ENDÜSTRİYEL NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ

Gökhan MUTLU

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ENDÜSTRİYEL NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ

Gökhan MUTLU
(https://orcid.org/0000-0002-7906-5637)

Dr. Öğr. Üyesi Cengiz TOĞAY
(https://orcid.org/0000-0001-5739-1784)
(Danışman)

YÜKSEK LİSANS TEZİ

ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2019

TEZ ONAYI

Gökhan MUTLU tarafından hazırlanan "ENDÜSTRİYEL NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ" adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Cengiz TOĞAY
(https://orcid.org/0000-0001-5739-1784)

Başkan : Dr. Öğr. Üyesi Cengiz TOĞAY
(https://orcid.org/0000-0001-5739-1784)
Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Anabilim Dalı
İmza

Üye : Doç. Dr. Ersen YILMAZ
(https://orcid.org/0000-0002-6620-655X)
Bursa Uludağ Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendisliği Anabilim Dalı
İmza

Üye : Doç. Dr. Cemal HANİLÇİ
(https://orcid.org/0000-0002-9174-0367)
Bursa Teknik Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi, Elektrik-Elektronik Mühendisliği Anabilim Dalı
İmza

Yukarıdaki sonucu onaylarım

Prof. Dr. Hüseyin Aksel EREN
Enstitü Müdürü
../...../.....

U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

17/09/2019

Gökhan MUTLU

ÖZET

Yüksek Lisans Tezi

ENDÜSTRİYEL NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ

Gökhan MUTLU

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Elektronik Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Cengiz TOĞAY

Nesnelerin interneti cihazları, endüstriyel gömülü sistemler, otomobilller, akıllı ev aygıtları, endüstriyel kablolu ya da kablosuz sensörler gibi birbirine bağlı cihazlardan meydana gelmektedir. İnternete bağlanma imkanı olmayan cihazlar, ağ geçitleri sayesinde nesnelerin interneti sisteminin bir parçası olabilirler. Cihazların internete bağlı hale gelmesi ile sistemlerin etkinlikleri artıyor olsa da aynı derecede güvenlik riskleri de artmaktadır. Buna göre, daha önce erişilemedikleri için saldırganların hedefi olmayan cihazlar, artık doğrudan DDoS ataklarının hedefi haline gelmekte ve ağ içerisindeki bu cihazlar aracılığı ile sistemlere zarar verilebilmektedir. Cihazların sisteme bağlanırken, kimliklerinin doğrulanması ve cihazların sistemdeki diğer cihazlar ile güvenli bir şekilde iletişim kurması bu tezin temel amaçlarıdır. Nesnelerin interneti sistemlerinde aracı sunucular (Broker), verilerin anonimleşmesi, protokol (MQTT, AMQP ve COAP) bağımsız sistemlerin kurulabilmesi ve yüksek performans ihtiyaçları nedeni ile sıklıkla tercih edilmektedir. İletişim güvenliğinde her ne kadar TLS protokolü kullanılabiliyor olsa da kısıtlı cihazların, asimetrik şifreleme algoritmalarının ihtiyacı olan açık anahtarları ve algoritma kodlarını kalıcı bellekte saklaması problem oluşturabilmektedir. Güvenli iletişim ve kimlik doğrulamada, TÜBİTAK TEYDEB 1505 projesi kapsamında değiştirilen bir broker ile birlikte güvenli anahtar depolama, gerçek rastgele sayı üreteci ve 128-bit AES şifreleme algoritması gibi özelliklere sahip olan ATAES132A modülünü baz alan bir yaklaşım kullanılmıştır. Çalışma kapsamında, bir test kartının (ARM Cortex-M3 içerir) yanı sıra geliştirilen yeni bir ağ geçidi (ARM Cortex-M0 içerir) kartı test ve uygulama çalışmalarında kullanılmıştır. Tez çalışmasında ayrıca ARM Cortex-M3'ün fiziksel I^2C özelliğinin kullanımıyla, eş zamanlı şifreleme ve MQTT tabanlı iletişim sağlayan yeni bir yöntem önerilmiştir. Son olarak iletişimde mesajların bütünlüğünün denetlenmesi için kriptografik karma (MD5, SHA-1 ve SHA-2 gibi) veya Döngüsel Artıklık Denetimi (CRC32/64) algoritmalarının performans açısından değerlendirilmesi yapılmıştır.

Anahtar Kelimeler: Endüstriyel Nesnelerin İnterneti, Gömülü Sistem, Ağ Geçidi, Güvenli İletişim

2019, vii + 43 sayfa.

ABSTRACT

MSc Thesis

SECURE GATEWAY FOR INDUSTRIAL INTERNET OF THINGS

Gökhan MUTLU

Bursa Uludağ University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronic Engineering

Supervisor: Dr. Cengiz TOĞAY

Internet of Things devices consists of interconnected devices such as industrial embedded systems, automobiles, smart home devices, industrial wired, or wireless sensors. The devices that cannot connect to the Internet can be part of the Internet of things system through gateways. When the devices connected to the Internet, the efficiency of the systems increases, but also the security risks increase. Previously unreachable devices are now becoming the target of direct DDoS attacks and can be utilized to damage systems within the network. The main objectives of this thesis are to verify the identity of the devices when connecting to the system and to ensure that devices communicate securely with other devices in the system. A broker is often preferred in the Internet of Things systems due to the anonymization of data, the establishment of protocol-independent systems (MQTT, AMQP, and COAP) and high-performance requirements. Although TLS protocol can be used in communication security, it can be problematic that limited devices store public keys and algorithm codes required by asymmetric encryption algorithms in non-volatile memory. In the secure communication and authentication, an approach based on ATAES132A module which has features such as secure key storage, real random number generator, and 128-bit AES encryption algorithm was used with a broker we changed within the scope of a TÜBİTAK TEYDEB 1505 project. In the scope of the study, a test card (includes ARM Cortex-M3) as well as a new gateway card (which includes ARM Cortex-M0) was used in test and application studies. In this thesis, a new method which provides simultaneous encryption and MQTT based communication is proposed by using the physical I^2C feature of ARM Cortex-M3. Finally, cryptographic hash (such as MD5, SHA-1 and SHA-2) or Cyclic Redundancy Check (CRC32 / 64) algorithms are evaluated for performance in order to check the integrity of messages in communication.

Key words: Industrial Internet of Things, Embedded System, Gateway, Secure Communication

2019, vii + 43 pages.

TEŞEKKÜR

Bu tezin yürütülmesi sırasında sabır göstererek desteğini eksik etmeyen danışmanım Dr. Öğr. Üyesi Cengiz Toğay'a, verdiği ümit ve motivasyondan dolayı sevgili aileme, sağladığı çalışma ortamı ve manevi destek nedeniyle EMKO Elektronik'e, bursiyer olarak görev aldığım 5170033 nolu TÜBİTAK 1505 projemizdeki desteklerinden dolayı TÜBİTAK'a ve çalışmam sırasında yardımı bulunan herkese teşekkürlerimi sunarım.

Gökhan MUTLU
17/09/2019

İÇİNDEKİLER

Sayfa

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	**Açıklama**
K_n	n. Anahtar Dizini
keyId_n	n. Anahtarın Kodu
IV	Başlangıç Vektörü
SK	Oturum Anahtarı
C_a	Şifreli Dizin

Kısaltmalar	**Açıklama**
IoT	Nesnelerin İnterneti
IIoT	Endüstriyel Nesnelerin İnterneti
TCP/IP	İletim Denetimi Protokolü / Internet Protokolü
QoS	Servis Kalitesi Hizmeti
DDoS	Dağıtılmış Hizmet Reddi
AES	Gelişmiş Şifreleme Standardı
PRNG	Sözde Rastgele Sayı Üreteci
UDP	Kullanıcı Datagram Protokolü
ARP	Adres Çözümleme Protokolü
IPv4	İnternet Protokolü Versiyon 4
ICMP	İnternet Kontrol Mesajı Protokolü
IGMP	İnternet Grubu Yönetimi Protokolü
PPPoE	Ethernet Üzerinden Noktadan Noktaya Protokolü
ROM	Sadece Okunabilir Bellek
UART	Evrensel Asenkron Alıcı / Verici
PHY	Fiziksel Katman
OSI	Açık Sistem Arabağlantısı
PLC	Programlanabilir Mantık Denetleyici
RF	Radyo Frekansı
LSB	En Az Önemli Bayt
MSB	En Çok Önemli Bayt
EEPROM	Elektronik Olarak Silinebilir Programlanabilir Salt Okunur Bellek
FIPS	Federal Bilgi İşleme Standartları

ŞEKİLLER DİZİNİ

Sayfa

ÇİZELGELER DİZİNİ

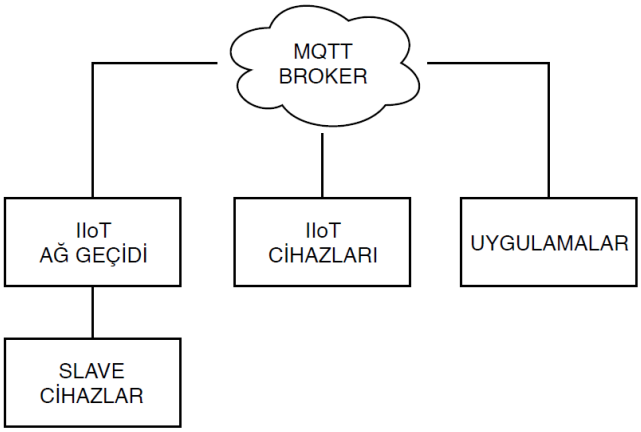
Sayfa

1. GİRİŞ

Günümüzde akıllı tarım, akıllı şehir, akıllı ulaşım, akıllı şebeke ve endüstriyel çözümlerde Nesnelerin İnterneti (IoT) cihazları yaygın olarak kullanılmaktadır. IoT sistemleri; sensör, aktüatör, PLC gibi çok çeşitli cihazlardan oluşmaktadır. Endüstriyel Nesnelerin İnterneti (IIoT) ise IoT olarak adlandırılan yapının bir alt kümesidir. IoT, herşeyi kapsarken IIoT sadece endüstriyel sistemleri kapsamaktadır. Bu nedenle IIoT, lojistik, havacılık, sağlık, enerji üretimi, petrol ve gaz üretimi gibi pek çok endüstriyel alanda gelişmeye açıktır. Temel olarak M2M (Makine-Makine) teknolojisinin yapısına dayanmaktadır ve M2M'nin bir evrimi olarak görülmektedir (Gilchrist 2016). Günümüzde M2M yapısına sahip endüstriyel sistemler zaten sensör, aktüatör ve PLC gibi bileşenler ile iletişim kurabilmektedirler. IIoT yaklaşımındaki temel fark, endüstriyel sistemlerin internete bağlanarak daha fazla iş akışı ve analizi amacıyla yönetilmelerini sağlamaktır.

IIoT gibi birden çok cihazın yönetildiği sistemlerde, haberleşme önemli bir unsurdur. Haberleşmenin işlevi; tüm sensör, aktüatör, PLC gibi IIoT cihazlarının birbirleri ile etkin ve güvenli bir şekilde iletişim kurabilmesini sağlamaktır. Ancak tek bir homojen sistem uygulamak her zaman mümkün değildir. Hali hazırda kurulu olan endüstriyel sistemlerde çeşitli haberleşme protokollerinin aynı anda kullanılabilmesine ihtiyaç olabilmektedir. Bu durumda haberleşme şekillerinin veya protokollerin çevrilmesi için ağ geçitlerinin kullanılması gereklidir. Örneğin, bir endüstriyel işletmede Modbus, Canbus, Profibus gibi bir dizi protokol kullanan yüzlerce cihaz olabilir. Bu cihazları Message Queuing Telemetry Transport (MQTT) (Banks ve Gupta 2014) protokolü ile haberleşen bir IIoT sistemine dahil etmek için ağ geçitlerine ihtiyaç bulunmaktadır. Endüstriyel cihazlar genellikle RS-232 ya da RS-485 gibi seri iletişim arayüzleri üzerinden haberleşmektedir. Ağ geçitleri, seri iletişim arayüzlerine bağlı cihazlar ile sunucu uygulamaları ya da MQTT gibi bir protokolle iletişim kuran cihazlar arasında veri alışverişine olanak sağlayabilmektedir. İnternet erişimi olmayan fiziksel cihazlar da, Şekil 1.1'de gösterildiği gibi ağ geçitleri sayesinde IIoT sisteminin bir parçası olabilir. Ağ geçitleri, mevcut cihazların protokol ve arayüz gereksinimlerinden oluşacak karmaşıklığı giderecek ve özellikle kısıtlı cihazların internete açılmalarını sağlayacaktır. Ayrıca ağ geçitleri,

özellikle tezde hedeflenen veri birleştirme, kimlik doğrulama ve güvenli veri iletişimi gibi işlevlerden sorumlu olabilmektedirler.



Şekil 1.1. MQTT protokolünü kullanan bir IIoT sistemi

Endüstriyel haberleşme sistemlerinde hem veriler hem de komutlar çok değerlidir. IIoT cihazları, ağı dinleme veya ağ trafiğini engelleme gibi çeşitli siber saldırılara karşı hazırlıklı olmalıdır. Bir saldırgan veri veya komutları değiştirdiğinde, üretim süreçleri etkilenebilir veya bazı tehlikeli durumlar oluşabilir. IIoT cihazları, bellek, bilgi işlem performansı, enerji tüketimi ve ağ bağlantısının güvenilirliği gibi bazı kısıtlamalara bağlı olarak ayrılabilirler. Bu cihazlara uygulanabilecek güvenlik yaklaşımları, bu sınırlamalardan dolayı kısıtlıdır. İşlem performansı yeterince yüksek değilse, küçük kod boyutuna sahip hafif bir güvenlik protokolü kullanmak gerekir (Andy ve ark. 2017). Yukarıda bahsedilen çeşitli gerekçelerle güvenlik açısından zayıf olan IIoT cihazlarının, güvenli bir ağ geçidi kullanarak internet ortamında daha güvenli bir şekilde iletişim kurmaları sağlanabilir.

Tezde özellikle kısıtlı cihazların bulunduğu ortamlarda sunucu performansını da gözeterek, kendisine bağlı cihazlardan veri toplayan, bu verileri AES gibi standart ve güvenli bir protokol aracılığı ile sunuculara ileten, benzer şekilde sunucudan güvenli bir şekilde gelen bilgi/komut ve verileri ilgili cihazlara ileten bir ağ geçidi tasarlanmıştır. Ağ geçidinin tasarımı sırasında, TEYDEB 1505 projesi kapsamında geliştirilmiş olan bir Broker sunucusu ile paylaşımlı anahtar yöntemine bağlı olarak oturum anahtarının

oluşturulması ve kimlik doğrulamanın gerçekleştirilmesine yönelik yeni protokol çalışmalarına yer verilmiştir. Geliştirilen yöntem ile MQTT protokolünde bir değişiklik yapılmadan şifre alanında normal bir şifre yerine bir dizi bilgi gönderimi gerçekleştirilmektedir. Böylece, her bağlantıda değişen kimlik doğrulama bileşenleri ile sunucuda kimlik doğrulaması sağlanırken aynı zamanda şifreli iletişim için gerekli olan simetrik anahtarın oluşması sağlanmıştır. Ağ geçidinde, anahtarların güvenli bir şekilde cihazda saklanmasına yönelik olarak ATAES132A entegresi kullanılmıştır. İletişim performansı ve güvenlik konusundaki çalışmaları gerçekleştirmek üzere, proje ortağı EMKO Elektronik ile bir ağ geçidi cihazı (ARM Cortex-M0) geliştirilmiş ve farklı işlemci mimarisine (ARM Cortex-M3) sahip bir başka geliştirme kartı kullanılmıştır. Sunulan çalışma ile ağ geçidinin bulunduğu ağın bant genişliği, ağ geçidinin işlemci yükü, cihazların ürettiği veri miktarı ve ağ geçidinin kullandığı işlemci modeline bağlı olarak farklı senaryolara yönelik çözümlerin oluşturulmasına olanak sağlayacak sonuçlara ulaşılmıştır.

