KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ TASARIMI

Sinan DİVARCI

KOCAELİ 2018

KOCAELİ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE HABERLEŞME MÜHENDİSLİĞİ

ANABİLİM DALI

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ TASARIMI

Sinan DİVARCI

Prof. Dr. Oğuzhan URHAN
Danışman, Kocaeli Üniversitesi .....

Dr. Öğr. Üyesi Oğuzhan KARAHAN
Jüri Üyesi, Kocaeli Üniversitesi .....

Prof.Dr. CEM ÜNSALAN
Jüri Üyesi, Marmara Üniversitesi .....

Tezin Savunulduğu Tarih: 02.07.2018

ÖNSÖZ VE TEŞEKKÜR

Bu tez çalışmasında, nesnelerin İnterneti için bir güvenli ağ geçidi tasarımı çalışması yapılmıştır. Yapılan çalışmaların bu alanda çalışacak araştırmacılara yararlı olmasını dilerim.

Bu konuda çalışma yapmama olanak sağlayan değerli hocam Prof. Dr. Oğuzhan Urhan'a, bilgi birikimi ile desteğini eksik etmeyen başta Ali Can KESEREL olmak üzere Netaş'taki tüm çalışma arkadaşlarıma, hayatım boyunca hem maddi hem de manevi olarak benden desteklerini esirgemeyen ve benim bugünlere gelmemde en büyük pay sahibi olan aileme ve bana destek olan tüm dostlarıma teşekkürü borç bilirim.

Haziran - 2018 Sinan DİVARCI

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

SİMGELER VE KISALTMALAR DİZİNİ

Kısaltmalar

3DES : Triple Data Encryption Algorithm (Üçlü Veri Şifreleme Standardı)
ADC : Analog to Digital Converter (Analogdan Dijitale Dönüştürücü)
AES : Advanced Encryption Standard (Gelişmiş Şifreleme Standardı)
AH : Authentication Header (Doğrulama Başlığı)
ARM : Acorn RISC Machine (RISC Tabanlı Bir İşlemci Mimarisi)
ARP : Address Resolution Protocol (Adres Çözümleme Protokolü)
ASCII : American Standard Code for Information Interchange (Bilgi Değişimi İçin Amerikan Standart Kodlama Sistemi)
AUTH : Authentication (Kimlik Doğrulama)
CAN : Controller Area Network (Denetleyici Alanı Ağı)
CPU : Central Processing Unit (Merkezi İşlem Birimi)
CRC : Cyclic Redundancy Check (Döngüsel Artıklık Denetimi)
DES : Data Encryption Standart (Veri Şifreleme Standardı)
DH : Diffie-Hellman
DHCP : Dynamic Host Configuration Protocol (Dinamik Ana Bilgisayar Yapılandırma Protokolü)
DMIPS : Dhrystone Millions of Instructions Per Second (Dhrystone Saniye Başına Milyon Komutu)
DTLS : Datagram Transport Layer Security (Veri birimi Aktarım Katmanı Güvenliği Protokolü)
EEPROM : Electrically Erasable Programmable Read-Only Memory (Elektronik Olarak Silinebilir Programlanabilir Salt Okunur Bellek)
ENC : Explicit Congestion Notification (Açık Tıkanıklık Bildirimi)
ESP : Encapsulating Security Payload (Kapsüllenmiş Güvenlik Yükü)
MAC : Medium Access Control (Ortam Erişim Kontrolü)
HDR : IKE Header (IKE Başlığı)
HMAC : Hash-Based Message Authentication Code (Özet Tabanlı Mesaj Doğrulama Kodu)
I2C : Inter-Integrated Circuit (Entegre Devreler Arası Haberleşme)
IAB : Internet Architecture Board (İnternet Mimarisi Kurulu)
IANA : Internet Assigned Numbers Authority (İnternet Atanan Sayılar Kurumu)
ICMP : Internet Control Message Protocol (Internet Kontrol Mesaj İletişim Kuralı)
ID : Identification (Kimlik)
IETF : Internet Engineering Task Force (İnternet Mühendisliği Görev Grubu)
IKEv2 : Internet Key Exchange Version 2 (Anahtar Değişim Protokolü Versiyon 2)
IP : Internet Protocol (İnternet Protokolü)

IPsec : Internet Protocol Security (İnternet Protokolü Güvenliği)
IPv4 : Internet Protocol Version 4 (İnternet Protokol Versiyon 4)
IPv6 : Internet Protocol Version 6 (İnternet Protokol Versiyon 6)
KB : Kilobyte (Kilobayt)
KE : Key Exchange(Anahtar Değişim)
lwIP : Lightweight Ip (Hafif Ip)
MCU : Microcontroller (Mikrodenetleyici)
MD5 : Message-Digest Algorithm 5 (Mesaj-Özet Algoritması 5)
Ni, Nr : Nonce Initiator (Nonce İlklendirici), Nonce Responder (Nonce Cevap Veren)
PHY : Physical Layer (Fiziksel Katman)
PRF : Pseudorandom Function (Sözde Rastgele Fonksiyonu)
PSK : Pre-Shared Key (Ön Paylaşımlı Anahtar)
QSSI : Quad Synchronous Serial Interface (Dörtlü Senkron Seri Arabirim)
RAM : Random Access Memory (Rastgele Erişimli Hafıza)
RFC : Request for Comments (Yorum Isteği)
SA : Security Association (Güvenlik Oturumu)
SCTP : Stream Control Transmission Protocol (Akış Kontrol Iletişim Protokolü)
SHA1 : Secure Hash Algorithm (Güvenli Hash Algoritması)
SK : Shared Key (Paylaşılan Anahtar)
SPD : Security Policy Database (Güvenlik Politikası Veritabanı)
SPI : Security Parameter Index (Güvenlik Parametresi Dizini)
SSL : Secure Sockets Layer (Güvenli Giriş Katmanı)
TCP : Transmission Control Protocol (Geçiş Kontrol Protokolü)
TLS : Transport Layer Security (Taşıma Katmanı Güvenliği)
ToS : Type of Service (Servis Türü)
TS : Traffic Selector (Trafik Seçici)
TTL : Time to Live (Yaşam Süresi)
UART : Universal Asynchronous Receiver-Transmitter (Evrensel Asenkron Alıcı - Verici)
UDP : User Datagram Protocol (Kullanıcı Veribloğu İletişim Kuralları)
USB : Universal Serial Bus (Evrensel Seri Veriyolu)
VPN : Virtual Private Network (Sanal Özel Ağ)

NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ TASARIMI

ÖZET

Teknolojinin gelişmesiyle hayatımızın her alanında yer alan elektronik cihazlar İnternet erişme yetkinliği kazanarak kişisel verilerimizi veya stratejik öneme sahip sistemlere ait verileri İnternet üzerinden iletebilmektedirler. Kritik öneme sahip bu gibi verilerin İnternet üzerinden iletilmesi sırasında artan güvenlik ihtiyaçlarının karşılanması amacıyla IPsec protokolünün kullanımı da artmıştır.

Bu tez çalışması kapsamında düşük kaynak ve maliyet gerektiren nesnelerin İnterneti uygulamalarında yüksek seviyeli veri gizliliği ve bütünlüğü sağlayabilmek için kriptografik altyapıya sahip bir mikrodenetleyici kullanılarak IPsec ve IKEv2 protokolleri gerçeklenmiştir. Tasarlanan güvenli ağ geçidinde gerçeklemesi yapılan IPsec ve IKEv2 protokolleri ve bu protokollerin tasarlanan sistemde gerçeklenmesi anlatılarak sonuçları paylaşılmıştır.

Anahtar Kelimeler: Diffie-Hellman, IKEv2, IPsec, Nesnelerin İnterneti.

SECURE GATEWAY DESIGN FOR INTERNET OF THINGS

ABSTRACT

With the development of technology, electronic devices in all areas of our lives gain Internet access flexibility and can transmit personal data or data belonging to systems with strategic presets via the Internet. The use of the IPsec protocol has also increased in order to meet increased security requirements when transmitting such critical data over the Internet.

In this thesis study, IPsec and IKEv2 protocols are implemented using a microcontroller with cryptographic infrastructure to provide high level data privacy and integrity in Internet of Things applications requiring low resource and cost. IPsec and IKEv2 protocols implemented in the designed secure gateway and the implementation of these protocols in the designed system are explained and the results are shared.

Key Words: Diffie-Hellman, IKEv2, IPsec, Internet of Things.

GİRİŞ

Teknolojinin gelişip küçüldüğü ve günlük hayatımızın her alanında yer almaya başladığı günümüzde basit elektronik aletler de İnternet yetkinliği kazanarak birbirleriyle bağlantılı hale gelmeye başlamışlardır. İnternet yetkinliği kazanan cihazlar günlük hayatımızdaki kişisel verilerimizi kullanıp bir başka noktaya iletebildikleri gibi güç kontrol sistemleri, alarm sistemleri gibi stratejik öneme sahip sistemlere ait verilerle de çalışmaktadırlar.

İnternete erişebilen cihazlar kritik bilgileri güvenlik protokolleri tarafından korumadan bir noktadan diğer bir noktaya İnternet veya yerel ağ aracılığı ile iletirken bu verilerin ağdaki diğer cihazlar tarafından kayıt edilmesinin ya da değiştirilmesinin önünde herhangi bir engel bulunmamaktadır. Bu durum veri mahremiyetinin ve veri bütünlüğünün sağlanmasını gerekli kılmaktadır. Günümüzde veri bütünlüğü ve/veya mahremiyetinin sağlanması için SSL, TLS, IPsec, VPN, DTLS gibi güvenlik protokolleri kullanılmaktadır. Uygulanacak sistemin kabiliyetlerine ve güvenlik ihtiyaçlarına göre kullanılacak güvenlik protokolü belirlenmektedir. Güvenlik protokolleri tek başına veya birbirleri ile uyumluluk durumlarına göre beraber kullanılabilirler.

İnsanlar ile doğrudan sensörler veya butonlar gibi fiziksel algılayıcılar ile etkileşime geçebilen, İnternet ile birbirine bağlanarak birbiriyle kablolu/kablosuz haberleşme teknolojilerini kullanarak topladıkları verileri iletebilen küçük cihazların oluşturduğu ekosistem "Nesnelerin İnterneti" olarak tanımlanabilir. "Nesnelerin İnterneti" teriminde "Nesneler" bilgisayarları kapsamamaktadır. Birbirleriyle İnternet üzerinden veri aktarıp haberleşebilen gömülü sistemler ve sensörler "Nesneler" olarak adlandırılmaktadır.

Nesnelerin İnterneti ekosisteminde düşük donanım kaynaklarına sahip nesnelerin İnternet üzerinden veri alışverişi yapmaları sonucu ortaya çıkan bilgi gizliliği ve güvenliği ihtiyacı için bu sistemlerin karşılayabileceği düşük kaynak gerektiren güvenlik uygulamaları önerilmektedir. Ancak bu düşük kaynaklı güvenlik uygulamaları güvenlik anlamında uygulamaların ihtiyaçlarını yeterince karşılayamamaktadırlar [1].

Tez kapsamında, nesnelerin İnterneti sistemlerinin İnternete açıldığı noktada çalışarak İnternet üzerinden aktarılacak olan verilerin şifrelenip veri bütünlük kontrolünü sağlayabilen bir güvenli ağ geçidi tasarımı yapılmıştır. IoT ağından bir veri başka bir noktaya İnternet üzerinden gönderilmek istendiğinde bu veri tasarlanan ağ geçidi üzerinden iletilecektir. Tasarlanan ağ geçidi IPsec protokolünü kullanarak ağ seviyesinde paketlerin gizliliğini ve bütünlüğü koruma altına almaktadır.

Tasarlanan sisteme IKEv2 protokolü yeteneği kazandırılarak güvenli ağ geçidinde kullanılacak şifreleme ve özet çıkarma algoritmaları için gerekli kripto anahtarları otomatik olarak üretilmiştir. IKEv2 protokolü sayesinde IPsec daha esnek ve ölçeklenebilir bir güvenlik protokolü haline getirilmiştir.

Tez içeriğinin ilk bölümünde önerilen sistemde de kullanılan IPsec protokolünün ayrıntıları anlatılmıştır. İkinci bölümde IKEv2 protokolünün ayrıntılarına değinilmiştir. Üçüncü bölümde Diffie-Hellman anahtar değişimi protokolü bir örnek ile açıklanmıştır. Dördüncü bölümde tasarlanan sistemin ayrıntıları verilerek, yazılım tasarımı, sistemin akış diyagramları ve seçilen donanım açıklanmıştır. Son bölümde ise tasarlanan sistemde çalışan IPsec ve IKEv2 protokollerinin testleri, tasarlanan rastgele sayı üretecinin testi ve sistemin bant genişliği testleri yapılarak değerlendirmelerde bulunulmuştur.

1. IPSEC PROTOKOL TAKIMI

1994 yılında IAB tarafından yayınlanan, "İnternet Mimarisinde Güvenlik "konulu 1636 numaralı RFC'de TCP/IP protokolünün güvenlik zafiyetleri nedeniyle İnternetin daha fazla ve daha iyi güvenlik protokollerine ihtiyaç duyduğu konusunda bir fikir birliğine varılmış ve geliştirilmesi gereken anahtar noktalara değinilmiştir. Yeni nesil IP protokolü olan IPv6'da dahil yeni nesil ağ iletişim teknolojilerinde bu güvenlik protokollerinin dahili olarak bulunmasının zorunlu olması gerektiği vurgulanmıştır [2]. IAB tarafından yayınlanan rapordaki güvenlik gereksinimleri günümüzde IPsec ile desteklenmiş VPN teknolojileriyle sağlanabilmektedir. Günümüzde IPsec'i tanımlayan birkaç adet RFC bulunmakla beraber 40'ın üzerinde RFC taslağı IPsec'in esnekliğinin ve ölçeklenebilirliğinin çeşitli yönlerini ele almaktadır [2]. IPsec IETF tarafından 2401 numaralı RFC ile standartlaştırılmıştır.

IPsec, İnternet üzerinden yapılan haberleşmeyi uçtan uca şifreleyip veri bütünlüğü kontrolünü sağlayabilen bir güvenlik protokolüdür. Halihazırdaki IP altyapısı ile birlikte çalışabilir, yüksek güvenirlikli, kriptografi tabanlı bir güvenlik protokolü olarak tasarlanmıştır. Erişim kontrolü, veri bütünlüğü, yeniden aynı veriyi gönderme (replay) saldırılarını tespit etme ve engelleme ve veri trafiği gizliliği sağlar. Bu güvenlik sistemlerini IP katmanında sağlayarak IP protokolünü kullanan tüm trafiğin güvenliğini sağlamaktadır [3]. IPsec, tünel ve iletim modu olmak üzere iki farklı modda çalışabilmektedir. Tünel modu ile VPN oluşturmak için kullanılabilmektedir. İletim modu ise genellikle iki bilgisayar arasında güvenli haberleşmeyi sağlayabilmek için kullanılmaktadır.

IPsec, anahtar değişimi ve kullanılacak güvenlik protokolleri üzerinde iki güvenli ağ geçidinin anlaşması gibi işlemleri gerçekleştiremez. IPsec için hali hazırda bir güvenlik oturumunun harici bir uygulama ile veya el ile -manuel olarak- önceden sağlanması gerekmektedir [3]. Güvenlik Oturumu (Security Association – SA), iki uç noktanın güvenlik protokolünde kullanılacak şifreleme ve bütünlük algoritmaları

üzerinde anlaşması ve bu algoritmalarda kullanılacak anahtarların her iki uçta da bulunması durumunu tanımlamaktadır[4]. IPsec, doğrudan IP katmanı üzerinde erişim kontrolüne sahip olduğu için temel düzeyde güvenlik duvarı özelliği gösterir. IPsec gerçeklenirken daha karmaşık ve gelişmiş güvenlik duvarı özelliklerinin kazandırılması IPsec gerçeklemesinin nasıl yapılmak istendiğine bağlıdır.

IPsec çalışma adımları aşağıdaki gibi sıralanabilir:

- IPsec protokolünün çalışması için öncelikle bir anahtar yönetimi protokolü ile veya el ile güvenlik oturumu kurulmalıdır.
- Güvenlik oturumu tarafından sağlanan IPsec anahtarları, IP paketlerinin şifrelenmesi ve kaynak doğrulanması için kullanılır.
- DES, 3DES, AES gibi bir şifreleme algoritması kullanılarak IP paketi şifrelenir.
- SHA1, SHA2, MD5 gibi bir kimlik doğrulama algoritması ile IP paketinin kaynağının doğrulanabilmesi için gerekli veriler oluşturulur.
- Şifrelenip kaynak doğrulaması yapılan IP paketleri alıcı tarafa gönderilir.

Tasarlanan sistemin ölçeklenebilir ve esnek olabilmesi için güvenlik oturumu yönetimi IKEv2 protokolü ile sağlanmıştır. Tasarlanan sistemde IPsec ve IKEv2 protokolleri beraber gerçeklenerek kullanılmıştır.

1.1. IPsec Kullanım Nedenleri ve Görevleri

IPsec protokolünün en büyük avantajı IP katmanında çalışarak ağın üst katmanlarında çalışan uygulamalarda herhangi bir değişiklik yapılmasını gerektirmemesidir. IPsec üst katman uygulamaları tarafından fark edilemez. Uygulamalara ait trafiğin TCP ya da UDP ile gönderilmesi arasında IPsec protokolü için bir farklılık yoktur.

IPsec genellikle iki farklı ağ arasında kurulmaktadır. Ağlar arasındaki IP paketi alışverişi güvenli ağ geçitleri tarafından yapıldığı için bu ağlara eklenecek olan bilgisayarların IPsec desteklemelerine gerek yoktur, güvenli ağ geçidi tarafından sağlanan güvenlik hizmetlerinden yararlanabilirler[5]. Güvenli ağ geçitleri tarafından İnternet üzerinde ilerleyen IP paketlerinin güvenliği IPsec ile sağlanabilir. Böylelikle IPsec destekleyemeyecek düşük kaynaklı cihazlar da bu güvenli ağ geçitleri üzerinden ağa çıkartılarak ağ katmanı güvenliği sağlanmış olur. Ayrıca ağ içinde sadece güvenli

ağ geçidinde güvenlik oturumu yönetimi ve IPsec çalıştırılarak sistem karmaşıklığı azaltılmış olur.

IPsec, ağda iletilen verilerde oluşabilecek bozulmaları algılayabilir, ağ trafiğindeki veri hırsızlıklarının, şifrelerin ve hesapların ele geçirilmesinin önüne geçebilir ve ağ kaynaklı saldırıları engelleyebilir. Böylelikle ağ üzerinde ilerleyen IP paketlerinin güvenliğini sağlar.

IPsec ağ trafiğini filtreleyerek belirlenen IP aralığındaki istemciler ile IPsec protokolü ile haberleşmeyi sağlayabilir. Böylelikle IPsec özelliği olamayan istemciler ile IPsec protokolü kullanılmadan da paket alışverişi sağlanabilir.

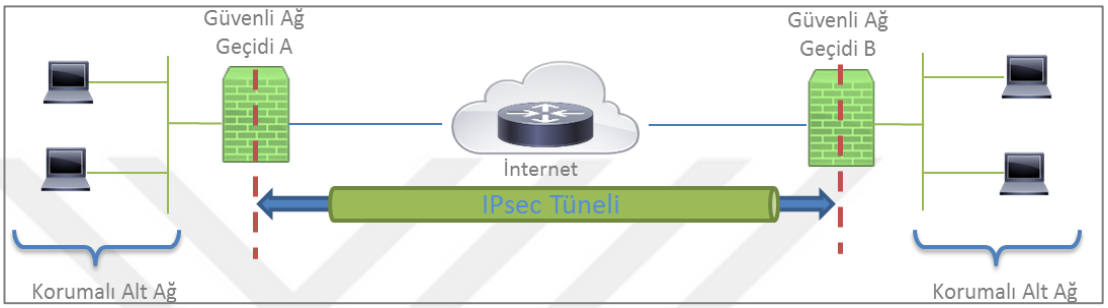
1.2. IPsec Çalışma Şekli

IPsec ağ geçitleri arasında, ağ geçidi ile istemci arasında veya doğrudan istemci bilgisayarlar arasında çalışabilmektedir. IPv4 ve IPv6 için güvenlik servislerini sağlamakta kullanılan AH ve ESP olmak üzere iki farklı protokol kullanılır. Bu protokoller birçok farklı şekilde kullanılarak IP güvenlik mekanizması işletilir [5].

Doğrulama başlığı IP paketlerinde gizlilik sağlamadan (şifrelemeden) bütünlük kontrolü ve kaynak doğrulaması sağlar. Gizlilik sağlamamasına rağmen üst katman uygulamalarında gerçekleştirilen veri şifreleme protokolleri ile kullanılabilmektedir. Kapsüllenmiş Güvenlik Yükü IP paketlerine veri gizliliği sağlamasının yanı sıra AH gibi bütünlük kontrolü ve kaynak doğrulaması da sağlayabilir. ESP ve AH tek tek kullanılabildiği gibi istenilen şekillerde bir arada kullanılabilmektedirler. Her iki protokol de iletim ve tünel modlarında kullanılabilmektedirler. İletim modunda kullanılan protokoller IP katmanının üzerindeki protokolleri korurken tünel modunda tüm IP paketi korunmaktadır.

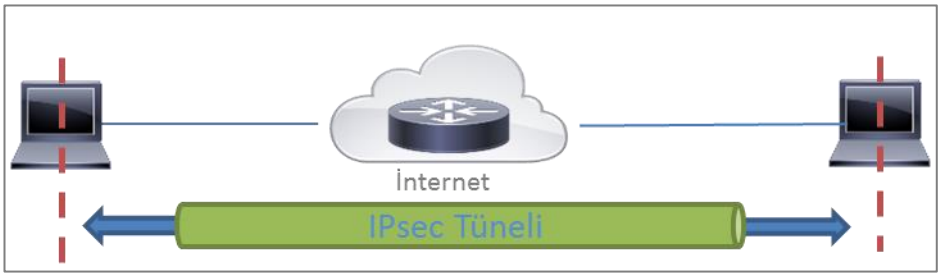
Ağ geçitleri arasında IPsec tüneli kurulması durumunda korumalı ağ içindeki bilgisayarlarda IPsec çalıştırılmasına gerek yoktur. Ağ üzerindeki güvenli ağ geçitleri ağ içindeki bilgisayarların ağ üzerinden gönderdikleri paketlerin korunması görevini gerçekleştirirler. Ağ geçitleri arasında IPsec tüneli kurulması durumunda gerçekleştirilen ağ koruması ağdaki bilgisayarlar tarafından algılanamaz bu nedenle bilgisayarlarda bir değişiklik veya IPsec için bir ayarlama yapmak gerekmez.

Korumalı ağda çalışan cihazlar ağ üzerinden IP paketi göndermek istediklerinde paketleri yerel güvenli ağ geçidine iletirler. Örnek olarak Şekil 1.1'de gösterilen Güvenli ağ geçidi A, paketi gönderen bilgisayarın ve hedef IP adresinin bulunduğu IP başlığını kapsülleyerek orijinal IP başlığının üzerine kaynak IP adresi olarak kendi IP adresini, hedef IP adresi olarak güvenli ağ geçidi B'nin IP adresini yazarak paketi ağa gönderir. Güvenli Ağ Geçidi B gelen paketten orijinal IP paketini çıkartarak koruduğu ağda IP paketinin hedef bilgisayara iletilmesini sağlar [11].



Şekil 1.1. Güvenli Ağ Geçitleri Arasında IPsec Tüneli Kurulması

Bilgisayarlar arasında IPsec kurulması durumunda IPsec tünelinin işletilmesi bilgisayarların sorumluluğunda olduğu için her iki bilgisayar da IPsec protokolünü desteklemek zorundadırlar. Şekil 1.2'de gösterilen bu senaryoda IPsec tünelinden gönderilen paketlerde kaynak ve hedef IP adresleri doğrudan uç noktada bulunan bilgisayarların IP adresleridir.



Şekil 1.2. Bilgisayarlar Arasında IPsec Tüneli Kurulması

1.3. Güvenlik Oturumu ve Anahtar Yönetimi

IPsec, tünel kurulacak iki uç nokta arasında hangi kripto algoritmalarının kullanılacağını ve bu algoritmalara ait anahtarların ne olacağını yönetebilecek bir mekanizmaya sahip değildir. Bu alt yapının IPsec protokolünün çalışabilmesi için önceden hazır olması gerekmektedir. Bütün IPsec gerçeklemeleri manuel ve

otomatikleştirilmiş güvenlik oturumu ve anahtar yönetimi sistemlerini desteklemelidir [5]. El ile manuel olarak güvenlik oturumunun yönetilmesi daha çok küçük ölçekli ağlarda uygulanabilmektedir. IPsec çalışırken el ile kripto anahtarlarının ve algoritmalarının değiştirilmesi pratik değildir. IPsec protokolü güvenlik oturumunun otomatik olarak yönetilebilmesi için IKEv2 protokolü önerilmektedir [5].

Anahtar yönetimi IPsec gibi bilgi gizliliği ve bütünlüğü sağlayan güvenlik uygulamaları için oldukça önemlidir. Anahtar ve güvenlik oturumu yönetim araçları sistemler arasındaki güvenlik protokollerinin kurulmasını, bakımını ve kontrolünü üstlenirler [6].

1.3.1. Kullanıcı tarafından güvenlik oturumu ve anahtar yönetimi

En basit anahtar ve güvenlik oturumu yönetimi şeklidir. İki güvenlik noktası arasında kullanılacak anahtarlar önceden belirlenmiştir. Güvenlik oturumu boyunca aynı şifreleme ve doğrulama anahtarları kullanılmaya devam edilir. Şifreli haberleşme için kullanılan anahtarlar ele geçirilirse anahtarlar değiştirilene kadar şifreli haberleşme çözülebilir.

Gerçek kullanımda manuel güvenlik oturumu yönetimi göründüğünden daha karmaşık ve hataya açık olabilmektedir. Sistemde kullanılan tüm cihazlar ayarlanarak sistemlere birbirleri ile haberleşmede kullanacakları anahtarlar ve kripto algoritmaları doğru olarak girilmelidir. Manuel güvenlik oturumu kullanılması IPsec'in esnekliğini engellemektedir. Anahtar değişimi ve güvenlik oturumuna özel anahtar yönetimi manuel güvenlik oturumunda mümkün olmamaktadır [6].

IPsec ile haberleşecek iki güvenli ağ geçidine de şifreleme anahtarları ve güvenlik oturumunda kullanılacak kripto algoritmaları ile ilgili ayarlamalar el ile girilir. Manuel anahtar yönetimi iyi ölçeklenemediği için küçük ve statik ağlarda kullanılabilirdir. Eğer IPsec ile VPN kurulacak ağ sayısı küçükse ve tüm ağlar tek bir yönetici tarafından yönetiliyorsa manuel anahtar yönetimi uygun olabilir. Bu senaryoda güvenli ağ geçidi sadece ayarlanan ağlardan gelen trafiği korurken diğer ağlar ile olan haberleşmeyi korumayabilir. Sadece seçilen haberleşme yönü güvenlik altına alınabilir. Manuel ağ yönetimi tekniğinde genellikle statik olarak yapılandırılmış simetrik anahtarlar kullanılır [7].

