6TiSCH Protokolü için Dağıtık Kullanıcı Kimlik Doğrulama Mekanizması
A Distributed User Authentication Mechanism for 6TiSCH Protocol

Hakan Aydın
Bilgisayar Mühendisliği
Karadeniz Teknik Üniversitesi
Trabzon, Türkiye
hakanaydin@ktu.edu.tr

Sedat Görmüş
Bilgisayar Mühendisliği
Karadeniz Teknik Üniversitesi
Trabzon, Türkiye
sedatgormus@ktu.edu.tr

***Özetçe* —Nesnelerin İnterneti (IoT) son zamanlarda popüler bir araştırma konusu haline gelmiştir. IoT ağlarının milyarlarca küçük cihazı İnternet'e entegre etmesi ve şehirlerin otomasyonundan yaşlı nüfus için ev tabanlı sağlık çözümlerine kadar sayısız uygulamayı mümkün kılmaktadır. Sayısız uygulama alanına sahip IoT ağı, güvenilirlik, düşük güç ve düşük gecikme gibi zorlukları da içermektedir. Bu tür internet tabanlı düşük güçlü cihazların, İnternet'in oluşturduğu güvenlik sorunlarıyla başa çıkabilmesi için ihtiyaç duyulan güvenlik gereksinimleri ile yapılandırılması gerekmektedir. Ayrıca güvenlik gereksinimleri, kısıtlı bir işlem gücüne ve mümkün olduğunca düşük güç tüketen mikro denetleyici aracılığıyla bu tür benzersiz zorluklara değinmek zorundadır.**

Bu bildiri, kimlik doğrulama anahtarlarının etkin bir önyükleme işlemini sağlamak için IoT ağının güvenilir düğümleri içinde dağıtıldığı 6TiSCH protokolünün güvenli önyükleme mekanizmasına bir eklenti sunar. Önerilen yaklaşım kullanılarak, standart 6TiSCH kimlik doğrulama mekanizmasının iletişim yükünü azaltmayı ve IoT ağının kenarında kimlik doğrulama belirteçlerini tutarak ağın enerji verimliliğini artırmayı amaçlıyoruz.

Anahtar Kelimeler*—*Nesnelerin İterneti, 6TiSCH, Önyükleme, Kimlik Doğrulama.

***Abstract*—The Internet of Things (IoT) has become a popular research topic in recent times. IoT networks enable billions of small devices to be integrated into the Internet and numerous applications ranging from automation of cities to home based health solutions for the elderly population. IoT network with numerous application areas also includes difficulties such as reliability, low power and low latency. Such low-powered Internet-based devices need to be configured with the security requirements needed to address the security problems that the Internet creates. In addition, security requirements must address such unique challenges through limited processing power and through a micro controller that consumes as little power as possible.**

This paper presents a plug-in to the secure boot mechanism of the 6TiSCH protocol that is deployed in the trusted nodes of the IoT network to provide an efficient boot process for authentication keys. By using the proposed approach, we aim to reduce the communication load of the standard 6TiSCH authentication mechanism and increase the energy efficiency of the network by keeping the authentication tokens at the edge of the IoT network.

Keywords*—*IoT, 6TiSCH, Bootstrapping, Authentication.

