_1 \longleftarrow T_1 \oplus F_0(RK_{4i}, T_0),$

$T_3 \longleftarrow T_3 \oplus F_1(RK_{4i+1}, T_2),$

$T_5 \longleftarrow T_5 \oplus F_0(RK_{4i+2}, T_4),$

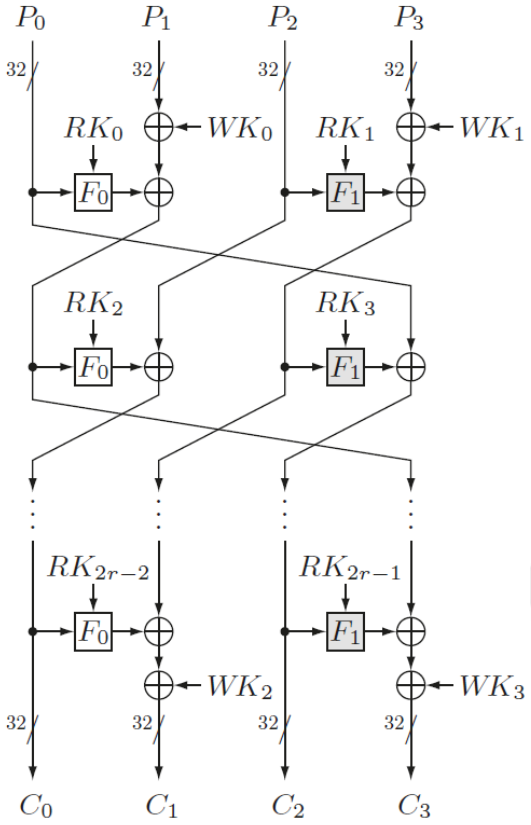
$T_7 \longleftarrow T_7 \oplus F_1(RK_{4i+3}, T_6)$

$Adım2.2: T_0|T_1|\ldots|T_7 \longleftarrow T_1|T_2|\ldots|T_0$

$Adım3: Y_0|Y_1|\ldots|Y_7 \longleftarrow T_7|T_0|\ldots|T_5|T_6$

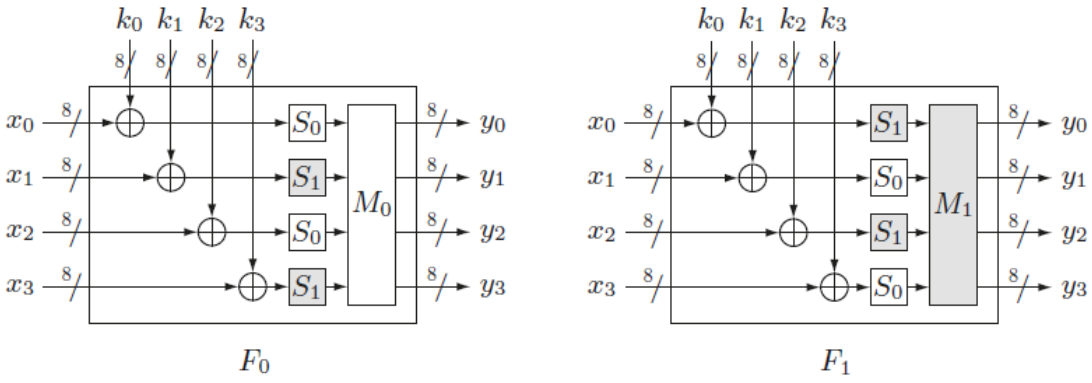
Şifreleme işleminde P açık metin ve C şifreli metin olmak üzere $P, C \in \{0,1\}^{128}$ şeklinde bir ifadedir. Ayrıca $P_i, C_i \in \{0,1\}^{32}$ $(0 \leq i < 4)$ olmak üzere $P = P_0|P_1|P_2|P_3$ biçimindeki 4 parça veri yollarında işlenerek $C = C_0|C_1|C_2|C_3$ şeklindeki şifreli metin elde edilir. İlk ve son döngüde anahtar beyazlatma işlemi için $WK_0, WK_1, WK_2, WK_3 \in \{0,1\}^{32}$ kullanılır. Anahtar üretim aşamasından gelen döngü anahtarları r döngü sayısı olmak üzere $RK_i \in= \{0,1\}^{32}$ $(0 \leq i < 2r)$ olarak belirtilir.

İlk aşama olarak açık metnin P_1 ve P_3 bölümü WK_0 ve WK_1 ile XOR işlemine alınır. Daha sonrasında P_0 bloğu RK_0 anahtarı ile F_0 fonksiyonuna alınarak $F_0(RK_0, P_0)$ işlemi gerçekleştirilir. Çıkan sonuç $P_1 \oplus WK_0$ işleminin sonucu ile XOR işlemine alınır. Aynı şekilde P_2 bloğu RK_1 anahtarı ile F_1 fonksiyonuna alınarak $F_1(RK_1, P_2)$ işlemi gerçekleştirerek çıkan sonuç $P_3 \oplus WK_1$ işleminin sonucu ile XOR işlemine alınır. Döngü sonunda oluşan $\acute{P_0}|\acute{P_1}|\acute{P_2}|\acute{P_3}$ bloğu $\acute{P_0} \rightarrow P_3$, $\acute{P_1} \rightarrow P_0$, $\acute{P_2} \rightarrow P_1$, $\acute{P_3} \rightarrow P_2$ olacak şekilde değiştirilerek bir sonraki döngüye geçilir. Son döngünün fonksiyon çıkışlarındaki P_1 ve P_3 bölümü WK_2 ve WK_3 ile XOR işlemine alınır. Bu işlemler Şekil 6.2'de verilmiştir.



Şekil 6.2. CLEFIA şifreleme döngü yapısı

Şifreleme işlemi içerisinde F_0 ve F_1 fonksiyonları $F_0, F_1 : (RK, x) \rightarrow y$ işlemini Algoritma 6.3 ve Algoritma 6.4'e göre gerçekleştirmektedir. Ayrıca fonksiyonların diyagramları Şekil 6.3'te verilmiştir.



Şekil 6.3. F_0 ve F_1 fonksiyonları

Algoritma 6.3. F_0 fonksiyonun algoritması

$Adım1: T \leftarrow RK \oplus x$

$Adım2: T = T_0|T_1|T_2|T_3 \; ve \; T_i \in \{0,1\}^8 \; olmak$ üzere

$T_0 \leftarrow S_0(T_0)$

$T_1 \leftarrow S_1(T_1)$

$T_2 \leftarrow S_0(T_2)$

$T_3 \leftarrow S_1(T_3)$

$Adım3: y = y_0|y|y_2|y_3 \; ve \; y_i \in \{0,1\}^8 \; olmak$ üzere

$(y_0, y_1, y_2, y_3)^t = M_0(T_0, T_1, T_2, T_3)^t$

Algoritma 6.4. F_1 fonksiyonun algoritması

$Adım1: T \leftarrow RK \oplus x$

$Adım2: T = T_0|T_1|T_2|T_3 \; ve \; T_i \in \{0,1\}^8 \; olmak$ üzere

$T_0 \leftarrow S_1(T_0)$

$T_1 \leftarrow S_0(T_1)$

$T_2 \leftarrow S_1(T_2)$

$T_3 \leftarrow S_0(T_3)$

$Adım3: y = y_0|y|y_2|y_3 \; ve \; y_i \in \{0,1\}^8 \; olmak$ üzere

$(y_0, y_1, y_2, y_3)^t = M_1(T_0, T_1, T_2, T_3)^t$

Algoritmalarda belirtilen S_0 ve S_1 doğrusal olmayan 8 bitlik S-Kutularıdır. S-Kutularının F_0 ve F_1 fonksiyonlarındaki kullanım sırası farklıdır. CLEFIA algoritmasında kullanılan S-Kutuları Çizelge 6.2'de verilmiştir.

Çizelge 6.2. CLEFIA S_0 ve S_1 S-Kutuları

S_0 S-Kutusu																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	57	49	D1	C6	2F	33	74	FB	95	6D	82	EA	0E	B0	A8	1C
1	28	D0	4B	92	5C	EE	85	B1	C4	0A	76	3D	63	F9	17	AF
2	BF	A1	19	65	F7	7A	32	20	06	CE	E4	83	9D	5B	4C	D8
3	42	5D	2E	E8	D4	9B	0F	13	3C	89	67	C0	71	AA	B6	F5
4	A4	BE	FD	8C	12	00	97	DA	78	E1	CF	6B	39	43	55	26
5	30	98	CC	DD	EB	54	B3	8F	4E	16	FA	22	A5	77	09	61
6	D6	2A	53	37	45	C1	6C	AE	EF	70	08	99	8B	1D	F2	B4
7	E9	C7	9F	4A	31	25	FE	7C	D3	A2	BD	56	14	88	60	0B
8	CD	E2	34	50	9E	DC	11	05	2B	B7	A9	48	FF	66	8A	73
9	03	75	86	F1	6A	A7	40	C2	B9	2C	DB	1F	58	94	3E	ED
A	FC	1B	A0	04	B8	8D	E6	59	62	93	35	7E	CA	21	DF	47
B	15	F3	BA	7F	A6	69	C8	4D	87	3B	9C	01	E0	DE	24	52
C	7B	0C	68	1E	80	B2	5A	E7	AD	D5	23	F4	46	3F	91	C9
D	6E	84	72	BB	0D	18	D9	96	F0	5F	41	AC	27	C5	E3	3A
E	81	6F	07	A3	79	F6	2D	38	1A	44	5E	B5	D2	EC	CB	90
F	9A	36	E5	29	C3	4F	AB	64	51	F8	10	D7	BC	02	7D	8E

S_1 S-Kutusu																
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	6C	DA	C3	E9	4E	9D	0A	3D	B8	36	B4	38	13	34	0C	D9
1	BF	74	94	8F	B7	9C	E5	DC	9E	07	49	4F	98	2C	B0	93
2	12	EB	CD	B3	92	E7	41	60	E3	21	27	3B	E6	19	D2	0E
3	91	11	C7	3F	2A	8E	A1	BC	2B	C8	C5	0F	5B	F3	87	8B
4	FB	F5	DE	20	C6	A7	84	CE	D8	65	51	C9	A4	EF	43	53
5	25	5D	9B	31	E8	3E	0D	D7	80	FF	69	8A	BA	0B	73	5C
6	6E	54	15	62	F6	35	30	52	A3	16	D3	28	32	FA	AA	5E
7	CF	EA	ED	78	33	58	09	7B	63	C0	C1	46	1E	DF	A9	99
8	55	04	C4	86	39	77	82	EC	40	18	90	97	59	DD	83	1F
9	9A	37	06	24	64	7C	A5	56	48	08	85	D0	61	26	CA	6F
A	7E	6A	B6	71	A0	70	05	D1	45	8C	23	1C	F0	EE	89	AD
B	7A	4B	C2	2F	DB	5A	4D	76	67	17	2D	F4	CB	B1	4A	A8
C	B5	22	47	3A	D5	10	4C	72	CC	00	F9	E0	FD	E2	FE	AE
D	F8	5F	AB	F1	1B	42	81	D6	BE	44	29	A6	57	B9	AF	F2
E	D4	75	66	BB	68	9F	50	02	01	3C	7F	8D	1A	88	BD	AC
F	F7	E4	79	96	A2	FC	6D	B2	6B	03	E1	2E	7D	14	95	1D

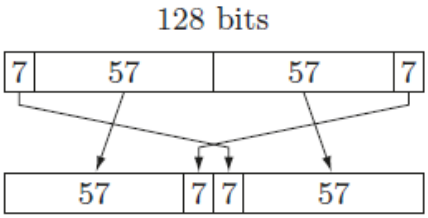
Algoritmalarda kullanılan M_0 ve M_1 doğrusal dönüşüm matrisleri hadamard formda 4×4 boyutunda olup matris çarpımları sonucu $GF(2^8)$ de $x^8 + x^4 + x^3 + x^2 + 1$ indirgenemez polinomu ile oluşturulan cisim içerisinde tanımlıdır. M_0 ve M_1 matrisleri ifade 6.4'te verilmiştir.

$$M_0 = \begin{bmatrix} 01_h & 02_h & 04_h & 06_h \\ 02_h & 01_h & 06_h & 04_h \\ 04_h & 06_h & 01_h & 02_h \\ 06_h & 04_h & 02_h & 01_h \end{bmatrix} \quad M_1 = \begin{bmatrix} 01_h & 08_h & 02_h & 0A_h \\ 08_h & 01_h & 0A_h & 02_h \\ 02_h & 0A_h & 01_h & 08_h \\ 0A_h & 02_h & 08_h & 01_h \end{bmatrix} \tag{6.4}$$

CLEFIA şifreleme algoritmasının anahtar planlama kısmı 128, 192 ve 256 bit anahtarları destekler. Planlama aşaması sonunda WK_i $(0 \leq i < 4)$ beyazlatma anahtarı ve RK_i $(0 \leq i < 2r)$ döngü anahtarları üretilir. Bu aşama içerisinde giriş bitlerinin yerlerini değiştirerek yeni çıkış bitlerini üreten *DoubleSwap* adında özel bir fonksiyon kullanılır. *DoubleSwap* fonksiyonu $\{0,1\}^{128} \rightarrow \{0,1\}^{128}$ olmak üzere ifade 6.5'te tanımlanmıştır.

$$\begin{gathered} X_{128} \mapsto Y_{128} \\ Y = X[7-63] \mid X[121-127] \mid X[0-6] \mid X[64-120] \end{gathered} \tag{6.5}$$

Burada $X[a-b]$ işlemi X giriş verisinin a bitinden b bitine kadar olan kısmın alınması olarak özetlenebilir. Bu işlem Şekil 6.4'te verilmiştir.



Şekil 6.4. DoubleSwap işlemi

Anahtar planlama 2 aşamada tamamlanmaktadır. İlk olarak 128, 192 veya 256 bitlik K anahtarı üzerinden ara anahtar L üretilir, ikinci olarak ise K ve L aracılığı ile beyazlatma anahtarları WK_i ve döngü anahtarları RK_i üretilir.

Anahtar planlama aşamasında 128, 192 ve 256 bitlik K anahtarına göre sıra ile 60, 84 ve 92 adet 32-bit uzunlukta $k = 128, 192, 256$ olmak üzere $CON_i^{(k)}$ sabit değerleri kullanılır. Bu değerler $P = B7E1 = (e-2)\cdot 2^{16}$ ve $Q = 243F = (\pi - 3)\cdot 2^{16}$ olmak üzere Algoritma 6.5 ile hesaplanır. Verilen algoritmada Adım 2.3 bölümündeki çarpma işlemleri sonucu $GF(2^{16})$'da $x^{16} + x^{15} + x^{13} + x^{11} + x^5 + x^4 + 1$ indirgenemez polinomu ile oluşturulan cisim içerisinde tanımlıdır. Ayrıca algoritmada belirtilen tekrar sayıları l^k ve başlangıç değerleri (Initial Values) IV^k ifadelerinin alması gereken değerler ise Çizelge 6.3'te verilmiştir.

Algoritma 6.5. $CON_i^{(k)}$ sabit değerlerinin hesaplanma algoritması

$Adım1: T \leftarrow IV^k$

$Adım2: for\ \ i = 0\ to\ \ l^k - 1$

$Adım2.1: CON_{(2i)}^{(k)} \leftarrow (T \oplus P)\ |\ (\bar{T} \lll 1)$

$Adım2.2: CON_{(2i+1)}^{(k)} \leftarrow (\bar{T} \oplus Q)\ |\ (\bar{T} \lll 8)$

$Adım2.3: T \leftarrow T\ .0x0002^{-1}$

Çizelge 6.3. Sabit sayılar için gereken değerler

k	CON Sayısı	l^k	IV^k
128	60	30	428A $= \left(\sqrt[3]{2} - 1\right)\cdot 2^{16}$
192	84	42	7137 $= \left(\sqrt[3]{3} - 1\right)\cdot 2^{16}$
256	92	46	B5C0 $= \left(\sqrt[3]{5} - 1\right)\cdot 2^{16}$

Anahtar planlama aşamasında 128 bitlik K anahtarı ile WK_i ve RK_j değerlerinin üretimi Algoritma 6.6'da verilmiştir. Buna göre $K = K_0|K_1|K_2|K_3$ anahtarı ve 24 adet 32 bitlik sabit CON_i^{128} $(0 \le i < 24)$ değerleri $GFN_{4,12}$ işlemine alınarak L değeri elde edilir. Daha sonra K ve L değerleri kullanılarak WK_i $(0 \le i < 4)$ ve RK_j $(0 \le j < 4)$ değerleri

algoritmanın 3. adımı ile üretilir. Bu aşamada 36 adet 32 bitlik sabit $CON_i^{128}(24 \leq i < 60)$ sabit değerleri kullanılmaktadır. Algoritmada belirtilen Σ fonksiyonu *doubleswap* işlemidir.

Algoritma 6.6. 128-bit anahtar için anahtar planlama aşaması

$Adım1: L \longleftarrow GFN_{4,12}(CON_0^{128}, \ldots, CON_{23}^{128}, K_0, K_1, K_2, K_3)$

$Adım2: WK_0|WK_1|WK_2|WK_3 \longleftarrow K$

$Adım3: for\ i = 0\ to\ 8$

$Adım3.1: T \longleftarrow L \oplus (CON_{24+4i}^{128}|CON_{24+4i+1}^{128}|CON_{24+4i+2}^{128}|CON_{24+4i+3}^{128})$

$Adım3.2: L \longleftarrow \Sigma(L)$

$Adım3.3: Eğer\ i\ \ tek\ ise\ T \longleftarrow T \oplus K$

$Adım3.4: RK_{4i}|RK_{4i+1}|RK_{4i+2}|RK_{4i+3} \longleftarrow T$

---

192 bitlik K anahtarı ile WK_ive RK_j değerleri ise Algoritma 6.7 ile üretilir. Buna göre öncelikle 192 bitlik $K = K_0|K_1|K_2|K_3|K_4|K_5$ anahtarı üzerinden 128 bitlik $K_L$ve K_R değerleri üretilir. Daha sonra 128 bitlik $K_L$ve K_R değerleri ve CON_i^{192} $(0 \leq i < 40)$ sabitleri $GFN_{8,10}$ işlemine alınarak $L_L$ve L_R değerlerine ulaşılır. Son olarak WK_i $(0 \leq i < 4)$ ve RK_j $(0 \leq j < 44)$ anahtarlarının üretilmesi için K_L, K_R, L_L, L_R ve CON_i^{192} $(40 \leq i < 84)$ değerleri algoritmada belirtildiği gibi kullanılır. Eğer 256 bitlik K anahtarı ile WK_ive RK_j değerleri üretilecek ise aynı algoritma $k = 256$ değeri ile çalıştırılır. Fakat bu durumda $L_L$ve L_R değerlerini üretmek için CON_i^{256} $(0 \leq i < 40)$ sabitleri ve daha sonra RK_j $(0 \leq j < 52)$ üretimi için ise CON_i^{256} $(40 \leq i < 92)$ sabitleri kullanılır.

Algoritma 6.7. 192-bit ve 256-bit anahtar için anahtar planlama aşaması

$Adım1: k = 192\ veya\ k = 256\ olarak\ ata$

$Adım2: Eğer\ k = 192\ ise\ \ K_L \leftarrow K_0|K_1|K_2|K_3 \quad K_R \leftarrow K_4|K_5|\overline{K_0}|\overline{K_1}$

$Eğer\ k = 256\ ise\ \ K_L \leftarrow K_0|K_1|K_2|K_3 \quad K_R \leftarrow K_4|K_5|K_6|K_7$

$Adım3: K_L = K_{L0}|K_{L1}|K_{L2}|K_{L3} \ \ ve\ K_R = K_{R0}|K_{R1}|K_{R2}|K_{R3}\ olmak$ üzere

$L \leftarrow GFN_{8,10}\left(CON_0^{(k)}, \dots, CON_{39}^{(k)}, K_{L0}, \dots, K_{L3}, K_{R0}, \dots, K_{R3}\right)$

$Adım4: WK_0|WK_1|WK_2|WK_3 \leftarrow K_L \oplus K_R$

$Adım5: for\ i = 0\ to\ 10\ (eğer\ k = 192)\ veya\ 12\ (eğer\ k = 256)$

$Adım5.1: Eğer\ i\ mod\ 4 = 0\ veya\ 1\ ise$

$T \leftarrow L_L \oplus (CON_{40+4i}^{(k)}|CON_{40+4i+1}^{(k)}|CON_{40+4i+2}^{(k)}|CON_{40+4i+3}^{(k)})$

$L_L \leftarrow \Sigma(L_L)$

$Adım5.2: Eğer\ i\ tek\ ise\ T \leftarrow T \oplus K_R$

$Adım5.3: Diğer\ durumda$

$T \leftarrow L_R \oplus (CON_{40+4i}^{(k)}|CON_{40+4i+1}^{(k)}|CON_{40+4i+2}^{(k)}|CON_{40+4i+3}^{(k)})$

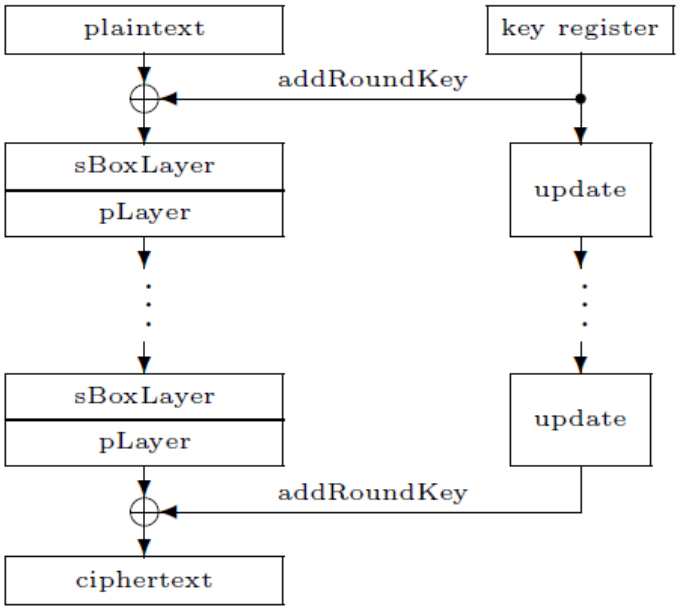
$L_R \leftarrow \Sigma(L_R)$

$Adım6: Eğer\ i\ tek\ ise\ T \leftarrow T \oplus K_L$

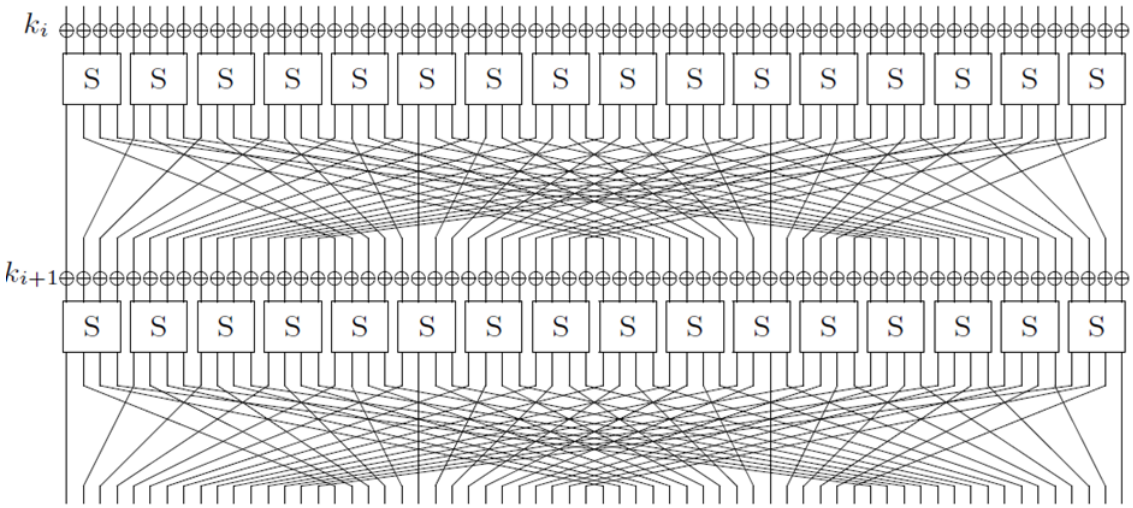
$Adım7: RK_{4i}|RK_{4i+1}|RK_{4i+2}|RK_{4i+3} \leftarrow$T

6.4.2. PRESENT Blok Şifreleme Algoritması

PRESENT, Fransa Orange Labs, Almanya Ruhr Üniversitesi ve Danimarka Teknik Üniversitesi tarafından 2007 yılında geliştirilen bir blok şifredir. Tasarımcıları A. Bogdanov, A. Poschmann, C. Paar, C. Vikkelsoe, G. Leander, L.R. Knudsen, M.J.B. Robshaw ve Y. Seurin'dur. CLEFIA gibi ISO/IEC 29192-2:2012 dokümanı ile hafif sıklet blok şifreleme algoritması olarak standartlaşmıştır. 64 bit blok uzunluğuna sahip olan PRESENT 80 veya 128 bit anahtar seçeneklerini desteklemektedir. Algoritma SPN mimarisi ile tasarlanmıştır ve 31 döngüden oluşmaktadır. PRESENT şifreleme algoritmasının diyagramı ve döngü yapısı Şekil 6.5 ve Şekil 6.6'da verilmiştir. Her döngü anahtar ekleme, doğrusal olmayan yer değiştirme ve doğrusal bitsel permütasyon katmanlarından oluşmaktadır.



Şekil 6.5. PRESENT şifreleme algoritması diyagramı



Şekil 6.6. PRESENT şifreleme döngü yapısı

Anahtar ekleme aşamasında $1 \leq i \leq 32$ olmak üzere döngü anahtarı $K_i = k_{63}^i, \dots, k_0^i$ ve döngünün giriş bitleri $b_{63}, \dots, b_0$ ile XOR işlemine ifade 6.6'daki gibi alınır. Burada 32. XOR işlemi son beyazlatma için kullanılmaktadır.

$$b_j \leftarrow b_j \oplus k_j^i \quad (0 \leq j \leq 63 \;\; ve \;\; 1 \leq i \leq 32\,) \tag{6.6}$$

Doğrusal olmayan yer değiştirme işleminde 4 bitlik S: $\mathbb{F}_2^4 \rightarrow \mathbb{F}_2^4$ şeklinde tanımlı Çizelge 6.4'te verilen S-Kutusu kullanılmaktadır.

Çizelge 6.4. PRESENT S-Kutusu

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Doğrusal bitsel permütasyon işlemi Çizelge 6.5'e göre yapılmıştır. Buna göre i konumundaki bit $P(i)$ konumuna aktarılır.

Çizelge 6.5. PRESENT permütasyon tablosu

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

PRESENT algoritması 80 ve 128 bit anahtarlar kullanarak şifreleme ve şifre çözme işlemini yapmaktadır. 80 bit için anahtar planlama aşamasında $k_{79}, k_{78}, \ldots, k_0$ biçimindeki K anahtarı bir kaydedici içerisinde saklanmaktadır. i döngüsünde kaydedici içerisindeki soldaki yani ilk 64 bit döngü anahtarı $K_i = k_{63}^i, \ldots, k_0^i$ olarak seçilir. Daha sonra 80 bitlik kaydedici Algoritma 6.8'e göre bir sonraki döngü için güncellenir.

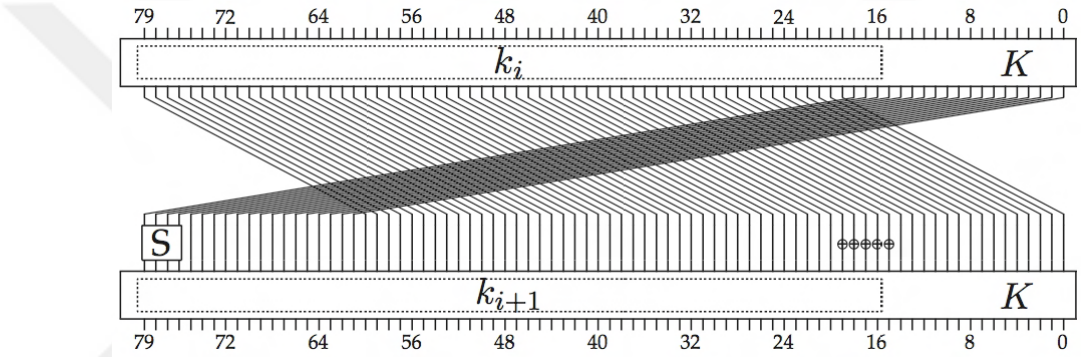
Algoritmaya göre kaydedici içerisindeki anahtar sola 61 bit döndürülür, en sol 4 bit S-Kutusu ile haritalanır ve son olarak 19-15 bit aralığındaki değerler döngü sayacı ile XOR işlemine alınır. Bu işlem Şekil 6.7'de özetlenmiştir.

