NÖHÜ Müh. Bilim. Derg. / NOHU J. Eng. Sci., 16 (2026), 1794041
Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi
Niğde Ömer Halisdemir University Journal of Engineering Sciences
Araştırma makalesi / Research article
www.dergipark.org.tr/tr/pub/ngumuh / www.dergipark.org.tr/en/pub/ngumuh

Nesnelerin interneti (IoT) uygulamalarında güvenlik ve gizliliğe duyarlı veri iletiminin enerji tüketimi

Energy consumption of security and privacy-preserving data transmission in IoT applications

Emirhan Uğurlu[1,*], Nazlı Tekin[2]

[1] *Erciyes Üniversitesi, Bilgisayar Mühendisliği Bölümü, 38260, Kayseri Türkiye*
[2] *Erciyes Üniversitesi, Yazılım Mühendisliği Bölümü, 38260, Kayseri, Türkiye*

Öz

Bu çalışma, Nesnelerin İnterneti (IoT) uygulamalarında güvenli ve gizliliğe duyarlı veri iletiminin enerji maliyetini incelemektedir. Çalışmada, güvenlik ve gizliliği sağlamak amacıyla kullanılan kriptografik algoritmaların IoT cihazlarında neden olduğu enerji tüketimi, deneysel ölçümlerden elde edilen nicel analizlere dayanarak değerlendirilmiş ve karşılaştırılmıştır. Deneysel düzende, gerçek sensör verilerine benzer özellikler taşıyan veri blokları kullanılmıştır. Yazılım geliştirme Arduino IDE ortamında gerçekleştirilmiş olup ESP8266 NodeMCU ve Heltec ESP32 V3 IoT cihazları test platformu olarak seçilmiştir. Çalışmada AES, AESSmall, AESTiny, SPECK, SPECKSmall, SPECKTiny, ChaCha, PRESENT, ACORN128 ve ASCON128 algoritmaları incelenmiştir. Her bir algoritma için şifreleme ve şifre çözme işlemleri gerçekleştirilmiş, bu süreçte süreler kaydedilmiş ve güç tüketimi ölçülmüştür. Elde edilen veriler kullanılarak enerji tüketimleri hesaplanmış ve algoritmaların performansları karşılaştırmalı olarak analiz edilmiştir. Bulgular, algoritmalar ve cihazlar arasında belirgin performans farklılıklarını ortaya koymaktadır. ESP8266 üzerinde Present, ESP32 üzerinde ise ACORN128 algoritması en düşük enerji tüketimini göstermiştir. Çalışma, IoT uygulamalarında güvenli ve enerji verimli veri iletimi için algoritma seçiminde yol gösterici niteliktedir.

Anahtar kelimeler: Nesnelerin interneti (IoT), Kriptografi, Şifreleme algoritmaları, Enerji tüketimi

Abstract

This study investigates the energy cost of secure and privacy-preserving data transmission in Internet of Things (IoT) applications. The energy consumption introduced by cryptographic algorithms employed to ensure security and privacy is evaluated and compared through quantitative analysis based on experimental measurements on IoT devices. In the experimental setup, data blocks with properties resembling real sensor outputs were utilized. Software development was carried out in the Arduino IDE environment, and ESP8266 NodeMCU and Heltec ESP32 V3 IoT devices were selected as the test platforms. The study examined AES, AESSmall, AESTiny, SPECK, SPECKSmall, SPECKTiny, ChaCha, PRESENT, ACORN128, and ASCON128 algorithms. For each algorithm, encryption and decryption operations were executed, execution times were recorded, and power consumption was measured. Using these data, the energy consumption of the algorithms was calculated, and their performance was analyzed in a comparative manner. The results reveal significant performance differences across algorithms and devices. Present demonstrated the lowest energy consumption on the ESP8266, while Acorn128 showed the most efficient performance on the ESP32. The findings provide guidance for algorithm selection to achieve secure and energy-efficient data transmission in IoT applications.

Keywords: Internet of Things (IoT), Cryptography, Encryption algorithms, Energy consumption

1 Giriş

Nesnelerin İnterneti (Internet of Things – IoT) fiziksel nesnelerin internet aracılığıyla birbirleriyle iletişim kurmasını, kontrol edilmesini ve veri akışında bulunmasını sağlayan bir sistemdir. Günümüzde IoT uygulamaları, akıllı tarımda sulama ve ürün takibi [1], akıllı şehirlerde trafik yönetimi, aydınlatma ve enerji verimliliği çözümleri [2] ve sınır güvenliğinde hareket algılama ve gözetim sistemleri [3] gibi birçok farklı alanlarda yaygın olarak kullanılmaktadır.

Bu geniş kullanım yelpazesi, IoT uygulamalarının toplumsal, ekonomik ve endüstriyel süreçlerde kritik bir rol üstlenmesini sağlamaktadır. IoT cihazlarının kullanımındaki hızlı artış, verilerin güvenli ve verimli taşınmasını daha da önemli hale getirmektedir. Ancak, IoT cihazlarının açık kablosuz ağlar üzerinden haberleşmesi onları çeşitli saldırılara karşı savunmasız bırakmaktadır. Özellikle dinleme (eavesdropping), veri manipülasyonu, kimlik sahtekârlığı ve yetkisiz erişim gibi saldırılar, IoT uygulamalarında güvenlik ve gizlilik tehditlerini artırmaktadır [4].

Veri şifreleme yöntemleri, IoT uygulamalarında güvenliği artırmanın en temel çözümlerinden biridir.

* Sorumlu yazar / Corresponding author, e-posta / e-mail: emirhanugurlu06@icloud.com (E. Uğurlu)
Geliş / Received: 30.09.2025 Kabul / Accepted: 05.01.2026 Yayımlanma / Published: 26.01.2026
doi: 10.28948/ngumuh.1794041

Bununla birlikte, şifreleme algoritmalarının getirdiği hesaplama yükü, sınırlı işlem gücü, bellek ve enerji kapasitesine sahip IoT cihazları için önemli bir sorun oluşturmaktadır. Özellikle batarya ile çalışan cihazlarda, şifreleme işlemleri sırasında ortaya çıkan ek işlem yükü enerji tüketimini artırarak cihazın kullanım ömrünü kısaltmaktadır. Bu nedenle, IoT senaryolarında güvenlik ve enerji tüketimi arasında kaçınılmaz bir denge (trade-off) söz konusudur.

