Protocoale de Securitate

Securitatea Comunicatiei

- IPsec
- Firewalls
- Virtual Private Networks

IP Security Protocol - IPSec

- Implementat la nivel IP
- Construieste o legatura securizata unidirectionala intre transmitator si receptor
  - numita Security Association - SA
  - asigura
    - autentificarea mesajelor sau
    - autentificarea si criptarea
- Securizarea ambelor sensuri -> 2 x SA

Parametri de securitate

- SA nu este legata de un singur algoritm de criptare sau de o singura cheie – se pot specifica:
  - algoritmul si modul de criptare (ex. DES in mod block-chaining)
  - cheia de criptare
  - parmetrii de criptare (ex. Initialization Vector)
  - protocolul de autentificare si cheia
  - durata de viata a unei asociatii (permite sesiuni lungi cu schimbarea cheii daca este necesar)
  - adresa capatului opus al asociatiei
  - nivelul de senzitivitate al datelor protejate.

SA Database

Un sistem pastreaza o baza de date cu asociatiile de securitate

- Pentru fiecare SA pastreaza parametrii de securitate (slide precedent) si

- **contor** numere de secventa: pentru antete de securitate
- Indicator **overflow** pentru contor numere de secventa: ce-i de facut la depasire limita contor
- fereastra **anti-replay**: determina daca un pachet este o copie
- **Path MTU**: path Maximum Transmission Unit (pentru evitare fragmentare)

SA Database (2)

Fiecare intrare unic identificata de:

– **Security Parameters Index (SPI)**: identificare SA la receptor
– **IP Destination Address**
– **Security Protocol Identifier**

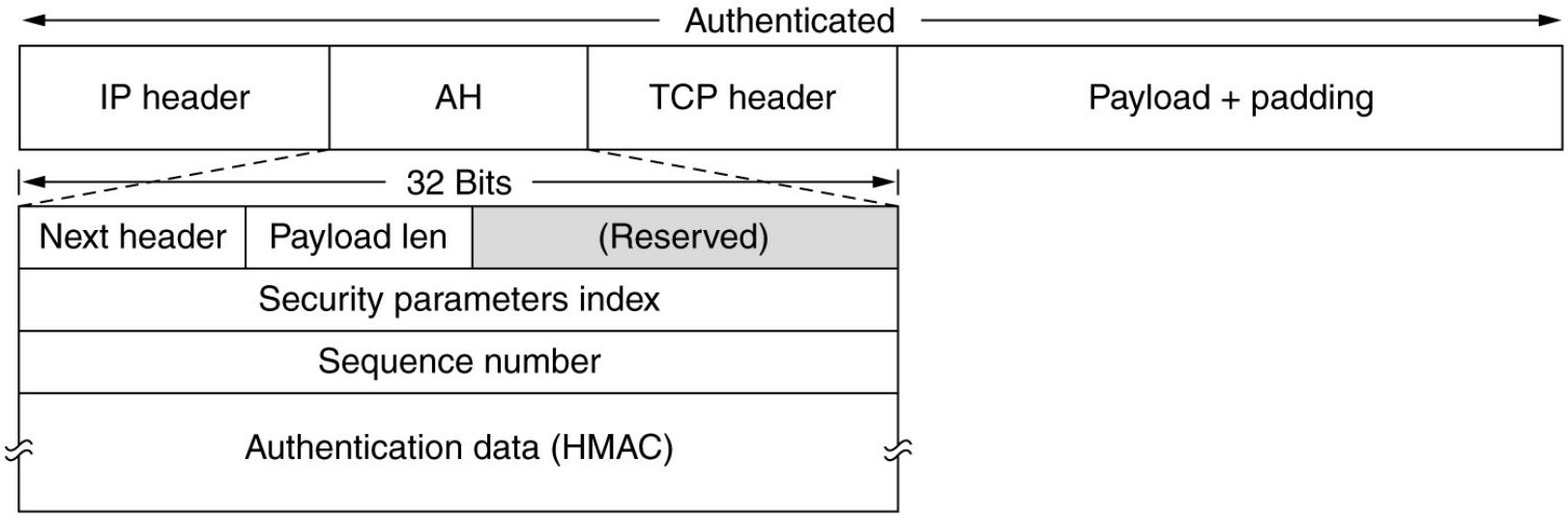
Doua protocoale de securitate:

– AH (Authentication Header) - protocol de autentificare
– ESP (Encapsulating Security Payload) - protocol combinat criptare/autentificare

Si doua moduri de lucru

- transport
- tunel

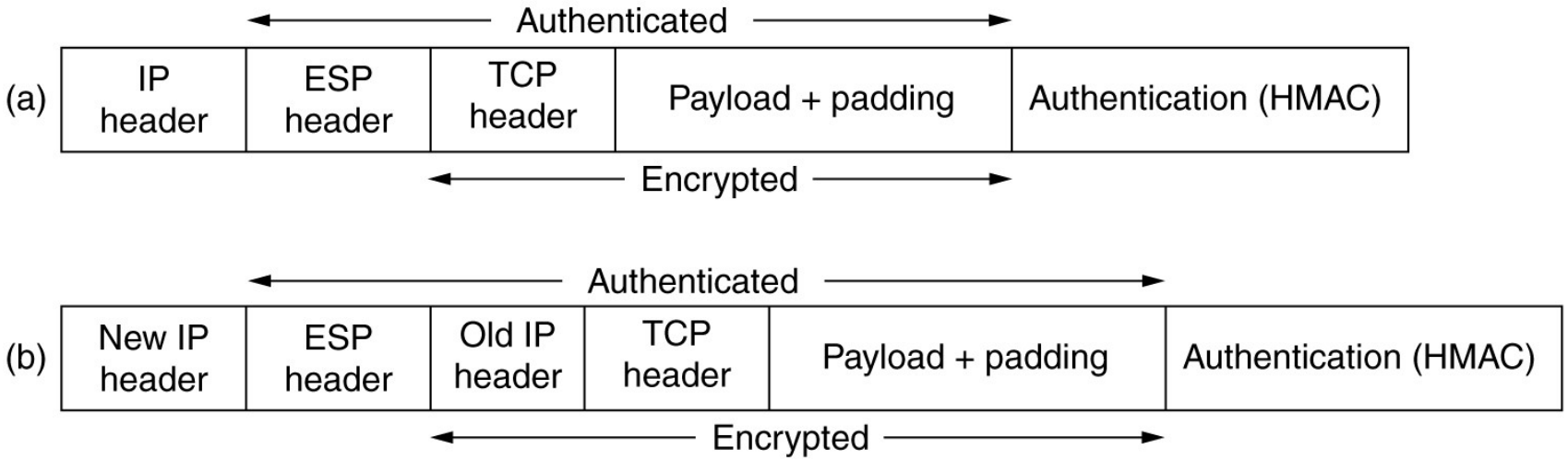
Protocol AH – in mod transport pentru IPv4



Authentication Header – inserat in datagrama IP

- Next header – preluat din IP header unde este inlocuit cu 51
- Payload len – lungime AH (nr cuvinte 32 biti) minus 2
- Security Parameters Index – indica inregistrarea din BD a receptorului
- Sequence number - evitare atacuri prin replica
- HMAC – Hashed Message Authentication Code
    - Utilizeaza cheia simetrica
    - Calculeaza rezumat peste intreaga datagrama (campurile variabile neincluse) + cheia simetrica

ESP in modurile transport si tunel



ESP – Encapsulating Security Payload

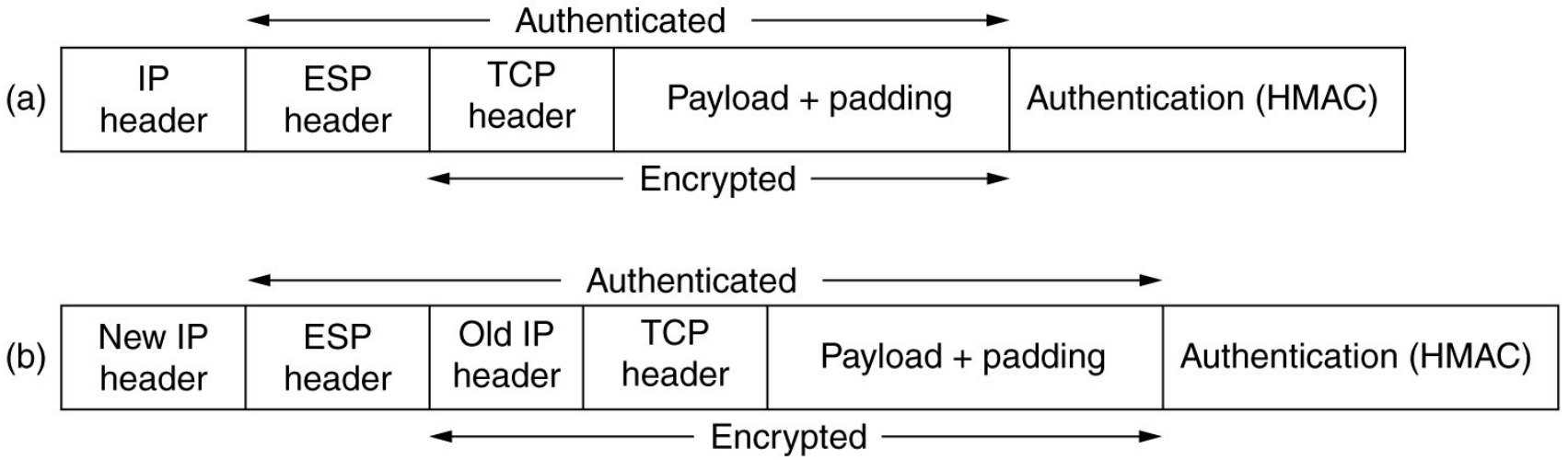
(a) ESP in mod transport.

- antetul ESP este plasat intre antetele IP si TCP
- campul “protocol” din antetul IP este modificat si arata ca urmeaza un antet IPsec

(b) ESP in mod tunel.

- la pachetul IP se adauga antetul IPsec si un nou antet IP
- tunelul se poate termina inainte de destinatie (de ex. la un firewall)

ESP in modurile transport si tunel



ESP – Encapsulating Security Payload

(a) ESP in mod transport. (b) ESP in mod tunel.

- criptarea protejeaza incarcatura;
- autentificarea protejeaza antet ESP + criptograma

ESP header include

Security Parameters Index

Numar de Secventa

Vector de initalizare (pentru criptare date)

La sfarsit: HMAC – Hashed Message Authentication Code

HMAC

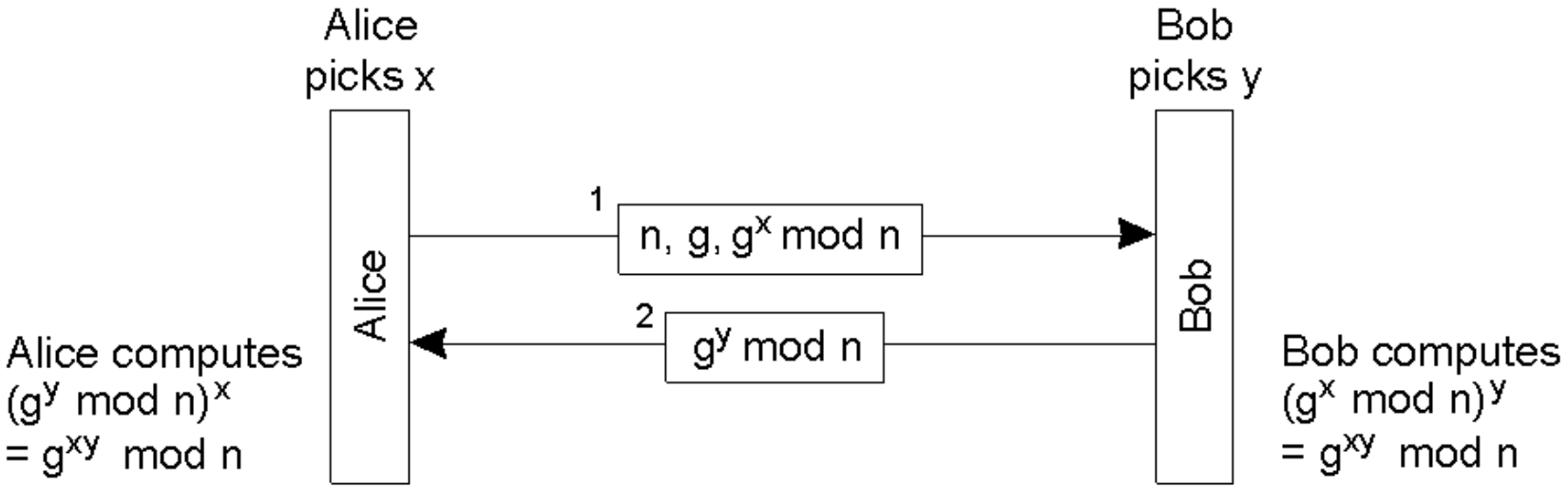
- HMAC = hash(MAC_write_secret XOR pad_2 |
- hash(MAC_write_secret XOR pad_1 | **MSG**))

