Kompiuterių tinklai - Tinklų saugumas

Dešimta paskaita (8 skyrius),
http://computernetworks5e.org/chap08.html

lekt. Vytautas Jančauskas

Kriptografija

Kriptografija (I)

- Kriptografijos tikslas padaryti informaciją nesuprantamą trečioms šalims.
- Iki kompiuterių pranešimų šifravimu užsiimdavo tam apmokyti žmonės.
- Informacija kurią reikia užšifruoti (**plaintext**) perleidžiamą per funkciją parametrizuojamą raktu (**key**).
- Šifravimo rezultas yra **ciphertext**. Jis ir yra perduodamas.
- Tarsime, kad kažkas yra suinteresuotas sužinoti kokia informacija perduodama.

$$C = E_K(P) \tag{1}$$

$$D_K(E_K(P)) = P \tag{2}$$

Kriptografija (II)

- Vienas pagrindinių principų saugume yra manyti, kad šifravimo algoritmas yra žinomas potencialiems pasiklausytojams.
- Kerchoff'o principas teigia: "Visi algoritmai turi būti vieši; tik raktai yra slapti"
- Kai tariamas saugumas užtikrinamas nuslepiant informaciją apie naudojamus algoritmus tai vadinama "security through obscurity".
- Kadangi raktas yra vienintelis slaptas dalykas, jo ilgis turi lemiamą reikšmę. Kuo ilgesnis raktas tuo galimų raktų yra daugiau. Praktikoje norima bent 256 bitų ilgio raktų.

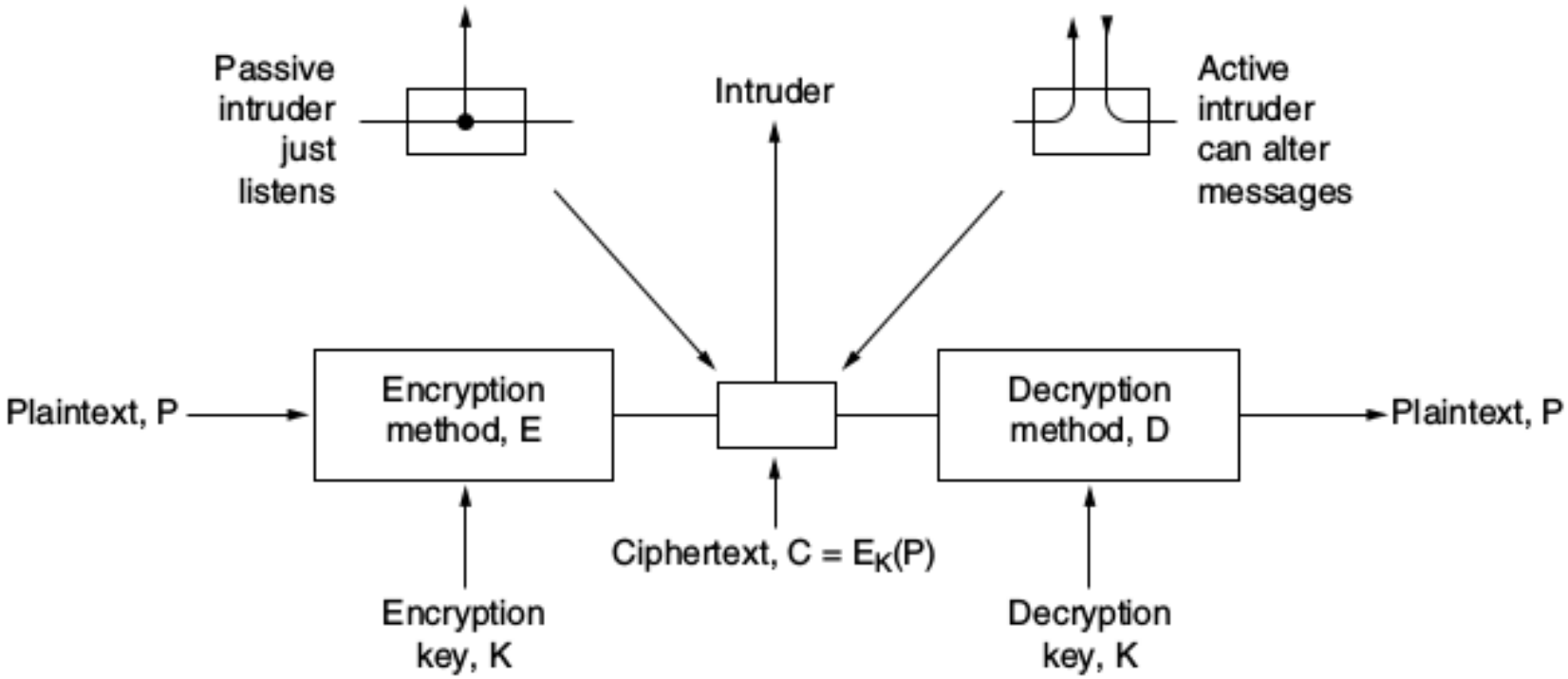

Figure 8-2. The encryption model (for a symmetric-key cipher).

Kriptoanalizė

- Norint atstatyti pradinį tekstą yra trys variantai.
  - **ciphertext-only** problema yra kai turimas tik užšifruotas tekstas.
  - **known plaintext** kai turimas ir užšifruotas tekstas ir originalus pranešimas.
  - **chosen plaintext** kai yra galimybė užšifruoti duomenis pačiam.
- Paprastai manoma, kad vienintelė grėsmė yra **ciphertext-only** atakos. Kitaip tariant jei šifras atsparus tokioms atakoms jis yra saugus.
- Praktikoje taip nėra. Kriptoanalitikas, pvz., gali neretai atspėti **plaintext** iš konteksto.

Keitimo šifrai

- Paprasti ir dažnai naudojami šifrai pakeičia raides ar raidžių grupes kitomis raidėmis ar raidžių grupėmis.
- Cezario šifras yra kai raidės pastumiamos į kairę per 3 vietas. Pavyzdžiui a tampa D, b tampa E ir t.t.
- Bendru atveju galima pastumti per kitą skaičių vietų. Šiuo atveju raktas yra per kiek raidžių pastumta.
- Geresnis variantas kai kiekvienai raidei priskiriama kokia nors kita raidė.
- Jis paprastai įveikiamas išnaudojant statistinius raidžių dažnius. Atspėjus keletą raidžių pranešimą atšifruoti tampa labai lengva.
- Kitas variantas yra bandyti surasti žodžius kurie tikėtina būs pranešime.

Transpozicijos šifrai

- Transpozicijos šifrai palieka raides tas pačias, tačiau sumaišo jų vietas.
- Paprasčiausiu atveju pranešimas surašomas į kažkokį skaičių eilučių, o jų stulpeliai sumaišomi vietomis. Galima naudoti slaptažodį ir sumaišyti pagal jo raidžių vietą abėcėlėje.
- Kriptoanalitikas turi atspėti, visų pirma, kad pranešimas užkoduotas transpozicijos šriftu. Tai galima padaryti pagal raidžių dažnius.
- Tada reikia atspėti, kiek stulpelių buvo naudojama. Tai galima padaryti atspėjus pranešime pasitaikantį žodį.
- Žinant stulpelių skaičių galima tiesiog išbandyti visus variantus. Arba žiūrėti kurios stulpelių pozicijos duoda tikėtinas raidžių kombinacijas.

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Figure 8-3. A transposition cipher.

Vienkartinis šifras

- Sukurti nenulaužiamą šifrą yra labai paprasta.
- Pranešima konvertuojame į bitų seką, sugeneruojame raktą atsitiktinai susidedantį iš tiek bitų kaip ir pranešimas.
- Atliekame XOR pabičiui tarp pranešimo ir rakto.
- Gautas užkoduotas pranešimas yra atsparus visoms įmanomoms dabartinėms ir ateities atakoms.
- Kodėl?
- Kokie jo trūkumai?

Message 1:	1001001	0100000	1101100	1101111	1110110	1100101	0100000	1111001	1101111	1110101	0101110
Pad 1:	1010010	1001011	1110010	1010101	1010010	1100011	0001011	0101010	1010111	1100110	0101011
Ciphertext:	0011011	1101011	0011110	0111010	0100100	0000110	0101011	1010011	0111000	0010011	0000101
Pad 2:	1011110	0000111	1101000	1010011	1010111	0100110	1000111	0111010	1001110	1110110	1110110
Plaintext 2:	1000101	1101100	1110110	1101001	1110011	0100000	1101100	1101001	1110110	1100101	1110011

Figure 8-4. The use of a one-time pad for encryption and the possibility of getting any possible plaintext from the ciphertext by the use of some other pad.

Simetriniai šifravimo algoritmai

- Simetrinio rakto algoritmai naudoja tą patį raktą tiek šifravimui tiek dešifravimui.
- Nagrinėsime blokinius šifrus, kurie užkoduoja n-bitų teksto į n-bitų šifruoto teksto.
- P-box sukeičia bitus vietomis.
- S-box pakeičia vieną bitų seką kita.
- Šios dėžutės gali būti kombinuojamos.