Bu çalışmada, farklı şifreleme algoritmalarının kaynak kısıtlı ve batarya ile çalışan IoT cihazları üzerindeki enerji tüketimi karşılaştırmalı olarak incelenmiştir. Deneysel ortamda gerçekleştirilen ölçümlerle şifreleme süresi, güç tüketimi ve toplam enerji tüketimi hesaplanarak, güvenlik ve gizliliğin sağlanmasının enerji maliyeti ortaya konulmuştur. Çalışmanın temel amacı, IoT uygulamalarında kullanılan şifreleme algoritmalarının sınırlı işlem gücüne sahip cihazlardaki performansını değerlendirmek ve güvenlik–enerji dengesi üzerine bulgular sunmaktır. Bu kapsamda, farklı IoT cihazlarında çeşitli şifreleme algoritmaları uygulanmış; elde edilen ölçümler ve hesaplamalar üzerinden karşılaştırmalı değerlendirmeler gerçekleştirilmiştir.

Bu çalışmada, şifreleme algoritmalarının blok boyutları esas alınarak üretilen veri setleri kullanılmış ve farklı algoritmaların IoT cihazları üzerindeki performansları deneysel olarak analiz edilmiştir. Blok boyutları, ilgili algoritmanın tek bir seferde şifreleyebileceği byte sayısını ifade etmektedir. Önerilen yaklaşım şu adımları içermektedir: (i) verilerin şifrelenmesi ve çözülmesi, (ii) süre ölçümlerinin yapılması, (iii) güç tüketiminin ölçülmesi ve (iv) toplam enerji tüketiminin hesaplanması. Deneyler ESP32 ve NodeMCU ESP8266 cihazları üzerinde yürütülmüş; batarya ile çalışan IoT cihazlarında güvenlik ve gizliliği sağlamaya yönelik hafif şifreleme algoritmalarının performansı ayrıntılı biçimde incelenmiştir.

Elde edilen sonuçlar, şifreleme algoritmalarının batarya ile çalışan IoT cihazlarında güvenlik ve gizlilik sağlama sürecinde nasıl bir enerji maliyeti yarattığını ortaya koymakta; mevcut güvenlik çözümlerinin pratik IoT senaryolarındaki uygulanabilirliğini değerlendirmeye yönelik karşılaştırmalı bulgular sunmaktadır.

Literatürdeki çalışmalar, gizlilik ve güvenlik odaklı veri iletiminde kullanılan geleneksel ve yenilikçi şifreleme algoritmaları arasındaki farklılıkları çoğunlukla göz ardı etmiştir. Önerilen çalışmada ise geleneksel (AES ve varyantları) ve yenilikçi (ChaCha, SPECK gibi) algoritmaların enerji tüketimi açısından karşılaştırılması yapılmıştır.

Indu Radhakrishnan vd. hafif kriptografi algoritmalarının IoT cihazlarında çalışma süresi, bellek kullanımı, gecikme süresi, veri aktarım hızı ve güvenlik performanslarını görmek için Arduino Nano ve Arduino Micro cihazları ile çalışmalar yapmışlardır [5].

Luke E. Kane vd. belirli kriptografi algoritmalarını karşılaştırdıkları makaleyi yayımlamışlardır. Makalede yaptıkları çalışmada ATMEGA328, STM32, ESP8266 cihazlarında AES, ChaCha, ACORN algoritmalarının performanslarını test etmişlerdir [6].

Rizwan Iqbal vd. geliştirdikleri algoritmanın hafif kriptografi algoritmaları içinde performansını değerlendirmek amacıyla çalışma yapmışlardır. Bu çalışmada Arduino Uno, Raspberry Pi 4, STM32 Nucleo cihazlarını kullanmışlardır [7].

Siranjeevi Srinivasa Raghavan tarafından yayımlanan makalenin amacı otomotiv sektöründe güvenlik ihtiyacını karşılayabilecek hafif şifreleme algoritmalarını analiz etmektir. Çalışma kapsamında simülatör üzerinde AES, RSA, Speck, ECC, PRESENT algoritmalarının performans-güvenlik dengesi test edilmiştir [8].

Jesús Soto-Cruz vd. IoT cihazlarının sahip olduğu sınırlı işlem gücü ve bellek kapasitesi özellikleri ile çalışırken güvenliğin arttırılması için simetrik şifreleme algoritmalarına ağırlık vermişlerdir. bu çalışmayı yaparken ESP32 Dev Module, NodeMCU ESP8266, MSP430, NUCLEO-G0B1RE, Arduino Nano cihazlarında PRESENT, SPECK, PICCOLO, CRAFT, HUMMINGIBIRD-2, ASCON, ACORN, LIZARD, FRUIT-80, TRIVIUM algoritmalarını kullanmışlardır [9].

Ievgeniia Kuzminykh vd. IoT cihazlarında, cihazın enerji tüketimi ve kullanım ömrünün incelenmesi adına Arduino Mega 2560 cihazında yaptıkları çalışmalarını anlattıkları makaleyi yayımlamışlardır. Makalede AES, XTEA, HIGHT, KLEIN, ECC, BLOWFISH, TWOFISH, SERPENT, PICCOLO, PRESENT algoritmaları ile çalışmışlardır [10].

Shapina Abdullah vd. IoT projelerinde güvenliğe yeterince dikkat edilmemesi üzerine SHA-256 ve AES-128 algoritmaları ile çalışmışlardır. Bu çalışmada Arduino Uno cihazı kullanılmıştır [11].

L. Sleem vd. Speck algoritması üzerinde çalışarak algoritmanın geliştirilmiş versiyonu olan SPECK-R algoritmasını tanıtmışlardır. Bu çalışmada ATmega328P, Teensy 3.6, DOIT ESP32 cihazlarında SPECK ve AES algoritmalarını test etmişlerdir [12].

M. Ramalho vd. yayımladıkları makalede IoT'de veri güvenliğine dikkat çekmişlerdir. Çalışmada afet önleme amacıyla kullanılan uzaktan çevresel izleme sistemi baz alınarak ESP8266 80 Mhz, ESP8266 160 Mhz, Arduino Uno cihazlarında AES, AESSmall, AESTiny, SPECK, SPECKSmall, SPECKTiny algoritmaları üzerinde çalışmışlardır [13].

Önerilen çalışmamızda ise, önceki çalışmalardan farklı olarak yalnızca standart algoritmalar değil, daha önce test edilmemiş varyantlar (AESSmall128, AESTiny128, ChaCha farklı tur sayıları, SPECKSmall, SPECKTiny) da değerlendirilmiştir. Ayrıca, bu algoritmalar günümüzde IoT projelerinde sıklıkla tercih edilen NodeMCU ESP8266 ve ESP32 cihazlarında uygulanmıştır. En önemlisi, yalnızca şifreleme süresi değil, aynı zamanda enerji tüketimi de doğrudan deneysel ölçümlerle incelenmiştir. Bu yönüyle çalışma, literatürdeki önceki araştırmalardan ayrışarak, güvenlik ile enerji verimliliği arasındaki dengeyi ortaya koyan özgün bir katkı sunmaktadır.

