# Düşük Enerjili Orta ve Büyük Ölçekli Kablosuz Algılayıcı Ağlar için Güvenli Yönlendirme Protokolü

# Secure Routing Protocol for Low-Power Medium/Large Scale Wireless Sensor Networks

Farzad Kiani[1]
[1]Bilgisayar Mühendisliği Bölümü, Mühendislik ve Doğa Bilimleri Fakültesi, İstanbul Sabahattin Zaim Üniversitesi, 34303, İstanbul, Türkiye
farzad.kiani@izu.edu.tr

## Öz

*Kablosuz Algılayıcı Ağlar, çok sayıda düşük güçlü sensör düğümünden oluşan ve değişkenlik gösterebilen ağlardır. Bu makalede, yeni anahtar yönetim metodu ile çeşitli saldırılara karşı iki güvenli yönlendirme protokolü önerilmektedir. Bu protokoller iki farklı ortamları için önerilmektedir. Birinci protokol küçük/orta ölçekli ağlara, ikinci protokol ise, büyük ölçekli ortamlar için tasarlanmaktadır. Her iki protokol, ağ oluşum fazı, güvenlik ve anahtarlama fazı ve veri transferi fazı olarak üç fazdan oluşmaktadır. Birinci ve üçüncü fazları aynı olan protokoller, sadece ikinci fazda farklılıklar göstermektedir. Bu iki fazda önerilen yeni sanal katmanı kullanarak ağ oluşturulmuş olacak ve veri transferi verimli olarak yapılacaktır. Birinci protokolde 'çoklu simetrik anahtar' tekniğinin geliştirilmesiyle sistemin yükü azaltılarak enerji tüketiminin verimli hale getirilmesi Wormhole, Sybil, Hello flood, DOS gibi yönlendirme saldırılarına karşı başarılı olmaktadır. İkinci protokolde ise, yalnızca küme başları/aktif düğümler ve anahtar sunucuları tarafından kullanılarak enerji tüketiminin verimli hale getirilmesi ve DoS, trafik analizi ve fiziksel saldırıları engellemiş olmaktadır. Geliştirilen protokoller, literatürdeki birkaç yöntemle karşılaştırıldığında performansta başarılı olduğu görülmektedir.*

***Anahtar sözcükler:*** *Telsiz ağlar, Güvenli yönlendirme protokolü*

Gönderim ve kabul tarihi : 21.06.2016 - 18.10.2016

*Wireless Sensor Networks consist of multiple dynamic sensor nodes. In this paper, two secure routing protocols will be developed in order to block the different attacks using new key management method. These protocols are suggested for two different environments. First protocol for small / medium sized networks, the second protocol is designed for large-scale environments. Each protocol has three phases as network deployment phase, security and key management phase and data transferring phase. The first and last phases are same in the protocols. In the two phases is used the suggested new virtual layer approach for network formation and data transferring as energy efficiency. The protocol is planned to make the system more efficient reducing the energy consumption by developing multiple symmetrical keys method. Also, it will be resistant to routing attacks such as Wormhole, Sybil, Hello flood, and DOS. Second protocol is planned to be used on large-scale networks' cluster heads (active nodes), and the server keys. It will also prevent the DOS, traffic analysis, and physical attacks. The developed protocol shows that successful performance compared to several methods in the literature.*

***Keywords****: Wireless sensor networks, Secure routing protocol*

## 1. GİRİŞ

Kablosuz Ad-hoc ağlar ailesinden olan Kablosuz Algılayıcı Ağlar (KAA), çok sayıda sensör düğümünden oluşan ve değişkenlik gösterebilen ağlardır. KAA'lar nem, sıcaklık, basınç gibi durumsal değişiklikleri takip edebilecek yapıdaki termik, manyetik ve görsel birçok farklı tipte

algılayıcı içerebilmektedir. Bu düğümlerin algıladığı bilgiler, baz istasyonunda toplanır ve oradan sunucuya iletilir. Sunucu tarafı merkezi veri toplama ve analiz sistemidir. Sunucu tarafında olan operatörler, ağ bileşenlerinin durumlarını sürekli izleyebilirler. KAA'lar, minimum konfigürasyonla kısa bir süre içerisinde kendi kendilerini yapılandırabilir ve geniş bir alanda kolayca yayılabilmektedir [1]. Günümüzdeki standardı IEEE 802.15.4 olan bu ağlar, pil üzerinden beslendikleri için minimum sistem kaynağı ile haberleşmesi istenen ağlardır. Bu ağlardaki düğümler, genellikle pilleri bitince kullanım ömürlerini doldururlar. Pil sorunu dışında sensör düğümlerinin hafızaları da sınırlıdır [2]. Algılayıcı düğümlerinin sınırlı işlemci gücü, hafıza, bant genişliği ve besleme kaynağı, KAA'larını çeşitli saldırılar karşısında riskli duruma getirebilir.

KAA'ların kısıtlı kaynaklarından dolayı (güvenli yönlendirme gerekçesi, güvenli yer tespit gerekçesi, sürekli topoloji değişiklik özelliği, güvenli anahtar oluşturulması, güvenli veri kümelemesi vb.) ad-hoc dışındaki ağların güvenlik mekanizmaları, bu ağlarda kullanılamaz [3]. Askeri ve tıbbi takip sistemleri gibi alanlarda bu ağların güvenlik konusu büyük önem taşımaktadır. KAA güvenliği; Anahtar Yönetimi (Key Management), Saldırı Tespiti (Intrusion Detection), Güvenli Yönlendirme (Secure Routing) ve Güvenli Konumlandırma (Secure Positioning) olarak dört başlıkta incelenmektedir [4]. KAA'nın dinamik yapısından, düğümlerin kolayca uzlaştırılabilmesinden ve kendi kendini örgütleme özelliğinden dolayı, anahtar yönetimi çok karmaşıktır. Limitli iletişim ve hesaplama kapasitesinden dolayı KAA'lar çeşitli saldırılara açıktır. Birçok durumda, bu ağların nasıl tasarlandığının hiçbir önemi olmadan, saldırganlar ağa sızmanın bir yolunu bulabilmektedir. Saldırı tespit sistemleri, bu saldırıları, kural dışı olay tabanlı tespit edebilmektedir. KAA, orta seviyedeki düğümün veri mesaj içeriğine ulaşmaya ihtiyaç duyduğundan dolayı veriyi iletmek için multi-hop yönlendirme ve kablosuz iletişim kullanmaktadır. Dolayısıyla, bu ağlar, birçok yönlendirme saldırı tiplerine maruz kalabilmektedir. Geleneksel uçtan uca güvenlik mekanizmaları, kablosuz ağ sistemlerinde iletişim sağlayamadığından yönlendirme güvenliğini ele alan birçok yaklaşım bulunmaktadır. KAA'da, düşman çevrelerdeki koordinatlara ulaşma gibi bazı uygulamalarda konum bilgisi çok önemlidir. Bu tür uygulamalarda, birçok yönlendirme protokolünde veya diğer güvenlik mekanizmalarında konum bilgisi veya komşu düğümler arasındaki uzaklık bilgisine ihtiyaç duyulmaktadır. KAA saldırıları, iç ve dış saldırılar olmak üzere ikiye ayrılmaktadır [5]. İç saldırılarda; saldırgan, algılayıcı düğümleri ve gizli anahtarı ele geçirerek saldırır. Dış saldırılarda ise saldırgan gizli anahtar bilgisine sahip olmadan kendi algılayıcı düğümleri ile hedefindeki algılayıcı ağın işlevini bozacak şekilde saldırılar düzenler. Bu ataklar, düğüm yakalama saldırısı, kanal saldırısı, servis reddi saldırıları, yönlendirme saldırıları, trafik analiz saldırısı ve Sybil saldırısı olabilir. Bu ataklardan en önemlisi ve en tehlikelisi ise servis reddi (Denial of Service-DoS) ataklarıdır. DoS atakları, KAA'dan beklenen görevin engellenmesi veya performansının büyük ölçüde düşürülmesi olarak tanımlanmaktadır. DoS atakları, geliştirilen sistem üzerinde KAA'nın yapısından dolayı kolayca gerçekleşebilmektedir. Genelde DoS atağında, saldırgan bir düğümü ele geçirerek ağdaki kaynakları tüketmek için gereksiz paketler gönderir ve diğer algılayıcı düğümlerin, ağın kaynaklarından veya servislerinden yararlanmasını engellemektedir. Ayrıca, DoS atakları ağın diğer katmanlarında da gerçekleştirilebilir. Fiziksel katmanda DoS atakları, gürültü ve sıkıştırma oluşturarak iletişimi engeller. DoS atakları bağlantı katmanında ise, çarpışma, yorma ve eşit davranmama seklinde gerçekleşir. Ağ ve yönlendirme katmanında, paket düşürme ve hatalı yönlendirme, kara ve solucan delikleri oluşturma seklinde DoS atakları vardır. Taşıma katmanında ise, senkronizasyonun bozulması seklinde DoS atakları bulunmaktadır. Bu ataklara karşı kaynak kullanımını ücretlendirme, güçlü kimlik doğrulama, trafik tanımlama ve anahtarlama yönetim metotları kullanılarak güvenli yönlendirme protokollerinin gerçekleştirilmesi mümkündür.

