

İskenderun Teknik Üniversitesi



Fakülte: Mühendislik ve Doğa Bilimleri Fakültesi

Bölüm: Bilgisayar Mühendisliği

Öğretim üyesi: Doç.Dr. İpek Abasıkeleş Turgut

Ders: Bilgi ve Ağ Güvenliği

- Genel Tanımlar, Saldırıları Tanıma
- Ağ Güvenliği Kavramları
- Kriptoloji – Simetrik ve Public Anahtar
- MAC, Dijital İmza
- Uç Nokta Kimlik Doğrulaması
- Katmanlarda Güvenlik – Her katman için örnek üzerinden
- İşletme Güvenliği (Firewall, IDS vb.)

Ders İçeriği



Doç.Dr.İpek Abasıkeleş-Turgut

**DİKKAT! Bu ders, CISCO
Enterprise Networking, Security,
and Automation v7.0
(ENSA) altındaki «Module 3:
Network Security Concepts»
konusu ders sunumlarını ve
konularını içermektedir.**



İÇERİK

Amaç: Zafiyet, tehdit, girişim kavramları

Başlıklar

Siber Güvenliğin Mevcut Durumu

Tehdit Aktörleri

Malware

Yaygın Ağ Atakları

IP Zafiyet ve Atakları

TCP and UDP Zafiyetleri

IP Servisleri

Ağ Güvenliğindeki Uygulamalar

Kriptoloji Giriş

Etik «Hack» Sözleşmesi

- DİKKAT!

Siber GüvenliĐin Mevcut Durumu



Doç.Dr.İpek Abasıkeleş-Turgut

Siber Güvenliğin Mevcut Durumu

- Saldırganların araçları gittikçe gelişiyor.
- Güvenlik oldukça önemli bir konu.
- Önemli terimler:

Terimler

VARLIK Assets

ZAFİYET Vulnerability

TEHDİT Threat

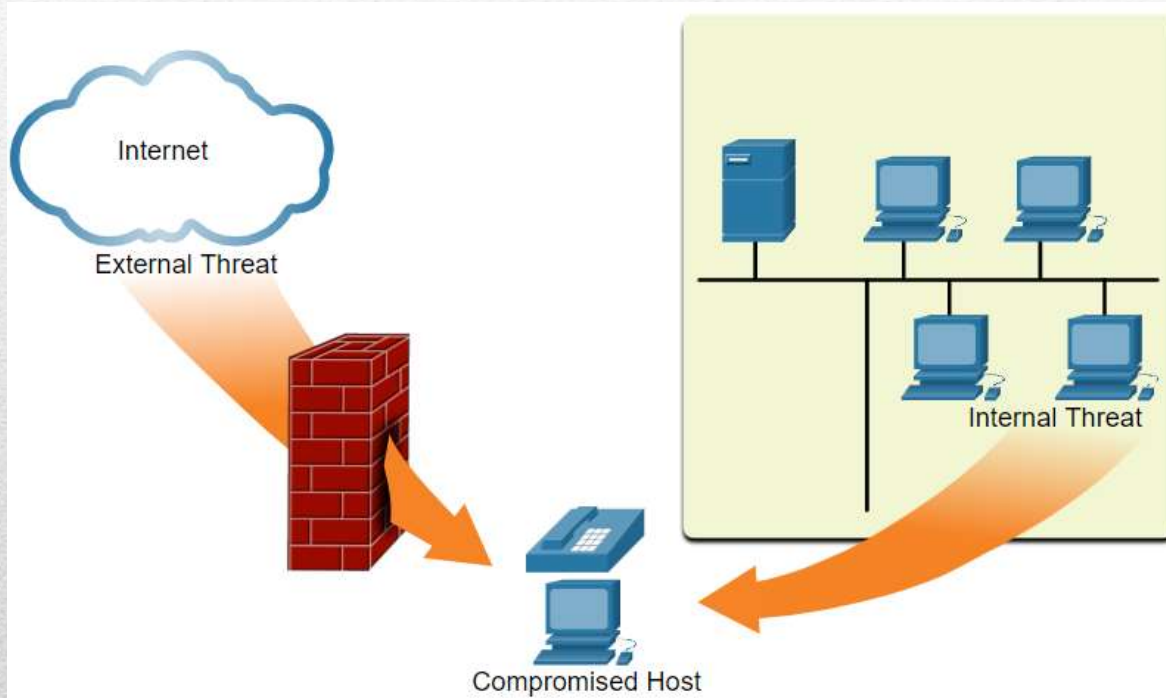
KÖTÜYE KULLANIM Exploit

AZALTMA Mitigation

Risk

SALDIRI VEKTÖRLERİ

- İÇ VE DIŞ ATAKLAR



VERİ KAYBI

SONUÇLAR:

- İtibar kaybı
- Rekabet avantaj kaybı
- Müşteri kaybı
- Gelir kaybı
- Yasal işlemler
- Tarafları bilgilendirme maliyeti

Data Loss Prevention (DLP)- Veri Kaybı Önleme kontrolleri uygulanmalı

Veri Kayıp Vektörleri

Vektörler

Email/Sosyal Ağ

Şifrelenmemiş Aygıtlar

Bulut Depolama Aygıtları

Çıkarılabilir Medya

Basılı Kopya

Uygun olmayan erişim kontrolü

Tehdit Aktörleri



Doç.Dr.İpek Abasıkeleş-Turgut

HACKER : BİLGİSAYAR KORSANI

Genel ifade: Hacker

Tipleri

White Hat Hackers – Beyaz Şapkalılar

Gray Hat Hackers – Gri Şapkalılar

Black Hat Hackers – Siyah Şapkalılar

Hacker Evrimi

Modern hacker terimleri

Terimler

Script Kiddies

Vulnerability Broker – Güvenlik Açığı Arayıcısı

Hacktivists

Cyber criminals – Siber Suçlular

State-Sponsored – Devlet Destekliler

Siber Suçlular

Tahmini milyarlarca dolar çalma

Yeraltı ekonomisi

Saldırı araç kiti, kod vb. alım-satım

Özel bilgi alım satım

Hem küçük işletme hem büyük endüstriler

Hacktivistler

Anonymous ve the Syrian Electronic Army.

Çoğu iyi organize değil

Basit ve ücretsiz araç kullanımı

Devlet Destekliler

Keşfedilmemiş açıkları kullanıp özel saldırılar yaparlar

Örn Stuxnet

Tehdit Aktörlerinin Araçları



Doç.Dr.İpek Abasıkeleş-Turgut

Gittikçe daha da evriliyor.
Daha az teknik bilgi gerektiriyor.

Araçlar

Yaygın penetrasyon (sızma) test araçları

Araç

Password Crackers – Şifre Kırıcı

John the Ripper, Ophcrack, L0phtCrack, THC Hydra, Rainbow Crack, Medusa.

Wireless Hacking Tools – Kablosuz Hacking Araçları - include Aircrack-ng, Kismet, InSSIDer, KisMAC, Firesheep, ViStumbler.

Network Scanning and Hacking Tools – Ağ Tarama ve Hacking Araçları - Nmap, SuperScan, Angry IP Scanner, and NetScanTools

Packet Crafting Tools – Paket Üretme Araçları - Hping, Scapy, Socat, Yersinia, Netcat, Nping, and Nemesis

Packet Sniffers – Paket Dinleyiciler - Wireshark, Tcpdump, Ettercap, Dsniff, EtherApe, Paros, Fiddler, Ratproxy, and TLSstrip

Araç

Rootkit Algılayıcı - AIDE, Netfilter, and PF: OpenBSD Packet Filter

Arama Zafiyetlerine karşı Fuzzer - Skipfish, Wapiti, and W3af.

Adli araçlar - Sleuth Kit, Helix, Maltego, and Encase

Debugger - GDB, WinDbg, IDA Pro, and Immunity Debugger

Hack işletim sistemleri - Kali Linux, BackBox Linux

Encryption Tools - Şifreleme araçları - VeraCrypt, CipherShed, OpenSSH, OpenTLS, Tor, OpenVPN, and Stunnel

Vulnerability Exploitation Tools - Zafiyet sömürü araçları - Metasploit, Core Impact, Sqlmap, Social Engineer Toolkit, and Netsparker

Vulnerability Scanners - Zafiyet tarayıcılar - Nipper, Core Impact, Nessus, SAINT, and OpenVAS

Saldırı Türleri

Türler

Eavesdropping – Gizlice dinleme

Data Modification – Veri modifiye etme

IP Address Spoofing – IP adres kandırması

Password-Based – Şifre tabanlı

Denial of Service Attack - DoS

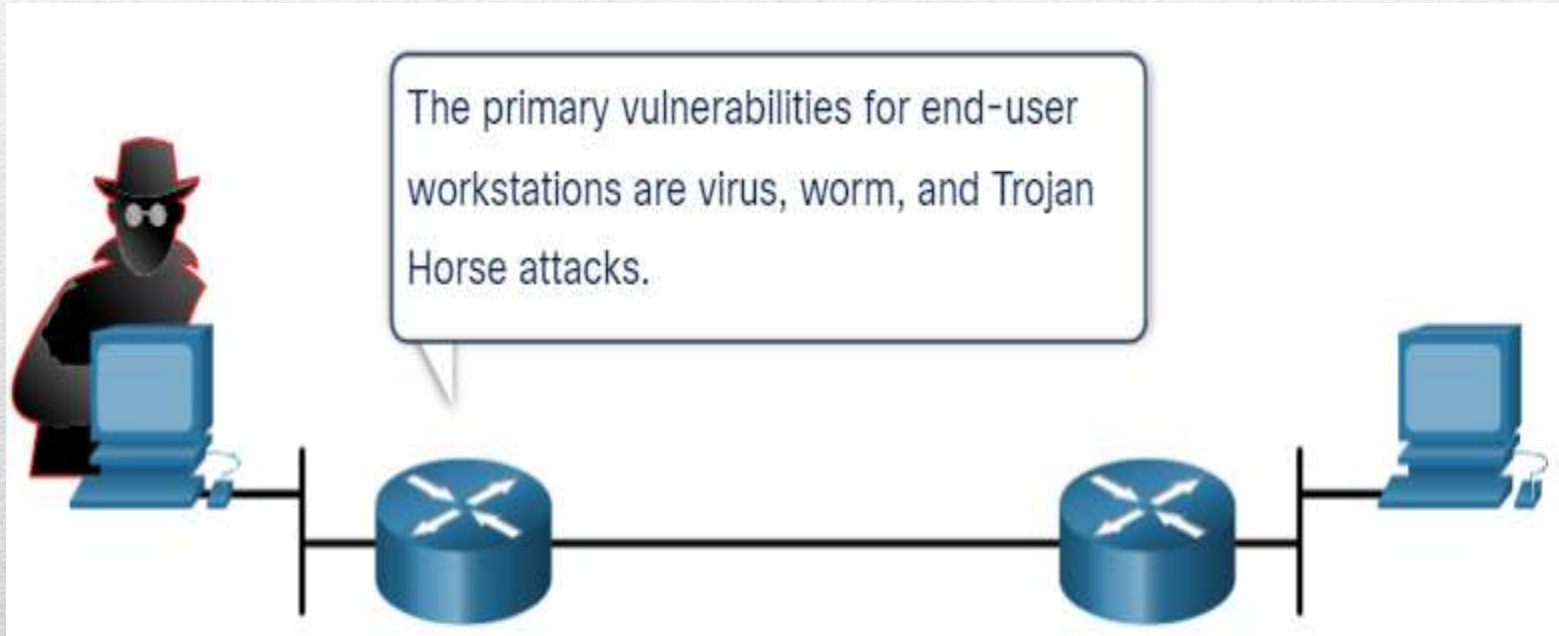
Man-in-the-Middle – Aradaki adam

Compromised-Key – Ele geçirilmiş anahtar

Sniffer - Dinleme

Malware (Kötü Amaçlı Yazılım)

- Uç sistemler malware ataklarına açık
- Uç sistem iş istasyonları için temel zafiyetler virüsler, solucanlar ve Trojan atları



Virüs

- En yaygın ve ilk
- İnsan müdahalesi lazım
- Gizler -> açılınca çalışır
- Yarattığı sorunlar:
 - Silme & karıştırma
 - Boot sorunları
 - Hassas bilgi
 - Email
 - Uykuda bekleme

Modern Virüsler

Virüs Türleri

Boot sector virus

Firmware virus – aygıt yazılım

Macro virus

Program virus

Script virus

Trojan Atları

- Yararlı görünen zararlı programlar

Türleri

Remote-access – Uzak erişim

Data-sending

Destructive –Zarar Verici

Proxy

FTP

Security software disabler – Devre dışı bırakma

Denial of Service (DoS)

Keylogger

Diğer Malware Türleri

Diğer Malware Türleri

Adware – reklam yazılımı

Ransomware – fidye yazılımı

Rootkit – kök setleri

Spyware – Casus yazılım

Worm - Solucan

Yaygın Ağ Saldırıları

- 3 tür:
- Keşif Saldırıları – Reconnaissance
- Erişim Saldırıları- Access
- DoS Saldırıları

Keşif Saldırıları

- Bilgi toplama için. Diğerlerinden önce

Kullanılan Teknikler

Hedefin bilgi sorgusunu gerçekleştirme

Hedef ağda ping taraması başlatma

Etkin IP adresleri için bir bağlantı noktası taraması başlatma

Güvenlik açığı tarayıcılarını çalıştırma

Sömürü araçlarını çalıştırma

Erişim Atakları

- Parola Saldırıları
- Spoofing – Sahtecilik
- Trust exploitations - Güven istismarı
- Port redirections - Bağlantı noktası yönlendirmeleri
- Man-in-the-middle attacks - Ortadaki adam saldırıları
- Buffer overflow attacks - Tampon bellek taşması saldırıları

Sosyal Mühendislik Saldırıları - Social Engineering Attacks

Saldırılar

Pretexting

Phishing - Yemleme

Spear phishing – Zıpkınla avlama

Spam

Something for Something

Baiting – Rahatsız etme

Impersonation – Kimliğe bürünme

Tailgating – öndeki arabanın dibinden gitme

Shoulder surfing – Gizlice dinleme

Dumpster diving – Çöp karıştırma

DoS ve DDoS Atakları

- Hizmet kesintisi
- 2 tür:
 - Aşırı Trafik Miktarı
 - Kötü Amaçlı Formatlanmış Paketler

IP Zafiyetleri ve Tehditler

- IP paketin kaynağını doğrulamaz -> Sahte IP kullanmak mümkün

Teknik

ICMP saldırıları

Amplifikasyon (Genişletme) ve reflection (yansıtma) saldırıları

Adres sahteciliği saldırıları

Ortadaki adam saldırısı (MITM)

Session hijacking - Oturum çalma

ICMP Saldırıları

- Keşif ve tarama saldırıları için ICMP kullanılır.
- Not: IPv4 için ICMP (ICMPv4) ve IPv6 için ICMP (ICMPv6) benzer saldırı türlerine karşı hassas
- ACL – Access control list – erişim kontrol listesi
- IDS – intrusion detection system – izinsiz giriş tespit sistemleri

Korsanlar tarafından kullanılan ICMP Mesajları

ICMP echo request ve echo reply

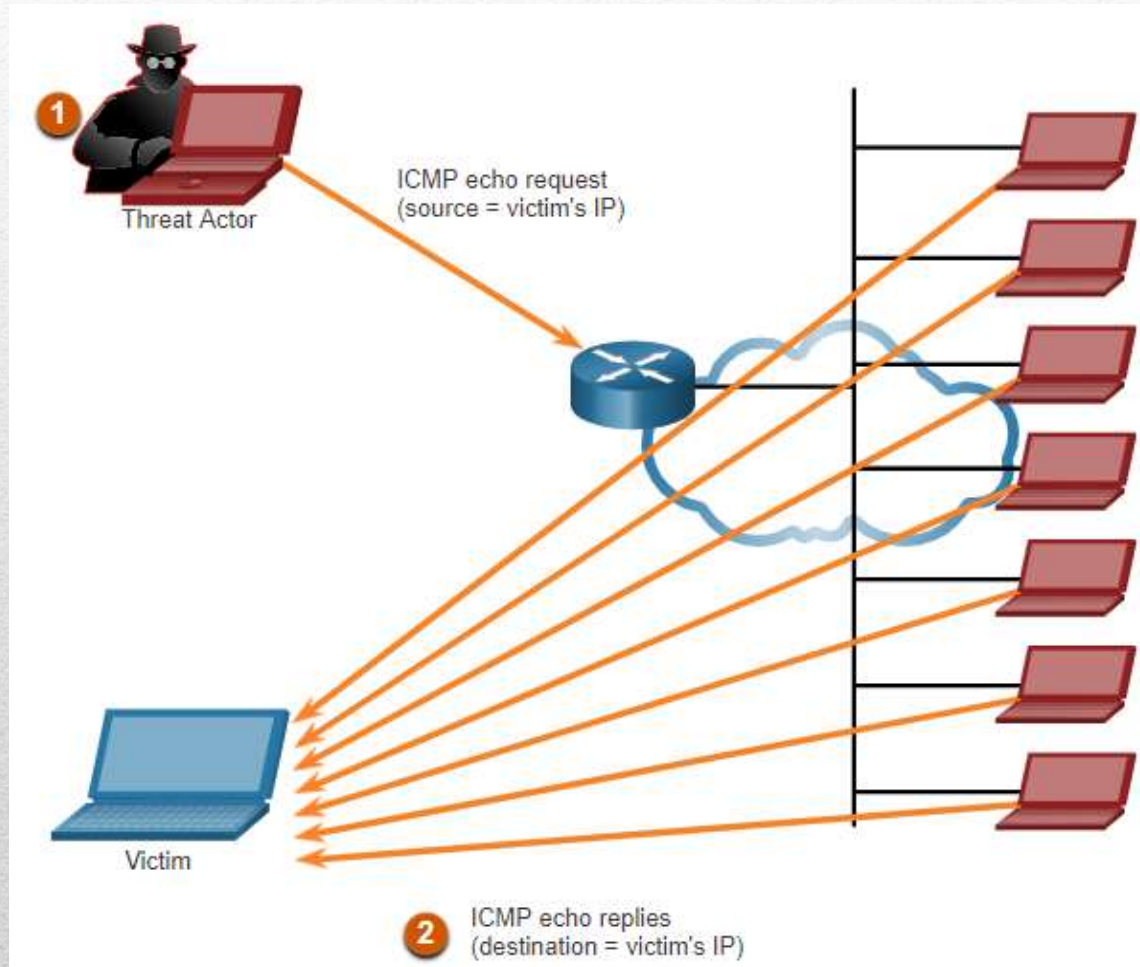
ICMP unreachable

ICMP mask reply

ICMP redirects

ICMP router discovery

Büyütme ve Yansıtma Atakları (Amplification and Reflection Attacks)

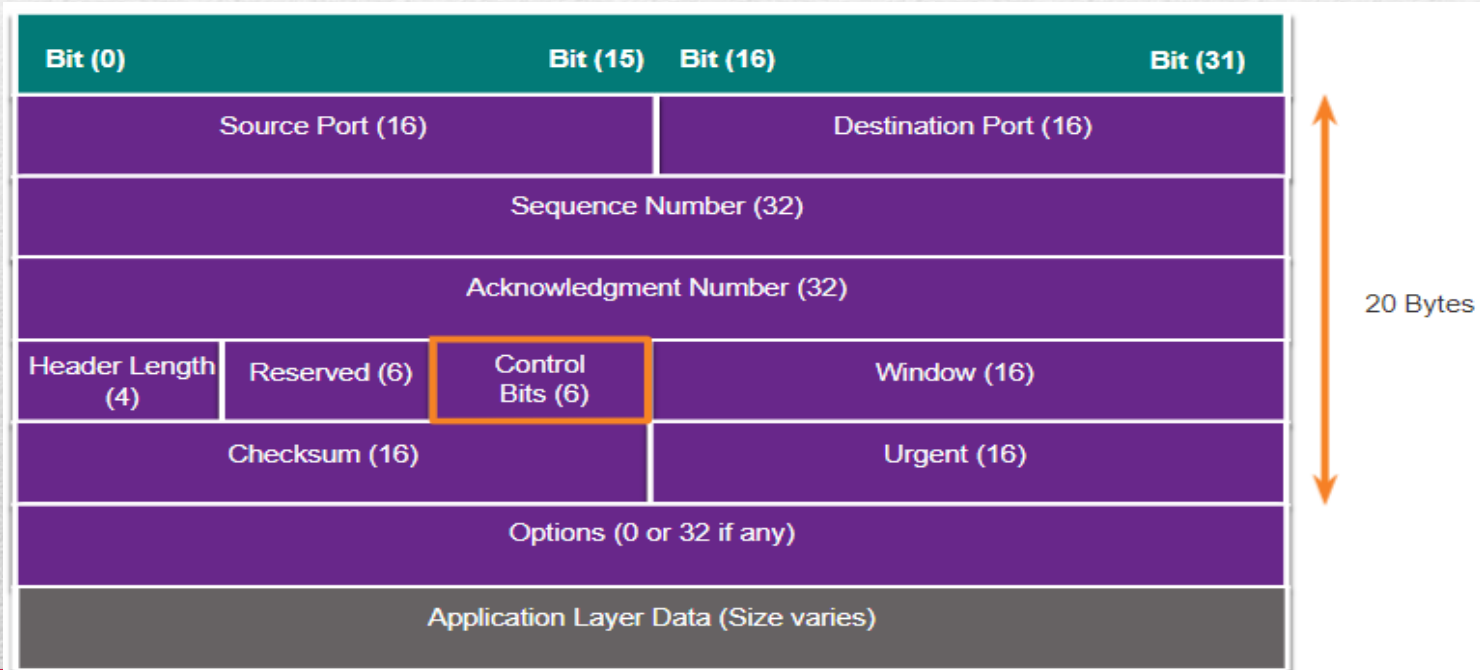


Sahte Adres Atağı (Address Spoofing Attacks)

- Yanlış kaynak IP
- Smurf vb. dahil edilir.
- Blind (kör) & non-blind (kör olmayan)
- MAC adres sahtekarlığı

TCP ve UDP Zafiyetleri

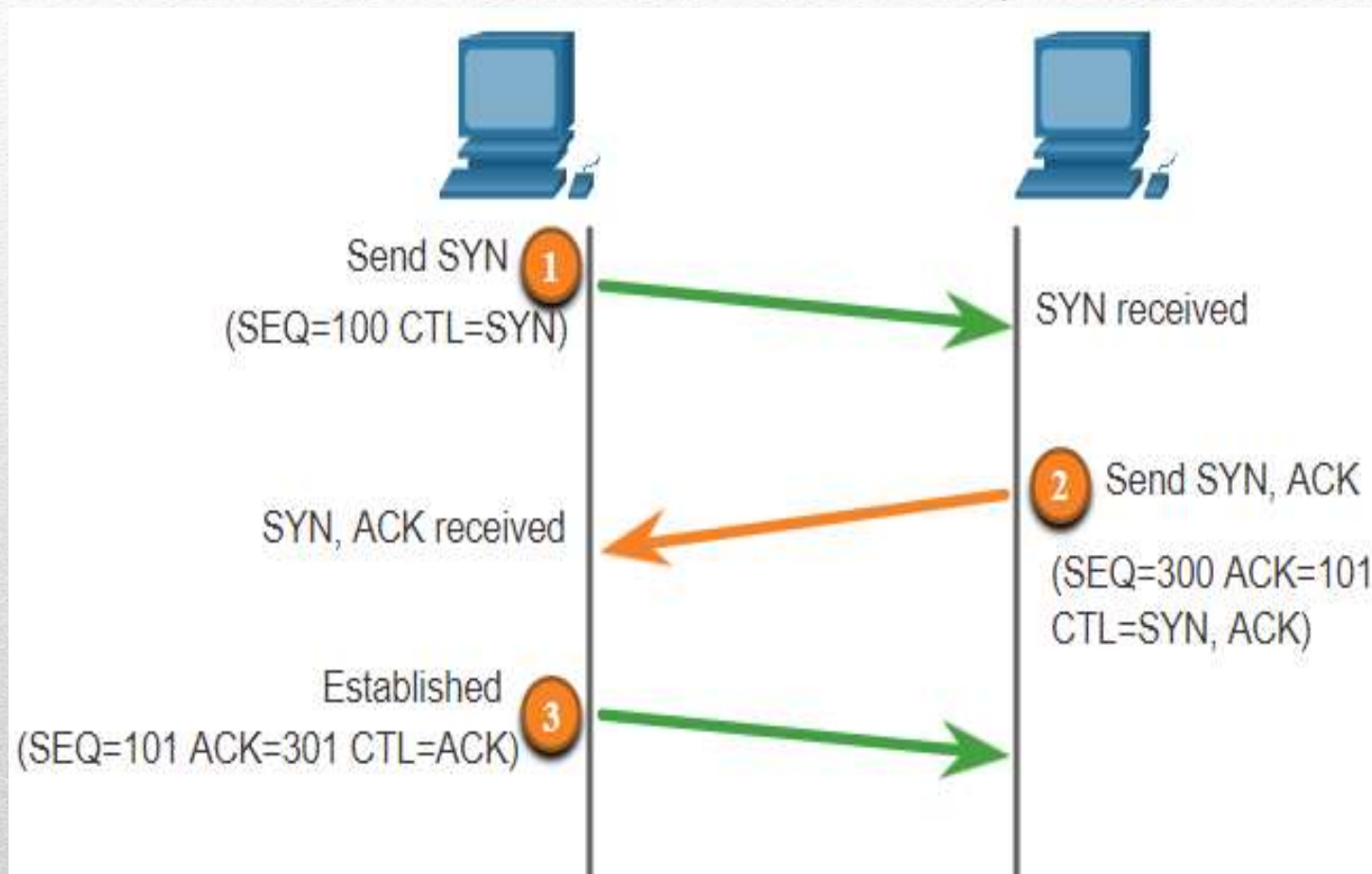
- URG - Urgent pointer field significant – acil işaretçi alanı önemli
- ACK - Acknowledgment field significant
- PSH - Push function – itme işlemi
- RST- Reset the connection – bağlantıyı sıfırla
- SYN - Synchronize sequence numbers – sıra no.larını senkronize et
- FIN - No more data from sender – göndericinin daha fazla verisi yok



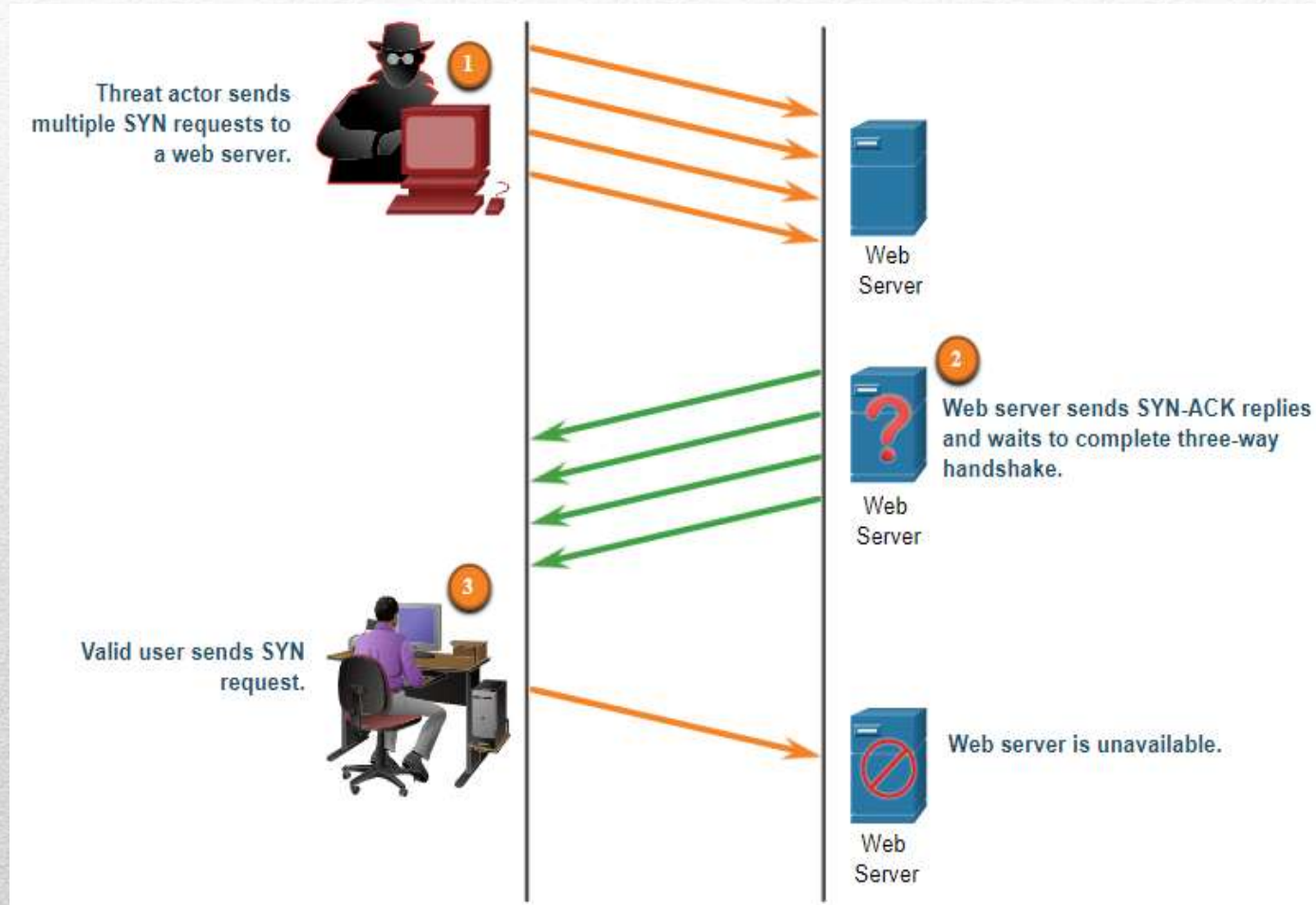
TCP'nin kullandığı servisler

- Güvenilir transfer
- Akış kontrolü (flow control)
- Durum bilgili (stateful) iletişim

