KABLOSUZ ALGILAYICI AĞLARDA GÜVENLİ BİR VERİ BAĞI KATMANI PROTOKOLÜ TASARIMI VE GERÇEKLEŞTİRİLMESİ

Murat DENER

DOKTORA TEZİ
ELEKTRONİK – BİLGİSAYAR EĞİTİMİ

GAZİ ÜNİVERSİTESİ
BİLİŞİM ENSTİTÜSÜ

EKİM 2012
ANKARA

Murat DENER tarafından hazırlanan KABLOSUZ ALGILAYICI AĞLARDA GÜVENLİ BİR VERİ BAĞI KATMANI PROTOKOLÜ TASARIMI VE GERÇEKLEŞTİRİLMESİ adlı bu tezin Doktora tezi olarak uygun olduğunu onaylarım.

Prof.Dr. Ömer Faruk BAY
Tez Yöneticisi

Bu çalışma, jürimiz tarafından oy birliği / ~~oy çokluğu~~ ile Elektronik-Bilgisayar Eğitimi Anabilim Dalında Doktora tezi olarak kabul edilmiştir.

Başkan: : Prof.Dr. İsmail ERTÜRK

Üye : Prof.Dr. Ömer Faruk BAY

Üye : Doç.Dr. O. Ayhan ERDEM

Üye : Doç.Dr. Bülent TAVLI

Üye : Yrd.Doç.Dr. Hüseyin POLAT

Tarih : 17/10/2012

Bu tez, Gazi Üniversitesi Bilişim Enstitüsü tez yazım kurallarına uygundur.

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

Murat DENER

KABLOSUZ ALGILAYICI AĞLARDA GÜVENLİ BİR VERİ BAĞI KATMANI PROTOKOLÜ TASARIMI VE GERÇEKLEŞTİRİLMESİ

(Doktora Tezi)

Murat DENER

GAZİ ÜNİVERSİTESİ

BİLİŞİM ENSTİTÜSÜ

Ekim 2012

ÖZET

Algılayıcı düğümlerin donanımsal kısıtları, kablosuz iletişim ortamı, gerçek zamanda işlem ihtiyacı, heterojen yapısı, düğüm sayısının fazlalığı, ölçeklenebilirlik ihtiyacı, gezginlik, uygulama ortam şartlarının ağırlığı ve maliyet gibi hususlardan kaynaklanan nedenlerle Kablosuz Algılayıcı Ağlar (KAA) pek çok güvenlik açığıyla karşı karşıyadır. Düşman hatlarının gözetlenmesi ya da sınır bölgelerinin gözetlenmesi gibi hassas KAA uygulamalarında, algılayıcılardan baz istasyonuna gizli veri aktarımını sağlayan güvenlik protokolleri mutlaka kullanılmalıdır. KAA için gereken güvenlik gereksinimlerinden (Veri Gizliliği, Veri Bütünlüğü, Veri Tazeliği, Kimlik Doğrulama, Kullanılırlık) her birini karşılayan, ama her bir gereksinimi karşılamak için yüksek güvenlik, düşük enerji tüketimi düşüncesiyle güvenlik protokolü geliştirmek gerekmektedir. Ayrıca önerilen çoğu güvenlik çözümünün benzetim ortamında kalması, algılayıcı platformlar üzerinde önerilen çözümün değerlendirilmemiş olması yapılan araştırmalar için bir eksikliktir. Bu yüzden; önerilen protokollerin doğrudan güvenlik gerektiren uygulamalarda kullanılabilmesi için algılayıcı düğümler üzerinde de uygulamasının yapılması gerekmektedir. Literatürde bulunan güvenlik çözümlerinden sadece TinySec ve MiniSec algılayıcı düğüm üzerinde uygulanmıştır. Fakat, mevcut güvenlik protokollerinin anahtar boyutu,

kullanılırlık, veri bütünlüğü hususlarında eksiklikleri vardır. Bu çalışmada, belirtilen eksikliklerin giderilerek, KAA için gereken güvenlik gereksinimlerinden veri gizliliği, veri bütünlüğü, veri tazeliği, kimlik doğrulama ve kullanılırlık prensiplerinden hepsini karşılayan ve bununla birlikte enerji verimli yeni bir veri bağı katmanı güvenlik protokolü geliştirilmiştir. Aynı zamanda, geliştirilen protokol algılayıcı düğümler üzerinde uygulanmıştır. Geliştirilen protokolde hem güvenlik artırılmış hem de KAA için önem teşkil eden enerji, bellek kullanımı gibi kriterler de TinySec'e göre daha iyi sonuçlar elde edilmiştir. Gecikme kriterinde ise TinySec'e oranla ihmal edilebilir bir düzeyde artış göstermiştir.

Bilim Kodu : 702.3.006
Anahtar Kelime : kablosuz algılayıcı ağlar, veri bağı katmanı, ortam erişim kontrolü, güvenlik protokolü
Sayfa Adedi : 90
Tez Yöneticisi : Prof.Dr. Ömer Faruk BAY

DESIGN AND IMPLEMENTATION OF A SECURE DATA LINK LAYER PROTOCOL FOR WIRELESS SENSOR NETWORKs
(PhD. Thesis)

Murat DENER

GAZİ UNIVERSITY

INFORMATION INSTITUTE

October 2012

ABSTRACT

Wireless Sensor Networks (WSNs) is caused by a lot of vulnerability because of factors such as hardware constraints of the sensor nodes, wireless communication medium, real-time computing, heterogeneous structure, large number of nodes, scalability, mobility, weight and cost requirements of application environment. In sensitive WSNs applications like surveillance of enemy lines or border areas security protocols must be used which providing confidential data transfer from sensors to base station. Security protocol must be developed with security requirements for WSNs which are Data Confidentiality, Data Integrity, Data Freshness, Data Authentication, Availability and the idea with high security – low energy consumption. Also, it is deficiency that most of the proposed security solutions evaulated on simulation environment not on sensor platforms. Therefore, security protocols must be implemented on sensor nodes for they can be used directly in applications requiring security. Security solutions in the literature only found TinySec and MiniSec implemented on sensor nodes. However, the existing security protocols have deficiency such as key size, usability, data integrity, etc. In this study, a new data link layer security protocol developed which have data confidentiality, data integrity, data freshness, data authentication, availability and also energy efficient. Moreover, the protocol implemented on the sensor nodes. The security has been increased in the developed protocol. Also we have been obtained better results than

TinySec in energy consumption and memory usage which are very important for WSNs. However in the delay criterion, our protocol increased negligible level than TinySec.

Science Code : **702.3.006**
Key Words : **wireless sensor networks, data link layer, medium access control, security protocol**
Page Number : **90**
Adviser : **Prof.Dr. Ömer Faruk BAY**

TEŞEKKÜR

Çalışmalarım boyunca değerli yardım ve katkılarıyla beni yönlendiren tez danışmanım Prof.Dr. Ömer Faruk BAY'a ve manevi destekleriyle beni hiçbir zaman yalnız bırakmayan aileme sonsuz teşekkürlerimi sunarım.

İÇİNDEKİLER

Sayfa

ÇİZELGELERİN LİSTESİ

Sayfa

ŞEKİLLERİN LİSTESİ

Sayfa

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı kısaltmalar açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	**Açıklama**
KAA	Kablosuz Algılayıcı Ağlar
XXTEA	Corrected Block Tiny Encyrption Algorithm (Düzeltilmiş Blok Mini Şifreleme Algortiması)
OCB	Offset Codebook Mode (Ofset Kod Çizelgesi Modu)
CBC	Cipher Block Chaining (Zincirleme Blok Şifreleme)
ADC	Analog Digital Converter (Analog Sayısal Çevirici)
FPGA	Field Programmable Gate Array (Alan Programlanabilir Kapı Dizileri)
RF	Radyo Frekansı
MAC	Medium Access Control (Ortam Erişim Kontrolü)
DoS	Denial of Service (Servis Reddetme)
FIFO	First In First Out (İlk Giren İlk Çıkar)
CSMA	Carrier Sense Multiple Access (Taşıyıcıyı Dinleyen Çoklu Erişim)

1. GİRİŞ

Düşük maliyetli algılayıcı mimarilerindeki gelişmeler Kablosuz Algılayıcı Ağlarını (KAA) yeni ve bilinen araştırma alanı yapmıştır [1]. Bu ağlar çok sayıda sınırlı kapasiteli, kısa mesafeli vericiye sahip, düşük güçlü ve düşük maliyetli algılayıcının kolayca erişilemeyen ve çoğu zaman güvenilir olmayan bir ortama rasgele bırakılmasıyla oluşur [2]. Her bir düğüm hesaplama, algılama ve iletişim yeteneklerine sahiptir [3]. Gözlem yapılacak ortama rasgele dağıtılabilen bu düğümler, birbirlerini tanıyabilmekte ve ortak gayret sarf ederek geniş bir alanda ölçüm vazifesini gerçekleştirebilmektedir. Bu özelliklerinden dolayı sağlık alanlarından askeri alanlara, bir binanın güvenliğinin sağlanmasından orman yangınlarının önceden tespitine kadar çok çeşitli alanlarda kullanılabilmektedirler[4].

Algılayıcı düğümlerin donanımsal kısıtları, kablosuz iletişim ortamı, gerçek zamanda işlem ihtiyacı, heterojen yapısı, düğüm sayısının fazlalığı, ölçeklenebilirlik ihtiyacı, gezginlik, uygulama ortam şartlarının ağırlığı ve maliyet gibi hususlardan kaynaklanan nedenlerle KAA pek çok güvenlik açığıyla karşı karşıyadır. Güvenliğin temel hedefi olan gizlilik, bütünlük ve kullanılabilirliğin sağlanması, zaman ve hayati önemdeki amaçların gerçekleştirilebilmesi için çözülmesi gereken en önemli problemlerden birini oluşturmaktadır [5]. Kişisel ya da dizüstü bilgisayarlar gibi donanımsal ve yazılımsal olarak güçlü düğümlerden meydana gelen ve kablolu bir mimari ile oluşturulan klasik bilgisayar ağları ile karşılaştırıldığında KAA kendilerine özgü pek çok özellik göstermektedirler. Bu özelliklerin birçoğu güvenlik probleminin çözümünü büyük ölçüde güçleştirmektedir. Ancak, saldırganların da genellikle aynı kısıtlamalar ile bağlı olmaları, bu özelliklerin bazı durumlar için ihtiyaçlar doğrultusunda kullanılabilmesini de mümkün kılmaktadır

Düşman hatlarının gözetlenmesi ya da sınır bölgelerinin gözetlenmesi gibi hassas KAA uygulamalarında, algılayıcılardan baz istasyonuna gizli veri aktarımını sağlayan güvenlik protokolleri mutlaka kullanılmalıdır. Ancak, algılayıcıların düşük işlemci ve radyo kapasiteleri geleneksel güvenlik protokollerinin KAA'larda uygulanmasına olanak tanımamaktadır [6].

Algılayıcı ağları pratikte kullanılabilir hale getirmek için günümüzde yazılımda ve donanımda birçok iyileştirme çalışmaları yapılmaktadır. Ancak yapılan her çalışmanın temelinde birim başına düşen enerji sarfiyatının azaltılması ve dolayısıyla algılayıcı düğümlerin ömrünün uzatılması vardır. Bunun temel sebebi çalışma ortamında bulunan algılayıcıların ve kullanılan enerji kaynaklarının değiştirilmesinin veya yeniden doldurulmasının çoğu zaman imkânsız ve aşırı maliyetli olmasıdır [7].

Literatürde ortam erişimi konusunda yapılan çalışmaların çoğunda enerji tüketimini en aza indirecek yaklaşımlar geliştirmek ilk amaç olmasına rağmen; KAA'lar, özellikle askeri uygulamalar başta olmak üzere birçok alanda veri gizliliği, veri bütünlüğü, veri tazeliği, kimlik doğrulaması, kullanılırlık gibi güvenlik gereksinimlerini sağlamak zorundadır [8]. Geleneksel ağlar için tasarlanan ve günümüzde birçok uygulamada yaygın olarak kullanılan güvenlik yöntemleri, algılayıcı düğümlerin, kısıtlı enerji kaynaklarına, yetersiz bellek kapasitelerine ve sınırlı işlem kabiliyetlerine sahip olmalarından dolayı, KAA'larda doğrudan uygulanamamaktadır [9]. Günümüzde, KAA uygulamalarının ve içerdikleri düğümlerin bu özellikleri göz önüne alınarak çeşitli güvenlik protokolleri geliştirilmektedir. Güvenlik konusunda yapılan araştırmalar ve uygulamalar, KAA alanındaki diğer tüm çalışmalarda olduğu gibi, genellikle enerji ve diğer kaynakların etkin kullanımı üzerine odaklanmaktadır [10].

KAA'daki güvenlik gereksinimlerinden çoğu, açık anahtarlı şifreleme ile sağlanmasına rağmen, açık anahtarlı şifreleme tekniğinde bellek boyutu gereksiniminin büyük olması ve performansın yavaş olmasından dolayı KAA'da direk kullanılamamaktadır.

Bundan dolayı KAA'da çoğu zaman gizli anahtarlı şifreleme kullanılmaktadır. Bununla beraber, gizli anahtarlı şifrelemede sadece veri gizliliği garanti edilmekte, geri kalan güvenlik gereksinimleri (veri bütünlüğü, kimlik doğrulama, veri tazeliği, kullanılırlık) karşılanamamaktadır. Yapılan deneysel sonuçlara göre, veri gizliliği için 64 bitlik anahtar kullanan şifreleme algoritmaları, saniyede 10^{12} şifre

deneyebilen süper bilgisayarlar tarafından 3.5 ayda kırılabilmektedir. Bu zaman değeri 128 bitlik anahtar kullananlar için $5.4x10^{18}$ yıldır [11]. Burada 128 bitlik şifreleme algoritması kullanmak mantıklı görünse de, belki de bu algoritmaya ait gereken bellek miktarı yada şifre açma/çözme süresi uzun olacağından kullanılması doğru olmayacaktır. Bununla birlikte şifreleme algoritmasının kullanılabilmesi için gereken çalışma kipi (CBC [12], OBC [13])' de önem arz etmektedir. Yani, sadece veri gizliliğini sağlamak için bile detaylı olarak hangi algoritma ve hangi çalışma kipi kullanılacağı düşünülmelidir.

Diğer güvenlik gereksinimlerinden veri bütünlüğü ve kimlik doğrulama için çoğu zaman mesaj doğrulama kodları (Message Authentication Code, MAC) kullanılmaktadır. Bu kod sayesinde veri bütünlüğü ve kimlik doğrulama sağlanmasına rağmen bu kodun mesaja eklenmesi, doğal olarak enerji tüketimini artıracaktır. Veri tazeliği için mesaja eklenen sayaç değeri, bununla birlikte; kullanılırlık prensibini karşılamak için gerçekleştirilen savunma mekanizması da enerjinin fazlaca tüketimine sebep olacaktır.

Kullanılırlık prensibinin karşılanması demek, o protokolün DoS ataklarına karşı [14] dayanıklı olmasını gerektirir. Mesaj değiştirme, tekrarlama atakları, paket çakışması atağı (Collision attack), adaletsizlik atağı (Unfairness attack), tüketim atağı (Exhaustion attack), DoS ataklarına örnek verilebilir. Bu ataklardan herhangi birinin karşılanamaması, o protokolü güvensiz kılar. Sonuçta güvenliği garanti etmek için, mevcut olabilecek açıkların hepsi kapatılmalıdır.

Bütün gereksinimleri karşılarken, KAA'nın sınırlı donanımsal kaynaklara sahip olduğu ve KAA'ın ilk amacının enerji verimliliği olduğu unutulmamalıdır. Aksi takdirde, bütün güvenlik önlemlerini karşılasa bile fazlaca enerji tüketen bir protokol, KAA için kullanışsız olacaktır.

KAA için gereken güvenlik gereksinimlerinden (Veri Gizliliği, Veri Bütünlüğü, Veri Tazeliği, Kimlik Doğrulama, Kullanılırlık) [15] her birini karşılayan, ama her bir gereksinimi karşılamak için yüksek güvenlik, düşük enerji tüketimi düşüncesiyle

güvenlik protokolü geliştirmek gerekmektedir. Ayrıca önerilen çoğu güvenlik çözümünün benzetim ortamında kalması, algılayıcı platformlar üzerinde önerilen çözümün değerlendirilmemiş olması yapılan araştırmalar için bir eksikliktir. Bu yüzden; önerilen protokollerin doğrudan güvenlik gerektiren uygulamalarda kullanılabilmesi için yalnız benzetim ortamında kalmaması, algılayıcı düğümler üzerinde de uygulamasının yapılması gerekmektedir.

Literatürde bulunan güvenlik çözümlerinden sadece TinySec [16] ve MiniSec [17] algılayıcı düğüm üzerinde uygulanmıştır. IEEE 802.15.4 [18] ise Kablosuz Özel Alan Ağları için geliştirilmiş olmasına rağmen düşük güç tüketimi, düşük maliyet ve esnek oluşundan dolayı KAA'da kullanılmaktadır. Diğer güvenlik protokolleri düğüm üzerine uygulanmamıştır. TinySec ve MiniSec veri gizliliğini garanti etmek için 80 bit anahtar boyutlu Skipjack algoritmasını kullanmıştır. Yapılan araştırmalar göstermektedir ki veri gizliliği için anahtar boyutunun en az 128 bitlik olması gerekmektedir. TinySec mesaj tekrar yayınlama ataklarını önleyemezken, MiniSec'te verinin bütünlüğü garanti edilememektedir. Ayrıca bu protokoller KAA için güvenlik gereksinimlerinden kullanılırlık ilkesini sağlayamamaktadır. Kullanılırlık gereksiniminin karşılanmaması demek, o protokolün DoS saldırılarına karşı dayanıksız olması anlamına gelmektedir.

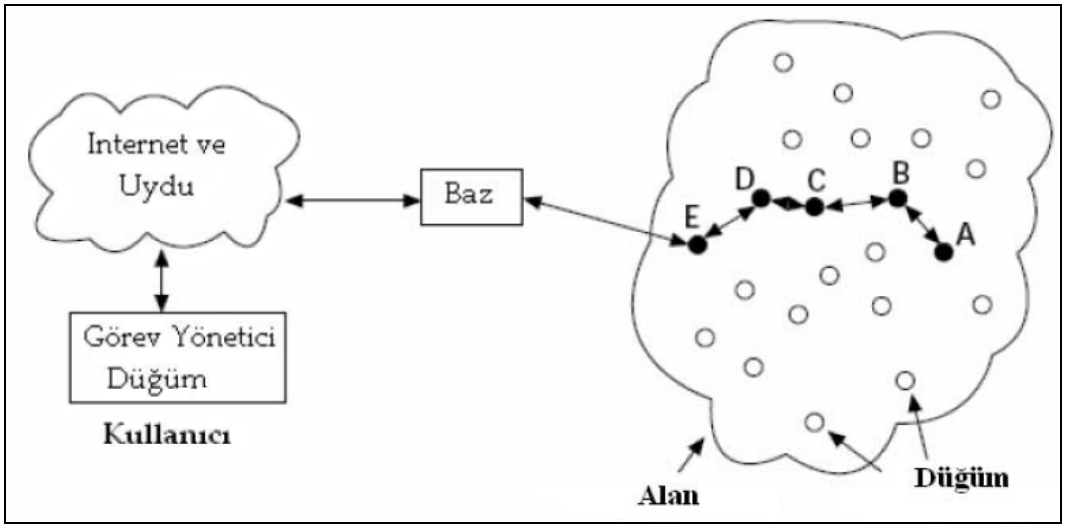
Mevcut güvenlik protokolleri, KAA'da gereken güvenlik gereksinimlerinin (Veri Gizliliği, Veri Bütünlüğü, Veri Tazeliği, Kimlik Doğrulama, Kullanılırlık) hepsini karşılayamadığından güvenlik zaafı meydana gelmektedir. Geliştirilen protokolde yukarıda verilen gereksinimlerin tümü yüksek güvenlik, düşük enerji tüketimi ilkesi doğrultusunda gerçekleştirilmiştir. Ayrıca geliştirilen protokol algılayıcı düğümler üzerinde uygulanmıştır.

Bu çalışmada, belirtilen eksikliklerin giderilerek, KAA için gereken güvenlik gereksinimlerinden veri gizliliği, veri bütünlüğü, veri tazeliği, kimlik doğrulama ve kullanılırlık prensiplerinden hepsini karşılayan ve bununla birlikte enerji verimli yeni bir veri bağı katmanı güvenlik protokolü geliştirilmiştir. Aynı zamanda, geliştirilen protokol algılayıcı düğümler üzerinde uygulanmıştır.

İkinci bölümde, algılayıcı düğüm bileşenleri, KAA'ya ait uygulama alanları, karakteristikleri, güvenlik gereksinimleri, avantaj ve dezavantajları, TinyOS işletim sistemi ve NesC programlama dili, son olarak da KAA'da şifreleme anlatılmaktadır. Üçüncü bölümde, literatürde bulunan mevcut güvenlik protokollerinden bahsedilmektedir. Dördüncü bölümde geliştirilen protokol anlatılırken, Beşinci bölümde ise geliştirilen protokolün analizi yer almaktadır.

2. KABLOSUZ ALGILAYICI AĞLAR (KAA)

Kablosuz Algılayıcı Ağlar, ortama yerleştirilmiş küçük boyutlu algılayıcı düğümlerden oluşur. Bu düğümler fiziksel bir alanda iş birliği içerisine girerek fiziksel dünyadan öğrendiklerini sanal dünya ortamına taşımaktadırlar [19].



Şekil 2.1. Kablosuz algılayıcı ağlar

Algılayıcı ağlarda fiziksel dünyadan, çeşitli algılayıcılar yardımıyla algılanan veriler kablosuz bir biçimde kulaktan kulağa olarak adlandırılan işbirliği yöntemiyle hedefleri olan bilgi işlem ağına aktarılmaktadır. Bilgi işlem ağına olan geçit Baz istasyonu olarak adlandırılmaktadır. Bu istasyon hem algılayıcı düğümleri hem de haberleşme ağı ile iletişim kurabilen özel bir düğümdür. Baz düğümü enerji problemi olmayan statik ve hesaplama kabiliyeti yüksek bir düğüm olarak kabul edilir.

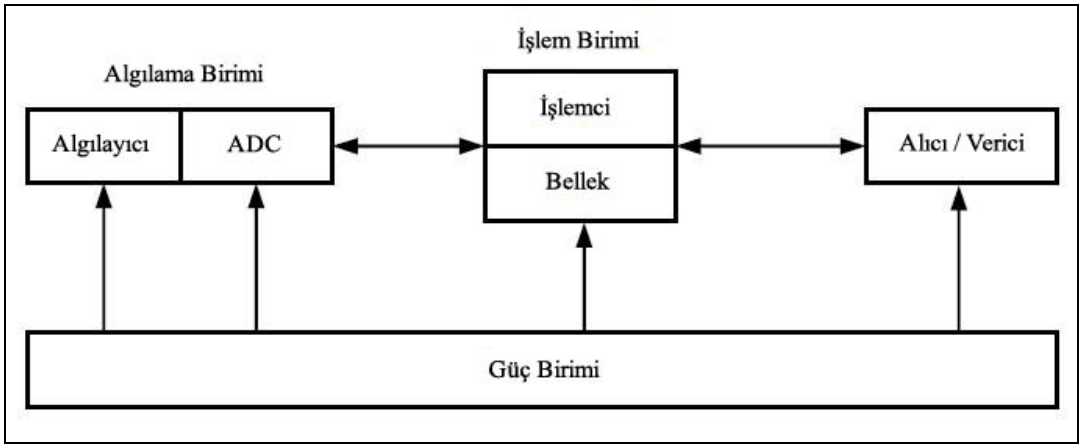
Algılayıcı düğümleri ise kablosuz ve genellikle radyo teknolojisi ile iletişim kuran, enerji ve hesaplama kabiliyetleri kısıtlı birimlerdir. Bu birimler algılama alanındaki bazı durumları ve olayları algılamak ve takip etmek amacıyla otomatik olarak yerleştirilmekte ve kurulmaktadırlar. Sayıları ise uygulamaya göre yüzlerce hatta binlerce olabilmektedir. Küçük boyutlara sahip olmaları ise kullanılabilirlik açısından fiziksel bir gereksinimdir.

Şekil 2.2'de [20] örnek bir algılayıcı düğüm gösterilmiştir.

Şekil 2.2. TelosB düğümü

2.1. Algılayıcı Düğüm Bileşenleri

Algılayıcı düğümler Şekil 2.3.'de gösterildiği gibi algılama birimi, işlem birimi, alıcı-verici ve güç biriminden oluşmaktadır. Her bir bileşen aşağıda tanımlanmıştır.



Şekil 2.3. Algılayıcı düğüm bileşenleri

Algılama birimi

Algılama Biriminin ana işlevi algılama ve hedef alandaki verilerin fiziksel olarak ölçülmesidir. Analog gerilim veya sinyal gözlemlenen olay neticesinde algılayıcı tarafından oluşturulur [21]. Sürekli dalga, analog-digital çevirici (ADC) tarafından sayısallaştırılır ve sonra analiz için işlem birimine iletilir [22,23]. Algılama

teknolojileri yarı iletken teknolojilerinden daha yavaş ilerlemesinden dolayı algılama birimi mevcut teknolojide bir darboğazdır [24].

İşlem birimi

İşlem Birimi, önceden tanımlanmış görevleri başarmak için algılayıcılar arasındaki işbirliği yönetiminde önemli bir rol oynar. Mikrodenetleyiciler, mikroişlemciler ve alan programlanabilir kapı dizileri (field-programmable gate arrays, FPGAs) olmak üzere bu birimde çeşitli aileler vardır [25]. FPGA'ler fazla enerji tüketir ve geleneksel programlama metodolojileri ile uyumlu değildir. Bununla birlikte, FPGA'lar dağıtım maliyetlerini ortadan kaldırmak için tekrar programlanabilir ve tekrar konfigure edilebilir [26].

Uçucu olmayan bellek ve ADC ler gibi arayüzler tek bir entegre devre üzerine entegre edilebilir. İşlem birimi görevlerini yürütmek için belleğe ihtiyaç duyar ve yerel işleme ve veri toplama yoluyla iletilen mesajları en aza indirger. Flash bellek, depolama kapasitesi ve fiyatı nedeniyle yaygın olarak kullanılır.

Alıcı / Verici

Algılayıcılarda, optik iletişim (lazer), kızılötesi, radyo frekansı (RF) olmak üzere üç farklı iletişim düzeni vardır. Lazer, radyoya göre daha az enerji tüketir ve yüksek güvenlik sağlar, fakat görüş hattı (line of sight) gerektirir ve atmosferik şartlarda hassastır. Kızılötesi, lazer gibi antene ihtiyaç duymaz fakat yayın kapasitesi sınırlıdır. RF'yi kullanmak çok kolaydır fakat antene ihtiyaç duyulur [21].

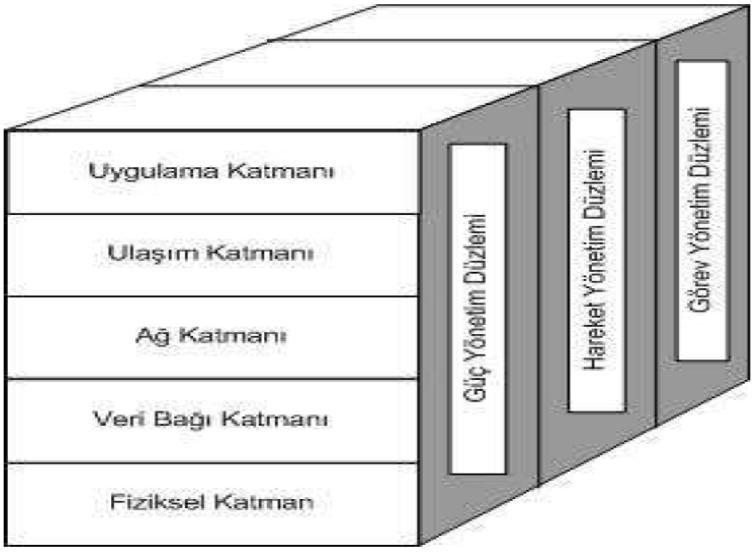
Modülasyon, filtreleme, demodülasyon gibi çeşitli enerji tüketimi azaltma stratejileri geliştirilmiştir. Genlik ve frekans modülasyonu, standard mekanizmalardır. Genlik modülasyonu basittir fakat gürültüye karşı hassastır. RF Monolithics TR1000 ve Chipcon 1000 ticari radyolardır ve çeşitli uygulamalarda yaygın olarak kullanılır. Chipcon 1000, 300 ve 1000 MHz arasındaki frekanslarda işlem yapmak için kolayca programlanabilir [26].