- Functii hash suportate:
  - MD5
  - SHA-1

- pad_1 este 0x36 repetat
- pad_2 este 0x5C repetat

Gestiunea cheilor

- ISAKMP – Internet Security Association Key Management Protocol
- Genereaza o cheie distincta pentru fiecare asociatie
- Implementat cu IKE (ISAKMP Key Exchange)
  - Foloseste Diffie – Hellman
- Pentru Alice:
  - x este cheia privata
  - $g^x \bmod n$ este cheia publica
  - $K_{A,B} = g^{xy} \bmod n$ este cheia secreta partajata cu Bob

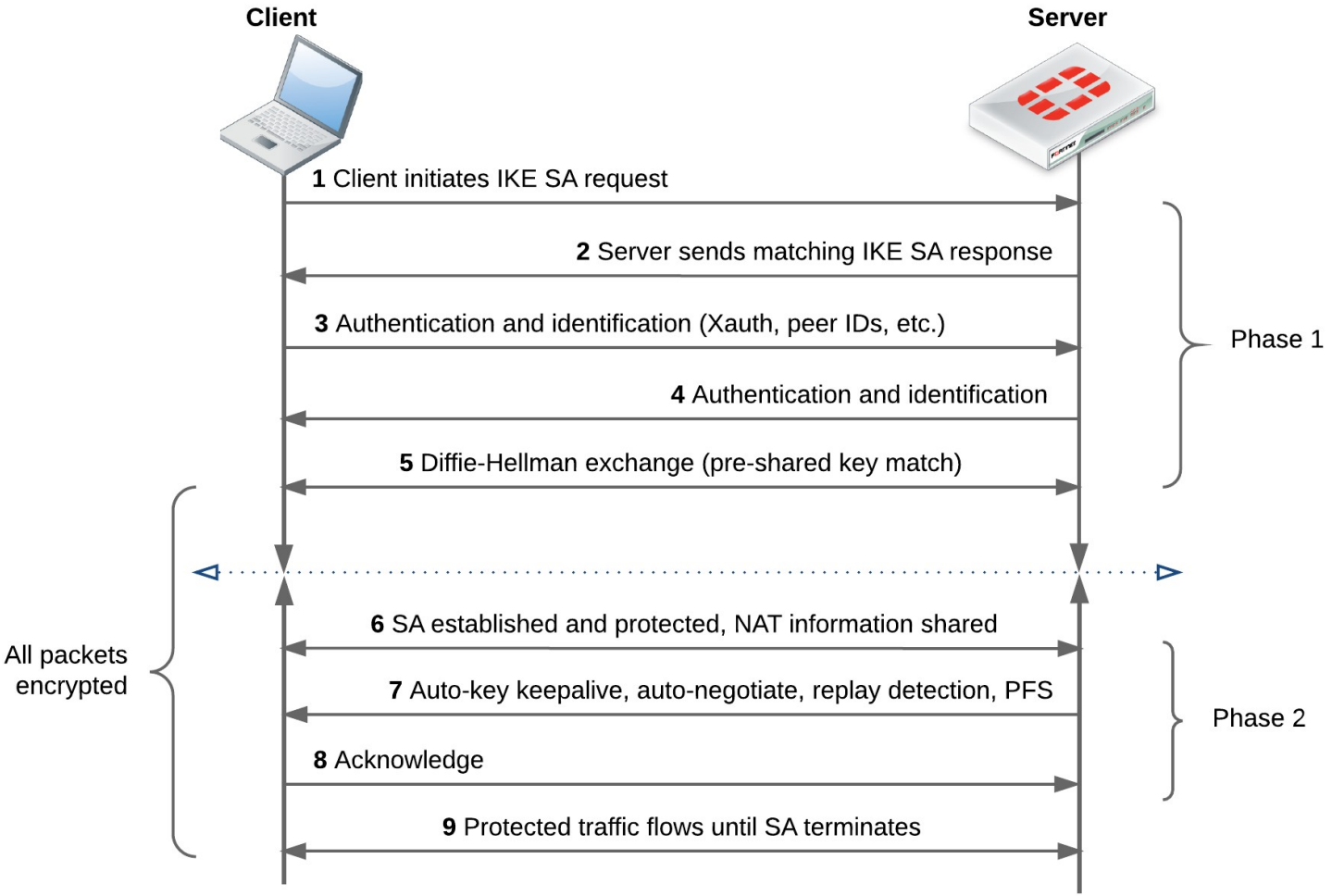


IKE

- IKEv1
  - Phase 1:
    - Clientul trimite o cerere de tipul IKE Security Association (SA)
    - Se trimit identitatile (ID-uri, certificate, etc.)
    - Include schimbul de chei Diffie-Hellman
  - Phase 2:
    - SA este stabilit si securizat
    - Parametrii SA sunt renegociati dupa un interval fix de timp

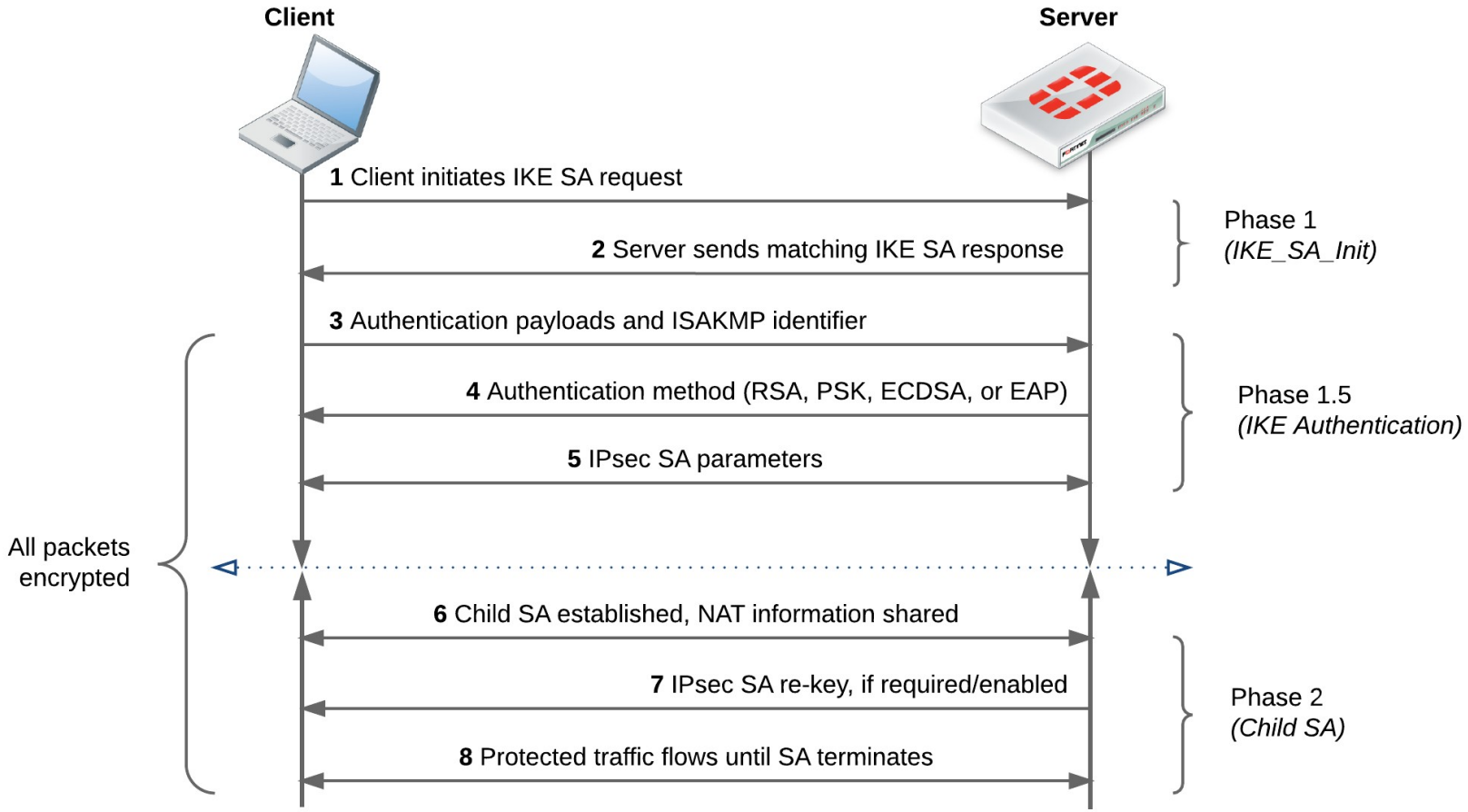
IKE

- IKEv1


Client
Server
1 Client initiates IKE SA request
2 Server sends matching IKE SA response
3 Authentication and identification (Xauth, peer IDs, etc.)
4 Authentication and identification
5 Diffie-Hellman exchange (pre-shared key match)
Phase 1
6 SA established and protected, NAT information shared
7 Auto-key keepalive, auto-negotiate, replay detection, PFS
8 Acknowledge
9 Protected traffic flows until SA terminates
Phase 2
All packets encrypted

IKE

- IKEv2
  - Pasul de autentificare este criptat
  - Poate include protocoale de autentificare diferite
  - Autentificarea este un pas separat (1.5)

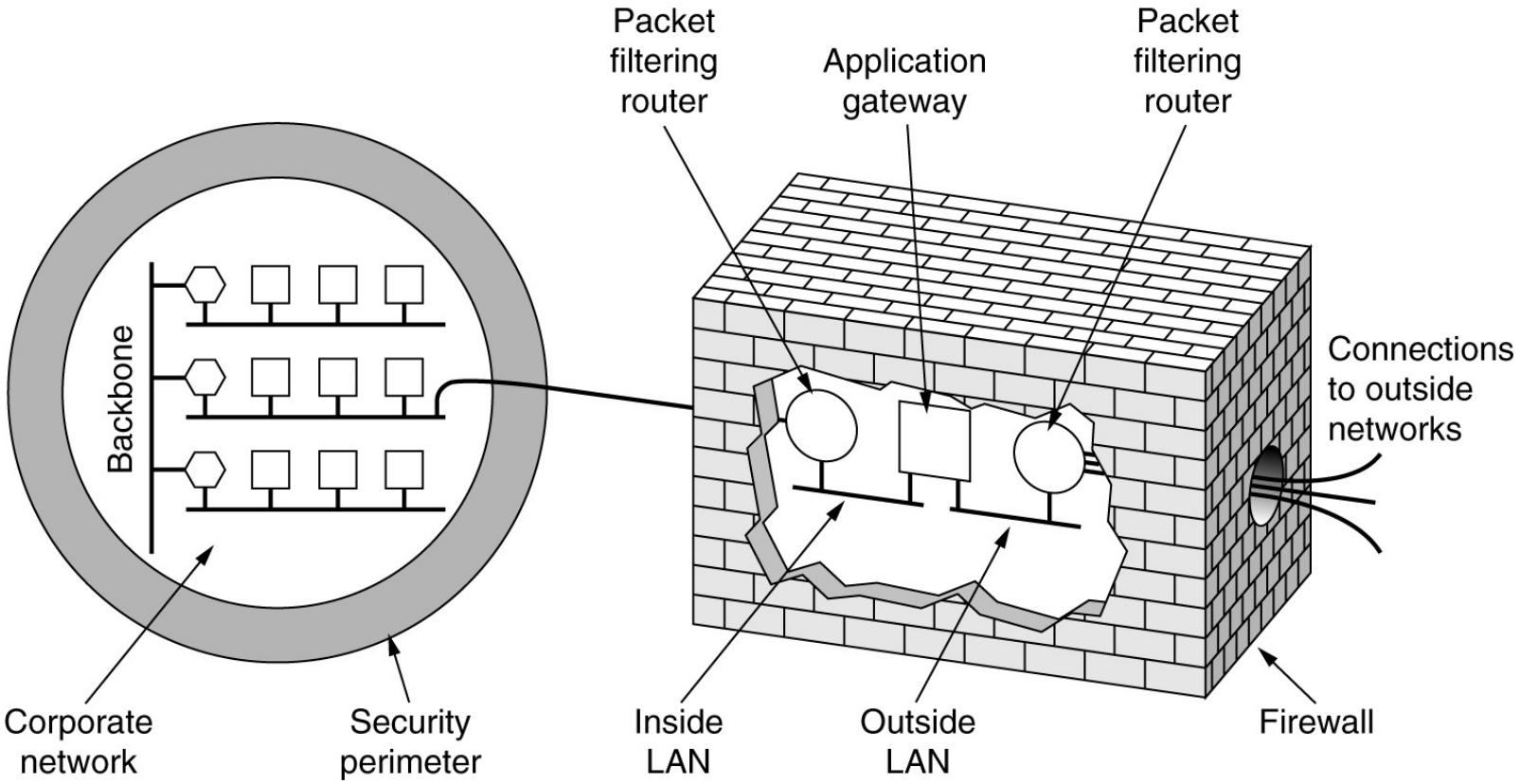


Caracteristici Protocol IPSEC

- IPSec este orientat pe conexiune (desi apartine nivelului retea)
- Permite selectia intre mai multi algoritmi
  - criptare: DES in mod CBC, 3DES, IDEA, …
  - autentificare: MD5, SHA (trunchiat la 96 biti)
  - “deschis” la adaugare algoritmi noi
- Permite stabilirea cheilor de criptare
- Permite alegerea intre mai multe servicii
  - confidentialitate
  - integritate
  - protectie la atacuri prin replica
- Permite alegerea granularitatii
  - conexiune TCP
  - toate legaturile intre doua calculatoare (tunel)
  - toate legaturile intre doua rutere, …