978-1-5386-4184-2/18/$31.00 ©2018 IEEE

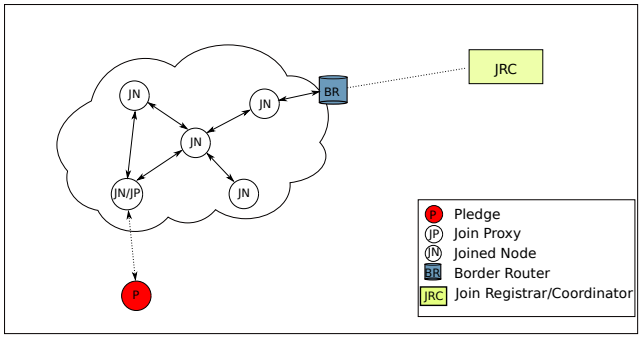
I. GİRİŞ

Nesnelerin İnterneti ağlarının İnternet tabanlı çevreleri önemli ölçüde değiştirmesi beklenmektedir. İnternnet protokolu teknolojisini kullanarak birbirleriyle haberleşen küçük cihazlar, ağırlıklı olarak algılayıcı bilgilerini taşıyan küçük paketler içerdiği tahmin edilen İnternet trafiğinde önemli bir artış sağlayacaktır. Bu tür küçük paketlerin, İnternetin verimliliğini olumsuz yönde etkilemesi beklenmektedir çünkü bunlar çoğu zaman, paket içinde taşınan bilgilerden daha büyük olan başlıklarda kapsülleneceklerdir. Bu tür sorunların üstesinden gelmek için araştırmacılar, yeni tanıtılan IoT cihazlarının yarattığı bu ani trafikle baş edebilmek için kenar hesaplama ve içerik merkezli yönlendirme gibi çeşitli önerileri öne sürdüler [1, 2]. Kenar hesaplaması durumunda, duyarga verileri olayın yakınında ağın kenarında işlenir, böylece merkezi bir varlık ile pahalı iletişim gerektirmeden uygun bir eylem gerçekle¸stirilebilir [2–4]. Kenar hesaplama, İnternet'teki IoT trafiğinin etkisini azaltmaya yardımcı olsa da ağın iletişim verimliliğini arttırmak için benzer bir duyarga verisinin bir IoT ağı içinde toplanmasını da göz önünde bulundurmak gerekir. Bu gibi çözümler, veri paketlerinin, Düşük Güçlü ve Kayıplı Ağlarda (RPL) IPv6 Yönlendirme Protokolü kullanarak 6LoWPAN'ın önerdiği gibi belli noktalara ulaşmak için veri atlamalarına ihtiyaç duyacağı kablosuz IoT mesh ağlarının ölçeklenebilirliğini de geliştirir. Bu tür ölçeklenebilirlik sorunları, IoT cihazlarının ağa önyüklenmesinde geçerli olan merkezi bir varlıkla yapıldığı herhangi bir merkezi protokol için sorun olmaya devam etmektedir.

Duyarga ağ iletişiminde, merkezi protokollerin çoklu duyarga ağı için iyi ölçeklenemediği bilinen bir gerçektir. Bu zorlukların üstesinden gelmek için, düğümlerin merkezi bir varlıktan edinilen önbelleğe alınmış bilgilere veya yerel ağı dinleyerek edinilen bilgilere dayanarak yerel kararlar verebileceği dağıtık algoritmalar geliştirilmelidir. Yerel kararların,merkezi bir varlığın belirli bir süreci hızlandırmak için düğümlere bilgi gönderdiği durumlarda, önbelleğe alınan bilgilerin yerel

karar vermenin kendi zorluklarına sahip olduğu da bilinir. 6TiSCH [5] çalışma grubu, IoT cihazlarının önyüklenmesi için "sıfır dokunuşlu" bir kimlik doğrulama mekanizması sunmuşlardır. Mevcut teklif, birleştirme vekil yardımcısı aracılığıyla düğümlerin kendilerini bu merkezi varlığa doğruladığı Şekil 1'de tasvir edildiği gibi, Katılma Kayıtçısı/Koordinatörü (JRC) olarak adlandırılan kimlik doğrulamasını etkinleştirmek için merkezi bir varlık içermektedir.

Bu bildiride, 6TiSCH ağında doğrulama parametrelerinin yerel düğümlerde depolandığı merkezi olmayan bir kimlik doğrulama yaklaşımı önerilmektedir. Protokol, Contiki [6] kullanılarak uygulanmış ve Cooja emulator [7] ile simüle edilmiştir. Bölüm II'de, ağaç tabanlı topolojilerde 6LoWPAN ve RPL protokollerini kullanan IoT kimlik doğrulama mekanizmaları için yapılan çalışmalar özetlenmiştir. Bölüm III, uygulanan merkezi ve dağıtık kimlik doğrulama mekanizmaları için ağ modelini sunar. 6TiSCH kimlik doğrulama mekanizmalarının ayrıntıları, uygulama özelliklerinin Contiki OS için tartışıldığı bölüm IV'tür. Önerilen mekanizmanın performans sonuçları, Contiki OS'nin Cooja simülasyon platformu kullanılarak bölüm V'te analiz edildi. Bölüm VI'da çalışmanın sonuçları irdelenmiş ve gelecekteki çalışmalara değinilmiştir.



