Aantekeningen Basic Security

# Symmetrische Crypto

## Classificatie op basis van 3 eigenschappen:

* **Operaties om van X -> Y te gaan**
  + Substitutie
  + Transpositie
* **Gebruikte aantal sleutels**
  + Symmetrisch – secret key – dezelfde sleutel
  + Asymmetrisch – public key – verschillende sleutel
* **Operaties om van input -> X te gaan**
  + Blokvercijfering
  + Stroomvercijfering

## Steganografie

* **Informatie verbergen**
* **Meestal in LSB (least Significant bit)**
* **Foto**
  + RGB Waarden van een pixel (8 bits, 8 bits, 8 bits)
  + Laatste bit van iedere kleur vervangen door bit van de te verbergen data
* **Geluid**
  + Menselijk gehoor kan maar bepaalde frequenties waarnemen
  + Niet waarneembare frequenties gebruiken om data in te verstoppen
* **In combinatie met cryptografie** kan men dit ook nog eens onleesbaar maken

## Types cijfersystemen

2 Hoofdgroepen:

* **Substitutiesystemen**: vervanging van tekens (confusion)
  + Mono-alfabetische substitutie (Caesar)
  + Poly-alfabetische substitutie (Vigenère)
* **Transpositiesystemen**: veranderen van volgorde (diffusion – permutation)

## Substitutiecijfersystemen

### Caesar substitutie

* Alfabet: ABCDEFGHEIJKLMNOPQRSTUVWXYZ
* Caesar5: FGHIJKLMNOPQRSTUVWXYZABCDE
* X: DITISEENBOODSCHAP
* Y: INYNXJJSGTTIXHMFU
* Aantal plaatsen te verschuiven = sleutel
* Brute aanval-gevoelig #sleutels = 26
  + Zelfs voor 26! Nog gevoelig
  + Gebruik maken van feq – analyse van de taal

### Vigenère cijfersysteem

Vigenère tableau (poly-alfabetisch)

* Met sleutelwoord (key)! Hier gebruiken we crypto als key
* X: dit is een
* K: cry cr cry
* Y: FZR KJ GVL

## Transpositiecijfersystemen

* Veranderd volgorde tekens
* X: een voorbeeld van kolomtranspositie
* K: BLOK (=sleutel)
* Volgorde: 1342 (B komt 1ste in alfabet, dan de K,...)
* X: EENV

OORB

EELD

VANK

OLOM

TRAN

SPOS

ITIE

* Y: EOEV OTSI VBDK MNSE EOEA LRPT NRLN OAOI
* Sparta: scytale
* Alleen met de juiste diameter van de cylinder is het leesbaar !

## Enigma

* Duitsers tijdens WO II
* 3 rotors met 26 tekens (alfabet)
* Initiële instellingen moest gekend zijn
* Caesar met een twist

## Symmetrische Cryptografie (Secret Key)

* Data Encryption Standard (DES)
* 3DES ( Triple DES)
* Advanced Encryption Standard (AES)
* International Data Encryption Algorithm (IDEA)
* Dezelfde geheime sleutel, algorithme is publiek
* Snel (belangrijkste eigenschap)
* Geheimhouding sleutel moet door beide partijen gebeuren, dus geen authenticatie en onweerlegbaarheid mogelijk

## DES

* OUD: 1977 standaard in USA
* Tekst wordt omgezet in binaire waarde (0,1)
* Symmetrisch:
  + Encryptie en decryptie met zelfde sleutel
* Tekst wordt gesplitst in blokken van 64 bits
* Sleutel is 64 bits, met 8 controlebits, dus eigenlijk is K: 56 bits
* Decryptie = encryptie alleen wordt de volgorde van bewerkingen omgedraaid