Örnek olarak 5 farklı ağ arasında IPsec kullanarak VPN kurmak istendiğini ve her ağın birbiri ile doğrudan bağlandığını düşünelim. Bu güvenli ağ haberleşmesi için her ağ arasında kurulacak VPN'de farklı anahtarlar kullanılırsa sistem için gerekli olan anahtar sayısı n(n-1)/2 formülü ile hesaplanabilir. Bu örnekte n ağ sayısını göstermek üzere toplam 10 adet farklı kripto anahtarı gerekmektedir. Eğer manuel güvenlik oturumu ile 25 adet ağ yönetilmek istenseydi ihtiyaç duyulan farklı anahtar sayısı 300 olacaktı.

1.3.2. Otomatikleştirilmiş güvenlik oturumu ve anahtar yönetimi

Manuel güvenlik oturumunun dezavantajlarını ortadan kaldırabilmek için otomatikleştirilmiş güvenlik oturumu ve anahtar yönetimi protokollerine ihtiyaç duyulmaktadır. IPsec'in geniş ve dinamik ağlarda kullanılarak anahtar değişiminin yapılabilmesi için standartlaştırılmış, ölçeklenebilir, otomatik bir güvenlik oturumu yönetim protokolüne ihtiyaç duyulmaktadır. IPsec için varsayılan otomatik anahtar yönetimi protokolü olarak IKEv2 seçilmiştir. Öte yandan başka anahtar yönetimi protokollerinin kullanılmasının önünde bir engel bulunmamaktadır.

IPsec protokolü için anahtar değişimi bir güvenlik oturumuna ait anahtarların ele geçirilmesi durumunda bu anahtarların yerine yenisini kullanarak güvenli haberleşmenin devamlılığın sağlanması amacıyla yapılmaktadır. Bu süreçte mevcut güvenlik oturumunun yanında yeni bir güvenlik oturumu oluşturulur. Yeni güvenlik oturumu devreye alındığında eski güvenlik oturumu kapatılarak haberleşmede kesinti olmasının önüne geçilir. Anahtar değişimleri belli periyotlarla yapılsa da bu periyot süreleri rastgele miktarlarda uzatılıp kısaltılarak tahmin edilemez bir davranış sergilemek saldırganlar karşısında nispeten değerli bir güvenlik yöntemi olarak kabul edilir [6].

IPsec için otomatik bir anahtar yönetimi protokolü kullanıldığında bu protokolün çıktıları tek güvenlik oturumunda kullanılacak birden çok güvenlik anahtarı ihtiyacını karşılayabilmektedir. Bu yetenek aynı zamanda IKE tarafından oluşturulan farklı güvenlik oturumuna farklı güvenlik anahtarları üretilmesini de sağlar. Eğer IPsec hem bütünlük hem gizlilik kontrolü görevlerini yerine getirecekse kullanılacak algoritmalar için en az dört farklı anahtar gerekmektedir.

1.4. IPsec Protokolleri

IPsec tek başına çalışan bir protokol değildir, IP ağında bütünüyle bir güvenlik çözümü sunan servis ve protokollerden oluşmaktadır. IPsec protokol paketini oluşturan iki ana bileşen vardır. İlki Kimlik Denetimi Başlığı (AH), veri gizliliği olmadan veri bütünlüğü ve kaynak doğrulaması sağlar. İkinci olarak Kapsüllenen Güvenlik Yükü (ESP), veri gizliliği sağlamasının yanı sıra konfigürasyona bağlı olarak AH'in sağladığı güvenlik özelliklerini de sağlar.

1.4.1. Kimlik denetimi başlığı protokolü

Bu protokol IPsec için veri bütünlüğü ve kaynak doğrulama servislerini sağlar. Alıcı tarafa ulaşan bir IP paketinin paket içinde yazan orijinal gönderici tarafından gönderilip gönderilmediğinin kontrolünü sağlar, paketin alıcı cihaza ulaştığı rota üzerindeki ara cihazların alınan paket içindeki veriyi değiştirip değiştirmediğini doğrular. Ayrıca bir IP paketinin yetkisiz bir kullanıcı tarafından ele geçirilerek yeniden gönderme saldırısı olarak adlandırılan paketi tekrar tekrar göndererek yapılan saldırılara karşı koruma sağlar [8]. AH protokolü IP protokolünün üzerinde çalışır. IANA tarafından IP protokol numarası 51 olarak belirlenmiştir. IP paketinin IP başlığında ağda iletim sırasında değişebilecek olan servis tipi, bayraklar, bayrak ofseti, TTL ve IPv4 başlık sağlaması alanları hariç IP paketinin tamamını koruma altına alır. Şekil 1.3'te kimlik denetimi başlığı protokolünün paket başlığı gösterilmiştir.

1 Bayt	1 Bayt	2 Bayt
Sonraki Başlık	Yük Uzunluğu	Rezerve
Güvenlik Parametre İndeksi		
Sıra Numarası		
Bütünlük Kontrol Değeri		

Şekil 1.3. Kimlik Denetimi Başlığı Paket Yapısı

Sonraki başlık alanı toplam 1 bayt uzunluğundadır. Kimlik denetimi başlığından sonra gelen protokolün tipini belirtir. Bu alana yazılacak protokol numarası IANA tarafından tanımlanmıştır. IANA tarafından tanımlanan protokol numaralarına örnek olarak ICMP için 1, IPv4 için 4, TCP için 6, IPv6 için 41, SCTP için 132 tanımlanmıştır.

Yük uzunluğu alanı toplam 1 bayt uzunluğunda tanımlanmaktadır. IP paketine eklenen kimlik denetimi başlığının toplam boyutunun 2 eksiğinin 4 bayt karşılığı yazılır. Örnek olarak 96 bit bütünlük kontrol verisi var ise bu alanda 4 yazılır. 1 word 32 bit uzunluğu ifade etmek üzere, sonraki Başlık + Yük Uzunluğu + Rezerve 1 word tutmaktadır. SPI ve sıra numarası alanları toplam 2 word uzunluğundadır. 96 bitlik bütünlük kontrol versi de toplam 3 word tutmaktadır.

Rezerve alanı toplam 16 bit uzunluğundadır. Gelecekte kullanılabilecek durumlar için ayrılmıştır. Şu anki kullanımda bu alan 0 ile doldurulmalıdır.

Güvenlik parametre indeksi (SPI) alanı 32 bit uzunluğundadır. Alınan IP paketinin hangi güvenlik oturumuna ait olduğunun alıcı tarafta belirlenebilmesi için gönderici taraf tarafından doldurulur.

Sıra numarası alanı toplam 32 bit uzunluğundadır. Her güvenlik oturumu için ayrı ayrı sayaç tutularak gönderilen paket ile bu sayaç bir arttırılır. Tekrar saldırılarından korunmak için kullanılan bu sayaçlar eğer maksimum değerin üzerine çıkmışsa yeniden bir güvenlik oturumu kurmak gereklidir. Bu alanın doldurulması alıcı tarafın tekrar saldırılarına karşı güvenlik önlemi almaması durumunda bile zorunludur. Sıra numarası alanının kullanılması alıcının tercihine bırakılmıştır ancak tüm AH destekleyen sistemler sıra numarası üretimini ve doğrulamasını desteklemelidirler. Güvenlik oturumu kurulduğu anda alıcı ve gönderici taraflarda sıra numaraları 0 ile ilklendirilir böylece gönderilen ilk paketin sıra numarası 1 olur. Tekrar saldırısı koruması etkinleştirilmişse sıra numarası 2^{32} paket sonunda sıfırlanmaz, mevcut güvenlik oturumu sonlandırılarak yeni bir güvenlik oturumu kurulur.

Bütünlük kontrol değeri alanı 32 bit ve katları olmak üzere değişik uzunluklarda olabilmektedir. Bütünlük kontrol değeri üretimi için kullanılan özet çıkarma algoritmalarının çıkış uzunluğuna bağlı olarak değişmektedir. Algoritma çıktı uzunluğu 32 bit ve katlarında değilse dolgu işlemi yapılarak bütünlük kontrol değeri 32 bit ve katları uzunluğunda olacak şekilde ayarlanır. Kullanılacak algoritma güvenlik oturumu kurulurken tünel uç noktaları tarafından belirlenir.

1.4.2. Kapsüllenen güvenlik yükü

Bu protokol IPsec için veri bütünlüğü ve kaynak doğrulama servislerini sağlamanın yanı sıra IP paketini şifreleyerek veri gizliliği de sağlar. AH protokolünün aksine eğer ESP protokolü taşıma modunda kullanılırsa tüm IP paketine koruma sağlayamaz. Tüm IP paketinde koruma isteniyorsa ESP protokolü tünel modunda kullanılmalıdır. ESP protokolü IP protokolünün üzerinde çalışır. IANA tarafından protokol numarası 50 olarak belirlenmiştir. Şekil 1.4'te kapsüllenen güvenlik yükü protokolünün formatı gösterilmiştir.

Şekil 1.4. Kapsüllenen Güvenlik Yükü Paket Yapısı

ESP paketi 4 bayt uzunluğundaki SPI ve sıra numarası alanları ile başlamaktadır. Bu iki alandan sonra ESP protokolünün koruma altına aldığı yük verisi alanı gelmektedir. Yük verisi alanını dolgu, dolgu uzunluğu ve sonraki başlık alanları takip etmektedir. Eğer ESP protokolü bütünlük koruması da sağlayacaksa bu alanların sonuna bütünlük kontrol değeri eklenerek ESP paketinin formatı tamamlanmış olur. Dolgu alanı, dolgu uzunluğu alanı ve sonraki başlık alanı ESP trailer olarak adlandırılır.

ESP protokolü bütünlük kontrolü sağlamak için kullanılıyorsa güvenlik parametre indeksi ile sonraki başlık alanının sonuna kadar olan alandaki veriler kullanılarak bütünlük kontrol değeri hesaplanmaktadır. ESP protokolü veri gizliliği sağlamak amacıyla kullanıldığından yük verisi alanının başından başlanarak sonraki başlık alanın sonuna kadar olan alandaki veriler şifrelenerek veri gizliliği sağlanır. ESP protokolünün veri gizliliğini sağlayabilmek için kullandığı bazı şifreleme algoritmaları çalışabilmek için ilklendirme vektörüne ihtiyaç duymaktadırlar. ESP protokolünde ilklendirme vektörü kullanılması gereken bir şifreleme algoritması kullanıldığında şifreli verinin alıcı tarafta çözülebilmesi için yük verisi alanının üzerine ilklendirme

vektörü alanı eklenerek paketi şifrelemekte kullanılan ilklendirme vektörü bu alanda gönderilmektedir.

Güvenlik parametre indeksi ve sıra numarası alanlarının işlevleri AH protokolü ile aynıdır. ESP protokolünde kullanılan şifreleme algoritması blok şifreleme yöntemiyle çalışıyorsa algoritmaya şifrelemesi için verilecek verinin uzunluğu algoritmanın blok uzunluğunun katlarında olması gerekmektedir. Şifrelenecek veri kullanılan şifreleme algoritmasının blok uzunluğunun katlarında değilse verinin sonundaki dolgu alanı doldurularak veri uzunluğunun blok uzunlu veya katlarında olması sağlanmaktadır.

Dolgu uzunluğu alanı 1 bayt uzunluğundadır. Giriş verisinin uzunluğunu şifreleme algoritmasının blok uzunluğuna uydurmak için dolgu alanına kaç byte dolgu yapıldığı bu alana yazılmaktadır.

Sonraki başlık alanı 1 bayt uzunluğundadır. Bu alana ESP başlığından sonra gelen protokolün IANA tarafından belirlenen protokol numarası yazılmaktadır.

1.5. IPsec Modları

ESP ve AH protokolleri iletim ve tünel modu olmak üzere iki farklı modda çalışabilmektedirler. Bu protokoller IP paketine güvenlik bilgisini içeren bir başlık ekleyerek çalışmaktadır. Protokollerin çalıştıkları mod IP paketinin hangi kısımlarının korunduğunu ve korumu işlemi için paket üzerinde yapılan eklemelerin nasıl yapılacağını belirler.

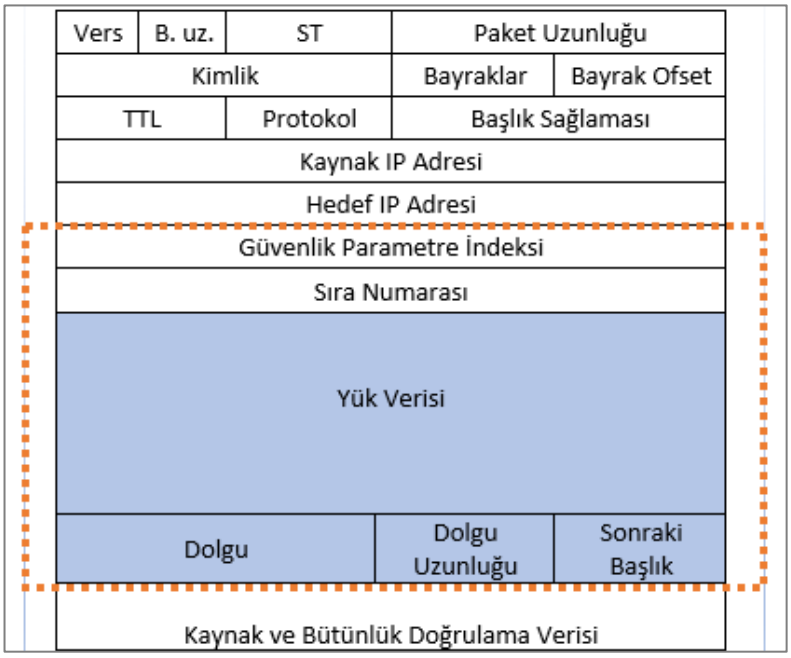
1.5.1. İletim modu

İletim modunda IP paketinin taşıdığı yük kısmı koruma altına alınır. AH ve/veya ESP başlığı, IP başlığı ile IP paketinin yükü arasına yerleştirilerek IP katmanının üzerinde çalışan protokollerde koruma sağlamaktadır [9]. Gerçek IP başlığında bir değişiklik yapılmadığı için paketi gönderen ve alan cihazların IP adresleri korumasız olarak gönderilmektedir. Ağ içerisindeki bilgisayarlar tarafından IPsec protokolü iletim modunda kullanılabilmektedir. Uçtan uca haberleşmenin korunması amacıyla kullanılan iletim modunda, IPsec protokolü ağ geçitlerinde veya yönlendiricilerde değil, Şekil 1.5'te gösterildiği gibi IPsec tarafından korunan son cihazda

çalıştırılmaktadır. İletim modunda orijinal IP başlığında bir değişiklik yapılmadığı için iç IP adresi ve dış IP adresi kavramlarından bahsedilemez.

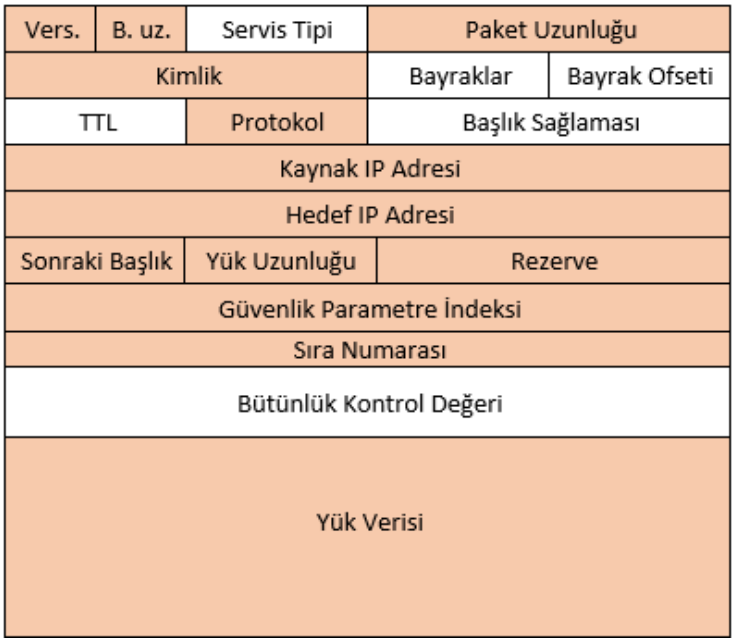
Şekil 1.5. İletim Modunda IPsec Protokolü

ESP protokolü iletim modunda kullanıldığında orijinal IP başlığı ile IP paketinin yükü arasına bir ESP başlığı getirilmektedir. Şekil 1.6'da örnek bir IP paketinin ESP protokolü ile iletim modunda işlenmesi gösterilmiştir. Koyu arka plan ile gösterilen kısımlar şifrelenerek veri gizliliği sağlanmaktadır. Kesikli çizgiler ile gösterilen kısımlarda ise veri bütünlüğü ve kaynak doğrulaması sağlanmaktadır. Kesikli çizgiler ile gösterilen kısımlar kullanılarak üretilen doğrulama verisi IP paketinin sonundaki kaynak ve bütünlük doğrulama verisi alanına eklenerek gönderilmektedir. İletim modunda ESP protokolü IP başlığından sonrasını koruma altına aldığı için IP paketinin yük kısmı yani IP protokolünün üzerinde çalışan TPC/UDP gibi protokollere ait veriler koruma altına alınmaktadır.



Şekil 1.6. İletim Modunda ESP Protokolünün Kullanımı

AH protokolü iletim modunda kullanıldığında ise ESP protokolünde olduğu gibi orijinal IP başlığı ile IP paketinin yükü arasına bir AH başlığı getirilmektedir. İletim modunda AH protokolü IP başlığı da dahil olmak üzere tüm IP paketini koruma altına alınmaktadır. Koruma altına alınmayan bölümler ise paketin ağda ilerlerken üzerinden geçtiği ağ cihazları tarafında değiştirilmesi gereken TTL, Servis Tipi, IP Başlık Doğrulaması gibi kısımlardır. Şekil 1.7'de örnek bir IP paketinin AH protokolü ile iletim modunda işlenmesi gösterilmiştir. Koyu arka plan ile işaretli kısımlar AH protokolü tarafından veri bütünlüğü ve kaynak koruması altına alınmaktadır.

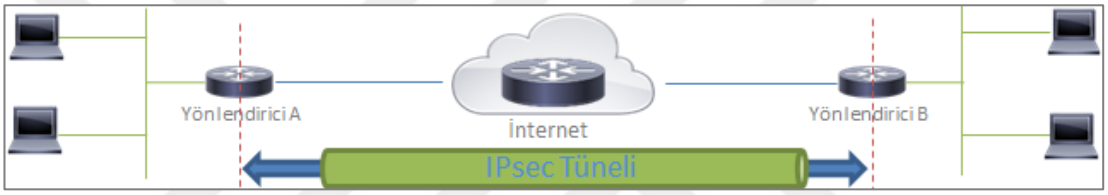


Şekil 1.7. İletim Modunda AH Protokolünün Kullanımı

1.5.2. Tünel modu

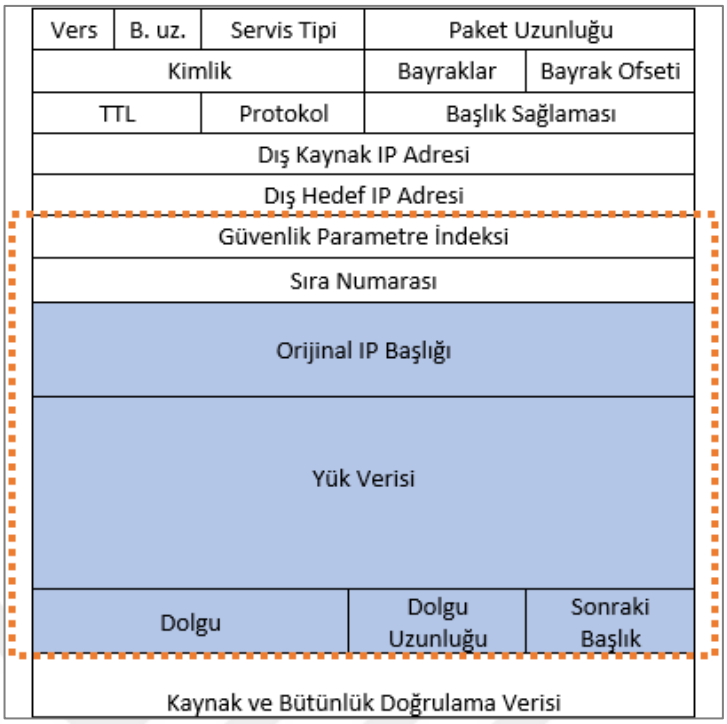
Tünel modunda IPsec protokolü tüm IP paketini kapsülleyerek koruma altına almaktadır. AH veya ESP başlığı orijinal IP başlığının önüne getirilerek orijinal IP paketi yeni bir IP paketi olarak kapsüllenmektedir. Bu mod yerel ağdaki cihazları koruma altına almak amacıyla güvenli ağ geçitleri arasında çalıştırılabildiği gibi bir güvenli ağ geçidi ile bilgisayar arasında da çalıştırılabilmektedir [8]. Şekil 1.8'de gösterilen ağda, ağa giriş-çıkış yapan paketler güvenli ağ geçidi üzerinden geçerek paketler için güvenlik ihtiyacı karşılanmış olduğundan yerel ağda çalışan cihazlarda IPsec çalıştırmak gerekmemektedir.

Tünel modunda orijinal IP başlığında yazmakta olan hedef ve kaynak IP adresleri iç IP adresi olarak adlandırılmaktadır. Orijinal IP paketini kapsüllemek için kullanılan IP başlığında yazmakta olan hedef ve kaynak IP adresleri ise dış IP adresi olarak isimlendirilmektedirler. İç IP adresleri paketi gönderen gerçek gönderici ve alıcı bilgisayarlara ait IP adresleridir. Dış IP adresi ile bu paketlerin güvenliğinden sorumlu olan güvenli ağ geçitlerine ait IP adresleridir. Tünel modunda IP paketinin gönderici ve alıcı adresleri de IPsec protokolü tarafından koruma altına alınmaktadır. Böylelikle ağ üzerinde IP paketi ilerlerken geçtiği yönlendirici gibi cihazlar alıcı ve gönderici IP adreslerini elde edemezler. Orijinal IP adreslerinin gizlenmesi güvenliğe katkıda bulunmaktadır. Genellikle farklı ağlar arası haberleşme amacıyla kullanılan tünel modu ağlar arasında VPN oluşturmak için kullanılmaktadır.



Şekil 1.8. Tünel Modunda IPsec Protokolü

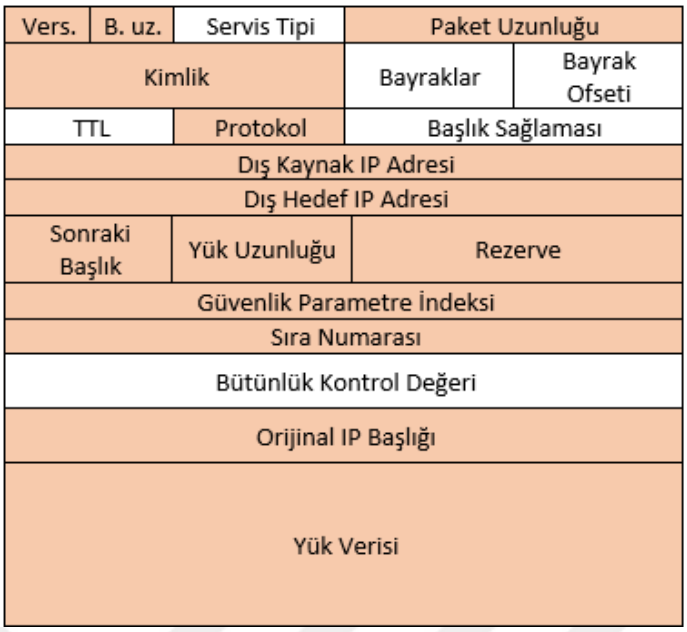
ESP protokolü tünel modunda kullanıldığında orijinal IP başlığının üzerinde bir ESP başlığı getirilir. ESP başlığından sonra ise bir dış IP başlığı eklenerek orijinal IP paketinin tamamı kapsüllenmektedir.



Şekil 1.9. Tünel Modunda ESP Protokolü

Şekil 1.9'da örnek bir IP paketinin ESP protokolü ile tünel modunda kapsüllenmesi gösterilmiştir. Koyu arka plan ile gösterilen kısımlar şifrelenerek veri gizliliği sağlanmaktadır. Kesikli çizgiler işaretli kısımlarda ise veri bütünlüğü ve kaynak doğrulaması sağlanır. Veri bütünlüğü ve kaynak doğrulaması için hesaplanan kontrol verisi paketin sonuna eklenerek gönderilmektedir. Tünel modunda ESP protokolü IP başlığı da dahil olmak üzere tüm IP paketini koruma altına almaktadır.

AH protokolü tünel modunda kullanıldığında ise ESP protokolünde olduğu gibi orijinal IP başlığının önüne bir AH başlığı getirilir. AH başlığının üzerine güvenli ağ geçitlerine ait IP adreslerinin yazılı olduğu dış IP başlığı getirilerek orjinal IP paketi kapsüllenir. Tünel modunda AH protokolü IP başlığı da dahil olmak üzere tüm IP paketini ve yeni eklenen dış IP başlığını koruma altına alır. Koruma altına alınmayan bölümler ise paketin ağda ilerlerken üzerinden geçtiği ağ cihazları tarafında değiştirilmesi gereken TTL, servis tipi, IP başlık sağlaması gibi kısımlardır. Şekil 1.10'da örnek bir IP paketinin AH protokolü ile iletim modunda işlenmesi gösterilmiştir. Koyu arka plan ile işaretli kısımlar AH protokolü tarafından veri bütünlüğü ve kaynak koruması altına alınmaktadır.



Şekil 1.10. Tünel Modunda AH Protokolü

2. IKEV2 PROTOKOLÜ

IPsec protokolünün çalışabilmesi için tünel uç noktalarında bulunan cihazlar arasında bir güvenlik oturumu hali hazırda bulunmalıdır. Güvenlik oturumu IPsec tüneli kuracak cihazlar arasında hangi kripto algoritmalarının kullanılacağının belirlenip cihazların bu güvenli oturum boyunca kullanacakları anahtarları paylaşmaları durumudur. Güvenlik oturumu yönetimi elle yapılabileceği gibi IPsec ile beraber bir güvenlik oturumu yönetim protokolü de kullanılabilir.