Tez çalışmasında ayrıca ARM Cortex-M3'ün fiziksel I^2C özelliğinin kullanımıyla, eş zamanlı şifreleme ve MQTT tabanlı iletişim sağlayan yeni bir yöntem önerilmiştir. Bu yöntem ile sunucuya veri gönderiminin yavaş olduğu ortamlarda, Slave cihazlardan gelen verilerin ağ geçidinde veri kaybı olmaksızın toplanması ve gönderilmek üzere MQTT'nin gönderim kuyruğuna yazılması işlemi gerçekleşirken aynı zamanda fiziksel I^2C özelliği ile kuyruktaki önceki verilerin şifrelenmesi sağlanmıştır.

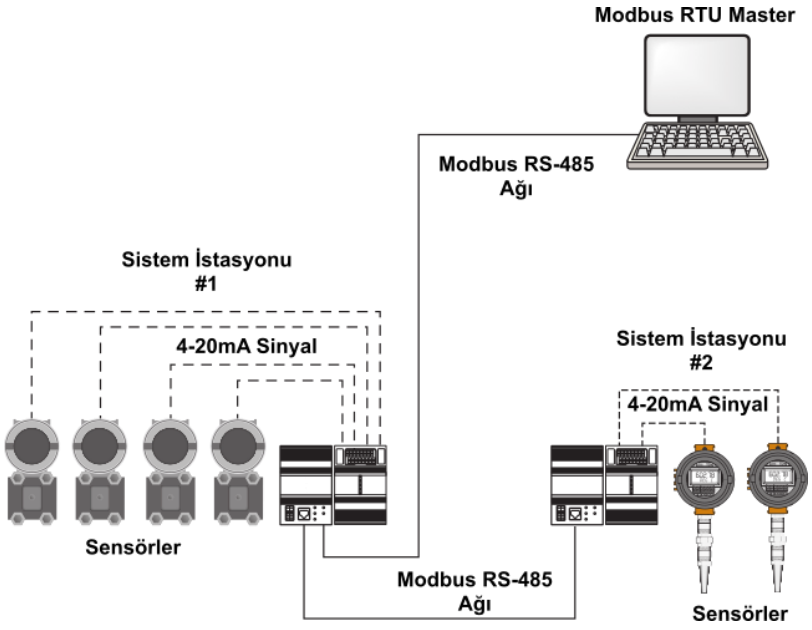
Son olarak iletişimde mesajların bütünlüğünün denetlenmesi için kriptografik karma (MD5, SHA-1 ve SHA-2 gibi) veya Döngüsel Artıklık Denetimi (CRC32/64) algoritmalarının performans açısından değerlendirilmesi yapılmıştır. Algoritmalar verilerin alıcıya ulaşması sonrasında ağda herhangi bir değişikliğe uğrayıp uğramadıklarının tespiti için kullanılmaktadır. Hangi algoritmanın tercih edileceği kullanılan mikrodenetleyiciye, bellek miktarına, performans ve güvenlik gereksinimlerine göre belirlenebilir. Bu kapsamda performansa yönelik yapılan test sonuçları tezde sunulmuştur.

Tezde yer verilen temel kavramlardan Modbus, MQTT protokolü ve AES şifreleme algoritmasına ait bilgilere Bölüm 2'de yer verilmiştir. Bölüm 3'de tezde kullanılan materyaller olan geliştirilen yeni ağ geçidi kartı, farklı mikrodenetleyiciye sahip olan bir geliştirme kartı, anahtarların saklanması için ATAES132A entegresi ve ağ geçidinde iletişim için geliştirilen yazılım bileşenlerine yer verilmiştir. Tezde elde edilen bulgular Bölüm 4'de ifade edilmiştir. Son olarak Bölüm 5'de sonuçlar değerlendirilmiştir.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

2.1. Modbus Haberleşme Protokolü

Modbus, (Anonim 1996) OSI modelinin yedinci seviyesi olan uygulama katmanına yerleştirilmiş bir haberleşme protokolüdür. Master-Slave yapısıyla çalışır ve fonksiyon kodlarıyla belirtilen işlevleri yürütür (Anonim 2005). 1979 yılında, Modicon (Schneider Electric) firması tarafından, birden çok cihazın iletişim kurması için geliştirilmiştir (McConahay 2011). İlk olarak RS-232 arayüzü üzerinde çalıştırılmış, ancak haberleşme hızının arttırılması, ağın genişletilmesi gibi ihtiyaçlardan dolayı, RS-485'te çalışacak şekilde uyarlanmıştır. Kullanımının, hızlı bir şekilde artmasıyla endüstriyel bir haberleşme standardı haline gelmiştir. Modbus temelde, bir Master cihazın bir veya birden fazla Slave cihaz ile iletişim kurmasına olanak sağlayan bir sistemdir (Şekil 2.1). Master ve Slave cihazlar RS-232 veya RS-485 gibi seri haberleşme ara yüzlerine sahip olan uç birimleri ifade etmektedir.



Şekil 2.1. Örnek bir Modbus ağı (McConahay 2011)

Günümüzde, yaygın olarak ASCII, RTU ve TCP şeklindeki üç tip Modbus modeli kullanılmaktadır. Tüm Modbus paketleri Şekil 2.2'de gösterildiği gibi aynı formatta iletilir. Bu üç tür arasındaki tek fark, mesajların kodlanma şeklidir.

MODBUS			
Slave Cihaz Kodu	Fonksiyon Kodu	Veri	Hata Kontrolü Değeri

Şekil 2.2. Genel Modbus paketi

Modbus ASCII'de, tüm mesajlar 4-bit ASCII karakterleri kullanılarak, on altılık sayı biçiminde kodlanmıştır. Çoğunlukla telefon, modem veya RF bağlantılarında tercih edilir. Modbus ASCII bir mesajı sınırlamak için belli bir özel karakter kullanır. Bu şekilde, iletim ortamındaki herhangi bir gecikmenin, alıcı cihaz tarafından yanlış değerlendirilmesi engellenir. Modbus ASCII, bu üç protokolden en yavaş olanıdır.

Modbus RTU'da veriler ikili olarak kodlanır. 1200 bps ile 115200 bps arasındaki hızlarda RS-232 veya RS-485 ağları üzerinden kullanım için idealdir. Yaygın olarak 9600 bps ve 19200 bps hızları kullanılır. Modbus RTU, en çok kullanılan Modbus modelidir. Tezde Modbus RTU protokolü testler sırasında kullanılmıştır.

Modbus TCP, Ethernet üzerinden yürütülen Modbus modelidir. Bu modelde, Slave cihazlarla haberleşmek için cihaz adresleri yerine, IP adresleri kullanılır. Modbus TCP ile Modbus verileri bir TCP/IP paketinin içine yerleştirilerek taşınır.

2.1.1. Modbus Veri ve Adres Yapısı

Slave konumundaki her cihaz kendine özgü parametrelerin saklandığı bir hafızaya sahiptir. Modbus bu verilerin nasıl sınıflandırılacağını ve verilere nasıl erişileceğini belirler. Ancak bu verilerin cihaz hafızasında nereye yerleştirileceği konusunda bir sınırlama getirmez (McConahay 2011). Modbus verileri Çizelge 2.1'de gösterildiği gibi, özelliklerine göre dört bölüme ayrılmaktadır.

Veri İsmi	Bellek Türü	Adres
Bobinler	Okunabilir ve Yazılabilir	1-9999
Ayrık Girişler	Sadece Okunabilir	10001-19999
Giriş Kaydedicileri	Sadece Okunabilir	30001-39999
Saklayıcı Kaydedicileri	Okunabilir ve Yazılabilir	40001-49999

Çizelge 2.1. Modbus veri türleri **(**McConahay 2011**)**

Ayrık girişler ve bobinler, bir bitlik değerlerdir ve her biri bir adrese sahiptir. Giriş kaydedicileri, genellikle ölçüm değerleri gibi sadece okunabilen kayıtları belirtir. Saklayıcı kaydedicileri ise yazılıp okunabilen kayıtları belirtir. Giriş ve saklayıcı kaydedicileri bellekte 16 bitlik yer kaplar. Bir Slave cihaz tasarımında veriler, bellekte bu dört başlıkta sınıflandırılır ve cihazın Modbus bellek haritası kullanıcıya bildirilir. Kullanıcı Master cihazı bu haritaya göre programlar ve Slave cihazla iletişimi kolayca gerçekleştirir. Modbus protokolü, adresleme konusunda bir takım kurallar tanımlar. Çizelge 2.1'de gösterildiği üzere her veri bloğu 0 ile 65535 arasında adreslenir ve blokların Modbus adresleri, 10000, 30000 ve 40000 gibi başlangıç sayıları ile birbirinden ayrılır (Anonim 2016).

2.1.2. Modbus RTU

Modbus RTU, cihaz adresi, fonksiyon kodu, veri ve Döngüsel Artıklık Denetimi'nden (CRC) oluşan bir paket yapısına sahiptir. Cihaz adresi 0 ile 247 arasında bir sayıdır. 0 adresi, özel olarak yayın mesajları (broadcast message) için tahsis edilmiştir. 0 cihaz adresi ile gönderilen paketler, tüm Slave cihazlar tarafından kabul edilir ve değerlendirilir. 1-247 arasındaki diğer sayılar, Slave cihazlara adres olarak atanabilir. Slave cihazlar yayın mesajları dışındaki, tüm Modbus mesajlarına yanıt verirler. Böylece Master cihaz, mesajın Slave cihaz tarafından alındığını bilir. Fonksiyon kodu, veri okuma, veri yazma gibi işlevleri tanımlayan 1 ile 255 arasında bir değerdir.

Modbus RTU'da veri gösterimi, diğer endüstriyel protokollere göre daha basittir. Modbus RTU, verileri "Big Endian" bellek yapısında ikili olarak kodlar. Bu nedenle on altı bitlik veriler için MSB değeri, LSB değerinden önce kodlanır. Örneğin, 0x012C on altılık

değeri 0000 0001 0010 1100 şeklinde kodlanır. 0x01 MSB değeri olduğundan haberleşmede önce iletilir.

Modbus RTU'da hata kontrolü, CRC ile yapılır. CRC, 16 bitlik sayısal bir değerdir ve mesajı gönderen birim tarafından üretilir. Alıcı cihaz gelen pakete göre CRC değerini hesaplar ve karşı tarafın gönderdiği değer ile karşılaştırır. CRC değeri eşleşmez ise mesajın yeniden iletilmesi istenir.

2.1.3. Modbus Fonksiyonları

Fonksiyon kodları, Slave cihazın yürütmesi gereken işlevi ifade etmektedir. Bu kodlar Modbus paketinde sekiz bitlik yer kaplar. Modbus fonksiyon kodlarını üç kategoriye ayırmıştır. Bunlar Genel Fonksiyon Kodları, Kullanıcı Tanımlı Fonksiyon Kodları ve Rezerve Fonksiyon Kodları'dır. Uygulamalarda sıklıkla kullanılan Modbus fonksiyonları Çizelge 2.2'de belirtilmiştir.

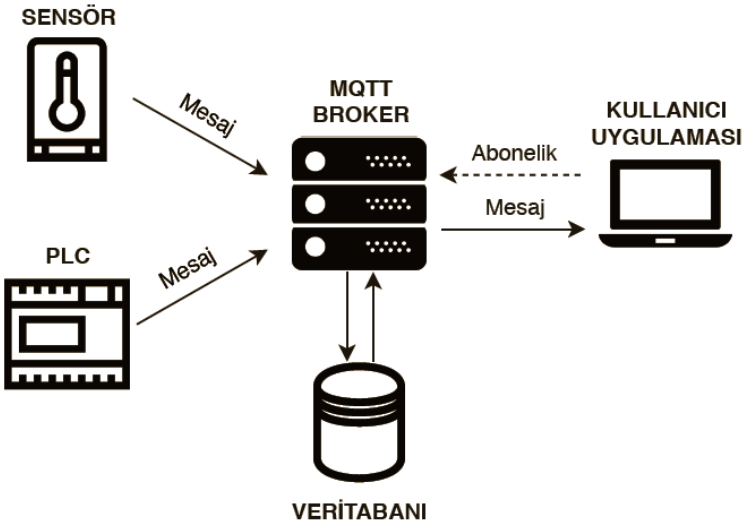
Fonksiyon Kodu	Fonksiyon İsmi
1	Dijital Çıkışları Okuma
2	Dijital Girişleri Okuma
3	Saklayıcı Kaydedicilerini Okuma
4	Giriş Kaydedicilerini Okuma
5	Tek Dijital Çıkışa Yazma
6	Tek Saklayıcı Kaydediciye Yazma
15	Çoklu Dijital Çıkışlara Yazma
16	Çoklu Saklayıcı Kaydedicilere Yazma

Çizelge 2.2. Sık kullanılan Modbus fonksiyonları **(**McConahay 2011**)**

2.2. MQTT

MQTT (Message Queuing Telemetry Transport) (Banks ve Gupta 2014), istemci ve sunucu arasındaYyayınla/Abone Ol yöntemine dayanan ISO tarafından standart haline getirilmiş bir mesajlaşma protokolüdür. Protokolün son versiyonu MQTT 3.1.1 (Banks

ve Gupta 2014) olarak yayınlanmıştır. Hafif paket yapısı, bellek tüketimi, farklı servis kalitesi desteği ve az enerji tüketilmesine neden olmasından dolayı endüstride nesnelerin interneti projelerinde tercih edilmektedir. Genellikle, TCP/IP ağ protokolü üzerinden çalışması tercih edilmektedir. MQTT, Yayınla/Abone Ol modelini baz almaktadır. Yaygın olarak kullanılan İstemci/Sunucu modelinde kullanıcı doğrudan bir uç nokta ile iletişim kurabilirken Yayınla/Abone Ol modelinde ise yayıncı, istemci veya abonelerden biri ile doğrudan iletişim kuramaz. Aralarındaki bağlantı Şekil 2.3'de görüldüğü üzere Broker adı verilen bir sunucu tarafından gerçekleştirilir. Günümüzde ActiveMQ (Anonim 2019a) ve HiveMQ (Anonim 2019b) gibi yaygın olarak kullanılmakta olan Broker sunucuları bulunmaktadır. Sunucular cihaz sayılarının artmasında bağlı olarak ölçeklenebilir mimariye uygun olarak tasarlanmışlardır. Örnek olarak, ActiveMQ sunucu başına yaklaşık 10000 adet cihazın bağlanmasına olanak sağlamaktadır. Ancak daha çok sayıda aktif bağlantı gerekmesi durumunda bir ya da daha fazla ActiveMQ sunucusunun birlikte çalışması sağlanabilmektedir.