Şekil 1: IETF 6TiSCH Önyükleme Protokolündeki Elemanlar

II. Yapılan Çalışmalar

Son yıllarda, küçük cihazların IoT ağlarına bağlanması için bol miktarda çalışma vardır. Bunun için farklı karmaşıklık seviyelerine sahip çeşitli kimlik doğrulama mekanizmaları önerilmiştir. Yazarlar [8] makalelerinde, düğümlerin merkezi bir kimlik doğrulama öğesi aracılığıyla doğrulandığı küçük cihazlar için kimlik doğrulama mekanizması önermektedir. Cihazlar, IoT ağının bir üyesi olmalarını sağlayan parametrelerle önceden yapılandırılmıştır. Yazarların kimlik doğrulaması için XOR tabanlı bir güvenli mesajlaşma protokolü kullandıkları M2M ağları için basitleştirilmiş bir kimlik doğrulama mekanizması önerilmiştir [9]. Kısıtlı Uygulama Protokolü (CoAP) [10] protokolü üzerinden Datagram Aktarım Katmanı Güvenliği (DTLS) kullanılmasının, kimlik doğrulama işleminin merkezi bir kimlik doğrulama sunucusunda gerçekleştirildiği ağa önemli bir iletişim ek yükü getirdiğinden bahsedilmektedir. Yine, böyle bir merkezi kimlik doğrulama mekanizması, birleştirme düğümü ve kimlik doğrulama sunucusu arasında ileri/geri yönlü birçok kimlik doğrulama mesajı trafiği oluşturacaktır. Bu, düğümlerin sınırlı iletişim bant genişliğine sahip olacağı M2M ağının ölçeklenebilirliğini sınırlayacaktır.

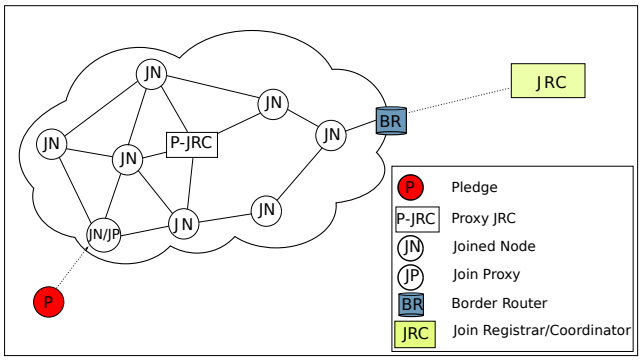
Daha öncede bahsedildiği gibi, DTLS CoAP protokolü mesajları için önerilen güvenli iletişim mekanizmasıdır. Ancak DTLS, belirli CoAP uygulamaları ile kullanıldığında ağın kaynak kullanımını önemli ölçüde arttırabilir. CoAP protokolü, sakla-ilet ve yayınla-abone değiş tokuşları ile uygulamaları desteklemek için gereklidir. Öte yandan DTLS, depolama ve iletme senaryoları için cihazlar arasında ihtiyaç duyulan sekme güvenliğini destekler. Bu tür uygulamalarda DTLS'yi tamamlamak için, CoAP mesajlarını çoklu-hop linkleri üzerinden güvence altına almayı amaçlayan yeni bir yöntem geliştirilmiştir. Bu CoAP uzantısı CoAP (OSCOAP) [11] için Object Security olarak adlandırılır. OSCOAP, son zamanlarda önerilen merkezi önyükleme mekanizmalarının [12] işlevini yerine getirdiği başlıca güvenli iletişim yöntemidir. Bu ifadelerden, kaynak kısıtlı cihazların önyüklenmesi için hafif protokollerin kullanılmasının başarılı bir IoT ağı için hayati önem taşıdığını görmek kolaydır. Hafif bir protokole sahip olmak, önyükleme işleminin verimliliğini artırırken, ölçeklenebilir bir bootstrapping çözümü oluşturmak, dağıtık bir mimariden yararlanan çözümler gerektirecektir.

Bu çalışmada, 6TiSCH için dağıtık bir önyükleme protokolü, merkezi önyükleme işlemine kıyasla iletişim yükünü azaltmak amacıyla sunulmuştur. Önerilen dağıtık önyükleme işleminde kullanılan kimlik doğrulama parametreleri, 6TiSCH ağı içindeki güvenilir düğümlerde saklanır. Bu adım, birleştirme düğümlerinin, merkezi kimlik doğrulama varlığına başvurmak zorunda kalmadan, güvenilir düğümlerden kimlik doğrulama belirteçleri edinerek ağa kimlik doğrulamasını sağlar.