Bu ağlarda, verimliliği ve kullanılabilirliği hakkında birçok çalışma olmasına rağmen, söz konusu ağların güvenliği alanında yeterli çalışma bulunmamaktadır. Bu makalede, sunulan yeni anahtar yönetim metotları ve güvenli yönlendirme protokolleri veri gizliliği, kimlik doğrulama, veri bütünlüğü, kullanılabilirlik, veri güncelliği, zaman senkronizasyonu ve güvenli konumlandırma amaçlarını gerçekleştirecektir.

Bu makalenin ikinci bölümünde, literatürdeki çalışmalar incelenmektedir. Üçüncü ve dördüncü bölümlerde, önerilen protokollerin geliştirilmesi ve

performans analizleri anlatılmaktadır. Son bölümde ise sonuçlar ve öneriler yer almaktadır.

## 2. Literatür Çalışmaları

KAA'larda, güvenliği sağlayabilen en önemli teknikler, anahtar yönetimi ve şifreleme metotlarıdır [6]. Ancak, bu metotlar, güvenliği sağlamakla birlikte enerji ve hafıza harcamanın artırılmasına da neden olmaktadır. Bu yüzden, sunulan yöntemler, her iki parametreyi de dengeli olarak gerçekleştirmelidir. Şifreleme yöntemleri simetrik ve asimetrik olarak iki sınıfa ayrılmaktadır. Asimetrik kriptolama yöntemleri, hafıza ve enerji tüketiminde verimli olmadıklarından bu ağlarda kullanılması uygun görülmemektedir [7]. Bu yüzden, genelde KAA'lar için literatürdeki çalışmalar, simetrik yöntemler üzerinde gerçekleştirilmektedir. Bu simetrik yöntemler, iki kategoriye ayrılmaktadır.

### 2.1. Merkezi Yapı

Anahtar dağıtım işleminin gerçekleştirilmesi için merkezi bir yapı kullanmaktadır. Bu kategorideki en çok tanınmış çalışmalar, SPINS [8], SNEP [9] ve Karlof [10] protokolleridir. SPINS protokolü, gizlilik amacına ulaşmak için SNEP'ı, ayrıca, kimlik doğrulama özelliğini sağlamak için µTESLA [11] yöntemini kullanmaktadır. Ancak, bu protokoller, merkez tabanlı bir yapı kullandıkları için sistemin erken çökmesine yol açabilmektedir. Zira, merkezi mimariler yıldız topolojisine benzemektedir. Ayrıca, bu yapıdaki yöntemlerin ölçeklenebilirlik özellikleri sınırlı olup çeşitli saldırılara da açıktır.

### 2.2. Dağıtık Yapı

Simetrik anahtarlama yöntemlerinin ikinci kategorisi, dağıtık yapıyı kullanan protokollerdir. Bu kategoride, protokoller, dağıtık bir yapı içerisinde anahtar dağıtımını yapmaktadır. Bu kategori kendi içerisinde iki sınıfa ayrılmaktadır.

#### 2.2.1. Rastgele Dağıtık Yapı

Bu yapıda, her düğüm, komşusu ile iletişim kurmak için rastgele ortak anahtarlar üretmeye çalışmaktadır. Düğümlerin birbiriyle iletişime geçebilmesi için protokolün düğümlerde aynı anahtarı üretmesi gerekmektedir. Bu işlem aynı anahtarı üretene kadar devam etmek zorundadır. Bu yüzden, ağın fazla enerji tüketimine neden olmakta ve büyük ölçekli sistemlerde uygun görülmemektedir. Diğer yandan, bu sınıftaki protokoller, stabil bir duruma sahip değildir [12-16]. Bu sınıftaki yöntemlerde, "anahtar ön dağıtımı" ve "olasılıklı anahtar" teknikleri kullanılmaktadır. Random pairwise [15] protokolü, tek bir anahtar havuzu oluşturarak tüm düğümlere rastgele anahtarlar dağıtmaktadır. Koordinatör olarak seçilen bazı düğümler, tüm anahtar bilgilerine sahip olacak, dolayısıyla, bu düğümler, düşman tarafından ele geçirildiğinde sistemin güvenliği büyük bir riske girecektir. Bu sorunu gidermek için q-Composite [14] protokolü önerilmiştir. Bu protokol, düğümler arasındaki anahtar paylaşımında kimlik doğrulama işlemini tek taraflıdan iki taraflıya yükseltmiştir. Düğümler arasında tek taraflı ve çift taraflı kimlik doğrulama çeşitleri bulunmaktadır. Tek taraflı kimlik doğrulama tekniğinde alıcı düğüm, kendi kimliğini, gönderici düğüme doğrulaması yeterli olacaktadır. Ancak, çift taraflı metotlarda, iletişim kurmak isteyen her iki düğüm, kimliklerini birbirlerine doğrulaması gerekmektedir. q-Composite protokolü, belli periyotlarda anahtar yenileme mekanizmasını da kullanılmıştır. Ancak, tüm düğümlerin anahtar bilgileri birkaç seçilmiş düğümde tutulduğu için yine de bu seçilmiş düğümler, tehlikeye uğradıklarında sistem riske girecektir. Dolayısıyla, bu protokol, sadece birazcık düşmanın işini zorlaştırmış olacaktır. Bu sınıftaki diğer bir yöntem, Polynomial-based [13] protokolüdür. Bu protokolde, polinom anahtar havuzu kullanarak sistemin güvenliğini arttırmak hedeflenmiştir. Ancak, bu protokolde, sistemin hafıza kullanımı arttırılmış olup kötü niyetli düğümlerin bulunması da zorlaştırılmış olacaktır. Bu sorunu gidermek için Grid-based [16] protokolü önerilmiştir. Bu protokol, hamming distance ve payton fonksiyonelini kullanmıştır. Ancak, bu protokolün de fazla enerji tüketim sorunu devam etmektedir. Tüm bu yöntemler, sadece az sayıda düğüme sahip olan ağlarda ve enerjinin önemi olmayan ortamlarda kabul edilmektedir. Dolayısıyla, rastgele anahtar dağıtım yöntemi ve merkezi yapıya sahip protokoller, enerji sorunu olmayan küçük ölçekli sistemler içerisinde kullanılabilmektedir. Enerji sorunu olmayan ortamlar, düğümlere ait pillerin kolaylıkla şarj edilebileceği ortamlardır. Bu makalede, enerji sorunu olmayan ortamlar için önerilen protokol, bu kategorideki temel sorunları gidermek için sunulmaktadır.