Literatürde yapılan çalışmalar, gizlilik ve güvenlik odaklı veri iletimi konusunda genellikle kullanılan şifreleme algoritmalarını tanıtmakta, ancak bu algoritmaların enerji tüketimi üzerindeki etkilerini kapsamlı biçimde karşılaştırmamaktadır.

Tablo 1. Literatürde yer alan IoT ve kriptoloji çalışmalarının özeti

Çalışma	Tarih	Cihazlar	Şifreleme Türü
I. Radhakrishnan [5]	2024	Arduino Nano, Arduino Micro	AES-128, SPECK, ASCON
L. E. Kane [6]	2020	ATMEGA328, STM32, ESP8266	AES, ChaCha, ACORN
R. Iqbal [7]	2025	Arduino Uno, Raspberry Pi 4, STM32 Nucleo	PRESENT, SIMON, SPECK, HIGHT, KATAN
S. S. Raghavan [8]	2023	OMNeT++ (Simülatör)	AES, RSA, SPECK, ECC, PRESEN
J. Soto-Cruz [9]	2024	ESP32 Dev Module , NodeMCU ESP8266, MSP430, NUCLEO-G0B1RE, Arduino Nano	PRESENT, SPECK, PICCOLO, CRAFT, HUMMINGIBIRD-2, ASCON, ACORN, LIZARD, FRUIT-80, TRIVIUM
I. Kuzminykh [10]	2023	Arduino Mega 2560	AES, XTEA, HIGHT, KLEIN, ECC, BLOWFISH, TWOFISH, SERPENT, PICCOLO, PRESENT
S. Abdullah [11]	2022	Arduino Uno	SHA256, AES128
L. Sleem [12]	2021	ATmega328P, Teensy 3.6, DOIT ESP32	SPECK, AES
M. Ramalho [13]	2025	ESP8266 80 Mhz, ESP8266 160 Mhz, Arduino Uno	AES, AESSmall, AESTiny, SPECK, SPECKSmall, SPECKTiny
Önerilen Çalışma	2025	NodeMCU ESP8266, ESP32	AES128, AESSmall128, AESTiny128, ChaCha (20,12,8 rounds) SPECK, SPECKSmall, SPECKTiny, PRESENT, ACORN128, ASCON128

Bu çalışmada ise geleneksel ve yenilikçi algoritmaların enerji tüketimi bakımından karşılaştırılması gerçekleştirilmiştir. Tablo 1'de literatürde yer alan IoT ve kriptoloji çalışmalarının özeti verilmiştir. Önerilen çalışma, özellikle 2024 yılında yapılan çalışmalardan farklı olarak NodeMCU ESP8266 ve ESP32 gibi farklı IoT cihazlarında ChaCha, SPECK, PRESENT, ASCON VE ACORN gibi yenilikçi, AES gibi daha geleneksel algoritmaların uygulanması ve enerji tüketimlerinin analiz edilmesiyle literatüre katkı sağlamaktadır. Böylece önerilen çalışmanın, yalnızca güvenlik değil aynı zamanda enerji verimliliği perspektifinden de değerlendirme yapması, onu mevcut çalışmalardan ayırarak değerli kılmaktadır.

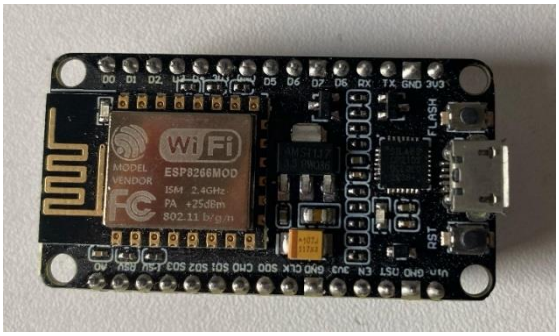
2 Deneysel yöntemler

2.1 Deneyde kullanılan IoT cihazları ve geliştirme ortamı

Bu çalışmada iki farklı IoT cihazı (Heltec ESP32 LoRa V3 ve ESP8266) ile bir adet güç ölçüm cihazı kullanılmıştır.

NodeMCU ESP8266 CP2102 Geliştirme Kartı (SK336)

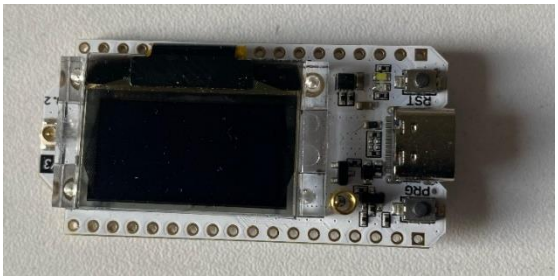
ESP8266 mobil, giyilebilir teknolojiler ve IoT uygulamaları için en düşük güç tüketimini hedefleyerek tasarlanmıştır. ESP8266 kendi içinde yeterli bir WiFi ağ çözümüne sahiptir. Entegre önbelleğe sahiptir. ESP8266 Tensilica L106 32-bit mikrodenetleyici ile entegre edilmiştir. Bu mikrodenetleyici, çok düşük güç tüketimi ve 16-bit RISC mimarisi ile öne çıkar. 80 MHz CPU saat hızına sahiptir. Bu değer 160 MHz'e kadar çıkabilir. 128 KB RAM ve 4 MB Flash memory'e sahiptir. 16 adet GPIO pini bulunur. Micro USB kablosu ile güç verilebilir veya programlanabilir. Arduino geliştirme ortamlarında programlama desteğine sahiptir. [14]



Şekil 1. NodeMCU ESP8266 geliştirme kartı

WiFi LoRa 32 V3.2 LoRa Node Development Kit

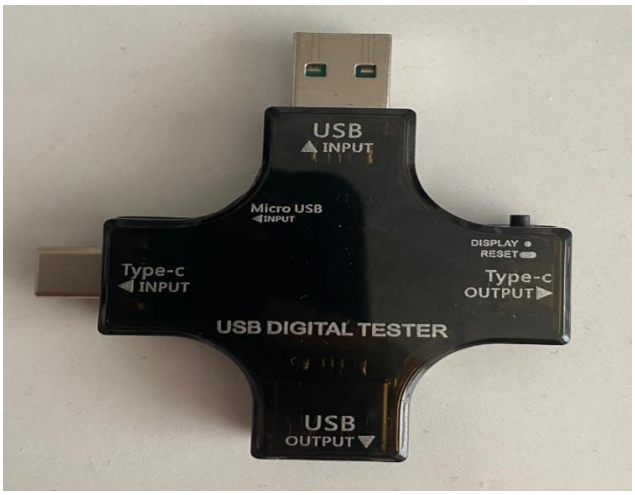
WiFi LoRa 32 Heltec Automation tarafından 2017'de geliştirilmiştir. IoT projelerinde kullanılmak üzere tasarlanmıştır. Geliştirme kiti ESP32-S3FN8 işlemcisi ve SX1262 LoRa node chip ile birlikte gelir. Entegre WiFi, LoRa, Bluetooth ağlarına sahiptir. Type C portu ile güç sağlanabilir ve programlanabilir. Arduino geliştirme ortamlarında programlama desteği vardır. 512KB SRAM ve 8MB SiP Flash memory'ye sahiptir. [15]