Güvenlik oturumunu manuel olarak yönetmek küçük ağlarda çalışan sistemler için pratik olsa da iyi ölçeklenemediği için ağda IPsec protokolü çalıştıran cihaz sayısı arttıkça pratik olmaktan çıkar. Bu nedenle ölçeklenebilecek ağlarda güvenlik oturumunu otomatikleştirip dinamik olarak yönetebilecek bir güvenlik oturumu yönetim protokolüne ihtiyaç duyulmaktadır. IPsec protokolü için varsayılan otomatik güvenlik oturumu yönetim protokolü IKEv2'dir [10].

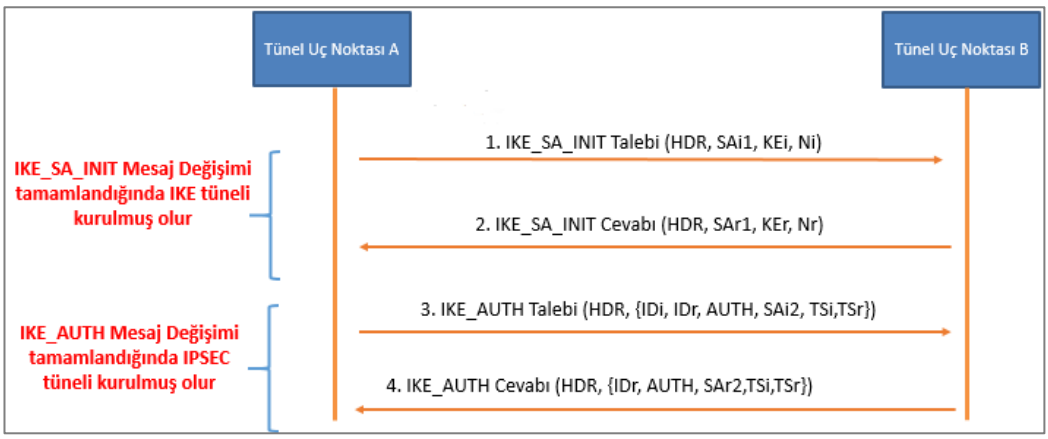
IKE tünel uç noktaları arasında karşılıklı kimlik doğrulaması gerçekleştirerek IKE güvenlik oturumunu (Security Association- SA) oluşturur. Ardından kurulan IKE SA'yı kullanarak IP trafiğinin güvenliğini sağlamak amacıyla kullanılacak olan ESP veya AH protokolleri için gerekli olan SA IKE protokolü tarafından kurulmaktadır. ESP veya AH protokolleri için kurulan SA'ya IKE protokolünde "Child SA" ismi verilir [10].

Uç noktalar arasında talep ve cevap mesaj çiftleri ile IKE haberleşmesi gerçekleştirilir. Karşılıklı gönderilen mesaj çiftleri "exchange" olarak isimlendirilir. IKE SA kurulurken gerçekleştirilen ilk iki mesaj değişimine IKE_SA_INIT ve IKE_AUTH mesaj değişimi olarak adlandırılmaktadır. Bu iki mesaj değişiminden sonra CREATE_CHILD_SA ve INFORMATIONAL mesaj değişimleri gerçekleştirilebilmektedir.

IKE SA ve IPsec SA'nın kurulması için ilk iki mesaj değişimi yeterli olmaktadır [11]. İlk iki mesaj değişiminden sonraki mesaj değişimleri IPsec uç noktaları arasında birden fazla Child SA kurmak, IKE ve IPsec güvenlik oturumlarının anahtarlarını yenilemek ve IKE tünelinin bakımını gerçekleştirmek amacıyla gerçekleştirilmektedir. Çoğu durumda IPsec uç noktaları için tek bir Child SA yeterli olduğu için ilk iki mesaj değişiminden sonra başka bir mesaj değişimi gerçekleştirilmemektedir.

IKE mesajlaşmalarında bir talep mesajına mutlaka cevap mesajı verilmelidir. Talep mesajlarına cevap gelip gelmediğinin kontrolü talep mesajını gönderen istemci uç noktanın sorumluluğundadır. Eğer talep mesajı tekrar iletildiği halde cevap gelmez ise IKE bağlantısı istemci uç nokta tarafından kapatılabilir.

İlk mesaj değişimi olan IKE_SA_INIT mesaj değişiminde IKE güvenlik oturumumun güvenlik parametreleri, nonce değerleri, Diffie-Hellman değerleri karşılıklı olarak uç noktalar arasında iletilmektedir. İkinci mesaj değişimi olan IKE_AUTH mesaj değişiminde uç noktaların kimlik bilgileri, kimlik doğrulama verileri, IPsec güvenlik oturumunun (Child SA) güvenlik parametreleri ve IPsec tünelinden geçecek IP trafiğinin belirlendiği trafik seçiciler karşılıklı olarak uç noktalar arasında iletilmektedir [11]. IKE SA ve IPsec SA kurulumu için gerekli olan ilk iki mesaj değişimi sırasında uç noktalar arasında gerçekleştirilen mesajlaşmalar Şekil 2.1'de gösterilmiştir. Şekil 2.1'de IKE_AUTH mesaj değişiminde şifrelenmiş yük içinde gönderilen yükler süslü parantez içinde gösterilmiştir.



Şekil 2.1. IKE_SA_INIT ve IKE_AUTH Mesaj Değişimleri

İlk iki mesaj değişiminden sonra gerçekleşebilecek olan mesaj değişimleri için beklenen bir sıralama yoktur. CREATE_CHILD_SA mesaj değişimi ile yeni bir Child

SA kurulabilir. INFORMATIONAL mesaj değişimi ile SA'lar silinebilir, hata durumları uç noktalar arasında raporlanabilir veya IKE tüneli ile alakalı bakım fonksiyonları yerine getirilebilir. Örnek olarak INFORMATIONAL mesajı tipi uç noktaların IKE tünelini ayakta tutup tutmadıklarının kontrolü amacıyla içeriğinde yük olmadan gönderilebilmektedir.

2.1. IKEv2 Protokolü Kullanım Senaryoları

2.1.1. Güvenli ağ geçitleri arasında tünel modu

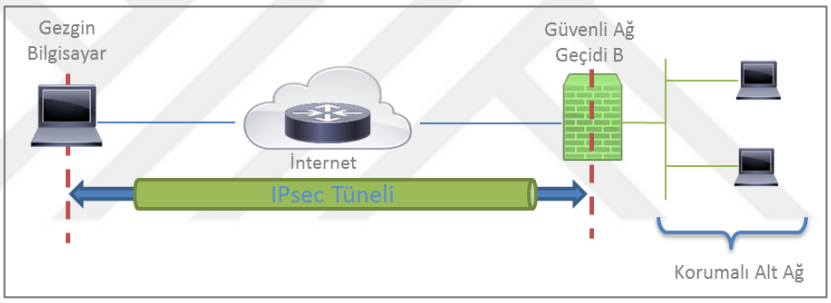
Bu kullanım senaryosunda korunan ağ içerisindeki bilgisayarlardan hiçbirisi IPsec protokolünü çalıştırmaz. Bunun yerine IP paketlerinin yolu üzerindeki IPsec tünelinden sorumlu cihazlar (Güvenli Ağ Geçitleri) ağ güvenliğinden sorumludurlar. Uç noktalar güvenli ağ geçitleri tarafından gerçekleştirilen ağ korumasından etkilenmezler. IPsec için gerekli paket işleme işinin yapılması için IP paketlerinin güvenli ağ geçitlerine iletilmesi yeterlidir. IPsec tünel modunda çalıştığı için korumalı ağ içerindeki cihazların gönderdikleri IP paketleri güvenli ağ geçitleri tarafından kapsüllenerek hedef ağa ait güvenli ağ geçidine gönderilir. Kapsüllenmiş IP paketini alan güvenli ağ geçidi, IPsec protokolüne ait kapsül kısmını paketten ayırarak orijinal IP paketini korumalı ağdaki paketin gerçek alıcısına yönlendirir. IPsec Şekil 1.2'de gösterilen bu mod ile kullanıldığında IKE protokolü sadece güvenli ağ geçitlerinde çalışmaktadır.

2.1.2. Uç noktalar arasında iletim modu

Bu kullanım senaryosunda ağ içerisindeki uç noktalarda IPsec protokolü çalıştırılarak ağ katmanı güvenliğinden uç noktadaki cihazın kendisi sorumlu olur. Bu güvenlik oturumunda ESP ve/veya AH protokolü iletim modunda kullanıldığı için paketler IP paketleri kapsüllenmezler. IP paketleri kapsüllenmediği için "iç IP başlığı" ve "dış IP başlığı" kavramlarından bahsedilemez [3]. Bu modda IPsec çalıştırıldığında uç noktalardaki cihazlar uygulama katmanındaki erişim kontrollerini IPsec'in kimlik doğrulamasına bağlı olarak çalıştırabilir. IPsec Şekil 1.3'de gösterilen bu mod ile kullanıldığında IKE protokolü ağ içindeki bilgisayarlarda çalışmaktadır.

2.1.3. Uç nokta ile güvenli ağ geçidi arasında tünel modu

Şekil 2.2'de gösterilen gezgin bir bilgisayarın yerel bir uzak ağa IPsec kullanarak korumalı tünel üzerinden bağlanması durumunda bu senaryo işletilmektedir. Gezgin sistem uzak yerel ağdaki bilgisayarlar ile güvenli bir bağlantı üzerinden haberleşebileceği gibi, uzak yerel ağ İnternet tabanlı saldırılara karşı bir güvenlik duvarı tarafından korunuyorsa bütün İnternet trafiğini IPsec tüneli üzerinden gerçekleştirerek güvenlik duvarının sağladığı avantajlardan da faydalanabilmektedir. İki durumda da tünel modunda çalışıldığı için gezgin bilgisayar güvenli ağ geçidi ile haberleşebilmek için uzak yerel ağa ait bir iç IP adresine ihtiyaç duyacaktır. Böylece kapsüllenen IP paketlerinin iç IP adresi gezgin bilgisayara uzak yerel ağ tarafından atanan IP adresi olmaktadır. Dış IP adresi ise gezgin bilgisayarın İnternete bağlandığı konuma göre değişiklik gösterebilmektedir.



Şekil 2.2. Gezgin Cihaz ile Güvenli Ağ Geçidi Arasında Tünel Modu

2.2. IKEv2 İlklendirme Mesajları

IKE tünelinin kurulabilmesi için IKE_SA_INIT ve IKE_AUTH mesaj değişimlerinin tamamlanması gerekmektedir. Her mesaj değişimi bir talep/cevap mesaj çiftinden oluştuğu için IKE tünelinin kurulabilmesi için IKE tünelini kuracak cihazlar arasında toplam dört tane IKE paketi alışverişi yapılmaktadır. Cookie gibi mesajlaşma miktarını arttıran yükler IKE tüneli kuran cihazlar tarafından gönderilirse IKE tüneli kurabilmek için gerek mesaj sayısı artabilmektedir. IKE_SA_INIT mesaj değişimi tamamlandıktan sonra IKE tüneli kurulduğu için IKE_SA_INIT mesaj değişimi için gönderilen paketler şifrelenmeden açıktan gönderilmektedirler. IKE tüneli kurulduktan sonra yapılan tüm mesajlaşmalar şifrelenerek IKE tüneli içerisinden

gönderilmektedirler. Şekil 2.1'de IKE ve IPsec tünellerinin kurulma sıraları gösterilmiştir.

Her IKE mesajı IKE başlığı kısmında bir mesaj kimlik numarası (Message ID) taşımaktadır. Bu numara ile talep ve cevap mesajları birbirleri ile eşleştirilir ve IKE paketlerinin kaybolup iletilememesi durumunda aynı mesajı tekrar gönderebilmek amacıyla kullanılmaktadır.

2.2.1. IKE_SA_INIT mesaj değişimi

İlk mesaj değişimi olan IKE_SA_INIT mesaj değişiminde IKE tünelinde kullanılacak kriptografik algoritmalar, nonce değeri ve Diffie-Hellman anahtar değişimi için gerekli olan sayılar hesaplanıp gönderilmektedir. IKE_SA_INIT mesaj değişimi tamamlandıktan sonra yapılan tüm mesaj değişimleri IKE_SA_INIT mesaj değişiminde belirlenen kriptografik algoritmalar ve anahtarlar tarafından korumalı olarak gerçekleştirilmektedir.

Şekil 2.1'de IKE tüneli kurulması için gerekli mesaj değişimleri ve içerikleri gösterilmiştir. IKE tünelini kurmak için gönderilen IKE SA INIT talep mesajının içeriğinde yer alan protokol başlıkları ve yükler aşağıda açıklanmıştır.

- HDR: IKE başlığını ifade eder. IKE başlığında Güvenlik Parametre İndeksi (SPI), IKE versiyon numarası, mesaj değişim tipi, mesaj numarası ve bayraklar yer almaktadır.
- SAi1: IKE bağlantısını başlatan istemcinin IKE Tünelinde kullanabilmek için desteklediği ve kullanmayı tercih ettiği şifreleme algoritmaları, doğrulama algoritmaları, sözde rastgele fonksiyonları (PRF) ve Diffie-Hellman grup numaraları iletilir.
- KEi: Diffie-Hellman anahtar değişimini gerçekleştirmek için iletilir. Diffie-Hellman grubu, rastgele üretilen ilklendirici uç noktaya ait gizli anahtar ve Diffie-Hellman üreteci kullanılarak üretilir.
- Ni: IKE ve IPSEC tünellerinde kullanılacak şifreleme ve doğrulama algoritmaları için anahtarların üretilmesinde ve şifresiz olarak iletilen IKE_SA_INIT mesajlarını doğrulamak için bir sonraki IKE_AUTH mesaj değişiminde kullanılır.

Cevap veren uç nokta istemci uç nokta tarafından önerilen kripto algoritmalarından cevap veren uç nokta için seçmesi ayarlanan kripto algoritmalarını seçerek IKE_SA_INIT talep mesajına cevap gönderir. Cevap mesajının içinde cevap veren uç noktanın hesapladığı Diffie-Hellman değeri ve nonce değeri de gönderilir. Bu noktada IKE SA için gerekli olan anahtarların türetildiği SKEYSEED değeri iki uç noktada da hesaplanmış olur.

IKE mesajlarının şifrelenmesi ve bütünlük kontrolünün sağlanması için kullanılan kripto algoritmalarının kullandığı anahtarlar SKEYSEED değerinden türetilen SK_e (encryption) ve SK_a (authentication) anahtarlarıdır. SK_e ve SK_a anahtarları alış ve atış yönleri için ayrı ayrı iki uç noktada da hesaplanır. İstemci uç nokta cevap veren uç noktanın hangi Diffie-Hellman grubunu kullanmak isteyeceğini tahmin ederek KEi değerini bu tahmine göre hesaplayıp gönderir. Eğer cevap veren uç nokta istemci uç noktanın tahmin ettiğinden farklı bir Diffie-Hellman grubu kullanmayı tercih ediyor ise hangi Diffie-Hellman grubunu kullanması gerektiğini INVALID_KE_PAYLOAD tipinde bir bilgilendirme mesajı ile cevap veren uç nokta tarafından gönderir. İstemci uç nokta cevap veren uç noktanın tercih ettiği DH grubuna göre KEi değerini yeniden hesaplayarak göndermelidir.

2.2.2. IKE_AUTH mesaj değişimi

IKE_AUTH mesaj değişimi ile IKE oturumunun önceki mesajlaşmalarının kimlik doğrulaması sağlanır, kimlik bilgileri ve sertifikalar iletilerek ilk Child SA kurulur. IKE AUTH mesajının içeriği IKE_SA_INIT mesaj değişimi sırasında üretilen anahtarlar ile şifrelenir ve veri bütünlüğü kontrolü sağlanır. Böylece ağ üzerinde izinsiz olarak uç noktalar arasındaki haberleşmeyi dinleyen cihazlardan IKE AUTH mesajının içeriği gizlenmiş olur. Ortadaki adam saldırısı yapılır ise istemci uç noktanın gönderdiği şifreli IKE_AUTH mesajı çözülüp içeriği okunabilir ancak ortadaki adam kimlik doğrulamasını tamamlayamayacağı için IKE ve IPsec tünelleri kurulamamaktadır [11].

IKE_AUTH mesaj değişimi ile istemci uç nokta kimlik bilgilerini IDi yükü ile iletir. İstemci uç noktanın IDr yükünü iletmesi zorunlu değildir. IDr yükü ile cevap veren uç noktada hangi responder ID ile IKE tüneli kurulmak istendiğini belirtir. Böylece cevap veren uç nokta birden çok ID ile çalışıyorsa istemcinin hangi ID ile IKE SA kurmak

istediği belirtilmiş olur. Kimlik doğrulama için PSK kullanılıyor ise hangi PSK'nın kurulacak olan IKE SA'da kullanılacağı IDr yükü ile gönderilen kimlik bilgisi kullanılarak cevap veren uç noktada belirlenmektedir. Eğer istemci uç noktanın belirttiği IDr cevap veren uç nokta tarafta kabul edilmez ise istemci uç nokta farklı bir IDr yükü göndererek IKE_AUTH mesaj değişimini tamamlayabilir. Bu durumda cevap veren uç noktanın gönderdiği IDr istemci uç noktada kabul edilmez ise IKE SA sonlandırılmaktadır.

IKE tünelini kurmak için gönderilen IKE AUTH talep mesajı ile gönderilen yükler Şekil 2.1'de gösterilmiştir. Bu mesaj ile gönderilen yükler aşağıda açıklanmıştır.

- HDR: IKE başlığını ifade eder. IKE başlığında güvenlik parametre indeksi(SPI), IKE versiyon numarası, mesaj değişim tipi, mesaj numarası ve bayraklar yer almaktadır.
- IDi: IKE bağlantısını başlatan istemci uç noktanın kimlik bilgisini bu yük ile iletilir.
- IDr: IKE bağlantısını başlatan istemci uç noktada cevap veren uç noktayı tanımlayan kimlik bilgisini bu yük ile iletilir.
- AUTH: İstemci uç noktanın kimliğinin doğrulanabilmesi için ürettiği oturum doğrulama verisi bu yük ile iletilir.
- SAi2: IKE bağlantısını başlatan istemci uç noktanın IPsec Tünelinde kullanabilmek için desteklediği şifreleme algoritmaları ve doğrulama algoritmaları bu yük ile iletilir.
- TSi: İstemci uç noktanın IPSEC tüneli içinde kullanacağı IP adresi aralığını bu yük ile iletilmektedir.
- TSr: Cevap veren uç noktanın IPSEC tüneli içinde kullanacağı IP adresi aralığı bu yük ile iletilmektedir.

Cevap veren uç nokta kimlik bilgilerini IDr yükü ile birlikte gönderir. Kimlik doğrulamasını ve IKE_AUTH mesajının bütünlük korumasını sağlayarak AUTH yükünü hesaplayıp gönderir. Uç noktalar arasında kimlik doğrulama için PSK kullanılmışsa ID yükündeki isimler AUTH yükünü oluşturmakta kullanılmaktadır.

IKE_AUTH mesaj değişimi sırasında oluşabilecek IPsec tüneli kurulması ile ilgili bir hata IPsec tünelinin kurulmasına engel olsa bile IKE SA çalışmaya devam edebilmektedir. IPsec tünelin kurulamamasına yol açacak hata durumlarında

INFORMATIONAL tipinde bir mesaj değişimi ile hatanın sebebi uç noktalar arasında paylaşılmaktadır. Notify mesajı ile iletilebilecek hata mesajlarından bazıları aşağıda açıklanmıştır.

- NO_PROPOSAL_CHOSEN: İstemci uç nokta tarafından gönderilen IPsec SA tekliflerinin cevap veren uç nokta tarafından kabul edilmemesi durumunda cevap veren uç nokta tarafından gönderilir.
- TS_UNACCEPTABLE: IPsec tüneli içerisinden geçecek IP trafiğine ait IP adresi aralığının cevap veren uç nokta tarafından kabul edilmemesi durumunda cevap veren uç nokta tarafından gönderilir.
- SINGLE_PAIR_REQUIRED: İstemci uç nokta tarafından birden fazla sunulan trafik seçicinin cevap veren uç nokta tarafından desteklenmemesi durumunda cevap veren uç nokta tarafından gönderilir. Cevap veren uç nokta sadece bir çift trafik seçici desteklemektedir.

IKE_AUTH mesaj değişimi sırasında oluşabilecek IKE SA kurulması ile ilgili bir hata IKE SA'nın kapatılarak IPsec ve IKE tünelinin kurulmasını engeller. Bu durumda gönderilecek hata mesajları da şifreli ve veri bütünlüğü korumalı olarak iletilmektedirler.

IKE_AUTH mesaj değişimi sırasında anahtar değişimi ve Nonce yükleri iletilmez. Bu nedenle IKE_AUTH mesaj değişimi mesajlarında iletilen güvenlik oturumu yükünde Diffie-Hellman grubu iletilmemektedir.

2.3. Diğer Mesaj Değişimleri

2.3.1. CREATE_CHILD_SA mesaj değişimi

CREATE_CHILD_SA mesaj değişimi yeni bir Child SA kurarak hem IKE SA'nın hem de Child SA'nın anahtarlarının yenilenmesini sağlamak amacıyla kullanılmaktadır. CREATE_CHILD_SA mesaj değişimi IKE_SA_INIT ve IKE_AUTH mesaj değişimleri tamamlandıktan sonra cevap veren veya istemci uç noktaların herhangi biri tarafından başlatılabilmektedir. IKE protokolünün çalıştırıldığı uç noktalarda istenirse IKE SA için CREATE_CHILD_SA talepleri reddedilebilir.

Güvenlik oturumlarında anahtar değişiminin yapılabilmesi için eski güvenlik oturumu kapatılmadan önce yenisi kurulmaktadır. Yeni güvenlik oturumu başarılı bir şekilde kurulduktan sonra eskisi kapatılarak IPsec veya IKE tünellerinden gerçekleşen haberleşmenin kesintiye uğramasının önüne geçilmektedir.

Normal şartlarda Child SA'nın anahtar değişimi sırasında anahtar değişimi yükü gönderilmemektedir. CREATE_CHILD_SA mesaj değişimi sistem ayarlarına göre daha yüksek güvenlik sağlamak için yeniden Diffie-Hellman anahtar değişimini tetikleyebilmektedir. Bunun için talep mesajı atan uç nokta mesajın içinde anahtar değişimi yükünü de göndermelidir. Yeniden Diffie-Hellman anahtar değişimi yapılarak güvenlik oturumunun herhangi bir anında kripto algoritmalarının kullandığı anahtarları ele geçiren bir saldırganın yeni kurulan güvenlik oturumuna ait anahtarı hesaplayabilmesinin önüne geçilmiş olmaktadır. IKE SA'ya ait anahtarın yenilenmesi sırasında Diffie-Hellman anahtar değişiminin yeniden yapılması zorunludur [11].

2.3.2. INFORMATIONAL mesaj değişimi

IKE protokolünün çalışması sırasında IKE tünelinin uç noktalarındaki cihazlar birbirlerine INFORMATIONAL mesaj değişimi ile hata durumlarını veya bildirimleri iletebilmek amacıyla kontrol mesajları iletmek isteyebilmektedirler. INFORMATIONAL mesaj değişiminin şifreli olarak gerçekleştirilmesi gerektiği için ilk iki mesaj değişiminden sonra mesaj değişimi yapılabilmektedir. Hem IKE SA hem de Child SA ile ilgili kontrol mesajları IKE SA tünelinden gönderilmelidirler.

INFORMATIONAL mesaj değişimi ile hiç yük olmadan boş bir bildirim mesajı gönderilebileceği gibi; bildirim, SA silme ve konfigürasyon için de bildirim mesajı gönderilebilmektedir. Talep mesajını alan cevap veren uç nokta mesaj içeriği boş olsa bile mutlaka bu mesaja cevap göndermelidir. Eğer cevap veren uç nokta cevap göndermez ise istemci gönderdiği mesajın ağda kaybolduğunu varsayarak aynı paketi tekrar göndermektedir. Boş bildirim mesajı IKE tünelinin uç noktalarındaki cihazların birbirlerini sağlıklı bir şekilde hala çalışıp çalışamadıklarını kontrol edebilmeleri için kullanılmaktadır.

IKE SA veya IPsec SA bir uç birim tarafından kapatılmak istendiğinde bunu karşı uç birime de bildirmektedirler. Kapatma işlemini başlatan uç noktada da güvenlik

oturumu alış yönünde kapatılıp karşı uç birime güvenlik oturumunun kapatılacağı bilgisini içeren bir bildirim mesajı gönderilerek karşı uç birimin de güvenlik oturumunu sonlandırması sağlanmaktadır. Bir Güvenlik Oturumunu sonlandırmak için içinde bir veya daha fazla Delete yükü içeren bildirim mesajı gönderilmelidir. Delete mesajı ile kapatılması istenen güvenlik oturumuna ait SPI değerleri karşı uç birime gönderilmektedir. IKE SA kapatılmak istendiğinde IKE SA'ya ait Child SA'lar da kapatılmaktadır. IKE SA'yı kapatmak için gönderilen bir talep mesajına cevap veren uç nokta tarafından yüksüz, boş bir bildirim mesajı ile cevap gönderilmelidir.

2.4. IKEv2 Protokol Detayları

IKE protokolü UDP protokolünü kullanarak varsayılan port numarası 500'den haberleşmektedir. UDP protokolü bağlantı kurmadan ağ üzerinden paketleri gönderdiği için paketin hedefe ulaşıp ulaşmadığını kontrol edemez veya paketin ulaştığı noktadan geri bildirim alamaz. IKE protokolü kendi içinde paket kaybı durumlarına karşı bir tekrar gönderme mekanizması içermektedir. IKE protokolü son gönderdiği paketin hedefe ulaştığına dair teyit almadan yeni IKE paketi göndermemektedir.

IKE protokolünün büyük paketleri küçük parçalara ayıracak -Fragmante edecek- bir mekanizması bulunmamaktadır. Büyük boyutlu IKE paketlerinin Fragmente edilmesi IP protokolüne bırakılmaktadır. IKE paketlerinin boyutu ağda iletilebilecek maksimum paket boyutundan büyükse UDP paketleri IP protokolü tarafında fragmente edilerek ağda gönderilir. IKEv2 protokolü uygulanan sistemler 1280 bayt uzunluğundaki paketleri işleyebilmek zorunda olmakla berber sistemlerin 3000 bayt uzunluğundaki paketlere kadar destek vermesi beklenmektedir.

