T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NESNELERİN İNTERNETİ UYGULAMALARI İÇİN GÜVENLİ BİR HABERLEŞME GERÇEKLEŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Muhammed Saadetdin KAYA

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Kenan İNCE

Aralık, 2021

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

NESNELERİN İNTERNETİ UYGULAMALARI İÇİN GÜVENLİ BİR HABERLEŞME GERÇEKLEŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Muhammed Saadetdin KAYA
36193619025

Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Kenan İNCE

Aralık, 2021

TEŞEKKÜR VE ÖNSÖZ

Bu tez çalışmasının her aşamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgemeden beni her konuda yönlendiren danışman hocam Sayın Dr. Öğr. Üyesi. Kenan İNCE'ye,

Çalışmalarımda ayrıca tüm hayatım boyunca olduğu gibi bu çalışmalarım süresincede benden her türlü desteklerini aileme,

Tezin uygulama aşamasında vermiş oldukları maddi ve manevi destekten dolayı, İnönü Üniversitesi Bilimsel Araştırma Projeleri Bölümü'ne

teşekkür ederim.

ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum “Nesnelerin İnterneti Uygulamaları İçin Güvenli Bir Haberleşme Gerçekleştirilmesi” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Muhammed Saadetdin KAYA

İÇİNDEKİLER

ÇİZELGELER DİZİNİ

ŞEKİLLER DİZİNİ

SEMBOLLER VE KISALTMALAR

IoT : Internet of Things
ZA : Zamanlama Analizi
ZAS : Zamanlama Analizi Saldırıları
YA : Yan-kanal Analizi
YAS : Yan-kanal Analizi Saldırıları
KRB : Kontrollü Rassal Bekleme
DoS : Denial of Service
DDoS : Distrubuted Denial of Service
TLS : Transport Layer Security
SSL : Secure Sockets Layer
DNS : Domain Name System
XMPP : Extensible Messagging and Presence Protocol
DDS : Data Distribution Service
RFID : Radio Frequency Identification
Wi-Fi : Wireless Fidelity
NFC : Near Field Communication
Ω : En İyi Durum (0% eşleşme oranı)
O : En Kötü Durum (100% eşleşme oranı)
GHz : Gigahertz
LFSR : Linear Feedback Shift Register
NLFSR : Nonlinear Feedback Shift Register
s_i : LFSR fonksiyonu bileşenleri
b_i : NLSFR fonksiyonu bileşenleri

ÖZET

Yüksek Lisans Tezi

NESNELERİN İNTERNETİ UYGULAMALARI İÇİN GÜVENLİ BİR HABERLEŞME GERÇEKLEŞTİRİLMESİ

Muhammed Saadetdin KAYA

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

50+VII sayfa

2021

Danışman: Dr. Öğr. Üyesi. Kenan İNCE

Hızla gelişen internet teknolojileri ile birlikte gittikçe yaygınlaşan internet kullanımı ve internete bağlı cihazların çeşitliliği ile birlikte internet kullanımı artık yalnızca bilgisayar, telefon ve tablet gibi cihazlarla sınırlı kalmamaktadır. Akıllı ev aletleri, otonom araçlar ve akıllı şehirler gibi projlerlere hayatın her alanında Nesnelerin İnterneti (IoT) kavramı ile birlikte görmeye başladığımız akıllı ve internete bağlı cihazlar, yalnızca insan-cihaz etkileşimini değil aynı zamanda cihaz-cihaz etkileşimini de başka bir boyuta taşımaktadır. Kullanım alanları gereği uygulamaların çoğunda kişisel bilgi alışverişini içeren IoT sistemleri, güvenlik ve gizlilik tehditlerinin hedefi olmaktadır. Bu sistemleri oluşturan elementler masaüstü bilgisayarlar, tabletler gibi güçlü bilgi işlem aygıtlarından RFID etiketleri ve sensörler gibi düşük kaynak kapasitesine sahip aygıtlara kadar değişiklik göstermektedirler. Özellikle bu kısıtlı aygıtlar söz konusu olduğunda, geleneksel şifreleme algoritmaları gerekli güvenlik ve performansı sağlayamamaktadır. Yapılan tez çalışmasında IoT sistemlerinde sıkça kullanılan mimarilerden, haberleşme teknolojilerinden, güvenlik unsurlarından ve potansiyel tehditlerle birlikte literatürde bulunan kriptografik çözümlere dikkat çekilmiş olup sisteme yeni bir IoT nesnesi katılması durumunda yaşanan sorunların çözülmesi adına yeni ve hibrit bir hafif siklet yaklaşım sunulmaktadır. Güvenilirliği herkesçe bilinen bir hafif siklet dizi şifreleme algoritması olan Grain 128a ile kontrollü rassal bekleme yapılarak sistemin doğrusallığının bozulması amaçlanmıştır. Sunulan yaklaşım belirtilen senaryoya uygun şekilde kurulan bir düzenekte uygulanmış olup sonuçları yorumlanmaktadır.

Anahtar Kelimeler: Nesnelerin İnterneti (IoT), Kriptgorafi, Grain, Güvenlik Yönetimi, Zamanlama Analizi Saldırıları

ABSTRACT

Master Thesis

PERFORMING A SECURE COMMUNICATION FOR INTERNET OF THINGS APPLICATIONS
Muhammed Saadetdin KAYA

Inonu University
Graduate School of Nature and Applied Sciences
Department of Computer Engineering

50+VII pages

2021

Supervisor: Asst. Prof. Dr. Kenan İNCE

With rapidly evolving internet technologies, the use of the Internet and the diversity of connected devices, the use of the internet is no longer limited to devices such as computers, phones and tablets. Smart and Internet-connected devices that we start seeing with the Internet of things (IoT) concept in all areas of life to projects such as smart home appliances, autonomous vehicles and smart cities take not only human-device interaction, but device-device interaction to another dimension. Due to their usage areas, IoT systems, which include the exchange of personal information in most of their applications, are the target of security and privacy threats. The elements that make up these systems vary from powerful computing devices such as desktop computers, tablets, to devices with low resource capacity such as RFID tags and sensors. Especially when it comes to these restricted devices, traditional encryption algorithms cannot provide the necessary security and performance. In the thesis study architecture commonly used in IoT systems, communication technologies, security elements, and drawn attention to potential threats in the literature together with cryptographic solutions IoT system into a new object in the case of join and in the name of solving the problems and lightweight new hybrid approach is presented. It is aimed to disrupt the linearity of the system by performing controlled random waiting with Grain 128a, a lightweight stream encryption algorithm whose reliability is known. The presented approach has been applied in a set-up in accordance with the specified scenario and results are being interpreted.

Keywords: Internet of Things, Cryptography, Grain, Security Management, Timing Analysis Attacks

1. GİRİŞ

Günümüzde kahve makinesi, buzdolabı ve bulaşık makinesi gibi gündelik eşyalardan araba ve tren gibi ulaşım araçlarına kadar birbirinden bağımsız birçok alanda internete bağlı cihazlara rastlamak mümkün olmaktadır. İnternete bağlı cihazlar tarafından oluşturulan ve Nesnelerin İnterneti (IoT – Internet of Things) olarak adlandırılan bu ekosistem, interneti kullanan nesnelerin çeşitliliği ve bu nesnelerin sürekli iletişim halinde olmaları sebebiyle doğaları gereği çeşitli riskleri de beraberinde getirmektedir. Genellikle fiziksel olarak bu cihazlara erişimin zor olmasının yanında bu cihazların birbirleriyle haberleşirken internete ihtiyaç duymaları sebebiyle –ağ korumalarının yeterli olmadığı durumlarda- kullandıkları ağlara erişim oldukça kolay olmaktadır. Genellikle fiziksel olarak bu cihazlara erişimin zor olmasının yanında gerekli güvenliğin sağlanamamasının sonucu olarak bu nesnelerin tamamen ele geçirilmeleri veya haberleşmeleri esnasında araya girilerek gizli kalması istenen bilgilerin sızdırılması gibi istenmeyen durumlarla karşılaşılması oldukça olasıdır (Birkel ve Hartmann, 2020; Abbas vd., 2019). Örneğin; elektrik hatlarını kontrol eden bir anahtar sisteminin ele geçirilmesi sonucunda saldırgan tarafından tüm elektrik hattına zarar verilebilir veya bir kamera sisteminin kontrol edilmesi ile o ortama ait görüntülerin sızdırılması mümkün olabilir. Bu gibi durumlar IoT sistemlerinde güvenli haberleşmenin önemini arttırmaktadır.

Fiziksel olarak bu cihazlara erişim olmasa bile bir işlemin ne kadar sürede yapıldığı, yapılırken harcanan güç, ortaya çıkan elektromanyetik yayınım, işlem süresince çıkan sesin şiddeti gibi sistem dışına istemsiz çıkışlar kullanılarak çeşitli analizler yapılabilmektedir (Zhao ve Ge, 2013). Bu istemsiz çıkışlar sistemin çözümlenmesinde kullanılabilecek nitelikte ise bu bilgiler yan-kanal bilgisi, bu bilgiler kullanılarak yapılan analizler ise yan-kanal analizi olarak adlandırılmaktadırlar. Yan-kanal analiz saldırıları (YAS), bu analizler vasıtasıyla sistem hakkında bilgi edinmeyi veya çözümlenme sağlanmasını hedefleyen saldırılardır (Joy Persial vd., 2011).

Analiz işleminin ve uygulanmasının nispeten daha kolay olduğu Zamanlama Analizi Saldırıları (ZAS) bir işlemin veya algoritmanın çeşitli şartlar altında değişen işleme süresinin yorumlanmasıyla sistem hakkında bilgi edinmek amacıyla yapılan bir saldırı türüdür. Çeşitli çalışmalar, bu saldırı türü ile sistem hakkındaki en kritik bilgilerin ortaya çıkarılabileceğini göstermiştir (Kocher, 1996; Janke ve Laackmann, 2002).

IoT'de cihazlar kendi aralarında ve kullanıcılarla sıklıkla ortak internet ağları üzerinden iletişim kurarlar ve genellikle düşük bellek, düşük güç ve düşük işlem yeteneklerine sahip olan bu cihazlar için daha az kaynak kullanımı ile güçlü şifreleme çözümleri tasarlanması gerekmektedir (Mukherjee, 2015). Düşük kaynak kapasitesi sebebiyle bilgisayar tabanlı konvansiyonel kriptografik çözümlerin birçoğunun tam olarak doğrudan uygulanamaması, hafif-siklet algoritmaların bu alanda sıklıkla tercih edilmesi sonucunu doğurmuştur (Kim, 2017).

Hafif siklet kriptografik şifreleme, RIFD etiketleri, sensörler, temassız akıllı kartlar vb. kısıtlı ortamlarda uygulamalar için uyarlanmış, özel senaryolar için özel çözümler öneren kriptografik algoritmalar ve protokollerdir (Katagi ve Moriai, 2008). Genellikle bilgisayar tabanlı kriptografik çözümlerin hafifletilmeleriyle IoT sistemlerine uygulanan bu algoritmalar, hem yapıları gereği hem de hafifletilmeleri sebebiyle bazı zaafiyetleri bünyelerinde bulundurmaktadırlar (Kaps, 2008; Kim ve Yoon, 2014). Bu zafiyetler ve yan-kanal saldırıları kullanılarak şifrelemede kullanılan gizli anahtara ulaşılması veya sistemin çözülmesi mümkün olmaktadır (Kim ve Yoon, 2014; Williams, 2008; Zhao vd., 2009).

Bu alanda haberleşme güvenliğinin sağlanması için birçok çalışma bulunmaktadır ancak çalışmaların dayanağı olan en az kaynak tüketimiyle en çok güvenliğin sağlanması ve bunların performans kaybı olmadan gerçekleştirilmesi problemi bu alanda yapılacak olan her çalışmayı özel yapmaktadır. Bu durum, internete bağlı cihazların her geçen gün daha da artmasının ve uygulama alanlarının çok olmasının yanında sürekli gelişen teknoloji ile birlikte güvenliğin arttırılması için önerilebilecek yaklaşımların da artmasından kaynaklanmaktadır. Örneğin; bir nesne için kaynağın asgari düzeyde tutulması bir istek değil zorunluluk olabilirken aynı sisteme bağlı başka bir nesne için kullanılabilecek kaynak miktarı daha esnek iken güvenliğin en üst düzeye çıkarılması zorunluluk olabilmektedir.

Karşılaşılabilecek senaryoların çokluğu ve bir yaklaşımın her senaryoda doğrudan uygulanamaması; bu alanda yapılacak bir çalışmanın belirli senaryolar için çözüm olabileceğini veya belirli senaryolar için çözüm olamayacağını da gösterecektir. Bu nedenle literatürde olan çeşitli çözümlerin bir sistem üzerinde uygulanması ve aynı sistem için önerilen yeni bir yaklaşımın bu çözümlerle karşılaştırılması literatüre, yeni bir çözüm olarak olmasa dahi, doğrudan katkı sağlayacaktır.

Tez çalışmasında, öncelikle kuramsal temellerden bahsedilmiş olup, literatürde bulunan çalışmalar incelenmiştir. Yapılan inceleme sonucunda güvenilirliği kabul edilmiş ve bilinen

bir algoritma olan Grain 128a, önerilen bir hafif-siklet yöntem ile güçlendirilip, bir tanışma senaryosunda uygulanarak sistemin ZAS'a karşı güçlendirilmesi hedeflenmiştir.

2. IOT

IoT kavramı ilk olarak 1999 yılında Kevin Auston tarafından ortaya atılmış olsa da nesnelerin interneti konsepti 2003 yılından itibaren radyo frekanslı tanımlama (RFID) teknolojisinin gerek askeri gerek günlük yaşamda kullanımının artmasıyla popülerlik kazanmaya başlamıştır. Bu nedenle RFID teknolojisi IoT konsepti için temel teknoloji olarak kabul edilmektedir (Khalil ve Özdemir, 2018).

Her geçen gün daha da genişleyen bir kavram olan IoT etki ettiği alanların da genişlemesi sebebiyle çeşitli şekillerde tanımlanmaktadır. Sadece insan-nesne arası ilişkinin yanında nesne-nesne ilişkisinin de giderek gelişmesiyle insanların hayatını kolaylaştıran bu teknoloji Uluslararası Telekomünikasyon Birliği tarafından "Bilgi toplumu için küresel bir altyapı, mevcut ve gelişen, birlikte çalışabilir bilgi ve iletişim teknolojilerine dayanan (fiziksel ve sanal) şeyleri birbirinine bağlayarak gelişmiş hizmetler sunmak" olarak tanımlamaktadır (Union I.T., 2012). IoT Avrupa Araştırma Topluluğu, IoT'u "Kendi kendine yapılandırma yapabilen fiziksel ve sanal "şeylerin"; bir kimliğe, fiziksel yeteneklere, sanal kişiliklere sahip olduğu, akıllı arayüzleri kullanbildiği ve standart iletişim protokolleri vasıtasıyla sorunsuz bir şekilde bilgi ağına entegre olabildiği dinamik küresel bir ağ altyapısı" olarak tanımlamaktadır (IERC Cluster SRIA, 2014). Gubbi vd. (2013) IoT'u "Algılama ve çalıştırma cihazlarının birbirine bağlanması, birleşik bir çerçeve aracılığıyla platformlar arasında bilgi paylaşma yeteneği sağlanması, yenilikçi uygulamaların etkinleştirilmesi için ortak bir çalışma resminin ortaya koyulması" olarak tanımlamaktadır.

Sürekli gelişen teknoloji ile birlikte internet ve internete bağlı cihazların kullanımı IoT artan bir hızda yaygınlaşmakta olup 2025 yılında, dünya genelinde, yaklaşık 75.4 milyar IoT sisteminin var olacağı tahmin edilmektedir (Statista Research Department, 2016).

2.1 Bileşenler

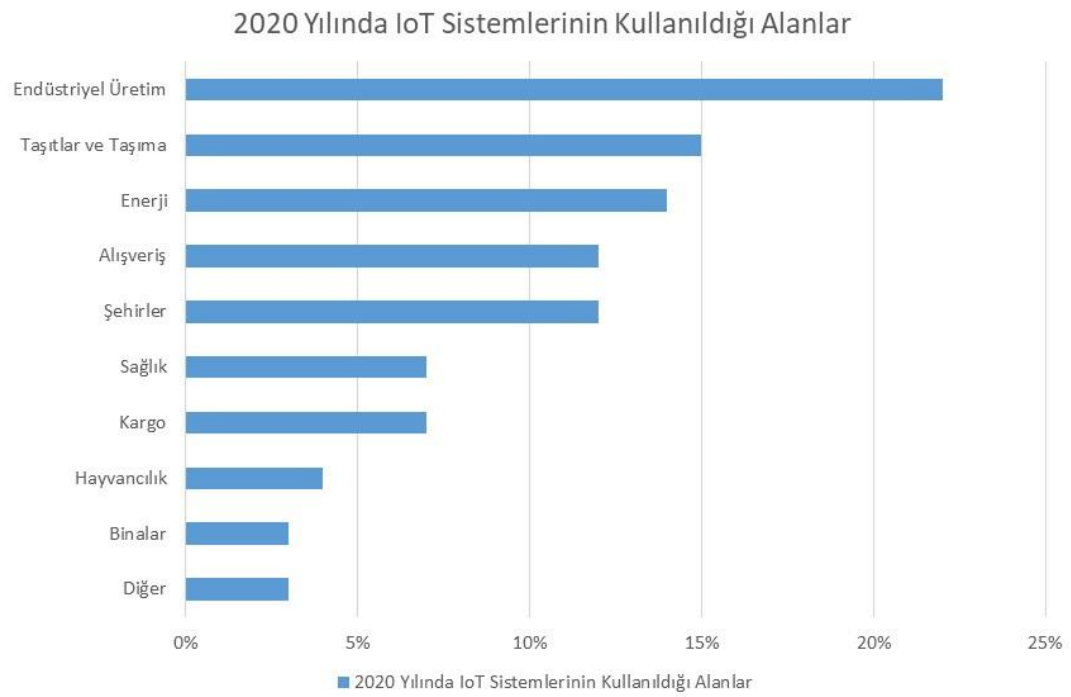
IoT beş temel bileşenden oluşmakta olup bu beş temel bileşen Tanımlama, Algılama, İletişim, İşleme ve Hizmetler olarak adlandırılmaktadır. Tanımlama yöntemleri, ağdaki her nesne için net bir kimlik sağlamak amacıyla kullanılmaktadır. Algılama; ağ içindeki ilgili nesnelerden veri toplamak ve bunları bir veri tabanına veya buluta göndermek için kullanılmaktadır. İletişim teknolojileri, belirli akıllı hizmetler sunmak için farklı nesneleri birbirine bağlamak amacayıla kullanılmaktadır. İşleme birimleri ve yazılım uygulamaları, IoT sistemlerinin beynini ve hesaplama yeteneğini temsil etmektedir. Hizmet bileşeni ise

IoT sistemlerini belirli bir amaca hizmet edecek şekilde kullanılmasıdır (Al-Fuqaha vd., 2015).