Şekil 2.3. MQTT haberleşmesi

Broker aracılığı ile sağlanan haberleşmede uç birimler (cihazlar ve uygulamalar) doğrudan iletişim kurmadıkları için birbirlerinin IP adresi ve port gibi bilgilerine ihtiyaç duymazlar. Mesajlaşmak için uç birimlerin aracının IP adresi ve bağlantı noktasını bilmesi yeterlidir. Uç birimlerin kimlik doğrulama ve bilgi paylaşımının

denetlenmesinden Broker sorumludur. Broker'ların bir diğer avantajı ise farklı protokoller arasında çevrim yapılmasına olanak sağlamasıdır. Broker'lar yaygın olarak kullanılmakta olan REST, WebSocket, AMQP, OpenWire gibi protokoller arasında çevrim yapabilmektedirler. Dolayısı ile AMQP protokolü ile bağlı olan bir sunucu MQTT protokolü üzerinden gelen bir veriye, Broker aracılığı ile ulaşabilmektedir. Böylece, nesnelerin interneti için protokolden bağımsız bir platform kurulması mümkün olmaktadır. Broker sunucuları, yüksek performans ile çalışacak şekilde geliştirildikleri için verileri toplayan ya da işleyen sunuculardan kaynaklı problemlerin sistemi etkilemesine izin vermez. Broker, cihazdan gelen veriyi tüm abonelere tanımlı servisine bağlı olarak aktarmaktadır. Ayrıca, Broker'lar abonesi olmayan verileri saklayarak bir abone bağlanması durumunda aktarma olanağına da sahiptir. Dolayısı ile verilerin aktarımı kapsamında oluşabilecek problemlere dayanıklı bir mimari Broker'lar aracılığı ile kurulabilmektedir. Broker'lar mesajları konu ya da kuyruk mekanizmasına göre kabul eder ve abonelerine iletir. Broker'lardaki yetkilendirme mekanizması, kullanıcıların hangi konu ya da kuyruğa veri yazabileceğinin kontrolünü sağlamaktadır. Ayrıca, Broker'lar mesajların içeriğine müdahale edilmesine olanak sağlamaktadır. TÜBİTAK projesi kapsamında ActiveMQ sunucusu üzerinde kural tabanlı bir mesaj işleme mekanizması (Toğay 2017) geliştirilmiş ve uygulanmıştır. Ayrıca sunucuya tez kapsamında yeni anahtar takas yöntemi ile uçlar arasında şifreli iletişim ve kimlik doğrulama imkanı sağlanmıştır.

MQTT, haberleşmede TCP/IP protokolünü kullanır. Herhangi bir istemci Broker ile bağlantı kurmak istediğinde CONNECT mesajını gönderir. Broker mesajı değerlendirerek istemcinin kimliğini doğrulamaya çalışır. Broker sonucu CONNACK paketi ile istemciye bildirir. Bu pakette bağlantı kabul bilgisi ya da bağlantı reddini içeren beş farklı dönüş kodu (hatalı kullanıcı adı veya şifre, yetkisiz kullanıcı, çalışmayan sunucu gibi) belirtilir. CONNECT ve CONNACK paket yapıları Şekil 2.4'de gösterilmiştir.

CONNECT					
İstemci Kodu	Oturum Bayrağı	Kullanıcı Adı	Parola	LWT Parametreleri	Aktif Kalma Süresi
CONNACK					
Mevcut Oturum Bayrağı	Dönüş Kodu				

Şekil 2.4. CONNECT ve CONNACK paket yapıları

Broker tarafında yetkiler kullanıcı ve konu başlığına bağlı olarak belirlenmektedir. Hangi kullanıcının hangi konulara yayıncı ya da istemci olacağı yönetici tarafından daha önce belirlenir. Bir istemci Broker ile bağlantı kurduktan sonra PUBLISH paketini kullanarak mesaj yayınlamaya yetkisine bağlı olarak başlayabilir. PUBLISH paket yapısı Şekil 2.5'de gösterilmiştir. İstemci bu paket ile mesajın kendisini, konu adını (Örneğin, Cihaz/Kanal1/Nem), servis kalitesi seviyesini (QoS) ve saklanma durumunu Broker'a bildirir. Gönderilen mesaj bir metin dizesi, ikili veri ya da şifreli bir veri olabilir. Hangi veri tipinin kullanılacağı tamamen kullanıcı tarafından belirlenir.

PUBLISH					
Paket Kodu	Konu Adı	QoS	Tutma Bayrağı	Mesaj/Veri	DUP Bayrağı

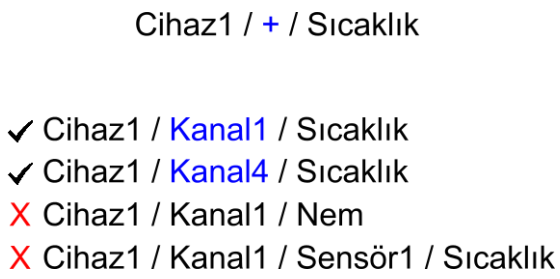
Şekil 2.5. PUBLISH paket yapısı

Broker'ın yayın mesajını yönlendirebilmesi için ilgili konuya ait abonelerin olması gerekir. Abonelik işlemleri SUBSCRIBE, SUBACK ve UNSUBSCRIBE paketleri (Şekil 2.6) ile yönetilir. Bir veya birden fazla konu için yapılacak abonelikler, servis kalitesi seviyeleri de belirtilerek SUBSCRIBE paketi ile Broker'a bildirilir. Broker abonelik talep edilen her bir konu için servis kalitesi seviyelerini de değerlendirerek sonucu SUBACK paketi ile istemciye gönderir. İstemci, konu aboneliklerinden ayrılmak istediğinde, ilgili konu isimlerini içeren UNSUBSCRIBE mesajını kullanır.

SUBSCRIBE					
Paket Kodu	QoS 1	Konu 1	QoS 2	Konu 2	…
SUBACK					
Paket Kodu	Dönüş Kodu 1	Dönüş Kodu 2	…		
UNSUBSCRIBE					
Paket Kodu	Konu 1	Konu 2	…		

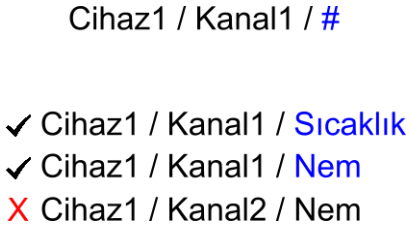
Şekil 2.6. SUBSCRIBE, SUBACK ve UNSUBSCRIBE paket yapıları

MQTT konu parametresinin oluşturulmasında UTF-8 dizelerini kullanır, küçük büyük harf duyarlılığına sahiptir ve geçerli olması için en az bir karakter içermesi gerekir. Konu birden fazla bölüm içerebilir. Bu durumda her bir bölüm '/' karakteri ile ayrılmalıdır. Bir konu dizesi ile tek bir abonelik gerçekleştirilir. Aynı adla birden çok konuya abone olunmak istenirse bir takım özel karakterlerden faydalanılır. Bir konu aboneliğinde '+' karakteri kullanıldığında konu dizesinde bu karakterin yerine rastgele bir dize gelebilir. Bu iki konunun da birbiriyle eşleşmesini sağlar. Bu durumu gösteren bir örnek Şekil 2.7'de belirtilmiştir.



Şekil 2.7. '+' karakteri kullanılmış bir konu örneği

Konu dizesinde kullanılan '+' karakteri ile sadece belirli bir bölüm değişken yapılabilirken, '#' karakteri kullanılarak birden fazla bölüm değişken yapılabilmektedir. '#' karakterli bir konuya abone olunduğunda, konunun uzunluğuna bakılmadan '#' karakterinden önceki dize esas alınır. Bu durumu gösteren bir örnek Şekil 2.8'de belirtilmiştir.



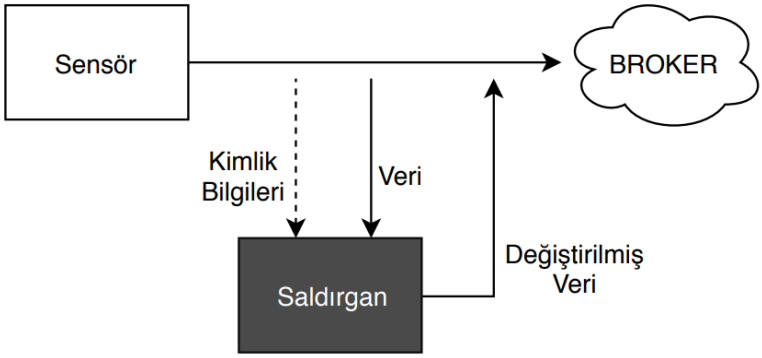
Şekil 2.8. '#' karakteri kullanılmış bir konu örneği

MQTT'deki bir diğer önemli parametre de servis kalitesi seviyesi (QoS)'dir. QoS 0, QoS 1 ve QoS 2 olmak üzere üç türü vardır. Bu parametre gönderici ile alıcı arasındaki teslimat durumunu tanımlar. QoS 0'da gönderici mesajı bir kez gönderir ve Broker herhangi bir dönüş mesajı beklemez. Bu yüzden QoS 0 en düşük servis kalitesi seviyesidir. QoS 1'de gönderici mesajı gönderir ve alıcıdan PUBACK onay paketinin gelmesini bekler. Onay paketi gelmez ise gönderici onay paketini alana kadar aynı mesajı göndermeye devam eder. Bu şekilde mesajın Broker'a en az bir kez gönderildiği garanti edilir. QoS 2'de ise gönderici mesajı gönderir ve Broker'dan onay anlamı taşıyan PUBREC paketini bekler. Gönderici PUBREC paketini aldıktan sonra alıcıya yanıt olarak PUBREL paketini gönderir. Alıcı ise PUBREL paketini aldıktan sonra PUBCOMP paketini göndericiye yanıt olarak gönderir. Bu şekilde her iki tarafında mesajların iletildiğinden emin olması sağlanır. QoS 2 en yüksek servis kalitesi seviyesidir.

2.2.1. MQTT ve Güvenlik

MQTT'nin IIoT uygulamalarında kullanımının artmasıyla birlikte siber güvenlik konusu önem kazanmıştır. Örneğin, IIoT sistemiyle yürütülen bir fabrikada güvenlik açıklarından dolayı sistem verilerinin sızması ya da verilerin içeriğinin değiştirilmesi fabrikaya büyük zararlar verebilmektedir. MQTT ağ, ulaşım ve uygulama seviyesinde bir takım güvenlik tedbirlerinin alınmasına olanak sağlar. Kullanılan haberleşme ağı herhangi bir dış bağlantıdan izole edildiğinde veya bir Sanal Özel Ağ (VPN) kullanıldığında ağ seviyesinde en yüksek güvenlik sağlanmış olur. Ulaşım seviyesinde ise yaygın olarak verilerin şifrelenerek taşınmasını sağlayan TLS/SSL gibi protokoller kullanılır. Bu protokoller şifreli veri iletiminin yanında her iki tarafında kimliğini doğrulamak için sertifika tabanlı kimlik doğrulama sistemleri kullanırlar. Ancak bu protokoller yoğun

işlemler ve bir bellek alanı gerektirdiğinden, kısıtlı cihazlara uygulanmaları güçlükler barındırmaktadır. MQTT uygulama seviyesinde, kimlik doğrulama için istemci kimliği, kullanıcı adı ve şifre parametrelerine sahiptir. İstemci bir Broker'a bağlanırken bu parametreleri CONNECT mesajı ile iletir. Broker ise bu bilgiler aracılığıyla istemciyi tanımlar. Ancak tüm bu bilgiler düz metin olarak iletildiğinden Şekil 2.9'da gösterildiği üzere saldırganlar tarafından kolayca elde edilebilir. Bu nedenle güvenli bir iletişim için TLS ya da AES gibi simetrik şifreleme algoritmalarına başvurulmalıdır.

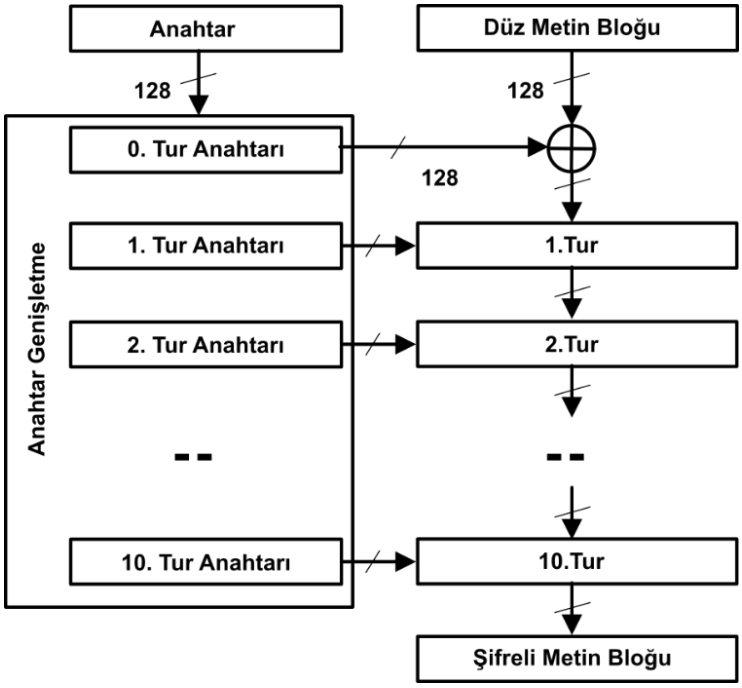


Şekil 2.9. Bir MQTT ağından kimlik bilgilerinin elde edilmesi

2.3. AES

Gelişmiş Şifreleme Standardı (AES), Kasım 2001'den beri yaygın olarak kullanılmaktadır (Anonim 2001a). AES, 128 bitlik blok büyüklüğüne ve 128, 192 ve 256 bitlik olmak üzere üç adet anahtar boyuta sahiptir. AES'de bazı turlar anahtar boyutlarına göre tanımlanır (Anonim 2001a). On turdan oluşan AES128 algoritmasının yapısı Şekil 2.10'da gösterilmiştir. AES algoritması hem donanım hem de yazılım tarafından gerçekleştirilebilir (Wardhani ve ark. 2017). Bellek, işlem ağırlığı ve güç tüketimi gibi performans ve kaynaklar dikkate alınarak uygulanır. Bazı mikrodenetleyicilerde AES şifrelemesi yapabilen özel bir donanım vardır (Schwabe ve Stoffelen 2016). Ancak bu özellik güç tüketimini ve maliyeti arttırmaktadır. AES yazılımına ait uygulamalardan biri, ARM Cortex-M3 işlemciler için bellek konusunda optimize edilmiştir (Wardhani ve ark. 2017). Bir başka yaklaşımda, AES hızını arttırmak için ARM komut setleri kullanılmış ve yan kanal saldırılarına ilişkin bazı önlemler verilmiştir (Schwabe ve Stoffelen 2016).

Tez çalışması kapsamında bir donanım (ATAES132A) ve mbedTLS (Anonim 2008) ve TinyCrypt (Anonim 2019) olmak üzere iki yazılım uygulaması incelenmiştir. Uygulamalar test vektörleri ile test edilmiştir (Bassham 2002). TinyCrypt kütüphanesi kısıtlı cihazları hedefler ve HMAC-PRNG, AES-CTR-PRNG rasgele sayı üreteçlerinden, AES-128 blok şifrelemesinden (AES-CBC, AES-CTR ve AES-CMAC şifreleme modlarıyla), AES-CCM ve ECC_DSA dijital imza algoritmalarından oluşur. Yazılım uygulamaları, yan kanal saldırılarına maruz kalabilmektedirler. Yan kanal saldırılarına karşı bazı önlemler uygulanabilir ancak bunlar genel kod boyutunu arttırmaktadır. TinyCrypt genel zamanlama saldırıları için bazı önlemler içerir. MbedTLS ise simetrik şifreleme (Blowfish, Triple-DES (3DES), DES, ARC4, Camellia, XTEA), karma (MD2, MD4, MD5, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, RIPEMD-160), RSA / PKCS # 1, Diffie-Hellman / PKCS # 3 ve Eliptik Eğri Kriptografisi (ECC) gibi algoritmalara sahiptir.