Güç birimi

Güç tüketimi algılayıcı ağlarda önemli bir zayıflıktır. Herhangi bir enerji koruma düzeni sensör ömrünü uzatmak için yardımcı olabilir. Algılayıcılarda kullanılan piller iki grupta kategorize edilebilir. – şarj edilebilir – şarj edilemez. Çoğunlukla zorlu ortamlarda pili şarj etmek veya pili değiştirmek imkansızdır. Günümüzdeki algılayıcılar yenilenebilir enerji kaynaklarını da (güneş enerjisi, ısı enerjisi, titreşim enerjisi vb.) kullanabilecek şekilde geliştirilmektedir [22,24].

İki önemli güç tasarrufu politikası mevcuttur [26]. Birinci yaklaşımda, kullanılmayan aygıtlar kapatılabilir ve gerektiği zaman aktif edilebilir. Dinamik güç yönetimi (Dynamic Power Management, DPM) olarak bilinen bu olay gelecekteki olayların tahmini için stokastik analiz ve işletim sistemi desteği gerektirir. Diğer bir yaklaşım ise dinamik gerilim planlama (Dynamic Voltage Scheduling ,DVS)'dır.

2.2. Protokol Yığını



Şekil 2.4. Protokol yığını

Protokol yığını, fiziksel katman, veri bağı katmanı, ağ katmanı, ulaşım katmanı, uygulama katmanı, güç yönetim düzlemi, hareket yönetim düzlemi ve görev yönetim düzlemi olarak bölümlendirilmiştir.

Fiziksel katman, frekans seçimi, taşıyıcı frekansın oluşumu, sinyal algılama, modülasyon işlemlerinden sorumlu olan katmandır [27].

Veri bağı katmanı, ortama erişim kontrolü (MAC), hata kontrolü, veri çerçevesinin algılanması, veri şifrelemesi olaylarından sorumlu olan katmandır [27].

Ağ katmanı, ulaşım katmanı tarafından iletilen verinin yol bulma işlemiyle ilgilenmektedir [27].

Ulaşım katmanı, veri akış kontrolü yapar. Algılama işlemleri sayesinde farklı tipte uygulamalar geliştirilebilir [27].

Bu uygulamalar, uygulama katmanı sayesinde kullanılmaktadır [27].

Bu katmanların yanı sıra protokol yapısında, güç, hareket ve görev yönetim düzlemleri bulunmaktadır. Bu düzlemler, düğümlerin algılama işlemlerini düzenlemek ve enerji tüketimini düşük seviyede tutmaya yardımcı olmaktadırlar.

Güç yönetim düzlemi, düğümlerin güç kullanımını yönetmektedir. Düğüm, herhangi bir komşu düğümden paketi aldıktan sonra kapatılarak aynı paketi tekrar alması önlenebilir. Düğümün enerji seviyesi alt eşik değerinin altına düşerse, komşu düğümlere mesaj göndererek yol bulma işlemlerine katılamayacağı bildirebilir. Bu sayede düğüm kalan enerjisini sadece algılama işlemi için kullanabilir [27].

Hareket yönetim düzlemi, düğümlerin hareketlerini algılayarak kaydetmektedir. Bu sayede komşu düğümler ve geri dönüş yolu rahatlıkla bulmaktadır. Komşu düğümlerin bilinmesi, düğümün görev ve güç yönetimini de dengelemektedir [27].

Görev yönetim düzlemi, düğümlerin algılama görevlerini düzenler ve dengeler [27].

Yönetim düzlemleri, düğümlerin güçlerini verimli bir şekilde kullanmalarını, hareketli KAA'da verinin yol bulmasını ve düğümler arasında kaynakların paylaşımını sağladıklarından KAA için büyük bir önem taşımaktadır.

2.3. Uygulama Alanları

KAA'ın kullanım alanları her geçen gün artmaktadır. Askeri, çevre, sağlık, ticari, ev otomasyonu vb. alanlarında yaygın olarak kullanılmaktadır.

Askeri uygulamalar

Savaş alanlarının gözetim altında tutulması, düşman hareketlerinin izlenmesi, arazi hakkında keşifte bulunmak, personel ve askeri araçların takip edilmesi, dost kuvvetlerin izlenmesi ve hedeflerin hız ve konumlarının tespit edilmesinde kullanılmaktadır.

Çevresel uygulamalar

Hava durumu, hava kirliliğinin tespiti, sel, deprem, orman yangını gibi doğal afetlerin takip edilmesi, tarımsal faaliyetlerin izlenmesi gibi uygulamalarda kullanılır.

Sağlık uygulamaları

Hastanede bulunan doktorların yerinin tespit edilmesi, hastaların durumlarının takip edilmesi, yaşlıların gözetim altında tutulması ve çeşitli sağlıksal parametrelerin takip edilmesinde kullanılır.

Ticari uygulamalar

Araçların izlenmesi ve tespit edilmesi, enerji hatlarının izlenmesi, küçük çocukların aileleri tarafından takip edilmesi, ışıklandırma kontrolü, trafik ışıklarının kontrolü, yangın sistemleri gibi alanlarda kullanılır.

Ev otomasyon uygulamaları

Zeki ev ortamları ve bina güvenlik sistemlerinde kullanılır.

2.4. Karakteristikleri

Geleneksel güvenlik protokollerini KAA'larda kullanmayı engelleyen ve sadece KAA'lara ait karakteristikler aşağıda özetlenmiştir. Açıklanan karakteristiklerin protokol tasarımı ve geliştirilmesi sırasında dikkate alınması protokolün kullanılabilirliğini artırmaktadır [28].

Büyük ölçek

KAA'ların genel uygulamaları (örneğin askeri gözetleme uygulamaları) coğrafi açıdan geniş bir alanın kapsanmasını gerektirir [30]. Ayrıca düğümlerin yüksek ölüm oranları, kısıtlı radyo kapasiteleri, güvenilirliği düşük ucuz algılayıcılar sebebiyle KAA'lar genelde çok büyük ölçekte olabilir ve bir KAA'daki düğüm sayısı on binleri asabilir [29].

Kısıtlı kaynak

KAA'ların düşük kurulum ve işletim maliyetli olma zorunluluğu algılayıcı düğümlerinin donanım açısından sade olmasını gerektirir. Bu nedenle KAA'larda işlem ve iletişim kaynakları kısıtlıdır. Örneğin genel bir algılayıcı türü olan (TelosB) 16 bitlik, 8 MHz işlemci, 48KB ana hafıza, 1024 KB anlık belleğe sahiptir [31]. İşlemci kapasitesinin düşüklüğü, hafıza ve radyo iletiminin kısıtlı olması, ağ ömrünün batarya ömrü ile sınırlı olması KAA'lar için tasarlanan her protokolü etkilemektedir [30].

Artıklılık

Düğüm artıklılığı nedeniyle her olay birden fazla algılayıcı düğümü tarafından algılanır ve dolayısı ile ağda taşınması gereken veri miktarı artar [30]. Başka bir

deyişle artıklık baz istasyonuna gönderilen verilerin miktarını artırmakta ve ağın yasam süresini azaltmaktadır [30]. Veri artıklılığından kurtulmak için veri kümeleme protokolleri kullanılmaktadır.

Güvenlik

Askeri sistemler ve tıbbi takip sistemleri gibi KAA uygulamaları güvenlik açısından çok hassastırlar. Algılayıcı düğümlerinin kısıtlı kaynaklarından dolayı geleneksel güvenlik mekanizmaları KAA'larda kullanılamaz. Buna ek olarak KAA'larda geleneksel ağlarda görülmeyen algılayıcıların fiziksel güvenliklerinin olmaması sorunu vardır. Algılayıcıların fiziksel güvenlikleri sağlanamadığından, ağdaki algılayıcı düğümleri her an kötü niyetli kişilerce ele geçirilip, kötü amaçlar için kullanılabilirler. Bu nedenlerden dolayı KAA'ların güvenlik mekanizmaları algılayıcı düğümlerinin kaynak kısıtları ve kötücül algılayıcılar göz önüne tutularak tasarlanmalıdır [30].

Veri merkezli işleme

Veri merkezli işleme KAA'ların en önemli özelliklerindendir. Algılayıcı düğümlerin ID'leri uygulamalar için çoğu zaman önemli değildir. Bu nedenle KAA uygulamalarında adlandırma düzeni çoğunlukla veriye yöneliktir (data oriented). Örneğin bir çevre gözetim sisteminde sıcaklık ölçümü yapmak için "X,Y ve Z düğümlerinden sıcaklık ölçüm değerlerini topla" şeklinde değil, "(X1,Y1,X2,Y2) koordinatları ile sınırlandırılmış bölgeden sıcaklık ölçüm değerlerini topla" seklinde olur. O bölgedeki algılayıcı düğümlerinin ID lerinin uygulama için bir önemi yoktur [30].

Tahmin edilemezlik

Algılayıcıların donanımlarının fiyatının çok düşük olması, hava durumu ve zor çevre koşulları gibi nedenlerle algılayıcı düğümlerinde ölçüm hataları çok yaygındır. Bu

nedenle KAA tasarımında çevrim-içi (on-line) gözetim ve geri beslemeli kontrol, yüksek servis kalitesini (quality-of-service) sağlamak için gereklidir [30].

Gerçek zamanlı kısıtları

KAA'lar gerçek dünya işlemlerinde kullanıldıklarından çoğu zaman gerçek zaman kısıtlamalarına uymaları gerekmektedir. Gözetim sistemlerinde, örneğin iletişimdeki gecikme doğrudan doğruya uygulamanın hedef bulma niteliğini olumsuz yönde etkilemektedir [30].

2.5. Güvenlik Gereksinimleri

KAA'da güvenlik gereksinimleri olarak veri gizliliği, veri bütünlüğü, veri tazeliği, kimlik doğrulama ve kullanılırlık sayılmaktadır [32,33,34,35].

2.5.1. Veri gizliliği

Veri gizliliği KAA'larda, toplanan veriye yetkisiz kişilerin erişiminin engellenmesini garantiye almaktadır ve hassas KAA uygulamalarında en önemli gereksinimden biridir. Bir algılayıcı düğümün çevreden okuduğu verileri komşularına sızdırmaması gerekir. Özellikle askeri uygulamalarda düğümlerde depolanan veriler çok hassas olabilir. Ayrıca birçok uygulamalarda düğümler çok hassas verileri, (örneğin, anahtar dağılımı) kablosuz iletim ortamı üzerinden diğer algılayıcı düğümlerine aktarmak zorundadırlar. Bunlara ilaveten yönlendirme verileri de kötücül düğümlere karsı gizli tutulmalıdır. Çünkü kötücül düğümler bu verilerden yararlanarak ağın performansını düşürebilirler. Bu nedenlerle KAA'larda veri aktarımı için güvenli bir iletişim kanalı oluşturulması çok önemlidir. Hassas verileri gizli tutmak için standart yaklaşım, verinin bir gizli anahtar ile şifrelenmesidir. Düşük enerji tüketimlerinden dolayı KAA' larda gizli anahtar altyapısına dayalı şifreleme algoritmaları kullanılmaktadır.

2.5.2. Veri bütünlüğü

Veri gizliliği kötücül düğümlerin veriyi ele geçirememesini garanti edebilir ama verinin yetkisiz kişilerce değiştirilmesini engelleyemez. Veri bütünlüğü iletişimde mesajın değiştirilmemesini garanti etmektedir. Bir kötücül düğüm mesajları bozarak ağın düzgün çalışmamasına neden olabilir. Dahası, doğrudan doğruya bir kötücül düğüm olmadan da mesajlar aktarım esnasında da bozulabilir. Bu nedenle veri bütünlüğü için mesaj doğrulama kodları (message authentication codes) ya da dairesel kodları (cyclic codes) kullanmak zorunludur.

2.5.3. Kimlik doğrulama

KAA'lar ortak kablosuz ortamı kullandığından, kötücül düğümlerden gelen mesajları veya yanıltma paketlerini bulmak için, kimlik doğrulama mekanizmalarına ihtiyaç duyarlar. Kimlik doğrulama metotları bir düğümün iletişim halinde olduğu düğümün kimliğini doğrulayabilmesini sağlamaktadır. Bir kötücül düğüm, kimlik doğrulama olmadan bir başka düğümün rolünü yaparak hassas bilgileri elde edebilir ve başka düğümlerin işleyişlerine engel olabilir. Eğer sadece iki düğüm iletişimdeyse kimlik doğrulama gizli anahtar kriptografisi (symmetric key cryptography) ile yapılabilir. Alıcı ve verici bir ortak gizli anahtar (secret key) paylaşımı ile tüm gönderilen mesajların doğrulama kodunu hesaplayabilir.

2.5.4. Kullanılırlık

Kullanılırlık KAA'ların servis devamlılığını servis reddi (denial-of-service -DoS) atakları sırasında da devam ettirebilmesidir. DoS saldırı çeşidi bir hizmet aksatma yöntemidir. Bir kişinin bir sisteme düzenli veya arka arkaya yaptığı saldırılar sonucunda hedef sistemin kimseye hizmet veremez hale gelmesi veya o sisteme ait tüm kaynakların tüketimini amaçlayan bir saldırı çeşididir. Ortada bir ele geçirme, zaptetme ya da teknik deyimiyle 'hack' etme yoktur. Yapılan iş, kurban sitenin kaynaklarını kullanmaya zorlayarak, sistemin hizmet verememesini sağlamaktır. DoS atakları KAA'ın her protokol katmanında gerçekleştirilebilir ve seçilen kurban düğümleri etkisiz hale getirebilir. DoS ataklarına ek olarak aşırı iletişim ya da aşırı

hesaplama yükü düğümün bataryasını beklenenden daha çabuk bitirebilir. KAA'nın kullanılırlığın sağlanamaması çok ciddi sonuçlara yol açabilir. Örneğin askeri bir izleme uygulamasında, eğer bir kaç tane düğüm doğru çalışmazsa, düşman birlikleri KAA'nın çalışmayan bu bölümünden içeri sızabilirler.

2.5.5. Veri tazeliği

KAA yapılarında algılayıcı elemanlar belirli zamanlarda bulundukları ortama ait ölçüm verileri göndermektedirler ve ölçüm değerlerinin ulaştırılma zamanları önemli olmaktadır. Saldırıcı bir unsur tarafından eski ölçüm değerleri kopyalarının yeniden yayınlanması söz konusu olabilmektedir. Dolayısıyla verinin güncel bir bilgi olduğunun kontrolü önemli olmaktadır.

2.6. Avantaj ve Dezavantajları

KAA'nın avantajları ve dezavantajları aşağıda belirtilmiştir.

Avantajları

KAA'nın kullanılması ile ilgili avantajları özetleyecek olursak;

Hatalara karşı toleranslı olması: KAA'da çok sayıda düğüm bulunmaktadır. Bu düğümlerin, sınırlı güç, fiziksel hasar ve çevresel sebeplerden dolayı hata verip işlemleri durabilir. Bu durumda KAA'nın performansının fazla etkilenmeden işlemlerine devam etmesi gerekmektedir. Bu durum KAA'nın hatalara karşı toleranslı olmasından kaynaklanmaktadır.

Ölçeklenebilirlik (Scalability): İzleme yapılacak olan bölgede çok sayıda düğüm bulunabilir. Yeni düğümlerin algılama alanına katılması mümkündür. Yeni KAA'nın, izleme bölgesinde bulunan KAA'ya katılması mümkün olmaktadır. KAA'nın bu katılımlardan sonrada işlevine devam etmesi ölçeklenebilirlikle alakalıdır.

Üretim maliyeti: KAA çok sayıda düğümden meydana geldiğinden tek bir düğümün maliyeti tüm KAA'nın maliyeti için önem taşımaktadır. Gelişen mikroişlemci teknolojisi sayesinde düğümlerin maliyeti ucuzlamıştır.

Kullanım kolaylığı: Düğümler herhangi bir ayar gerektirmeden kendi aralarında organize olarak dinamik bir şekilde çalışmakta ve değişen koşullara ayak uydurabilmektedirler.

Yeniden kullanılması: Düğümler fiziksel ortamdan çeşitli olayları algılamak için kullanılmaktadır. Bu sebepten dolayı farklı uygulamalarda ilgili olayları algılamak için yeniden kullanılmaktadırlar.

Taşınabilir olması: Kablo ve enerji alt yapısı gerektirmediğinden KAA'nın bir yerden başka bir yere kolaylıkla taşınması gerçekleştirilmektedir.

Algılayıcı düğümlerin hareketi: Algılama alanında kablosuz olarak haberleşen düğümler herhangi bir kısıtlama olmadan hareket edebilmektedirler.

Dezavantajları

KAA'nın sağladığı avantajların yanı sıra dezavantajları da bulunmaktadır. Kaynakların kısıtlı olması, düğümlerin izlenmesi ve yönetilmesi ile ilgili zorluklar, hata olasılığının yüksek olması ve servis kalitesinin istenen seviyede olmaması gibi dezavantajları bulunmaktadır.

En önemli dezavantajı kaynakların kısıtlı olmasıdır. Enerji ve bellek, düğüm için önemli bir yer teşkil etmektedir. Belleğinin kısıtlı olması sebebiyle ileri derece algoritmik işlemlerin yapılması zor olmaktadır. Enerjinin sınırlı olması sebebiyle düğümün hayatta kalma süresi de sınırlı olmaktadır.

2.7. TinyOS İşletim Sistemi ve NesC Dili

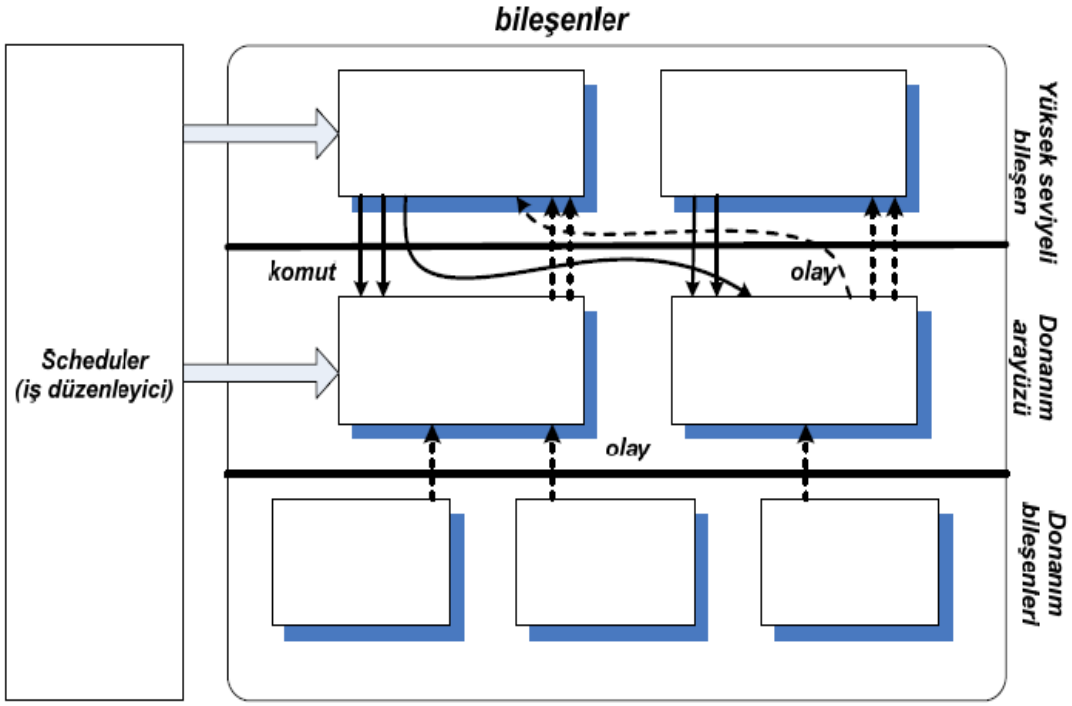
KAA'yı oluşturan algılayıcı düğümler üzerinde TinyOS [36] İşletim Sistemi yüklüdür. TinyOS, NesC [37] programlama dilinde kodlanmıştır. Bu kodlama sayesinde düğümlere yeni özellikler kazandırılabilir. Tasarlanan algoritma ya da protokoller NesC programlama dili kullanılmak suretiyle düğümlere yüklenebilir.
TinyOS işletim sistemi, kablosuz algılayıcı ağlarının gereksinimlerini destekleyecek şekilde tasarlanmıştır [38]. TinyOS, kablosuz algılayıcı ağlarında kullanılmak üzere tamamen ücretsiz ve açık kaynak kodlu olarak dağıtılan bir gömülü işletim sistemidir. California ve Berkeley Üniversitelerinin ve Intel'in iş birliği ile geliştirilmesine başlanmıştır. Daha sonraları gelişerek TinyOS Alliances adında uluslararası bir birlik kurulmuştur.

TinyOS işletim sistemi, C programlama dilinin bir varyasyonu olan NesC programlama dili ile yazılmıştır.

Bileşen tabanlı bir mimariye sahiptir. Klasik işletim sistemlerinden farklı olarak, işletim sisteminde çekirdek ve kullanıcı uygulamaları diye bir ayrım bulunmamaktadır. Bu yapı, programının tamamının analizinin ve eniyilemenin daha etkin bir şekilde yapılabilmesine olanak sağlar. İşletim sisteminin çekirdeği sadece 400 byte kod ve veri içerir. Statik bellek yönetimini kullanır. Böylece malloc/free dinamik bellek yönetimlerinden oluşabilecek bellek sızıntısı da engellenmiş olur. Sanal bellek oluşturulmaz ve her yerden ulaşılabilecek bir tek evrensel bellek kullanır.

TinyOS işletim sistemi, güç tüketimini azaltmak ve çalışma ömrünü arttırmak için "acele et ve uyu" diye bilinen bir strateji izler. Bu strateji, olabildiğince az güç harcamak için mikrodenetleyicinin mümkün olduğunca uyuması prensibine dayanır. Görev, olabilecek en kısa sürede bitirilip, işlemci en az güç harcayacağı uyku modunda bekletilir. İlaveten, uygulamaları sonsuz döngülerde bekletmek yerine kesmeler veya zamanlayıcı/sayıcı fonksiyonları kullanılır, fonksiyon çağrıları yerine makro ve "inline" gibi C programlama deyimleri kullanılarak kod eniyilemesini

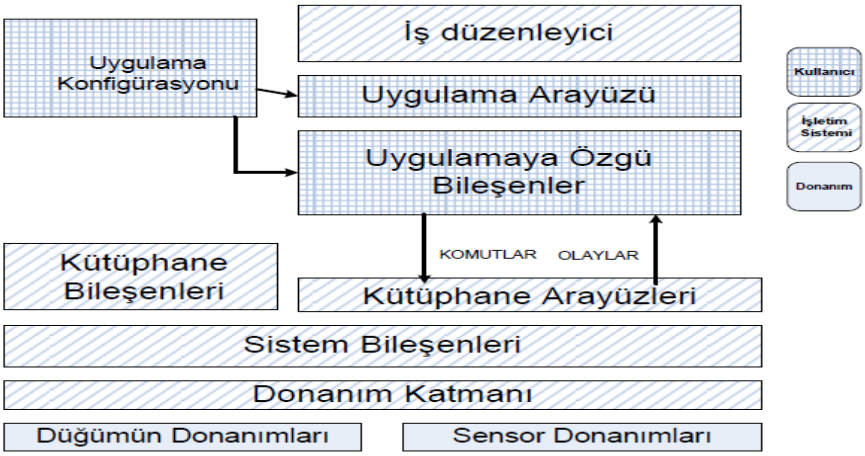
arttırır ve işlem süresini kısaltarak güç tüketimini azaltmayı hedefler. Bunların gerçekleştirilebilmesi içinse, TinyOS işletim sisteminde nesneye yönelik programlama yöntemi kullanılır. Şekil 2.5.'te [38] TinyOS işletim sisteminin temel bileşenlerini ve birbirleri ile olan bağlantılarını göstermektedir.



Şekil 2.5. TinyOS işletim sisteminin bileşenleri ve birbirleriyle olan bağlantıları

İşletim sistemi, tekrar kullanılabilen ve başka bileşenler tarafından çağırılabilen bileşenlerden ve bunların birbiriyle haberleşmesini sağlayan komut ve olaylardan oluşur. İş düzenleyici de komut ve olaylar tarafından gönderilen görevlerin hangi sırada yapılacağını kontrol eder. FIFO (ilk giren ilk çıkar) yapısında çalışır, öncelik tanımlı değildir.

Şekil 2.6.'da[38] TinyOS işletim sisteminin katman yapısı gösterilmektedir.



Şekil 2.6. TinyOS işletim sisteminin katmanları

Nesneye yönelik bir mimaride, tek bir uygulama birbirinden bağımsız görevler tarafından paylaşılarak yapılır. TinyOS işletim sisteminde, her bir sistem bileşeni, sürekli olarak kendilerine gelen olaylara cevap vererek çalışır. Bir olay meydana geldiğinde, ilgili görev çalışır ve istenilenleri yerine getirir sonra tekrar sisteme geri döner. Yüksek performanslı işlem gerektiren alanlarda çalışan araştırmacılar, olay tabanlı uygulamalarda nesneye yönelik programlama yönteminin kullanılmasının gerekliliğini daha önceki çalışmalarda görmüşlerdir [39,40].

Nesneye yönelik programlamanın bir dezavantajı, çok uzun süren hesaplamaların, oluşum zamanı önemli olan olayların gecikmesine neden olabilmesidir. Aynı şekilde, bir hatadan dolayı, bir olay hiçbir zaman bitmezse, diğer sistem fonksiyonlarının durmasına diğer bir değişle hiçbir zaman yürütülememesine neden olabilecektir. Çok uzun süren hesaplamaların yapılabilmesi için, TinyOS görev adı verilen bir yürütme mekanizmasına sahiptir. Bir görev, arka planda çalışan ve ilgili hesaplama işlerini olaylara ve kesmelere engel olmadan yapan yürütme mekanizmasıdır. Görevler, iş düzenleyici tarafından işlemcinin boş olduğu zamanlarda yerine getirilir. İlaveten; görevler, kendilerinden daha düşük seviyeli sistem uygulamaları tarafından kesilebilirler. Böylece bir olay oluştuğunda görevler beklemeye geçer ve sistemin olayları zaman geciktirmeden işlemesine imkân tanınmış olur. İş düzenleyici, FIFO mimarisinde çalışır. Görevler arasında öncelik yoktur, iş düzenleyiciye ilk verilen görev bitmeden diğer görevler işlenemez. İş düzenleyici de yürütülecek herhangi bir

görev yoksa işlemci saati haricindeki bütün uygulama ve donanımları uyku moduna sokar. Böylece gereksiz yere enerji harcanmaz. Donanım tarafından olaylar oluşturulmakta, olayların bazıları bir komut ile işini bitirirken, bazıları ise iş düzenleyiciye görevler vermektedir. Görevler yürütülürken, yeni komutlar çağırılabilir veya iş bitince sisteme geri dönülebilir.

Çok uzun süren hesaplamalar için görev yapısı tanımlandığı gibi, TinyOS, yürütülmesi sırasında birbirinden ayrılmaması gereken kodlar için de atomikleştirme adında bir mekanizma tanımlamıştır. Nesneye yönelik programlamada hatalara yol açan durumlardan biri farklı modüllerin ulaşabildiği bir bellek alanına aynı anda erişimin oluşturduğu yarışma durumu olarak adlandırılan durumlardır. İşlemin yapıldığı yer bir görevin içiyse ve görev bitene kadar daha öncelikli bir kodun bu alana erişmesi engellenmek isteniyorsa, bu bellek alanına erişilen kod parçası atomik kod olarak tanımlanır.