Algoritma 6.8. 80 bitlik anahtar için güncelleme algoritması

$Adım1: k_{79}k_{78} \ \ldots \ k_1k_0 \leftarrow k_{18}k_{17} \ \ldots \ k_{20}\, k_{19}$

$Adım2: k_{79}k_{78}k_{77}k_{76} \ \leftarrow S[k_{79}k_{78}k_{77}k_{76}]$

$Adım3: k_{19}k_{18}k_{17}k_{16}k_{15} \ \leftarrow k_{19}k_{18}k_{17}k_{16}k_{15} \ \oplus dongu_sayaci$



Şekil 6.7. 80 bit anahtar planlama

128 bit anahtar üretim aşamasında ise 80 bite benzer bir şekilde $k_{127}, k_{126}, \ldots, k_0$ biçimindeki K anahtarı bir kaydedici içerisinde saklanmaktadır. i döngüsünde kaydedici içerisindeki en sol bitler 64 bit döngü anahtarı $K_i = k_{63}^i, \ldots, k_0^i$ olarak seçilir 128 bitlik kaydedici Algoritma 6.9'a göre bir sonraki döngü için güncellenir. Algoritmaya göre kaydedici içerisindeki anahtar sola 61 bit döndürülür, 127-124 ve 123-120 bit aralığı S-Kutusu ile haritalanır ve son olarak 66-62 bit aralığındaki değerler döngü sayacı ile XOR işlemine alınır. Bu işlem Şekil 6.8'de özetlenmiştir.

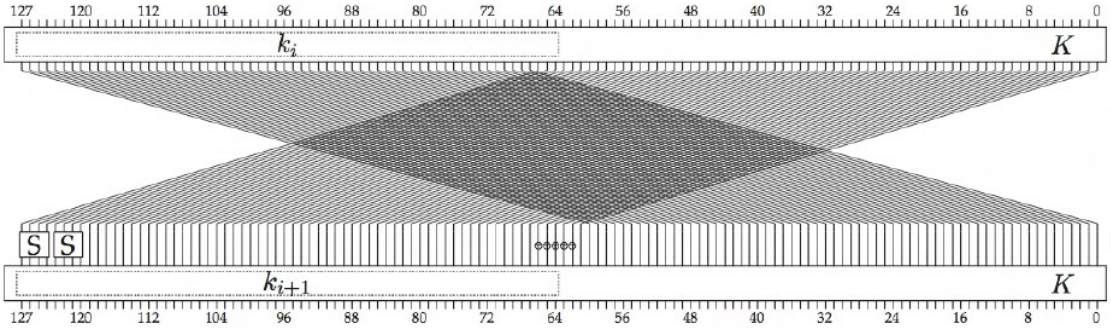
Algoritma 6.9. 128 bitlik anahtar için güncelleme algoritması

$Adım1$: $k_{127}k_{126}\ \ldots\ k_1k_0 \leftarrow k_{66}k_{65}\ \ldots\ k_{68}\,k_{67}$

$Adım2$: $k_{127}k_{126}k_{125}k_{124} \leftarrow S[k_{127}k_{126}k_{125}k_{124}]$

$Adım3$: $k_{123}k_{122}k_{121}k_{120} \leftarrow S[k_{123}k_{122}k_{121}k_{120}]$

$Adım4$: $k_{66}k_{65}k_{64}k_{63}k_{62} \leftarrow k_{66}k_{65}k_{64}k_{63}k_{62} \oplus dongu_sayaci$



Şekil 6.8.128 bit anahtar planlama

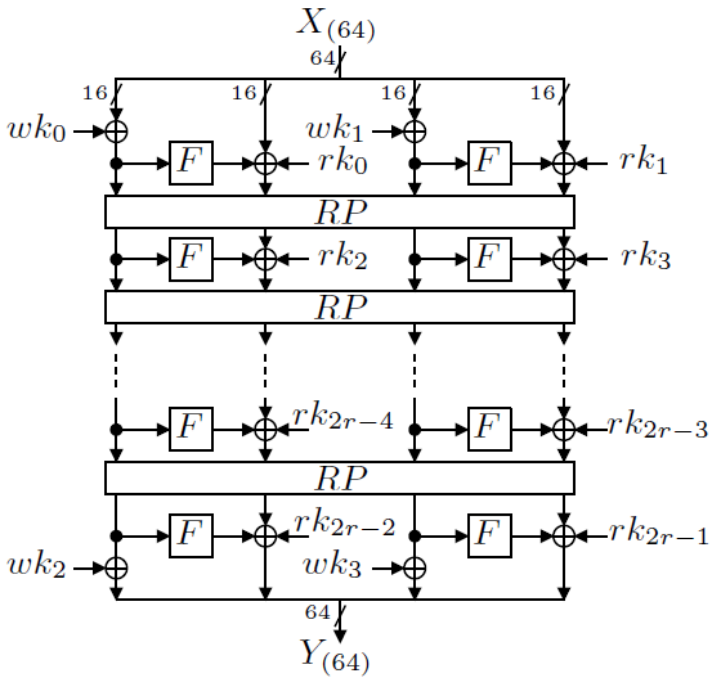
6.4.3. PICCOLO Blok Şifreleme Algoritması

PICCOLO, CLEFIA gibi Sony tarafından son derece kısıtlı kapasitedeki cihazlar için optimize edilerek geliştirilmiş 64 bit veri bloklarını 80 ve 128 bit anahtar seçenekleri ile şifreleyen bir algoritmadır. Algoritmanın tasarımcıları Atsushi Mitsuda, Harunaga Hiwatari, Kyoji Shibutani, Taizo Shirai, Takanori Isobe ve Toru Akishita'dır. Feistel mimarisini kullanan algoritma 80 bit anahtar uzunluğu için 25 döngüde şifreleme yaparken 128 bit anahtar uzunlukları için 31 döngüde şifreleme yapmaktadır.

Genel olarak şifreleme aşaması r adet döngüden oluşmakta ve her döngü G_r ile ifade edilmektedir. G_r, $X \in \{0,1\}^{64}$ olmak üzere 64 bitlik açık metin, $wk_i \in \{0,1\}^{16}$ $(0 \leq i < 4)$ ifadesini sağlayan ve sadece ilk ve son döngüde kullanılacak 4 adet 16 bitlik beyazlatma anahtarı ve $2r$ adet $rk_i \in \{0,1\}^{16}$ $(0 \leq i < 2r)$ ifadesini sağlayan giriş verilerini alarak 64 bitlik $Y \in \{0,1\}^{64}$ şifreli metni üretmektedir. G_r ifadesi ifade 6.7'deki gibi tanımlanmıştır.

$$G_r: \begin{cases} \{0,1\}^{64} \times \{\{0,1\}^{16}\}^4 \times \{\{0,1\}^{16}\}^{2r} \to \{0,1\}^{64} \\ \left(X_{(64)}, wk_{0(16)}, \dots, wk_{3(16)}, rk_{0(16)}, \dots, rk_{2r-1(16)}\right) \mapsto Y_{(64)} \end{cases} \tag{6.7}$$

Şifreleme aşamasında 64 bitlik açık metin 16 bitlik 4 parçaya ayrılır. 1. ve 3. Parça beyazlatma anahtarı wk_0 ve wk_1 ile XOR işlemine alındıktan sonra sonuçlar F fonksiyonu ile işlenir. Fonksiyon çıkışı döngü anahtarı rk_0 ve rk_1 ile XOR işlemine tabi tutulur. Çıkan sonuçlar bir sonraki döngüye aktarılmadan önce RP katmanı ile permütasyon işlemine alınır. PICCOLO şifreleme işleminin döngü yapısı Şekil 6.9'da ve algoritması Algoritma 6.10'da verilmiştir.



Şekil 6.9. PICCOLO şifreleme döngü yapısı

Algoritma 6.10. PICCOLO G_r şifreleme algoritması

$Adım1: X_{0(16)}|X_{1(16)}|X_{2(16)}|X_{3(16)} \leftarrow X_{(64)}$

$Adım2: X_0 \leftarrow X_0 \oplus wk_0\,,\quad X_2 \leftarrow X_2 \oplus wk_1$

$Adım3: for\ i = 0\ \ to\ r-2$

$Adım3.1: X_1 \leftarrow X_1 \oplus F(X_0) \oplus rk_{2i},$

$X_3 \leftarrow X_3 \oplus F(X_2) \oplus rk_{2i+1},$

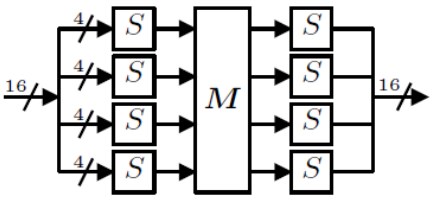
$Adım3.2: X_0|X_1|X_2|X_3 \leftarrow RP(X_0|X_1|X_2|X_3)$

$Adım4: X_1 \leftarrow X_1 \oplus F(X_0) \oplus rk_{2r-2},$

$X_3 \leftarrow X_3 \oplus F(X_2) \oplus rk_{2r+1}$

$Adım5: X_0 \leftarrow X_0 \oplus wk_2\,,\quad X_2 \leftarrow X_2 \oplus wk_3$

$Adım6: Y_{(64)} \leftarrow X_0|X_1|X_2|X_3$



Şekil 6.10. F fonksiyonu

PICCOLO algoritmasındaki döngülerde kullanılan $F: \{0,1\}^{16} \rightarrow \{0,1\}^{16}$ şeklinde tanımlanmış F fonksiyonu Şekil 6.10'da verilmiştir. Buna göre 16 bit giriş değeri öncelikli olarak 4 bitlik S-Kutuları ile ifade 6.8'deki şekilde işlenir.

$$(x_0, x_1, x_2, x_3) \leftarrow \big(S(x_0), S(x_1), S(x_2), S(x_3)\big) \tag{6.8}$$

Bu aşamada kullanılan S-Kutusu Çizelge 6.6'da verilmiştir.

Çizelge 6.6. PICCOLO S-Kutusu

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	E	4	B	2	3	8	0	9	1	A	7	F	6	C	5	D

Daha sonra M doğrusal dönüşümü ile difüzyon işlemine alınır ve son olarak difüzyon çıkış değerleri tekrar S-Kutularından geçirilir. Değerlerin M matrisi çarpımları sonucu $GF(2^4)$ te $x^4 + x + 1$ indirgenemez polinomu ile oluşturulan cisim içerisinde tanımlıdır. Bu işlem ifade 6.9'daki gibidir ve kullanılan M matrisi ifade 6.10'da verilmiştir.

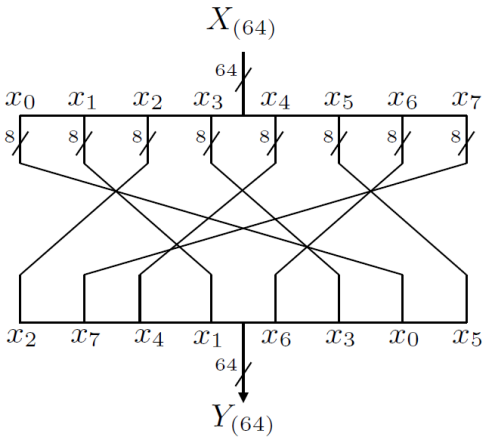
$$^{t}(x_0, x_1, x_2, x_3) \leftarrow M.\ ^{t}(x_0, x_1, x_2, x_3) \tag{6.9}$$

$$M = \begin{bmatrix} 2_h & 3_h & 1_h & 1_h \\ 1_h & 2_h & 3_h & 1_h \\ 1_h & 1_h & 2_h & 3_h \\ 3_h & 1_h & 1_h & 2_h \end{bmatrix} \tag{6.10}$$

PICCOLO algoritmasında kullanılan permütasyon yapısı $RP: \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ şeklinde tanımlanmış olup 64 bitlik $X_{(64)}$ giriş değerini $x_{0(8)}|x_{1(8)}|\ldots|x_{7(8)}$ şeklinde 8 bitlik bölümlere ayırarak ifade 6.11'deki şekilde permütasyon işlemine alır.

$$RP: (x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7) \leftarrow (x_2, x_7, x_4, x_1, x_6, x_3, x_0, x_5) \tag{6.11}$$

Daha sonrasında bitler birleştirilerek diğer döngüye aktarılır. Son döngüde permütasyon işlemi gerçekleştirilmez. RP permütasyon işlemi Şekil 6.11'de verilmiştir.



Şekil 6.11. RP permütasyon işlemi

PICCOLO algoritmasının şifre çözme işlemi olan G_r^{-1}, şifreleme işlemi olan G_r işlemine çok benzemektedir. G_r^{-1} içerisinde sadece beyazlatma ve döngü anahtarlarının yerleri değişmektedir. G_r^{-1} ifade 6.12'deki gibi tanımlıdır ve algoritması Algoritma 6.11'de verilmiştir.

$$G_r^{-1}: \begin{cases} \{0,1\}^{64} \times \{\{0,1\}^{16}\}^4 \times \{\{0,1\}^{16}\}^{2r} \to \{0,1\}^{64} \\ \left(Y_{(64)}, wk_{0(16)}, \dots, wk_{3(16)}, rk_{0(16)}, \dots, rk_{2r-1(16)}\right) \mapsto X_{(64)} \end{cases} \tag{6.12}$$

Algoritma 6.11. PICCOLO G_r^{-1} şifre çözme algoritması

$Adım1$: $wk'_0 \leftarrow wk_2$, $wk'_1 \leftarrow wk_3$, $wk'_2 \leftarrow wk_0$, $wk'_3 \leftarrow wk_1$

$Adım2$: $for\ i = 0\ \ to\ r-1$

$$Adım2.1: rk'_{2i}|rk'_{2i+1} \leftarrow \begin{cases} rk_{2r-2i-2}|rk_{2r-2i-1} & (Eğer\ i\ mod\ 2 = 0) \\ rk_{2r-2i-1}|rk_{2r-2i-2} & (Eğer\ i\ mod\ 2 = 1) \end{cases}$$

$Adım3$: $X_{(64)} \leftarrow G_r(Y, wk'_0, \dots, wk'_3, rk'_0, \dots rk'_{2r-1})$

PICCOLO algoritmasının anahtar planlama evresi 80 ve 128 bitlik anahtarlar üzerinden $wk_i \in \{0,1\}^{16}$ $(0 \le i < 4)$ ifadesini sağlayan 4 adet beyazlatma anahtarını ve $rk_i \in \{0,1\}^{16}$ $(0 \le i < 2r)$ ifadesini sağlayan $2r$ adet döngü anahtarlarını üretmektedir. Anahtar planlama evreleri 80 bit için KS_r^{80} ve 128 bit için ise KS_r^{128} olarak

simgelenmektedir. Anahtar planlama için ilk olarak 16 bitlik sabit con_i^{80} ifade 6.13 ve con_i^{128} ifade 6.14'teki gibi üretilmektedir. Burada c_i, i değerinin 5 bitlik ikili ifadesini belirtmektedir. Örneğin $c_{11} = \{01011\}_2$ olarak düşünülebilir.

$$con_{2i}^{80}|con_{2i+1}^{80} \leftarrow (c_{i+1}|c_0|c_{i+1}|\{00\}_2|c_{i+1}|c_0|c_{i+1}) \oplus \{0F1E2D3C\}_{16} \quad (6.13)$$

$$con_{2i}^{128}|con_{2i+1}^{128} \leftarrow (c_{i+1}|c_0|c_{i+1}|\{00\}_2|c_{i+1}|c_0|c_{i+1}) \oplus \{6547A98B\}_{16} \quad (6.14)$$

80 Bitlik anahtar planlama aşaması KS_r^{80} için 80 bitlik gizli anahtar $K_{(80)}$, $k_{i(16)}$ $(0 \leq i < 5)$ şeklinde 16 bitlik 5 parçaya ayrılarak $wk_i (0 \leq i < 4)$ beyazlatma anahtarını ve $rk_i (0 \leq i < 2r)$ döngü anahtarını Algoritma 6.12'ye göre oluşturur. Burada k_i^L ve k_i^R ifadeleri k_i ifadesinin sağ ve sol 8 bitini belirtmektedir.

Algoritma 6.12. PICCOLO KS_r^{80} anahtar planlama algoritması

$Adım1: wk_0 \leftarrow k_0^L|k_1^R,\ wk_1 \leftarrow k_1^L|k_0^R,\ wk_2 \leftarrow k_4^L|k_3^R, wk_3 \leftarrow k_3^L|k_4^R$

$Adım2: for\ i = 0\ to\ r-1$

$$Adım2.1: (rk_{2i}, rk_{2i+1}) \leftarrow (con_{2i}^{80}, con_{2i+1}^{80}) \oplus \begin{cases} (k_2, k_3) Eğer\ i\ mod\ 5 = 0\ veya\ 2 \\ (k_0, k_1) Eğer\ i\ mod 5 = 1\ veya\ 4 \\ (k_4, k_4) Eğer\ i\ mod 5 = 3 \end{cases}$$

128 Bitlik anahtar planlama aşaması olan KS_r^{128} algoritmasında ise 128 bitlik gizli anahtar $K_{(128)}$, $k_{i(16)}$ $(0 \leq i < 8)$ şeklinde 16 bitlik 8 parçaya ayrılarak $wk_i (0 \leq i < 4)$ beyazlatma anahtarını ve $rk_i (0 \leq i < 2r)$ döngü anahtarını Algoritma 6.13'e göre oluşturur.

Algoritma 6.13. PICCOLO KS_r^{128} anahtar planlama algoritması

$Adım1: wk_0 \leftarrow k_0^L|k_1^R,\ wk_1 \leftarrow k_1^L|k_0^R, wk_2 \leftarrow k_4^L|k_7^R, wk_3 \leftarrow k_7^L|k_4^R$

$Adım2: for\ i = 0\ to\ 2r-1$

$Adım2.1: Eğer\ (i+2)\ mod = 0$

$(k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7) \leftarrow (k_2, k_1, k_6, k_7, k_0, k_3, k_4, k_5)$

$Adım2.2: rk_i \leftarrow k_{(i+2)mod\ 8} \oplus con_i^{128}$

6.4.4. PRINCE Blok Şifreleme Algoritması

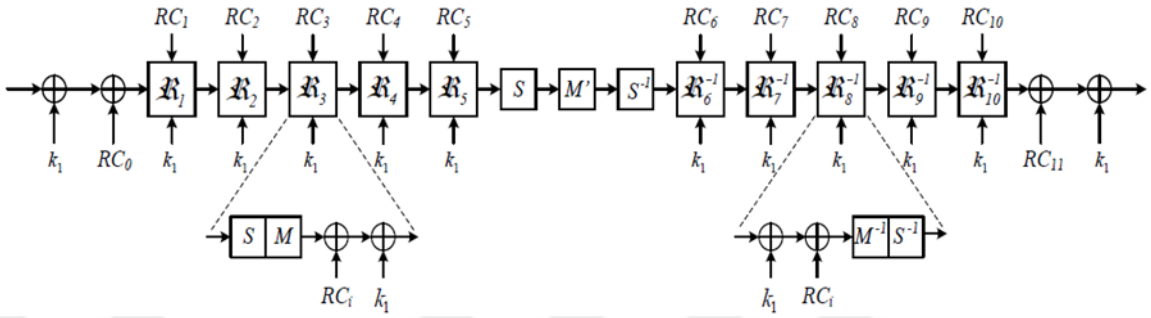
PRINCE, 64 bit veri bloklarını 128 bit anahtar ile şifreleyen bir algoritmadır. Donanım üzerine optimize edilmiş olan PRINCE Anne Canteaut, Christian Rechberger, Christof Paar, Elif Bilge Kavun, Gregor Leander, Julia Borghoff, Lars R. Knudsen, Miroslav Knezevic, Peter Rombouts, Søren S. Thomsen, Tim Guneysu, Tolga Yalçın ve Ventzislav Nikov tarafından tasarlanmış ve Asiacrypt 2012'de sunulmuştur. FX adı verilen bir yapıya sahip olan algoritma şifre çözmeyi farklı bir anahtar kullanarak şifreleme işlemi üzerinden gerçekleştirir.

Tasarımcıların önerisine göre 12 adımda şifreleme yapan algoritma 128 bitlik k anahtarını 64 bitlik iki parçaya $k = k_0|k_1$ şeklinde böler. Ardından 192 bite genişletme işlemi $k_0' = (k_0 \ggg 1) \oplus (k_0 \gg 63)$ olacak şekilde $k_0|k_0'|k_1$ dönüşümü gerçekleştirilir. Burada k_0 ve k_0' beyazlatma anahtarları olarak kullanılırken k_1 ise 12 adım için döngü anahtarı olarak kullanılır. Dolayısı ile algoritmanın detaylı bir anahtar üretim aşaması yoktur ve her döngüde aynı anahtar kullanılır. Anahtar beyazlatma işlemi Şekil 6.12'deki gibi m açık metni ile k_0 değerinin XOR işlemine alınması ve 12 adım sonrasındaki çıkış değerinin k_0' ile XOR'lanması şeklinde yapılır.



Şekil 6.12. PRINCE anahtar beyazlatma işlemi

PRINCE şifreleme algoritmasının diyagramı Şekil 6.13'te verilmiştir. Her adım döngü anahtarı k_1 ile XOR'lama, S-Kutusu ile haritalama, doğrusal dönüşüm işlemi ve döngü sabiti RC_i ile XOR'lama işlemini barındırmaktadır. İlk 5 döngü çalıştıktan sonra ara işlem olarak S-Kutusu, doğrusal dönüşüm işlemi ve S-Kutusunun tersi işlemi yapılır.



Şekil 6.13. PRINCE Şifreleme algoritması diyagramı

PRINCE algoritmasında kullanılan 4 bitlik S-Kutusunu ve tersini Çizelge 6.7'de, döngü adımların da kullanılan döngü sabitleri ise Çizelge 6.8'de verilmiştir.

Çizelge 6.7. PRINCE S-Kutusu ve tersi.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S(x)$	B	F	3	2	A	C	9	1	6	7	8	0	E	5	D	4
$S^{-1}(x)$	B	7	3	2	F	D	8	9	A	6	4	0	5	E	C	1

Çizelge 6.8. PRINCE döngü sabitleri

RC_0	0000000000000000
RC_1	13198A2E03707344
RC_2	A4093822299F31D0
RC_3	082EFA98EC4E6C89
RC_4	452821E638D01377
RC_5	BE5466CF34E90C6C
RC_6	7EF84F78FD955CB1
RC_7	85840851F1AC43AA
RC_8	C882D32F25323C54
RC_9	64A51195E0E3610D
RC_{10}	D3B5A399CA0C2399
RC_{11}	C0AC29B7C97C50DD

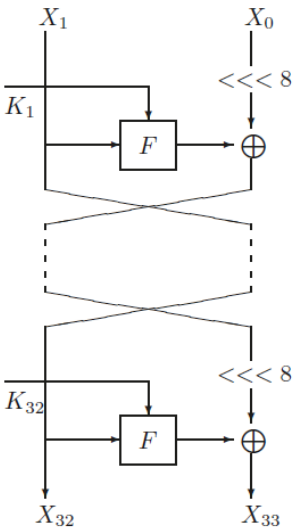
Döngü sabit değerlerinde dikkat edilirse $0 \leq i \leq 11$ olmak üzere $RC_i \oplus RC_{11-i}$ işlemi sabit değer olan $\alpha = \text{c0ac29b7c97c50dd}$ değerini vermektedir. Buradaki α sabit değeri $\pi = 3.141\ldots$ değerinden türemiştir. Bu özellik ve ilk 5 döngü son 5' döngünün tersi olarak algoritmanın tasarımı sayesinde k anahtarı ile şifrelenen metnin açık metne dönüştürülmesi için $k \oplus \alpha$ ile şifreleme algoritmasının çalıştırılması yeterlidir.

6.4.5. LBLOCK Blok Şifreleme Algoritması

LBLOCK, Wenling Wu ve Lei Zhang tarafından önerilmiş 64 bit veri bloklarını 80 bit anahtar ile şifreleyen bir algoritmadır. Sınırlı kaynaklar ile çalışan donanımlar üzerinde verimli ve güvenli olacak şekilde tasarlanmıştır. Feistel mimarisini kullanan

algoritma 32 döngüde şifreleme işlemini tamamlamaktadır.

Şifreleme işleminde öncelikle 64 bitlik M açık metni $M = X_1|X_0$ şeklinde 32 bitlik iki parçaya ayrılır. X_1 değeri işleme alınmadan bir sonraki döngüye aktarılır. Daha sonra X_1 ve K_1 döngü anahtarı F döngü fonksiyonuna alınarak 8 bit sola döngüsel kaydırılmış $X_0 \lll 8$ sonucu ile XOR işlemine alınır. Çıkan sonuç bir sonraki döngünün X_2 değeri olarak kaydedilir. 32 döngü sonunda M açık metni $C = X_{32}|X_{33}$ şeklinde şifrelenmiş olur. LBLOCK döngü yapısı ise Şekil 6.14'de ve şifreleme işlemleri Algoritma 6.14'de ve verilmiştir.



Şekil 6.14. LBLOCK şifreleme döngü yapısı

Algoritma 6.14. LBLOCK şifreleme algoritması

$Adım1$: $X_1|X_0 \leftarrow M$

$Adım2$: $for\ i = 2\ \ to\ 33$

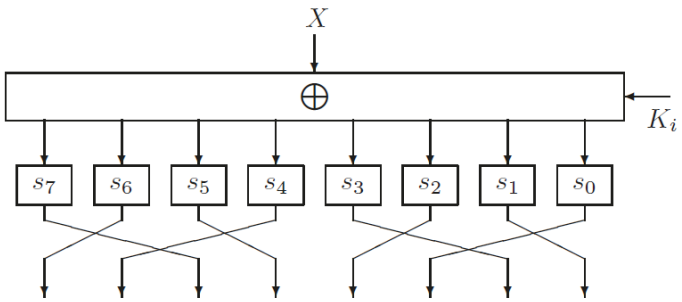
$Adım2.1$: $X_i \leftarrow F(X_{i-1}, K_{i-1}) \oplus (X_{i-2} \lll 8)$

$Adım3$: $C \leftarrow X_{32}|X_{33}$

LBLOCK şifreleme algoritmasında geçen F döngü fonksiyonu giriş değerlerinin XOR'laranak sonucun S-Kutularından geçirilmesi ve devamında difüzyon işleminin uygulanması adımlarını barındırmaktadır. F döngü fonksiyonu ifade 6.15'deki gibidir ve

diyagramı ise Şekil 6.15'de verilmiştir.

$$F: \begin{matrix} \{0,1\}^{32} \times \{0,1\}^{32} \longrightarrow \{0,1\}^{32} \\ (X, K_i) \longrightarrow U = P(S(X \oplus K_i)) \end{matrix} \tag{6.15}$$



Şekil 6.15. LBLOCK F fonksiyonu

F döngü fonksiyonun içerisinde birbirinden farklı 8 adet 4 bitlik S-Kutuları kullanılmaktadır. Buna göre K_i döngü anahtarı ile XOR'lana 32 bitlik X değeri 4 bitlik 8 parçaya $X = X_7|X_6|X_5|X_4|X_3|X_2|X_1|X_0$ şeklinde ayrılır ve $s_7, s_6, s_5, s_4, s_3, s_2, s_1, s_0$ S-Kutularından geçirilerek difüzyon tabakasına aktarılır. LBOCK içerisinde kullanılan S-Kutuları Çizelge 6.9'da verilmiştir. Bu S-Kutularından s_8 ve s_9 anahtar üretim aşamasında kullanılmaktadır.