III. Sistem Modeli

6TiSCH ağına katılmak isteyen kablosuz cihaz, kanal atlama sırası, slotframe süresi ve slot zamanlaması gibi parametreler hakkında bilgi edinmek için ağa senkronize edilmek zorundadır. Bu adımı takiben, uygun kimlik doğrulama bilgisi cihazda mevcut ise, ağa dahil olma işlemi başlatılır. 6TiSCH ağına katılma işlemi, işaretçilerin doğrulanması için önceden paylaşılmış bir anahtar veya JRC'ye aygıtın doğrulanması için sertifika gibi ön koşullara sahiptir. 6TiSCH ağına katılmak isteyen düğüm Pledge ve ağın parçası haline gelen düğüm ise Joined Node olarak adlandırılır. JRC'ye katılma isteğini kesintisiz bir şekilde ileten düğüme Join Proxy denir. JRC, düğümlerin yetkilendirilmesinden sorumlu olan güçlü bir cihazı ifade eder. Önerilen 6TiSCH birleştirme sürecindeki aktörler Şekil 2'de gösterilmiştir.

6TiSCH önyükleme modelinde, Pledge'in, dahil olmak istediği ağ için önceden paylaşılmış bir anahtar veya geçerli bir sertifikaya sahip olduğu varsayılır. 6TiSCH ağına katılmak isteyen düğüm, önce ağa senkronize olur ve kanal atlama, zamanlama parametreleri gibi değerlerin öğrenilmesi için bir EB mesajını dinler [5]. Pledge bir işaretçi aldığında, önce işaretçinin önceden bilinen bir anahtarla doğrulanmasını sağlar. Bu adım, işaretçinin 6TiSCH işaretçisi olduğunu doğrulamak için oluşturulmuştur. Pledge'nin yaymış olduğu işaretçinin geçerli bir 6TiSCH mesajı olarak doğrulanmasından sonra, cihaz ağa senkronize olur. Pledge için bir sonraki adım, CoAP protokolü ile işaretçisini aldığı düğüme bir Join Request mesajı göndermesidir. Join Proxy, isteğin JRC'ye iletilmesinden sorumludur. İletim işlemi, proxy düğümü tarafından OSCOAP protokolü aracılığıyla yapılır. JRC düğümü, önceden paylaşılmış anahtarı



Şekil 2: Sunulan 6TiSCH kimlik doğrulama mekanizmasının genel yapısı

veya önceden yüklenmiş sertifikayı kullanarak isteği doğrular ve Join Response mesajını Join Proxy öğesine geri gönderir. Join Proxy yanıt mesajını Pledge'e iletir ve Pledge, JRC'nin geçerli bir JRC'den katılım yanıtı aldığından emin olmak için kimlik doğrulama bilgilerini içeren yanıt mesajını doğrular. Katılım süreci bu adım ile sonlandırılır.

Bu çalışmada önyükleme modeli, JRC için kimlik doğrulama belirteçleri içerebilecek bir Proxy JRC (P-JRC) içerecek şekilde genişletilmiştir ve bu öğe dağıtık önyükleme modeli oluşturan JRC adına ağdaki düğümlerin kimliğini doğrulayabilir. P-JRC cihazı, JNs içinden veya gelişmiş donanıma sahip önceden yapılandırılmış cihazlar arasından seçilebilir. Önerilen çalışmanın ana motivasyonu, ağdaki kimlik doğrulama belirteçlerinin güncelleme işleminin iletişim yükünü azaltmaktır. P-JRC, ağ içinde tek bir güvenilir varlık olarak uygulanabilir veya birkaç düğümün, merkezileştirilmiş JRC varlığı için kimlik doğrulama bilgilerini birlikte depolayabileceği şekilde dağıtık şekilde uygulanabilir. Bu çalışmada, ağda Şekil 2 'de gösterildiği gibi bir tek P-JRC kullanılmıştır. P-JRC, kimlik doğrulama bilgilerinin cihazın epromunda saklandığı düşük güç exp5438 düğümüne uygulanır. Bu yaklaşımda ağa dahil olma süreci, 6TiSCH önyükleme modelinde olduğu gibi Pledge'in Join Proxy öğesine katılım isteği göndererek başlar.

Join Proxy, kesintisiz bir şekilde katılma isteğini merkezileştirilmiş JRC öğesinin hedef IPv6 adresine iletir. Mesaj ağ üzerinden sekme boyunca iletilirken, katılma isteği P-JRC üzerinden geçebilir. Mesajın P-JRC öğesi üzerinden iletilmesi durumunda, kimlik doğrulama işlemi P-JRC tarafından Pledge için sahip olduğu ortak anahtar değeri ile başlatılır. P-JRC kimlik doğrulama mesajını tamamlamak için Join Proxy 'ye cevap mesajı gönderir. JRC öğesi, kimlik doğrulama parametrelerini güncellemek için belli aralıklarla P-JRC öğesine mesaj gönderir. Bu durum, düğümlerin kimlik doğrulama belirteçlerinin merkezi olarak doğrulanması ve güncellenmesi ile karşılaştırıldığında daha düşük bir iletişim yükü sağlar.