Şekil 2.10. AES-128 algoritması (Hall 2013)

3. MATERYAL VE YÖNTEM

3.1. Tasarlanan Ağ Geçidi Cihazı

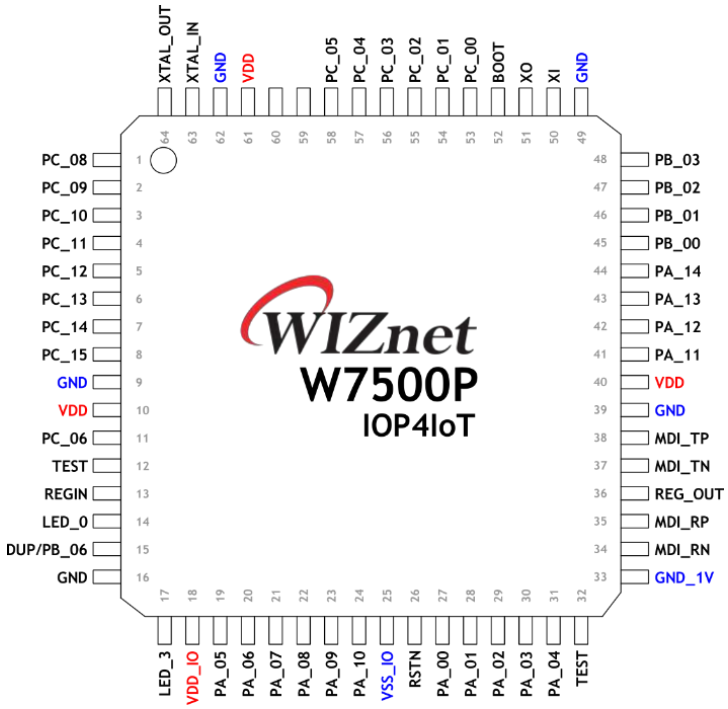
Tez kapsamındaki IIoT çözümünde, cihazların Broker ile güvenli haberleşmesi, MQTT protokolü ile iletişimin sağlanması ve kısıtlı cihazların ihtiyaç duyduğu mesaj zenginleştirme gibi operasyonlar için TÜBİTAK projesi kapsamında bir ağ geçidi geliştirilmiştir (Şekil 3.1). Kart temelde W7500P (ARM Cortex-M0) (Anonim 2015a) mikrodenetleyicisi, RS-232, RS-485, ATAES132A ve Ethernet bağlantılarını içermektedir.

Şekil 3.1. Tasarlanan Ağ Geçidi

3.1.1. W7500P Özellikleri

W7500P (ARM Cortex-M0) (Anonim 2015a), WIZNET firması tarafından özellikle IoT sistemlerini gerektiren gömülü uygulamalar için geliştirilmiş, düşük maliyetli bir mikrodenetleyicidir (Şekil 3.2). Yüksek performanslı ARM Cortex-M0 32-bit RISC, 48

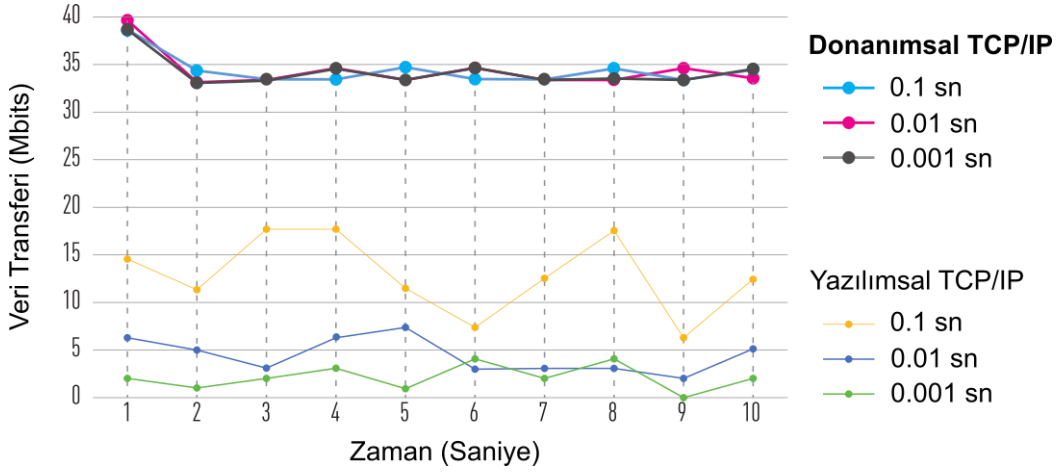
MHz frekansında çalışabilen çekirdek, 128 Kbyte FLASH bellek, 16 Kbyte SRAM bellek, donanımsal TCP/IP ve I/O birimlerini içermektedir. Standart haberleşme arayüzü olarak SPI ve UART çevrebirimlerine sahiptir. W7500P, önyükleme kodu için 6 KB ROM bellek bulundurur. Bu sayede, açılışta boot pini aktif edilerek mikrodenetleyici önyükleme moduna geçirilebilir ve herhangi bir programlayıcıya ihtiyaç duymadan flash alanı programlanabilir.



Şekil 3.2. W7500P mikrodenetleyicisi (Anonim 2015a)

Ethernet haberleşmesi, fiziksel katman işlemlerinin gerçekleştirilebilmesi için ayrı bir Ethernet fiziksel birimine (PHY) ihtiyaç duymaktadır. Ancak, tasarlanan yeni ağ geçidi kartında PHY birimini de içinde bulunduran W7500P mikrodenetleyicisi kullanılmıştır. Bu sayede maliyet azaltılırken kart çiziminde yer tasarrufu sağlanmıştır. W7500P, aynı zamanda donanımsal TCP/IP birimine sahiptir. Bu birim TCP, UDP, IPv4, ICMP, ARP, IGMP ve PPPoE protokollerini destekleyen çeşitli uygulamalarda kullanılır. Şekil 3.3'de belirtildiği gibi DDoS saldırılarında gerçekleşen yüksek veri trafiğini donanımsal TCP/IP, yazılımsal TCP/IP'e göre daha başarılı bir şekilde işleyebilmektedir (Anonim 2015b).

W7500P, 0-+70 ℃ sıcaklık aralığı ile 2,7–3.6 V besleme aralığında çalışmaktadır (Anonim 2015a).

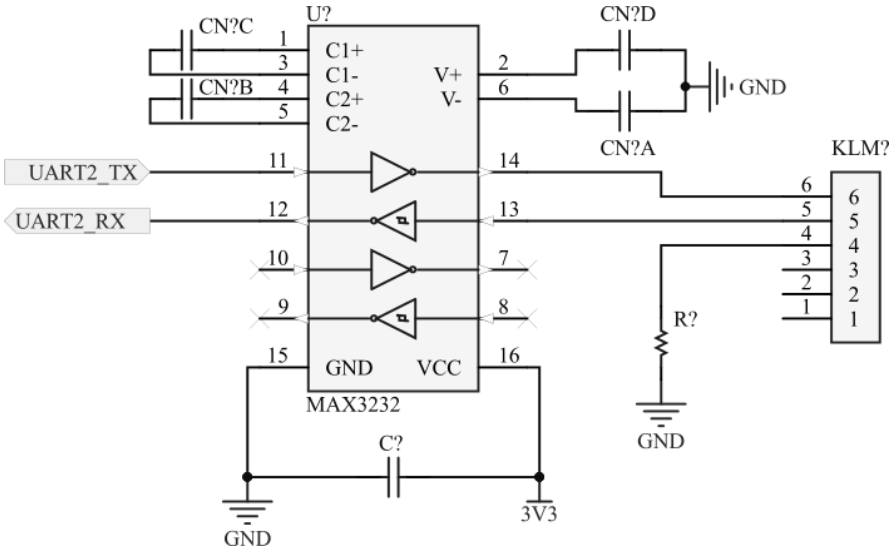


Şekil 3.3. DDoS saldırısı için donanımsal ve yazılımsal iletişim performansının karşılaştırılması (Anonim 2015b)

3.1.2. RS-232 Bağlantısı

RS-232, verilerin seri iletimi için 1960 yılında tanıtılan bir haberleşme standardıdır (Anonim 2001b). Endüstriyel otomasyon ve kontrol sistemlerinde yaygın olarak kullanılmaktadır. Yaygın olarak kullanılmasının en önemli sebebi standardın sade ve kolay olmasıdır. Kullanıcıların seri portlar ile doğrudan iletişim kurmasını sağlar. RS-232 haberleşme standardı, Evrensel Asenkron Alıcı Verici (UART) ile birlikte çalışır. UART işlemci veya kontrol ünitesine entegre edilmiş bir çevrebirimidir.

Tasarlanan deneysel kartta RS-232 hat sürücü olarak Texas Instruments firmasının ürettiği MAX3232 (Anonim 2017a) entegresi kullanılmıştır. Bu entegre 3,3V ile 5,5V aralığında besleme gerilimine sahiptir ve iki adet alıcı/verici içerir. RS-232 hattı için ±15 kV ESD (Elektrostatik Deşarj) korumasına sahiptir. MAX3232 ile oluşturulan RS-232 devre şematiğinde (Şekil 3.4) besleme gerilimi W7500P ile uyumlu olması açısından 3,3V kullanılmıştır. RS-232 çıkışı ise TX, RX ve GND olmak üzere üç uçtan oluşmaktadır.

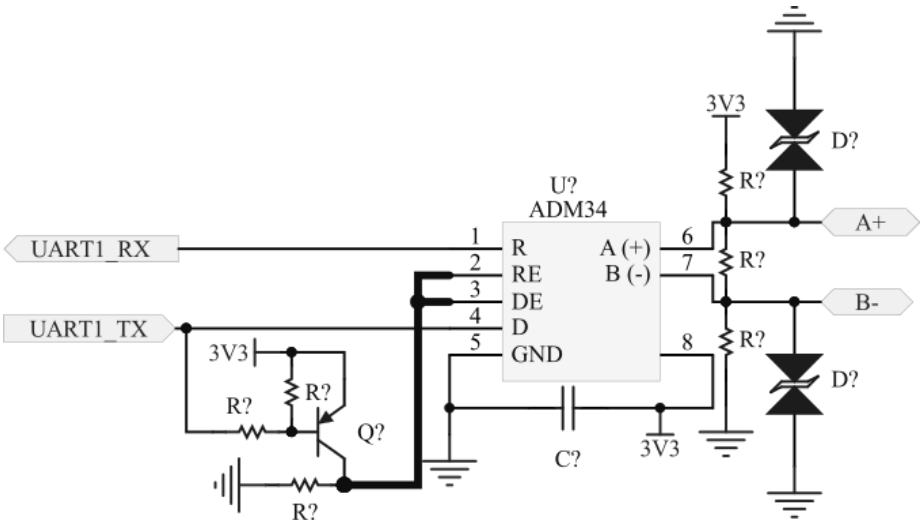


Şekil 3.4. RS-232 bağlantı şematiği

3.1.3 RS-485 Bağlantısı

RS-485 olarak bilinen TIA-485 standardı, seri iletişim sistemleri için sürücü ve alıcıların elektriksel özelliklerini tanımlayan bir standarttır. Bu standart Telekomünikasyon Endüstrisi Birliği (TIA) ve Elektronik Endüstrileri Birliği (EIA) tarafından 1983 yılında yayınlanmıştır (Kugelstadt 2016). Standart uzun mesafeli ve elektriksel gürültülü iletişim ağlarında, etkili bir şekilde kullanılmaktadır. RS-485, çok uçlu bir veri ağına birden fazla alıcının bağlanabilmesini sağlar. Bu sayede endüstriyel kontrol sistemlerinde geniş bir uygulama alanına sahip olmuştur.

Tasarlanan deneysel kartta, RS-485 devresi için Analog Devices firmasının ürettiği ADM3483 (Anonim 2011) entegresi kullanılmış ve devre şeması Şekil 3.5'de gösterildiği gibi oluşturulmuştur. Bu alıcı/verici entegre çok noktalı veri yolu iletim hatlarında çift yönlü veri iletimi için tasarlanmıştır. Entegre, RS-422 ve RS-485 haberleşme standartlarını desteklemektedir. ADM3483, 3,3V besleme gerilimi ile çalışan 3 durumlu diferansiyel hat sürücüsü ve giriş hattı alıcısından oluşur. Sürücü ve alıcı yön kontrolü için harici olarak birbirine bağlanabilen iki girişe sahiptir.



Şekil 3.5. RS-485 bağlantı şematiği

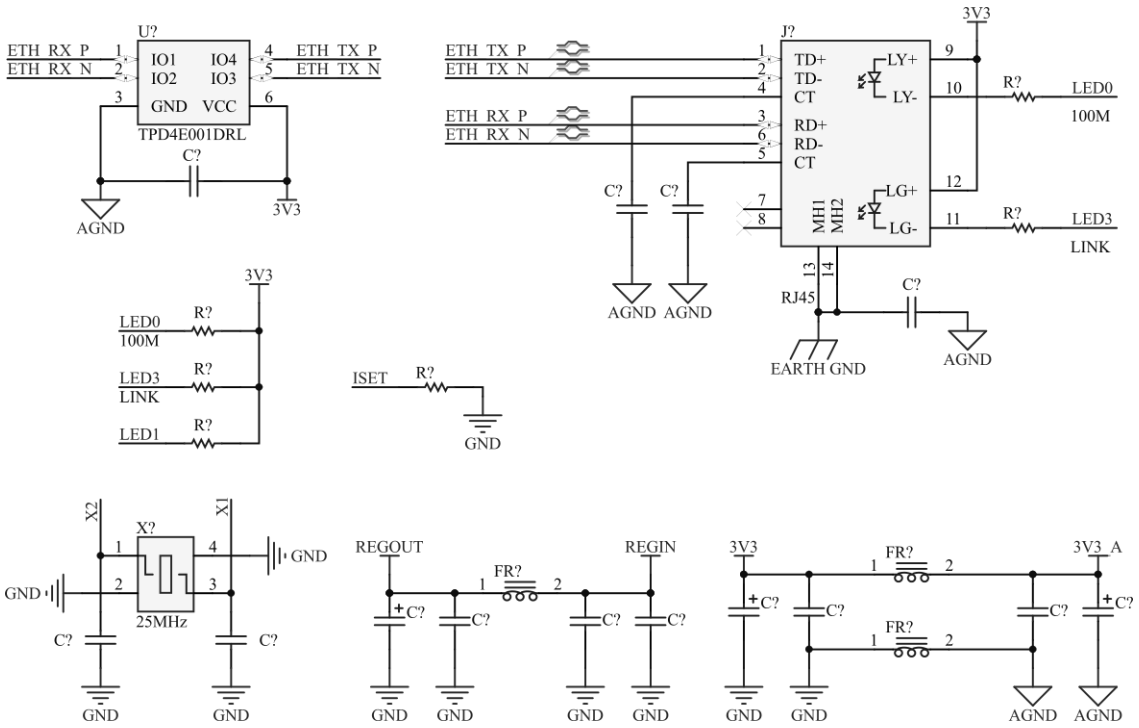
3.1.4 Ethernet Bağlantıları

Ethernet haberleşmesi için Ethernet MAC (Media Access Controller) birimi, Ethernet PHY birimi ve soket bağlantıları gerekmektedir. Ethernet MAC, IEE-802.3 Ethernet standardı (Hollenbeck 2001) olarak tanımlanmaktadır. Bu birim veri bağlantı katmanı işlevlerini uygular. Ethernet PHY birimi ise IEEE-802.3 standardı ile tanımlanan fiziksel katman işlevlerini yürüten bir alıcı-vericidir. Bu iki birimin haberleşmesi için MII (Media Independent Interface)ve RMII (Reduced Media Independent Interface) olmak üzere iki standart tanımlanmıştır (Simmons 2008).

W7500P, Ethernet MAC biriminin yanında Ethernet PHY birimini de dahili olarak bulundurmaktadır. Bu iki birim MII arayüzü ile entegre içerisinde birbirine bağlanmıştır. Ethernet PHY biriminin portları, bağlantıların yapılabilmesi için W7500P'nin pinlerine entegre içerisinde bağlanmıştır. W7500P, Ethernet PHY birimi olarak IP101G tümdevre yapısına sahiptir. IP101G (Anonim 2012), 100Mbps veya 10Mbps işlemler için IEEE 802.3/802.3u uyumlu hızlı Ethernet Alıcı-Vericisidir. Gelişmiş CMOS (85nm) teknolojisi ile üretilmiştir.