## DES

* 64 bit invoer wordt gesplitst in 2 x 32bits door een initiële permutatie (IP)
* Geeft een linker blok van 32bits, en een rechterblok van 32bits
* IP = permutatie met permutatie matrix
* Even positites van boven = L
* Oneven posities van onder = R
* 16 iteraties met 16 subsleutels afgeleid van K
* Iteratie
  + L’ = R
  + R’ = L XOR f(R,Ki)
* Op het einde een inverse IP, met een iets andere IP matrix
* Bij iteratie 16 wordt er niet meer geswitched van kant
  + Is om de symmetrie te bewaren, zodat voor decryptie met K16 kan gestart worden, dan K15,...
* F(R,Ki): de kracht van DES!
* voor iedere iteratie wordt er een subkey gemaakt van k = 48 bits
* Om R te kunnen XOR met Ki moet R eerst groter gemaakt worden van 32bits -> 48 bits
* Expansie, ook door een matrix
* Na expansie kan XOR met subkey van 48 bits
* Resultaat = 48 bits -> verdelen over 8 s-boxen
* Iedere S-box krijgt dus 6 bits
* S box neemt die 6 bits en vormt die om tot 4 bits
* Omvorming in de s-box:
  + D.m.v. S-Box tabellen (zijn 8 verschillende)
  + Gebruik tabellen:
    - Neem eerste en laatste bit van de 6 inputbits -> geeft nummer van rij aan
    - Middenste 4 bits geeft kolom aan
* Vb: input s-box: (011011)
* Rij: (0 1) – binair getal, is dus in decimale cijfers 1
* Kolom: (1 1 0 1), decimaal is dit 13
* Resultaat van rij 1 en kolom 13 is 5 ( voor S1 )
* 5 in binair = 0101 = output van S1
* Output van S1 = 4 bits
* Voor de 8 s-boxen krijgen we dan 32 bits
* 32 bits hadden we immers nodig om te kunnen XOR-en
* Het 32 bits resultaat wordt dan nog eens gepermuteerd door een permutatie-matrix (=transpositie) -> van plaats verwisselen / door elkaar schudden
* Decryptie = idem als encryptie (pas op met volgorde van sleutels)
* Input = ge-encrypteerde data
* Volg zelfde schema met IP; maar
* Ipv K1 eerst toe te passen, past men eerst K16 toe, dan K15,..
* Daarna dezelfde IP-1
* De subsleutels:
  + 16 subsleutels van 48 bits worden gemaakt van K die 64 bits is
  + Herinner dat van die 64 bits, 8 controle bits waren dus K=56 bits
  + Hoe worden die 64 -> 56 bits omgezet?
    - Via PC-1 matrix
  + Die 56bits worden weer gesplitst in 2 delen: linker en rechter, ieder 28 bits
  + C en D worden dan geshift naar links met 1 of 2 positites (Afhankelijk van de ronde)
  + Na die shift worden C en D weer samengevoegd
  + Die 56 bits worden dan door PC-2 matrix weer teruggebracht naar 48 bits
* Avalanche-effect:
  + DES is zo goed dat een kleine verandering in X of K een cascade effect genereert en voor een totaal andere Y zorgt (door de 16 iteraties wordt een kleine fout telkens iets groter x16)
* Zwakke sleutels (weak keys)
  + Als C allemaal 1’en is en D allemaal 0’en of omgekeerd
    - * Shift levert niets op want blijft idem
    - Subsleutels blijven dan hetzelfde: E(X,K) = D(X,K)
    - 2x vercijfering tekst geeft dus dezelfde tekst als resultaat
    - Die sleutels zijn wel gekend, dus men gebruikt die niet

## 3DES

* Brute force attack op DES: 2^56 = doenbaar!
  + (brute force) alle mogelijkheden uittesten
* 3DES: 3 X DES met verschillende sleutels
* Meestal: E(X,K1) – D(X’,K2) – E(X’’,K3)
  + Is voor achterwaardse compatibiliteit met 1DES
  + E: encryptie, D:decryptie
  + DES wordt in elektronische chips gemaakt
    - Dus voor 3DES kan men gemakkelijk dezelfde DES-chip gebruiken
  + 3DES: 3 X sleutel van 56 bits = 168 bits
    - = onkraakbaar!

## AES

* Advanced Encryption Standard
* Sleutels van 128, 192, 256 bits
* Bloks van 128 bits
* 10 iteraties met 1 eind iteratie
* Gebaseerd op zijn voorganger DES

# PKI ( public key infrastructure )

## Digitale certificaten

Certificaat verbindt een publieke sleutel aan een individu of organisatie

* Elektronische identiteitskaart
* Uitgereikt door Certification Authority (CA)
  + Thrusted third party
  + Gebaseerd op X.509v3 – standard, ze zijn dus gestandariseerd
* Inhoud certificaat:
  + Public key + extra gegevens
  + Elektronisch ondertekend door CA
  + Door die handtekening garandeert de CA dat de public key in het certificaat effectief bij een bepaald persoon hoort
  + Als je CA vertrouwt, vertrouw je automatisch de inhoud van het certificaat
  + X.509 certificat:
    - Versienummer, serienummer, uitgever, geldigheidsperiode, handtekening CA
    - Verbindt een distinguished name (DN) aan een publieke sleutei
      * DN: serie van naam value pairs
        + OU (organisational unit) = jansen, CN ( certificate name ) = Kiwi wabbe
* Soorten Cerficiaten
  + Persoonsgebonden certificaat
    - Gekoppeld aan een persoon
  + Rolgebonden certificaat
    - Gebonden aan de functie van een persoon binnen een onderneming
  + Toepassingsgebonden certificaat
    - Gebonden aan een applicatie die zelf (veilig) gegevens verstuurd, gaan deze gegevens ondertekenen. Bijv een server die logs stuurt naar admins.
  + SSL-servercertificaat
    - Beveilige verbinding tussen client en server
    - Identificatie van het IP – address en domain name van de server
    - Gebruiker zit op de juiste website en gegevens die hij invult kunnen niet door derden worden afgeluisterd
  + Signing certificaat
    - Ondertekening van software: code afkomsting van ondergetekende