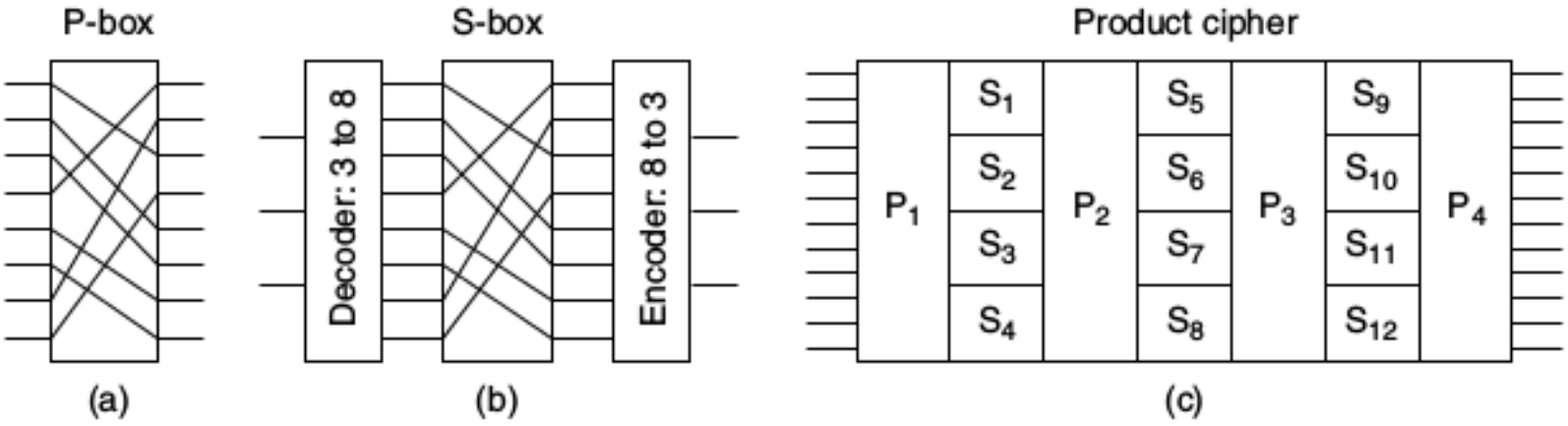

Figure 8-6. Basic elements of product ciphers. (a) P-box. (b) S-box. (c) Product.

DES - The Data Encryption Standard

- DES buvo sukurtas 1977 JAV vyriausybės užsakymu IBM.
- Duomenys šifruojami 64 bitų blokais.
- Algoritmas susideda iš 16 identiškų iteracijų.
- Kiekvienos iteracijos metu 56 bitų raktas yra pakeičiamas.
- IBM sukūrė algoritmą naudojantį 128 bitų raktus, tačiau po pokalbio su NSA rakto ilgis susitraukė iki 56 bitų.
- Šiuo metu DES yra gana nesunkiai atšifruojamas naudojant nedidelį kiekį plain ir cipher teksto.

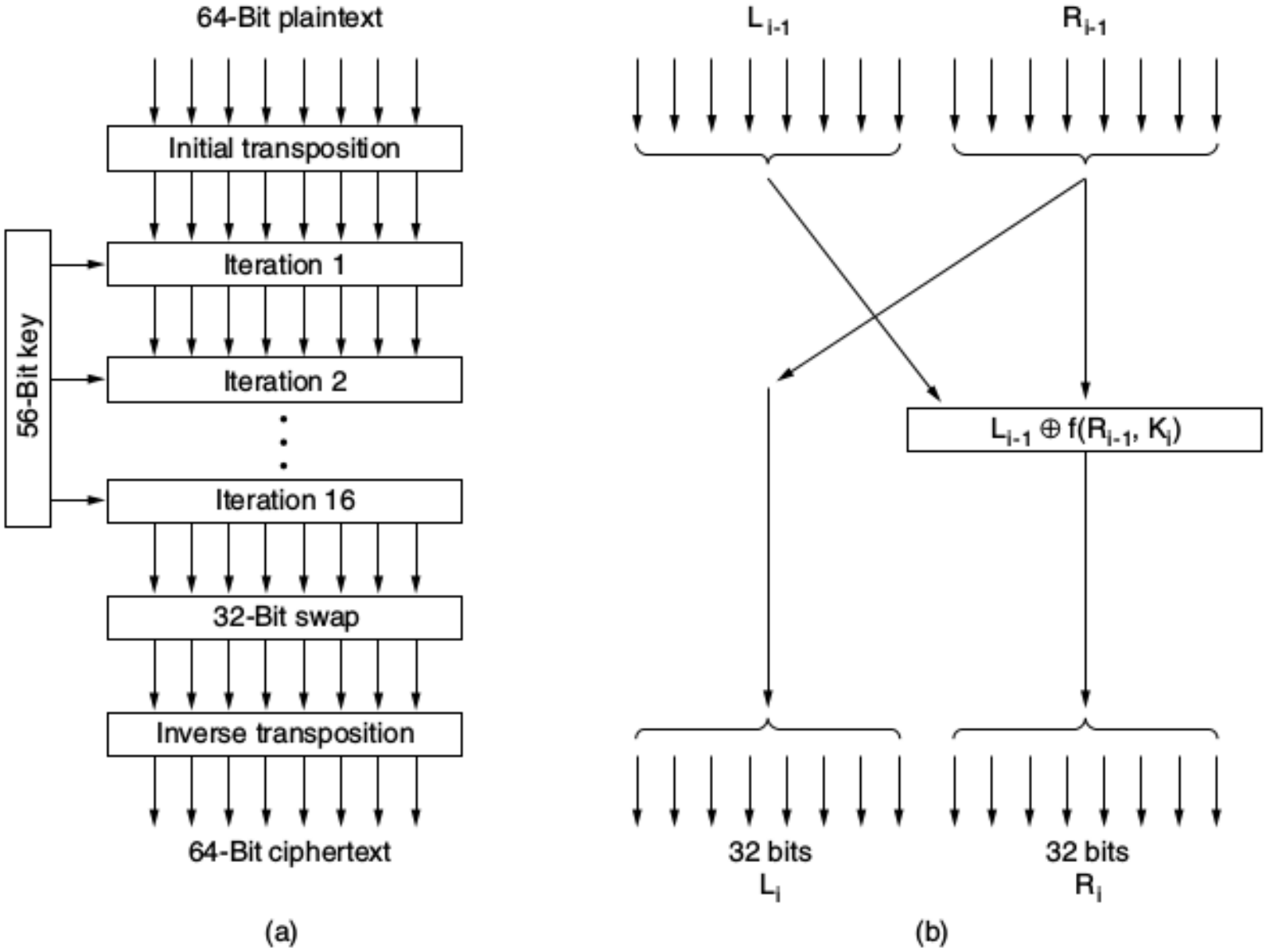

Figure 8-7. The Data Encryption Standard. (a) General outline. (b) Detail of one iteration. The circled + means exclusive OR.

Triple DES

- Kad DES raktas yra per trumpas buvo suprasta jau 1979 metais.
- Pasiūlyta labai paprasta idėja kaip panaudoti tą patį DES, tačiau rakto ilgį padvigubinti iki 112 bitų.
- Naudojami trys DES iš eilės.
- Pirma užkoduojama naudojant K_1, tada atkoduojama naudojant K_2 ir vėl užkoduojama naudojant K_1.
- Atkoduojant pirma atkoduojama naudojant K_1, užkoduojama naudojant K_2 ir atkoduojama naudojant K_1.
- Norint padaryti DES ir Triple DES suderinamus galima naudoti Triple DES su $K_1 = K_2$.

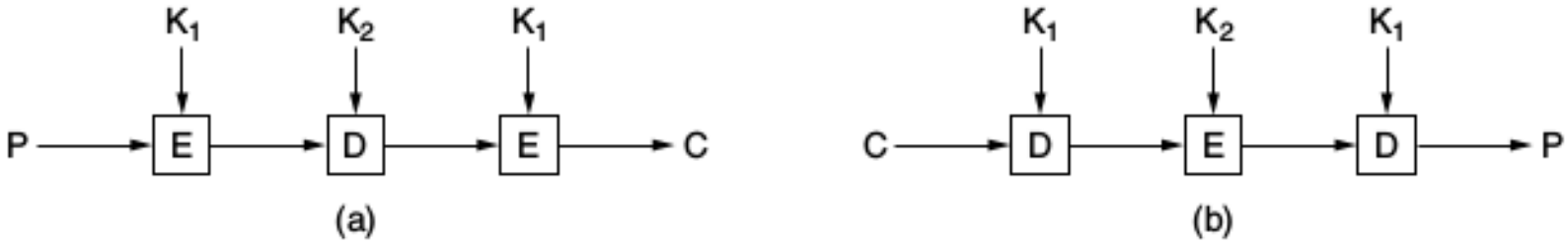

Figure 8-8. (a) Triple encryption using DES. (b) Decryption.

AES - The Advanced Encryption Standard

- Net su triple DES, DES naudojimas eina į pabaigą.
- Iškilo poreikis standartui leidžiančiam ilgesnius raktus.
- Buvo paskelbtas konkursas sukurti AES (Advanced Encryption Standard)
- Jam buvo keliami šie reikalavimai:
  1. Algoritmas turi būti simetrinis ir blokinis.
  2. Dizainas turi būti viešas.
  3. Turi palaikyti 128, 192 ir 256 bitų ilgio raktus.
  4. Turi būti galima įgyventinti tiek programiškai tiek aparatūriniame lygmenyje.
  5. Algoritmas turi būti viešas ir jo licenzija turi būti atvira.
- Buvo išrinktas Rijndael algoritmas.

Viešo rakto algoritmai

- Pasidalinimas raktais yra istoriškai silpniausia šifravimo taikymo praktikoje vieta.
- Viena iš išeičių yra naudoti skirtingus raktus šifravimo ir atšifravimui.
- Reikia, kad nebūtų galima atstatyti vieno rakto naudojant kitą.
- Tokiu atveju žmogus paskelbia savo viešą raktą. Kiti šifruoja juo pranešimus ir siunčia jam.
- Atšifruoti tokius pranešimus galima tik turint privatų dešifravimo raktą.