2.2 Uygulama Alanları

IoT artık bir hevesin daha ötesinde ivme kazanarak hayatımızı şekillendiren bir teknoloji olmaya başlamaktadır. Bu nesneler kullanılarak oluşturulan sistemler, insanlığın uygun ve bağlantılı bir yaşam tarzına öncülük etme, insan gücüne olan ihtiyacı azaltma ve insan hatalarına bağlı sorunların ortadan kaldırılması ihtiyacına karşı aktif olarak kullanılmaya başlanmıştır.

Günümüzde bu cihazların birbirlerine ve internete bağlı hale getirilip makine öğrenmimi, yapay sinir ağları gibi konseptlerle birleştirilerek bu cihazların veri toplamalarına, iletmelerine ve karar almalarına izin verilmektedir. Oluşturulan bu karmaşık sistemler akıllı ev teknolojileri, akıllı şehirler, otonom araçlar, çiftçilik, giyilebilir teknoloji, akıllı sağlık teknolojileri gibi isimlerle karşımıza çıkmaktadırlar. IoT sistemleri yaygın olarak birçok alanda kullanılmakla beraber Şekil 2.1.'de 2020 yılına ait en yaygın on kullanım alanına yer verilmiştir.



Şekil 2.1 : 2020 yılında IoT sistemlerinin en yaygın olarak kullanıldığı alanlar (Leuth, 2021).

Özellikle insan gücünün ve insana bağlı hataların asgari düzeyde tutulması istenen endüstriyel alanlarda diğer alanlara kıyasla IoT sistemlerine yüklenen sorumluluğun daha fazla olduğu görülmektedir.

2.3 Mimari

IoT için evrensel olarak mütabık olunan tek bir katmanlı yapı bulunmamaktadır. Çeşitli mimariler farklı araştırmacılar tarafından önerilmiş ve kullanılmıştır. Bazı araştırmacılara göre IoT mimarisi üç katmandan oluşmaktayken bazı araştırmacılar dört, bazı araştırmacılar ise beş katmanlı mimariyi savunmaktadırlar. Son dönemde artan güvenlik ihtiyacının karşılanması sebebiyle katman sayısının yediye kadar çıktığı görülebilmektedir (Kumar ve Mallick, 2018). Şekil 2.2'de son dönemde yapılan sıkça kullanılan katmanlı mimariler ve bunların katmanları görülmektedir.

Şekil 2.2 IoT mimarileri katmanları (Kumar ve Mallick, 2018).

Katmanlar arası benzerlik bulunmasına rağmen her katmanın her mimaride kullanılmaması, kullanım şekillerinin ve sıralamalarının değişmesi gibi farklılıklar görülmektedir.

2.3.1 Algılama katmanı (Perceptron layer)

Alıcı/Sensör katmanı olarak da bilinen bu katman bir bakıma IoT sistemlerinin duyu organlarıdır. Bir şeyleri algılama, tanımlama ve onlardan bilgi edinme sorumluluğu vardır. Bu katmanda kullanılan sensörler uygulama katmanındaki gereksinimlere göre seçilirler. Elde edilen veriler diğer katmanlar aracılığıyla anlamdırılarak işlenirler.

2.3.2 Ağ katmanı (Network layer)

Ağ katmanı insan vücudundaki sinirler gibi düşünülebilir. Algılama katmanında toplanan verilerin diğer katmanlara aktarılmasını sağlayan bu katman, üç katmanlı mimaride doğrudan Algılama Katmanı ile Uygulama Katmanı arasında bir köprü görevi görmektedir. Fiziksel nesnelerden toplanan verileri taşır ve iletir, aynı zamanda IoT nesnelerini, ağ aygıtlarını ve ağları birbirine bağlar.

2.3.3 Destek katmanı (Support layer)

Destek katmanı, genellikle dört katmanlı mimaride bulunan ve IoT sistemlerinin güvenliğinin arttırılması için kullanılan bir katmandır. Üç katmanlı mimaride doğrudan ağ katmanına gönderilen veri, destek katmanına sahip dört katmanda öncelikle destek katmanına sonrasında ağ katmanına gönderilir. Destek katmanının temel olarak iki sorumluluğu vardır. Bunlar; bilgilerin gerçek kullanıcılar tarafından gönderildiğini ve potansiyel bir tehdit taşımadığını onaylamaktır.

2.3.4 İşleme katmanı (Processing layer)

İşleme katmanı toplanan bilgilerin işlenmesini gerçekleştiren, hiçbir anlamı olmayan veya hasarlı bilgileri ayıklayan katmandır. Büyük IoT sistemlerinde kullanılan sensör çokluğu ve gelen verinin sürekliliği sebebiyle ortaya çıkan büyük veri sorununu çözmek amacıyla kullanılmaktadır.

2.3.5 Uygulama Katmanı (Application layer)

Uygulama katmanı, IoT teknolojisini kullanan uygulamaları tanımlar. IoT uygulamaları, akıllı evler, akıllı sağlık hizmetleri ve akıllı şehirler gibi sistemlere hizmet sağlamaktadırlar. IoT sistemleri tarafından verilen bu hizmetler alıcılar tarafından toplanan bilgilere ve bu

bilgilerin işlenmesine bağlı olduğundan dolayı her sistem özelinde farklılık göstermektedirler.

2.3.6 İş katmanı (Bussiness layer)

İş katmanı bir uygulamanın ve sistemin var oluş amacını temsil eder. Tüm sistemin yöneticisi gibi davranır ve IoT sistemlerini, iş ve kar modellerini yönetme ve kontrol etme gibi sorumlulukları vardır. Bilgilerin nasıl oluşturulabileceği, saklanabileceği ve değiştirilebileceği gibi sistem özelliklerini belirleme yeteneğine de sahiptir.

2.4 Haberleşme

Haberleşme, farklı nesnelerin veya farklı IoT sistemlerinin birbirlerine bağlandığı ve bilgi alışverişi yaptığı, IoT'un temel elementlerinden ve aynı zamanda da temel amaçlarından biridir. Haberleşme, cihazlar veya sistemler arası mesaj, dosya veya diğer bilgiler gönderilmesi ile sağlanır.

2.5.1 Kablosuz Haberleşme

Radyo Frekansı Tanımlama (RFID) (Want R., 2006), Bluetooth (McDermott-Wells P., 2004), Wi-Fi (Ferro ve Potorti, 2005), Yakın Alan İletişimi (NFC) (Want R., 2011), ZigBee (Wang vd., 2011) gibi kablosuz haberleşmeyi sağlayan çeşitli teknolojiler bulunmaktadır. Kablosuz haberleşmenin sağlanması için kullanılan her teknoloji çeşitli güvenlik önlemlerine ve güvenlik açıklıklarına sahiptirler.

- Bluetooth: Kısa mesafeli iletişim amacıyla kullanılan ve RF (radyo frekansı) tabanlı kablosuz haberleşme teknolojisidir (McDermott-Wells P., 2004).
- RFID: Radyo frekansları kullanılarak iletişimin sağlandığı kablosuz haberleşme teknolojisidir (Want R., 2006).
- NFC: Kısa mesafeli iletişim amacıyla kullanılan yüksek frekansa sahip kablosuz haberleşme teknolojisidir (Want R., 2011).
- ZigBee: Düşük kaynak kullanımına sahip düşük hızda veri aktarımına olanak sağlayan kısa mesafeli kablosuz haberleşme teknolojisidir (Wang vd., 2011).

- Wi-Fi: Şifreli veri iletimine sahip, yoğun kaynak kullanımıyla yüksek hızda veri aktarımına olanak sağlayan kablosuz haberleşme teknolojisidir (Ferro ve Potorti, 2005).

3. IOT SİSTEMLERİNDE GÜVENLİ HABERLEŞME

IoT sistemlerinde en sık tercih edilen haberleşme yöntemi kablosuz haberleşmek olmakla beraber, bu amacın gerçekleştirilmesi için kullanılan teknolojiler ve bu teknolojilerin doğuştan gelen güvenlik önlemleri çeşitli durumlarda yalın halleriyle yeterli gelemeyebilmekte veya kullanım kısıtlamaları sebebiyle her sistemde kullanılamamaktadırlar. Örneğin; Bluetooth teknolojisi kendisinden mesaj şifreleme yeteneğine sahipken uzun mesafeli haberleşmelerde bu teknolojinin kullanılması mümkün olmamaktadır. Bu nedenle, daha uzun mesafeli haberleşme ihtiyacına sahip sistemler için güvenlik açısından görece olarak daha kötü olan RFID teknolojisi tercih edilebilmektedir.

3.1 Güvenli Haberleşmenin Önemi

IoT sistemleri yapıları gereği birçok güvenlik açığına sahiptirler. Sistemi oluşturmak için kullanılan cihazların fazlalığı ve çeşitliliği, kaynak kullanımı kısıtlı olan nesneler olmaları, genellikle çalışılabilirliğin ve çalışma verimliliğinin güvenli ve stabil sistemler oluşturmak yerine tercih edilmesi gibi sebepler bu güvenlik açıklarına temel hazırlamaktadır. Saldırılar genellikle akıllı ev sistemleri, işyerlerinde bulunan akıllı güvenlik sistemleri ve akıllı tıbbi sistemler gibi kritik sistemlerde bulunan ve saldırılara sistem genelinden daha az güvenli olan IoT cihazlarını hedef alırlar. Bu cihazları hassas noktalara erişim amacıyla bir basamak olarak veya sistemden bilgi sızdırma amacıyla kullanılmaktadır. Bu nedenle kullanılan cihazların sistemdeki rolünün büyük veya küçük olmasına bakılmaksızın ciddi tehdit oluşturabileceği unutulmamalıdır. IoT sistemlerine yapılan saldırıların ciddiyeti geçmişte yaşanmış olaylar göz önüne alındığında daha da iyi anlaşılmaktadır.

Geçtiğimiz yıllarda Amazon'a bağlı bir şirket olan Ring iki farklı güvenlik zaafiyeti ile gündeme gelmiş, saldırganlar zayıf kimlik bilgilerini kullanarak çeşitli evlerin etrafındaki kameralara erişmiş ve sonrasında haberleşme zaafiyetlerini de kullanarak evlerde bulunan hoparlörleri ve mikrofonları kullanarak ev sahipleri ile iletişime geçmişlerdir (Vanwell, 2021).

IoT cihazları doğaları gereği, genellikle, herhangi bir şifreleme olmadan verileri bir bulut sunucuya iletirler. St. Jude Medical's hastanesinde yapılan bir çalışmada bir IoT cihazı kullanılarak implante edilebilir kardiyak cihazlarına erişilebildiği görülmüştür. Kardiyak cihazlarına erişimin olması durumunda ise saldırganın hastaların pillerini tüketebileceği veya hastalara şok uygulayabileceği belirtilmiştir (Vanwell, 2021).

Check Point'teki güvenlik araştırmacıları Yaniv Balmas ve Eyal Itkin, faks makinelerinin yalnızca bir telefon hattı ve faks numarası kullanarak bir şirket ağı üzerinden veri çalmasına izin verebilecek güvenlik açıklarına sahip olduğunu keşfetmişlerdir. Araştırmacılar ayrıca DEFCON 26 konferansında bir Hewlett Packard yazıcıdaki güvenlik kusurlarından nasıl yararlanabildiklerini de göstermişlerdir. Potansiyel tehdidi açıklayan araştırmacılar, saldırganların özel olarak oluşturulmuş kötü amaçlı yazılım kodlu görüntü dosyalarını faks yoluyla hedeflenen ağlara gönderebileceklerini söylemiş ve faks makinesindeki kötü amaçlı yazılımın dosyaların şifrelerini çözmeye ve bunları kendi belleğine yüklemesine olanak tanıyabileceğini ortaya koymuşlardır (Srinivas, 2020).

3.2 Güvenlik Unsurları

CIA üçlüsü olarak da bilinen bu unsurlar, bir sistem içindeki bilgi güvenliği politikalarını yönlendirmek için tasarlanmış bir modeldir. Model bazen AIC üçlüsü olarak da adlandırılır. Bu model,

- Gizlilik (Confidentiality)
- Bütünlük (Integrity)
- Erişilebilirlik (Availability)

olmak üzere üç unsurdan oluşmaktadır. Bu bağlamda temel olarak gizlilik, bilgiye erişimi sınırlayan kurallar bütününü, bütünlük bilginin güvenilir ve doğru olduğunu, kullanılabilirlik ise yetkili cihazlar tarafından bilgiye güvenilir erişimin sağlanmasını temsil etmektedir.

3.2.1 Gizlilik

Gizlilik, bilgilerin yetkisiz erişim girişimlerini önlemek için tasarlanmıştır. Verilerin yanlış ellere geçmesi durumunda verilebilecek hasarın miktarına ve türüne göre kategorize edilmesi yaygındır. Bu unsurun IoT sistemlerine uygulanabilmesi için sistemin bir bütün olarak ele alınması önemlidir. Verilerin iletilmeden önce şifrelenmesi ve şifreleme anahtarlarının güvenliğinin sağlanması bilgilerin korunması için kullanılabilecek yöntemlerden biridir (Chinau vd., 2018).

3.2.2 Bütünlük

Bütünlük, sistemin tüm yaşam döngüsü boyunca verilerin tutarlılığını, doğruluğunu ve güvenilirliğini korumayı amaçlamaktadır. Veriler transit olarak değişitirilmemeli ve verilerin yetkisiz kişilerce değiştirilmemesini, değiştirilse bile bu verilerin tutarlılığının sağlanması zorunluluğunu getirmektedir.

IoT cihazları genel olarak hassas ve değerli bilgilerin elde edilmesini, işlenmesini veya iletilmesini sağlamaktadırlar. Bu nedenle taşınan verilerin tutarlı olması ve güvenilirliğinin tartışılamaz olması gerekmektedir. Sistemin çalışması sırasında bu durumun sağlanamaması sonucunda yanlış veya hasarlı bilgilerin veri akışına dahil olması durumunda bu veriler sistemin tasarım amacına uygun olacak şekilde kullanılamaz olacaktır. Dolayısıyla bu verileri kullanan hizmetlerin güvenilirliği sorgulanır hale gelecektir. Örneğin; bir akıllı yangın söndürme sisteminde verilerin kasıtlı yahut kasıtsız olarak tahrif edilmesi sonucunda tüm sistem alarm moduna geçebilir veya nabız ölçümü yapan bir cihaz hasta hakkında yanıltıcı bilgiler verebilir. Bu tarz durumlarla karşılaşılmaması için bütünlük ilkesinin sağlandığından emin olunmalıdır.

3.2.3 Erişilebilirlik

Erişilebilirlik, bilgilerin yalnızca yetkili cihazlar tarafından kolay ve sürekli şekilde erişilebilir olmasını sağlamak amacıyla kullanılan unsurdur. Bilgilerin tutarlılığının ve gizliliğinin sağlanması için erişilebilirlik unsurunun IoT sistemlerinde doğru bir şekilde uygulanması gerekmektedir. Bu, donanım ve teknik altyapının ve bilgileri tutan ve görüntüleyen sistemlerin düzgün bir şekilde bakımını da içermektedir.

IoT sistemlerinde erişilebilirik, yetkili kullanıcıların veya cihazların sistem tarafından sağlanan tüm hizmetlere ve verilere ihtiyaç duyulduğu anda doğrudan erişiminin sağlanması anlamına gelmektedir (Mosenia ve Jha, 2016). Bir akıllı yangın söndürme sisteminde, sistemin veri aktarımının kesilmesi veya aktarım sürecinde yaşanan gecikme can ve mal kaybıyla sonuçlanabilir. Erişilebilirlik ilkesinin bütünlük ve gizlilik unsurlarından kaynaklanabilecek sorunlar sebebiyle sekteye uğrayabileceği ve bu durumun yaşanmaması için sistemin tüm unsurlarla bütün olarak değerlendirilerek tasarlanması önemlidir.

3.3 Geleneksel Güvenlik Önlemlerinin Uygulanamaması

Cihazlarda depolanan ve internet üzerinden iletilen bilgilerin güvenli bir şekilde iletilmesinin ve güvenli bir şekilde korunmasının büyük bir problem olduğu bilinmektedir. IoT sistemlerinin güvenliğinin sağlanması, bu teknolojinin doğasına uygun olacak şekilde yeni yöntemlerin geliştirilmesi veya mevcut sistemlerin hafifletilerek sistemlere entegre edilmesi ile mümkün olmaktadır. Bunun nedeni, geleneksel bilgisayar sistemleri ile IoT aygıtlarının bellek, işlevsellik, güç kaynağı ve veri yeteneği gibi kaynaklar açısından farklı olmasıdır (Wei vd., 2016). Bu fark, başlangıçta geleneksel bilgisayar sistemlerini korumak için tasarlanan konvansiyonel korunma yöntemlerinin IoT sistemlerini korumak için kullanılmasını zorlaştırır. IoT sistemlerinin güvenliği, cihazların fiziksel dünyaya doğrudan bağlı olmaları, aynı zamanda gerçek zamanlı veri akışına sahip olmaları ve özel bilgilerin taşınması sebebiyle oldukça güçlü olmalıdır. Bu cihazların kullanım amaçlarına göre uygulanacak yöntemlerin veri akış hızını ve veri sağlığını etkilemeyecek şekilde tasarlanması gerekmektedir (Katagi ve Moriai, 2008).

3.4 Potansiyel Tehditler

IoT sistemleri kullanılan oluşturulurken kullanılan mimariye göre farklılıklar gösterebilen tehditlerle karşı karşıya kalmaktadırlar. Karşılaşılan bu tehditler uygulanabilmeleri için hedeflenen katmana göre fiziksel erişim veya ağ erişimi gibi özelliklere ihtiyaç duymaktadırlar. Örneğin; güç analizi kullanılan cihazlara veya sensörlere fiziksel erişim gerektirirken, zararlı kod enjekte edilerek yapılacak bir saldırı genellikle ağ erişimine ihtiyaç duymaktadır. IoT sistemlerinin en temel özelliklerinden olan mobil ve kompakt tasarımlar sebebyile genellikle fiziksel erişimin mümkün olmaması veya erişilse bile kolaylıkla fark edilmesi sebebiyle fiziksel erişime ihtiyaç duyan saldırıların uygulanması daha zor olmaktadır. Bu katmanlara yapılan saldırılardan bir kısmı yan kanal saldırıları, DoS/DDoS, geleneksel bilgisayar ağlarına yapılan saldırılar olabilecekken bir kısmı ise yalnızca IoT ve sensörler özelinde uygulanabilirliği olan saldırılardır (Frustaci vd., 2017). Bir saldırının birden fazla katman için tehdit oluşturabileceği unutulmamalıdır.