### 2.2.2. Deterministik Dağıtık Yapı

Bu yapıyı kullanan protokoller, bahsedilen iki yapıdan, daha verimli bir şekilde sistemin amaçlarını ve güvenliğini sağlamaktadır. Genelde, deterministik yapıdaki protokoller, düğümler arası güvenli iletişimin sağlanması ve kimlik doğrulaması için master adlı anahtar kullanmaktadır. Bu yöntemlerde, dengeli olarak sistemin güvenliği fazla enerji ve hafıza kullanmadan gerçekleştirilebilmektedir. Bu yapıyı, bu makalenin iki protokolünde enerji tüketimi önemli olan ortamlar için kullanılmaktadır. Bu alanda, literatürdeki çalışmaların eksik olduğu da ayrıca görülmektedir. Deterministik yapıda en çok tanınmış çalışmalar, LEAP[17] ve LEAP+ [18] protokolleridir. Bu protokoller, küçük ve orta ölçekli sistemler için sunulmuştur. Ancak, bu protokollerin de bazı sorunları vardır. Bu çalışmanın birinci protokolü, bu yöntemlerin üzerinde odaklanıp eksik noktalarının giderilmesi için bir yönlendirme mekanizmasının oluşturulmasını planlamaktadır. LEAP protokolünde, ortak ikili anahtar üretiminin tek taraflı olması güvenlik eksikliklerine yol açmaktadır. Örneğin "u" düğümü, "v" düğümü ile haberleşmek istediğinde sadece "v" düğümü, kimlik doğrulama işlemi yapması gerekmekte ve "u" düğümünün, kendi kimliğini doğrulatması gerekmemektedir. Dolayısıyla, bu sorun, LEAP'ın birinci eksik yönüdür. Bu makalenin birinci protokolünde, çift taraflı kimlik doğrulama tekniğini kullanarak bu sorunun giderilmesi planlanmaktadır. LEAP'ın ikinci eksik yönü, baz istasyonundan veri paketini alan düğümlerin, baz istasyonu kapsama alanında olmayan düğümlere, multi-hop şeklinde, deşifreleme ve şifreleme gibi işlemlere gerek duyularak veri transferinin gerçekleştirilebilmesidir. Bu yüzden, bu teknik, fazla enerji tüketimine yol açmaktadır. Ayrıca, LEAP protokolünde, tüm master anahtarlar, sadece baz istasyonu tarafından tahsis edilmektedir. Bu yüzden, baz istasyonu, düşman eline geçerse sistemin güvenliği riske girecektir. Deterministik yapılarda bu teknik, birçok protokolde kullanılmıştır [17-19]. Bu sorun, LEAP'ın üçüncü eksik yönüdür. Diğer taraftan, baz istasyonu ile düğümler arası mesafe, enerji harcama açısından da önem taşımaktadır. Dolayısıyla, uzak mesafelerde, düğümlerin fazla enerji ve hafıza kullanmaları kaçınılmaz olacaktır. Master anahtar dağıtım ve baz istasyonundan uzak mesafelerde bulunan düğümlerin sorunlarını gidermek için literatürdeki yeni çalışmalarda, bir hiyerarşik yapının kullanılması önerilmektedir. Bu yapı sayesinde, protokoller, baz istasyonunun bazı görevlerini, her kümenin küme başı elemanlarına dağıtmış olacaktır. Ancak, bu teknikte, küme başı elemanlarına fazla yük aktarıldığı için enerji verimliliği konusunda sıkıntılar devam etmektedir. Bu makalede önerilen birinci protokol, bu sorunu temel olarak giderip optimize enerji tüketimini sağlayabilmek için hiyerarşi yöntemi yerine yeni sanal katman yapısını geliştirilecektir. Bu yeni sanal katman yapısı, bu alanda yeni bir yöntem ortaya koyacaktır.

Küçük/orta ölçekli sistemler için deterministik ve hiyerarşik tabanlı çalışmalarda en çok tanınan protokol, BROSK[21] protokolüdür. Bu protokol, tamamen oturum ve müzakere tabanlı bir yöntemdir. Bu yöntemde de LEAP gibi, iki düğüm arası anahtar tahsisi için baz istasyonundan verilen master anahtarı kullanılmaktadır. Dolayısıyla, bu protokol, baz istasyonunun düğümlere uzak olduğu durumlarda enerji verimliliğini kaybetmektedir. LEAP ile farkı ise, bir tane master anahtarı kullanmasıdır. Bir tane master anahtarının düşman eline geçebilme ihtimali daha yüksek olduğu için LEAP protokolü BROSK'a göre daha güvenlidir.

Literatürde, küçük/orta ölçekli ağlar için tasarlanmış birçok çalışma büyük ölçekli ağlara uygulandığında düğümlerin enerji verimliliğinin azaldığı görülmektedir. Makaledeki önerilen ikinci protokolde, büyük ölçekli ağlarda anahtarlama yönteminin yalnızca küme başları/aktif düğüm ve anahtar sunucuları tarafından kullanılarak enerji tüketiminin verimli hale getirilmesi planlanmaktadır. Literatürdeki büyük ölçekli sistemler için dağıtık yapı çerçevesinde tasarlanmış olan önemli çalışmalardan birisi, LİSP [20] protokolüdür. Bu protokol, enerji verimliliğini sağlayabilmesi için her oturumda yeni anahtar tahsis etmeyip belli periyotlarda bu anahtarları atamaktadır. Bu yöntem, kümeleme yapısını kullandığı için literatürdeki başarılı yöntemlerden biridir. Ancak, bu protokolde, küme başı elemanlarına fazla yük aktarıldığı için enerji verimliliği konusunda sıkıntılar devam etmektedir. LİSP protokolünün diğer eksik yönü ise, küme başı elemanlarının değişmemesidir. Bu yüzden, saldırgan, bu düğümleri ele geçirerek tüm sistemi tehdit edebilmektedir. Dolayısıyla, önerilen ikinci protokolde de sanal katman yapısının kullanılması öngörülmektedir. Bu yüzden, LİSP ve benzeri

protokollerin sorunları giderilmiş olacaktır. LİSP protokolünde, düğümlere tahsis edilmiş olan anahtarların yenileme zamanı gelene kadar, düğümlerin aynı anahtarları kullanmaları gerekmektedir. Dolayısıyla, LEAP ve BROSK'a göre anahtarları daha uzun süre düğümlerde tutması gerekmektedir. Bu yüzden, hafıza kullanımı da verimli olmayacaktır. Önerilen ikinci protokol, bu sorunları gidermek için sanal katman yapısında anahtarları bulunduran aktif düğümler, belli periyotlarda enerji seviyelerine ve baz istasyonlarından uzaklıklarına göre mod değişimini gerçekleştirmektedir. Aktif düğümler uyku haline geçip, uyku halinde olan düğümler ise aktif hale geçip görevi devralacaktır. Dolayısıyla, enerji ve hafızanın verimli kullanılması sağlanacaktır.

Literatürdeki birçok protokol, genelde, güvenliği sağlamak için sadece anahtar dağıtımı yapmakta olup veri transferi ve yönlendirme konusu ile ilgili çok odaklı çalışma yapmamaktadır. Bu çalışmada, yeni anahtarlama yöntemleri kullanılarak düğümler arasında güvenli iletişimin ve veri transferinin sağlanması da planlanmaktadır. Bu doğrultuda, iki yeni güvenli yönlendirme protokol farklı ortamlar için önerilmektedir.

## 3. Orta/Büyük Ölçekli Ortamlar İçin Güvenli Yönlendirme Protokolü

Literatürdeki çalışmalarda genelde güvenlik hedefleri farklı kategorilerde incelenmektedir [8, 12]. Bu makalede ise, birincil ve ikincil olmak üzere iki sınıfta kategorize edilmektedir. Birincil hedefler, gizlilik, bütünlük, kimlik doğrulama ve kullanılabilirlik gibi güvenlik kriterleridir. İkincil hedeflerde ise, veri güncelliği, zaman senkronizasyonu ve güvenli konumlandırma gibi etkenlerdir. Birçok uygulamada, düğümler yüksek hassasiyetli verileri veya gizli anahtarları taşıdıklarından dolayı data paketlerini başkalarına sızdırmamaları gerekmektedir. Bu makalede, hassas verileri gizli tutmak için veriyi sadece ilgili alıcının ulaşabileceği gizli bir anahtarlama yöntemi ile şifrelemek ve düğümler arasında güvenli kanalların oluşturulması düşünülmektedir. Ayrıca, bu veriler, pasif saldırganlardan gizli tutulması gerekmektedir. Ağ üzerinden iletilen herhangi bir mesajın gizli kalabilmesi için veri gizliliği özelliğinin sağlanması gerekmektedir. Genelde, veri gizliliği; trafik analiz ve fiziksel saldırganların hedefidir. Bir hiyerarşi yapıda güvenli kanal (TDMA ve CDMA) ve şifreleme yöntemlerini kullanarak gizlice dinlemelerin engellenmesi, anahtarlama yöntemlerini ağaç tabanlı topolojilere gömerek (pre-distribution, post-security) fiziksel saldırıların engellenmesi ve düğümlerin kimlik gibi genel bilgilerini kripto ederek, trafik analizi saldırılarının engellenmesi ön görülmektedir. Düşman/kötü niyetli düğümlerin, yetkisiz erişimlerini engellemek için ise kimlik doğrulaması gerekmektedir. KAA, paylaşımlı kablosuz ortam kullandıklarından dolayı, kötü niyetli algılayıcı düğümleri veya sahte paket tespitleri için doğrulama mekanizmalarına ihtiyaç duymaktadır.