RSA

- RSA veikimo principas yra paprastas:
  1. Parenkami du dideli pirminiai skaičiai p ir q (paparastai 1024 bitų ilgio).
  2. Suskaičiuojama $n = p \times q$ ir $z = (p-1) \times (q-1)$
  3. Parenkamas santykinai z pirminis skaičius d.
  4. Randamas e, toks, kad $e \times d = 1 \mod z$.
- Žinutė padalinama blokais P, taip, kad P yra skaičius $0 \leq P < n$.
- Užšifruojant skaičiuojama $C = P^e(\mod n)$, atšifruojant skaičiuojama $P = C^d(\mod n)$.
- Metodo saugumas pagrįstas tuo, kad didelių skaičių faktorizavimas yra sunkus uždavinys.

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E
Sender's computation				Receiver's computation		

Figure 8-17. An example of the RSA algorithm.

Skaitmeninis parašas

- Popierinių dokumentų autentiškumą užtikrina parašai (tame tarpe ir teisiškai).
- Sukurti atitikmenį skaitmeniniams dokumentams nėra paprastas ir reikia išspręsti šias problemas:
  1. Gavėjas turi turėti galimybę patvirtinti siuntėjo tapatybę.
  2. Siuntėjas negali vėliau keisti žinutės turinio.
  3. Gavėjas neturi turėti galimybės pats sukurti pasirašytos žinutės.

Simetrinio rakto parašai

- Naudojamas centralizuota institucija per kurią siunčiami pranešimai.
- Alice šifruoja pranešimą P savo raktu ir siunčia BB $A, K_A(B, R_A, t, P)$. Čia B yra Bob tapatybė.
- BB atšifruoja pranešimą, peršifruoja ir siunčia $K_B(A, R_A, t, P, K_{BB}(A, t, P))$.

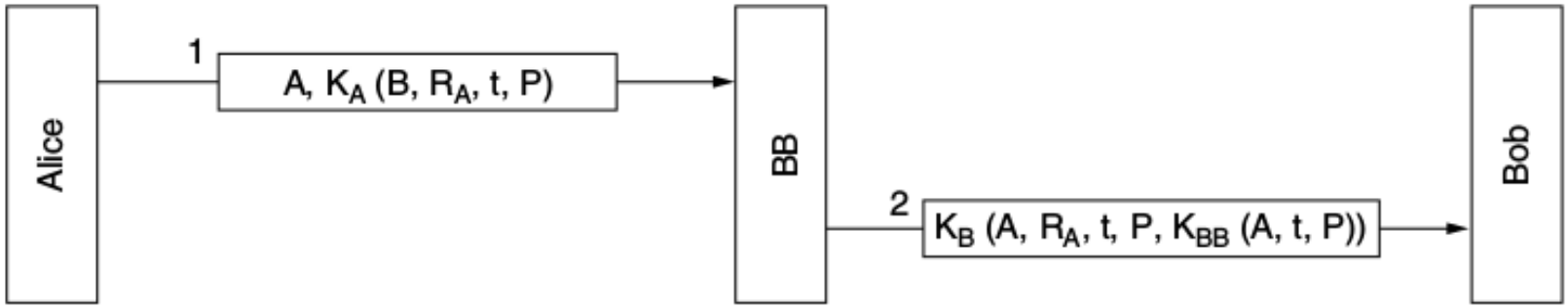

Figure 8-18. Digital signatures with Big Brother.

Viešo rakto parašai

- RSA pasižymi savybe $E(D(P)) = P$ ir $D(E(P))$.
- Alice siunčia Bob $E_B(D_A(P))$. Alice žino savo privatų raktą D_A ir Bob viešą raktą E_B.
- Bob gavęs žinutę naudoja savo privatų raktą ir gauna $D_A(P)$. Išsaugo ir tada naudoja E_A (Alice viešą raktą) ir gauna originalų pranešimą.
- Kas atsitinka jeigu Alice vėliau neigia išsiuntus pranešimą?

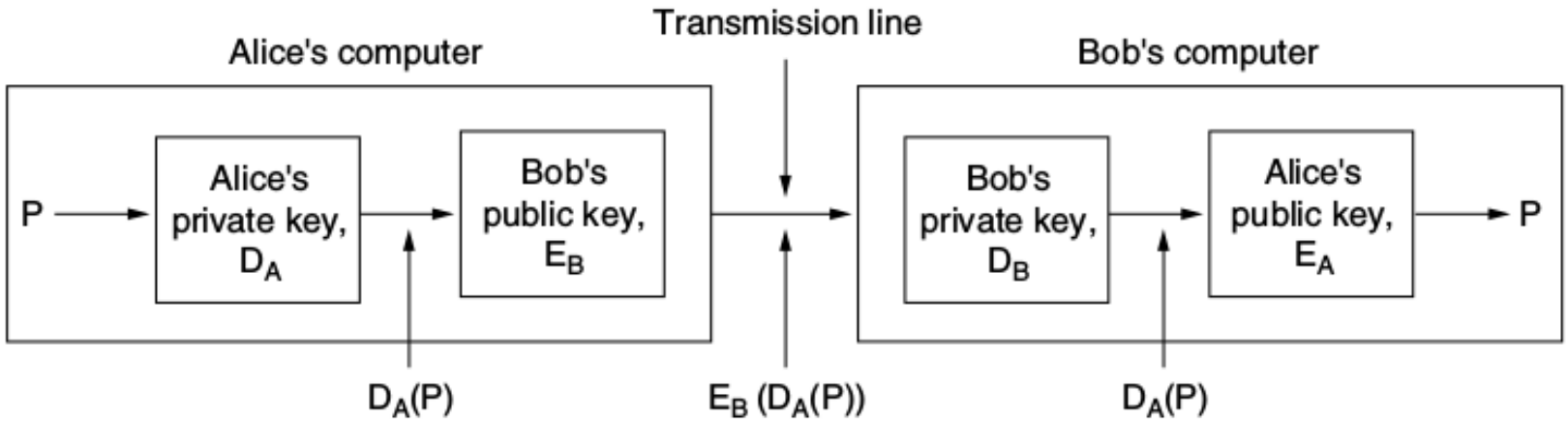

Figure 8-19. Digital signatures using public-key cryptography.

Komunikacijų saugumas

IPsec (I)

- Nemažai saugumo ekspertų mano, kad geriausia “vieta saugumui” yra taikomajame lygije. Tačiau tam reikia, kad būtų perrašytos visos nesaugios taikomosios programos.
- Dalis ekspertų mano, kad naudotojai ir taikomųjų programų programuotojai saugumo nesupranta ir niekada nesupras. Todėl reikalinga užtikrinti saugumą tinklo (IP paketų) lygyje.
- **IPsec (IP security)** aprašytas RFC 2401, 2402 ir 2406.
- **IPsec** nepriklauso nuo konkrečių kriptografijos algoritmų.
- Įdomu tai, kad **IPsec** yra orientuotas į sujungimus. Jie vadinami **SA (Security Association)**. **SA** yra simplex sujungimas tarp taškų ir su juo susieti duomenys.

IPsec (II)

- IPsec gali būti naudojamas dviem rėžimais: transporto ir tuneliavimo.
- Transporto režimu IPsec antraštė įterpiama iš karto po IP antraštės. IP paketo *Protocol* lauke nurodoma, kad tai yra IPsec paketas.
- Tuneliavimo režimu visas IP paketas yra įdedamas į naują IP paketą su nauja antrašte.

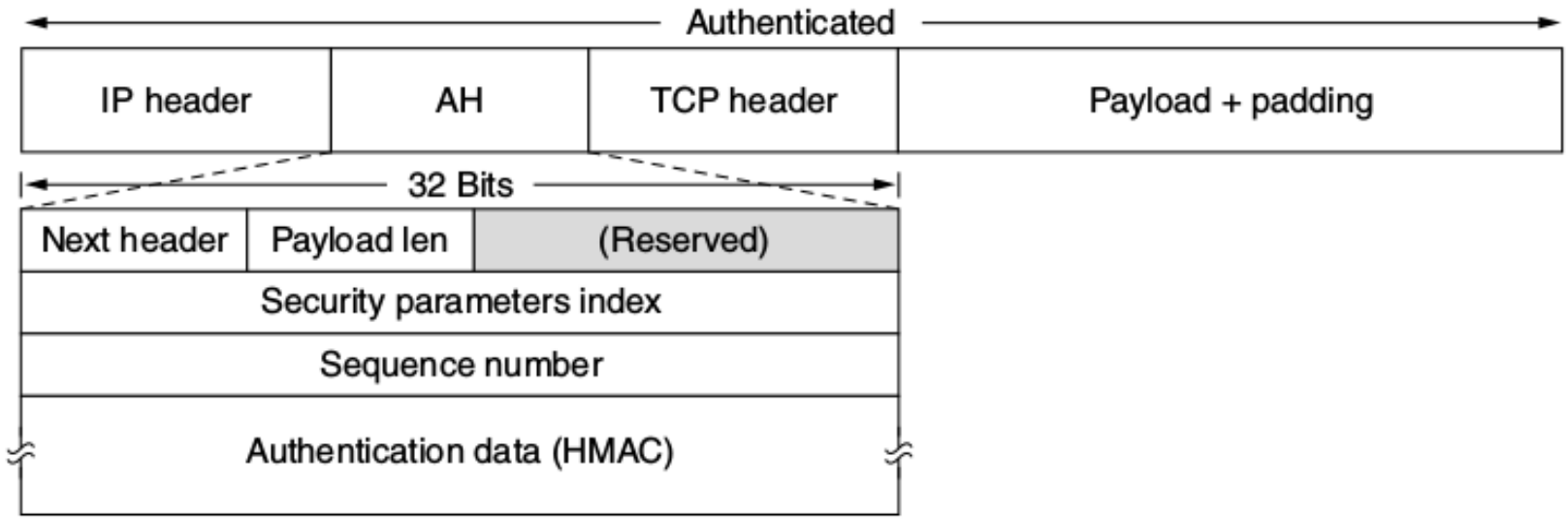

Figure 8-27. The IPsec authentication header in transport mode for IPv4.