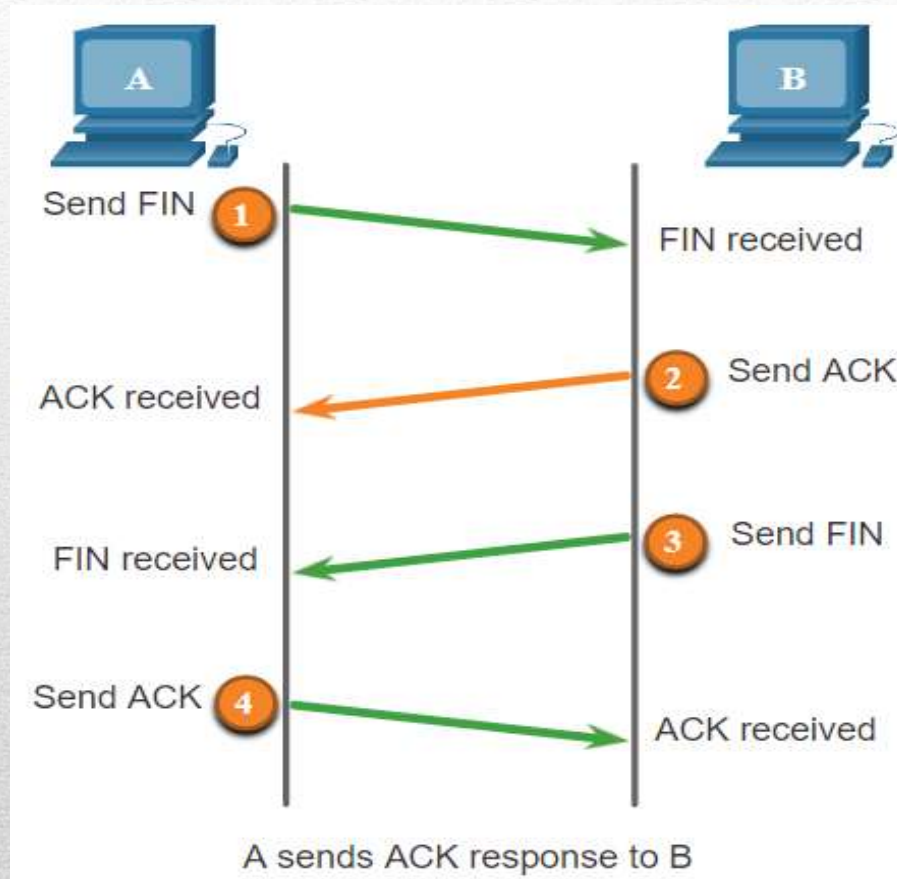
TCP Bağlantısı Oluşturma



TCP SYN Flood Atığı



TCP Oturum Sonlandırma

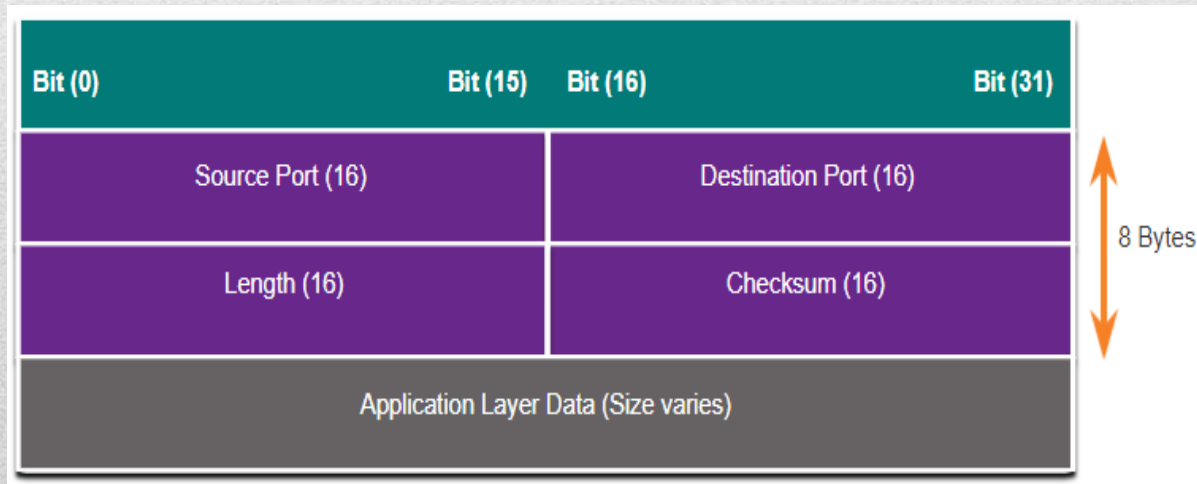


TCP Zafiyetleri

- TCP Sıfırlama - Reset saldırısı
- TCP sesion hijacking – ele geçirme

UDP

- DNS, TFTP, NFS, SNMP vb. tarafından
- Medya akışı, VoIP – gerçek zamanlı
- Bağlantısız
- Yük az
- Güvenilirlik mekanizmaları yok
- Şifreleme yok



UDP Flood Atakları

- UDP Unicorn veya Low Orbit Ion Cannon ile
- ICMP port unreachable
- DoS ile benzer sonuç

IP Servisleri

- ARP Zafiyeti:
- karşılıksız ARP" – “gratuitous ARP
- Herhangi bir aygıt, herhangi bir adrese sahip olduğunu iddia edebilir
- MITM -> ARP ön bellek zehirlenmesi

ARP Cache Poisoning

- çeşitli MITM atakları için
- Örnek
- 1 – PC-A, default gateway MAC talebi -> ARP req.broadcast – 192.168.10.1 için
- R1, PC-A'yı kaydeder ve ARP reply ile cevap
- PC-A, R1'i kaydeder.
- Saldırgan – hedef IP – kendi MAC ile sahte ARP reply
- Pasif veya aktif

DNS Saldırıları

- Kaynak ada karşılık IP adresi sunar
- Otomatik
- Sorgu (query) ve yanıt (response) formatı
- RRs – resource records ile DNS yanıt tipi
- DNS saldırıları:
 - 1 - DNS open resolver attacks - açık çözümleyici
 - 2 - DNS stealth attacks - gizlilik
 - 3 - DNS domain shadowing attacks – alan adı gölgeleme
 - 4 - DNS tunneling attacks – tünelleme

1 – DNS açık çözümleyici atağı

DNS Resolver Hassasiyetleri

DNS cache poisoning attacks – DNS ön bellek zehirleme saldırıları

DNS amplification and reflection attacks – DNS büyütme ve yansıtma saldırıları

DNS resource utilization attacks – DNS kaynak kullanım atakları

2 - DNS gizlilik saldırıları

DNS Gizlilik Teknikleri

Fast Flux – Hızlı Akış

Double IP Flux

Domain Generation Algorithms – Alan adı üretime algoritmaları

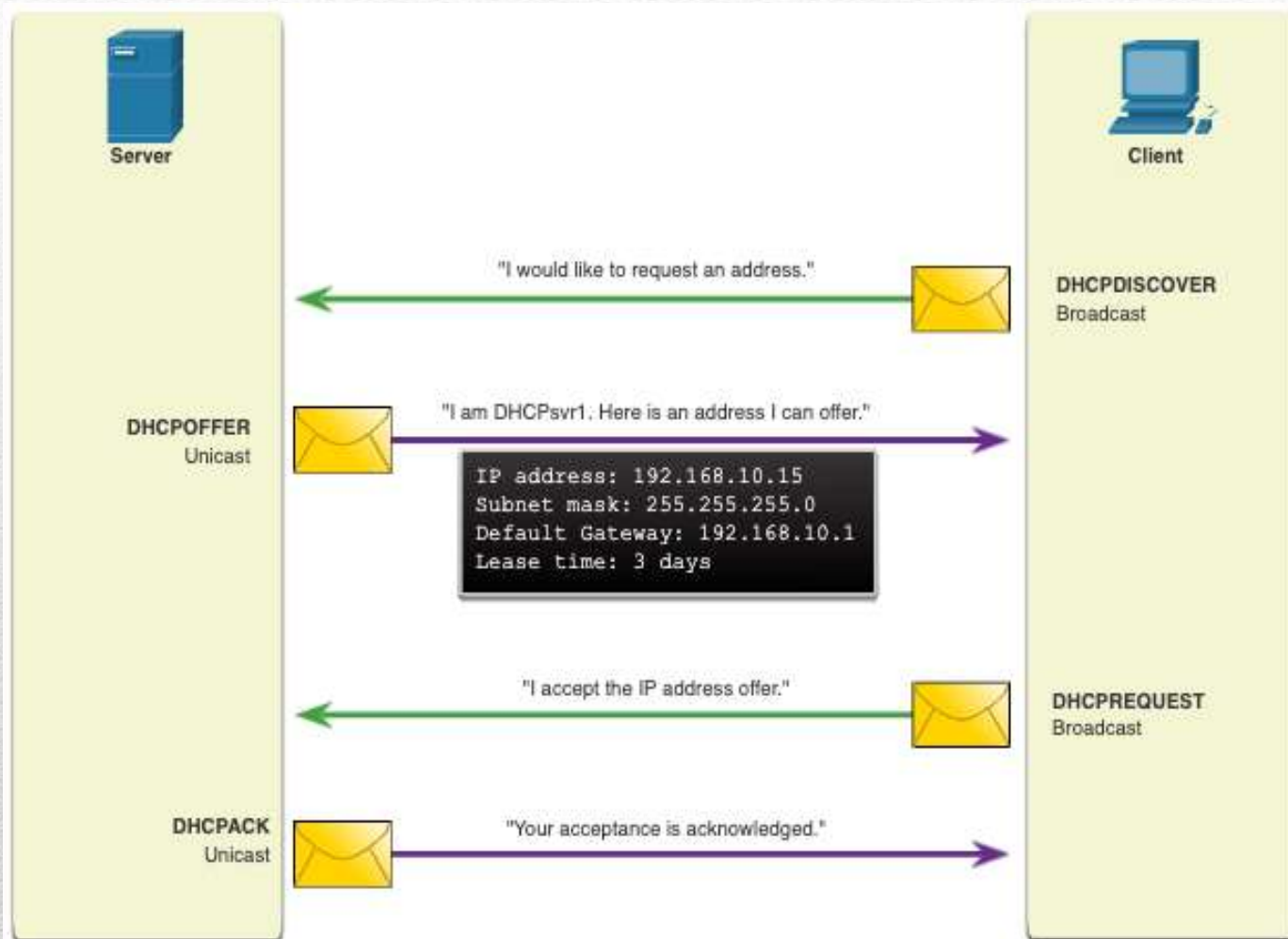
3- DNS Alan Gölgeleme Atakları

- Kimlik bilgisi toplama -> alt domain yaratma -> saldırıda kullanma

4 – DNS Tünelleme Atağı

- Non-DNS trafiği
- CnC komutları:
- 1 – chunk – parçalar
- 2 – parçalar –alt seviye alan adı etiketine
- 3 – ISS DNS sunucuları
- 4 – saldırganın yetkili ad sunucusu
- 5 – proses devam
- 6 – CnC komutları
- 7 – malware
- Engelleme?

DHCP



DHCP Spoofing Atağı

- Yanlış varsayılan ağ geçici
- Yanlış DNS server
- Yanlış IP
- Örnek:
 - 1 – İstemci DHCP discover -> hem yasal hem sahte
 - 2 – İkisi de yanıtlar, ilki seçilir
 - 3 – Sahteyi aldı – yanıt
 - 4 – Yasal olan işlem yapmaz

AĞ GÜVENLİĞİ



Doç.Dr.İpek Abasıkeleş-Turgut

- Güvenli iletişim?
- 3 özelliği sağlamalı:
 - *Confidentiality* Yetkisiz erişimden gizlenmeli – gizlilik. İçeriği sadece gönderici/alıcı bilmeli.
 - *Integrity* Yetkisiz değişimden korunmalı – bozulmamışlık – hata veya kötü niyet
 - *Availability* Gerektiğinde yetkili erişim ile ulaşılabilmeli – uygunluk/erişilebilirlik

Ağ Güvenliği



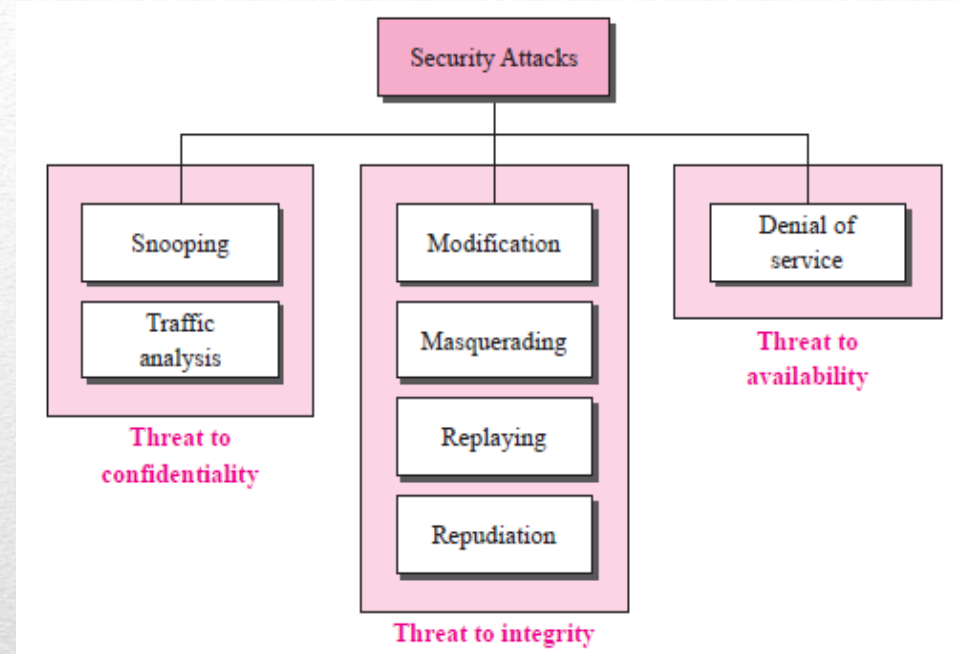
Doç.Dr.İpek Abasıkeleş-Turgut

- End-point authentication - Uç nokta kimlik doğrulaması – hem gönderen hem alan, iletişime dahil olan diğer tarafın kimliğini teyid edebilmeli.
- Operational security – İşletme güvenliği – organizasyonlar, şirketler, üni.ler vb.
- Internet'e bağlı olan tüm birimler ataklara açık.
- Firewall – public ağ ve şirket ağı arasında yer alarak paket akışını kontrol eder,
- IDS – deep packet inspection yaparak ağ yöneticisini uyarır

Ağ Güvenliği



- *Gizliliği tehdit eden ataklar:*
 - Snooping – yetkisiz erişim ile veriyi izleme/dinleme. Örneğin gönderilen dosyanın içinden gizli bilgileri edinme
 - Trafik Analizi – online trafiği izleyerek bilgi edinme.
- *Bütünlüğü tehdit eden ataklar:*
 - Modifikasyon – veriye eriştikten sonra kendine uygun şekilde değiştirme.
 - Masquerading – spoofing - gizlenme/yanıltma – kendisini başkası olarak tanıtmak.
 - Replaying – tekrar etme – kullanıcının paketinin kopyasını alıp daha sonra tekrar etme
 - Repudiation – inkar – biraz farklı. Ya gönderici ya alıcı olmak lazım. Yaptığı işlemi inkar etme, kabul etmeme
- *Erişilebilirliği tehdit eden ataklar:*
 - Denial of Service – DoS atak – çok yaygın. Sistemi ya yavaşlatma ya da tamamen durdurma. Çok yolla yapılabilir. Sunucuya çok fazla istek göndererek ağır yük altında bırakabilir, sunucunun gönderdiği cevabı durdurabilir veya silebilir – kullanıcı sunucunun yanıt vermediğini sanır. Kullanıcı isteklerini durdurabilir – böylece sürekli istek yaparak sisteme yüklenmelerini sağlar.



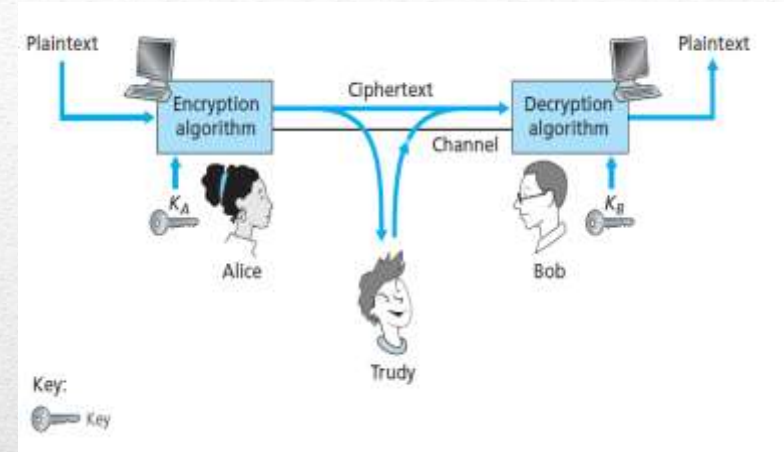
Ataklar

KRİPTOLOJİ



Doç.Dr.İpek Abasıkeleş-Turgut

- Cryptography – Yunanca bir kelime – secret writing
- Eskiden sadece mesajların şifrelenip çözülmesini – encryption/decryption ifade ederken günümüzde 3 farklı mekanizma içeriyor:
 - symmetric-key encoding,
 - asymmetric-key (public key) encoding, ve
 - Hashing
- Terimler: plaintext (cleartext), Encryption alg., ciphertext, key
- Şifreleme yöntemi bilinir – public
- Peki koruma? – anahtar ile
- Plaintext üzerinde algoritma(anahtar) => ciphertext



Kriptoloji

Doç.Dr.İpek Abasıkeleş-Turgut

- Gizlilik – Confidentiality- şifreleme ile sağlanabilir.
- Tarihte Sezar'dan beri var.
- Geleneksel şifreleyiciler – simetrik anahtar kullanır = secret-key şifrelemesi.
- bidirectional – yani iki yönlü iletişimde hem şifreleme hem şifre çözme için aynı anahtar kullanılır.
- Anahtar – şifreleme algoritmasının üzerinde çalışacağı değer
- şifreleme ve çözme algoritmalarını birbirinin tersidir – inverse.
- Encryption: $C = E_k(P)$
- Decryption: $P = D_k(C)$
- $D_k(E_k(x)) = E_k(D_k(x)) = x$

Geleneksel Şifreleme



- Şifreleme algoritmaları public, şifreler gizli.
- Peki şifre nasıl paylaşılacak? – güvenilir başka bir kanaldan - mesela yüzyüze
- Symmetric-key şifreleyicileri 2 gruba ayrılır:
 - substitution (yerine koyma) ,
 - transposition (yer değiştirme)

Geleneksel Şifreleme

- bir sembol yerine başka bir sembol koyma - Caesar cipher (Julius Caesar)
- Mesela A harfini D ile değiştirme gibi.
- Substitution şifreleme de kendi içinde 2 ye ayrılır:
 - Monoalphabetic
 - Polyalphabetic
- Monoalphabetic - textteki karakter, pozisyonundan bağımsız olarak hep aynı karakterle/sembolle değiştirilir.
- Yani plaintext – ciphertext harfleri arasındaki ilişki birebirdir – one-to-one.
- Buna bir örnek additive veya shift cipher.
- Örnek: Alfabe karakterlerini modüle 26 olarak düşünelim - a – 0, z -25 olacak.
- Şifreleme algoritması anahtarı karakterin sayı karşılığına ekleyecek, şifre çözme ise çıkaracak.
- Mesela anahtar 3 ise, a = 0 olduğu için, şifrelemede $0 + 3 = 3 = d$ olacak. Şifre çözülürken ise $3 - 3 = 0$ 'dan a ya geri çevrilecek.

Substitution



- Bu yöntem, exhaustive anahtar taraması kullanan brute-force benzeri ataklara çok açık.
- Anahtar uzayı çok dar – 26 karakter - 26! Olasılık var.
- Dolayısı ile brute-force ile rahatça kırılır.
- Daha iyi bir yöntem – her bir plaintext karakteri ile ilgili ciphertext karakteri arasında mapping yaratmak.
- Örneğin:

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

this message is easy to encrypt but hard to find the key

ICFVQRVVNEFVRNVSIYRGASLHIOJICNHTIYBFGTICRXRS

Monoalphabetic

- istatiksel olarak tipik bir İngilizce metinde e ve t harfleri çok fazla kullanılır - %13 ve %9 oranlarında.
- Veya in, it, the, ion, ing gibi bazı kalıplar da sıklıkla kullanılır.
- Dolayısı ile kodu kırmak kolaylaşır.
- Eğer intruder, mesajın olası içeriği hakkında biraz bilgi sahibi ise, kodu kırmak daha da kolaylaşır.
- Mesajı kırmak için intruder'ın ne bildiği önemli:
- 1 - Ciphertext-only attack – sadece Ciphertext erişimi var. Plaintext ile ilgili herhangi bir bilgiye sahip değil.
- 2 - Known-plaintext attack – bir şekilde içerik hakkında bilgi var - a, l, i, c, e, b, ve o. Harflerinin karşılığını bilir. Yani bazı plaintext-ciphertext çiftini bilir.
- 3 - Chosen-plaintext attack – saldırgan, istediği mesajı şifreleterek, plaintext mesajına ve karşılığındaki ciphertext formuna erişebilir.

Monoalphabetic



- plaintext – ciphertext harfleri arasındaki ilişki one-to-many.
- Örneğin a yerine textin başında ise d, ortasında ise n konulması gibi.
- Avantajı, harf frekansını gizlemesi
 - Böylece tek-harf frekans istatistiği ile kırılmaz.
- Burada ciphertext karakterleri, plaintextteki karakterlerle birlikte bu karakterlerin yerlerine de bağlı
- Anahtar aslında bir alt anahtar stream'i
 - $k = (k_1, k_2, k_3, \dots)$.
- K_i = plaintextteki i.karakteri, ciphertextteki i.karaktere çevirmede kullanılan anahtar.

Polyalphabetic



- Şekilde 2 sezar cipher'ı var – $k=5$ ve $K=19$ ile.
- Birine C_1 , diğerine C_2 dersek; C_1, C_2, C_2, C_1, C_2 şeklinde bir dizi oluşturabiliriz.
- Bu durumda plaintextteki ilk karakteri C_1 ile ikinciyi C_2 ile 3.yü C_2 ile vb. şekilde şifreleyebiliriz.
- Pattern tekrarlı olursa – 6.yı yine C_1 vb. şekilde gider.

Plaintext letter:	a b c d e f g h i j k l m n o p q r s t u v w x y z
$C_1(k=5)$:	f g h i j k l m n o p q r s t u v w x y z a b c d e
$C_2(k=19)$:	t u v w x y z a b c d e f g h i j k l m n o p q r s

Polyalphabetic



$$P = P_1 P_2 P_3 \dots \quad C = C_1 C_2 C_3 \dots \quad k = (k_1, P_1, P_2, \dots)$$

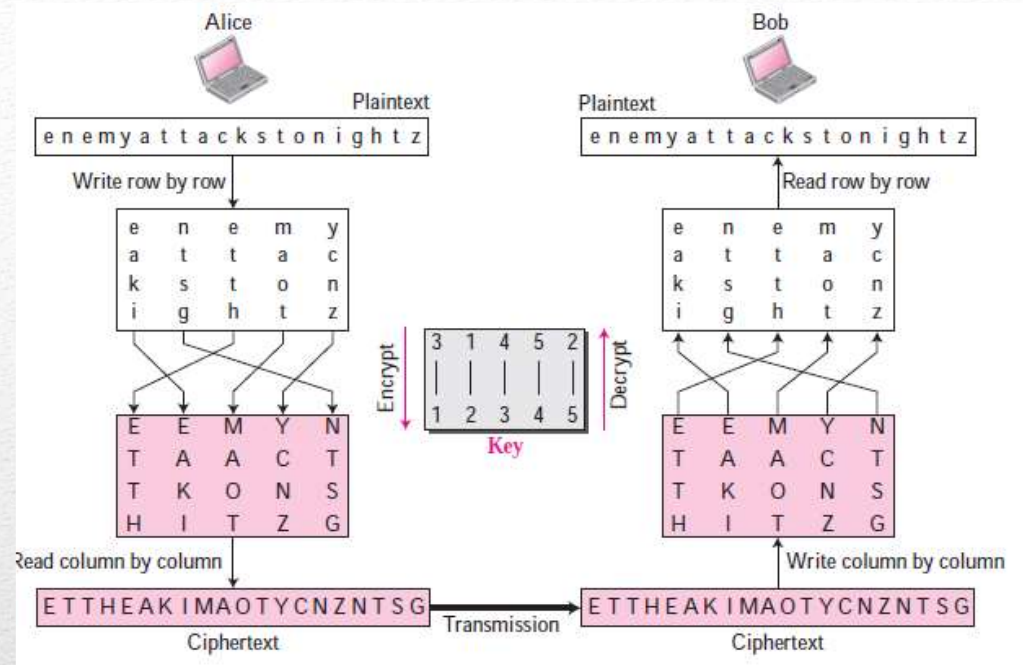
$$\text{Encryption: } C_i = (P_i + k_i) \bmod 26 \quad \text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

- Örnek: autokey şifreleyici.
- Özelliği, plaintext'i anahtarın içine dahil etmesi.
- İlk anahtar taraflar arasında gizlice belirlenir.
- İkincisi, ilk plaintext karakteridir (0-25 arasında).
- 3. İse 2.plaintext karakteridir ve bu şekilde devam eder.
- 2 türü var: key autokey ve text autokey.
- key autokey, anahtar streamindeki bir önceki alt anahtarı kullanarak yeni alt anahtarı üretiyor.
- text autokey ise plaintext 'teki bir önceki karakteri kullanarak yeni alt anahtarı üretiyor.
- Modern kriptografide kullanılan self-synchronizing stream ciphers – bir autokey şifreleyicisidir.
- Burada her karakter, bir pseudorandom digit stream (keystream) ile kombine edilir.

Polyalphabetic



- karakterin yeri değiştirilir.
- Reorder veya transpose.
- Örneğin şekilde şifreleme için plaintext satır satır yazılır.
- Kolonlar anahtar kullanılarak yer değiştirilir.
- Ciphertext ise, yeni tablonun kolon kolon okunarak oluşturulmasıdır.
- Alıcı ise şifreyi çözmek için önce sırayla kolonları oluşturur.
- Anahtara göre yeniden düzenler ve son olarak satır satır okur.
- Yani yapılan işlemleri tersinden yapar.



Transposition

KRİPTOLOJİ – MODERN ŞİFRELEYİCİLER

*Kaynak : James F. Kurose & Keith W. Ross - Computer networking: A top-down
approach 6th Edition*

Behrouz A. Forouzan - TCP/IP protocol suite 4th edition



- stream ve block cipher
- Birçok Internet protokolünde – PGP (secure e-mail) , TLS (secure TCP connection) ve IPsec (ağ-katman iletimini güvenli hale getirme) block cipher'lar kullanılır.
- mesaj k bitlik bloklar halinde şifrelenir. – önce bölünür, sonra bağımsız şifrelenir.
- cipher, one-to-many mapping ile k-bitlik plaintext bloğunu -> k-bit ciphertext bloğuna dönüştürür.

Modern Şifreleyiciler

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

- Diyelim ki $k=3$ olsun.
- Blok şifreleyicisi 3-bitlik girişleri 3 bitlik çıkışlara haritalar.
- Şekilde bir olasılık var.
- Burada one-to-one mapping kullanılmış – yani her giriş için farklı bir çıkış var.
- Örneğin giriş: 010110001111, çıkış: 101000111001
- Peki 3-bitlik blok için kaç olası mapping var?
- – tüm olasılıkların permütasyonu – 2^3 olası giriş var.
- Dolayısı ile $8! = 40.320$ farklı şekilde haritalama yapılabilir.

Block Cipher



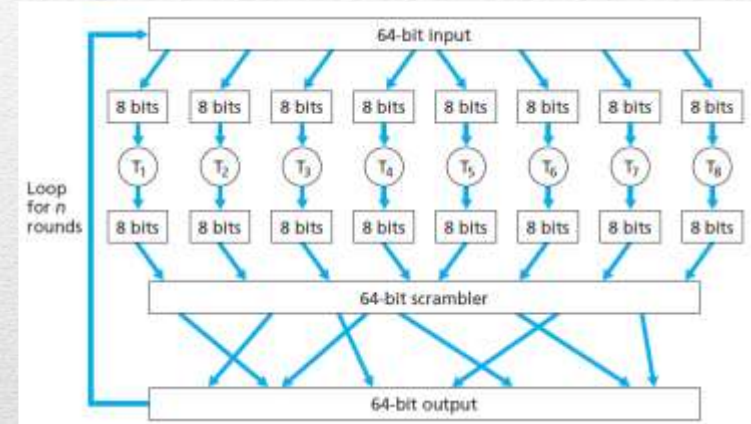
Doç.Dr.İpek Abasıkeleş-Turgut

- Brute-force atakları tüm mapping'leri kullanarak ciphertext'i kırmaya çalışır.
- Masaüstü bir PC ile rahatlıkla kırılır.
- O nedenle blok cipher'ları daha geniş bloklar kullanır – $k = 64$ ve fazlası.
- K-blok cipher için olası mapping sayısı $2^k!$ dir.
- full-table block cipher, k 'nın orta değerleri için saldırılara dayanıklı ama uygulama zor
 - $K = 64$ için hem gönderici hem de alıcının 2^{64} giriş değeri için gerekli olan haritayı elde etmesi lazım.
 - Dahası, anahtar değişirse, tablo yeniden oluşturulmalı – pratikten uzak.

Block Cipher



- Bunun yerine rasgele deęişen tablolar simüle eden fonksiyonlar kullanılır
- Fonksiyon öncelikle 64-bit blokları 8 bitlik 8 adet yığına böler.
- Her yığın parçası, 8-bite 8-bitlik tablolar tarafından işlenir.
- Tablolardan çıkan 8 bitlik output'lar 64-bitlik blok olacak şekilde yeniden biraraya getirilir.
- Blok içindeki 64 bitin pozisyonu 64-bitlik output'u üretmek için karıştırılır (permüte edilir - deęiştirilir)
- Ardından bu output, yeni bir 64-bitlik giriş halini alır ve yeni bir döngü başlar.
- N döngüden sonra fonksiyon 64 bitlik ciphertext üretir.
- Döngü amacı – her girişin daha fazla sayıda final bitini etkilemesini sağlama
- Anahtar –8 adet permütasyon tablosu
 - (deęiştirme fonksiyonunun herkes tarafından bilindięini varsayalım.)