Şekil 2. ESP32 geliştirme kartı

Güç ölçüm cihazı

Güç ölçüm cihazı (powermeter), elektronik cihazların güç tüketimini ölçmek için kullanılan bir cihazdır. Gerilim olarak 3.30V - 30.0V aralığında, akım olarak 0.00 - 5.10A aralığında ölçüm yapabilir. Kolayca kullanılabilir boyutlardadır. Kullanıcıya gerçek zamanlı izleme imkanı sunar. Bağlantı noktaları USB-A giriş, USB-A çıkış, Micro-USB giriş, USB-C giriş, USB-C çıkış şeklindedir. Cihaz üzerinde bulunan buton aracılığıyla ekranlar arasında geçiş ve kolay erişim sağlanmaktadır. Çeşitli bağlantı noktalarına sahip olması, pratik kullanımı ve ölçüm kolaylığı sebebiyle tercih edilmiştir.



Şekil 3. Güç ölçüm cihazı

Tablo 2. IoT cihazları ve özellikleri

IoT Cihazı	CPU Hızı	RAM	Flash Bellek	Mimari
ESP8266	160MHz	128KB	4MB	32-bit MCU
ESP32	240MHz	512KB	8MB	32-bit Xtensa LX7

IoT cihazlarının yazılım geliştirme süreci Arduino IDE ortamında gerçekleştirilmiştir. Arduino IDE; geniş kütüphane desteği, yaygın kullanıcı tabanı, basit arayüzü ve fonksiyonel yapısı nedeniyle tercih edilmiştir. Çalışmada kullanılan şifreleme algoritmaları (PRESENT hariç) Arduino Crypto Library [16] kullanılarak optimize edilmiş biçimde IoT cihazları üzerinde çalıştırılmıştır.

2.2 Şifreleme algoritmaları

IoT teknolojisi her alanda yaygınlaşıp, verimliliği arttırırken güvenlik risklerini de arttırmaktadır. IoT cihazlarının genel özellikleri olan sınırlı bellek, düşük güç ve işlemci kapasitesi; şifreleme yaparken hafif (lightweight) şifreleme algoritmalarını kullanmayı zorunlu hale getirir. Bu algoritmalar minimum hesaplama, bellek ve enerji kullanımıyla veri güvenliğini sağlar.

Şifreleme bir metni, anahtar ile şifreli metne dönüştürerek veri güvenliği sağlar. Simetrik şifreleme algoritmalarında şifreleme ve şifre çözme için tek anahtar kullanılır. Bu sayede, işlem süresi açısından oldukça hızlıdır. Tek anahtar kullandığı için de az kaynak kullanır. İşlem süresi ve kaynak kullanımı dikkate alınarak büyük miktarda verilerin güvenli aktarımı için tercih edilir. Asimetrik şifreleme algoritmaları uygulanırken şifreleme ve şifre çözmede farklı anahtarlar kullanılır. Kullanılan anahtarların biri açık diğerinin gizli olması sebebiyle sadece ilgili kullanıcı şifreli metni çözebilir. Hash fonksiyonları girdi olarak verilen metni sabit uzunlukta, geniş çapta değişen bir metin haline getirir.

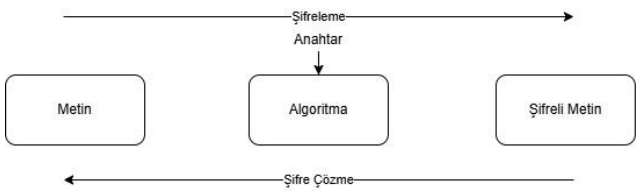
Projede donanımsal olarak dikkat edilmesi gereken unsurlar sınırlı hafıza, düşük pil gücü, sınırlı hesaplama gücüdür. Ek olarak süre kullanımını da dikkat edilmelidir. Önerilen çalışmada yer alan şifreleme algoritmaları arasında proje gereksinimleri, çalışılan konu üzerinde yapılan güncel çalışmalar göz önünde bulundurularak, ayrıntılı incelenmek ve literatür araştırması yapılmak üzere seçilmiş olanlar AES, SPECK, ChaCha, PRESENT, ASCON, ACORN algoritmaları ve varyantlarıdır. Veri iletimi sırasında gizlilik ve güvenliği sağlamak, sınırlı kaynakları etkin kullanmak gibi amaçlarla farklı türden algoritmalara yer verilmiştir.

AES algoritması ABD Ulusal Standart ve Teknoloji Enstitüsü (National Institute of Standards and Technology – NIST) tarafından 2001 yılında geliştirilmiş bir algoritmadır. AES veriyi 16 byte boyutunda bloklar halinde alır ve çıktı olarak yine 16 byte boyutunda çıktı oluşturur. Veri boyutu daha fazla ise 16 byte olarak bölünür. Simetrik şifreleme algoritması olduğu için açık anahtar kullanılır. Anahtar, her turda hesaplamaya tabi tutulur. Böylece her turda farklı anahtar oluşturulur. AES-128, AES-192 ve AES-256 için tur sayıları farklıdır. AES- 128 için 10 tur, AES-192 için 12, AES-256 için 14 tur bulunur. Her turda dört işlem yapılır. Bu işlemler byte değiştirme, satır kaydırma, sütun karıştırma ve tur anahtarı ile toplama şeklindedir. AESSmall ve AESTiny, AES algoritmasının gömülü sistemler ve IoT cihazları için uyarlanmış hafifletilmiş sürümleridir. AESSmall, özellikle bellek kullanımını azaltmaya yönelik tasarlanmış olup sınırlı RAM ve Flash kapasitesine sahip mikrodenetleyicilerde kullanılmaktadır. Bu yaklaşım kod boyutunu küçültmekte ve bellek tüketimini en aza indirmekte başarılı olmakla birlikte, işlem hızında belirli ölçüde kayıplara neden olmaktadır. AESTiny ise daha da minimal bir yapıya sahip olup, 8-bit gibi kısıtlı donanımlarda dahi çalışabilecek şekilde optimize edilmiştir. Çoğunlukla C dili ile geliştirilmiş olan bu sürüm, taşınabilirlik açısından avantaj sağlamakta ve donanım bağımsız çalışabilmektedir. Özetle, AESSmall sürümü bellek tasarrufunu ön planda tutarken AESTiny, düşük bellek kullanımı ile birlikte basitlik ve uyarlanabilirliği ön plana çıkarmaktadır. Bu nedenle, söz konusu sürümlerin tercih edilmesi cihazın işlem kapasitesi, bellek kısıtları ve uygulamanın performans gereksinimlerine bağlı olarak değişiklik göstermektedir. SPECK algoritması 2013 yılında yayınlanan yeni nesil bir hafif şifreleme algoritmasıdır. Gelen veri ikiye bölünerek kullanılır. Her turda sağa döndürme, sola döndürme, toplama, XOR işlemleri uygulanır. Algoritmaya göre blok boyutları 32, 48, 64, 96, 128 bit , anahtar boyutu s 64, 72, 96, 128, 144, 192, 256 bit, tur sayısı 22-34 arasında olabilir. ChaCha algoritması ilk olarak 2005 yılında yayınlanmış bir akış şifreleme türüdür. Chacha8, ChaCha12, ChaCha20 olmak üzere tur sayısına