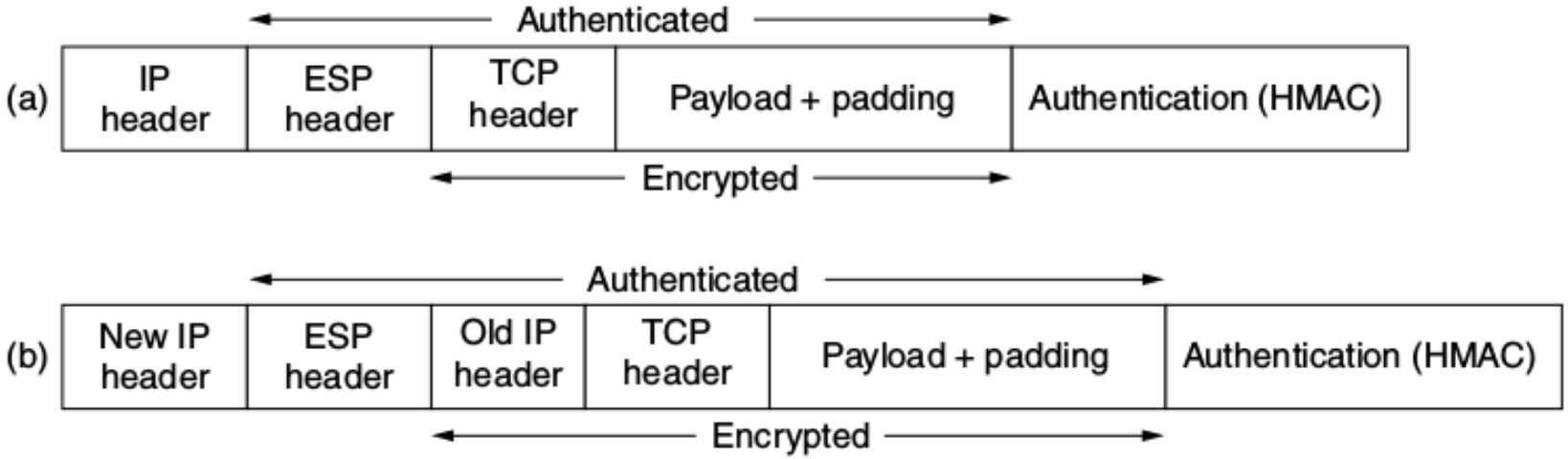

Figure 8-28. (a) ESP in transport mode. (b) ESP in tunnel mode.

Ugniasienės (I)

- Ugniasienės veikia visus į vidinį tinklą ateinančius paketus praleisdamos pro vieną maršrutizatorių ir filtruodamos IP paketus.
- Paketai atitinkantys tam tikrus konfigūracijoje nustatytus parametrus yra praleidžiami, kiti yra atmetami.
- Paprastai taisyklėse yra nurodoma iš ko galima priimti ir kam siųsti IP paketus.
- Norimus portus galima tiesiog užblokuoti visai.
- Galima rašyti ir sudėtingesnes taisykles, kurios atsižvelgtų į taikomojo sluoksnio protokolus. Pavyzdžiui galima blokuoti tam tikrus kriterijus atitinkančius HTTP užklausimus.

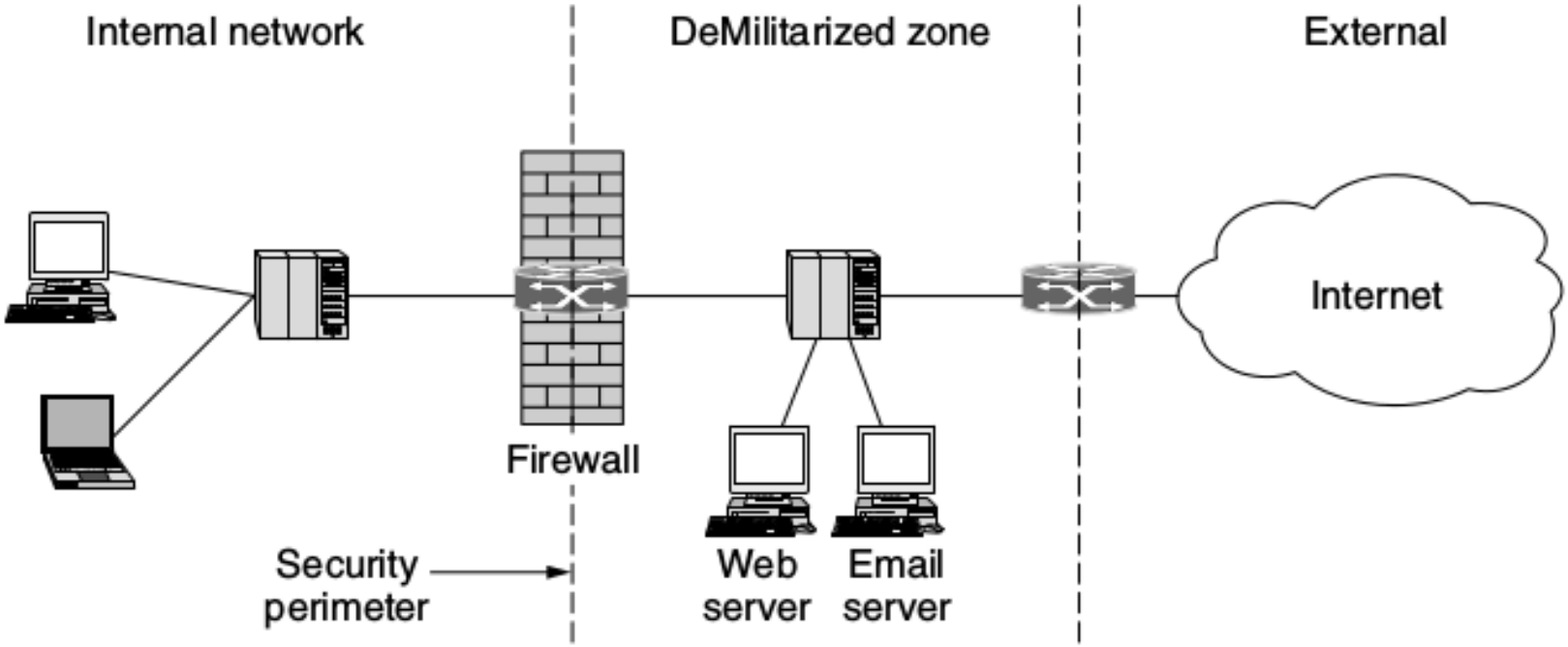

Figure 8-29. A firewall protecting an internal network.

Ugniasienės (II)

- Ugniasienės pažeidžia interneto architektūros principus. Jos yra tinklo sluoksnio įrenginiai, tačiau naudoja informaciją apie transporto ir taikomojo sluoksnio protokolus.
- Niekas netrukdo padirbti IP paketus įrašant į juos IP adresą, kurio ugniasienė nefiltruoja.
- Slaptus duomenis galima išsiųsti iš ugniasienės ginamo tinklo vidaus paštu juos užšifravus ir pan.
- Ugniasienės neapsaugo nuo **DDoS** ir panašių atakų.

Virtual Private Networks

- Neretai kompanijų ofisai yra įvairiose šalyse ar įvairiose šalies vietose.
- Tokiu atveju vistiek norima turėti vidinį kompanijos tinklą.
- Vienas sprendimas yra išsinuomoti privačias komunikacijos linijas (pvz. telefono). Tačiau tai yra brangu.
- Kitas sprendimas yra VPN (Virtual Private Network).
- Naudojant VPN Ineternetu siunčiami IPsec (pvz.) paketai naudojant tunelius.
- Kitiem interneto naudotojams jie atrodo kaip paprasti paketai, tačiau juose yra idėtas ir užšifruotas paketas skirtas vidiniam tinklui.
- VPN naudotojams atrodo, kad kompiuteriai sujungti tiesiogiai.
- Paketų filtravimui naudojamos ugiasienės.