Block Cipher

- Popüler blok cipher'lar - DES - Data Encryption Standard, 3DES ve AES (Advanced Encryption Standard).
- önceden belirlenmiş tablolar yerine fonksiyonlar kullanırlar.
- anahtar olarak bir bit stream kullanırlar.
- Örneğin DES, 56-bit anahtarlı 64-bit bloklar kullanır.
- AES, 128-bit bloklar kullanır ve 128, 192 ve 256 bit uzunluktaki anahtarlar ile çalışabilir.

Block Cipher



- Algoritmanın anahtarı, spesifik mini tablo haritalamalarını ve permütasyonları belirler.
- Bu cipher'lar için brute force ataklarının tüm anahtarlar boyunca tarama yapması lazım – şifre çözme algoritmasını her anahtar için deneyerek.
- N uzunluktaki bir anahtar için 2^n adet olası anahtar var.
- NIST'in ölçümlerine göre bir makine 56-bit DES anahtarını 1 saniye içinde kırabiliyor – 2^{56} anahtarı 1 saniye içinde deneyebiliyor.
- 128-bitlik AES anahtarını kırmak ise 149 trilyon yıl alır anlamına geliyor.

Block Cipher



- Genelde uzun mesaj şifreleme gerekir.
- Blok cipher kullanmada bir sorun ?
- Ya plaintext blokları aynı olursa – Örn. HTTP/1.1 gibi
- Aynı ciphertext -> eğer saldırgan protokol yapısını bilirse tahmini kolaylaşır – aynı ciphertext blokları ile de çözebilir.
- Çözüm? Ciphertext içine rasgelelik katma
- Böylece aynı plaintext – farklı ciphertext

Cipher-Blok Zinciri



- $m(i)$ – i.plaintext bloğu, $c(i)$ ise i.ciphertext bloğu.
- a XOR b işlemi a ve b adında 2 bit stream'inin XOR işlemini yapsın – bu işlemin bit bit yapıldığını ve aynı ise 0, farklı ise 1 verdiğini hatırlayalım.
- Blok cipher şifremiz S, algoritmamız ise K_s olsun.
- Gönderici $r(i)$ ile ifade edilen i.blok için rasgele bir k-bitlik sayı üretir ve $c(i) = K_s(m(i) \text{ XOR } r(i))$ k-bitlik rasgele sayı her blok için ayrı ayrı seçilir.
- Gönderici $c(1), r(1), c(2), r(2), \dots$ gönderir.
- Alıcı ise $c(i), r(i)$ leri alarak $m(i) = K_s(c(i)) \text{ XOR } r(i)$ işlemi ile plaintexti elde eder.

Cipher-Blok Zinciri



- $r(i)$ saldırgan tarafından ele geçirilse bile, $m(i)$ yani plaintexti elde edemez çünkü anahtarı – S - bilmiyor.
- iki blok, i ve m – aynı olsun. $m(i)$ ve $m(j)$ aynı olsa bile $c(i)$ ve $c(j)$ farklıdır çünkü $r(i)$ ve $r(j)$ farklıdır – çok yüksek olasılıkla
- Şekildeki 3 bitlik blok cipher için plaintext'imiz 010010010
- doğrudan şifreleme – rasgelelik olmadan – ciphertext 101101101
- Eğer saldırgan ciphertexti incelerse – her blok aynı olduğunu fark eder.
- Ama eğer rasgelelik kullanırsa, örneğin $r(1) = 001$, $r(2) = 111$, ve $r(3) = 100$ olsun.
- Bu durumda $c(1) = 100$, $c(2) = 010$, ve $c(3) = 000$ olacağı için blokların aynı olduğu fark edilmeyecek.
- gönderen, alıcıya $c(1)$, $r(1)$, $c(2)$, $r(2)$, $c(3)$ ve $r(3)$ 'ü yolluyor.
- plaintexti elde etmek için ancak K_s 'nin bilinmesi lazım.

input	output	input	output
000	110	100	011
001	111	101	010
010	101	110	000
011	100	111	001

Table 8.1 ♦ A specific 3-bit block cipher

Cipher-Blok Zinciri

- Rasgelelik bahsedilen problemi çözer ama yeni bir problem çıkarır – gönderen, kendi bitleri kadar daha bit yollar.
- bandgeniřlięi 2 kat fazla
- Bu amaçla blok cipher'lar CBC – Cipher Block Chaining denilen bir teknik kullanırlar.
- Burada amaç – sadece ilk mesajda 1 rasgele deęer yollamak, gönderici ve alıcının dięer rasgele sayılar yerine hesaplanan kod blokları kullanmalarıdır.

CBC



- 1 – mesaj şifrelenmeden önce gönderici k-bit string üretir - Initialization Vector (IV) Bu vektör $c(0)$ 'ı temsil eder. Gönderici bunu alıcıya plaintext içinde yollar.
- 2 – İlk blok için gönderici $m(1)$ XOR $c(0)$ işlemi uygular ve ardından algoritmaya sokarak ciphertext bloğunu üretir. $c(1) = K_S(m(1) \text{ XOR } c(0))$. Gönderici şifrelenmiş $c(1)$ 'i alıcıya yollar.
- 3 – i.blok için göndericinin ürettiği ciphertext bloğu: $c(i) = K_S(m(i) \text{ XOR } c(i - 1))$ 'dir.

CBC



- alıcı hala orijinal mesajı elde etme imkanına sahip - Alıcı $c(i)$ 'yi alır, $c(i-1)$ 'i zaten bildiği için ve K_s ile çözerek $m(i)$ 'yi elde eder.
- Burada iki blok aynı bile olsa ciphertext'leri neredeyse her zaman farklı olacaktır.
- Ayrıca gönderici IV'yi açık şekilde yolladığı halde saldırgan ciphertext'i çözemez çünkü anahtarı bilmiyor.
- Son olarak gönderici sadece 1 blok yollar – IV – dolayısı ile bantgenişliği kullanımı yüzlerce blokluk veri yollandığı düşünüldüğünde önemsiz olacak şekilde azdır.

CBC



- Örneğin az önceki tablo için 3-bit blok cipher düşünelim.
- Plaintext : 010010010 ve $IV = c(0) = 001$ olsun.
- Bu durumda $c(1) = K_S(m(1) \text{ XOR } c(0)) = 100$, $c(2) = K_S(m(2) \text{ XOR } c(1)) = 000$ ve $c(3) = 101$ olur.
- Alıcı, IV ve K_S 'yi bildiği için orijinal texti çözer.
- CBC ile güvenli ağ protokolü tasarlarken önemli bir sonuç var: IV'yi göndericiden alıcıya dağıtırken bir mekanizma sunmak lazım.

CBC



Public Key KRİPTOLOJİ

Kaynak : James F. Kurose & Keith W. Ross - Computer networking: A top-down approach 6th Edition

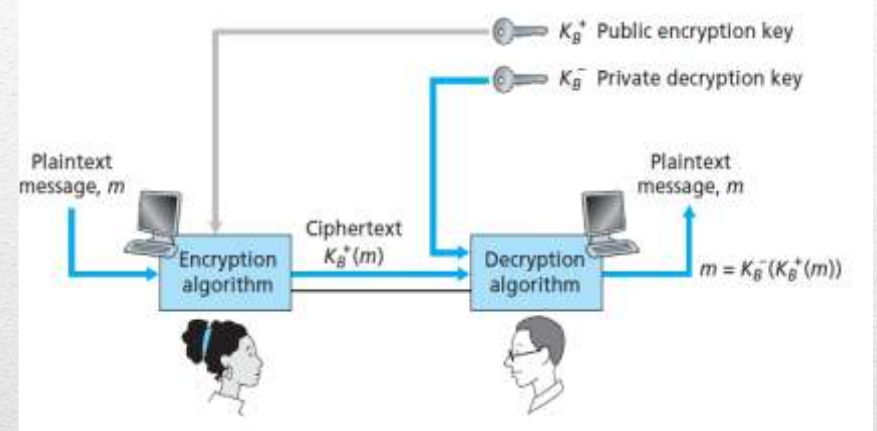


- Sezar'dan 1970'li yıllara hem şifreleme hem şifre çözme için aynı anahtarı – simetrik anahtar kullanımı.
- Zorluk – anahtarın paylaşımı
- Tamamen güvenilir olmalı. Nasıl?
- Peki taraflar paylaşılan bir gizli anahtar olmadan nasıl şifreli haberleşme yapabilir?
- 1976 - Diffie ve Hellman bir algoritma öneriyorlar.
- Bugün sadece şifrelemede değil kimlik doğrulama ve dijital imzada da kullanılıyor.

Asymmetric Key = Public Key Cryptography



- Gizli bir anahtar paylaşmak yerine 2 anahtar
- birisi herkese görünür olan public anahtar, diğeri ise sadece alıcı tarafından bilinen gizli - private anahtar.
- K_B^+ ve K_B^- = public ve private anahtarlar
- Alice m adlı mesajını Bob'un public anahtarı ve bilinen (standart) bir şifreleme algoritması ile şifrelesin = $K_B^+(m)$
- Bob, Alice'in şifreli mesajını alarak kendi gizli anahtarını ve bilinen bir şifre çözme mesajını kullanarak mesajı elde etsin = $K_B^-(K_B^+(m))$.



Public Key Cryptography

- Öyle encryption/decryption algoritmaları ve public/private anahtar seçim teknikleri var ki: $K_B^- (K_B^+ (m)) = m$ 'dir.
- Yani Bob'un public anahtarı m mesajına uygulanarak şifrelenir ve bu şifreli mesaj Bob'un private anahtarı ile açılır.
- Böylece Alice, Bob'a herkesçe bilinebilecek bir anahtar ile gizli mesaj gönderebilir.
- Hem de karşılıklı gizli bir anahtar paylaşmadan.
- Ayrıca public-private key şifrelemesini değiştirerek $K_B^- (K_B^+ (m)) = K_B^+ (K_B^- (m)) = m$ 'nin nasıl elde edildiğine kısaca bakacağız.

Public Key Cryptography

- Basit ancak 2 sorun var
- Saldırgan Bob'un public anahtarı ve Alice'in şifreli mesajını biliyor.
- chosen-plaintext atağı ile Alice'in göndereceğini tahmin ettiği bir mesajı şifreleyebilir.
- Eğer public key cryptography doğru şekilde çalışırsa – anahtar seçimi ve şifreleme/şifre çözme işlemleri uygun seçilirse – saldırgan için Bob'un gizli şifresini tahmin etmek de Alice'in Bob'a gönderdiği mesajı çözmek de imkansız olur.
- İkinci sorun, Bob'un şifreleme anahtarı public olduğu için herhangi biri Alice olduğunu iddia ederek mesaj yollayabilir.
- Eğer tek bir paylaşılan gizli şifre olsaydı, alıcı eşsiz olurdu ama public key olduğu için herhangi biri de mesaj yollayabilir.
- Dijital imza kısmında göndericinin nasıl bir mesaja bağlanması gerektiği açıklanacak.

Public Key Cryptography



- Public key cryptography için birçok algoritma var ama en yaygını – RSA - Ron Rivest, Adi Shamir, and Leonard Adleman.
- RSA, mod-n aritmetiğinin geniş kullanımı.
- Kısaca modüler aritmetiğe bakacak olursak – $x \bmod n = \text{kalan } x/n$.
- Örneğin $19 \bmod 5 = 4$.
- Modüler aritmetik ile ekleme, çarpma, üs alma gibi işlemler yapılabilir.
- Ancak her işlemin sonucu, sonucun n ile bölünmesinden kalan olarak değiştirilir.

RSA



- $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
- $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
- $[(a \bmod n) \cdot (b \bmod n)] \bmod n = (a \cdot b) \bmod n$
- $(a \bmod n)^d \bmod n = a^d \bmod n \rightarrow \text{önemli}$
- Diyelim ki Alice, Bob'a bir RSA-şifreli mesaj yollamak istiyor.
- Mesajın sadece bir bit dizisi olduğunu biliyoruz.
- Her bit dizisi ise bir integer ile ifade edilebilir.
- Örneğin mesajın bir kısmı 1001 ise, bu onluk olarak 9'a karşılık gelir.
- RSA algoritması ile mesaj şifrelenirken, mesajın ifade ettiği integer sayı şifrelenir.

RSA



- RSA'nin birbiri ile bağlantılı 2 bileşeni var:
- 1 – public ve private anahtar seçimleri nasıl olacak?
- 2 - encryption ve decryption algoritmaları nasıl olacak?

RSA



- Bob, public ve private anahtarları üretmek için şu adımları izler:
- 1 – 2 adet büyük asal sayı seçer – p ve q olsun. Peki ne kadar büyük? Eğer çok büyük olursa RSA'nın kırılması zorlaşır ama bu sefer de şifreleme ve çözme işleri zorlaşır. Yapılan denemeler sonucunda p ve q sayılarının çarpımının yaklaşık 1024 bit olmasının uygun olacağı söylenmiştir.
- 2 – $n = pq$ ve $z = (p-1)(q-1)$ hesaplanır.
- 3 – n 'den daha küçük olan, e adından bir sayı seçilir. Öyle ki z ve e arasında 1 haricinde ortak çarpan olmayacak yani e ve z aralarında asal olacak. e değeri encryption işleminde kullanılacak.
- 4 – Öyle bir d sayısı bul ki $ed - 1$, z ile tam olarak bölünsün. d değeri şifre çözmede kullanılacak. Dolayısı ile $ed \bmod z = 1$ olacak.

RSA



- 5 – Bob’un üreteceği public key: $K_B^+ = (n,e)$ sayı çifti, private key: $K_B^- = (n,d)$ sayı çifti.
- Alice ve Bob arasındaki şifreleme işlemi şu şekilde yapılır:
- Diyelim ki Alice, Bob’a bir bit dizisi yollayacak. Öncelikle bunun integer karşılığını hesaplar – m olsun. $m < n$ olacak şekilde. Alice öncelikle m^e ’yi hesaplar ve n ’ye göre mod alır. Şifrelenen değer, c , elde edilir.
- $c = m^e \bmod n$
- Bit dizisine karşılık gelen ciphertext c , Bob’a yollanır.
- Bu mesajı çözmek için Bob şunu hesaplar:
- $m = c^d \bmod n$

RSA



- Örneğin Bob $p = 5$, $q = 7$ değerini seçsin (gerçekte bu sayılar güvenlik için çok küçük).
- Bu durumda $n = 35$, $z = 24$ olacak.
- Bob $e = 5$ seçsin.
- 5 ile 24 arasında hiç ortak çarpan yok.
- Son olarak $d = 29$ seçsin çünkü $5 \cdot 29 - 1$ (144) 24'e tam olarak bölünüyor.
- Bob, $n = 35$ ve $e = 5$ değerini public, $d = 29$ değerini private yapar.

RSA



- Diyelim ki Alice love stringini yollamak istiyor.
- Her harf 1 ve 26 arasında bir değer olacak – alfabenin integer karşılığı, $a = 1$, $z = 26$ olmak üzere.
- Aşağıda mesajların şifrlenmesi ve çözülmesi işlemleri var:

Plaintext Letter	m : numeric representation	m^e	Ciphertext $c = m^e \bmod n$
l	12	248832	17
o	15	759375	15
v	22	5153632	22
e	5	3125	10

Ciphertext c	c^d	$m = c^d \bmod n$	Plaintext Letter
17	4819685721067509150915091411825223071697	12	l
15	127834039403948858939111232757568359375	15	o
22	851643319086537701956194499721106030592	22	v
10	10000000000000000000000000000000	5	e

RSA



- Bu örnekte 4 harfî ayrı bir mesaj gibi düşündük.
- Gerçekte bu harfler 8-bit ASCII şekline dönüştürülür ve ardından 32-bitlik bit dizisi integer karşılığı ile şifrelenir.
- Bu konu ile ilgili ayrıntılar: bu kadar büyük asal sayılar nasıl seçilir? E ve d sayıları nasıl seçilir? Bu kadar büyük sayılarla üssel işlemler nasıl yapılır? Vb. – bu dersin konusu haricinde

RSA



- RSA'daki üssel işlemler çok fazla zaman alıcı.
- Buna karşılık DES, yazılımda 100, donanımda 1000 - 10000 kat daha hızlı.
- O nedenle pratikte RSA, simetrik key cryp. ile birlikte kullanılır.
- Diyelim ki Alice, Bob'a büyük miktarda şifreli veri yollayacak.
- Önce kendisi verisini şifrelemek istediği bir anahtar seçer – session key – K_s .
- Bu anahtar simetrik key cipher tarafından kullanılacak bir ortak simetrik anahtar olacağı için Bob'a haber vermesi lazım.
- O nedenle bu anahtarı Bob'un public anahtarını kullanarak şifreler ve Bob'a yollar – RSA benzeri şekilde - Diffie-Hellman (DH)
- $c = (K_s)^e \text{ mod } n$ 'yi hesaplar ve gönderir.
- Daha sonraki veri gönderme işlemleri simetrik anahtar kullanılarak yapılabilir.

Session Keys – Oturum Anahtarları



- RSA'nın güvenliği, bir sayıyı hızlı bir şekilde çarpanlarına ayırabilen bilinen bir algoritma olmadığı gerçeğine dayanır.
- Public değer n 'yi q ve p 'ye ayırma.
- Eğer p ve q değerleri bilinirse, public e değeri sayesinde gizli olan d değerine kolayca erişilebilir.
- Her ne kadar bilinen bir algoritma olmasa da, n yi p ve q 'ya hızlı ayıran yöntemler olabilir.
- Özellikle quantum hesaplama ve quantum bilgisayarlar ile
- Güncel bir alternatif teknoloji: Elliptic Curve Cryptography (ECC)
- ABD Ulusal Standartlar ve Teknoloji Enstitüsü, kuantum dirençli açık anahtarlı kriptografik algoritmalar için üç taslak standart talep etmiş, değerlendirmiş ve önermiştir [NIST 2025].

RSA



MESAJ BÜTÜNLÜĞÜ – DİJİTAL İMZA

Kaynak : James F. Kurose & Keith W. Ross - Computer networking: A top-down approach 6th Edition



- Mesajın güvenilirliği – confidentiality – için şifrelemenin nasıl kullanıldığını gördük.
- Şimdi en az güvenlik kadar önemli olan integrity – diğer bir ifade ile mesaj authentication (kimlik doğrulama) – nasıl sağlandığına bakacağız.
- Bu alanda 2 konu var: dijital imza ve end-point authentication.
- Diyelim ki Bob bir mesaj aldı – (ister plaintext halinde isterse ciphertext) Mesajın kimlik doğrulama işlemi için:
 - 1 – mesajın gerçekten Alice’den kaynaklandığının,
 - 2 – mesajın yolda kurlanmadığının (tamper) doğrulanması gerekir.

Message Integrity & Dijital İmza

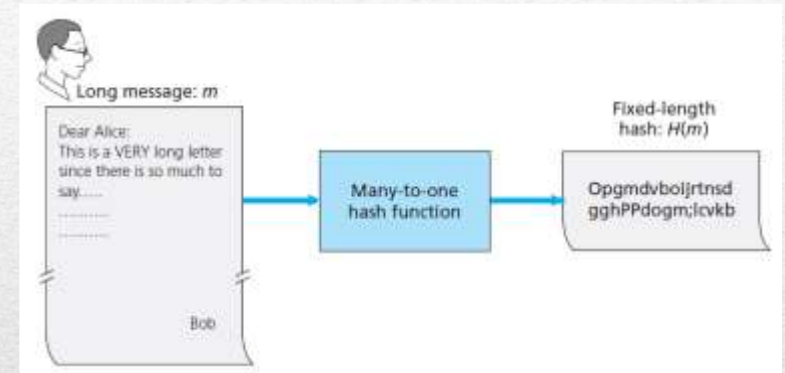


- Spesifik bir örnek: diyelim ki bir bilgisayar ağı, router çiftleri arasındaki yolu belirlemek için link-state routing algoritmasını kullanıyor.
- Örneğin OSPF.
- Bu algorithmada her router ağıdaki diğer tüm router'lara, doğrudan bağlı olduğu komşularının ve bu komşularına olan uzaklığının/maliyetinin listesini içeren bir link-state mesajı broadcast eder.
- Router'lar bu mesajları aldıktan sonra ağın tam haritasını çıkarırlar ve en-düşük maliyetli routing algoritmasını çalıştırarak kendi iletim tablolarını konfigüre ederler.
- Routing algoritması üzerinde basit bir atak, yanlış link-state bilgisi içeren sahte link-state mesajları dağıtmak.
- Dolayısı ile integrity sağlanması lazım – router B, router A'dan bir link-state mesajı aldığında bu mesajın gerçekten A tarafından yaratıldığından ve yolda herhangi bir kötü niyetli işleme maruz kalmadığından emin olması lazım.

Message Integrity & Dijital İmza



- Şekildeki gibi bir giriş alır, m , ve ardından sabit boyutlu hash = $H(m)$ adı verilen string'i hesaplar.
- Internet checksum ve CRC bu tanıma uyar.
- Cryptographic Hash fonk.ise ek bir özellik içerir:
- Sayısal olarak x ve y adında iki farklı mesajın aynı hash'e sahip olması - $H(x) = H(y)$ imkansızdır.
- Bu durumda bir saldırganın hash fonk.ile korunan bir mesaj yerine başka bir mesaj koymasla mümkün değildir.
- Yani diyelim ki göndericinin yollayacağı mesaj m için hash fonk. $H(m)$ olsun.
- Saldırganın başka bir y mesajı için orijinal mesajla aynı olan bir hash değeri uydurması olası değildir.



Cryptographic Hash Fonksiyonları

- Basit bir örnek olarak checksum alalım – Internet checksum.
- Zayıf bir Cryptographic Hash fonksiyonudur.
- İkilik tümleyen aritmetiği yerine her karakteri bir byte olarak düşünüp, byte’ları birbirine ekleyerek tek seferde 4-byte ‘lık yığın olduğunu varsayalım.
- Diyelim ki Bob Alice’e borçlu olsun ve şu texti içeren bir fatura yollasın: IOU100.99BOB.
- Bu harflerin hexadecimal formda ASCII gösterimi: 49, 4F, 55, 31, 30, 30, 2E, 39, 39, 42, 4F, 42’dir.
- Şekilde görüldüğü gibi bu mesajın 4-byte checksum değeri B2 C1 D2 AC’dir.
- Şekilde altta başka bir mesaj - “IOU900.19BOB” ve checksum karşılığı vardır.
- Her ikisi de aynıdır.
- Bu basit checksum algoritması bahsedilen gereksinimi ihlal eder – orijinal veri haricinde başka bir veri de aynı checksum’ı vermiştir.
- Güvenlik gerekçesiyle checksum’dan daha güçlü hash fonksiyonlarına ihtiyaç vardır.

Message	ASCII Representation				
I O U 1	49	4F	55	31	
0 0 . 9	30	30	2E	39	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

Message	ASCII Representation				
I O U 9	49	4F	55	39	
0 0 . 1	30	30	2E	31	
9 B O B	39	42	4F	42	
	B2	C1	D2	AC	Checksum

- Bunlardan bir tanesi MD5 hash algoritmasıdır.
- 4 aşamalık işlem den sonra 128-bit hash hesaplar.
- Birisi padding aşaması – istenilen şartları sağlaması için mesaja yeterli miktarda 0 doldurma (padding).
- Diğer ekleme (appending) adımı – doldurma yapmadan önce mesajın boyutunun 64-bitlik karşılığını mesaja ekleme.
- Diğer aşamalar akümülatör başlatma (initialization) ve final döngü adımı.
- Diğer bir hash algoritması Secure Hash Algorithm (SHA-1).
- MD4 (MD5'in atası) ile benzerdir.
- US federal standardıdır.
- 160-bit mesaj özetine sahiptir. Daha geniş çıktı, daha güvenli.

Cryptographic Hash Fonksiyonları



- Mesaj bütünlüğünü sağlamak için:
- 1 – Alice yarattığı m mesajı için $H(m)$ hash'ini hesaplar. Örneğin SHA-1 ile
- 2 – Alice, m mesajına $H(m)$ 'yi ekler ve oluşan genişletilmiş mesajı $(m, H(m))$ Bob'a yollar.
- 3 – Bob genişletilmiş mesajı alır (m, h) ve $H(m)$ 'yi hesaplar. Eğer $H(m) = h$ ise herşey yolunda demektir.
- Sorun?

Message Authentication Code



- Bu yaklaşım görünürde hatalıdır.
- Saldırgan Alice olduğunu iddia ederek sahte bir m' mesajı yaratır ve $H(m')$ mesajlar ardından Bob'a yollar – $m', H(m')$.
- Bob 3.adıma göre hesaplama yapar ve herhangi bir sorun ile karşılaşmaz.
- Dolayısı ile bu yöntemle yani mesajın bütünlüğü için cryptographic hash fonk.kullanmanın yanında paylaşılan bir gizli anahtar s 'ye de ihtiyaç vardır.
- Bu paylaşılan anahtar, bir bit string'i olup, authentication key olarak adlandırılır.

Message Authentication Code

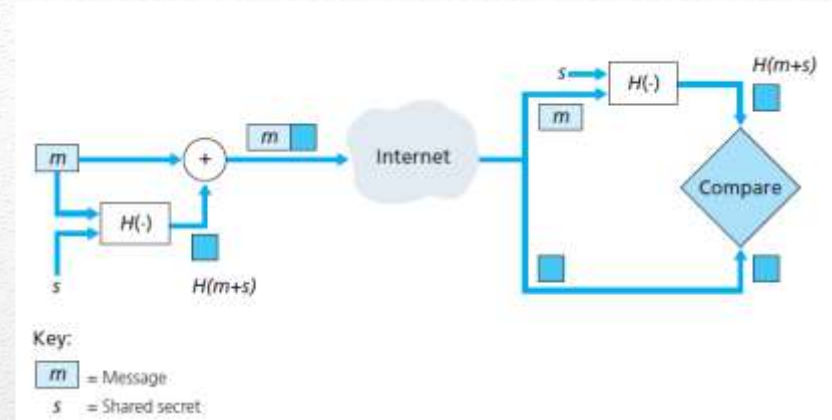


- Bu gizli anahtar kullanımı ile mesaj bütünlüğü şu şekilde sağlanır:
- 1 – Alice m mesajını oluşturur ve m 'yi s ile bağlayarak $m + s$ 'yi yaratır. Ardından $H(m+s)$ için hash hesaplar. Örneğin SHA-1 ile. $H(m+s)$ 'ye mesaj authentication code (MAC) denilir.
- 2 – Alice m mesajına MAC'i ekler ve genişletilmiş mesajı Bob'a yollar – $(m, H(m + s))$
- 3 – Bob, aldığı mesaj (m, h) ve daha önce bildiği s ile MAC'i hesaplar $H(m + s)$
- Eğer $H(m + s) = h$ ise herşey yolunda demektir.

Message Authentication Code



- MAC'in bir avantajı bir şifreleme algoritmasına ihtiyaç duymamasıdır.
- Böylece hem mesaj bütünlüğü hem de güvenliği kompleks şifreleme algoritmaları kullanılmadan sağlanmış olur.
- MAC için birçok standart vardır.
- En popülerlerinden biri HMAC.
- İster MD5 ister SHA-1 ile uygulanabilir.
- Sorun?



Message Authentication Code

- Paylaşılan authentication anahtarı nasıl dağıtılacak?
- yani taraflar bu anahtardan nasıl haberdar edilecek?
- Örneğin link-state routing alg.da bu gizli anahtarın bir şekilde tüm yönlendiricilere dağıtılması lazım – hepsi aynı anahtarı kullanabilir.
- Bir ağ yöneticisi hepsini tek tek gezerek – fiziksel olarak – anahtar tanımlaması yapabilir veya her router’ın kendi public anahtarı olur, ve ağ yöneticisi bu anahtarları kullanarak gizli anahtarı ağ üzerinden mesaj ile yollar.

Message Authentication Code

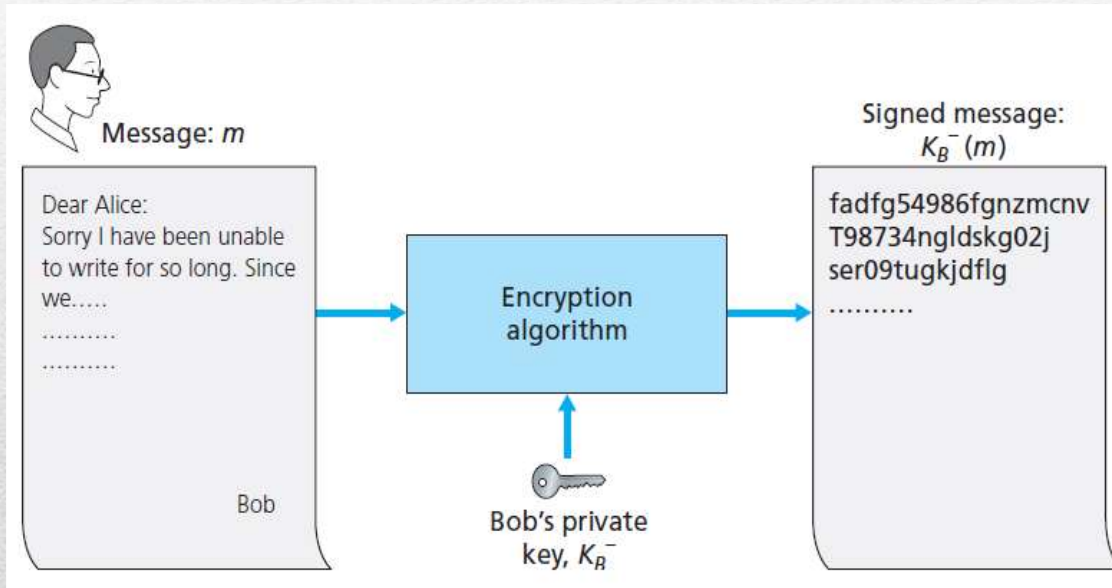


- Dijital dünyada, bir dokümanın yaratıcısı veya sahibi olduğunuzu belirtmek için veya bir doküman içeriğini onaylamak için imza kullanımını gerekebilir.
- Dijital imza, bu amaçla yaratılmış cryptographic tekniktir.
- Doğrulanabilir ve taklidi yapılamaz olması lazım.
- MAC kullanılabilir mi?
- Alıcı için de MAC lazım → iki kişi bilemez → sorun
- Çözüm – public key cry.