Atomik kodların çok uzun tutulması sistemde oluşan kesmelerin yürütülmesinin gecikmesine neden olabilir. Bu yüzden atomik olarak tanımlanan kod parçaları mümkün olduğunca kısa olmalıdır. Genel kullanım yerleri, diğer görev veya fonksiyonların erişebildikleri bellek bölgelerindeki işlemler olarak bilinir.

TinyOS işletim sisteminin nesneye yönelik olan yapısına ek olarak, TinyOS işletim sistemine özgü bir bileşen mimarisi vardır. Bileşen mimarisi, modüler ve kolayca birleştirilebilir bir şekilde geliştirilmiştir. Bileşen mimarisi, uygulama geliştirici için, birbirinden bağımsız bileşenleri kendi uygulamasına özgü bir şekilde bağlayarak, kolayca yeni bir uygulama geliştirmesine olanak sağlayacak şekildedir. Başka bir deyişle, TinyOS işletim sisteminde geliştirilen bir uygulama aslında o uygulamada kullanılan bileşenlerin listesi ve bunların birbirine bağlantılarını gösteren konfigürasyon dosyalarından oluşur.

TinyOS işletim sisteminde her bir bileşen kendisine ait komutları ve olayları içerir. Bileşenin oluşturabildiği olaylar ve yürütmek için başka bileşenlere gönderdiği

komutlar da genel olarak o bileşenin arayüzü olarak tanımlanır. Her bir bileşen aşağıdaki şekilde gösterildiği üzere dört bölümden oluşur:

• Görevler
• Komutlar
• Olaylar
• Yerel değişkenler ve yerel fonksiyonlar

TinyOS işletim sistemi bileşen içeriği

Modülerliğin sağlanabilmesi için, her bir bileşen kendisine sunulabilecek komutları ve kendisinin oluşturabileceği olayları bileşen dosyanın başında bildirir.

Bileşenler içindeki yerel değişkenlerin bellekteki yerleri statik olarak belirlenir. Böylece derleme zamanında o bileşenin ihtiyaç duyacağı bellek alanı tespit edilebilir. Böyle olması aynı zamanda yerel değişken veya fonksiyonlara başka bileşenlerin dışardan erişmesini imkânsız kılar ve hafıza korumasına ihtiyaç duyulmamasını sağlar. Bellek alanının statik olarak belirlenmesi ve dinamik bellek mimarisine izin verilmemesi, hatalı işaretçi kullanımından dolayı meydana gelebilecek hataların da önüne geçilmesini sağlar. Bir başka olumlu yönü ise, değişkenlere işaretçiler üzerinden erişim yerine değişkenler hafızadaki yerlerine statik olarak derleme zamanında yerleştirilerek yürütme zamanı kısaltılmış olur. TinyOS işletim sisteminde komutlar daha düşük katmanlardaki bileşenlere bloke edilemez olarak gönderilen mesajlardır. Tipik olarak, her komut çağırıcısına başarılı veya başarısız olduğunu belirten bir geri dönüş değeri döndürür. Komutun başarılı veya başarısız olduğunu bildiren geri dönüş değeri, o işlemin başarısını değil, komutun ilgili bileşen tarafından alınıp alınmadığını gösterir. Bunun nedeni, her bileşenin dışarıya sadece arayüzlerini göstermesi ve arayüzler aşağıdaki şekil tarafından başarılı olarak alınan bir çağrının sorunsuz olarak yürütülmesinin bileşenin kendi sorumluluğunda olmasındandır.

Olaylar, donanıma yakın modüllerin veya başka bileşenler tarafından kurulmuş sayıcı kesmeleri sonucunda oluşan ve bileşenleri veya görevleri tetikleyen mesajlardır. Olaylar, komutları çağırabilir, görevleri başlatabilir veya başka olayları tetikleyebilir. Donanım tarafından tetiklenen olaylar, en düşük seviyeli başka bir deyişle en yüksek öncelikli olaylardır. Hızlı çalışması, mümkün olduğunca kısa süre yürütülerek işlemcinin olaylarda çok fazla zaman geçirmesi engellenmelidir.

Tipik olarak, bileşenler üç kategoriye ayrılabilir. Donanım bileşenleri, donanım arayüzleri ve yüksek seviyeli bileşenler. Donanım bileşenleri, TinyOS bileşen mimarisinde donanımın yerini tutar. RFM bileşeni bu sınıftandır. Bu bileşen gelen komutlara göre, alıcı/verici modülünün giriş ve çıkışlarını ayarlar, modülün veya haberleşmenin durumu hakkında diğer bileşenler için olaylar yaratır. Yerel değişkenlerinde, alıcı/verici modülünün haberleşme hızı, aktif olan modu, gibi o anki durumu tutulur. RFM bileşeni tamamıyla kesmelerden oluşmuştur. TX veya RX bitinde oluşan kesmeleri üst katmandaki bileşenle olay oluşturarak bildirir. Yerel hafıza bölgesinde herhangi bir görev bulundurmaz.

Donanım arayüz bileşenleri, daha karmaşık donanım davranışlarının yapılmasını sağlar. Buna iyi bir örnek “kablosuz haberleşme byte” olarak adlandırılan bileşendir. Kendisine gönderilen veriyi, RFM bileşenine teker teker kaydırır ve bir bytelık veri aktarımı bitince üst katmana bir olay oluşturur. Yerel hafıza bölgesinde, basit kodlama algoritmalarını yürütür. Üst katmanlar için bu verileri kablosuz haberleşme yolu ile göndermek için bir arayüzdür. Yüksek seviyeli bileşenler, kontrol, yönlendirme ve veri transferleri gibi karmaşık uygulamaları barındırırlar. Buna bir örnek olarak mesajlaşma bileşeni gösterilebilir.

TinyOS işletim sistemine, C programlama dilinin bir varyasyonu olan NesC programlama dili ile program kodu yazılır. Kablosuz algılayıcı ağlarının düşük hafıza gereksinimi göz önüne alınarak C programlama diline olay tabanlı çalışabilme ve nesneye yönelik programlama özelliklerini ekler. NesC programlama dili, NesC önderleyicisi ile standart C koduna dönüştürülür ve açık kaynak kodlu C derleyicisi

(gnu-gcc) ile ikili düzende PXA271 mikrodenetleyicisine yüklenmeye hazır dosya haline getirilir.

TinyOS işletim sisteminin statik bellek yönetimi sayesinde derleme zamanında programın tam çalışma şekli bilinir. Bu nedenle derleme zamanında program kodunda optimizasyon yapılır. Olası birden fazla bileşenin aynı bellek bölgesine aynı anda yazması şeklinde adlandırılan yarışma durumlarını tespit edebilir. NesC programlama dilinde iki tip dosya bulunur: Modüller ve Konfigürasyonlar.

Modüller, modüllerin dışarıya verecekleri servisleri ve dışardan alacakları servisleri içeren arayüz tanımlarını içerirler. Dışarıya servis veren veya alan bir modül, aldığı veya verdiği veriyi kendi içinde de işleyebilir. Bunun için gerekli olan kod yazımı da modüller içerisinde yazılır ve ANSI-C yazım şekli kullanılır.

Konfigürasyonlar ise, bileşenlerin birbirleri ile olan ilişkilerini içerir. Bağlama olarak adlandırılan yazım şekli ile bir bileşen başka bileşenleri birbirine bağlayarak bir görevin yerine getirilmesini sağlayabilir.

Bir bileşen hem modül hem konfigürasyon dosyalarını içerebilir. Bu, o bileşenin hem başka bileşenleri kullandığı hem de kendi içinde C kodu bulundurduğu anlamına gelir.

Modüller ve konfigürasyonlar

NesC programlama dilinde arayüzler, bir bileşenin başka bileşenler ile olan ilişkilerini düzenler. Bu ilişkiler hizmet verme veya hizmet alma şeklinde olabilir.

Bileşenler arası bağlantılar

Standartlaştırabilmek amacıyla, her bir modül, daha önceden belirlenmiş kontrol arayüzlerini içerir:

• init();

• start();

• stop();

Aynı şekilde tanımlanmış görevlerde iki aşamada ele alınır:

• send();

• sendDone();

NesC programlama dilinde modüller öncelikle bileşenin sunduğu hizmetlerin ve alacağı hizmetlerin sırasıyla “provides” ve “uses” anahtar kelimeleriyle listelendiği “module” fonksiyon tanımı ile başlarlar. Sonrasında ise, “implementation” fonksiyon tanımı ile modül içinde bulunan C kodunun yazılımı gelir. Bu kısım da yerel değişkenlerin tanımlanması ve fonksiyonların tanımlanması şeklinde iki kısımda incelenebilir.

Konfigürasyon dosyaları, “configuration” fonksiyonu ile başlarlar. Burada kullanılacak ve verilecek olan hizmetler belirtilir. Ardından “implementation” fonksiyonu ile hangi bileşen ve arayüzlerin birbirlerine bağlanacağı belirtilir. Burada herhangi bir C kodu yazılmaz. Bağlama amacıyla kullanılan “−>, <−, =” ifadeleri bulunur.

Bağlanacak olan bileşenlerin birbirleriyle uyumlu olması gerekliliği vardır. Arayüz – arayüz, komut – komut ve olay – olay arasında bağlantılara izin vardır. Örnek olarak, “send” arayüzü, “receive” arayüzüne bağlanamaz. Ancak “send – send” arayüzü birbirine bağlanabilir.

Bir bileşen, başka bir bileşenden veya donanımdan bir olay çağrısı aldığında, bunu hemen koşmaya başlar. Aynı şekilde, bir bileşen başka bir bileşene “command” çağrısı yaptığında, ilgili bileşen kendi C kodundaki ilgili “command” fonksiyonunu işletir. Komutlar, daha düşük seviyedeki bileşenlere yapılırken, olaylar, komutlara veya donanımsal kesmelere cevaben daha yukarı seviyelere yapılır.

2.8. KAA ve Şifreleme

Şifreleme, haberleşen iki veya daha fazla tarafın bilgi alışverişini emniyetli olarak yapmasını sağlayan, temeli matematiksel zor problemlere dayanan tekniklerin ve uygulamaların bütünüdür [41].

Haberleşen iki tarafın güvenlikle ilgili çeşitli beklentileri vardır. Bu beklentiler haberleşmenin emniyet öğeleri olarak sınıflandırılmıştır. Haberleşmede emniyet öğeleri aşağıda verilmektedir [41]:

Gizlilik: Taşınan bilginin içeriğinin gizli kalmasıdır.
Bütünlük: Taşınan bilginin içeriğinin yolda değiştirilememesidir.
Kimlik Doğrulama: Bilgiyi gönderen kişinin kimliğinin doğruluğundan emin olmaktır.
İnkâr Edememezlik: Bilgiyi gönderen veya işleyen kişinin yaptığı işi sonradan inkar edememesidir.
Haberleşmenin Sürekliliği: Haberleşmenin kesintiye uğramadan yapılmasıdır.

Şifreleme, bir bilginin özel bir yöntemle değiştirilerek farklı bir şekle sokulması olarak tanımlanabilir. Şifreleme işlemi sonucunda ortaya çıkan yeni biçimdeki bilgi, şifre çözme işlemine tabi tutularak ilk haline dönüştürülebilir.

Şifreleme yönteminde aranan bir takım özellikler vardır. Bunlar aşağıda listelenmiştir[41]:

- Şifreleme ve şifre çözme işleminin zorluğu ihtiyaç duyulan güvenlikle doğru orantılı olmalıdır. Çok önemli olmayan bir bilginin şifrelenmesi için bilginin kendisinden daha fazla işgücü ve zaman harcanması verimli olmayacaktır.

- Anahtar seçimi ve şifreleme algoritması özel koşullara bağlı olmamalıdır. Şifreleme yöntemi her türlü bilgi için aynı şekilde çalışmalıdır.

- Sürecin gerçeklenmesi mümkün olduğunca basit olmalıdır. Çok karışık bir sistemin gerçeklenmesi hem hatalara sebep olabilir hem de performans açısından tatmin edici olmayabilir.

- Şifrelemede yapılan hatalar sonraki adımlara yansımamalı ve mesajın tamamını bozmamalıdır. Saldırılara karşı bu özellik koruyucu olacaktır. Ayrıca haberleşme hattında meydana gelen bir hata bütün mesajın bozulmasına neden olmayacağı için bu özellik tercih edilmektedir.

- Kullanılan algoritmanın karıştırma özelliği olmalıdır. Mesajın şifrelenmiş hali ile açık hali arasında ilişki kurulması çok zor olmalıdır.

- Kullanılan algoritmanın dağıtma özelliği olmalıdır. Mesajın açık hali şifreli hale gelirken içerdiği kelime ve harf grupları şifreli mesajın içinde olabildiğince dağıtılmalıdır.

Güvenli şifreleme yöntemleri klasik şifreleme yöntemlerinin zayıf yönlerini ortadan kaldıran ve kriptoanalize karşı dirençli olan algoritmalarla gerçeklenir. Bu yöntemler elektronik sistemlerde (bilgisayar, telekomünikasyon vb) kullanılır ve ikili düzende (binary) saklanan ve taşınan bilgi üzerinde uygulanır. Bu nedenle anahtar olarak bit dizileri kullanılır.

Bir şifreleme algoritmasının güvenliğini belirleyen en önemli değişkenlerden birisi anahtar uzunluğudur. Şifrelemede bu anahtarlardan herhangi birisi kullanılabileceği için bu anahtarı tahmin yoluyla elde etme olasılığı çok düşüktür.
64 bitlik Anahtar = 1100101010110001 0001101000000111 0110100010011110 1100111010011011

Güvenli şifreleme temel olarak iki çeşittir [41]: Simetrik Kriptografi, Asimetrik Kriptografi

2.8.1. Simetrik kriptografi

Simetrik kriptografi, şifreleme ve şifre açma işlemi aynı anahtar ile yapılır. Simetrik kriptografide bu anahtar gizli tutulmalıdır. Bu nedenle, bu tip sistemlere gizli anahtarlı şifreleme sistemi adı da verilmektedir [41].

Bu sistemde haberleşen taraflar:

Aynı şifreleme algoritmasını kullanırlar
Birbirine uyumlu gerçeklemeler kullanırlar
Aynı anahtarı kullanırlar

Simetrik kriptografide Artılar Eksiler

Güçlü yönleri aşağıdaki gibi özetlenebilir:

- Algoritmalar hızlıdır
- Algoritmaların donanımla gerçeklenmesi kolaydır
- "Gizlilik" güvenlik hizmetini yerine getirir

Zayıf yönleri aşağıdaki gibidir:

- Ölçeklenebilir değil
- Emniyetli anahtar dağıtımı zor
- "Bütünlük" ve "Kimlik Doğrulama" güvenlik hizmetlerini gerçeklemek zor

Simetrik kriptografi algoritmaları başlıca iki sınıfta ele alınabilir:

<u>Blok şifreleme algoritmaları</u>

Bu tip algoritmalar şifrelenecek veriyi sabit uzunlukta bloklar olarak şifreleme fonksiyonuna alırlar ve aynı uzunlukta şifrelenmiş veri blokları üretirler. Bu

algoritmalara örnek olarak AES[57], DES[58], Skipjack[13], RC5[59] vb. verilebilir. Bu algoritmalar aşağıdaki özellikleri gerçeklemeye çalışırlar:

Karıştırma: Anahtar ve şifrelenmiş mesaj arasındaki ilişki olabildiğince karışık olmalıdır.
Dağıtma: Tek bir açık mesaj karakterinin etkisi olabildiğince fazla şifrelenmiş karaktere yansıtılmalıdır.
Transpoze İşlemi: Şifrelemeye başlamadan önce açık mesajın içeriği değişik bir sıraya konur.
Yer Değiştirme İşlemi: Tekrar eden kalıplar başka kalıplarla değiştirilir.

Bit katarı (dizi) şifreleme algoritmaları

Bu tip algoritmalar veriyi akan bir bit dizisi olarak alırlar. Vernam tipindeki bu algoritmalarda rastgele bit dizisi üretiminin kendini tekrarlamayan bir yapıda olması gereklidir. Örnek algoritmalar RC2, RC4 vb.

2.8.2. Asimetrik kriptografi

Asimetrik kriptografi, şifreleme ve şifre çözme işlemi farklı anahtarlar ile yapılır. Bu anahtar çiftini oluşturan anahtarlara açık ve özel anahtar adı verilir. Bu kriptografi yönteminde özel anahtar gizli tutulmalıdır fakat açık anahtar gerekli kişilere verilebilir ve başka kişilerle paylaşılabilir. Bu özelliğinden dolayı asimetrik kriptografi, açık anahtarlı şifreleme adıyla da anılır [41].
Bu sistemi kullanarak haberleşen taraflar:

Aynı şifreleme algoritmasını kullanırlar
Birbiriyle uyumlu gerçeklemeler kullanırlar
Gerekli anahtarlara erişebilirler

Şifreleme sistemlerinin karşılaştırması

Asimetrik ve simetrik şifreleme sistemlerinin özellikleri Çizelge 2.1.'de verilmektedir.

Çizelge 2.1. Asimetrik ve simetrik şifreleme sistemlerinin karşılaştırılması

Konu	**Simetrik Kriptografi**	**Asimetrik Kriptografi**
Gizlilik	+	+
Bütünlük	-	+
Kimlik doğrulama	-	+
İnkâr Edememezlik	-	+
Performans	Hızlı	Yavaş
Güvenlik	Anahtar uzunluğuna bağlı	Anahtar uzunluğuna bağlı

Gizli anahtar şifrelemede hem alıcı hem verici şifreleme işlemleri için aynı anahtarı kullanırlar. Açık anahtar şifrelemede alıcı ve verici farklı anahtarları kullanır. Açık anahtar şifreleme gizli anahtar sisteminden daha güçlüdür ve daha iyi güvenlik ve mesaj gizliliği sağlar. Ancak bu sistemin en büyük dezavantajı hızdır. Açık anahtar sistemi karmaşıktır ve bazı durumlarda pratik olmayabilir.

Şifreleme algoritmalarının karşılaştırılması

40 bitlik bir anahtar için $n=2^{40}$ veya n=1 099 511 627 776 (bir trilyon doksan dokuz milyar beş yüz on bir milyon altı yüz yirmi yedi bin yedi yüz yetmiş altı) olası anahtar söz konusudur. 1995'de yapılan bir yarışmada RC4 algoritması ile 40 bitlik bir anahtarla şifrelenmiş internet üzerinden yapılan bir kredi kartı işlemi, elinde sadece mütevazi bir bilgisayar laboratuvarı olan bir öğrenci tarafından 3 buçuk saatte çözülmüştür [11].

Anahtarın deneme-yanılma yöntemiyle bulunmasını engellemek için, bugünkü süper bilgisayarlardan milyonlarca kat daha hızlı çalışan bir bilgisayarla bile milyarlarca

yıl sürmesi için, kullanılan anahtarların uzunluğunun mümkün olduğunca büyük olması gerekmektedir.

Çizelge 2.2'de farklı anahtar boyları için, saniyede bir milyon, bir milyar ve bir trilyon şifre deneyebilen bilgisayarlar için anahtar çözme süreleri verilmiştir [11].

Çizelge 2.2. Farklı anahtar boyutları için anahtar çözme süreleri

Anahtar Uzunluğu (n)	**Olası değer sayısı (2^n)**	**10^6 şifre/s hızında ortalama çözme süresi**	**10^9 şifre/s hızında ortalama çözme süresi**	**10^{12} şifre/s hızında ortalama çözme süresi**
32 bit	~$4x10^9$	36 dak	2.16 s	2.16 ms
40 bit	~10^{12}	6 gün	9 dak	1 s
56 bit	~$7.2x10^{16}$	1142 yıl	1 yıl 2 ay	10 saat
64 bit	$1.8x10^{19}$	292 000 yıl	292 yıl	3.5 ay
128 bit	$1.7x10^{38}$	$5.4x10^{24}$ yıl	$5.4x10^{21}$ yıl	$5.4x10^{18}$ yıl

Bir şifreleme algoritmasının performansı şu kriterlere göre belirlenebilir:

- Sistemin kırılabilme süresinin uzunluğu,
- Şifreleme ve çözme işlemlerine harcanan süre,
- Şifreleme ve çözme işleminde ihtiyaç duyulan bellek miktarı,
- Algoritmanın kurulacak sisteme uygunluğu.

Blok şifreler [42], Shannon'un önerdiği karıştırma (confusion) ve yayılma (diffusion) tekniklerine dayanır. Karıştırma şifreli metin ve açık metin arasındaki ilişkiyi gizlemeyi amaçlarken, yayılma açık metindeki izlerin şifreli metinde sezilmemesini sağlamak için kullanılır.

Anahtar

Blok şifreleme algoritmalarında anahtarın uzunluğu yada bit sayısı en temel saldırı olan geniş anahtar arama saldırısına karşın güçlü olmalıdır. Örneğin DES [43] algoritması 56-bit anahtar kullanırken AES [44], algoritması DES'in bu zaafını örter niteliktedir ve 128, 192, 256 bit anahtar seçenekleri mevcuttur. Ayrıca anahtarın rastlantısal olması gerekmektedir.

Döngü sayısı

Blok şifreleme algoritmalarında döngü sayısı iyi seçilmek zorundadır. Çünkü lineer transformasyon ve yer değiştirmelerin bu seçilen değerle algoritmaya yeterli gücü vermesi gerekmektedir. Ayrıca yapılan saldırıların başarısız olması için en önemli şartlardan biridir.

Düğümlerdeki güç, enerji, hesaplama ve iletişim sınırlamalarından dolayı asimetrik kriptosistemler, KAA'da güvenli iletişim için kullanılamamaktadır. Her bir protokol, farklı bir algoritmayı seçmiştir. Algoritma seçilirken, güvenli olması, az bellek kaplaması gibi şartlar gereklidir. Berkeley Üniversitesi tarafından geliştirilen SPINS [45], RC5 [46] algoritmasını kullanmaktadır.

KAA için kullanılan güvenlik protokollerinden TinySec [16], SenSec[47] ve MiniSec [17]'in Skipjack şifreleme algoritmasını (80-bit anahtar boyutu) kullanılmaktadır. Yazarlar [48] Corrected Block Tiny Encryption Algorithm (XXTEA)[49] (128-bit anahtar boyutlu)' nın Skipjack'e göre daha iyi bir alternatif olacağını benzetim ve analiz sonuçları ile göstermişlerdir. Bu çalışma TinyOS üzerinde NesC dili kullanılarak yapılmıştır. Ayrıca TOSSIM [50] ve AVRORA [51] simülatörlerinde de sonuçlar alınmıştır.

Veri bağı güvenlik protokolleri, güvenlik özelliklerinden olan veri gizliliği, veri bütünlüğü ve veri doğruluğunu kanıtlama ve tekrar yayımlama ataklarına karşı veri tazeliğini sağlamak zorundadırlar.

Önerilen çoğu veri bağı güvenlik mimarilerinin 80 bit anahtar boyutlu Skipjack şifreleme algoritmasını kullanmasına rağmen, güvenliğin sağlanması için 128 bit anahtar boyutunu kullanmak daha olumlu olacaktır. Örneğin ileri şifreleme standardı olarak kabul edilen AES 128 bit anahtar boyutludur.

KAA için gereken güvenlik çözümleri aşağıda verilmektedir [52].

a) Veri Gizliliği: Simetrik anahtar blok şifreleme modu kullanılarak gizlilik sağlanabilir. 1977'de tanıtılan DES (53 bit anahtar boyutlu) 1982'de kırılmıştır. Bundan yola çıkarak yapılan araştırmalar bir şifreleme algoritmasının en az 128 bit anahtar boyutlu olması gerektiğine karar kılmıştır.

b) Veri Bütünlüğü: Tipik olarak, veri bütünlüğünü garanti etmek için, unkeyed hash fonksiyonları kullanılır. Veri bütünlüğü ile birlikte diğer gerekli bir özellik olan veri doğruluğunu kanıtlamak için Mesaj doğrulama Kodu Message Authentication Code (MAC) kullanılmaktadır. KAA için MAC değerinin bit sayısı genellikle 4-8 byte'dır.

c) Tekrar Yayınlama Koruması: Saldırgan düğüm, aynı veri paketini alıcıya defalarca göndererek, düğümün kaynaklarını tüketebilir. Bu atağa engel olmak için birbirini izleyen sıra numaraları kullanılır. Herhangi gönderilen bir veri paketinin sıra numarası, en yüksek sıra numarasından küçük olursa bu tekrarlanan paket olarak işlev görür ve atılır.

d) Veri Tazeliği: Alıcının aldığı veri paketinin, önceki bir paket olmadığını gösterir. Tazelik parametresi kaynaktan verinin iletimi ve onun alıcıya teslim etmesi arasındaki farkı ele geçirir. Tekrar yayınlama koruması, veri tazeliğini garanti etmenin özel bir durumudur.

e)Kullanılırlık: DoS ataklarına karşı kullanılan protokolün güçlü olmasını gerektirir.

f) Düşük Ek yük: KAA ortamı kısıtlı olduğundan dolayı gerçekleştirilen güvenlik çözümlerinde bu olayın dikkate alınması gerekir.

Skipjack[53] 80 bit anahtar boyutu, 64 bit blok boyutu ve 32 döngüye sahiptir.

TEA 128 bit anahtar boyutu, 64 bit blok boyutu ve 64 döngüye sahiptir. Bununla birlikte, çeşitli kriptografik ataklar TEA için sunulmuştur.

TEA'nın bu sınırlamalarının üstesinden gelmek için Needham tarafından XTEA önerilmiştir. Bununla birlikte, XTEA'da bazı açıklar olduğu öngörülmüştür. Ardından bu eksiklikleri gidermekle birlikte daha da güçlenen XXTEA, Wheeler tarafından önerilmiştir. XXTEA, 128 bit anahtar boyutludur. Şu an için bilinen kriptaanaliz zayıflığı bulunmamaktadır. Tabiki XXTEA sadece 128 bit anahtar boyutlu olduğu için değil, dahası basit ve minumum olduğundan da önerilmektedir.

Soren Rinne[54] TEA, XTEA, SEA, AES, HIGHT ve DES algoritmalarını karşılaştırmıştır. Yazarlar, eğer bellek kritik bir kıstas ise TEA veya XTEA nın iyi bir seçim olacağını dile getirmişlerdir.

Liu [55], TEA algoritmasının Berkeley düğüm platformlarına uygulanmasını tartışmıştır. Yazarlar, zamansal olarak TEA'nın performansının hızlı olduğunu göstermişlerdir.

Großshädl Johann [56], blok şifreleme algoritmalarını enerji bakımından değerlendirmişlerdir. Yazarlar, makalede performans, enerji tüketimi, çalışma zamanı bellek gereksinimleri üzerinde durmuşlardır. Yazarlar, RC6, AES, Serpent, Twofish and XTEA algoritmalarını karşılaştırmıştır. XTEA en iyi sonuçları vermiştir.

Diğer bir çalışmada [57] şifreleme, şifre çözme ve anahtar düzenleme işlemleri süresince harcanan işlemci çevrimi açısından karşılaştırma yapılmaktadır. Sadece şifreleme ve şifre çözme ele alındığında Skipjack algoritması, TEA ailesine göre daha iyi sonuçlar vermektedir. Bununla birlikte, Anahtar belirleme için Skipjack fazla miktarda işlemci çevirimine gereksinim duyar.

Başka bir çalışmada ise [58] Enerji tüketimi açısından Skipjack ve TEA ailesi karşılaştırılmıştır. Skipjack'e göre XXTEA 6.27%, XXTEAO 5.03% daha fazla enerji harcamaktadır. KAA düğümlerinde enerji kritik bir kaynaktır. Fakat XXTEA'nın anahtar boyutu 128 bit, Skipjack'in ise 80 bit'tir.

Tüm deneysel sonuçlardan elde edilen analizlere göre XXTEA, KAA için en uygun şifreleme yöntemidir. Şifreleme ve şifre çözme işlemlerinde Skipjack şifreleme yöntemi daha hızlı ve enerji etkin olmasına rağmen, Skipjack'te anahtar yayılımda çok yüksek enerji tüketir. Böylece, tüm enerji gereksinimleri Skipjack'te daha yüksektir.