3.4.1 Algılama katmanı tehditleri

Algılama katmanı, genellikle sensörler ve mikrodenetleyicilerden oluşmaktadır (Kumar vd., 2017). Bu elemanlar sayesinde ortam hakkında veri toplama işleminin gerçekleştirildiği

katman olduğu için; bu katmanda işlememiz gereken tüm verilerin toplandığından, veri güvenliğine dikkat edildiğinden ve bu verilerin gerçek bir bilgi olduğundan emin olunması gerekmektedir (Rizvi vd., 2018). Bu katman hedeflenerek gerçekleştirilebilen saldırılardan bazıları şöyledir:

- Kurcalama Saldırıları: Bu saldırılar genellikle donanım bileşenlerine odaklanır ve saldırganın IoT sisteminde fiziksel olarak bulunması ve sistemi meşgul etmek için sürecine devam etmesi gerekir (Frustaci vd., 2017).
- Yönlendirme Saldırıları: Veri toplama ve iletme işleminde, ara kötü amaçlı düğüm yönlendirme yolunu değiştirir ve sistemi meşgul eder (Singh vd., 2020).
- Veri Aktarımı Saldırıları: Koklama (Sniffing), Ortadaki adam (Man in the Middle) gibi çeşitli saldırılar veri aktarımı sırasında bütünlük ve gizliliğe saldırır (Singh vd., 2020).

3.4.2 Ağ katmanı tehditleri

Bu katman algılama katmanından aldığı verileri ağ üzerinden iletimini sağlamadığı için sıklıkla hedef alınmaktadır. Bu katman için en önemli hedefler veri gizliliğinin korunması, bütünlüğünün sağlanması ve alınan verilerin hızlı bir şekilde gerekli katmanlara iletilmesinin sağlanmasıdır ancak gelen verinin büyüklüğü ve sürekliliği bu hedefelere ulaşılmasını zorlaştırmaktadır.

- Yönlendirme Saldırıları: Veri toplama ve iletme işlemi sırasında ara kötü amaçlı düğümler yönlendirme yolunu değiştirebilir ve sisteme bulaşabilir.
- Sahte Yönlendirme Bilgileri: Saldırganlar, ağlardaki trafiği rahatsız etmek için IP adresini taklit eder, değiştirir veya yeniden oynatır: sonuçta yönlendirme döngüleri, sahte hata mesajı ve kısaltılmış yollar vb.
- DoS/DDoS Saldırıları: DoS ve DDoS saldırısında saldırgan bağlantı isteyip erişimi aldıktan sonra ağ hizmetlerini kullanılamaz hale getirmeyi amaçlar. Sık ve sürekli paket gönderimi sağlayarak sisteme kaldıramayacağından fazla istek göndererek sistemi yavaşlatmayı veya durdurmayı amaçlar.
- Ortadaki Adam Saldırısı: IoT ağındaki tüm düğümler iletişim içinde ve bir ağ geçidine bağlıdır. Saldırgan bu ağ geçidine saldırarak verilerin iletimini engellemeyi veya verilere erişim sağlamayı hedeflemektedir.

3.4.3 Uygulama katmanı tehditleri

Kullanıcıdan gelen isteklerin yerine getirildiği ve aynı zamanda bulut katmanı olarak da bilinen ağ katmanında, verilerin gizliliği bu katman için en büyük endişe kaynağıdır. Farklı cihazların, korunması gereken, büyük miktarda veri alışverişi yapması, bu katmanın sorumluluğu olan; kullanıcı isteklerine göre gerekli verilerin sağlanması ve gizlilik görevlerinin yerine getirilmesinin önündeki en büyük zorluktur (Rizvi vd., 2018). Bu katmanda gizlilik şartının sağlanması için TLS, SSL, DNS vb. veri gizleme teknikleri uygulanabilmektedir. Yine aynı zamanda Websocket, XMPP, DDS ve http protokollerinden bazıları veri gerçekliğini, veri gizliliğini ve veri bütünlüğünü sağlamak amacıyla kullanılabilmektedir (Swamy vd., 2017). Bu katman hedeflenerek gerçekleştirilebilen saldırılardan bazıları şöyledir:

- DoS/DDoS saldırıları: DoS ve DDoS saldırısında saldırgan bağlantı isteyip erişimi aldıktan sonra ağ hizmetlerini kullanılamaz hale getirmeyi amaçlar. Sık ve sürekli paket gönderimi sağlayarak sisteme kaldıramayacağından fazla istek göndererek sistemi yavaşlatmayı veya durdurmayı amaçlar.
- Ara Saldırı: Bu saldırı, kötü amaçlı düğüm olan ve yönlendirme yolunu değiştirebilen bir ara düğüm kullanılması ile gerçekleştirilebilir. Kullanıcı isteklerinin yerine getirilmemesi veya yanlış şekilde yerine getirilmesini hedefler.
- Koklama Saldırısı: Saldırgan tarafından, cihazlar ve ağdan gelen tüm bilgileri toplayan bir ara koklama uygulaması kullanılarak yapılır. Sistem hakkındaki bilgi toplamayı veya kullanıcı verilerine erişmeyi hedefler (Swamy vd., 2017).

3.4.5 Zamanlama analizi saldırıları

Zamanlama analizi (ZA), yapılan işlemlerin değişken şartlarda sızdırdıkları zamanlama bilgilerini kullanarak yapılan işlemler veya sistem hakkında bilgi edinmek amacıyla yapılan analiz türüdür (Kocher, 1996).

ZAS, bir kriptografik sistemin çeşitli koşullarda gerçekleştirilirken geçen sürelerin farklılıkları kullanılarak yapılan bir yan kanal saldırısı türüdür. Girdiye bağlı olarak değişen işleme süresine sahip algoritmaların sızdırdığı zamanlama yan-kanal bilgisinden faydalanılır (Kocher, 1996; Janke ve Laackmann 2002; Popp vd., 2007). Farklı işlemci komutları uygulanırken, toplama, üs alımı veya bölme gibi matematiksel işlemler gerçekleştirilirken farklı işleme süreleri ortaya çıkmaktadır. Örneğin; kullanıcıdan girdi alınarak yapılan bir üs

alma işlemi büyüyen girdiler için daha uzun işleme sahip olacaktır. Bu durum algoritma çözümlenmese bile gizlenmek istenilen veri hakkında bilgi sahibi olunmasını sağlar. Genellikle fiziksel müdahaleye ihtiyaç duyulmadan gerçekleştirilebilmeleri sebebiyle ZAS, IoT alanında uygulanması nispeten daha kolay olan bir saldırı türüdür.

Bu alanda yapılan çeşitli çalışmalar işlemlerin yürütülme sürelerindeki farklardan faydalanarak elde edilen zamanlama yan-kanal bilgisi ile gizlenmek istenen bilgiye ulaşmayı başarmışlardır (Kocher, 1996; Janke ve Laackmann 2002; Perianin vd., 2020; Lerman vd., 2011; Won vd., 2021).

Kocher (1996), yapmış olduğu çalışmada algoritma adımlarının birinde gizli anahtara bağlı olarak üs alma işlemi sırasında işlem süresinin değişkenliğinden faydalanılarak yapılan bir ZAS örneği görülmüştür. Yine aynı şekilde Janke ve Laackmann (2002), yapmış oldukları çalışmada ortaya koyulan senaryoda anahtara bağlı olarak ortaya çıkan dallanma işlemleri sonucunda yan-kanal zamanlama bilgisi kullanılarak anahtara ulaşılabileceği ortaya koyulmuştur.

ZAS tek başına uygulanabileceği gibi, diğer yan-kanal saldırıları ile birlikte de uygulanabilir. Walter ve Thompson 2001 yılında yapmış oldukları çalışmada zamanlama analizi ve güç analizi saldırıları birlikte kullanılarak modüler üs alıcılarla ilgili gizli bilgilere ulaşılmıştır (Walter ve Thompson, 2001).

4. IOT SİSTEMLERİNDE KRİPTOGRAFİ

IoT'de kullanılan kriptografik temeller, işlem gören mesajın ve alt yapının temel güvenlik hedeflerine ulaştırılması amacıyla kullanılırlar (Hoyland vd., 2014). Bu hedefler altı ana başlıkta kategorilendirilebilirler:

1. Gizlilik: Mesaj sadece onaylanmış öğelere, istemcilere ve yönetimlere sunulur. Özel bilgiler ve anahtarlar güvenlik nitelikleri onaylanmamış unsurlardan korunulmalıdır (Hosseinian vd. 2019).
2. Bütünlük: IoT sistemlerinde, çeşitli uygulamalar farklı yetkinlik gereksinimlerine sahip olabilirler (Hosseinian vd. 2019; Kang vd. 2014). Bu nedenle sistemin ihtiyaçlarına göre tasarlanmalıdır.
3. Kimlik Doğrulama: Ortak ağlar gibi güvenli olmayan bir ağ üzerinden bilgi alışverişi yapılırken verileri korumak ve erişimi kontrol etmek için bir IoT sistemini oluşturan aygıtların birbirlerini tanıyabilmeleri adına kullanılan bir modeldir (El-Hajj vd., 2019; Kim ve Lee, 2017). Sistem tarafından tanınmayan aygıtların sisteme katılması engellenmelidir.
4. Yetkilendirme: IoT sistemindeki her uç noktanın kimliklerinin doğrulanması sonucunda sistemde çeşitli yetkilerle donatılarak sisteme katılması işlemidir (Kim ve Lee, 2017). Her aygıt yetkilendirme işlemi sırasında kendi kimliğine uygun şekilde yetkilendirilmelidir.
5. Kullanılabilirlik: Kullanılan altyapı, sistemin ihtiyaçlarını sürekli olarak ve güvenli bir şekilde sağlamalıdır (Xiong vd., 2018).
6. Hesap verebilirlik: IoT sistemlerinin takip edilebilir olması adına yapılan işlemlerin kayıt altında anlaşılabilir halde tutulması gerekmektedir. Tutulan bu verilerin sistem dışında ve yine gizlilik esaslarını sağlayacak şekilde saklanması gerekmektedir.

4.1 Kriptografi

Kriptografi kelimesinin kökeni Yunanca "kryptos" (gizli) ve "graphein" (yazıt) kelimelerine dayanmakta olup steganografinin aksine veri yapısını değiştirerek verinin gizlenmesini sağlamaktadır (Pawlan, 1998). 4,000 yıllık bir geçmişe sahip olan kriptografi, geçmişte, mesajın bir yerden diğerine taşındığı süre boyunca mesajın içeriğini korumak için mesajları okunamayan şekil gruplarına dönüştürmekle ilgilenirken; günümüzde, temel mesaj gizliliğinden, mesaj bütünlüğü kontrolü, gönderen/alıcı kimlik doğrulaması ve dijital

imzaların bazı aşamalarını içerecek şekilde genişlemiştir (Cohen, 1995). Bu durum, geniş çaplı korumaya ihtiyaç duyan IoT sistemlerinde kriptografik yöntemlerin tercih edilmesinde önemli bir etken olmuştur (Thakor vd., 2020). Şifreleme yöntemleri kullanılan anahtarın özelliklerine ve çeşidine göre Simetrik Şifreleme, Asimetrik Şifreleme ve Hibrit Şifreleme olarak kategorilendirilebilirler. Asimetrik ve hibrit şifreleme teknikleri IoT cihazlarının düşük bilgisayar gücüne sahip olmaları sebebiyle pek tercih edilmemektedir.

4.1.1 Simetrik şifreleme

Simetrik şifreleme, bilgileri hem şifrelemek hem de şifresini çözmek için yalnızca bir anahtarın (gizli anahtar) kullanıldığı bir şifreleme türüdür. Simetrik şifreleme yoluyla iletişim kuran cihazlar, şifre çözme işleminde kullanılabilmesi için anahtarı değiştirmelidir. Bu şifreleme yöntemi, iletileri şifrelemek ve şifresini çözmek için bir ortak ve bir özel anahtar çiftinin kullanıldığı asimetrik şifrelemeden farklıdır.

Simetrik şifreleme algoritmaları kullanılarak veriler, şifresini çözmek için gizli anahtara sahip olmayan herkes tarafından anlaşılamayan bir forma dönüştürülür. Anahtara sahip olan hedeflenen alıcı iletiyi aldıktan sonra algoritma, iletinin özgün ve anlaşılabilir biçimine döndürülmesi için eylemini tersine çevirir.

Simetrik şifreleme algoritmalarında genellikle karışıklık ve difüzyon olarak isimlendirilen iki strateji kullanılmaktadır. Karışıklık, şifreli metnin orijinal düz metin hakkında hiçbir ipucu vermediğini garanti ederken; difüzyon, düz metnin fazlalığını satırlara ve sütunlara yayarak düz metnin fazlalığını geliştirmek amacıyla kullanılan bir stratejidir. Çizelge 4.1'de bu yöntemlerin karşılaştırılması görülmektedir.

Çizelge 4.1 : Difüzyon ve karışıklık stratejilerinin karşılaştırılması.

Karışıklık	Difüzyon
Zayıf şifreli metinler oluşturmak için kullanılan bir şifreleme tekniğidir.	Şifreli düz metinler oluşturmak için kullanılır.
Yerine koyma algoritmaları ile yapılır.	Taşıma algoritmaları ile yapılır.
Şifrelenmiş metnin içindeki bir bit değiştirilirse metindeki bitlerin çoğu veya tümü de değişir.	Düz metnin içindeki bir görüntü değiştirilirse, şifreli metnin içindeki görüntünün çoğu veya tamamı değişir.
Sonuç olarak belirsizlik artar.	Sonuç olarak fazlalık artar.

Simetrik şifreleme daha eski bir şifreleme yöntemi olsa da, veri boyutu ve yoğun CPU kullanımı ile ilgili performans sorunları nedeniyle ağları olumsuz etkileyen asimetrik şifrelemeden daha hızlı ve daha verimlidir. Daha iyi performans ve daha hızlı simetrik şifreleme hızı (asimetrik şifrelemeye kıyasla) nedeniyle, simetrik şifreleme genellikle büyük miktarda veriyi toplu olarak şifrelemek için kullanılır (Simmons G. J., 1979). Genellikle kullanılan şifreleme yöntemine göre;

1. Blok Şifreleme
2. Dizi Şifreleme

olmak üzere iki başlık altında toplanırlar. Blok şifreleme algoritmalarında veri ayarlanan bit uzunlukları, belirli bir gizli anahtar kullanılarak veri blokları halinde şifrelenir. Veriler şifrelenirken, sistem tüm blokları beklerken verileri belleğinde tutar. Dizi algoritmalarında ise veriler sistem belleğinde saklanmak yerine akış esnasında bit-bit olarak şifrelenir (Sharif ve Mansoor, 2010). Çizelge 4.2'de bu iki simetrik şifreleme yönteminin karşılaştırılması görülmektedir.

Çizelge 4.2 : Blok şifreleme ve dizi şifreleme algoritmalarının genel özellikleri.

Blok Şifreleme	Dizi Şifreleme
Düz metnin bloğunu tek seferde alarak düz metni şifreli metne dönüştürür.	Tek seferde 1 bayt düz metin alarak düz metni şifreli metne dönüştürür.
Genellikle 64 ve üzeri bit kullanır.	Genellikle 8 bit kullanır.
Genellikle şifreleme işlemi karmaşıklığı daha azdır.	Genellikle şifreleme işlemi karmaşıklığı daha fazladır.
Difüzyon ve karışıklık kullanılır.	Karışıklık kullanılır.
Şifrelenmiş metni tersine çevirmek zordur.	Şifrelenmiş metni tersine çevirmek daha kolaydır.
Genellikle ECB ve CBC algoritma modları kullanılır.	Genellikle CFB ve OFB algoritma modları kullanılır.
Genellikle rail-fence tekniği ve transpozisyon tekniği gibi yer değiştirme teknikleri ile çalışır.	Genellikle sezar şifreleme ve poligram ikame şifresi gibi yer değiştirme teknikleri ile çalışır.
Dizi şifrelemeye göre daha yavaştır.	Blok şifrelemeye göre daha hızlıdır.

4.1.2 Asimetrik şifreleme

Simetrik şifrelemenin aksine asimetrik şifreleme, verileri iki ayrı ancak matematiksel olarak bağlı şifreleme anahtarları kullanarak şifreler ve çözümler. Bu anahtarlar "Genel/Ortak Anahtar" ve "Özel Anahtar" olarak isimlendirilmekte olup birlikte "Ortak ve Özel Anahtar Çifti" olarak adlandırılırlar. Asimetrik şifrelemede, ortak anahtar şifreleme için kullanılırken şifreleme için kullanılırken, özel anahtar şifre çözme işlemi için kullanılmaktadır. Adından da anlaşılacağı gibi, özel anahtar yalnızca kimliği doğrulanmış alıcının ileti şifresini çözebilmesi için özel ve gizli olması amaçlanmıştır. Bu şifreleme yönteminin temelinde bir kriptografik şifreleme algoritması yatar. Bu algoritma bir anahtar çifti oluşturmak için bir anahtar oluşturma protokolü (matematiksel bir fonksiyon) kullanır. Anahtarlar arasındaki ilişki bir algoritmadan diğerine farklılık göstermektedir (Simmons G. J., 1979).

Şifreleme açısından daha güçlü olmasına rağmen asimetrik şifreleme yöntemlerii genellikle IoT sistemlerinde bulunandan daha fazla işlem gücü, bellek ve depolama alanı gibi kaynaklar gerektirdiğinden dolayı sıklıkla doğrudan kullanılmamaktadır. İhtiyaç duyulan kaynak kapasitesinin yanında IoT sistemlerinde sıkça kullanılan akan verinin şifrelenmesi ihtiyacı için yavaş kalmaları sebebiyle de tercih edilmemektedirler.

4.1.3 Hibrit şifreleme

Hibrit şifreleme, iki veya daha fazla şifreleme yönteminin birlikte kullanılmasıyla ortaya çıkan güçlendirilmiş bir şifreleme yöntemidir. Her şifreleme yönteminin daha güçlü taraflarından faydalanmayı amaçlamaktadır. Sıklıkla bir asimetrik ve bir simterik şifreleme yönteminin birlikte kullanılmasıyla oluşturulan hibrit şifreleme yöntemleri, simetrik şifrelemenin hızından ve asimetrik şifrelemenin şifreleme gücünden faydalanmayı hedeflemektedirler. Hibrit şifreleme yöntemleri, genel ve özel anahtarların gizliliği korunduğu sürece en güvenli şifreleme yöntemleri olarak kabul edilmektedirler (Mushtaq vd., 2017).