IKE protokolünde kullanılan tüm mesajlaşmalar talep ve cevap mesaj çiftlerinden oluşmaktadır. IKE SA kurulması için normal şartlarda ilk iki mesaj değişiminin yapılması yeterli olmaktadır. IKE SA kurulana kadar gerçekleştirilen ilk iki mesaj değişiminde talep mesajları istemci uç nokta tarafından gönderilmektedir. IKE SA kurulduktan sonra ise her iki uç nokta da talep mesajı gönderebilmektedir. Gönderilen mesajların IKE başlıklarına talep ya da cevap mesajı oldukları yazılmaktadır. IKE mesaj çiftlerinin iletilip iletilmediğinin sorumluluğu talep mesajını atan uç noktaya aittir. Cevap veren uç nokta hiçbir zaman yeniden iletme talebi almadan gönderdiği

bir mesajı tekrar göndermektedir. Talep mesajı gönderen uç nokta gönderdiği mesajı cevap alana kadar tutmaktadır ve cevap mesajı alana kadar tekrar göndermeye devam etmektedir. İstemci uç nokta ayarlanan miktarda paketi tekrar gönderdiği halde cevap alamaz ise IKE SA'yı sonlandırabilmektedir.

Her IKE mesajının IKE protokol başlığı bölümünde bir mesaj ID'si bulunur. Bu mesaj ID alanı talep ve cevap mesajlarını birbirleriyle eşleştirerek yeniden iletme durumunun anlaşılması amacıyla kullanılmaktadır. Yeniden iletilen IKE mesajı orijinal IKE mesajı ile aynı mesaj ID'ye sahip olmalıdır.

Mesaj ID alanı 32 bit uzunluğundadır. IKE_SA_INIT mesajında sıfır ayarlanarak gönderilir ve takip eden tüm mesaj değişimlerinde bir arttırılarak gönderilmektedir. Dolayısıyla IKE_SA_INIT mesaj değişiminden sonra gönderilen IKE_AUTH mesajının mesaj ID değeri 1 olmaktadır. IKE SA için anahtar değişimi yapıldığında mesaj ID tekrar sıfırlanarak mesajlar baştan numaralandırılmaya başlanır. Her iki uç noktada da beklenen Mesaj ID değeri ve gönderilecek mesajlara verilecek Mesaj ID değeri ayrı ayrı tutularak son gönderilen paketin ağda iletilirken kaybolup kaybolmadığı anlaşılmaktadır. Beklenen Mesaj ID'den önceki Mesaj ID ile bir talep mesajı gelirse cevap veren uç nokta daha önce gönderdiği paketin alıcıya ulaşmadığını anlayarak paketi tekrar göndermektedir.

IKE protokol başlığında ilk 16 bayt uzunluğundaki alanda 8 bayt uzunluğundan cevap veren uç noktaya ait SPI ve istemci uç noktaya ait SPI alanları bulunmaktadır. Bu alanlar paketlerin hangi IKE SA'ya ait olduklarının belirlenmesinde kullanılmaktadırlar. IKE tünelinin ilk kurulum aşamasında gerçekleştirilen IKE_SA_INIT mesaj değişimi sırasında istemci uç nokta henüz cevap veren uç noktanın hangi SPI'ı seçeceğini bilmediği için cevap veren uç noktanın SPI alanını sıfır ile doldurarak göndermektedir. Ağdan gelen IKE paketleri sadece SPI adreslerine bakılarak ilgili IKE SA'ya yönlendirilirler. IP adresine bakılarak Güvenlik Oturumu eşleştirmesi yapılmamaktadır.

ESP ve AH protokollerinde paket başlığında sadece alıcının SPI değeri yazılırken IKE paketlerinde hem alıcı hem de paketi gönderen uç noktaya ait SPI değeriler IKE protokol başlığında yazılı gönderilmektedir.

IKE mesajları içindeki SA yükü ile IKE, ESP veya AH protokollerinin güvenlik oturumlarında kullanılacak olan kripto algoritmaları önerileri uç noktalar arasında paylaşılmaktadır. Güvenlik oturumunda kullanılacak algoritmalar "Dönüşüm" adı verilen yapılar ile iletilmektedir. Dönüşümlerin tamamını tanımlayan öneri kümesi "Teklif" olarak isimlendirilmiştir. IKE paketlerindeki SA yüklerinde ile birden fazla teklif bulunulabilir. Her teklif içerisinde hangi güvenlik oturumunu tanımladığına bağlı olarak değişen (IKE, ESP veya AH) dönüşümler ile güvenlik oturumu için gerekli olan tüm kripto algoritmaları uç noktalar tarafından önerilmektedir. Dönüşümlerin de "Nitelik" adı verilen algoritma ile alakalı detay bilgilerin bulunduğu alanları olabilmektedir. Nitelik alanı sadece dönüşüm ismi algoritmayı tanımlamakta yetersiz kalıyorsa -Anahtar uzunluğunu belirtmek için vs.- kullanılmaktadır.

Her dönüşümde sadece bir adet protokol bulunmaktadır. Eğer cevaplayan uç nokta SA yükü içindeki bir teklifi kabul ettiyse içeriğindeki diğer dönüşümleri de değiştirmeden kabul etmesi gerekmektedir. Cevap veren uç nokta gönderilen teklifler arasından sadece bir tanesini seçmelidir. Eğer tekliflerden hiç birisi cevap veren uç nokta tarafından kabul edilmezse NO_PROPOSAL_CHOSEN tipinde bir bilgilendirme mesajıyla istemciyi bilgilendirmektedir.

Her IPsec teklifi bir veya daha fazla dönüşüm içerebilmektedir. Güvenlik oturumu için kullanılacak algoritmalar tek bir teklif içerisinden seçilmelidir. Örnek olarak IPsec SA için gönderilen teklif ENCR_3DES, ENCR_AES (nitelik: anahtar uzunluğu = 128), ENCR_AES (Nitelik: anahtar uzunluğu = 256), AUTH_HMAC_MD5 ve AUTH_HMAC_SHA dönüşümlerini içeriyor olsun. Cevap veren uç nokta dönüşümler arasından bir adet şifreleme(ENCR_) ve bir adet doğrulama(AUTH_) algoritması seçmelidir. Dolayısıyla gönderilen teklifler arasından 6 farklı kombinasyon ile kriptografik algoritmalar seçilerek IPsec SA oluşturulabilmektedir.

2.4.1. IPsec trafiğinin belirlenmesi

IPsec çalışmakta olan bir sistemde IPsec yazılım bloğuna gelen bir IP paketi sistemin politikalarına bakılarak korumalı olarak belirlenirse IPsec yazılım bloğu IPsec SA'da belirlenen güvenlik protokollerini IP paketine uygulamaktadır. Eğer henüz IPsec SA yoksa SA kurulum görevi IKE protokolüne bırakılmaktadır. Sistemin politika veri

bankası IPsec SA kurulduktan sonra IKE protokolü tarafından güncellenerek IPsec protokolünün doğru paketlerde güvenlik protokollerini çalıştırması sağlanmaktadır.

TS yükü ile uç noktalar birbirlerine kendilerine ait IP trafiği politikalarını tanıtmaktadır. TS yükü ile uç noktalar yeni kurulacak olan IPsec SA'dan geçecek olan IP trafiğinin IP adres aralığını belirlemektedir. TS yükü Child SA kurulurken yapılan mesaj değişiminde mesaj çiftinin ikisinde de bulunmaktadır. TS yükü ile birden fazla trafik seçici içinde IP adresi aralıkları, port aralıkları ve IP protokol ID'leri bulunmaktadır.

TS yükü Child SA kurulurken yapılan mesaj değişiminde TSi ve TSr olmak üzere iki adet bulunmaktadır. İstemci uç nokta TSi yükü ile hangi IP adres aralığındaki IP paketlerini cevap veren uç noktaya göndereceği belirtmektedir. Cevap veren uç nokta TSr yükü ile hangi IP adres aralığındaki IP paketlerini istemciye göndereceği belirtmektedir. Cevap veren uç nokta bu talebi kabul eder ise aynı TSi yükünü cevap mesajında geri göndermektedir.

IKEv2 protokolü cevap veren uç noktaya istemci uç noktanın sunduğu IP aralığını küçülterek daha küçük bir alt ağ erişimi sağlayabilme imkânı sağlamaktadır. Bu durum genellikle uç noktalar arasındaki konfigürasyonun değiştiği ancak bu değişikliğin sadece bir uç noktada güncellendiği durumda meydana gelmektedir [11]. Uç noktalar farklı kişiler tarafından ayarlanabileceği için uç noktalar arasındaki IP adresi aralığı uyumsuzluğu bir hataya yol açmadan SA'ların çalışmasına olanak sağlayabilmektedir. Uç noktalar TSi veya TSr yüklerinin içerisine birden fazla TS yükü ekleyerek belli port numaralarından gelen belli protokolleri daha dar bir IP adresi aralığına yönlendirebilirler. Bu IP aralığı daraltma işlemine IKEv2 protokolünde "narrowing" adı verilmektedir [11].

2.4.2. Nonce

IKE_SA_INIT ve CREATE_CHILD_SA mesaj değişimi sırasında karşılıklı gönderilen IKE mesajlarının içinde Nonce alanları bulunmaktadır. Nonce değerleri kripto algoritmalarına giriş olarak verilerek tekrar saldırılarına karşı koruma sağlamaktadır [11]. Nonce değeri bir rastgele sayı üreteci kullanılarak en az 128 bit uzunluğunda ve IKE SA'da kullanılan Sözde-Rastgele Fonksiyon'un (PRF) anahtar

uzunluğunun yarısı uzunluğunda üretilmelidir. Karşılıklı olarak iletilen Nonce değerleri Child SA için kullanılacak anahtarların türetilmesi sırasında rastgeleliği arttırmak ve Diffie-Hellman anahtar değişiminde rastgele bir giriş verisi olarak kullanılmaktadır. İstemci uç nokta IKE SA kurulmadan önce nonce değerini üreteceği için, SA yükü ile önerdiği PRF fonksiyonları için yeterli uzunlukta bir nonce göndermelidir.

2.4.3. Güvenlik oturumları için anahtarların üretilmesi

IKEv2 protokolüyle IKE SA kurulumu için IKE_SA_INIT mesaj değişimi sırasında uç noktalar arasında kullanılacak olan şifreleme algoritması, bütünlük koruma algoritması, Diffie-Hellman grubu ve PRF üzerinde anlaşma sağlanmaktadır. Sözde rastgele fonksiyonu (PRF), IKE SA ve IPsec SA için anahtar üretiminde kullanılan kriptografik bir algoritmadır.

Şifreleme ve bütünlük koruma için değişken uzunlukta anahtarlar ile çalışabilen kriptografik algoritmalar kullanılacak ise bu algoritmaların karşılıklı tanıtılmaları sırasında sabit bir anahtar uzunluğu SA yükü ile iletilerek belirlenmektedir. Bütünlük kontrolü amacıyla kullanılan HMAC algoritmaları için gerekli anahtar uzunluğu bu algoritmalarda kullanılan özet algoritmalarının çıkış uzunluğu kadardır.

PRF algoritmaları herhangi uzunluktaki bir anahtar ile çalışabilirler ancak IKE protokolü ile SA kurulurken ortak bir anahtar uzunluğu üzerinde anlaşılmalıdır. HMAC algoritmalarını kullanan PRF algoritmalarının anahtar uzunlukları HMAC algoritmasının kullandığı özet algoritmasının çıkış uzunluğuna eşit olmaktadır. HMAC haricinde bir algoritma ile çalışan PRF algoritmaları için gerekli anahtar uzunluğu SA yükü ile belirtilmektedir.

IKEv2 protokolünde kriptografik algoritmalar için gerekli anahtarlar IKE SA'da anlaşılan PRF algoritmasının çıktıları kullanılarak üretilmektedir. Gerekli olan toplam anahtar uzunluğu PRF algoritmasının çıktı uzunluğundan daha fazla olduğu için PRF algoritması tekrarlı olarak kullanılarak gerekli anahtar uzunluğuna ulaşılmaktadır.

Anahtar üretiminin anlatımında PRF algoritmasının giriş verisi, çıkış verisi ve anahtar değerinin gösterimi Denklem (2.1)'de;

Çıkış Verisi=prf(Anahtar, Giriş Verisi) (2.1)

verilmiştir.

IKEv2 protokolü anahtar üretimi için PRF algoritmasını tekrarlı olarak kullandığı için prf+ ile PRF algoritmasının tekrarlı kullanımı gösterilmiştir. Anahtar üretimi için öncelikle ihtiyaç duyulan toplam anahtar uzunluğu hesaplanarak PRF algoritmasının çıkış uzunluğuna bölünmüştür. Elde edilen bölüm tam sayı değil ise yukarı yuvarlanarak ihtiyaç olan anahtar uzunluğu karşılanmaktadır. PRF algoritmasının çıktısı art arda eklenerek IKE protokolünün detaylarında anlatılacak olan sıra ile anahtarlar elde edilmektedir.

Aşağıdaki gösterilen örnekte dört tekrar ile ihtiyaç duyulan anahtar uzunluğuna erişildiği varsayılmıştır. IKEv2 protokol tanımına göre ilk tekrardan itibaren giriş verisinin sonuna iterasyon numarası eklenerek PRF algoritması çalıştırılır. n. iterasyon çıktısı İÇn olarak gösterilmiştir. | sembolü ile PRF çıktılarının birbirine bağlandığı gösterilmiştir. Buna göre ilk iterasyon Denklem (2.2)'deki gibi;

İÇ1=prf(Anahtar, Giriş Verisi |0x01) (2.2)

hesaplanır[11].

İlk tekrardan sonraki tekrarlarda bir önceki PRF'den üretilen çıktı ile giriş verisi birleştirilerek sonuna ilk tekrarda olduğu gibi tekrar numarası birleştirilen verinin sonuna yazılır. Buna göre kalan tekrarlar sırasıyla Denklem (2.3), (2.4) ve (2.5)'deki gibi;

İÇ2=prf(Anahtar, İÇ1|Giriş Verisi|0x02) (2.3)

İÇ3=prf(Anahtar, İÇ2|Giriş Verisi|0x03) (2.4)

İÇ4=prf(Anahtar, İÇ3|Giriş Verisi|0x04) (2.5)

hesaplanır[11].

Buna göre n adet tekrar ile gerçekleştirilen prf+ Denklem (2.6)'daki gibi;

prf+(Anahtar, Giriş Verisi)=İÇ1|İÇ2|..|İÇn (2.6)

gösterilebilir.

prf+ Fonksiyonunun çıktısı birleştirme sırasında oluşan sınırlar dikkate alınmadan ihtiyaç duyulan uzunlukta anahtar olarak kullanılmaktadır.

2.4.4. IKE güvenlik oturumu için şifreleme anahtarlarının üretilmesi

Ortak anahtarların üretilmesi sırasında kullanılan SKEYSEED değeri IKE_SA_INIT mesaj değişimi sırasında uç noktalar arasında paylaşılan nonce değerlerinden ve hesaplanan Diffie-Hellman değerinden PRF fonksiyonu kullanılarak üretilmektedir. Uç noktalar tarafından karşılıklı paylaşılan nonce değerleri önce istemci uç noktaya ait nonce sonra cevap veren uç noktaya ait nonce olmak üzere art arda birleştirilmektedir. Nonce değeri ve hesaplanan g^{ir} değeri PRF fonksiyonuna sokularak SKEYSEED değeri üretilmektedir. PRF fonksiyonuna birleştirilen nonce değerleri anahtar olarak, g^{ir} değeri giriş verisi olarak verilerek Denklem (2.7)'deki gibi;

$$SKEYSEED=prf(Ni|Nr, g^{ir}) \quad (2.7)$$

hesaplanır[11].

prf+ fonksiyonu ile nihai anahtarların üretimi için uç noktalar arasında paylaşılan nonce ve SPI değerleri art arda birleştirilerek veri olarak kullanılmaktadır. Hesaplanan SKEYSEED değeri ise anahtar olarak prf+ fonksiyonuna verilerek IKE SA için anahtar üretimi tamamlanmaktadır. Üretilen prf+ sonucundan Denklem (2.8)'de gösterildiği gibi;

$$prf+(SKEYSEED, Ni|Nr|SPIi|SPIr)= (SK_d|SK_{ai}|SK_{ar}|SK_{ei}|SK_{er}|SK_{pi}|SK_{pr}) \quad (2.8)$$

7 farklı anahtar elde edilmektedir[11].

SK_d: Child SA için üretilecek anahtarlar bu anahtar kullanılarak türetilir.

SK_{ai}: IKE_SA_INIT mesaj değişiminden sonra IKE tünelinden yapılacak mesaj değişimlerinde istemci uç nokta gönderdiği mesajlarda kimlik doğrulama ve veri bütünlüğünü bu anahtarı kullanarak sağlanmaktadır.

SK_{ar}: IKE_SA_INIT mesaj değişiminden sonra IKE tünelinden yapılacak mesaj değişimlerinde cevap veren uç nokta gönderdiği mesajlarda kimlik doğrulama ve veri bütünlüğünü bu anahtarı kullanarak sağlamaktadır.

SK_{ei}: IKE_SA_INIT mesaj değişiminden sonra IKE tünelinden yapılacak mesaj değişimlerinde istemci uç nokta gönderdiği mesajları şifreleyerek veri gizliliğini sağlayabilmek için bu anahtarı kullanmaktadır.

SK_{er}: IKE_SA_INIT mesaj değişiminden sonra IKE tünelinden yapılacak mesaj değişimlerinde cevap veren uç nokta gönderdiği mesajları şifreleyerek veri gizliliğini sağlayabilmek için bu anahtarı kullanmaktadır.

SK_{pi}: IKE_AUTH mesaj değişimi sırasında istemciye ait kimlik doğrulama verisinin paylaşıldığı kimlik doğrulama yükü bu anahtar kullanılarak üretilir ve cevap veren uç nokta tarafından doğrulanır.

SK_{pr}: IKE_AUTH mesaj değişimi sırasında cevap veren uç noktaya ait kimlik doğrulama verisinin paylaşıldığı kimlik doğrulama yükü bu anahtar kullanılarak üretilir ve istemci tarafından doğrulanır.

SK_d, SK_{pi}ve SK_{pr} anahtarlarının uzunlukları IKE_SA_INIT anahtar değişimi sırasında üzerinde anlaşılan PRF algoritmasının kullanması gereken uzunlukta olmaktadır.

g^{ir} değeri karşı uç noktadan gelen anahtar değişimi yükü, uç noktada üretilen gizli anahtar ve Diffie-Hellman grubu kullanılarak Denklem (2.9)'da gösterildiği gibi;

$$g^{ir} = \text{Anahtar Değişimi Değeri}^{\text{Gizli Anahtar}} \bmod(\text{DH Asal Sayısı}) \quad (2.9)$$

hesaplanır[11].

Diffie-Hellman ile g^{ir} ve anahtar değişimi yükünün nasıl hesaplandığı Diffie-Hellman başlığında detaylı olarak anlatılacaktır.

2.4.5. IKE güvenlik oturumunun doğrulanması

IKE SA kuran uç noktalar IKE_SA_INIT ve IKE_AUTH mesaj değişimi sırasında paylaşılan verilerin belli kısımlarını üzerinde anlaşılan bir HMAC algoritması ile

imzalayarak karşılıklı olarak birlerinin kimliklerini doğrulamaktadırlar. Kimlik doğrulamak için öncelikle uç noktalar kendi oturum doğrulama yüklerini hesaplayarak IKE_AUTH mesaj değişimi ile paylaşmaktadırlar. Daha sonra karşı uç noktadan gelen kimlik doğrulama yükünü tekrar hesaplanıp doğruluğunun kontrolü yapılarak kimlik doğrulaması sağlanmış olur. Kimlik doğrulaması başarısız olursa hata mesajı gönderilerek IKE SA sonlandırılmaktadır.

Uç noktalar oturum doğrulama yükü oluştururken önceden uç noktalar arasında paylaşılmış bir PSK veya sertifika kullanabilirler. Bu tez kapsamında PSK kullanılarak kimlik doğrulaması sağlandığı için PSK ile kimlik doğrulama yöntemi anlatılacaktır. Herhangi bir rastgelelik içermeden sadece kullanıcı tarafından seçilen bir paroladan üretilen PSK değerinin kullanılması yeterli tahmin edilemezliğe sahip olamayabileceğinden sözlük saldırılarına karşı dirençsizdirler. Bu saldırılara karşı PSK ile yapılan kimlik doğrulama yaygın ancak genellikle güvensiz bir yöntemdir [11]. Bu nedenle seçilen PSK tahmin edilebilirlikten uzak olmalıdır.

Uç noktaların mesaj değişimleri sırasında gönderdikleri verilerin imzalanacak kısımlarının birleştirilmesiyle oluşan veri bütününe SignedOctets adı verilmektedir. İstemci uç noktanın hesapladığı InitiatorSignedOctets değeri sırasıyla Denklem (2.10) ve (2.11)'de gösterildiği;

$$\text{MACedIDForI} = \text{prf}(SK_{pi}, \text{RestOfInitIDPayload}) \quad (2.10)$$

$$\text{InitiatorSignedOctets} = \text{RealMessageI}|\text{NonceRData}|\text{MACedIDForI} \quad (2.11)$$

gibi hesaplanmaktadır[11].

RestOfInitIDPayload: İstemci uç noktanın gönderdiği ID yükünün yük başlığı altında kalan alanın tamamını ifade etmektedir (ID Type, Reserved ve InitIDData alanları).

RealMessageI: İstemci uç noktanın gönderdiği IKE_SA_INIT talep mesajının IP başlığının altında kalan kısmının tamamını ifade etmektedir. IKE protokol başlığı dahil (IKE protokol bağlındaki ilk SPI değerinden başlayarak) tüm gönderilen IKE mesajı bu alana dahildir.

NonceRData: Cevap veren uç nokta tarafından gönderilen Nonce yükünün yük başlığının altında yer alan Nonce verisi alanının tamamını ifade etmektedir.

MACedIDForI: IKE SA için üretilen SK_{pi} anahtarı ile RestOfInitIDPayload verisi üzerinde anlaşılan PRF fonksiyonuna sokularak üretilir.

Responder uç noktanın hesapladığı ResponderSignedOctets değeri sırasıyla Denklem (2.12) ve (2.13)'de;

$$MACedIDForR = prf(SK_{pr}, RestOfRespIDPayload) \quad (2.12)$$

$$ResponderSignedOctets = RealMessageR|NonceIData|MACedIDForR \quad (2.13)$$

gösterildiği gibi hesaplanmaktadır[11].

Signed Octest hesaplandıktan sonra AUTH yükünün hesaplanabilmesi için PSK ve "Key Pad for IKEv2" dizisi kullanılarak HashedPSK değeri üretilmektedir. "Key Pad for IKEv2" dizisi 17 adet ASCII karakterden oluşur ve NULL sonlandırıcı karakteri içermemektedir. Hesaplanan HashedPSK ve SignedOctets kullanılarak IKE_AUTH mesaj değişiminde AUTH yükü ile iletilecek olan nihayi kimlik doğrulama verisi üretilmiş olur. AUTH yükü sırasıyla Denklem (2.14) ve (2.15)'te gösterildiği gibi;

$$HashedPSK = prf(PSK, "Key Pad for IKEv2") \quad (2.14)$$

$$AUTH = prf(HashedPsk, SignedOctest) \quad (2.15)$$

hesaplanmaktadır[11].

PSK, "Key Pad for IKEv2" dizisiyle PRF işleminden geçirilip saklanarak PSK'nın düz metin olarak saklanmasının önüne geçilmiş olunmaktadır. PSK eğer bir paroladan üretildiyse IKEv2 protokolü haricinde bir uygulamanın PSK'yı parola olarak kullanması engellenmesi amaçlanmaktadır. PSK'yı bir paroladan türetmek güvenli olmamasına rağmen uç noktaları ayarlayan insanların PSK'yı yine de bir paroladan türetebileceği tahmin edildiği için bu şekilde bir yapı kullanılmaktadır [11].

2.4.6. IPsec güvenlik oturumu için şifreleme anahtarlarının üretilmesi

IKE SA için anahtar üretimi sonlandıktan sonra bu anahtarlar kullanılarak kurulan IKE tünelinden IKE_AUTH mesaj değişimi gerçekleştirilir. Eğer IKE_AUTH mesaj değişimi başarılı bir şekilde gerçekleşirse ilk Child SA kurulumu tamamlanmış olur. Child SA için anahtar üretilirken daha önce IKE SA için üretilmiş olan SK_d anahtarı ve karşılık olarak IKE_SA_INIT mesaj değişiminde veya CREATE_CHILD_SA mesaj değişiminde paylaşılan Nonce değerleri kullanılır. Bu değerlerden prf+ fonksiyonu kullanılarak Child SA'nın kullanacağı 4 farklı anahtar Denklem (2.16)'te gösterildiği gibi;

$$prf+(SK_d, Ni|Nr) = CA\_SK_{ie}|CA\_SK_{ia}|CA\_SK_{re}|CA\_SK_{ra} \quad (2.16)$$

üretilmektedir[11].

CA_SK_{ie}: Child SA tarafından kurulacak olan IPsec tünelinde istemci uç noktanın IP paketi gönderirken trafiği şifrelemek için kullanacağı anahtardır. Cevap veren uç nokta IPsec tünelinden gelen trafiği çözmek için bu anahtarı kullanır.

CA_SK_{ia}: Child SA tarafından kurulacak olan IPsec tünelinde istemci uç noktanın IP paketi gönderirken kimliğinin doğrulanabilmesi için kullanacağı anahtardır. Cevap veren uç nokta IPsec tünelinden gelen trafiğin kimliğini doğrulamak için bu anahtarı kullanır.

CA_SK_{re} : Child SA tarafından kurulacak olan IPsec tünelinde Cevap veren uç noktanın IP paketi gönderirken trafiği şifrelemek için kullanacağı anahtardır. İstemci uç nokta IPsec tünelinden gelen trafiği çözmek için bu anahtarı kullanır.

CA_SK_{ra}: Child SA tarafından kurulacak olan IPsec tünelinde Cevap veren uç noktanın IP paketi gönderirken kimliğinin doğrulanabilmesi için kullanacağı anahtardır. İstemci uç nokta IPsec tünelinden gelen trafiğin kimliğini doğrulamak için bu anahtarı kullanır.

Eğer CREATE_CHILD_SA mesaj değişimi sırasında Diffie-Hellman anahtar değişimi gerçekleştirilirse yeni hesaplanan g^{ir} değeri ile Nonce değerleri birleştirilerek Denklem (2.17)'daki gibi;

$$prf+\left(SK_d, g^{ir}|Ni|Nr\right) = CA\_SK_{ie}|CA\_SK_{ia}|CA\_SK_{re}|CA\_SK_{ra} \quad (2.17)$$

anahtarlar hesaplanır[11].

3. DİFFİE-HELLMAN ANAHTAR DEĞİŞİMİ

Diffie-Hellman anahtar değişimi yöntemi Whitfield Diffie ve Martin Hellman tarafından 1976 yılında "New Directions in Cryptography" isimli bir makale ile duyurulmuştur. Duyurulan bu yöntem 1977 yılında US4200770A numarası ile patent altına alınmış ancak 1997 yılı itibariyle bu patenti süresi dolmuştur [12].