Diğer bir çalışmada[59] AES, XXTEA ve Skipjack algoritması Cipher Block Chaining (CBC) ve Output Codebook Block (OCB) modları kullanılarak ayrı ayrı değerlendirilmiştir. Çalışma sonucunda KAA'da Veri Bağı Katmanı Güvenlik Mimarisi için XXTEA'nın optimum algoritma ve OCB'nin de optimum mod olduğu ortaya çıkmıştır. Bu yüzden geliştirilen protokolde şifreleme algoritması olarak XXTEA, şifreleme modu olarak da OCB seçilmiştir.

3. MEVCUT GÜVENLİK PROTOKOLLERİ

Kablosuz Algılayıcı Ağlarda önerilen çoğu MAC protokolü enerji verimliliği veya gecikme duyarlı uygulamalar için tasarlanmıştır. Araştırılan protokollerden IEEE 802.11[60], S-MAC[61], T-MAC[62], P-MAC[63], PAMAS[64], TRAMA[65], ALOHA with Preamble Sampling[66], WiseMAC[67], B-MAC[68], X-MAC[69], Z-MAC[70]'in birincil amacı enerji verimliliği iken, DSMAC[71] ve Optimized MAC[72] protokollerinin birincil amacı gecikmeyi azaltmaktır. Aşağıdaki bölümde ise KAA için önerilen güvenlik protokolleri sunulmuştur.

3.1. TinySec

Berkeley Üniversitesi tarafından geliştirilen TinySec[16], TinyOS sürümü içerisine dahil edilmiş bir bağlantı katmanı (link layer) güvenlik mimarisidir. Tasarımda kullanım kolaylığı ve algılayıcı ağına en az ek yük getirmesi esas olarak alınmıştır. Klasik bilgisayar ağlarında mesaj doğrulaması, bütünlüğü ve gizliliği genellikle sondan-sona (end-to-end) güvenlik mekanizmalarıyla gerçekleştirilmektedir. Aradaki geçitler mesajın içeriği ile ilgilenmemekte, sadece mesajın başlığına bakarak yönlendirme yapmaktadırlar. Algılayıcı ağlarında ise en az güç tüketimi ve band genişliğinin optimum kullanımı için, kümeleme ve aynı mesajların elenmesi gibi "ağ içi işleme (in-network processing)" yapılmaktadır. Bunun başarılabilmesi için arada yönlendirme işlemi yapan düğümlerin mesaj içeriğine ulaşmaları, değişiklik yapmaları ve belki de düşürmeleri gerekmektedir. Bu nedenle, TinySec geliştirilirken, klasik ağlardaki sondan-sona güvenlik mekanizmalarının duyarga ağları için iyi bir çözüm olamayacağı, bu işlemin bağlantı katmanında yapılması gerektiği kabul edilmiştir. Ayrıca sorunu bu katmanda çözmenin, yetkisiz mesajların baz istasyonuna ulaşmadan, henüz ağa ilk girişinde yakalanmasına ve dolayısıyla DoS saldırılarına karşı duyarga ağının daha güvenli olmasına katkıda bulunacağı düşünülmüştür. TinySec iki farklı güvenlik seçeneğini desteklemektedir:

• Kimlik doğrulamalı (authentication) şifreleme,

• Sadece kimlik doğrulama.

Kimlik doğrulamalı şifrelemede veri şifrelenir ve pakete bir kimlik doğrulama kodu (MAC) eklenir. Sadece kimlik doğrulamada ise veri şifrelenmez, sadece paket

doğrulaması yine bir MAC ile gerçekleştirilir. Bundan da anlaşılacağı üzere TinySec'de kimlik doğrulama her paket için bir zorunluluk, verinin şifrelenmesi ise uygulamaya göre karar verilebilecek bir seçenektir. Mesajların şifrelenmesinde Skipjack blok şifrelemesi, 8 baytlık bir başlangıç vektörü (IV) ve şifre bloğu zincirlemesi (CBC) ile kullanılmaktadır. Anahtarlama yöntemi için herhangi bir sınır getirilmemiş olup, uygulamada arzu edilen güvenlik seviyesine göre tüm ağ için tek bir anahtar çifti (biri verinin şifrelenmesi, diğeri ise MAC'ların hesaplanması için) seçilmektedir. TinySec kimlik doğrulamalı şifrelemenin kullanıldığı en sıkı güvenlik seviyesinde enerji, gecikme ve band genişliğine %10 ek yük getirmektedir. Sadece doğrulamanın kullanıldığı durumlarda ise bu oran %3'e düşmektedir.

3.2. Spins

Berkeley Üniversitesi tarafından geliştirilen SPINS[73], kimlik doğrulamalı yayımda kullanılan µTESLA (Micro Version of Timed, Efficient, Streaming, Loss-tolerant Authentication Protocol) protokolü, gizliliği, iki düğüm arası kimlik doğrulamayı ve verinin tazeliğini (data freshness) sağlayan SNEP (Secure Network Encryption Protocol) protokolü yapı taşlarından ve bunlar üzerine oturtulmuş bir yönlendirme protokolünden meydana gelmektedir. SNEP aşağıdaki imkanları sunmaktadır:

• Semantik güvenlik: Ağı dinleyen bir saldırganın, aynı düz metnin birden fazla şifreli kopyasını aldığında dahi bunlardan düz metin hakkında herhangi bir bilgi edinememesi anlamın gelen semantik güvenlik, alıcı ve gönderen arasında paylaşılan ve her mesaj alış-verişinde artırılan bir sayaç sayesinde gerçekleştirilmektedir.

• Kimlik doğrulaması: Alıcı düğüm gönderinin kimliğini kullanılan MAC ile kanıtlamaktadır.

• Tekrar koruması: MAC içerisindeki sayaç eski mesajların tekrar gönderilmesine karşı koruma sağlamaktadır.

• Zayıf tazelik: Semantik güvenlik maksadıyla alıcı ve gönderen arasında kullanılan sayaç, alınan mesajın bir önceki mesajdan sonra gönderildiğini garantilemektedir.

• Düşük haberleşme ek yükü: Sayacın alıcı ve gönderen üzerinde tutulması, mesaj içerisine konulmaması haberleşme ek yükünü azaltmaktadır.

Klasik yaklaşımlarda kimlik doğrulaması asimetrik yöntemlerle yapılmaktadır. Ancak algılayıcıların donanımsal kısıtlamaları, oldukça pahalı olan asimetrik yöntemler için son derece yetersizdir. µTESLA kimlik doğrulamasına asimetriklik mantığını simetrik yöntemlerle kazandırmaktadır. Gönderen, yayınlanacak mesaj paketleri için sadece kendisi tarafından bilinen bir anahtar ve tek yönlü bir fonksiyon kullanarak bir MAC oluşturur. Mesaja ait anahtarı mesajın yayımından belli bir süre sonra yayınlar. Böylece paketin içeriğinin değiştirilebilmesi ihtimali ortadan kaldırılmış olur. Alıcı tarafında, bu anahtar kullanılarak bir arabellekte tutulan paketin doğruluğu kontrol edilir. Şifreleme işleminde RC5 kullanılmaktadır. Tüm bu kimlik kanıtlama işlemi için µTESLA alıcı ve gönderen arasında gevşek de olsa bir eş zamanlamaya ihtiyaç duymaktadır.

3.3. Lisp

Sınırlı kaynaklı algılayıcı düğümlerden oluşan, büyük ölçekli kablosuz ağlarda güvenlik çözümlerini amaçlar. Çok sayıda algılayıcı düğümden oluşan ağları ölçeklemek için kümelere ayırmaktadır. Her küme için küme başı seçer ve anahtar sunucu oluşturur. LISP (Lightweight Security Protocol)[74] güvenlik protokolü yeni bir anahtarlama mekanizmasını içermektedir. Anahtarlama yöntemini, küme başlarını ve anahtar sunucuları kullanarak uygulamaktadır. Bu yöntemin çeşitli avantajları şunlardır:

- ACK'ların gönderilmesini gerektirmeyen etkin anahtar yayını kullanır,
- Veri mesajına eklenmeksizin oluşturulan doğruluk bitlerini kullanır,
- Kaybolan anahtarları kurtarabilir,
- Veri şifreleme/çözme olmaksızın anahtar yeniler, LISP, saldırılara karşı kritik bilgileri korumak için sağladığı faydalar ise şu şekilde özetlenebilir:
- Veri bütünlüğü, gönderilen verilerin değiştirilmesini engeller.
- Erişim kontrolü, ağa girişlerin kontrol edilmesiyle sağlanır.

- Anahtar yenileme, ağı tehlikeye atacak düğümlere karşı koruma sağlar.

LISP protokolü, güvenlikle beraber diğer servisleri de (yönlendirme, veri dağıtımı, konum) birleştirebilmektedir. LISP esnek ve enerji duyarlı bir protokoldür. Ayrıca ACK ve diğer kontrol paketlerine gerek duymadığından DoS [75] saldırılarına karşı oldukça güçlüdür.

3.4. IEEE 802.15.4

IEEE 802.15.4 [18,76] Kablosuz Özel Alan Ağları (Wireless Private Area Networks, WPANs) için Ortam Erişimi ve Fiziksel katmanlarını belirler. Bu protokol KAA için geliştirilmemiş olmasına rağmen, düşük güç tüketimi, düşük maliyet ve esnek oluşundan dolayı KAA'da kullanılmaktadır. Halihazırda, Crossbow[77] firması tarafından üretilen Micaz, TelosB düğümleri üzerinde bu protokol çalışmaktadır.
ZigBee güçlü şifreleme AES-128 kullanılmaktadır.
Zigbee yenilik (freshness) sağlamaktadır.

- Yeniliği kontrol etmek tekrarlama saldırısından önler.
- Sayaç yeni anahtar oluştuğunda reset edilir.

Zigbee bütünlüğü(intergrity) sağlamaktadır.

- Saldırganın mesajı değiştirmesini önler.
- Bütünluk seçeneği 0,32,64,128 bit
- Default 64 bit

Zigbee doğrulamayı sağlamaktadır.

- Doğrulama, doğru kişiye erişilip erişilmediğini sınar.
- Saldırganın aygıtları başka aygıtmış gibi göstermesini önler.
- Doğrulama ağ düzeyinde ve aygıt seviyesinde mümkündür.
- Ağ seviyesindeki doğrulama ortak ağ anahtarı kullanılarak sağlar.

- Aygıt seviyesindeki doğrulama aygıtlar arasındaki tek Bağlantı Anahtarını kullanarak sağlar.

Zigbee şifrelemeyi sağlamaktadır.

- Saldırganın araya girip dinlemesini önler.
- Zigbee 128 bit AES encryption kullanılmaktadır.

Şifreleme koruması ağ seviyesinde ve aygıt seviyesinde sağlanmaktadır.

- Ağ seviyesindeki şifrelemede ortak ağ anahtarı kullanılır. Bu çok az bellek kullanarak saldırgandan önler.
- Aygıt seviyesindeki şifreleme ortak bağlantı anahtarı kullanır.

Zigbee üç çeşit anahtar kullanmaktadır.

- Master Anahtarı(Master key)
  İki aygıt arasındakı uzun süreli güvenliği sağlar.
- Bağlantı anahtarı(Link Key)
  İki aygıt arasındakı güvenliği sağlar.
- Ağ anahtarı(Network Key)
  Ağdaki güvenliği sağlar.

3.5. Lsec

Lsec[78], basit güvenli anahtar değiş tokuş düzeni ile kimlik doğrulama ve yetkilendirme sağlar. Bundan başka, veri gizliliği ve ihlal yada kuraldışı olaylara karşı koruma mekanizması vardır.

Algılayıcı ağlarda çeşitli güvenlik atakları mevcuttur. Bunlara örnek olarak, DoS, gizlice dinleme, tekrarlama atakları, mesaj değiştirilmesi, kötücül düğümler sayılabilir. Bu ataklara karşı koymak için LSec'te veri gizliliği, kimlik doğrulama, veri bütünlüğü, davetsiz misafirlere karşı savunma ve bazı güvenlik mekanizmaları

kullanmıştır. Algılayıcı düğümler arasındaki iletişim şifrelendiğinde kısmen bu problemler çözülebilir fakat tamamen çözülebilmesi için güçlü anahtar değiş-tokuş ve dağıtım düzeni gerekmektedir.

LSec'in sağladıkları şunlardır:

- Kimlik doğrulama ve yetkilendirme
- Basit güvenli anahtar değiş tokuş düzeni
- Kural dışı ve ihlallere karşı savunma mekanizması
- Veri gizliliği
- Asimetrik ve simetrik şifrelemelerinin birlikte kullanımı.

LSec protokolünün, Sensor Network Simulator and Emulator (SENSE)[79] üzerinde benzetimi yapılmıştır. Uygulaması bulunmamaktadır.

3.6. Lisa

LISA[80] aşağıda sayılan güvenlik çözümlerini içermektedir.
-Semantik Güvenlik: Her bir veriden sonra sayaç değeri artırılarak, aynı veri farklı şekilde şifrelenmektedir.
- Kimlik Doğrulama: Verinin doğru düğümden geldiğini garanti etmektedir.
-Tekrarlama Ataklarına karşı koruma: Eski mesajların tekrarlanmasını engellemektedir.
- Zayıf tazelik: Baz istasyonu, üretilen mesajın önceki mesajdan sonra olduğunu doğrulamaktadır.

3.7. MiniSec

MiniSec[17] Telos[20] platformu üzerinde uygulanmıştır. TinySec, düşük enerji tüketiminde düşük güvenlik sağlarken, Zigbee[81] yüksek enerji tüketiminde yüksek güvenlik sağlamaktadır. Yazarlara göre MiniSec düşük enerji tüketiminde yüksek güvenlik sağlamaktadır. Bunu başarmak için 3 teknik kullanılmıştır. Birincisi, gizlilik ve doğrulamayı sağlamak için blok şifreleme modu kullanılmıştır. Fakat;

sadece veri üzerinde bir tane geçiş olmuştur. İkincisi, IV az miktarda bit olarak kullanılmıştır. Üçüncüsü, unicast ve broadcast iletişim arasında esas ayrımlardan faydalanılmıştır. Unicast modda, extra hesaplama yerine getirerek ve eşzamanlı sayaçlar kullanarak radyonun enerji tüketimini azaltmışlardır. Broadcast modda, bloom filter mekanizması kullanılmıştır. Şifreleme algoritması olarak Skipjack, şifreleme modu olarakta OCB kullanılmıştır. DoS saldırılarına karşı savunmasızdır.

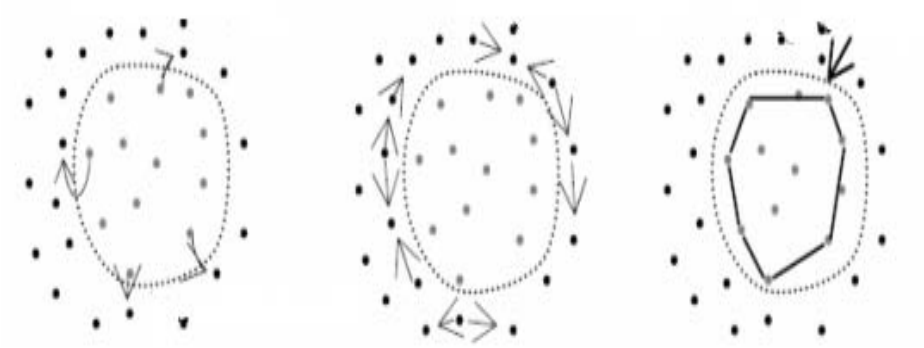
3.8. Llsp

LLSP[82], sadece simetrik güvenlik algoritmalarını kullanarak minimum maliyetle kimlik doğrulama, veri bütünlüğü ve semantik güvenliği sağlamıştır. Anahtar mekanizması, KAA'da anahtar yönetimi konularını belirlemektedir. Kriptograf anahtarlar nasıl dağıtılır, paylaşılır ve güncellenir sorularını içermektedir. Uygun bir anahtarlama mekanizması, hedef tehlike modeli, uygulamadaki ağ iletişimi ve güvenlik gereksinimleri ve kolay kullanım gibi bazı özel faktörlere bağlıdır. Anahtarlama mekanizması bu makalede ele alınmamıştır.

Literatürde bulunan mevcut güvenlik protokollerinden uygulaması olmayan LISP hariç diğer protokoller DoS saldırılarına karşı savunmasızdır. Literatürde KAA'da MAC katmanında meydana gelen DoS ataklarını önleme konusunda sınırlı sayıda çalışma bulunmaktadır. Mevcut çalışmalar aşağıdaki bölümde anlatılmıştır.

3.9. DoS Ataklarını Önleme Konusunda Yapılan Çalışmalar

3.9.1. Haritalama protokolü (Jammed-Area Mapping, JAM)



Şekil 3.1. Haritalama protokolü

Haritalama protokolü (Şekil 3.1) MAC katmanında gerçekleşen DoS ataklarını önlemek için tasarlanmıştır. Düğümler atak meydana gelip gelmediğini kontrol ederler. Atağı algılamak için kullanılan parametreler: Kablosuz alana erişimde tekrarlanan başarısızlık, alanlardaki anormal değişiklikler, protokol uyumsuzlukları, kayıp ACK2'lar, aşırı kabul edilen sinyal seviyesi, düşük sinyal-gürültü oranı, tekrarlanan çakışmalar, bekleme süresi. Şekilde görüldüğü gibi atağa maruz kalan düğümler komşularına mesaj göndermektedir. Atak altında düşman düğümlerin ortamı işgal ettiğinden dolayı, bir düğümün mesaj göndermesi oldukça zordur. Bu durumun üstesinden gelmek için Jammed mesajları önceliklendirilmiştir. Komşu düğümler gelen mesajları toplayarak jammed düğümlerin listesini oluşturmaktadırlar, böylece jammed bölgesi belirlenmektedir. Bununla birlikte, JAM [83] protokolünün henüz uygulaması gerçekleştirilmemiş olup, sadece benzetimi gerçekleştirilmiştir.

3.9.2. FS - MAC

IEEE 802.11 MAC protokolüne tespit ve savunma birimi eklenerek oluşturulmuştur. Paket çakışmalarının fazlalığı paket çakışması atağı, RTS paketlerinin fazlalığı tüketim atağı, paketlerin bekleme zamanı da adaletsizlik atağını teşkil etmektedir. Atağı belirlemek için bulanık mantık kullanılmıştır. Savunma yöntemi olarak ise atağa maruz kalan (jammed) düğümlerin atak bitene kadar kısa aralıklarla uyku moduna geçip uyanmasıdır. Bu protokolünde uygulaması gerçekleştirilmemiştir [84].

3.9.3. G - MAC

G-MAC [85] DoS ataklarına karşı merkezi grup metodu kullanmaktadır. Gruptaki düğümler, gruptaki diğer düğümlerle iletişime geçmek için gateway sensor'u kullanmaktadırlar. Diğer kaynaklardan alınan paketler ihmal edilmektedir, böylece deceptive jammer ataklarından kaçınılmaktadır. Bununla birlikte, bu protokol constant jammers, random jammers and reactive jammers lara karşı tam olarak güvenlik savunmasını yerine getirememektedir. G-MAC'te frame iki periyoda bölünmektedir. Bunlar toplanma ve dağılım. Toplanma periyodunda, gateway düğümü ağ trafiğini yönetmektedir. Gateway sensorü periyodik olarak enerji seviyesi en yüksek olan düğüm olarak seçilmektedir.

3.9.4. Diğer çalışmalar

Wenyuan Xu, et al. [86] 4 tip atak tanımlamış ve bunları belirlemek için bazı metodlar geliştirmişlerdir. Birinci metot sinyal gücüne bağlıdır. Çünkü atak durumunda sinyal gücünde anormal değişiklikler görülebilmektedir. İkinci metot, Carrier sense aralıklarıdır. Atak sırasında Carrier sense aralığı genişlemektedir. Diğer metot ise Paket varış oranını kontrol etmektir. Tabiki bu değerler tek başına yeterli değildir. Ağ içindeki mevcut düğümlerde atak olmaksızın bazı şartlarda bu oranlar yine anormal değişiklikler gösterebilmektedir. Bu yüzden Xu Çizelge 3.1.'de görüldüğü gibi bir yöntem geliştirmiştir.

Çizelge 3.1. Xu'ya ait DoS ataklarına karşı savunma yöntemi

Paket Teslim Oranı	Sinyal Gücü	Yorum
0	Düşük	Komşu düğüm zarar gördü
0	Yüksek	Atak var
Düşük	Düşük	Komşu düğümler gönderme alanından uzak
Düşük	Yüksek	Atak var

4. GELİŞTİRİLEN PROTOKOL

Bundan böyle DoSSec olarak adlandırılacak olan önerilen protokolde KAA'da güvenlik gereksinimlerden Veri Gizliliği, Veri Bütünlüğü, Kimlik Doğrulama, Veri Tazeliği'ni sağlamak için XXTEA+OCB kullanılmıştır.

4.1. Veri Gizliliği, Veri Bütünlüğü, Kimlik Doğrulama, Veri Tazeliği

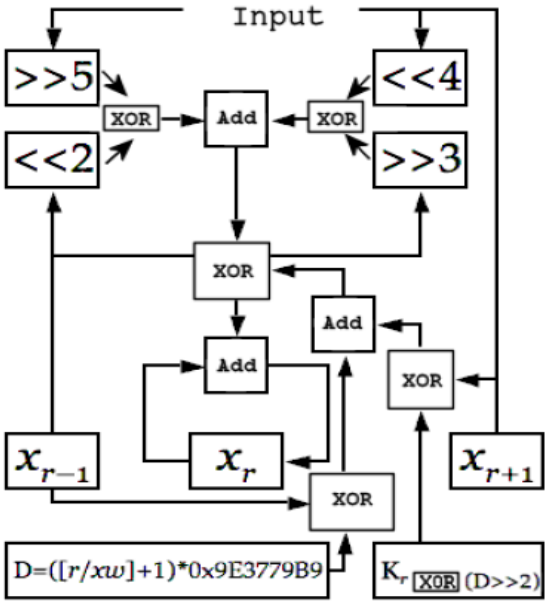
4.1.1. XXTEA (Corrected Block Tea)

Block TEA(XTEA) algoritmasının zayıf yönleri düzeltilerek dizayn edilmiştir. Cambridge Üniversitesinden Roger Needham ve David Wheeler bu algoritmayı geliştirmişlerdir. Algoritmayı tanıtan makale 1998 yılında yayınlanmıştır. XXTEA[87,88,89,90] algoritmasına ait özellikler aşağıda verilmiştir.

- Anahtar boyutu = 128 bit
- Blok uzunluğu = 64 bit
- Döngü sayısı = 32
- Güvenlik = 2076 yılına kadar
- Bilinen atak = Yok

XXTEA, 8 aşamadan oluşur. 4. aşamadan 8. aşamaya kadar olan kısım 32 kere tekrarlanır. XXTEA, sola kaydırma, sağa kaydırma, toplama ve XOR operatörlerini kullanan bir algoritmadır.

XXTEA'da bir turda yapılan işlemler Şekil 4.1.'de gösterilmiştir.



Şekil 4.1. XXTEA

XXTEA'da yapılan işlemler aşağıda anlatılmıştır.

Öncelikle giriş parametreleri ayarlanır.

- Şifrelemede kullanılacak 128 bitlik anahtar belirlenir.
- Şifrelenecek mesaj 64 bitlik bloklara ayrılır.
- Son mesaj 64 bit olana kadar 0 eklenir.
- Sum değeri 0'lanır.

Ardından 32 kere aşağıdaki işlemler tekrarlanır.

- Sum değeri ile Delta değeri toplanır. (sum += DELTA)
- Yandaki işlem ile e değeri belirlenir. (e = (sum >> 2) & 3)
- z mesajın ilk bloğu, y mesajın ikinci bloğu, k şifreleme anahtarı olmak üzere bu şekilde tüm mesaj blokları için aşağıdaki işlem gerçekleşir.

  (z>>5^y<<2) + (y>>3^z<<4)^(sum^y) + (k[p&3^e]^z)

Bu şekilde tüm mesaj blokları şifrelenir.

4.1.2. OCB (Offset Codebook Mode)

Kimlik doğrulama (authentication) ve gizlilik (privacy) prensiplerinin birlikte karşılandığı şifreleme modudur. Diğer modlar (ECB,CBC,CTR..) Veri gizliliği için şifreleme modu ve kimlik doğrulama için MAC olmak üzere iki ayrı sistem kullanmaktadır. OCB[91,92,93,94,95,96,97] bunları birleştirmiştir. Sadece gizliliği garanti eden CBC moduna göre daha az maliyetlidir. Buna rağmen OCB modu basit ve kolaylıkla yazılıma entegre edilebilir. OCB modunda kullanılan nonce(başlangıç vektörü) değeri random değildir. Counter olarak işlem yapar. OCB işlem moduna ait algoritma 6 adımdan oluşmaktadır. Bu adımlar aşağıda verilmiştir.

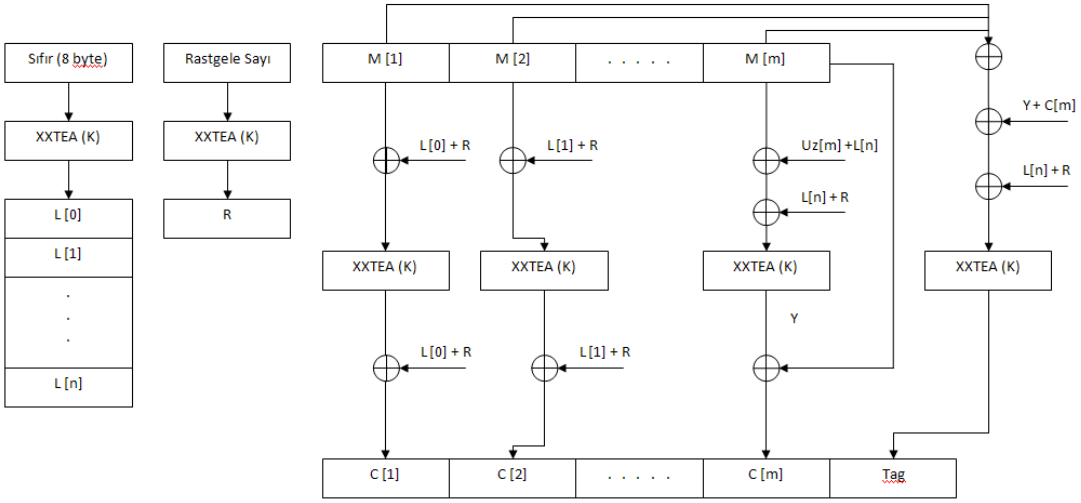
Birinci Adım: Girilen mesaj 64 bit bloklara ayrılır.
İkinci Adım: Son mesaj bloğu 64 bit olmazsa, geriye kalan bitler 0 ile tamamlanır.
Üçüncü Adım: Rastgele bir değer üretilir. Üretilen değer şifreleme algoritması kullanılarak, NONCE değerini üretmek için şifrelenir. Bu NONCE değeri ofset değeri üretiminde kullanılır.
Dördüncü Adım: OCB modunun önemli özelliklerinden bir tanesi L değeridir. Farklı mesaj blokları için bu L değeri farklıdır. L değeri şifreleme metodundaki karmaşıklığı arttırır.
- Başlangıçta L değerini üretmek için 8 byte 0 (8*8=64 bit) içeren değer alınır ve şifreleme algoritması yardımıyla şifrelenir. Şifrelenmiş değer L[0]'a eşittir.
- Mesaj bloklarının sayısına eşit oluncaya kadar L değeri üretilir.
- İlerdeki L değerleri (L2,L3..) aşağıdaki mantıkla üretilir.

    L[i] 'in MSB biti =0 ise L[i+1]= L[i]<<1
    L[i] 'in MSB biti =1 ise L[i+1]= L[i]>>1

Böylece bu L değerlerini kullanarak ofset üretilir. Her bir L değeri ile NONCE XOR'lanır. Yani i. mesaj bloğuna ait offset L[i] xor NONCE' dır. Bu ofset değeri şifrelemeden öncede sonrada mesaj ile XOR'lanır.
Beşinci Adım: Hash fonksiyonu yardımıyla tag değeri üretilir. Hash fonksiyonu orijinal mesaja aittir. Hash fonksiyonu her bir mesaj bloğunun XOR'lanmış halidir.