göre isimlendirilmiş 3 versiyonu vardır. Chacha20 versiyonunun anahtar boyutu 256 bit, blok boyutu 512 bitten oluşmaktadır. Simetrik algoritma türü olduğu için şifreleme ve şifre çözmede aynı anahtar kullanılır. Algoritma anahtar üretme, başlatma, şifreleme ve sonuç adımlarından oluşur. Veriler sabit boyutlu bloklar yerine rastgele oluşturulan bitler kullanarak şifrelenir. PRESENT algoritması blok şifreleme algoritmasıdır. 2007 yılında geliştirilmiş olup, 80 bit ve 128 bit boyutunda anahtar kullanan versiyonları vardır. Her turda tur anahtarı ile toplama, S-box layer, P-layer, tur anahtarı ile toplama şeklinde 4 adım uygulanır. Her turda 64 bitlik tur anahtarı üretilir. ASCON algoritması 2014 yılında geliştirilmiş sünger yapılı (sponge-based) bir İlişkili Verilerle Kimliği Doğrulanmış Şifreleme (Authenticated Encryption with Associated Data – AEAD) algoritmasıdır. 2023 yılında NIST tarafından hafif kriptografi uygulamaları için standartlaştırılmasına karar verilmiştir. Şifreleme ve kimlik doğrulamayı birlikte sağlar. Dört adımda çalışır: Başlatma, ilişkili verilerin işlenmesi, şifreleme ve final. 128 bit anahtar ve 16 byte blok boyutu ile işlem yapar. ACORN (Additive Congruential Random Number) 1989 yılında yayınlanmış bir hafif kriptografi algoritmasıdır. Algoritma 128 bit anahtar ve 16 byte blok boyutu ile çalışır.

ESP8266 üzerinde geliştirilen yazılımlarda, cihazın çalışmasını yöneten iki temel fonksiyon kullanılmıştır. Başlangıç fonksiyonu, cihaz ilk açıldığında veya yeniden başlatıldığında bir kez çalışarak, pin modlarının ayarlanması, seri haberleşmenin başlatılması, modüllerin etkinleştirilmesi ve sensörlerin tanımlanması gibi başlangıç işlemlerini yürütür. Sürekli döngü fonksiyonu ise cihaz çalıştığı sürece tekrarlanan işlemleri yönetir; sensör verilerinin okunması, veri işleme ve karar mekanizmalarının işletilmesi gibi görevler bu fonksiyon içinde gerçekleştirilir. Her iki fonksiyon da geriye değer döndürmemektedir.

Bu çalışmada, deneysel yapının gereği olarak gerçek sensör verileri yerine kukla veriler kullanılmış ve şifreleme işlemleri, cihazın başlangıç aşamasında (başlangıç fonksiyonu içinde) gerçekleştirilmiştir. Böylece, veri işleme ve şifreleme süreci doğrudan cihaz üzerinde başlatılmış ve ölçümler için gerekli koşullar sağlanmıştır. Veriler, IoT cihazlarının belleğinde bloklar halinde tanımlanmıştır. Her şifreleme algoritması farklı blok boyutuna sahip olmakla birlikte, şifrelenmemiş metinler tüm algoritmalarda aynı tutulmuştur. Tüm algoritmalarda hem şifreleme hem de şifre çözme işlemleri uygulanmış; ayrıca süre ölçümleri ve enerji tüketimleri hesaplanmıştır. Şifreleme sürecine ait akış diyagramı Şekil 4'te verilmiştir.



Şekil 4. Şifreleme akış diyagramı

Şifreleme algoritmalarının testlerinde haberleşme hızı 115200 baud olarak belirlenmiştir. Süre ölçümleri, Arduino IDE geliştirme ortamında yapılmıştır. Şifreleme süresi ölçümünde kullanılan fonksiyon, şifreleme ve şifre çözme fonksiyonlarının öncesinde ve sonrasında çağrılarak doğru süre hesaplamaları yapılmıştır.

Enerji tüketimi için güç ölçüm cihazı ile güç hesaplaması yapılmıştır. Şifreleme fonksiyonları döngü içerisinde cihazın içerisinde sonsuz kez çalışacak şekilde ayarlanmıştır. Cihaz çalıştığı süre boyunca tükettiği güç ölçüm cihazı ile değerler alınmıştır. Güç tüketim hesaplaması aşağıdaki gibidir:

$$Güç\ (W)\ =\ Voltaj\ (V)\ x\ Akım\ (A) \qquad (1)$$

Yukarıda yer alan güç tüketim hesabına göre voltaj ve akım değerleri güç ölçüm cihazından alınarak çarpma işlemine tabi tutulmuştur. Ölçülen şifreleme ve şifre çözme süreleri ile de elde edilen güç değeri çarpılarak enerji tüketimi hesaplanmıştır.

$$Enerji\ Tüketimi\ (J)\ =\ Güç\ (W)\ x\ Süre\ (s) \qquad (2)$$

3 Bulgular ve tartışma

Her cihazda ve algoritmada ayrı ayrı süre ölçümleri yapılmıştır. Bu ölçümlerin değerleri Tablo 3 ve Tablo 4'te sunulmuştur. Tablo 3 ve Tablo 4'te yer alan süre ölçüm değerleri optimize edilmiş kodlar ile elde edilmiştir.