IV. 6TISCH KIMLIK DOĞRULAMA MEKANİZMASI

RPL protokolü, düşük güçlü aygıttan ağdaki kök düğüme ve kök düğümden IPv6 adresleme [13] kullanan düşük güç aygıta iki yönlü trafiği etkinleştirmek için DODAG tabanlı topoloji oluşturur. Çalışmadaki düğümler, yerel alan ağı ile simüle edilen ağ arasındaki sınır yönlendirici düğümü vasıtasıyla kendilerini JRC'ye doğrulatabilir. Ayrıca JRC'nin, simetrik anahtarların ve cihazların dinamik kimliklerinin depolandığı güvenilir bir varlık olduğu varsayılır. JRC tarafından doğrulanmış olan cihazlar Joined Node haline gelir ve ağ oluşumuna katılabilir.

Uygulanan kimlik doğrulama mekanizması, Pledge'in kayıt ve kimlik doğrulamasından sorumlu iki adımı kapsar. Uygulanan kimlik doğrulama işlemi için kullanılan kısaltmalar, Tablo I'de tanımlanmıştır.

TABLO I: Kısaltmaların anlamları

Kısaltma	Tanım
ID_u	Düğümün kimlik bilgisi
DID_u	Düğümün dinamik kimlik bilgisi
PWD_u	Düğümün parolası
$H(PWD_u)$	Düğümün parolasının özüt değeri
GK_{JRC-u}	Düğüm ile JRC arasındaki simetrik anahtar değeri
PK_{JRC-u}	Kimlik doğrulamasını takiben JRC ve düğüm arasındaki gizli anahtar(simetrik)
TS	Zaman belirteci
$H(.)$	Tek yönlü özüt fonksiyonu (SHA-1)
$\|\|$	Metin birleştirme operatörü

A. Kayıt Aşaması

Bu aşamada (ID_u) ve (PWD_u, $H(PWD_u)$) parametreleri her cihazda oluşturulur ve saklanır. PWD_u, her aygıt için oluşturulmuş 20 bayt rastgele bayt dizisini temsil eder. Sisteme kayıtlı her cihazda JRC öğesi ile haberleşebilmesi için simetrik bir anahtar değeri üretilir ve bu anahtar değeri hem cihazda hem de JRC öğesinde kayıt altına alınır. Üretilen anahtar değeri, kimlik doğrulama sürecinde JRC mesajının doğruluğunu kontrol eder.

$$(DID_u) = (H(ID_u||H(PWD_u))) \quad (1)$$

Oluşturulan DID_u ve $H(PWD_u)$ parametreleri, JRC cihazında güvenli bir şekilde saklanır. Kimlik bilgilerinin özet değerleri JRC öğesinde depolandığından, cihazların göndermiş olduğu mesaj bilgisine erişebilen bir saldırgan, bu karma bilgiden cihazların orijinal kimliğini elde edemez. Diğer yandan, JRC gelen kimlik doğrulama mesajı ile Pledge için kayıtlı kimlik bilgilerinden üretilen DID ile karşılaştırarak cihazın geçerli bir cihaz olup olmadığını doğrulayabilir.

B. Doğrulama Aşaması

Bir cihaz 6TiSCH ağına katılmak istediğinde, doğrulama aşamasında açıklanan prosedürleri takip ederek ağa dahil olabilmektedir. Daha önce de belirtildiği gibi 6TiSCH, EB'lerin ağa dahil olma sürecinde önceden yapılandırılmış veya elde edilmiş bir anahtar kullanılarak doğrulanmasını gerektirir. Bir düğüm ağdan EB aldığında ona senkronize olur. Alınan EB mesajından slot bilgisi, slotframe süresi vb. gibi ağ parametreleri öğrenilir. Pledge daha sonra bağlantı-yerel IPv6 adresini yapılandırır ve Join Proxy (JP) düğümüne senkronize olur. Bu adımdan sonra Pledge, katılım işlemini başlatır. Bu bildiride önerilen katılma işlemi, [14] 'de tanımlanan teklifin biraz değiştirilmiş bir sürümünü kullanır.