Ethernet bağlantıları için gereken devreler referans şematiklerde belirtildiği gibi oluşturulmuştur (Şekil 3.6). Burada REGOUT, REGIN, ISET, X1, X2, AVDD, AGND, LED0, LED3, MDI_TP, MDI_TN, MDI_RP ve MDI_RN pinleri Ethernet PHY birimi

ile ilgilidir. X1 ve X2 pinlerine bağlanan 25MHz kristal Ethernet PHY ve Ethernet MAC birimini çalıştıran saat sinyallerini üretmek için kullanılır. MDI_TP, MDI_TN, MDI_RP ve MDI_RN soket üzerinden veri alışverişini sağlayan hatlardır. LED0, 100Mbps hız durumunu, LED1 ise bağlantı durumunu belirten gösterge bağlantılarıdır. Diğer pinler Ethernet PHY biriminin besleme ve iç gerilimlerini oluşturan bağlantıları ve komponentleri içerir. Veri hatlarında sonlandırma direnci olarak kullanılan 49,9 ohm dirençler, Ethernet PHY biriminin içinde dahili olarak bulunduğundan şematiğe eklenmemiştir.

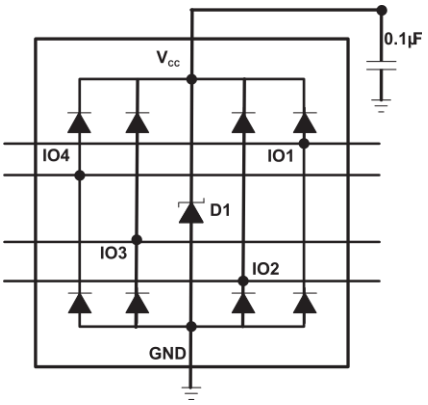


Şekil 3.6. Ethernet bağlantı şematiği

Ethernet hatları, izolasyon trafolu RJ45 soketi ile dışarı açılmaktadır. RJ45 soketinin şasi toprağı, sistem toprağına 1nF/2kV kondansatör ile bağlanmıştır. 1nF/2kV kondansatör 100 MHz'de yaklaşık 1,6 ohm bir empedansa sahiptir. Bu sayede kablo ekranı ile toprak arasındaki RF gürültüleri için düşük bir empedans yolu sağlanmış olur.

Ethernet hatlarındaki bir diğer önemli unsur ESD (Electrostatic Discharge) korumasıdır. ESD, iki nesne temas ettiğinde, şarj olmuş nesneden diğerine elektriğin aniden boşalmasıdır. Bu durum insanlar için oldukça zararsız iken hassas entegre devreler için

ciddi arızalara neden olabilmektedir. Elektronik tasarımlarda bu durumu önlemek için ESD koruma diyotları kullanılır. ESD koruma diyotları, arayüz konnektörü ile PHY arasındaki sinyal hattına paralel bağlanır. Herhangi bir elektrostatik boşalma durumunda diyotlar aktif olarak mevcut akımı toprağa yönlendirir ve entegre devreyi korumuş olur. Ethernet Tx/Rx sinyal hatları için 4 kanallı bir ESD koruması gerekir. Şematikte bu amaçla Texas Istruments firmasının ürettiği TPD4E001 (Anonim 2019c) entegresi kullanılmıştır. TPD4E001, yüksek hızlı veri hatları için geliştirilmiş, düşük kapasitanslı, 4 kanallı bir ESD koruma entegresidir (Şekil 3.7).

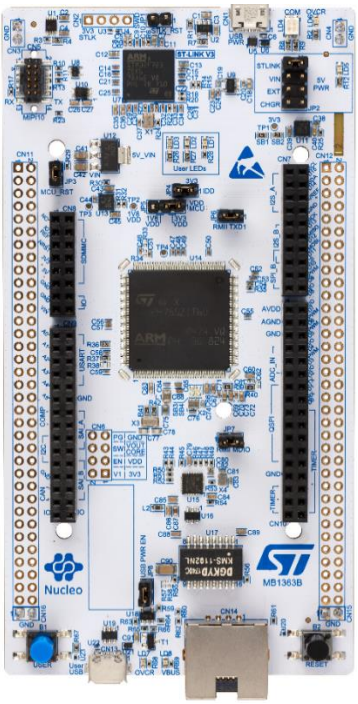


Şekil 3.7. TPD4E001 entegre yapısı (Anonim 2019c)

3.2. STM32 NUCLEO – 144 Geliştirme Kartı Özellikleri

NUCLEO-144 (Anonim 2017b), STMicroelectronics firması tarafından geliştirilmiş kullanıcılara yeni tasarımlar denemek ve prototipler oluşturmak için imkân sağlayan, uygun maliyetli bir geliştirme kartıdır (Şekil 3.8). Aynı zamanda STM32Cube MCU paketi ile sunulan ücretsiz yazılım kütüphaneleri ve örneklerine doğrudan erişim imkanı sağlar. Bu geliştirme kartı, ST-LINK programlayıcı devresini üzerinde bulundurduğu için ayrı bir programlayıcıya gerek duymadan kolayca programlanabilir. Mikrodenetleyici olarak, STMicroelectronics firmasının geliştirdiği STM32F207ZG (Anonim 2019d) entegresini kullanır. STM32F207ZG mikrodenetleyicisi, 120 MHz'e kadar çalışabilen, yüksek performanslı Arm Cortex–M3 32-bit RISC çekirdeğini temel alır. Kartın çalışmada kullanılmasındaki temel amaç, donanımsal I^2C biriminden faydalanmaktır. Bir Mbyte flash bellek, 128 Kbyte SRAM bellek alanına sahiptir. Kartın tezde kullanılmasındaki temel amaç, donanımsal I^2C imkanı sunan Arm Cortex–M3 işlemcisini

barındırmasıdır. Donanımsal I^2C aracılığı ile işlemci üzerinde MQTT paketlerinin gönderimi yapılırken aynı zaman da şifreleme gibi işlemci tabanlı görevler gerçekleştirilebilmektedir. Tezde, donanımsal I^2C aracılığı ile düşük hızlı internet bağlantısının olduğu ortamlarda, işlemcinin performansının optimum kullanılarak şifreli bir iletişim sağlanması hedeflenmiştir.



Şekil 3.8. STM32 NUCLEO–144 geliştirme kartı (Anonim 2017b)

3.3. ATAES132A

ATAES132A (Anonim 2018), Microchip firması tarafından, kimlik doğrulama, kalıcı olarak gizli veri depolama gibi yeteneklere sahip güvenlik uygulamaları için geliştirilmiş bir entegredir (Şekil 3.9). Temel özellikleri aşağıdaki gibidir:

- ATAES132A, AES-128 şifreleme algoritmasına sahiptir ve kimlik doğrulama, depolanan veriyi şifreleme, şifre çözme ve mesaj doğrulama kodları üretmek için AES-CCM modunu kullanır. ATAES132A ile şifrelenen veri, aynı anahtarı kullanan başka bir cihaz tarafından çözülebilir.
- Rastgele sayı üretimi için Federal Information Processing Standards (FIPS) tarafından onaylı rastgele sayı üretecine sahiptir.

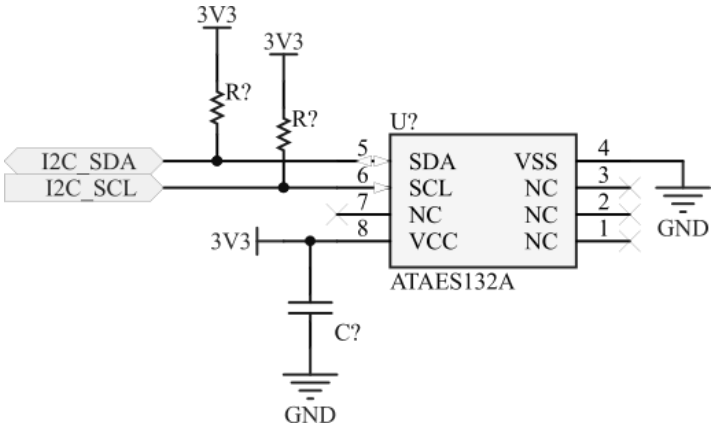
- 32 Kbyte bellek alanı olan Seri EEPROM özelliğine sahiptir. Bu alan 2 Kbyte bölümlere ayrılarak, entegre içinde 16 adet kullanıcı bölgesi oluşturulmuştur. Her bir kullanıcı bölgesi, kullanıcı tarafından erişim kısıtlamalarına göre önceden yapılandırılabilmektedir. Anahtar belleğinde 128 bit uzunluğunda 16 adet anahtarı güvenli bir şekilde saklanabilmektedir. Bu anahtarlar, yalnızca kullanıcı tarafından konfigürasyonu yapılmış etkin kriptografik işlevlerde kullanılabilir. ATAES132A, anahtarları güvenli bir şekilde sakladığından kriptografik işlemler gerçekleştirilmeden önce değiştirilmeleri gerekmez. Anahtar belleği bir kere kilitlendikten sonra anahtarlar dışarıdan birebir elde edilemez.
- ATAES132A, bir saldırganın dahili gizli verileri belirlemesini engellemek için bazı fiziksel güvenlik önlemleri içerir. Gerilim, sıcaklık, frekans ve ışık gibi nicelikler ile gerçekleştirilebilecek saldırılar için tamper dedektörleri bulundurur. Ayrıca devre üzerinde aktif bir metal korumaya ve dahili bellek şifrelemesine sahiptir.
- ATAES132A, fiziksel tasarımı ve şifreleme protokolü ile yan kanal saldırılarını önlemek için veya önemli ölçüde zorlaştırmak için özel olarak tasarlanmıştır.

ATAES132A'nın pin çıkışları elektronik kart çizimlerini kolaylaştırmak için diğer Seri EEPROM entegreleri ile uyumlu yapılmıştır. Entegre haberleşme seçeneği olarak 1MHz'e kadar çalışabilen I^2C protokolünü ve 10 MHz'e kadar çalışabilen SPI protokolünü sunmaktadır.

Şekil 3.9. ATAES132A entegresi (Anonim 2018)

3.3.1 ATAES132A Bağlantısı ve Haberleşme Protokolü

Her iki kart üzerinde anahtar takası ve şifreli iletişim için ATAES132A entegre devresi uygulanmıştır. ATAES132A, I^2C ve SPI haberleşme seçeneklerine sahiptir. Hali hazırda kullanılan EEPROM entegreleri ile uyumlu olması açısından haberleşme için I^2C protokolü tercih edilmiş ve şematik buna göre oluşturulmuştur (Şekil 3.10). I^2C, bir Master cihaz ile Slave konumundaki cihazların iletişim kurmasını sağlayan ve yaygın olarak kullanılan etkili bir haberleşme standardıdır (Anonim 2002). Diğer haberleşme standartları ile karşılaştırıldığında en büyük avantajı sadece iki veri yolu (SDA ve SCL) kullanmasıdır. Aynı veri yolu üzerinde birden fazla cihazın (farklı adreslere sahip olmak koşuluyla) haberleşmesini sağlayabilir. Genellikle 100 - 400 Kbps hızları arasında kullanılır.

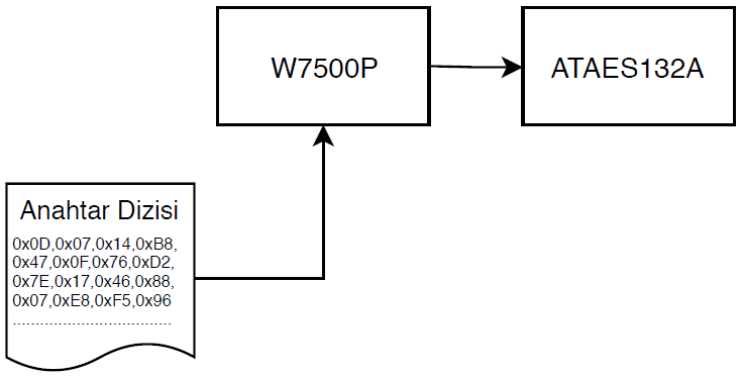


Şekil 3.10. ATAES132A bağlantı şematiği

3.3.2 ATAES132A Anahtarların Yüklenmesi

Daha önce ifade edildiği üzere, ATAES132A, her biri 128 bit uzunluğunda 16 adet anahtarı güvenli bir şekilde saklayabilen anahtar belleğine sahiptir. Her anahtar konfigürasyon belleğindeki ilgili kayda göre konfigüre edilebilmektedir. Anahtarlar sadece konfigürasyon kaydında etkinleştirilen şifreleme işlemleri için kullanılabilir. Çalışmada kullanılan anahtarlar rastgele sayı üreteci ile üretilmiştir. Üretilen anahtarlar 0xF200 ile 0xF2FF adresleri arasındaki anahtar belleğine (anahtar belleği kilitlenmeden önce) yazma komutu ile I^2C üzerinden yazılmış (Şekil 3.11) ve anahtar belleği

kilitlenmiştir. Bu işlem bir kere yapılmıştır ve kaydedilen anahtar değerleri hiçbir koşulda ATAES132A'dan okunamamaktadır. Anahtar yükleme işlemi ATAES132A PCB üzerinde takılı iken mikrodenetleyici ile gerçekleştirilmiştir. Bu işlem için mikrodenetleyici üzerinde tek sefer çalıştırılan ayrı bir kod parçacığı yazılmıştır.



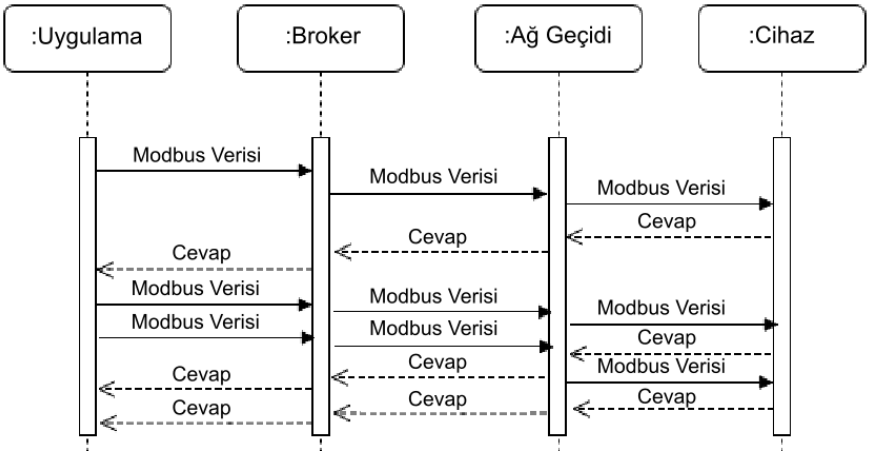
Şekil 3.11. Anahtarların ATAES132A'ya yüklenmesi

Anahtar yazma işlemi, kullanıcı tarafından istenirse entegrenin komut setlerinden EncWrite komutu ile şifreli bir şekilde de yapılabilmektedir. Kilitleme işleminden sonra, anahtar kayıtları KeyCreate, KeyImport, KeyLoad ve KeyTransfer entegre komutları ile yönetilir. Bu komutların kullanılabilmesi için kullanılan anahtara ait konfigürasyon kaydında ilgili izinlerin verilmiş olması gerekir.