Çizelge 6.9. LBLOCK S-Kutuları

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$s_0(x)$	E	9	F	0	D	4	A	B	1	2	8	3	7	6	C	5
$s_1(x)$	4	B	E	9	F	D	0	A	7	C	5	6	2	8	1	3
$s_2(x)$	1	E	7	C	F	D	0	6	B	5	9	3	2	4	8	A
$s_3(x)$	7	6	8	B	0	F	3	E	9	A	C	D	5	2	4	1
$s_4(x)$	E	5	F	0	7	2	C	D	1	8	4	9	B	A	6	3
$s_5(x)$	2	D	B	C	F	E	0	9	7	A	6	3	1	8	4	5
$s_6(x)$	B	9	4	E	0	F	A	D	6	C	5	7	3	8	1	2
$s_7(x)$	D	A	F	0	E	4	9	B	2	1	8	3	7	5	C	6
$s_8(x)$	8	7	E	5	F	D	0	6	B	C	9	A	2	4	1	3
$s_9(x)$	B	5	F	0	7	2	9	D	4	8	1	C	E	A	3	6

F döngü fonksiyonun içerisinde S-Kutularından geçen X değeri difüzyon işlemine alınır. LBLOCK algoritması için difüzyon işlemi 4 bitlik 8 değerin permütasyonu olarak tasarlanmıştır. Buna göre difüzyon aşamasına gelen $X_7|X_6|X_5|X_4|X_3|X_2|X_1|X_0$ girişleri $X_6|X_4|X_7|X_5|X_2|X_0|X_3|X_1$ şeklinde sıralanır.

LBLOCK şifre çözme algoritması, şifreleme işleminin çok benzeridir. 64 bitlik C şifreli metni $C = X_{32}|X_{33}$ şeklinde 32 bitlik iki parçaya ayrılır. Şifreleme algoritması benzer şekilde çalıştırılır. Buradaki tek fark F döngü fonksiyonu çıkışı 8 bit sağa döngüsel kaydırılmış $X_i \ggg 8$ sonucu ile XOR işlemine alınmasıdır. LBLOCK şifre çözme algoritması Algoritma 6.15'de verilmiştir.

Algoritma 6.15. LBLOCK şifre çözme algoritması

$Adım1: X_{32}|X_{33} \leftarrow C$

$Adım2: for\ i = 31\ \ to\ 0$

$Adım2.1: X_i \leftarrow F(X_{i+1}, K_{i+1}) \oplus (X_{i+2} \ggg 8)$

$Adım3: M \leftarrow X_1|X_0$

LBLOCK anahtar üretim aşamasında K gizli anahtarı $K = k_{79}, k_{78}, \dots, k_0$ şeklinde bir kaydedicide saklanır. Kaydedicide bulunan 80 bitlik anahtarın en soldaki 32 biti döngü anahtarı K_i olarak kullanılır. Dolayısıyla ilk döngü anahtarı K_1, gizli anahtar K değerinin ilk 32 bitlik bölümüdür. Diğer anahtarlar ise Algoritma 6.16'e göre üretilir.

Algoritma 6.16. LBLOCK anahtar güncelleme algoritması

$Adım1: for\ i = 1\ \ to\ 31$

$Adım1.1: K \lll 29$

$Adım1.2: [k_{79}k_{78}k_{77}k_{76}] \leftarrow s_9([k_{79}k_{78}k_{77}k_{76}])$

$Adım1.3: [k_{75}k_{74}k_{73}k_{72}] \leftarrow s_8([k_{75}k_{74}k_{73}k_{72}])$

$Adım1.4: [k_{50}k_{49}k_{48}k_{47}k_{46}] \oplus i_2$

$Adım1.5: K_{i+1} \leftarrow [k_{79}k_{78} \dots k_{49}k_{48}]$

BÖLÜM 7

7. SEÇİLEN HAFİF SIKLET ŞİFRELEME ALGORİTMALARININ PERFORMANS ANALİZİ

7.1. Giriş

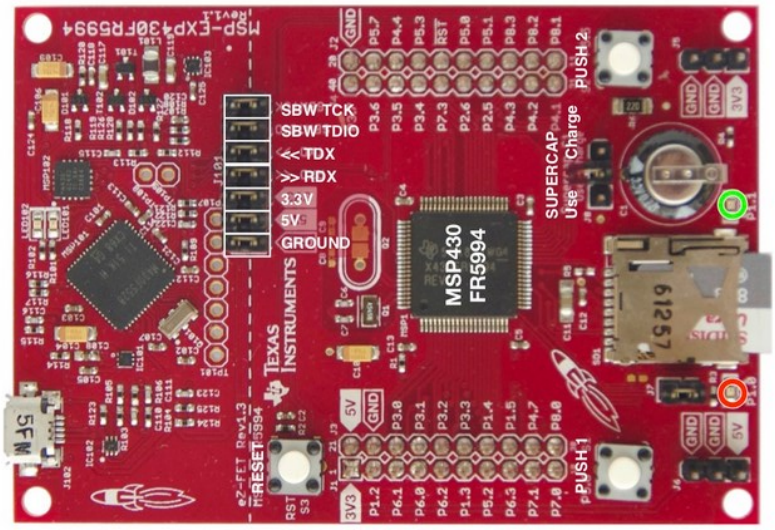
Tezin 6. bölümünde detayları verilen hafif sıklet kriptografik algoritmalar TI (Texas Instruments) firmasının ürünü olan MSP-EXP430FR5994 üzerinde belirlenen senaryoya göre gerçeklenmiştir. Algoritmalar belirlenen senaryoya göre çalıştırılıp enerji ve güç tüketimleri analiz edilmiştir. Sonuçlar tablo halinde bölüm sonunda verilmiştir.

7.2. Test Ortamı

Donanım olarak Şekil 7.1'de verilen ve endüstride kullanılan TI firmasının ürünü olan MSP430 ailesinden MSP-EXP430FR5994 LaunchPad Development Kit seçilmiştir. Kullanılan kit MSP430FR5994 mikro denetleyicisi için hazırlanmış bir geliştirme platformudur. 16MHz saat frekansına sahip mikro denetleyici, 256KB büyüklüğünde ultra düşük güç tüketimli FRAM (Ferroelektrik Rasgele Erişim Belleği) kalıcı belleğe sahiptir. MSP430FR5994, sahip olduğu yeni bir teknoloji olan Low-Energy Accelerator (LEA) sayesinde düşük güç tüketimine sahip olup oldukça verimlidir ve analog veriyi gerçek zamanlı olarak işleyebilmektedir (TI, 2019). Dahili eZ-FET hata ayıklama özelliği bulunmaktadır. Bu özellik sayesinde geliştirilen şifreleme kodunun performansı herhangi bir cihaza bağlı kalmaksızın test ve analiz edilebilmektedir (TI, 2020b).

Geliştirme ortamı olarak yine TI firmasının ürettiği cihazlarda kullanılabilen eclipse tabanlı Code Composer Studio (CCS) kullanılmıştır (TI, 2020a). Tüm hafif sıklet şifreleme algoritmaları CCS içerisinde C kodu ile derlenerek MSP430FR5994 cihazına

aktarılmıştır.



Şekil 7.1. MSP-EXP430FR5994 LaunchPad Development Kit

Algoritmaların enerji, güç, akım ölçümleri EnergyTrace™ yazılımı aracılığı ile gerçekleştirilmiştir. EnergyTrace™ teknolojisi, uygulamanın enerji profilini ölçen ve görüntüleyen ayrıca ultra düşük güç tüketimi için optimize etmeye yardımcı olan enerji tabanlı bir kod analiz aracıdır. Code Composer Studio içerisinde entegre olarak çalışmaktadır. EnergyTrace ve EnergyTrace++ olmak üzere iki moda sahiptir. EnergyTrace modu ile temel enerji ölçümleri yapılabilir. Enerji ve gücü ölçmek için mikro denetleyicideki besleme gerilimi sürekli olarak örneklenir. Bu mod, hata ayıklayıcıya erişmeden uygulamanın enerji tüketimini doğrulamak için kullanılabilir. EnergyTrace++ modu ise temel enerji ölçümlerinin yanı sıra RAM kullanımı, enerji modları gibi mikro denetleyicinin iç durumu hakkında da bilgi sağlar (BrittanyFinch, 2014).

7.3. Test Senaryosu

MSP430 gibi kenar cihazlarının görevi sensörlerden aldıkları veriyi nesnelerin interneti uygulamasının amacına göre işledikten sonra bulut, sunucu veya broker rolündeki cihazlara iletmeleri veya gelen veriye göre sensör üzerinde çeşitli işlemlerin yapılmasını sağlamak olarak özetlenebilir. Bölüm 5'te anlatıldığı üzere MQTT ve CoAP iletişiminde veri güvenliğini sağlayabilmek için iletimden önce veri şifrelenmelidir.

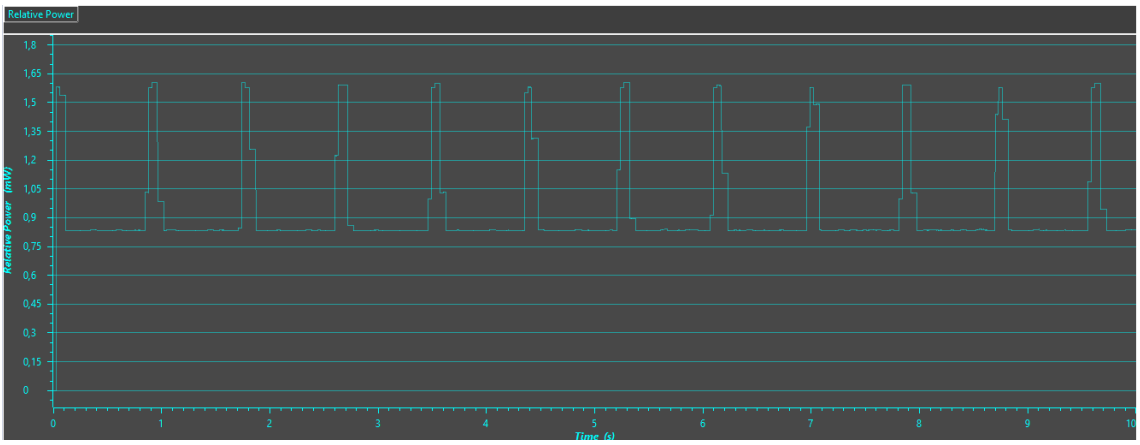
Ayrıca şifreli gelen veri yine bu aşamada şifre çözme işlemi ile açık metne dönüştürülmelidir.

Belirlenen test senaryosuna göre MSP430FR5994 cihazı çalıştıktan sonra her bir saniyede AM'e (aktif mod) geçerek şifreleme veya şifre çözme işlemi yapacak daha sonra güç tasarrufu modu olan LPM (Düşük Güç Modu) moduna geçecektir. Senaryoya göre cihaz bu şekilde 10 saniye boyunca çalıştırılarak EnergyTrace™ yazılımı ile algoritmaların enerji, güç ve akım verileri ölçülecektir.

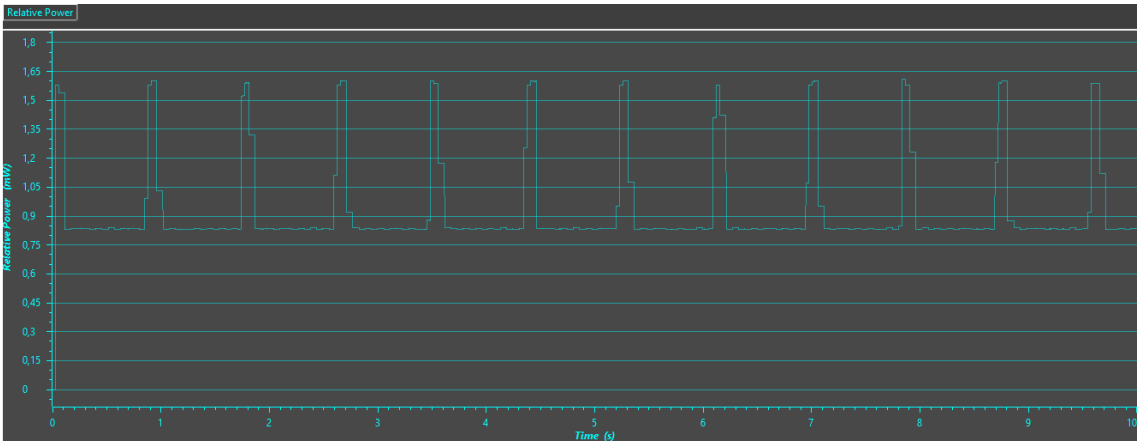
7.4. CLEFIA Blok Şifreleme Algoritması

7.4.1. 128 Bit Şifreleme

CLEFIA algoritması 128 bit blok uzunluğu ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 9,128 mJ enerji harcamıştır. Şekil 7.2'de görüldüğü gibi ortalama olarak 0,9186 mW en fazla ise 1,6036 mW güç tüketmiştir. Ortalama 0,2795 mA en fazla ise 0,4877 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 10 Ay 2 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 9,128 mJ enerji harcamıştır. Şekil 7.3'te görüldüğü gibi ortalama olarak 0,9187 mW en fazla ise 1,6123 mW güç tüketmiştir. Ortalama 0,2795 mA en fazla ise 0,4905 mA akım çekilmiştir. Cihaz bu şekilde 10 Ay 3 gün çalışabilir.

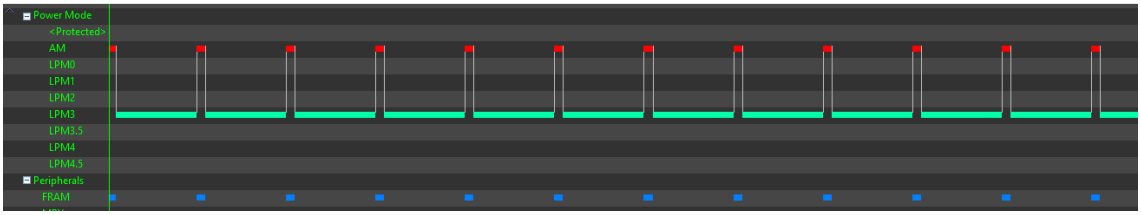


Şekil 7.2. CLEFIA 128 bit şifreleme güç tüketimi

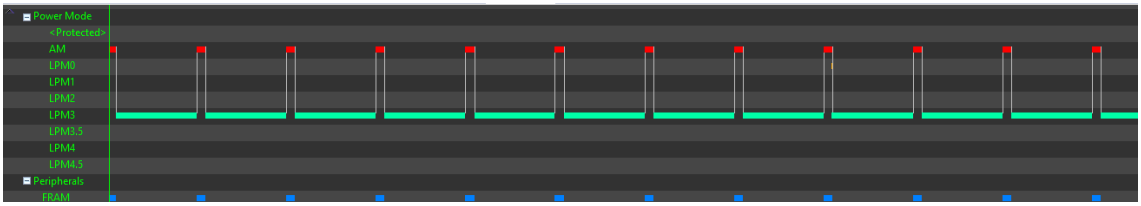


Şekil 7.3. CLEFIA 128 bit şifre çözme güç tüketimi

MSP430FR5994, CLEFIA 128 bit şifreleme işlemini 10 saniyelik çalışma zamanının %10'luk kısmında aktif mod olarak tamamlamış geri kalan %90'lık kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.4'te görülebilir. Cihaz toplam enerji tüketiminin sadece %14.8'ini aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994 yine %10'luk kısmında aktif mod olarak tamamlamış geri kalan %90'lık kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin sadece %14.8'ini aktif modda şifre çözme yaparken harcamıştır.



(Şifreleme)

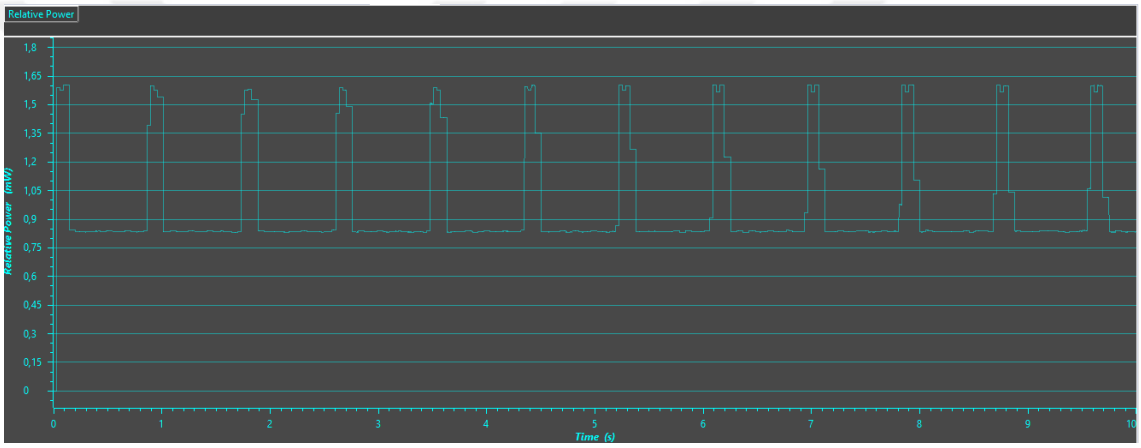


(Şifre Çözme)

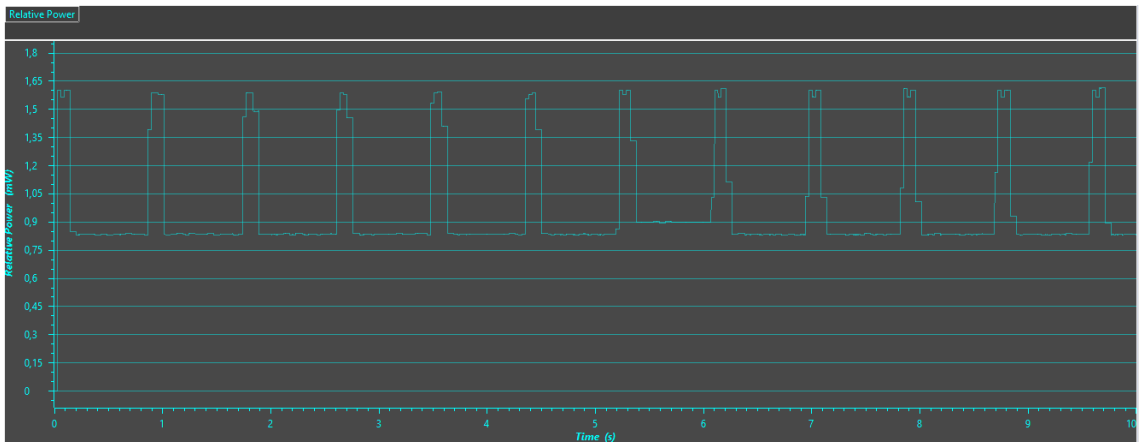
Şekil 7.4. CLEFIA 128 bit şifreleme ve şifre çözme için AM-LPM geçişleri

7.4.2. 192 Bit Şifreleme

CLEFIA algoritması 192 bit blok uzunluğu ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 9,452 mJ enerji harcamıştır. Şekil 7.5'te görüldüğü gibi ortalama olarak 0,9482 mW en fazla ise 1,6040 mW güç tüketmiştir. Ortalama 0,2885 mA en fazla ise 0,4877 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 9 Ay 23 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 9,494 mJ enerji harcamıştır. Şekil 7.6'da görüldüğü gibi ortalama olarak 0,9527 mW en fazla ise 1,6142 mW güç tüketmiştir. Ortalama 0,2899 mA en fazla ise 0,4911 mA akım çekilmiştir. Cihaz bu şekilde 9 Ay 22 gün çalışabilir.

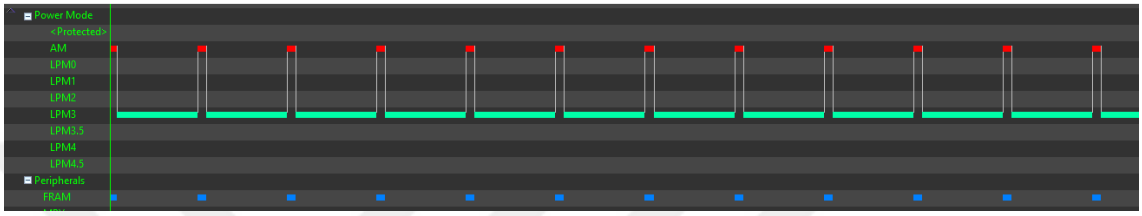


Şekil 7.5. CLEFIA 192 bit şifreleme güç tüketimi

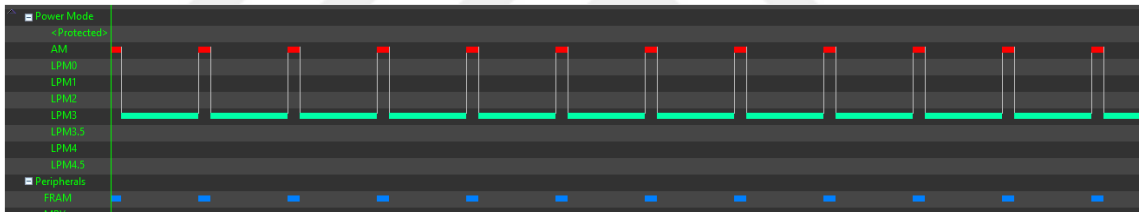


Şekil 7.6. CLEFIA 192 bit şifre çözme güç tüketimi

MSP430FR5994, CLEFIA 192 bit şifreleme işlemini 10 saniyelik çalışma zamanının %14'lük kısmında aktif mod olarak tamamlamış geri kalan %86'lık kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.7'de görülebilir. Cihaz toplam enerji tüketiminin sadece %20.4'ünü aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994, %14'lük kısmında aktif mod olarak tamamlamış geri kalan %86'lık kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin %20.3'ünü aktif modda şifre çözme işlemi yaparken harcamıştır.



(Şifreleme)



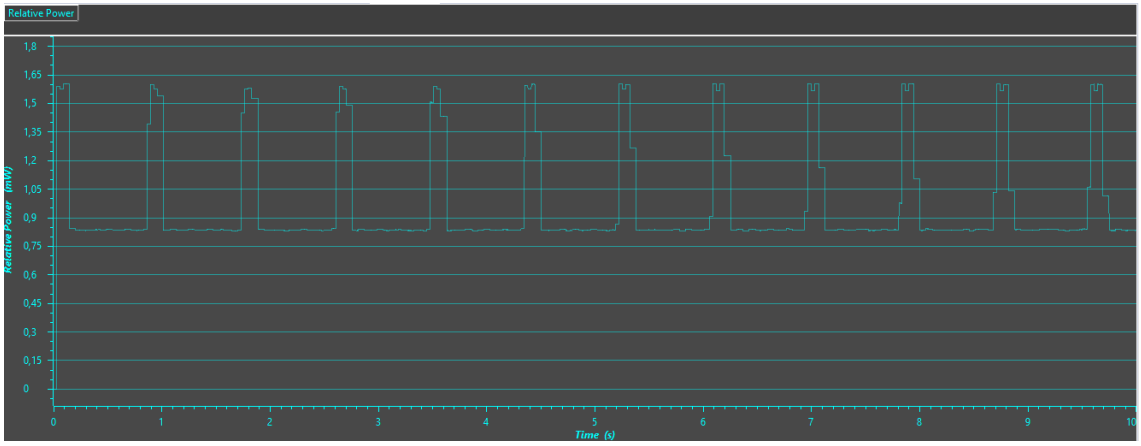
(Şifre çözme)

Şekil 7.7. CLEFIA 192 bit şifreleme ve şifre çözme için AM-LPM geçişleri

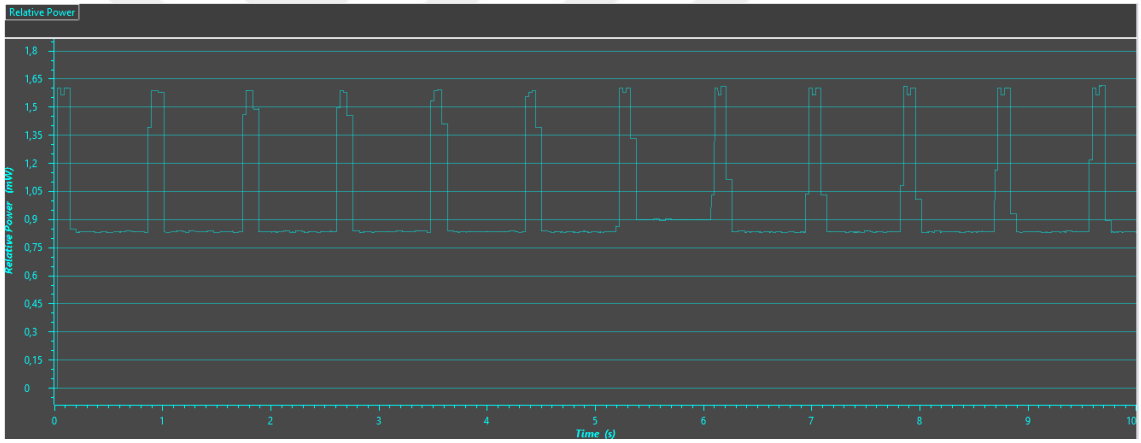
7.4.3. 256 Bit Şifreleme

CLEFIA algoritması 256 bit blok uzunluğu ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 9,452 mJ enerji harcamıştır. Şekil 7.8'de görüldüğü gibi ortalama olarak 0,9482 mW en fazla ise 1,6040 mW güç tüketmiştir. Ortalama 0,2885 mA en fazla ise 0,4877 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 9 Ay 23 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 9,575 mJ enerji harcamıştır. Şekil 7.9'da görüldüğü gibi ortalama olarak 0,9579 mW en fazla ise 1,6141 mW güç tüketmiştir. Ortalama 0,2914 mA en fazla ise 0,4909 mA akım çekilmiştir. Cihaz

bu şekilde 9 Ay 19 gün çalışabilir.

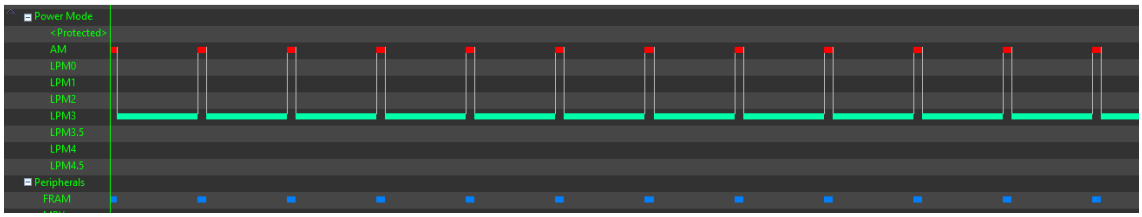


Şekil 7.8. CLEFIA 256 bit şifreleme güç tüketimi

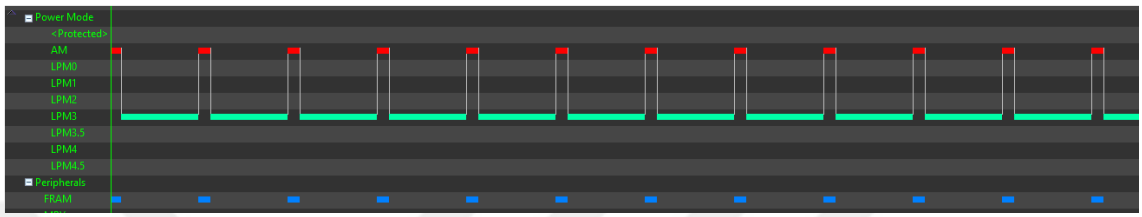


Şekil 7.9. CLEFIA 256 bit şifre çözme güç tüketimi

MSP430FR5994, CLEFIA 256 bit şifreleme işlemini 10 saniyelik çalışma zamanının %15,4'lük kısmında aktif mod olarak tamamlamış geri kalan %84,6'lık kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.10'da görülebilir. Cihaz toplam enerji tüketiminin %22.1'ini aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994, %15,4'lük kısmında aktif mod olarak tamamlamış geri kalan %84,6'lık kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin %22.1'ini aktif modda şifre çözme yaparken harcamıştır.