Firewall-uri

- Firewall: are posibilitatea de a bloca sau de a modifica traficul la nivelul retea



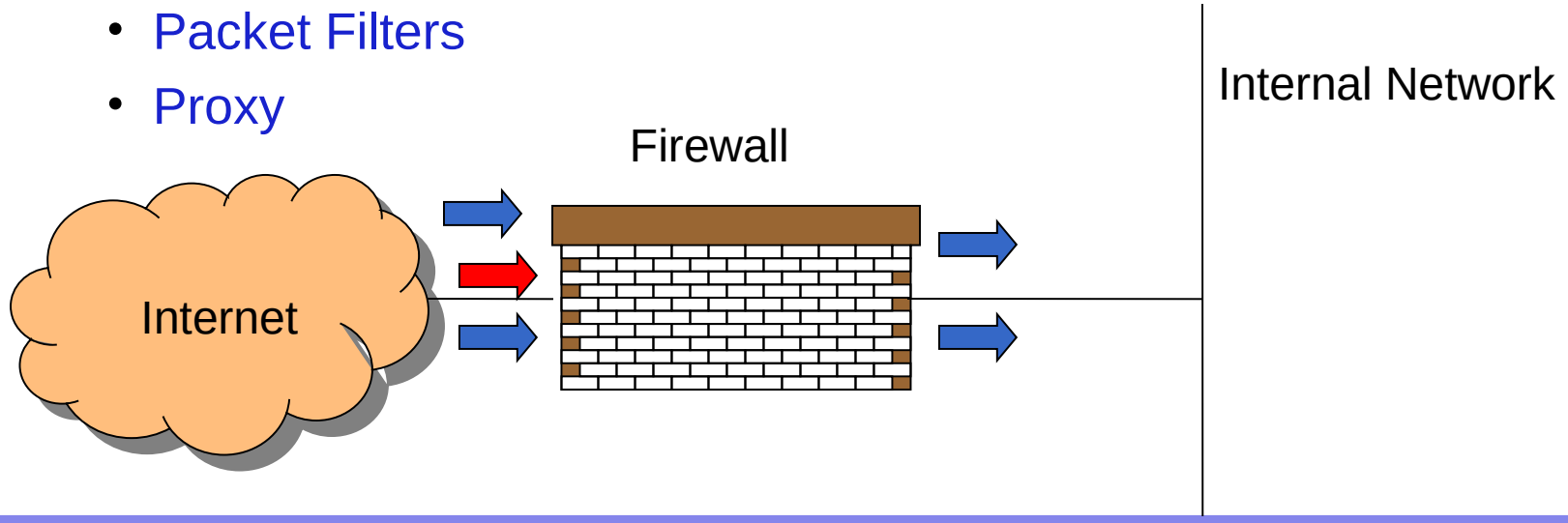
Firewall (1)

Functionalitate firewall:

- Inspecteaza traficul ce trece prin el
- Permite doar trecerea traficului de retea ce corespunde anumitor reguli
- Orice altceva este blocat

2 tipuri de firewall:

- Packet Filters
- Proxy



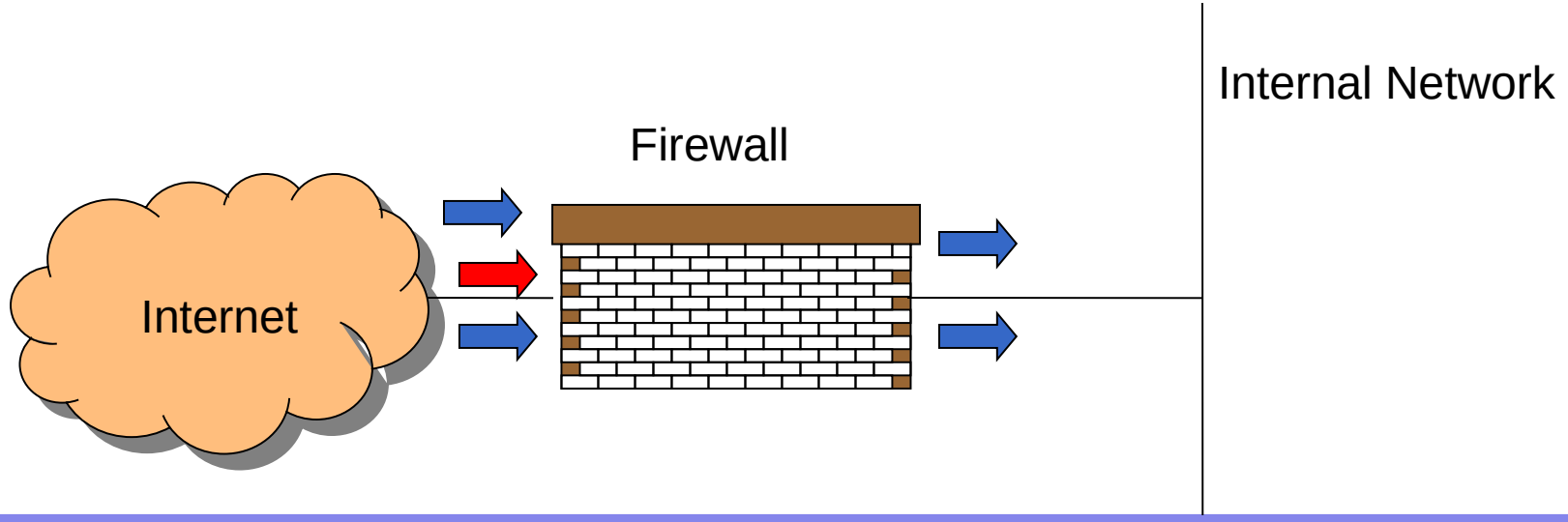
Firewalls (2)

- Foloseşte reguli la nivel reţea.
- Filtrare după:
  - Adrese origine şi destinaţie.
  - Protocol (TCP sau UDP).
  - Numere port.
- Nepotrivit pentru medii care cer analiza mai detaliată pentru protocoale de nivel superior (adică proxy servers).

Firewall (3)

Actiuni posibile:

- Permite trecerea pachetului
- Drop
- Alterarea pachetului (NAT?)
- Logging



Exemple reguli

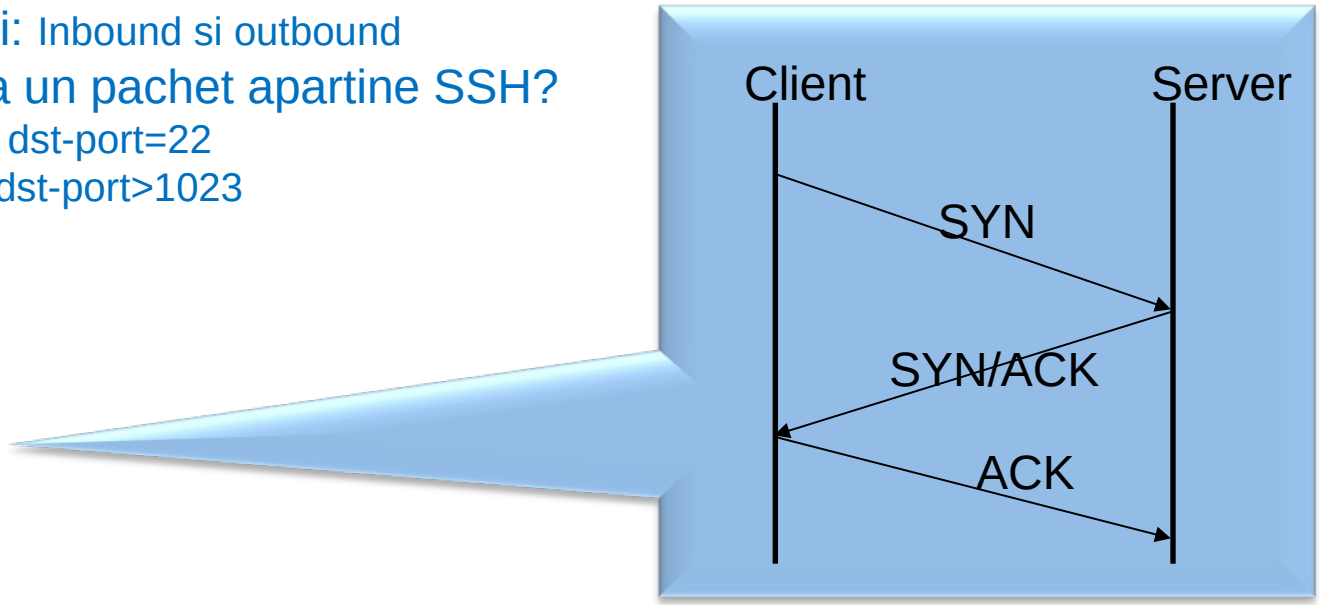
FROM <target a> TO <target b> <action> <protocol> <port>

- Blocheaza toate pachetele din exterior, mai putin pentru anumite protocoale
- Blocheaza tot traficul dintr-o lista de domenii
- Blocheaza tot traficul spre anumite domenii

- Ingress filtering
  - Toate pachetele din exterior spre interiorul retelei vor fi aruncate
- Egress filtering
  - Toate pachetele din reteaua locala spre adrese din exterior vor fi aruncate

Exemplu reguli firewall

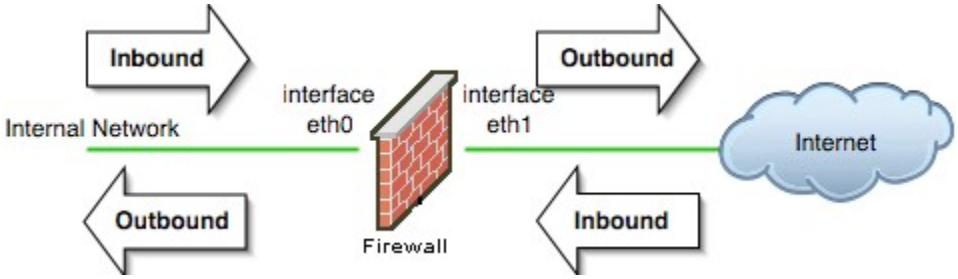
- Permite conexiuni SSH de la host-uri externe spre host-uri interne
- Este nevoie de 2 reguli: Inbound si outbound
- Cum se poate stii daca un pachet apartine SSH?
  - Inbound: src-port>1023, dst-port=22
  - Outbound: src-port=22, dst-port>1023
  - Protocol=TCP
- Ack bit Set?