Son mesaj bloğu hariç olmak üzere bütün mesaj blokları XOR'lanır. Ardından şifreleme algoritması ile şifrelenir. Bu üretilen tag değeri kimlik doğrulama için kullanılır.

Altıncı Adım: Son adım bütün değerleri şifrelemek ve tag değeri ile şifreli metni üretmektir. OCB'ye ait şema Şekil 4.2.'de verilmiştir.



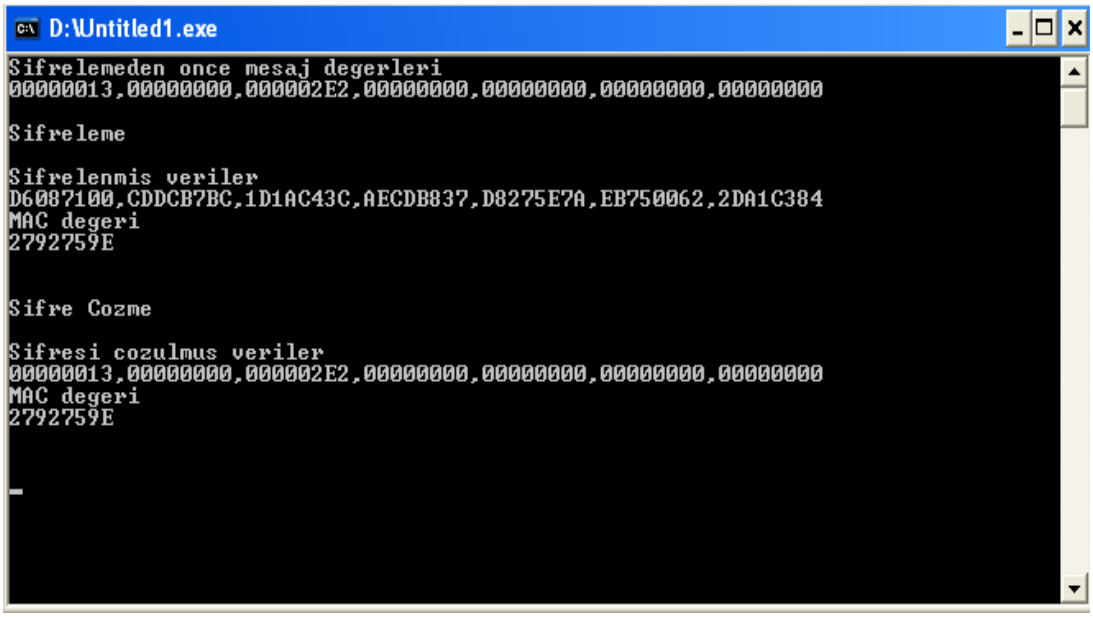
Şekil 4.2. OCB işlem modu

Şifreleme algoritmalarından aynı açık metni iki kez şifrelediklerinde farklı iki şifreli metin üretmeleri istenir. Algoritma her çalıştırıldığında, şifrelemede kullanılan bir vektörü (IV) değiştirmek bunu sağlar. IV uzarsa paket boyu ve yük artar. IV kısalırsa aynı IV'nin tekrarlanma sıklığı artar. Geliştirilen DoSSec protokolünde başlangıç vektörü için Nonce ile oluşturulmus IV seçilmiştir. Bu sayede mesaj tazeliği de sağlanmış olacaktır. Ayrı bir algoritmaya (CTR..) gerek duyulmayacaktır. TinySec ve MiniSec sayaç ile oluşturulmuş IV kullandıklarından dolayı pakete ek yük getirmişlerdir.

4.1.4. Benzetim sonuçları

XXTEA+OCB öncelikle C programlama dili ile kodlanmıştır. (350 satır) Aşağıda verilecek olan TOSSIM benzetim sonuçlarında kullanılan mesaj değerlerinin

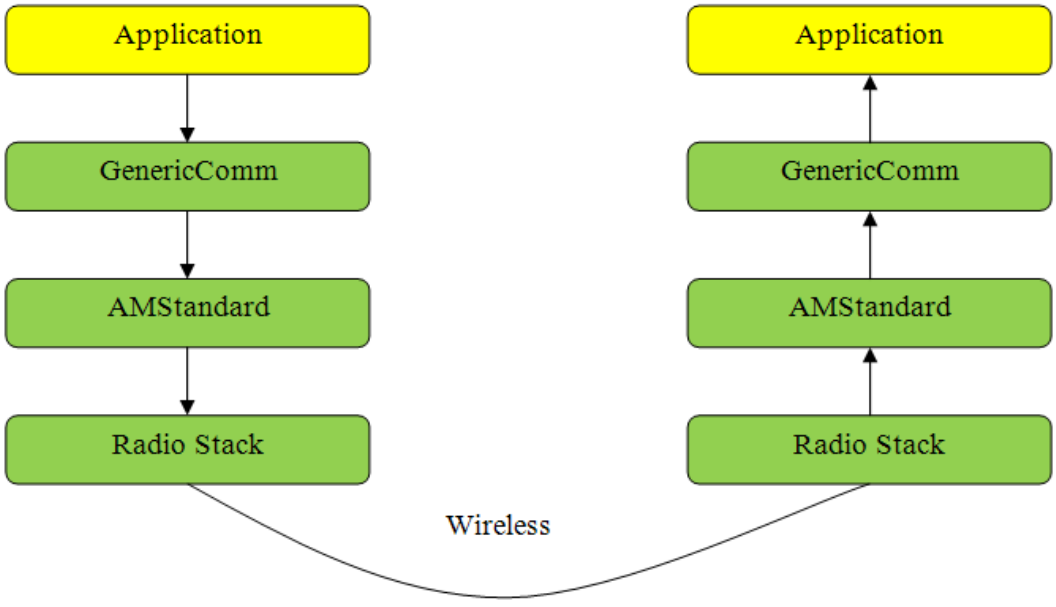
şifrelenmesi ve MAC değerinin hesaplanması işlemleri öncelikle bu programda yapılmıştır. Programa ait ekran çıktısı Şekil 4.3.'te verilmiştir.



Şekil 4.3. C programala dili ile kodlanan XXTEA+OCB'ye ait ekran çıktısı

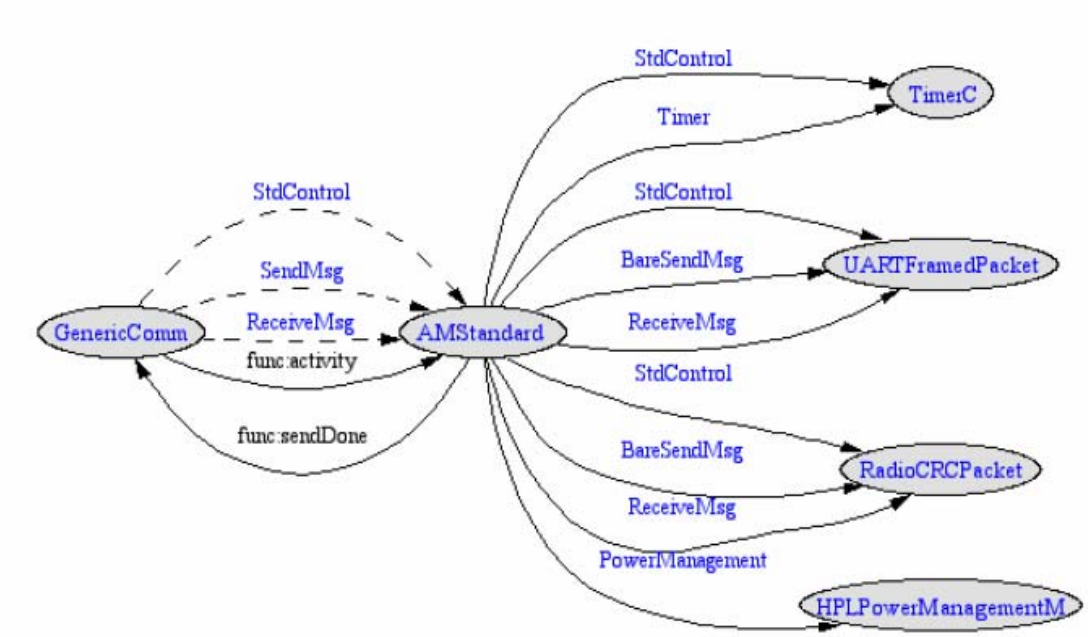
Ardından bu ikilinin çalışma prensibi anlaşıldıktan sonra yazılan kod NesC koduna çevrilmiştir.

TinyOS'ta mesaj gönderimi Şekil 4.4.'te verilmiştir.



Şekil 4.4. TinyOS'ta mesaj gönderimi

XXTEA ve OCB için yazılan kodlar ayrı bir dosya halinde modül olarak yazılmıştır. AMStandarda gelen mesaj paketi burada OCBMode arayüzü yardımıyla şifrelemeye gönderilmiştir. Alıcı düğümde de AMStandarda gelen mesaj paketi yine OCBMode arayüzü yardımıyla şifre çözmeye gönderilmiştir. Yani farklı şifreleme algoritması yada şifreleme modu kullanmak isteyen kullanıcıların, kendi algoritma yada modlarıyla bu dosyaları değiştirmeleri yeterli olacaktır. Bu şekilde yazılan kod hem modüler hem de kolay kullanım özelliklerine sahiptir. TinyOS'ta mesaj gönderimini gösteren diyagram Şekil 4.5.'te verilmiştir.



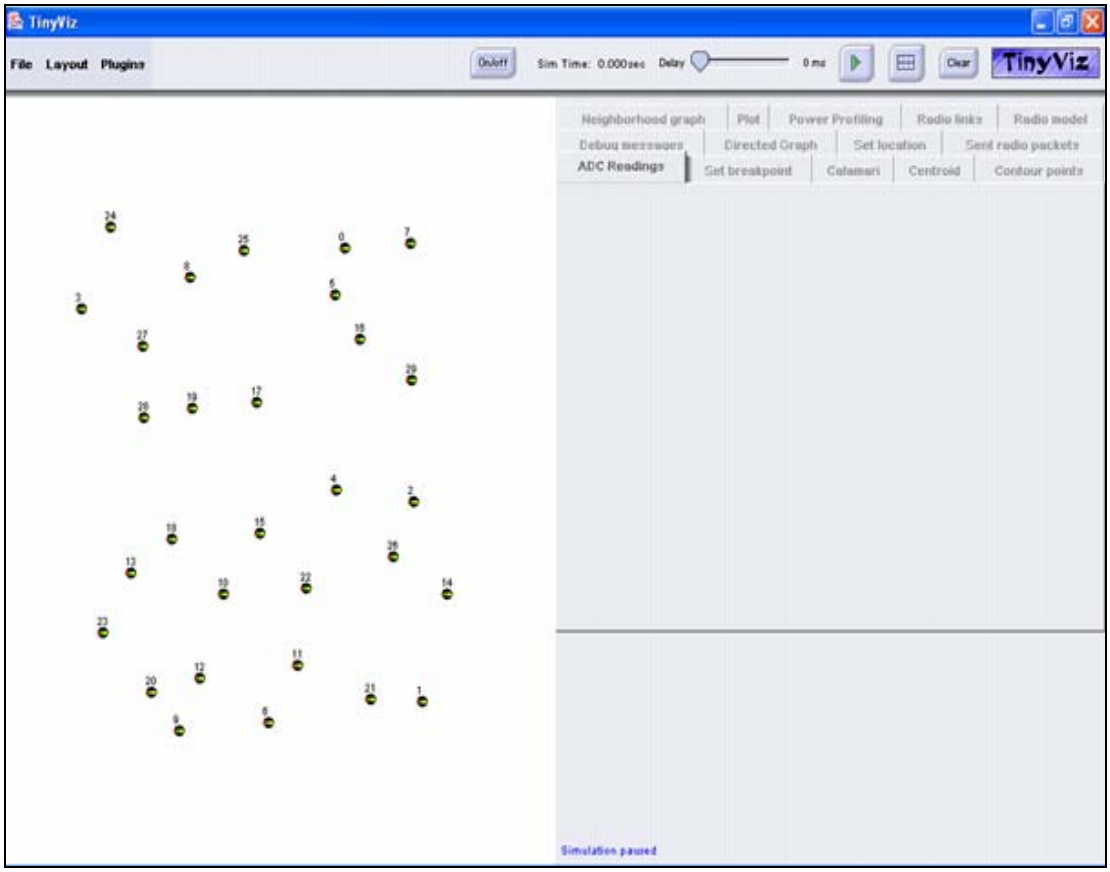
Şekil 4.5. TinyOS mesaj gönderimini gösteren diyagram

Bu işlem gerçekleştirildikten sonra yazılan NesC kodunun düğüme yüklemeden önce doğru işlem yapıp yapmadığını test etmek için TOSSIM simülasyon programı yardımıyla 30 düğüm üzerinde benzetim yapılmış sonuçlar alınmıştır. TOSSIM simülasyon programı TinyOS içerisinde bulunan, fakat sonradan başlıktan çıkarılan group id ile çalışmaktadır. Bu yüzden bu bölümde

- Düğümün ürettiği mesaj,
- Mesajın şifrelenmesi ve MAC'in hesaplanması,
- Mesajın gönderilmesi,

- Mesajın alınması,
- Şifresinin çözülmesi,
- MAC'in hesaplanıp karşılaştırılması,
- MAC'ler eşitse mesajın kabul edilmesi işlemleri gösterilmiştir.

Ayrıca TOSSIM benzetim programında her bir düğümün farklı bir şekilde çalışması sağlanamamaktadır. TOSSIM buna imkân vermemektedir. Bu yüzden bu bölümde de her bir düğüme mesaj alıp veren bir uygulama NesC ile kodlanarak benzetim çalıştırılmıştır. Benzetime ait ekran görüntüsü Şekil 4.6.'da verilmiştir.



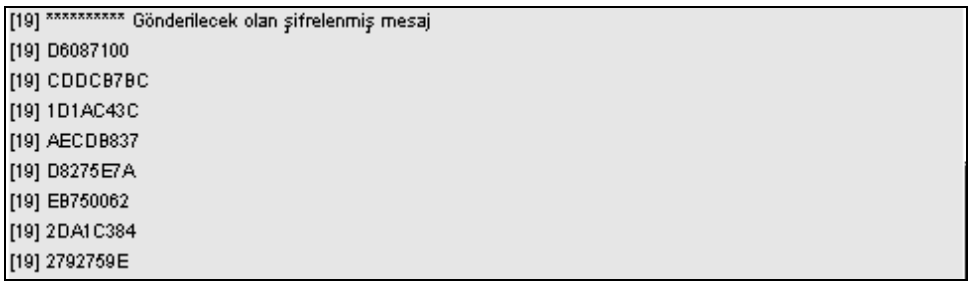
Şekil 4.6. TinyViz

Sistem şu şekilde çalışmaktadır.
19 nolu düğüm algılama işlemini gerçekleştirdikten sonra bir mesaj üretir.

Üretilen mesajın içeriği Şekil 4.7.'de verilmiştir.

Şekil 4.7. Üretilen mesajın içeriği

Ardından mesajı göndermeden önce şifreleme işlemini gerçekleştirir. Gönderilecek olan şifreli mesaj Şekil 4.8.'de verilmiştir.

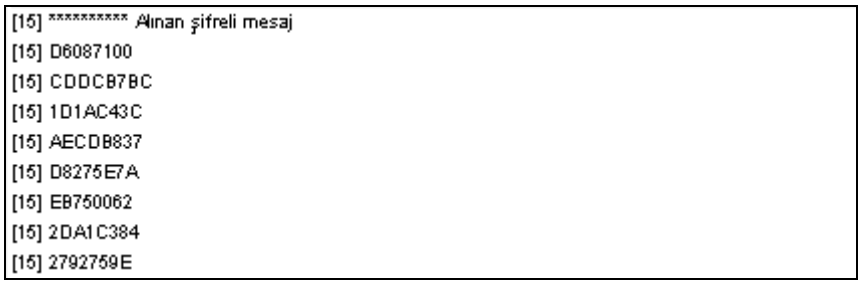


Şekil 4.8. Gönderilecek olan şifreli mesaj

Şifreli mesaj ile birlikte hesaplanan MAC'te gönderilecek olan mesaj paketine eklenmiştir. MAC değeri Şekil 4.9.'da verilmiştir.

Şekil 4.9. MAC değeri

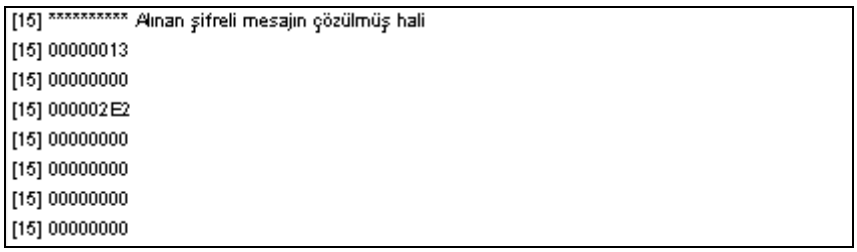
19 nolu düğüm tarafından gönderilen mesajı 15 nolu düğüm alır. Alınan şifreli mesaj Şekil 4.10.'da verilmiştir.



Şekil 4.10. Alınan şifreli mesaj

Alınan mesajın şifresi çözülür ve MAC değeri hesaplanır.

Alınan şifreli mesajın çözülmüş hali Şekil 4.11.'de verilmiştir.

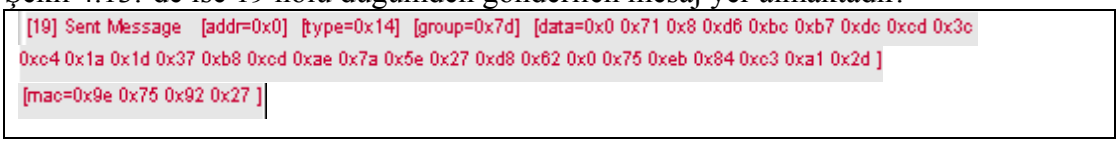


Şekil 4.11. Alınan şifreli mesajın çözülmüş hali

Hesaplanan MAC değeri, gelen MAC değeri ile aynı olduğundan mesaj işleme konulur. MAC değeri Şekil 4.12.'de verilmiştir.

Şekil 4.12. MAC değeri

Şekil 4.13.'de ise 19 nolu düğümden gönderilen mesaj yer almaktadır.



Şekil 4.13. Gönderilen mesaj

Yazılan NesC kodunun benzetimde sorunsuz çalıştığı gözlemlendikten sonra yazılan kod düğümlere yüklenmiştir.

4.2. Kullanılırlık

KAA'da düğümlerin içerisine yüklenen TinyOS işletim sistemi erişim yöntemi olarak CSMA kullanmaktadır. CSMA (Carrier Sense Multiple Access) birden çok kullanıcısı olan veri taşıma ortamlarında (paylaşımlı) göndericinin herhangi bir veriyi göndermeden önce veri yolunda trafiğin olup olmadığını kontrol ettiği bir iletişim kuralıdır.

Carrier Sence: Bir göndericinin herhangi bir veriyi göndermeden önce Carrier Wave (Taşıma Sinyali-Bilgi taşınmasına izin verme amaçlı gönderilen bir sinyal) sinyalini dinlemesi-beklemesi durumunu tarif eder. Bu, veri gönderiminden önce başka bir istasyon tarafından kodlanmış bir sinyalin veri yolunda bulunup bulunmadığının

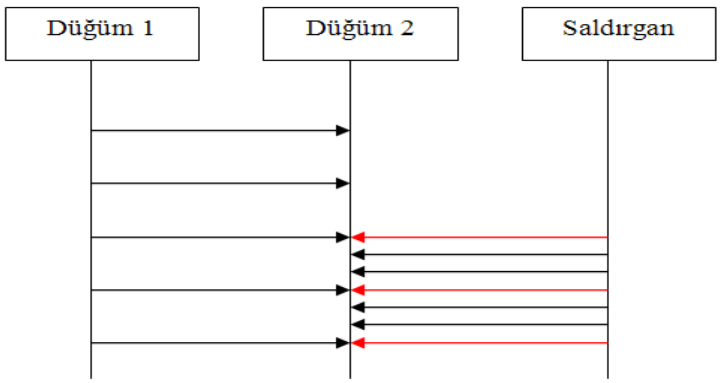
taranması işlemidir. Eğer herhangi bir gönderim işlemi bulunmuşsa istasyon iletime başlamadan önce daha önceki gönderim işleminin bitmesini bekler.

Multiple Access: Herhangi bir fiziksel ortamın birden çok istasyon tarafından kullanılması durumunu tarif eder.

Yani, "ortam boşsa veriyi gönder. Ortam boş değilse bekle ve bir süre sonra tekrar göndermeyi dene". Gerçekleştirilen güvenli veri bağı katmanı protokolü göz önüne alındığında veri alışverişi şu şekilde olmaktadır. Ortam boşsa veriyi gönderirken XXTEA+OCB kullanılacaktır. Mesajı gönderen düğüm şifreleme işlemini gerçekleştirirken, mesajı alan düğüm ise şifre çözme işlemini yerine getirecektir. Bu kısımda veri çalınsa bile karşı taraf için hiçbir şey ifade etmeyecektir. DoS atakları ise veriyi zaten elde edememektedir. Gerçekleştirilen DoS ataklarının amacı ağ trafiğini felç etmektir. Şöyle ki, ağdaki bir düğüm diğerine veri gönderirken saldırgan düğümünde ortama küçük boyutlu veri gönderip çakışmaya meydan vermek (paket çakışması) veya o düğümü meşgul etmek (tüketim), bir düğüme ulaşmak isteyen düğümlere meydan vermemek (adaletsizlik). Özetle, Veri Bağı Katmanındaki DoS atakları 3 tanedir. Bunlar Paket çakışması[98], Tüketim[99] ve Adaletsizliktir[100].
Ortam Erişim Kontrol Protokolü komşu-komşu iletişim için kanal tayini sağlamaktadır. Yalnızca ortam boş ise gönderme yapan CS (Carrier Sense, Taşıyıcı Algılama) modelini kullandığından dolayı DoS Ataklarına karşı savunmasızdır.

4.2.1. Paket çakışması ve tüketim atakları

Şekil 4.14.'te Paket Çakışması ve Tüketim ataklarına ait gösterim verilmiştir.

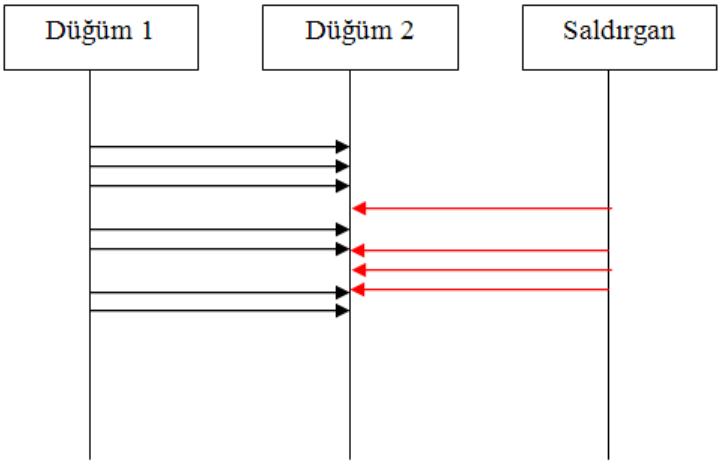


Şekil 4.14. Paket çakışması ve tüketim atakları

Düğüm 1 ortamda algılama işlemini gerçekleştirdikten sonra Düğüm 2'ye mesaj gönderir. Bu arada Saldırgan Düğüm'de CSMA iptal edildiği için, ortamın boş olmasını beklemeden Düğüm 2'ye devamlı olarak mesaj gönderir. Böylece paket çakışması ve tüketim atağı meydana gelmiş olur.

4.2.2. Adaletsizlik atağı

Şekil 4.15.'te Adaletsizlik atağına ait gösterim verilmiştir.



Şekil 4.15. Adaletsizlik atağı

CSMA tabanlı Ortam Erişimi Protokollerinde her bir düğüm için ortamı kullanma süresi eşittir. Her düğüm ortamı ele geçirmek için çaba sabreder ve bu adil olarak paylaştırılır. Saldırgan düğüm bu özelliği kullanmak suretiyle ağa paket göndermektedir. Böyle yaparak, kanalı ağa ait düğümler yerine bu saldırgan düğümler kullanmaktadır.

4.2.3. Geliştirilen uygulama

Geliştirilen uygulamada atak düğümler ve tespit ve savunma birimi oluşturulmuştur.

Atak Düğümler

Paket Çakışması ve Tüketim Atağı

- Düğümlerde CSMA iptal edilmiştir.

- Ortama küçük boyutlu paketlerle veri gönderme işlemi yapan kod düğüme yüklenmiştir.

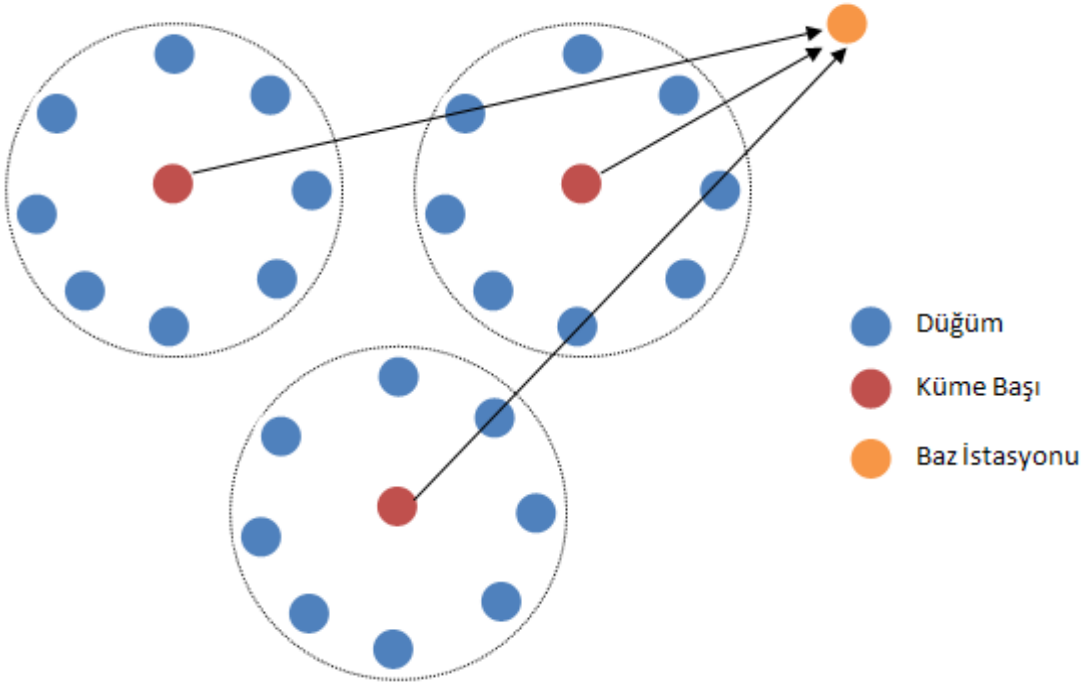
Adaletsizlik Atağı

- Ortama normal paketlerle veri gönderme işlemi yapan kod düğüme yüklenmiştir.

4.2.3. Tespit ve savunma birimi

Paket çakışması ve tüketim atağı

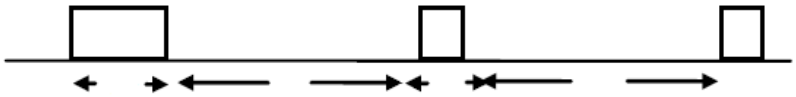
Kablosuz Algılayıcı Ağlarda baz istasyonuna veri gönderimi Şekil 4.16.'da verilmiştir.