4.2 Hafif Siklet Kriptografi

IoT'de geleneksel güvenlik önlemlerinin uygulanamamasının önündeki engeller şifreleme algoritmaları için de geçerlidir. Geleneksel bilgisayar işlemlerinde kullanılan konvansiyonel şifreleme algoritmalarının düşük kaynaklı cihazlara doğrudan uygulanmaları:

- Boyut
- Kaynak İhtiyacı
- Güç Tüketimi
- İşleme Hızı

gibi faktörlere bağlıdır (Eisenbarth vd., 2007). IoT sistemleri, doğaları gereği genellikle, asgari fiziksel boyut, asgari güç tüketimi, asgari kaynak ihtiyacı ve en azami işleme hızına sahip olacak şekilde tasarlanmalıdırlar. Bu nedenle güvenli sistem tasarımında önemli basamaklardan biri olan güvenilirliği genel olarak tescillenmiş ve herkesçe bilinen yöntemlerin, IoT sistemlerine uygun hale getirilerek uygulanmaları gerekmektedir (Dutta vd., 2019). Bu da kriptografik yöntemler için, tescillenmiş şifreleme algoritmalarının

hafifletilerek uygulanması veya yeni hafif çözümlerin ortaya çıkarılması anlamına gelmektedir.

4.3 IoT Sistemlerinde Kriptografi Uygulamaları

IoT'de web arayüzü yapılandırması, ağ servisleri, yetkilendirme, gizlilik, veri taşınması gibi güvenlik zaafiyeti oluşturan çeşitli alanlar bulunmaktadır. Bu zaafiyetlerden faydalanmayı hedefleyen saldırılar Andrea vd. (2015) tarafından yazılım, fiziksel ve ağ olmak üzere hedef aldıkları bölgeye göre üç ayrı kategoride toplanmışlardır.

Zhou ve Chao (2011) yapmış oldukları çalışmada, şifreleme işlemini, mesaj metninin bir algoritma ile anlaşılamaz ve tahmin edilemez şekle dönüştürülmesi olarak tanımlamıştır. Şifreleme işleminin bu şartı sağlaması halinde mesaja ulaşılsa bile şifreleme yönteminin çözülemediği sürece mesajın açık içeriğine ulaşılamayacağını belirtmişlerdir. Bu yaklaşım IoT sistemleri ve kriptografi ilişkisinin temellerinden olup günümüzde hala kabul görmektedir.

Fan vd. (2017) çalışmalarında yeni bir nesnenin genel ağa katılması sırasında güvenlik zaafiyetlerinin bulunabileceğini ortaya koymuşlardır. Haberleşme esnasında şifrelemeyi sağlayan ağ anahtarının bilinen bir bağlantı anahtarıyla şifrelenip gönderildiğini gördükten sonra bu anahtarla paketlerin şifrelerini çözmüş ve ağ anahtarına ulaşmayı başarmışlardır.

IoT sistemlerinde cihazlar kendi aralarında ve kullanıcılarla sıklıkla ortak internet ağları üzerinden iletişim kurarlar. Genellikle düşük bellek, düşük güç ve düşük işlem yeteneklerine sahip olduğundan dolayı daha az kaynak kullanımı ile daha güçlü şifreleme çözümleri tasarlanması gerekmektedir. IoT sistemlerinin birçok senaryoda kullanılabilir olmaları sebebiyle şifreleme çözümleri de yine her senaryo için kendi özelinde tasarlanmalıdır. Mukherjee (2015) yapmış olduğu çalışmada IoT'de düşük kaynak kullanımında şifreleme işleminin gerçekleştirilmesinin bir zorunluluk olduğunu belirtmiştir.

Kim (2017), düşük sistem özellikleri sebebiyle bilgisayar tabanlı konvensiyonel kriptografik çözümlerinin birçoğunun tam olarak doğrudan uygulanamadığını, uygulanan çözümlerin birçoğunun da sistem üzerindeki yükü çokça artırdığını ortaya koymuş ve yapmış olduğu çalışmada bu soruna çözüm üretmek için daha düşük güç tüketimine ihtiyaç duyan bir hafif siklet algoritma önermiştir.

Hafif siklet kriptografik şifreleme, RIFD etiketleri, sensörler, temassız akıllı kartlar vb. Dahil olmak üzere kısıtlı ortamlarda uygulamalar için uyarlanmış, spesifik senaryolar için spesifik

çözümler öneren kriptografik algoritmalar ve protokollerdir (Katagi ve Moriai, 2008). Genellikle bilgisayar tabanlı kriptografik çözümlerin hafifletilmeleriyle IoT sistemlerine entegre edilmişlerdir. En sık kullanılan hafif siklet kriptografik şifreleme algoritmaları genel olarak simetrik algoritmalar ve asimetrik algoritmalar olmak üzere iki grupta sınıflandırılabilirler. Bu iki şifreleme yönteminin arasındaki en temel fark, simetrik şifreleme algoritmalarında tek anahtar kullanılırken, asimetrik şifreleme algoritmalarında birbirinden farklı ancak birbiriyle bağlantılı iki farklı anahtar kullanılmasıdır. IoT sistemleri için simetrik anahtarların üretimi ve dağıtımı; üretim sırasında anahtarın cihaza gömülmesi, donanımı tahrif edilmekten korumak için güvenlik modelinin sağlanması, çalışması sırasında anahtarların işlenmesi ve daha iyi kriptografik temellerin seçimi gibi alt görevlere sahip temel iki görevdir. Bu nedenle kısıtlı kaynaklara sahip IoT sistemleri için hafif siklet ortak anahtar şifrelemesine sahip olmak düşük kaynak tüketiminde etkin şifreleme sağlamaya olanak sağlayacaktır (Rao ve Prema, 2020).

AES (Advanced Encryption Standard), NIST (National Institute of Standards and Technology) tarafından standart olarak kabul edilmiş (PUB, 2001), 128-bitlik blok ve 128, 192 ve 256 bit uzunluğunda bir anahtar kullanılarak uygulanan bir hafif siklet kriptografik şifreleme algoritmasıdır (Moradi vd., 2011). Bu algoritmanın asgari fiziksel alan ihtiyacı, bilinen halinden yaklaşık olarak 23% daha düşük (Moradi vd., 2011) olmasına rağmen, Bansod vd. (2014) tarafından çeşitli gerçek-zamanlı uygulamalar için kaynak kullanımı açısından hala ağır kaldığı ortaya koyulmuştur.

Standart olarak kabul edilen bir başka algoritma olan PRESENT 64-bitlik bir blok ve 80-bitlik ve 128-bitlik iki farklı anahtar çeşidine sahip olan bir hafif siklet kriptografik şifreleme algoritmasıdır (Bogdanov vd., 2007). RECTANGLE yapısı PRESENT'a benzeyen ancak şifreleme için 25 tur kullanan ve çeşitli uygulama alanlarına sahip olan bir başka hafif siklet kriptografi algoritmasıdır (Zhang vd., 2015).

TWINE, PRESENT benzeri bir yapıya sahip ancak uygulama aşamasında karşılaşılan birçok sorunu aşabilen bir hafif siklet kriptografik şifreleme algoritmasıdır. Hız kıyaslaması olarak AES'in uygulanabildiği IoT sistemlerinde daha yavaş kalmasına rağmen AES'in uygulanamadığı sistemlerde PRESNT'tan yaklaşık olarak %250 daha hızlı işleme süresi sunmaktadır (Thakor vd., 2020).

Tiny Encryption Algorithm (TEA), çok küçük, işlem gücü olarak zayıf ve düşük donanıma sahip uygulamalar için uygun bir algoritma türüdür (Appel vd, 2016). Sağlamış olduğu hızın

yanında kullanmış olduğu basit anahtar zamanlaması ve şifre çözme işlemi için kullandığı üç eş anahtar sebebiyle kaba kuvvet saldırılarına ve tahminleme saldırılarına karşı algoritmayı savunmasız kılmaktadır (Williams, 2008; Sekar vd., 2011). TEA'nın gelişmiş hali olan eXtended TEA (XTEA), daha karmaşık bir anahtar zamanlaması sunmaktadır (Kaps, 2008). Ancak bu karmaşıklık ilgili-anahtar saldırısı için yeterli koruma sağlayamamıştır (Lu, 2009). XXTEA isimli XTEA'nın geliştirilmiş Wheeler vd. (1998) tarafından ortaya koyulmuştur.

Camelia, AES benzeri bir yapıya sahip olmakla beraber şifreleme işlemini 18 tur veya 24 tur iki farklı seçenek sunarak yapmaktadır (Aoki vd., 2000). AES algoritmasına benzer bir güvenlik seviyesi sunan Camelia, donanımsal uygulamalarına yakın seviyede yazılımsal uygulamalar sağlamasıyla bilinmektedir (Satoh ve Morioka, 2003). Zhao vd. (2009), yapmış oldukları çalışmada Camelia algoritmasının zamanlama saldırısına karşı zaafiyeti ortaya koymuşlardır.

Scalable Encryption Algorithm (SEA), düşük hafıza gereksinimleri, küçük kod boyutu ve anahtar erişimine karşı koruma sunan; 96-bitlik anahtar uzunluğuna, 96-bitlik ve 8-bitlik blok seçeneklerine sahip kısıtlı kaynaklarda azami korumayı hedefleyen bir blok şifreleme algoritmasıdır (Standeart vd., 2006; Mace vd., 2007).

CLEFIA, 2007 yılında NIST tarafından standart olarak kabul edilen 128-bitlik blok, 128-bitlik, 192-bitlik ve 256-bitlik anahtar uzunluğunun yanında 18, 22 ve 26 turda şifreleme seçenekleri sunmaktadır (Akishta ve Hiwatari, 2011). CLEFIA, çeşitli saldırılara karşı başarılı performans sergilemesine rağmen talep ettiği yüksek hafıza kullanımı sebebiyle çok küçük IoT sistemlerinde tam olarak kullanılamamaktadır (Bansod vd., 2014).

Saldırılara karşı iyi bir performans ortaya koyan HISEC algoritması, 80-bitlik bir anahtar ve 64-bitlik bir blok vasıtasıyla 15 turda şifreleme imkanı sunmakta ve karakteristiksel olarak PRESENT algoritmasına benzemektedir (Thakor vd., 2020; AlDabbagh vd., 2014).

Lightweight Block Encryption (LEA), 32-bitlik işlemciler için tasarlanan bir hafif siklet kriptografik şifreleme algoritmasıdır (Hong vd., 2013). 128-bitlik blok, 128-bit, 192-bit ve 256-bit anahtar uzunluğuna sahip 24,28 veya 32 turluk şifreleme sçeneği sunan bu algoritmanın 128-bitlik anahtar kullanımında güç analizine karşı savunmasız olduğu Kim ve Yoon (2014) tarafından ortaya koyulmuştur.

Ring, hem donanımsal olarak hem de yazılımsal olarak uygulanabilen bu algoritma; 80-bitlik anahtar, 64-bitlik blok uzunluğu ve 24 turluk şifreleme sunmaktadır. Yazılımsal performansı PRESENT algoritmasından 3 kat daha verimli çalışmaktadır (Das, 2014).

Hummingbird, blok şifreleme ve akış şifreleme yöntemlerini bir arada kullanan, 256-bit key uzunluğuna sahip, 16-bitlik blok ve 20 tur şifreleme sunan bir hafif siklet kriptografik şifreleme algoritmasıdır (Engels vd., 2010). Hummingbird-2 ise düşük sistem özelliklerine sahip mikrokontroller için oluşturulmuş, 128-bit anahtar uzunluğuna ve 64-bitlik başlatma vektörlerine sahip hem yazılımsal olarak hem de donanımsal olarak kullanılabilen bir şifreleme algoritmasıdır (Engels vd. 2011).

IoT sistemlerinin kullanım alanlarının farklılıkları, kullanılan aygıtların, sensörlerin ve diğer ekipmanların da farklılığını doğurmaktadır. Algoritmaların etkili veya etkili değil olarak nitelendirilememesini beraberinde getiren bu durum sonucunda her geçen gün farklı senaryolar için farklı algoritmalar ortaya çıkmaktadır. Sürekli olan gelişen ve genişleyen bu alanda uygun algoritma seçimi için de çeşitli çalışmalar yapılmaktadır. Yapılan çalışmalarda algoritmalar kaynak kullanımı, saldırılara karşı sunulan güvenlik, uygulanabilirlik gibi çeşitli alanlardan ele alınarak karşılaştırılmaktadırlar (Naru vd., 2017; Mustafa vd., 2018; Damghani vd., 2019; Thakor vd., 2020). Belirli bir doğrunun olmayışı, uygulama alanlarını çeşitliliği ve ihtiyaca göre değişkenlik gösteren öncelikler sebebiyle bu alanda yapılacak olan her çalışma literature doğrudan katkı sağlayacaktır.

5. GRAIN DİZİ ŞİFRELEME ALGORİTMALARI

Grain dizi şifreleme algoritmaları, 2004 yılında avrupa araştırma kuruluşlarından oluşan bir topluluk tarafından başlatılan ve Avrupa Birliği tarafından finanse edilmekte olan verimliliğe ve kompaktlığa odaklanarak yaygın olarak kullanılacak yeni dizi şifreleme algoritmalarının tasarlanması amacıyla başlatılan eSTREAM projesi kapsamında geliştirilmiştir (Robshaw ve Billet, 2008). Proje yazılım odaklı şifreleme algoritmaları (profil 1) ve donanım odaklı şifreleme algoritmaları (profil 2) olmak üzere oluşturulan algoritmaları iki başlık altında değerlendirmiştir.

- Profil 1: Yazılımda uygulama için optimize edilmeli ve uygun bir dizi şifreleme modunda AES'den önemli ölçüde daha iyi performans göstermelidir. Ayrıca, en az 128 bit güvenlik düzeyine sahip olmalı ve 64 bit ve 128 bit başlatma vektörü (IV)'leri desteklemelidirler. Bu profilin ana odağı, tek bir başlatma işleminden sonra büyük miktarda veri için yüksek ham şifreleme hızına ulaşmaktır (Babbage vd., 2008).
- Profil 2: Çok kısıtlı donanım ortamlarında uygulama için optimize edilmelidir. Somut olarak, sınırlı bir ortamda en az bir önemli konuda AES'den önemli ölçüde daha iyi performans göstermelidirler. Bunun yanı sıra, 80-bit güvenlik seviyesine sahip olmalı ve 32-bit ve 64-bit IV'leri desteklemelidirler (Babbage vd., 2008).

Profil 2 olarak kategorilendirilen Grain algoritmaları, kapı sayısının, güç tüketiminin ve belleğin sınırlı olduğu kısıtlı donanım ortamlarını hedefleyen bir dizi şifreleme algoritmaları ailesidir. İlk hali 2005 yılında ortaya çıkmış olsa da barındırmış olduğu güvenlik zaafiyetleri sebebiyle genel kullanım için uygun bulunmamıştır (Hell vd., 2007). Bu nedenle ilk hali Grain v0 olarak isimlendirilmiştir. Tüm Grain algoritmalarının en önemli özelliklerinden biri, algoritmaların hızının daha iyi donanım şartları altında arttırılabilmesidir. Yani kullanıcı, mevcut donanım miktarına bağlı olarak şifreleme hızına karar verebilmektedir. Tüm sürümler;

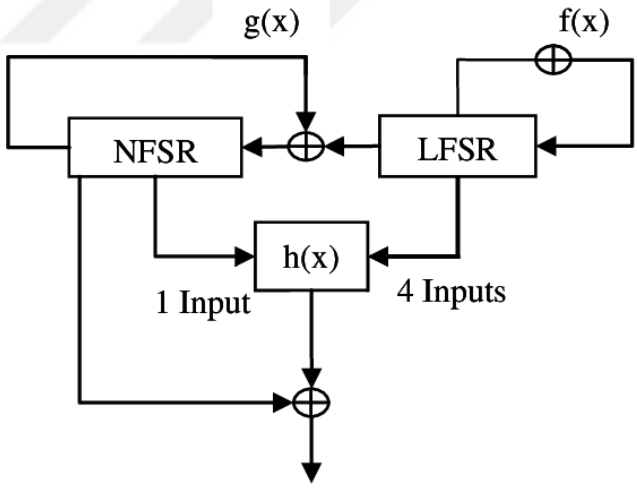
- Doğrusal geri besleme kaydırma kaydı (Linear Feedback Shift Register – LFSR),
- Doğrusal olmayan geri besleme kaydırma kaydı (Nonlinear Feedback Shift Register – NLFSR),
- Boolean filtreleme fonksiyonu

olmak üzere üç temel yapı taşından oluşmaktadır.

eSTREAM projesi kapsamında projeye katılan ve 34 aday arasından nihai 7 finalist arasında yerini alan Grain v0, birkaç bağımsız araştırmacının çıktı fonksiyonun seçiminde keşfettikleri zayıflık sonrasında NLFSR fonksiyonun güncellenerek Grain v1 sürümü olarak revize edildi ve Grain-128 (Hell vd., 2006) varyantı ile birlikte yeniden ikinci değerlendirme aşamasına sunuldu (Hell vd., 2007). Yapılan son değerlendirmeler sonucunda eSTREAM portföyünün yedi şifreleme yönteminden biri oldu (Robshaw ve Billet, 2008). 2011 yılında Grain-128'e temellendirilen yeni bir sürüm olan Grain-128a, değişken etiket boyutlarına sahip isteğe bağlı ileti kimlik doğrulamasını destekler ve ayrıca daha farklı bir doğrusal olmayan güncelleme fonksiyonu kullanarak, o dönemde Grain-128'in sahip olduğu tüm zaafiyetlerine karşı güçlendirilerek yayınlandı (Agren vd., 2011). Günümüzde 128 bitlik koruma için Grain-128 algoritması, 80 bit uzunluğunda güvenlik için Grain v1 algoritması bu ailenin en güçlü algoritmalarıdır.

5.1 Grain v1

Grain v1, 80 bitlik gizli anahtar boyutuna ve 64 bitlik IV'ye sahip bir algoritmadır. Şekil 5.1'de Grain şifrelemenin temeli görülmektedir.



Şekil 5.1 Grain ailesi genel şifreleme blok diyagramı (Bokhari vd., 2014).