Rule	Dir	Src Addr	Src Port	Dst Addr	Dst Port	Proto	Ack Set?	Action
SSH-1	In	Ext	> 1023	Int	22	TCP	Any	Allow
SSH-2	Out	Int	22	Ext	> 1023	TCP	Yes	Alow

Firewall rule example (default)

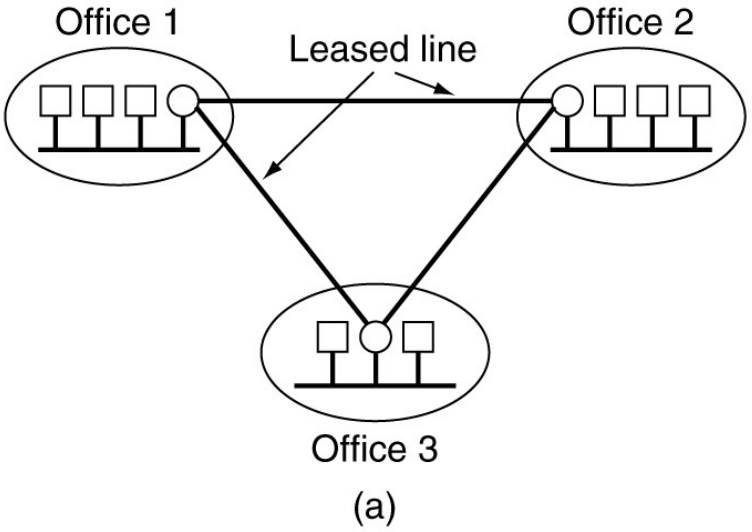
- Egress Filtering
  - Outbound traffic from external address Drop
- Ingress Filtering
  - Inbound Traffic from internal address Drop
- Default Deny

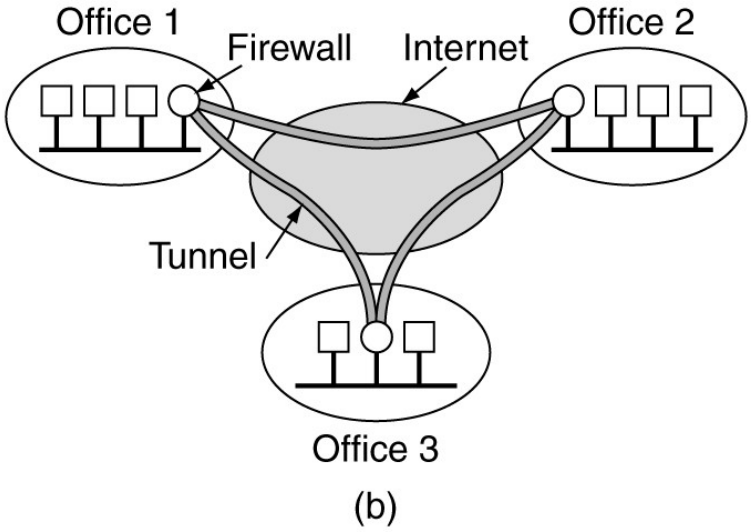


Rule	Dir	Src Addr	Src Port	Dst Addr	Dst Port	Proto	Ack Set?	Action
Egress	Out	Ext	Any	Ext	Any	Any	Any	Deny
Ingress	In	Int	Any	Int	Any	Any	Any	Deny
Default	Any	Any	Any	Any	Any	Any	Any	Deny

Virtual Private Networks

- Reţea privată:
  - (a) Cu linii închiriate.
  - (b) VPN.
- VPN se construiesc peste Internet:
  - Fiecare birou are un firewall, se creează tunele între firewalls.
  - IPsec folosit pentru tunneling (ESP în mod tunel).
  - În Internet pachetele apar ca şi cele obişnuite.
  - VPN este transparent pentru software de aplicaţii.





Protocoale de Autentificare

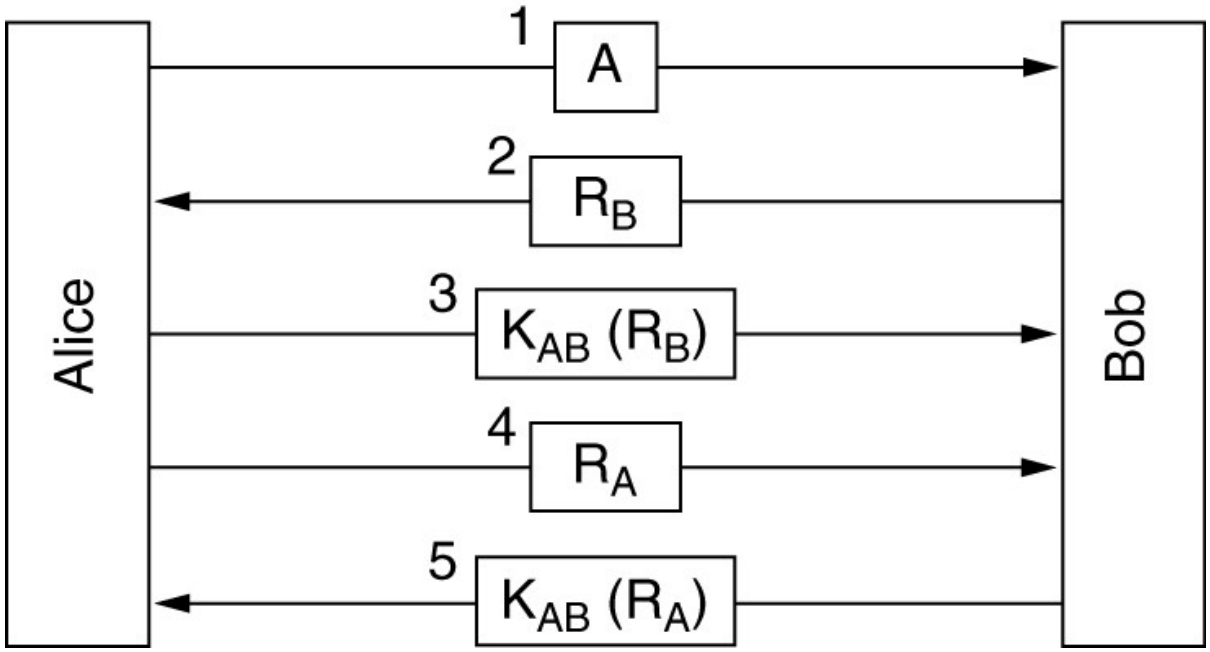
Determina daca o entitate (utilizator, proces) este cu adevarat cine / ce pretinde ca este

- diferita de autorizare
- se bazeaza pe un schimb de mesaje prin Internet (prezentate, de regula, ca schimb intre Alice si Bob)
- mesajele pot fi interceptate si folosite de alte entitati (de regula Trudy)
- protocolul genereaza si o cheie de sesiune

Folosesc criptografia cu

- Chei secrete partajate
- Chei publice

Autentificare cu cheie secreta partajata

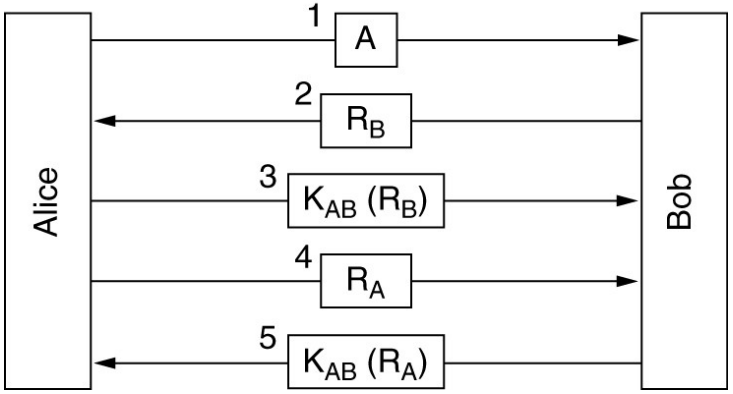


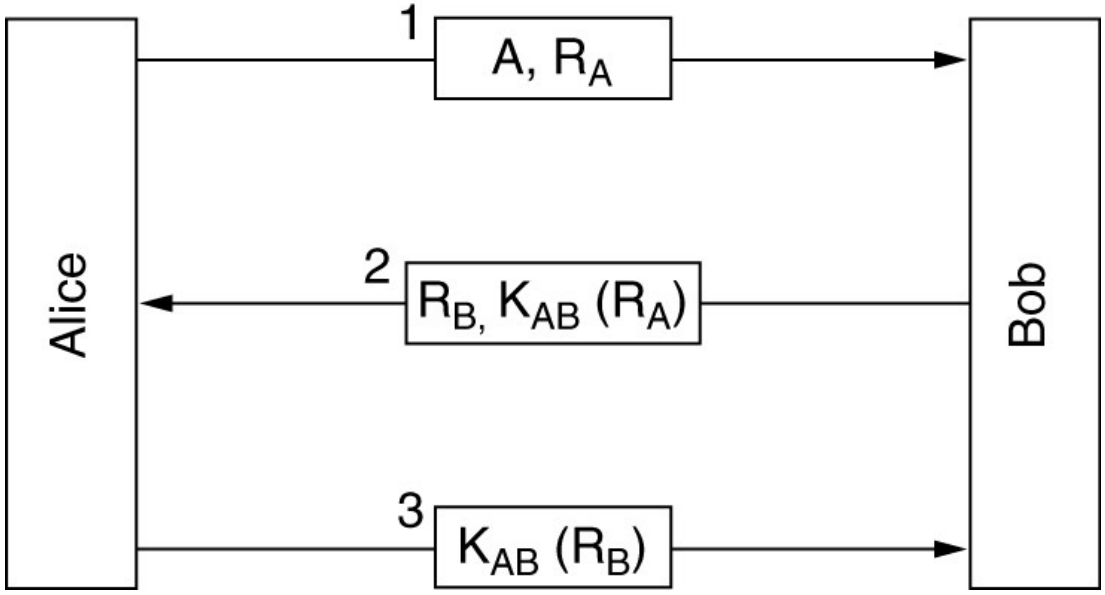
Autentificare reciproca cu un protocol challenge-response

Alice si Bob partajeaza cheia $\mathbf{K_{AB}}$

$\mathbf{R_A}$, $\mathbf{R_B}$ - numere aleatoare foarte mari, folosite contra atac prin replica

Autentificare cu cheie secreta partajata (2)