Dijital İmza



- Bob, private anahtarı ile şifreler -> Alice Bob'un public anahtarı ile açar.

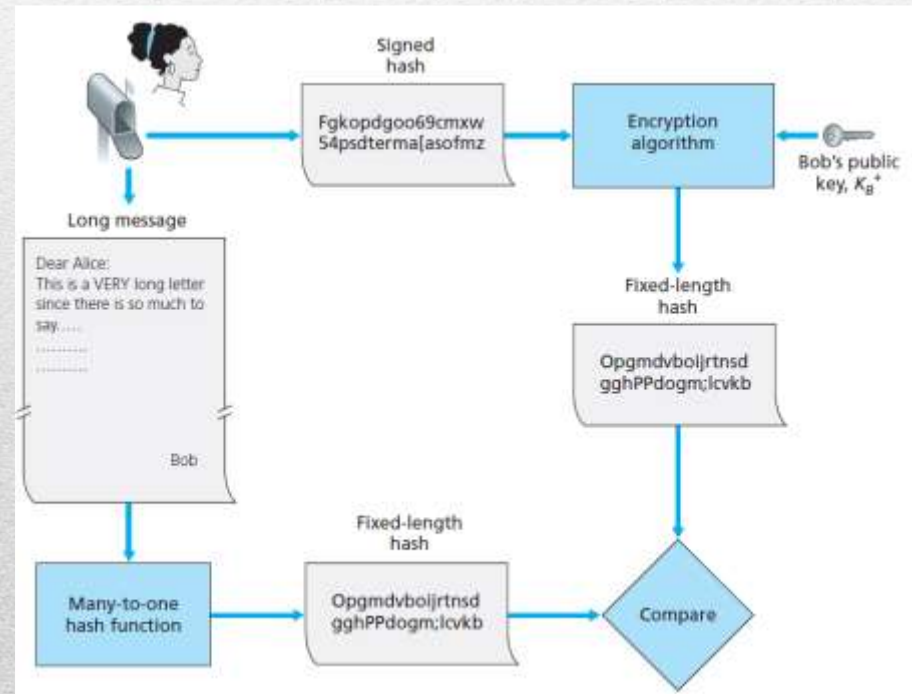
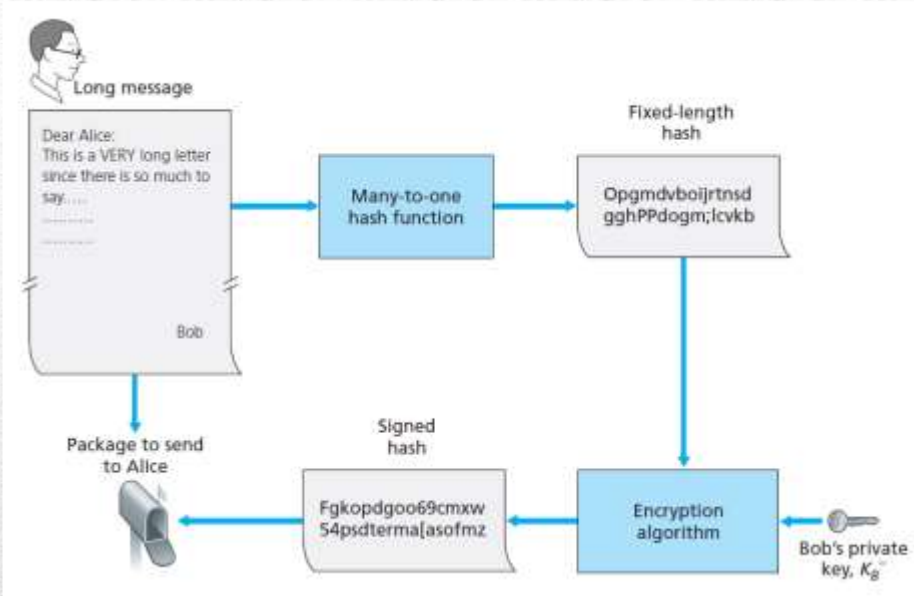


Dijital İmza

- Orijinal metin değişirse – $m' \Rightarrow K_B^+ (K_B^- (m)) \neq m'$ olduğu için integrity sağlanır.
- Sorun – şifreleme & çözme işlemlerinin maliyeti
- Çözüm – hash fonk.
- Bob önce hash'e sokar, ardından hash'i imzalar $\rightarrow K_B^- (H(m))$
- Bob, orijinal mesaj ile dijital imzalı hash'i yollar.
- Alice, dijital imzalı hash'den orijinal mesajı elde eder. Kendisine gelen orijinal ile kıyaslar. Eğer eşitse sorun yok.

Dijital İmza





- İkisi de mesaj ile başlar.
- MAC = anahtar ekleyip hash'e sokar – şifreleme yok (ne public ne simetrik)
- Dijital imza = hash alınır ve public key cry. ile şifrelenir.
- Dijital imza daha ağır bir iş
- Sertifikasyon mercileri ile birlikte Public Key Infrastructure (PKI) gerektirir.

Dijital İmza vs MAC

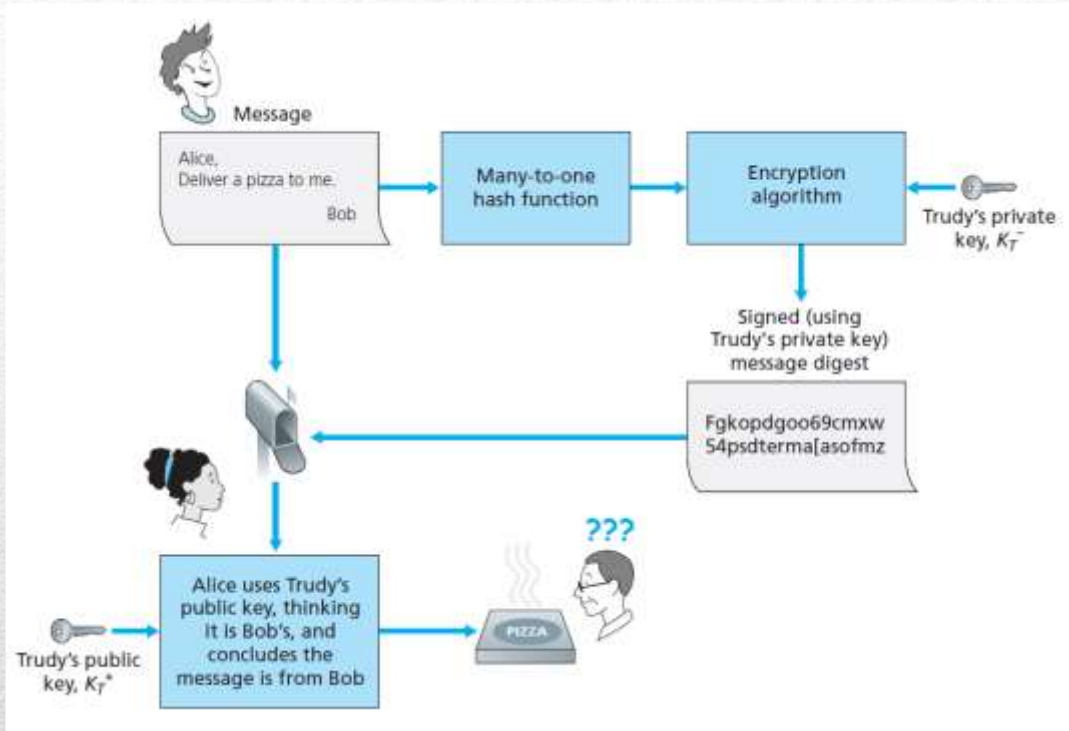


- Dijital imzanın önemli bir uygulaması: public key sertifikasyonudur.
- Public key'in spesifik bir kişiye bağlandığını onaylamak.
- Popüler güvenli ağ protokollerinde kullanılır – Ipsec ve TLS gibi.

Public Key Certification

- Alice, pizza dağıtıcısı ve Internet üzerinden sipariş alıyor olsun.
- Bob da bir pizza sever olarak Alice'e ev adresini ve istediği tür pizzayı içeren bir plaintext yollasın.
- Ayrıca mesajında dijital imza kullansın.
- Yani orijinal plaintext'in imzalı hash'ini yollayarak mesajın kaynağı olduğunu ispatlasın.
- İmzayı doğrulamak için Alice, Bob'un public anahtarını elde etsin – public key sunucu veya e-mail mesajı ile.
- Bu şekilde, public anahtarla çözülen mesajın Bob'un mesajı olduğunu anlar.
- Peki herşey yolunda ilerler mi?
- Hayır.
- Diyelim ki saldırgan Bob olduğunu söylese ve Bob'un ev adresi ile pizza sipariş etse.
- Ayrıca mesaja kendi public anahtarını koysa ve dijital imza olarak da kendi private anahtarını kullansa.
- Alice mesajı aldığı anda Bob'dan geldiğini düşünerek public key kullanıp mesajı elde etse – dijital imzalı bu mesajı sadece public key açabilir – sorun yok?

Pizza Dağıtım Örneği

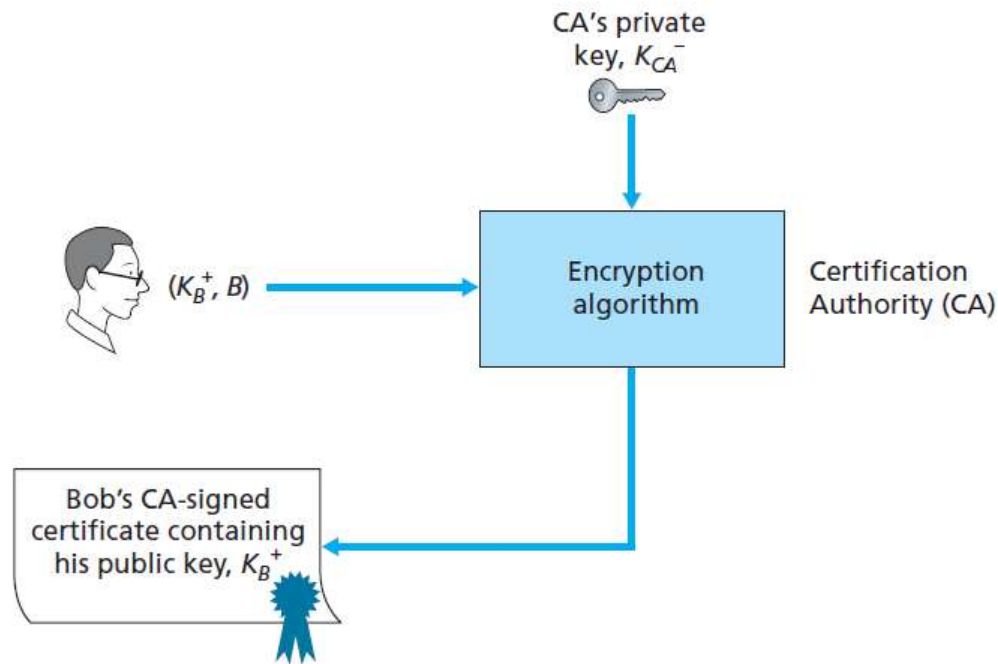


- public key cry. kullanımının işe yaraması için public key sahibi olanın – router, browser veya kişi olabilir – o anahtarın kendisine ait olduğunu kanıtlaması lazım.
- Yani Alice, Bob'un public key'ini aldığı anda onun gerçekten Bob'a ait olduğundan emin olması lazım.
- Bu işlem – yani bir public key'in sahibinin tescili işlemi, Certification Authority (CA) tarafından yapılır.
- Görevleri kimlikleri doğrulamak ve sertifika işlemlerini yapmak.

- 1 – CA, kiři, router vb. řeylerin doęrulanmasını saęlar – yani gerekten o olduęunun saęlamasını yapar.
- Sertifikasyonun nasıl yapılacaęı konusunda sabit bir prosedür yok. CA’ya gúvenmek gerekiyor. Diyelim ki saldırgan eęer Fly-by-Night CA’ya gidip ben Bob’um dedięinde sertifika alabiliyorsa, o zaman Fly-by-Night CA tarafından sertifikası alınan public anahtarlar gúvenilir deęildir. Bununla beraber bir devlet programı kapsamında olan CA’ların daha gúvenli olduęu dűřünülebilir.
- 2 – CA, kimlięi doęruladıktan sonra bir sertifika hazırlar ve public key’i sizle baędařtırır – bind. Sertifikada public anahtar ile, onun sahibi olanın global olarak kendisine özel ve eřsiz olan bilgisi yer alır (isim, IP adresi gibi) Sertifika CA tarafından dijital olarak imzalanır.

Gúrevleri





- Pizza örneğine dönelim.
- Bob, siparişi ile birlikte CA-signed sertifikasını yollar.
- Alice Bob'un CA public anahtarını kullanarak Bob'un sertifikasının doğruluğunu test eder.
- Ardından Bob'un public anahtarını elde eder.

- Hem International Telecommunication Union (ITU) hem de IETF, CA'lar için standartlar geliştirmişlerdir.
- ITU X.509, hem authentication servisi hem de sertifikalar için spesifik bir sentaks tanımlamıştır.
- [RFC 1422]'de güvenli Internet e-mail kullanımı için CA tabanlı anahtar yönetimi tanımlanmıştır.
- Şekilde bir sertifikada yer alan bazı önemli alanlar görülmektedir:

Field Name	Description
Version	Version number of X.509 specification
Serial number	CA-issued unique identifier for a certificate
Signature	Specifies the algorithm used by CA to sign this certificate
Issuer name	Identity of CA issuing this certificate, in distinguished name (DN) [RFC 4514] format
Validity period	Start and end of period of validity for certificate
Subject name	Identity of entity whose public key is associated with this certificate, in DN format
Subject public key	The subject's public key as well indication of the public key algorithm (and algorithm parameters) to be used with this key

Public Key Certification

END POINT AUTHENTICATION

Kaynak : James F. Kurose & Keith W. Ross - Computer networking: A top-down approach 6th Edition



- uç nokta kimlik doğrulaması, bir tarafın kendi kimliğini bilgisayar ağı üzerinden diğer tarafa kanıtlaması işlemi.
- Örneğin kullanıcının kimliğini email sunucusuna ispatlaması.
- İnsanlar bu işi çeşitli şekillerde yapar: yüzyüze, telefonda ses ile, pasaporttaki resimle vb.
- Peki bir taraf diğer tarafa ağda iletişim kurarken kimliğini nasıl ispatlar?
- odaklanacağımız konu “live” party kimlik doğrulaması yani iletişim halihazırda gerçekleşirken yapılan.

End-Point Authentication

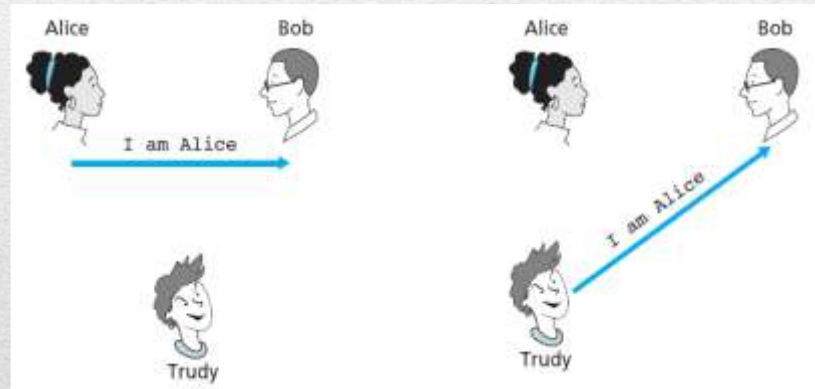
- Somut bir örnek – e-mail sunucusuna kendi kimliğini doğrulayan kişi.
- Ağ üzerinde kimlik doğrulaması yapılırken, iletişim kuran taraflar, görüntü veya ses gibi biometrik bilgilere güvenmez.
- Ayrıca sadece insan değil, router veya istemci/sunucu proseslerinin de iletişiminde kimlik doğrulama işlemi gereklidir.
- Burada kimlik doğrulama işlemi, authentication protocol'ün bir parçası olarak tamamen mesaj ve veri değişimi temeline dayanır.

End-Point Authentication

- Tipik olarak authentication protocol'ü, her iki taraf başka bir protokol çalıştırmadan önce (örneğin güvenilir veri transfer protokolü, routing bilgisi değişim protokolü, e-mail protokolü gibi) çalışır ve her iki tarafında karşının kimliğinden emin olmasını sağlar.
- Nasıl rdt için adım adım ilerlediysek, authentication protocol için de adım adım ilerleyeceğiz.
- Kısaca ap diyeceğiz.
- Diyelim ki Alice, Bob'a kimliğini doğrulayacak.

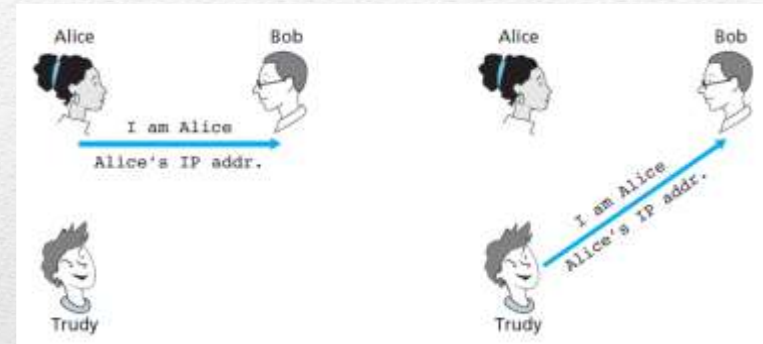
End-Point Authentication

- En basit senaryo.
- Alice Bob'a basit bir şekilde mesaj atarak Alice olduğunu söyler.
- Sorun çok açık – Bob'un, bu mesajı alınca karşısındakinin gerçekten Alice olduğunu bilemez.
- Şekildeki gibi bir durum da olabilir.



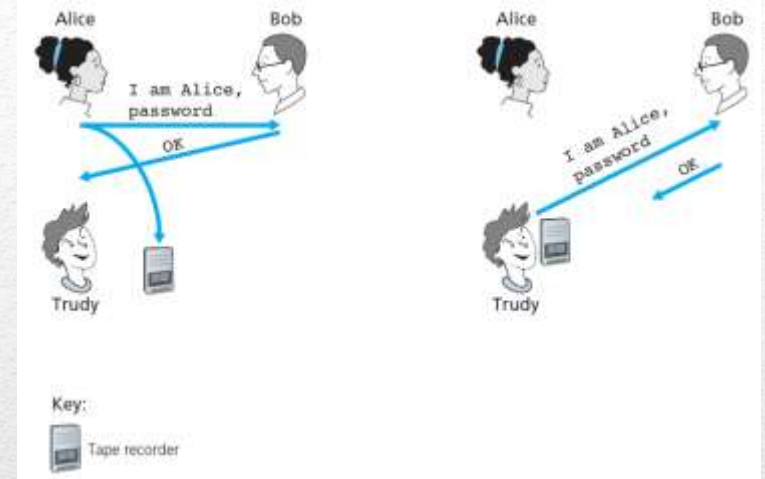
Authentication Protocol ap1.0

- Eğer Alice, her zaman iletişim kurduğu, belirli bir ağ adresine sahipse – IP adresi – Bob IP datagramının kaynak adresine bakarak, gönderenin Alice olduğunu doğrulayabilir.
- Ancak saldırgan, işletim sistemine müdahale ederek, IP datagram yaratabilir ve istediği IP adresini buna koyabilir.
- Ardından bu datagramı veri bağlantı katmanından üzerinden ilk adım router'a yollar.
- Bu bir çeşit IP spoofing'dir.
- IP spoofing, saldırganın ilk-adım router'ı tarafından önlenemez – sadece saldırganın kendi IP'sinin kaynak olduğu datagramları iletecek şekilde konfigüre edilerek.
- Ancak bu global olarak uygulanan veya zorlanacak bir yöntem değildir.



Authentication Protocol ap2.0

- Kimlik doğrulamada kullanılan klasik bir yaklaşım gizli password kullanmak.
- Password, kimliği denetleyen ve kimliğini ispat etmesi gereken taraflar arasında paylaşılan bir gizli anahtardır.
- Gmail, Facebook, telnet, FTP ve diğer birçok servis password authentication kullanır.
- Bu yöntem çok yaygın olarak kullanıldığı için güvenli olduğunu düşünebilirsiniz ama değildir.
- Saldırgan Alice'in iletişimini gizlice dinleyerek – eavesdrop – password'ü öğrenebilir.
- Eğer bunun yaygın olmayan bir durum olduğunu düşünürseniz -> Telnet, başka bir makineye bağlanırken ve log in olurken kullanılan password'ü encrypt olmadan yani şifrelenmeden Telnet sunucusuna iletir.
- Böylece istemcinin veya sunucunun LAN'ında bulunan bir saldırgan Lan üzerinde iletilen tüm paketleri sniff (okuyup-depolama) ederek login password'ü rahatlıkla elde edebilir.
- Bu yöntem şifre çalmada eskiden beri kullanılan klasik bir yöntemdir.



Authentication Protocol ap3.0

- Bir sonraki fikir, password'ü şifreleme (encrypt).
- Böylece Alice'in şifresini saldırgan öğrenemez.
- Diyelim ki Alice ve Bob, bir simetrik gizli anahtar paylaştılar - K_{A-B} -
- ardından Alice password'ünü şifreledi ve kimlik mesajı - I am Alice ile birlikte şifrelenmiş password'ünü Bob'a gönderdi.
- Bob password'ü decrypt etti ve Alice'in kimliğini doğruladı.
- Sadece Alice password ve onu şifrelemek için gerekli olan paylaşılan gizli anahtarı bildiği için herhangi bir sorun olmadığını düşündü.
- Ancak şifreleme işlemi kimlik doğrulama sorununu çözmez çünkü Bob playback (yeniden oynatma) atağına maruz kalabilir.
- Saldırgan Alice'in iletişimi dinleyerek şifreli password'ü öğrenir ve bunu kullanarak Bob'a mesaj yollayabilir.
- saldırganın password'ü öğrenmesine gerek yok - şifreli hali de yeterli. Dolayısı ile 3.0 versiyonundan bunun hiçbir farkı yok.

Authentication Protocol

ap3.1



- 3.1'deki sorun, Bob'un, Alice'in orijinal kimlik doğrulaması ile daha sonra Alice'in orijinal kimlik doğrulamasının playback'ı arasındaki farkı ayırt edememesinden kaynaklanır.
- Yani Bob, Alice'in “live” yani o anda bağlantının diğer ucunda olup olmadığı hakkında bilgi sahibi değil.
- Aldığı mesaj, Alice'in daha önceki kimlik doğrulanmış mesajının kaydedilmiş bir kopyası olabilir.
- Aslında TCP el sıkışma protokolünün de aynı sorunu çözmesi lazım.
- TCP'nin server tarafı, aldığı bir SYN mesajının, daha önceki bağlantılarında kullandığı eski bir SYN mesajının kopyası olmasını istemez.
- Peki TCP sunucu tarafı, istemcinin o anda gerçekten “live” olduğunu nasıl bilecek?
- Bunun için bir sequence sayısı seçer – çok uzun zaman önce kullandığı.
- Bu sayıyı istemciye yollar sonra ondan bu sayıyı içeren bir ack mesajı bekler. Aynı mantığı burada da kullanacağız.

Authentication Protocol

ap4.0



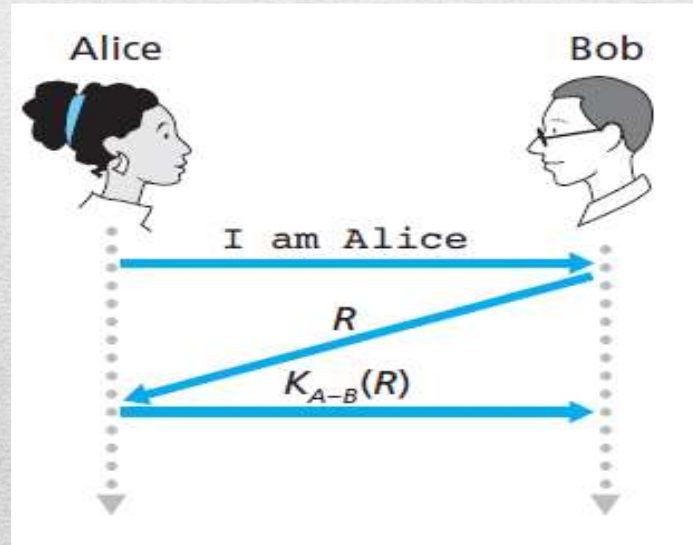
- “nonce”, bir protokolün yaşam süresi boyunca 1 kere kullanacağı bir sayıdır.
- Protokol 1 kez nonce kullanır ve asla bir daha kullanmaz. ap4.0 şu şekilde çalışır:
- 1 – Alice, Bob’a “I am Alice” mesajını yollar.
- 2 – Bob bir nonce seçer. Diyelim ki R. Bunu Alice’e yollar.
- 3 – Alice bu sayıyı Alice and Bob’un paylaşılan gizli anahtarı K_{A-B} ile şifreler ve şifreli nonce’ı $K_{A-B}(R)$ Bob’a yollar. Tıpkı ap3.1’de olduğu gibi sadece Alice K_{A-B} ’yi bildiği için şifreleme yapabilir. Nonce kullanmanın amacı, Alice’in “live” olduğundan emin olmaktır.
- 4 – Bob, mesajın şifresini çözer ve elde ettiği nonce ile kendi yolladığı aynı ise Alice’in kimliğini doğrular.

Authentication Protocol

ap4.0



- Şekilde görüldüğü gibi, one-in-a-lifetime değeri R kullanarak ve geriye dönen $K_{A-B}(R)$ değerini kontrol ederek kimlik doğrulama yapılabilir.
- Bu çözümde nonce ve simetrik key enc. kullandık.
- Peki nonce ve public key cryptography kullanabilir miyiz?



Katmanlarda Güvenlik Önlemleri – Uygulama Katmanı

Kaynak : James F. Kurose & Keith W. Ross - Computer networking: A top-down approach 6th Edition



- Ağ güvenliğinin temel konularından bahsettik – simetrik key ve public key cryp. , end-point authentication, anahtar dağıtımı, mesaj bütünlüğü ve dijital imza.
- Peki bunlar kullanılarak Internette güvenlik nasıl sağlanır?
- Internetin en üst 4 katmanında güvenlik servisi sunmak mümkün.
- Uygulama = sadece o uygulama için
- İletim = kullanan tüm uygulamalar
- Ağ = kullanan tüm iletim & uygulama protokolleri
- Veri Bağlantı = kullanan tüm ağ & iletim & uygulama protokolleri – «blanket coverage»
- Önce uygulama katmanından başlayıp aşağı doğru ineceğiz.

Giriş



- Uygulama katmanı içinden özel bir uygulama seçelim – e-mail.
- Sonra iletim katmanı için TLS, ağ katmanı için IPsec ve son olarak veri bağlantı katmanı için IEEE 802.11 wireless LAN için güvenlik konularına bakacağız.
- Güvenlik servisi neden Internette birden çok katmanda yapılıyor?
- Tek bir katmanda örneğin ağ katmanında önlemler alınsa yetmez mi?
- Sadece client ve server uygulama kodu gerektiren PGP, Internette kullanılan ilk güvenlik teknolojilerinden birisi

Giriş

Doç.Dr.İpek Abasıkeleş-Turgut



- Alice ve Bob birbirlerine e-mail yolluyorlar. Amacımız bunu saldırıgandan korumak.
- Secure e-mail sistemi tasarlamadan önce hangi güvenlik özelliklerinin istendiğine bakalım.
- İlk olarak confidentiality – gizlilik şart.
- İkincisi gönderici için authentication - sender authentication.
- Üçüncü istenilen özellik ise message integrity – mesajın modifiye edilmemiş olması lazım.
- Son olarak da receiver authentication sağlaması lazım. Yani mesajın sadece Bob’a gitmesi.

E-Mail Güvenliği



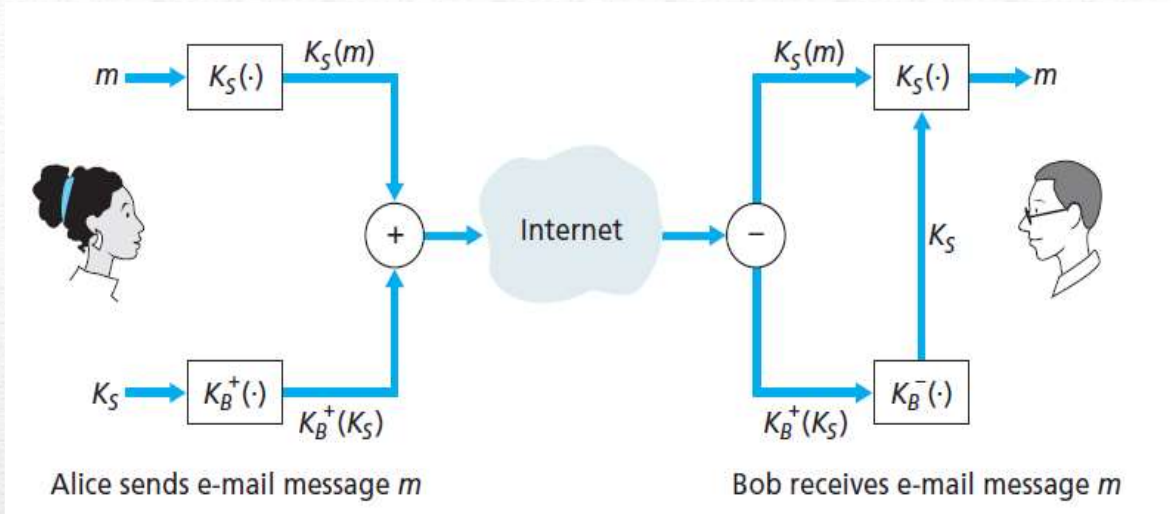
- Sırasıyla sorunları çözelim.
- Önce gizlilik.
- En basit yol, Alice'in mesajı DES veya AES gibi bir simetrik key teknolojisi ile şifrelemesi ve Bob'un da bunu çözmesi.
- Daha önce söylendiği gibi, eğer yeterince uzun bir şifre kullanılırsa ve sadece Alice ve Bob bu anahtara sahip olursa saldırganın mesajı çözmesi oldukça zor olur.
- Burada sorun simetrik anahtarı Alice ve Bob arasında dağıtmak.