Şekil 4.16. Kablosuz algılayıcı ağlarda baz istasyonuna veri gönderim şekli

Düğümler ortama bırakıldıktan sonra her grup kendine ait bir küme başı belirler. Bu düğümler diğer düğümlere göre enerjisi yüksek, önceden belirlenen düğümlerdir. Veri akışı şu şekilde gerçekleşmektedir. Ortamdaki düğümler küme başlarına veri gönderirken, küme başları da baz istasyonuna veri göndermektedir. Ataklara ait tespit ve savunma birimi geliştirirken bu özellik esas alınmıştır. Ortamdaki düğümler

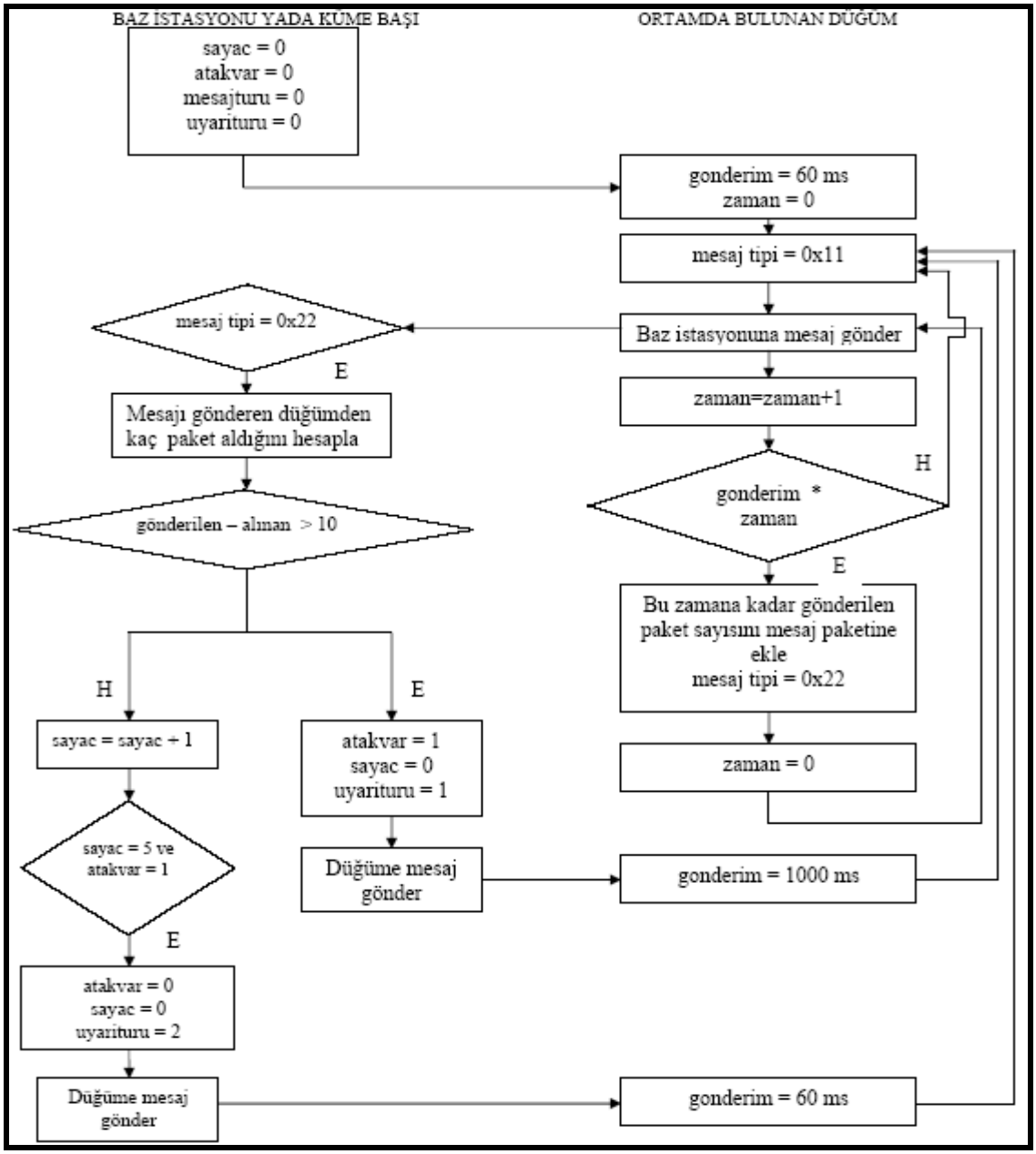
mesaj tipi 0x11 olacak şekilde 60 ms de bir küme başına veya baz istasyonuna veri göndermektedir. Bu süre 1 dakika (60000 ms) ya ulaştığında düğüm bu ana kadar kaç paket gönderdiğini mesaj paketine ekleyerek, mesaj tipi 0x22 olacak şekilde gönderir. Bu mesajı alan küme başı veya baz istasyonu gelen paketten bu paketin hangi düğüme ait olduğunu bulur. Ardından bu düğümden alınan paket sayısını bulur. Eğer o düğümün gönderdiği paket sayısı ile alınan paket sayısı arasında fark yoksa ağ güzel bir şekilde işlemektedir. Eğer varsa paket gelirken havada kaybolmuş, yani paket çakışması meydana gelmiştir. Bunu anlayan küme başı veya baz istasyonu o düğüme uyarı mesajı gönderir ve bu mesajı alan düğümde mesaj gönderim sıklığını 1000 ms yapmak suretiyle mesaj göndermesine devam etmektedir. Literatürde bilindiği gibi atağı engellemek için Hız Sınırlama Tekniği (Rate Limiting) kullanılmıştır. Hız sınırlama Tekniğine ait işlem Şekil 4.17.'de verilmiştir.



Şekil 4.17. Hız sınırlama tekniği

Yukarıdaki şekilde görüldüğü gibi radyonun aktif olma süresi kısaltılmıştır. Dost düğümlerin iletişim anları ile saldırganın saldırı anları çakışmaz ise saldırganın etkisi kalmayacaktır. Bu olasılığı en aza düşürmenin yollarından birisi de düğümlerin dinleme sürelerini azaltmaktır. Özetle, bir dinleme/uyuma periyodu içerisinde daha uzun süre uyumak ve daha kısa uyanık kalarak iletişimi gerçekleştirmektir. Böylelikle saldırganın saldırı paketlerini düğümlerin iletişim anlarına denk getirme olasılığı çok daha azalır. Bu teknik sayesinde ağın yaşam süresi oldukça artar. Radyo tarafından muhafaza edilen ve alınan veri miktarı azaltılarak, oluşan atağın etkisi önemli ölçüde azaltılmaktadır. Atak meydana geldikten belli bir süre sonra saldırgan düğümün enerjisinin bitmiş olması ya da kapanmış olması ihtimali göz önüne alınmıştır. Bu süreçte ağın veri gönderim hızının yavaş olarak devam etmemesi için şu yol izlenmiştir. 5 dakika içinde herhangi bir paket kaybı yoksa küme başı veya baz istasyonu mesaj göndererek düğümün tekrar 60 ms de veri göndermesini

sağlamaktadır. Paket Çakışması ve Tüketim Atağı için geliştirilen Tespit ve Savunma Birimine ait akış şeması Şekil 4.18.'de verilmiştir.



Şekil 4.18. Paket çakışması ve tüketim atakları için oluşturulan tespit ve savunma birimine ait akış şeması

Adaletsizlik atağı

CSMA tabanlı Ortam Erişimi Protokollerinde her bir düğüm için ortamı kullanma süresi eşittir. Örneğin 5 tane düğüm küme başı veya baz istasyonuna veri gönderecekse her birisi ortamın % 20'sini kullanmış olmalıdır.

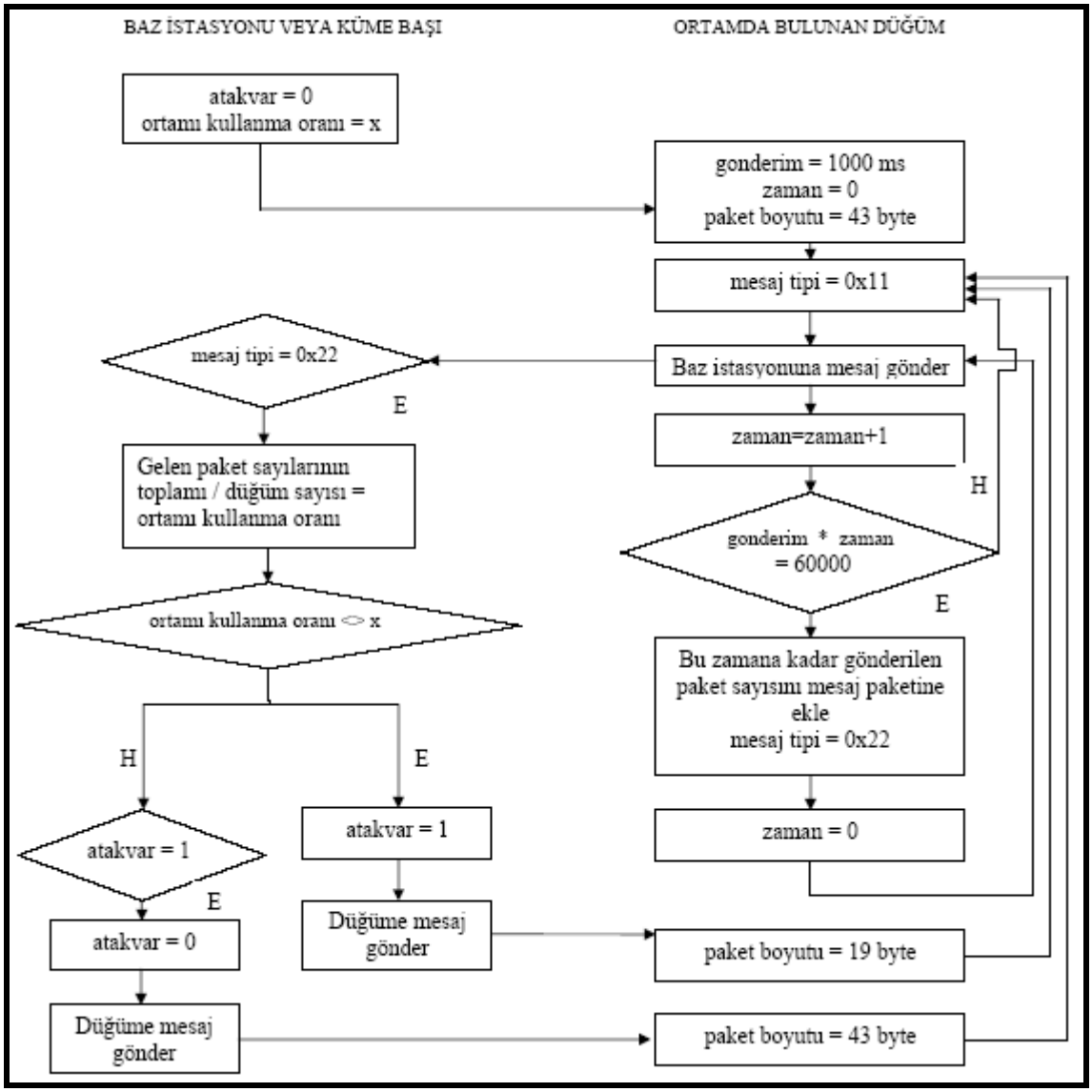
Ortamı Kullanma Oranı = (x düğümünün gönderdiği paket sayısı / düğümlerin gönderdiği paket sayılarının toplamı) * 100

Yukarıdaki eşitlikten de görüldüğü gibi her düğüm eşit sayıda küme başına veya baz istasyonuna paket gönderecek (örneğin 50 tane) ve formüle göre;

Ortamı Kulllanma Oranı = (50 / 50+50+50+50+50) * 100
Ortamı Kulllanma Oranı = (50 / 250) * 100
Ortamı Kulllanma Oranı = 20 bulunmaktadır.

Yani küme başı veya baz istasyonu muhatap olduğu düğüm sayısını bildiğinden dolayı her 1000 ms'de bir düğümler mesaj gönderdiğinde ortamı kullanma oranını hesaplamaktadır. Bu değer küme başı veya baz istasyonunun hesapladığı değere eşitse ağ güzel bir şekilde işlemektedir. Fakat bu değer olması gerekenden az çıkarsa o zaman ortamı saldırgan düğüm kullanıyor demektir. Bunu anlayan küme başı veya baz istasyonu düğüme mesaj gönderir ve o düğüm artık mesaj gönderme işlemine paket boyutunu küçültmüş bir şekilde devam eder. Küçük boyutlu paketler, büyük boyutlu paketler ile karşılaştırıldığında düşük iletim gücüne ihtiyaç duyarlar. Bir hata olma durumu büyük boyutlu paketlere göre oldukça azdır. Küçük boyutlu paketler sayesinde ağa ait düğümler ortamı kullanma oranını artırabilmektedir. Atağın bittiği küme başı veya baz istasyonu tarafından anlaşıldığında ise tekrar düğümlere mesaj gönderip eski gönderim şekillerine devam etmeleri sağlanmaktadır.

Adaletsizlik Atağı için geliştirilen Tespit ve Savunma Birimine ait akış şeması Şekil 4.19.'da verilmiştir.

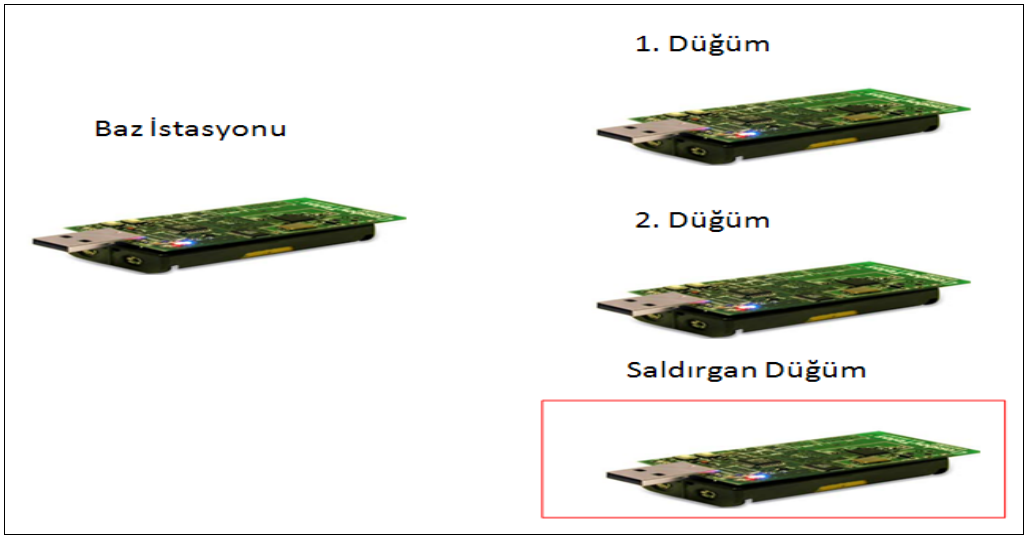


Şekil 4.19. Adaletsizlik atağı için oluşturulan tespit ve savunma birimine ait akış şeması

5. GELİŞTİRİLEN PROTOKOLÜN ANALİZİ

5.1. Kullanılırlık Prensibi

DoS Atakları ve tespiti esnasında değerlendirme yapabilmek amacıyla 4 tane algılayıcı düğüm kullanılmıştır. Bu düğümlerden birincisi baz istasyonu, ikincisi saldırgan düğüm ve geri kalanlar ise ortamda bulunan düğümlerdir. Düğümlere ait gösterim Şekil 5.1.'de verilmektedir. Ortamda bulunan düğümler baz istasyonuna mesaj göndermekte iken baz istasyonuda gelen mesajları seri porta aktarmaktadır. Seri porta gelen mesajlarda TinyOS klasörü içerisinde bulunan Listen.java programı yardımıyla bilgisayar ekranında görülebilmektedir. Listen.java programında bazı eklemeler yapılmıştır. Yapılan eklemelerle seri porta gelen veriler veritabanına kaydedilmektedir. Veritabanı yönetim sistemi olarak PostgreSQL kullanılmıştır. Veritabanına kaydedilen verilerden, çıkarım yapılmak suretiyle, sonuçları görebilmek için Delphi programı kullanılarak bir arayüz yazılmıştır. Atakların meydana gelmesi ve önlenmesi hususunda Şekil 5.1. referans alınmaktadır. Çerçeve içindeki düğüm saldırgan düğümü kastetmektedir.



Şekil 5.1. Atakların meydana gelmesi ve önlenmesi hususunda referans alınan senaryo

5.1.1. Paket çakışması ve tüketim atağı

Senaryo şu şekilde düzenlenmiştir. Ortamdaki düğümler baz sitasyonuna paket göndermektedir. İlk 1 dakika içerisinde saldırgan düğüm kapalı olacaktır. 2. dakikanın başından itibaren saldırgan düğüm açılacak ve atak başlayacaktır.

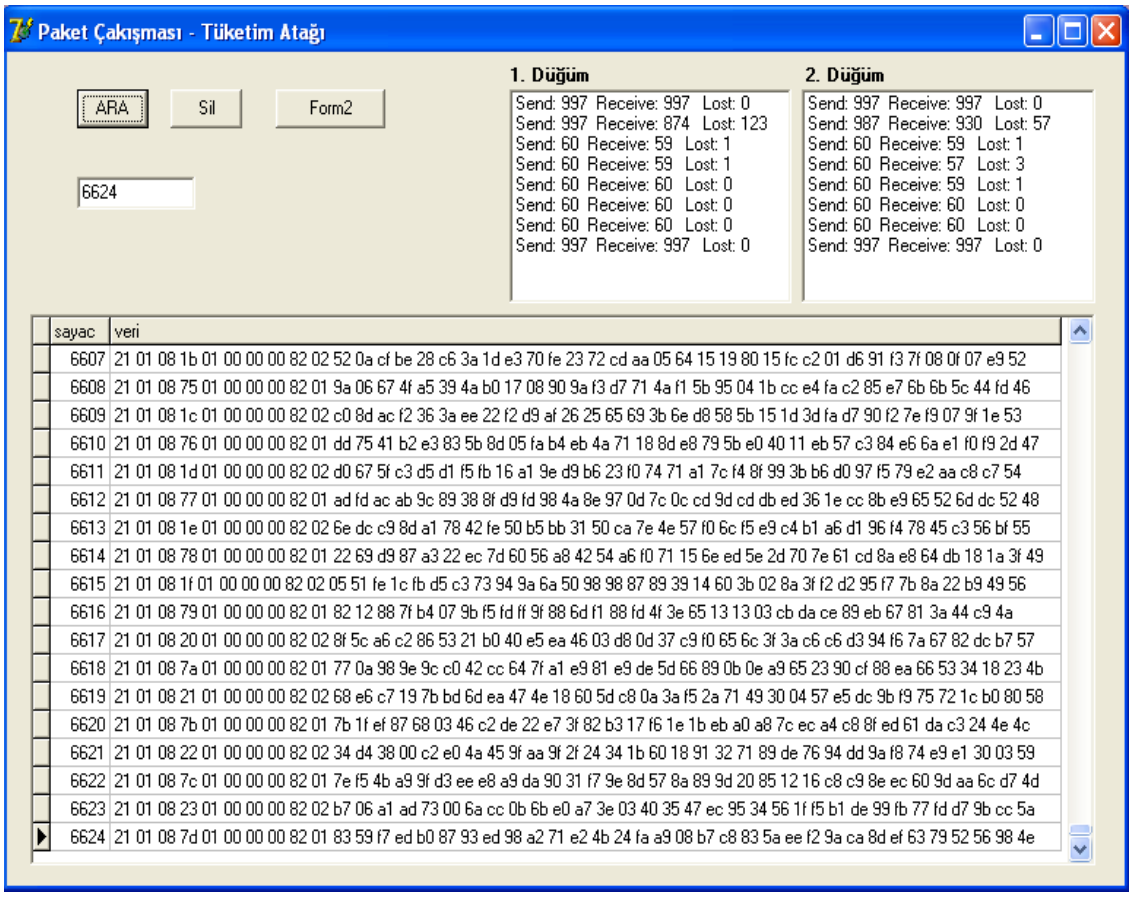
Paket Çakışması ve Tüketim ataklarının gösterimi için düğümler ilk başta mesaj paketlerini 60 ms'de bir göndermektedir. Atak anında hız sınırlamak için 60 ms'yi artırmak gerekmektedir. Hangi değere kadar artırılması gerektiğine farklı zaman birimleri verilmek suretiyle karşılaştırma sonucu ile bulunmuştur. Bu sonuçlar Çizelge 5.1.'de görülebilmektedir.

Çizelge 5.1. Atak anında başarılı olarak gönderilen paket oranı

Atak anında (ms)	Ortalama Gönderilen paket sayısı (adet)	Ortalama Paket Kaybı (adet)	Başarılı olarak gönderilen paket oranı (%)
60	997	120	87.96
80	749	103	86.25
100	558	89	84.05
120	497	57	88.53
150	398	36	90.95
200	298	28	90.60
250	237	20	91.56
300	199	14	92.96
400	148	11	92.57
500	118	9	92.37
600	100	8	92.00
750	79	7	91.14
800	74	6	91.89
1000	60	1	98.33
1200	50	1	98.00

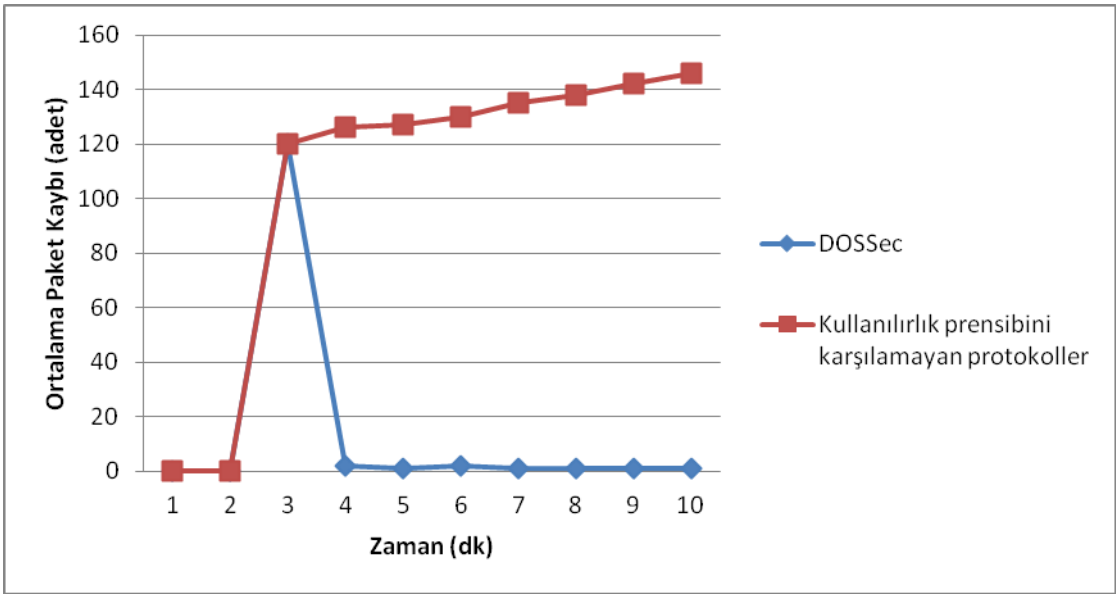
Atak anında başarılı olarak gönderilen paket oranı en yüksek 1000 ms'de olduğu için, atak meydana geldiğinde paket gönderim sıklığı 1000 ms olarak ayarlanmıştır. Normalde (60 ms) 997 paket gönderilirken, atak anında (1000 ms) bunun 60 pakete düşmesi şu şekilde açıklanabilir. Atak meydana geldiğinde asıl amaç paket kaybını

azaltmaktır. Zaten 1 dakika içerisinde gelen 60 paketten, 997 paket gönderildiğinde gelen değerlerin örneklemesi bulunmaktadır. Bu yüzden atak meydana geldiğinde başarılı olarak gönderilen paket oranının en yüksek olduğu değeri kullanmak daha makul olacaktır. Sonuçlar Şekil 5.2.'de verilmiştir. Şekil 5.2.'de düğüm yazılarının altında bulunan liste kutusunda her bir satır 1 dakikayı ifade etmektedir. Görüldüğü gibi 1.dakikada saldırgan düğüm kapalı olduğundan dolayı herhangi bir kayıp gözlenmemiştir. 2. dakikanın başında saldırgan düğüm açılmış ve 1. düğümden 123, 2. düğümden ise 57 adet paket kaybolmuştur. Bunun üzerine baz istasyonu ortamdaki düğümlere hızlarını yavaşlatmaları için mesaj göndermektedir. 3. dakikadan itibaren saldırgan düğüm atağına devam etmesine rağmen herhangi bir kayıp meydana gelmemektedir. Belli bir süre kayıp olmazsa, baz istasyonu tarafından düğümlere tekrar mesaj gönderilecek ve düğüm eski gönderim şekline devam edecektir.



Şekil 5.2. Paket çakışması ve tüketim atakları

Atak uygulaması 30 defa çalıştırılmış, ortalama sonuçlar Şekil 5.3.'de verilmiştir.



Şekil 5.3. Paket çakışması ve tüketim atakları esnasında meydana gelen ortalama paket kaybı

İlk 1 dakikada atak olmadığı için paket kaybı 0'dır. 1. dakika bittikten sonra 2. dakikaya kadar saldırgan düğüm atağa başlamış ve ortalama 120 paket kayıp olmuştur. 3. dakikadan sonra DoSSec protokolü atağı algılamış baz istasyonunun uyarı mesajıyla hızını sınırlayan düğümde hızını azaltarak bundan sonraki zamanlarda paket kaybını 0-1'e indirgemiştir. Fakat kullanılırlık prensibini karşılamayan protokollerde atak devam ettiği için ve ortalama paket kaybı da artarak devam edecektir.

5.1.2. Adaletsizlik atağı

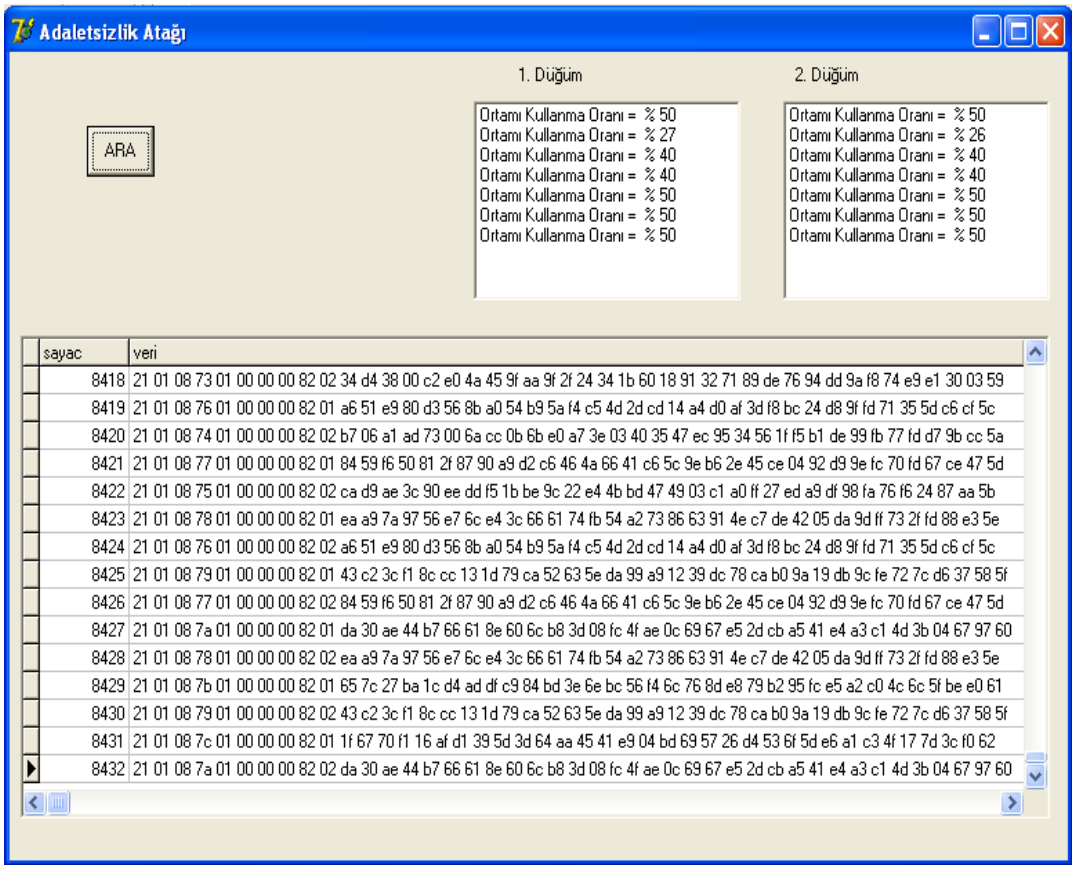
Senaryo şu şekilde düzenlenmiştir. Ortamdaki düğümler baz istasyonuna paket göndermektedir. İlk 1 dakika içerisinde saldırgan düğüm kapalı olacaktır. 2. dakikanın başından itibaren saldırgan düğüm açılacak ve atak başlayacaktır.