## PKI Componenten !

* Certification Authority (CA)
  + Genereren van certificaten
  + Distributie van certificaten
  + Terugtrekken van certificaten
* Registration Authority (RA)
  + Registreren van gebruikers
  + Vaststellen identiteit van gebruikers
* Validation Authority (VA)
  + Verifiëren of certificaat al dan niet is ingetrokken of opgeschort (CRL)
* Repository/Diretory
  + Opslag van certificaten
  + Opslag blacklist (Certificate Revocation List)
* Eindentiteiten
  + Aanvraag van certificaat
  + Aanmaken van public en private keypair
* Certificate policies and Certificate Practice Statement

## Certificate Policy en Certificate Practice Statement

* Beleid van CA is vastgesteld in 2 beleidsdocumenten
  + CP: ( Certificate Policy)
    - Regels en normen die aan CA wordt gesteld
    - Beleidsniveau
    - Gestandarizeerd document: hoe willen ze uitgifte aanpakken, certificaat formaat-CRL formaat, contractuele bepalingen
  + CPS:
    - De door de CA gevolgde procedures om aan die regels en normen te voldoen
    - Hoe ze de CP concreet invullen
  + Door deze beleidsdocumenten kan de gebruiker bepalen hoeveel vertrouwen hij heeft in het certificaat
    - Er bestaan immers verschillende ‘levels’ van vertrouwen voor verschillende toepassingen (geen superbeveiligingscertificaat gebruiken als je toch gewoon maar wat wilt babbelen over koetjes en kalfjes, maar een mindere ‘vertrouwensklasse’)
    - Vertrouwensklassen zijn gestandariseerd (IETF)

## PKI – architectuur

* 2 soorten architectuur
  + Hierarchisch-architectuur
  + Mesh-architectuur
* Hierarchische structuur
  + Veel toegepast
  + Iedereen kent public key van root CA
  + Root CA:
    - Signeert CA’s van onderliggende levels, die dan op hun beurt weer signeren voor levels daaronder, en uiteindelijk het certificaat kunnen uitreiken op laatste level
  + Public Key van CA juist boven eindleven wordt gebruikt om validiteit te controleren
* Voordeel: duidelijkheid in opbouw vertrouwens-relatie
* Nadeel: compromittering in structuur heeft grote gevolgen (vertrouwen tussen 2 CA’s valt weg -> hele structuur valt in elkaar )
* Mesh PKI-architectuur
  + Alle CA’s hebben gelijkwaardige relaties met elkaar
  + Geen root of hierarchie
  + Iedereen kent public key van locale CA waar hij bij aangesloten is
  + Verificatie kan langs verschillende paden
  + Cross-certificering !
* Voordeel: flexibiliteit, uitvallen van CA is geen probleem
* Nadeel: niet duidelijk hoe de vertrouwensrelatie in elkaar zit
  + Cross-certificering !
    - Verderzijdse erkening van 2 CA’s in een verschillende omgeving (zij vertrouwen elkaar)
    - Op basis van CPS (certificate practice statements)

## Certificaat aanvragen

* A geeft haar public key + persoonlijke gegevens aan RA -> controle -> ok? Volgende stap
* Certificaat voor A wordt aangevraagd bij CA
* CA keurt dit goed, creatie + ondertekening certificaat
* Terugsturen naar A + toevoeging aan public db (directory service)
* B is ook gebruiker->zoekt certificaat A op in db
* B controleert certificaat met public key CA om te kijken of het echt van CA komt
  + Heeft nu public key van A en is zeker dat het van A is
* Communicatie via public key crypto

## Intrekken van certificaat

* Wanneer intrekken ?
  + Vermoeden dat certificaat niet meer integer is
  + Private key compromised
  + Geldigheidsduur verlopen
* Gevolg:
  + Certificaat uit public directory halen (revoke)
* Certificate revocation list (CRL:black list) (!)
* Intrekking wordt gepubliceerd door CA
* Structuur van CRL is volgens standard X.509

## Online Certificate Status Protocol (OCSP) (!)

* Online Realtime controle van certificaat (freshness)
* B Vraagt statusinfo aan OCSP server (naam, CA, serienummer, versienummer) – staat in rechtstreeks contact met CA
* Verwerking door OCSP stuurt terug
  + Good: niet op CRL
  + Revoked: op CRL
  + Unknown: OCSP kent CA niet

## Key Escrow (!)

* VS ziet crypto als een militair aspect
  + Verbiedt de uitvoer van sterke crypto-systemen
  + Zwakke crypto mag wel (korte sleutels)
  + Sterke crypto is soms toegestaan, maar dan moet de gebruiker zijn PRIVATE KEY (decryptie) ook laten beheren door de CA (TTP)
    - Zo kunnen ze dus de communicatie bekijken als ze een gerechtelijk bevel hebben zodat de CA de private key moet geven -> nodig in de strijd tegen terrorisme
    - =Key Escrow
      * BIG BROTHER toestanden
      * Nooit de private signeringskey uit handen geven
      * Key escrow: beheren van private key door TTP
        + Mag onder bepaalde omstandigheden door anderen opgevraagd worden
        + **(Escrow: key in beheer geven aan TTP)**