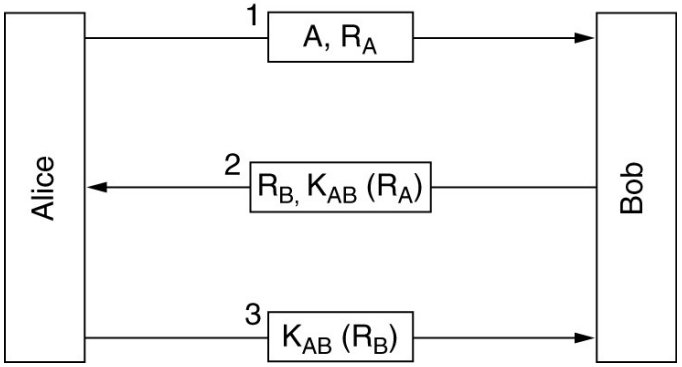
Reducere numar de pasi

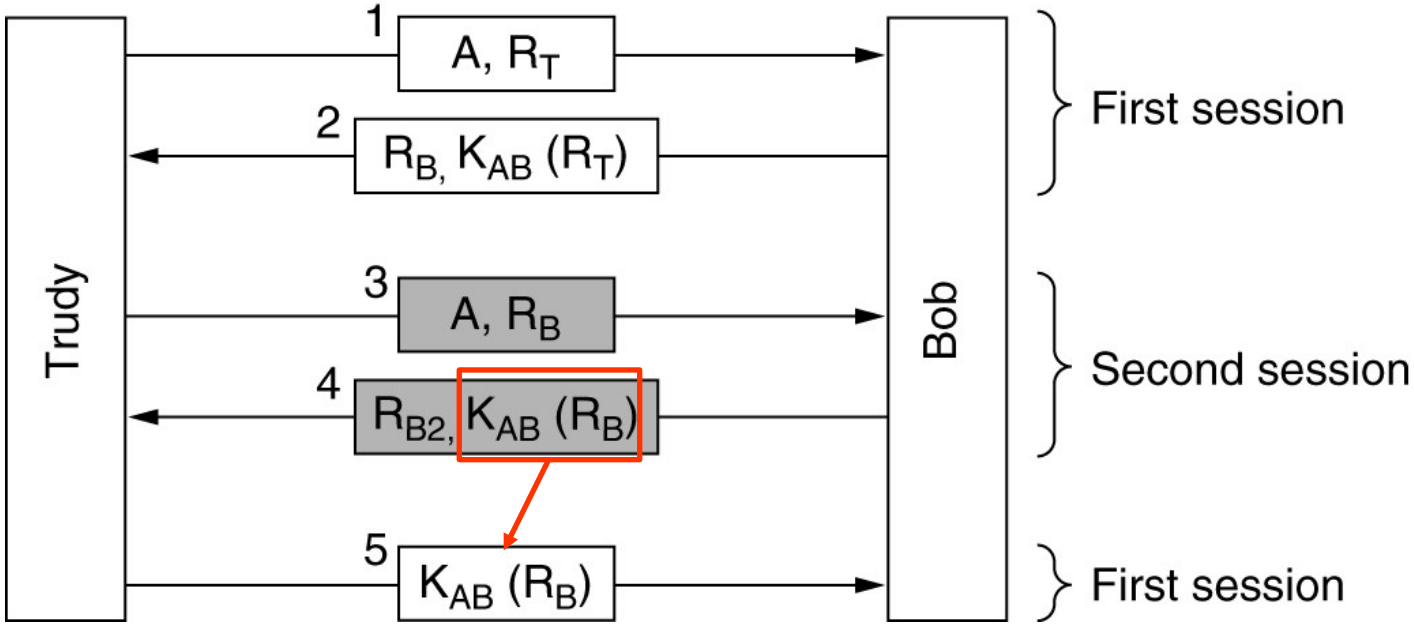
Atacul prin reflexie

Trudy nu poate cripta R_B din mesajul 2

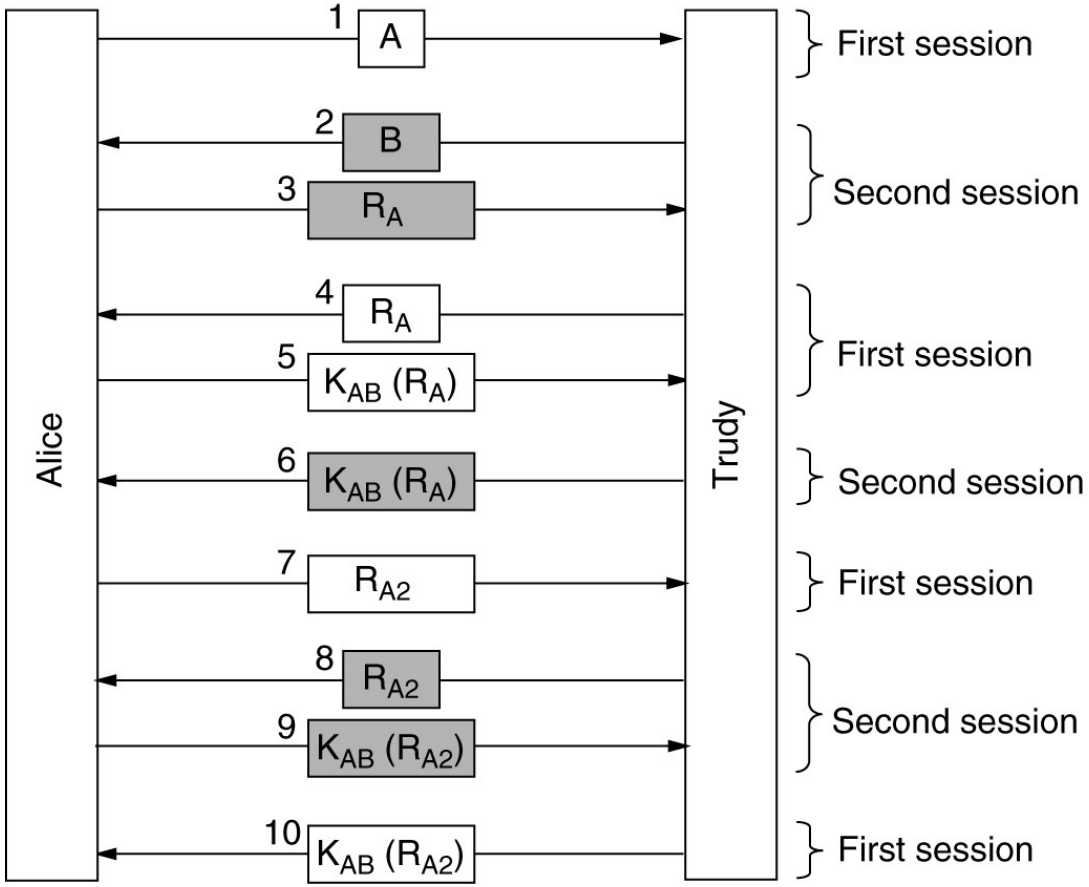
Dar retransmite un mesaj produs de Bob (4) si

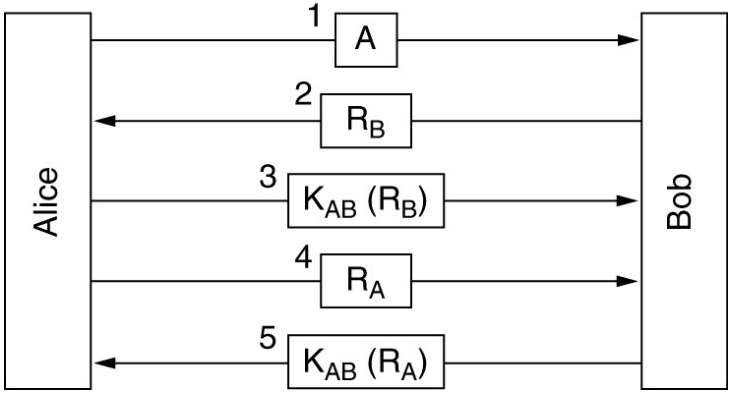
reuseste sa stabileasca o sesiune autentificata cu Bob





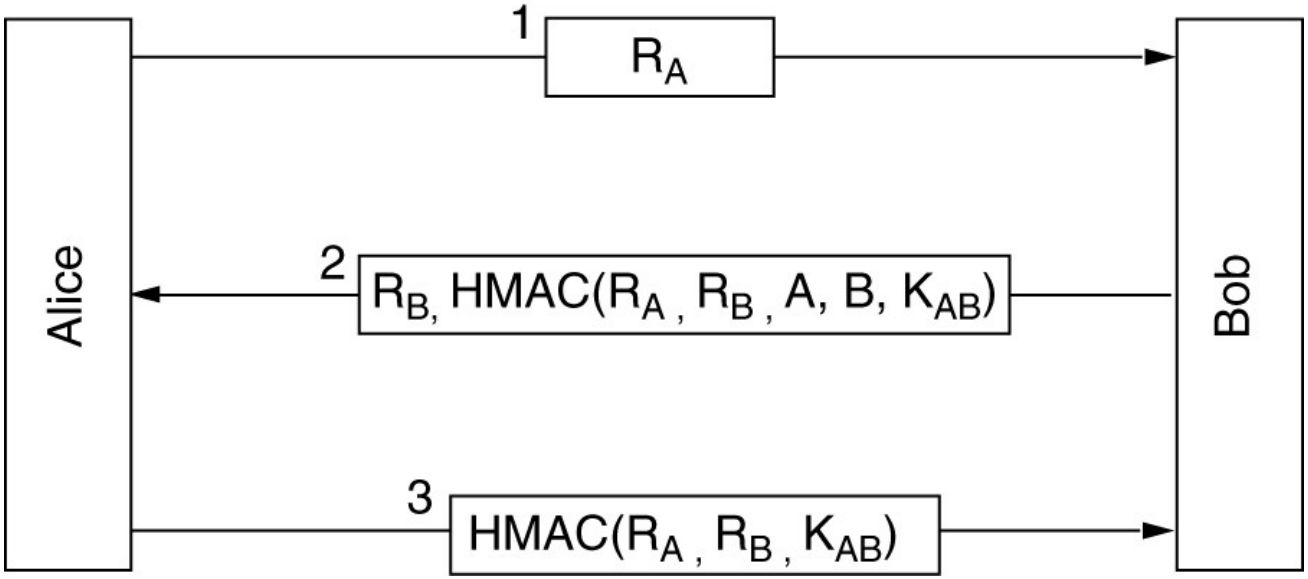
Atacul prin reflexie pe protocolul initial





Refolosind mesajele 5 si 9, Trudy reuseste sa stabileasca doua sesiuni autentificate cu Alice

Autentificarea cu HMAC



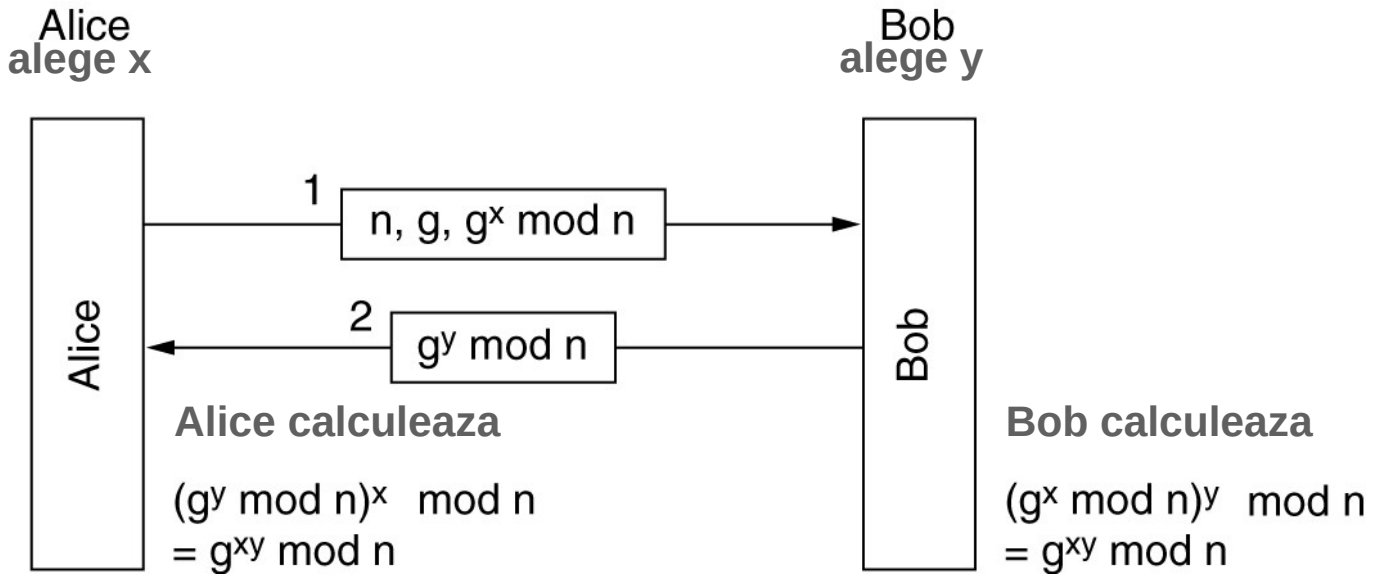
Alice si Bob partajează cheia K_{AB}

Fiecare parte poate calcula rezumatul HMAC - Hashed Message Authentication Code – (deoarece contine doar valori cunoscute)

- Hash-based Message Authentication Code (de ex. folosind SHA-1)

Trudy nu poate forța pe Alice sau Bob să cripteze sau să rezume o valoare impusă de ea

Stabilire cheie partajata: Diffie-Hellman key exchange



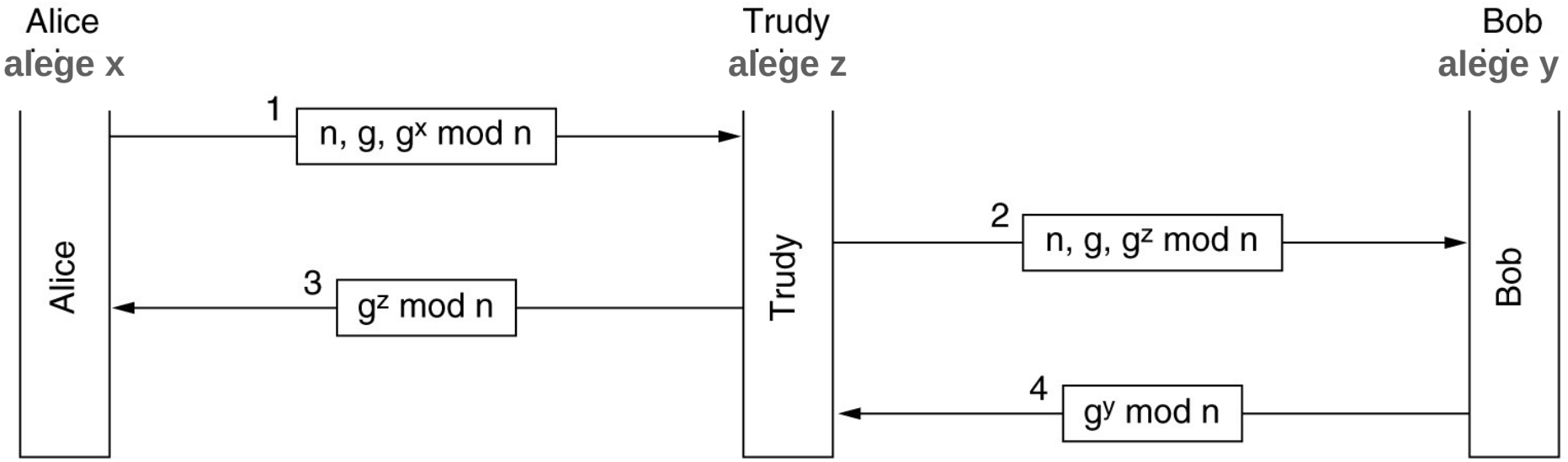
n, g – numere mari
n prim
(n-1)/2 prim

x nu poate fi calculat din $g^x \bmod n$
$g^{xy} \bmod n$ nu poate fi calculat din $g^x \bmod n$ si $g^y \bmod n$ cand n este mare

g < n (generator) are proprietatea: orice p poate fi scris ca $g^k \bmod n$

adica: pentru fiecare *p* intre 1 si *n*-1 inclusiv, exista o putere *k* a lui *g* astfel ca $p = g^k \bmod n$.