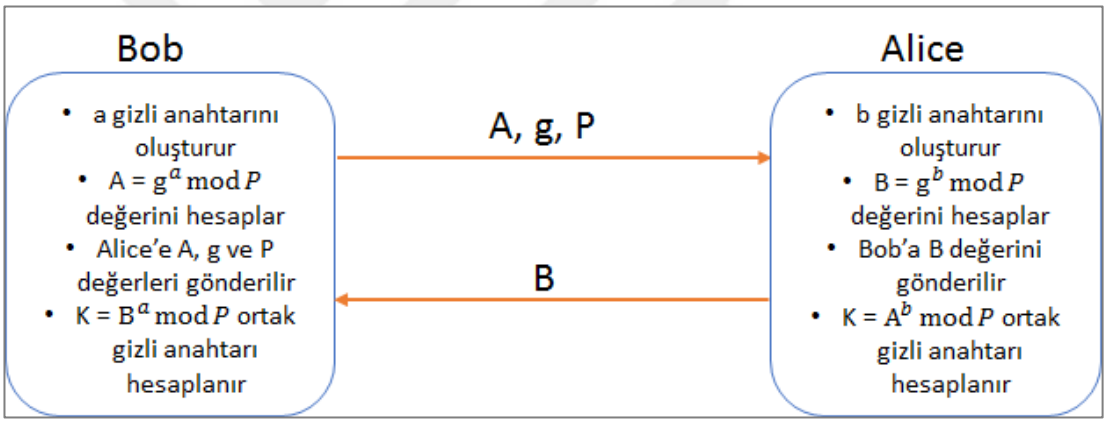
Diffie-Hellman yöntemi ile uç noktalar güvensiz bir ağ üzerinden ortak gizli bir anahtar elde edebilirler. Elde edilen bu ortak gizli anahtar uç noktaların güvensiz ağ üzerinden şifreli olarak haberleşerek güvenli bir hat oluşturmalarını sağlar. Ağ üzerinden güvenli haberleşmenin bir zorunluluk haline geldiği günümüzde Diffie-Hellman ağ altyapısının da en çok kullanılan protokollerden bir tanesidir. Web üzerinden kurulan bağlantıların güvenliğini sağlamak amaçlı kullanılan SSL veya TLS protokollerinde de Diffie-Hellman anahtar değişimi kullanıcıların büyük çoğunluğu farkında olmasalar bile kullanılmaktadır. Diffie-Hellman anahtar değişimi IPsec protokolünün de bir parçası olduğu için IPsec tabanlı tüm VPN servislerinde Diffie-Hellman kullanılmaktadır [13].

Diffie-Hellman anahtar değişimi protokolü IKEv2 protokolünün temelini oluşturur. IKEv2 protokolü IKE SA ve IPsec SA için gerekli olan anahtarları üretmek için kullandığı “Ortak Gizli Anahtar” değerini Diffie-Hellman anahtar değişimi protokolünü kullanarak hesaplar. Sonrasında IKE ve IPsec tünelleri kurularak bu tüneller üzerinden güvenli haberleşme sağlanır.

DH anahtar değişiminin güvensiz bir ağ üzerinden gerçekleştirildiği varsayıldığı için anahtar değişimi sırasında uç noktalar arasında paylaşılan sayılar üçüncü kişiler tarafından elde edilebilir. Ancak paylaşılan bu sayıları üçüncü bir kişi elde edebilse bile DH algoritması ayrık logaritma problemine dayandığı için uç noktaların ortak oluşturdukları “Ortak Gizli Anahtar” değerini geri hesaplayamaz [14].

DH anahtar değişimi sırasında yapılan aritmetik işlemler kavramsal olarak basittir. Temel matematiksel işlemlerden üs alma ve modüler aritmetik ile DH anahtar değişimi gerçekleştirilir. Uç noktalar kendilerine ait gizli anahtarları hiçbir zaman birbirleri ile paylaşmazlar. Bu gizli anahtarları kullanarak hesapladıkları sayıyı ve bu sayıyı hesaplarken kullandıkları sabit değerleri ağ üzerinden paylaşırlar.

DH anahtar değişimi sırasında uç noktalar tarafından gerçekleştirilen işlemler anlatım kolaylığı için uç noktalara isim verilerek açıklanmıştır. DH anahtar değişimini gerçekleştirecek uç noktalardan birisinin adı Alice diğerinin adı Bob olsun. Güvensiz ağ üzerinden yapılan anahtar değişimini dinleyen üçüncü kişi Eve olsun. Anahtar değişimini başlatan uç noktanın Alice olduğunu varsayarsak Alice ve Bob'un gerçekleştirdiği aritmetik işlemler ve karşılıklı iletilen veriler Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Diffie-Hellman Anahtar Değişimi Adımları

- DH anahtar değişimi için Alice tarafından a gizli anahtarı oluşturulur.
- Alice, bir g ve P sayısı belirler. Bu sayıları kullanarak A açık anahtarını hesaplayıp Bob'a g ve P sayıları ile birlikte gönderir.
- Bob g, P sayılarını ve A açık anahtarını aldıktan sonra kendi B açık anahtarını hesaplayıp Alice'e gönderir. K ortak gizli anahtarını A açık anahtarını, g ve P sayılarını kullanıp hesaplayarak DH anahtar değişimini tamamlar.
- Alice, Bob'un B açık anahtarını aldıktan sonra K ortak gizli anahtarını B açık anahtarı, g ve P sayılarını kullanıp hesaplayarak DH anahtar değişimini tamamlar.
- Bu noktada Alice ve Bob arasındaki haberleşmeyi dinleyebilen Eve A ve B açık anahtarları ile g ve P sayılarını ele geçirmiş olur.

DH anahtar değişimi sırasında açıktan paylaşılan A ve B açık anahtarlarını ile g ve P sayılarını ele geçiren Eve, Alice'e ait gizli anahtarı Denklem (3.1)'den;

$$A= g^{a} \bmod P \tag{3.1}$$

Bob'a ait gizli anahtarı Denklem (3.2)'den;

$$B= g^{b} \bmod P \tag{3.2}$$

geri hesaplayabilir[15]. Eve'nin yapacağı bu geri hesaplama problemi ayrık logaritma problemi olarak bilinmektedir. Ayrık logaritma problemi, P sayısı bir asal sayı olarak seçildiğinde ve uzunluğu 1024 bit ve üzerinde seçildiğinde çözülmesi makul bir sürede ve maliyette gerçekleştirilememektedir [15]. Diffie-Hellman üreteci olarak isimlendirilen g sayısının büyük bir sayı olmasına gerek yoktur.

David Adrian ve arkadaşları[15] tarafından Ekim 2015'te yapılan çalışmalarda 512 bit uzunluğundaki asal sayı kullanıldığında 18 çekirdekli bir İntel Xeon işlemci kullanılarak bir dakika civarında bir sürede ayrık logaritma probleminin çözülebildiğini göstermiştir. Aynı çalışmada 1024 bit uzunluğunda bir asal sayı seçilmesi durumunda problemin çözümünün maliyetinin yaklaşık 100 milyon dolar olacağı öngörülmüştür. 2048 bit uzunluğundaki asal sayılar kullanıldığı durumda problem çözümünün 1024 bit kullanılmasına oranla 10^9 kat daha zor olacağı belirtilerek günümüzdeki sistemlerde 1024 bit ve altındaki uzunluklarda DH gruplarının kullanılmaması önerilmiştir [15]. Bu tez çalışmasının yazım tarihinde tanımlı olan DH gruplarında Diffie-Hellman üreteci 2 olarak belirlenmiştir [16, 17].

DH anahtar değişimi örneğinde gösterilen g ve P sayıları IKEv2 protokolü tarafından IKE SA kurulurken uç noktalar tarafından belirlenen Diffie-Hellman grup numaralarından gelmektedir. DH grup numaraları IANA tarafından belirlenmiştir. [11] ve [17]'de tanımlanan DH gruplarında kullanılan asal sayı (P) uzunluğu ve üreteç değeri Tablo 3.1'de gösterilmiştir.

Tablo 3.1. DH Grupları

DH Grup Numarası	Asal Sayı Uzunluğu	Üreteç
1	768 Bit	2
2	1024 Bit	2
5	1536 Bit	2
14	2048 Bit	2
15	3072 Bit	2
16	4096 Bit	2
17	6144 Bit	2
18	8198 Bit	2

4. NESNELERİN İNTERNETİ İÇİN GÜVENLİ AĞ GEÇİDİ TASARIMI

Bu bölümde nesnelerin İnterneti uygulamalarında kullanılan ağ geçitlerinin tanım ve görevlerinden bahsedilerek nesnelerin İnterneti uygulamalarında veri gizliliği ve güvenliği sağlamak amacıyla yapılan çalışmalara değinilip tez kapsamında gerçekleştirilen güvenli ağ geçidi tasarımı anlatılmıştır.

4.1. Giriş

Nesnelerin İnterneti uygulamalarında kullanılan nesnelerin sayılarına ve haberleşmek için kullandıkları protokollerin çeşitliliğine bağlı olarak diğer sistemlerle ve İnternet ile daha iyi bütünleşebilen uygulamalar gerçeklenmek istendiğinde nesnelerin İnterneti için ağ geçitlerine ihtiyaç duyulmaktadır. Nesnelerin İnterneti ağ geçitleri birçok protokol ve çözümü farklı ölçeklerde gerçekleştirerek nesneleri başka bir ağa, buluta ya da veri merkezleri gibi birçok noktaya köprüleyebildikleri için birçok farklı formda karşımıza çıkabilmektedirler.

Nesnelerin İnterneti yapısında ağ geçitleri birçok görevi yerine getirebilirler:

- Nesnelerin İnterneti ağlarını birbirleriyle haberleştirirler, nesnelerin İnterneti ağından gelen verilerin filtrelenip toplarlar, nesnelerin İnternet erişimlerini sağlayarak kullanıcıların nesnelere erişimini ve yönetimini sağlayabilir.
- Nesnelerin kısıtlı donanımları ile gerçekleştiremeyecekleri ağır işlem yükü getiren uygulamaları gerçekleştirir, protokoller arası dönüşümleri gerçekleştirir, karar verme mekanizması olarak çalışabilir.
- Nesnelerin güvenlik ihtiyaçlarını karşılayabilir.

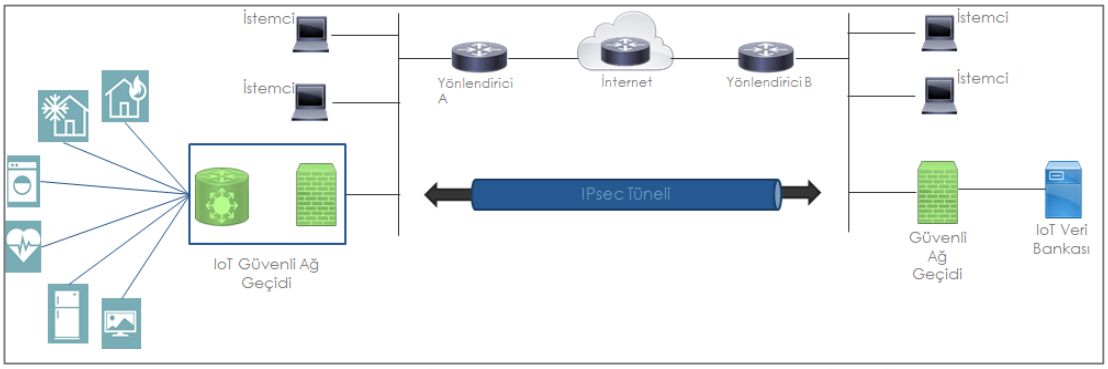
[18]'de IPsec AH, IPsec ESP ve DTLS protokollerinin bir nesnelerin İnterneti uygulamasında gerçekleştirerek performans karşılaştırmaları yapılmıştır. Geliştirilen yazılımlar ST Microelectronics tarafından geliştirilen 24 MHz saat frekansında çalışan 32-bit ARM Cortex M3 tabanlı bir MCU kullanılan MB851 geliştirme kartında gerçeklenmiştir. Kullanılan MCU'nun düşük güç tüketerek yüksek performans ile çalıştığı vurgulanmıştır. Yapılan çalışmada IPsec AH ve IPsec ESP için gerekli olan güvenlik oturumu IKEv2 protokolü olmadan şifreleme ve kaynak doğrulama için gerekli anahtarlar el ile sistemlere girilerek test edilmiştir. IKEv2 gibi bir otomatik güvenlik oturumu protokolünün sistemde kullanılan MCU için ağır kriptografik işlemler gerektirdiği için kullanılmadığı belirtilerek IKEv2 protokolü kullanılmamıştır.

[19]'da sensörler ve istemci arasında uçtan uca haberleşme güvenliğinin sağlanması için DTLS kullanılması önerilmiştir. Çalışmada Atmel tarafından üretilen 48 MHz saat frekansında çalışan 32-bit ARM Cortex M3 tabanlı bir MCU kullanılmıştır. Yapılan çalışmada IPsec protokolü ağ katmanında çalıştığı için IPsec tüneli kurarak güvenli haberleşecek iki uç noktanın da ağ alt yapısının aynı güvenlik protokollerini desteklemek zorunda olduğu, DTLS protokolünün ise iletim ve uygulama katmanı arasında yer alarak uç noktalar arasında ağ katmanı güvenlik protokollerinin uyumlu olması gibi bir gerekliliği bulunmadığı vurgulanmıştır. Çalışmada farklı kripto algoritmaları kullanıldığında sistemin güç tüketimi, ağ gecikmeleri gibi performans değerlendirmeleri yapılmıştır.

Bu tez çalışmasında nesnelerin İnterneti uygulamalarında nesnelerin başka ağlar ile İnternet üzerinden yaptıkları haberleşmenin ağ seviyesinde güvenliğini sağlayabilecek bir güvenli ağ geçidi tasarımı yapılmıştır. Yapılan çalışmada donanımsal olarak kriptografik şifreleme ve doğrulama fonksiyonlarını gerçekleştirebilen bir MCU seçilerek güvenli ağ geçidinin ağır işlem yükü getiren kriptografik fonksiyonların iş yükünden kurtarılması hedeflenmiştir. Seçilecek MCU'da bulunan donanımsal kriptografik hızlandırıcılar yardımıyla sistemde otomatik güvenlik oturumu yönetimi protokolü gerçeklenmiştir.

Tasarlanan sistem nesnelerin interneti ağından gelen verilerin ağ seviyesinde güveliğini sağlayarak internet üzerinden verilerin güvenli olarak iletilmesini garanti

altına almaktadır. Tasarlanan sistem nesnelerin interneti ağı ile internet arasında çalışmaktadır. Tasarlanan sistemin nesnelerin İnterneti uygulamalarındaki yeri Şekil 4.1'de gösterilmiştir.



Şekil 4.1. Tasarlanan Sistemin Nesnelerin İnterneti Uygulamalarındaki Yeri

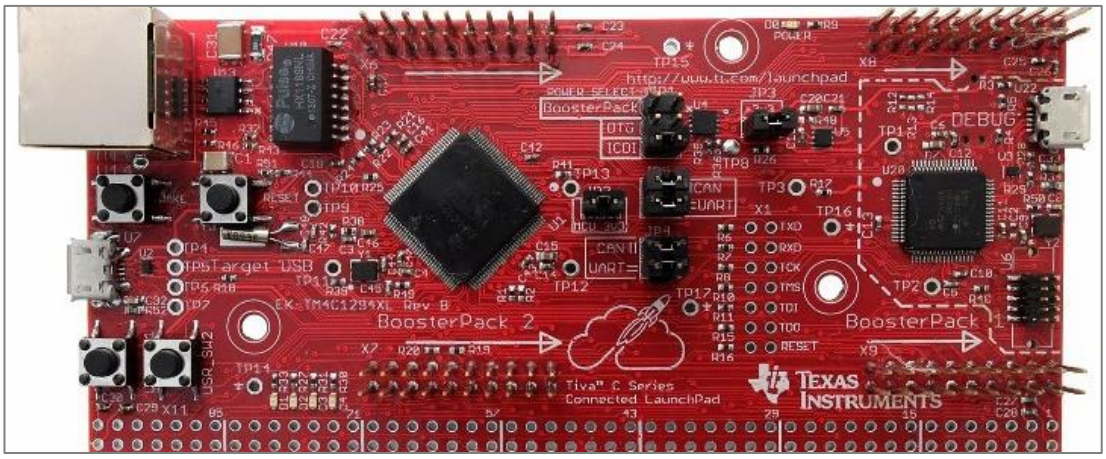
Nesnelerin İnterneti ağında çalışarak dış dünyadan ölçümler yapan sensörlere ait verileri toplayan, aktüatörleri yöneten ve durumları hakkında bilgi toplayan, düşük kaynaklara sahip, az yer kaplayan, düşük güç tüketimli, İnternet üzerinden gerçekleştirilecek haberleşmede ağ katmanında veri mahremiyeti ve gizliliği sağlayabilen bir ağ geçidi tasarımı yapılmıştır. Tasarlanan ağ geçidi nesneleri İnternet üzerinden haberleştirirken ağ katmanında veri gizliliği ve bütünlüğünü koruyarak bir güvenli ağ geçidi olarak çalışabilmektedir.

4.2. Sistem Tasarımı

Tasarlanan güvenli ağ geçidinin geniş ölçekli uygulamalarda da kullanılabileceği göz önüne alınarak sisteme esneklik ve yönetim kolaylığı sağlayabilmesi için IKEv2 otomatik güvenlik oturumu yönetim protokolü gerçeklenmiştir. Tasarlanan sistemde çalışacak olan kriptografik uygulamalar için önemli bir ihtiyaç olan rastgele sayı üreteci MCU'nun dahili sıcaklık sensörü ve SHA-1 hızlandırıcısı kullanılarak X9.82 Pseudo-Random sayı üreteci yöntemi kullanılarak tasarlanmıştır [20]. Sistemde gerçeklenen IKEv2 protokolü Diffie-Hellman anahtar değişimi yöntemi ile güvenlik oturumlarına ait anahtarları üretmektedir. Diffie-Hellman anahtar değişimi yönteminde çok büyük sayılarla üstel işlemler ve mod alma işlemleri yapılmaktadır. Bu aritmetik işlemleri yapabilmek için açık kaynak kodlu GMP Kütüphanesi tasarlanan sisteme dahil edilerek kullanılmıştır [21]. GMP kütüphanesi çalışabilmek için dinamik bellek yönetimine ihtiyaç duymaktadır. Tasarlanan sistemde GMP

kütüphanesinin çalışabilmesi için bir RAM alanı ayrılarak bu RAM alanını yönetecek bir dinamik bellek yönetimi tasarımı yapılmıştır. Tasarlanan güvenli ağ geçidinde TCP/IP, ICMP, DHCP İstemci, ARP gibi İnternet iletişim kuralları dizisi açık kaynaklı lwIP (lightweight IP) kütüphanesi kullanılarak sağlanmıştır [22]. Bu kütüphaneye IPsec altyapısı eklenmiş ve bu alt yapının MCU'nun donanımsal şifreleme hızlandırıcısını kullanabilmesi sağlanmıştır.

Tasarlanan sistemde kullanılacak donanımın nesnelerin İnterneti yapısına uygun olarak düşük maliyetli ve düşük güç tüketimine sahip olması hedeflenmiştir. Düşük maliyet ve düşük güç tüketimine sahip MCU'ların genellikle düşük işlem kapasitesine sahip CPU'lar içerdikleri ve sistemde gerçeklenecek olan güvenlik uygulamalarının kullandıkları kriptografik uygulamaların yüksek işlem gücü gerektirdiği göz önüne alınmıştır. Bu gereklilikler doğrultusunda sistemde performans kayıpları olmaması açısından seçilecek olan MCU'nun gerçeklenecek olan güvelik uygulamalarının kullanacağı kriptografik uygulamaları donanımsal olarak gerçekleştirebilecek çevre birimlere sahip olması ihtiyacı olduğu tespit edilmiştir. Tasarlanan güvenli ağ geçidinin gerçekleneceği donanımda I^2C, UART gibi seri haberleşme protokollerini desteklemesine dikkat edilerek ağ geçidinin Bluetooth, ZigBee gibi protokolleri kullanarak haberleşmesi gerektiği durumlarda bu protokoller için gerekli donanımların bir dönüştürücü ihtiyacı olmadan doğrudan ağ geçidine bağlanabilmesi hedeflenmiştir. Tasarlanan sisteme esneklik kazandırmak amacıyla ihtiyaç duyulması halinde sistemin gerçekleneceği donanım üzerindeki giriş/çıkış portları ve ADC'ler kullanılarak ağ geçidi üzerine de algılayıcı ve/veya aktüatör bağlanabilmesi hedeflenmiştir. Bu ihtiyaçlar doğrultusunda Texas Instruments firması tarafından üretilen düşük güç tüketimine rağmen yüksek işlem gücü sağlayan ve kriptografik fonksiyonları donanımsal olarak gerçekleştirmesiyle ön plana çıkan TM4C1294NCPDT mikrodenetleyicisinin kullanıldığı Şekil 4.2'de gösterilen TM4C1294[26] geliştirme kartı kullanılmak üzere seçilmiştir. Geliştirme kartının üzerinde Ethernet bağlantı alt yapısının bulunması da tercih sebebi olmuştur.



Şekil 4.2. TM4C129 Geliştirme Kartı[26]

Sistemde kullanılmak üzere seçilen geliştirme kartında kullanılan mikrodenetleyicinin öne çıkan özellikleri aşağıda listelenmiştir:

- 120 MHz saat frekansında 150 DMIPS [23] performans gösteren ARM Cortex-M4 tabanlı CPU
- 1024 KB Flash bellek, 256 KB RAM, 6 KB EEPROM
- CRC, AES, DES, SHA ve MD5 hesaplamalarını destekleyen güvenlik hızlandırıcısı
- UART, QSSI, I^2C, CAN, Ethernet MAC, Ethernet PHY ve USB haberleşme ara yüzleri

IKE SA ve IPsec SA için kullanılacak kriptografik algoritmalar güvenli ağ geçidi için seçilen MCU'nun donanımsal olarak gerçekleştirebildiği AES, DES, 3DES, SHA-1, SHA-2, ve MD5 algoritmalardan seçilerek kullanılmıştır. Donanımsal kripto hızlandırıcıları CPU'da koşturulacak yazılımlardan daha hızlı kriptografik işlemleri gerçekleştirmek için tasarlandıklarından bu donanımlar sistemde kullanılarak MCU'nun düşük işlem kapasitesine sahip CPU'su ağır işlem yükü gerektiren şifreleme ve özet çıkarma gibi kriptografik işlemlerin iş yükünden kurtarılmıştır. Böylelikle tasarlanan sistemde ağ gecikmelerinin ve güç tüketiminin düşürülmesi hedeflenmiştir.

4.3. Yazılım Tasarımı

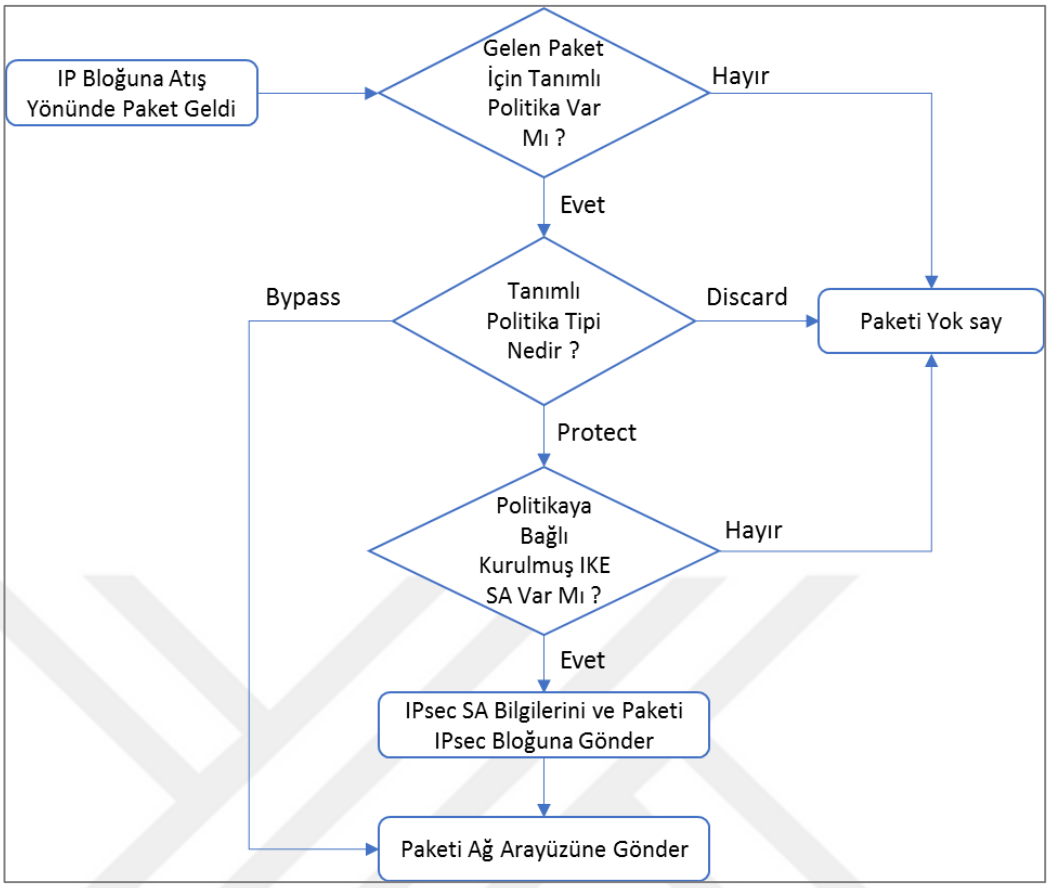
Tasarlanan sistem için gerçeklenen yazılım Code Composer Studio ortamında C programlama dili kullanılarak yazılmıştır. Tasarlanan gömülü sistemde bir işletim sistemi kullanılmamıştır.