(Şifreleme)

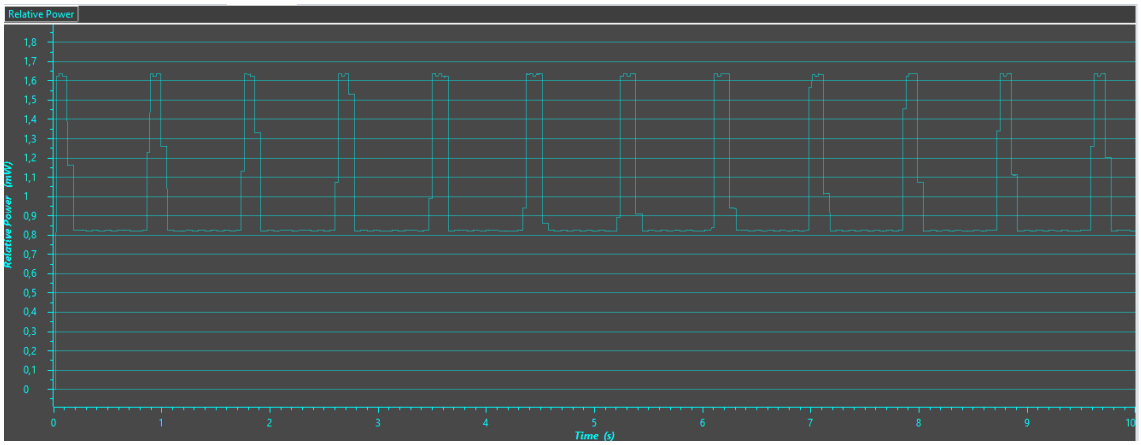


(Şifre çözme)

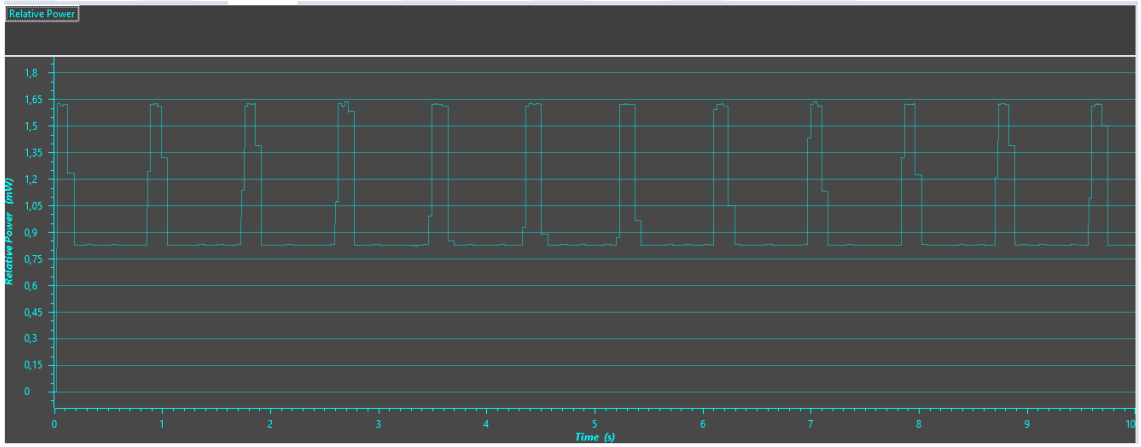
Şekil 7.10. CLEFIA 256 bit şifreleme ve şifre çözme için AM-LPM geçişleri

7.5. PRESENT Blok Şifreleme Algoritması

PRESENT algoritması ele alındığında, şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 9,523 mJ enerji harcamıştır. Şekil 7.11'de görüldüğü gibi ortalama olarak 0,9548 mW en fazla ise 1,6383 mW güç tüketmiştir. Ortalama 0,2905 mA en fazla ise 0,4982 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 9 Ay 21 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 9,573 mJ enerji harcamıştır. Şekil 7.12'de görüldüğü gibi ortalama olarak 0,9591 mW en fazla ise 1,6374 mW güç tüketmiştir. Ortalama 0,2918 mA en fazla ise 0,4980 mA akım çekilmiştir. Cihaz bu şekilde 9 Ay 19 gün çalışabilir.

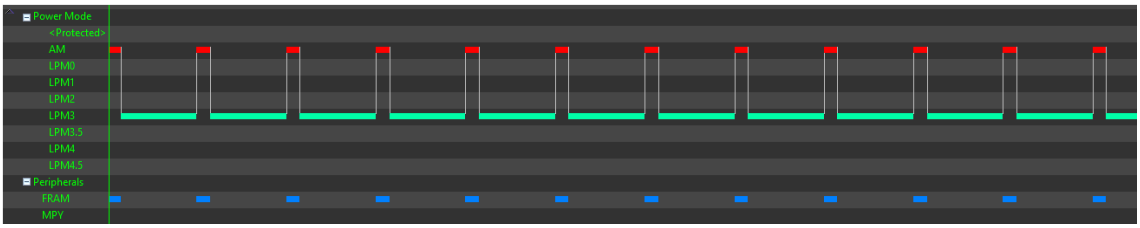


Şekil 7.11. PRESENT şifreleme güç tüketimi

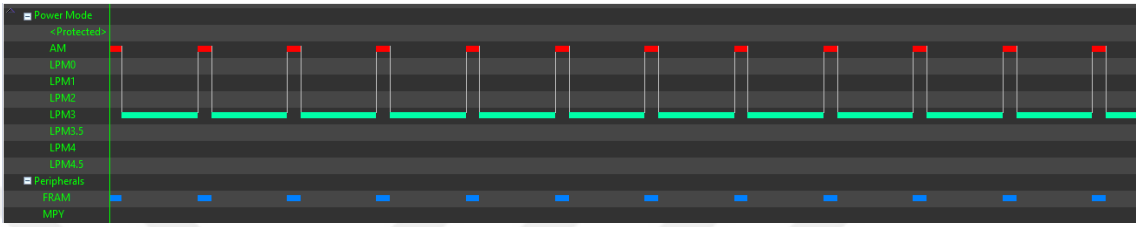


Şekil 7.12. PRESENT şifre çözme güç tüketimi

MSP430FR5994, PRESENT şifreleme işlemini 10 saniyelik çalışma zamanının %15,5'lik kısmında aktif mod olarak tamamlamış geri kalan %84,5'lik kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.13'te görülebilir. Cihaz toplam enerji tüketiminin sadece %23'ünü aktif modda şifreleme yaparken harcamıştır. Benzer sonuçlara şifre çözme işleminde de ulaşılmıştır.



(Şifreleme)



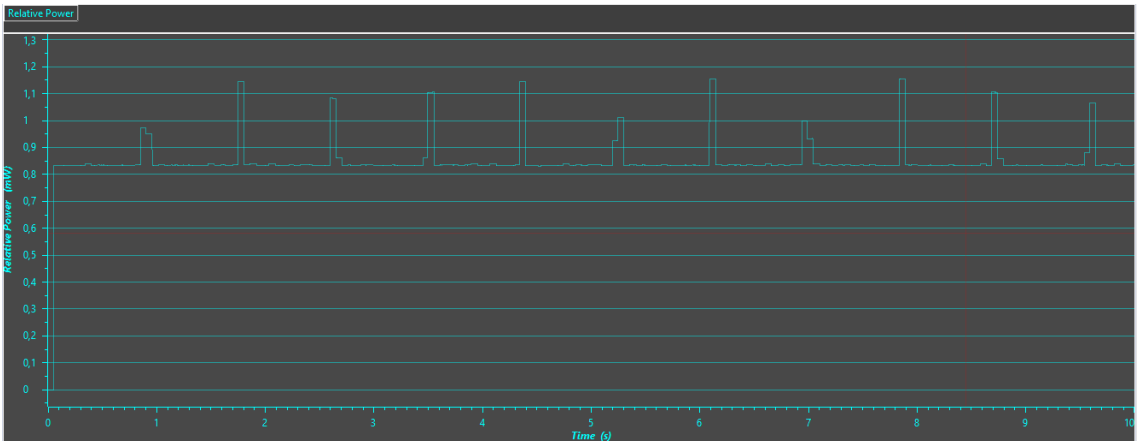
(Şifre çözme)

Şekil 7.13. PRESENT şifreleme ve şifre çözme için AM-LPM geçişleri

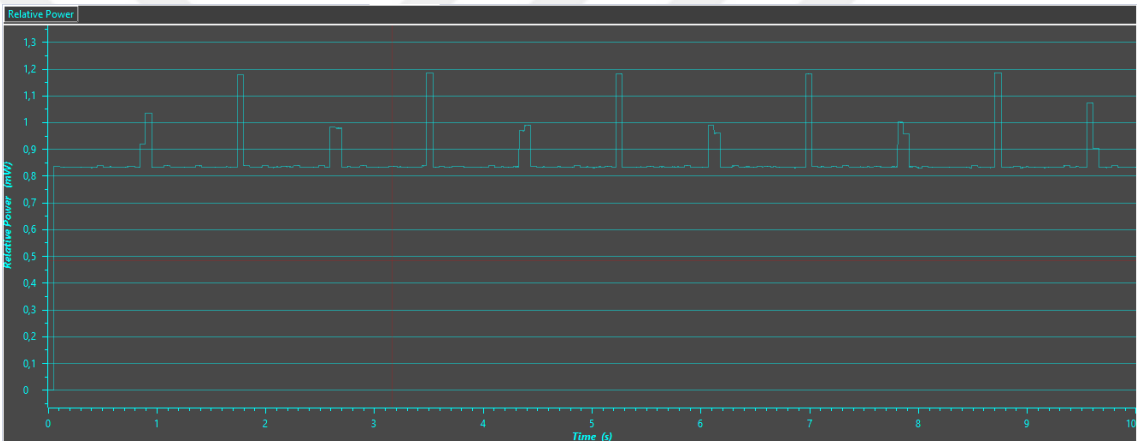
7.6. PICCOLO Blok Şifreleme Algoritması

7.6.1. 80 Bit Şifreleme

PICCOLO algoritması ile 80 bit blok uzunluğu ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 8,487 mJ enerji harcamıştır. Şekil 7.14'te görüldüğü gibi ortalama olarak 0,8439 mW en fazla ise 1,1419 mW güç tüketmiştir. Ortalama 0,2568 mA en fazla ise 0,3472 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 10 Ay 27 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 8,490 mJ enerji harcamıştır. Şekil 7.15'te görüldüğü gibi ortalama olarak 0,8449 mW en fazla ise 1,1768 mW güç tüketmiştir. Ortalama 0,2570 mA en fazla ise 0,3579 mA akım çekilmiştir. Cihaz bu şekilde 10 Ay 26 gün çalışabilir.

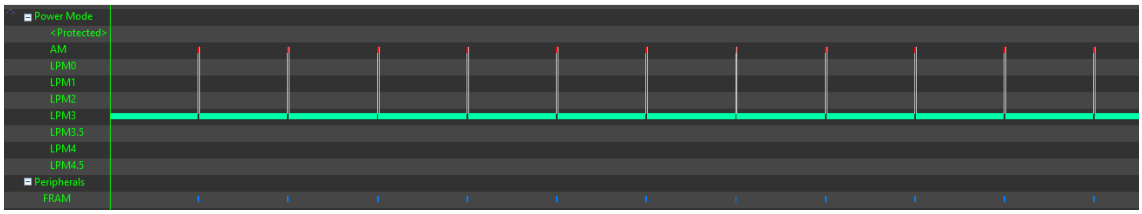


Şekil 7.14. PICCOLO 80 bit şifreleme güç tüketimi

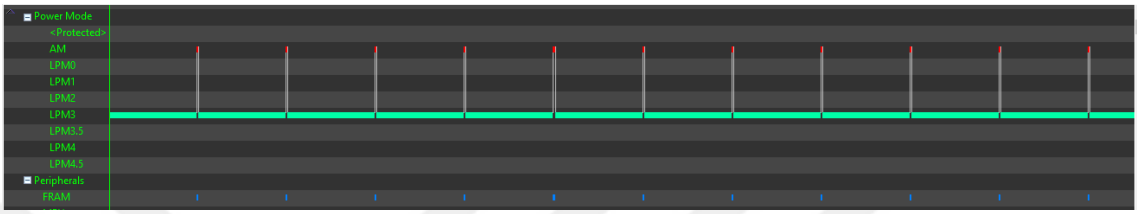


Şekil 7.15. PICCOLO 80 bit şifre çözme güç tüketimi

MSP430FR5994, PICCOLO 80 Bit şifreleme işlemini 10 saniyelik çalışma zamanının %1,8'lik kısmında aktif mod olarak tamamlamış geri kalan %98,2'lik kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.16'te görülebilir. Cihaz toplam enerji tüketiminin sadece %2,9'unu aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994, %1,9'luk kısmında aktif mod olarak tamamlamış geri kalan %98,1'lik kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin %3.1'ini aktif modda şifre çözme yaparken harcamıştır.



(Şifreleme)

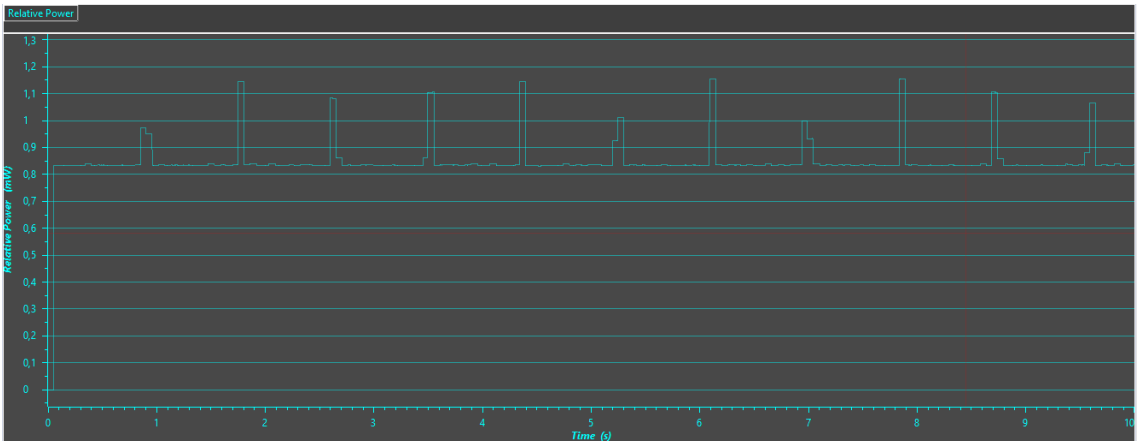


(Şifre çözme)

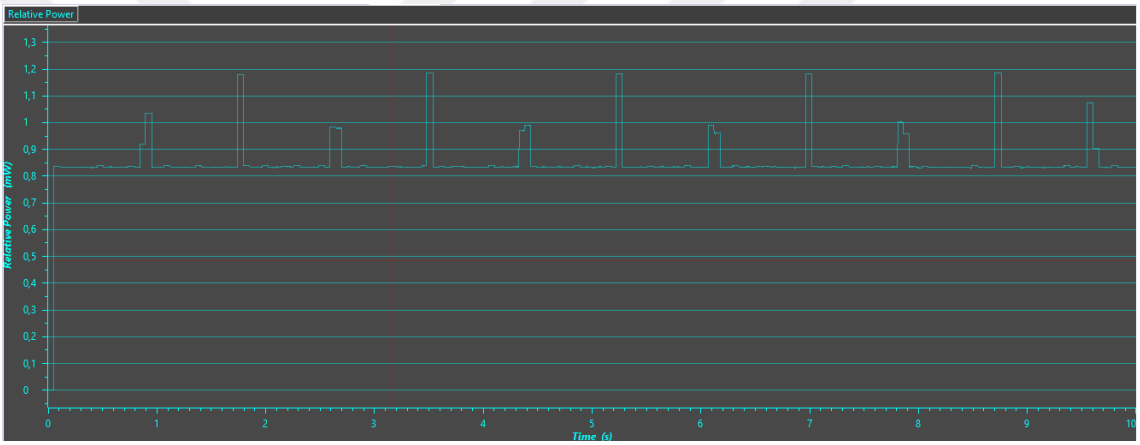
Şekil 7.16. PICCOLO 80 bit şifreleme ve şifre çözme için AM-LPM geçişleri

7.6.2. 128 Bit Şifreleme

PICCOLO algoritması ile 128 bit blok uzunluğu ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 8,533 mJ enerji harcamıştır. Şekil 7.17'da görüldüğü gibi ortalama olarak 0,8498 mW en fazla ise 1,1886 mW güç tüketmiştir. Ortalama 0,2585 mA en fazla ise 0,3614 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 10 Ay 25 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 8,547 mJ enerji harcamıştır. Şekil 7.18'de görüldüğü gibi ortalama olarak 0,8516 mW en fazla ise 1,2431 mW güç tüketmiştir. Ortalama 0,2591 mA en fazla ise 0,3782 mA akım çekilmiştir. Cihaz bu şekilde 10 Ay 24 gün çalışabilir.

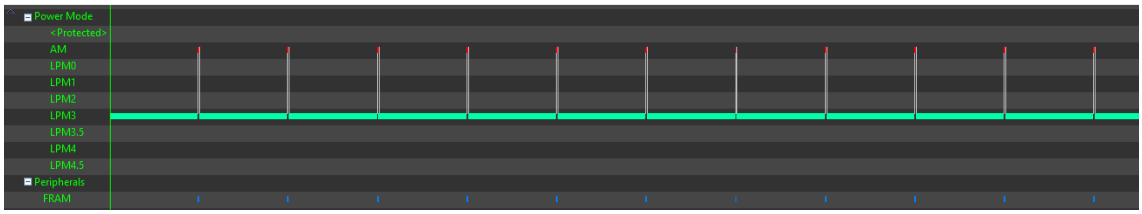


Şekil 7.17. PICCOLO 128 bit şifreleme güç tüketimi

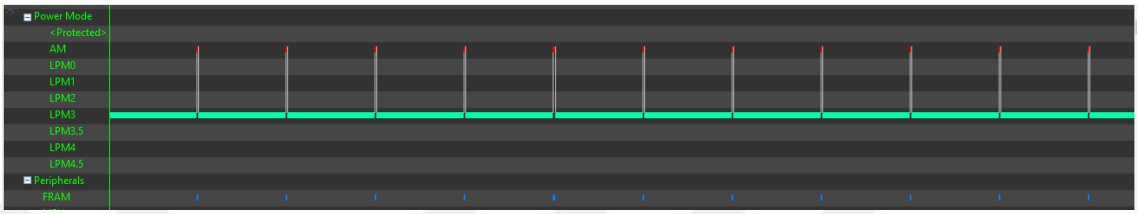


Şekil 7.18. PICCOLO 128 bit şifre çözme güç tüketimi

MSP430FR5994, PICCOLO 128 Bit şifreleme işlemini 10 saniyelik çalışma zamanının %1,8'lik kısmında aktif mod olarak tamamlamış geri kalan %98,2'lik kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.19'de görülebilir. Cihaz toplam enerji tüketiminin sadece %2,9'unu aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994, %1,9'luk kısmında aktif mod olarak tamamlamış geri kalan %98,1'lik kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin %3.1'ini aktif modda şifre çözme yaparken harcamıştır.



(Şifreleme)

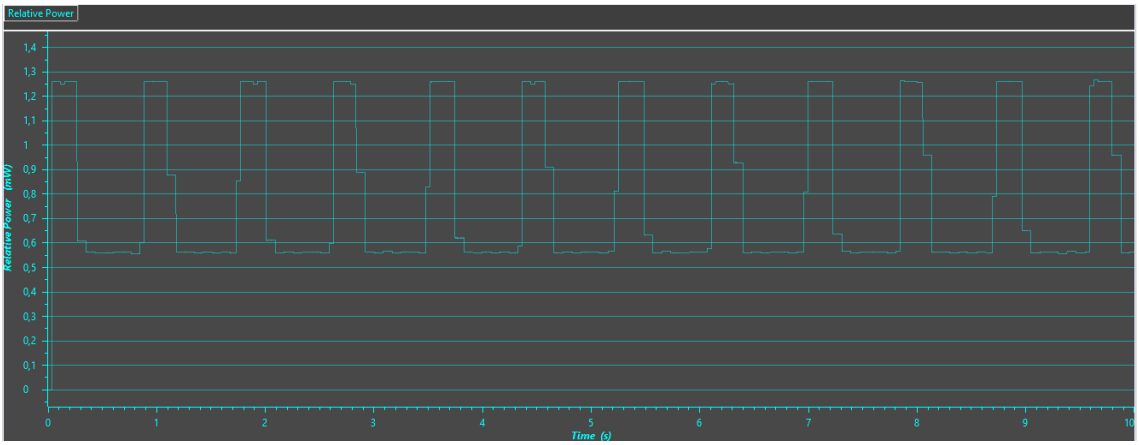


(Şifre çözme)

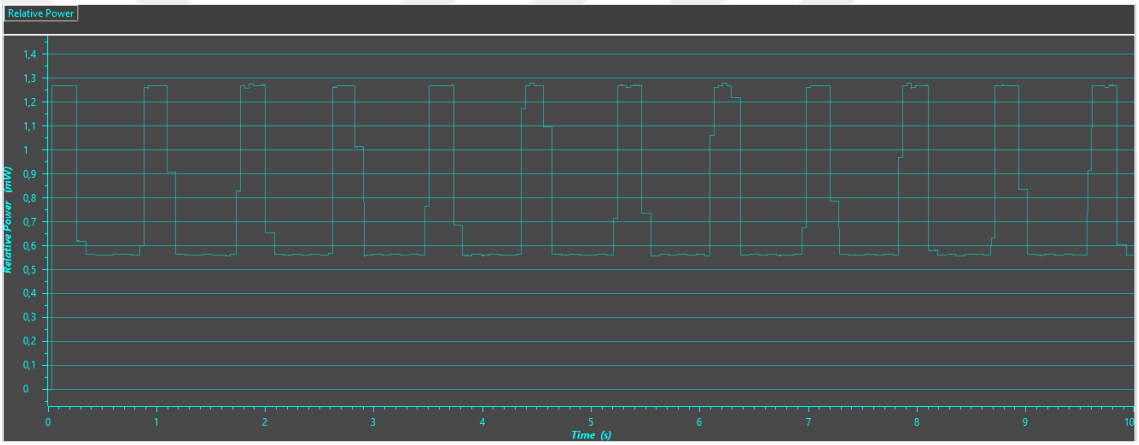
Şekil 7.19. PICCOLO 128 bit şifreleme ve şifre çözme için AM-LPM geçişleri

7.7. PRINCE Blok Şifreleme Algoritması

PRINCE algoritması ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 7,513 mJ enerji harcamıştır. Şekil 7.20'da görüldüğü gibi ortalama olarak 0,7578 mW en fazla ise 1,1694 mW güç tüketmiştir. Ortalama 0,2306 mA en fazla ise 0,3863 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 1 yıl 9 gün çalışabilir. Şifre çözme işlemi aynı senaryoya göre 7,549 mJ enerji harcamıştır. Şekil 7.21'de görüldüğü gibi ortalama olarak 0,7621 mW en fazla ise 1,2797 mW güç tüketmiştir. Ortalama 0,2319 mA en fazla ise 0,3892 mA akım çekilmiştir. Cihaz bu şekilde 1 yıl 7 gün çalışabilir.

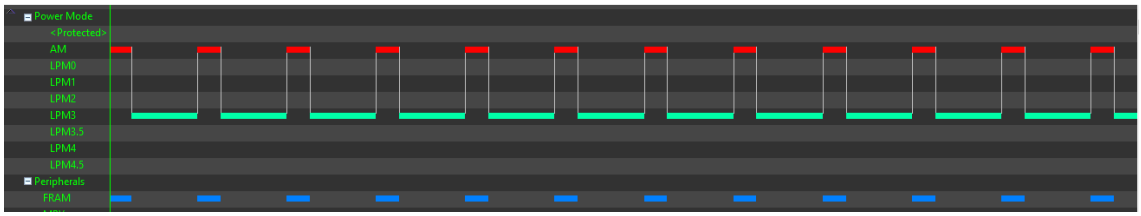


Şekil 7.20. PRINCE şifreleme güç tüketimi

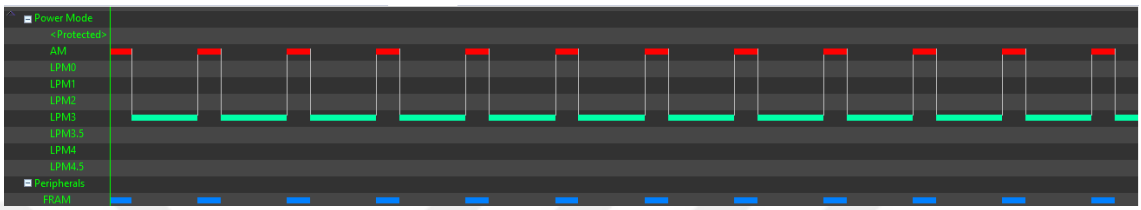


Şekil 7.21. PRINCE şifre çözme güç tüketimi

MSP430FR5994, PRINCE şifreleme işlemini 10 saniyelik çalışma zamanının %26,8'lik kısmında aktif mod olarak tamamlamış geri kalan %73,2'lik kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.22'de görülebilir. Cihaz toplam enerji tüketiminin sadece %38,8'ini aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994, %26,9'luk kısmında aktif mod olarak tamamlamış geri kalan %71,3'lik kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin %39'unu aktif modda şifre çözme yaparken harcamıştır.



(Şifreleme)

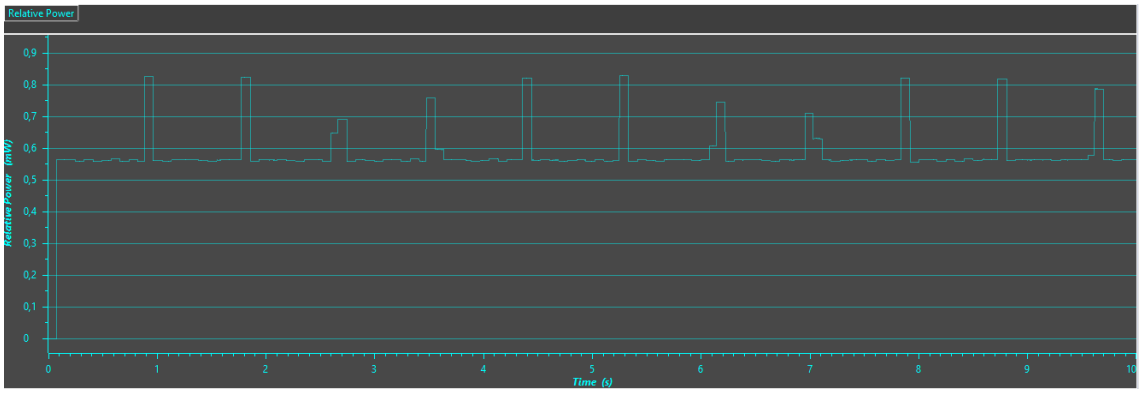


(Şifre çözme)

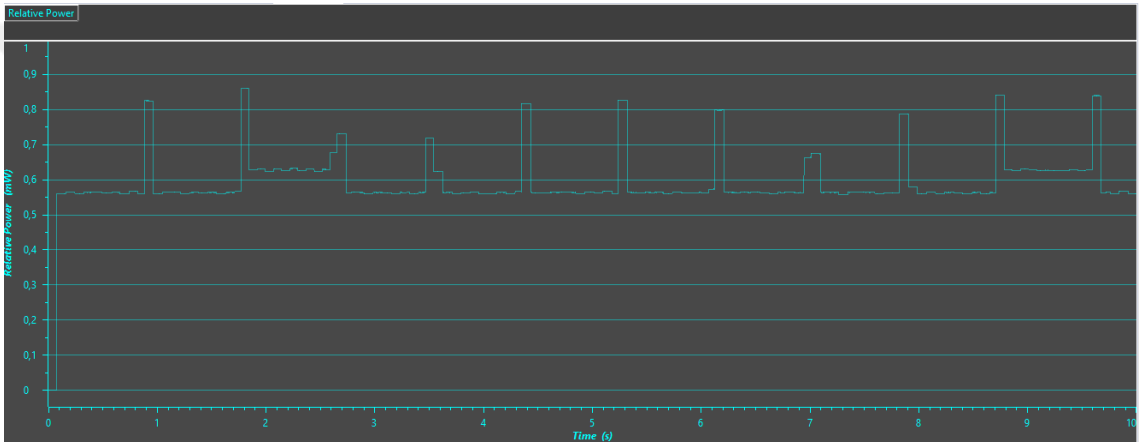
Şekil 7.22. PRINCE şifreleme ve şifre çözme için AM-LPM geçişleri

7.8. LBLOCK Blok Şifreleme Algoritması

LBLOCK algoritması ile şifreleme işlemi MSP430FR5994 üzerinde 10 saniye boyunca çalışarak 5,812 mJ enerji harcamıştır. Şekil 7.23'de görüldüğü gibi ortalama olarak 0,5799 mW en fazla ise 0,8295 mW güç tüketmiştir. Ortalama 0,1764 mA en fazla ise 0,2522 mA akım çekilmiştir. Cihaz bu şekilde 2 adet kalem pil (AA) ile 1 yıl 3 ay çalışabilir. Şifre çözme işlemi aynı senaryoya göre 5,924 mJ enerji harcamıştır. Şekil 7.24'te görüldüğü gibi ortalama olarak 0,5894 mW en fazla ise 0,8605 mW güç tüketmiştir. Ortalama 0,1793 mA en fazla ise 0,2617 mA akım çekilmiştir. Cihaz bu şekilde 1 yıl 3 ay çalışabilir.