6TiSCH ağına katılmak isteyen düğüm, Join Proxy aracılığıyla JRC adresine uçtan uca şifreli bir mesaj gönderir. Join Request iletisinin şifrelenmesi ortak bir anahtar veya

Pledge'de yüklü sertifikalar kullanılarak gerçekleştirilir. Ortak anahtarlar kullanan kimlik doğrulama mekanizması, Pledge ve JRC arasında daha az mesaj trafiğine sebep olmaktadır. Bu işlem, ortak anahtar tabanlı kimlik doğrulama sürecini, sertifika tabanlı kimlik doğrulama ile karşılaştırıldığında daha elverişli hale getirir.

Bu bildiride, merkezi kimlik doğrulama sürecinin getirdiği iletişim yükünü azaltmak amacıyla 6TiSCH ağı içinde JRC benzeri bir öğe yer almaktadır. Elbette, tüm bir ağ için doğrulama belirteçlerinin düşük güçte bir cihaza uymayacağı öngörülmektedir. Ancak, ağdaki bazı cihazların, ağın parçası için dikkatlice seçilmiş bir kimlik doğrulama bilgisi alt kümesi depoladığını varsayarsak, daha kısa kimlik doğrulama yollarından yararlanarak ağın verimliliğini arttırmak mümkündür. Bu çalışmada, yalnızca dağıtık bir kimlik doğrulama mekanizmasının ağın trafik ve enerji tüketimi üzerindeki etkisine odaklanıyoruz. JRC öğesine benzer özellik gösteren düğümlerin seçimi bu yazının kapsamı dışındadır ve daha sonra ele alınacaktır. Önerilen çalışmada, P-JRC düğümünün ağda ideal konumda olduğu ve ağ için kimlik doğrulama parametrelerini saklamada yeterli bellek alanına sahip olduğu varsayılmaktadır. Şekil 2'de görülebileceği gibi, P-JRC kimlik doğrulama isteği alırsa ve bu istek için kimlik doğrulama bilgilerine sahipse P-JRC, Pledge cihazını doğrular ve Pledge için kimlik doğrulama parametreleri ile birlikte bu cihaza bir yanıt mesajı gönderir. Diğer yandan P-JRC ağa dahil olmak isteyen düğümün yönlendirme yolunda değilse, Şekil 1 'da açıklandığı gibi kimlik doğrulama işlemi JRC'de gerçekleşir.

V. Deneysel Sonuçlar

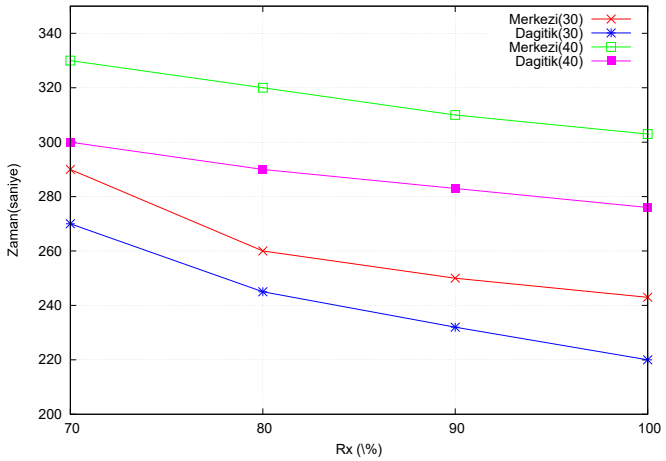
Önerilen önyükleme protokolü Contiki işletim sisteminde exp5438 gömülü platform [15] için Cooja emülatöründe simule edilmiştir. Tablo II'de değerlendirme için kullanılan parametreler verilmiştir. 30 ve 40 düğüm senaryoları farklı besleme (seed) değerlerinde 5 kez çalıştırılarak ortalama doğrulama gecikmesi [1] hesaplanmıştır. Her iki senaryo için ideal olarak yerleştirilmiş P-JRC kullanılır. Düğümler belirtilen adımları gerçekleştirip ağa senkron olduktan sonra 30 ile 60 saniye arası bir sürede Join Proxy (JP)'ye kimlik doğrulama isteği gönderir.