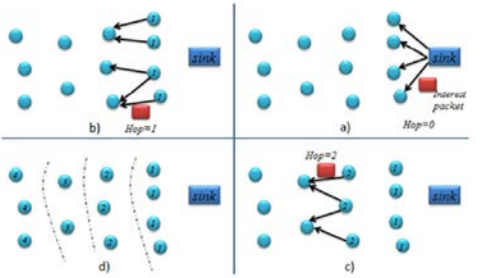
Bu makalede, iç ve dış saldırılar için iki farklı kimlik doğrulama yöntemi izlenmektedir. İç saldırılara karşı, düğümler arasında gizli anahtarların paylaşılmasında simetrik mekanizmaların kullanılması ön görülmektedir. Dış saldırılarda ise saldırı tespit yöntemi kullanılması planlanmaktadır. Bu yönetim mekanizmasında, sistem ve kullanıcı faaliyetleri monitör olunur, sistemin konfigürasyonu ve anormal aktivite desenleri analiz edilir, sistem ve dosya bütünlüğünün değerlendirilmesi de yapılır. Veri bütünlüğü, verilerin gizli tutulup doğrulaması yapıldıktan sonra otomatik olarak güvenilir yönlendirme protokolü ile sağlanmaktadır. Güvenilirliği sağlamak için olay-tabanlı ve sorgu-tabanlı yöntemler kullanılmaktadır. Kullanılabilirlik hedefine ulaşmak için çok sayıda düğüm kullanmak ve enerji tasarruflu yöntemler (sleep/wake-up vb. yöntemler [3]) kullanılabilmektedir. Kullanılabilirlik özelliği, ağ servislerini, DoS saldırılarına karşı korur. Veri güncelliğini sağlamak için her veri paketine adım sayısı (hop) ve yaşam süresi (TTL) parametreleri eklenmektedir. Bu yüzden, ağ içerisinde aynı paketlerin sürekli tekrarlanması engellenmiş olup enerji ve hafıza verimliliği de sağlanmış olacaktır. Zaman senkronizasyonu hedefini sağlamak için paket delay (gecikme) parametresini protokole ekleyerek ağdaki gecikmeler takip edilmektedir. Paketlerin ACK'lerinde gecikme yaşandığı takdirde, protokol, bu durumu fark edip tekrar paketin gönderilmesine karar verecektir. Ayrıca, gecikme kaynağı da analiz edilebilmektedir. Bu makaledeki son hedef, güvenli konumlandırma özelliğini sağlamaktır. Genelde, bu özelliği gerçekleştirmek için GPS konum tespit sistemi, sensör düğümlerine entegre edilmesi gerekmektedir. Konum tespit özelliğine rağmen, saldırganlar güvenli olmayan konumlara hatalı sinyaller rapor ederek kolayca o bölgedeki

düğümleri manipüle edebilecektir. Dolayısıyla, konum tespiti dışında lokasyonların da güvenli olması gerekmektedir. Diğer yandan, her düğüme monte olan GPS, sistemin maliyetini artıracaktır. Bu makale kapsamında geliştirilmiş yeni yazılımsal yöntemle bu özellik GPS'e ihtiyaç duyulmaksızın gerçekleştirilmektedir.

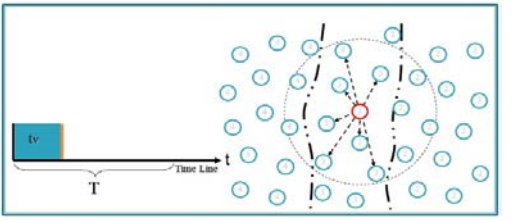
Bu makalede iki farklı ortam için iki protokol çözüm olarak önerilmektedir. Birinci olarak, küçük/orta ölçekli ortamlar için bir protokol tasarlanmaktadır. Diğeri ise, büyük ölçekli ağlar için ikinci protokol geliştirilmektedir. Her iki protokol üç fazdan oluşmaktadır. Birinci, ağ topolojisinin oluşumu, ikinci faz ise, anahtarlama ve şifreleme yöntemlerinin uygulaması ve son fazda, veri transferi yapılmaktadır.

### 3.1. Ağ Oluşturma Fazı

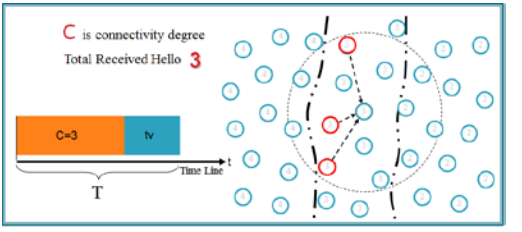
Bu makalenin ana teması, yeni güvenli anahtarlama yönlendirme protokollerinin geliştirilmesidir. Doğru anahtarlama sayesinde kimlik doğrulamaları yapılıp yeni anahtar yönetim metotları ile güvenli yönlendirme protokolü geliştirilerek iç ve dış saldırıların engellenmesi amaçlanmaktadır. Bu amaçlara ulaşmak için önerilen her iki protokolde sanal katman yapısı kullanılmaktadır. Bu yeni sanal Katman oluşumunda düğümler ve baz istasyonu arasındaki hop sayısını belirleyerek her düğüm hangi katmanda bulunduğu tespit edilmektedir. Bu doğrultuda, baz istasyonu tarafından "Interest Online" mesaj paketi, tüm düğümlere broadcast (yayın) edilmektedir. Bu paketin ilk değeri sıfır olarak tanımlanmaktadır (HOP=0). Baz istasyonunun radyo alanında olan düğümler, bu mesajı aldıktan sonra HOP ismindeki değişkenin sayısını 1 arttırarak bu mesajı tekrar diğer komşularına gönderecektir. Bu aşamada, yeni mesajı alan düğümler, HOP değişkeninin sayısını 1 arttırarak HOP değerini 2' ye yükseltmiş olup aynı şekilde kendi komşularına mesajı ileteceklerdir. Dolayısıyla, bu mesaj, tüm düğümlere ulaştığında her düğümün bir HOP değeri olacak ve HOP değerleri aynı olan düğümlerin aynı katmanda oldukları anlamına gelecektir. Düğümlerin baz istasyonuna olan uzaklıkları HOP değerlerine göre belirlenmektedir. Tüm bu işlemler şekil 1'de özetlenmiştir. Örneğin, HOP değeri 2 olan düğümün ikinci katmanda bulunduğu ve baz istasyonuna iki adım uzaklıkta olduğu anlamına gelmektedir. Dolayısıyla, bu protokolde, GPS' e ihtiyaç duymadan düğümlerin konumları tespit edilecek ve sistemin enerji harcaması da verimli olacaktır. Ağın enerji verimliğini arttırmak için her katman birkaç düğümün aktif kalması yeterlidir (şekil 2). Bu aktif düğümler komşu katmanlardaki düğümlerle her zaman iletişimde olmaları gerekmekte olup belli periyotlar arasında uyku modunda olan düğümlerle yer değiştirmeleri gerekmektedir. Protokolümüz, bu zaman aralığını T adında parametre ile gerçekleştirmektedir. T zamanı dolunca yeni bir oturum açılmış olup her katmandaki aktif düğüm, uyku moduna geçip uyku modundaki ise aktif moduna geçer. Aktif kalması gereken düğüm sayısı ise, Connection değişkeni ile gerçekleştirilmektedir (şekil 3).




**Şekil-1: Sanal katman oluşumu**




**Şekil-2:** Aktif düğüm $t_v$ kadar bekledikten sonra kendi katmanındaki komşularına HELLO mesajı yayınlamaktadır




**Şekil-3:** Connection değeri kadar komşulardan cevap gelince aktif düğüm uyku moduna geçmektedir

## 3.2. Güvenlik Fazı

### 3.2.1. Düşük Güçlü Enerji Kaynağına Sahip Küçük/Orta Ölçekli Ağlar İçin Protokol

Merkezi ve rastgele dağıtık yapıları kompleks ve ağır anahtarlama yöntemlerini kullandıkları için enerji tüketimi konusunda genelde verimli değillerdir. Bu nedenle, bu makaledeki protokoller verimli enerji tüketimini dikkate alarak güvenliği sağlamaktadır. Önerilen birinci protokolde 'çoklu simetrik anahtar' tekniğinin geliştirilmesiyle sistemin yükü azaltılarak enerji tüketiminin verimli hale getirilmesi planlanmaktadır. Bu teknik Wormhole, Sybil, Helloflood gibi yönlendirme saldırılarının başarı olasılıklarını düşürüp sistem yönetimini de kolaylaştırmaktadır. Alternatif yol bulması ve yeni enerji tasarruflu yönlendirme algoritması, bu protokolün diğer özgün değerlerindendir. Literatürdeki LEAP [17] ve bu alanda birçok çalışmanın kümeler arası haberleşme ve anahtarlama eksikliğini de akıllı ve çoklu küme başı dağıtım yöntemiyle giderilmesi planlanmaktadır. Genelde, bu yöntemlerde, master anahtar, baz istasyonundan tüm düğümlere dağıtılmış olup onların fazla enerji ve hafıza tüketimine neden olmaktadır. Ayrıca, baz istasyonun kendisi kötü niyetli biri tarafından ele geçilirse tüm sistem tehlikeye girecektir. Bu protokolde, iki taraflı kimlik doğrulama tekniği ve dinamik baz istasyonlarının kullanılması düşünülmektedir. Böyle ortamlarda, baz istasyonu ile düğümlerin mesafeleri enerji tüketimi açısından önem taşımaktadır. Dolayısıyla, uzak mesafelerde, düğümlerin fazla enerji ve hafıza kullanılmaları kaçınılmaz olacaktır. Bu sorunları gidermek için literatürdeki yeni çalışmalar hiyerarşik yapının kullanılmasını önermektedir. Ancak, bahsedildiği gibi, bu hiyerarşik yapı da çok fazla başarıya erişememektedir. Bu sorun, yeni sanal katman yöntemi (şekil 1.) ile giderilecek olup enerji verimliliği de arttırılacaktır.