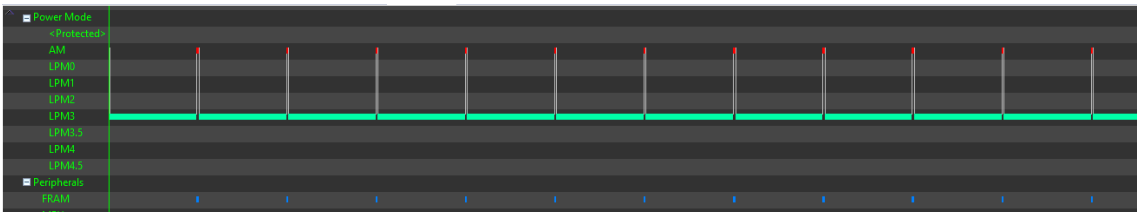


Şekil 7.23. LBLOCK şifreleme güç tüketimi

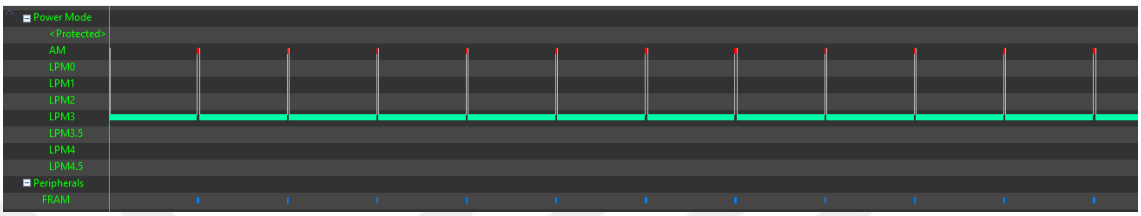


Şekil 7.24. LBLOCK şifre çözme güç tüketimi

MSP430FR5994, LBLOCK şifreleme işlemini 10 saniyelik çalışma zamanının %26,8'lik kısmında aktif mod olarak tamamlamış geri kalan %73,2'lik kısımda ise LPM modunda geçirmiştir. Bu durum Şekil 7.25'te görülebilir. Cihaz toplam enerji tüketiminin sadece %38,8'ini aktif modda şifreleme yaparken harcamıştır. Şifre çözme aşamasında ise MSP430FR5994, %26,9'luk kısmında aktif mod olarak tamamlamış geri kalan %71,3'lik kısımda ise LPM modunda geçirmiştir. Cihaz toplam enerji tüketiminin %39'unu aktif modda şifre çözme yaparken harcamıştır.



(Şifreleme)



(Şifre çözme)

Şekil 7.25. LBLOCK şifreleme ve şifre çözme için AM-LPM geçişleri

7.9. Test Sonuçları

6. Bölümde incelenen algoritmaların enerji ölçümleri Çizelge 7.1'de, güç ölçümleri Çizelge 7.2'de, akım ölçümleri Çizelge 7.3'te, tahmini pil ömürleri Çizelge 7.4'te, çalışma süresi ve harcanan enerjiye göre aktif mod oranları ise Çizelge 7.5'te verilmiştir.

Çizelge 7.1. İncelenen algoritmaların enerji tüketimleri

Şifre	Mimari	Blok Uzunluğu	Anahtar Uzunluğu	Döngü Sayısı	İşlem	Enerji (mJ)
PRESENT	SPN	64	80	31	Şifreleme	9,523
					Şifre çözme	9,573
CLEFIA	Feistel	128	128	18	Şifreleme	9,128
					Şifre çözme	9,128
			192	22	Şifreleme	9,452
					Şifre çözme	9,494
			256	26	Şifreleme	9,452
					Şifre çözme	9,575
PICCOLO	Feistel	64	80	25	Şifreleme	8,487
					Şifre çözme	8,490
			128	31	Şifreleme	8,533
					Şifre çözme	8,547
PRINCE	SPN	64	128	12	Şifreleme	7,513
					Şifre çözme	7,549
LBLOCK	Feistel	64	80	32	Şifreleme	5,812
					Şifre çözme	5,924

Çizelge 7.2. İncelenen algoritmaların güç tüketimleri

Şifre	**Anahtar Uzunluğu**	**Döngü Sayısı**	**İşlem**	**Güç (mW)**	
				MAK	ORT
PRESENT	80	31	Şifreleme	1,6383	0,9548
			Şifre çözme	1,6374	0,9591
CLEFIA	128	18	Şifreleme	1,6036	0,9186
			Şifre çözme	1,6123	0,9187
	192	22	Şifreleme	1,6040	0,9482
			Şifre çözme	1,6142	0,9527
	256	26	Şifreleme	1,6040	0,9482
			Şifre çözme	1,6141	0,9579
PICCOLO	80	25	Şifreleme	1,1419	0,8439
			Şifre çözme	1,1768	0,8449
	128	31	Şifreleme	1,1886	0,8495
			Şifre çözme	1,2431	0,8516
PRINCE	128	12	Şifreleme	1,1694	0,7578
			Şifre çözme	1,2797	0,7621
LBLOCK	80	32	Şifreleme	0,8295	0,5799
			Şifre çözme	0,8605	0,5894

Çizelge 7.3. İncelenen algoritmaların akım ölçümleri

Şifre	Anahtar Uzunluğu	Döngü Sayısı	İşlem	Akım (mA)	
				MAK	ORT
PRESENT	80	31	Şifreleme	0,4982	0,2905
			Şifre çözme	0,4980	0,2918
CLEFIA	128	18	Şifreleme	0,4877	0,2795
			Şifre çözme	0,4905	0,2795
	192	22	Şifreleme	0,4877	0,2885
			Şifre çözme	0,4911	0,2899
	256	26	Şifreleme	0,4877	0,2885
			Şifre çözme	0,4909	0,2914
PICCOLO	80	25	Şifreleme	0,3472	0,2568
			Şifre çözme	0,3579	0,2570
	128	31	Şifreleme	0,3614	0,2585
			Şifre çözme	0,3782	0,2591
PRINCE	128	12	Şifreleme	0,3863	0,2306
			Şifre çözme	0,3892	0,2319
LBLOCK	80	32	Şifreleme	0,2522	0,1764
			Şifre çözme	0,2617	0,1793

Çizelge 7.4. İncelenen algoritmalar için tahmini pil ömürleri

Şifre	**Anahtar Uzunluğu**	**Döngü Sayısı**	**İşlem**	**Pil Ömrü (2xAA pil)**
PRESENT	80	31	Şifreleme	9 ay 21 Gün
			Şifre çözme	9 ay 19 gün
CLEFIA	128	18	Şifreleme	10 ay 2 gün
			Şifre çözme	10 ay 3 gün
	192	22	Şifreleme	9 ay 23 gün
			Şifre çözme	9 ay 22 gün
	256	26	Şifreleme	9 ay 23 gün
			Şifre çözme	9 ay 19 gün
PICCOLO	80	25	Şifreleme	10 ay 27 gün
			Şifre çözme	10 ay 26 gün
	128	31	Şifreleme	10 ay 25 gün
			Şifre çözme	10 ay 24 gün
PRINCE	128	12	Şifreleme	1 yıl 9 gün
			Şifre çözme	1 yıl 7 gün
LBLOCK	80	32	Şifreleme	1 yıl 3 ay
			Şifre çözme	1 yıl 3 ay

Çizelge 7.5. İncelenen algoritmalar için aktif mod oranları

Şifre	Anahtar Uzunluğu	Döngü Sayısı	İşlem	Aktif Mod (Süre)	Aktif Mod (Enerji)
PRESENT	80	31	Şifreleme	%15,5	%23,0
			Şifre çözme	%15,8	%23,0
CLEFIA	128	18	Şifreleme	%9,9	%14,7
			Şifre çözme	%10,0	%14,8
	192	22	Şifreleme	%14,0	%20,4
			Şifre çözme	%14,0	%20,3
	256	26	Şifreleme	%15,4	%22,1
			Şifre çözme	%15,4	%22,1
PICCOLO	80	25	Şifreleme	%1,8	%2,9
			Şifre çözme	%1,9	%3,1
	128	31	Şifreleme	%1,8	%2,9
			Şifre çözme	%1,9	%3,1
PRINCE	128	12	Şifreleme	%26,8	%38,8
			Şifre çözme	%26,9	%39,0
LBLOCK	80	32	Şifreleme	%2,2	%3,9
			Şifre çözme	%2,2	%3,8

BÖLÜM 8

8. GÜVENLİ İLETİŞİM İÇİN TEST ORTAMININ KURULMASI VE TEST EDİLMESİ

8.1. Giriş

Nesnelerin interneti mimarisi içerisinde yer alan katmanlardan olan uygulama katmanı, cihazların iletişiminde uygulama düzeyinde iletilen mesajlar için önemli protokolleri barındırmaktadır. Bu katmanda tanımlı olan MQTT ve CoAP protokolleri en sık kullanılanlardır. Tez kapsamında kurulan iletişim ortamı MQTT ve CoAP için ayrı ayrı hazırlanmıştır. Hazırlanan test ortamlarındaki ağ paketleri broker veya sunucu cihaz içindeki Wireshark[15] yazılımı ile yakalanarak dinlenmiştir. Paketleri yakalamak için ağa bir saldırı düzenlenmemiştir. Wireshark, ağ üzerindeki paketlerinin analizi için kullanılan açık kaynak kodlu ve ücretsiz bir yazılımdır. Ağ trafiğinin anlık olarak izlenmesini, izlenen bu trafiğin kayıt edilmesini ve daha sonra incelenmesini sağlamaktadır. (Orebaugh, Ramirez, & Beale, 2006).

8.2. MQTT Test Ortamı

MQTT için hazırlanan test ortamında broker cihaz olarak Raspberry Pi 3 Model B+ kullanılmıştır. Raspberry Pi cihazını çalıştırabilmek için NOOBS[16] aracılığı ile Raspbian işletim sistemi kurulmuştur (Upton & Halfacree, 2014). Test ortamında MQTT iletişimini sağlayabilmek için açık kaynak olarak hazırlanmış Mosquitto[17] MQTT Broker

[15] https://www.wireshark.org/
[16] New Out Of the Box Software, Raspberry Pi cihazı için işletim sistemi kurmak amacıyla geliştirilmiş bir ön yükleyicidir.
[17] https://mosquitto.org/

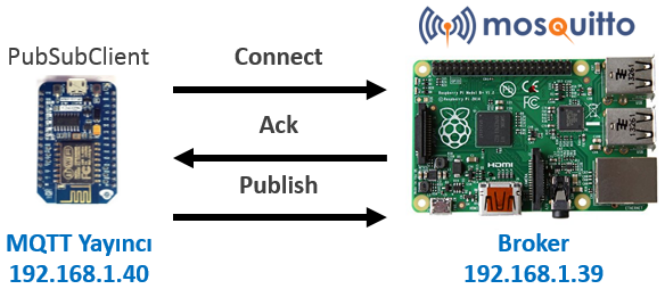
yazılımı seçilmiştir. Raspberry Pi içerisine Mosquitto kurabilmek için öncelikle Raspbian paketleri konsol ekranında aşağıdaki kod satırları ile güncellenmiştir.

```
sudo apt-get update

sudo apt-get upgrade

sudo reboot
```

Paketler güncellendikten sonra aşağıdaki kod satırı ile Raspberry Pi içerisine Mosquitto MQTT broker kurulmuştur (Hillar, 2017).

```
sudo apt install -y mosquitto mosquitto-clients
```

Test ortamında yayıncı ve abone olarak Arduino tabanlı NodeMCU LoLin geliştirme kartı tercih edilmiştir. NodeMCU LoLin üzerinde kablosuz iletişim için ESP8266 Wi-Fi modülü bulunan açık kaynak kodlu bir nesnelerin interneti platformudur. Genellikle uç cihaz olarak üzerinde sensörler yerleştirilmiş halde kullanılır (Al Dahoud & Fezari, 2018). LoLin kartının MQTT yayın ve abone mesajlarını iletmesi için PubSubClient[18] kütüphanesi kullanılmıştır. Bu kütüphane, MQTT broker ile temel yayın/abone mesajlarını iletmek için tasarlanmıştır (Oliveira vd., 2018). MQTT iletişimi için hazırlanan test ortamı Şekil 8.1'de verilmiştir.



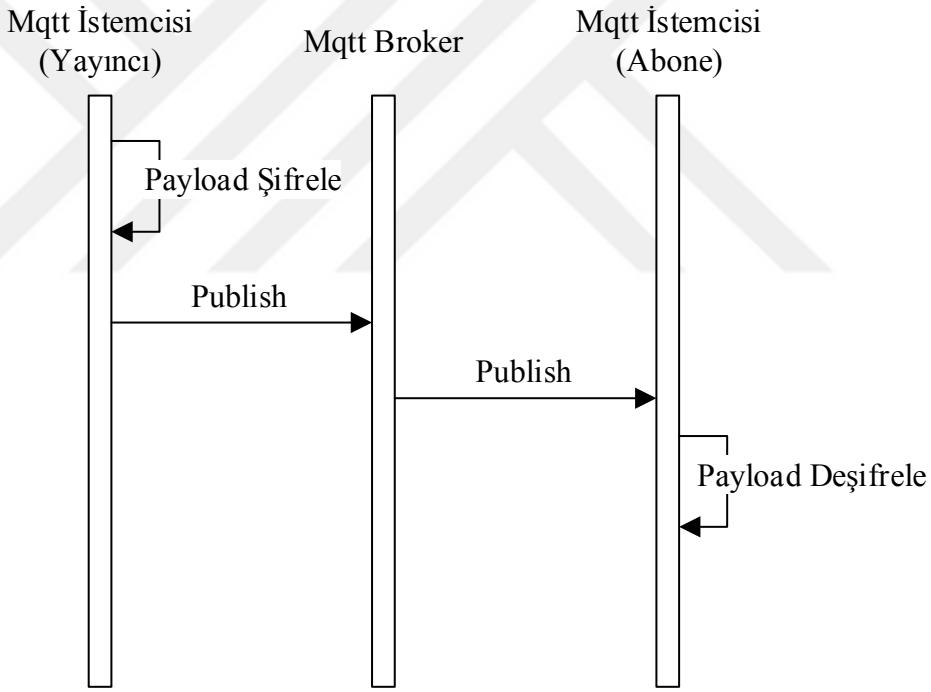
Şekil 8.1. MQTT test ortamı

MQTT güvenlik odaklı bir protokol değildir. Bu sebeple üçüncü şahıslar tarafından erişilmemesi gereken bilgiler iletilecekse paketler oluşturulmadan önce MQTT

[18] https://github.com/knolleary/pubsubclient

istemcisi mutlaka şifreleme yaparak mesajları iletmelidir. Bunun için istemci-istemci veya istemci-broker şifreleme/şifre çözme yaklaşımları tercih edilebilir.

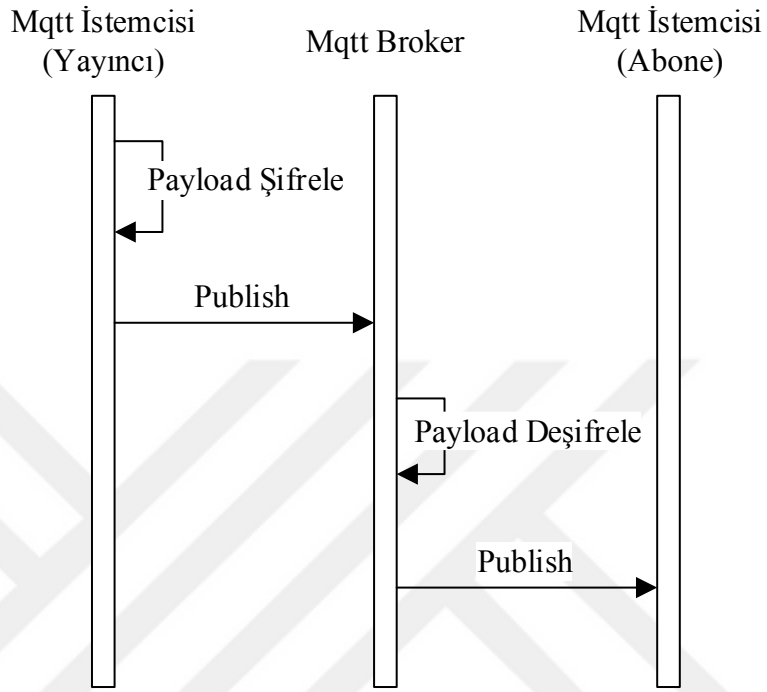
İstemci-istemci (yayıncı-abone) şifreleme/şifre çözme yaklaşımında PUBLISH paketi içerisindeki diğer verilerde bir değişiklik yapılmadan sadece payload kısmı şifrelenerek uçtan uca şifreleme gerçekleştirilebilir. PUBLISH paketi içindeki şifreli veriler broker tarafından incelenmeyecek, doğrudan abone olunan istemcilere gönderileceğinden dolayı mesaj ilgili abonelerde şifre çözme işlemi ile açık metne dönüştürülecektir. Eğer bir saldırgan MQTT paketini ele geçirirse anahtarı bilmediğinden şifreli mesajı çözemeyecektir. Bu durum Şekil 8.2'de verilmiştir. Eğer kimlik doğrulama veya TLS gibi güvenlik önlemleri kullanılmıyorsa uçtan uca şifreleme verileri korumak için en etkili yöntemdir.



Şekil 8.2. MQTT uçtan uca şifreleme

İstemci-broker şifreleme yaklaşımında ise iletinin payload kısmının istemci ve broker arasındaki iletişimde şifrelenmesi veya şifre çözme işlemi yapılması söz konusudur. İstemci iletiyi göndermeden önce şifreler ve iletiyi alan broker şifre çözme işlemini gerçekleştirerek üreteceği yeni paketi abonelere şifresiz olarak dağıtır. Bu durum özellikle güvenilir abone istemcileri için tercih edilebilir. Bu aboneler TLS bağlantısı ile

veriyi daha güvenli olarak alabilir. İstemci-broker şifreleme yaklaşımında yayıncı istemci ile broker arasında TLS bağlantısına ihtiyaç olmaz. Bu durum Şekil 8.3'te özetlenmiştir.



Şekil 8.3. MQTT istemci-broker şifreleme

Tez kapsamında hazırlanan güvenli MQTT iletişimi test ortamında Şekil 8.2 ve Şekil 8.3'te gösterildiği gibi iletilecek olan verinin payload kısmı şifrelenerek iletişim sağlanmıştır.

8.3. CoAP Test Ortamı

CoAP için hazırlanan test ortamında sunucu olarak Raspberry Pi 3 Model B+ ve CoAP sunucu yazılımı olarak txThings[19] kullanılmıştır (Tanganelli, Vallati, & Mingozzi, 2015). Raspberry Pi içerisine konsol üzerinden aşağıdaki kod satırı ile txThings uygulaması kurulmuştur.

```
sudo pip install twisted==15.1.0
```

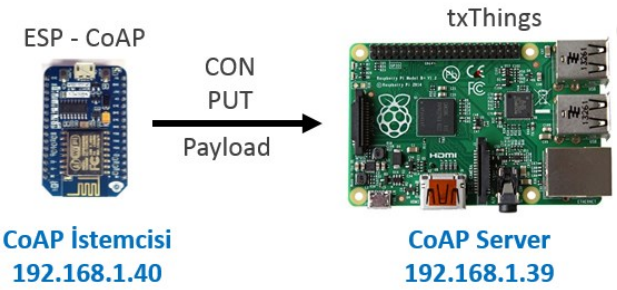
[19] https://github.com/mwasilak/txThings

```
sudo pip install txthings
```

Kurulum gerçekleştikten sonra ilgili klasör üzerinden Raspberry Pi içerisine CoAP sunucusu aşağıdaki kod satırları ile başlatılmıştır.

```
sudo python server.py
```

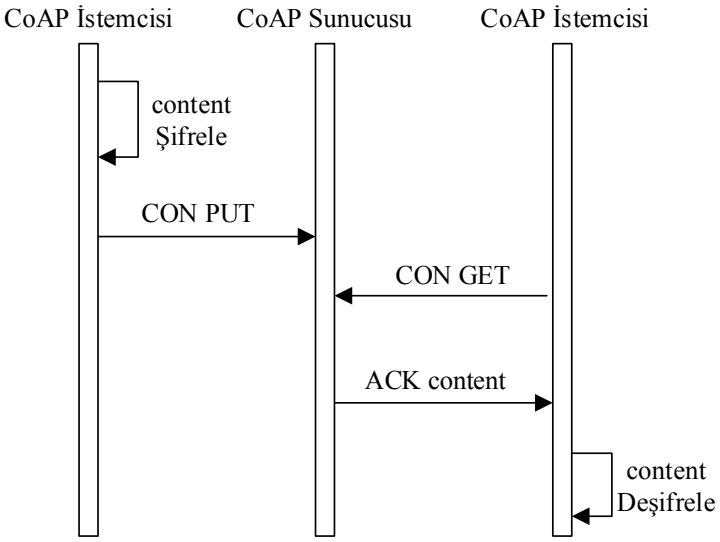
CoAP istemcisi olarak yine Arduino tabanlı NodeMCU LoLin geliştirme kartı tercih edilmiştir. İstemcinin CoAP mesajları gönderip alabilmesi için ESP-CoAP Kütüphanesi[20] kullanılmıştır (Prayogo, Mukhlis, & Yakti, 2019). CoAP iletişimi için hazırlanan test ortamı Şekil 8.4'te verilmiştir.



Şekil 8.4. CoAP test ortamı

CoAP, MQTT'ye benzer bir şekilde güvenlik odaklı bir protokol değildir. İletim ortamında veri istemciden sunucuya veya sunucudan istemciye aktarılırken şifrelenmeden aktarılır. Güvenliği sağlamak için istemci-sunucu-istemci arasında iletilecek veri şifrelenerek aktarım yapılabilir. İstemci ileteceği veriyi şifreleyerek CON PUT mesajı ile sunucuya iletir. Sunucu CON GET mesajı ile veriyi talep eden istemciye ACK content mesajı ile şifreli veriyi gönderir. Şifreli veriyi alan istemci şifre çözme işlemini yaparak açık metne ulaşır. Şekil 8.5'te bu durum gösterilmiştir. Tez kapsamında hazırlanan güvenli CoAP iletişimi test ortamında iletilecek olan verinin content kısmı şifrelenerek iletişim sağlanmıştır.

[20] https://github.com/automote/ESP-CoAP



Şekil 8.5. CoAP istemci-sunucu-istemci şifreleme

8.4. Test Senaryoları

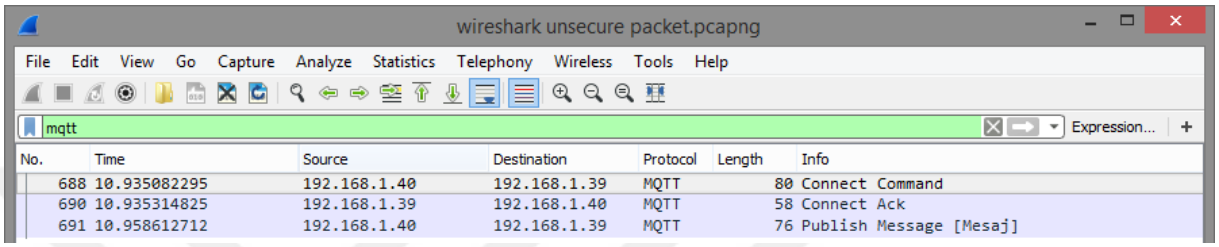
MQTT ve CoAP protokolleri için güvenli ve güvensiz iletişimi test etmek amacıyla senaryolar belirlenmiştir. Bu senaryolar Çizelge 8.1'de verilmiştir. Tez metninde sadece hafif sıklet algoritmalardan LBLOCK şifreleme algoritması ile yapılan güvenli iletişim ortamı gösterilmiştir. Fakat tez çalışması boyunca tüm algoritmalar ile bu senaryolar gerçeklenmiştir.

Çizelge 8.1. Test senaryoları

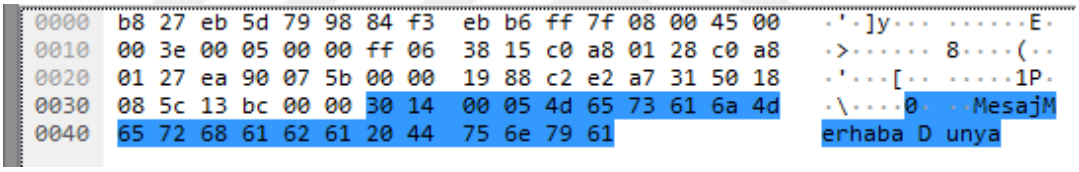
Senaryo Numarası	**İletişim Protokolü**	**Şifreleme Yapıldı mı?**	**İletilen Konu**	**İletilen Mesaj**
Test Senaryosu 1	MQTT	Hayır	Mesaj	Merhaba Dunya
Test Senaryosu 2	CoAP	Hayır	Mesaj	Merhaba Dunya
Test Senaryosu 3	MQTT	Evet	Mesaj	Merhaba Dunya
Test Senaryosu 4	CoAP	Evet	Mesaj	Merhaba Dunya

8.4.1. Test Senaryosu 1

192.168.1.40 IP adresli yayıncı ile 192.168.1.39 IP adresli MQTT broker arasındaki Şekil 8.1'de gösterilen iletişim ortamı kurularak gerçekleşen MQTT haberleşmesi Wireshark yazılımı ile yakalanmıştır. Buna göre "Mesaj" konusuna şifreleme yapılmadan "Merhaba Dunya" mesajı iletilmiştir. Yakalanan paketler ve paketlerin içeriği Şekil 8.6 ve Şekil 8.7'de verilmiştir.



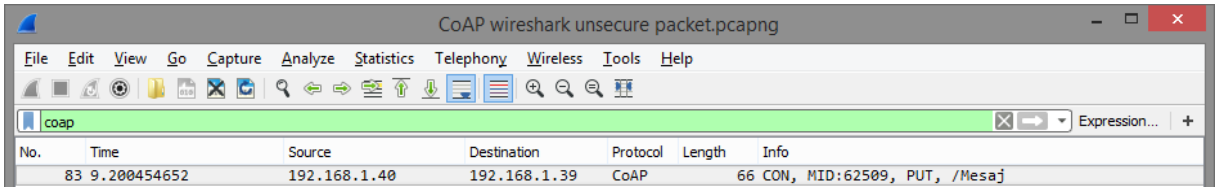
Şekil 8.6. Wireshark ile yakalanan MQTT paketleri (*Test Senaryosu1*)



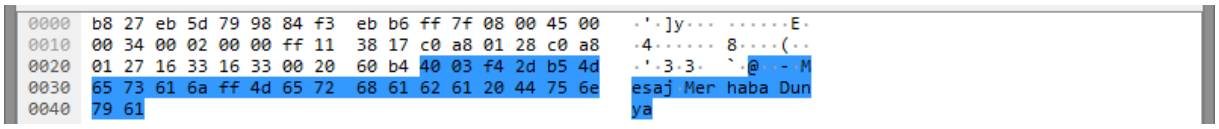
Şekil 8.7. Yakalanan MQTT yayın paketinin içeriği (*Test Senaryosu1*)

8.4.2. Test Senaryosu 2

192.168.1.40 IP adresli istemci ile 192.168.1.39 IP adresli CoAP sunucusu arasındaki Şekil 8.4'te gösterilen iletişim ortamı kurularak gerçekleşen CoAP haberleşmesi Wireshark yazılımı ile yakalanmıştır. Yapılan haberleşmede "Mesaj" kaynağına şifreleme yapılmadan "Merhaba Dunya" mesajı iletilmiştir. Yakalanan paketler ve paketlerin içeriği Şekil 8.8 ve Şekil 8.9'da verilmiştir.