## Veilige sleutellengten voor de toekomst

* Sleutellengte bepaald veiligheid
* Als sleutel maar beperkte levensduur heeft, hoeft de lengte niet zo groot te zijn
* Er bestaan tabellen die aangeven hoe groot een sleutel moet zijn voor een bepaald systeem, om toch nog veilig te zijn (ook projecties in de toekomst)
  + Uitgedrukt in Mips Jaar: hoeveelheid berekeningen die uitgevoerd kunnen worden in 1 jaar
  + (mips: million instructions per second)

Forward Operation:

hoelang de normale encryptie/decryptie duurt

Inverse Operation:

hoelang een brute force attack zou duren

## De met PKI beveiligde systemen

* Secure e-mail
* Code signing (applets)
* Virtual private networks
* Secure Single Sign-On
  + Gebruiker heeft meerdere applicaties die authenticate verwachten
    - Maar 1 keer inloggen om al die toepassingen te gebruiken

# Beveiliging van het web

Inleiding: het web groeit exponentieel er komen alle dagen veel websites bij, maar dit brengt ook veel risicos met zich mee. Communicatie gebeurt meestal van server -> client. Maar het kan ook gebeuren in speciale gevallen dat er communicatie van client naar server gebeurt. Dit is kwetsbaar en de dag van vandaag is men te weinig bewust van de veiligheids risico’s.

Methoden tot beveiliging van web zijn bijna het zelfde qua services en mechanismen, maar verschillen in toepassingsbereik en plaats op de TCP/IP protocol stack

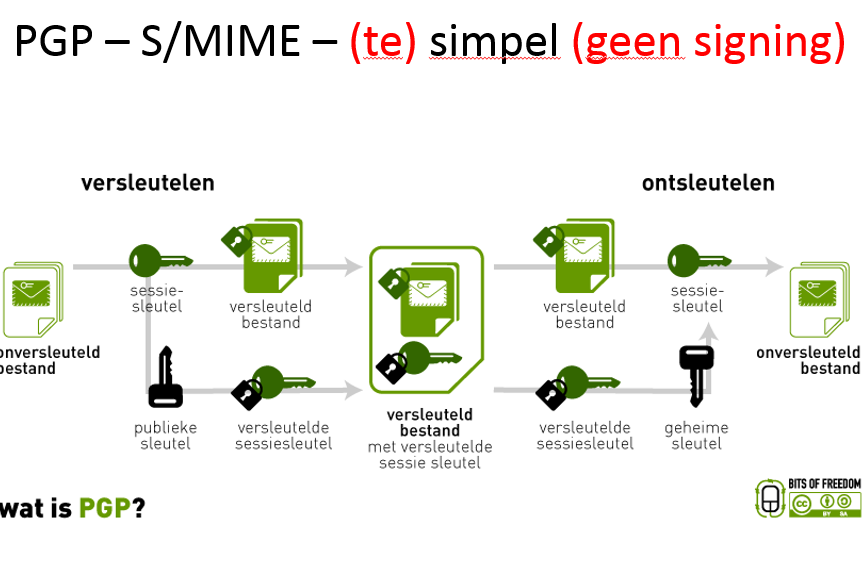
* Methode 1: gebruik van IPSec
  + Voordelen
    - Transparant voor eindgebruikers en toepassingen
    - Universele oplossing
    - Filtermogelijkheid: alleen verkeer dat beveiligd moet worden heeft de IPSec overhead
* Methode 2: SSL en TLS
  + Beveiliging direct bovenop TCP
  + De meeste browsers en webservers hebben SSL geimplementeerd
  + Applicatie moet zo ontworpen zijn om SSL/TLS te gebruiken
* Methode 3: Toepassingsgebonden beveiligingsservices
  + Service kan nu op maat van de toepassing worden gemaakt
  + S/MIME – PGP bijvoorbeeld
* Samenvatting
  + Applicatielaag
    - PGP, SET, S/MIME, Kerberos
  + Transportlaag
    - SSL/TLS
  + Netwerklaag
    - IPsec
  + Data link laag
    - Beveiliging van wireless communication
      * IEEE 802.1X, EAP, Radius, WEP, WPA
      * Wordt niet expliciet gezien in dit vak
    - Beveiliging van ‘gewone’ laag 2 verkeer
      * Hardware-matig, meestal P2P (PPTP:P2P tunneling protoc., of Frame Relay – ATM encryptie)
      * Wordt niet expliciet gezien in dit vak

## Electronic mail

* E-mail veel gebruikte toepassing op internet
* Crypto toepassen op e-mail om zo veilige post te bekomen
* 2 systemen
  + Secure Multipurpose Internet Mail Extension (S/MIME)
  + Pretty Good Privacy (PGP)