Birinci protokolde, güvenliği ve enerji verimliliği arttırmak için yeni anahtarlama mekanizması aşağıdaki gibi sunulmaktadır.

Çoklu simetrik anahtar tekniği kullanılarak Wormhole ve Sybil gibi saldırılar engellenmiş olacaktır. Bu teknikte, düğümler ile baz istasyonu arasındaki iletişim için "kişisel anahtarlar", bir düğüm ile komşuları arasında iletişimin kurulabilmesi için "düğümler arası anahtarlar", düğümlerin katmanlar arası iletişimi için "katman anahtarları", ve yayın (broadcast) paylaşımları için "genel anahtarlar" olmak üzere dört çeşit anahtar kullanılacaktır. Anahtarların tahsisi için aşağıdaki aşamaların gerçekleştirilmesi planlanmaktadır.

1. Kişisel Anahtarlar: Düğümler ortama dağıtılmadan önce her düğüme seçkin anahtar tahsis edilecektir. Bu anahtarların bilgileri baz istasyonunda veya sunucuda tutulacaktır. Her sensör düğümüne seçkin anahtar tahsis etmek için aşağıdaki formülün kullanılması düşünülmektedir.

$$K_u^m = f\ K_s^m(u)$$

$K_u^m$ ifadesi; "u" düğümüne master (m) anahtarı tahsis edilmesi anlamına gelmektedir. Bu master anahtar, kontrolör rolü oynamakla birlikte düğümlere benzersiz anahtar tahsisi için kullanılmaktadır. $f$ ifadesi ise, sözde-rastgele (Pseudo-Random) fonksiyonudur [17]. Bu protokolde, her sensör düğüme benzersiz bir "id" tanımlanacaktır. Tüm anahtarların üretimi için sadece bir master anahtarın kullanılması hafıza tüketiminde verim sağlayacaktır. Bu teknik, baz istasyonu ile düğümler arası iletişimin gerçekleştirilebilmesi için kullanılacaktır.

2. Düğümler arası Anahtarlar: Her düğüm ile bir adım mesafede olan komşuları arasında güvenli iletişimin sağlanabilmesi için bu yöntemin kullanılması planlanmaktadır. Kontrolör, başlangıç anahtarını ($K_1$,$K_2$,..) düğümlere tahsis edecektir.

$$K_u = f K_1(u)$$

Daha sonra, protokol, düğümlere, komşuları ile iletişime geçebilmeleri için belli bir zamanı hazırlık süresi olarak tanıyacaktır. İki düğüm arasında veri transferi yapılmadan önce kimlik doğrulama işleminin gerçekleştirilmesi gerekmektedir. Kimlik doğrulama işlemi ve veri transferi arasındaki geçen sürece "hazırlık süresi" denilmektedir. Tüm düğümler, tanınan hazırlık sürecinde, kendi komşularını bulmak ile yükümlüdür.

$$\text{Node } u \leftarrow T_{min}\ (\text{ready})$$

Bu süre içerisinde, düğümler "HELLO-discover" mesajını broadcast ettikten sonra cevap bekleyip radyo kapsama alanında olan düğümlerden cevap alınca onları, komşu listelerine ekleyecektir. Kimlik doğrulama işlemi ise master anahtar tekniği ile gerçekleştirilecektir.

$u \rightarrow * : \text{Nonce}_u$

Bilgisayar bilimlerinde özellikle veri güvenliğinde, Nonce terimi "number used once" kelimesinin baş harflerinden oluşmakta olup anlam olarak bir sefer kullanılan sayı demektir. Her HELLO-discover mesajı, bir tane Nonce sayısını kapsamaktadır.

$v \rightarrow u: v, \text{MAC}(K_v, \text{Nonce}_u \mid v)$

"u" düğümü, komşularından gelen cevap ile (örneğin "v" adlı komşusundan) kendisi ve komşusu arasında ikili ortak anahtar üretecektir.

$K_{uv} = f K_v(u)$

"u" düğümüne tanımlanmış olan süre bitince $K_1$ anahtarı ve tüm komşularına discovery aşamasında tahsis edilmiş master anahtarlarının silinmesi gerekmektedir. Bu işlem, her oturum için geçerlidir. Bu sayede, sistemin güvenliği artacaktır. Her oturumun sonunda tahsis edilmiş ortak ikili anahtarlar silinecektir. Dolayısıyla, herhangi bir düğüm, tehlikeye düştüğünde diğer düğümlerin güvenliği korunmuş olacaktır.

3. Katmanlar arası Anahtarlar: Bazen düğümlerin komşuları, başka katmanın üyesi olabilir. Bu durumda, acil olarak iki farklı kümede olan komşu düğümler arasında anahtarlama işlemi yapılacaktır. Bu durumda, "u" düğümü rastgele anahtar ($K_u^C$) üreterek şifreleme işlemini gerçekleştirecektir.

$u \rightarrow v_i : (K_u^C)\, k_{UVi}$

Daha sonra, "$v_i$" düğümü, $K_u^C$ anahtarlarını deşifre ederek onları bir tabloda kaydedecektir. "u" düğümünün herhangi bir komşusu herhangi bir nedenden dolayı devre dışı bırakılırsa (düşman eline geçerse veya pili biterse), "u" düğümü, katman anahtarını iptal edip veya yenileyip diğer komşularına da haber verecektir. Bu sayede, düğümlerin anahtar tabloları, güncelliğini korumuş olacaktır. "u" düğümü, kendisinin ve komşusunun katman numarasına göre aynı kümede olup olmadığını anlamaktadır. Eğer, aynı katmanda ise, düğümler arası anahtarlama tekniği, değilse katmanlar arası anahtarlama tekniği kullanılacaktır.

4. Genel Anahtar: Bazen ihtiyaca göre, yeni görevler, veri paketleri şeklinde baz istasyonundan ağın tüm düğümlerine aktarılabilmektedir. Böyle durumlarda, genel anahtarlama tekniği ile gönderilen görev paketinin güvenliğini garanti altına almak mümkündür. Anahtarlama işlemi baz istasyonundan yapılacaktır. Bu paket, baz istasyonun kendi anahtarı ile ($K_{all}^m$) şifrelenip hop-by-hop olarak tüm düğümlere gönderilecektir. Enerji verimliliğinin sağlanabilmesi için baz istasyonundan gönderilen paketlerin iletim işlemini, küme başları üstlenecektir. Paketi $K_c^m$ olarak şifreleyen küme başı, bulunduğu küme üyesi olan diğer düğümlere bu paketin gönderimini devam ettirecektir. Dolayısıyla, ağdaki tüm düğümler, paket iletimi için sürekli deşifreleme ve şifreleme işlemini yapmak zorunda kalmayacaktır. Çünkü bu işlemi, artık küme başı elemanları yapacaktır. Ancak, küme başı görevini üstlenen düğümler fazla enerji tüketecektir. Protokol, bu görevi belli periyotlarda diğer küme üyelerine devrederek bu sorunu gidermiş olacaktır.