3.4. Ağ Geçidi Yazılımı

3.4.1. Ağ Geçidinin Modbus RTU Ağı ile Haberleşmesi

Tasarlanan deneysel ağ geçidinin seri bağlantı arayüzleri kullanılarak Slave cihazlar ile iletişim kurulabilir. Örneğin, RS-485 hattı kullanılarak bir Modbus RTU ağı oluşturulabilir. Ağ geçidi, Ethernet üzerinden gelen istekleri Modbus protokolü ile Slave cihazlara yönlendirir. Slave cihazlardan Modbus protokolü ile gelen cevapları ise MQTT protokolüne çevirerek Ethernet üzerinden MQTT Broker'ına iletir. Bu durumda Ağ geçidi Modbus ağı için Master cihaz konumunda olmaktadır (Şekil 3.12).



Şekil 3.12. Modbus ile MQTT üzerinden uygulama ve cihaz arasındaki iletişim akışı

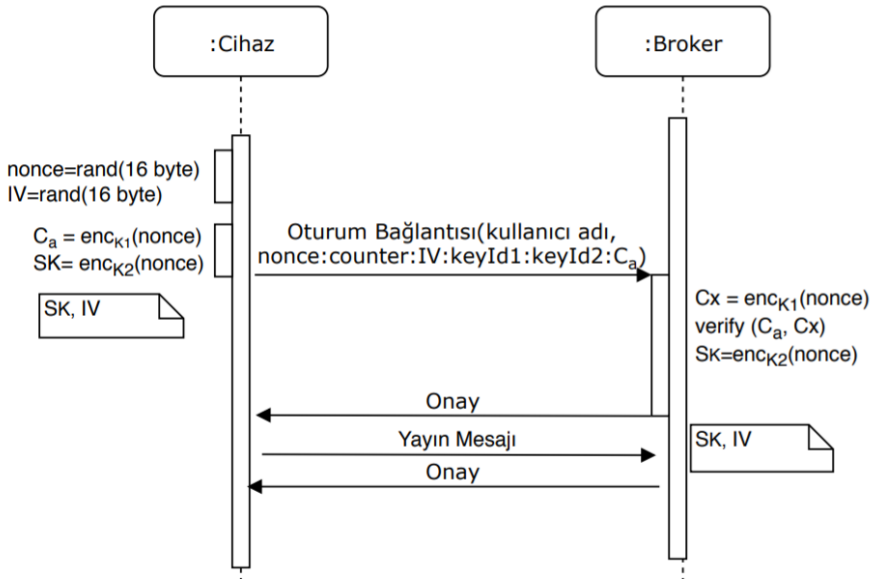
Ağ geçidi, Slave cihazlarla haberleşmek için fonksiyon 3, 4, 6 gibi standart Modbus fonksiyonlarını kullanır. Ağ Geçidinin Modbus ağındaki Slave cihazlar ile ilgili önceden kaydedilmiş bir verisi yoktur. IIoT platformundaki uygulama, Ağ Geçidine bağlı Slave cihazların cihaz kodları, kayıt adresleri gibi bilgilerini bilmek zorundadır. Bu yüzden Ethernet üzerinden gelen istek paketinde, cihaz kodu, işlev kodu, adres aralığı ve veri (yazma işlemi yapılacaksa) gibi Modbus sorgusu için gereken parametreler ağ geçidine bildirilir. Ağ geçidi bu parametreleri uygun şekilde birleştirir ve hata kontrolü değerini de ekleyerek bir sorgu paketi hazırlar ve paketi Modbus ağına gönderir.

Ağ Geçidine istekler art arda gelirse Modbus ağının durumuna göre hareket edilir. Modbus sorgu/cevap yapısıyla çalışan bir protokol olduğundan işlenen istek paketleri art arda Modbus ağına gönderilemez. Bu durumda Modbus ağı içinde ayrı bir kuyruk yapısı oluşturulmasına ihtiyaç vardır.

3.4.2. Ağ Geçidinin Broker Bağlantısı ve Kimlik Doğrulama

Broker olarak, TÜBİTAK 1505 projesi kapsamında değiştirilen ActiveMQ (Anonim 2019a) tabanlı sunucu kullanılmıştır. Geliştirilen Broker cihazların doğrulanmasından, mesajların şifreleme işlemlerinden ve mesaj bütünlüğünün sağlanmasından sorumludur. Her bir ağ geçidi, Broker ile önceden paylaşılmış 16 adet simetrik anahtara sahiptir. Örneğin, Broker ile kurulacak bir bağlantı sırasında K_1 anahtarı kimlik doğrulama için C_a üretiminde K_2 anahtarı ise SK (oturum anahtarı) üretiminde Şekil 3.13'de gösterildiği

üzere kullanılır. K_1 ve K_2 Broker'ın veritabanında sırasıyla keyId_1 ve keyId_2 ile ilişkilendirilir. Cihaz Broker ile bir TCP/IP bağlantısı kurduğunda ATAES132A rastgele 12 byte nonce ve 16 byte başlatma vektörü (IV) değerini oluşturur. Nonce değeri 32 bitlik counter değeri ile genişletilir. Counter, keyId_1 ve keyId_2 ile tanımlı anahtarların kullanım sayısını ifade etmektedir. Cihaz kullanıcı adını ve şifreyi "nonce:counter:IV:keyId_1:keyId_2:C_a" şeklinde göndererek kimlik doğrulamayı başlatır. Broker, Şekil 3.13'de gösterildiği üzere C_a değerini aldığında, kullanıcıya tanımlı ilgili keyId_1 ile ilişkili anahtarı kullanarak nonce değerini doğrular. Sonrasında ağ geçidi ile Broker arasındaki şifreli iletişimde kullanılacak olan oturum anahtarı (SK) keyId_2'ye tanımlı simetrik anahtar kullanılarak üretilir. Bu işlemden sonra her iki tarafta mesaj şifrelemek ve çözmek için gereken oturum anahtarı (SK) ve başlangıç vektörüne (IV) sahip olmuş olur. Kimlik doğrulama işlemi başarılı olunca Broker veritabanındaki sayacın yeni değerini belirler.

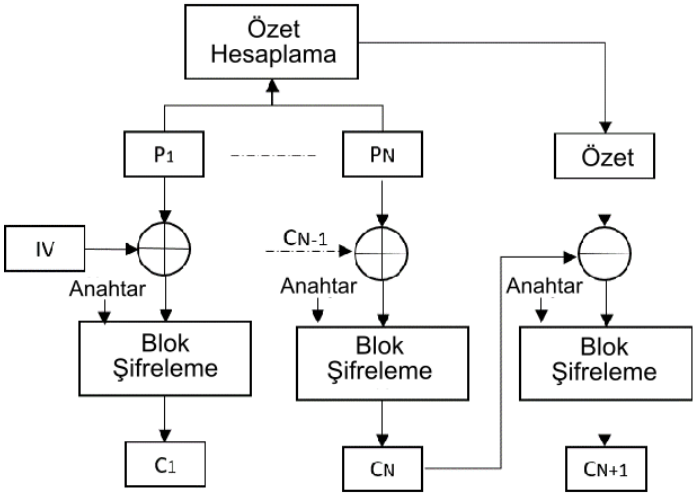


Şekil 3.13. Kimlik doğrulama mekanizması (Toğay 2019)

3.4.3. AES-128 ile Şifreleme ve Mesaj Bütünlüğü

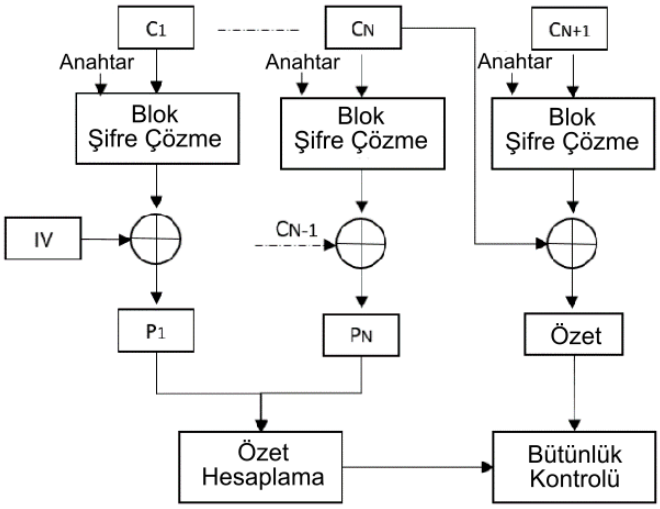
Kimlik doğrulama sonrasında, Broker ve ağ geçidinde oluşturulan SK ve IV bir MQTT verilerinin şifrelenmesi veya şifreli içeriğin çözülmesinde kullanılır. TÜBİTAK projesi kapsamında tasarlanan Broker ile cihazlar arasındaki iletişim şifreli bir şekilde gerçekleştirilir. Her cihaz ya da uygulamanın oturum anahtarı farklıdır. Örneğin, Ağ Geçidi veriyi şifreleyerek mesajı Broker'a gönderir. Şifreli veri Broker tarafından çözülür. Daha sonra Broker ilgili abonenin oturum anahtarını kullanarak veriyi şifreler ve aboneye iletir.

Veri güvenliğinin bir diğer önemli kısmı da veri bütünlüğünün sağlanmasıdır. Bazı kriptografik fonksiyonlar verinin özetini almak için kullanılabilir. Alıcı tarafından değiştirilmeden elde edilen özet, verinin bütünlüğünü doğrulamak için kullanılır. Bu nedenle özet değeri alıcıya güvenli bir şekilde ulaştırılmalıdır. Özet, bir etiket değeri olarak verinin sonuna eklenebilir ve farklı bir anahtarla şifrelenebilir. Ancak çalışmada, performans sorunları nedeniyle aynı anahtar kullanılmıştır. Özet değeri, MD-5, SHA-1, SHA-2 veya Döngüsel Artıklık Kontrolü (CRC) gibi algoritmalarla hesaplanabilir. MD-5 doğrudan 128 bit veri üretir. CRC algoritması, sırasıyla CRC-32 ve CRC-64 için 32 ve 64 bit veri üretir. Bazı özel mikrodenetleyiciler, CRC ve özet algoritmaları donanımsal olarak gerçekleştiren çevrebirimlerine sahiptir. Buradaki yaklaşım maliyet, performans ve donanım sorunları gibi parametrelere dayalı olarak tasarlanmıştır (Toğay 2019).
128 bitlik N adet bloklar halindeki veriler ve bu blokların özet değeri, oturum IV değeri ve SK ile şifrelenir (Şekil 3.14). Elde edilen 128 bitlik şifreli bloklar (C_1, C_2, .., C_N ve C_{N+1}) alıcıya gönderilir.



Şekil 3.14. AES-128 şifreleme (Toğay 2019)

Alıcı, gelen 128 bitlik şifreli blokların (C_1, C_2, .., C_N ve C_{N+1}) IV değeri ve SK ile şifresini çözer (Şekil 3.15). Tüm blokların şifresi çözüldüğünde özet değeri alıcı tarafından tekrar hesaplanır ve gönderici tarafından hesaplanan özet değeri ile doğrulanır. Böylece, veri aktarımı sırasında şifreli veri değiştirildiğinde bu durum alıcı tarafından tespit edilebilir. Burada hangi özet algoritmasının kullanılacağı işlemci, maliyet ve performans gereksinimlerine bağlı olarak belirlenir.



Şekil 3.15. AES-128 şifre çözme **(**Toğay 2019)

4. BULGULAR ve TARTIŞMA

4.1. Güvenlik Değerlendirmesi

Nesnelerin interneti sistemlerinde, güvenliğin sağlanmasına yönelik olarak kimlik doğrulama ve şifreli veri aktarımı konularında araştırma çalışmaları yapılmaktadır (Toğay 2019) (Andy 2017) (Chowdhury 2018) (Naik ve Maral 2018). Saldırganlar, herhangi bir zaman aralığında iletim hattına erişim sağladığında şifreli metni ve nonce gibi kurulum parametrelerini dinleyebilir. Ancak mesaj içeriğinin elde edilememesi ve mesaj hakkında varsayımda bulunulmaması temel kabul ve gereksinimlerdir. Buna bağlı olarak geliştirilen yönteme ait güvenlik risk değerlendirmesi maddeler halinde aşağıda sunulmuştur:

- Kullanıcı adı veya cihaz kimliği, (Naik ve Maral 2018)'de tanımlandığı gibi gizli bir bilgi olarak değerlendirilebilir. Bu parametreler iletişim sırasında simetrik bir anahtarla şifrelenebilir. Ancak simetrik anahtar tüm cihazlarda aynı olacağından fiziksel saldırılar ile cihazdan anahtarın elde edilmesi durumunda tüm veri trafiği tehlikeye girebilir. Bu şekilde iletişim hattı dinlenerek elde edilen anahtarla şifreli verinin şifresi çözülebilir. Sunulan, çalışmada bir anahtar takası gerçekleştirilmemektedir ancak her kullanıcıya özel anahtarlar sunucuda yer almaktadır. Sadece, anahtarlar kullanıcı adına bağlı olarak veri tabanında sorgulandığından kullanıcı adı şifrelenmeden iletilmektedir.

- Kullanıcı adı ve şifre tabanlı yaklaşımlarda kimlik bilgilerinin periyodik olarak değiştirilmesi gerekir (Andy 2017). Şifreler cihazlardan elde edilebilir. Bu nedenle her cihazın benzersiz bir şifresi olmalıdır. Önerilen yaklaşımda tüm cihazların benzersiz anahtarları vardır. Ancak anahtarlar farklı olsa bile tekrarlama saldırılarına maruz kalabilirler. Tekrarlama saldırılarını önlemek için bir sayaç eklenebilir. Uygulamada, bu değerlerin bakımı bir sorun olabilir. Örneğin, sayaçlar veya cihazın saati yeniden başlatıldığında eşler arasındaki senkronizasyon bozulabilir. ATAES132A entegresi ile güvenilir bir sayaç mekanizması kullanılabilir. Entegre, her anahtarla ilişkili yüksek dayanıklı sayaçlar sağlar. Sayaçlar, söz konusu anahtarın her kullanımında (şifreleme, şifre çözme vb.) artırılır. Sayaç değeri daha sonra şifre değerine dahil edilebilir. Bu

şekilde, Broker önceki mesajları takip edebilir ve daha önce işlenmiş olan verileri reddedebilir. Önerilen bu yaklaşım tekrarlama saldırıları açısından güvenlidir. Saldırgan, tekrarlama saldırısını üç şekilde başlatabilir:

- Bir oturum sırasındaki tüm paketler kimlik doğrulama ve dinlenen veri paketlerini içerebilir. Nonce değeri sayacın eski değerini belirttiğinden, yetkilendirme başarısız olur.
- Saldırgan dinlediği kimlik doğrulama paketleri ile etkin bir saldırı başlatabilir. Bu durumda yetkilendirilebilir. Ancak saldırganın K_2'yi bilmediği varsayılırsa mesajları şifreleyemez ve şifresini çözemez. Saldırgan tarafından bir mesaj gönderildiğinde Broker mesajın değiştirildiğini veya kırıldığını (bkz. Şekil 3.13) değerlendirerek yöneticiler için bir alarm oluşturulabilir.
- Bir oturum sırasında dinlenen herhangi bir paket Broker'a gönderilebilir. Her mesaj ve MAC değeri CBC modda şifrelendiğinden önceki paketin herhangi bir şekilde değiştirilmesi veya tekrar edilmesi (bkz. Şekil 3.13) mesaj bütünlüğünün bozulmasına neden olur.