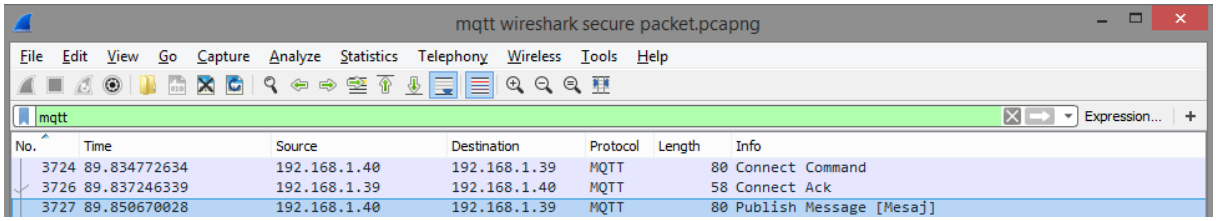
Şekil 8.8. Wireshark ile yakalanan CoAP paketi (*Test Senaryosu2*)



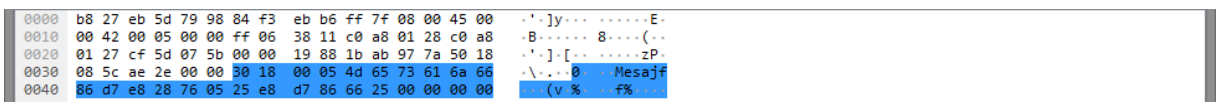
Şekil 8.9. Yakalanan CoAP paketinin içeriği (*Test Senaryosu2*)

8.4.3. Test Senaryosu 3

192.168.1.40 IP adresli yayıncı ile 192.168.1.39 IP adresli MQTT broker arasındaki Şekil 8.1'de gösterilen iletişim ortamı kurularak gerçekleşen MQTT haberleşmesi Wireshark yazılımı ile yakalanmıştır. Buna göre "Mesaj" konusu tezin 6. bölümünde incelenen hafif sıklet LBLOCK algoritması ile şifrelenerek "Merhaba Dunya" mesajı iletilmiştir. Yakalanan paketler ve paketlerin içeriği Şekil 8.10 ve Şekil 8.11'de verilmiştir.



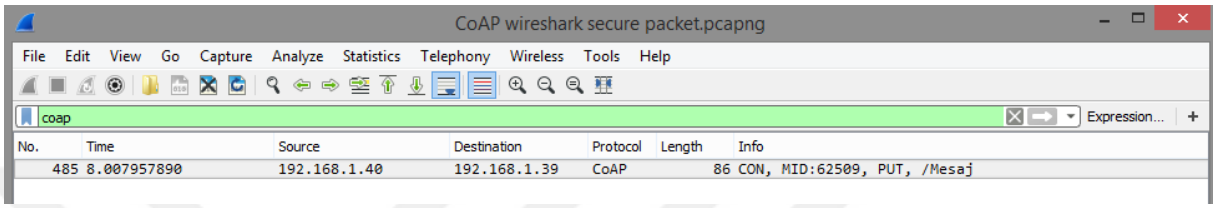
Şekil 8.10. Wireshark ile yakalanan MQTT paketleri (*Test Senaryosu3*)



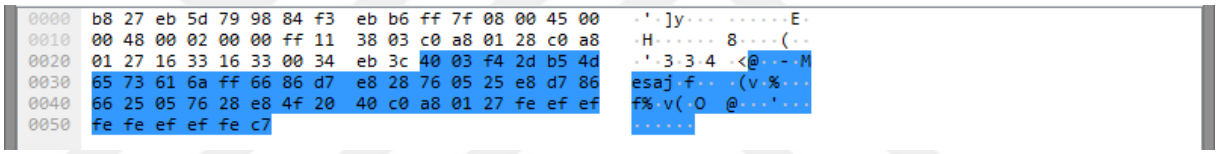
Şekil 8.11. Yakalanan MQTT yayın paketinin içeriği (*Test Senaryosu3*)

8.4.4. Test Senaryosu 4

192.168.1.40 IP adresli istemci ile 192.168.1.39 IP adresli CoAP sunucusu arasındaki Şekil 8.4'te gösterilen iletişim ortamı kurularak gerçekleşen CoAP haberleşmesi Wireshark yazılımı ile yakalanmıştır. Yapılan haberleşmede "Mesaj" kaynağı LBLOCK algoritması ile şifrelenerek "Merhaba Dunya" mesajı iletilmiştir. Yakalanan paketler ve paketlerin içeriği Şekil 8.12 ve Şekil 8.13'te verilmiştir.



Şekil 8.12. Wireshark ile yakalanan CoAP paketi (*Test Senaryosu4*)



Şekil 8.13. Yakalanan CoAP paketinin içeriği (*Test Senaryosu4*)

8.5. Test Sonuçları

Şekil 8.7 ve Şekil 8.9'daki ağ paketlerinin içeriği incelendiğinde, ağ üzerindeki MQTT ve CoAP iletişiminin güvensiz bir şekilde olduğu görülebilir. Öte yandan bu güvenlik zafiyeti MQTT için Şekil 8.2 ve Şekil 8.3, CoAP için ise Şekil 8.5'te olduğu gibi istemci veya yayıncı cihaz üzerine şifreleme ve şifre çözme algoritmaları eklenerek aşılabilir. Bu durum Şekil 8.11 ve Şekil 8.13'te yakalanan ağ paketlerinde gözlemlenmiştir.

BÖLÜM 9

9. TEST ORTAMINA SALDIRI SENARYOSUNUN UYGULANMASI

9.1. Giriş

Bu bölümde, tezin 8. bölümde kurulan MQTT ve CoAP için güvenli iletişim test ortamına günümüzde yerel ağlarda rastlanan ve en tehlikeli saldırılardan biri olan ortadaki adam saldırısı karşısında test edilmiştir. Saldırı nesnelerin interneti uygulamasının çalıştığı bir yerel ağa bağlanan saldırgan bilgisayar tarafından yapılmıştır. Saldırı yazılımı olarak Kali Linux işletim sistemi içindeki WebSploit tercih edilmiştir. Saldırının başarı ile sonuçlanmasından sonra saldırgan bilgisayar üzerindeki Wireshark yazılımı ile MQTT ve CoAP paketleri analiz edilmiştir.

9.2. Ortadaki Adam Saldırısı (Man-In-The-Middle MITM)

MITM, bilişim güvenliğinde güvenlik uzmanları tarafından endişe yaratan en iyi bilinen saldırılardan biridir ve bilgisayar korsanlarının yerel ağda ilk başvurduğu dinleme saldırıları arasındadır. Ayrıca MITM saldırısı, DoS ve DNS kimlik sahtekârlığı gibi daha karışık saldırıları başlatmak için de ön hazırlık aşamasında kullanılır. MITM saldırısı özellikle ARP zehirlenmesi (ARP poisoning) yapılabilecek bir LAN ortamında mümkündür. Bu tip dinleme saldırıları sonunda, kullanıcı kimliği, parola, oturum bilgileri gibi kullanıcılar için önemli olan veriler çalınabilir (Conti, Dragoni, & Lesyk, 2016).

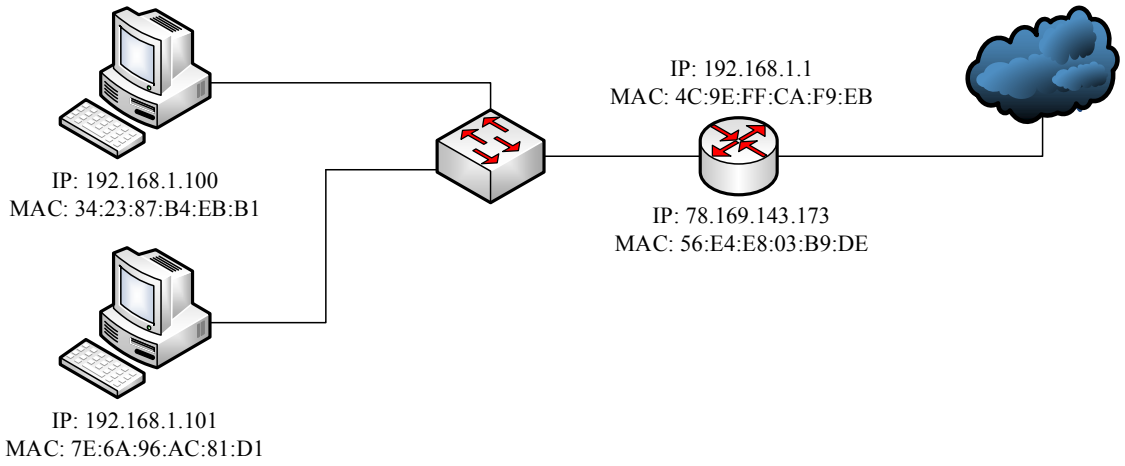
9.3. ARP Zehirlenmesi

Yerel ağ ortamında veri aktarımı MAC adresleri kullanılarak gerçekleştirilir. MAC adresi TCP/IP protokol kümesinin veri bağlantısı katmanında çalışır. Bir yerel ağda iletişim kurmak için IP adresini MAC adresine dönüştürmek önemlidir. MAC adresleri 48 bit, IP adresleri 32 bittir. Bu işlemi gerçekleştirmek için kullanılan protokole ARP (Adres Çözümleme Protokolü) denir. Ağdaki bir cihaz veri göndermek istediğinde, IP adresini bildiği alıcı cihazın MAC adresini bulmak için ARP önbelleğini (cache) kullanır (Forouzan, 2009).

TCP/IP kümesindeki ARP paketleri, ARP isteği (request) ve ARP yanıtı (reply) olmak üzere iki türdür. Gönderen, veri iletmek istediği hedefin MAC adresini öğrenmek istediğinde ARP isteğini ağdaki her cihaza yayınlar. Hedef cihaz ARP yanıtını, tek noktaya yayın (unicast) modunda içerisine MAC adresi yazılmış bir şekilde gönderene iletir. Yanıt alındıktan sonra, alıcı yani ARP istek paketini gönderen gelecekteki iletişimi hızlandırmak için IP ve MAC adresi eşleşmesini önbelleğe kaydeder. Normal bir koşulda tüm cihazlar bu sistematiğe bağlı kalarak doğru IP ve MAC adresi ile iletişim kurarlar.

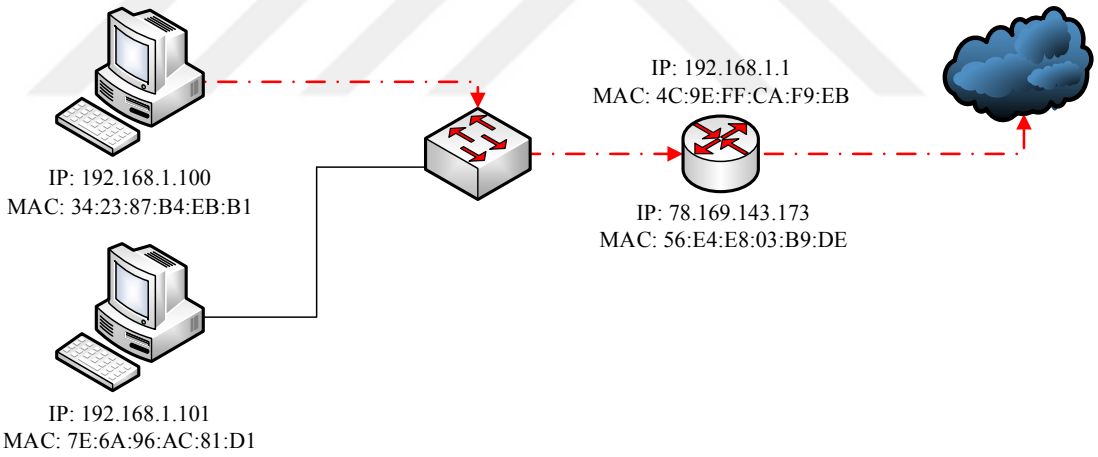
ARP sisteminin kendi sınırlamaları vardır. ARP herhangi bir kimlik doğrulaması yapmaz diğer bir deyişle IP-MAC adresi eşleşmesinin doğru olup olmadığını test etmez. Öte yandan ARP, ağdaki diğer cihazların ARP yanıtı göndermesini engellemez. Bu kontrollerin olmaması yerel ağda MAC adresi sahtekârlığı işleminin gerçekleşmesine zemin oluşturmaktadır. Kimlik doğrulama kontrolü yapılmadığından saldırgan ağ trafiğini aynı yerel ağdaki kendi cihazına veya başka bir cihaza yönlendirebilir. Temel olarak bu tip saldırılarda saldırgan, ARP isteği ile sorgulanan IP adreslerine sahip olduğunu iddia eden sahte bir ARP yanıt paketi gönderir. Böylelikle istekte bulunan cihazın önbelleğine sahte veriler kaydedilir. Bu işleme ARP zehirlenmesi denir. Bu zehirlenme tüm ağ trafiğinin saldırganın cihazına iletilmesine olanak sağlar.

Şekil 9.1'de, 192.168.1.100 ve 192.168.1.101 IP adresine sahip cihazlar katman 2 anahtar üzerinden 192.168.1.1 adresine sahip bir ağ geçidi ile WAN'a bağlanmaktadır.



Şekil 9.1. ARP zehirlenmesi için örnek senaryo

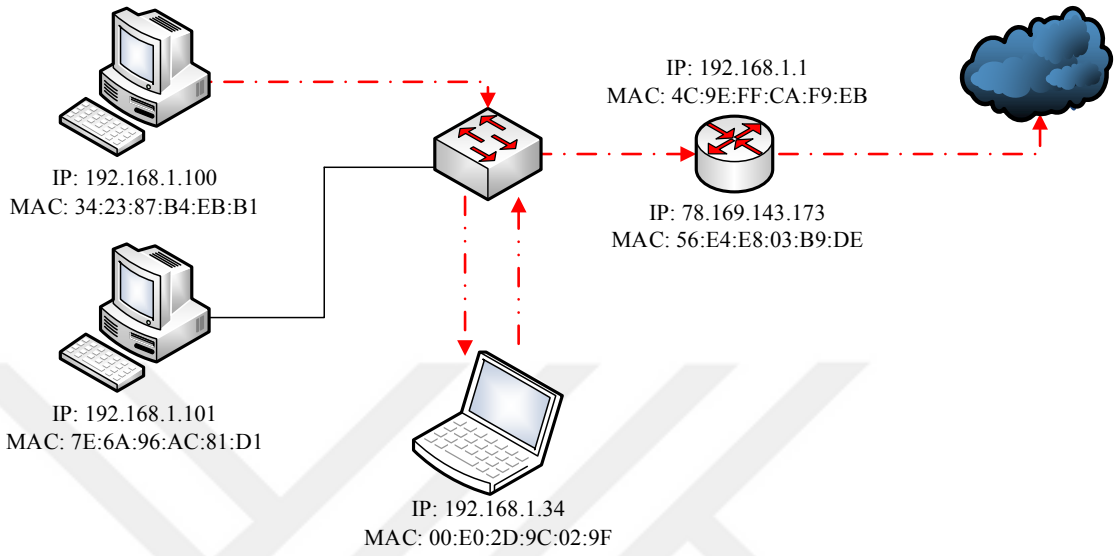
192.168.1.100 IP adresli bilgisayar dış ağa veri göndereceği zaman ağ paketini oluşturur ve anahtara iletir. Paket, anahtar üzerinden yönlendiriciye ve oradan da dış ağa iletilir. Bu veri iletiminin yönü Şekil 9.2'de gösterilmiştir.



Şekil 9.2. ARP zehirlenmesi öncesi veri iletimi

Aynı yerel ağa bağlı olan 192.168.1.34 IP adresli saldırgan ARP zehirlenmesi başlatarak kendi MAC adresini ağ geçidinin MAC adresi olarak yerel ağa duyurur. Bu durumda ağa bağlı olan tüm cihazlar ağ geçidi olarak saldırganın cihazını kabul ederler ve geniş alan ağına iletilecek tüm paketleri saldırgana gönderirler. Saldırgan paketleri alır, analiz eder ve daha sonrasında 192.168.1.1 IP adresli gerçek ağ geçidine iletir. Dış ağdan

paketlerin geri dönüşü de aynı şekilde saldırgan üzerinden gerçekleşir. Dış ağ ile iletişim kurabilen kurban cihazlar böylelikle ağda herhangi bir aksama hissetmemiş olurlar. ARP zehirlenmesi sonrasında veri iletimi ise Şekil 9.3'te gösterilmiştir.



Şekil 9.3. ARP zehirlenmesi sonrası veri iletimi

ARP zehirlenmesi sonrasında saldırgan ağdaki paketleri aldığı için paket analizi yaparak koklama (sniffing) veya kimlik sahtekârlığı (spoofing) gibi birçok saldırıyı gerçekleştirebilir. ARP zehirlenmesini önlemek veya güvenliği sağlayabilmek için bazı yöntemler mevcuttur. Bu yöntemlerden bazıları aşağıdaki gibi sıralanabilir.

- Statik MAC adresi kullanma,
- VPN kullanımı,
- Kontrollü paket analizi,
- Şifreleme.

Bu yöntemlerin bazıları sistem yöneticilerine ciddi iş yükü getirmektedir. Örneğin cihazların statik IP ve MAC adresleriyle eşlenmesi, bu tip saldırıları kısmen engelleyebilirken özellikle büyük ağlarda yönetimi zorlaştırmaktadır. Öte yandan bazı paket analiz ve filtreleme yazılımları ile ağ trafiğindeki anomaliler tespit edilerek saldırılar önlenebilir. Fakat bu durum ağ trafiğinde fark edilebilecek bir boyutta yavaşlamaya neden olabilir. VPN kullanılması ise bir başka çözüm olarak sunulabilir ama

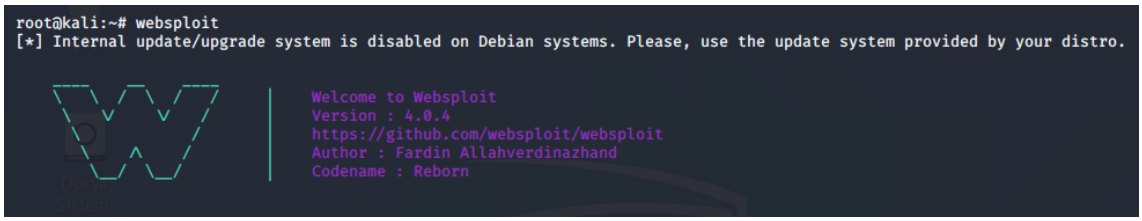
bu durum maliyeti ve karmaşıklığı artırır. Diğer bir çözüm olarak ağın içerisinde HTTPS ve SSL gibi güvenli protokollerin kullanılması mümkündür. Elbette bu çözüm de özelikle büyük ağlarda maliyeti artıracaktır. Nesnelerin interneti uygulamalarındaki bu güvenlik sorununu ortadan kaldırabilecek en iyi ve maliyet açısından makul çözümlerden biri, ağın işleyişini değiştirmeden verilerin güvenli bir şekilde iletilmesini sağlamak amacıyla paketlerin şifrelenmesi olarak düşünülebilir.

9.4. Kali Linux ile MITM Saldırısı

Kali Linux, bilgi güvenliği eğitimi ve penetrasyon testi gibi hizmetleri sağlayan Offensive Security şirketi tarafından geliştirilen ve finanse edilen açık kaynaklı Debian tabanlı bir Linux dağıtımıdır. Kali Linux, penetrasyon testi, güvenlik zafiyet araştırması, adli tıp bilişimi ve tersine mühendislik gibi bilgi güvenliği alanına yönelik birçok aracı içerisinde barındırmaktadır. Kali Linux içerisinde MITM saldırısı düzenlemek için birçok araç mevcuttur (Najera-Gutierrez & Ansari, 2018). Bu çalışmada WebSploit kullanılarak MITM saldırısı düzenlenmiştir. Ayrıca Wireshark yazılımı ile ağ üzerindeki paketler analiz edilmiştir.

9.5. WebSploit

WebSploit, çeşitli güvenlik açıklarını bulmak amacıyla sistemi taramak ve analiz etmek için kullanılan açık kaynaklı python programlama dili ile geliştirilmiş bir yazılımdır. İçerisinde ARP zehirlenmesi, HTTP trafiğini koklama, ağdaki cihazları tespit etmek için ağı tarama, kablosuz ağları tarama ve sahte kablosuz ağlar oluşturmak ile ilgili birçok araç mevcuttur. WebSploit, Şekil 9.4'te gösterildiği gibi konsol ekranında **`websploit`** komutu ile çalıştırılabilir.



Şekil 9.4. WebSploit yazılımının çalıştırılması

WebSploit içerisindeki araçları ve açıklamalarını görebilmek için **show** veya **show modules** komutu konsol üzerinde çalıştırılabilir. 4.0.4 versiyonunda 7 adet araç mevcuttur. Bu araçlardan bazıları ağı tarayıp bilgi almak için kullanılırken bazıları da kimlik sahtekârlığı için kullanılmaktadır. WebSploit içindeki saldırı araçları Şekil 9.5'te gösterilmiştir.



Şekil 9.5. WebSploit içerisindeki saldırı araçları

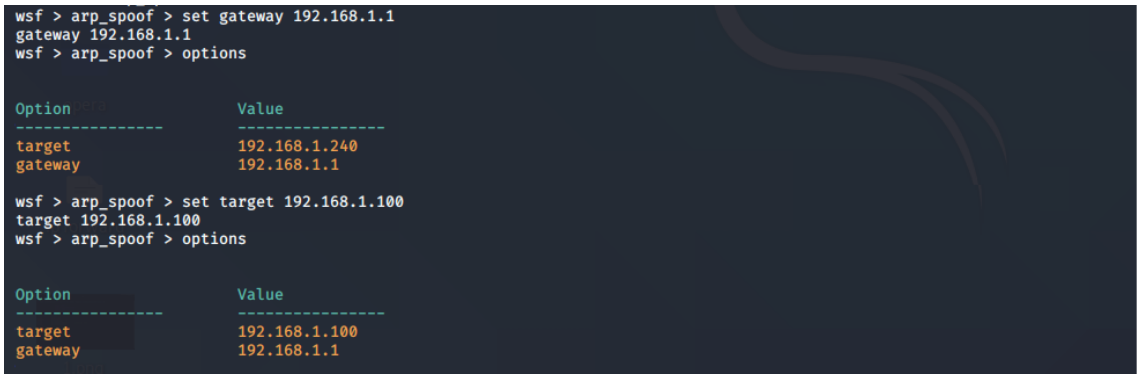
Bir yerel ağ içinde ARP zehirlenmesi saldırısı yapabilmek için WebSploit içindeki arp_spoof aracı kullanılmaktadır. Bunun için WebSploit başlatıldıktan sonra **use arp_spoof** komutu çalıştırılır. Ağın hedef ve ağ geçidi bilgilerini görebilmek için **options** komutu çalıştırılabilir. Bu işlem Şekil 9.6'da gösterilmiştir.



Şekil 9.6. WebSploit içerisindeki arp_spoof aracı

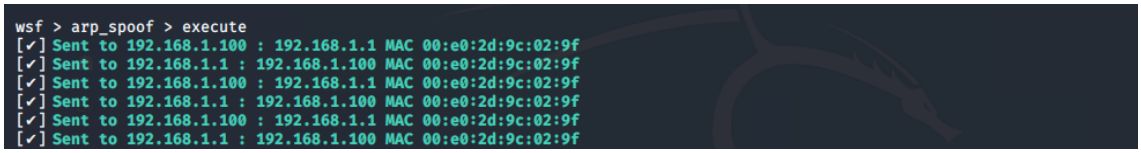
Şekil 9.6'da listelenen hedef (target) ve ağ geçidi (gateway) IP adresleri WebSploit için varsayılan değerlerdir ve ARP zehirlenmesi saldırısı yapılacak ağa göre ayarlanmalıdır. Bunun için **set gateway** ve **set target** komutları kullanılır. Şekil 9.3'te verilen senaryoya göre 192.168.1.34 IP adresli saldırgan ağın trafiğini ele geçirmek istemekte ve 192.168.1.100 IP adresli kurbanın trafiğini dinlemek istemektedir. Bu senaryoda ARP zehirlenmesi saldırısında saldırgan kendini ağa 192.168.1.1 IP adresli ağ geçidi olarak tanıtmak durumunda ve hedef olarak da 192.168.1.100 IP adresli bilgisayarı belirlemelidir. Saldırgan WebSploit içerisindeki arp_spoof aracında

`set target 192.168.1.100` ve **`set gateway 192.168.1.1`** komutları ile saldırı yapacağı ağın bilgilerini ayarlar. Bu durum Şekil 9.7'de verilmiştir.



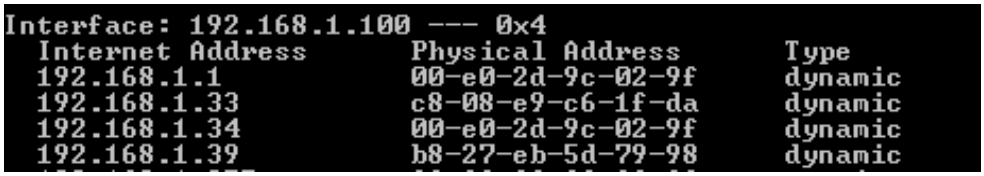
Şekil 9.7. WebSploit ARP zehirlenmesi saldırısı için bilgilerin ayarlanması

Gerekli ayarlamalar yapıldıktan sonra saldırgan WebSploit içerisindeki arp_spoof aracında Şekil 9.8'de olduğu gibi **`execute`** komutu ile ARP zehirlenmesi saldırısını başlatır. Saldırı başladıktan sonra saldırgan, hedefe kendisinin ağ geçidi olduğunu belirten ARP paketleri göndererek kurbanın ARP önbelleğini doldurur ve zehirlenme işlemini tamamlar. Bundan sonra kurban tüm veri iletişiminde ağ paketlerini saldırgana gönderecektir.



Şekil 9.8. ARP zehirlenme saldırısının başlatılması

ARP zehirlenmesi yapılan ağda hedef bilgisayarın ARP önbellek tablosuna konsolda **`arp -a`** komutu ile ulaşılabilir. Bu önbellek tablosu Şekil 9.9'da verilmiştir. Buna göre şekil incelendiğinde 192.168.1.34 IP adresli saldırgan ile 192.168.1.1 IP adresli ağ geçidinin MAC adresinin aynı olduğu görülebilir. Diğer bir deyişle saldırgan yerel ağın tüm protokol kurallarını iptal ederek kendini ağ geçidi olarak hedef bilgisayara zorla kabul ettirmiştir.