### 3.2.2. Düşük Güçlü Enerji Kaynağına Sahip Büyük Ölçekli Ağlar İçin Protokol

Literatürde, küçük/orta ölçekli ağlar için tasarlanmış birçok çalışmanın büyük ölçekli ağlara uygulandığında enerji verimliliğinin azaldığı görülmektedir. Önerilen ikinci protokolde, büyük ölçekli ağlarda anahtarlama yönteminin yalnızca küme başları/aktif düğüm ve anahtar sunucuları tarafından kullanılarak enerji tüketiminin verimli hale getirilmesi planlanmaktadır. Dolayısıyla, ağın üzerindeki anahtarlama ve şifrelemeden kaynaklı olan overhead azaltılacaktır. Bu protokol, literatürdeki LİSP [20] ve benzeri çalışmaların [23,24] küme başına bağımlılığını gidererek daha ölçeklenebilir olup özgünlüğünü arttırmış olacaktır. Büyük ölçekli sistemlere tasarlanan protokol sadece belli periyotlar içerisinde yeni anahtarlar atayacaktır. Dolayısıyla, bu yöntem, sanal katman yapısı sayesinde baz istasyonu veya küme başlarının yükü hafifletilmiş olup enerji tasarrufu sağlamakla birlikte güvenlik risk olasılıkları da düşecektir. Bu yapıda, anahtarları bulunduran aktif düğümler, belli periyotlarda enerji seviyelerine ve baz istasyonlarından uzaklıklarına göre mod değişimini gerçekleştirmektedir. Dolayısıyla, hafızanın verimli kullanılması da sağlanacaktır. Ayrıca, DoS, trafik analizi ve fiziksel saldırılara da daha dayanaklı olacaktır. Bu yapıdaki aktif düğümler, küme başı gibi rol oynayacak ancak hiyerarşi yapısının aksine tek bir eleman üzerine düşen yük/görev artmayacaktır.

Bu protokolde kullanılan anahtarlama teknikleri aşağıda anlatılmaktadır.

1. Anahtarlama yöntemi, yalnızca küme başları/aktif düğümler ve anahtar sunucularına uygulanarak enerji verimliliğini büyük ölçekli ağlarda sağlamış olup DoS saldırılarını da engellemiş olacak,

2. ACK'ların gönderilmesini gerektirmeyen etkin anahtar yayını kullanıp trafik analiz saldırılarını önlemiş olacak,

3. Veri mesajına eklenmeksizin oluşturulan doğruluk bitlerini kullanıp fiziksel saldırılarını önlemiş olacaktır. Ayrıca erişim kontrolü, ağa girişlerin kontrol edilmesi ve anahtar yenileme özelliklerini de sağlamış olacaktır.

Bu protokolde, enerji tasarrufu için basit hashing algoritmaların kullanılması planlanmaktadır. Ayrıca, bu protokol gereğince kaynak kullanımı fazla olduğundan sanal katmanlı bir yapı kullanarak bu fazlalığı minimize edip enerji verimliliğinin optimum seviyede kalması düşünülmektedir. Büyük ölçekli ağlarda enerji tüketiminin artışı kaçınılmazdır. Ancak kontrollü bir şekilde tüketilmesi için düğümler arası iletişimlerde yeni bir oturumun açılması gerekmemektedir. Bu yüzden, anahtarlar her oturumda değil, belli periyotlarda yenilenmesi öngörülmektedir. Dolayısıyla, süreli anahtar yenileme tekniği sayesinde, enerji verimliliği optimize edilmiş olacaktır. Bu protokolde, farklı katmanlara ait olan düğümler için aynı anahtarlar kullanılabilmektedir. (Örneğin, "A" katmanındaki düğümlere tahsis edilmiş anahtarlar, diğer katmanlarda yer alan düğümlerin haberleşmesinde de kullanılabilmektedir). Ayrıca, küme başı (aktif düğüm) olma görevi belli periyotlarda diğer düğümlere devredilebildiğinden dolayı yeni seçilen küme başları için geçici anahtar listeleri güncellenecektir. Bu protokol, deterministik ve stabil bir anahtarlama yöntemi önermektedir. Bu doğrultuda,

A) Paketleri kriptolamak için $T_k$ adlı geçici anahtar tahsis edilecektir.

B) Tahsis edilen $T_k$ anahtarlarını unicast olarak tüm düğümlere göndermek için master anahtarının kullanılması gerekecektir. Bu master anahtarı, $K_s$ olarak kullanılacaktır. "s" baz istasyonu veya küme başı elemanı/aktif düğümü olabilir.

C) Kümelere yeni eklenen düğümlerin kimlik doğrulama işlemi, geçici anahtar tahsisi, küme içindeki diğer düğümlere haber verilmesi ve kümenin anahtar tablosunun güncellenmesi görevini küme başı elemanı üstlenecektir.

Bu protokolde, en kritik olan kısım, önceki aşamada tahsis edilen $T_k$'ların yönetimidir. Bu yüzden, yeni $T_k$'ların tahsis edilmesinde güvenli ve güvenilir bir yöntemin kullanılması gerekmektedir. Bu güvenliği sağlamak için gizlilik ve kimlik doğrulama işlemlerinin gerçekleşmesi gerekmektedir. Güvenilirliğin sağlanması için ise kaybolan $T_k$'ları restore edilmesi gerekmektedir. Ayrıca, bu yeni $T_k$'ların tahsis edilme sürecinde veri transfer işleminin aksamaması gerekmektedir. Bu yüzden, zaman senkronizasyon [26] algoritmalarına ihtiyaç duyulmaktadır. Bu doğrultuda,

A) Bir $T_k$ dizisinin oluşturulması gerekmektedir. Bu işlemi gerçekleştirmek için tek taraflı kriptolama yöntemi kullanılması planlanmaktadır. Önerilen bu yöntem, S/key[27] tekniğine benzemektedir.

B) Kriptolama öncesi, uygun ve güvenli $T_k$ dağıtımının yapılması gerekmektedir. Bu görevi, baz istasyonu veya küme başı/aktif düğüm, kontrolör paketler ile üstlenecektir. Bu paket, aşağıdaki parametreleri içermektedir.

1. t (Anahtarın arabellek boyutu): t parametresinin büyüklüğüne göre arabellek (buffer) boyutu da belirlenecektir. Dolayısıyla, hata dayanıklılık seviyesi de artmış olacaktır.
2. İnitial $T_k$: Başlangıç anahtarı olarak kullanılmaktadır.
3. $T_k$ yenileme aralığı: Kaybolan Tk'lara karşı tekrar gönderim yapmadan anahtar kullanım sürecidir.
4. $T_{refresh}$: Yeni $T_k$'lar için kullanılacaktır. Bu parametrenin değeri yenileme aralığından küçük olması gerekmektedir. Bu yüzden, paket iletimi ve anahtar yenileme süresindeki gecikme oranları optimum seviyede tutulacaktır. Ayrıca, çarpışmalar da engellenmiş olacaktır.

Tüm bu parametrelerin kullanılması aşağıdaki formülde gösterilmektedir.

$$K_s \rightarrow m: fMK_m (t \mid T_k \mid T_{refresh})$$

C) Tk-buffering tekniğinin tüm düğümler üzerinde uygulanması gerekmektedir.

D) Tahsis edilen $T_k$'ların onayı ve kaybolan $T_k$'ların keşif ve geri kurtarma işlemlerinin gerçekleştirilmesi gerekmektedir. Tahsis edilen $T_k$'ların kontrolü, oluşturulmuş diziye bağlıdır. $K_s$, sistemin yapılandırılması aşamasında, bu diziyi oluşturacaktır.

$$T_{ki} = H(T_{ki+1}) , i<n ; \{ T_{ki} \mid i=1...n\}$$

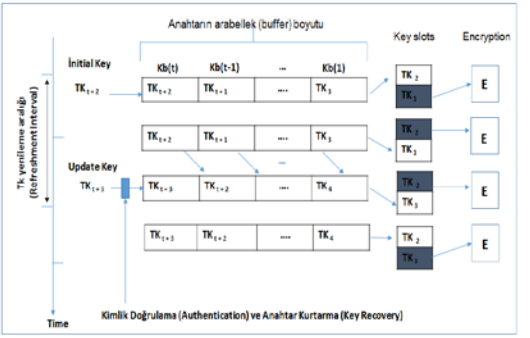
n değerinin büyük sayı olması gerekmektedir (örneğin n=100).

$K_s$, dizi oluşturma görevi dışında anahtar yenileme görevini de üstlenecektir. Bu işlem, aşağıdaki formül kullanılarak gerçekleşecektir.

$$K_s \rightarrow m: fMK_m (t \mid T_{kt+2} \mid T_{refresh}) \mid MAC (t \mid T_{kt+2} \mid T_{refresh})$$

Bu protokolde, kimlik doğrulama işlemi, yöntemin içerisinde gizli olarak gerçekleştirilmiş olup kontrol paketlerinin boyutu da yüksek oranda azalmış olacaktır. Dolayısıyla, enerji tasarrufu sağlanmış olacaktır. Ayrıca, bu paketlerin yönetimi için üç farklı anahtarın kullanılması planlanmaktadır.