E-Mail Güvenliği

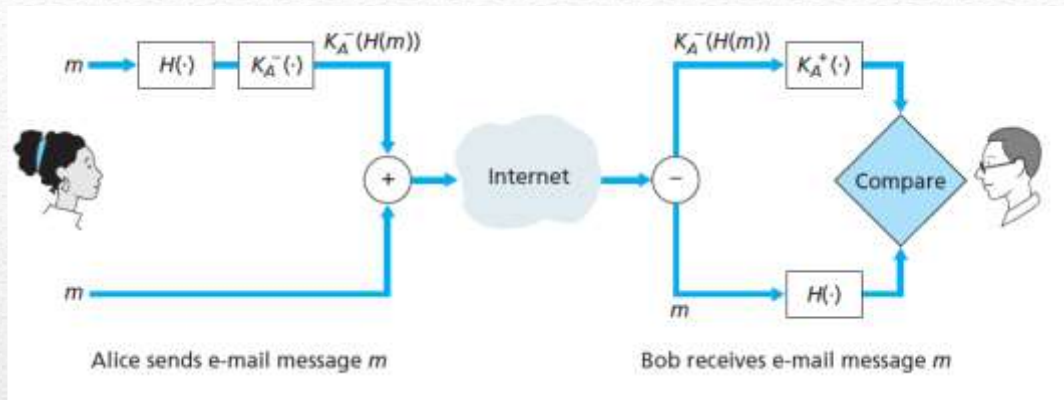


- Alternatif yaklaşım public key teknolojisi – örneğin RSA kullanmak.
- Burada Bob kendi public anahtarını yaratır ve bunu bir public key sunucusu veya kendi web sitesi üzerinden ilan eder.
- Alice mesajını Bob'un public anahtarı ile şifreler ve e-mailine yollar.
- Bob bunu kendi gizli anahtarı ile çözer.
- Eğer Alice, elindeki public anahtarın Bob'un olduğuna emin olursa, bu yöntem ile gizlilik güçlü şekilde sağlanır.
- Burada sorun, public key şifrelemesinin özellikle uzun mesajlarda kullanışsız olması.

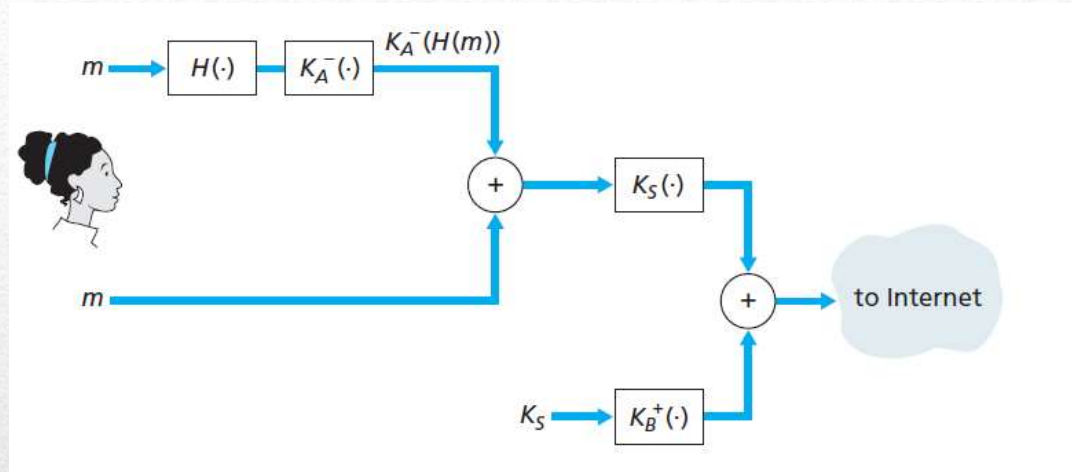
E-Mail Güvenliği



- Efektiflik probleminin üstesinden gelmek için oturum anahtarları kullanılsın.
- Şekildeki gibi, Alice, rasgele bir simetrik session key seçer – K_S .
- Mesajını simetrik anahtar ile, simetrik anahtarını ise Bob'un public anahtarı K_B^+ , ile şifreler.
- Şifrelenmiş mesaj ve şifrelenmiş simetrik anahtar içeren paketini Bob'un mailine gönderir.
- Şekilde yuvarlak içinde + işlemi birbirine bağlama, - ise ayırtırmadır.
- Bob paketi alınca gizli anahtarını kullanarak önce simetrik anahtarı elde eder sonra da onu kullanarak mesajı elde eder.
- Böylece gizlilik sorunu çözülmüş olur.



- Hem gönderici kimlik doğrulaması hem de mesaj bütünlüğünü nasıl sağlayacağımıza bakalım.
- dijital imza ve mesaj özeti – digest – kullanılabilir.
- Alice, mesajına H hash fonk.nu (örneğin MD5) uygular ve böylece m mesajının özetini elde eder.
- Sonra bu özetini gizli anahtarı ile şifreler K_A^- , ve böylece dijital imzasını yaratır.
- Ardından orijinal mesajı – şifrelenmemiş - imzası ile birleştirerek paketini oluşturur.
- Bob'un mailine yollar.
- Bob paketi alınca Alice'in public anahtarını kullanarak mesaj özeti elde eder.
- Bunu, gelen mesajın kendisindeki hash fonk.a sokulması ile elde ettiği ile kıyaslar.
- her ikisi de aynı ise, bunun Alice'den geldiğini ve değiştirilmediğini anlar.



- diyelim ki bir e-mail sistemi gizlilik, gönderici kimlik doğrulama ve mesaj bütünlüğünü sağlayacak şekilde tasarlanmış olsun.
- Yukarıda verilen iki şeklin kombine edilmesi ile olur.
- Alice önce bir başlangıç (preliminary package) paketi yaratır – orijinal mesajı ve bu mesajın dijital imzalı özetini içeren.
- Sonra bu başlangıç paketini gizlilik adımlarına göre düzenler.

- Bob paketi aldığıında önce gizlilik için çözer, sonra diğer adımları yapar.
- Böylece gizlilik, gönderici kimlik doğrulama ve mesaj bütünlüğü sağlandı.
- Hem Alice hem Bob burada public key crp. 2 kere kullanıyor. 1 – Kendi gizli anahtarı ile, 2 – Diğerinin public anahtarı ile.
- Secure e-mail tasarımı, kullanıcılarına yeterli güvenliği sağlar.
- Ancak çözülmesi gereken önemli bir sorun var.
- Alice Bob'un, Bob ise Alice'in public anahtarlarını elde etmeli.
- Bu dağıtım sırasında saldırgan kendi yarattığı anahtarını kullanmamalı – pizzacı örneğindeki gibi.
- Bunun için popüler yaklaşım – public anahtarları sertifikalandırmak – CA

E-Mail Güvenliği

- Phil Zimmermann tarafından 1991’de yaratılmıştır.
- Bugün artık e-mail için standart hale gelmiştir.
- OpenPGP standardı IETF tarafından oluşturulmuş ve RFC 4880’de tanımlanmıştır.
<http://openpgp.org/>
- Versiyonuna bağlı olarak mesaj özeti için MD5 ve SHA’yı kullanır.
- Simetrik key şifreleme için CAST, triple-DES, veya IDEA kullanır.
- Public key şifreleme içinse RSA kullanır.
- PGP kurulduğu anda, yazılım kullanıcı için bir public anahtar çifti üretir.
- Public anahtar kullanıcı web sitesinde veya public key sunucularında yer alabilir.
- Private anahtar ise password kullanımı ile saklanır.
- Kullanıcı gizli anahtarına her erişiminde password girmelidir.
- PGP ile mesajı dijital imzalama, şifreleme ve hem dijital imzalayıp hem de şifreleme imkanına sahip olursunuz.

PGP - Pretty Good Privacy (PGP)



- Şekilde PGP imzalı mesaj görülmektedir. Bu mesaj, MIME başlığından sonra görülür. – SMTP’nin kullandığı format
- mesajdaki şifreli veri yani $KA - (H(m)) = \text{mesajın özetinin dijital olarak imzalı hali}$.

```
-----BEGIN PGP SIGNED MESSAGE-----  
Hash: SHA1  
Bob:  
Can I see you tonight?  
Passionately yours, Alice  
-----BEGIN PGP SIGNATURE-----  
Version: PGP for Personal Privacy 5.0  
Charset: noconv  
yhHJRhhGJGhg/12EpJ+lo8gE4vB3mqJhFEvZP9t6n7G6m5Gw2  
-----END PGP SIGNATURE-----
```

PGP - Pretty Good Privacy (PGP)



- şekilde gizli bir PGP mesajı var.
- Bu da MIME başlığından sonra görülür.
- Gizli e-mail mesajında plaintext mesaja dahil edilmez.
- Gönderici hem gizlilik hem bütünlük istiyorsa, ikinci tipte mesaj yollamalı.

```
-----BEGIN PGP MESSAGE-----  
Version: PGP for Personal Privacy 5.0  
u2R4d+/jKmn8Bc5+hgDsqaewsDfrGdszX68liKm5F6Gc4sDfcXyt  
RfdS10juHgbcfDssWe7/K=lKhnmikLo0+1/BvcX4t==Ujk9PbcD4  
Thdf2awQfgHbnmKlok8iy6gThlp  
-----END PGP MESSAGE
```

PGP - Pretty Good Privacy (PGP)



Katmanlarda Güvenlik Önlemleri – İletim Katmanı

Kaynak : James F. Kurose & Keith W. Ross - Computer networking: A top-down approach 6th Edition



- Transport Layer Security
- QUIC protokolünün bir parçası olarak UDP üzerinde de çalışıyor (yeni HTTP versiyonlarında)
- TLS (TLS'in atası) Netscape tarafından tasarlanmıştır.
- Popüler Web browserları ve sunucuları, neredeyse tüm e-ticaret siteleri tarafından desteklenmiştir.
- Her yıl TLS üzerinden milyarlarca dolar harcanmaktadır.
- Diyelim ki Internet üzerinden kredi kartınızla birşeyler aldınız.
- Browser'ınız ve sunucu arasında muhtemelen TLS üzerinden iletişim kuruldu.
- Eğer linkiniz http değil de https ise tarayıcınız TLS kullanıyor demektir.
- Mevcut TLS versiyonu – RFC 8446
- Eski versiyonlardan farkı – hızlı handshake ve eski kriptografik algoritmaların kaldırılması

Güvenli TCP & HTTP Bağlantıları: TLS



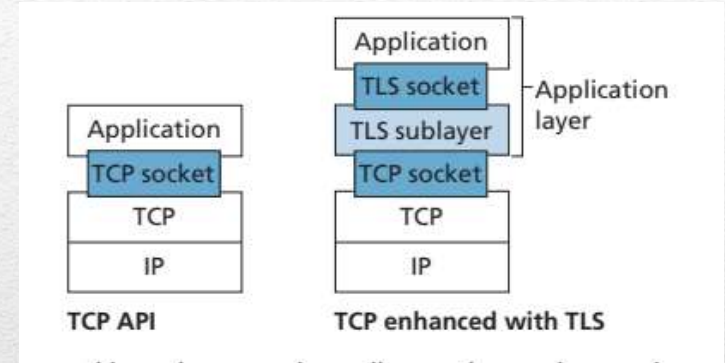
- neden TLS'e ihtiyaç olsun?
- Tipik bir internette ticaret senaryosuna bakalım:
- Bob, Web'de geziniyor ve Alice'in parfüm satan sitesine rastlıyor.
- Sitede bir form var.
- Bob adresini, adını, istediği parfüm markasını ve miktarını yazarak (hatta kredi kartı numarasını) sipariş verebiliyor.
- Eğer herhangi bir güvenlik önlemi alınmazsa şu sorunlar olabilir:
- Eğer gizlilik – confidentiality için önlem alınmazsa saldırgan Bob'un siparişini dinleyerek kart bilgilerini elde edebilir. Ardından kart bilgilerini kendisi için kullanabilir.
- Eğer integrity sağlanmazsa, saldırgan Bob'un siparişini değiştirebilir – mesela istediğinin 10 katı kadar parfüm sipariş etmiş gibi gösterebilir.
- Eğer sunucu kimlik doğrulaması kullanılmazsa, saldırgan kendisini Alice olarak göstererek bir site tasarlanmış olabilir ve böylece Bob'un tüm bilgilerine erişecek olabilir.

Güvenli TCP Bağlantıları:

TLS



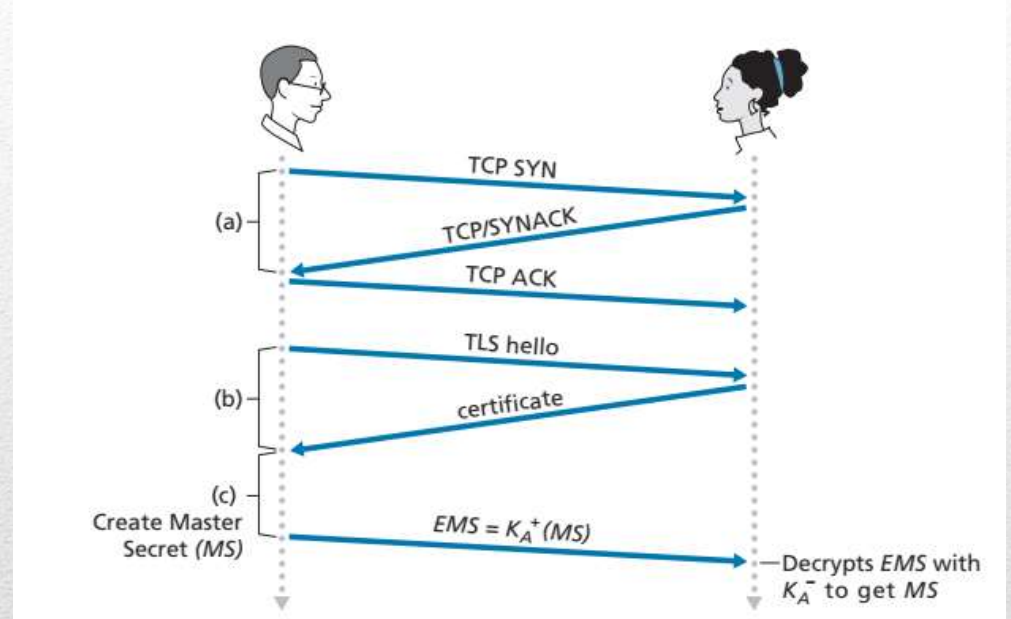
- TLS, TCP'yi geliştirerek gizlilik, integrity, server ve client authentication sorunlarını çözer. (Şu anda UDP üzerinden de prosesler arası güvenli iletişim – QUIC içinde)
- TLS genellikle HTTP iletimlerine güvenlik sağlar.
- Ancak TCP kullanan herhangi bir uygulama da aslında faydalanabilir.
- TLS, TCP'nin API'sine benzer basit bir Application Programmer Interface (API) sağlar.
- Eğer bir uygulama TLS kullanmak isterse TLS class/library'lerini include eder.
- Her ne kadar developer gözünden teknik olarak uygulama katmanındaymış gibi görünse de, TLS, TCP servislerine güvenlik hizmeti sunar.



- TLS'in basitleştirilmiş bir versiyonuna bakarak neden ve nasıl TLS sorularına yanıtları aramaya çalışalım.
- Bu basit TLS versiyonuna “almost-TLS” (hemen hemen, neredeyse) diyelim.
- Ardından ayrıntıları doldurarak gerçek TLS'den bahsedebiliriz.
- Almost-TLS (ve TLS), 3 aşamadan oluşur: handshake (el sıkışma), key derivation (anahtar türetme) ve data transferi
- Bob (client) ve Alice (server) arasında bir iletişimde bu 3 aşamanın nasıl olduğunu tanımlayalım.
- Alice'in kendi public/private anahtar çiftine sahip olduğunu ve public anahtarına kimliğini bağlamak için sertifikası olduğunu varsayıyoruz.

«The Big Picture»

- Bob'un (a) Alice ile TCP bağlantısı kurması, (b) Alice'in gerçekten Alice olduğundan emin olması ve (c) Alice'e asıl gizli anahtarını yollaması lazım.
- Bu ana gizli anahtar, hem Alice hem Bob tarafından kullanılacaktır ve TLS oturumunda gerekli olan tüm simetrik anahtarları üretecek olan ana anahtar olacaktır.



1 – Handshake

- Aslında, MS, Bob ve Alice tarafından artık bilindiğine göre, tüm ardışık şifreleme ve veri bütünlüğü kontrol işlemlerinde simetrik oturum anahtarı olarak kullanılabilir.
- Ancak, Alice ve Bob için hem farklı cryptolojik anahtarlar hem de farklı şifreleme ve integrity kontrol anahtarı kullanmanın genellikle daha güvenli olduğu varsayılır.
- Dolayısı ile hem Alice hem Bob MS'yi kullanarak 4 farklı anahtar üretirler.
- Mesela basitçe anahtarı 4'e bölebilirler (gerçek TLS'de daha karmaşık).
- Bu aşamadan sonra hem Alice hem de Bob 4 anahtara sahip olur: 2 MAC (verinin bütünlüğünün sağlanması için), 2 encry. anahtarı (verinin şifrlenmesi için).
- HMAC – standart hash fonk.

2 – Key Derivation

- E_B = session encryption key for data sent from Bob to Alice – Bob’dan Alice’e giden veri için şifreleme oturum anahtarı
- M_B = session MAC key for data sent from Bob to Alice - Bob’dan Alice’e giden veri için MAC oturum anahtarı
- E_A = session encryption key for data sent from Alice to Bob
- M_A = session MAC key for data sent from Alice to Bob

2 – Key Derivation



- Alice ve Bob şu anda aynı 4 oturum anahtarına sahip: (E_B , M_B , E_A , ve M_A).
- Artık TCP bağlantısı üzerinden birbirlerine güvenli veri yollayabilirler.
- TCP, byte-stream bir protokol olduğu için, TLS için doğal yaklaşım, uygulama verisini olduğu gibi, hemen şifrelemek ve şifreli veriyi TCP'ye geçirmektir.
- Sorun?
- Sorun – Tüm verinin integrity doğrulaması için TCP oturum sonuna kadar beklemek lazım.
- Bu sorunu çözmek için TLS, data stream'i **record** adı verilen kısımlara böler.
- Her record'a bütünlüğü kontrol etmek amacıyla bir MAC atar ve record+MAC'i şifreler.
- MAC'i oluşturmak için Bob, record datasını input olarak alır ve M_B ile hash fonk.a sokar
- Ardından record+MAC'den oluşan paketini şifrelemek için, oturum şifreleme anahtarını E_B kullanır.
- Bu şifreli paket, internete gönderilmek üzere TCP'ye geçer.

3 – Data Transferi



- Güvenli mi?
- sıra numarası kullanımı
- TLS şu şekilde çalışır: Bob bir sıra numarası sayıcı kullanır: 0'dan başlar ve gönderdiği her TLS record'u için 1 artırır.
- Bob gerçekte record'un kendisine bir sıra nosu koymaz.
- Bunun yerine MAC hesabına sıra numarasını da dahil eder.
- Dolayısı ile artık $MAC = \text{hash}(\text{data} + M_B + \text{mevcut sıra nosu})$
- Güvenli mi?

3 – Data Transferi



- Hem TLS, hem almost-TLS için.
- Tip, versiyon, uzunluk, veri ve MAC alanı var.
- İlk 3 kısım şifreli değil.
- Tip alanında, handshake mesajı mı, uygulama verisi taşıyan mesaj mı olduğu belirtilir.
- Aynı zamanda TLS bağlantısını sonlandırmada da kullanılır.
- Alıcı taraftaki TLS, uzunluk alanını kullanarak, gelen TCP byte stream'i içinden TLS record'larını ayırır.
- Versiyon zaten belli.



TLS Record



- almost-TLS'i inceledik. Gerçek TLS ile ilgili ayrıntılara geçelim.
- **TLS Handshake**
- TLS, Alice ve Bob'a spesifik bir simetrik anahtar algoritması, spesifik bir public-key algoritması kullanmayı zorunlu kılmaz.
- Bunun yerine TLS oturumunun başında, handshake aşamasında, Alice ve Bob'un kriptolojik algoritmalar üzerinde anlaşmalarını sağlar.
- Ayrıca, handshake aşamasında, Alice ve Bob birbirlerine “nonce” yollarlar.
- Bunun sayesinde oturum anahtarları üretilir: (E_B , M_B , E_A , ve M_A).

Daha Geniş Pencere



- Gerçek TLS handshake aşamaları:
- 1- İstemci, kendi nonce'i ile birlikte desteklediği kriptolojik algoritmaların bir listesini yollar.
- 2- Server, listeden bir simetrik key algoritma (mesela AES), bir public key algoritma (mesela spesifik bir anahtar uzunluklu RSA) ve HMAC algoritması (MD5 veya SHA-1) seçer. İstemciye tercihleri, sertifikası ve kendi “nonce” ile döner.
- 3 – İstemci sertifikayı doğrular, server'ın public anahtarını ayırır ve Pre-Master Secret (PMS) üretir. PMS'yi server'ın public anahtarı ile şifreler ve şifreli PMS'yi sunucuya yollar.
- 4 – Aynı key türetme fonksiyonunu kullanarak (TLS standartlarında tanımlı) hem istemci hem de sunucu bağımsız olarak PMS ve nonce'lar ile MS'yi hesaplarlar. Ardından MS, dilimlenerek 2 encyp, 2 MAC anahtarına dönüşür. Ayrıca eğer seçilen simetrik cipher, CBC kullanıyorsa (mesela 3DES ve AES'te olduğu gibi) 2 adet Initialization Vector (IV) – her biri bağlantının bir ucu için - de MS'den elde edilir. Böylece istemci ve sunucu arasındaki tüm mesajlar hem şifrelenecek, hem de kimliği doğrulanacaktır (MAC ile)
- 5 – İstemci, tüm handshake mesajları için bir HMAC yollar.
- 6 – Server, tüm handshake mesajları için bir HMAC yollar.
- HMAC kullanım amacı?

Daha Geniş Pencere

Doç.Dr.İpek Abasıkeleş-Turgut



- handshake mesajlarını “tamper” atağından koruma
- Adım 1’de algoritmalar listesi şifresiz yollanıyor => değiştirilebilir.
- Örn. Güçlüleri silme, zayıf bırakma vb.
- Hesaplanan vs alınan HMAC arasında tutarsızlık varsa bağlantı iptal edilir.
- Adım 2’deki nonce neden kullanılıyor?
- Sıra sayısı ile zaten segment replay ataklarına karşı koruma sağlanmıyor mu?

Daha Geniş Pencere



- Evet ama connection replay ataklarına karşı koruma yok
- Saldırgan, Bob'un bir önceki gün yolladığı mesajların aynısını yollayabilir – aynı sırada ise integrity testinden geçer.
- Her TCP oturumu için farklı nonce ile played-back TLS kayıtlarına karşı savunma gerçekleşir.

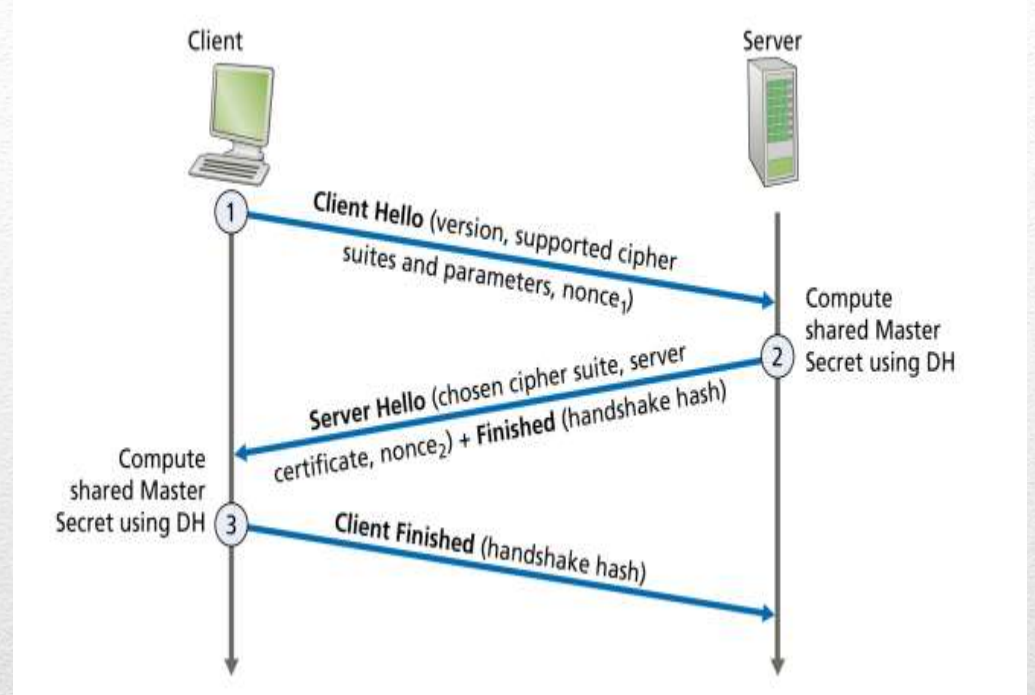
Daha Geniş Pencere



- Oturum sonlandırma?
- TCP FIN
- **truncation** atağına açık -> saldırgan sonlandırabilir.
- Çözüm?
- Record içindeki tip alanı
- Tip alanında TLS oturumunun bitip bitmediği belirtilir.
- Sorun?

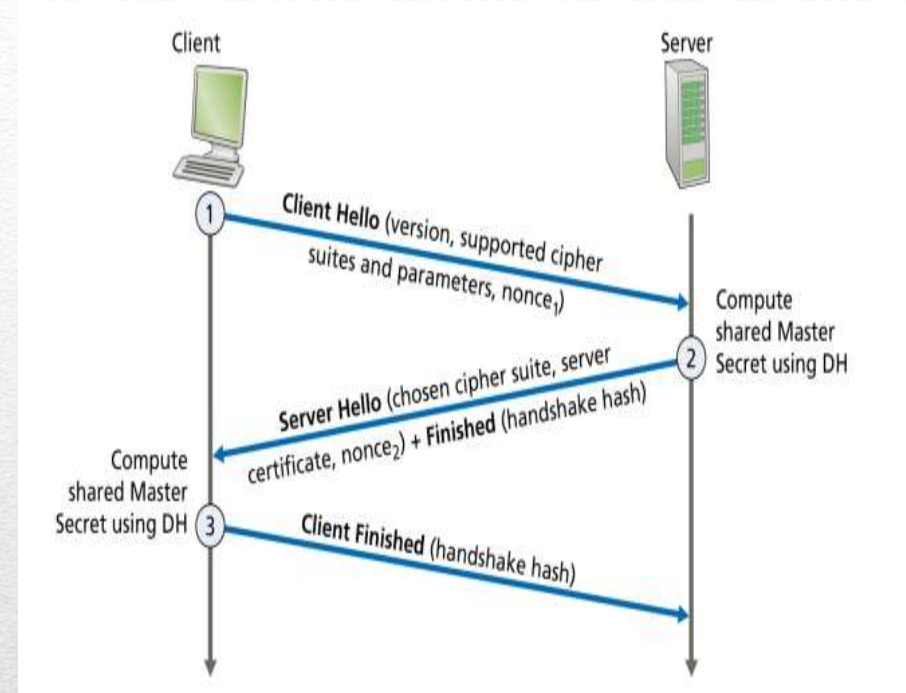
Bağlantının Kapatılması

- Handshake aşaması,
- 1 - Client Hello – TLS versiyon no, 3 cipher süite içinden birini seçmesi için liste, nonce sayısı
- Cipher süite içeriği :
- a - MS üretmek için anahtar değiştirme tekniği, MS'den ek anahtarlar üretmek için kullanılan SHA algoritması
- b – AEAD (Authenticated Encryption with Associated Data) – data transfer için ayrı ayrı şifreleme ve bütünlük yerine tek algorithmada ikisi de var. MS anahtarı üretmek için DH – diffie hellman'ın bir varyasyonu kullanılıyor.

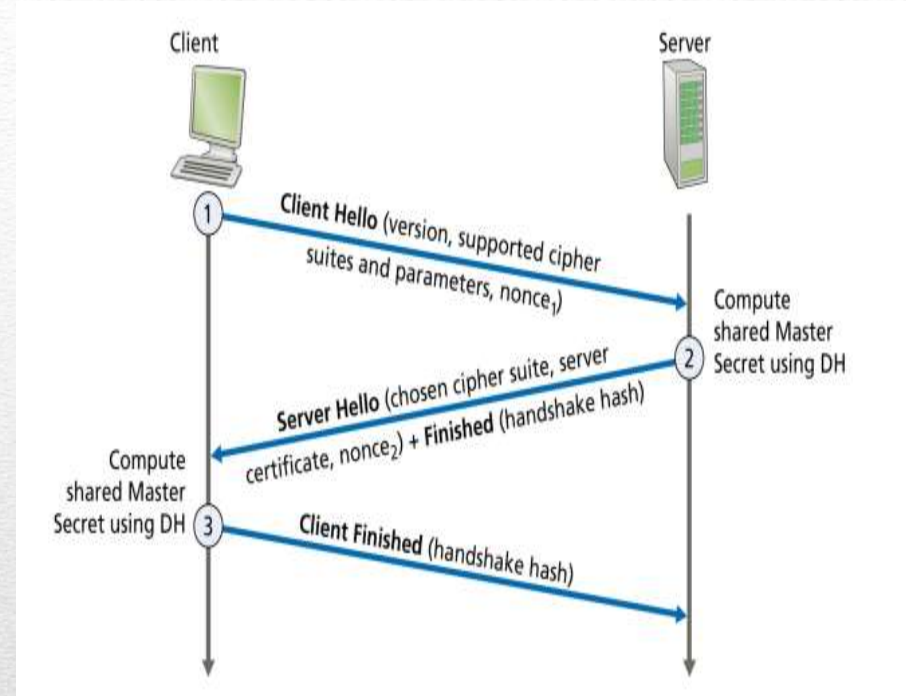


TLS 1.3

- 2 - Server Hello, Finished:
Server hello'su sunucunun sertifikasını, seçilen cipher süiti ve nonce'ı içerir. Client hello mesajında DH için gereken public değerler olduğu için sunucu kendi public ve private DH değerlerini üretir ve ardından paylaşılan MS anahtarını – istemcinin public değeri ve kendi private DH değerinden üretir. Kendi public DH değerini istemciye gönderir.