Şekil 9.9. Hedef bilgisayarın ARP önbellek tablosu

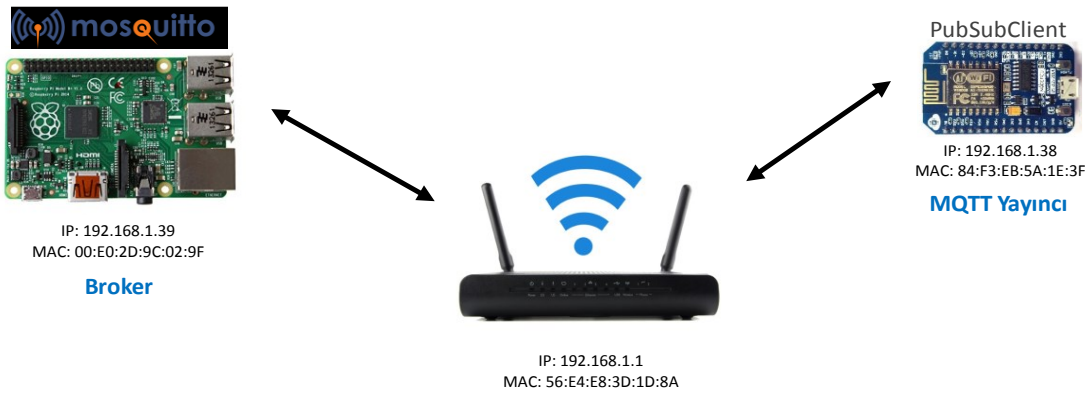
Şekil 9.10'da yerel ağın ARP zehirlenmesi saldırısı esnasındaki ARP paketlerinin bazıları Wireshark yazılımı ile analiz edilerek listelenmektedir. İlk olarak 846 numaralı pakette, saldırgan ARP istek paketi ile 192.168.1.100 IP adresli hedefin MAC adresini sorgulamaktadır. TCP/IP kuralları gereği kurban bu isteğe ARP yanıt paketi ile cevap vererek 887 numaralı paketi göndermiştir. Daha sonra saldırgan 892 numaralı paket ile 192.168.1.1 IP adresli ağ geçidinin kendi MAC adresi ile eşleştiğini kurbana iletmektedir. Ayrıca saldırgan 909 numaralı paket ile ağ geçidinin MAC adresini öğrenmek istediğini ARP istek paketi ile tüm ağa iletmiş ve ağ geçidi saldırgana MAC adresini göndermek için bu isteği 962 numaralı paket ile yanıtlamıştır. 1024 numaralı paket ile saldırgan ağ geçidine, kurbana gönderilecek olan paketleri alabilmek için 192.168.1.100 IP adresinin kendisine ait olduğunu iletmiştir. Bu esnada ağ geçidi 192.168.1.1 IP adresinin çakıştığını belirtse de veri bağlantısı katmanında işleyen MAC adres tabanlı iletişim bozulmaz. ARP zehirlenmesi saldırısı bu şekilde hedef cihazların ARP önbelleği dolana kadar devam eder. Sonuçta ağ trafiği ister istemez saldırgan üzerinden akarak sağlanır ve MITM saldırısı başarı ile sonuçlanır.

No.	Source	Destination	Protocol	Length	Info
846	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.100? Tell 192.168.1.34
887	HonHaiPr_b4:eb:b1	Innomedi_9c:02:9f	ARP	42	192.168.1.100 is at 34:23:87:b4:eb:b1
892	Innomedi_9c:02:9f	HonHaiPr_b4:eb:b1	ARP	42	192.168.1.1 is at 00:e0:2d:9c:02:9f
909	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.34
962	ZyxelCom_ca:f9:eb	Innomedi_9c:02:9f	ARP	42	192.168.1.1 is at 4c:9e:ff:ca:f9:eb
1024	Innomedi_9c:02:9f	ZyxelCom_ca:f9:eb	ARP	42	192.168.1.100 is at 00:e0:2d:9c:02:9f (duplicate use of 192.168.1.1 detected!)
1229	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.100? Tell 192.168.1.34
1282	HonHaiPr_b4:eb:b1	Innomedi_9c:02:9f	ARP	42	192.168.1.100 is at 34:23:87:b4:eb:b1
1371	Innomedi_9c:02:9f	HonHaiPr_b4:eb:b1	ARP	42	192.168.1.1 is at 00:e0:2d:9c:02:9f
1414	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.34
1417	ZyxelCom_ca:f9:eb	Innomedi_9c:02:9f	ARP	42	192.168.1.1 is at 4c:9e:ff:ca:f9:eb
1422	Innomedi_9c:02:9f	ZyxelCom_ca:f9:eb	ARP	42	192.168.1.100 is at 00:e0:2d:9c:02:9f (duplicate use of 192.168.1.1 detected!)
1616	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.100? Tell 192.168.1.34
2335	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.102? Tell 192.168.1.34
2345	SamsungE_90:8a:7b	Innomedi_9c:02:9f	ARP	42	192.168.1.102 is at d4:e6:b7:90:8a:7b
2347	SamsungE_90:8a:7b	Innomedi_9c:02:9f	ARP	42	192.168.1.102 is at d4:e6:b7:90:8a:7b
3330	Innomedi_9c:02:9f	HonHaiPr_b4:eb:b1	ARP	42	192.168.1.1 is at 00:e0:2d:9c:02:9f
3343	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.34
3344	ZyxelCom_ca:f9:eb	Innomedi_9c:02:9f	ARP	42	192.168.1.1 is at 4c:9e:ff:ca:f9:eb
3377	Innomedi_9c:02:9f	ZyxelCom_ca:f9:eb	ARP	42	192.168.1.100 is at 00:e0:2d:9c:02:9f (duplicate use of 192.168.1.1 detected!)
3511	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.100? Tell 192.168.1.34
3528	HonHaiPr_b4:eb:b1	Innomedi_9c:02:9f	ARP	42	192.168.1.100 is at 34:23:87:b4:eb:b1
3553	Innomedi_9c:02:9f	HonHaiPr_b4:eb:b1	ARP	42	192.168.1.1 is at 00:e0:2d:9c:02:9f
3571	Innomedi_9c:02:9f	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.34
3572	ZyxelCom_ca:f9:eb	Innomedi_9c:02:9f	ARP	42	192.168.1.1 is at 4c:9e:ff:ca:f9:eb
3585	Innomedi_9c:02:9f	ZyxelCom_ca:f9:eb	ARP	42	192.168.1.100 is at 00:e0:2d:9c:02:9f (duplicate use of 192.168.1.1 detected!)

Şekil 9.10. ARP zehirlenmesi saldırısı esnasındaki ARP paketleri

9.6. MQTT iletişim ortamına MITM Saldırı Senaryosu

Tezin 8. bölümünde test edilen MQTT iletişim ortamına MITM saldırısı düzenlenmiştir. Basit bir MQTT ortamı Şekil 9.11'de verilmiştir. Buna göre 192.168.1.39 IP adresli broker ve 192.168.1.38 IP adresli yayıncı 192.168.1.1 IP adresli ağ geçidine kablosuz (IEEE 802.11) ağ üzerinden bağlanmıştır. Broker ve yayıncı cihazlar birbirleri ile MQTT iletişimi kurmaktadır.



Şekil 9.11. Tez çalışmasında MQTT iletişim için kullanılan ortam

Hazırlanan senaryoya göre Şekil 9.11'de verilen MQTT iletişim ortamındaki yerel ağa bağlanan 192.168.1.34 IP adresli saldırgan, yayıncının broker cihaza gönderdiği

mesajları alabilmek için ARP zehirlenmesi saldırısı yaparak ortadaki adam rolünü almak istemektedir. Bu durum Şekil 9.12'de gösterilmiştir.



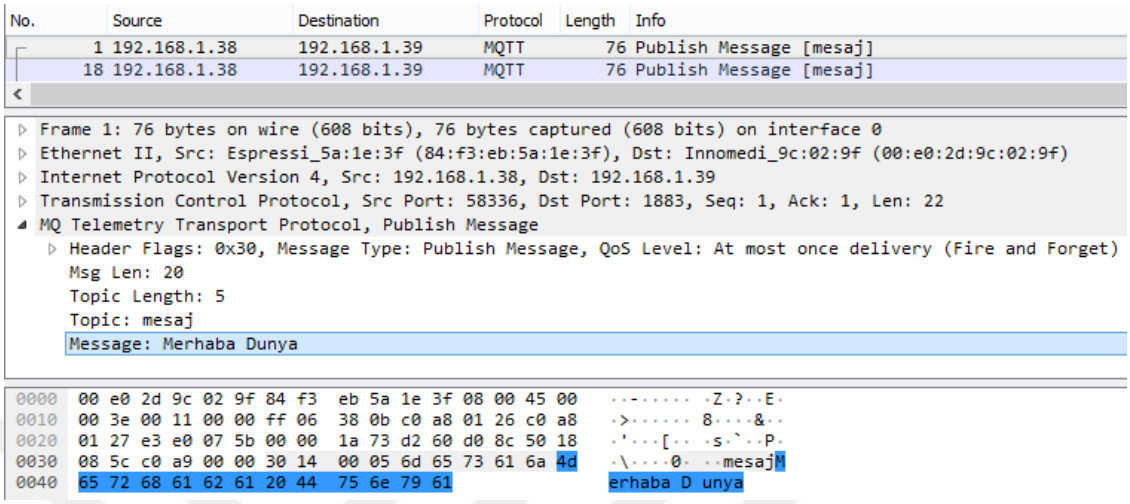
Şekil 9.12. MQTT iletişimi için MITM saldırısı

Saldırgan ağ trafiğini ele geçirebilmek için ARP zehirlenmesi saldırısında hedef olarak yayıncıyı, ağ geçidi olarak ise broker cihazı seçmelidir. WebSploit kullanarak yapılan saldırıda aşağıdaki komutlar kullanılır. İşlemler tamamlandıktan sonra saldırgan, broker ve yayıncı arasındaki ağ trafiğini kendi üzerinden geçirerek paketleri analiz edebilir.

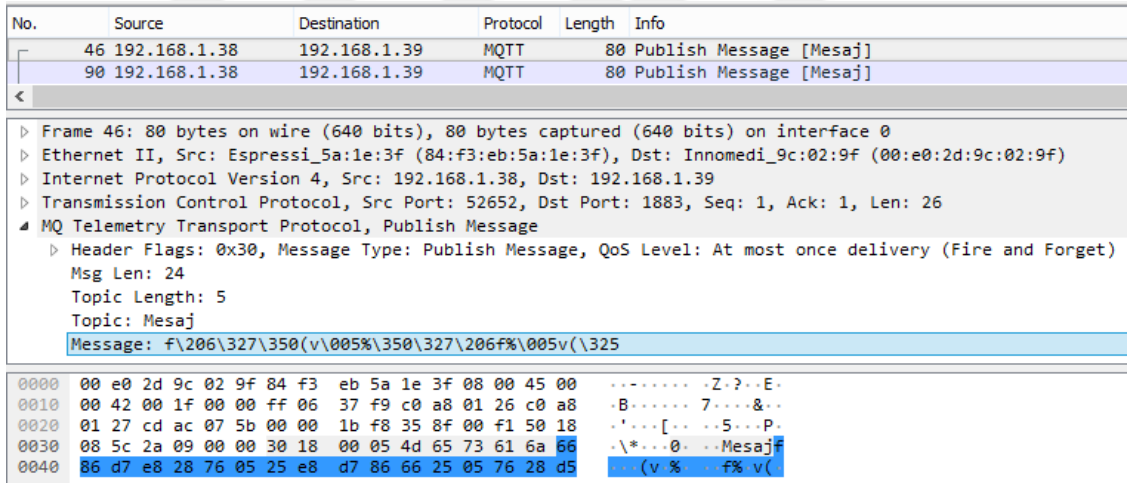
```
root@kali:~# websploit
wsf > use arp_spoof
wsf > arp_spoof > set target 192.168.1.38
target 192.168.1.38
wsf > arp_spoof > set gateway 192.168.1.39
gateway 192.168.1.39
wsf > arp_spoof > execute
```

Şekil 9.13'te verilen saldırganın ele geçirdiği MQTT paketleri incelendiğinde şifreleme yapılmadan gönderilen paketlerin tüm içeriğinin kolaylıkla ele geçirebildiği görülebilir. Buradan mesaj başlığının (topic) "mesaj" ve içeriğin (payload) "Merhaba Dunya" olduğu rahatlıkla söylenebilir. Öte yandan Şekil 9.14'te saldırganın yakaladığı

şifrelenmiş MQTT paketi görülmektedir. Burada mesaj başlığının “mesaj” olduğu belirli olmasına rağmen içerik şifrelendiği için anlaşılamaz durumdadır.



Şekil 9.13. Saldırganın yakaladığı şifrelenmemiş MQTT paketleri



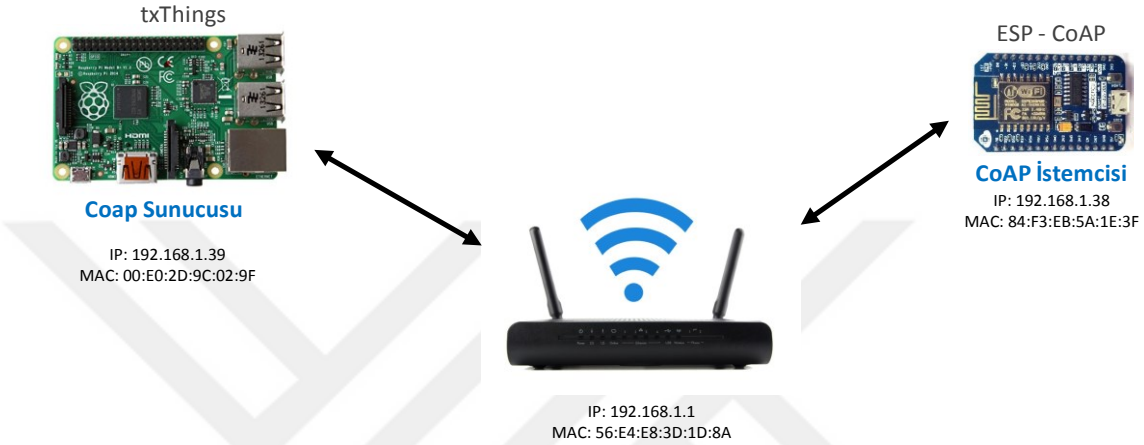
Şekil 9.14. Saldırganın yakaladığı şifrelenmiş MQTT paketleri

9.7. CoAP iletişim ortamına MITM Saldırı Senaryosu

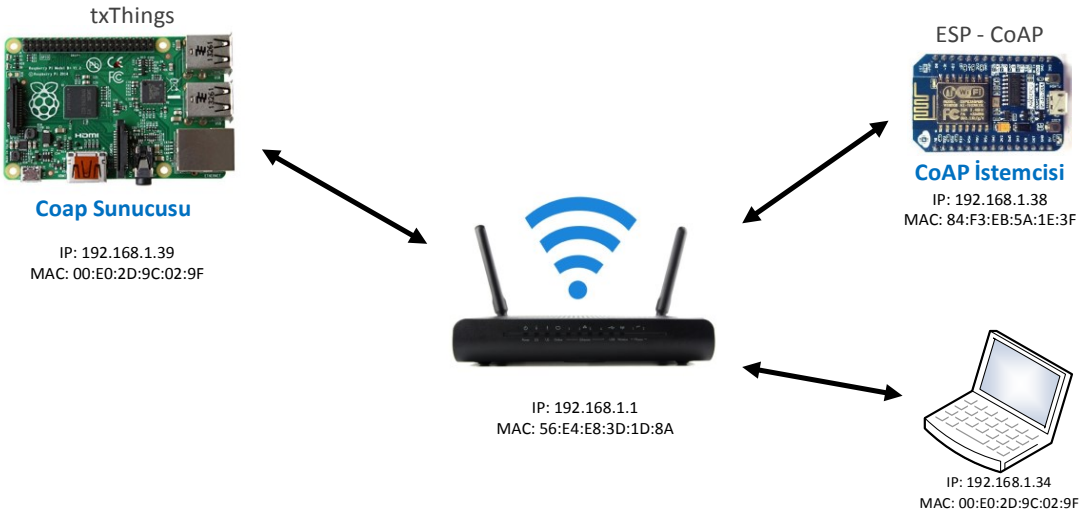
Tezin 8. bölümünde test edilen CoAP iletişim ortamına MITM saldırısı düzenlenmiştir. Basit bir CoAP iletişim ortamı Şekil 9.15’te verilmiştir. Buna göre 192.168.1.39 IP adresli CoAP sunucusu ve 192.168.1.38 IP adresli CoAP istemcisi 192.168.1.1 IP adresli ağ geçidine kablosuz (IEEE 802.11) bağlanmıştır. İstemci ve

sunucu cihazlar birbirleri ile CoAP iletişimi kurmaktadır.

Hazırlanan senaryoya göre Şekil 9.15'te verilen CoAP iletişim ortamındaki yerel ağa bağlanan 192.168.1.34 IP adresli saldırgan, istemcinin sunucu cihaza gönderdiği mesajları alabilmek için ARP zehirlenmesi saldırısı yaparak ortadaki adam rolünü almak istemektedir. Bu durum Şekil 9.16'da gösterilmiştir.



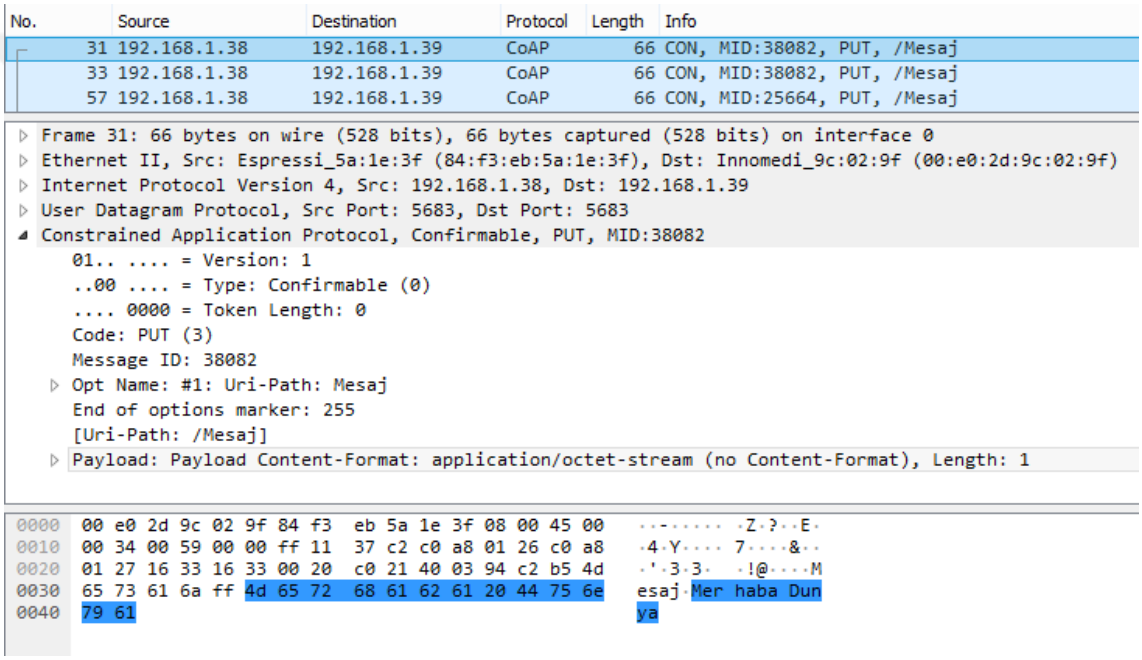
Şekil 9.15. CoAP iletişimi için kullanılan ortam



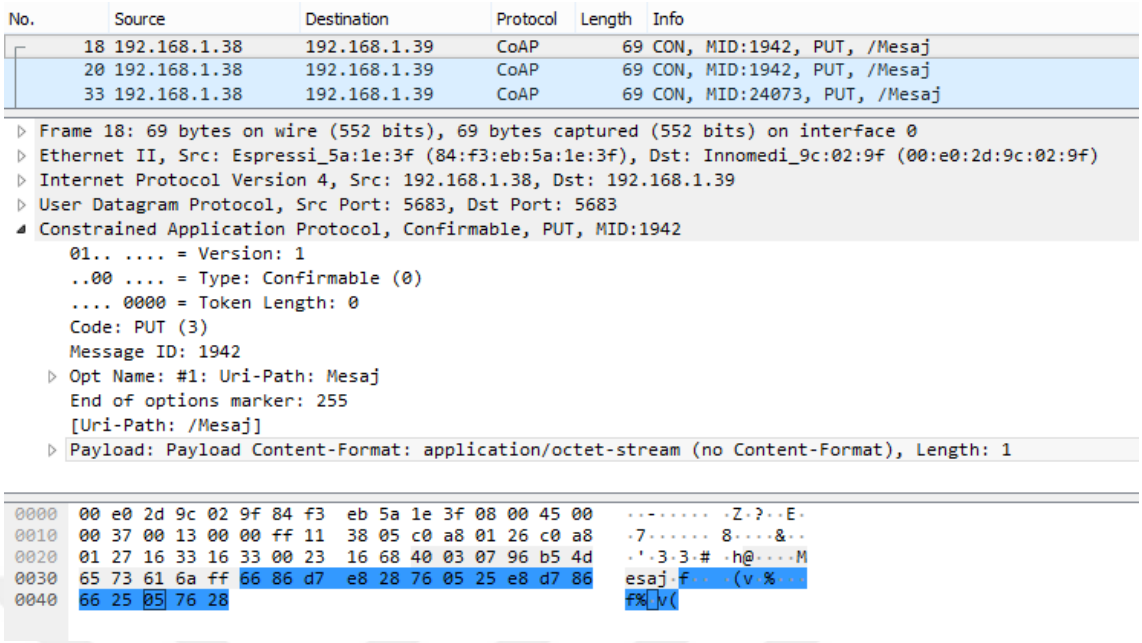
Şekil 9.16. CoAP iletişimi için MITM saldırısı

Saldırgan veri iletişimini ele geçirebilmek için ARP zehirlenmesi saldırısında hedef olarak CoAP istemcisini, ağ geçidi olarak ise CoAP sunucusunu seçmelidir. MQTT

MITM saldırısında olduğu gibi bu senaryoda da WebSploit kullanarak yapılan saldırıda aynı komutlar kullanılmıştır. İşlemler tamamlandıktan sonra saldırgan CoAP sunucusu ve istemcisi arasındaki ağ trafiğini kendi üzerinden geçirerek paketleri analiz edebilir. Saldırganın ele geçirdiği CoAP paketleri incelendiğinde şifreleme yapılmadan paketlerin gönderilmesi ile tüm paket içeriği kolaylıkla görülebilir. Şekil 9.17'te mesaj kaynağının (uri-path) "mesaj" ve içeriğin (payload) "Merhaba Dunya" olduğu gösterilmektedir. Öte yandan Şekil 9.18'de ise saldırganın yakaladığı şifrelenmiş CoAP paketi görülmektedir. Burada mesaj kaynağının "mesaj" olduğu belirli olmasına rağmen içerik şifrelendiği için anlaşılamaz durumdadır.



Şekil 9.17. Saldırganın yakaladığı şifrelenmemiş CoAP paketleri



Şekil 9.18. Saldırganın yakaladığı şifrelenmiş CoAP paketleri

9.8. Test Sonuçları

Bu bölümde MQTT ve CoAP güvenli iletişim ortamına MITM saldırısı düzenlenmiştir. Saldırgan WebSploit yazılımını kullanarak ağdaki trafiği kendi üzerinden geçirerek elde ettiği MQTT ve CoAP paketlerini Wireshark yazılımı ile analiz etmiştir. Saldırgan her ne kadar şifreleme yapılarak gönderilen paketler ele geçirmiş olsa da şifreleme algoritmasını ve anahtarı bilmediği için mesaj içeriğine ulaşamamaktadır.

BÖLÜM 10

10. SONUÇLAR VE TARTIŞMA

Nesnelerin interneti uygulamalarının iletişim güvenliğini artırabilmek amacıyla gerçekleştirilen bu tez çalışmasında öncelikli olarak nesnelerin interneti uygulamalarının kullanım alanları güncel verilere göre araştırılmıştır. Buna göre 2020 yılında geliştirilen uygulamaların sektörel dağılımı açısından ilk sıralarda endüstriyel üretim, taşımacılık, enerji, perakendecilik, akıllı şehirler, sağlık, tedarik zinciri, tarım ve hayvancılık, akıllı binalar bulunmaktadır.

Nesnelerin interneti uygulamaları için literatürde farklı mimari yaklaşımlar mevcuttur. Bu konuda henüz TCP/IP veya OSI gibi bir standart kabul görmemiştir. Bunun nedeni, uygulamaların farklı alanlarda gerçeklenmesi, kullanılan protokollerin ve teknolojilerinin birbirinden farklılık göstermesi olarak düşünülebilir. Yakın gelecekte bir standarttın oluşması da mümkün görülmemektedir. Fakat M2M uygulamaların artması ile farklı uygulamalarda kullanılan cihazların birbirleri ile veri iletişimi yapması açısından standardın oluşması zorunludur. Bu konuda çalışmalar yapılmaktadır. Tez çalışmasının 2. bölümünde literatürdeki mevcut mimariler incelenmiştir. En sık kullanılanı 4 katmanlı mimari olduğu belirlenmiştir. Fakat özellikle akıllı iş süreçleri, veri analitiği, büyük veri işleme gibi nesnelerin interneti uygulamalarının kritik aşamalarının artması ile 5 katmanlı mimariye geçişin yaygınlaşacağı düşünülmektedir. Mimari katmanların içerisindeki teknoloji ve protokoller tezin 3. bölümünde ele alınmıştır. Özellikle ağa bağlı cihaz sayısı, veri transfer hızı, gücü ve mesafesi düşünüldüğünde birçok farklı teknoloji ve protokol mevcuttur. Uygulama tipine bağlı olarak bu seçimin yapılması gereklidir. Örneğin akıllı tarım uygulamaları geniş araziye yayılması gerektiğinden Sigfox ve LoRaWAN gibi geniş alan ağları tercih edilmelidir. Akıllı ev sistemlerinde ise özellikle bu alan için geliştirilmiş Z-Wave veya ZigBee gibi protokoller seçilebilir. Öte yandan tüketici

elektroniği alanında genellikle Bluetooth LE teknolojisinin kullanıldığı görülmektedir. Nesnelerin interneti uygulamalarında ağa bağlı cihaz sayısı her geçen yıl üssel olarak artmaktadır. Günümüzde yaygın olarak kullanılan IPv4 ile bu sürecin yönetilmesi imkansızdır. Bu sebeple her cihazın IP adresi alabilmesi için IPv6 adresleme sistemine geçiş oldukça önemlidir. Bu konuda dünyada ilk sırada %54.41 uyumla Belçika yer almaktadır fakat tüm dünya düşünüldüğünde hızlı bir ilerleme kaydedilmemektedir. Cihazların tekil olarak adreslenememesi yönetim ve güvenlik açısından sıkıntılar doğurmaktadır.

Her ne kadar nesnelerin interneti uygulamaları insanlığın üretim ve yaşam tarzını radikal olarak değiştirse de güvenlik açısından bakıldığında ciddi sorunlara neden olmaktadır. Nesnelerin interneti uygulamalarındaki cihaz trafiğinin sadece %2'si şifrelenmiş bir şekilde iletilmektedir. Bu durum kişisel ve sektörel verilerin güvenliğini tehlikeye atmaktadır. Tezin 4. bölümünde birçok siber güvenlik ihlaline yer verilmiştir. Güvenlik uzmanları her yıl içerisinde binlerce nesnelerin interneti cihazlarından oluşan botnet ağları tespit etmektedir. Yakın gelecekte birçok saldırının bu ağlar üzerinden yapılacağı düşünülmektedir. Tez çalışmasında nesnelerin interneti uygulamalarına yönelik tehditler mimari katmanlara göre sınıflandırılmıştır. Buna göre cihazın ele geçirilmesi, algılama katmanı düzeyindeki saldırılar ile mümkün olurken iletişimin ve dolayısı ile verilerin ele geçirilmesi, ağ katmanına yapılan saldırılar ile gerçekleştirilmektedir. Veri güvenliği ağ katmanında sağlanmalıdır. Tezin 5. bölümünde nesnelerin interneti uygulamalarında ağ ve uygulama düzeyinde cihazların veri iletişimini sağlayan ve sıkça kullanılan MQTT ve CoAP protokolleri anlatılmıştır. Algılama katmanından gelen sensör verilerini sunucuya veya diğer cihazlara ileten bu protokoller güvenlik kaygısı güdülmeden veri iletim performansına bakılarak geliştirilmiştir. Çoğu uygulamada her ne kadar veri, sunucu düzeyinde SSL ve TLS gibi protokoller ile güvenli iletilse de sensörden sunucuya veri iletimi şifrelenmemiş ve güvensiz bir şekilde yapılmaktadır.