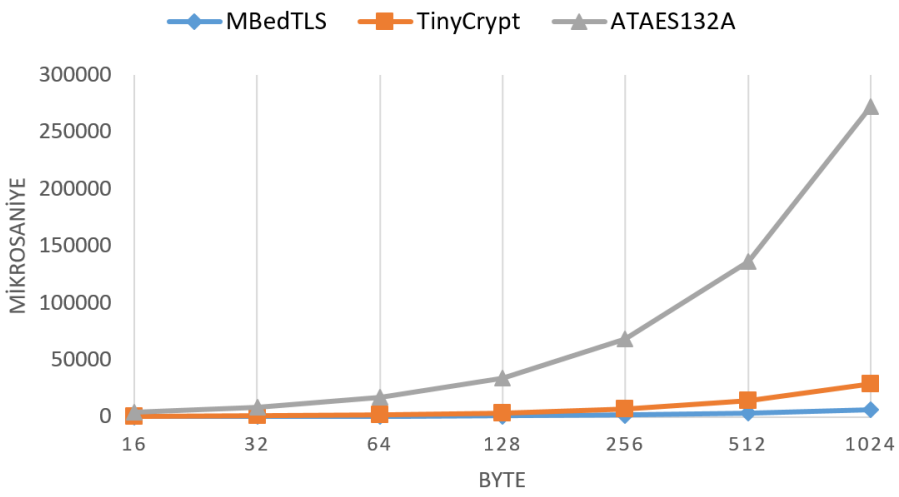
- Bağlantılar için farklı anahtarlar ve IV değerleri kullanılmalıdır. Her bağlantı için iki anahtar kullanılır ve bunların sayaçları yukarıda ifade edildiği üzere ATAES132A entegresinde arttırılır. Bir anahtarın kullanım sayacı sınıra ulaştığında farklı bir anahtar kullanılmalıdır. Saldırgan bu durumdan faydalanarak Broker olarak davranabilir ve cihazın yeniden bağlanmasını sağlamak için cihaza hata mesajları gönderebilir. Cihaz bağlantıyı denemeye devam ettiği sürece anahtarları kullanır. Sonuç olarak Broker'a bağlanacak anahtar kalmayacaktır. Bu nedenle cihaz artık Broker'a bağlanamaz. Bunun bir önlemi olarak yeniden bağlantı için bir süre beklenebilir. Bu şekilde anahtarların kısa sürede tüketilmesi engellenmiş olur.
- Broker varsayılan olarak 1883 numaralı bağlantı portunu dinler. Herhangi bir TCP soket uygulaması, Broker'a bağlantı açabilir. Bu nedenle, Broker için DDoS saldırıları mümkün olabilir. Bunun için güvenlik duvarı kullanımı değerlendirilebilir.
- Önerilen yaklaşımda, oturum anahtarı AES algoritması ve rastgele sayılar aracılığıyla üretilir. Rastgele sayı üretimi için sözde sayı üreteci (PRNG) kullanılabilir (Petit ve ark. 2007). PRNG'nin çekirdek değeri rastgele olmalıdır.

Ancak kriptografik işlemlerde, gerçek rastgele sayıların kullanılması önemlidir. Bu nedenle, harici entegredeki gerçek rastgele sayı üreteci (TRNG) kullanılmıştır. Bazı özel mikrodenetleyiciler donanımsal olarak rasgele sayı üreteçlerine sahiptir.

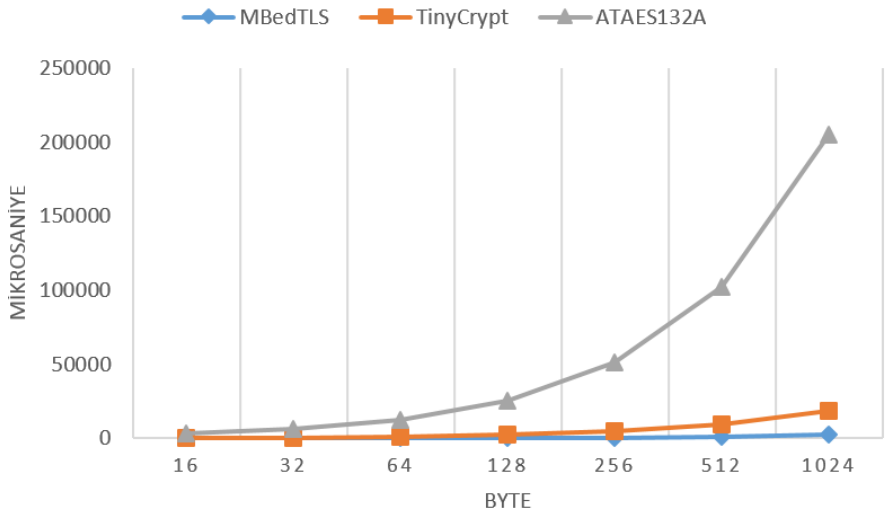
- Yazılım tabanlı AES uygulamaları; zamanlama, güç analizi ve yan kanal saldırılarına karşı savunmasızdır. (Schwabe ve Stoffelen 2016). ATAES132A entegresi bu tür saldırılar için korumalara sahiptir. Ağ geçidinin veri transfer perfomansı, ATAES132A entegresine yakın olursa, yazılım uygulamaları yerine verileri şifrelemek için entegre kullanılabilir.
- Özet uzunluğu 32 bitten daha fazla olabilir, ancak 64 bitten az olması pek önerilmemektedir (Dworkin 2004). Ancak mikrodenetleyicinin donanımsal CRC birimi olması durumunda 32 bit CRC tercih edilebilir. Performans bölümünde ifade edildiği üzere, CRC 64 bit ve MD5'in yazılım uygulamaları birbirlerine yakın verime sahiptir. Performans ve mikrodenetleyici'ye bağlı olarak tercih yapılabilir.

4.2. Şifreleme Performansı

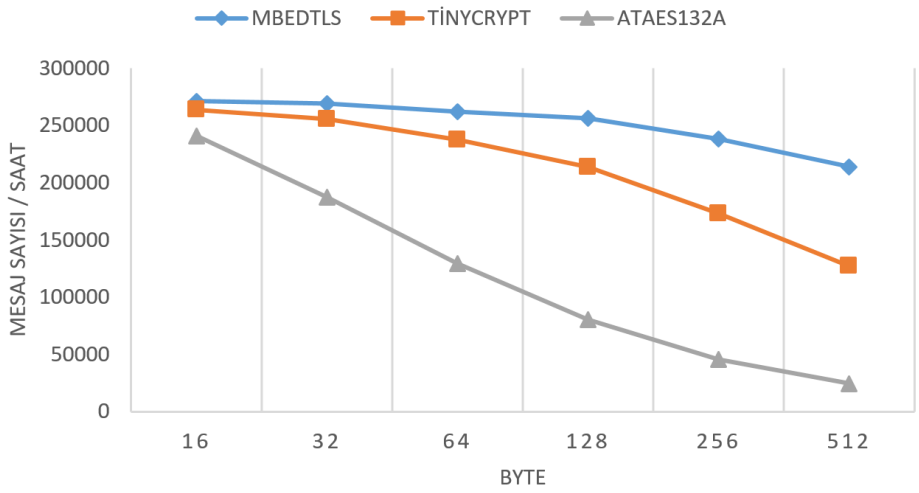
Karşılaştırmak amacıyla, tasarlanan ağ geçidi ve NUCLEO-144 geliştirme kartı ile mbedTLS ve TinyCrypt olmak üzere iki yazılım uygulaması test edilmiştir. Ağ geçidi ARM Cortex-M0'da, 1024 bayt için mbedTLS, TinyCrypt ve ATAES132A ile yapılan şifreleme işleminde sırasıyla 6272, 28912 ve 272742 mikrosaniye zaman harcanmıştır (Şekil 4.1). Geliştirme kartı ARM Cortex-M3'de 1024 bayt için mbedTLS, TinyCrypt ve ATAES132A ile yapılan şifreleme işleminde sırasıyla 2543, 18670 ve 204800 mikrosaniye zaman harcanmıştır (Şekil 4.2). Şekil 4.1 ve Şekil 4.2'de gösterildiği gibi, ARM Cortex-M3, ARM Cortex-M0'dan yaklaşık iki kat daha iyi performansa sahiptir. Bu değerlere ve MQTT mesaj yayınlama sürelerine dayanarak, cihazların veri iletim performansları Şekil 4.3 ve Şekil 4.4'de gösterilmektedir. ARM Cortex-M3'ün (STM32F207VGT6) fiyatının, ARM Cortex-M0'a (W7500P) göre daha pahalı olması nedeni ile ağ geçitlerinin tasarımında fiyat-performans dengesi gözetilmelidir.



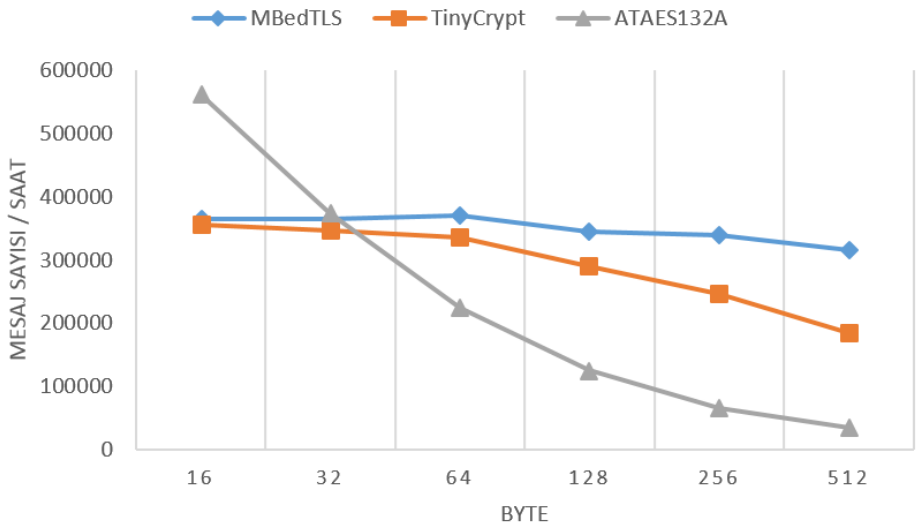
Şekil 4.1. ARM Cortex –M0 için AES-128 şifreleme (Toğay 2019)



Şekil 4.2. ARM Cortex –M3 için AES-128 şifreleme **(**Toğay 2019)



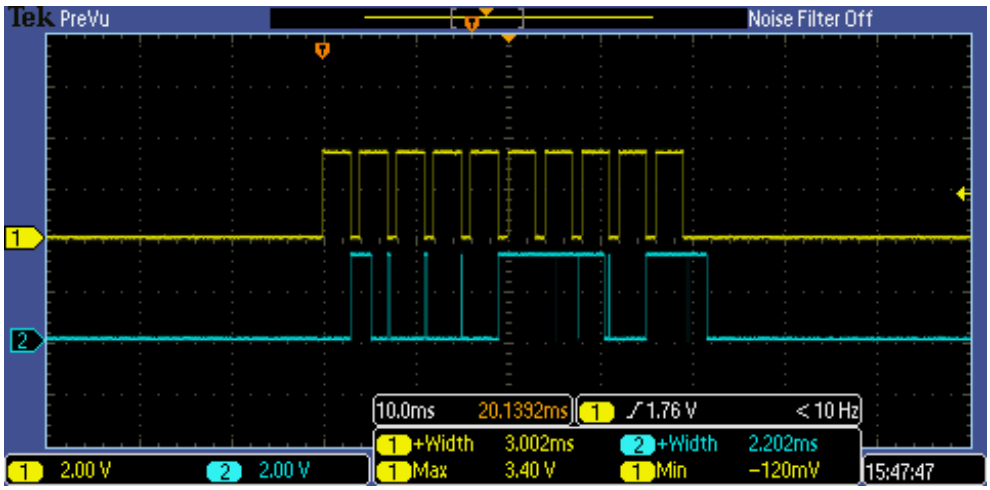
Şekil 4.3. ARM Cortex –M0 için şifreleme performansı **(**Toğay 2019)



Şekil 4.4. ARM Cortex-M3 için şifreleme performansı **(**Toğay 2019)

Yazılım şifreleme kütüphaneleri (mbedTLS ve TinyCrypt) işlem süresince mikrodenetleyiciyi meşgul eder. Bu nedenle, W7500P ile tasarlanan ağ geçidinde şifreleme ile mesaj yayınlamayı birlikte yürütmek mümkün değildir. Ancak, NUCLEO-144 geliştirme kartında donanımsal I^2C birimi bulunduğundan MQTT iletişimi ve AES şifrelemesi, Şekil 4.5'de gösterildiği gibi aynı anda gerçekleştirilebilir. Çalışmada uygulamaların performansını ölçmek için bir Tektronix MSO 2014 dijital osiloskop kullanılmıştır.

MQTT verisi gönderilirken, ATAES132A entegresi birkaç yeni veriyi şifreleyebilir. Şifreleme işlemi yaklaşık 3200 mikrosaniye sürmektedir. ATAES132A entegresi, MQTT işlemlerinin 3200 mikrosaniyeden daha fazla sürdüğü yerlerde etkili bir şekilde kullanılabilir. Bu durumda, şifreleme maliyeti sıfıra düşer. Verimlilik, blok boyutundan ve ağ gecikmesinden etkilenir. ARM Cortex-M3'e sahip geliştirme kartı, şifreleme ve mesaj yayınlama işini eş zamanlı olarak yürütebildiğinden, ağ gecikmesinin 3200 mikrosaniyeden fazla olduğu ve blok boyutunun Şekil 4.5'de gösterildiği gibi küçük olduğu durumlarda verim yüksek olabilir. Bu nedenle, harici entegre tabanlı çözüm, mikrodenetleyicinin donanımsal I^2C birimine sahip olduğu ve iletişim gecikmesinin yüksek olduğu durumlarda şifreleme verimini arttırır.



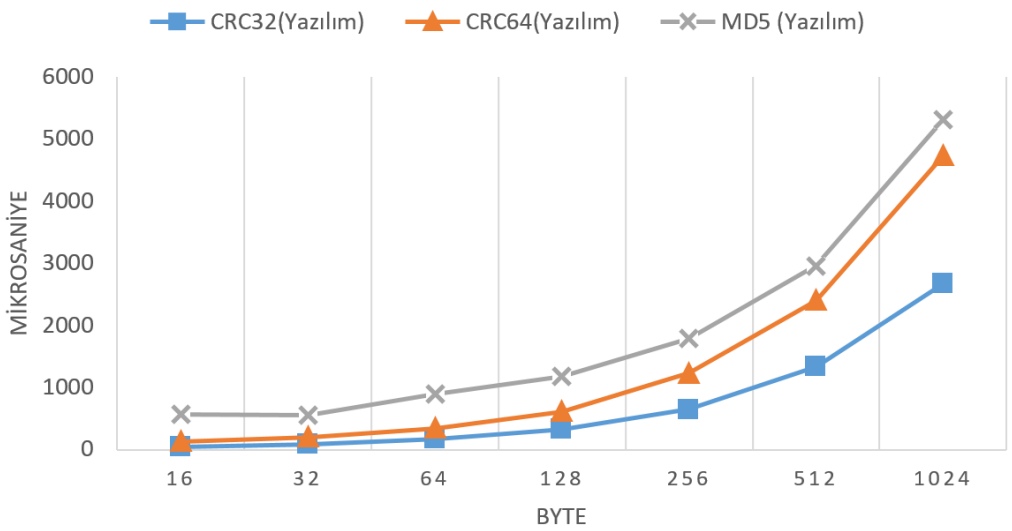
Şekil 4.5. Şifreleme (Kanal 1) ve Mesaj İletimi (Kanal 2) **(**Toğay 2019)

Çalışmada, test işlemleri uzak bir Broker üzerinden yürütülmüştür. Ağ geçidi 1024 byte (64 blok) şifreli veriyi Broker'a 3425 ile 8104 mikrosaniye arasında göndermektedir. Performans değerlendirmesine göre ortalama yayın süresi 6400 mikrosaniye olarak kabul edilmiştir. Bu durumda MQTT veriminin mikrodenetleyici hızından, ağ gecikmesinden ve Broker'ın performansından etkilendiği belirtilmelidir.