## PGP en S/MIME

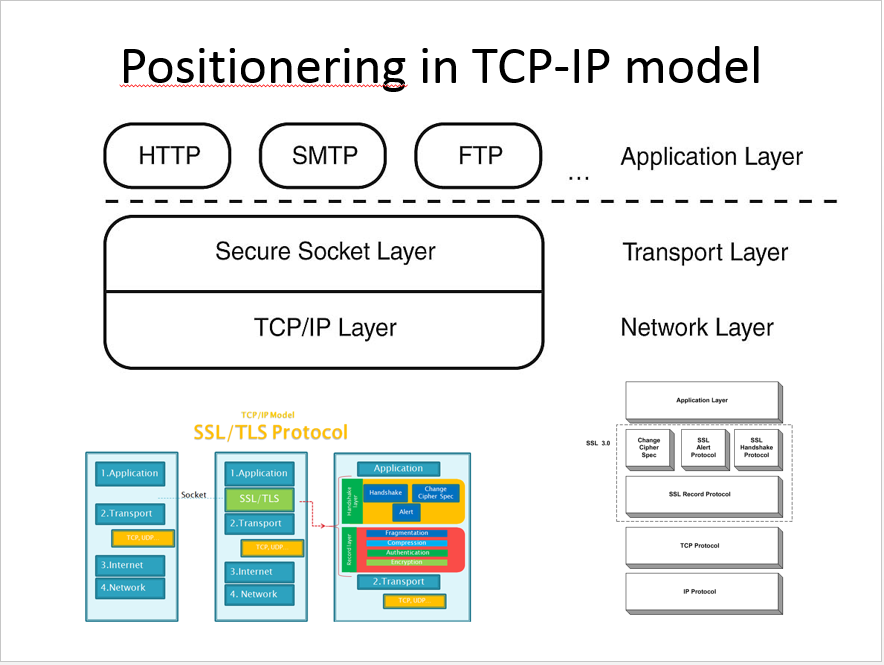
* Gebruiken beide het principe van hybrid crypto om e-mails te versturen
* Grootste verschil
  + Gebruik van certificaten: garantie op de linking tussen ID en public key
* S/MIME
  + Gebruik PKI X.509v3 certificaten, dus uitgereikt door een CA
    - Linking is gegarandeerd door CA
  + Meestal gebruikt in web-based mailing systemen
* PGP:
  + Gebruikt zelf gemaakte certificaten, die vertrouwd worden door een ‘Web of Trust (WoT)’
  + Als X- aantal van uw vrienden een certificaat vertrouwen, dan vertrouw jij dit ook
  + PGP wordt door vele Itérs gebruikt, maar het onderling vertrouwen van certificaten is ’too much hassle’ voor non-IT users



## Secure Socket Layer (SSL) Transport Layer Security (TLS)

* SSL ontwikkeld door Netscape
* SSLv3 is meest gebruikte
* TLS-werkgroep opgericht om SSLv3 te standariseren,
  + Dus in principe is TLS = SSLv3.1
* SSL beveiligt verbinding tussen webbrowsers en webservers
  + HTTPS
    - = http + SSL (en op een andere poort)
* Heel flexibel: #soorten encryptiealgorithmen (=framework)

## Positionering in TCP-IP model



## SSL-architectuur

* SSL gebruikt TCP
  + Om betrouwbare end-to-end service te leveren
* Bestaat uit 2 lagen van protocollen
  + Laag 1
    - SSL record protocol
      * Levert elementaire beveiligingsservices aan hogere lagen, bv aan HTTP
      * Beveiligde data transfer
  + Laag 2
    - Wordt voor het beheer van SSL uitwisseling gebruikt
      * SSL Handshake protocol
      * SSL Change Cipher Sec protocol
      * SSL alert protocol

## Laag 1: SSL Record protocol

* Elementaire beveiligingsservices
  + Vertrouwelijkheid
    - Secret key nr. 1: voor bv AES, in Handshake protocol afgesproken
  + Berichtintegriteit
    - Secret key nr. 2: voor MAC (Message Authentication Code), ook in handshake afgesproken
    - = hash met secret key (secret key = kinda like salt)
      * Pas op: dis is niet hetzelfde als een hash signeren met je persoonlijke secret key (public key crypto)

## SSL Record Protocol

* Record protocol: data afkomstig van Applicatielaag-HTTP

1. Fragmenteert data in hanteerbare blokken
2. Comprimeert (optioneel)
   1. We voegen immers overhead toe voor de security
3. Past MAC toe (integriteit met hash met MAC secret key, nr 2)
4. Versleutelt (bv AES met andere secret key, nr 1)
5. Voegt header toe en stuurt door naar TCP-laag (transport laag)

## SSL Architectuur – Laag 2

* Handshake protocol
  + Authenticate + onderhandelen over te gebruiken cryptoalgorithme
  + Moet in orde zijn voordat er beveiligde communicate kan bestaan
* Change Cipher Spec Protocol
  + Geeft gewoon aan dat de afspraken in orde zijn
    - Vanaf dan beveiligde communicatie zal zijn, dus met een sleutel (onderhandelingen waren dus ‘onbeveiligd’)
* Alert protocol
  + Iets mis in de communicatie -> alert
  + Onderhandelingen zijn afgebroken -> alert
* Bedoeling van handshake
  + Cryptografische parameters afspreken zoals welke encryptie-algorithme (is immers framework), sleutels, ...