- 3 - Client, Finished: İstemci imzayı ve sertifikayı doğrular ve kendi private DH değeri ile sunucunun public DH değerini kullanarak MS üretir.
- Finish mesajında imzalı hash yollar – Client ve Server hello mesajları için
- Finish mesajı şifrelidir ve MS tarafından üretilen anahtar kullanılır.



Katmanlarda Güvenlik Önlemleri – Ağ Katmanı

*Kaynak : James F. Kurose & Keith W. Ross - Computer
networking: A top-down approach 6th Edition*



- IPsec, ağ katmanında güvenlik sağlayan protokoldür.
- 2 ağ-katman aygıtı arasında (host veya router) hareket eden IP datagramlarının güvenliğini sağlar.
- Birçok şirket IPsec kullanarak public Internet üzerinde çalışan VPN'lerini yaratır.
- Ağ katmanında gizlilik sağlamak ne demek?
- Gönderen aygıt alıcıya gönderdiği datagramın veri kısmını şifreler.
- Bu veri içinde üst katman verileri yer alır yani TCP/UDP segmenti veya ICMP mesajı vb.
- Eğer ağ katmanında güvenlik hizmeti sağlanırsa - “blanket coverage” sağlanır. Neden?

Ağ Katmanında Güvenlik: IPSec ve VPNs (Virtual Private Networks)



- Gizliliğin yanında ağ-katman protokolü diğer güvenlik hizmetlerini de sağlayabilir.
- Örneğin kaynak authentication ile alıcının kaynağı doğrulaması sağlanır.
- Ayrıca integrity sağlanarak datagramı değiştirilmediğinden emin olunur.
- Ayrıca replay-attack önlenabilir – yani eğer saldırgan duplicate datagram enjekte ederse Bob tespit edebilir.

Ağ Katmanında Güvenlik: IPSec ve VPNs (Virtual Private Networks)

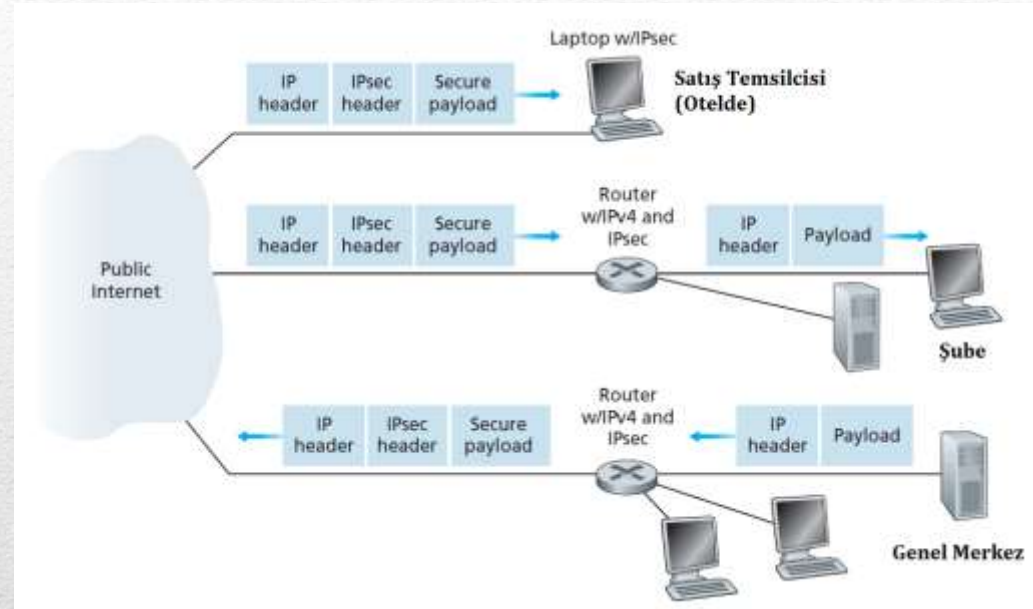


- Geniş çaplı şirketler aygıtları ve sunucuları arasındaki iletişimin güvenli olması amacıyla kendi IP ağlarının olmasını ister.
- Bu amaçla, stand-alone bir fiziksel ağ kurulabilir.
- Yani içerisinde router, link, DNS alt yapısı vb. herşeyin dahil olduğu bir sistem.
- Bu sistem public Internetten tamamen ayrı/bağımsız olur.
- Bu ağlara private networks denilir.
- Bu ağların kurulum maliyeti çok yüksektir.
- Bu nedenle bunun yerine şirketler, public Internet üzerinde VPN oluşturmayı tercih ederler.

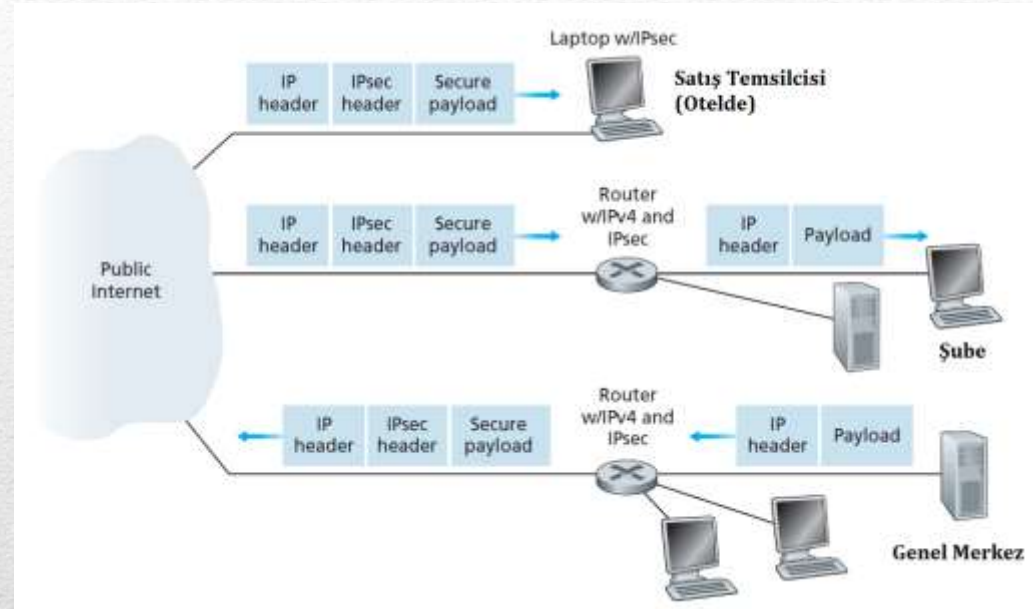
IPsec ve Virtual Private Networks (VPNs)



- VPN sayesinde şirket içi trafik, fiziksel olarak internetten bağımsız bir ağ üzerinden değil de public internet üzerinden akar.
- Gizliliği sağlamak içinse şirket içi trafik public internete çıkmadan önce şifrelenir.
- basit bir VPN yapısı şekilde görülmektedir: Örneğin bir şirket yapısı, bir ana merkezden, 1 şubeden ve Internet'e otel odasından bağlanan bir satış temsilcisinden oluşabilir.
- Bu tür bir VPN'de, genel merkezde veya şubede yer alan 2 aygıt haberleşmek isterse, IPsec kullanmadan standart IPv4 kullanabilirler.
- Ancak 2 aygıtın datagramları public Internet üzerinden geçecekse, IPsec kullanmaları gerekir.



- örneğin genel merkezden satış temsilcisine bir datagram gidecek olsun.
- Genel merkezdeki gateway router, standart IPv4 datagramını IPsec datagramına çevirir ve internetten yollar.
- IPsec datagramında standart IPv4 başlığı yer alır.
- Bunun amacı internet üzerindeki router'ların yönlendirme işini yapabilmesi içindir.
- IPv4 veri kısmı ise aslında IPsec datagramıdır – başlık + data.
- Data kısmı şifrelenmiştir.
- IPsec datagramı satış temsilcisinin laptop'una ulaştığında veri kısmının şifresi çözülür ve üst katman bilgisi elde edilir.

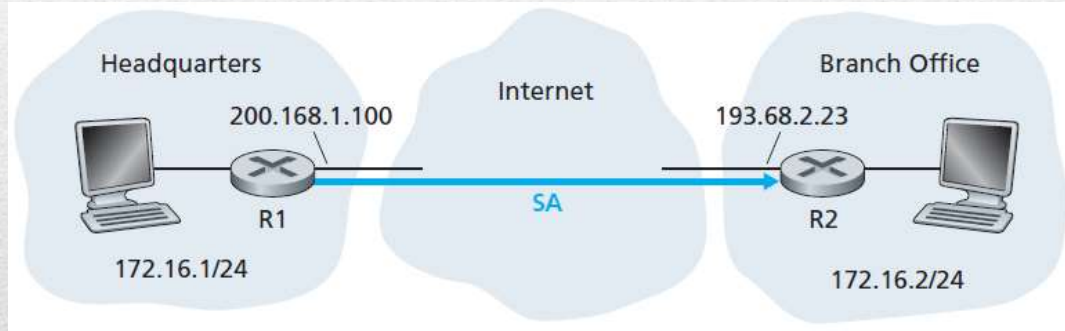


- Ipsec oldukça geniş kapsamlı.
- 10'dan fazla RFC'de tanımlı.
- 2 önemli RFC 4301 ve 6071.
- Genel IP güvenlik altyapısı ilkinde, Ipsec protokol suiti ise 2.de mevcut.
- IPsec protokol suiti 2 temel protokol içerir: Authentication Header (AH) protocol ve Encapsulation Security Payload (ESP) protocol
- Kaynak IPsec aygıtı (host veya router), hedefe (host veya router) güvenli datagram göndermek istediğinde ya AH ya da ESP protokolünü kullanır.
- AH, authentication ve integrity sağlarken, gizlilik sağlamaz.
- ESP ise hepsini sağlar. O nedenle ESP daha yaygın kullanılır.

AH ve ESP Protocols



- IPsec datagramları aygıtlar arasında gönderilmeden önce kaynak ve hedef arasında ağ-katman mantıksal bağlantısı yaratılır.
- Bu mantıksal bağlantıya security association (SA) denilir.
- SA, basit bir mantıksal bağlantıdır – kaynak ve hedef arasında tek yönlü.
- Eğer her iki taraf da birbirine datagram gönderecekse o zaman her biri bir yön için toplam 2 SA kurulmalıdır.



Security Associations (Güvenlik Bağlantısı)

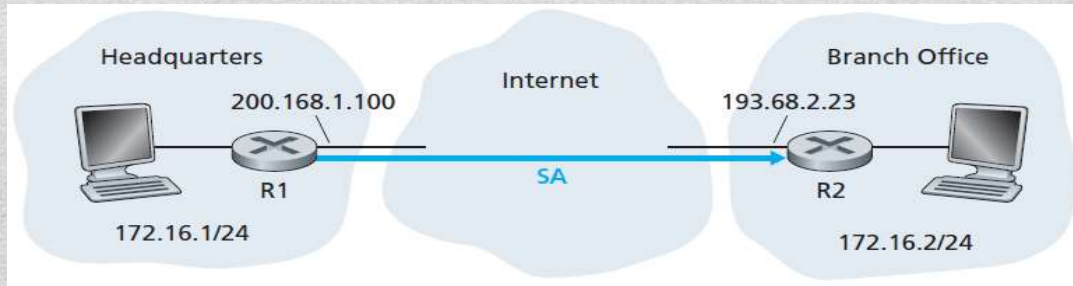
- Diyelim ki az önceki örnekte gördüğümüz şirkette 1 genel merkez, 1 şube ve n adet satış temsilcisi olsun.
- Diyelim ki genel merkez ve şube arasında çift yönlü IPsec ve genel merkez ve satış temsilcileri arasında da çift yönlü IPsec olsun.
- Bu durumda VPN, kaç adet SA içerir?
- Gateway router'lar veya satış temsilcisi laptopları üzerinden akan her trafik IPsec değildir.
- Mesela genel merkezdeki bir kişi, Google veya Amazon sunucularına bağlanacak.
- Her ne kadar public internetten geçse de IPsec değildir.

Security Associations

(Güvenlik Bağlantısı)



- SA yapısında ne var? Genel merkezden, şubeye doğru olan SA'yı düşünelim. Genel merkezin router'ından - R1- çıkan datagram, şubenin router'ına – R2 – varana kadarki yoldaki içeriğe bakalım.
- R1 router'ı bu SA için durum bilgisi sağlar. Bu bilgiler;
- 32 bitlik SA identifier = SPI – Security Parameter Index,
- SA'nın kaynağı (burada 200.168.1.100) ve hedefi (burada 193.68.2.23)
- Kullanılan şifreleme yöntemi (Örneğin CBC'li 3DES)
- Şifreleme anahtarı (encryption key)
- Bütünlük kontrol yöntemi (integrity check) – Örn. MD5'li HMAC
- kimlik doğrulama anahtarı (authentication key)

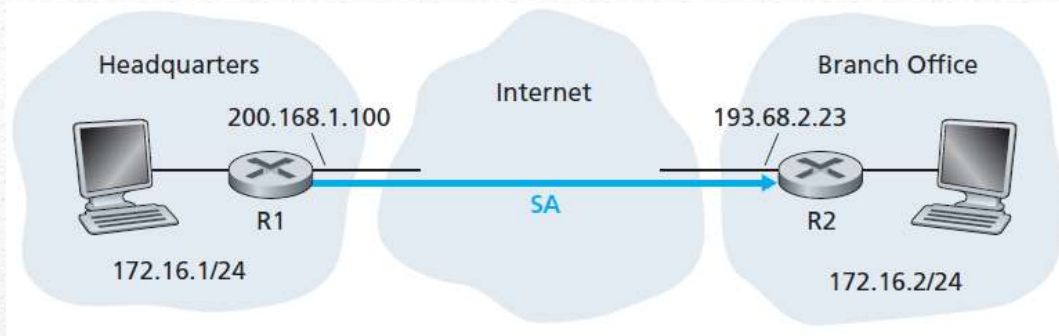


- R1, bu SA üzerinden ne zaman IPsec datagramı yollamak isterse, o zaman datagramı nasıl şifreleyeceğine ve doğrulayacağına karar vermek için bu durum bilgisine erişir.
- Benzer şekilde R2 de bu durum bilgisini kullanarak SA'dan gelen tüm datagramların doğrulama ve şifre çözme işlemlerini yapar.
- Çoğunlukla bir IPsec aygıtı (router veya host) çok sayıda SA için durum bilgisi sağlar.
- IPsec aygıtı bu durum bilgilerinin hepsini Security Association Database (SADB) inde saklar.
- Bu, aygıtın işletim sistem çekirdeği içindeki bir veri yapısıdır.

Security Associations

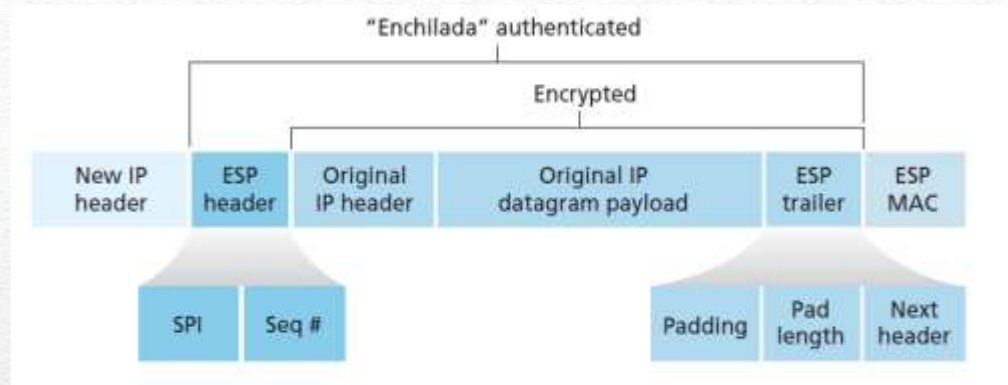
(Güvenlik Bağlantısı)





- IPsec 2 farklı paket formuna sahiptir: tunnel mod ve transport mod.
- Tunnel modu VPN ler için daha uygundur ve transport moda göre daha geniş kullanım alanına sahiptir.
- Bu derste tunnel moduna odaklanacağız.
- Diyelim ki R1, genel merkezdeki bir host olan IP = 172.16.1.17 ‘den bir IPv4 datagramı aldı.
- Bu datagramın hedefi şubede yer alan bir host= IP = 172.16.2.48.
- R1, bu orijinal IPv4 datagramını IPsec datagramına dönüştürmek için şunları yapar:

IPsec datagram



- IPv4 datagramının orijinal veri sonrası başlık kısmını “ESP trailer” alanına atar.
- Sonucu SA’da belirtilen algoritma ve anahtarı kullanarak şifreler.
- Şifrelenen kısmın önüne “ESP header” atar. Bu son pakete “enchilada” denilir. *The whole enchilada = tamamı, bütünü*
- Tüm kısım için SA’da belirtilen algoritma ve anahtar kullanılarak MAC üretilir.
- MAC, enchilada sonuna eklenir. İkisi birlikte yeni payload olur.
- Bu payload için, klasik bir IP başlığı oluşturulur ve önüne eklenir.

- Son IPsec datagramı aslında klasik bir IPv4 datagramıdır.
- IPv4 başlığı ve payload kısmı vardır.
- orijinal Ip datagramının kaynak IP adresi 172.16.1.17, hedef IP adresi 172.16.2.48'dir.
- IPsec datagramında bu adresler şifreli şekilde, IPsec paketinin payload kısmına dahil edilir.
- Ancak yeni IP paketinin kaynak ve hedef IP adresleri farklıdır: tünelin başında ve sonunda yer alan router arayüzlerinin IP'si: 200.168.1.100 ve 193.68.2.23.
- Ayrıca bu yeni IPv4 teki başlık alanı protokol numarası TCP, UDP, veya SMTP değil, 50'dir = ESP protokolü kullanan IPsec datagramı demektir.

IPsec datagram



- R1, IPsec datagramını public Internet'e gönderdikten sonra birçok router'dan geçer ve R2'ye ulaşır.
- Datagram sıradan bir IPv4 datagramı gibi görüldüğü için yolda sorunla karşılaşmaz.
- Hedef IP adresi, R2 olduğu için – en dış başlıkta – datagram R2'ye ulaşır.

IPsec datagram



- ençilada'ya ayrıntılı bakalım.
- ESP trailer içinde 3 alan var: padding, pad length ve next header.
- Block cipher'lar, mesajın uzunluğunun, blok uzunluğu x bir integer sayı olmasını gerektirirler.
- padding yani anlamsız byte'lar ile doldurma işlemi, mesajın blok uzunluğunun tam katı olmasını sağlamak içindir.
- pad-length ise ne kadar padding yapıldığını gösterir.
- next header ise, payload data alanındaki verinin cinsidir – mesela UDP gibi.
- orijinal IP datagramı ile ESP trailer kısmı birleştirildikten sonra şifrelenir.

IPsec datagram



- Şifrelenen kısmın başında yer alan ESP başlığı ise SPI ve sıra sayısı olmak üzere 2 alandan oluşur.
- Alıcı aygıt, uygun kimlik doğrulama/şifre çözme algoritmalarını ve anahtarlarını bulmak için SADB 'yi SPI ile indexler.
- Sıra sayısı ise replay ataklarına karşıdır.
- Gönderici aygıt aynı zamanda bir authentication MAC atar.
- Gönderici aygıt tüm ençilada üzerinden MAC hesaplar.
- Bu hesaplama için ençilada'ya bir gizli MAC anahtarı ekler ve sonucun sabit uzunluklu hash'ini hesaplar.

IPsec datagram

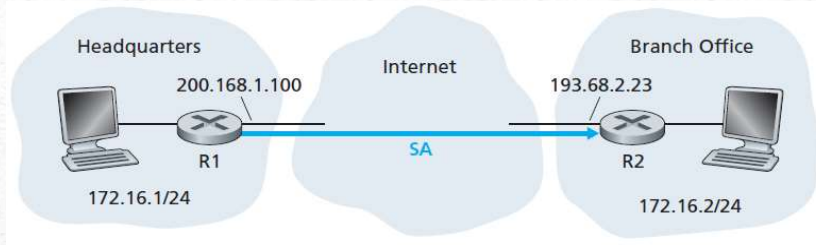


- R2, IPsec datagramını aldığıında, hedef IP'nin kendisi olduğunu görür ve datagramı işleme koyar.
- Protokol alanı 50 olduğu için, datagramı IPsec ESP'ye göre işler.
- Önce enchiladaya bakar.
- SPI'yi kullanarak hangi SA'ya ait olduğunu bulur.
- Ardından tüm enchilada için MAC hesaplar ve bunu ESP MAC alanı ile karşılaştırır.
- Eğer sorun yoksa, enchilada'nın R1 den geldiğini ve yolda değiştirilmediğini anlar.
- 3. olarak sıra no alanını kontrol eder ve datagramın yeni mi yoksa replayed bir datagram mı olup olmadığını anlar.
- 4. olarak şifreli kısmı şifre çözme algoritması ve SA'ya atanmış anahtar ile çözer.
- 5.olarak padding'i kaldırır ve orijinal IP datagramına erişir.
- Son olarak da orijinal datagramı şubedeki gerçek alıcısına iletir.

- İlk bakışta göze çarpmayan ama önemli olan bir sorun var?
- R1, merkezin ağından, şifrelenmemiş bir datagram alsın.
- Bu datagram da merkezin dışında bir adrese gidecek olsun.
- R1 bunu IPsec datagramına çevirip çevirmeyeceğini nereden bilecek? Hatta diyelim ki IPsec olarak işledi. Bunu hangi SA'ya göre yapacak?
- SADB ile birlikte IPsec, bir veri yapısı daha sağlar: Security Policy Database (SPDB)
- SPDB, hangi tip datagramların IPsec olarak işleneceğini ve hangi SA'ların kullanılacağını belirtir.
- Dolayısıyla ile SPDB – gelen datagram ile “ne” yapılacağını, SADB ise “nasıl” yapacağını söyler.

IPsec datagram





- IPsec tam olarak hangi servisleri sunuyor?
- Saldırgan – R1 ve R2 arasında man-in-the-middle – gözünden bakalım.
- Saldırgan SA tarafından kullanılan authentication & Şifreleme anahtarlarını bilmiyor.
- Orijinal datagramı göremez.
- Sadece data değil, Protocol no, kaynak IP, hedef IP de görülemez.
- SA üzerinden gönderilen datagram için sadece 172.16.1.0/24 ‘e ait bir host olduğunu bilebilir.
- 172.16.2.0/24’te olan bir ağıta gideceğini bilir.
- TCP, UDP veya ICMP; HTTP, SMTP vb. taşıdığını bilmez.
- Bu nedenle güvenilirlik TLS’den yüksek seviyede.

IPsec Servisleri

- Saldırgan datagramın bazı bitlerinin yerini değiştirerek tamper atağı yapsın.
- Bu değişmiş datagram R2'ye vardığında integrity testinden geçemez (MAC kullanımı)
- Saldırgan kendisini R1 gibi gösterson ve 200.168.1.100 kaynaklı, 193.68.2.23 hedefli bir sahte datagram yaratsın.
- Bu datagram da R2'deki integrity testine takılır.
- Sıra no sayesinde replay (tekrar) ataklarına karşı da korumalıdır.

IPsec Servisleri



- Eğer VPN az sayıda uç noktaya sahipse (örnekteki gibi 2 router), ağ yöneticisi manuel olarak SA bilgilerini uç noktalardaki SADB'lere girer.
- SA bilgileri : şifreleme/kimlik doğrulama algoritmaları ve anahtarlar, SPI'lar
- Geniş VPN'ler içinse – yüzlerce, hatta binlerce IPsec router ve hosta sahip - manuel giriş pratik değil
- SA'ların otomatik olarak oluşturulmasını sağlayan bir mekanizma lazım
- IKE – Internet Key Exchange Protokolü bunun içindir.
- RFC 5996'da tanımlanmıştır.

IKE: Key Management in IPsec



- TLS'deki el sıkışmaya benzerdir.
- IPsec elemanlarının bir sertifikası vardır – elemanın public anahtarını içeren
- TLS'deki gibi
 - 2 taraf sertifikalarını değiştirir;
 - authentication ve şifreleme algoritmaları konusunda anlaşır;
 - IPsec SA'da kullanılacak oturum anahtarlarını yaratmak için gerekli anahtar bilgilerini güvenli şekilde değiş-tokuş yapar.
- TLS'den farklı olarak bu işleri 2 aşamada yapar.

IKE



- Bu 2 aşamayı R1 ve R2 için inceleyelim.
- Aşama 1 – Mesaj çiftlerinin değişimini içeren 2 kısımdan olur.
- 1. Kısım: Her iki taraf da Diffie-Hellman kullanarak çift yönlü IKE SA yaratır. IKE SA, IPsec SA'dan farklıdır.
- IKE SA, router'lar arasında authenticated ve encrypted bir kanal sağlar.
- Bu 1. mesaj çifti değiş tokuş kısmında, IKE SA için encryption ve authentication anahtarları belirlenir.
- Aynı zamanda 2. aşamada IPSec SA anahtarlarını hesaplamak için kullanılacak olan master secret (MS) belirlenir.

IKE



- Bu kısımda RSA public ve private anahtarları kullanılmaz.
- Ayrıca, ne R1 ne de R2 private anahtarları ile mesajlarını imzalayarak kendi kimliklerini doğrulamazlar.
- 2. Kısım: Mesajlarını imzalayarak her iki taraf da kimliğini doğrular.
- Bu kimlikler pasif dinleyiciye görünür değildir – güvenli IKE SA kanalı
- Ayrıca her iki taraf da IPsec SA tarafından kullanılacak encryption and authentication algoritmalarını belirlerler.

IKE



- 2. Aşamada her iki taraf da her yöne IPSec SA yaratır.
- Bu aşamanın sonunda 2 SA'nın her iki tarafı için de encryption ve authentication oturum anahtarları belirlenir.
- Her iki taraf da SA'yı kullanarak güvenli datagram gönderebilir – daha önce anlatıldığı gibi.
- IKE'de neden 2 aşama vardır?
- Asıl hedef hesaplama maliyeti – 2.aşamada herhangi bir public anahtar kriptolojisi kullanılmadığı için 2 IPsec elemanı arasında relatif olarak az maliyetle çok sayıda SA yaratılabilir.

IKE



Katmanlarda Güvenlik Önlemleri – Veri Bağlantı Katmanı Wireless LANs

*Kaynak : James F. Kurose & Keith W. Ross - Computer
networking: A top-down approach 6th Edition*



- Kablosuz ağlarda hem WLAN hem hücresel ağlarda güvenlik önemli – saldırgan gizlice dinleyebilir.
- 2 kritik güvenlik sorunu var:
- 1 – Mutual authentication – Bir mobil aygıt AP'ye atanarak datagram alışverişi yapmadan önce kimlik doğrulaması istenir – erişim önceliklerini ve aygıtı tanımak için.
- Aynı şekilde mobil aygıt da atandığı ağın kimliğini doğrulamak ister. Buna mutual authentication denilir.
- 2 – Şifreleme – Kablosuz kanallar dinlenebilir ve manipüle edilebilir – mobil aygıt ve AP arası gezen link-level çerçevelerin şifrlenmesi lazım. Simetrik anahtar kriptolojisi kullanılır – çünkü hız önemli. Anahtar ihtiyacı var – her iki tarafın da.

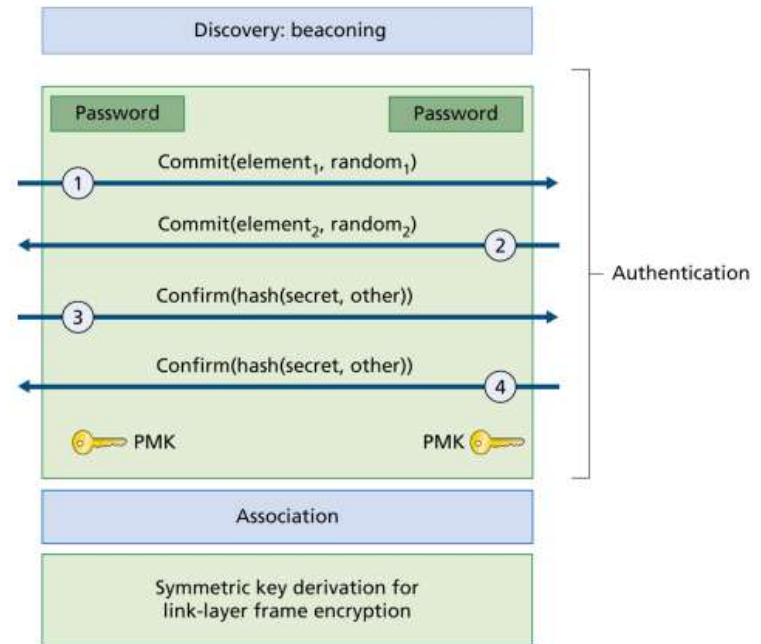
Giriş

- WiFi authentication – ağ ayarlarına bağlıdır.
- «personal» ayarlarda istemci aygıt ağa bağlanmak ister – AP doğrudan kimlik doğrular.
- «enterprise» ayarlarda aygıt merkezi bir kimlik doğrulama sunucusuna gider ve o ağın hem LAN/WLAN lerine hem de diğer kaynak ve servislerine erişim sağlanır.
- Enterprise ayarlarında AP aracıdır. İşlem istemci ve authentication sunucu arasındadır.