LFSR'nin içeriği s_i, s_{i+1}, …, s_{i+79} olarak ve NLFSR'nin içeriği ise b_i, b_{i+1}, …, b_{i+79} olarak gösterilmiştir. Kaydırma kayıtlarını güncellemek için farklı dokunma noktaları kullanılır. Bunlar LFSR'nin nasıl güncelleneceğini belirlemek için sonlu alan aritmetik içinde polinom mod 2 olarak veya tek dokunma noktalarını gösteren bir güncelleme fonksiyonu olarak ifade edilebilir (Saluja K. K., 1987). Bu fonksiyon

$$f(x) = 1 + x^{18} + x^{29} + x^{42} + x^{57} + x^{67} + x^{80} \quad (5.1)$$

olacak şekilde tanımlanır ve bu denklem (Denklem 5.1) dokunma noktalarına karşılık gelir. Dokunma noktalarına karşı gelen denklem 5.1 kullanılarak denklem 5.2'deki güncelleme fonksiyonu oluşturulur.

$$s_{i+80} = s_{i+62} + s_{i+51} + s_{i+38} + s_{i+23} + s_{i+13} + s_i \quad (5.2)$$

Denklem 5.2'deki dokunma noktaları kullanılarak NLFSR'nin geri besleme fonksiyonu g(x) denklem 5.3'te görüldüğü gibi oluşturulur.

$$\begin{aligned} g(x) = 1 &+ x^{18} + x^{20} + x^{28} + x^{35} + x^{43} + x^{47} + x^{52} + x^{59} + x^{66} + x^{71} \\ &+ x^{80} + x^{17}x^{20} + x^{43}x^{47} + x^{65}x^{71} + x^{20}x^{28}x^{35} + x^{47}x^{52}x^{59} \\ &+ x^{17}x^{35}x^{52}x^{71} + x^{20}x^{28}x^{43}x^{47} + x^{17}x^{20}x^{59}x^{65} \\ &+ x^{17}x^{20}x^{28}x^{35}x^{43} + x^{47}x^{52}x^{59}x^{65}x^{71} + x^{28}x^{35}x^{43}x^{52}x^{59} \end{aligned} \quad (5.3)$$

NLFSR, korelasyon saldırılarının ve bilgi sızıntısının önlenmesi için iki dirençli (iki kayıtlı) olarak seçilmiştir. Bu iki kayıt, şifrenin geçerli durumunu tanımlar. NLFSR'nin girişi, NLFSR'nin dengelenemesini sağlamak için LFSR'nin çıkışıyla maskelenir (Hu ve Xiao, 2003). Denklem 5.4'te görülen dengeli boolean (filtreleme) fonksiyonu h(x) LFSR'den 4 biti, NLFSR'den bir biti girdi olarak alır.

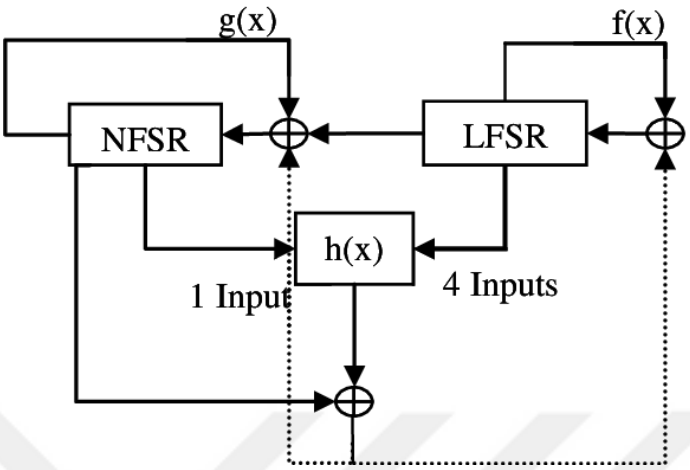
$$\begin{aligned} h(x) = x_1 + x_4 + x_0x_3 + x_2x_3 + x_3x_4 + x_0x_1x_2 + x_0x_2x_3 + x_0x_2x_4 \\ + x_1x_2x_4 + x_2x_3x_4 \end{aligned} \quad (5.4)$$

Çıkış biti A={1,2,4,10,31,43,56} için denklem 5.5'te görüldüğü şekilde oluşturulur.

$$z_i = \sum_{k \in A} b_{i+k} + h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63}) \quad (5.5)$$

Grain v1 alogritması için anahtar başlatma işlemi iki aşamadan oluştur. Bu aşamalardan birincisi, ikinci aşamada bir anahtar dizisi oluşturmak için kullanılmadan önce şifreleme işlemini başlatır. Bitleri k_i olarak gösterilen 80 bit uzunluğundaki k anahtarını alır ve NLFSR'yi bu anahtar bitleri ile oluşturur. Bundan sonra LFSR'nin ilk 64 biti IV_i ile ifade edilen 80 bit uzunluğundaki başlatma vektörüne aktarır. Böylece ilk 64 bit için IV_i ve s_i değerleri eşitlenmiş olur. LFSR için kalan 16 bit ise 1 değerini taşıyacak şekilde güncellenir. Bu işlemin de tamamlanmasıyla birlikte, algoritma çıkış bitleri üretmeden 160 kez

zamanlanmış olur. Bu anahtar başlatma işlemlerinin nedeni, anahtar akışı oluşturulmadan önce kaydırma kayıtlarının içeriğini karıştırmaktır. Yapılan işlemlerin blok diyagramı Şekil 5.2'de gösterilmiştir.



Şekil 5.2 Grain ailesi anahtar başlatma işlemi (Bokhari vd., 2014).

5.2 Grain 128a

Anahtar uzunluğu K olan herhangi bir dizi şifreleme algoritması için zaman/bellek/veri değiş tokuşu saldırısı $O(^{K/2})$ zaman karmaşıklığında gerçekleştirilecektir (Biryukov ve Shamir, 2000). Bu durum göz önüne alındığında 80 bit uzunluğundaki bir şifrenin olduğu algoritma için bu saldırının gerçekleşimi $O(2^{40})$ zaman karmaşıklığında olacaktır ve bu karmaşıklık günümüz teknolojisinde ulaşılması hiç de güç olmayan bir zaman karmaşıklığı anlamına gelmektedir. 128 bitlik bir anahtar uzunluğu için zaman karmaşıklığının $O(2^{64})$ olacağı düşünüldüğünde Grain ailesi için 128 bitlik anahtar uzunluğuna sahip varyantalarının oluşturulması bir zorunluluk olmuştur. 128 bitlik Grain şifrelemesinin Grain v1'den farkı 128 bitlik LFSR ve 128 bitlik NLFSR kullanılması ve h(x) filtreleme fonksiyonun değiştirilerek buna uygun hale getirilmesidir. Denklem 5.6'da da görüleceği şekilde LFSR'nin geri besleme polinomu f(x) güncellenmiş ve polinom seviyesi 128'e çıkarılmıştır. Filtreleme fonksiyonu h(x) ise yine 128 bitlik anahtar uzunluğunun ihtiyaçlarını karşılamak için Denklem 5.7'de görüldüğü şekline güncellenmiştir.

$$f(x) = 1 + x^{32} + x^{47} + x^{58} + x^{90} + x^{121} + x^{128} \tag{5.6}$$

$$h(x) = x_0x_1 + x_2x_3 + x_4x_5 + x_6x_7 + x_0x_4x_8 \tag{5.7}$$

Geri kalan tasarım ilkeleri aynı kalmıştır. Grain 128a versiyonun Grain 128'den farkı ise İleti Kimlik Doğrulama Kodu (Message Authentication Code – MAC) barındırmasıdır (Agren vd., 2011). Grain 128a 128 bitlik anahtar uzunluğu ve bir doğrulama mekanizması barındırması sebebiyle Grain ailesinin en güçlü üyesidir (Agren vd., 2011). LFSR ve filtreleme fonksiyonu Grain 128 ile aynı olmasına karşın NLFSR'nin geri besleme polinomu g(x) güçlendirilmiştir ve Denklem 5.8'deki hali almıştır.

$$g(x) = 1 + x^{32} + x^{37} + x^{72} + x^{102} + x^{128} + x^{44}x^{60} + x^{61}x^{125} + x^{63}x^{67} + x^{69}x^{101} + x^{80}x^{88} + x^{110}x^{111} + x^{115}x^{117} + x^{46}x^{50}x^{58} + x^{103}x^{104}x^{106} + x^{33}x^{35}x^{36}x^{40} \quad (5.8)$$

NSFR'nin güncelleme fonksiyonun değiştirilmesine ek olarak başlatma aşaması da 128 bitlik anahtara geçiş sebebiyle değiştirilmiştir. Grain 128a, NLFSR'yi 128 bit anahtarla, LFSR'yi ise 96 bitlik IV ile yükler. LFSR'nin kalan 32 bitlik alanın 31 biti ise 1'ler ile doldurulur. Benzer IV'lerin yüklenmesi sebebiyle oluşabilecek zaafiyetlerin önlenmesi amacıyla sonuncu bit 0 değerine sahip olacak şekilde güncellenir. Şifreleyici 256 kere tikledikten sonra Grain v1'dek olduğu gibi çıktı yeniden kaydırma kayıtlarına geri beslenir.

Kimlik doğrulama modunun eklenmesi ile birlikte çıkış fonksiyonunda da değişiklik yapılması gerektiğinden dolayı yazarlar Denklem 5.9'da görüleceği şekilde A={2,15,36,45,64,73,89} için bir ön-çıkış fonksiyonu tanımlamışlardır.

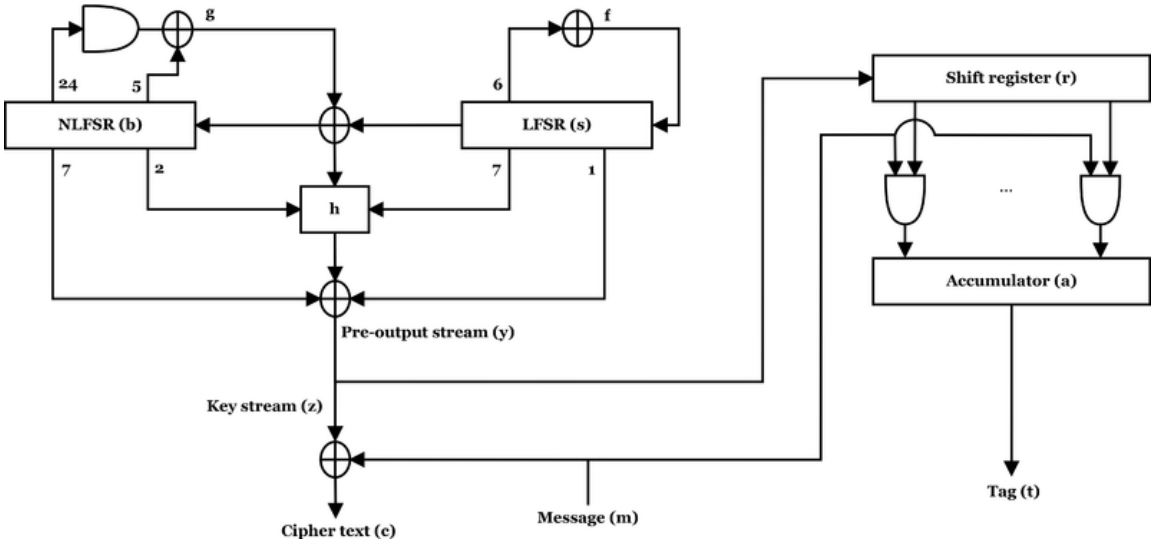
$$y_i = \sum_{k \in A} b_{i+k} + h(x) + \mathrm{s_{i+93}} \quad (5.9)$$

Denklem 5.9'da görülen çıktı kimlik doğrulama modunun aktif olup olmamasına göre değişiklik göstermektedir. Kimlik doğrulama modu aktif değil ise çıktı 5.10'daki denkleme, eğer aktif ise 5.11'deki denkleme eşit olacaktır.

$$z_i = y_i \quad (5.10)$$

$$z_i = y_{64+2i} \quad (5.11)$$

Kimlik doğrulama modunun açık olduğu durumda görülen denklem, ihtiyaç anında çıktıdaki 64 bit ihmal edildikten sonraki ikinci bit çıktı biti olarak kabul edileceği anlamına gelmektedir. Şekil 5.3'te kimlik doğrulama özelliği aktif olan Grain 128a algoritmasının blok diyagramı görülmektedir.



Şekil 5.3 : Grain 128a kimlik doğrulamalı blok diyagramı (Vikipedi, 2021).

5.3 Güvenlik

Grain algoritmaları düşük donanımsal ihtiyaçları ve güçlü koruma özellikleri sebebiyle kullanıcılara için cazip bir seçenek sunmaktadır. Bu nedenle ortaya çıktıkları ilk günlerden beri sık sık kriptanaliz çalışmalarına maruz kalmışlardır (Berbain vd., 2006; Zhang ve Wang, 2009; Maximov, 2006; Lehman ve Meier, 2012; Bokhari vd., 2014).

Küçük (Küçük, 2006) tarafından yapılan bir çalışmada kayma resenkronizasyon saldırısı kullanılarak Grain v1'in başlatma aşamasında yapılan bir saldırı ile ilişkili anahtarların ve IV'lerin bulunması sağlanmıştır. Daha sonrasında De Canniére ve diğ. (2008) yapmış oldukları çalışmada Grain algoritmalarındaki kayma özelliğinin başlatma kısmındaki varlığının anahtar arama maliyetini yarı yarıya azaltacak şekilde kullanılabileceğini göstererek bu çalışmayı daha da genişleterek yayınlamışlardır.

Küp saldırısı, ayırt edici saldırıya çok benzeyen, matematiksel bir bilinen düz metin saldırısıdır. Anahtarı tamamen ortaya çıkarabilen bu saldırı Dinur ve Shamir (2009) tarafından yayınlanmıştır. Bu saldırı daha sonra Aumasson ve diğ. (2009) tarafından Grain-128 ile korunan bir FPGA'ya başarılı bir şekilde uygulanmıştır. Daha sonrasında Dinur ve diğ. (2011) tarafından dinamik küp saldırısı isminde daha gelişmiş bir küp saldırısı 2011 yılında Grain-128 algoritmasına uygulanmıştır. Bu saldırı, küp testlerinden elde edilen farklılıkları kullanarak gizli anahtarı ortaya çıkarmayı hedeflemektedir.

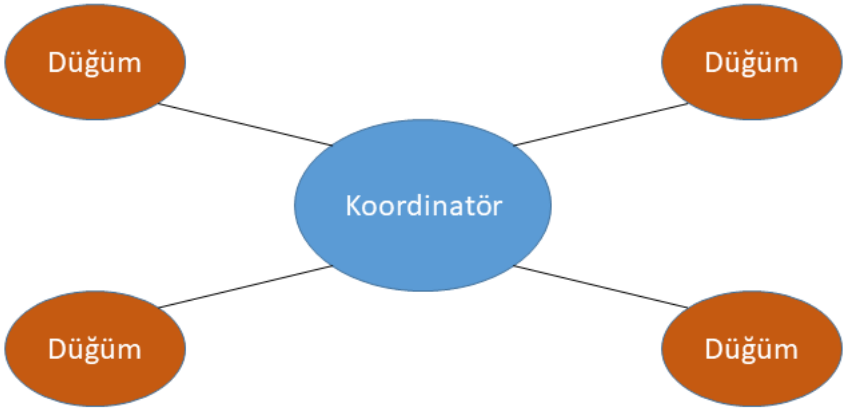
Bir yan kanal saldırısı olan hata saldırısı olan kusur saldırısı, şifrenin veya yapılan işlemlerin çeşitli durumlarda arızalara neden olduğu ve daha sonra sistemi kırmak için arızasız bir

şifreli metne kıyasla hatalı şifreli metnin analiz edildiği bir saldırı türüdür. Grain-128'e ilk kez 2009 yılında Berzati ve diğ. (2009) tarafından uygulanmıştır. Daha sonrasında Banik ve diğ. (2012) tarafından geliştirilerek Grain 128a üzerinde uygulanmıştır. Karmakar ve Chowdhury (2014) yapmış oldukları çalışma ise diğer çalışmaların aksine hem LFSR'nin hem de LFSR ile birlikte NLFSR'nin de hedef alınabileceğini ve bu nedenle her iki kaydın da korunması gerektiğini ortaya koymuştur.

Bu şifreleme ailesin üzerinde yapılan araştırmalar göz önüne alındığında haricinde dinamik küp saldırısı ile kusur saldırıları haricinde bilinen ciddi zaafiyeti olmadığını göstermektedir. Bunun yanında Lehmann ve Meier'in (2012) önermiş oldukları yöntem ile birlikte Grain 128a algoritmasının bu iki saldırı türüne karşı daha dirayetli olduğu ve dolayısıyla Grain ailesinin en güçlü algoritması olduğu bilinmektedir.

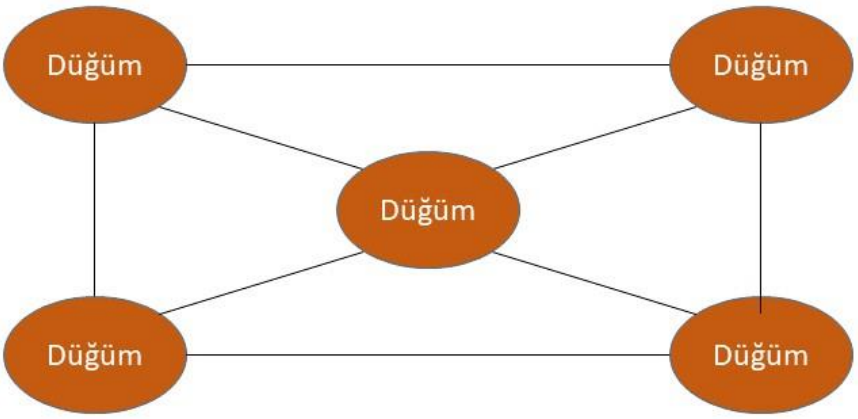
6. UYGULAMA

IoT sistemlerinde güvenli haberleşmenin sağlanması için çeşitli yöntemler kullanılmaktadır. Bu yöntemlerden en temel olanlarından biri iletişimin şifreli olarak sağlanmasıdır. Şifreli olarak haberleşmenin sağlanması için kullanılan yöntemlerden biri olan kriptografik yöntemlerin nasıl olduğunu ve literatürde nasıl bir yere sahip olduğu Bölüm 4'te anlatılmıştır. IoT sistemlerinde haberleşme güvenliğinin en zaruri olduğu noktalardan biri sisteme yeni bir düğümün katılımı sırasında ortaya çıkan zaafiyetlerin giderilmesidir. Sisteme katılacak olan nesnelerin güvenilir olup olmadıkları veya hangi yetkilerle sistem içinde hangi rolü alacakları önceden tahmin edilebilir olmalıdır. IoT sistemlerinde düğümler kendi aralarında doğrudan veya bir koordinatör vasıtasıyla haberleşme işlemini gerçekleştirebilirler. Şekil 6.1'de koordinatör üzerinden haberleşen bir IoT sistemi modeli görülmektedir. Bu modelde düğümler arası herhangi bir bağlantı bulunmamaktadır, tüm iletişim koordinatör üzerinden sağlanmaktadır.