4.3.1. IP yazılım bloğu tasarımı

Tasarlanan sistem nesnelerin İnterneti uygulamalarında nesneleri İnternete bağlayan bir ağ geçidi olarak çalışacağından ağ üzerinden temel haberleşmenin yapılabilmesi için ARP, DHCP Client, UDP, TCP ve ICMP gibi protokollere de ihtiyaç duyulmuştur. LwIP kütüphanesinin sisteme kazandırdığı özellikler aşağıda listelenmiştir:

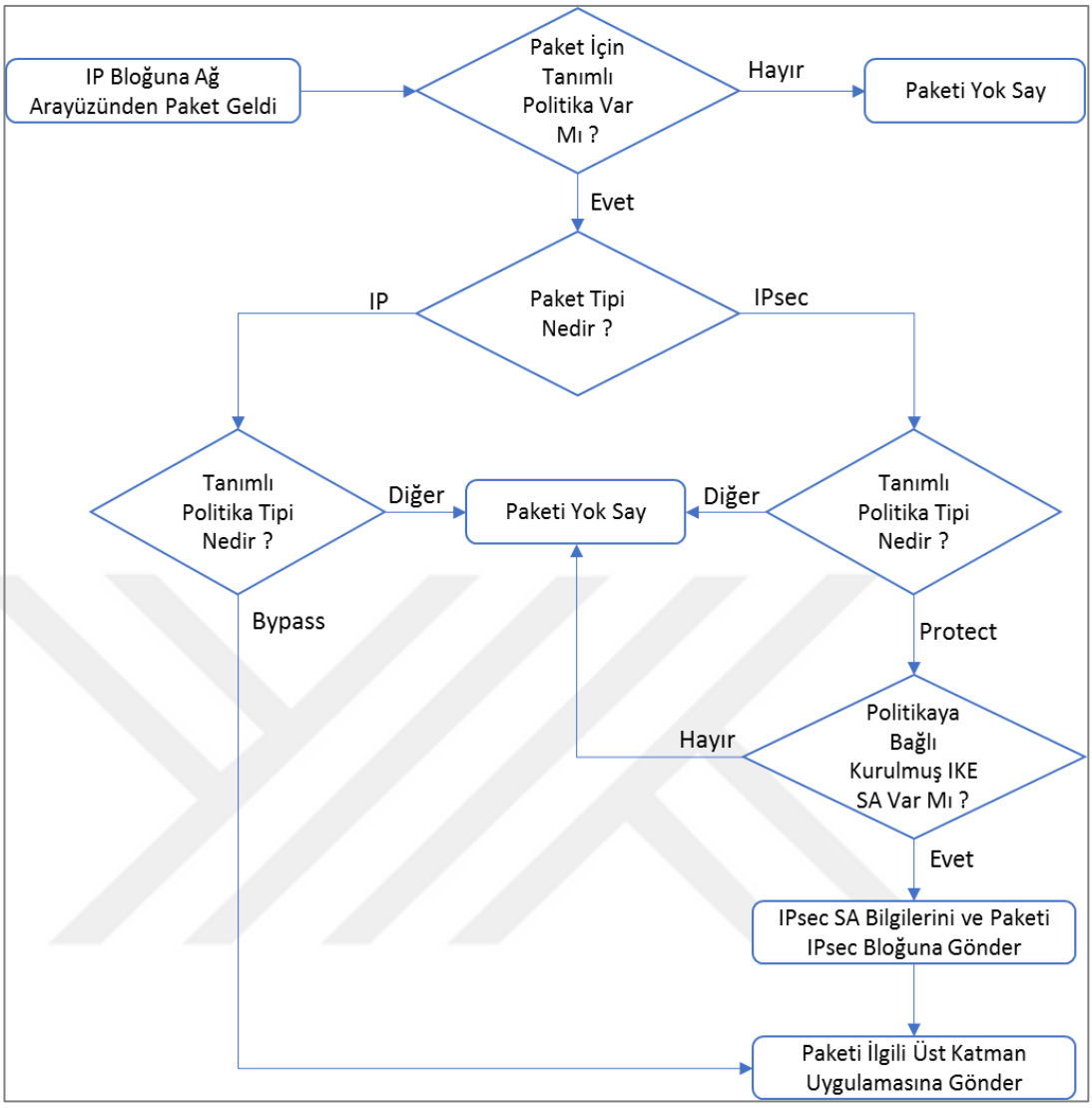
- İnternet Protokolü Versiyon 4 (IPv4)
- İnternet Kontrol Mesajı Protokolü (ICMP)
- Kullanıcı Veri Birimi Protokolü (UDP)
- İletim Kontrol Protokolü (TCP)
- Adres Çözümleme Protokolü (ARP)
- Dinamik Makine Yapılandırma Protokolü (DHCP)

Tasarlanan IP bloğu yazılımında paket alış ve atış yönleri ayrı ayrı ele alınmıştır. Tasarlanan güvenli ağ geçidinde çalışan uygulamalardan ağ arayüzüne iletilmesi için IP bloğuna iletilen paketlerin IP adresleri için öncelikle sistemde ayarlı bir güvenlik politikası olması gerekmektedir. IKE SA tarafından belirlenen güvenlik politikaları IKE çalışma dosyalarında saklanmıştır. Gönderilecek olan IP paketi için IKE çalışma dosyaları taranarak uygun bir politika bulunamazsa veya bulunan politika "Discard" tipindeyse paket yok sayılarak ağa iletilmemiştir. Paket için bulunan politika tipi "Bypass" ise pakete IPsec uygulanmadan doğrudan ağa iletilmiştir. Paket için bulunan politika tipi "Protect" ise pakete IPsec uygulanabilmesi için bulunan politika için kurulmuş olan IKE SA'lar taranmaktadır. Politika için kurulmuş bir IKE SA yok ise "Protect" politika için pakete IPsec uygulayabilecek güvenlik oturumu henüz oluşturulmadığı için paket yok sayılarak silinmiştir. Politika için kurulmuş bir IKE SA bulunursa bulunan IKE SA'ya ait IKE çalışma dosyasından IPsec tünelinde kullanılacak algoritmalar ve anahtarlar alınarak paket ile birlikte IPsec bloğuna teslim edilmiştir. Paket, IPsec bloğu tarafından şifrelenip kimlik ve veri bütünlüğü korumaları sağlandıktan sonra IP bloğuna geri teslim edilmiştir. IPsec bloğu tarafından işlenen paket IP bloğu tarafından ağ ara yüzüne teslim edilerek paket ağa iletilmektedir. Üst katman uygulamalarından IP bloğuna atış yönünde gönderilen paketlerin IP bloğu tarafında işlenmesi Şekil 4.3'te gösterilmiştir.



Şekil 4.3. Atış Yönündeki Paketlerin IP Bloğunda İşlenmesi

Tasarlanan sistemde ağ ara yüzünden gelen paketler için öncelikle tanımlı bir politika olup olmadığı kontrol edilmiştir. Tanımlı bir politika ile eşleştirilemeyen paketler yok sayılmıştır. Bir politikayla eşleşen paketlerin IPsec ile korumalı gönderilmiş bir paket olup olmadığı kontrol edilerek "Bypass" tipinde bir politika ile eşleşmeyen tüm IPsec korumasız paketler yok sayılmıştır. IPsec korumalı olarak gelen paketlerin ise "Protect" tipinde bir politika ile eşleşmiş olması gerekmektedir. Aksi halde bu paketler de yok sayılmıştır. "Protect" tipinde bir politika ile eşleşen paketler için politika için kurulu bir IKE SA olup olmadığı kontrol edilerek IKE SA olmadığı durumlarda bu paketleri IPsec bloğu işleyemeyeceği için paketler yok sayılmıştır.



Şekil 4.4. Alış Yönündeki Paketlerin IP Bloğunda İşlenmesi

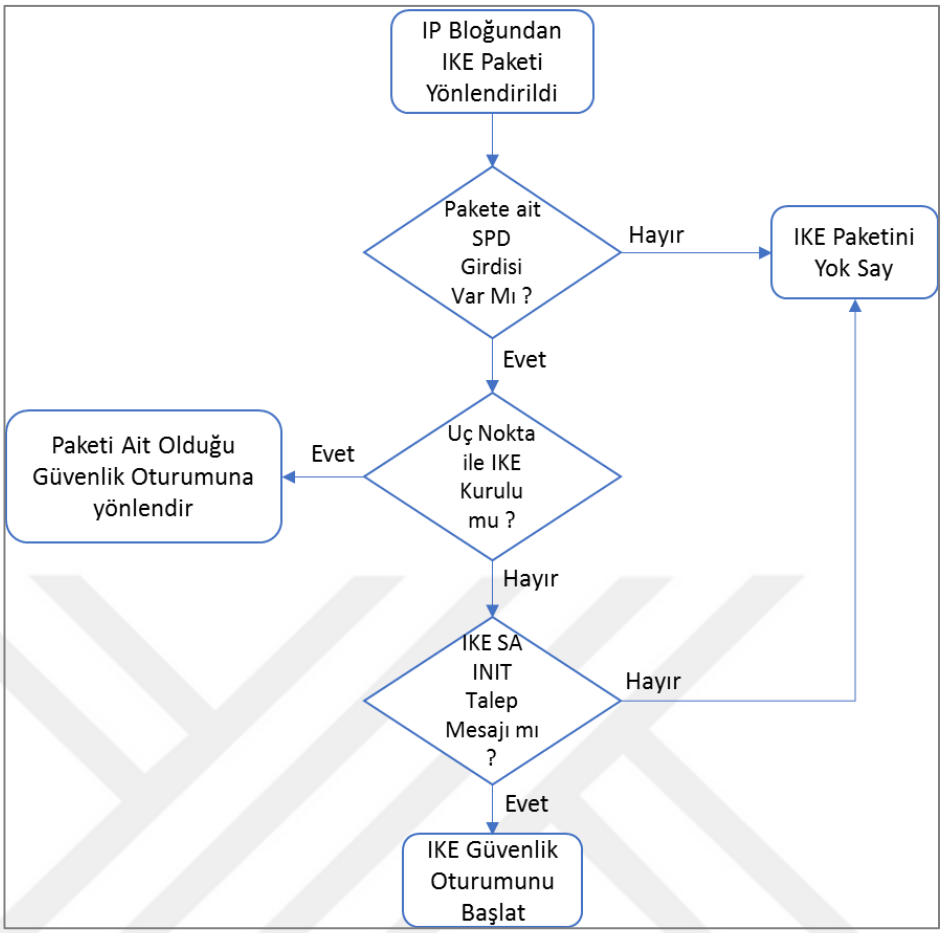
Politika için kurulu bir IKE SA mevcut ise IKE SA'ya ait IKE çalışma dosyası içinden IPsec tünelinde kullanılan algoritma ve anahtarlar alınarak paket ile birlikte IPsec bloğuna teslim edilmiştir. IPsec bloğu tarafından paketin şifresi çözülüp kimlik doğrulaması ve bütünlük kontrolü yapıldıktan sonra paket IP bloğuna geri teslim edilmiştir. IP bloğu tarafından teslim alınan şifresi çözülmüş paket işlenerek ilgili üst katman yazılımına gönderilmektedir. Ağ ara yüzünden IP bloğuna alış yönünde gelen paketlerin IP bloğu tarafında işlenmesi Şekil 4.4'te gösterilmiştir.

4.3.2. IKEv2 protokolü yazılım bloğu tasarımı

IKEv2 protokolü UDP kullanarak port numarası 500'den paket gönderir ve alır. Tasarlanan sistemde IP bloğu tarafından UDP port 500'e gelen paketler IKEv2

bloğuna yönlendirilmektedir. Tasarlanan sistemde kurulan IKE SA'ların bilgilerini tutabilmek için yaklaşık 20 KB RAM alanı ayrılmıştır. Bu RAM alanı 3 parçaya bölünerek parçalara "IKE çalışma dosyası" adı verilmiştir. Tasarlanan sistemde 3 adet IKE çalışma dosyası oluşturularak sistemde aynı anda en fazla 3 adet IKE SA kurulabilmesi için yeterli RAM alanı ayrılmıştır.

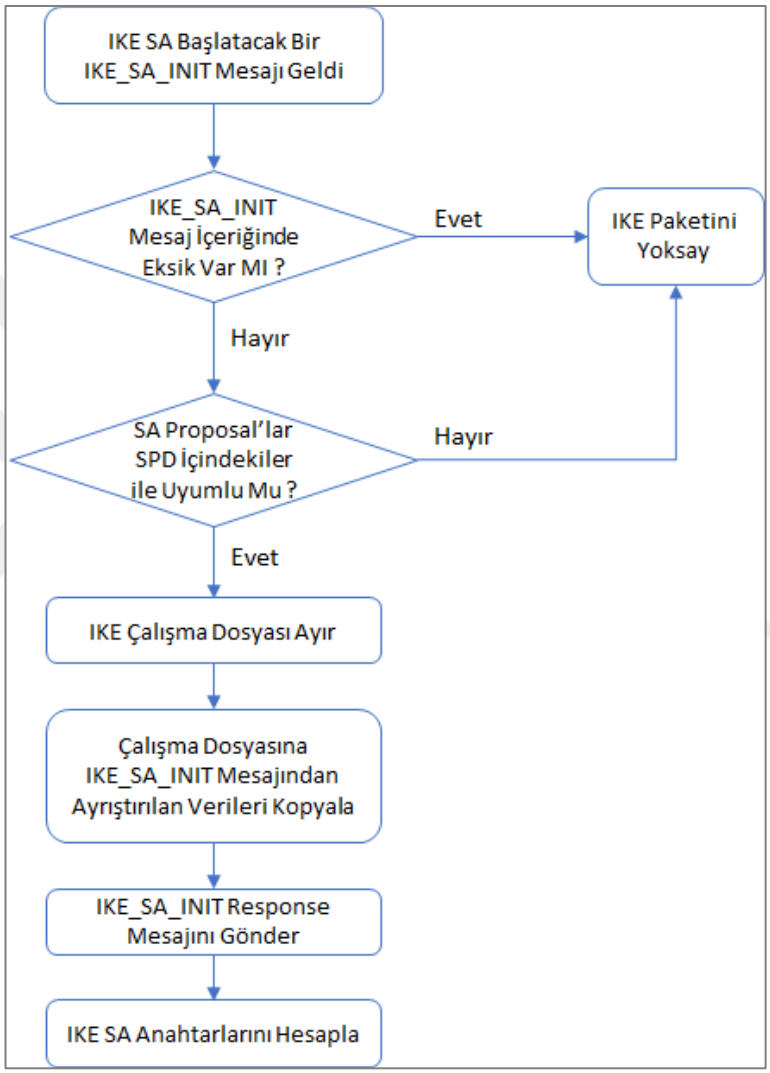
Tasarlanan IKEv2 bloğuna IP bloğundan yönlendirilen IKE paketinin güvenli ağ geçidinde IKE tüneli kurulması için ayarlanan bir uç noktadan gelip gelmediğinin anlaşılabilmesi için SPD girdileri taranmaktadır. Eğer gelen paket sistemde ayarlanan bir uç noktadan gelmiyor ise herhangi bir SPD girdisi bulunamayacağı için gelen IKE paketi yok sayılarak silinir ve herhangi bir işlem yapılmaz. Gelen paket sistemde ayarlanan bir uç noktadan geliyor ise paketi gönderen uç nokta ile hali hazırda kurulu bir IKE SA olup olmadığı kontrol edilir. Sistemde ayarlanan uç nokta ile kurulu bir güvenlik oturumu varsa gelen paket o güvenlik oturumu için ayrıştırılarak paket içeriğine göre IKE protokol detaylarında belirtilen işlemler gerçekleştirilmektedir. Sistemde ayarlanan uç nokta ile kurulu bir güvenlik oturumu yoksa gelen paketin IKE güvenlik oturumunu başlatabilecek bir IKE_SA_INIT talep mesajı olup olmadığına bakılır. Güvenlik oturumu başlatabilecek bir paket ise IKE protokol detaylarında açıklanan IKE güvenlik oturumu başlatma işlemleri gerçekleştirilmektedir. Gelen paket IKE güvenlik oturumu başlatamayacak bir mesaj değişimine ait ise hatalı paket geldiği varsayımı yapılarak paket ret edilmektedir. IKEv2 bloğuna gelen paketlerin işlenmesi Şekil 4.5'te gösterilmiştir.



Şekil 4.5. IKEv2 Bloğuna Gelen Paketlerin İşlenmesi

Tasarlanan IKEv2 bloğuna IP bloğundan yönlendirilen IKE paketi güvenlik oturumunu başlatabilecek bir IKE_SA_INIT talep mesajı ise öncelikle gelen mesaj içeriğinde IKE SA kurulabilmesi için bulunması zorunlu olan anahtar değişimi, güvenlik oturumu ve Nonce yüklerinin varlığı kontrol edilmektedir. Zorunlu yüklerde eksik varsa gelen IKE paketi yok sayılmaktadır. Paket içeriği kontrol edildikten sonra güvenlik oturumu yükü içinde alınan şifreleme algoritmalarının, doğrulama algoritmalarının, PRF fonksiyonlarının ve DH grubunun SPD'de ayarlanan güvenlik algoritmaları ile uyumlulukları kontrol edilmektedir. Uç noktadan alınan teklifler SPD ile uyumsuz ise IKE paketi yok sayılmaktadır. Alınan teklifler SPD ile uyumlu ise IKE SA kurulumunu başlatmak için öncelikle yeni bir IKE çalışma dosyası ayrılmaktadır. IKE SA kurulumu için ayrılan çalışma dosyasında IKE SA ile ilgili veriler tutulmaktadır. Gelen paket için bulunan SPD girdisindeki veriler ayrılan IKE çalışma dosyasına kopyalanmaktadır. Gelen paketin IKE protokol başlığında güvenlik oturumunu belirlemek için yer alan SPI değeri IKE SA'ları ayırt edebilmek için ayrılan

IKE çalışma dosyasına kopyalanmıştır. Güvenlik oturumu yükünde sistem ile uyumlu olduğu tespit edilen Proposal numarası gönderilecek olan IKE_SA_INIT cevap mesajında kullanılmak üzere IKE çalışma dosyasına kopyalanmıştır. Uç noktanın gönderdiği IKE mesajının yükleri ayrıştırılarak IKE çalışma dosyasına daha sonraki işlemlerde kullanılmak üzere kopyalanmaktadır.



Şekil 4.6. IKE_SA_INIT Mesajının İşlenmesi

Yük ayrıştırma işlemi tamamlandıktan sonra IKE_SA_INIT mesajına cevap gönderilerek mesaj değişimi tamamlanmıştır. Mesaj gönderimi tamamlandıktan sonra IKE_AUTH mesaj değişiminde gelecek olan paketler şifreli olacağı için IKE SA'ya ait anahtarlar üretilerek IKE çalışma dosyasına kopyalanmıştır. Aşağıdaki şekilde IKE_SA_INIT mesajı sisteme geldikten sonra yapılan işlemler Şekil 4.6'da gösterilmiştir.

Tasarlanan sistem tarafından IKE_SA_INIT mesajına içinde güvenlik oturumu, anahtar değişim ve Nonce yükleri bulunan bir cevap mesajı hazırlanarak gönderildikten sonra IKE SA için gerekli anahtarlar üretilmektedir. Cevap gönderilirken öncelikle rastgele sayı üreteci kullanılarak 8 bayt uzunluğunda SPI üretilmektedir. Üretilen SPI IKE başlığına yazıldıktan sonra IKE SA için kullanılacak olan algoritmaların hangileri olduğu IKE çalışma dosyasında okunarak güvenlik oturumu yükü oluşturulmaktadır.

Anahtar değişimi yükü ile paylaşılacak olan anahtar değişim verisi hesaplanırken Diffie-Hellman anahtar değişimi yöntemi kullanılmaktadır. DH anahtar değişimini gerçekleştirebilmek için taban ve modülüs değerleri güvenlik oturumu yükünde seçilen DH grubundan gelmektedir. Üstel olarak ise uç noktaların gizli anahtarları kullanılmaktadır. Tasarlanan sistemde rastgele sayı üreteci kullanılarak toplam 64 bayt uzunluğunda gizli anahtar üretilmektedir. Gizli anahtar üretildikten sonra DH protokol detaylarında anlatıldığı gibi matematiksel işlemler tamamlanarak anahtar değişimi yükü oluşturulur.

Nonce yükü ile paylaşılacak nonce verisi her mesaj için farklı olarak üretilmeli ve bir kez kullanıldıktan sonra tekrar kullanılmamalıdır. Nonce verisi rastgele sayı üreteci kullanılarak toplam 96 bayt uzunluğunda üretilip Nonce yükü oluşturulmaktadır.

Uç noktadan alınan IKE_SA_INIT mesajına gönderilecek mesajın oluşturulması tamamlandıktan sonra oluşturulan paket IP bloğuna teslim edilmiştir. IP bloğu yazılımı tarafından IKE mesajının ağ üzerinden uç noktaya gönderilmesi sağlanmıştır.

4.3.2.1. IKE SA anahtarlarının hesaplanması

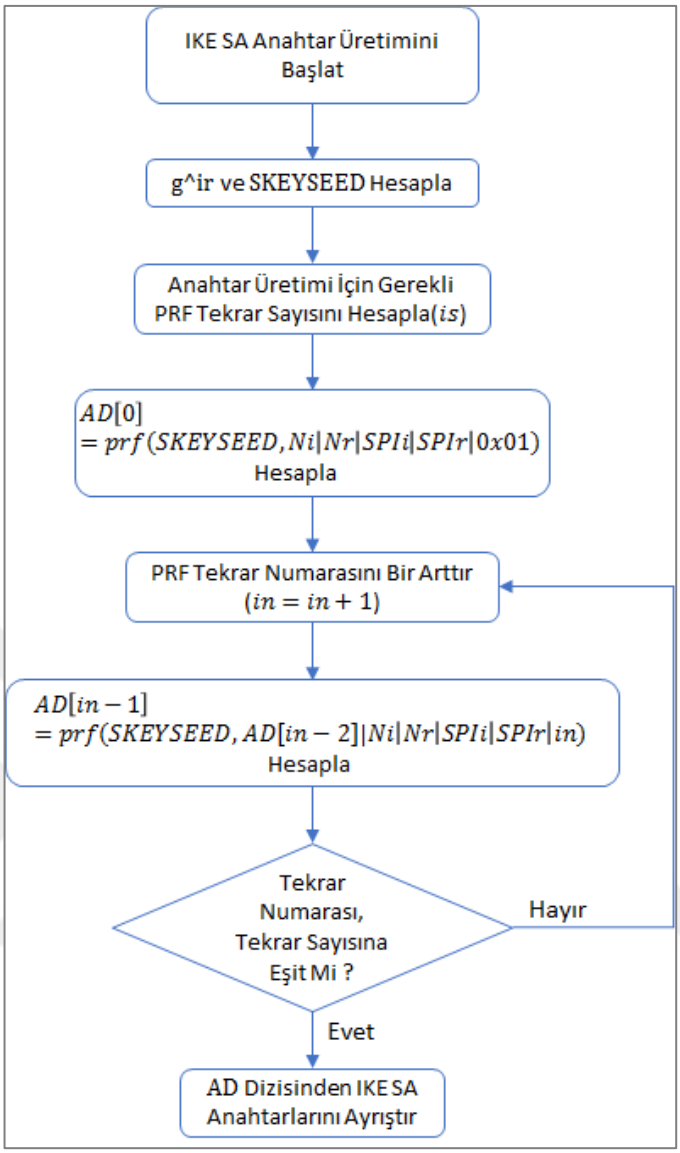
IKE SA anahtarlarının üretilebilmesi için öncelikle DH anahtar değişimi tamamlanmalıdır. IKE_SA_INIT mesajı oluşturulurken üretilen gizli anahtar ve DH grubu kullanılarak g^{ir}(Ortak Gizli Anahtar) sayısı Denklem (4.1)'de gösterildiği gibi;

$$g^{ir} = KEi^{\text{Gizli Anahtar}} \text{Mod DH Asal Sayısı} \quad (4.1)$$

hesaplanmaktadır[11].

Denklem (4.1)'de gösterilen aritmetik işlemler çok büyük sayılar ile yapıldığı için bu denklemin hesaplanabilmesi GMP kütüphanesinin mpz_powm fonksiyonu kullanılmıştır. Hesaplanan ortak gizli anahtar kullanılarak SKEYSEED değeri hesaplanmaktadır. SKEYSEED hesaplanırken donanımsal HMAC hızlandırıcısına anahtar olarak uç noktaların Nonce değerleri birleştirilerek kullanılacağı için birleştirilmiş Nonce uzunluğu ile anahtar uzunluğunun uyumluluğu kontrol edilmektedir. Eğer birleştirilmiş Nonce uzunluğu gerekli anahtar uzunluğundan daha az ise birleştirilmiş Nonce sayısının sonuna gerekli anahtar uzunluğuna erişinceye kadar sıfır eklenmektedir. Eğer birleştirilmiş Nonce uzunluğu gerekli anahtar uzunluğundan daha fazla ise donanımsal HASH hızlandırıcısı ile birleştirilmiş Nonce'ın özeti çıkartılarak çıkartılan özet HMAC hızlandırıcısına anahtar olarak verilmektedir. Birleştirilmiş Nonce ile ilgili kontroller ve gerekli ise düzeltmeler gerçekleştirildikten sonra donanımsal HMAC hızlandırıcına veri olarak hesaplanan ortak gizli anahtar, anahtar olarak birleştirilmiş Nonce verilerek SKEYSEED değerinin hesaplanması beklenmektedir.

SKEYSEED hesaplaması tamamlandıktan sonra IKE SA'da kullanılan algoritmalar için gerekli toplam anahtar uzunluğu hesaplanarak PRF ile kaç tekrar sonrasında gerekli anahtarların tamamının hesaplanabileceği bulunmaktadır. IKEv2 protokol detaylarında prf+ olarak anlatılan fonksiyon donanımsal HMAC hızlandırıcısı tekrarlı olarak kullanılarak IKE SA'ya ait anahtarların tamamı üretilmektedir. Üretilen anahtarlar IKE protokol detaylarında anlatılan sıra ile ayrıştırılarak IKE çalışma dosyasına kopyalanmıştır. Sistemde IKE SA anahtarlarının nasıl üretildiği ve prf+ fonksiyonunun gerçeklenmesi Şekil 4.7'de gösterilmiştir. Şekil 4.7'de hesaplanan gerekli tekrar sayısı is kısaltması ile, gerçekleştirilen tekrar sayısı in kısaltması ile gösterilmiştir. Tekrar sayısı 1'den başlamaktadır.