Atacul man-in-the-middle

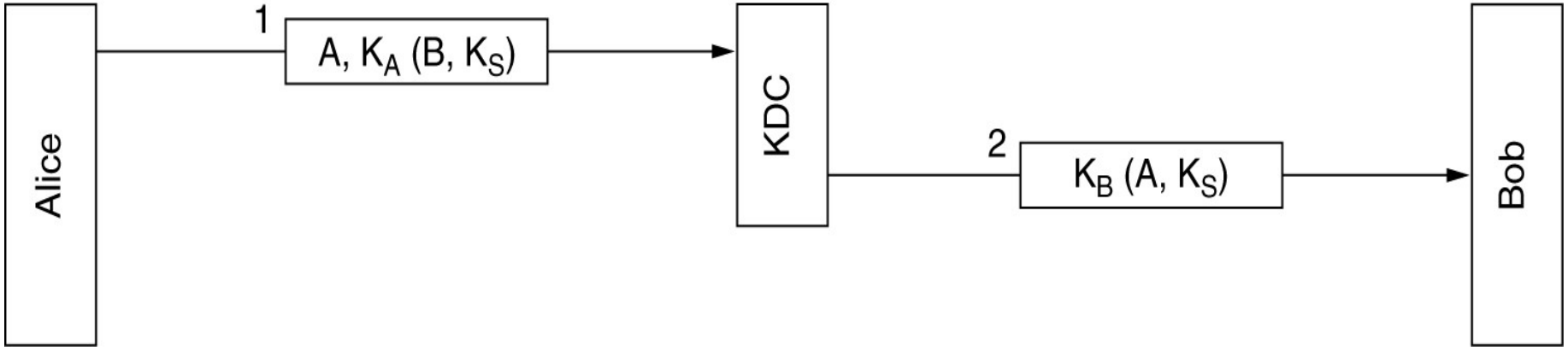


Vulnerabilitate – **g** si **n** sunt publici

- permite stabilirea a doua chei: intre Alice si Trudy - $\mathbf{g^{xz} \bmod n}$ si intre Trudy si Bob - $\mathbf{g^{zy} \bmod n}$

Rezolvare: Alice si Bob semneaza mesajele schimbate intre ei
Trudy nu poate modifica mesajele

Autentificarea folosind Key Distribution Center



Alice si Bob folosesc un Centru de distributie a cheilor

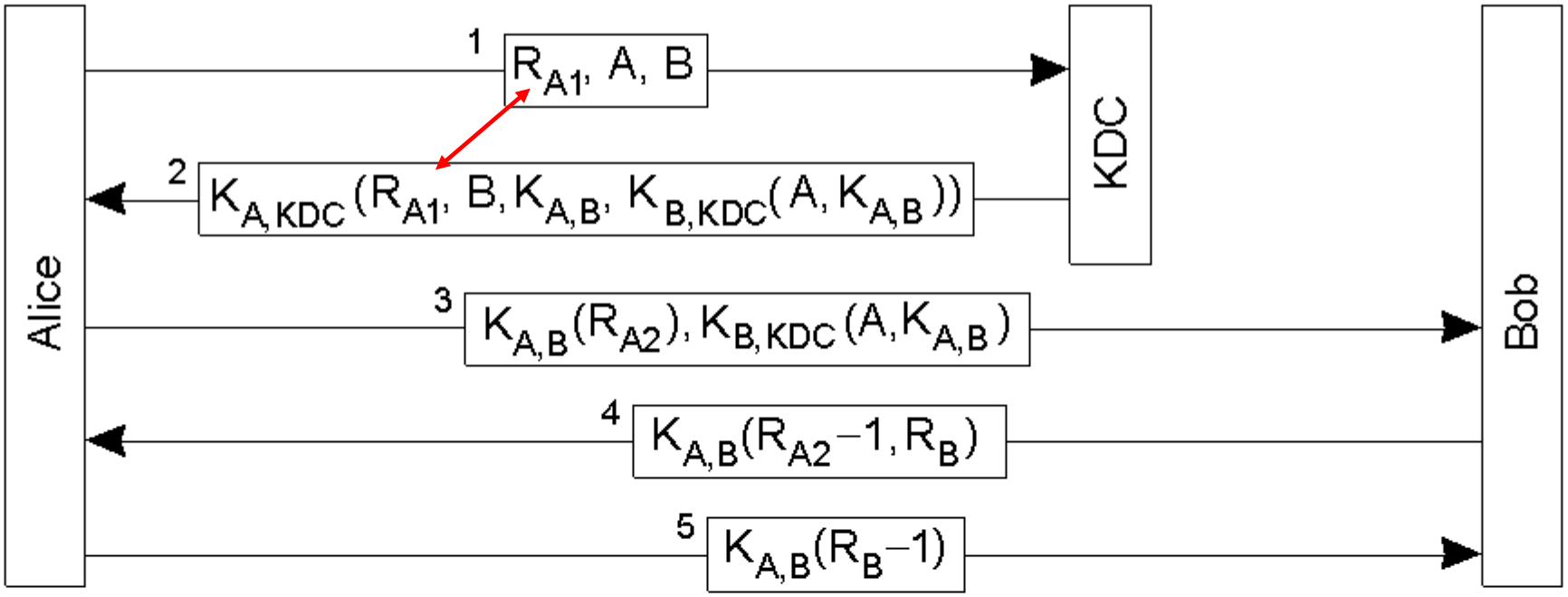
- in care au incredere
- cu care impart cheile secrete $\mathbf{K_A}$ respectiv $\mathbf{K_B}$

Prima incercare, vulnerabila la replay attack

Trudy **retransmite** mesajul 2 si

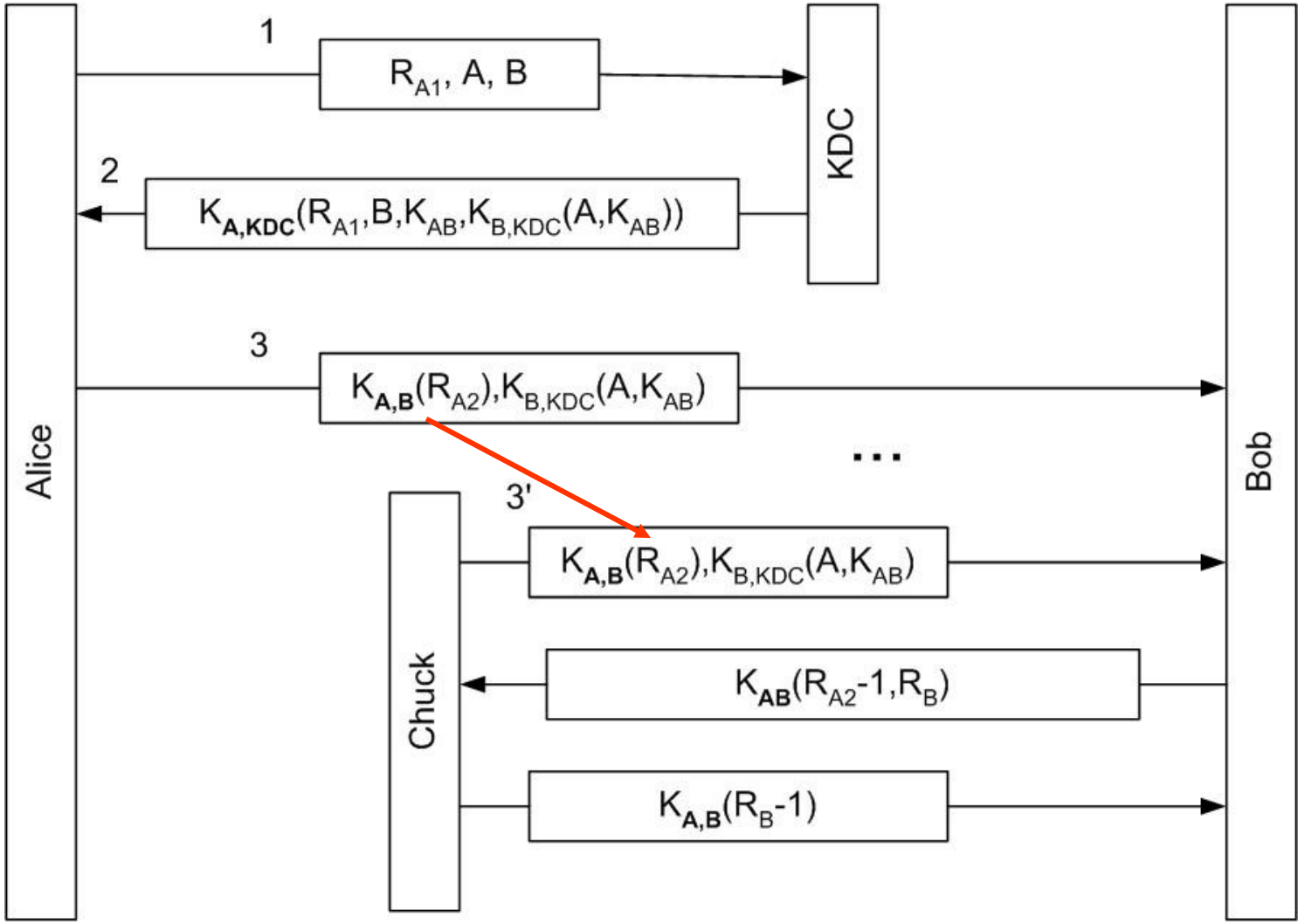
mesajul asociat cu el, criptat deja cu $\mathbf{K_S}$ (de ex. extragerea din contul lui Alice a unei sume de bani)

Autentificarea cu protocolul Needham-Schroeder



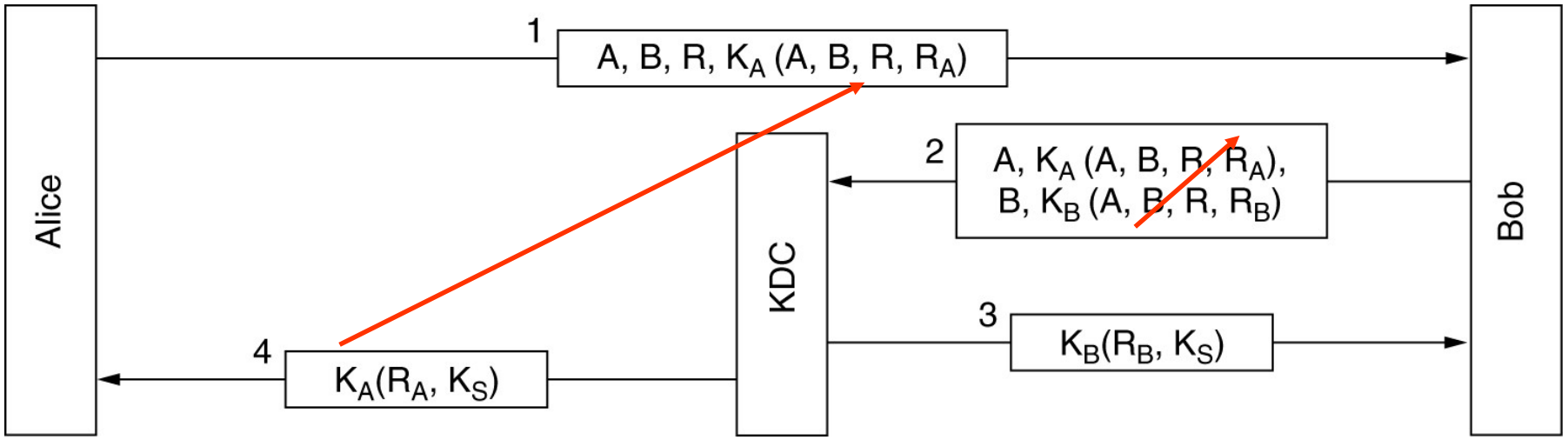
- foloseste tichete - ex. **$K_{B,KDC}(A, K_{AB})$**
  - Alice nu poate intelege sau modifica tichetul, Bob poate
  - Bob capata incredere in cheia **K_{AB}** (care vine de la KDC)
- numere aleatoare (nonce) ex. **R_{A1}**, folosite contra atac prin replica
  - ex. Alice afla ca mesajul 2 este un raspuns la 1, nu un mesaj rejucat de Trudy

Slabiciune Needham-Schroeder



Chuck afla cheia K_{AB} si retrimite mesajul 3, pretinzând ca e Alice

Autentificarea folosind Protocolul Otway-Rees



Protocolul Otway-Rees (simplificat).