## Handshake protocol

* Bedoeling van handshake:
  + Cryptografische parameters afspreken zoals welke encryptie-algorithme, sleutels,...
* 4 fasen:
  + Fase 1:
    - Client stuurt ‘Client hello’ – bericht
      * Parameters
        + SSL versie van de client
        + Sessie-ID
        + Lijst met vercijferingsalgorithmen die de client kent
        + Lijst met compressiealgorithmen die de client kent (zip + hash)
        + Random getal van de client
    - Server stuurt ‘server hello’ – bericht
      * Parameters
        + SSL versie van de server
        + Sessie-ID
        + Gekozen encryptie algorithme
        + Gekozen compressie algorithme (gekozen hash functie)
        + Random getal van de server

Nu weet de server dat ze een sessie eventueel kunnen starten, en met welke algorithmes ze gaan werken, dus gaat zijn certificaat sturen.

* + Fase 2:
    - Servers authenticatie: stuurt
      * Zijn sleutelcertificaat (zit de public key in)
        + Server heeft geen certificaat ?

Stuurt ‘server key exchange’ bericht

Afspraken hoe een master key kan gemaakt worden op andere manier

(client versleutelt later ‘pre-master secret’ met pub key server

Kan gebruikt worden om het ‘master secret’ te maken (=> key voor bv AES ))

* + - * Certificate request
        + Vraagt client zijn certificaat
      * Server Hello-done bericht
        + Hello-bericht gedeelte is afgelopen voor de server

Nu gaat de client nagaan of server-hello acceptabel is (check van certificaat server bij CA/VA) – OCSP

* + Fase 3:
    - Clients authenticatie stuurt:
      * Zijn sleutelcertificaat (zit de public key in)
        + Heeft geen certificaat?

‘no certificate’ – alert naar server sturen

Server neemt beslissing of communicatie verder kan

* + - * Certificate verification-bericht
        + Maakt hash van alle berichten die tot nu toe werden uitgewisseld in de handshake

Versleuteld met prive key van client (=signature)

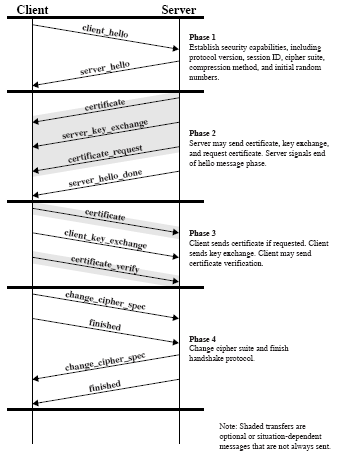
* + - * + Server heeft berichten ook

Berekent eigen hash, decrypt hash van de client, komt overeen

Er zeker van dat client prive key heeft die bij de pub key van certificaat hoort

(server moet dus niet de CA checken van het certificaat van de client

* + - * Client key exchange
        + Client maakt een 48 bytes ‘pre-master secret’ aan
        + Versleutelt dit met publieke sleutel van de server, die hij dan ook terugstuurt
* Gevolg fase 3:
  + Client en server hebben nu beiden de premaster secret
    - Beiden berekenen uit premaster secret de master secret
      * Gebruik random getallen die in het begin werden uitgewisseld (‘hello’)
      * Beiden hebben nu dezelfde master secret
    - Berekenen uit master secret o.a. de encryptie key (ook MAC key)
      * Encryptie key is bv AES secret key
      * (MAC = Message Authentication Code)
        + (soort van beveiligde hash functie, heeft secret key nodig)
  + Pre-master secret -> Master secret -> Encryptie-key + MAC key
* ***Functie van de random getallen***
  + Tegen **replay attack**
    - Aanvaller houdt alle berichten bij die gecommuniceerd worden met de server, en stuurt die op een ander tijdstip gewoonweg terug naar de server -> bv opnieuw een order plaatsen
      * Gaat niet omdat er random waarden in het spel zijn en de server zal een ander random getal kiezen voor andere sessie
  + Fase 4
    - Client stuurt change cipher spec aan server
      * Dus vanaf nu beveiligde berichten gebruiken
      * Stuurt ook een finished-bericht
    - Server stuurt op zijn beurt ook een change cipher spec bericht en een finished bericht



## Virtual Private Networks

* Eigen, afgesloten network dat gebruik maakt van het internet
  + Remote access van medewerker tot bedrijf
  + E-commerce: gedeelte van het netwerk openstellen voor klanten
  + Extranet: partners die mee kunnen inloggen om bv voorraad op te vragen
* Private network:
  + Beveiligd door allerlei mechanismen, dus privé
* Virtual
  + Is geen directe lijn, maar over het internet door gebruik te maken van tunneling techniek = virtueel
* Technologieen
  + PPTP: point to point tunneling protocol (laag 2 OSI)
  + IPSec (laag 3 OSI)
  + SSL?TLS