1- Başlangıç anahtarları (İnitial key): Yeniden anahtar üretimi için KS tarafından kullanılacaktır.
2- Güncel anahtarlar (Update key): Periyodik anahtar dağıtımı için kullanılacaktır.
3- İstek anahtarları (Request key): Küme-içi düğümlere, periyodik anahtar dağıtımında herhangi bir anahtar atanmadığı taktirde bu tip anahtarlar kullanılacaktır.




**Şekil-4:**Tahsis edilen $T_k$'ların yönetim aşamaları [21]

## 3.3. Veri Transferi Fazı

Literatürdeki birçok protokol, genelde, güvenliği sağlamak için sadece anahtar dağıtımı yapmakta olup veri transferi ve yönlendirme konusu ile ilgili çok odaklı çalışma yapmamaktadır. Bu çalışmanın her iki protokolü için veri transfer fazında aynı strateji uygulanmaktadır. Güvenli enerji tasarruflu algoritma geliştirilerek ağ iletişimindeki sistem yükü azaltılacaktır. Düğümlerin sürekli aktif olmalarının yerine belli periyotlarda uyku moduna çevirerek; düğümler verilerini göndermeden birleştirerek veri fazlalığı ve tekrarlanmaları engelleyerek bu amaca ulaşılmak mümkün olmaktadır.

Bu çalışmada, sanal katman dışında, aşağıdaki tekniklerin kullanılması veri transferinde verimli bir şekilde enerji tüketimini sağlayacaktır.

1. Graf (graph) ve süzme (percolation) teorilerini sanal katmanda kullanarak düğümler arasındaki uyku/aktif vardiyaları akıllı olarak kontrol edilecektir. Bu yöntemde, aktif modunda olan düğüm, belli süre sonra komşularına görevinin bittiğini bildirecek ve en az C sayısı kadar komşusundan gelen cevap ile uyku moduna geçebilecektir. Bu yöntemin gerçekleştirilmesinde özel olarak T (time period) ve C (connection value) adlı değişkenler kullanılacaktır. Bu haberleşme modelini gerçekleştirebilmek için graf ve süzme tekniklerinin kullanılması planlanmaktadır.
2. Literatürdeki birçok çalışma, uyku modundan aktif moduna geçme tekniğini yazılımsal olarak kullanmış ve bu projeler sadece

simülasyon aşamasında kalmıştır. Uyku/aktif mod değişimi, gerçek alanlarda yazılımsal boyutundan ziyade donanım ve elektronik bilimlerine ihtiyaç duyulmaktadır. Bu makalede, önerilen mekanizma, iyi kısa-kapsam taşıyıcılarından birisi olan TR1000 modülü [3] kullanılarak gerçek uygulama alanlarındaki mod değişimini sağlayacaktır. TR1000 modülü kullanıldığında uyku halinde olan düğümün gönderici ünitesi uyku modunda olup sadece alıcı ünitesi aktif kalacaktır. Bu modülde, uyku halinde olan düğümün alıcı ünitesi, gönderici ünitesine göre ¼ oranında enerji harcayacaktır. Dolayısıyla, bu modül sayesinde, daha az enerji tüketilecektir.

3. Katmanlar arası aktif düğümlerin haberleşebilmesi için alternatif yollar kullanılacaktır. Dolayısıyla, katmanlar arasındaki kopukluk ihtimali veya kötü niyetli düğümlerin oluşma riski ciddi derecede düşecektir. Ayrıca, hem güvenilirlik hem de fault tolerance özelliği optimize edilmiş olacaktır.
4. Veri transferinde güvenliği yüksek seviyede tutabilmek için düğümler arası haberleşmede Kapsayan Ağaç (Spanning Tree) tekniği kullanılacak aynı zamanda enerji tasarrufu da sağlanmış olacaktır.

## 4. Performans Analizi

Bu makalede önerilen her iki protokolümüz, eşit parametrelerle geliştirilmiştir. Bu parametreler değerleriyle birlikte Tablo 1'de gösterilmektedir. Bu protokoller, OMNET++ [28] ile simülasyon yapılmış ve literatürdeki en iyi üç protokol ile karşılaştırılmıştır. Bu deneyde MEMSIC Professional-Sensör KİT'i [29] kullanılmıştır. Küçük/orta ölçekli ortamlar için İstanbul Sabahattin Zaim Üniversitesi (İSZÜ) mühendislik fakültesinde önerilen protokolümüz ile birlikte LEAP, LEAP+ ve BROSK protokolleri programlanarak MEMSIC sensör kitlerin üzerine gömülü olarak çalıştırılmış ve sonuçları, OMNET++ simülasyon yazılımından elde edilmiştir. Karşılaştırma sonuçları ise şekil 5'de gösterilmektedir. Büyük ölçekli ortamlar ise, ikinci protokolümüz, LISP, Password ve SABR protokolleri İSZÜ yerleşkesinde gerçekleşerek denetilmiştir. Kullanılan donanımlar ve yazılımlar ise küçük/orta ölçekli ortamlardakilerin aynısıdır. Karşılaştırma sonuçları ise şekil 6'de gösterilmiştir. Tüm protokollerin gerçekleştirilmesi için gerekli input parametreler, Tablo 1'deki değerler ile kullanılmıştır.

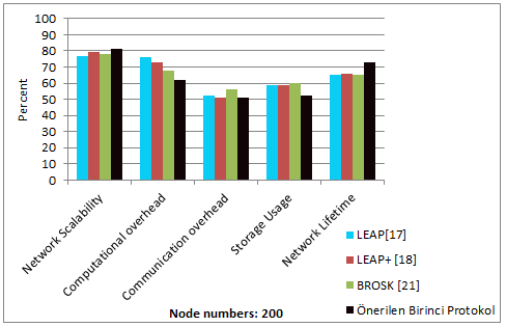
Birinci protokol, düşük güç kaynaklarına sahip küçük/orta ortamlar için geliştirilmiş ve çalışma performansının analizi için literatürdeki [17, 18, 21] çalışmalar ile karşılaştırılmıştır. İkinci protokol ise büyük ölçekli ortamlar için uygulanmış ve çalışma performansının analizi için literatürdeki [20, 24, 25] çalışmalar ile karşılaştırılmıştır.

**Tablo 1. Tüm protokollere uygulanan input parametrelerin adı ve değerleri**

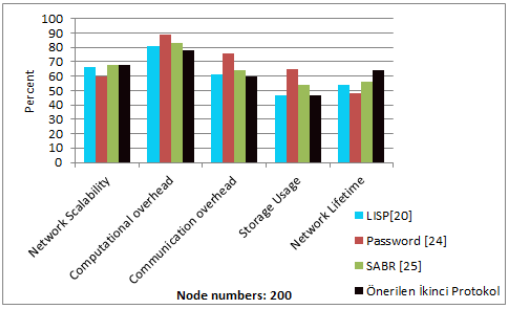
| *Parametre adı* | *Parametre değerleri* | *Parametre adı* | *Parametre değerleri* |
|---|---|---|---|
| Initial (max) energy | 0.9 J/bit | Receive buffer size | 10000 bytes |
| Radio/ Sensor energy consumption | 30 nJ/bit | Send buffer size | 10000 bytes |
| Transmit process cost | 30 nJ/bit | Deployment area size | (600 x 600) m |
| Receive/sense process cost | 5 nJ/bit | Send/receive buffer counts | 25 |
| Data packet size | 400 bytes | Sink/BS position | (600 x 300) m |
| Sensing Radius | 7.5 m | Transmission Radius | 15 m |

Geliştirilen protokollerimizin sonuçlarını diğer literatürdeki protokoller ile çeşitli çıktı-parametreleri (output parameters) ele alarak karşılaştırılmıştır. Bu parametreler, ölçeklenebilirlik (scalability), hesaplama yük fazlalığı (computational overhead), haberleşme yük fazlalığı (communication overhead), ve ağın ömrüdür (network lifetime). En iyi performansa sahip olan protokol, ölçeklenebilirlik ve ağın ömrü parametrelerde yüksek olasılığı sahip aynı zamanda diğer üç parametrede en düşük değerleri olan protokolüdür. Elde edilen sonuçlara göre 100% bir iyileşme söz konusu değil, ancak, bu makalede önerilen protokoller önerilen ortamlar için kabul edilebilir performansa eriştikleri görünmektedir. Ayrıca, sonuçlara göre sanal katman yapısında uygulanan protokollerimiz bu iyileştirmede büyük bir rol oynamaktadır. Sanal katman yapısında anahtarları bulunduran aktif düğümler, belli periyotlarda enerji seviyelerine ve baz istasyonlarından uzaklıklarına göre mod değişimini

gerçekleştirmektedir. Aktif düğümler uyku haline geçip, uyku halinde olan düğümler ise aktif hale geçip görevi devralacaktır. Dolayısıyla, enerji ve hafızanın da verimli kullanılması sağlanmıştır. Şekil 5'de makalede önerilen birinci protokolü, 200 sensör düğümle çalıştırılmış ve diğer üç protokol ile sonuçları karşılaştırılmıştır. Şekil 6'de ise aynı işlem önerilen ikinci protokolümüz için yapılmış ve büyük ölçekli ortamlara önerilen en uygun üç protokol (LISP, Password ve SABR) ile karşılaştırılmıştır.