TABLO II: Kullanılan parametreler

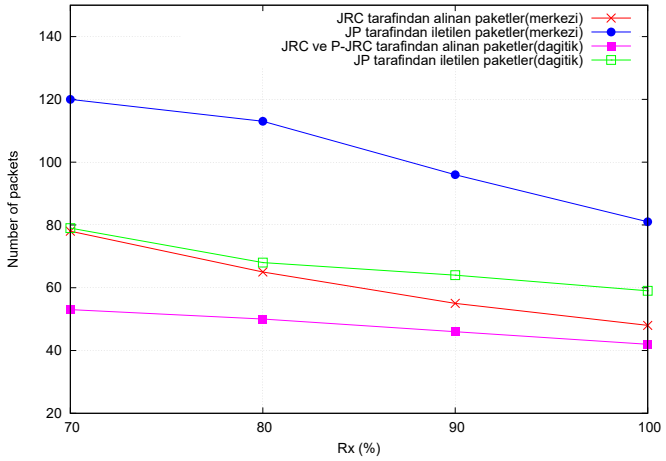
Parametreler	**Değer**
Düğüm Sayısı	30-40
Gecikme(dakika)	30
Kimlik doğrulama isteği(saniye)	30-60
Rx(%)	70-80-90-100
Yayılım Modu	Cooja UDGM

Uygulanan kimlik doğrulama mekanizmalarının performans sonuçları Şekil 3'te gösterilmektedir. Ağın bağlantı kalitesine bağlı olarak düğümlerin kimliklerinin doğrulama süresinin değiştiği gözlemlenmiştir. Düşük bağlantı kalitesinde önyükleme adımı daha uzun süre almaktadır. Tüm bağlantı kalitesi olasılıkları için dağıtık kimlik doğrulama mekanizmasındaki ortalama ağa dahil olma süresi daha kısadır. Merkezi kimlik doğrulama mekanizmasına göre dağıtık kimlik doğrulama süresi, yaklaşık % 38 daha iyi performans gösterir.

[1] Bu metrik, değerlendirmelerimizin ana performans ölçütüdür.



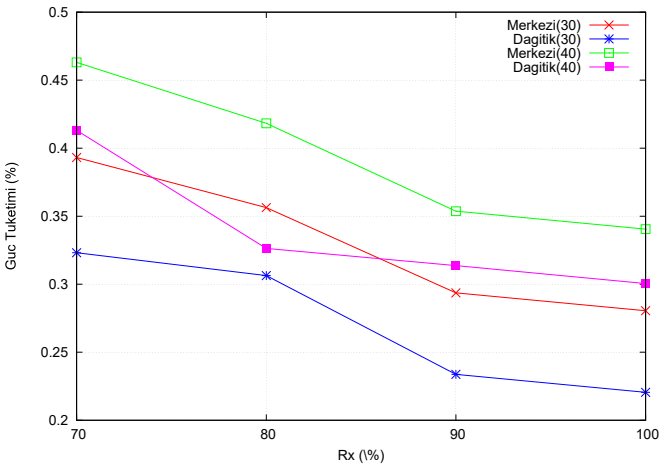
Şekil 3: 6TiSCH ağı için ortalama kimlik doğrulama süresi



Şekil 4: 40 düğüme sahip ağ için iletilen kimlik doğrulama mesajlarının sayısı

Şekil 4, 40 düğümlü ağ senaryosu için merkezi ve dağıtık kimlik doğrulama mekanizmalarında gönderilen ve alınan doğrulama paketlerinin sayısını göstermektedir. Şekilden görüleceği gibi bağlantı başarısı oranı ile iletilen paketlerin toplam sayısı orantılı olarak gösterilmiştir. Tüm ağı dolaşmak zorunda kalan toplam kimlik doğrulama paket sayısı, dağıtık kimlik doğrulama mekanizmasına kıyasla merkezi bir kimlik doğrulama yaklaşımı kullanıldığında daha fazla paket trafiğine sebep olur. 6TiSCH ağları için önerilen kimlik doğrulama mekanizmasının sonuçları, uygulanan merkezi kimlik doğrulama ile karşılaştırıldığında % 40 daha az paket trafiğine neden olur.

Şekil. 5, 3.3V-100 mAh pilin kullanıldığı pil kapasitesine göre düğümlerin ağa önyüklenmesinde tüketilen enerji miktarını verir. Bu çalışmada, exp5438 platformunun enerji tüketim rakamlarını [16]'da verilen bilgiler kullanılarak simüle edilir.