Ortamdaki düğümler 60000 ms'de bir (1 dakika) baz istasyonuna ne kadar paket gönderdiğini bildirmektedir. Baz istasyonuna aynı sayıda paket gönderen 1. ve 2. düğümlerin ortamı kullanma oranları aşağıdaki eşitlik yardımıyla bulunabilmektedir.

Ortamı Kullanma Oranı = (x düğümünün gönderdiği paket sayısı / düğümlerin gönderdiği paket sayılarının toplamı) * 100

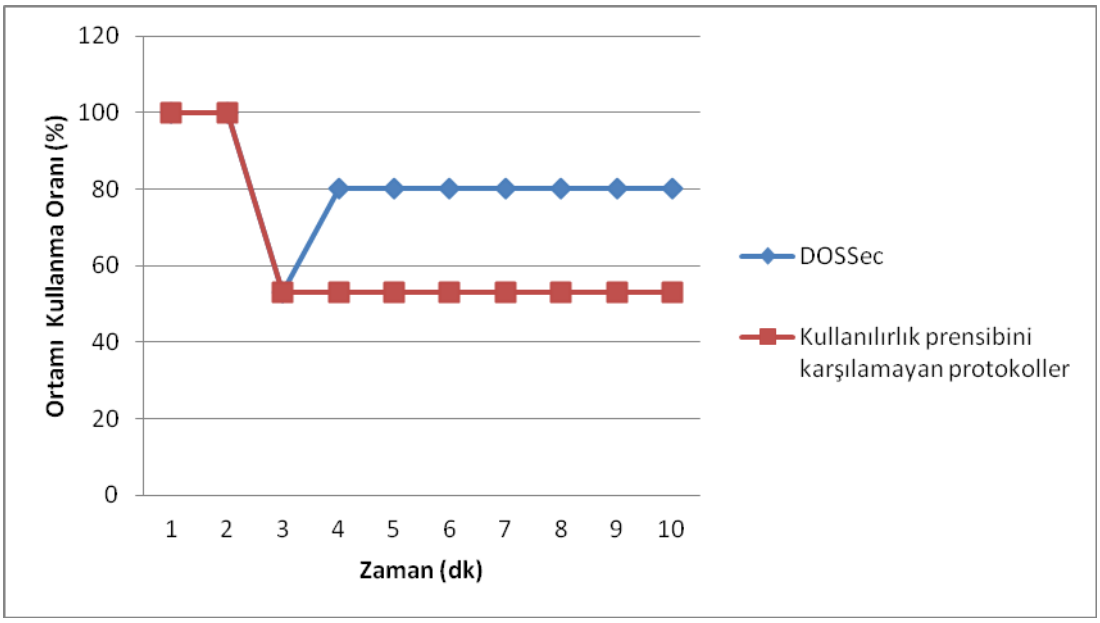
1. ve 2. düğümüm 40 paket gönderdiği varsayılırsa,
1. veya 2. düğümün ortamı kullanma oranı = (40 / 80) * 100
1. veya 2. düğümün ortamı kullanma oranı = %50'dir.

Bu demektir ki ortamda başka bir saldırgan düğüm olmadığı müddetçe ortamı kullanma oranı % 50 olmalıdır. Sonuçlar Şekil 5.4.'de verilmiştir. Şekil 5.4.'de düğüm yazılarının altında bulunan liste kutusunda her bir satır 1 dakikayı ifade etmektedir. Görüldüğü gibi 1. dakikada düğümlerin ortamı kullanma oranları toplamı % 100 iken, 2. dakikada saldırgan düğüm açıldığından dolayı düğümlerin ortamı kullanma oranları toplamı (26+27=53) % 53'e gerilemiştir. Dolayısıyla 1. ve 2. düğümlerin ortamı kullanma oranı neredeyse yarı yarıya azalmıştır. Bunu anlayan baz istasyonu düğümlere uyarı mesajı gönderecek ve bu mesajı alan düğümlere artık küçük boyutlu paketler göndererek görevlerine devam edecektir. Küçük boyutlu paket gönderildiğinde saldırgan düğümün ortamı kullanma oranı azalırken, ortamdaki düğümlerin ortamı kullanma oranları % 80'ene yükselmiştir. Atak devam ettiği sürece ortamı kullanım oranları bu seyirde devam edecektir. Atak sonladığında düğümlerin ortamı kullanım oranları % 100 olacak ve tekrar eski şekilde gönderimine devam edeceklerdir.



Şekil 5.4. Adaletsizlik atağı

Atak uygulaması 30 defa çalıştırılmış, ortalama sonuçlar Şekil 5.5.'te verilmiştir.



Şekil 5.5. Adaletsizlik atağı esnasında meydana gelen ortamı kullanma oranı

İlk 1 dakikada atak olmadığı için düğümlerin toplam ortamı kullanma oranı % 100'dür. 1. dakika bittikten sonra saldırgan düğüm atağa başlamış ve ortamı kullanma oranı % 53'e kadar gerilemiştir. 3. dakikadan sonra DoSSec protokolü atağı algılamış baz istasyonunun uyarı mesajıyla paket boyutunu küçülterek gönderim yapan düğümlerde ortamı kullanma oranı % 80'ene ulaşmıştır. Fakat kullanılırlık prensibini karşılamayan protokollerde atak devam ettiği için ortamı kullanma oranı % 53'de kalmıştır.

5.2. Diğer Kriterler

Çizelge 5.2.'de mevcut protokoller ve önerilen protokol (DoSSec) farklı kriterler üzerinden karşılaştırılmıştır.

Çizelge 5.2. KAA için güvenlik gereksinimleri / protokoller

KAA için güvenlik gereksinimleri / Protokoller	Veri gizliliği (Data confidentiality)	Veri bütünlüğü (Data integrity)	Verinin doğruluğunu kanıtlama (Data authentication)	Veri Tazeliği (Data freshness)	Kullanılırlık (Availability)	Uygulaması
TinySec	+	+	+	-	-	TinyOS (Mica2)
SPINS	+	+	+	+	-	-
MiniSEC	+	-	+	+	-	TinyOS (TelosB)
LSec	+	-	+	-	-	-
LLSP	+	+	+	+	-	-
LISA	+	+	+	+	-	-
IEEE 802.15.4	+	+	+	+	-	TinyOS (MicaZ, TelosB)
LISP	+	+	+	-	+	-
DoSSec	+	+	+	+	+	TinyOS (TelosB)

Çizelge 5.2.'de görüldüğü gibi geliştirilen DoSSec protokolünde KAA güvenlik gereksinimleri sağlanmıştır. Aynı zamanda DoSSec protokolü TelosB düğüm üzerinde uygulanmıştır. Kullanılırlık prensibini sağlayan Lisp protokolünün

uygulamasının olmaması bir eksikliktir. Uygulaması olan TinySec, MiniSec ve IEEE 802.15.4 güvenlik çözümleri ise KAA prensiplerinden hepsini karşılayamadığından güvenlik açıkları meydana gelmektedir.

Çizelge 5.3.'de [16,17,18,76] uygulaması olan protokollere ait Şifreleme algoritmaları ve modları sunulmaktadır.

Çizelge 5.3. Uygulaması olan protokollere ait şifreleme algoritmalar ve modları

	Şifreleme algoritması	Şifreleme Modu
TinySec	Skipjack	CBC+CTR
MiniSec	Skipjack	OCB
IEEE 802.15.4	AES	CBC
DoSSec	XXTEA	OCB

TinySec ve MiniSec Skipjack şifreleme algoritmasını kullanırken, IEEE 802.15.4 AES şifreleme algoritmasını kullanmıştır. Şifreleme modu olarak ta TinySec ve IEEE 802.15.4 CBC'yi MiniSec ise OCB'yi tercih etmiştir. DoSSec protokolünde ise şifreleme algoritması olarak XXTEA, şifreleme modu olarak ta OCB kulanılmıştır. Çizelge 5.4.'te [13,49,57] uygulaması olan protokollerin kullandığı şifreleme algoritmalarının özellikleri yer almaktadır.

Çizelge 5.4. Uygulaması olan protokollerde kullanılan şifreleme algoritmalarının özellikleri

Protokoller / Değişkenler	Tarihi	Blok Uzunluğu (bit)	Anahtar Uzunluğu (bit)	Döngü Sayısı
Skipjack	1993	64	80	32
AES	2002	128	128,192,256	10,12,34
XXTEA	1998	64	128	32

Skipjack şifreleme algoritmasının anahtar uzunluğu 80 bit iken, XXTEA'nın 128 bittir. AES'in anahtar uzunluğunda ise 128,192,256 olmak üzere farklı seçenekleri mevcuttur. Çizelge 5.5.'te farklı anahtar boyutları ve güvenlik konulu yapılan araştırmalara ait [11] sonuçlara yer verilmektedir.

Çizelge 5.5. Farklı anahtar boyutları için anahtar çözme süreleri

Anahtar Uzunluğu (n)	Olası değer sayısı (2^n)	10^6 şifre/s hızında ortalama çözme süresi	10^9 şifre/s hızında ortalama çözme süresi	10^{12} şifre/s hızında ortalama çözme süresi
32 bit	~$4x10^9$	36 dak	2.16 s	2.16 ms
40 bit	~10^{12}	6 gün	9 dak	1 s
56 bit	~$7.2x10^{16}$	1142 yıl	1 yıl 2 ay	10 saat
64 bit	$1.8x10^{19}$	292 000 yıl	292 yıl	3.5 ay
128 bit	$1.7x10^{38}$	$5.4x10^{24}$ yıl	$5.4x10^{21}$ yıl	$5.4x10^{18}$ yıl

Anahtar uzunluğu Hangi yıla kadar kullanılabileceği

- 56 bit 1982 (1977'de tanıtılan DES algoritması 1982'de kırılmıştır.)
- 64 bit 1994
- 80 bit 2013
- 128 bit 2076

Çizelge 5.5.'te görüldüğü gibi 80 bit anahtar uzunluğuna sahip olan şifreleme algoritmasının günümüzde güvenlik açığına sebep olma ihtimalinin yüksek olduğundan 128 bitlik anahtar uzunluğuna sahip olan XXTEA şifreleme algoritması DoSSec protokolünde kullanılmıştır. Bu sonuçlardan görüldüğü üzere 128 bit anahtar kullanan DoSSec, 80 bit anahtar kullanan TinySEC ve MiniSEC protokollerinden daha güvenlidir. Bundan sonraki kısımlarda karşılaştırma işlemleri TinySEC ve MiniSEC protokolleri ile yapılacaktır. Karşılaştırma işlemlerinde protokollerin tanıtımı için yayınlanan ilk makaleleri [16,17] referans alınmıştır.

5.2.1. Güvenlik

Uygulaması olan protokollere ait güvenlik karşılaştırması Çizelge 5.6.'da verilmektedir.

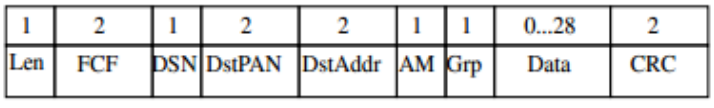
Çizelge 5.6. Uygulaması olan protokollere ait güvenlik karşılaştırması

	DoSSec	TinySec	MiniSec
Anahtar Uzunluğu	128 bit	80 bit	80 bit
Mesaj tekrar yayınlama ataklarını önleme	+	-	+
Verinin bütünlüğü	+	+	-
Kullanılırlık İlkesi	+	-	-

DoSSec protokolünün anahtar uzunluğunun 128 bit oluşu ve güvenlik gereksinimlerinden kullanılırlık, veri bütünlüğü, veri tazeliğini sağlamasından dolayı, geliştirilen DoSSec protokolünün TinySec ve MiniSec'e göre daha güvenli olduğu açıktır. Ayrıca Kablosuz Algılayıcı Ağların diğer kriterleri göz önüne alınarak karşılaştırma yapılmıştır. Sonuçlar 4 farklı şekilde sunulmaktadır. Bunlar; Paket Boyutu, Enerji, Gecikme, Bellek Kullanımı.

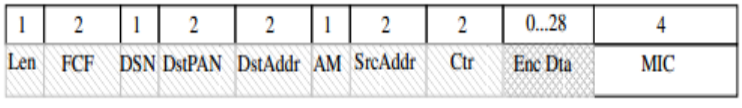
5.2.2. Paket boyutu

TinyOS'un paket boyutu 40 byte iken TinySec 5, MiniSec 3, DoSSec ise 3 byte TinyOS'a ek yük getirmişlerdir. TinyOS Paket Formatı Şekil 5.6., TinySec Paket Formatı Şekil 5.7., MiniSec Paket Formatı Şekil 5.8., DoSSec Paket Formatı ise Şekil 5.9.'da verilmiştir.



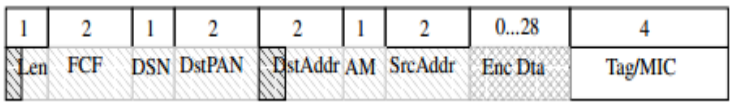
10 byte header
28 byte data
2 byte crc
40 Byte

Şekil 5.6. TinyOS paket formatı



13 byte header
28 byte data
4 byte mac
45 Byte

Şekil 5.7. TinySec paket formatı



11 byte header
28 byte data
4 byte mac

43 Byte

Şekil 5.8. MiniSec paket formatı

1	2	1	2	2	1	2	0..28	4
Len	FCF	DSN	DstPan	DstAddr	AM	SrcAddr	Enc Dta	Tag/MIC

11 byte header
28 byte data
4 byte mac

43 Byte

Şekil 5.9. DoSSec paket formatı

DoSSec, TinySec ve MiniSec protokollerinin paket formatı açısından karşılaştırması Çizelge 5.7.'de verilmektedir.

Çizelge 5.7. Paket boyutu karşılaştırması

	Toplam Paket Boyutu (byte)	Güvenlikten dolayı yapılan eklemeler (byte)
TinyOS	40	-
TinySec	45	5
MiniSec	43	3
DoSSec	43	3

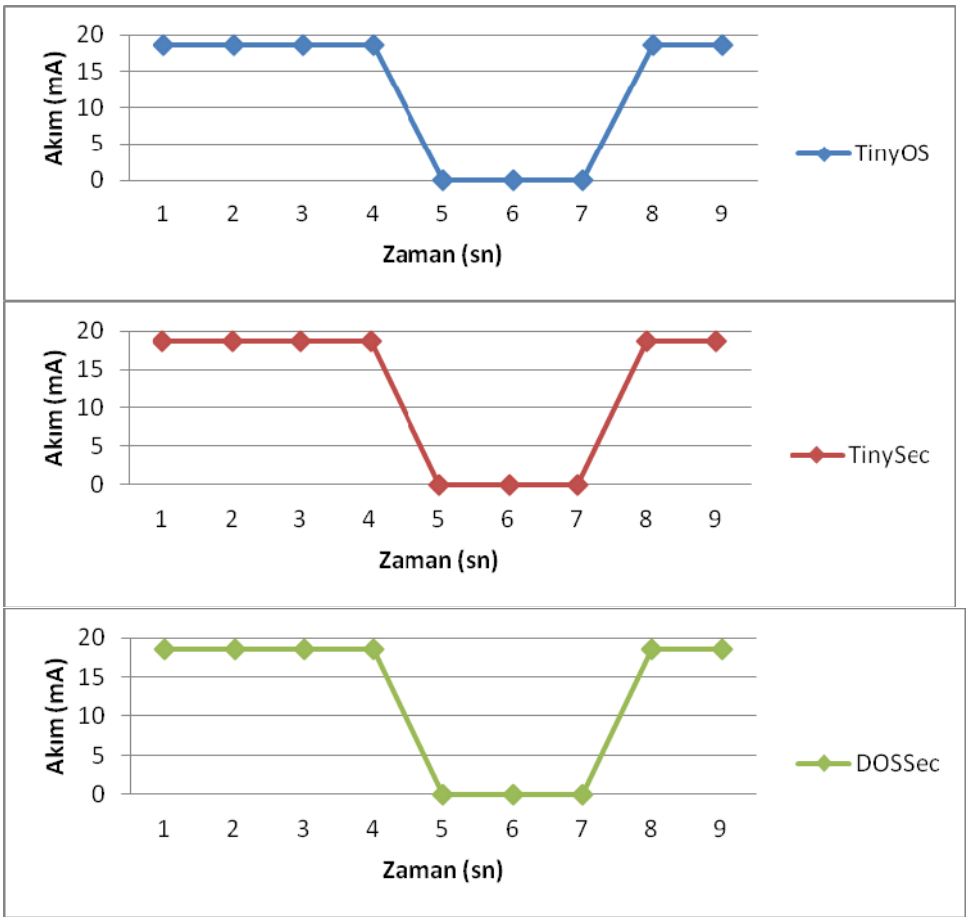
5.2.3. Enerji

Enerji ölçümlerini yapmak için dijital multimetre kullanılmıştır. Mevcut TinyOS kodu yüklü düğüm ve DoSSec protokolünün yüklü olduğu düğüm üzerinde ölçümler yapılarak sonuç bulunmak istenmiştir. Düğümler üzerinde Excel Alkaline bataryası takılıdır. TinySec ile karşılaştırma yapabilmek için ölçüm şekli olarak TinySec'in kullandığı işlem esas alınmıştır. İşlem şu şekildedir. Mevcut düğüm üzerinde akım ölçüm işlemi gerçekleşir. İlk değer not alınır (ilkdeger). Düğüm bu şekilde 1 saat çalışır. 1 saat sonlandıktan sonra tekrar ekrandaki değer alınır (sondeger). sondeger-ilkdeger işleminden çıkan sonuç bir saatte tüketilen toplam enerjiyi ifade etmektedir. Literatürde belirtildiği gibi [16] aynı gerilime sahip iki batarya enerji açısından karşılaştırılacaksa mAH değerinden ölçümler yapmak karşılaştırmanın doğruluğunu artıracaktır. mAH, bir saatte tüketilen toplam enerjiyi ifade etmektedir.

Çizelge 5.8. Enerji karşılaştırması

	Enerji (mAH)	TinyOS'a göre artan enerji miktarı
TinyOS	0.000160	-
TinySEC	0.000176	10%
DoSSec	0.000172	8%

Şekil 5.6.'da zamana göre ölçülen akım değerleri verilmiştir. Ölçülen akım 18,61 mA civarında olmaktadır.



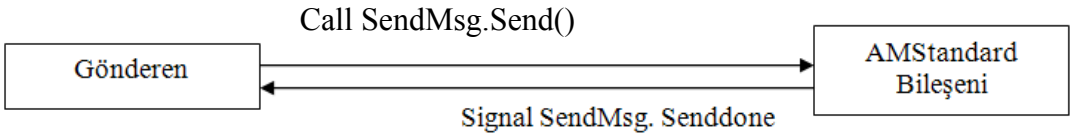
Şekil 5.10. Ölçülen akım değerleri

Şekil 5.10.'da görüldüğü gibi ölçülen akım değerleri miliAmper düzeyinde eşitlik göstermektedir. Geliştirilen protokollerde TinyOS'a göre artan enerji miktarı mikroAmper seviyesinde olmaktadır.

5.2.4. Gecikme

MiniSec [17] makalesinde gecikme sonuçlarına yer vermemiştir. TinySec [16] ise makalesinde TinyOS'a %7.9 luk gecikme kattığını belirtmektedir. DoSSec protokolünde gecikme sonuçlarını almak için aşağıdaki işlemler yapılmıştır. İşlemlerin kolay anlaşılması için Şekil 5.11.'de mesajı gönderen düğüm ve AMStandard bileşeni arasındaki bağlantı verilmiştir.

Düğüm, mesaj paketini hazırladıktan sonra SendMsg.Send fonksiyonunu çağırır. Gönderme işlemi tamamlandıktan sonra SendMsg.Senddone sinyali çağrılır.



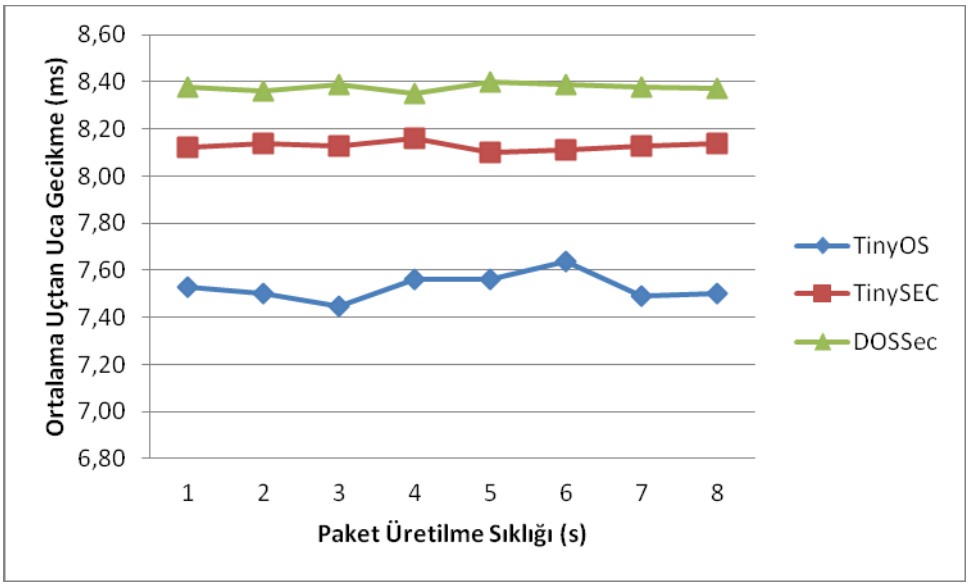
Şekil 5.11. SendMsg.Send fonksiyonu ve SendMsg.Senddone olayı

SendMsg.Send fonksiyonu ve SendMsg.Senddone olayı çağrılmadan önce sayaçlar eklenmiştir. SendMsg.Send fonksiyonu çağrılmadan önce sendsayac, SendMsg.Senddone olayı çağrıldığında ise senddonesayac eklenmiştir.
Düğümün gönderme anında kaybettiği zamanı bulabilmek için ise her gönderim sonunda senddonesayac – sendsayac ile oluşan farkları toplamak suretiyle 300 paket gönderimi sonucunda çıkan değer alınmıştır. 300 paket gönderildikten sonra sonucun alınmasının sebebi ortalama ve doğruya yakın bir değer elde etmek içindir. Ardından çıkan değer 300'e bölünmüş 7.53 ms değeri bulunmuştur.

Yukarıdaki işlemin aynısı birde XXTEA+OCB kısmı açılarak tekrarlanmıştır. Bulunan değer 8.38 ms'dir. TinyOS'a göre artan gecikme zamanı % 11.28 olmaktadır. Çizelge 5.9.'da DoSSec ve TinySec protokollerinin TinyOS'a göre artan gecikme miktarı verilmektedir.

Çizelge 5.9. TinyOS'a göre artan gecikme miktarı

	TinyOS'a göre artan gecikme miktarı
TinyOS	-
TinySEC	7.9%
DoSSec	11.28%



Şekil 5.12. Ortalama uçtan uca gecikme

Şekil 5.12'de görüldüğü gibi TinyOS'ta paket gönderimi esnasında ortalama 7.53 ms harcanmaktadır. Bu değer TinySEC'te 8.12 ms, DoSSec'te ise 8.38 ms'dir. Yani arttırılmış güvenliğe rağmen DoSSec yalnızca her paket gönderimi esnasında yalnıza 0.26 ms fark bir gecikme oluşturmaktadır. Ayrıca sonuçlar normalize edildiğinde, TinySec 1 kabul edilirse, DoSSec 1.03 olacaktır. Yani meydana gelen gecikme TinySec ile çok fazla bir fark oluşturmamaktadır.

5.2.5. Bellek kullanımı

Algılayıcı düğümlerde iki çeşit bellek türü vardır. Bunlar RAM ve ROM'dur. RAM, işlemcinin işleyeceği verilerin tutulduğu geçici bir depolama alanı iken; ROM ise TinyOS işletim sisteminin veya bir uygulamanın görevini yürütmesi için gereken

kalıcı bilgilerin tutulduğu depolama alanıdır. DoSSec ve TinySec protokollerine ait bellek kullanımı karşılaştırma sonuçları Çizelge 5.10.'da verilmiştir.

Çizelge 5.10. Bellek kullanımı karşılaştırması

	Bellek Kullanımı		TinyOS'a göre artan bellek kullanımı miktarı	
	ROM	RAM	ROM	RAM
TinySec	7146 byte	728 byte	5.45%	17.77%
DoSSec	2180 byte	493 byte	4.43%	4.81%

TinySEC Mica2 düğümü üzerinde uygulanmıştır. Mica2 düğümünün RAM kapasitesi 4K = 4 * 1024 = 4096 byte, ROM kapasitesi ise 128K = 128 * 1024 = 131072 byte'dır. TinySec RAM üzerinde % 17.77 lik, ROM üzerinde ise % 5.45 lik kullanım oluşturmaktadır.

DoSSec TelosB düğümü üzerine uygulanmıştır. TelosB düğümünün RAM kapasitesi 10K = 10 * 1024 = 10240 byte, ROM kapasitesi ise 48K = 48 * 1024 = 49152 byte'dır. DoSSec RAM üzerinde % 4.81 lik, ROM üzerinde ise % 4.43 lük kullanım oluşturmaktadır.

5.2.6. Genel değerlendirme

Çizelge 5.11'de görüldüğü gibi DoSSec protokolünde hem güvenlik artırılmış hem de KAA için önem teşkil eden enerji, bellek kullanımı gibi kriterler de TinySec'e göre iyi çıkmıştır. Gecikme olarak ta TinySec'e oranla ihmal edilebilir bir düzeyde artış göstermiştir. DoSSec'te anahtar boyutu olarak 80 bit yerine 128 bit kullanılmasına rağmen, TinySec'te bulunan 2 byte paket fazlalığı sonuçların bu şekilde çıkmasına neden olmuştur. Bununla birlikte OCB modunun, TinySec'te kullanılan CBC moduna göre daha hızlı ve daha az bellek gereksinimi duyması da neden olarak sayılabilir.

Çizelge 5.11. Genel değerlendirme

	DoSSec	TinySec
Anahtar Uzunluğu	128 bit	80 bit
Veri Gizliliği	+	+
Veri Bütünlüğü	+	+
Kimlik Doğrulama	+	+
Veri Tazeliği	+	-
Kullanılırlık	+	-
Güvenlikten dolayı yapılan eklemeler (byte)	3	5
TinyOS'a göre artan enerji miktarı	8%	10%
TinyOS'a göre artan gecikme miktarı	11.28% (8.38 ms – 1.03)	7.9% (8.12 ms - 1)
TinyOS'a göre artan bellek kullanımı miktarı (RAM)	4.81%	17.77%
TinyOS'a göre artan bellek kullanımı miktarı (ROM)	4.43%	5.45%

6. SONUÇLAR

KAA için gereken güvenlik gereksinimlerinden (Veri Gizliliği, Veri Bütünlüğü, Veri Tazeliği, Kimlik Doğrulama, Kullanılırlık) her birini karşılayan, ama her bir gereksinimi karşılamak için yüksek güvenlik, düşük enerji tüketimi düşüncesiyle güvenlik protokolü geliştirmek gerekmektedir. Ayrıca önerilen çoğu güvenlik çözümünün benzetim ortamında kalması, algılayıcı platformlar üzerinde önerilen çözümün değerlendirilmemiş olması yapılan araştırmalar için bir eksikliktir. Bu yüzden; önerilen protokollerin doğrudan güvenlik gerektiren uygulamalarda kullanılabilmesi için yalnız benzetim ortamında kalmaması, algılayıcı düğümler üzerinde de uygulamasının yapılması gerekmektedir.