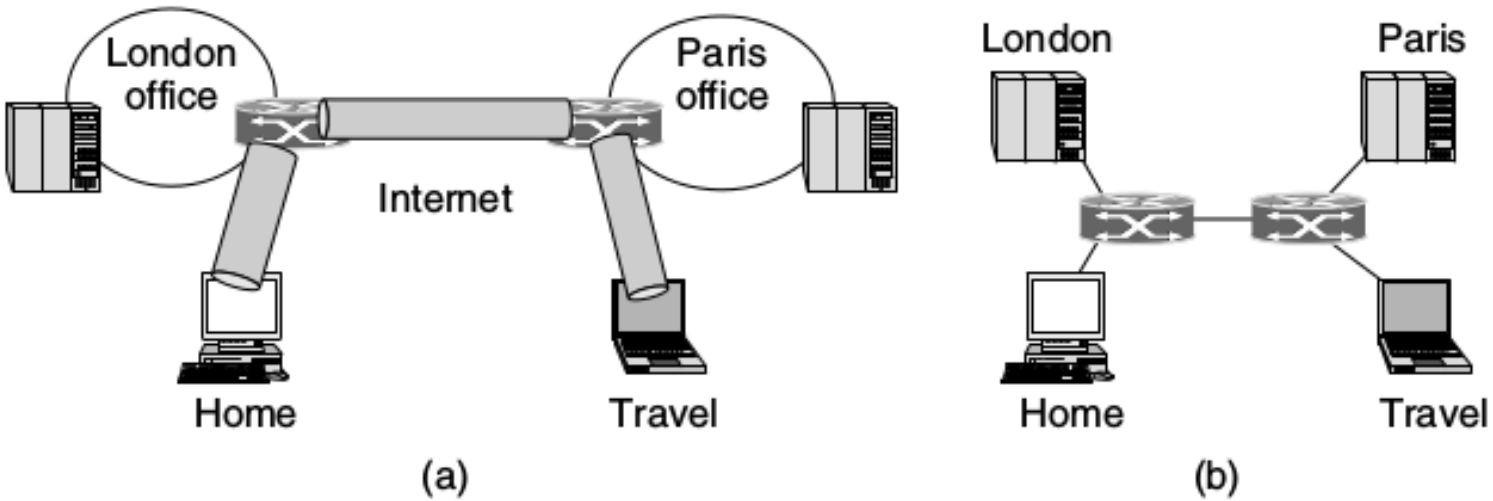

Figure 8-30. (a) A virtual private network. (b) Topology as seen from the inside.

802.11 saugumas

- Bevieliuose tinkluose šiuo metu naudojamas WPA2.
- Naudojant WPA2 namų kontekste naudojamas slaptažodis, kuris padalinamas visiems tinklo naudotojams.
- Šifravimo raktas sesijai yra gaunamas naudojant keturių paketų protokolą.
- Pirma AP ir klientas apsikeičia nonce.
- Antra AP nusiunčia klientui broadcast ir multicast naudojamą raktą, o klientas išsiunčia patvirtinimą.
- Sesijos raktas suformuojamas iš abiejų nonce, AP ir kliento MAC adresų.

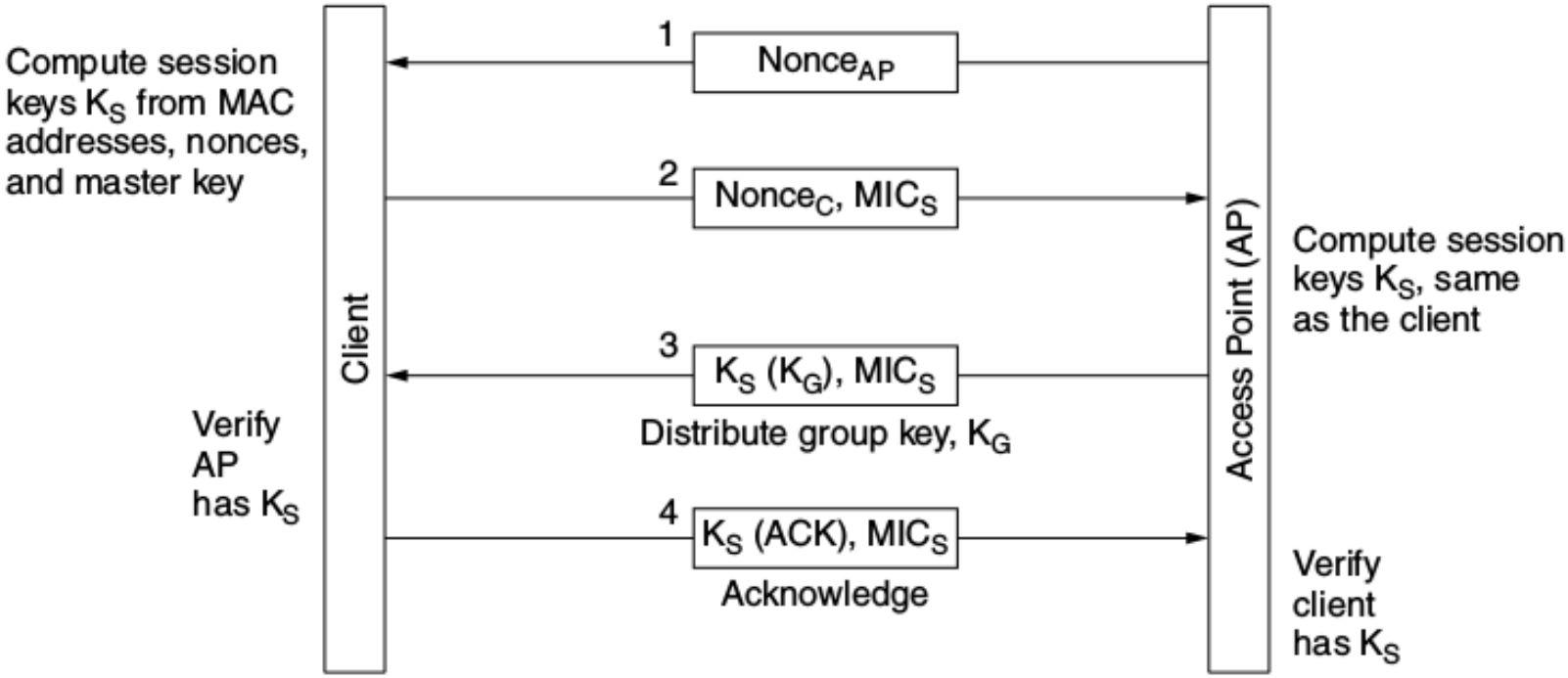

Figure 8-31. The 802.11i key setup handshake.

Bluetooth saugumas

- Bluetooth veikimo atstumas trumpesnis nei 802.11 taigi pasiklausyti pranešimų iš išorės yra sunkiau.
- Tačiau pvz. Bluetooth klaviatūros galima būtų pasiklausyti iš gretimo kambario. Taip pat printerio ir kitų įrenginių gaunamų paketų.
- Bluetooth siūlo saugumą keliuose sluoksniuose.
- Fiziniame sluoksnyje keičiamas perdavimo dažnis naudojant abiem žinomą seką.
- Duomenims perduoti naudojamas šifravimas.
- Įrenginiai yra autentifikuojami.

Saugumo spragos saityne

Grėsmės

- Apie saugumo spragų pasėkmes neretai pranešama žiniasklaidoje.
- Neretai "nulaužiami" organizacijų puslapiai ir pakeičiamas ten esantis turinys.
- Dažnos "Distributed Denial of Service (DDoS)" atakos kurių metu serveriai apkraunami nuolat siunčiamomis užklausomis iš daug kompiuterių, kuriuose veikia specialios programos paprastai įrašytos be savininko žinios.
- 1999 vienas švedas įsilaužė į Microsoft Hotmail puslapį ir padarė laiškus viešai prieinamus visiems.
- Devyniolikmetis rusas Maksimas įsilaužė į elektroninės komercijos svetainę ir vėliau negavęs $100000 išpirkos paviešino 300000 kredito kortelių duomenis.
- 23 metų studentas iš Kalifornijos išsiuntė elektroninį laišką spaudai Emulex korporacijos vardu, kad kompanija patyrė didelių nuostolių tą ketvirtį sukeldamas akcijų kainų nuvertėjimą 60%. Iš to užsidirbo 2 milijardus dolerių.

DNS Spoofing (I)

- Viena iš didesnių saugumo spragų žiniatinklyje yra tai, kad įvedęs į naršyklės langelį adresą, nesi 100% tikras, kad atsidaręs puslapis yra tikrai tas kurio norima.
- Įsilaužėlis perėmęs HTTP GET užklausą, gali gražinti savo puslapį ir tuo pasinaudojęs surinkti duomenis apie užklausėją, pvz. kredito kortelės numerį ir kitus duomenis.
- Tačiau tam reikia, kad įsilaužėlis galėtų patikimai įsiterpti tarp siuntėjo ir gavėjo.
- Tai yra sudėtinga, ypač jeigu naudojama pluoštinės optikos linija.
- Kitas būdas yra naudoti taip vadinamą DNS spoofing metodą.

DNS Spoofing (II)

- Atakos esmė yra tai, kad padaroma jog DNS serveris atsakytų į užklausą klaidingu IP adresu. Kitaip tariant jeigu Alice nori Bob adreso, Trudy padaro taip, kad užklausus DNS serverio Bob IP adreso būtų gražinamas Trudy IP adresas.
  1. Trudy išsiunčia Alice ISP DNS serveriui užklausą *bob.com* IP adresui.
  2. Tuo pačiu paruošia atsakymą kuriame nurodo, kad *bob.com* IP adresas yra, pvz., 42.9.9.9 (Trudy adresas).
  3. Jeigu Trudy sugeneruotas atsakymas pasieks DNS serverį anksčiau nei atsakymas iš com domeną administruojančio serverio į cache bus įrašytas klaidingas IP adresas.
- Toks metodas vadinamas **DNS spoofing**.
- Sugadintas DNS serverio cache vadinamas **poisoned cache**.