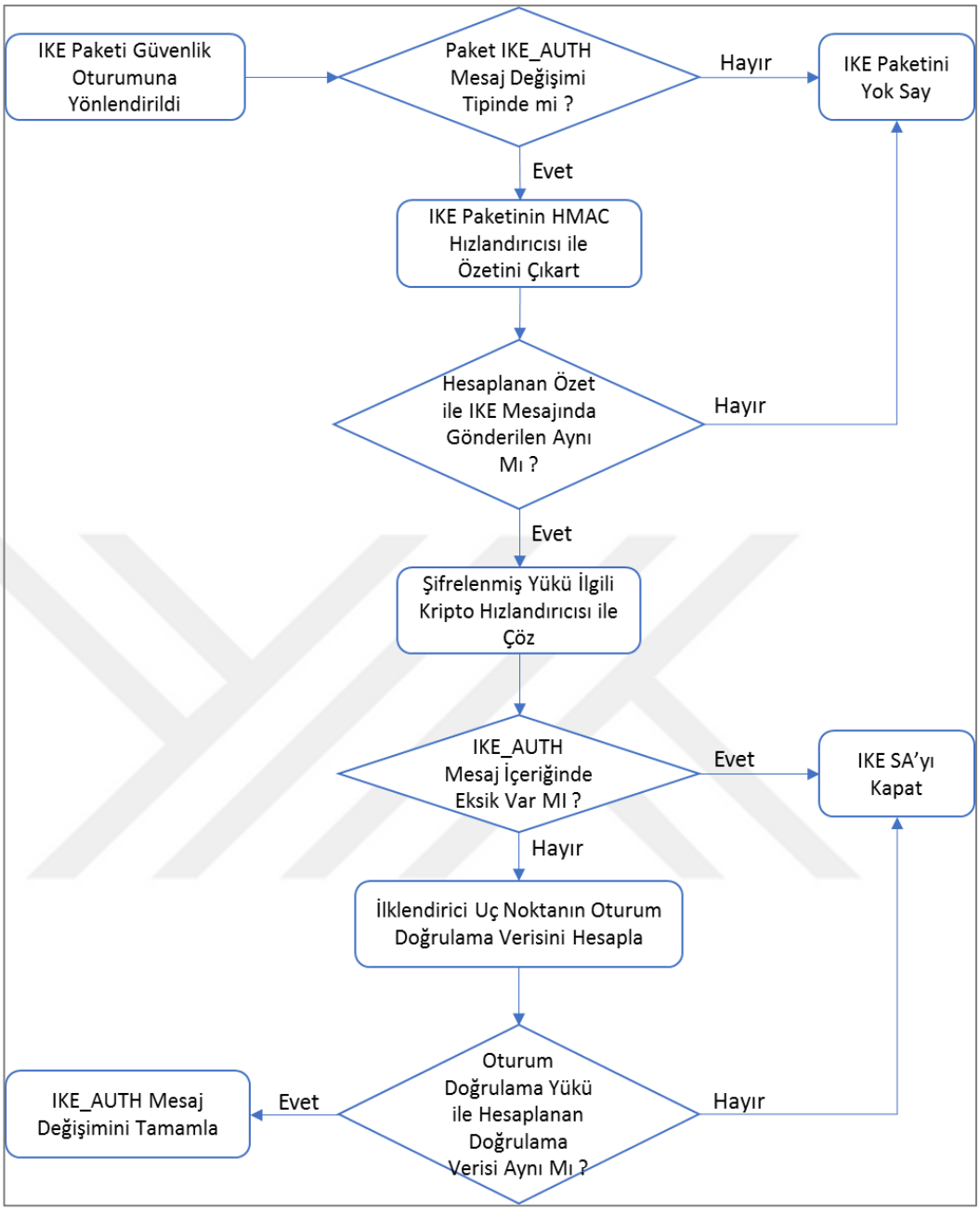
Şekil 4.7. IKE SA Anahtarlarının Hesaplanması

4.3.2.2. IKEv2 güvenlik oturumunun doğrulanması

Bu tez kapsamında önerilen IKEv2 bloğu tasarımında IKE SA ve IPsec SA kurulumu için zorunlu olmayan CREATE_CHILD_SA ve INFORMATIONAL mesaj değişimleri için işleyici yazılım yapıları gerçeklenmemiştir. Tasarlanan sistem IKE SA oturumu kurmak için karşı uç noktanın IKE_SA_INIT mesaj değişimini başlatmasını beklemektedir. IKE_SA_INIT mesaj değişimine ait mesajlar geldiğinde IKE güvenlik oturumu başlatılmaktadır. Bu nedenle güvenlik oturumuna yönlendirilen bir paketin IKE_AUTH mesaj değişimine ait olması gerekmektedir.

IKE_SA_INIT mesaj değişimi tamamlandıktan sonra gelen IKE paketinin IKE_AUTH mesaj değişimine ait olması beklenmektedir. IKE_AUTH mesaj değişimi haricinde bir mesaj değişimine ait paket yönlendirilirse sistem tarafından gelen paket işlenmeyerek yok sayılmaktadır. Bu nedenle öncelikle güvenlik oturumuna yönlendirilen bir paketin IKE_AUTH mesaj değişimine ait olup olmadığı kontrol edilir. IKE_AUTH mesaj değişimi için gönderilen paketler IKEv2 paket başlığından başlanarak ayrıştırılır.

Tasarlanan IKEv2 bloğuna gelen bir IKE paketi eğer hali hazırda kurulu bir IKE güvenlik oturuma gönderilmişse gelen paketin içindeki yükler ayrıştırılarak çıkarılan yüklere ve paketin hangi mesaj değişimi için gönderildiğine bağlı olarak paket içeriğinin değerlendirilmesi Şekil 4.8'de gösterilmiştir.



Şekil 4.8. IKEv2 Güvenlik Oturumunun Doğrulanması

IKE_AUTH mesaj değişimi IKE SA kullanılarak şifreli, veri bütünlüğü ve kaynak korumalı olarak gerçekleştirildiği için öncelikle IKE paketinin sonunda yer alan özet tabanlı mesaj doğrulama kodu (HMAC) tekrar hesaplanarak gelen mesajın bütünlüğü ve kaynağı doğrulanır. Gelen IKE_AUTH mesajının IKE protokol başlığındaki ilk SPI değerinin RAM adresi ve bu RAM adresinden ne kadar bayt sonrasına kadar HMAC hesaplanacağı donanımsal HMAC hızlandırıcısına giriş olarak verilir. Paketi gönderen uç noktanın IKE SA'da gönderdiği paketlerin kaynağının doğrulanabilmesi için HMAC hesaplarken kullandığı SK_ai anahtarı da donanımsal HMAC hızlandırıcısına

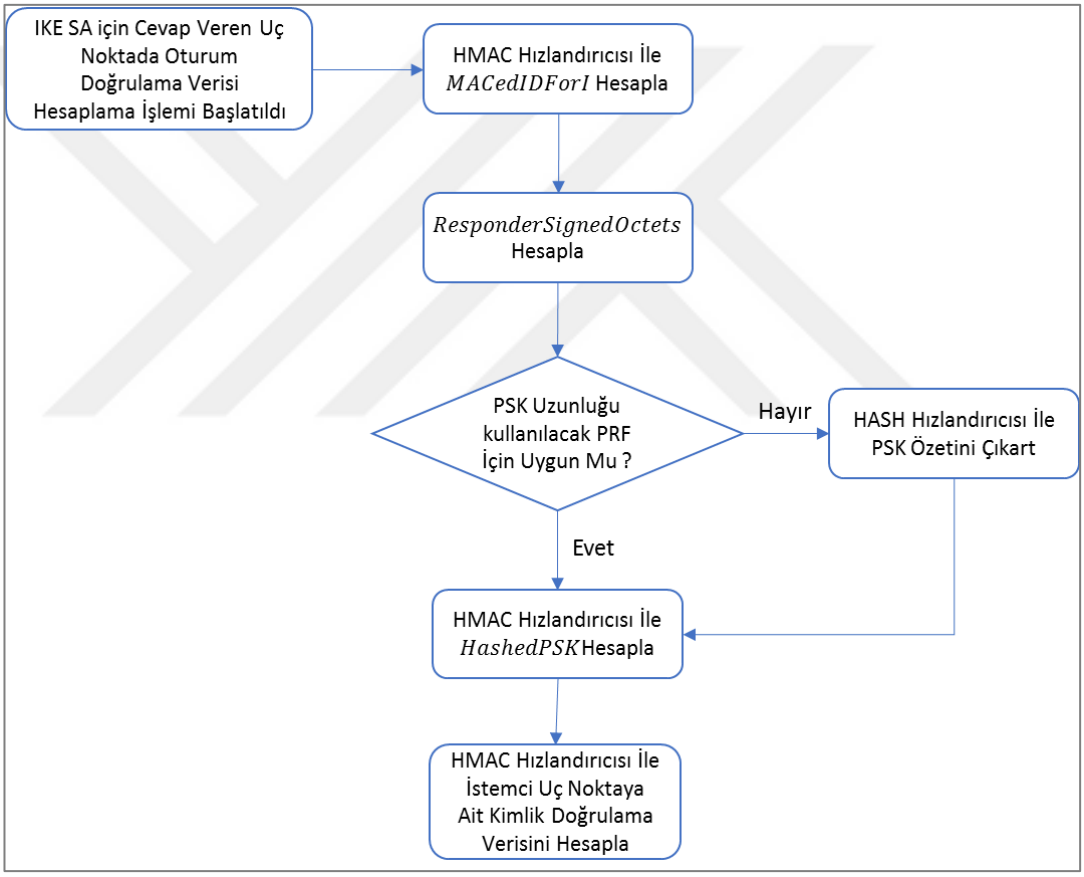
giriş olarak verilerek donanımın HMAC sonucunu hesaplaması beklenir. Sonuç hesaplandıktan sonra IKE paketi içindeki bütünlük kontrol değeri ile hesaplanan HMAC karşılaştırılarak paketin bütünlüğü ve kaynağı doğrulanır. Doğrulama sonucu başarısız olursa paket silinerek yok sayılır. Başarılı olursa paket içindeki şifreli yükün çözülmesi aşamasına geçilir.

IKE_AUTH mesaj değişiminde karşı uç noktadan şifrelenerek gönderilen şifreli yük verisi donanımsal şifre çözme hızlandırıcısı kullanılarak çözülür. Şifrelenmiş yük içerisindeki ilklendirme vektörünün ve şifrelenmiş yük içindeki şifreli verinin RAM'deki başlangıç adresleri donanımsal şifre çözme hızlandırıcısına verilir. Paketi gönderen uç noktanın IKE paketlerini şifrelemek için kullandığı SK_ei anahtarı ve şifreli verinin ne kadar bayt uzunluğunda olduğu da donanımsal şifre çözme hızlandırıcısına verilerek donanımın şifreli veriyi çözmesi beklenir. Şifreli verinin çözülme işlemi tamamlandıktan sonra çözülmüş veri içindeki yükler ayrıştırılmaya başlanır.

Şifrelenmiş veri içinde IKE SA ve IPsec SA kurulabilmesi için bulunması zorunlu olan IDi, AUTH, SA, TSi ve TSr yüklerinin varlığı kontrol edilir. Zorunlu yükler veri içerisinde mevcut ise bu veriler IKE SA'da kullanılmak üzere IKE SA'ya ait RAM'de ayrılan çalışma dosyasına kopyalanır.

Paket içeriği kontrol edildikten sonra paket içinden ayrıştırılan oturum doğrulama yükü tekrar hesaplanarak IKE_AUTH mesajını gönderen uç noktanın kimliğinin doğrulanması Şekil 4.9'da gösterilmiştir. Oturum doğrulama yükünü hesaplamak için öncelikle oturumun başında uç nokta tarafından IKE_SA_INIT mesajı ile gönderilen ID yükünün saklandığı RAM adresi, ID yükünün uzunluğu ve SK_pi anahtarı donanımsal HMAC hızlandırıcısına giriş olarak verilerek donanımın MACedIDForI değerini hesaplanması beklenmektedir. Hesaplama tamamlandıktan sonra hesaplanan değer kullanılarak InitiatorSignedOctets verisi oluşturulur. RAM'de sabit olarak tutulan "Key Pad for IKEv2" dizisinin başlangıç adresi, PSK değerinin başlangıç adresi ve "Key Pad for IKEv2" dizisinin uzunluğu donanımsal HMAC hızlandırıcısına giriş olarak verilerek donanımın $hashed\_psk$ değerini hesaplanması beklenmektedir. Bu aşamadan önce PSK uzunluğunun HMAC algoritmasının anahtar uzunluğu ile uyumlu olup olmadığı kontrol edilmelidir. Olması gereken anahtar uzunluğundan daha

uzun bir PSK ayarlanmışsa öncelikle PSK'nın HASH fonksiyonu ile özeti çıkartılarak HMAC algoritması için uygun boyuta indirilmesi gerekmektedir. InitiatorSignedOctets verisinin RAM adresi, InitiatorSignedOctets verisinin uzunluğu ve önceki aşamada hesaplanan *hashed_psk* değerinin RAM adresi donanımsal HMAC hızlandırıcısına giriş olarak verilerek donanımın ilklendirici uç noktaya ait oturum doğrulama verisini hesaplanması beklenmektedir. Hesaplanan değer ile ilklendirici uç nokta tarafından oturum doğrulama yükü ile gönderilen değer karşılaştırılarak IKE SA oturumu doğrulanarak IKE_AUTH mesajına cevap gönderilir. Eğer oturum doğrulanamazsa IKE SA kapatılır.



Şekil 4.9. İlklendirici Uç Noktanın Oturum Doğrulama Verisi

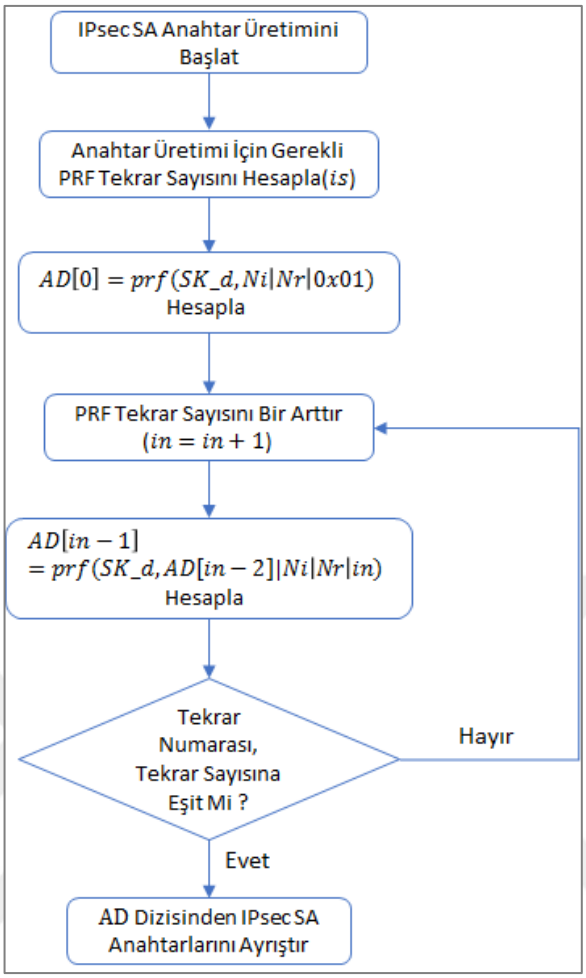
4.3.2.3. IPsec SA anahtarlarının hesaplanması

Tasarlanan sistem tarafından IKE_AUTH mesajına cevap gönderilmeden önce IPsec SA için gerekli anahtarlar hesaplanmaktadır. IPsec SA için gerekli anahtarlar IKE SA'da olduğu gibi donanımsal HMAC hızlandırıcısını tekrarlı olarak kullanan prf+ fonksiyonu kullanılarak üretilmiştir. IPsec SA'da kullanılacak algoritmalar için gerekli

toplam anahtar uzunluğu hesaplanarak PRF ile kaç tekrar sonrasında gerekli anahtarların tamamının hesaplanabileceği bulunmaktadır. Üretilen anahtarlar IKE protokol detaylarında anlatılan sıra ile ayrıştırılarak IKE çalışma dosyasına kopyalanmıştır. Sistemde IPsec SA anahtarlarının nasıl üretildiği ve prf+ fonksiyonunun gerçeklenmesi Şekil 4.10'da gösterilmiştir. Şekil 4.10'da hesaplanan gerekli tekrar sayısı is kısaltması ile, gerçekleştirilen tekrar sayısı in kısaltması ile gösterilmiştir. Tekrar sayısı 1'den başlamaktadır.

IPsec SA için anahtar üretimi tamamlandıktan sonra IKE_AUTH mesaj değişimini tamamlamak için içinde sadece şifrelenmiş yük bulunan bir IKE_AUTH cevap mesajı hazırlanarak gönderilmiştir. Oluşturulmaya başlanan IKE_AUTH mesajının IKE protokol başlığına IKE çalışma dosyasından okunan SPI ve mesaj ID numaraları yazılmaktadır. Sonraki yük kısmına mesaj içindeki tek yük olan şifrelenmiş yük numarası (35) yazılmıştır.

Şifrelenmiş yük başlığında yer alan ilklendirme vektörü alanına rastgele sayı üreteci kullanılarak IKE SA'da kullanılan şifreleme algoritması için gerekli uzunlukta bir sayı üretilmektedir. Oluşturulmaya başlanan şifrelenmiş yükün içine kimlik tanımlama, oturum doğrulama, güvenlik oturumu ve trafik seçme yükleri eklenerek şifrelenecek toplam veri boyutu hesaplanmıştır. Toplam veri boyutu, şifrelemede kullanılacak algoritmanın blok boyutu ile örtüşmüyorsa şifrelenecek verinin sonuna toplam veri boyutu şifreleme algoritmasının blok boyutu ve katları olana kadar dolgu işlemi yapılmıştır. Yapılan dolgu işleminin paket boyutunu değiştireceği dikkate alınmalıdır.



Şekil 4.10. IPsec SA Anahtarlarının Hesabı

Şifrelenmiş yük ve IKE protokol başlığının oluşturulması tamamlandıktan sonra paket için üretilen ilklendirme vektörü, şifrelenmiş yük içindeki şifreli veri alanının RAM'deki başlangıç adresi, şifrelenecek verinin toplam uzunluğu ve gönderilecek paketi şifrelemekte kullanılan SK_{er} anahtarı donanımsal şifreleme hızlandırıcısına verilerek donanımın veriyi şifrelemesi beklenmektedir. Şifreleme işlemi tamamlandıktan sonra şifreli veri şifrelenmiş yüke kopyalanarak paketin kimlik ve veri bütünlüğünün korunması aşamasına geçilmiştir.

İlklendirici uç noktanın gönderdiği oturum doğrulama verisinin tekrar hesaplanması aşamasında yapılan işlemler cevap veren uç nokta için de yapılarak oturum doğrulama verisi hesaplanmıştır. Hesaplanan oturum doğrulama verisi oturum doğrulama yükü ile gönderilmiştir.

IPsec SA için seçilen algoritmaların hangileri olduğu IKE çalışma dosyasından okunarak oluşturulan güvenlik yükü içindeki tekliflere yazılmıştır. Oluşturulan IKE paketinin IKE protokol başlığındaki ilk SPI değerinin RAM adresi, gönderilen pakette kimlik doğrulamasının sağlanması için kullanılan SK_{ar} anahtarı ve HMAC hesaplanacak verinin toplam uzunluğu donanımsal HMAC hızlandırıcısına giriş olarak verilerek donanımın ürettiği sonuç değeri IKE paketinin sonuna kopyalanmıştır. Böylelikle Şifrelenip, veri bütünlüğü ve kaynak koruması sağlanan IKE paketi IP bloğuna teslim edilerek ağ üzerinden karşı uç noktaya ulaştırılması sağlanmıştır.

IKE_AUTH mesajına cevap gönderildikten sonra IKE SA ve IPsec SA kurulumu tamamlanmaktadır. SA kurulumları tamamlandıktan sonra IKE ve IPsec tünellerinden güvenli haberleşme sağlanabilmektedir.

4.3.3. IPsec yazılım bloğu tasarımı

Bu tez çalışmasında önerilen sistemde IPsec protokolü ile işlenmesi gereken paketlere IP bloğunda karar verilecek bir tasarım yapılmıştır. IP bloğunda pakete IPsec protokolü uygulanacağına karar verildikten sonra IPsec tüneli için gerekli algoritmalar ve anahtarlar da IPsec yazılım bloğuna iletilerek paketin işlenmesi sağlanmıştır. Paket işlenirken yapılan kriptografik işlemler sistemde kullanılan MCU'nun ilgili donanımsal hızlandırıcıları kullanılarak gerçekleştirilmiştir.

IP yazılım bloğundan atış yönünde bir paket IPsec bloğuna gönderildiğinde öncelikle paket boyutunun şifreleme için kullanılacak olan algoritmanın blok boyutu ile örtüşmüyorsa şifrelenecek verinin sonuna toplam veri boyutu şifreleme algoritmasının blok boyutu ve katları olana kadar dolgu işlemi yapılmaktadır. Yapılan toplam dolgu uzunluğu ESP paketinin dolgu uzunluğu alanına yazılmıştır. Dolgu yapılarak şifreleme algoritmasının blok boyutu ile uyumlu hale getirilen paketin şifrelenebilmesi için gerekli olan ilklendirme vektörü rastgele sayı üreteci kullanılarak üretilmiştir. Üretilen ilklendirme vektörü ve CA_SK_{re} anahtarı ile donanımsal şifreleme hızlandırıcısı kullanılarak paketin şifrelenmesi tamamlanmıştır. Şifrelenmiş paketin üzerine yerleştirilen ESP başlığındaki paket sıra numarası ve SPI alanlarına IKE çalışma dosyasından IP bloğu tarafından yönlendirilen veriler kullanılarak doldurulmuştur. Paketi şifrelemek için üretilen ilklendirme vektörü de ESP başlığına kopyalanarak ESP paket formatı tamamlanmıştır.

ESP paketinin kaynak ve bütünlük korumasının sağlanabilmesi için ESP başlığından şifrelenmiş paketin sonuna kadar olan verinin donanımsal HMAC hızlandırıcısı kullanılarak CA_SK_{ra} anahtarı ile özeti çıkarılmıştır. Çıkartılan özet ESP paketinin sonuna kopyalanarak paketin kaynak ve bütünlük koruma işlemi tamamlanmıştır.

Son olarak ESP başlığının üzerine bir dış IP başlığı getirilerek paketin kapsüllenmesi gerçekleştirilmektedir. Dış IP başlığına tasarlanan güvenli ağ geçidinin DHCP sunucusundan aldığı IP adresi, karşı uç noktanın IP adresi, toplam paket boyutu ve protokol alanına ESP protokol numarası (50) yazılarak oluşturulan paket ağ ara yüzüne iletilmek üzere IP bloğuna gönderilmiştir.

IP yazılım bloğundan alış yönünde bir paket IPsec bloğuna gönderildiğinde öncelikle gelen paketin kaynak ve bütünlük koruması kontrol edilmektedir. Bunun için ESP başlığından şifrelenmiş paketin sonuna kadar olan verinin donanımsal HMAC hızlandırıcısı kullanılarak CA_SK_{ia} anahtarı ile özeti çıkarılmıştır. Çıkarılan özet ile ESP paketinin sonunda gönderilen kaynak ve bütünlük doğrulama verisi karşılaştırılarak bir farklılık tespit edilirse paket yok sayılarak silinmiştir. Üretilen veri ile paket sonundaki veri bire bir aynı ise şifreli paketin çözülmesi için ESP başlığında bulunan ilklendirme vektörü ve CA_SK_{ie} anahtarı kullanılarak donanımsal şifreleme hızlandırıcısı ile şifreli paket çözülmüştür. Çözülen paket üst katman uygulamalarına iletilmek üzere IP bloğuna geri gönderilmiştir.

4.3.4. Rastgele sayı üreteci tasarımı

Tasarlanan sistemde çalışacak olan kriptografik uygulamalar için önemli bir ihtiyaçtır. Tasarlanan güvenli ağ geçidinde kullanılmak üzere tasarlanan rastgele sayı üretecinde [20]'de entropi kaynağı olarak kullanılabileceği belirtilen mikroişlemcinin dahili sıcaklık sensörü ve SHA-1 hızlandırıcısı kullanılarak X9.82 Pseudo-Random sayı üreteci yöntemi kullanılarak tasarlanmıştır [20].

IKE, IPsec gibi tüm kriptografi sistemleri şifreleme için gerekli anahtar üretiminde gizli ve tahmin edilemez gerçek rastgele sayılara ihtiyaç duyarlar. Bu rastgele sayılar kriptografi sistemlerine yönelik olası bir saldırıda çok düşük tahmin edilebilirlik ve hesaplanabilirliğe sahip olmalıdırlar. Saat gibi kaynakların kullanıldığı rastgele sayı üreteçleri sınırlı sayıda çıkış aralığı verdikleri için çoğu rastgelelik testinden

geçemezler. Prensip olarak harici sensörler (radyo alıcıları, sıcaklık sensörleri) entropi kaynağı olarak kullanılabilmektedirler. Termal gürültü (entegre devrelerde Johnson gürültüsü olarakta anılır) veya radyoaktif bozunum kaynakları ve hızlı bir free-running osilatör doğrudan entropi kaynağı olarak kullanılabilirler [24].

X9.82'ye göre ilk olarak rastgele sayı üreteci ilklendirilmelidir. İlklendirme işlemine HMAC fonksiyonunun kullanacağı giriş datası(V) ve anahtar(K) sıfırlanarak başlanır. V ve K dizileri 0 ile doldurulduktan sonra Denklem (4.2)'de gösterildiği gibi;

K=HMAC(K, V|0x00|EntropiDeğeri) (4.2)

K değeri entropi kaynağından elde edilen veri kullanılarak tekrar hesaplanmıştır[20].

Hesaplanan K anahtarı ile HMAC fonksiyonu kullanılarak sırası ile Denklem (4.3) (4.4) ve (4.5)'te gösterildiği gibi;

V=HMAC(K, V) (4.3)

K=HMAC(K, V|0x01|EntropiDeğeri) (4.4)

V=HMAC(K, V) (4.5)

V giriş verisinin ve K anahtarının sıfırlanması tamamlanmaktadır[20].

Sıfırlama işlemi tamamlandıktan sonra ihtiyaç duyulan rastgele sayılar ihtiyaç duyulan uzunluğa erişinceye kadar tekrarlı olarak Denklem (4.6)'nın sonucu; V=HMAC(K, V) (4.6)

art arda eklenerek üretilmektedir[20].

Rastgele sayı üretimi tamamlandıktan sonra tekrar üretim yapılmadan önce K anahtarı ve V giriş verisi sırasıyla Denklem (4.7) ve (4.8)'de gösterildiği gibi;

K=HMAC(K, V|0x00) (4.7)

V=HMAC(K, V) (4.8)

sıfırlanarak kullanılmaktadır[20].