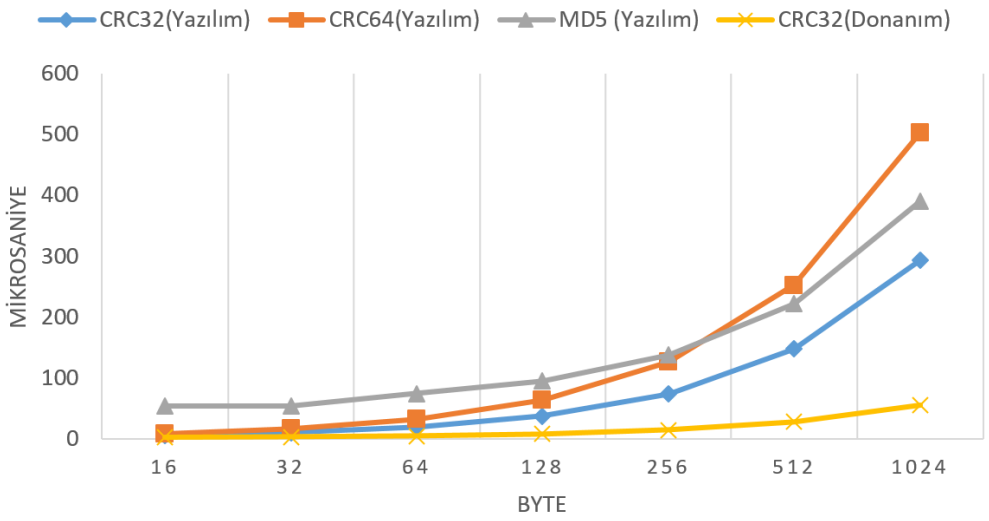
4.3. Mesaj Bütünlüğü Performansı

Önerilen yaklaşımda, mesajlardaki değişiklikler özet algoritmalarının kullanılması ile tespit edilebilir. Daha önce ifade edildiği üzere tüm mesaj çözüldükten sonra elde edilen

özet değeri ile mesajın içerisinde şifreli bir şekilde gelen özetin aynı olması beklenmektedir. Özet algoritmasında özetin uzunluğu ve algoritmanın çakışmalara karşı etkinliği güvenlik kapsamında önem kazanmaktadır. MD5, saldırılara karşı CRC-32 ve CRC-64'den daha güvenlidir. Testlerimiz sırasında yazılımsal ve donanımsal özet algoritmaları değerlendirilmiştir. Ağ geçidi ARM Cortex-M0'da 256 bayt verinin özet değeri için CRC-32 (yazılım), CRC-64 (yazılım) ve MD5 (yazılım) ile yapılan işlemlerin süreleri sırasıyla 648, 1228 ve 1787 mikrosaniyedir (Şekil 4.6). MD5 algoritması büyük boyutlu mesajlarda, güvenlik ve performans açısından CRC-64 yerine tercih edilebilir. Bu iki algoritma için bellek tüketimi ise Çizelge 4.1'de gösterildiği üzere birbirlerine çok yakındır. Bazı özel mikrodenetleyiciler donanımsal olarak CRC-32'ye sahiptir. Bu durumda, performans ve bellek boyutundan yararlanmak için CRC32 kullanılabilir. Geliştirme kartı ARM Cortex-M3'de 256 bayt verinin özet değeri için CRC-32 (donanım), CRC-32 (yazılım), CRC-64 (yazılım) ve MD5 (yazılım) ile yapılan işlemlerin süreleri sırasıyla 15, 74, 126 ve 137 mikrosaniyedir (Şekil 4.7). Verimliliğin gerekli olduğu durumlarda CRC-32 donanım uygulaması seçilebilir. STMicroelectronics firmasının STM32F217VG (Anonim 2016) gibi bazı modellerinde ayrıca AES, MD5 ve SHA-1 algoritmaları dahili olarak bulunur. Algoritmalar ve uygulamalar mikrodenetleyiciye ve istenilen verime bağlı olarak tanımlanmalıdır.



Şekil 4.6. ARM Cortex-M0 için
mesaj doğrulama algoritmalarının performansı **(**Toğay 2019)



Şekil 4.7. ARM Cortex-M3 için
mesaj doğrulama algoritmalarının performansı **(**Toğay 2019)

Algoritma İsmi	Toplam Flash Bellek Boyutu (Byte)
CRC32	1064
CRC64	2099
MD5	2508

Çizelge 4.1. Özet algoritmalarının Flash bellek boyutu (Toğay 2019)

5. TARTIŞMA ve SONUÇ

Bu tez çalışması ile uygun maliyetli ve güvenli bir IIoT ağ geçidinin geliştirilmesi amaçlanmıştır. Ağ geçidi, Modbus protokolüne sahip çeşitli cihazlardan toplanan verileri, işlenmek üzere bir Broker üzerinden ilgili birime (sunucu uygulaması, bir başka cihaz vb.) gönderir. MQTT, IIoT sistemlerinde TCP/IP üzerinden açık ya da TLS protokolü ile şifreli bir şekilde veri transferine olanak sağlayan standart haberleşme protokolüdür. Çoğu fabrika ortamında işlem gücünden tasarruf ve hız nedenleri ile veriler açık bir şekilde iletilmektedir. Ancak, verilerin açık bir şekilde (şifrelenmemiş olarak) gönderilmesi bilgilerin çalınmasına/değiştirilmesine bağlı olarak canlı hayatına mal olabilecek çok ciddi sıkıntıların oluşmasına neden olabilir. Örneğin, bir tavuk çiftliğinde; sıcaklık sensöründen veri toplayıp, bunu bir karar uygulamasına MQTT protokolü ile ileten bir cihazın gönderdiği verinin yolda değiştirilmesi sonucunda alınacak karar (ortam sıcaklığının çok düşük ya da çok yüksek hale getirilmesi) ile yüzlerce/binlerce tavuğun ölümüne neden olunabilir. MQTT ile şifresiz iletim sırasında cihazın kullanıcı adı ve parolası elde edilebilir. Buna bağlı olarak farklı bir cihazdan aynı kullanıcı bilgileri ile farklı veriler gönderilerek karar alma uygulamaları ya da bilgileri izleyen kullanıcıların izleme ekranlarında yanlış/tutarsız verilerin gösterimi sağlanabilir. Yangın varken kullanıcının yangın durumunu ekranında görememesi gibi durumlar ortaya çıkabilir. Dolayısı ile firewall arkasında dış ağdan yalıtılmış dahi olsa verilerin şifreli bir şekilde aktarılması güvenlik açısından bir gerekliliktir. Şifreli iletişimde AES ya da benzer bir algoritma ile veriler bir anahtar kullanılarak şifrelenir. Karşı tarafta aynı anahtarın olması durumunda içerik çözülerek işlenir. Dolayısı ile uçlar arasında bir anahtar takas yöntemine ihtiyaç vardır. Anahtarların cihazdan kolaylıkla elde edilebilecek olması nedeni ile paylaşımlı simetrik anahtarlarının kullanılması tercih edilmemektedir. Bu kapsamda asimetrik şifreleme algoritmaları ile anahtar takası gerçekleştirilebilir. IIoT cihazlarının genel olarak kısıtlı cihazlardan oluşması nedeni ile TLS'in son versiyonlarının uygulanması çoğu zaman bellek ve işlemci kısıtları nedeni ile mümkün olmamaktadır. Örnek olarak TLS 1.0 ile Amazon'un Broker'larına bağlanılamamaktadır. TLS'de anahtarın oluşturulması için RSA ya da ECC gibi asimetrik şifreleme algoritmalarının kullanılmasına ihtiyaç vardır. RSA algoritmasının uygulandığı cihazlarda RSA kütüphanesi ve RSA açık anahtarı ya da parmak izi (fingerprint) gömülü

cihazın geçici olmayan belleğinde saklanmalıdır. Bu nedenle TLS tabanlı anahtar değişimi kısıtlı cihazlar için uygun değildir. Bu çalışmada harici bir entegrede (ATAES132A) saklanan önceden paylaşılmış anahtarlardan yararlanılmıştır. Bu anahtarlar hem kimlik doğrulama hem de mesajın şifrelenmesi ve bütünlüğü için kullanılmıştır. Entegrenin simetrik anahtarları güvenli bir ortamda saklaması nedeni ile anahtarlar doğrudan elde edilememektedir. ATAES132A'nın kullanıldığı senaryolarda AES şifreleme algoritması, cihazın FLASH ve RAM belleğinde yer kaplamaması ve uygulama kodu için daha fazla bellek ayrılması nedeni ile tercih edilebilir. ATAES132A'nın bir diğer faydası da standartlara uygun rastgele sayı üreticisine sahip olmasıdır. Bu şekilde simetrik anahtarların belirlenmesinde kripto analiz yöntemlerine karşı olarak önemli bir konu olan rastgele olma gereksinimi sağlanmıştır. ARM Cortex-M3 gibi bazı mikrodenetleyicilerin barındırdıkları donanımsal I^2C birimi sayesinde tezde ve (Toğay 2019)'de ifade edildiği üzere MQTT paketlerinin gönderimi sırasında Slave cihazlardan veri toplama ve MQTT mesaj kuyruğuna ekleme işlemi aynı zamanda gerçekleştirilebilmektedir.

Gelecekteki çalışmalarda, işlemci ve entegre arasında daha verimli iletişim için ATAES132A'nın diğer yazılım uygulamalarının ve SPI bağlantı arayüzlerinin kullanımı değerlendirilebilir. Ayrıca, AES, MD5, SHA-1 gibi şifreleme algoritmalarını içeren ARM Cortex-M3 (STM32F217VGT6) kullanılabilir.

KAYNAKLAR

Anonim, 1996. Modicon Modbus Protocol Reference Guide. http://modbus.org/docs/PI_MBUS_300.pdf-(Erişim tarihi: 14.02.2019).
Anonim, 2001a. FIPS 197: Announcing the ADVANCED ENCRYPTION STANDARD (AES). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf-(Erişim tarihi: 10.05.2019).
Anonim, 2001b. RS-232. https://en.wikipedia.org/wiki/RS-232-(Erişim tarihi: 14.06.2019).
Anonim, 2011. 3.3 V Slew Rate Limited, Half- and Full-Duplex, RS-485/RS-422 Transceivers Datasheet. https://www.analog.com/media/en/technical-documentation/data-sheets/ADM3483_3485_3488_3490_3491.pdf-(Erişim tarihi: 02.04.2019).
Anonim, 2012. IP101G Preliminary Data Sheet. http://www.bdtic.com/DataSheet/ICplus/IP101G_DS_R01_20121224.pdf-(Erişim tarihi: 05.06.2019).
Anonim, 2015a. W7500P Datasheet Manual. https://cdn.sos.sk/productdata/11/0a/241b553d/w7500p.pdf-(Erişim tarihi: 18.06.2019).
Anonim, 2015b. WIZNET Internet Offload Processor. https://www.wiznet.io/wp-content/uploads/2015/08/wiznet_ebrochure_iop4iot.pdf-(Erişim tarihi: 02.04.2019).
Anonim, 2016. STM32F215xx, STM32F217xx Datasheet. https://www.st.com/resource/en/datasheet/stm32f217vg.pdf-(Erişim tarihi: 15.04.2019).
Anonim, 2017a. MAX3232 3-V to 5.5-V Multichannel RS-232 Line Driver/Receiver With ±15 kV ESD Protection Datasheet. http://www.ti.com/lit/ds/symlink/max3232.pdf-(Erişim tarihi: 14.04.2019).
Anonim, 2017b. UM1974 User manual STM32 Nucleo-144 boards. https://www.st.com/content/ccc/resource/technical/document/user_manual/group0/26/49/90/2e/33/0d/4a/da/DM00244518/files/DM00244518.pdf/jcr:content/translations/en.DM00244518.pdf -(Erişim tarihi: 20.07.2019).
Anonim, 2018. ATAES132A 32K AES Serial EEPROM Complete Datasheet. http://ww1.microchip.com/downloads/en/DeviceDoc/ATAES132A-Data-Sheet-40002023A.pdf-(Erişim tarihi: 17.01.2019).
Anonim, 2019a. ACTIVEMQ. https://activemq.apache.org/-(Erişim tarihi: 12.08.2019).
Anonim, 2019b. HIVEMQ. https://www.hivemq.com/-(Erişim tarihi: 02.08.2019).
Anonim, 2019c. TPD4E001 Low-Capacitance 4-Channel ESD-Protection for High-Speed Data Interfaces Datasheet. http://www.ti.com/lit/ds/symlink/tpd4e001.pdf-(Erişim tarihi: 09.08.2019).
Anonim, 2019d. STM32F205xx, STM32F207xx Datasheet. https://www.st.com/resource/en/datasheet/stm32f207zg.pdf-(Erişim tarihi: 02.08.2019).

Andy, S., Rahardjo, B., Hanindhito, B. 2017. Attack scenarios and security analysis of mqtt communication protocol in iot system. International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 4(September):, 600–604.
Banks, A., Gupta, R. 2014. MQTT Version 3.1.1.
Bassham, L. E. 2002. The Advanced Encryption Standard Algorithm Validation Suite (AESAVS). http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf-(Erişim tarihi: 15.05.2019).
Dworkin, M. 2004. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38c.pdf-(Erişim tarihi: 11.07.2019).
Gilchrist, A. 2016. Industry 4.0 The Industrial Internet of Things.
Hall, J. 2013. C Implementation of Cryptographic Algorithms. http://www.ti.com/lit/an/slaa547b/slaa547b.pdf-(Erişim tarihi: 05.08.2019).
Hollenbeck, R. 2001. The IEEE 802.3 Standard (Ethernet): An Overview of the Technology. http://blog.gitdns.org/2016/06/17/ethernet-mac-phy/Ethernet.pdf-(Erişim tarihi: 05.08.2019).
Kugelstadt, T. 2016. The RS-485 Design Guide. https://en.wikipedia.org/wiki/RS-232-(Erişim tarihi: 07.06.2019).
McConahay, J. 2011. Using Modbus for Process Control and Automation. *Control Engineering*, A12–A14.
Petit, C., Standaert, F.-X., Pereira, O., Malkin, T., ve Yung, M. 2007. A Block Cipher based PRNG Secure Against Side-Channel Key Recovery. , 1–22.
Schwabe, P., Stoffelen, K. 2016. All the AES You Need on Cortex-M3 and M4. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10532 LNCS:, 180–194.
Simmons, M. 2008. Ethernet Theory of Operation. http://ww1.microchip.com/downloads/en/appnotes/01120a.pdf-(Erişim tarihi: 07.06.2019).
Toğay, C. 2017. A New Message Processing Mechanism for Internet of Things, Uludağ University Journal of The Faculty of Engineering, Vol 23, Issue 2, pp. 55-66.
Toğay, C., Mutlu, G., Kurtuluş, D., Özgür, F. 2019. Secure Gateway for the Internet of Things, (16):, 414–426.
Wardhani, R. W., Ogi, D., Syahral, M., Dedy Septono Catur, P. 2017. Fast implementation of AES on Cortex-M3 for security information devices. QiR 2017 - 2017 15th International Conference on Quality in Research (QiR): International Symposium on Electrical and Computer Engineering, 2017–December:, 241–244.

ÖZGEÇMİŞ

Adı Soyadı : Gökhan MUTLU
Doğum Yeri ve Tarihi : ÇANAKKALE / 1992
Yabancı Dil : İngilizce

Eğitim Durumu
Lise : İbrahim Bodur Anadolu Lisesi, 2007
Lisans : Bursa Uludağ Üniversitesi, 2010
Yüksek Lisans : Bursa Uludağ Üniversitesi, 2016

Çalıştığı Kurum/Kurumlar : Emko Elektronik, 2015

İletişim (e-posta) : ggokhan.mutlu@gmail.com

Yayınları :

Toğay, C., Mutlu, G., Kurtuluş, D., Özgür, F. (2019). Secure Gateway for the Internet of Things. European Journal of Science and Technology, (16), 414-426.