KDC trimite cheia de sesiune K_S dupa ce verifica daca identificatorul comun R apare in ambele parti criptate ale mesajului 2

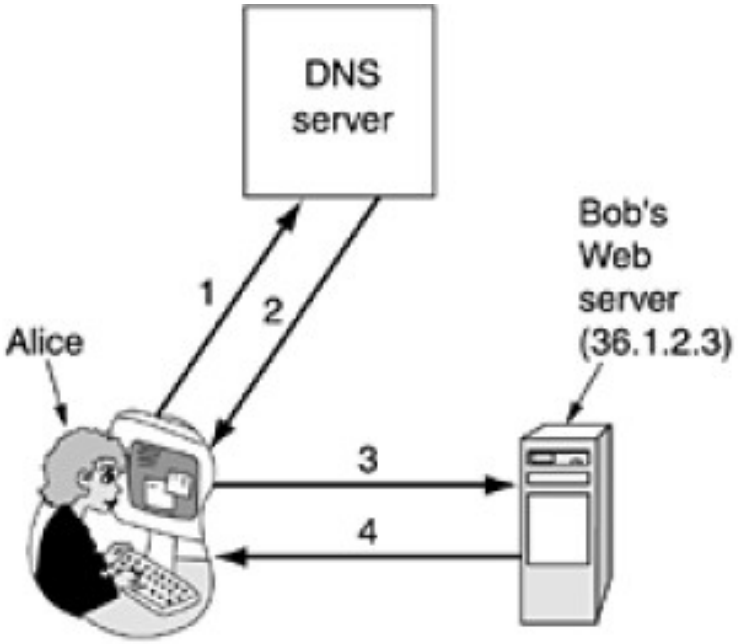
R_A, R_B – numere aleatoare folosite de KDC in mesajele 3 si 4 pentru a face legatura cu mesajele 1 si 2

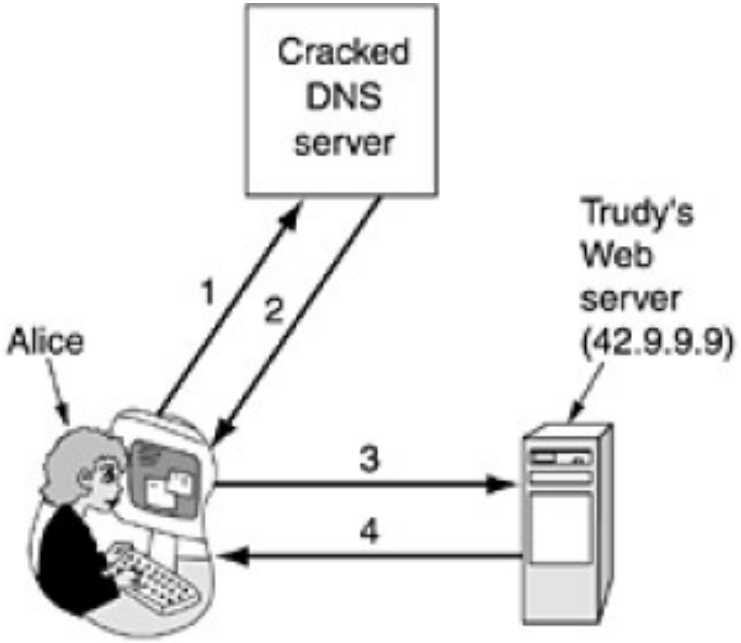
Problema: Alice ar putea folosi cheia secreta inainte ca Bob sa afle de ea

Securitatea Web

- Atacuri
  - inlocuire Home page
  - Denial-of-service
  - Citire mail-uri
  - Furt numere credit card
- Solutii
  - Secure Naming
  - SSL – The Secure Sockets Layer

Necesitate Secure Naming



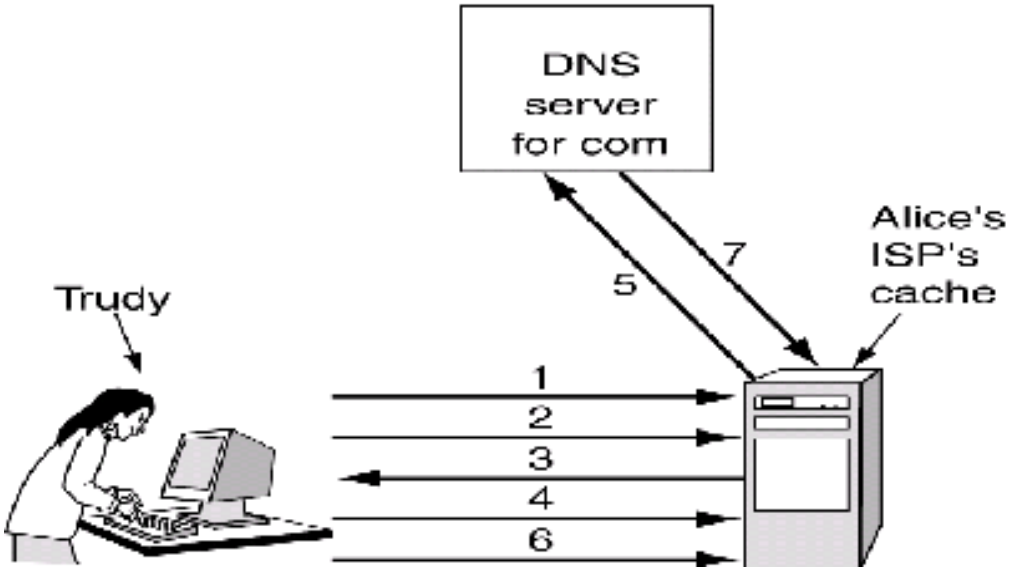


Situatie Normala.

1. Da-mi adresa IP Bob
2. 36.1.2.3 (adr IP Bob)
3. GET index.html
4. Pagina home Bob

Un atac bazat pe modificarea inregistrarii lui Bob in DNS.

1. Da-mi adresa IP Bob
2. 42.9.9.9 (adr IP Trudy)
3. GET index.html
4. Pagina Bob falsificata de Trudy



Trudy pacaleste ISP-ul lui Alice

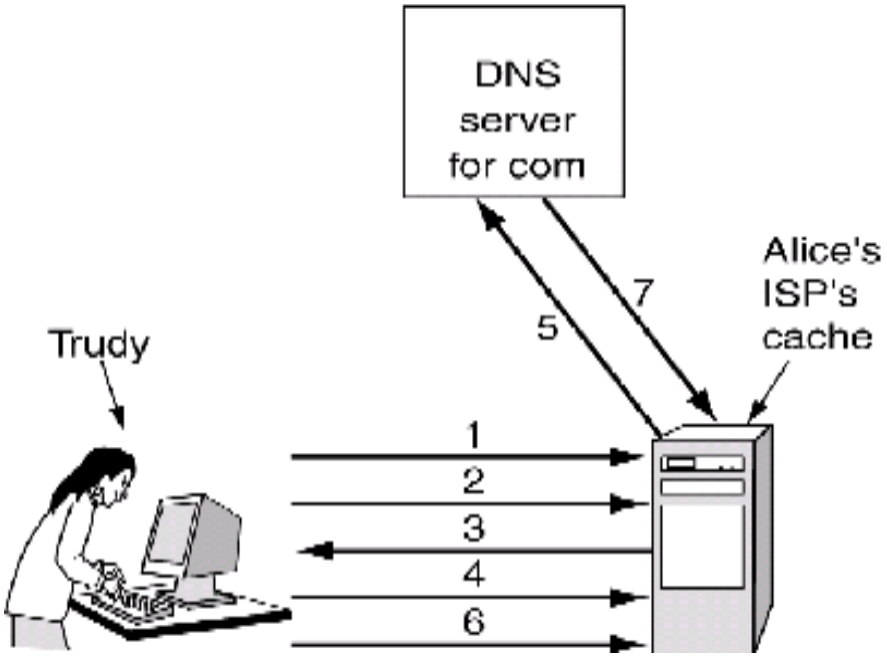
Probleme: ISP verifică adresa IP de la care vin răspunsurile DNS

- Trudy poate folosi adresa IP a unui server DNS de nivel inalt (care este publica) pentru a construi un raspuns fals

DNS se bazeaza pe UDP DNS foloseste

sequence numbers pentru a mapa cererile si raspunsurile

- Trudy inregistreaza un domeniu ***trudy-the-intruder.com*** (IP 42.9.9.9) si
- Creaza un server DNS ***dns.trudy-the-intruder.com*** (aceeasi IP 42.9.9.9)

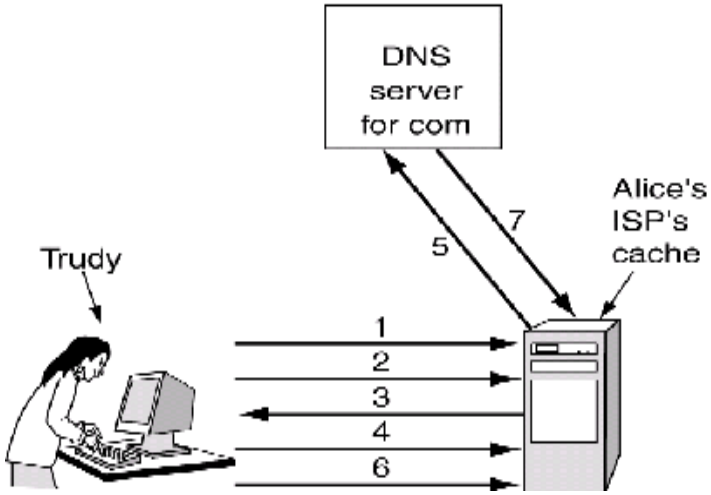


Trudy pacaleste ISP-ul lui Alice (2)

1. Cere adresa ***foobar.trudy-the-intruder.com*** **-** ISP-ul lui Alice afla de la serverul **com** despre noul ***dns.trudy-the-intruder.com*** si il pune in cache

2. Cere ISP-ului adresa pentru ***www.trudy-the-intruder.com***

3. ISP intreaba DNS-ul lui Trudy; intrebarea are un numar de secventa, **n** asteptat de Trudy



Trudy pacaleste ISP-ul lui Alice (3)

4. Repede, cere adresa **bob.com** (fortand ISP sa intrebe serverul **com** in pasul 5)

5. ISP transmite cererea cu nr secv n+1 catre serverul **com**

6. Trudy transmite repede un **raspuns fals** cu nr secv = **n+1**, adresa IP a serverului **com** drept sursa raspunsului si adresa sa 42.9.9.9 drept adresa lui Bob; raspunsul este considerat bun si este pus in cache-ul ISP

7. Cand soseste raspunsul adevarat, ISP il rejecteaza

Cand Alice cauta **bob.com** primeste **adresa falsa** din cache ISP

Secure DNS

Inregistrarile din DNS au forma

Domain name	Time to live	Class	Type	Value
bob.com.	86400	IN	A	36.1.2.3

Pentru securitate

fiecarei zone DNS i se aloca o **pereche de chei** publica/privata

Se adauga doua noi tipuri de inregistrari

***KEY* record** – cheia publica a unei zone, utilizator, host, etc.

***SIG* record** - **hash** semnat al inregistrarilor *A* si *KEY* pentru verificarea autenticitatii.

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...

Secure DNS (2)

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...

Gruparea obtinuta se numeste RRSet (Resource Record Set)

Clientii primesc de la DNS un RRS cu semnatura SIG

- aplica cheia publica a zonei pentru a decripta SIG
- calculeaza hash-ul pentru A si KEY
- compara cele doua valori (calculata si decriptata)

Analiza algoritmilor criptografici (Optional)

Problema:

Pot criptogramele interceptate de un intrus sa faciliteze spargerea confidentialitatii criptografice ?

O solutie este **Teoria Informatiei**

- care masoara cantitatea medie de informatie transmisa de o sursa
- si, echivalent, cantitatea de incertitudine inlaturata (in medie) de un mesaj

Bazata pe notiunile de entropie si entropie conditionata

Entropia

Entropia este cantitatea medie de informatie transmisa de o sursa

Fie S o sursa de informatii

- care transmite mesajele $X_1,..., X_n$

cu probabilitatile $p(X_1), ..., p(X_n)$, ptr. care $\sum_{i=1,n} p(X_i) = 1$.