Şekil 6.1 Koordinatörlü haberleşme modeli.

Şekil 6.2'de ise iletişimin düğümler arasında doğrudan gerçekleştirildiği ve bir düğümün birden fazla iletişim kanalını ihtiyacına göre kullanabildiği model görülmektedir.



Şekil 6.2 Koordinatörsüz haberleşme modeli.

Seçilen haberleşme modellerinden hangisi uygulanacak olursa olsun düğümler arası haberleşmenin güvenli olması ve sisteme katılacak olan yeni düğümün de güvenilir olduğundan emin olunması gerekmektedir. Haberleşme güvenliğinin sağlanması için bilinen ve güvenilirliği test edilmiş kriptografik çözümlere başvurulabileceği gibi sisteme katılacak yeni düğümün güvenilir olup olmadığının kontrolü için de yine konvansiyonel yöntemler tercih edilmelidir. Ancak hem şifreleme yönteminin seçiminde hem de düğüm katılımı esnasında yapılacak onaylama işleminin tercih edilmesinde IoT cihazlarının sahip oldukları düşük kaynak kapasiteleri ve ihtiyaç duydukları hızlı işlem gereksinimi göz önünde bulundurulmalıdır.

6.1 Sisteme Katılım

IoT sistemlerinde, sisteme katılma işleminin sağlanabilmesi için katılacak olan yeni cihazın güvenilirliği bilinmelidir. Bu ihtiyacın karşılanması için çeşitli matematiksel modeller kullanılarak her düğüm için öznel bir bağlantı anahtarı katılım esnasında oluşturulabilir veya her aygıt ortak olan ve daha önceden oluşturulmuş bir bağlantı anahtarıyla sisteme bağlanabilir. Bu işlem aynı zamanda bir el sıkışma (Keoh vd., 2014) işlemidir. Çizelge 6.1'de bu iki yöntemin karşılaştırılması görülmektedir.

Çizelge 6.1 Ortak Bağlantı Anahtarı ve Öznel Bağlantı Anahtarı yöntemlerinin kıyaslanması.

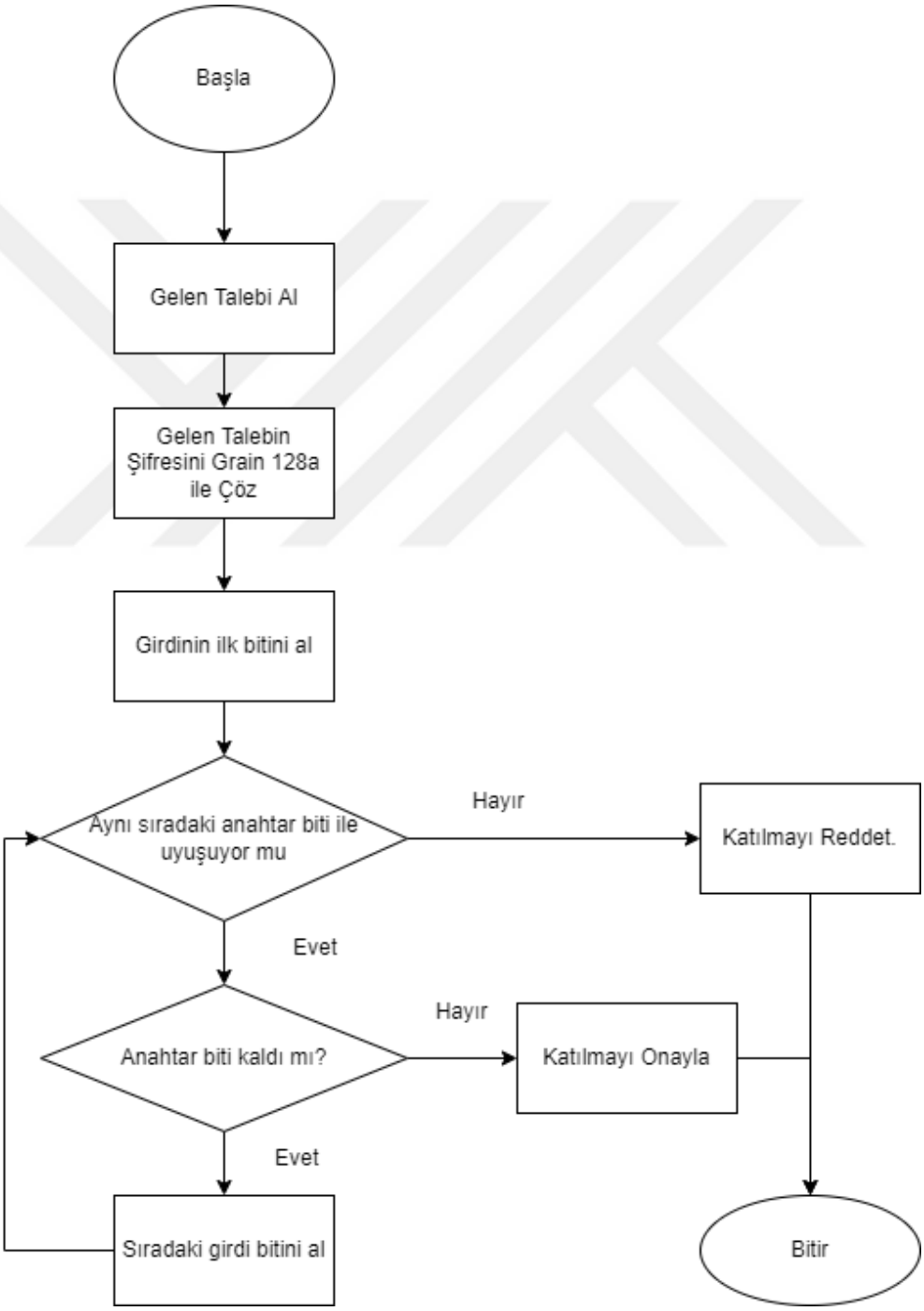
Ortak Bağlantı Anahtarı Kullanımı	Öznel Bağlantı Anahtarı Kullanımı
Kullanılan bağlantı anahtarı tüm düğümler için ortaktır.	Her düğüm için farklı bir anahtar oluşturulur, anahtarlar özneldir.
Anahtar önceden üretildiği için sisteme ek işlem yükü bindirmez.	Anahtar her düğüm için ağa katılım esnasında oluşturulduğu için işlem gücü gerektirir.
Tek anahtarın bulunması ile tüm cihazların anahtarları elde edilmiş olur.	Anahtar üretiminin arkasındaki matematiksel model çözülmedikçe anahtarlara ulaşım zordur.
Tek bir anahtar ile saldırgan birden fazla cihazı sisteme bağlayabilir.	Genellikle aynı anahtar birden fazla kez kullanılamaz.

Seçilecek olan bağlantı anahtarı kullanımı yönteminin uygulanacağı sistem özelinde sistem yeterliliklerinin ve ihtiyaçlarının göz önünde bulundurularak seçilmesi önemlidir. Örneğin; işlem hızı ihtiyacının yüksek, işlem gücü ve bellek kapasitelerinin düşük olduğu bir sistemde ortak bağlantı anahtarı kullanımı güvenlik açısından daha ciddi zaafiyetler doğurabilecek olmasına rağmen tercih edilebilir.

6.2 Önerilen Yöntem

Sisteme bağlanacak cihazları tek bir noktadan kontrol etmek, güvenli bir şekilde yeni düğümlerin ağa katılmasını sağlamak ve birbirleriyle iletişimlerini sağlamak amacıyla; zaafiyetleri bilinen hafif siklet bir dizi şifreleme algoritması olan Grain 128a'nın, yeni bir yöntem olan kontrollü rassal bekleme (KRB) yöntemi ile birlikte kullanılması önerilmektedir. Önerilen bu yöntem ile hem bağlantı güvenliğinin arttırılması hem de sisteme yeni bir düğüm eklenmesi esnasında zamanlama bilgisi sızımının düşük kaynak kullanımına sahip bu yöntemler ile engellenmesi amaçlanmaktadır.

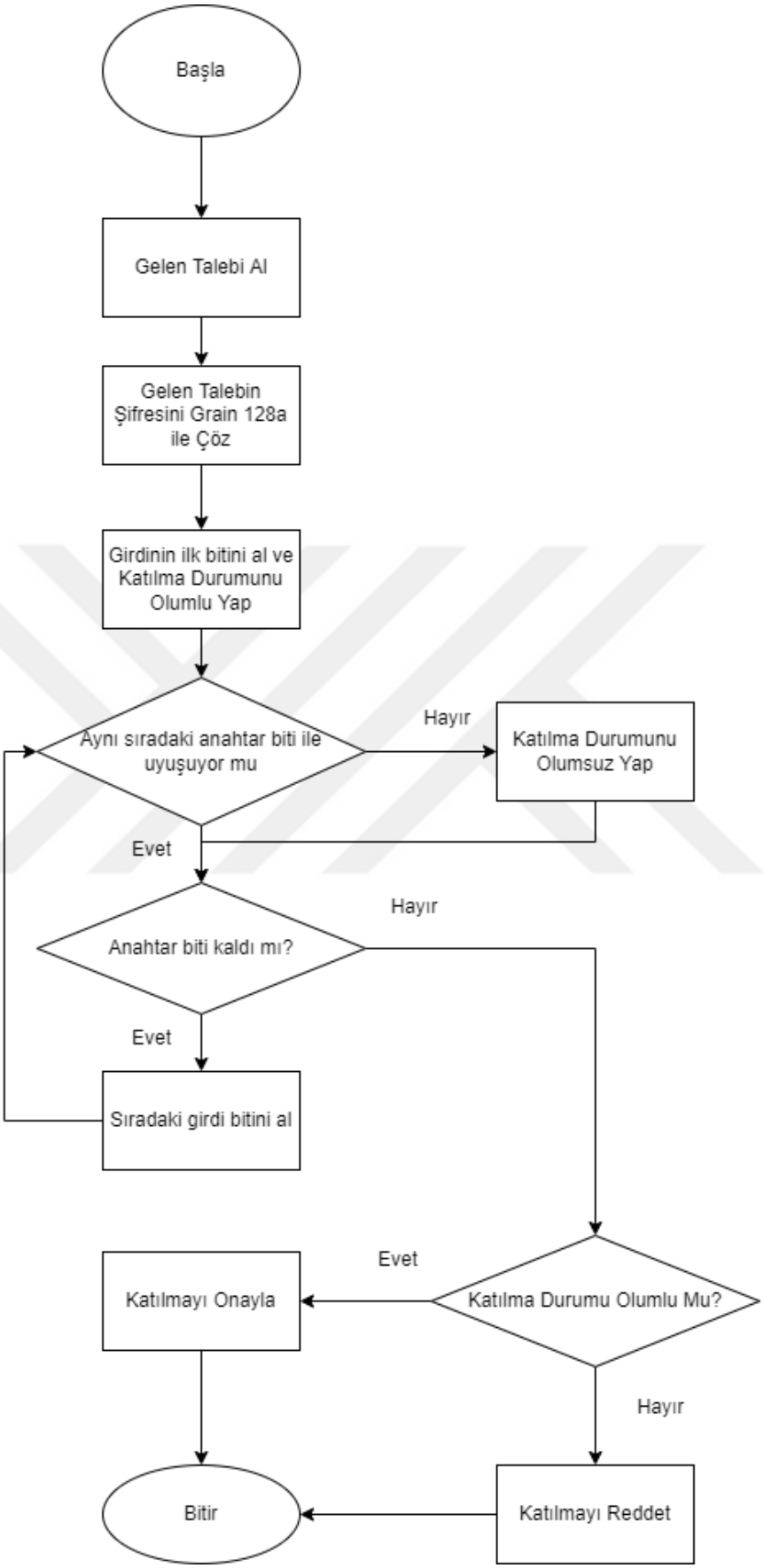
Sistemin ZAS'a karşı korunmadan önceki yalın halinin akış diyagramı Şekil 6.3'te görülmektedir. Bu yalın haldeki kullanımda gizli anahtar kıyaslama işleminin döngü içinde kaldığı süre eşleşme oranına göre değişiklik gösterdiği için sistem dışına Grain 128a algoritmasından kaynaklı olmamasına rağmen şifreleme ve şifre çözme işlemleri sonrası zamanlama bilgisi sızacaktır. Bu sızma durumunun engellenmesi için çeşitli yöntemler geliştirilmiştir (Käsper ve Schwabe, 2009; Ambrose vd., 2008; Dhem, 1998; Walter, 1999; Walter 2002; Hachez ve Quisquater, 2000).



Şekil 6.3 : Yalın halde Grain 128a kullanılan sisteme düğüm katılması işlemi.

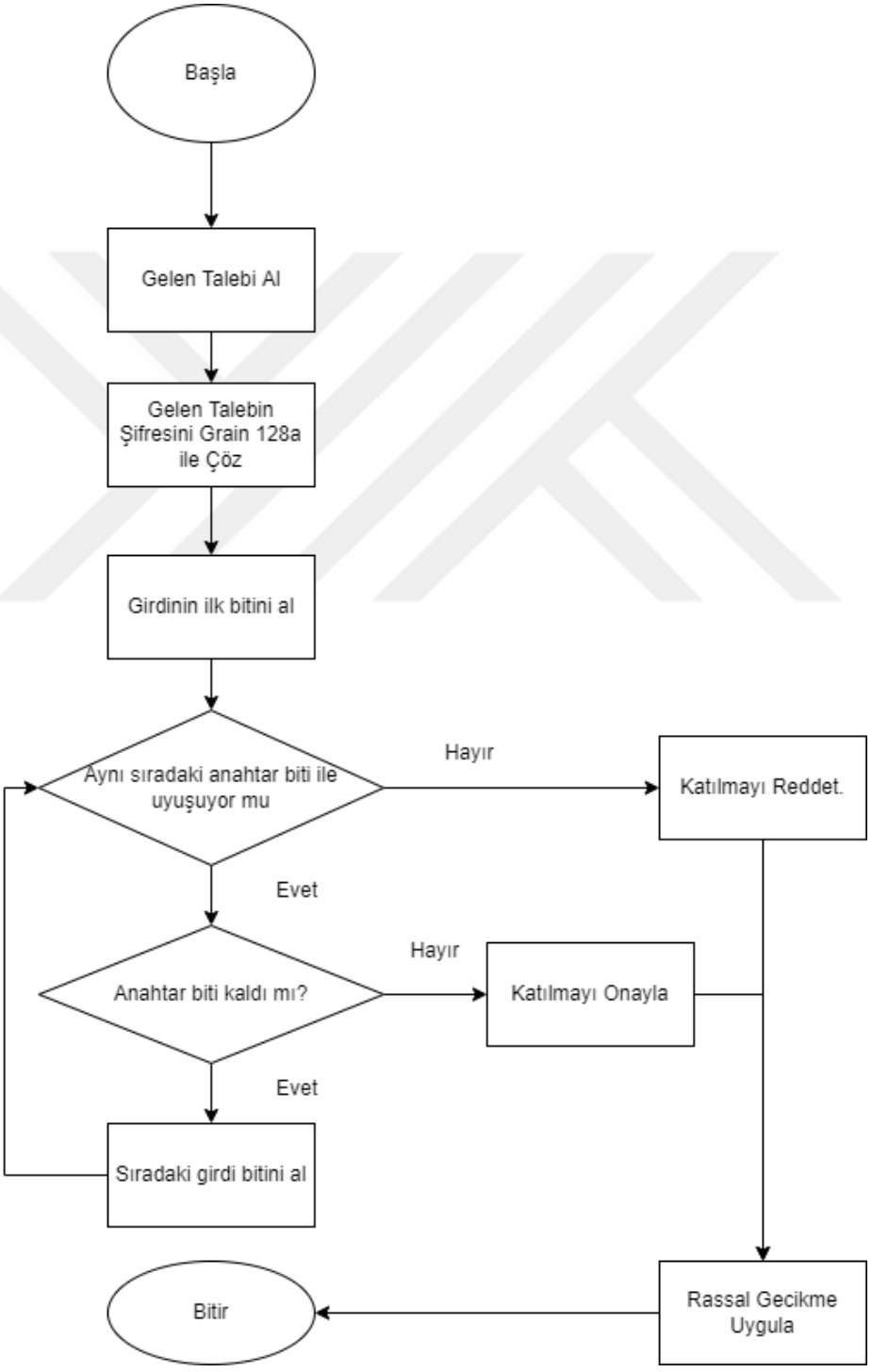
Donanımsal önlemler daha çok zamanlama ölçümü yapılmasını veya gürültüyü arttırarak ya da sızan zamanlama bilgisini sınırlandırarak yapılan saldırıyı zorlaştırmayı amaçlamaktadırlar. Örneğin; bir tümleşik devre içerisine yerleştirilen fiziksel bir rassal sayı üreteci ile gürültünün arttırılması donanımsal önlem olarak nitelendirilebilir (Kocher vd., 1999). Yazılımsal olarak alınabilecek önlemlerden biri yine donanımsal önlemlerdeki gibi işlem süresinin rastgele hale getirilmesidir (Coron, 1999). Bu korunma yönteminde, algoritma içerisine işlem sonucunu değiştirmeyecek çeşitli işlemler eklenir. Bu işlemlerin ortaya çıkaracağı gecikmenin rassal olarak oluşturulması sağlanır ve bu sayede saldırganın zamanlama bilgisini anlamlandırması zorlaştırılır. Bu yöntem tamamen rassal olması sebebiyle yapılan işlemlerin sisteme daha fazla yük bindirmesi ve genellikle hızlı tepki süresine ihtiyaç duyan IoT sistemleri için ağır kalabilmektedir.

En çok kullanılan yazılımsal korunma yöntemlerinden biri de kullanılan algoritmanın sabit zamanlı çalışacak şekilde tasarlanmasıdır (Ordu ve Yalçın, 2016). Bu yöntem ZAS'a karşı en güçlü korunma yöntemi olsa da, bir kriptografik şifreleme algoritmasının sabit zamanlı çalışmasının tek yolu her koşulda en kötü durumdaymış gibi tepki vermesiyle mümkün olmaktadır. Bu da kriptografik şifreleme algoritmalarının bile hafif-siklet olarak kullanılmasını gerektiren IoT alanı için istenmeyen bir durumdur. Bu yöntemin bahsi geçen senaryoda uygulanmış halinin temsili akış diyagramı Şekil 6.4'te görülmektedir.