Tablo 3'te şifreleme ve şifre çözme süreleri incelendiğinde ESP32 cihazında en hızlı çalışan 64 blok boyutluk ChaCha algoritmasının 8 turluk varyantı ~9.6 µsolarak ölçülmüştür. Aynı algoritmanın 12 ve 20 varyantları incelendiğinde ise tur sayısı arttıkça sürenin arttığı gözlemlenmiştir. Bu değerler hem şifreleme hem de şifre çözme fonksiyonları için benzerlik göstermektedir. Bu algoritmada şifreleme ve şifre çözme fonksiyonları için süreler aynı şartlarda oldukça benzerlik göstermektedir. Acorn128 algoritması ise 16 blok boyutu ile ChaCha algoritmasının 12 turluk varyantı ile oldukça yakın sonuçlar vermiştir. Speck algoritması 16 blok boyutu ile bu çalışmada tercih edilen algoritmalar arasında ortalamanın altında bir gerçekleşme süresi ortaya koymuştur. Small ve Tiny varyantları süre bakımından birbirlerine yakın sonuçlar vermiştir. Algoritmanın varyantları küçüldükçe sürenin de yaklaşık olarak arttığı gözlemlenmiştir.

Tablo 3. HELTEC ESP32 v3 cihazında farklı şifreleme algoritmaları için şifreleme ve çözme süreleri

Algoritma	Block Size (bytes)	Şifreleme Süresi (µs)	Şifre Çözme Süresi (µs)
AES128 (ECB modu)	16	252.121	29.625
AESSmall128 (ECB modu)	16	139.435	139.671
AESTiny128 (ECB modu)	16	139.498	×
ChaCha (20 rounds)	64	14.128	14.132
ChaCha (12 rounds)	64	11.104	11.133
ChaCha (8 rounds)	64	9.606	9.636
Speck (128-bit anahtar)	16	29.063	29.062
SpeckSmall (128-bit anahtar)	16	59.679	60.066
SpeckTiny (128-bit anahtar)	16	59.655	×
Present (128-bit anahtar)	8	920.273	639.102
Acorn128 (128-bit anahtar)	16	11.106	10.974
Ascon128 (128-bit anahtar)	16	48.215	47.942

Ascon128 algoritması 16 blok boyutu ile Speck algoritması ile benzer sonuçlar ortaya koymuştur. AES geleneksel algoritması ise 16 blok boyutu ile ortalamanın üzerinde bir şifreleme süresi ortaya koymuştur. Bu algoritmada şifreleme fonksiyonunun, şifre çözme fonksiyonuna kıyasla yaklaşık 8 kat daha yavaş çalıştığı olduğu ortaya koyulmuştur. Varyantları küçüldükçe şifreleme süresinde azalma olduğu ortaya koyulmuştur. Present algoritmasının 8 blok boyutuyla gerçekleşme süresi olarak en uzun süre çalışan algoritma olduğu bu cihaz özelinde ortaa koyulmuştur.

Tablo 4'te verilen şifreleme ve şifre çözme süreleri incelendiğinde ESP8266 cihazında, ESP32 ile kıyaslandığında PRESENT algoritmasının hem şifreleme fonksiyonunda hem de şifre çözme fonksiyonunda en kısa süreyi verdiği ortaya koyulmuştur. Gerçekleştirme süresi olarak yaklaşık 330 katlık bir fark olduğu ortaya koyulmuştur. Present algoritmasındaki gerçekleştirme süresi farkı her iki fonksiyonda da korunmuştur. ChaCha algoritmasının tüm varyantları birbirlerine yakın süreler olup ortalamanın altında gerçekleştirme süresine sahiptirler. ACORN128 algoritması ise ChaCha algoritmasının 12 turluk versiyonu ile benzerlik göstermektedir. SPECK algoritması ise şifreleme süresi olarak ortalamada kalmıştır. Varyantı küçüldükçe gerçekleştirme süresinde artış gözlemlenmiştir. AES algoritmasında ise şifreleme süresi ortalamanın üzerinde olup varyantlar küçüldükçe gerçekleştirme süresinde yaklaşık 5 kat artış gözlemlenmiştir.

Tablo 4. ESP8266 NodeMCU cihazında farklı şifreleme algoritmaları için şifreleme ve çözme süreleri

Algoritma	Block Size (bytes)	Şifreleme Süresi (µs)	Şifre Çözme Süresi (µs)
AES128 (ECB modu)	16	175.368	264.121
AESSmall128 (ECB modu)	16	614.255	853.439
AESTiny128 (ECB modu)	16	614.232	×
ChaCha (20 rounds)	64	42.950	42.945
ChaCha (12 rounds)	64	33.963	34.041
ChaCha (8 rounds)	64	29.513	29.591
Speck (128-bit anahtar)	16	89.790	93.602
SpeckSmall (128-bit anahtar)	16	154.071	153.141
SpeckTiny (128-bit anahtar)	16	154.008	×
Present (128-bit anahtar)	8	2.773	1.960
Acorn128 (128-bit anahtar)	16	31.360	31.338
Ascon128 (128-bit anahtar)	16	118.573	117.766

Tablo 3 ve Tablo 4'te verilen her iki cihaza ait farklı algoritmalara ait süre değerleri incelendiğinde ise ESP8266 cihazında PRESENT algoritmasının en hızlı, ESP32 cihazında ise PRESENT algoritmasının en yavaş algoritma olduğu ortaya koyulmuştur. ESP32 cihazında AES algoritmasında varyantlar küçüldükçe süre azalmışken ESP8266 cihazında ise sürede artış görülmüştür. Aynı algoritmaların iki farklı cihazda oldukça farklı sonuçlar vermesi, IoT cihazının donanım mimarisinin süre ölçümlerinde önemli bir etken olduğunu ortaya koymaktadır. Aynı blok boyutuna sahip iki algoritmanın kıyaslanması ile bu algoritmalar arasında gerçekleme süresi olarak oldukça büyük farklar olduğu ortaya koyulmuştur. Bu sayede blok boyutu ile süre arasında doğrudan bir bağlantı kurulamadığı sonucu ortaya koyulmuştur.

Enerji tüketimi sonuçları Tablo 5 ve Tablo 6'da sunulmuştur. Tablo 5'te gösterildiği üzere ESP8266 cihazında PRESENT algoritması 9.760×10^{-4} J ile en az enerji tüketen algoritmadır. Şifreleme ve şifre çözme fonksiyonları arasında yaklaşık 3 katlık enerji tüketimi farkı ortaya koyulmuştur. ACORN128 algoritması ortalamanın altında kalarak ESP32 cihazı ile benzer sonuçlar göstermiştir. ChaCha algoritması ise ortalamanın biraz altında enerji tüketmiş olsa da tur sayısı azalınca enerji tüketiminde büyük bir değişim olmadığı ortaya koyulmuştur. SPECK algoritması ortalama bir enerji tüketimine sahiptir. Varyantları azaldıkça enerji tüketiminde artış ortaya koyulmuştur. ASCON128 algoritması ise SPECK ve ChaCha algoritmaları arasında enerji tüketmiştir. AES algoritması ise en yüksek enerjiyi tüketen algoritma olarak ortaya koyulmuştur. Varyantları küçüldükçe enerji tüketiminde artış olduğu ortaya koyulmuştur.