DNS Spoofing (III)

- Viena iš papildomų komplijacijų yra tai, kad DNS serveris patikrins IP adresą, iš kurio atėjo UDP paketas. Tačiau Trudy užtenka ten įrašyti atitinkamo DNS serverio (šiuo atveju kažkurio iš *com*) IP adresą.
- Kita komplikacija yra tai, kad DNS serverių atsakymai ir užklausos turi eilės numerius. Pagal juos nustatoma, kurie atsakymai atitinka kurias užklausas. Trudy turi žinoti eilės numerį.
- Tam Trudy sukuria savo domeną *trudy-the-intruder.com* ir DNS serverį *dns.trudy-the-intruder.com*. Abu naudoja IP adresą 42.9.9.9.
- Tada Trudy užklausia Alice ISP DNS serverio, koks yra *foobar.trudy-the-intruder.com* IP adresas. Alice ISP DNS serveris iš *com* DNS serverio gauna, kad *trudy-the-intruder.com* DNS serveris yra *dns.trudy-the-intruder.com*.

DNS Spoofing (IV)

- *dns.trudy-the-intruder.com* įrašomas į Alice ISP DNS serverio cache.
- Dabar Trudy užtenka išsiųsti užklausimą, pvz. koks yra *www.trudy-the-intruder.com* IP adresas. Jame bus eilės numeris.
- Trudy tada išsiunčia atsakyma, kad *bob.com* IP adresas yra 42.9.9.9.
- Taip pat galima siųsti tokius atsakymus su vis didėjančiu IP numeriu, kuris nors atitiks.
- Nuo šiol, kai Alice norės atsidaryti *bob.com* ji gaus IP adresą 42.9.9.9.
- Paprastas būdas viską mums sugadinti yra naudoti atsitiktinius eilės numerius o ne einančius iš eilės. Tačiau ir tokiu atveju eilės numerį galima atspėti bandant daug variantų.

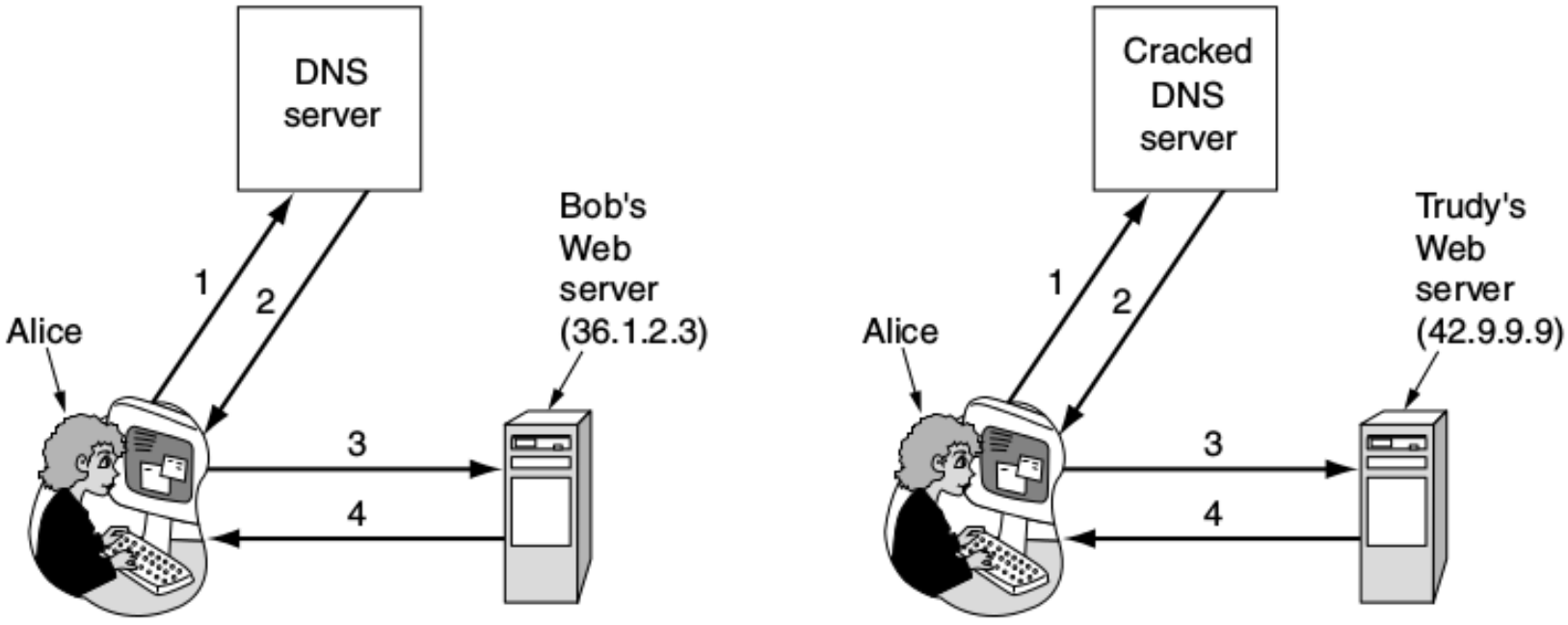

Figure 8-46. (a) Normal situation. (b) An attack based on breaking into a DNS server and modifying Bob's record.

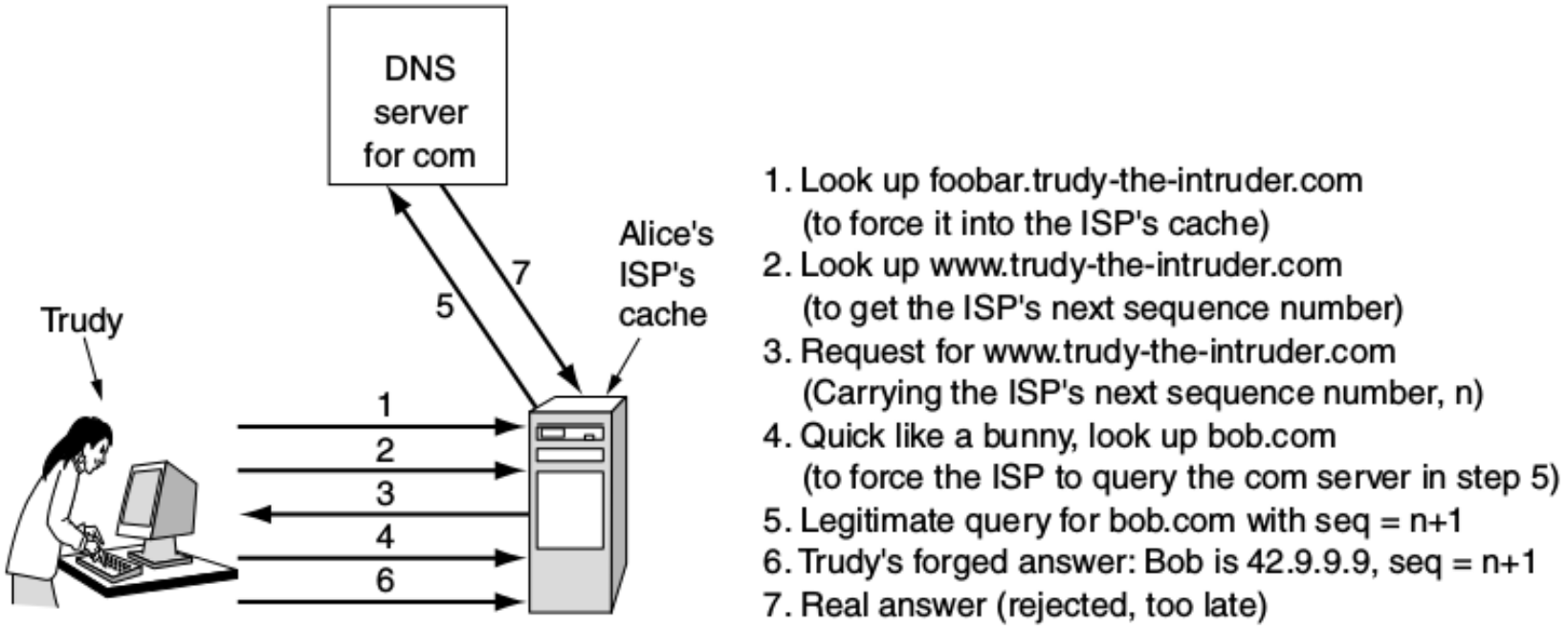

Figure 8-47. How Trudy spoofs Alice's ISP.

Secure DNS (I)

- **DNSsec (DNS Security)** buvo sukurtas kaip atsakas minėtoms DNS saugumo spragoms. **DNSsec** teikia šias paslaugas:
  1. Galima įrodyti duomenų autentiškumą.
  2. Leidžia dalintis viešais raktais.
  3. Tranzakcijos ir užklausų autentifikavimas.
- DNS įrašai grupuojami į aibes vadinamas **RRSets (Resource Record Sets)**. Įrašai, kurių pavadinimas, klasė ir tipas yra vienodi sudedami į vieną aibę.
- Kiekvienam **RRSet** yra paskaičiuojama SHA-1 maišos funkcijos reikšmė. Ši reikšmė yra pasirašoma (užšifruojama).
- Klientas gavęs pasirašytą **RRSet** atšifruoja jį naudodamas zonos viešą raktą ir patikrina **RRSet** maišos funkcijos reikšmę. Jei reikšmės sutampa reiškia duomenys geri.

Domain name	Time to live	Class	Type	Value
bob.com.	86400	IN	A	36.1.2.3
bob.com.	86400	IN	KEY	3682793A7B73F731029CE2737D...
bob.com.	86400	IN	SIG	86947503A8B848F5272E53930C...

Figure 8-48. An example RRSet for *bob.com*. The *KEY* record is Bob's public key. The *SIG* record is the top-level *com* server's signed hash of the *A* and *KEY* records to verify their authenticity.