**Şekil-5:** Orta ölçekli ortamlar için anahtar yönetim protokollerin karşılaştırılmalı performans analizi




**Şekil-6:** Büyük ölçekli ortamlar için anahtar yönetim protokollerin karşılaştırılmalı performans analizi

## 5. Sonuçlar ve Öneriler

Bu makalede, yeni anahtar yönetim metodu ile iç ve dış saldırılara karşı güvenli yönlendirme protokolü geliştirildi. Önerilen protokolde birinci fazda ağ oluşturulması için iki faklı yapı (sanal katman ve kümeleme) ile gerçekleştirildi ve farklı sonuçlar elde edildi. İkinci fazda anahtarlama, şifreleme ve saldırıları engellemek için veri gizliliği, kimlik doğrulama işlemi, veri bütünlüğü, kullanılabilirlik, veri güncelliği, zaman senkronizasyonu ve güvenli konumlandırma amaçların hepsini kapsayan protokolümüz ölçeklenebilirlik, hesaplama yük fazlalığı, iletişim yük fazlalığı, hafıza kullanımı ve ağın ömrü olarak beş farklı parametre ile sonuçları gösterildi. Elde edilen sonuçlar, önerilen protokolün iyi performans sergilediğini gösterdi.

Public/Private anahtarlarının tahsis edilme işlemi, baz istasyonu tarafından yapıldığı zaman güvenlik sorununun ortaya çıkma ihtimali halen devam etmektedir. Çünkü baz istasyonu, düşman tarafından ele geçirilirse tüm sistem tehlikeye düşecektir. Bu yüzden, projede önerilen tüm protokoller, anahtar tahsis işlemleri için küme başı elemanlarının, birden fazla baz istasyonunun, mobil sensör düğümlerinin, mobil baz istasyonlarının ve/veya İnsansız Hava Araçlarının (İHA) [30] kullanılması düşünülmektedir.

## KAYNAKLAR


[1] Kiani, F., Amiri, E., Zamani, M., Khodadadi T., AbdulManaf, A. Efficient Intelligent Energy Routing Protocol in Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, 2015(1), pp.1-14, 2015.

[2] Akyildiz, I. F., Sankarasubramaniam Y., Cayirci, E. *Wireless Sensor Networks: A Survey*, *Journal of Computer Networks*, 38(4), pp.393-422, 2003.

[3] Kiani, F. *Designing New Routing Algorithms Optimized for Wireless Sensor Network*, LAP LAMBERT Academic Publishing, Dusseldorf, Germany, 2014.

[4] Alrajeh, N.A., Khan, S., Shams, B. *Intrusion Detection Systems in Wireless Sensor Networks: A Review*, International Journal of Distributed Sensor Networks, 2013(1), pp.1-7, 2013.

[5] Ahmed, M., Hung, X., Sharma, D. *A Taxonomy of Internal Attacks in Wireless Sensor Network*, World Academy of Science, Engineering and Technology International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering, 6(2), pp.203-206, 2012.

[6] Mansour, I., Ghalhoub, G., Lafourcade, P. *Key Management in Wireless Sensor Networks*, J. Sens. Actuator Netw., 4, pp.251-273, 2015.

[7] Si, L., Ji, Z., Wang, Z. *RETRACTED: The Application of Symmetric Key Cryptographic Algorithms in Wireless Sensor Networks*, International Conference on Solid State Devices and Materials Science (Elsevier), 25, pp.552-559, 2012.

[8] Macedonio, D., Merro, M. *A Semantic Analysis of Key Management Protocols for Wireless Sensor Networks*, Sci. Computer Program, 1, 81, pp.53-78, 2014.

[9] Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., Culler, D. *SPINS: Security Protocols for Sensor Networks*, Wireless Networks, 8(5), pp.521-534, 2002.

[10] Karlof, C., Sastry, N., Wagner, D. *TinySec: A Link Layer Security Architecture for Wireless Sensor Networks*, International Conference on Embedded Networked Sensor Systems (SenSys), *ACM*, 1, pp.162-175, 2004.

[11] Du, W., Deng, J., Han, Y.S., Chen, S., Varshney, P. *A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge*, IEEE InfoCom, 1, pp.13-28, 2004.

[12] Anjum, F. *Location Dependent Key Management Using Random Key-Pre distribution in Sensor Networks*, Proceedings of the 5th ACM workshop on Wireless security (WiSe 06), 5, pp.21-30, 2006.

[13] Chan, H., Perrig, A, Song, D. *Random Key Pre-Distribution Schemes for Sensor Networks*, IEEE Symposium on Research in Security and Privacy, 3, pp.197-213, 2003.

[14] Huang, J.M., Yang, S., Dai, Ch. *An Efficient Key Management Scheme for Data-Centric Storage Wireless Sensor Networks*, International Conference on Electronic Engineering and Computer Science, 4, pp.25-31, 2013.

[15] Du, X., Xiao, Y., Guizani, M., Chen, H.H. *An Effective Key Management Scheme for Heterogeneous Sensor Networks*, Elsevier Ad Hoc Network, 5(1), pp.24-34, 2007.

[16] Ning, P., Li, R., Liu, D. *Establishing Pairwise Keys in Distributed Sensor Networks*, ACM Transaction Information System Security, 8(1), pp.41-77, 2005.

[17] Zhu, S., Setia, S. Jajodia, S. *LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks*, Conference on Computer and Communications Security, 10, pp.62-72, 2003.

[18] Zhu, S., Setia, S. Jajodia, S. *LEAP+: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks*, Journal ACM Transactions on Sensor Networks (TOSN), 2(4), pp.500-528, 2006.

[19] Wang, N., Fang, S. *A Hierarchical Key Management Scheme for Secure Group Communications in Mobile Ad Hoc Networks*, Journal of System Software, 80(10), pp.1667-1677, 2007.

[20] Ramox, A., Filho, R.H. *Sensor Data Security Level Estimation Scheme for Wireless Sensor Networks*, Sensors, 15(1), pp.2104-2136, 2015.

[21] Jemai, A. Mastouri, A. Eleuch, H. *Study of Key Pre-Distribution Schemes in Wireless Sensor Networks: Case of BROSK (use of WSNet)*, International Journal Applied Mathematics & Information Sciences, 5(3), pp.655-667, 2011.

[22] Younis, M.F., Ghumman, K., Eltoweissy, *M. Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks*, IEEE Transaction Parallel Distributed Systems, 17(8), pp.865-882, 2006.

[23] Qiu, Y., Zhou, J. Y., Baek, J. Authentication and Key Establishment in Dynamic Wireless Sensor Networks, *Sensors*, 10(4), pp.3718-3731, 2010.

[24] Kiani, F., Dalkilic, G. *Password Renewal Enhancement for Dynamic Authentication in Wireless Sensor Networks*, IEEE Conference on Computational Intelligence, Communication Systems, and Networks, 2, pp.143-146, 2010.

[25] Kanavalli, A. *SABR: Secure Authentication-Based Routing in Large Scale Wireless Sensor Network*, Emerging Research in Computing, Information, Communication and Applications, 223-229, 2015.

[26] Kiani, F. *A Novel Channel Allocation Method for Time Synchronization in Wireless Sensor Networks*, International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 2016, pp.1-11, 2016.

[27] Wang, Y., Attebury, G., Ramamurthy, B. *A survey of Security Issues in Wireless Sensor Networks*, IEEE Communications Surveys & Tutorials, 8(2), pp.2-23, 2006.

[28] OMNeT++website. http://whale.hit.bme.hu/omnetpp/

[29] www.memsic.com/userfiles/files/.../6020-0062-06_A_WSN_Professional_Series.pdf

[30] Sahingoz, K. *Large Scale Wireless Sensor Networks with Multi-Level Dynamic Key Management Scheme*, Journal of Systems Architecture, 59, pp.801–807, 2013.