Tablo 5. ESP8266 NodeMCU cihazında farklı şifreleme algoritmaları için enerji tüketimi hesabı

Algoritma	Block Size (bytes)	Şifreleme Enerji Tüketimi (J)	Şifre Çözme Enerji Tüketimi (J)
AES128 (ECB modu)	16	6.195×10^{-2}	9.398×10^{-2}
AESSmall128 (ECB modu)	16	2.267×10^{-1}	3.047×10^{-1}
AESTiny128 (ECB modu)	16	2.181×10^{-1}	×
ChaCha (20 rounds)	64	1.619×10^{-2}	1.598×10^{-2}
ChaCha (12 rounds)	64	1.291×10^{-2}	1.311×10^{-2}
ChaCha (8 rounds)	64	1.107×10^{-2}	1.119×10^{-2}
Speck (128-bit anahtar)	16	3.188×10^{-2}	3.360×10^{-2}
SpeckSmall	16	5.639×10^{-2}	5.651×10^{-2}
SpeckTiny	16	5.621×10^{-2}	×
Present (128-bit anahtar)	8	9.760×10^{-4}	3.700×10^{-4}
Acorn128 (128-bit anahtar)	16	5.927×10^{-3}	5.923×10^{-3}
Ascon128 (128-bit anahtar)	16	2.241×10^{-2}	2.226×10^{-2}

Tablo 6'da gösterildiği üzere, ESP32 cihazında ACORN128 algoritması 5.927×10^{-3} J ile en az enerjiyi tüketmiştir. ChaCha algoritmasının 8 turluk varyantı ile ACORN128 algoritması yakın sonuçlar vermektedir. Tur sayısı arttıkça algoritmanın enerji tüketiminde artış olduğu ortaya koyulmuştur. ASCON128 algoritması bu çalışma özelinde ortalama bir enerji tüketimine sahiptir. SPECK algoritması ise ACORN128 algoritmasından yaklaşık 20 kat daha fazla enerji tüketmektedir. Varyantların küçülmesiyle enerji tüketiminde artış olduğu ortaya koyulmuştur. AES algoritması ise SPECK algoritmasına kıyasla daha fazla enerji tüketmektedir. Varyantların küçülmesiyle enerji tüketiminde azalma görülmektedir. Bu algoritmada şifreleme ve şifre çözme fonksiyonları arasında enerji tüketimi açısından net bir fark ortaya koyulmuştur. PRESENT ise en yüksek enerji tüketimine sahip algoritmadır.

Tablo 6. HELTEC ESP32 v3 cihazında farklı şifreleme algoritmaları için enerji tüketimi hesabı

Algoritma	Block Size (bytes)	Şifreleme Enerji Tüketimi (J)	Şifre Çözme Enerji Tüketimi (J)
AES128 (ECB modu)	16	8.946×10^{-2}	1.035×10^{-1}
AESSmall128 (ECB modu)	16	5.117×10^{-2}	5.000×10^{-2}
AESTiny128 (ECB modu)	16	4.966×10^{-2}	×
ChaCha (20 rounds)	64	5.340×10^{-3}	5.299×10^{-3}
ChaCha (12 rounds)	64	4.220×10^{-3}	4.253×10^{-3}
ChaCha (8 rounds)	64	3.554×10^{-3}	3.585×10^{-3}
Speck (128-bit anahtar)	16	1.037×10^{-2}	1.046×10^{-2}
SpeckSmall	16	2.178×10^{-2}	2.198×10^{-2}
SpeckTiny	16	2.171×10^{-2}	×
Present (128-bit anahtar)	8	3.238×10^{-1}	1.431×10^{-1}
Acorn128 (128-bit anahtar)	16	2.099×10^{-3}	2.074×10^{-3}
Ascon128 (128-bit anahtar)	16	9.113×10^{-3}	9.061×10^{-3}

Bu bulgular, IoT cihazlarında gizlilik ve güvenlik hassasiyetli veri iletimi, performans ve enerji verimliliği açısından kritik öneme sahiptir. Daha kısa süreli şifreleme işlemleri, daha az enerji tüketimine karşılık gelmekte ve bu durum pil ile çalışan IoT cihazlarının ömrünü doğrudan uzatmaktadır. Özellikle ESP32'nin düşük süreli şifreleme performansı, gerçek zamanlı veri iletiminin gerekli olduğu IoT senaryolarında önemli bir avantaj sağlamaktadır. Daha önce yapılan çalışmalarda da AES'in görece uzun süre gerektirdiği, hafif yapılı algoritmaların ise IoT ortamları için daha uygun olduğu vurgulanmıştır. Bu çalışma, farklı şifreleme ve şifre çözme algoritmaların iki ayrı IoT cihazında (ESP32 ve ESP8266) kıyaslanmasıyla literatüre katkı sağlamaktadır.

4 Sonuçlar

Bu çalışmada, IoT cihazlarında güvenli ve gizlilik odaklı veri iletimini düşük enerji tüketimiyle gerçekleştirebilecek kriptografik çözümler ayrıntılı olarak incelenmiştir. Yapılan ölçümler, kullanılan cihazların donanım mimarisine bağlı olarak algoritmalar arasında kayda değer performans farklılıkları bulunduğunu göstermiştir. Genel olarak ESP32, ESP8266'ya kıyasla hem süre hem de enerji tüketimi açısından daha verimli sonuçlar ortaya koymuştur. Özellikle ChaCha ailesi süre bakımından öne çıkarken, ACORN128 algoritması düşük enerji tüketimi ile dikkat çekmiştir. IoT cihazlarının kısıtlı işlem gücü, bellek kapasitesi ve enerji kaynakları, kriptografik algoritmaların seçiminde temel belirleyici faktörler olmaktadır. Güvenliğin artırılması amacıyla daha karmaşık algoritmalar tercih edildiğinde, enerji tüketimi ve işlem süresi artabilmekte; bu da gerçek zamanlı veri aktarımı gereken uygulamalarda performans sorunlarına yol açabilmektedir. Dolayısıyla, IoT ortamlarında kullanılacak şifreleme çözümlerinde şu temel gereklilikler öne çıkmaktadır:

- Donanım-uyumlu algoritma seçimi: Farklı IoT cihaz mimarilerinin aynı algoritmalar için oldukça farklı sonuçlar vermesi, algoritma seçiminde cihazın işlemci yapısının ve donanım desteklerinin mutlaka göz önünde bulundurulması gerektiğini ortaya koymaktadır.
- Enerji verimliliği: Batarya ile çalışan IoT cihazlarında, enerji tüketiminin düşük tutulması kritik öneme sahiptir. Bu nedenle, enerji maliyeti düşük ve hafif kriptografik çözümler tercih edilmelidir.
- Gerçek zamanlılık ve düşük gecikme: Sağlık, endüstri veya güvenlik odaklı IoT uygulamalarında, kriptografik işlemlerin veri aktarımında ek gecikmeye yol açmaması gerekmektedir.
- Güvenlik–performans dengesi: Algoritmaların güvenlik seviyesinin, uygulamanın ihtiyaçlarına göre belirlenmesi ve bu noktada performans ile enerji tüketimi arasında dengeli bir seçim yapılması gerekmektedir.