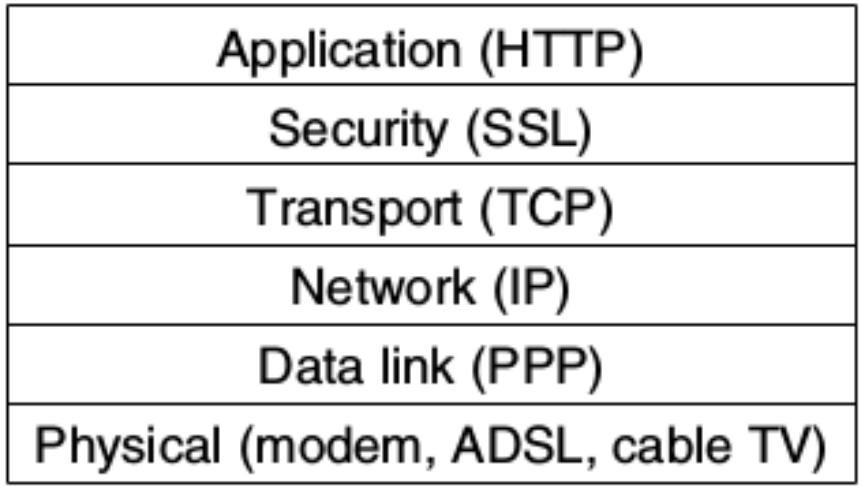

Figure 8-49. Layers (and protocols) for a home user browsing with SSL.

SSL – The Secure Sockets Layer (I)

- SSL buvo sukurtas dėl poreikio turėti saugius sujungimus plintant žiniatinklio naudojimui. Sukurtas 1995 Netscape Communications Corp.
- Sukuria saugu sujungimą tarp dviejų socketų:
  1. Susitaria dėl parametrų tarp kliento ir serverio.
  2. Autentifikuoja serverį klientui.
  3. Leidžia saugų bendravimą.
  4. Saugo duomenų integralumą.
- Įsiterpia tarp taikomojo ir transporto sluoksnio - naudoja transporto sluoksnio socketus ir teikia paslaugas taikomajam sluoksniui.
- Jei HTTP veikia per SSL jis vadinamas **HTTPS (Secure HTTP)** ir naudoja 443 o ne 80 portą.

SSL – The Secure Sockets Layer (II)

Saugių sujungimų sudarymas:

1. Alice išsiunčia Bob užklausimą 1 sudaryti sujungimui. Užklausime nurodoma SSL versija ir kokių Alice pageidauja suspaudimo ir kriptografijos algoritmų. Taip pat skaičių (nonce) R_A.
2. Bob pasirenka algoritmus iš tų kuriuos palaiko Alice. Siunčia savo nonce R_B. Išsiunčia žinutę 2. Žinutėje 3 nusiunčia sertifikatą su savo viešu raktu. Žinutėje 4 pasako, kad dabar Alice eilė.
3. Alice nusiunčia atsitiktinį 384 bitų **premaster** raktą užšifruotą Bob viešu raktu (žinutė 5). Žinutėje 6 Alice nurodo pereiti prie naujo šifro, ir žinutėje 7, kad ji baigė.
4. Bob patvirtina žinutėmis 8 ir 9.
5. Alice dabar žino kas yra Bob, bet Bob nežino, kas yra Alice. Todėl Bob paprašai prašo autentifikuotis naudojant prisijungimo vardą ir slaptažodį.

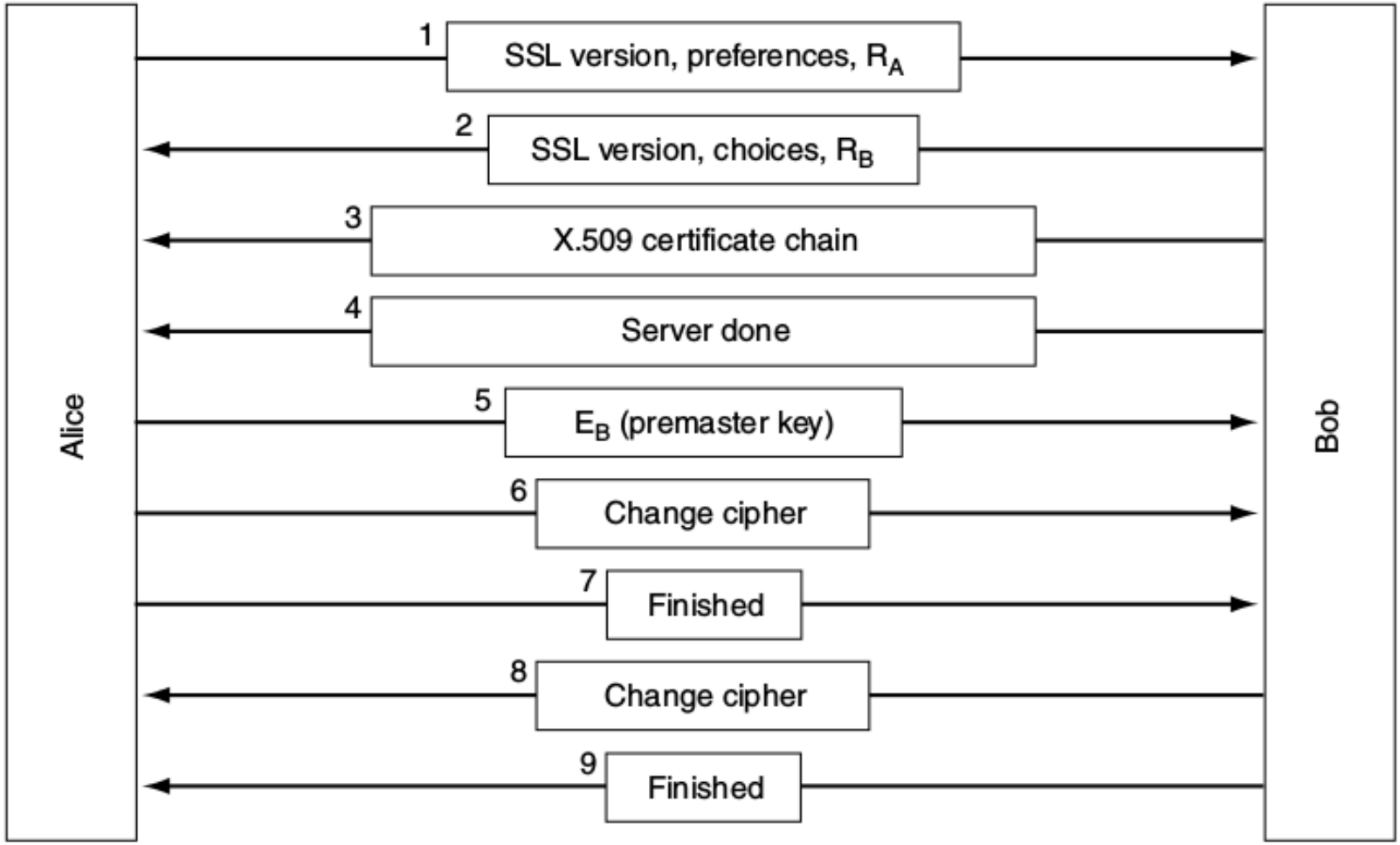

Figure 8-50. A simplified version of the SSL connection establishment subprotocol.

SSL – The Secure Sockets Layer (II)

- Pačiam duomenų perdavimui naudojamas kitas protokolas.
  1. Naršyklės pranešimai padalinami į iki 16KB gabaliukus.
  2. Kiekvienas gabaliukas suspaudžiamas atskirai.
  3. Po to slaptas raktas sudaromas iš dviejų nonce ir premaster rakto.
  4. Šis slaptas raktas sujungiamas su duomenimis ir naudojamas maišos algoritmas (dažniausiai MD5).
  5. Maišos algoritmo rezultatas pridedmas prie kiekvieno suspausdo fragmento. Viskas užšifruojama simetriniu šifravimo algoritmu.
  6. Pridedama fragmento antraštė ir siunčiama TCP sujungimu.