## IPSec

* Framework of open standards
  + Werkt op netwerklaag
  + Keuze tussen verschillende algorithms
* 4 keuzes te maken

1. Welke IPSec protocol
2. Welke (symm) encryptie
3. Welke authenticatie
4. Hoe afspreken shared key (diffie hellman)

## VPN modes

* 2 Modes
  + Transport mode: behouden van ori IP header
  + Tunnel mode: alles, inclusief IP header verbergen, en het een nieuwe IP header geven
    - (example: private IP addressen, gekend op bedrijfsnetwerk, maar niet op het internet, in een public IP pakket steken)

### VPN Transport mode

* Originele pakket wordt niet geïncapsuleerd in een ander IP-pakket, maar behoudt eigen header
* End-to-End security
* End computers doen de security processing
  + Niet belangrijk om ip addressen te verbergen voor iemand

### VPN Tunnel mode

* Originele pakket incapselen in een ander IP-pakket met een nieuwe IP-header
  + Verberging van de originele IP addressen (=privacy)!
* Als 1 van de eindgebruikers een gateway (router, firewall) is , dan moet tunneling-mode gebruikt worden
* Security voor
  + Network-to-network
  + Host-to-network
  + Host-to-Host

## Secure Electronic Transaction (SET)

* Encryptie op applicatie-laag
* Nieuwe standaard voor betalingen via creditcard over internet
* Zowel verkoper als koper moeten over certificaat beschikken, uitgegeven door CA
* Betrokken partijen bij een SET-transactie (onderstaand deel: lezen)
  + Cardholder: houder van de kaart
  + Issuer: financiele instituut dat de kaart uitreikte
  + Merchant: aanbieder van producten
  + Acquirer: bedrijf dat de merchants autorisatie- en vereffeningsdiensten geeft
  + Payment gateway: SET berichten terug omzetten naar standaard bericht
  + CA: uitreiking certificaten
* Encryptie
  + Combi van symmetrische (DES) en asymmetrische (RSA) crypto (weeral !!!)
    - Bericht versleutelen met DES, dan versleutelen met public key ontvanger via RSA (in envelop steken)
* Digitale handtekening toegevoegd via hash
  + Hash berekenen en versleutelen met private key sender
* Dubbele handtekening !
  + Zender wil 2 berichten versturen in 1 envelop, maar die mogen niet beiden gelezen worden door 1 persoon
  + (dient ook om die berichten aan elkaar te koppelen)
    - Vb: A stuurt order aan merchant, en betalingsinfo aan acquirer

### SET – Dubbele handtekening

* A maakt van 2 berichten hash: h1, h2
* h1 en h2 samenvoegen tot h3
* h3 vercijferen met prive key A (RSA)= dual signature
* Maakt 2 secret keys DES1 en DES2 aan en versleutelt
  + Betalingsinstructies, dual signature, certificaat van A, h2, met DES1 = X1
  + Bestelinformatie, dual signature, certificaat van A, h1, met DES2 = X2
* A stuurt naar Bank (=acquirer)
  + X1 + DES1 versleuteld met de public key van de Bank
* A stuurt naar merchant
  + X2 + DES2 versleuteld met de public key van de merchant

# Firewalls, Virussen en Wormen, Trojans en Rootkits, Sniffers

## Wat is een Firewall?

* Fundament waarop informatiebeveiligings-beleid van meeste organisaties rust
* Niet volledig waterdicht
* Firewall is elk apparaat dat als toegangscontrole-mechanisme voor een bepaald netwerk wordt gebruikt
  + Meestal controle op externe toegang tot netwerk
  + Gebruikt regels (beleid) om toegang te verlenen of niet, meestal via controle op:
    - Bronadres
    - Doeladres
    - Poortnummers
* Extra functionaliteiten in sommige firewalls
  + Filteren van inhoud (blokkering van toegang tot bepaalde sites, ... )
  + Virtual Private Networks (veilige tunnel van A->B)
  + Network Address Translation (Reserved Address Space)
  + Load Balancing (spreiding van belasting)
  + Intrusiedetectie (volledig ander product – belasting firewall?)
  + ...