Şekil 6.4 : Sabit zamanlı çıktı ve Grain 128a kullanılan sisteme düğüm katılması işlemi.

Önerilen KRB yönteminde ise öncelikle Şekil 6.3'te görülen algoritma çeşitli eşleşme oranlarındaki anahtarlar ile uygulanacak olup sistemin eşleşme oranlarına göre sızdırdığı zamanlama bilgisi analiz edilecektir. Yapılan analiz sonucunda elde edilen bilgiler kontrollü bir şekilde rassal bir bekleme yapılabilmesi için oluşturulacak olan modelde azami bekleme süresinin belirlenmesi için kullanılacaktır. Sistemin Şekil 6.5'te görüldüğü şekilde çalışması ve işlemler sonrasında fonksiyonun rassal olarak bekletilmesi sağlanacaktır.



Şekil 6.5 : KRB ve Grain 128a kullanılan sisteme düğüm katılması işlemi.

Yapılan analiz sonucunda sızan zamanlama bilgisi N, algoritma işleme zamanı K, en iyi durum Ω (%0 eşleşme oranı) ve en kötü durum O (%100 eşleşme oranı) değerleri kullanılarak bir matematiksel model oluşturulacaktır. Oluşturulan matematiksel model sistemin gelecek istekler karşısında bekletileceği sürenin rassal olarak oluşturulması aşamasında alt ve üst limitlerini belirleyecektir. Bekleme işlemi ise No-operation (nop) ile sağlanacaktır.

Şekil 6.3, Şekil 6.4 ve Şekil 6.5'te gösterilen algoritmalar çeşitli eşleşme oranına sahip anahtarlar için farklı sistem yükleri altında çalıştırılarak her aşama için ZA yapılacaktır. ZA sonuçları kıyaslanarak önerilen yeni yöntemin hem ZAS'a karşı direnci hem de literatürde bulunan kabul görmüş bir yönteme karşı kazanç sağlayıp sağlayamadığı gösterilecektir.

6.3 Önerilen Yöntemin Uygulanması ve Sonuçlar

Önerilen yöntem 1 adet Arduino Mega 2560 Rev 3, 1 adet Raspberry Pi 3 Model B ve Wireless NRF24L01 2.4 GHz Transceiver Modül - 2.4 GHz Alıcı Verici Modül kullanılarak bir sistem oluşturulmuş ve gizli anahtar eşleşme senaryosu kullanıcı tarafından anahtar girdisi alınarak simüle edilmiştir.

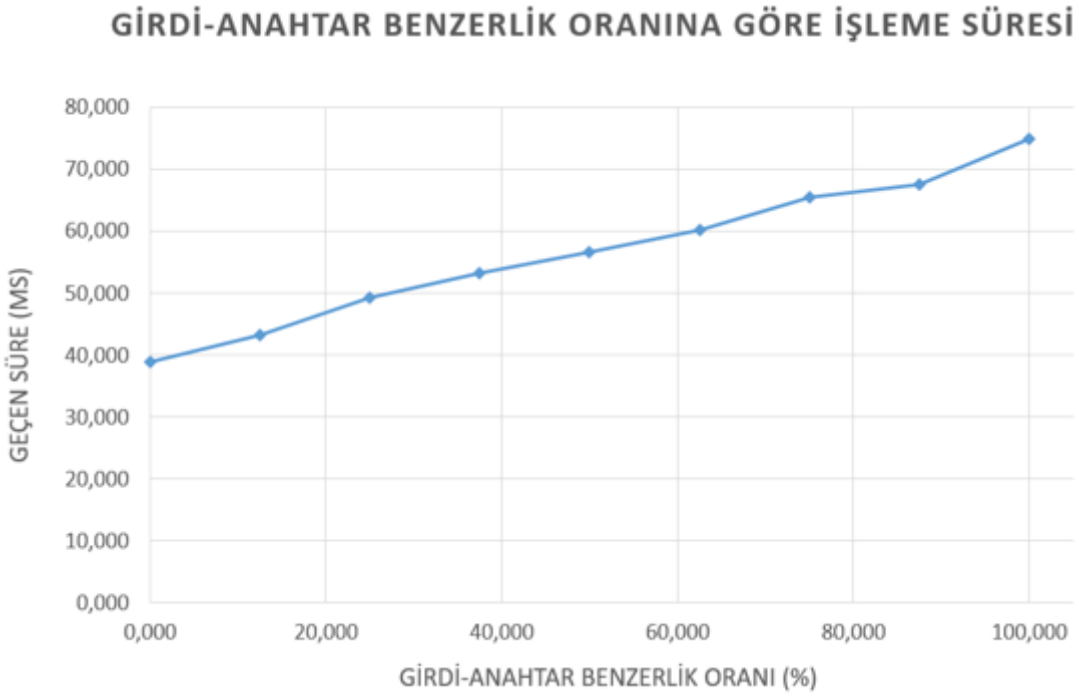
Sisteme öncellikle Şekil 6.3'teki haliyle bilinen ve çeşitli düzeyde eşleşme oranına sahip anahtarlar girdi olarak gönderilerek ZA uygulanmış ve yalın haldeki durum için en iyi ve en kötü durumdaki zamanlama bilgileri kayıt altına alınmıştır. Bu işlem her bir aşama için 30.000 kez tekrar edilerek sistem yükü değişiklikleri ve gürültü gibi etkenlerin önüne geçilmesi amacıyla bu kayıtların ortalamala değerleri alınarak her eşleşme oranı için tek değer belirlenmiştir.Sistem üzerinde gizli anahtarla farklı eşleşme oranlarına sahip girdilerle test edilerek uygulanmış olup Çizelge 6.2'deki sonuçlara ulaşılmıştır.

Çizelge 6.2 : Ω ve O durumlarının yalın algoritmadaki zamanlama bilgileri.

	Zamanlama Değeri (ms)		
Durum	En İyi	Ortalama	En Kötü
Ω	35,354	38,991	51,551
O	59,772	74,815	96,664

Herhangi bir önlem alınmadan eşleştirme işleminin yapılması sonucunda ortaya çıkan analiz sonuçları Şekil 6.6'da görülmektedir. Girdi-anahtar benzerliği arttıkça işleme süresinin de arttığı, benzerlik ile işleme süresi arasında doğru orantı olduğu görülmektedir. Bu durum,

farklı gerçekleşimlerin yalnızca sürelerine bakılarak o gerçekleşimde kullanılan girdinin anahtar ile benzerliğinin yorumlanabilmesini hatta gizli anahtarın tahmin edilebilmesini mümkün kılmaktadır.



Şekil 6.6 : Eşleşme fonksiyonunun zamanlama analizi sonuçları.

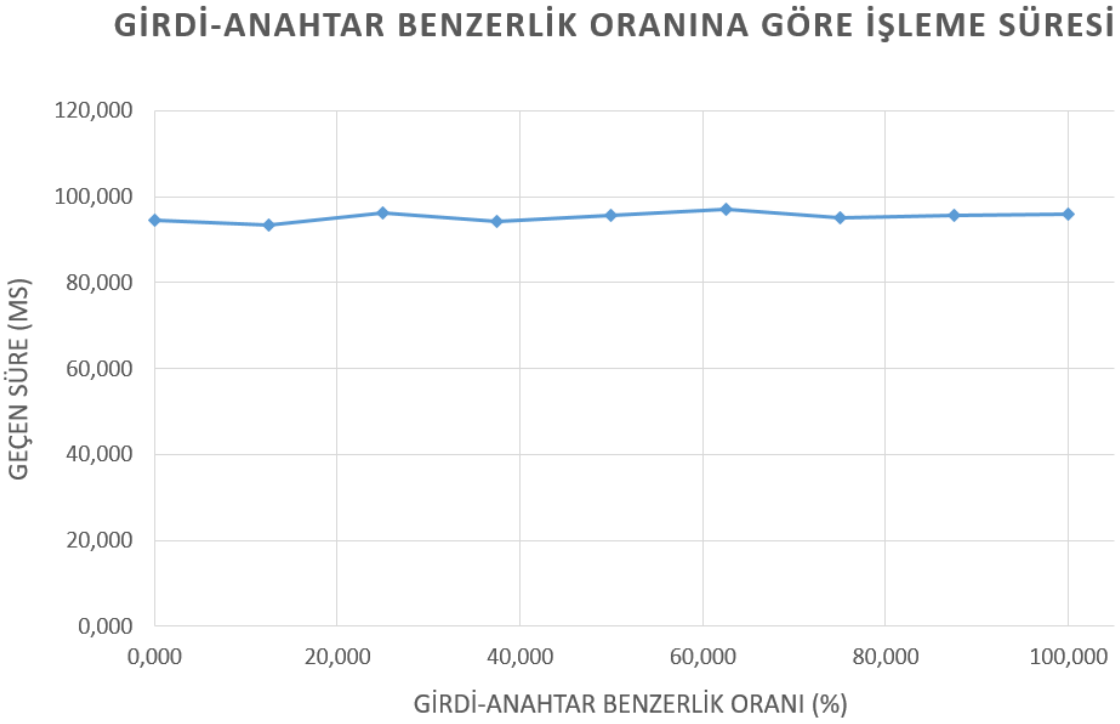
Algoritmanın yalın haline uygulanan ZA sonrasında algoritma girdi boyutundan bağımsız olarak cevap verecek şekilde düzenlenmiştir. Kaydedilen zamanlama bilgileri Çizelge 6.3'te görülmektedir.

Çizelge 6.3 : Ω ve O durumlarının sabit zamanlı algoritmadaki zamanlama bilgileri.

	Zamanlama Değeri (ms)		
Durum	En İyi	Ortalama	En Kötü
Ω	94,314	95,939	95,111
O	95,712	95,283	95,334

Şekil 6.7'de görülen sonuçlar sabit zamanlı işleme süresine sahip olacak şekilde eşleştirme işleminin düzenlenmesi sonucunda ortaya çıkan analiz sonuçlarıdır. Fonksiyon girdi boyutundan ve girdi-anahtar benzerliğinden bağımsız olarak en kötü durumda cevap vermiştir. Bu çözüm zamanlama analizini engellemeseine karşın karmaşık işlemlerin olduğu fonksiyonlarda veya anahtar boytunun çok büyük olduğu durumlarda sistem üzerindeki yükü

arttıracaktır. Bu da düşük kaynaklara sahip olan IoT uygulamalarında istenmeyen bir durumdur.



Şekil 6.7 : Sabit zamanlı eşleşme fonksiyonunun zamanlama analizi sonuçları.

Yalın halde yapılan analiz sonuçları kullanılarak Yapılan analiz sonucunda sızan zamanlama bilgisi N, algoritma işleme zamanı K, en iyi durum Ω (%0 eşleşme oranı) ve en kötü durum O (%100 eşleşme oranı) için;

$$\Omega \leq N \leq 2 \cdot O \; ve \; 0 \leq m \leq O \;\; için; \; N = K + m \tag{6.1}$$

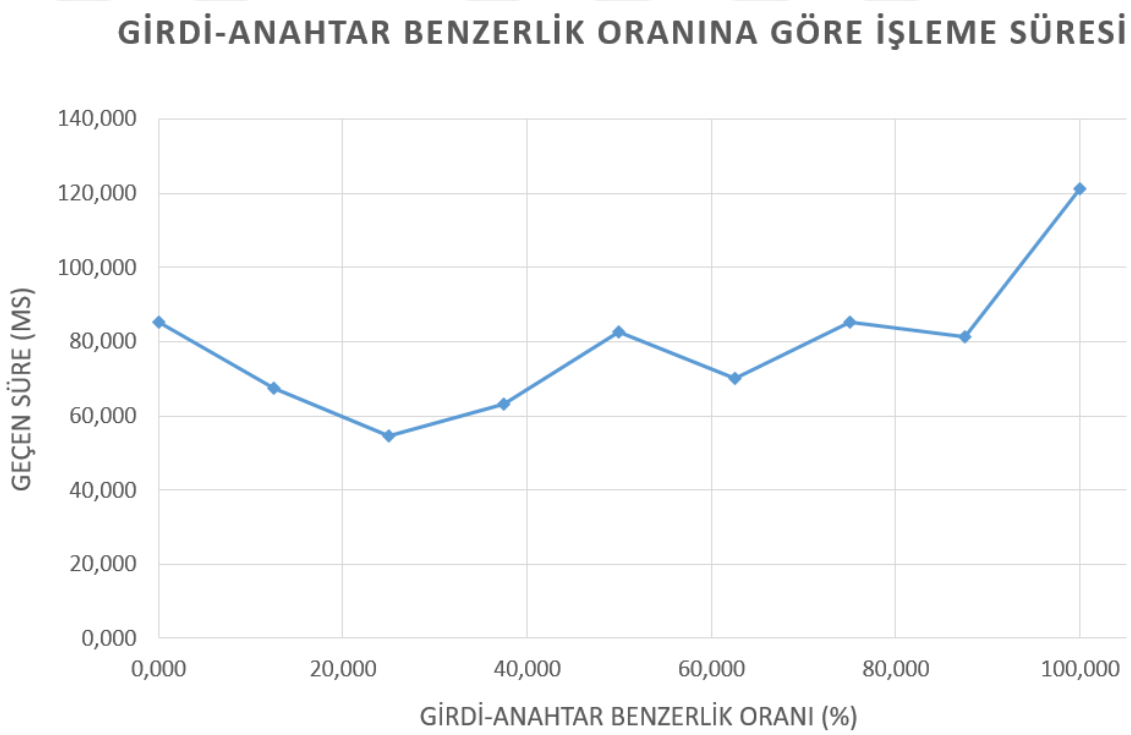
modeli oluşturulmuştur. N ve m değerlerinin, belirtilmiş aralıklarda olmaları sağlanacak şekilde, rassal sayı üreteci (Arduino için random(), Raspberry Pi için trng.h) kullanılarak m değerinin oluşturulması ve algoritmanın hiçbir şey yapmadan (nop) bekletilmesi sağlanmıştır. Yapılan bu işlem sonucunda; sisteme ekstra yük bindirmeden, tepki süresinin minimumda tutulması ve sızan zamanlama yan-kanal bilgisinin en anlamsız hale getirilmesi amaçlanmıştır.

Denklem 6.1'de oluşturulan model ve Çizelge 6.2'deki sonuçlar kullanılarak uygulanmış ve bu yeni uygulamaya ait ZA sonuçları Çizelge 6.4'te olduğu şekilde kaydedilmiştir.

Çizelge 6.4 : Ω ve O durumlarının önerilen yöntemin uygulandığı algoritmadaki zamanlama bilgileri.

	Zamanlama Değeri (ms)		
Durum	En İyi	Ortalama	En Kötü
Ω	43,931	85,220	132,310
O	78,302	121,123	140,160

Şekil 6.8'te, rastgele bekleme yapacak şekilde değiştirilmiş eşleştirme işleminin analiz sonuçları bulunmaktadır. Girdi-anahtar benzerliği ile işleme süresi arasında herhangi bir doğru ilişki Şekil 6.6'dakinin aksine grafiğe bakılarak görülememektedir.



Şekil 6.8 : Önerilen çözüme ait eşleşme fonksiyonunun zamanlama analizi sonuçları.

Son olarak girdi-anahtar eşleştirme algoritması yalın, sabit zamanlı ve önerilen yöntemli olmak üzere üç durumda da yeniden uygulanarak girdi-anahtar benzerliği oranına göre vermiş oldukları sonuçlar ve ortalama işleme süreleri Çizelge 6.5'teki gibi kayıt altına alınmıştır.

Çizelge 6.5 : Girdi-gizli anahtar benzerlik oranına göre algoritmaların zamanlama analizi sonuçları.

	Zamanlama Değeri (ms)		
Benzerlik (%)	Yalın	Sabit Zamanlı	Rassal ve Kontrollü Bekleme
0	38,991	94,821	85,220
12,5	43,301	93,939	67,330
25,0	49,366	96,199	54,440
37,5	53,287	94,200	63,010
50,0	56,599	95,298	82,695
62,5	60,110	97,011	70,012
75,0	65,467	95,046	85,327
87,5	67,626	95,142	81,194
100,0	74,815	95,889	121,123
ORTALAMA	**56,618**	**95,283**	**78,928**

Çizelge 6.5'te görüldüğü üzere önerilen rassal ve kontrollü bekleme yöntemi ortalama işleme süresi olarak yalın haldeki algoritmadan daha yavaş kalmakta ancak sabit zamanlı algoritmaya göre zamanlama bilgisini maskelemesine rağmen yaklaşık %17 daha hızlı tepki vermektedir.

6.4 Değerlendirme

Önerilen yöntem, düğümlerin sisteme katılırken onaylanmasını ve bu işlemin hem şifreli şekilde sağlanmasını hem de litetartürdeki çeşitli yöntemlerden daha hızlı olmasını sağlamaktadır. Ortak anahtar kullanımı hem daha az bellek kullanımına olanak sağlayacak hem de tek koordinatör üzerinden haberleşmenin sağlanacağı bir sistemde, sisteme katılım aşamasında diğer düğümlerde yaşanması olası gecikmeyi en aza indirecektir.

Yöntemde gerçekleştirilen bekleme işleminin ZA ile birlikte güç analizi benzeri yan kanal bilgi edinme yöntemlerinin de engellenmesi amacıyla lineer matematiksel işlemler veya literatürde bulunan benzer rassal yöntemler kullanılarak yapılması önerilebilir. Ancak bu durumun sistem üzerindeki yükü arttıracağı ve birden fazla düğümün sistem üzerinde senkronize olarak çalışacağı düşünüldüğünde istenmeyen gecikmelere yol açabileceği unutulmamalıdır.

Önerilen yöntemde kullanılan anahtar dağıtımı yönteminin ortak anahtar kullanımı yöntemi olması sebebiyle Çizelge 6.3'te bahsedilen güvenlik zaafiyetleri ortaya çıkacaktır ancak sisteme katılma işleminin yalnızca sistem üzerinde daha önceden tanımlanmış olan bir anahtar vasıtasıyla yapılması ve bu anahtarın Grain 128a algoritması ile şifrelenmiş bir şekilde taşınması sebebiyle, bir saldırı durumunda saldırganın hem Grain 128a algoritmasını çözmesi hem de kullanılan gizli anahtara erişilmesi gerekmektedir. Bu durumda da önerilen yöntem sisteme katılma sırasında çift katmanlı koruma sağlamış olacak ve anahtara izinsiz erişim ihtimalini oldukça düşürecektir.

Önerilen yöntemde oluşturulan matematiksel model uygulanacağı sistem özelinde optimize edilerek daha verimli bir şekilde oluşturulması sonucunda, çalışmada elde edilen %17'lik tepki hızı kazanımı daha da arttırılabilecek ve sistem üzerinde oluşturduğu yük hafifletilebilecektir.