Elde edilen bulgular, IoT ortamlarında enerji verimliliği odaklı kriptografik uygulamalar için uygun algoritma ve donanım seçiminin kritik olduğunu göstermektedir. Bu çalışma, güvenli veri iletimi için pratik bir yol haritası sunmakta ve gelecekte yapılacak çalışmalara özellikle şu açılardan temel teşkil etmektedir: enerji tüketimini daha da azaltacak donanım tabanlı hızlandırıcıların geliştirilmesi, farklı IoT senaryolarında algoritma performanslarının değerlendirilmesi ve hafif kriptografi standartlarının gerçek donanımlar üzerinde uygulanabilirliğinin test edilmesi. Ayrıca, elde edilen bulguların akademi–sanayi iş birliği çerçevesinde değerlendirilmesi, hem akademik dünyada yeni optimizasyon yaklaşımlarının geliştirilmesine hem de endüstride güvenli ve enerji verimli IoT çözümlerinin hayata geçirilmesine önemli katkılar sağlayacaktır. Bu bağlamda, akıllı şehirlerde sensör tabanlı trafik yönetim sistemleri, sağlık sektöründe giyilebilir IoT tabanlı hasta izleme cihazları, enerji şebekelerinde akıllı sayaç ve dağıtım altyapıları ve tarımda akıllı sulama ve toprak nem kontrol sistemleri gibi alanlarda yürütülecek ortak projeler, bu çalışmanın bulgularının doğrudan uygulamaya aktarılabileceği örnekler arasında değerlendirilebilir.

Teşekkür

Bu çalışma, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu (TÜBİTAK) tarafından 124E544 numaralı proje kapsamında desteklenmiştir. Yazar, sağladığı desteklerden dolayı TÜBİTAK'a teşekkür eder.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı (iThenticate): %9

Kaynaklar

[1] M. A. Ferrag, L. Shu, X. Yang, A. Derhab and L. Maglaras, Security and privacy for green IoT-based agriculture: Review, Blockchain Solutions, and Challenges. IEEE Access, 8, 32031-32053, 2020. https://doi.org/10.1109/ACCESS.2020.2973178

[2] M. Rana, Q. Mamun, and R. Islam, Lightweight cryptography in IoT networks: A survey. Future Generation Computer Systems, 129, 77-89, 2022. https://doi.org/10.1016/j.future.2021.11.011

[3] N. Fatima, S. A. Siddiqui, and A. Ahmad, IoT based border security system using machine learning. 2021 International Conference on Communication, Control and Information Sciences (ICCISc), Idukki, India, 1-6, 2021. https://doi.org/10.1109/ICCISc52257.2021.9484934

[4] İ. Avcı, M. Koca ve M. Atasoy, Windows tabanlı uygulamalarda SQL enjeksiyon siber saldırı senaryosu ve güvenlik önlemleri. Avrupa Bilim ve Teknoloji Dergisi (28), 213-219, 2021. https://doi.org/10.31590/ejosat.995697

[5] I. Radhakrishnan, S. Jadon, and P. B. Honnavalli, Efficiency and security evaluation of lightweight cryptographic algorithms for resource-constrained IoT devices. Sensors, 24(12), 4008, 2024. https://doi.org/10.3390/s24124008

[6] L. E. Kane, J. J. Chen, R. Thomas, V. Liu, and M. Mckague, Security and performance in IoT: A balancing act. IEEE Access, 2020. https://doi.org/10.1109/ACCESS.2020.3007536

[7] R. Iqbal, N. M. Ansari, M. ur R. Awan, M. Ismail, and H. Gul, Design and Evaluation of lightweight cryptographic algorithms for internet of things (IoT) devices: achieving optimal trade-offs between security, computational speed, and energy efficiency in resource-constrained environments. The Progress: A Journal of Multidisciplinary Studies, 6, 2025. https://doi.org/10.71016/tp/smfybz24

[8] S. Raghavan and V. Scholar, A lightweight cryptographic framework for securing real-time communication in automotive systems. International Journal of Information Technology and Management Information Systems, 87-103, 2023. https://doi.org/10.34218/IJITMIS_14_01_010

[9] J. Soto-Cruz, E. Ruiz-Ibarra, J. Vázquez-Castillo, A. Espinoza-Ruiz, A. Castillo-Atoche, and J. Mass-Sanchez, A survey of efficient lightweight cryptography for power-constrained microcontrollers. Technologies, 13(1), 3, 2025. https://doi.org/10.3390/technologies13010003

[10] I. Kuzminykh, M. Yevdokymenko, and V. Sokolov, Encryption algorithms in IoT: security vs lifetime. SSRN Electronic Journal, 2023. https://doi.org/10.2139/ssrn.4636161

[11] N. M. Shapina Abdullah, N. M. Noor, and A. Khalid, IOT security: data encryption for arduino-based IoT devices. Journal of Positive School Psychology, 2022. https://www.journalppw.com/index.php/jpsp/article/view/5116

[12] L. Sleem and R. Couturier, Speck-R: An ultra light-weight cryptographic scheme for Internet of Things. Multimedia Tools and Applications, 2021. https://link.springer.com/article/10.1007/s11042-020-09625-8

[13] M. Ramalho, G. Sampaio, N. Neves, R. Porto, V. A. Sobral, M. Rezende, and D. S. V. Medeiros, A comparative evaluation of symmetric cryptography algorithms for resource-constrained devices. Journal of Internet Services and Applications, 16, 2025. https://doi.org/10.5753/jisa.2025.4903

[14] Espressif Systems, ESP8266 NodeMCU Datasheet. https://components101.com/sites/default/files/component_datasheet/, 2015.

[15] Heltec Automation, WiFi LoRa 32 (V3) Datasheet – HTIT-WB32LA V3.2. https://resource.heltec.cn/download/WiFi_LoRa_32_V3/HTIT-WB32LA_V3.2.pdf, 2022.

[16] R. Weatherley, Arduino cryptography library. http://rweather.github.io/arduinolibs/crypto.html, Erişim tarihi: 26 Mayıs 2025.