## Soorten firewalls

* Firewalls die pakketten filteren (routers, cisco,...)
  + L4 OSI Model
* Stateful packet-filter firewalls (Checkpoint, PIX, ...)
  + L5 OSI model
* Firewalls die als proxy fungeren
  + L7 OSI model

## Packet-Filter firewall (L4)

* Meestal routers met pakketfilterfuncties
* Variabelen
  + Bronadres
  + Doeladres
  + Protocol
  + Poortnummer
* Voordeel:
  + Gemakkelijk te implementeren, meestal heeft men al een router
* Nadeel:
  + Niet echt voorbereid op DoS-aanval (is maar simpele router)
  + Kunnen meestal geen sessietoestanden bijhouden
    - Poorten boven 1024 moeten open blijven staan om de TCP sessies en de onderhandelingen netjes te laten verlopen
      * Geen goed idee om open poorten te hebben !!
  + Grote ACLs kunnen bij grote belasting netwerkprestaties verminderen
* Extra uitleg: verschil tussen gewone packetfiltering en stateful
  + Een ACL (Access Control List) is een lijst met allemaal rules (wie mag wat?)
    - Standaard wordt alle verkeer dat van het internet komt, en dat naar het intern netwerk wil, geblocked.
    - Maar als je bv wilt surfen:
      * Bericht vertrekt van intern naar internet
      * Het antwoord komt dan terug van het internet naar intern
        + Geblocked!
      * Oplossing van gewone packetfilters
        + Alle verkeer dat van het internet naar binnen wilt en een destination port heeft groter dan 1024, toelaten
        + Is immers waarschijnlijk een antwoord op een request
      * Oplossing van stateful packetfilters:
        + Houden de sessie bij, en laten enkel het verkeer binnen als de sessie gestart was in het interne netwerk
        + Houdt dus de sockets-info bij (zowel destination als source)

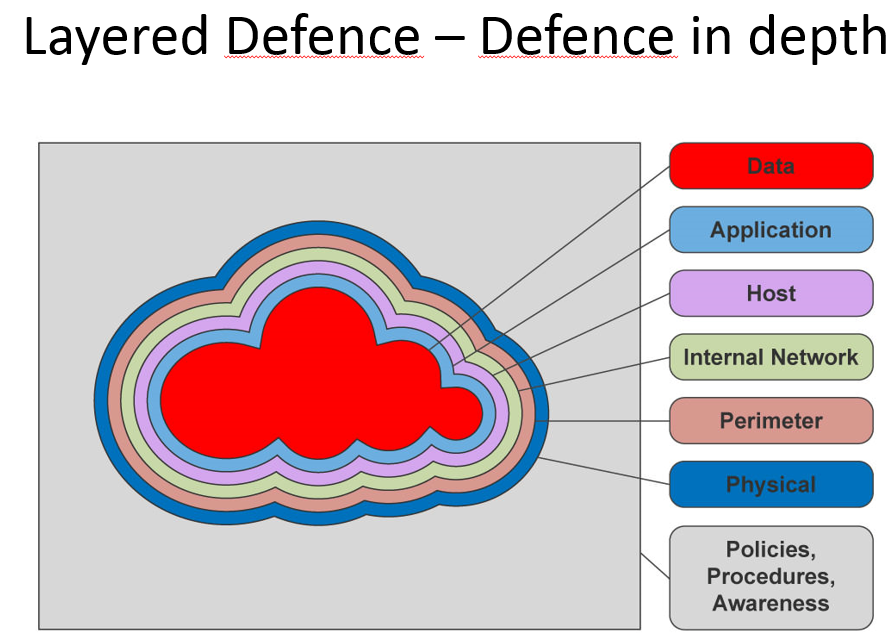
Bij de reply: checken tegen die originele socketinfo

## Proxyfirewalls (L7)

* Inspecteert het verkeer niet alleen op netwerk- en sessieniveau (packet filter firewalls)
  + Maar ook nog eens op toepassingsniveau
  + Deep packet inspection
    - Http pakket arriveert
      * Doorgeven aan HTTP-proxyprocedure
    - FTP pakket
      * Doorgeven aan FTP-proxyprocedure
  + ProxyFirewall is in principe dus veiliger, want begrijpt ook de toepassingsprotocols (HTTP, FTP, SMTP, POP,...)
  + Problemen
    - Meestal trager (belangrijker bij zware belasting netwerk)
    - Nieuw protocol wordt uitgevonden -> geen procedure voor

## Tekortkomingen van firewalls

* Hoe hoger de beveiliging (striktheid van de firewall), hoe minder de functionaliteit
  + Het gewone gebruik van het netwerk kan in gedrang komen
* Firewalls vormen soms een vals gevoel van veiligheid
  + Ze zijn niet absoluut onfeilbaar
    - Zorg ervoor dat het niet de enige line of defence is
      * Zorg voor layered defence
  + Kijk goed de firewall logs na
    - Iemand aan de deur aant rammelen ?



## Virussen en Wormen

### Wat is een computervirus ?

* Een programma dat zich kopieert door andere progs te infectere zodat zij ook een versie van het virus bevatten
* Grootste kenmerk van virus: replicatie
* Infectie:
  + Het zich hechten van virusprogramma aan 1 of meerdere progs op het doelsysteem. Prog start op – virus activeert zich
    - Meestal dus fysiek gehecht aan dat prog, maar hoeft niet
      * Booten kan al genoeg zijn om het virus te activeren
* De meeste virussen volgen de route:
  + User start ligitem prog (User interaction !)
  + Virus die zich genesteld had in de instructiereeks van dat prog, wordt uitgevoerd ipv het originele prog
  + Viruscode wordt beeindigd, en geeft de controle terug aan legitieme prog