Giriş

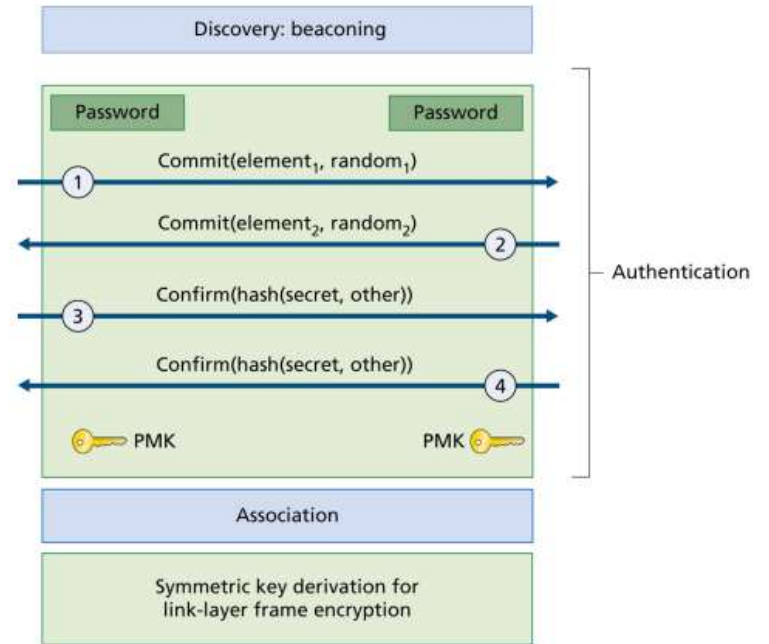


- Beaconing ve WLAN association – WLAN konusu.
- Personal WiFi ayarı içinkimlik doğrulama standardı – WiFi Protected Access 3 – Personal (WPA3-Personal)
- AP-istemci arasında password tabanlı kimlik doğrulama kullanılır.
- Aynı password tüm aygıtlar için kullanılır.
- Spesifik kimlik doğrulama protokolü – SAE - **Simultaneous Authentication of Equals**



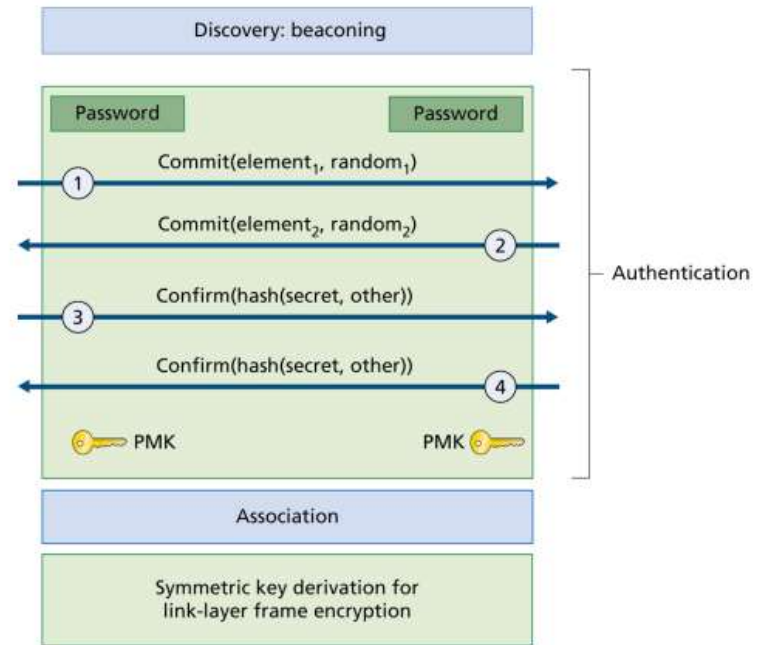
WPA3 - Personal

- Önce istemci AP'nin Beacon mesajını alır. Bu mesajda Personal desteği olduğu belirtilir.
- SAE protokolü 4 aşamalı Dragonfly Key Exchange protokolünü kullanır.
- Aşama 1 – Client commit – istemci password kullanarak yeni bir değer üretir – element1 – kriptolojik olarak güvenlidir yani element1 'den password çıkarılamaz.
- İstemci aynı zamanda nonce olarak kullanacağı random bir sayı atar. Bu değerleri SAE commit mesajı içinde AP'ye yollar.



WPA3 - Personal

- Aşama 2 – AP commit – AP, element1'den password'ü çıkarır ve doğrular. Ardından element2'yi üretir ve kendi rasgele sayısı ile istemciye yollar.
- Aşama 3 – istemci de benzer şekilde element2'yi gönderen AP'nin password'ü bildiğini doğrular. Replay ataklara karşı hash yollar – önceki aşamalarda yer alan değerler için
- Aşama 4 – AP aldığı hash'i inceler ve kendisi de hash yollar



WPA3 - Personal



Doç.Dr.İpek Abasıkeleş-Turgut

- Wired Equivalent Privacy (WEP)
- 802.11 spesifikasyonlarında standartlaştırılmış
- Amaç – kablosuz ağlarda güvenliği sağlamak
- 1999’da bir host ve Wireless Access Point (base station) arasında authentication ve encryption amacıyla tasarlanmış
- Simetrik anahtar yaklaşımı
- WEP, bir anahtar yönetim algoritması sağlamaz.
- Yani host ve Wireless AP, bir şekilde anahtar konusunda anlaşmış varsayılır.

WIRELESS LAN GÜVENLİĞİ

- Authentication şu şekilde sağlanır:
- 1 – Wireless host, AP'den authentication talep eder.
- 2 – AP, bu isteğe 128 byte lık bir nonce ile cevap verir.
- 3 - Wireless host, AP ile paylaştığı simetrik anahtarı kullanarak nonce'ı şifreler.
- 4 – AP, şifreleri nonce'ı çözer.
- Eğer şifresi çözülmüş nonce ile orijinal nonce aynı ise, host, AP tarafından kimliği doğrulanır.

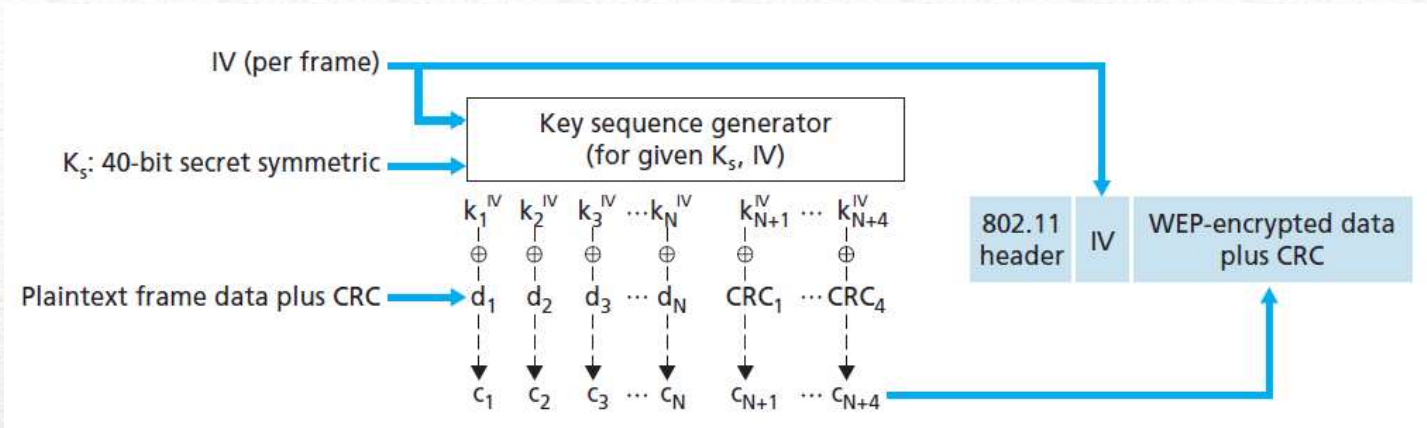
WEP



- Encryption:
- Gizli bir 40-bit simetrik anahtar – K_s – hem host hem AP tarafından bilindiği varsayılır.
- Ayrıca 24-bit IV, 40-bit gizli anahtara eklenerek 64-bitlik anahtar üretilir.
- Bu 64 bitlik anahtar tek bir çerçeveyi şifrelemede kullanılır.
- IV, her frame için farklıdır.

WEP





- Encryption: $c_i = d_i \oplus k_i^{IV}$
- 4 byte CRC değeri her data payload'u için hesaplanır.
- Data payload + CRC - RC4 stream cipher kullanılarak şifrelenir.
- RC4 alg. anahtar dizisi (stream) üretir.
- Bunlar şifrelemede kullanılır.
- Verinin i.byte ile anahtarın i.byte'ı XOR işleminden geçer – ciphertext'in i.byte'ı için

WEP

- IV değeri, WEB-şifreli 802.11 çerçevesinin başlık kısmındaki plaintext içine dahil edilir.
- Alıcı, gönderici ile paylaştığı 40 bitlik gizli anahtara IV yi ekler – 64 bitlik anahtarı elde eder.
- Ardından çerçeveyi çözer:
$$d_i = c_i \oplus k_i^{IV}$$
- RC4 alg.uygun kullanımında aynı 64-bit anahtar asla birden çok kez kullanılmaz.
- WEP anahtarı her çerçevede farklıdır.

WEP



- Verilen bir Ks değeri için 2^{24} olası farklı anahtar mevcut.
- Eğer bu anahtarlar rasgele seçilirse, aynı IV değerinin seçilme olasılığı 12.000. çerçeveden sonra %99'un üzerine çıkar.
- 1Kbyte çerçeve boyu, 11 Mbps iletim hızını düşünürsek, 12.000. çerçeveden önce sadece birkaç saniye var demektir.
- Ayrıca IV değeri plaintext içinde yollandığı için pasif dinleyici nerede duplicate IV kullanıldığını bilir.

WEP



- Duplicate key kullanmanın dezavantajını daha iyi anlamak saldırgan tarafından chosen-plaintext atağı yapıldığını varsayalım.
- Diyelim ki saldırgan IP spoofing kullanarak Alice'e, bilinen içeriğe $d1, d2, d3, d4, \dots$ sahip bir dosya yollamak için, bir HTTP veya FTP isteği yollasın.
- Saldırgan ayrıca şifreli veriyi de gözlemliyor: $c1, c2, c3, c4, \dots$.
- Biliyoruz ki : $d_i = c_i \oplus k_i^{IV}$
- Eğer bu eşitliğin her iki tarafını c_i ile XOR işlemine sokarsak;
- Dolayısı ile saldırgan bilinen d_i ve c_i değerlerini kullanarak ilgili anahtarları hesaplayabilir.

$$d_i \oplus c_i = k_i^{IV}$$

WEP



- IEEE 802.11'in 1999 yılında yayınlanmasının hemen ardından daha güçlü güvenlik mekanizmalarına sahip yeni ve gelişmiş versiyon olan IEEE 802.11i üzerine çalışmalar başlamıştır.
- Bu yeni standardın son hali 2004'te tamamlanmıştır.
- WEP
 - görece zayıf şifreleme sunduğu için,
 - authentication tek bir yolla yapıldığı için,
 - anahtar dağıtım mekanizması olmadığı için,

IEEE 802.11i



- IEEE 802.11i ‘ de güçlü şifreleme, çok sayıda kimlik doğrulama mekanizması ve anahtar dağıtım mekanizması vardır.
- Wireless istemci ve AP’ye ek olarak AP’nin iletişim kuracağı bir kimlik doğrulama sunucusu eklenmiştir.
- Bu sunucuyu AP’den ayırmak = 1 sunucu ile birden çok AP’ye hizmet etmek, kararları merkezi hale getirmek, AP maliyetini ve karmaşıklığını azaltmak

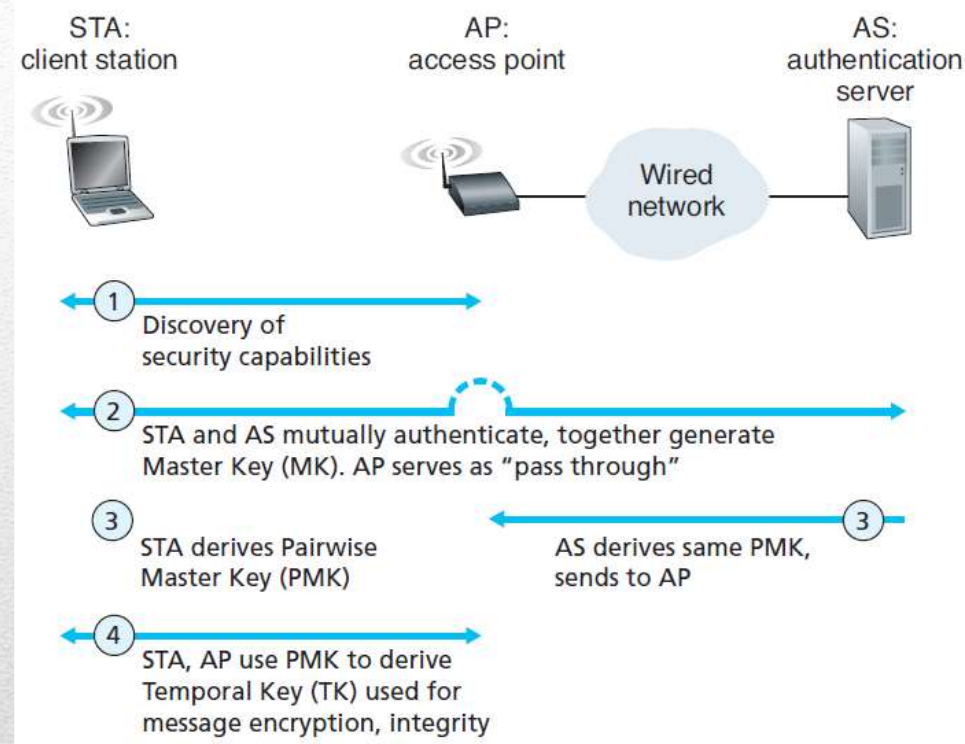
IEEE 802.11i



- 802.11i 4 aşamadan oluşur:
- 1 – Keşif – AP varlığını söyler, kablosuz istemci düğüme authentication & encryption çeşitlerini sunar.

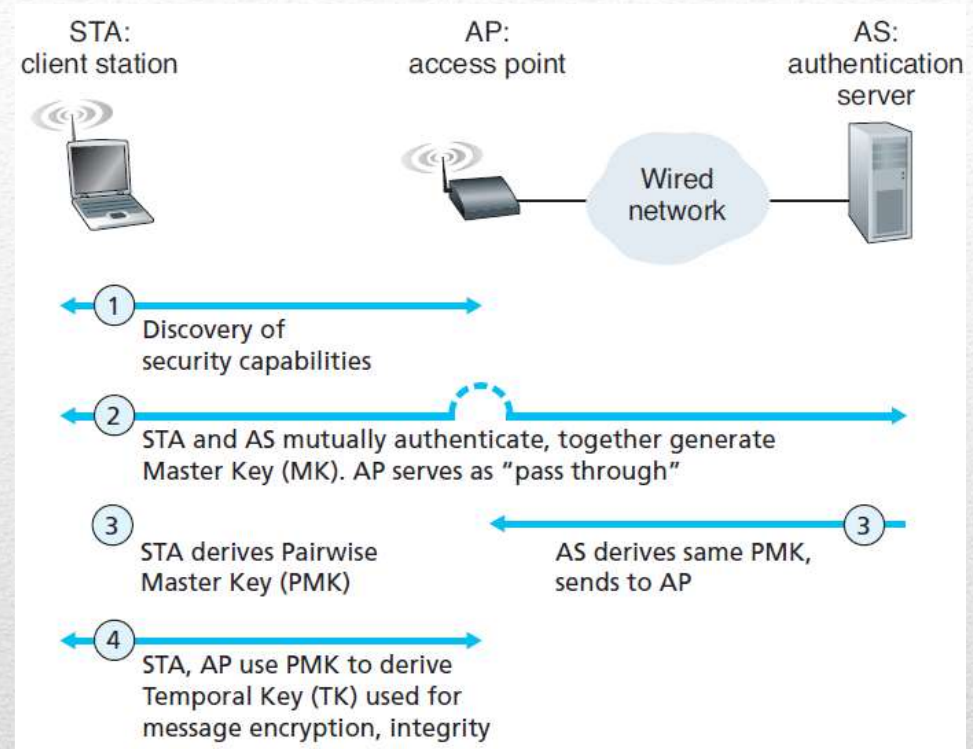
Ardından istemci, istediğini seçer. İstemci ve AP mesajlaşmalarına rağmen

İstemci henüz kimlik doğrulamadı ve Şifreleme anahtarı yok. Yani daha çok adım lazım



IEEE 802.11i

2 – Karşılıklı Kimlik Doğrulama
ve MK üretimi – Master Key
Kimlik doğrulama Client ile AS
arasında. Burada AP'nin rolü,
aracı olmak – mesaj iletimi
Extensible Authentication
Protocol
(EAP) bu mesajlaşmanın
kurallarını tanımlar.



IEEE 802.11i

Doç.Dr.İpek Abasıkeleş-Turgut

- EAP mesajları – Client ile AP arası - EAPoL kullanılarak kapsülленir ve 802.11 wireless link üzerinden yollanır.
- EAP mesajları AP’de kapsülden çıkarılır ve RADIUS protokolü ile yeniden kapsülленir. UDP/IP üzerinden AS’ye yollanır.
- RADIUS, 802.11i için şart değil ama de facto standart bileşenidir.
- RADIUS yerine artık **DIAMETER** protokolü standart haline getirildi.
- EAP ile AP kimlik doğrulama için birçok alternatiften birini seçebilir.
- 802.11i bu konuda bir zorlama yapmasa da genellikle EAP-TLS authentication şeması kullanılır.
- Bu şema, istemci ve AS arasında karşılıklı kimlik doğrulama ve MK üretimini içeren daha önce işlediğimiz public anahtar tekniklerini kullanır.

IEEE 802.11i



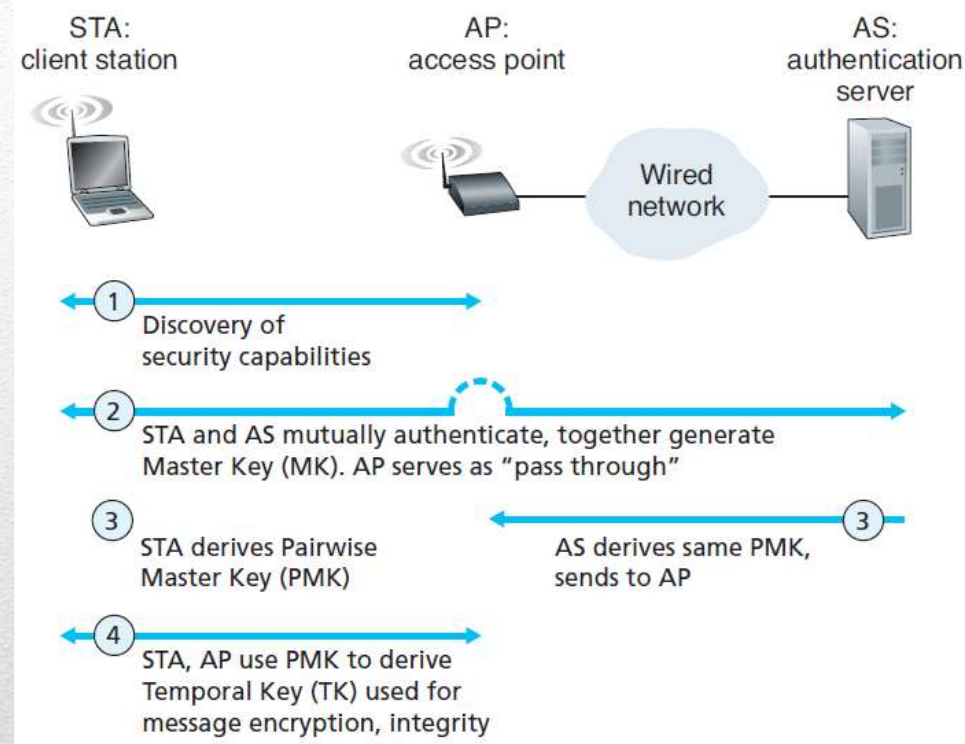
- 3 – İkili MK üretimi – PMK

MK, hem istemci hem AS tarafından

bilinir ve PMK üretiminde kullanılır.

AS, PMK'yı AP'ye gönderir.

Artık istemci ve AP shared key paylaşıyorlar ve karşılıklı kimlik doğrulama yaptılar -> Artık işe koyulabilirler!



IEEE 802.11i

Doç.Dr.İpek Abasıkeleş-Turgut

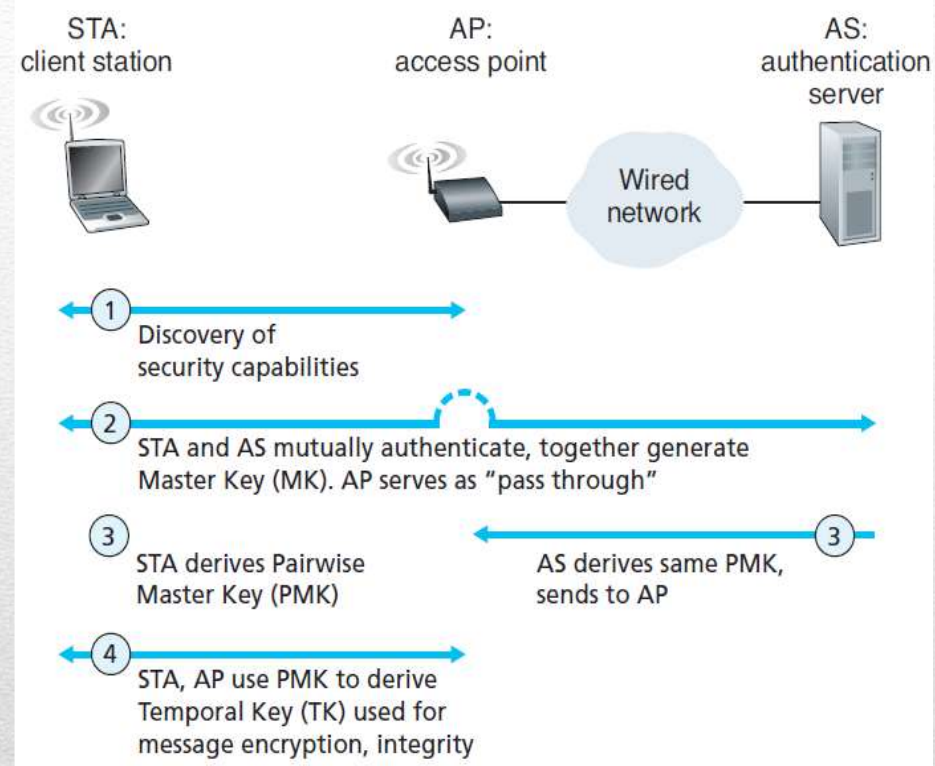
- 4 – Geçici Anahtar Üretimi (TK)
PMK ile istemci ve AP iletişimde kullanacakları ek anahtarları üretir.
TK – wireless link üzerinden giden verinin

veri bağlantı katman şifreleme işlemlerinde kullanılır.

802.11 çeşitli şifreleme yöntemleri sunar.

AES-tabanlı şifreleme ve güçlendirilmiş

WEP şifreleme bunlara dahildir.



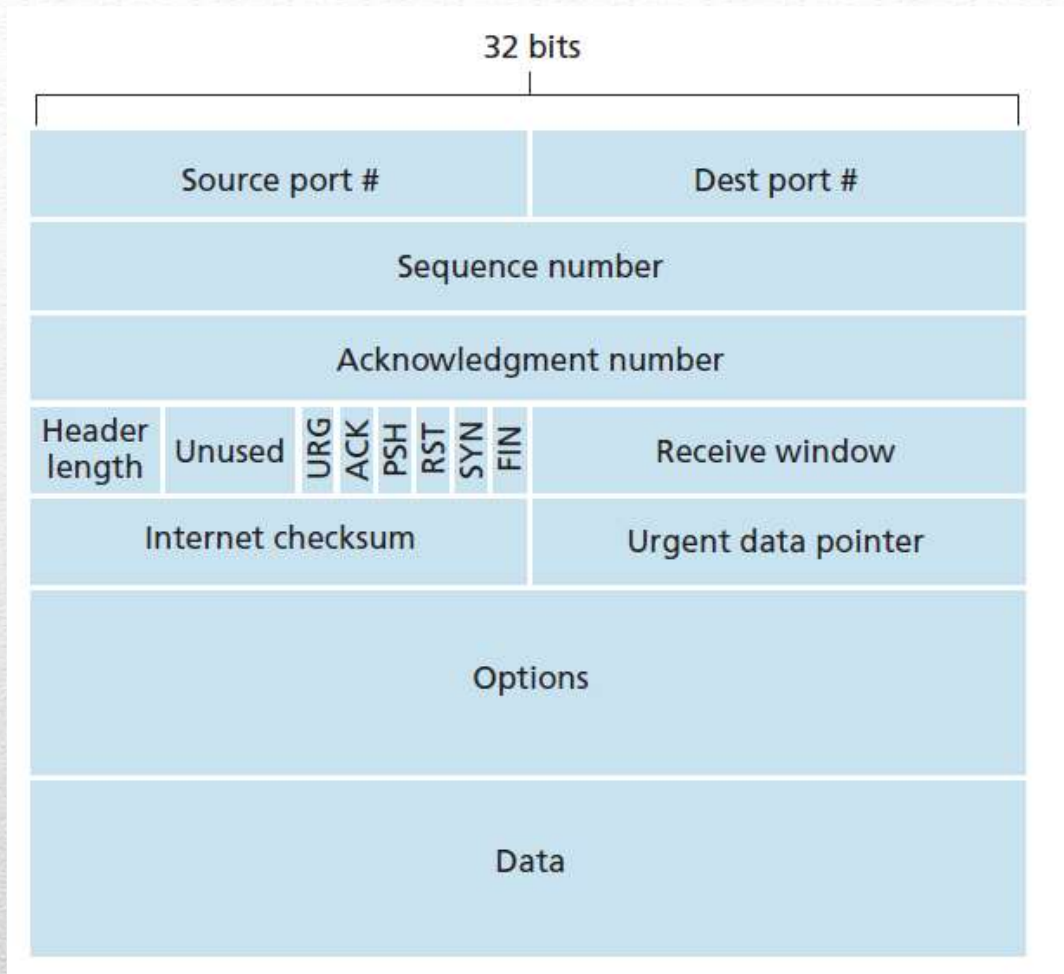
IEEE 802.11i

Doç.Dr.İpek Abasıkeleş-Turgut

Firewall



Doç.Dr.İpek Abasıkeleş-Turgut



ÖN BİLGİ : TCP BAĞLANTILARI



Doç.Dr.İpek Abasıkeleş-Turgut

- Bir aygıtta (istemci) çalışan bir proses, başka bir aygıtta (sunucu) çalışan bir proses ile bağlantı kurmak istesin.
- Adım 1: İstemci TCP, sunucu TCP'ye özel bir TCP segmenti gönderir.
- Bu özel segmentte uygulama katman verisi yoktur.
- Başlıkta yer alan SYN biti 1'dir.
- Bu nedenle bu özel segmente SYN segmenti denilir.
- Ayrıca istemci rasgele bir başlangıç sıra nosu (client_isn) üretir ve bunu ilk TCP SYN segmentinin sıra no kısmına koyar.

ÖN BİLGİ : TCP BAĞLANTILARI

- Adım 2: Sunucu TCP bu segmenti alınca TCP tamponlarında yer ayırır (allocate), bağlantı parametrelerini ayarlar ve istemci TCP'ye bağlantı onay mesajı yollar.
- Bu segmentte de uygulama katman verisi yoktur.
- Başlıkta: SYN biti 1'dir. ack alanı $client_isn+1$. Sunucu bir ilk sıra no seçer: $server_isn$, bunu başlık sıra nosu olarak koyar.
- Bu segmentin adı SYNACK'dır.
- Anlamı: client-isn sıra numaralı SYN paketini aldım, bağlantıyı kabul ediyorum ve benim ilk sıra numaram $server_isn$

ÖN BİLGİ : TCP BAĞLANTILARI

- Adım 3: SYNACK'yı alan istemci tamponlar için yer alır, bağlantı değişkenlerini yaratır.
- Ardından sunucuya yeni bir segment yollar. ACK segmenti.
- Sunucunun bağlantı onayına ack segmenti.
- Sıra no : $\text{client_isn}+1$
- Bu segmentteki ack alanına $\text{server_isn}+1$ değeri konulur.
- SYN biti 0 yapılır çünkü artık bağlantı kuruldu.
- Bu segment uygulama katman verisi taşıyabilir.

ÖN BİLGİ : TCP BAĞLANTILARI

- İstemci bağlantıyı sonlandırmak istiyor.
- Sunucu da isteyebilir.
- İstemci, Sunucuya bir TCP segmenti yollar ve FIN bitini 1 yapar.
- Ardından FIN_WAIT_1 durumuna geçer.
- Bu durumda istemci sunucudan ack içeren bir TCP segmenti bekler.
- Bu segmenti alınca, FIN_WAIT_2 durumuna geçer.
- Bu durumda istemci sunucudan FIN biti 1 olan bir segment bekler.
- Bu segmenti alınca, ack yollar ve TIME_WAIT durumuna geçer. (ACK kayıplarına karşı)
- Tipik olarak 30 sn., 1dk veya 2dk bu durumda kalır ve bu süre sonunda bağlantı resmi olarak sonlanır, değişkenler silinir vb.