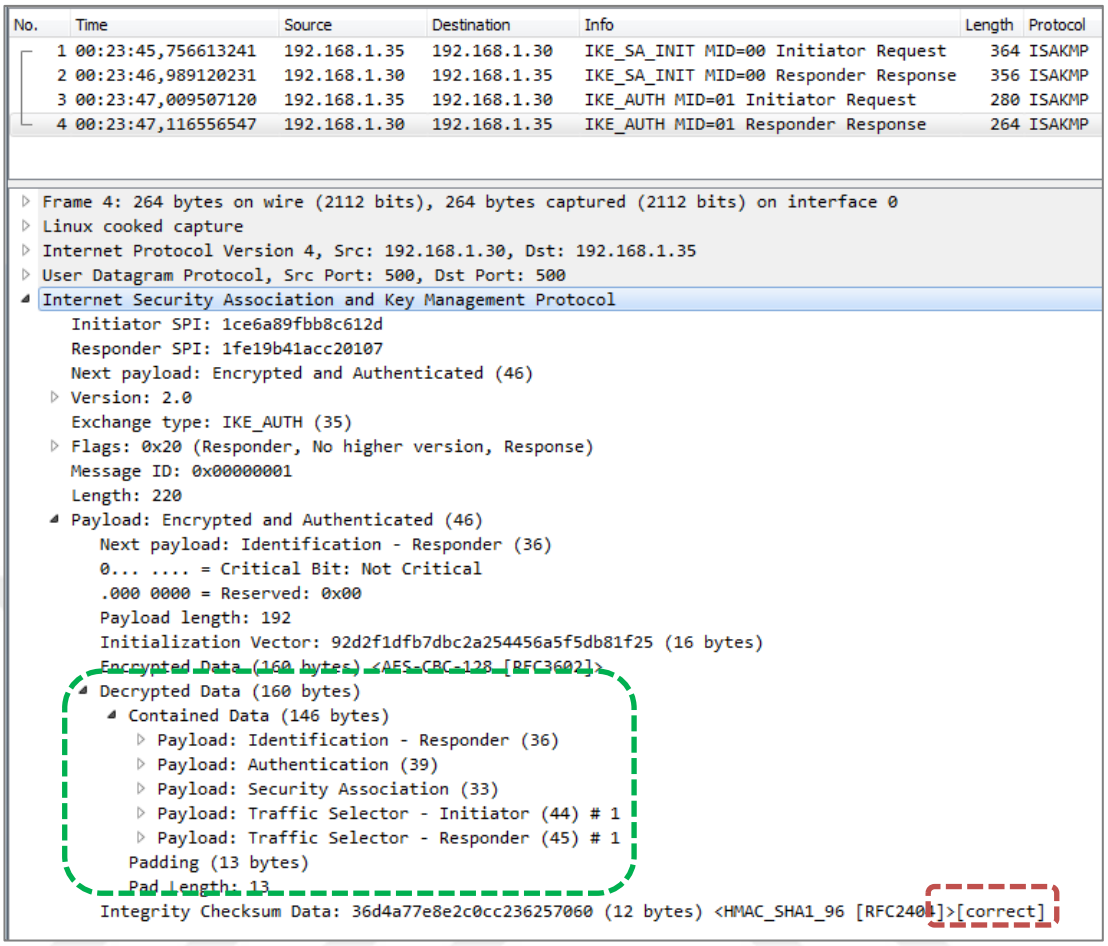
Tasarlanan rastgele sayı üretecinde rastgele sayı üretimi tamamlandıktan sonra yukarıdaki formüllerde belirtilen sıfırlama yöntemi yerine X9.82 yönteminin en başından başlanarak entropi değeri tekrar kullanılarak rastgele sayı üreteci sıfırlanmıştır.

Tasarlanan rastgele sayı üreteci IKEv2 protokolünde SPI (Security Payload Index) değerlerinin üretiminde, Nonce yükünün üretiminde ve Diffie-Hellman anahtar değişim protokolünde gizli anahtar üretimi için kullanılmıştır. Bu sayılar rastgele üretilerek şifreleme algoritmaları için üretilecek anahtarların tahmin edilememesi ve önceden tanımlı anahtar kümeleri denenerek anahtarların tahmin edilememesi amaçlanmaktadır [11].

5. TASARLANAN GÜVENLİ AĞ GEÇİDİNİN TEST EDİLMESİ

Tasarlanan güvenli ağ geçidi ile test bilgisayarı olarak kullanılan Ubuntu işletim sistemi ile çalışan bir bilgisayar arasında IKE SA ve IPsec SA kurularak test edilmiştir. Ubuntu işletim sistemli bilgisayarın IKE ve IPsec protokollerini çalıştırabilmesi için açık kaynak kodlu strongSwan [25] yazılımı kullanılmıştır. Oluşturulan test ortamında Ubuntu bilgisayarda wireshark kullanılarak tasarlanan sistem ve bilgisayar arasındaki IKE SA kurulumu sırasında gerçekleştirilen haberleşme yakalanarak incelenmiştir. Ubuntu bilgisayar ile IPsec tünelinden ICMP talebi gönderilerek tasarlanan sistemin bu talep mesajına IPsec tüneli içinden cevap mesajı gönderdiği görülmüştür. IPsec ve IKE tünellerinden yapılan şifreli haberleşmenin çözülerek şifreleme işleminin doğru yapılıp yapılmadığının kontrolü ve paketler için üretilen kaynak ve bütünlük koruma verisinin doğru üretilip üretilmediğinin kontrol için IKE SA ve IPsec SA kurulumu gerçekleştirildikten sonra bu güvenlik oturumlarına ait anahtarlar tasarlanan sistemin seri portundan gönderilecek değişiklik yapılmıştır.

Wireshark programına seri porttan alınan anahtarlar verilerek yakalanan paketlerin şifreleri çözülüp ve kaynak ve bütünlük doğrulama verilerinin kontrolü sağlanmıştır. Şekil 5.1'de Ubuntu bilgisayarda koşan strongSwan yazılımı ile tasarlanan sistem arasında IKE SA kurulurken gerçekleştirilen mesaj değişimleri gösterilmiştir. Şekil 5.1'de tasarlanan güvenli ağ geçidi tarafından şifreli olarak gönderilen IKE_AUTH mesajının SK_{er} anahtarı kullanılarak wireshark tarafından çözülmüş hali yeşil kesikli çizgi içinde gösterilmiştir. Tasarlanan güvenli ağ geçidi tarafından üretilen kaynak ve bütünlük doğrulama verisi SK_{ar} anahtarı kullanılarak wireshark programı tarafından tekrar hesaplanıp doğruluğu teyit edilmiştir. Paket ile gönderilen kaynak ve bütünlük doğrulama verisinin doğruluğunun wireshark programında test edilmesi Şekil 5.1'de turuncu kesikli çizgi içinde gösterilmiştir.



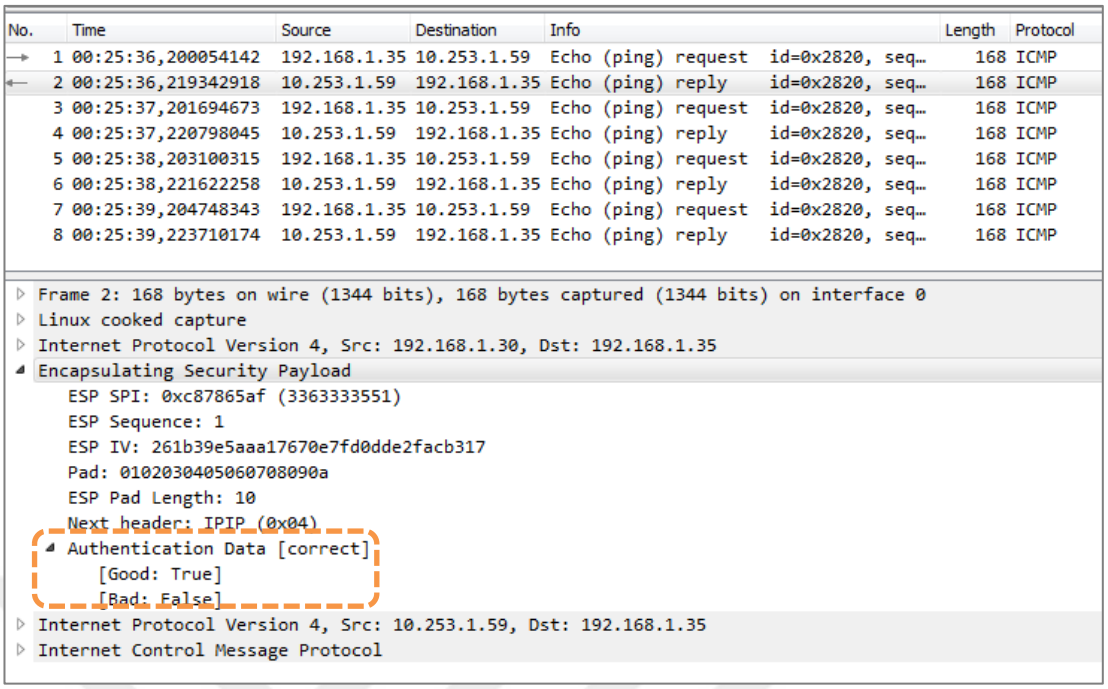
Şekil 5.1. IKE Tünelindeki Haberleşmenin İncelenmesi

Şekil 5.1'de gösterilen IKE_SA_INIT mesaj değişimi sırasında 1024 bit uzunluğunda DH grubu kullanılmıştır. IKE_SA_INIT mesaj değişimi sırasında DH anahtar değişimi için gerekli matematiksel hesaplamaların CPU tarafından hesaplanması uzun sürdüğü için 1024 bit uzunluğunda DH grubu kullanıldığında tasarlanan güvenli ağ geçidinin IKE_SA_INIT cevap mesajının yaklaşık 1,23 saniyede gönderilebildiği gözlenmiştir. Sistemde 768, 1024, 1536, 2048 ve 4096 bit uzunluğundaki DH grupları kullanılarak testler tekrarlandığında 4096 bit uzunluğundaki DH anahtar değişimi sistemde GMP kütüphanesi için ayrılan RAM alanı yetersiz kaldığı için tamamlanamamıştır. Diğer DH grupları için ortalama IKE_SA_INIT mesaj değişimini tamamlama süreleri Tablo 5.1'de gösterilmiştir.

Tablo 5.1. IKE_SA_INIT Mesaj Değişimi Tamamlama Süreleri

DH Grubu	IKE_SA_INIT Mesaj Değişimi Tamamlama Süresi
768	0,76 Saniye
1024	1,21 Saniye
1536	2,44 Saniye
2048	4,44 Saniye

IKE SA kurulumundan sonra IPsec SA kurulumunun da başarılı bir şekilde gerçekleştirildiğinin test edilmesi amacıyla Ubuntu bilgisayar ile IPsec tünelinden tasarlanan sisteme doğru Ping testi yapılmıştır. Ping uygulaması ile IPsec tünelinden tasarlanan güvenli ağ geçidine bir ICMP paketi atılmaktadır. ICMP paketine tasarlanan sistem tarafından IPsec tüneli içinden cevap paketi gönderilmesi beklenmektedir. Wireshark ile sistemlerin haberleşmesi dinlenerek yapılan testlerde kullanılan bilgisayar ile tasarlanan sistem arasında ping uygulaması sırasında ESP trafiğinin oluştuğu gözlenmiştir. Ubuntu bilgisayar tarafından gönderilen ESP paketlerinin şifrelerinin çözülmesi için CA_SK_{ie}, paketlerin kaynak ve bütünlük doğrulama verisinin sağlaması için CA_SK_{ia} anahtarları wireshark programına girilmiştir. Tasarlanan güvenli ağ geçidi tarafından gönderilen ESP paketlerinin şifrelerinin çözülmesi için CA_SK_{re}, paketlerin kaynak ve bütünlük doğrulama verisinin sağlaması için CA_SK_{ra} anahtarları wireshark programına girilerek IPsec tünelinin testi gerçekleştirilmiştir. Şekil 5.2'de IPsec tünelinden gerçekleştirilen haberleşmenin wireshark tarafından çözülmüş hali gösterilmiştir. ESP paketi ile gönderilen kaynak ve bütünlük doğrulama verisinin doğruluğunun wireshark programında teyit edilmesi Şekil 5.2'de turuncu kesikli çizgi içinde gösterilmiştir. Şekil 5.2'deki çözülmüş ESP paketinde detayları gösterilen ESP başlığının üzerinde dış IP başlığının bulunduğu bu dış IP başlığındaki kaynak IP adresinin tasarlanan sistemin DHCP sunucudan aldığı IP adresi (192.168.1.35) olduğu gözlenmiştir. Gösterilen ESP başlığının altında iç IP başlığı bulunduğu, bu dış IP başlığında bulunan kaynak IP adresinin IP yazılım bloğunda tasarlanan sistemde sabit olarak belirlenen IP adresi (10.253.1.59) olduğu gözlenmiştir.



Şekil 5.2. IPsec Tünelindeki Haberleşmenin İncelenmesi

Tasarlanan sistemin ESP protokolüyle test bilgisayarından ping uygulamasıyla gönderilen ICMP paketlerine ağ gecikmeleri dahil ortalama 18.5ms'de cevap gönderebildiği Şekil 5.2'de gözlenmiştir. Şekil 5.2'de 10.253.1.59 tasarlanan güvenli ağ geçidinin iç IP adresidir. 192.168.1.35 test bilgisayarına ait iç IP adresidir.

Test bilgisayarından ESP protokolü ile 1000 adet 284 bayt boyutunda ICMP paketi beklemesiz olarak gönderildiğinde ise ortalama cevap gönderme süresi 133,71 ms olmaktadır. Şekil 5.3'te gösterilen bu testin sonuçlarına göre test bilgisayarı ile tasarlanan güvenli ağ geçidi arasındaki ortalama 133,71 ms'de toplam 568 bayt veri alışverişi yapılabilmektedir.

```
strongswan@strongswan-VirtualBox:~$ sudo ping -f -c 1000 -s 256 10.253.1.59
PING 10.253.1.59 (10.253.1.59) 256(284) bytes of data.
.....................
--- 10.253.1.59 ping statistics ---
1000 packets transmitted, 979 received, 2% packet loss, time 21022ms
rtt min/avg/max/mdev = 19.280/133.719/167.921/27.725 ms, pipe 9, ipg/ewma 21.043/136.038 ms
```

Şekil 5.3. ICMP ile Ağ Hızı Testi

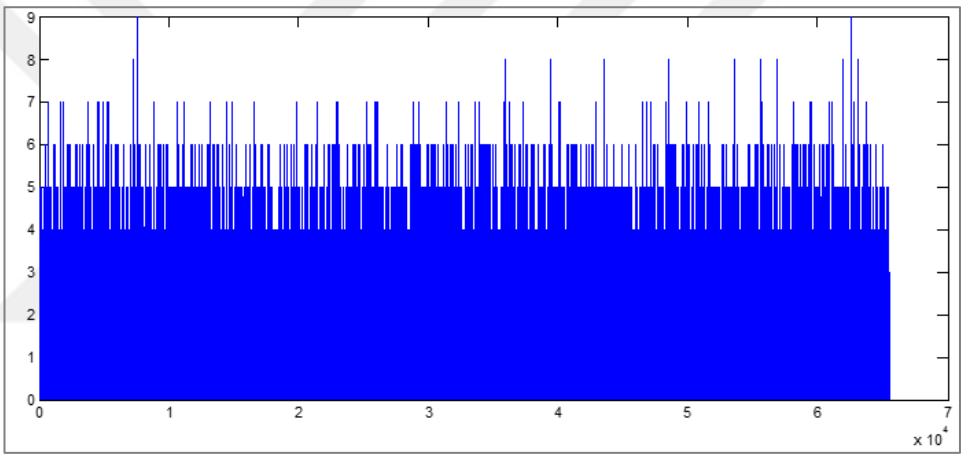
Bu sonuçlara göre tasarlanan güvenli ağ geçidi ile test bilgisayarı arasındaki bant genişliği yaklaşık 33,9 Kb/s olarak Denklem (5.1) ve (5.2)'de gösterildiği gibi;

Toplam Veri Alışverişi=(284×2)×8=4544 Bit (5.1)

$$\text{Bant Genişliği}=\frac{4544\ \text{Bit}}{0{,}13371\ \text{Saniye}}=33{,}9\text{Kb/S} \quad (5.2)$$

hesaplanabilir.

Tasarlanan sistemde çalışan IKEv2 ve IPsec protokolleri için ihtiyaç duyulan rastgele üretilmiş sayıları üretebilmek için sistem tasarımı detaylarında açıklanan bir rastgele sayı üreteci tasarlandı. Tasarlanan rastgele sayı üreteci tarafından 0 ile 65536 arasında değerleri değişen 100000 adet rastgele sayı üretilerek MATLAB çalışma ortamında incelenmiştir. 0 ile 65536 arasındaki sayılardan her birinin kaç kez üretildiğinin gösterildiği Şekil 5.3'de gösterilen grafikte rastgele üretilen sayıların belli bir aralıkta daha fazla ya da daha az üretilmeyip spektruma eşit dağıldığı gözlenmiştir.



Şekil 5.4. Rastgele Sayı Üreteci Çıkış Verisine Ait Histogram

Rastgele sayı üreteci ile üretilen örnek verinin rastgele bir süreç ile üretilip üretilmediklerinin değerlendirilmesi için Wald–Wolfowitz run testi kullanılabilmektedir. Bu testte sıfır hipotezi olarak dizi içindeki değerlerin rastgele olarak üretildiği kabul edilerek kurulan bu hipotez test edilir. Test sırasında hesaplanan p-değeri test için belirlenen anlamlılık seviyesinden [28] daha yüksek çıkarsa sıfır hipotezi reddedilememektedir [27].

Rastgele sayı üretecinden alınan örnekler MATLAB çalışma ortamında $\alpha = 0{,}01$ anlamlılık seviyesinde Wald–Wolfowitz run testinde sokulmuştur. Test sonucunda p-değeri 0,6536 olarak bulunmuştur. Buna göre dizinin rastgele üretildiği hipotezi (sıfır hipotezi) $p > \alpha$ olduğu için kabul edilmiştir [29-30]. Rastgele sayı üretecinden alınan örneklerin ortalaması 32780, standart sapması 18929, en büyük değeri 65535 ve en

küçük değeri 0 olarak hesaplanmıştır. Örneklerin ortalamasının rastgele sayı üretecinin üretebileceği sayı aralığının ortalaması olan 32767'ye yakın çıkması, standart sapmanın yüksek çıkması ve yapılan istatistiksel testler sonucunda tasarlanan rastgele sayı üretecinin sistemin ihtiyacı olan rastgeleliği karşıladığı sonucuna varılmıştır.

6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında nesnelerin İnterneti yapısında nesnelerin İnternet üzerinden gerçekleştirdikleri haberleşmeyi veri gizliliği ve bütünlüğü anlamında koruma altına alacak bir ağ geçidi tasarımı amaçlanmıştır. Tasarlanan ağ geçidinde IKEv2 ve IPsec protokolleri kullanılarak nesnelerin İnterneti konsepti ile uyumlu bir güvenli ağ geçidi tasarımı yapılmıştır. Tasarlanan sistemin düşük maliyetli ve düşük kaynaklara sahip bir gömülü platformda gerçeklenmiştir.

Yapılan testlerde sistemin başarılı bir şekilde güvenlik oturumlarını kurarak nesnelerin İnterneti ağının İnternet üzerinden yapacağı haberleşmede yüksek seviyeli gizlilik ve veri bütünlüğü sağlayabileceği gösterilmiştir. Ağ geçidinde çalışan kriptografik algoritmaların ihtiyaç duydukları rastgele sayıları üretebilmek için tasarlanan rastgele sayı üretecinin çıktıları istatistiksel olarak incelenerek sistemin ihtiyaç duyduğu rastgeleliği karşılayabildiği değerlendirilmiştir.

Geliştirilen sistemin düşük kaynak ve maliyet gerektiren nesnelerin İnterneti uygulamalarında mahremiyet ve veri bütünlüğü açısından örnek olacağı düşünülmektedir. Tasarlanan sistem için geliştirilen yazılım üzerinde yapılabilecek iyileştirmeler aşağıda listelenmiştir.

- Gerçeklemesi yapılan IKEv2 protokolü IKE SA ve IPsec SA anahtar yenileme işlemini yapabilir hale getirilebilir.
- Diffie-Hellman anahtar değişimi sırasında sistemde yaşanan gecikmeler için iyileştirme çalışmaları yapılabilir.
- GMP kütüphanesi için ayrılan RAM alanı yeterli olmadığı için tasarlanan sistemde 4096-bit uzunluğundaki Diffie-Hellman grubuyla anahtar değişimi tamamlanamamıştır.
- Tasarlanan rastgele sayı üreteci ile ilgili daha ayrıntılı testler yapılarak varsa zayıf noktaları giderilebilir.

- Geliştirilen yazılım ile kullanılan mikrodenetleyicinin RAM'i %93 doluluğa ulaşmıştır. RAM kullanımı ile ilgili iyileştirmeler yapılarak tasarlanan güvenli ağ geçidinde farklı uygulamalar için de yer açılabilir.

KAYNAKLAR

[1] Lavanya M., Natarajan V., Certificate-free Collaborative Key Agreement Based on IKEv2 for IoT, *2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, Chennai India, 27-28 February 2017.

[2] Tiller J. S., The Ipsec Standard, Data Security Management, http://www.ittoday.info/AIMS/DSM/87-10-27.pdf, (Ziyaret tarihi: 08 Haziran 2018).

[3] Kent S., Seo K., Security Architecture for the Internet Protocol, *İnternet Engineering Task Force Magazine*, DOI: 10.17487/RFC4301.

[4] Singh S. P., Bharti V., Singh B. M. K., Johri P., Sharma M., Formation of Security Association Database (SAD) in Internet Protocol Version 6 (IPv6), *2016 International Conference on Computing*, Noida India, 29-30 April 2016.

[5] Atkinson R., Security Architecture for the Internet Protocol, *İnternet Engineering Task Force Magazine,* DOI: 10.17487/RFC1825.

[6] Tiller J. S., Ipsec Key Management, Data Security Management, http://www.ittoday.info/AIMS/DSM/87-10-29.pdf, (Ziyaret tarihi: 08 Haziran 2018).

[7] Kent S., Atkinson R., Security Architecture for the Internet Protocol, *İnternet Engineering Task Force Magazine,* DOI: 10.17487/RFC2401.

[8] Kozierok C. M., *TCP/IP Guide,* No Starch Press. Inc., San Francisco, 2005.

[9] Li M., Policy-based IPsec Management, *IEEE Network*, 2003, **17**(6), 36-43.

[10] Kaufman C., Internet Key Exchange (IKEv2) Protocol, *İnternet Engineering Task Force Magazine*, DOI: 10.17487/RFC4306.

[11] Kaufman C., Hoffman P., Nir Y., Eronen P., Kivinen T., Internet Key Exchange Protocol Version 2 (IKEv2), *İnternet Engineering Task Force Magazine,* DOI: 10.17487/RFC7296.

[12] Hellman M. E., Diffie B. W., Merkle R. C., Cryptographic Apparatus and Method, 1977, US4200770A, Google Patents.

[13] Palmgren K., Diffie-Hellman Key Exchange- A Non-Mathematician's Explanation, CISSP, http://academic.regis.edu/cias/ia/palmgren_-_diffie-hellman_key_exchange.pdf, (Ziyaret Tarihi: 08 Haziran 2018).

[14] Boer B., Diffie-Hellman is as Strong as Discrete Log for Certain Primes, *Advances in Cryptology — Crypto '88*, New York, 1988.

[15] Adrian D., Bhargavan K., Durumeric Z., Gaudry P., Green M., Halderman J. A., Heninger N., Springall D., Thome E., Valenta L., Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice, *22nd ACM SIGSAC Conference on Computer and Communications Security*, United States of America, 2015.

[16] Orman H., The OAKLEY Key Determination Protocol, *İnternet Engineering Task Force Magazine*, DOI: 10.17487/RFC2412.

[17] Kivinen T., Kojo M., More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), *İnternet Engineering Task Force Magazine*, DOI: 10.17487/RFC3526.

[18] De Rubertis A., Mainetti L., Mighali V., Patrono L., Sergi I., Stefanizzi M. L., Pascali S., Performance Evaluation of End-To-End Security Protocols in an Internet of Things, *Software, Telecommunications and Computer Networks (SoftCOM)*, Croatia, 18-20 September 2013.

[19] Kothmayr T., Schmitt C., Hu W., Brunig M., Carle G., A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication, *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference*, United States of America, 2012.

[20] Eastlake D., Schiller J., Crokcer S., Randomness Requirements for Security, *İnternet Engineering Task Force Magazine*, DOI: 10.17487/RFC4086.

[21] GNU MP, https://gmplib.org/manual/, (Ziyaret Tarihi: 15 Şubat 2018).

[22] Dunkels A., lwIP- A Lightweight TCP/IP Stack – Summary, http://savannah.nongnu.org/projects/lwip/, (Ziyaret tarihi: 08 Haziran 2018).

[23] Weiss A. R., Dhrystone Benchmark- History, Analysis, Scores and Recommendations, http://www.johnloomis.org/NiosII/dhrystone/ECLDhrystone WhitePaper.pdf, (Ziyaret tarihi: 08 Haziran 2018).

[24] Eastlake D., Crocker S., Schiller J., Randomness Recommendations for Security, *İnternet Engineering Task Force Magazine*, DOI: 10.17487/RFC1750.

[25] StrongSwan – About, https://www.strongswan.org/, (Ziyare tarihi: 08 Haziran 2018)

[26] ARM Cortex M4F Based MCU TM4C1294 Connected Launchpad Evaluation Kit, http://www.ti.com/tool/EK-TM4C1294XL, (Ziyaret tarihi: 08 Haziran 2018).

[27] Wolfowitz W., Runs Test, https://en.wikipedia.org/wiki/Wald%E2%80%93 Wolfowitz_runs_test, (Ziyaret tarihi: 12 Haziran 2018).

[28] What Are Tests for Significance, https://web.csulb.edu/~msaintg/ ppa696/696stsig.htm, (Ziyaret tarihi: 12 Haziran 2018).

[29] Runs Test for Detecting Non-randomness, https://www.itl.nist.gov/div898/ handbook/eda/section3/eda35d.htm, (Ziyaret tarihi: 12 Haziran 2018).

[30] Run Test for Randomness, https://www.mathworks.com/help/stats/runstest. html, (Ziyaret tarihi: 12 Haziran 2018).

KİŞİSEL YAYINLAR VE ESERLER

[1] Gökbayrak A.B., **Divarcı S.**, Urhan O., Ev Otomasyonu Uygulamaları için Kablosuz Algılayıcı Ağı Ağgeçidi Tasarımı, *22. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Trabzon, Türkiye, 23-25 Nisan 2014.

[2] **Divarcı S.**, Demir S., Urhan O., Bilgisayarla Görü Yaklaşımı ile Hız Limit Asistanı, *24. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, Zonguldak, Türkiye, 16-19 Mayıs 2016.

[3] **Divarcı S.**, Urhan O., IoT Sistemlerde Ağ Katmanı Güvenliği İçin Güvenli Ağ Geçidi, *26. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı*, İzmir, Türkiye, 2-5 Mayıs 2018.

ÖZGEÇMİŞ

1993 yılında Muğla'da doğan Sinan Divarcı, ilköğretimini Silkar Günlükbaşı İlköğretim Okulu'nda, lise öğretimini Fethiye Ömer Özyer Anadolu Öğretmen Lisesi'nde tamamladı. Lisans derecesini 2011 yılında girdiği Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümünden 2015 yılında aldı. 2016 Haziran ayından beri Netaş'ta yazılım tasarım mühendisi olarak çalışmaktadır.

2015 yılında başladığı Kocaeli Üniversitesi Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Ana Bilim Dalı'ndaki Yüksek Lisans öğreniminden 2018 yılında mezun olma durumundadır.