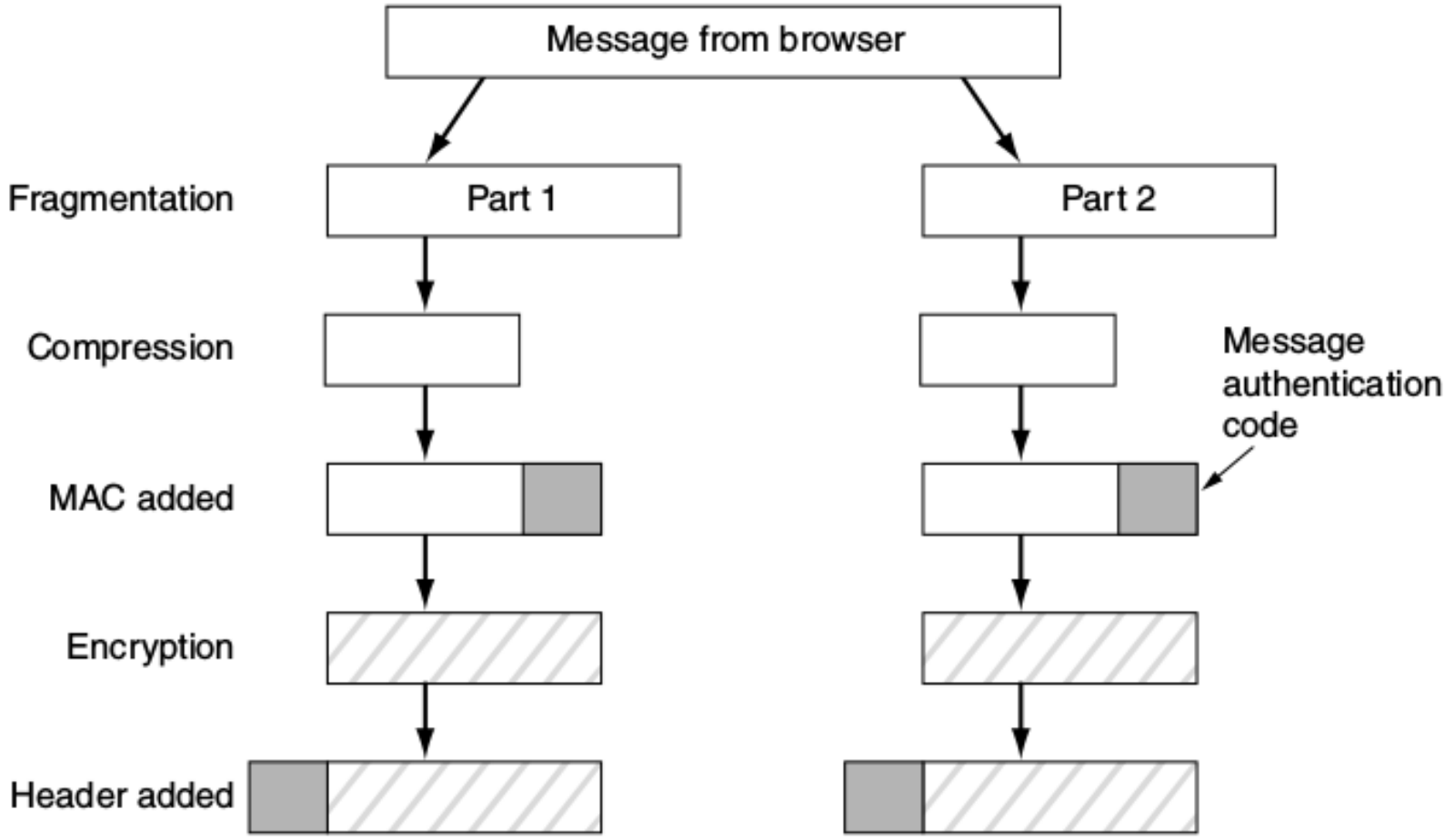

Figure 8-51. Data transmission using SSL.

Java programėlių saugumas

- Java programėlės yra Java programos sukompiliuotos į Java Virtual Machine mašininį kodą ir įterptos į Web puslapius bei užkraunamos su jais.
- Kai užkraunamas puslapis programėlės yra interpretuojamos virtualioje mašinoje paleistoje naršyklės.
- Jeigu programėle pasitikima galima sisteminius kvietinius vykdyti be klausimų.
- Paprastai jeigu programėle bando naudoti sistemos resursus jos kvietiniai yra tikrinami ir arba patvirtinami arba ne. Ar jie leidžiami ar ne priklauso nuo saugumo politikos.

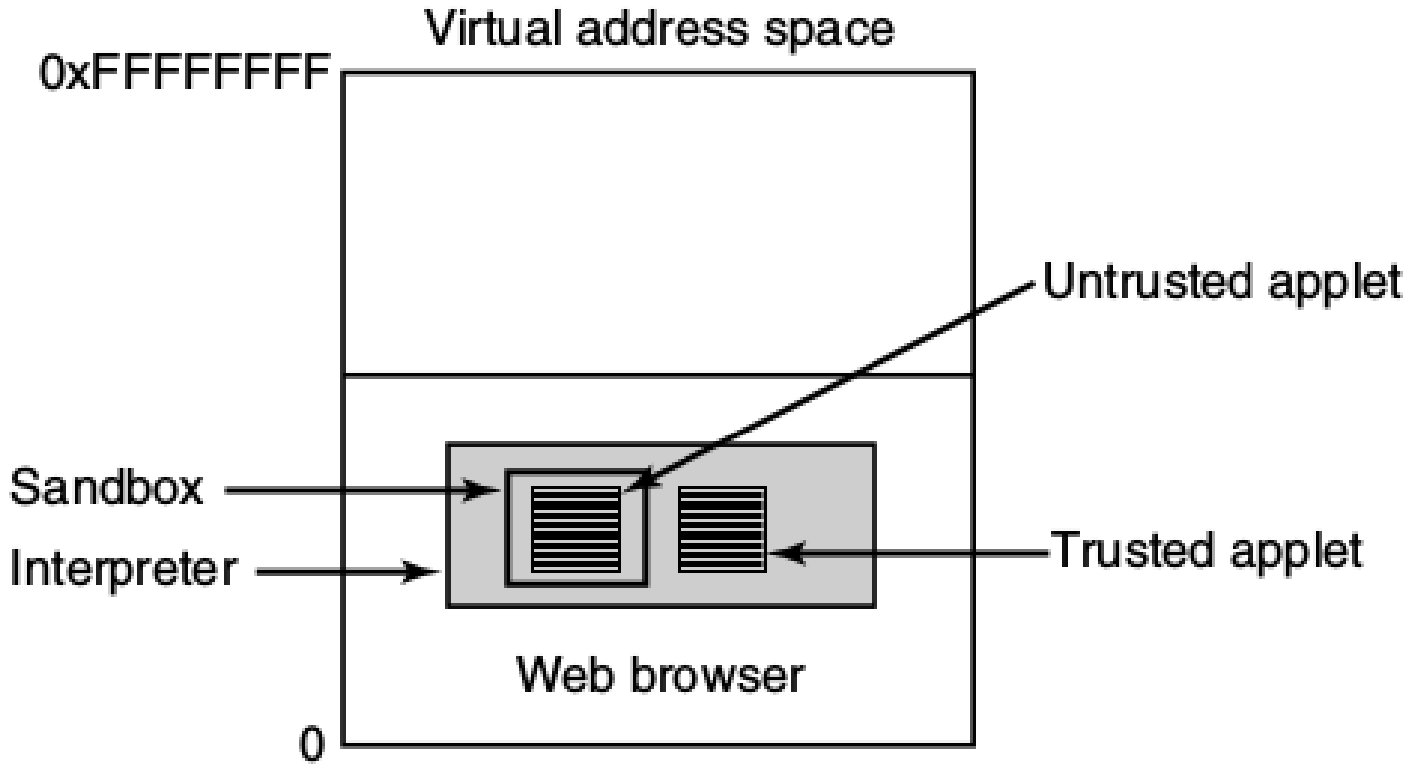

Figure 8-52. Applets can be interpreted by a Web browser.

ActiveX saugumas

- ActiveX yra x86 programos kurios gali būti paleidžiamos naršyklėje. Jos nėra interpretuojamos nei jų sisteminiai kvietiniai tikrinami.
- ActiveX programos yra pasirašomos skaitmeniniu parašu. Jeigu parašo savininku pasitikima programa paleidžiama. Taip pat galima patikrinti ar programos kodas nebuvo pakeistas jau pasirašius.
- Jeigu parašas geras tikrinama ar programos autorius patikimas.
- Programuotojas Sietle įkūrė kompaniją, užregistravo ją kaip patikimą ActiveX programų autorių ir paplatino ActiveX programą išjungiančią kompiuterį.
- Iš principo niekas netrukdo sukurti ActiveX programos kuri sugadintų kompiuterių negrįžtamai.

Naršyklės praplėtimai

- Naršyklės praplėtimai naudojami turiniui vaizduoti naršyklės lange.
- Paprastai kiekvienas MIME tipas gauna savo naršyklės praplėtimą.
- Paprasčiausiu atveju tokia programa tiesiog gali būti sukurtas kaip kenkėjiška.
- Sudėtingesniu atveju praplėtimai skirti duomenų tipams kurie patys yra programavimo kalbos. Pavyzdžiai yra PDF ir Flash.
- Tokiu atveju galioja tas pats, kas ir Java programėlių atveju.
- Gali būti pasinaudota klaidomis tų programavimo kalbų intepretatoriuose.

Virusai

- Skirtumas tarp paprastų programų ir virusų yra tas, kad virusai kuriami taip, kad galėtų kurti savo kopijas.
- Atidarius virusą, pavyzdžiui elektroninio pašto dėžutėje jis bandys save persiųsti į kuo daugiau kitų kompiuterių.
- Paprasčiausiu atveju tiesiog išsiųsdamas save visiem adresų knygos adresatams.
- Kai kurie virusai įsirašo į kietojo disko boot sektorių ir yra paleidžiami kiekvieną kartą užsikrovus kompiuteriui.
- Virusams nėra jokio akivaizdaus sprendimo.

Uždaviniai

Uždaviniai (I)

72. Smulkiai aprašykite kaip veikia DNS spoofing.
73. Aprašykite kaip veikia ir kam reikalingas viešo rakto skaitmeninis parašas.
74. Aprašykite kaip ir kodėl veikia ir kam reikalinga RSA.
75. Kodėl aptartas vienkartinio rakto algoritmas nors teoriškai neįveikiamas yra nenaudotinas praktikoje?

Uždaviniai (II)

76. Aprašykite kas tai yra ir kaip veikia ugniasienės.
77. Aprašykite kaip veikia ir kaip atšifruoti keitimo šifrus.
78. Aprašykite kaip veikia ir kaip atšifruoti transpozicijos šifrus.
79. Aprašykite kokie yra ir kaip veikia 802.11 saugumo protokolai.