7. SONUÇ VE ÖNERİLER

İnternet, insan hayatını kolaylaştırması sebebiyle yaşantımızda gün geçtikçe daha da yer edinmeye başlamaktadır. Geçmişten günümüze gelişerek günlük yaşamın içerisindeki etkisini arttırmış ve artık her yerde internet kullanımı zaruri hale gelmektedir. Gelişen teknoloji ile birlikte artık yalnızca insanlar değil çevremizde gördüğümüz nesneler de internet vasıtasıyla birbirleriyle etkileşimde bulunmaktadırlar.

İnsan-Nesne, Nesne-Nesne etkileşimlerinin internet üzerinden sağlandığı IoT teknolojisi de özellikle akıllı nesnelerin ortaya çıkması ile birlikte artık hayatın bir parçası olmaya başlamıştır. Ortamdan sensörler vasıtasıyla topladıkları bilgileri kullanarak insan hayatını kolaylaştırmayı hedefleyen bu teknolojinin güvenliği, oldukları yetkiler ve taşıdığı bilgiler göz önüne alındığında kötü niyetli kişiler tarafından önemli bir hedef olarak görülmektedir. Özellikle birden fazla akıllı nesnenin birlikte oluşturdukları ve birçok alanda kilit rol aldıkları bu IoT sistemleri, gizlilik ve güvenlik yönteminin doğru şekilde sağlanmaması durumunda bulundukları ortam için ciddi tehdit oluşturabilmektedirler.

Tez çalışmasında, öncelikle bu sistemler için "nedir?", "nerelerde kullanılır?", "güvenlik ihtiyaçları nelerdir ve neden önemlidir?" gibi sorular cevaplanmaktadır. Daha sonra literatürde bulunan çalışmalar incelenerek güvenlik ihtiyacının boyutu ve bu ihtiyaca yönelik çözümler analiz edilmektedir. Bu alanda haberleşme güvenliğinin sağlanması amacıyla yaygın olarak kullanılan kriptografik yöntemler ele alınmakta ve bir IoT sistemi oluşturulurken karşılaşılabilecek en temel sorunlardan biri olan yeni nesnelerin sisteme katılması aşamasında ortaya çıkan sisteme katılım sorunu irdelenmektedir. Ardından, bu sorunun giderilmesi için bir dizi şifreleme algoritması olan Grain 128a algoritması, Ortak Anahtar Kullanımı gibi bilinen yöntemlerle birlikte sisteme yönelik potansiyel tehditlerden biri olan Zamanlama Analizi Saldırıları'na karşı yeni bir hafif siklet yöntemin birlikte kullanılması önerilmektedir. Bu yöntem ile birlikte sisteme yeni bir düğüm ekleneceği zaman bir ağ koordinatörü vasıtasıyla katılacak düğümün güvenli bir şekilde onaylanması ve Zamanlama Analizi kullanılarak gerçekleştirilecek bir saldırı durumunda bu anahtarın ortaya çıkarılmasının engellenmesi birlikte gerçekleştirilmektedir.

Gelecek çalışmalarda sistemin bir geri besleme ile birlikte desteklenerek her katılma durumunda modelin tekrar güncellenmesi ve bu şekilde en optimum sonucun sistemdeki düğüm sayısının değişmesine bakılmaksızın sürekli olarak elde edilmesi gerçekleştirilebilir.

KAYNAKLAR

Abbass, W., Bakraouy, Z., Baina, A., & Bellafkih, M. (2019). Assessing the Internet of Things Security Risks. J. Commun., 14(10), 958-964.

Agren, M., Hell, M., Johansson, T., & Meier, W. (2011). Grain-128a: a new version of Grain-128 with optional authentication. International Journal of Wireless and Mobile Computing, 5(1), 48-59.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. IEEE communications surveys & tutorials, 17(4), 2347-2376.

Ambrose, J. A., Parameswaran, S., & Ignjatovic, A. (2008, November). MUTE-AES: A multiprocessor architecture to prevent power analysis based side channel attack of the AES algorithm. In 2008 IEEE/ACM International Conference on Computer-Aided Design (pp. 678-684). IEEE.

Anderson, R., & Kuhn, M. (1996, November). Tamper resistance-a cautionary note. In Proceedings of the second Usenix workshop on electronic commerce (Vol. 2, pp. 1-11).

Aumasson, J. P., Dinur, I., Henzen, L., Meier, W., & Shamir, A. (2009). Efficient FPGA implementations of high-dimensional cube testers on the stream cipher Grain-128. SHARCS, 9, 147.

Babbage, S., Canniere, C., Canteaut, A., Cid, C., Gilbert, H., Johansson, T., ... & Robshaw, M. (2008). The eSTREAM portfolio. eSTREAM, ECRYPT Stream Cipher Project, 1-6.

Banik, S., Maitra, S., & Sarkar, S. (2012, November). A differential fault attack on Grain-128a using MACs. In International Conference on Security, Privacy, and Applied Cryptography Engineering (pp. 111-125). Springer, Berlin, Heidelberg.

Berbain, C., Gilbert, H., & Maximov, A. (2006, March). Cryptanalysis of grain. In International Workshop on Fast Software Encryption (pp. 15-29). Springer, Berlin, Heidelberg.

Berzati, A., Canovas, C., Castagnos, G., Debraize, B., Goubin, L., Gouget, A., ... & Salgado, S. (2009, July). Fault analysis of GRAIN-128. In 2009 IEEE International Workshop on Hardware-Oriented Security and Trust (pp. 7-14). IEEE.

Birkel, H. S., & Hartmann, E. (2020). Internet of Things–the future of managing supply chain risks. Supply Chain Management: An International Journal.

Biryukov, A., & Shamir, A. (2000, December). Cryptanalytic time/memory/data tradeoffs for stream ciphers. In International Conference on the Theory and Application

of Cryptology and Information Security (pp. 1-13). Springer, Berlin, Heidelberg.

Bokhari, M. U., Alam, S., & Hasan, S. H. (2014). A Detailed Analysis of Grain family of Stream Ciphers. International Journal of Computer Network & Information Security, 6(6).

Chinanu, U. E., Oche, O. E., & Okah-Edemoh, J. O. (2018). Architectural layers of internet of things: analysis of security threats and their countermeasures. Scientific Review, 4(10), 80-89.

Coron, J. S. (1999, August). Resistance against differential power analysis for elliptic curve cryptosystems. In International workshop on cryptographic hardware and embedded systems (pp. 292-302). Springer, Berlin, Heidelberg.

De Canniere, C., Kücük, Ö., & Preneel, B. (2008, June). Analysis of Grain's initialization algorithm. In International Conference on Cryptology in Africa (pp. 276-289). Springer, Berlin, Heidelberg.

Dhem, J. F. (1998). Design of an efficient public-key cryptographic library for RISC-based smart cards (Doctoral dissertation, UCL-Université Catholique de Louvain).

Dinur, I., & Shamir, A. (2009, April). Cube attacks on tweakable black box polynomials. In Annual international conference on the theory and applications of cryptographic techniques (pp. 278-299). Springer, Berlin, Heidelberg.

Dinur, I., & Shamir, A. (2011, February). Breaking Grain-128 with dynamic cube attacks. In International Workshop on Fast Software Encryption (pp. 167-187). Springer, Berlin, Heidelberg.

Dutta, I. K., Ghosh, B., & Bayoumi, M. (2019, January). Lightweight cryptography for internet of insecure things: A survey. In 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0475-0481). IEEE.

Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., & Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. IEEE Design & Test of Computers, 24(6), 522-533.

Ferro, E., & Potorti, F. (2005). Bluetooth and Wi-Fi wireless protocols: a survey and a comparison. IEEE Wireless Communications, 12(1), 12-26.

Frustaci, M., Pace, P., Aloi, G., & Fortino, G. (2017). Evaluating critical security issues of the IoT world: Present and future challenges. IEEE Internet of things journal, 5(4), 2483-2495.

Goubin, L., & Patarin, J. (1999, August). DES and differential power analysis the "Duplication" method. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 158-172). Springer, Berlin, Heidelberg.

Hachez, G., & Quisquater, J. J. (2000, August). Montgomery exponentiation with no final subtractions: Improved results. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 293-301). Springer, Berlin, Heidelberg.

Hell, M., Johansson, T., & Meier, W. (2007). Grain: a stream cipher for constrained environments. International journal of wireless and mobile computing, 2(1), 86-93.

Hell, M., Johansson, T., Maximov, A., & Meier, W. (2006, July). A stream cipher proposal: Grain-128. In 2006 IEEE International Symposium on Information Theory (pp. 1614-1618). IEEE.

Hu, Y., & Xiao, G. (2003). Resilient functions over finite fields. IEEE Transactions on Information Theory, 49(8), 2040-2046.

Janke, M., & Laackmann, P. (2002). Power and timing analysis attacks against security controllers. Infineon Technologies AG, Technology Update, Smart Cards.

Joy Persial, G., Prabhu, M., & Shanmugalakshmi, R. (2011). Side channel attack-survey. Int J Adva Sci Res Rev, 1(4), 54-57.

Kaps, J. P. (2008, December). Chai-tea, cryptographic hardware implementations of xtea. In International Conference on Cryptology in India (pp. 363-375). Springer, Berlin, Heidelberg.

Karmakar, S., & Chowdhury, D. R. (2014). Fault Analysis of Grain Family of Stream Ciphers. IACR Cryptol. ePrint Arch., 2014, 261.

Katagi, M., & Moriai, S. (2008). Lightweight cryptography for the internet of things. Sony Corporation, 2008, 7-10.

Keoh, S. L., Kumar, S. S., & Tschofenig, H. (2014). Securing the internet of things: A standardization perspective. IEEE Internet of things Journal, 1(3), 265-275.

Kim, J. T. (2017, May). Analyses of secure authentication scheme for smart home system based on internet on things. In 2017 International Conference on Applied System Innovation (ICASI) (pp. 335-336). IEEE.

Kim, Y., & Yoon, H. (2014). First Experimental Result of Power Analysis Attacks on a FPGA Implementation of LEA. IACR Cryptol. ePrint Arch., 2014, 999.

Kocher, P. C. (1996, August). Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Annual International Cryptology Conference (pp. 104-113). Springer, Berlin, Heidelberg.

Kocher, P., Jaffe, J., & Jun, B. (1999, August). Differential power analysis. In Annual international cryptology conference (pp. 388-397). Springer, Berlin, Heidelberg.

Kumar, N. M., & Mallick, P. K. (2018). The Internet of Things: Insights into the building blocks, component interactions, and architecture layers. Procedia computer science, 132, 109-117.

Kumar, S., Sahoo, S., Mahapatra, A., Swain, A. K., & Mahapatra, K. K. (2017, December). Security enhancements to system on chip devices for IoT perception layer. In 2017 IEEE International Symposium on Nanoelectronic and Information Systems (iNIS) (pp. 151-156). IEEE.

Käsper, E., & Schwabe, P. (2009, September). Faster and timing-attack resistant AES-GCM. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 1-17). Springer, Berlin, Heidelberg.

Küçük, Ö. (2006). Slide resynchronization attack on the initialization of grain 1.0. eSTREAM, ECRYPT Stream Cipher Project, Report, 44, 2006.

Lehmann, M., & Meier, W. (2012, December). Conditional differential cryptanalysis of Grain-128a. In International Conference on Cryptology and Network Security (pp. 1-11). Springer, Berlin, Heidelberg.

Lerman, L., Bontempi, G., & Markowitch, O. (2011). Side channel attack: an approach based on machine learning. Center for Advanced Security Research Darmstadt, 29.

Leuth K. L., (2020, Haziran), Top 10 IoT applications in 2020. 29 Kasım 2021 tarihinde https://iot-analytics.com/top-10-iot-applications-in-2020/ adresinden alındı.

Maximov, A. (2006, March). Cryptanalysis of the" Grain" family of stream ciphers. In Proceedings of the 2006 ACM Symposium on Information, computer and communications security (pp. 283-288).

McDermott-Wells, P. (2004). Bluetooth scatternet models. IEEE potentials, 23(5), 36-39.

Mosenia, A., & Jha, N. K. (2016). A comprehensive study of security of internet-of-things. IEEE Transactions on emerging topics in computing, 5(4), 586-602.

Mukherjee, A. (2015). Physical-layer security in the Internet of Things: Sensing and communication confidentiality under resource constraints. Proceedings of the IEEE, 103(10), 1747-1761.

Mushtaq, M. F., Jamel, S., Disina, A. H., Pindar, Z. A., Shakir, N. S. A., & Deris, M. M. (2017). A survey on the cryptographic encryption algorithms. International Journal of Advanced Computer Science and Applications, 8(11), 333-344.

Ordu, L., & Yalçın, S. B. Ö. (2016, December) Yan-Kanal Analizi Saldırılarına Genel Bakış.

Perianin, T., Carré, S., Dyseryn, V., Facon, A., & Guilley, S. (2020). End-to-end automated cache-timing attack driven by machine learning. Journal of Cryptographic Engineering, 1-12.

Popp, T., Mangard, S., & Oswald, E. (2007). Power analysis attacks and countermeasures. IEEE Design & test of Computers, 24(6), 535-543.

Rizvi, S., Kurtz, A., Pfeffer, J., & Rizvi, M. (2018, August). Securing the internet of things (IoT): A security taxonomy for IoT. In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE) (pp. 163-168). IEEE.

Robshaw, M., & Billet, O. (Eds.). (2008). New stream cipher designs: the eSTREAM finalists (Vol. 4986). Springer.

Saluja, K. K. (1987). Linear feedback shift registers theory and applications. Department of Electrical and Computer Engineering, University of Wisconsin-Madison, 4.

Samani, A., Ghenniwa, H. H., & Wahaishi, A. (2015). Privacy in Internet of Things: A model and protection framework. Procedia Computer Science, 52, 606-613.

Samy, M. M., Anis, W. R., Abdel-Hafez, A. A., & Eldemerdash, H. D. (2021). An Optimized Protocol of M2M Authentication for Internet of Things (IoT). International Journal of Computer Network & Information Security, 13(2).

Schindler, W. (2015, September). Exclusive exponent blinding may not suffice to prevent timing attacks on RSA. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 229-247). Springer, Berlin, Heidelberg.

Sharif, S. O., & Mansoor, S. P. (2010, August). Performance analysis of stream and block cipher algorithms. In 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE) (Vol. 1, pp. V1-522). IEEE.

Simmons, G. J. (1979). Symmetric and asymmetric encryption. ACM Computing Surveys (CSUR), 11(4), 305-330.

Singh, D., Pushparaj, Mishra M.K., Lamba A., Swagatika S., (2020). Security issues in different layers of IoT and their possible mitigation. International Journal of Scientfic & Technology Research.

Srinivas, R. (2020, Ocak), 10 IoT Security Incidents That Make You Feel Less Secure. 17 Kasım 2021 tarihinde https://cisomag.eccouncil.org/10-iot-security-incidents-that-make-you-feel-less-secure/ adresinden alındı.

Swamy, S. N., Jadhav, D., & Kulkarni, N. (2017, February). Security threats in the application layer in IOT applications. In 2017 International conference on i-SMAC (iot in social, mobile, analytics and cloud)(i-SMAC) (pp. 477-480). IEEE.

Tiri, K., & Verbauwhede, I. (2003, September). Securing encryption algorithms against DPA at the logic level: Next generation smart card technology. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 125-136). Springer, Berlin, Heidelberg.

Vanwell, J. (2021, Ocak), IoT Security Breaches: 4 Real-World Examples. 17 Kasım 2021 tarihinde https://www.conosco.com/blog/iot-security-breaches-4-real-world-examples/ adresinden alındı.

Walter, C. D. (1999). Montgomery exponentiation needs no final subtractions. Electronics letters, 35(21), 1831-1832.

Walter, C. D. (2002, February). MIST: An efficient, randomized exponentiation algorithm for resisting power analysis. In Cryptographers' Track at the RSA Conference (pp. 53-66). Springer, Berlin, Heidelberg.

Walter, C. D., & Thompson, S. (2001, April). Distinguishing exponent digits by observing modular subtractions. In Cryptographers' Track at the RSA Conference (pp. 192-207). Springer, Berlin, Heidelberg.

Wang, W., He, G., & Wan, J. (2011, September). Research on Zigbee wireless communication technology. In 2011 International Conference on Electrical and Control Engineering (pp. 1245-1249). IEEE.

Want, R. (2006). An Introduction to RFID Technology, IEEE Pervasive Computing. Jan.

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

Wei, W., Yang, A. T., Shi, W., & Sha, K. (2016, October). Security in internet of things: Opportunities and challenges. In 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI) (pp. 512-518). IEEE.

Williams, D. (2008). The tiny encryption algorithm (tea). Network Security, 1-14.

Won, Y. S., Chatterjee, S., Jap, D., Bhasin, S., & Basu, A. (2021). Time to Leak: Cross-Device Timing Attack On Edge Deep Learning Accelerator. In 2021 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-4). IEEE.

Zhang, H., & Wang, X. (2009). Cryptanalysis of Stream Cipher Grain Family. IACR Cryptol. ePrint Arch., 2009, 109.

Zhao, K., & Ge, L. (2013, December). A survey on the internet of things security. In 2013 Ninth international conference on computational intelligence and security (pp. 663-667). IEEE.

Zhao, X. J., Wang, T., & Zheng, Y. (2009). Cache Timing Attacks on Camellia Block Cipher. IACR Cryptol. ePrint Arch., 2009, 354.

ÖZGEÇMİŞ

Ad-Soyad : Muhammed Saadetdin KAYA

ÖĞRENİM DURUMU:

- **Lisans :** 2019, Aydın Adnan Menderes Üniversitesi, Mühendislik Fakültesi, Elektrik-Elektronik Mühendsiliği (İng.)

YÜKSEK LİSANS TEZİNDEN TÜRETİLEN ÇALIŞMALAR

- **KAYA M.S. & İNCE K., (2021).** Nesnelerin İnternetinde Rassal ve Kontrollü Bekleme Süresi İle Zamanlama Analizi Saldırılarının Önlenmesi . Computer Science , 5th International Artificial Intelligence and Data Processing symposium , 61-69 . DOI: 10.53070/bbd.990915
- **KAYA M.S. & İNCE K., (2021).** Nesnelerin İnternetinde Çok Katmanlı Algılayıcı Kullanarak Zamanlama Analizi Saldırısı ile Özel Anahtar Tahminlemesi . Computer Science , 5th International Artificial Intelligence and Data Processing symposium , 385-390 . DOI: 10.53070/bbd.990849