Literatürde bulunan güvenlik çözümlerinden sadece TinySec ve MiniSec algılayıcı düğüm üzerinde uygulanmıştır. IEEE 802.15.4 ise Kablosuz Özel Alan Ağları için geliştirilmiş olmasına rağmen düşük güç tüketimi, düşük maliyet ve esnek oluşundan dolayı KAA'da kullanılmaktadır. Diğer güvenlik protokolleri düğüm üzerine uygulanmamıştır. TinySec ve MiniSec veri gizliliğini garanti etmek için 80 bit anahtar boyutlu Skipjack algoritmasını kullanmıştır. Yapılan araştırmalar göstermektedir ki veri gizliliği için anahtar boyutunun en az 128 bitlik olması gerekmektedir. TinySec mesaj tekrar yayınlama ataklarını önleyemezken, MiniSec'te verinin bütünlüğü garanti edilememektedir. Ayrıca bu protokoller KAA için güvenlik gereksinimlerinden kullanılırlık ilkesini sağlayamamaktadır. Kullanılırlık gereksiniminin karşılanmaması demek, o protokolün DoS saldırılarına karşı dayanıksız olması anlamına gelmektedir.

Düğüm üzerine uygulanan mevcut protokollerden TinySec ve MiniSec şifreleme algoritması olarak 80 bitlik Skipjack kullanmaları, kullanılırlık prensibini karşılamamaları ve bazı atakları önleyemedikleri için yeni bir güvenlik protokolü tasarlanmıştır.

Bu çalışmada, Kablosuz Algılayıcı Ağları için gereken güvenlik gereksinimlerinden Veri gizliliği, veri bütünlüğü, veri tazeliği, kimlik doğrulama ve kullanılırlık

prensiplerinden hepsini karşılayan ve bununla birlikte enerji verimli yeni bir veri bağı katmanı güvenlik protokolü geliştirilmiştir. Aynı zamanda, geliştirilen protokol algılayıcı düğümler üzerinde uygulanmıştır.

Protokole ait tasarım yapılmadan önce mevcut şifreleme algoritamaları ve şifreleme modları incelenmiş, KAA'nın sınırlı donanımsal kaynakları olduığundan dolayı, en uygun algoritmanın XXTEA, en uygun şifreleme modunun ise OCB olduğuna karar verilmiştir. Veri gizliliği, bütünlüğü, tazeliği, kimlik doğrulama gereksinimleri XXTEA+OCB ikilisi ile sağlanmıştır. Kullanılırlık prensibi için de Tespit+Savunma birimi geliştirilmiştir.

Geliştirilen protokolde hem güvenlik artırılmış hem de KAA için önem teşkil eden enerji, bellek kullanımı gibi kriterlerde TinySec'e göre daha iyi sonuçlar elde edilmiştir. Gecikme kriterinde ise TinySec'e oranla ihmal edilebilir bir düzeyde artış göstermiştir. Aynı zamanda geliştirilen protokol modüler bir yapıya sahiptir.

Bir güvenlik yaklaşımı geliştirirken, kablosuz algılayıcı düğüm kaynaklarının (bellek, işlemci, güç kaynağı) kapasitelerini göz önüne almak gerekmektedir. KAA uygulamalarında güvenliği artırmak için eklenen şifreleme mekanizmalarının düğüm enerji tüketim miktarlarını ve ortalama uçtan uca gecikme sürelerini arttırması beklenen bir sonuçtur. Burada uygulamaların ihtiyaçlarının iyi tespit edilmesi oldukça önemlidir. Basit bir geniş ölçekli çevre ya da endüstriyel KAA uygulamasında güvenlik çok fazla önem taşımazken enerji tüketiminin büyük önemi bulunmaktadır. Diğer yandan, askeri ve sağlık uygulamalarında ise güvenlik büyük önem taşırken düğüm enerji tüketimi nispeten göz ardı edilebilmektedir. Bu yüzden askeri ve sağlık uygulamalarında kullanılması için geliştirilen güvenlik çözümlerine uygun şifreleme algoritması, şifreleme modu seçmek oldukça önemli olmaktadır. Geliştirilen güvenlik çözümleri modüler olmalıdır. Yani, literatüre yeni giren şifreleme algoritması ve şifreleme modları güvenlik, enerji, bellek kullanımı, gecikme konularında daha iyi ise geliştirilen güvenlik çözümüne direk olarak entegre edilebilmelidir.

KAYNAKLAR

1. Akyıldız, I.F., Su, W., Sankarasubramaniam, Y., Çayırcı, E., "A survey on sensor networks", ***IEEE Communications Magazine***, 40(8), 102-114, (2002).

2. Özdemir, S., "Secure Data Aggregation In Wireless Sensor Networks Via Homomorphic Encryption", ***J. Fac. Eng. Arch. Gazi Univ.***, vol.23, no. 2, 365-373, (2008).

3. Chong, C-Y., Kumar, S.P., "Sensor Networks : Evolution, opportunities, and challenges", ***Proc IEEE***, Vol 91, No 8, 1247-1256, (2003).

4. Çakıroğlu, M., Özcerit, A.T., "Denial Of Service Attack Resistant Mac Protocol Design For Wireless Sensor Networks", ***J. Fac. Eng. Arch. Gazi Univ.***, vol.22, no. 4, 697-707, (2007).

5. Örencik, B., "Askeri İşbirlikli Nesne Ağlarında Güvenlik konulu araştırma raporu", ***İstanbul Teknik Üniv.,*** (2005).

6. Cam, H., Özdemir, S., Nair, P., Muthuavinashiappan, D., Sanli, H.O., "Energy-Efficient and secure pattern based data aggregation for wireless sensor Networks", ***Special Issue of Computer Communications on Sensor Networks***, 446-455, (2006).

7. Lin, R., Wang, Z., Sun, Y., "Energy Efficient Medium Access Control Protocols for Wireless Sensor Networks and Its State-of-Art", ***IEEE***, pp 669-674, (2004).

8. Wang, Y., Attebury, G., Ramamurthy, B., "A Survey Of Security Issues In Wireless Sensor Networks", ***IEEE Communications Surveys & Tutorials***, Volume 8, No. 2, 2nd Quarter, (2006).

9. Karaboğa, D., Ökdem, S., "Security Communication Techniques on Wireless Sensor Networks", ***Electronic Signature Semposium***, (2006).

10. Bandırmalı, N., Ertürk, İ., "Yeni Bir Kablosuz Algılayıcı Ağ Veri Bağı Katmanı Güvenlik Protokolü", ***Politeknik Dergisi***, Cilt 12 Sayı 4 s.235-242, (2009).

11. Kodaz, H., Cryptography for Security on Data Communication, M.Sc. Thesis, ***Selçuk University Institue of Science and Technology***, (2002).

12. İnternet: Wikipedia "CBC-MAC", http://en.wikipedia.org/wiki/CBC-MAC (2012).

13. İnternet: Ucdavis Computer Science "The OCB Authenticated-Encryption Algorithm", http://www.cs.ucdavis.edu/~rogaway/papers/draft-krovetz-ocb-00.txt (2012).

14. Mohit, S., "Security In Wireless Sensor Networks - A Layer Based Classification", ***Cerias Tech Report***, (2007).

15. Bandırmalı, N., Ertürk, İ., "WSNSec: A Scalable Data Link Layer Security Protocol for WSNs", ***Ad Hoc Networks***, (2011).

16. Karlof, C., Sastry, N., Wagner, D., TinySEC: A Link Layer Security Architecture for Wireless Sensor Networks, ***2nd ACM Conference on Embedded Networked Sensor Systems SENSYS 2004***, Maryland, USA, 162-175, (2004).

17. Luk, M., Mezzour, G., Perrig, A., Gligor, V., "MiniSec: A Secure Sensor Network Communication Architecture", ***ACM International Conference on Information Processing in Sensor Networks***, (2007).

18. IEEE-TG15.4, "Part 15.4: Wireless Medium Access Control (MAC) anf Physical Layer (PHY) Specifications for Low-Rate Wireless Personel Area Networks (LR-WPANs)", ***IEEE standart for Information Technology***, (2003).

19. Odabaşı, Ş. D., Zaim, A. H., "Kablosuz Sensör Ağlar ve Güvenlik Problemleri", ***3. Ağ ve Bilgi Güvenliği Ulusal Sempozyumu***, Ankara, (2010).

20. İnternet: Willow Technologies "TelosB Datasheet", http://www.willow.co.uk/TelosB_Datasheet.pdf (2012).

21. Khemapech, I., Duncan, I., Miller, A., "A survey of wireless sensor networks technology", ***Proceedings of the 6th Annual PostGraduate Symposium on the Convergence of Telecommunications Networking and Broadcasting***, (2005).

22. Hill, J.L., "System Architecture for Wireless Sensor Network," PhD Dissertation, ***University of California, Berkeley***, (2003).

23. Akyıldız, I.F. , Su, W., Sankarasubramaniam, Y., Çayırcı, E., "Wireless Sensor Networks: A Survey," ***Computer Networks***, (2002).

24. Feng, J., Koushanfar, F., Potkonjak, M., "System-Architecture for Sensor Networks Issues, Alternatives and Directions," ***ICCD'02***, (2002).

25. İnternet: USAMRMC Military Operational Medicine Research Program, "Warfighter Physiological Status Monitoring", http://www.momrp.org/publications/WPSM.pdf (2012).

26. Vieira, C.N., Coelho, J., Silva, D.C., Mata, J.M., "Survey on Wireless Sensor Network Devices," ***IEEE***, (2003).

27. Kavitha, T., Sridharan, D., "Security Vulnerabilities in Wireless Sensor Networks: A Survey", ***Journal of Information Assurance and Security***, vol 5, pp. 031-044, (2010).

28. Chan, H. , Perrig, A., "Security and privacy in sensor networks", ***IEEE Computer Magazine***, pp 103–105, (2003).

29. Carman, D.W. , Krus, P.S., Matt, B.J., "Constraints and approaches for distributed sensor network security", Technical Report 00-010, ***NAI Labs, Network Associates, Inc., Glenwood, MD***, (2000).

30. Meghdadi, M., Özdemir, S., Güler, İ., "Security in Wireless Sensor Networks: Problems and Solutions", ***International Journal Of Information Technologies***, vol 1, pp. 35-40, (2008).

31. İnternet: MEMSIC Powerful Sensing Solutions for a Better Life, "TelosB", http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html (2012).

32. Kumar, H., Sarma, D., Kar, A., "Security Threats in Wireless Sensor Networks", ***IEEE***, (2006).

33. Raymond, D.R., Midkiff, S.F, "Denial of Service in Wireless Sensor Network: Attacks and Defenses", ***IEEE Pervasive Computing***, Vol:7, Issue 1, PP: 74 – 81, (2008).

34. Bandırmalı, N., Ertürk, İ., "Increasing the Reliability of Security Protocols for WSNs", ***IEEE International Conference on Application of Information and Communication Technologies AICT 2009***, 1–5, Baku, Azerbaijan, (2009).

35. Özdemir. S., Wireless Sensor Network Security: A Comprehensive Overview, ***Journal of Politecnic***, pp 217-244, (2008).

36. İnternet: TinyOS, http://www.tinyos.net/ (2012).

37. İnternet: Wikipedia, "NesC", http://en.wikipedia.org/wiki/NesC (2012).

38. Erboral, S., "Kablosuz Duyarga Ağlarında Veri Birleştirilmesi Ve Değerlendirilmesi", Yüksek Lisans Tezi, ***İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü***, (2008).

39. Blumofe, R., "Cilk: An Efficient Multithreaded Runtime System", ***Proceedings of the 5th Symposium on Principles and Practice of Paralel Programming***, (1995).

40. Hu, J., Pyarali I., and Schmidt D., "Measuring the Impact of Event Dispatching and Concurrency Models on Web Server Performance Over High-speed Networks", ***Proceedings of the 2 nd Global Internet Conference IEEE***, (1997).

41. İnternet: TÜBİTAK UEKAE, "Açık Anahtar Altyapısı Eğitim Kitabı", http://www.kamusm.gov.tr/dosyalar/kitaplar/aaa/ (2012).

42. Keliher, L., Linear Cryptanalysis of Substitution-Permutation Networks, Ph.D. Thesis, ***Queen's University Canada***, (2002).

43. İnternet: Wikipedia, "DES", http://tr.wikipedia.org/wiki/DES (2012).

44. İnternet: Wikipedia, "AES", http://tr.wikipedia.org/wiki/AES (2012).

45. Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., Culler, D. E., "Spins: security protocols for sensor Networks", ***Wireless Networking***, vol.8, no.5, pp.521–534, (2002).

46. Rivest, R.L., "The RC5 encryption algorithm", ***In Proceedings of the 2nd Workshop on Fast Software Encryption***, pages 86-96, Springer, (1995).

47. Li, T., Wu, H., Wang, X., Bao, F., "SenSec Design, I^2R Sensor Network Flagship Project", Technical Report TR v1.0.

48. Devesh, C.J., Dhiren, R.P., Kankar, S.D., "Investigating and Analyzing the Light-weight ciphers for Wireless Sensor Networks", ***Journal of Computer Science***, (2009).

49. Wheeler, D., Needham, R. M., XXTEA: Corrections to XTEA. Technical report, ***Computer Laboratory, University of Cambridge***, (1998).

50. Levis, P., "Tossim: accurate and scalable simulation of entire tinyos applications"., ***SenSys '03***, pp. 126–137, (2003).

51. Titzer, B.L., Lee, D.K., Palsberg, J., "Avrora: a scalable sensor network simulation with precise timing", ***IEEE***, Piscataway, NJ, USA, IPSN '05. p. 67, (2005).

52. Bandırmalı, N., Ertürk, İ., Çeken, C., "Securing Data Transfer in Delay-sensitive and Energy-aware WSNs Using the Scalable Encryption Algorithm", ***IEEE International Symposium on Wireless and Pervasive Computing ISWPC 2009***, 288–293, Melbourne, Australia, (2009).

53. İnternet: National Institute of Standards and Technology, "Skipjack - a epresentative of a family of encryption algorithms as part of the NSA suite of algorithms", http://csrc.nist.gov/cryptval/des.htm (2012).

54. Rinne, S., Eisenbarth, T., Paar, C., "Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers", ***ECRYPT Workshop SPEED - Software Performance Enhancement for Encryption and Decryption***, Amsterdam, (2007).

55. Liu, S., Gavrylyako, O. V., Bradford, P. G., "Implementing the TEA algorithm on sensors", ***ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference***, pages 64–69, New York, NY, USA, (2004).

56. Großshädl, J., Tillich, S., Rechberger, C., Hofmann, M., Medwed, M, "Energy evaluation of software implementations of block ciphers under memory constraints", ***DATE '07: Proceedings of the conference on Design automation and test in Europe***, pages 1110–1115, San Jose, CA, USA, (2007).

57. Lee, J., Kapitanova, K., Son, S.H., "The price of security in wireless sensor networks", ***Computer Networks***, 54 pp 2967-2978, (2010).

58. Rehman, S., Bilal, M., Ahmad, B., Yahya K.M., Ullah, A., Rehman, O.U., "Comparison Based Analysis of Different Cryptographic and Encryption Techniques Using Message Authentication Code (MAC) in Wireless Sensor Networks (WSN)", ***IJCSI International Journal of Computer Science Issues***, Vol. 9, Issue 1, No 2, (2012).

59. Devesh, C.J., Dhiren, R., PatelKankar, S., Dasgupta, "Optimizing the Block Cipher Resource Overhead at the Link Layer Security Framework in the Wireless Sensor Networks", ***Proceedings of the World Congress on Engineering 2008 Vol I WCE 2008***, July 2 - 4, (2008).

60. IEEE Computer Society, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", ***IEEE Standards 802.11***, 195-200, (1999).

61. Ye, W., Heidemann, J., Estrin, D., "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", ***IEEE INFOCOM***, New York, Vol. 2, pp. 1567-1576, (2002).

62. Dam, T.V., Langendoen, K., "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", ***The First ACM Conference on Embedded Networked Sensor Systems (Sensys'03)***, Los Angeles, CA, USA, November, (2003).

63. Zheng, T., Radhakrishnan, S., Sarangan, V., "PMAC: An Adaptive energy-efficient MAC protocol for Wireless Sensor Networks", ***IPDPS '05: Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium***, (2005).

64. Singh, S., Raghavendra, C., "PAMAS: Power Aware Multi-Access Protocol with Signaling for Ad-hoc Network", ***ACM SIGCOMM Computer Communication Review***, (1998).

65. Rajendran, V., Obraczka, K., Gracia-Luna-Aceves, J.J., "Energy Efficient, Collision Free Medium Access Control for Wireless Sensor Networks", ***in ACM International Conference on Embedded Networked Sensor Systems (SenSys)***, pp. 181-192, (2003).

66. El-Hoiydi, A., "Aloha with Preamble Sampling for Sporadic Traffic in Ad-hoc Wireless Sensor Networks", ***in Proceedings of IEEE International Conference on Communications***, (2002).

67. Enz, C.C. , El-Hoiydi, A., Decotignie, J., Peiris, V., "WiseNET: An Ultralow-Power Wireless Sensor Network Solution", ***IEEE Computer***, Vol. 37, Issue 8, (2004).

68. Polastre, J. , Hill, J., Culler, D., "Versatile low Power Media Access for Wireless Sensor Networks", ***Proceedings of the 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04),*** Baltimore, MD, (2004).

69. Buettner, M. , Yee, G., Anderson, E., Han, R., "X-MAC: A Short Preamble MAC Protocol For Duty-Cycled Wireless Sensor Networks", Technical Report, ***Deparment of Computer Science University of Colorado at Boulder***, (2006).

70. Rhee, I., Warrier, A., Aia, M., Min, J., "Z-MAC: a Hybrid MAC for Wireless Sensor Networks", ***SenSys'05***, San Diego, California, USA, (2005).

71. Lin, P., Qiao, C., Wang, X., "Medium access control with a dynamic duty cycle for sensor networks", ***IEEE Wireless Communications and Networking Conference***, Volume: 3, Pages: 1534 - 1539, 21-25, (2004).

72. Yadav, R., Varma, S., Malaviya, N., "Optimized Medium Access Control for Wireless Sensor Network", ***IJCSNS International Journal of Computer Science and Network Security***, Vol. 8, No.2, pp. 334-338, (2008).

73. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E., "Spins:security protocols for sensor Networks", ***Wireless Networking***, vol.8, no.5, pp.521–534, (2002).

74. Park, T., Shin, K.G., "LiSP: A Lightweight Security Protocol for Wireless Sensor Networks", ***ACM Transactions on Embedded Computing Systems***, vol. 3(3), pp. 634–660, (2004).

75. Wood, A.D., Stankovic, J.A., "Denial of service in sensor networks", ***Computer Science***, 35(10), 54–62, (2002).

76. Koubaa, A., Alves, M., Tovar, E., "IEEE 802.15.4: a Federating Communication Protocol for Time-Sensitive Wireless Sensor Networks", ***Sensor Networks and Configurations: Fundamentals, Tecniques, Platforms and Experiments, Springer-Verlag***, Germany, pp 19-49, (2007).

77. İnternet : Crossbow Technology Inc., http://www.xbow.com (2010).

78. Shaikh, R.A., Lee, S., Khan, M.A., Song, Y.C., "LSec: Lightweight Security Protocol for Distributed Wireless Sensor Network", ***PWC***, pp 367-377, (2006).

79. İnternet: Sensor Network Simulator and Emulator (SENSE) http://www.cs.rpi.edu/~cheng3/sense/ (2012).

80. Tripathy, S., LISA: lightweight security algorithm for wireless sensor networks, ***Proceedings of the 4th international conference on Distributed computing and internet technology***, (2007).

81. ZigBee Alliance. Zigbee specification. Technical Report Document 053474r06, Version 1.0, ZigBee Alliance, (2005).

82. Leonard, E., Ren, L.J., Li. T., "An Energy Efficient Link-Layer Security Protocol for Wireless Sensor Networks", ***IEEE EIT***, (2007).

83. Wood, A., Stankovic, J., Son, S., "JAM: A jammed-area mapping service for sensor networks", ***24th IEEE Real-Time Systems Symposium***, pages 286- 297, (2003).

84. Ren, Q., Liang, Q., "Fuzzy logic-optimized secure media access control (FSMAC) protocol", ***CIHSPS***, (2005).

85. Brownfield, M., Gupta, Y., Davis, N., "Wireless sensor network denial of sleep attack", ***Systems, Man and Cybernetics (SMC) Information Assurance Workshop***, (2005).

86. Xu, W., Trappe, W., Zhang, Y., Wood, T., "The feasibility of launching and detecting jamming attacks in wireless Networks", **ACM MobiHoc '05**, (2005).

87. İnternet: Wikipedia, "XXTEA", http://en.wikipedia.org/wiki/XXTEA (2012).

88. Wheeler, D., Needham, R.M., "XXTEA: Corrections to XTEA Technical report", ***Computer Laboratory University of Cambridge***, (1998).

89. Schneier, B., "Applied Cryptography", Second Edition, John Wiley & Sons, Inc., New York, Ny, (1996).

90. Liu, S., Gavrylyako, O.V., Bradford, P. G., "Implementing the TEA algorithm on sensors" ***Proceedings of the 42nd annual Southeast regional conference***, pages 64–69, New York, NY, USA, (2004).

91. İnternet: Ucdavis Computer Science, "OCB Mode", http://www.cs.ucdavis.edu/~rogaway/ocb/ (2012).

92. İnternet: Wikipedia, "OCB Mode", http://en.wikipedia.org/wiki/OCB_mode (2012).

93. İnternet: Wikipedia, "Block Cipher Modes of Operation", http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation (2012).

94. Pauley, J.T., Bhatia, S.J, "American national standard for information systems—Data encryption algorithm—Modes of operation", ANSI X3.106-1983, American National Standards Institute, New York.

95. Rinne, S., Eisenbarth, T., Paar, C. , "Performance Analysis of Contemporary Light-Weight Block Ciphers on 8-bit Microcontrollers", ***ECRYPT Workshop SPEED - Software Performance Enhancement for Encryption and Decryption***, Amsterdam, (2007).

96. Großshädl, J., Tillich, S., Rechberger, C., Hofmann, M., and Medwed, M., "Energy evaluation of software implementations of block ciphers under memory constraints", ***DATE '07: Proceedings of the conference on Design, automation and test in Europe,*** pages 1110–1115, San Jose, CA, USA, (2007).

97. Devesh, C.J., Dhiren, R., Kankar, S.D., "Optimizing the Block Cipher Resource Overhead at the Link Layer Security Framework in the Wireless Sensor Networks", ***Proceedings of the World Congress on Engineering 2008 Vol I WCE 2008***, July 2 - 4, (2008).

98. Mohammadi, S., Jadidoleslamy, J., "A Comparison of Link Layer Attacks on Wireless Sensor Networks", ***International journal on applications of graph theory in wireless ad hoc networks and sensor Networks***, Vol.3, No.1, (2011).

99. Hong, S., Lim, S., Song, J., "Unified Modeling Language based Analysis of Security Attacks in Wireless Sensor Networks: A Survey", ***Ksii Transactions on Internet and Information Systems***, Vol. 5, No. 4, (2011).

100. Singh, S., Verma, H.K., "Security For Wireless Sensor Network", ***International Journal on Computer Science and Engineering***, Vol. 3 No. 6, (2011).

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı	: DENER, Murat
Uyruğu	: T.C.
Doğum tarihi ve yeri	: 10.05.1984 Yozgat
Medeni hali	: Bekâr
Telefon	: 0 (312) 202 37 28
Faks	: 0 (312) 202 37 10
e-mail	: muratdener@gazi.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet tarihi
Yüksek lisans	Gazi Üniversitesi Bilişim Enstitüsü Elektronik-Bilgisayar Eğitimi	2008
Lisans	Gazi Üniversitesi Teknik Eğitim Fakültesi Bilgisayar Sistemleri Öğretmenliği	2005

İş Deneyimi

Yıl	Yer	Görev
Aralık 2005 – Devam	Gazi Üniversitesi	Araştırma Görevlisi
Eylül 2005- Aralık 2005	Kastamonu Tosya Meslek Lisesi	Teknik Öğretmen

Yabancı Dil

İngilizce

Yayınlar

1. Dener, M., Bay, Ö.F., "Medium Access Control Protocols for Wireless Sensor Networks: Literature Survey", G.U. Journal of Science, Vol 25 (2): 455-564, (2012).

2. Dener, M., Akcayol, M.A., Toklu, S., Bay, Ö.F., "Genetic Algorithm Based a New Algorithm for Time Dynamic Shortest Path Problem", Journal of the Faculty of Engineering and Architecture of G.U, Vol 26, No 4, 915-928, (2011).

3. Dener, M., Bay, Ö.F., "Data Communication Between Motes on Wireless Sensor Networks", Sixth International Advanced Technologies Symposium, (2011).

4. Elmas, Ç., Orman, A., Dener, M., "Development of an Application for Information Trustworthiness on Internet", E-Journal of New World Sciences Academy Natural and Applied Sciences, 6(135-147), (2010).

5. Dener, M., Dörterler, M., Bay, Ö.F., "Software Development of a Purchasing Management for Sme's", E-Journal of New World Sciences Academy Natural and Applied Sciences, 3(267-278), (2010).

6. Bay, Ö.F., Dener, M., Dörterler, M., "Development of a Product Identification System and Software for Sme's", E-Journal of New World Sciences Academy Natural and Applied Sciences, 5(160-171), (2010).

7. Elmas, Ç., Orman, A., Dener, M., "Development of a Thesis Search Engine By Search Engine Optimizer", E-Journal of New World Sciences Academy Natural and Applied Sciences, 3(136-147), (2009).

8. Dener, M., Toklu, S., "Performance Evaluation of Dsdv and Dsr Manet Routing Protocols", Journal of Politecnic, Vol:12 No:3 pp.157-166, (2009).

9. Dener, M., Dörterler, M.,"Web Usability on Human Computer Interaction: An Example Application", Third International Computer and Educational Technologies Semposium, (2009).

10. Dörterler, M., Dener, M.,"Comparision Interaction Design of Learning Management System on Distanct Education", Third International Computer and Educational Technologies Semposium, (2009).

11. Dener, M., Dörterler, M., Orman, A., "Open Source Data Mining Softwares: Example Application on WEKA ", Eleventh Academic Informatics Conference, (2009).

12. Bay, Ö.F., Dener, M., Dörterler, M., "Development a Software for Materials Requirement Planning on SME's ", Fifth SME's and Efficiency Congress: Global Dynamics and SME's, (2008).

Projeler

1. Information Security and Management Systems, Leonardo da Vinci programme, Participant, Brussels, BELGIUM (1-21 April 2012).

2. Gazi Üniversitesi, Bilimsel Araştırma Projesi, Kablosuz Algılayıcı Ağlarda DoS Saldırılarına Dayanıklı bir MAC Protokolü Tasarımı ve Gerçekleştirilmesi, Araştırmacı, 2012.

3. Intensive English Program and Thesis Study, Participant, Georgia State University & Georgia Tech University, Atlanta,Georgia, U.S.A (30 July – 16 December 2011).

4. Improving the Quality of Education with Distance Learning System, Leonardo da Vinci programme, Participant, Lizbon, PORTUGAL (6-27 March 2011).

5. International Project Management Association (IPMA) World Congress, Organizing Committee Member, Istanbul, TURKEY. (1-3 November 2010).

6. Embed4Auto - Upskilling to Model-Based Software Development in the Automotive and Embedded Software Sector, Leonardo da Vinci programme, Participant, Thesseloniki, GREECE (18-23 September 2009).

7. Gazi Üniversitesi, Bilimsel Araştırma Projesi, KOBİ'ler için Ürün Kodlama Sistemi Yazılımının Geliştirilmesi, Araştırmacı, 2008.

İlgi Alanları

Kablosuz Algılayıcı Ağlar
Kriptoloji
Bilgi Güvenliği