ÖN BİLGİ : TCP BAĞLANTILARI



- İnternet güvenliğini inceledik.
- Şimdi, bir şirket ağını ve o ağı yöneten ağ yöneticisini düşünelim.
- Ağ yöneticisinin gözünden dünya 2'ye bölünür: «goog guys» & «bad guys»
- Şirket ağına bağlı olup, şirket bilgilerine erişme yetkisi olanlar & Diğer herkes
- Çoğu şirkette, şirkete girip/çıkanların güvenlik denetiminin yapıldığı tek bir giriş/çıkış noktası olur.
- Bilgisayar ağlarında, ağa giren/çıkan trafiğin güvenlik denetimi / log tutulması/ engellenmesi/ iletilmesi işleri operational devices – işletimsel aygıtlar tarafından yapılır.
- Bunlar: firewall, IDS ve IPS'dir.

Operational Security – İşletimsel Güvenlik

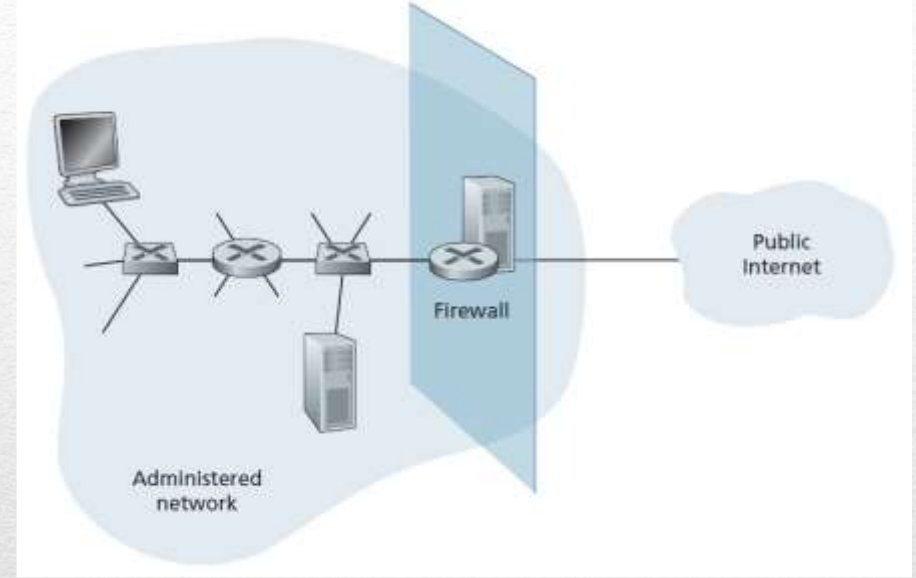


- Firewall, bir donanım ve yazılım birleşimidir.
- Amaç – şirketin iç/dahili ağını serbest Internetten izole etmek;
- Bazı paketlerin geçmesine izin verip; bazılarının bloklamak
- Firewall sayesinde ağ yöneticisi dış dünya ile yönetilen ağdaki kaynaklar arasındaki erişimi kontrol edebilir.
- Bunu, kaynaklardan çıkan ve kaynaklara gelen trafiği yöneterek yapar.

Firewalls



- Firewall'un 3 amacı vardır:
- **1** – Dışarıdan içeriye gelen & içeriden dışarıya çıkan tüm trafik firewall üzerinden geçer.
- Tam olarak iç ve dış dünya arasındaki sınırda yer alır.
- Geniş şirketlerde çok seviyeli firewall veya dağıtık firewall kullanılabilir.
- Onun dışında genellikle tek bir erişim noktasında yer alır.
- Böylece yönetimi kolay olur; güvenlik-erişim kurallarını daha rahat uygulayabilir.



Firewalls

- **2** – Sadece yetkilendirilmiş trafiğin geçmesine izin verilir.
- Yetkilendirilmiş trafik lokal güvenlik ilkelerinde/kurallarında tanımlanmıştır.
- Şirkete giren/çıkan tüm trafik firewall üzerinden geçer ve firewall yetkili ağa erişimi kısıtlayabilir.
- **3** – Ağ sızmaya karşı korumalıdır.
- Firewall'un kendisi de ağa bağlı bir aygıttır.
- Eğer iyi bir şekilde tasarlanmaz veya kurulamazsa, saldırgan rolü oynayabilir ve bu de hiç olmamasından daha tehlikeli bir durumdur.

Firewalls



- Cisco ve Check Point bilinen firewall şirketleri.
- Firewall 3 kategoriye bölünür:
- 1 – Geleneksel paket filtreleyiciler (traditional packet filters)
- 2 – Geniş kapsamlı filtreler (stateful filters)
- 3 – Uygulama ağ geçitleri (application gateways)

Firewalls



- Bir şirketin genellikle kendi dahili ağını, ISS'ye bağlayan bir gateway router'ı olur.
- Giren çıkan tüm trafik bu yönlendiriciden geçer.
- O nedenle filtreleme işlemi bu router üzerinde yapılır.
- Bir paket filtreleyicisi her datagramı ayrı ayrı inceler ve karar verir.
- Datagram ya geçecektir, ya da düşürülecektir.
- Bu kararı administrator-specific kurallara göre verir.

Geleneksel paket filtreleyiciler



- Filtreleme kararları tipik olarak şunlara bağlıdır:
- IP kaynak ve hedef adresleri
- IP datagramındaki protokol tipleri: TCP, UDP, ICMP, OSPF vb.
- TCP veya UDP kaynak hedef port adresleri
- TCP bayrak bitleri: SYN, ACK, vb.
- ICMP mesaj tipi
- Ağa giren/çıkan datagramlar için farklı kurallar
- Farklı router arayüzleri için farklı kurallar

Geleneksel paket filtreleyiciler



- Ağ yöneticisi firewall'ı şirketin kanunlarına göre konfigüre eder.
- Bu kanunlar şirketin güvenliği kadar kullanıcı üretkenliğini ve bant genişliği kullanımını da hesaba katabilir.
- Şekilde 130.27/16 ağ adresli bir şirketin olası kanunları listelenmiştir.
Web server IP: 130.207.244.203

Policy	Firewall Setting
No outside Web access.	Drop all outgoing packets to any IP address, port 80
No incoming TCP connections, except those for organization's public Web server only.	Drop all incoming TCP SYN packets to any IP except 130.207.244.203, port 80
Prevent Web-radios from eating up the available bandwidth.	Drop all incoming UDP packets — except DNS packets.
Prevent your network from being used for a smurf DoS attack.	Drop all ICMP ping packets going to a "broadcast" address (eg 130.207.255.255).
Prevent your network from being tracerouted	Drop all outgoing ICMP TTL expired traffic



- Bir filtreleme kanunu, adres ve port noları kombinasyonuna bağlı olabilir.
- Örneğin bir filtre router, spesifik IP adresleri listesi dışında gelen/giden tüm Telnet datagramlarını iletir. Port 23
- Bu kural sadece izin verilen aygıtların Telnet bağlantılarını onaylar.
- Ancak bu kural, IP spoofing ataklarına karşı maalesef koruma sağlamaz.

Geleneksel paket filtreleyiciler



- Filtreleme TCP ACK bitinin 1 veya 0 olmasına göre de yapılabilir.
- Ne için kullanılır?
- Şirket kendi dahili/ bünyesindeki kullanıcıların dış dünyadaki sunuculara bağlanmasına izin verip; harici kullanıcıların kendi dahili sunucularına bağlanmasını engellemek için
- Her TCP bağlantısının ilk segmentinde ACK biti 0, diğer segmentlerde 1
- Eğer şirket, dahili sunucularına harici kullanıcı bağlantılarını engellemek isterse?

Geleneksel paket filtreleyiciler



- Firewall kuralları yönlendiricilerin içine erişim kontrol listeleri şeklinde yerleştirilir.
- Her router ara yüzü kendi listesine sahiptir.
- 222.22/16 adresli bir şirket için erişim listesi örneği şekilde görülmektedir.
- Bu erişim kontrol listesi, yönlendiriciyi, şirketin harici ISS'sine bağlayan arayüzü içindir.

action	source address	dest address	protocol	source port	dest port	flag bit
allow	222.22/16	outside of 222.22/16	TCP	> 1023	80	any
allow	outside of 222.22/16	222.22/16	TCP	80	> 1023	ACK
allow	222.22/16	outside of 222.22/16	UDP	> 1023	53	—
allow	outside of 222.22/16	222.22/16	UDP	53	> 1023	—
deny	all	all	all	all	all	all

- Geleneksel filtrelerde filtreleme kararı her paket için ayrı olarak veriliyor.
- Geniş kapsamlı filtreler ise TCP bağlantılarını izleyerek karar veriyor.
- Örneğin bir önceki şekilde – 2. satır kullanılarak saldırganlar filtreden geçebilir,
 - dahili sunuculara istedikleri paketlerle saldırabilir;
 - DoS atağı gerçekleştirebilir;
 - dahili ağın haritasını çıkarabilir.
- Nasıl engellenir? – TCP ACK paketlerini de engelleyerek
- Sonuç ?

Geniş kapsamlı filtreler (Stateful filters)



- Stateful filtreler bu sorunu çözmek için bağlantı tablosundaki tüm TCP bağlantılarını izler.
- Bağlantıları nasıl izler?
- +Yeni bir bağlantının başlangıcını gözlemleyebilir.
 - 3 yollu el sıkışmayı izleyerek – SYN, SYNACK, ACK
- + bağlantının sonunu gözlemleyebilir.
 - FIN paketi ile
- + bağlantının sonlandığını tahmin edebilir – örn. 60 sn boyunca bir aktivite görmezse.

Geniş kapsamlı filtreler (Stateful filters)



source address	dest address	source port	dest port
222.22.1.7	37.96.87.123	12699	80
222.22.93.2	199.1.205.23	37654	80
222.22.65.143	203.77.240.43	48712	80

- Örnek bir bağlantı tablosu şeklindeki gibidir.
- Halihazırda 3 bağlantı var – her biri şirketten başlatılmış.
- Ayrıca stateful filtre erişim kontrol listesinde yeni bir kolon içerir: check connection

action	source address	dest address	protocol	source port	dest port	flag bit	check connection
allow	222.22/16	outside of 222.22/16	TCP	>1023	80	any	
allow	outside of 222.22/16	222.22/16	TCP	80	>1023	ACK	X
allow	222.22/16	outside of 222.22/16	UDP	>1023	53	—	
allow	outside of 222.22/16	222.22/16	UDP	53	>1023	—	X
deny	all	all	all	all	all	all	



- Bağlantının kontrolü 2 kurala bağlı.
- Diyelim ki saldırgan şirket ağına zararlı bir paket sokmaya çalışsın.
- Bunun için kaynak port 80 olan, ACK bayrağı 1 olan bir datagram yollasın.
- Ayrıca bu paketin kaynak port nosu 12543 ve kaynak IP adresi 150.23.23.155 olsun.
- Bu paket firewall'a geldiğinde, şirket ağına girmesine izin verilmeden önce bağlantı tablosunun kontrol edilmesi gerekir.
- Eğer bu paket, bağlantı tablosunun içinde yer alan bir TCP bağlantısına sahip değilse, paket atılır.

Geniş kapsamlı filtreler (Stateful filters)



- 2.örnek olarak diyelim ki dahili bir kullanıcı Web’de gezinmek istiyor.
- Önce TCP SYN segmenti yollar ve böylece bağlantı tablosunda isteği kaydedilir.
- Sunucu ona cevap döndüğünde (ACK biti 1) firewall bağlantı tablosunu inceler ve bağlantının mevcut olduğunu görür.
- Paketlerin geçmesine izin verir.
- Böylece geleneksel filtrelerin dezavantajı bu filtrelerde görülmez.

Geniş kapsamlı filtreler (Stateful filters)



- Paket filtreleme, IP ve TCP/UDP başlıklarına göre filtreleme yapma sağlıyor: IP adresi, port no, ack bitleri
- Diyelim ki şirket sadece belli bir dahili kullanıcı grubuna Telnet hizmeti sunmak istiyorsa?
- Hatta bu imtiyazlı kullanıcıların, dış dünya ile Telnet yaratmadan önce kimlik doğrulama işleminden geçmeleri istenirse?
- Geleneksel veya stateful filtreler bu yeteneklere sahip değil
- Dahili kullanıcıların kimlikleri uygulama katman verisidir ve IP/TCP/UDP başlıkları içine dahil değildir.

Uygulama Ağ Geçici (Application gateway)



- Firewall paket filtreleri ile uygulama ağ geçitlerini birleştirerek daha iyi bir güvenlik hizmeti sunabilirler.
- Application gateway - IP/TCP/UDP başlıklarının ötesinde uygulama verisine göre karar mekanizması sunar.
- Application gateway = tüm uygulama verilerinin (hem gelen hem giden) üzerinden geçmek zorunda olduğu uygulama-özel sunucu
- Birden çok application gateway aynı host üzerinde çalışabilir ancak her gateway, kendi prosesine sahip ayrı bir sunucudur.

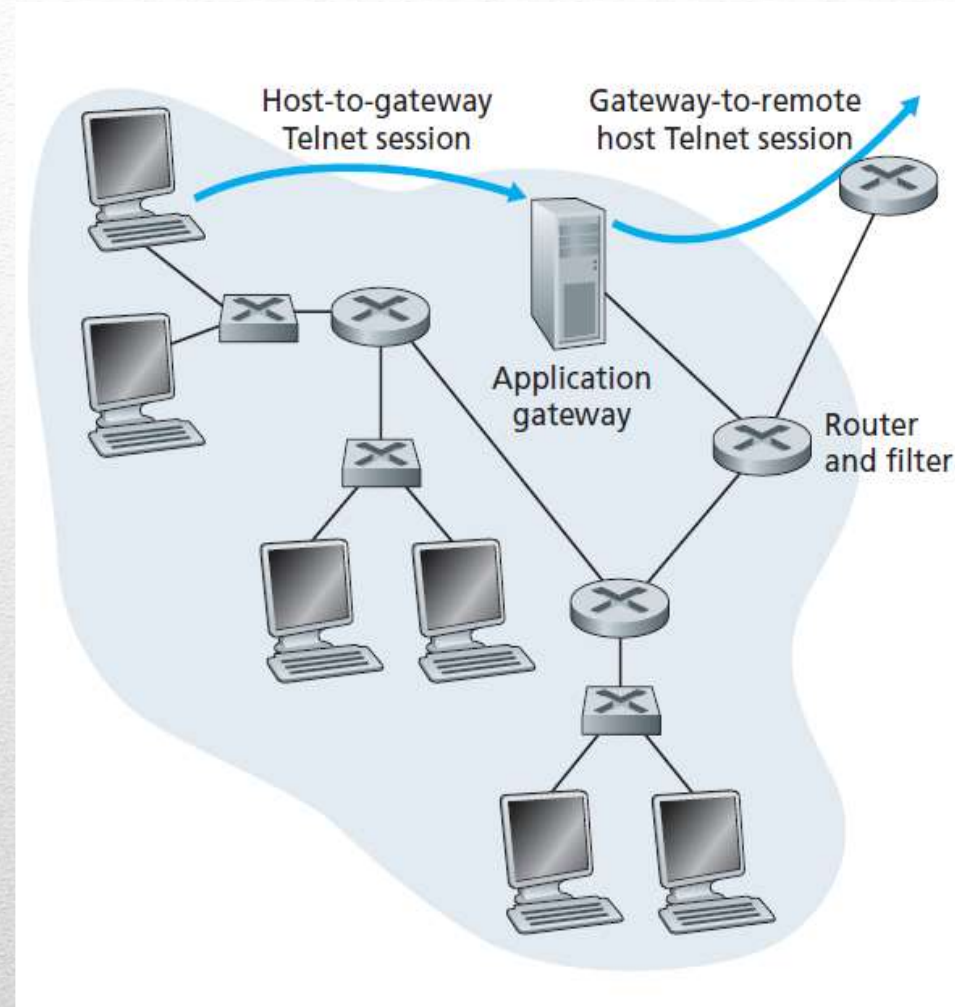
Application gateway

- Bir firewall sadece belirli ve sınırlı sayıda kullanıcının dış dünya ile Telnet bağlantısına izin versin; dışarıdan içeriye gelen tüm Telnetleri yasaklasın.
- Bu kural için router içinde bir paket filtreleyici ile Telnet uygulama ağ geçici kullanmak yeterli.
- Router – application gateway’in IP adreslerinden kaynaklanan Telnet bağlantıları dışındaki tüm Telnet bağlantılarını bloklayacak
- Böyle bir filtre konfigürasyonu sayesinde dışarıya çıkan tüm Telnet bağlantıları application gateway üzerinden geçmek zorunda kalır.

Application gateway



Diyelim ki dahili bir kullanıcı
Telnet ile dış dünyaya bağlanmak
istese.
Öncelikle app.gt. ile bir Telnet
oturumu başlatacak;
Gateway üzerinde çalışan ve gelen
Telnet oturumlarını dinleyen bir uyg.
kullanıcıdan user ID ve password
isteyecek;
Ardından kullanıcının dış dünyaya
Telnet yetkisinin olup olmadığını
kontrol edecek;
Yoksa – kullanıcı ile Telnet
bağlantısı app.gt.
tarafından sonlandırılacaktır.



Eğer kullanıcının yetkisi varsa
app.gt.;

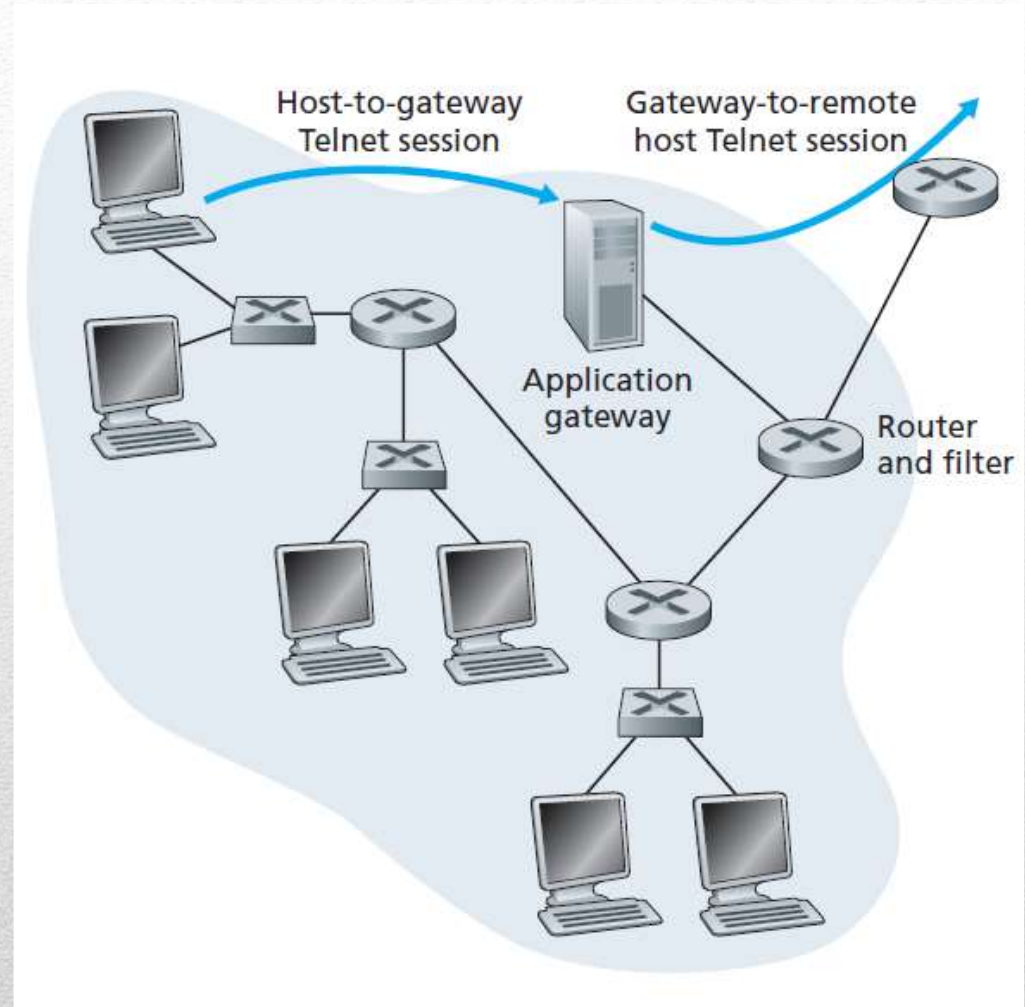
1 – kullanıcının bağlanmak
istediği

harici host adını sorar;

2 – Bu harici host ile arasında
Telnet oturumu başlatır;

3 – kullanıcıdan gelen datayı
harici hošta, ondan geleni de
kullanıcıya iletir.

App.gt. – sadece kullanıcı
kimlik doğrulaması değil, aynı
zamanda Telnet sunucu ve
Telnet istemci görevi görür.



- Harici ağlar çoğu zaman çoklu app.gt. sahiptirler.
- Örn. Telnet, HTTP, FTP ve e-mail gateway'leri gibi.
- Dezavantaj?
- 1 – Her uygulama için farklı app.gateway gerekli.
- 2 – tüm data gateway üzerinden akacağı için performans kaybı
- özellikle çoklu kullanıcı veya uygulamalar aynı gateway makinesini kullanıldığında daha büyük problem.
- 3 – kullanıcı istekte bulunacağı zaman istemci yazılımı gateway ile nasıl iletişime geçeceğini ve external sunucuyu ona nasıl söyleyeceğini bilmesi gerekiyor.

Application gateway



- Paket filtreler belirli tipte – HTTP, TCP, UDP vb. paketlerin geçişine izin veriyor / blokluyor.
- Çeşitli atakları tespit edebilmek için deep packet inspection yapılması lazım -> Başlık filtresinden öte
- app.gateway bazen bunu yapıyor ama o da sadece bazı uygulamalar için
- Öyle bir aygıt olmalı ki şüpheli paket (paket zinciri) görürse, bunların şirkete girmesine izin vermemeli veya sadece şüpheli oldukları için geçişe izin vermeli ama ağ yöneticisini uyarmalı.
- Böylece o, trafiği daha ayrıntılı inceleyerek uygun işlemleri yapabilir.

Intrusion Detection Systems



- Potansiyel saldırı trafiğini gözlemleyerek uyarı veren ağıta - **intrusion detection system (IDS)**
- Şüpheli trafiğin tamamını filtreleyen ağıta ise **intrusion prevention system (IPS)**
- IDS sayesinde çok çeşitli ataklar yakalanabilir: network mapping, port scan, TCP stack scan, DoS bandwidth-flooding attacks, worms, viruses, OS vulnerability attacks, application vulnerability attacks vb.
- Birçok şirket bu hizmeti sunuyor. Çoğu public-domain sistemler – Snort IDS gibi

Intrusion Detection Systems



- Snort IDS - public-domain & open source IDS
- yüzbinlerce uygulaması var.
- Linux, Unix ve Windows üzerinde çalışabilir.
- generic sniffing interface libpcap kullanır. – Tıpkı Wireshark vb. paket sniffer uygulamaları gibi
- 100 Mbps trafiği rahatlıkla kaldırır.
- Daha yüksek trafik oranları için çoklu Snort gerekir.

Snort



Doç.Dr.İpek Abasıkeleş-Turgut

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any  
(msg:"ICMP PING NMAP"; dsize: 0; itype: 8;)
```

- Örnek bir Snort imzası yukarıda görülmektedir.
- Şirket dışından içeriye gelen tip 8, boş payload'a sahip ICMP paketleri.
- nmap bu özelliklerde ping paketleri üretir.
- bu imza da nmap ping saldırılarını bulmak için kullanılır.
- bu imzaya uyan bir paket yakalanırsa – ICMP PING NMAP alarmı üretilir.
- yeni bir ataktan sonra sadece 1kaç saat içinde Snort atağın imzasını yazar ve dağıtır - yüzbinlerce diğer Snort yerleşkesine

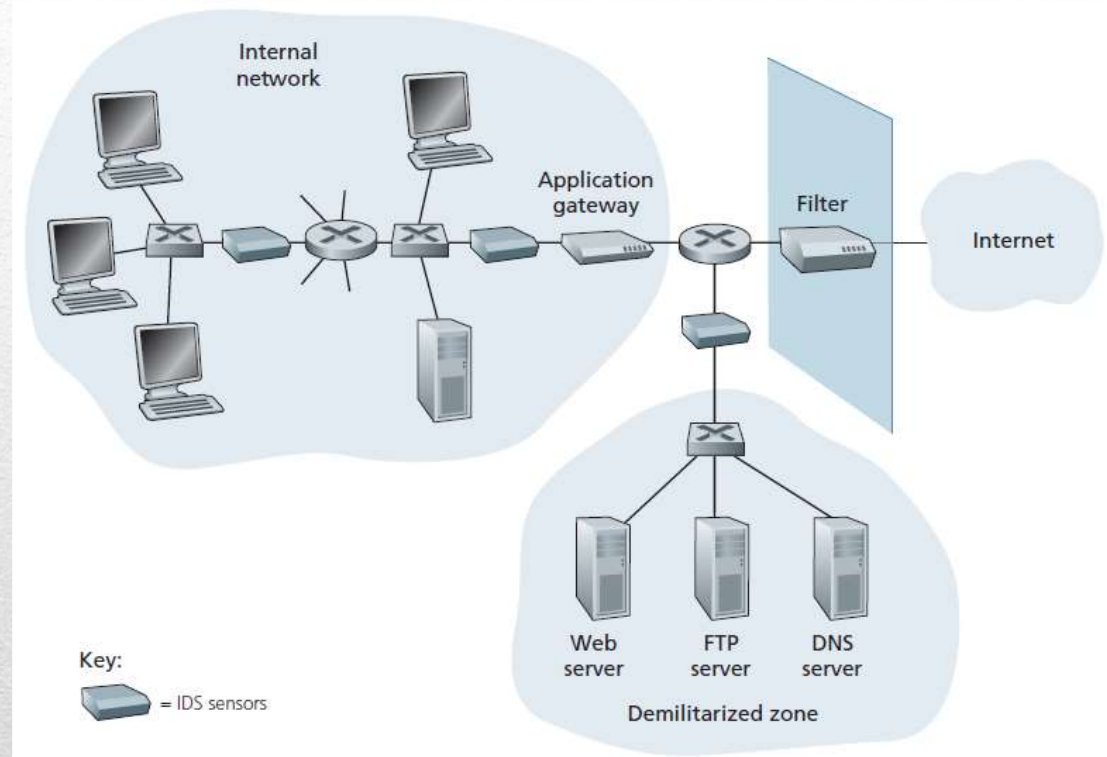
- Bir şirket 1 veya daha çok IDS sensörü alabilir.
- Şekilde 3 tane var.

Eğer birden çok sensör varsa, genelde ortak çalışarak şüpheli trafik bilgilerini merkezi bir IDS işlemcisine yollarlar.

Orada bilgiler toplanır, birleşir ve ağ yöneticisine alarm verilir.

Şekilde şirket ağı 2 parçaya ayrılmış: yüksek güvenli kısıım

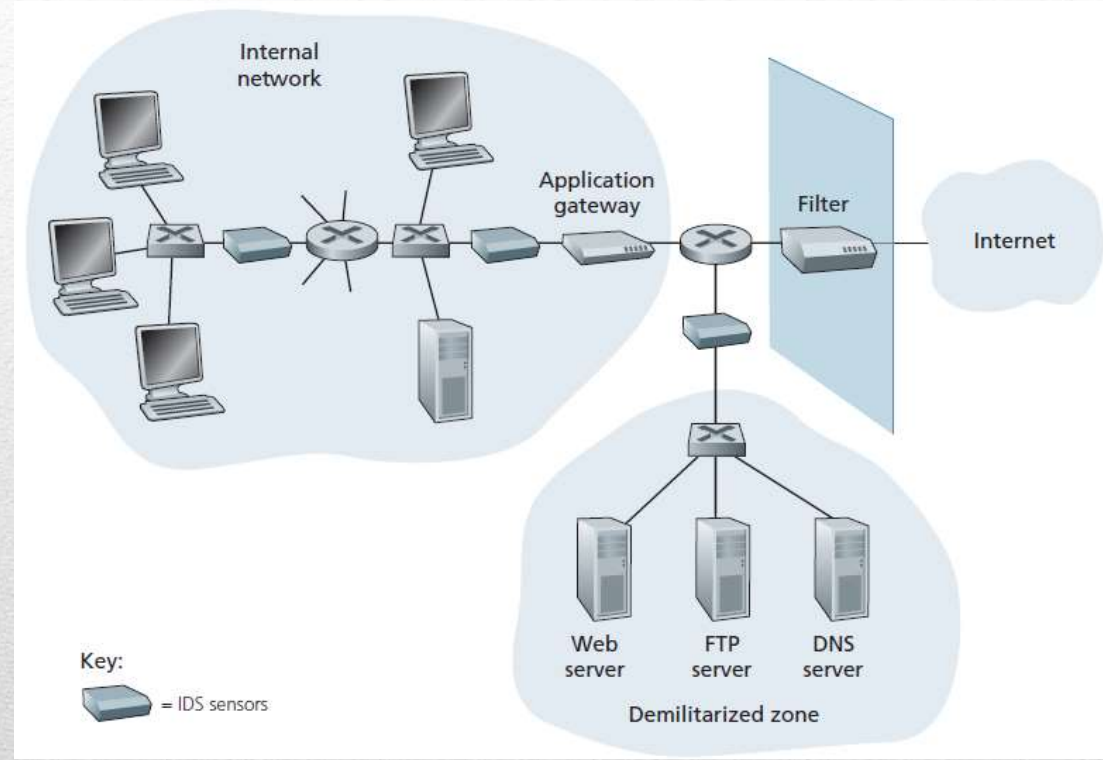
ve düşük güvenli kısıım



Y.G.Kısım bir paket filtresi, app.gt. tarafından korunup, ids tarafından takip ediliyor. D.G. kısım ise **demilitarized zone (DMZ)**, paket filtrekeyici tarafından korunuyor ve ayrıca ids sensörleri tarafından izleniyor.

DMZ sadece şirketin dış dünya ile iletişim kurması gereken sunucularını içeriyor.

Neden çoklu IDS? 1 tane filtre yanında hatta birleşik olsa?



- IDS sadece ayrıntılı paket incelemesi yapmaz – ayrıca her paketi onbinlerce «imza» ile karşılaştırır.
- bu ise ciddi anlamda bir işlem gerektirir.
- IDS’yi bu trafikten öteye yerleştirmek ile şirket trafiğinin bir kısmının görmeleri sağlanır – daha az işlem yükü

Intrusion Detection Systems