Şekil 5: Önyükleme işlemi sırasında tüketilen ortalama enerji miktarı

VI. Sonuç

Bu çalışmada, merkezi ve dağıtık ağa dahil olma mekanizmaları uygulanmış ve performansları karşılaştırılmıştır. 6TiSCH ağına dikkatlice yerleştirilmiş JRC benzeri yapıya sahip olmanın, bu tür ağların kimlik doğrulama performansını önemli ölçüde arttırabileceğini göstermiştir. Ek olarak, ağdan geçmesi gereken paketlerin miktari, ideal konuma yerleştirilmiş JRC ile % 40'a varan oranda azaltılabilir. Bu, gelecekteki araştırmalar için IoT ağında birden fazla JRC replikasyonunun yapılabileceği umuduyla gelecek vaat eder.

IoT ağında JRC öğesinin çoğaltılmasıyla ilgili olağanüstü zorluklar vardır. En belirgin olanı ise, kendi başına karmaşık bir problem olan ve gelecekteki araştırmalar olarak ele alınması gereken RPL yönlendirme protokolü kullanılarak bir 6TiSCH ağı içinde çoğaltılan JRC'nin ideal lokasyonunun seçilmesidir. Diğer zorluklar arasında, düşük güçlü ve düşük kapasiteli bir aygıtta kritik kimlik doğrulama belirteçlerinin güvenli bir şekilde nasıl saklanacağı, bu kimlik doğrulama belirteçlerinin ne sıklıkla yenilenmesi gerektiği, kimlik doğrulama belirteçlerinin ağ içinde ideal olarak nasıl saklanacağı gibi sorulardır. Gelecekteki çalışmalarımız, 6TiSCH tabanlı IoT ağları için güvenli ve ölçeklenebilir kimlik doğrulama mekanizmaları oluşturarak bu zorlukları ele alacaktır.

Kaynaklar

[1] Yichao Jin, Sedat Gormus, Parag Kulkarni, and Mahesh Sooriyabandara. Content centric routing in iot networks and its integration in rpl. *Computer Communications*, 89:87–104, 2016.

[2] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.

[3] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. Integration of cloud computing and internet of things: a survey. *Future Generation Computer Systems*, 56:684–700, 2016.

[4] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. Fog computing: A platform for internet of things and analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*, pages 169–186. Springer, 2014.

[5] IETF. Ipv6 over the tsch mode of ieee 802.15.4e (6tisch). Available: https://datatracker.ietf.org/wg/6tisch/charter/, [Online]: 28/10/2017.

[6] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. Contiki-a lightweight and flexible operating system for tiny networked sensors. In *Local Computer Networks, 2004. 29th Annual IEEE International Conference on*, pages 455–462. IEEE, 2004.

[7] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with cooja. In *Local computer networks, proceedings 2006 31st IEEE conference on*, pages 641–648. IEEE, 2006.

[8] Mahzad Azarmehr, Arash Ahmadi, and Rashid Rashidzadeh. Secure authentication and access mechanism for iot wireless sensors. In *Circuits and Systems (ISCAS), 2017 IEEE International Symposium on*, pages 1–4. IEEE, 2017.

[9] Alireza Esfahani, Georgios Mantas, Rainer Matischek, Firooz B Saghezchi, Jonathan Rodriguez, Ani Bicaku, Silia Maksuti, Markus Tauber, Christoph Schmittner, and Joaquim Bastos. A lightweight authentication mechanism for m2m communications in industrial iot environment. *IEEE Internet of Things Journal*, 2017.

[10] Zach Shelby, Klaus Hartke, and Carsten Bormann. The constrained application protocol (coap). 2014.

[11] G Selander, J Mattsson, F Palombini, and L Seitz. Object security of coap (oscoap). *IETF, Internet-Draft*, 2015.

[12] 6tisch secure join protocol. Available: https://tools.ietf.org/html/draft-ietf-6tisch-dtsecurity-secure-join-01, [Online]: 03/11/2017.

[13] Tim Winter. Rpl: Ipv6 routing protocol for low-power and lossy networks. 2012.

[14] Minimal security framework for 6tisch. Available: https://tools.ietf.org/html/draft-ietf-6tisch-minimal-security-02, [Online]: 01/11/2017.

[15] Msp-exp430f5438 experimenter board user's guide, texas instruments inc., da, texas, 2013. Available: http://www.ti.com/lit/ug/slau263i/slau263i.pdf, [Online]: 10/11/2017.

[16] Rafael Lajara, José Pelegrí-Sebastiá, and Juan J Perez Solano. Power consumption analysis of operating systems for wireless sensor networks. *Sensors*, 10(6):5809–5826, 2010.