Nesnelerin interneti uygulamalarında kriptografik olarak genellikle AES algoritması kullanılmaktadır. Fakat uç cihazlar olarak nitelendirilen sensör cihazları sınırlı disk, bellek ve işlem kapasitesine sahiptir. AES ise geleneksel ağlardaki standart veri iletişimi için tasarlanmıştır. Bu doğrultuda küçük boyutlu, az enerji kullanan ve az yer kaplayan sistemler için daha etkin olan hafif sıklet kriptografik algoritmalar

kullanılmalıdır. Tez çalışması kapsamındaki test ortamlarındaki şifreleme ve şifre çözme işlemleri CLEFIA, PRESENT, PICCOLO, PRINCE ve LBLOCK algoritmaları ile yapılmıştır. Tezin 6. bölümünde bu algoritmaların içyapıları incelenmiştir. İncelenen algoritmalar yazılımsal ve donanımsal olarak gerçeklenmiş ve çalışma performansları tezin 7. bölümünde verilmiştir. Buna göre enerji ölçümleri Çizelge 7.1'de, güç ölçümleri Çizelge 7.2'de, akım ölçümleri Çizelge 7.3'te, tahmini pil ömürleri Çizelge 7.4'te, çalışma süresi ve harcanan enerjiye göre aktif mod oranları ise Çizelge 7.5'te verilmiştir. En az enerji harcayan, 64 bit blok uzunluğunu 80 bit anahtar ile şifreleyen LBLOCK, en fazla enerji harcayan ise 128 bit blok uzunluğunu 256 bit anahtar ile CLEFIA şifre çözme algoritmasıdır. Güç tüketimlerinde LBLOCK algoritması açık ara birinci olurken diğer algoritmalar benzer değerleri vermiştir. Akım ölçümünde yine LBLOCK birinci sırada yer almaktadır. Şifreleme ve şifre çözme işlemlerinde cihazın aktif mod süreleri göz önüne alındığında PICCOLO ve LBLOCK ilk sıralarda yer alırken PRINCE algoritması oldukça kötü sonuçlar vermiştir. Son olarak aktif mod enerji oranları incelendiğinde ise PICCOLO ve LBLOCK algoritmalarının yine ilk sıralarda yer aldığı görülmektedir. Algoritmaların döngü sayısı ve blok boyutunun enerji tüketimi, akım ölçümü, aktif mod çalışma süresi ve aktif mod enerji tüketimi açısından fark yarattığı söylenebilir. CLEFIA, 128-bit blok uzunluğu ile incelenen algoritmalar arasında en büyük blok uzunluğuna sahip olan şifreleme algoritmasıdır diğer algoritmalarda ise blok uzunluğu olarak 64-bit tercih edilmiştir. Bu durum doğası gereği düşük kapasiteli olan nesnelerin interneti uygulamalarında çalışan cihazlar için önemlidir. Ayrıca algoritmalardan iyi sonuçlar verenler Feistel mimarisini kullanmaktadır çünkü bu mimaride bloğun yarısı işleme alınır. Öte yandan anahtar boyutunun artması enerji verimliliğini düşürmektedir. Elbette anahtar boyutu ne kadar büyükse güvenlik o kadar iyi sağlanmaktadır. Fakat nesnelerin interneti uygulamalarında 80-bit ve 128-bit anahtarlar ideal olarak kabul edilebilir. Algoritmaların içerisindeki yapıların basit ve fazla enerji tüketmeyecek şekilde seçilmesi verimliliği artırmaktadır. CLEFIA ve PICCOLO algoritmalarının içerisinde kullanılan indirgeme işlemleri, karışık çarpmalar gibi enerji tüketen yapılar bu durumu göstermektedir. LBLOCK algoritmasının bu ölçümlerde birinci sırada gelmesinin sebebinin yapısında kullanılan XOR işlemi ve küçük boyutlu S-Kutularından kaynaklı olduğu söylenebilir.

Tezin 8. bölümünde kurulan güvenli iletişim platformu MQTT ve CoAP

protokolleri için ayrı ayrı hazırlanmıştır. İletilecek olan verinin güvenliği için Şekil 8.3 ve Şekil 8.5'te olduğu gibi şifreleme yapılarak iletişim sağlanmıştır. Hazırlanan sistem çeşitli senaryolara göre test edilerek ağ paketlerinin incelenmesiyle güvenli veri iletiminin yapıldığı gösterilmiştir. Tezin 9. bölümünde, hazırlanan güvenli iletişim platformuna günümüzde yerel ağlarda sıkça rastlanan ve neredeyse %100 oranında başarılı olan ortadaki adam saldırısı gerçekleştirilerek başarılı bir şekilde MQTT ve CoAP ağ paketleri yakalanmıştır. Şekil 9.14 ve Şekil 9.18 incelendiğinde her ne kadar saldırgan başarılı bir şekilde ağ paketlerini yakalasa da paketler şifreli olduğu için saldırganın verilere ulaşamadığı gösterilmiştir.

Nesnelerin interneti uygulamaları geleceği değiştirmekte ve her geçen gün farklı alanlardaki uygulamalar ile hayatımıza nüfuz etmektedir. Bu aşamada veri güvenliği oldukça önemlidir. Bu bağlamda geliştirilen platform ile cihazların birbiriyle olan iletişimi güvenli bir şekilde yapılabilir. Ayrıca incelen hafif sıklet kriptografik algoritmaların donanımsal davranışları sonucunda yapılan çıkarımlar dikkate alınarak yeni algoritmalar geliştirilebilir. Gelecekte, tez kapsamında sınıflandırılan diğer saldırılar geliştirilen platforma uygulanarak sistemin güvenliği artırılabilir.

KAYNAKLAR

Abrams, M., & Weiss, J. (2008). *Malicious control system cyber security attack case study–Maroochy Water Services, Australia*. Retrieved from https://www.mitre.org/publications/technical-papers/malicious-control-system-cyber-security-attack-case-study-maroochy-water-services-australia

Adelantado, F., Vilajosana, X., Tuset-Peiro, P., Martinez, B., Melia-Segui, J., & Watteyne, T. (2017). Understanding the limits of LoRaWAN. *IEEE Communications magazine, 55*(9), 34-40.

Ahmed, N., Kanhere, S. S., & Jha, S. (2005). The Holes Problem in Wireless Sensor Networks: A Survey. *ACM SIGMOBILE Mobile Computing and Communications Review, 9*(2), 4-18.

Al Dahoud, A., & Fezari, M. (2018). NodeMCU V3 For Fast IoT Application Development. In: Faculty of IT, Al-Zaytoonah University Amman.

Aslan, B. (2013). *Blok şifreler için cebirsel ikili doğrusal dönüşüm tasarımı ve modern bir blok şifreye uygulanması.* (Doktora Tezi), Trakya Üniversitesi,

Baker, G. (2008). Schoolboy hacks into city's tram system. *The Telegraph*. Retrieved from https://www.telegraph.co.uk/news/worldnews/1575293/Schoolboy-hacks-into-citys-tram-system.html

Banks, A., Briggs, E., Borgendale, K., & Gupta, R. (2019). MQTT Version 5.0. In *OASIS Standard* (Vol. 7).

Bawany, N. Z., Shamsi, J. A., & Salah, K. (2017). DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions. *Arabian Journal for Science and Engineering, 42*(2), 425-441.

Baysal, A., & Şahin, S. (2015). *Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors*. Paper presented at the International Workshop on Lightweight Cryptography for Security and Privacy.

Bélissent, J. (2010). *Getting Clever About Smart Cities: New Opportunities Require New Business Models*. Retrieved from Cambridge, Massachusetts, USA:

Beltov, M. (2016). New Linux Trojan NyaDrop targets IoT Devices. Retrieved from https://bestsecuritysearch.com/new-linux-trojan-nyadrop-targets-iot-devices/

Bhattasali, T., Chaki, R., & Sanyal, S. (2012). Sleep Deprivation Attack Detection in Wireless Sensor Network. *International Journal of Computer Applications, 40*(15), 19-25.

Biham, E., & Shamir, A. (1991). Differential cryptanalysis of DES-like cryptosystems. *Journal of CRYPTOLOGY, 4*(1), 3-72.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., . . . Vikkelsoe, C. (2007). *PRESENT: An ultra-lightweight block cipher*. Paper presented at the International Workshop on Cryptographic Hardware and Embedded Systems.

Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knezevic, M., Knudsen, L. R., . . . Rechberger, C. (2012). *PRINCE–a low-latency block cipher for pervasive computing applications.* Paper presented at the International Conference on the Theory and Application of Cryptology and Information Security.

Bormann, C., Castellani, A. P., & Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing, 16*(2), 62-67.

Bris, A. C., Chan Heng;Lanvin,Bruno. (2019). *Smart City Index 2019*. Retrieved from

BrittanyFinch, W. G. (2014). *MSP430™ Advanced Power Optimizations: ULP Advisor™ Software and EnergyTrace™ Technology*. Retrieved from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiulZ3PqvzqAhXEvosKHY5sDnkQFjAAegQIBBAB&url=https%3A%2F%2Fwww.ti.com%2Flit%2Fpdf%2Fslaa603&usg=AOvVaw0cT5g21ybgPOmMYL0RVBRC

Business, T. (2018). Shell Selects C3 IoT as Strategic AI Software Platform. *Telecoms Business*. Retrieved from https://telecomsbusiness.com/industries/retail/shell-selects-c3-iot-as-strategic-ai-software-platform/

Business Wire. (2017). Rockwell Automation CEO Discusses Future of Industrial Internet of Things *Business Wire*. Retrieved from https://www.businesswire.com/news/home/20170524005658/en/Rockwell-Automation-CEO-Discusses-Future-Industrial-Internet

Büyüktanır, T., & Özer, B. (2017). *Internet of Things*: Kodlab.

Cárdenas, A. A., Amin, S., & Sastry, S. (2008). *Research Challenges for the Security of Control Systems.* Paper presented at the Hot topics in security, San Jose, CA, USA.

Chen, F., Wang, N., German, R., & Dressler, F. (2010). Simulation study of IEEE 802.15. 4 LR-WPAN for industrial applications. *Wireless Communications and Mobile Computing, 10*(5), 609-621.

Cisco, V. (2018). *Cisco visual networking index: Forecast and trends, 2017–2022*. Retrieved from https://www.cisco.com/c/dam/m/en_us/network-intelligence/service-provider/digital-transformation/knowledge-network-webinars/pdfs/1213-business-services-ckn.pdf

CoAP RFC 7252. (2014). The constrained application protocol (CoAP). In *Internet Engineering Task Force (IETF)*.

Conti, M., Dragoni, N., & Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE Communications Surveys and Tutorials, 18*(3), 2027-2051.

Coskun, V., Ok, K., & Ozdenizci, B. (2012). *Near field communication (NFC): From theory to practice*: John Wiley & Sons.

Deering, S., & Hinden. (1998). Internet protocol, version 6 (IPv6) specification. In *Network Working Group*.

Diginomica. (2018). Walmart as "innovation company" - the CEO perspective on retail digital transformation. *Diginomica* Retrieved from https://diginomica.com/walmart-as-innovation-company-the-ceo-perspective

Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., & Biryukov, A. (2016). *Design Strategies for ARX with Provable Bounds: Sparx and LAX.* Paper presented at the ASIACRYPT 2016.

Dizdarević, J., Carpio, F., Jukan, A., & Masip-Bruin, X. J. A. C. S. (2019). A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. *ACM Computing Surveys (CSUR), 51*(6), 1-29.

Dobbins, R. (2016). Mirai iot botnet description and ddos attack mitigation. Retrieved from https://www.netscout.com/blog/asert/mirai-iot-botnet-description-and-ddos-attack-mitigation

Douceur, J. R. (2002). *The sybil attack.* Paper presented at the International Workshop on Peer-to-Peer Systems.

EC. (2013). *Report on the Public Consultation on IoT Governance*. Retrieved from https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjiuKDXqvzqAhW5kcMKHQAkC0QQFjAAegQIARAB&url=http%3A%2F%2Fec.europa.eu%2Finformation_society%2Fnewsroom%2Fcf%2Fdae%2Fdocument.cfm%3Fdoc_id%3D1746&usg=AOvVaw0ehod_keQfvpNtG5Ws0aqg

Engels, D., Saarinen, M.-J. O., Schweitzer, P., & Smith, E. M. (2011). *The Hummingbird-2 lightweight authenticated encryption algorithm.* Paper presented at the International Workshop on Radio Frequency Identification: Security and Privacy Issues.

Erdem, Ö. (2015). *HoneyThing: nesnelerin interneti için tuzak sistem.* (Yüksek Lisans Tezi), İstanbul Şehir Üniversitesi,

EU-IoT. (2009). *Internet of Things – An Action Plan for Europe*. Retrieved from Brusseles: https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2009:0278:FIN:EN:PDF

Farahani, S. (2011). *ZigBee wireless networks and transceivers*: Newnes.

Farooq, M., Waseem, M., Khairi, A., & Mazhar, S. (2015). A critical analysis on the security concerns of internet of things (IoT). *International Journal of Computer Applications, 111*(7), 1-6.

Farwell, J. P., & Rohozinski, R. (2011). Stuxnet and the future of cyber war. *Global Politics and Strategy, 53*(1), 23-40.

Fereidooni, H., Classen, J., Spink, T., Patras, P., Miettinen, M., Sadeghi, A.-R., . . . Conti, M. (2017). *Breaking fitness records without moving: Reverse engineering and spoofing fitbit.* Paper presented at the International Symposium on Research in Attacks, Intrusions, and Defenses.

FIBS PUB197. (2001). Advanced encryption standard (AES). In *Federal Information Processing Standards Publication* (Vol. 197, pp. 0311).

FIBS PUB 46-2. (1988). Data Encryption Standard (DES). In *Federal Information Processing Standards Publication*

Forbes. (2018). Microsoft Soars: How Azure, AI And IoT Are Driving Cloud Hypergrowth At $20-Billion Scale. *Forbes*. Retrieved from https://www.forbes.com/sites/bobevans1/2018/05/03/microsoft-soars-10-factors-driving-satya-nadellas-20-billion-cloud-juggernaut/#234d406a2037

Forouzan, B. A. (2009). *TCP/IP protocol suite 4 edition*: McGraw-Hill, Inc.

Fulare, P. S., & Chavhan, N. (2011). False data detection in wireless sensor network with secure communication. *International Journal of Smart Sensors and AdHoc Networks (IJSSAN), 1*(1), 66-71.

Gong, Z., Nikova, S., & Law, Y. W. (2011). *KLEIN: a new family of lightweight block ciphers.* Paper presented at the International Workshop on Radio Frequency Identification: Security and Privacy Issues.

Google. (2020). *Google IPV6 Adoption*. Retrieved from https://www.google.com/intl/en/ipv6/statistics.html

Greenberg, A. (2015). After jeep hack, chrysler recalls 1.4 m vehicles for bug fix. *Wired, Conde Nast*. Retrieved from https://www.wired.com/2015/07/jeep-hack-chrysler-recalls-1-4m-vehicles-bug-fix/

Greenberg, A. (2017). The Reaper Botnet Has Already Infected a Million Networks. *Wired, Conde Nast*. Retrieved from https://www.wired.com/story/reaper-iot-botnet-infected-million-networks/

Grosso, V., Leurent, G., Standaert, F.-X., & Varıcı, K. (2014). *LS-designs: Bitslice encryption for efficient masked software implementations.* Paper presented at the International Workshop on Fast Software Encryption.

Halon, E. (2019). Israeli experts expose security flaw in Amazon's Ring Doorbell. *Jerusalem Post*. Retrieved from https://www.jpost.com/jpost-tech/israeli-experts-expose-security-flaw-in-amazons-ring-doorbell-582282

Hillar, G. C. (2017). *MQTT Essentials-A lightweight IoT protocol*: Packt Publishing Ltd.

Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B.-S., . . . Jeong, K. (2006). *HIGHT: A new block cipher suitable for low-resource device.* Paper presented at the International Workshop on Cryptographic Hardware and Embedded Systems.

I-CIO. (2018). John Deere: How information-enabled farming will feed the world *I-CIO*. Retrieved from https://www.i-cio.com/innovation/internet-of-things/item/john-deere-how-information-enabled-farming-will-feed-the-world

Izadi, M., Sadeghiyan, B., Sadeghian, S. S., & Khanooki, H. A. (2009). *MIBS: a new lightweight block cipher.* Paper presented at the International Conference on Cryptology and Network Security.

Juels, A., & Weis, S. A. (2005). *Authenticating pervasive devices with human protocols.* Paper presented at the Annual International Cryptology Conference.

Karakoç, F., Demirci, H., & Harmancı, A. E. (2013). *ITUbee: a software oriented lightweight block cipher.* Paper presented at the International Workshop on Lightweight Cryptography for Security and Privacy.

Keszthelyi, A. (2014). *NEW CHALLENGES OF A NEW ERA.* Paper presented at the Managerial Challenges of the Contemporary Society. Proceedings.

Khoo, B. (2011). *RFID as an enabler of the internet of things: issues of security and privacy.* Paper presented at the Internet of Things (iThings/CPSCom), 2011 International Conference on and 4th International Conference on Cyber, Physical and Social Computing.

Kim, D., & Solomon, M. G. (2016). *Fundamentals of information systems security 3rd Edition* Jones & Bartlett Learning.

Kolay, S., & Mukhopadhyay, D. (2014). *Khudra: a new lightweight block cipher for FPGAs.* Paper presented at the International Conference on Security, Privacy, and Applied Cryptography Engineering.

Krebs, B. (2014). Target hackers broke in via HVAC company. Retrieved from https://krebsonsecurity.com/2014/02/target-hackers-broke-in-via-hvac-company/

Krebs, B. (2016). KrebsOnSecurity hit with record DDoS. Retrieved from https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

Leander, G., Paar, C., Poschmann, A., & Schramm, K. (2007). *New lightweight DES variants.* Paper presented at the International Workshop on Fast Software Encryption.

Li, L. (2012). *Study on security architecture in the internet of things.* Paper presented at the Measurement, Information and Control (MIC), 2012 International Conference on.

Lim, C. H., & Korkishko, T. (2005). *mCrypton–a lightweight block cipher for security of low-cost RFID tags and sensors.* Paper presented at the International Workshop on Information Security Applications.
Matsui, M. (1993). *Linear cryptanalysis method for DES cipher.* Paper presented at the Workshop on the Theory and Application of of Cryptographic Techniques.
Miller, B. A., Bisdikian, C., & Foreword By-Siep, T. (2000). *Bluetooth revealed*: Prentice Hall.
Millman, R. (2017). Hajime malware now has 300,000 strong botnet at disposal say researchers. Retrieved from https://www.scmagazineuk.com/hajime-malware-300000-strong-botnet-disposal-say-researchers/article/1474756
Minerva, R., Biru, A., & Rotondi, D. (2015). Towards a definition of the Internet of Things (IoT). *IEEE Internet Initiative, 1*(1), 1-86.
Mitrokotsa, A., Rieback, M. R., & Tanenbaum, A. S. (2010). Classification of RFID attacks. *Gen, 15693*, 73-86.
Moradi, A., Poschmann, A., Ling, S., Paar, C., & Wang, H. (2011). *Pushing the limits: a very compact and a threshold implementation of AES.* Paper presented at the Annual International Conference on the Theory and Applications of Cryptographic Techniques.
Mouha, N., Mennink, B., Van Herrewege, A., Watanabe, D., Preneel, B., & Verbauwhede, I. (2014). *Chaskey: an efficient MAC algorithm for 32-bit microcontrollers.* Paper presented at the International Workshop on Selected Areas in Cryptography.
Mulligan, G. (2007). *The 6LoWPAN architecture.* Paper presented at the Proceedings of the 4th workshop on Embedded networked sensors, Cork, Ireland.
Najera-Gutierrez, G., & Ansari, J. A. (2018). *Web Penetration Testing with Kali Linux: Explore the methods and tools of ethical hacking with Kali Linux*: Packt Publishing Ltd.
Niland, M. (2003). Virus Disrupts Train Signals. *CBS News*. Retrieved from https://www.cbsnews.com/news/virus-disrupts-train-signals/
Oliveira, G. M., Costa, D. C., Cavalcanti, R. J., Oliveira, J. P., Silva, D. R., Nogueira, M. B., & Rodrigues, M. C. (2018). *Comparison between MQTT and WebSocket protocols for Iot applications using ESP8266.* Paper presented at the 2018 Workshop on Metrology for Industry 4.0 and IoT.
Orebaugh, A., Ramirez, G., & Beale, J. (2006). *Wireshark & Ethereal network protocol analyzer toolkit*: Elsevier.
Owen, M. (2020). Philips Hue smart bulb allows hackers to attack your network. Retrieved from https://appleinsider.com/articles/20/02/05/philips-hue-smart-bulb-allows-hackers-to-attack-your-network
Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011). Cloud computing: security issues and research challenges. *International Journal of Computer Science and Information Technology & Security (IJCSITS), 1*(2), 136-146.
Padmavathi, G., & Shanmugapriya, D. (2009). A survey of attacks, security mechanisms and challenges in wireless sensor networks. *International Journal of Computer Science and Information Security, 4*(1), 1-9.
Paetz, C. (2018). *Z-Wave Essentials*: CreateSpace Independent Publishing Platform.
Perlroth, N., & Krauss, C. (2018). A cyberattack in Saudi Arabia had a deadly goal. Experts fear another try. *New York Times*. Retrieved from

https://www.nytimes.com/2018/03/15/technology/saudi-arabia-hacks-cyberattacks.html
Pinto, A. D. (2020). Dark Nexus IoT Botnet: Analyzing and Detecting its Network Activity. Retrieved from https://www.nozominetworks.com/blog/dark-nexus-iot-botnet-analyzing-and-detecting-its-network-activity/
Prayogo, S. S., Mukhlis, Y., & Yakti, B. K. (2019). *The Use and Performance of MQTT and CoAP as Internet of Things Application Protocol using NodeMCU ESP8266.* Paper presented at the 2019 Fourth International Conference on Informatics and Computing (ICIC).
Presser, M. (2016). The rise of IoT–why today. *IEEE Internet of Things*. Retrieved from https://iot.ieee.org/newsletter/january-2016/the-rise-of-iot-why-today.html
Roberts, P. F. (2008). Zotob, PnP worms slam 13 DaimlerChrysler plants. Retrieved from https://www.eweek.com/security/zotob-pnp-worms-slam-13-daimlerchrysler-plants
Scully, P. (2020). Top 10 IoT applications in 2020. *IoT Analytics: Market Insights for the Internet of Things*. Retrieved from https://iot-analytics.com/top-10-iot-applications-in-2020/
SecureMac. (2020). Kaiji malware: a new IoT threat. Retrieved from https://www.securemac.com/news/kaiji-malware-a-new-iot-threat
Sethi, P., & Sarangi, S. R. (2017). Internet of things: architectures, protocols, and applications. *Journal of Electrical and Computer Engineering, 2017*, 1-25.
Shelby, Z., & Bormann, C. (2011). *6LoWPAN: The wireless embedded Internet* (Vol. 43): John Wiley & Sons.
Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T., & Shirai, T. (2011). *Piccolo: an ultra-lightweight blockcipher.* Paper presented at the International Workshop on Cryptographic Hardware and Embedded Systems.
Shirai, T., Shibutani, K., Akishita, T., Moriai, S., & Iwata, T. (2007). *The 128-bit blockcipher CLEFIA*. Paper presented at the International Workshop on Fast Software Encryption.
Suzaki, T., Minematsu, K., Morioka, S., & Kobayashi, E. (2011). *Twine: A lightweight, versatile block cipher.* Paper presented at the ECRYPT Workshop on Lightweight Cryptography.
Tanganelli, G., Vallati, C., & Mingozzi, E. (2015). *CoAPthon: Easy development of CoAP-based IoT applications with Python.* Paper presented at the 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT).
TI. (2019). MSP-EXP430FR5994 LaunchPad Development Kit. In T. Instruments (Ed.), *User's Guide*.
TI. (2020a). Code Composer Studio™ IDE v10.x forMSP430™ MCUs. In T. Instruments (Ed.), *User's Guide*.
TI. (2020b). MSP Debuggers. In T. Instruments (Ed.), *User's Guide*.
TrendLabs. (2019). Silex Malware Bricks IoT Devices with Weak Passwords. Retrieved from https://www.trendmicro.com/vinfo/hk-en/security/news/cybercrime-and-digital-threats/-silex-malware-bricks-iot-devices-with-weak-passwords
TU Automotive. (2019). Volvo Claims 1M Connected Car Milestone. *TU Automotive*. Retrieved from https://www.tu-auto.com/volvo-claims-1m-connected-car-milestone/
Unit42. (2020). *2020 Unit 42 IoT Threat Report*. Retrieved from
Upton, E., & Halfacree, G. (2014). *Raspberry Pi user guide*: John Wiley & Sons.

Uttarkar, M. R., & Kulkarni, R. (2014). Internet of Things: Architecture and Security. *International Journal of Computer Applications, 3*(4), 12-19.

Vejlgaard, B., Lauridsen, M., Nguyen, H., Kovács, I. Z., Mogensen, P., & Sorensen, M. (2017). *Coverage and capacity analysis of sigfox, lora, gprs, and nb-iot.* Paper presented at the 2017 IEEE 85th vehicular technology conference (VTC Spring).

Want, R. (2011). Near field communication. *IEEE Pervasive Computing, 10*(3), 4-7.

World Medical Innovation Forum. (2020). Calibrating Innovation Opportunity and Urgency. Retrieved from https://www.youtube.com/watch?v=VXyvxIMQi0c

Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., & Du, H.-Y. (2010). *Research on the architecture of Internet of Things.* Paper presented at the 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE).

Wu, W., & Zhang, L. (2011). *LBlock: a lightweight block cipher.* Paper presented at the International Conference on Applied Cryptography and Network Security.

Xiao, C., & Zheng, C. (2017). New IoT/Linux Malware Targets DVRs, Forms Botnet. Retrieved from https://unit42.paloaltonetworks.com/unit42-new-iotlinux-malware-targets-dvrs-forms-botnet/

Yavuzer Aslan, F., Sakallı, M. T., & Aslan, B. (2012, 1 - 3 Şubat 2012). *Önemli Blok Şifrelerde Kullanılan Doğrusal Dönüşümlerin İncelenmesi.* Paper presented at the Akademik Bilişim'12, Uşak Üniversitesi.

Yeh, T., Chiu, D., & Lu, K. (2017). Persirai: New internet of things (IoT) botnet targets IP cameras. Retrieved from https://blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras/

Yetimler, E. (2017). Internet of Things (Nesnelerin İnterneti) Nedir? Cihazların Etkileşim Trendleri. Retrieved from http://www.karel.com.tr/blog/internet-things-nesnelerin-interneti-nedir-cihazlarin-etkilesim-trendleri

ÖZGEÇMİŞ

Adı Soyadı : Füsun YAVUZER ASLAN

Doğum Yeri : Zonguldak

Doğum Tarihi : 07.10.1979

E-posta : fusunyavuzer@klu.edu.tr

Öğrenim Durumu

Lise : Zonguldak Mehmet Çelikel Lisesi

Lisans : Fizik Bölümü, Trakya Üniversitesi

Yüksek Lisans : Bilgisayar Mühendisliği, Trakya Üniversitesi

Yüksek Lisans Tez Başlığı ve Danışmanı

Blok Şifrelerde Kullanılan Doğrusal Dönüşüm Yapılarının İncelenmesi, Prof. Dr. M. Tolga SAKALLI

Mesleki Deneyim

Öğretim Görevlisi, Kırklareli Üniversitesi Lüleburgaz Meslek Yüksekokulu Bilgisayar Programcılığı Programı, 2014

TEZ İLE İLGİLİ BİLİMSEL FAALİYETLER

ASLAN Y.F., SAKALLI M.T., ASLAN B. (2019). Nesnelerin İnterneti Uygulamalarında Kullanılan Protokollerin İncelenmesi, Trakya Üniversiteler Birliği 4. Lisansüstü Öğrenci Kongresi, TUBLOK 2019, Kırklareli, Türkiye.

ASLAN B., ASLAN Y.F. (2018). IoT and Cyber Security, Istanbul Bosphorus International Conference On Cyberpolitics and Cybersecurity, İstanbul, Türkiye.