Entropia H(X) este:

$$\mathbf{H(X) = \sum_{i=1,n} p(X_i)*log\ (1/p(X_i)) = -\sum_{i=1,n} p(X_i)*log\ (p(X_i))}$$

- $log(1/p(X_i))$ = cantitatea de informatie primita la receptia lui X_i.
- baza logaritm = 2 -> cantitatea masurata in numar de biti

Exemplu

- aruncarea monedei – cap sau pajura
- probabilitati egale, 1/2

informatie 1 bit: $H(X) = \sum_{i=1,2} (1/2)*log\ (1/(1/2)) = 1$

Incertitudinea

Entropia = cantitatea de incertitudine inlaturata (in medie) de un mesaj

Exemplu:

o sursa poate trimite n = 4 mesaje, cu probabilitati egale p(X) = 1/4

$$H(X) = \Sigma_{i=1,4} (1/4)*\log (4) = 2$$

fiecare mesaj inlatura o incertitudine de 2 biti
(inainte nu se stia care din cele 4 mesaje va fi primit)
Entropia depinde de distributia probabilitatior mesajelor

- H(X) maxim când $p(X_1) = p(X_2) = ... = p(X_n) = 1/n$.

$$H(X) = \Sigma_{i=1,n} (1/n)*\log (n) = \log (n)$$

- H(X) descreşte când distribuţia mesajelor se restrânge.
- H(X) = 0 când $p(X_i) = 1$ pentru un mesaj i.

Entropie conditionata - Echivocitatea

Exemplu:

- mesajele X sunt conditionate de mesaje Y

Fie m=4 şi p(Y) = 1/4 pentru fiecare Y.
Presupunem ca fiecare mesaj Y restrânge X astfel:
dupa Y1: urmeaza X1 sau X2, fiecare cu prob. 1/2
dupa Y2: urmeaza X3 sau X4,
dupa Y3: urmeaza X2 sau X3,
dupa Y4: urmeaza X4 sau X1.
Problema:

- cum se calculeaza entropia lui X ?

Entropie conditionata – Echivocitatea (2)

Dat fiind Y din mulţimea mesajelor $Y_1,..., Y_n$ cu $\Sigma_{i=1,n}\ p(Y_i) = 1$,

- fie: $p_Y(X)$ probabilitatea mesajului X condiţionat de Y.

  p(X,Y) probabilitatea mesajelor X şi Y luate împreună:

  $$p(X,Y) = p_Y(X)*p(Y).$$

  - Echivocitatea este entropia lui X condiţionat de Y:

    $$H_Y(X) = \Sigma_{X,Y}\ p(X,Y)*\log\ (1/p_Y(X))$$
    $$H_Y(X) = \Sigma_{X,Y}\ p_Y(X)*p(Y)\ \log\ (1/p_Y(X))$$
    $$= \Sigma_Y\ \mathbf{p(Y)}\ \Sigma_X\ \mathbf{p_Y(X)*\log\ (1/p_Y(X))}.$$

Exemplu:

- n = 4 şi p(X) = 1/4 pentru fiecare X => H(X) = log 4 = 2.
- Fie m=4 şi p(Y) = 1/4 pentru fiecare Y.
- Presupunem că fiecare Y restrânge X:
- Y1: X1 sau X2, Y2: X3 sau X4, Y3: X2 sau X3, Y4: X4 sau X1.
- Echivocitatea este: $H_Y(X) = 4 ((1/4) 2 (1/2) \log 2) = \log 2 = 1$.

Adică: cunoaşterea lui Y reduce incertitudinea lui X la un bit.

Confidenţialitatea perfectă

Cunoasterea unor criptograme nu reduce confidentialitatea

Fie:

- M - multime texte clare cu probabilitatea p(M), Σ_M p(M) = 1.
- C - criptograme, cu probabilitatea p(C), Σ_C p(C) = 1.
- K - chei cu probabilitatea p(K), Σ_K p(K) = 1.
- $p_C(M)$ probabiltiatea să se fi transmis M când se recepţionează C.

Confidenţialitatea perfectă <=> $p_C(M) = p(M)$.

Fie $p_M(C)$ probabilitatea să se recepţioneze C când s-a transmis M:

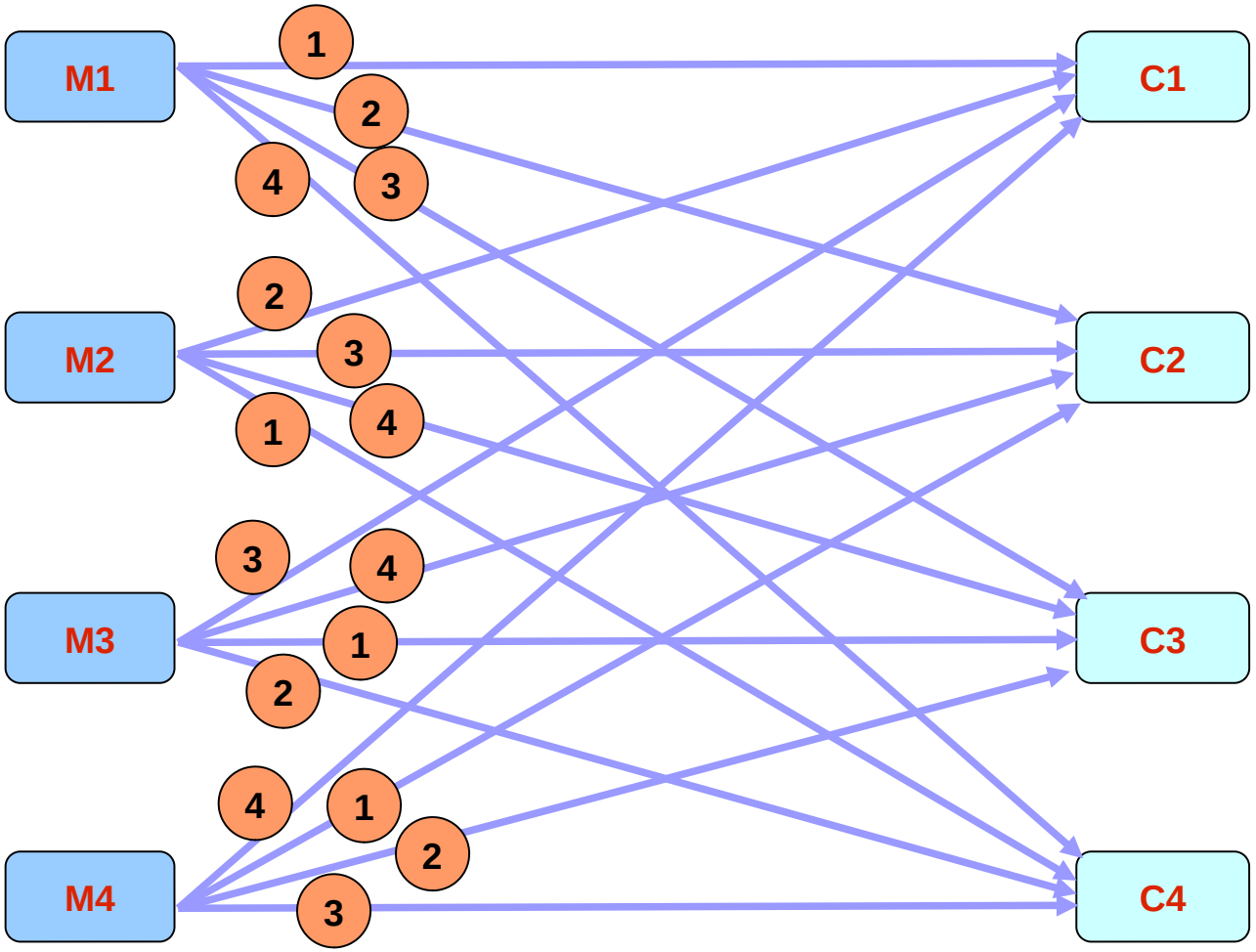
$p_M(C) = \Sigma_{k,\ Ek(M)=C}\ p(k)$.

- Confidenţialitatea perfectă:

$p_M(C) = p(C)$, pentru toate M şi orice C.

Confidenţialitatea perfectă

- Confidenţialitatea perfectă este posibilă dacă se folosesc chei la fel de lungi ca mesajele codificate.



Distanţa de unicitate

- "Spargerea" confidenţialităţii depinde de cantitatea de criptograme de care intrusul dispune
  - cantitatea de incertitudine în K cunoscand C, este exprimată ca entropia (echivocitatea) cheii conditionata de criptograme:

$$\mathbf{H_C(K) = \Sigma_C\, p(C) \Sigma_K\, p_C(K) \log (1/p_C(K))}$$

- Dacă echivocitatea $H_C(K)=0$ nu există incertitudine şi cifrul se poate sparge.
- Când creşte lungimea N a textelor cifrate echivocitatea scade.
- Distanţa de unicitate:
  - **Cel mai mic N pentru care $H_C(K)$ este foarte apropiat de 0**.
- Cifru neconditionat sigur:
  - $H_C(K)$ nu se apropie niciodată de 0.

Redundanta limbajului (1)

Pentru un limbaj, consideram mulţimea mesajelor X de lungime N

- cu entropia H(X)
- fiecare mesaj este o secventa de N simboluri dintr-un alfabet A care are L simboluri

Nu toate combinatiile de N simboluri au sens

- ex. anumite succesiuni de consoane, digrame, trigrame, etc.
- Redundanţa limbajului, D este

$$D = R - r$$

  - R este rata absolută a limbajului
  - r este rata limbajului
- D = 3.2 ... 3.7 pentru limba engleză.

Rata limbajului

Rata limbajului este entropia pe simbol daca nu toate combinatiile de N simboluri au sens

este raportul entropia mesaj / numar simboluri din mesaj:

$$r = H(X) / N$$

- r = numar de biti pentru un simbol

cu care se pot reprezenta (un numar de) 2^r simboluri

- pentru limba engleză r = 1 ... 1.5 biti pe litera

- Numarul total al mesajelor de lungime N cu sens este 2^{rN}

Rata absoluta a limbajului

Rata absoluta a limbajului este entropia pe simbol daca toate combinatiile de N simboluri ar avea sens

Se considera ca cele L simboluri au aceeasi probabilitate = 1/L

Rezulta:

$$R = \Sigma_{i=1,L} (1/L) \log (L) = \log L$$

- pentru limba engleză R = log 26 = 4.7 biţi pe literă

- Numarul de simboluri L se poate rescrie **$L = 2^R$**
- Numarul de mesaje de lungime N (cu sau fara sens) este **2^{RN}**

Calcul aproximativ distanţă unicitate (1)

Distanta de unicitate N este:

$$N = H(K) / D$$

Justificare

- Ipoteze:
  - Sunt 2^{RN} mesaje posibile, din care 2^{rN} au sens.
  - Toate mesajele cu sens au aceeaşi probabilitate, $1/2^{rN}$.
  - Toate mesajele fără sens au probabilitate 0.
  - Sunt $2^{H(K)}$ chei cu probabilităţi egale.
  - Cifrul este aleator:
    - Pentru fiecare k şi C, descifrarea $D_K(C)$ este variabilă aleatoare independentă uniform distribuită pe toate mesajele, cu sau fără sens.

Calcul aproximativ distanţă unicitate (2)

Fie criptograma $C = E_K(M)$.

- Criptanalistul are de ales între $2^{H(K)}$ chei, doar una este corectă.
- Rămân $2^{H(K)}-1$ chei care pot da o soluţie falsă (adica C se obţine criptând un alt mesaj M' cu înţeles)

cu aceeaşi probabilitate (= mesaje cu sens / total mesaje)

$$q = 2^{rN} / 2^{RN} = 2^{(r-R)N} = 2^{-DN}$$

- Numărul de soluţii false F (= nr chei incorecte * probabilitatea unei chei de a da solutie falsa):

$F = (2^{H(K)} -1)q = (2^{H(K)} -1)\, 2^{-DN} \approx 2^{H(K)-DN}$

conditia de unicitate -> log F = H(K)-DN = 0

N = H(K) / D

Analiza cifrării prin substituţie

- Substituţie monoalfabetică:
  - N = H(K) / D = log n! / D sunt n! chei posibile
  - Pentru limba engleză:
    - N = log 26! / 3.2 = 27.6
- Substituţie periodică cu perioada d si s simboluri in alfabet:
  - Sunt s^d chei posibile pentru fiecare substiţutie simplă:
    - N = H(K) / D = log s^d / D = (d*log s) / D
  - Pentru cifrul Vigenere s = 26:
    - N = d * 4.7 / 3.2 = 1.5 d

Analiza cifrării prin transpoziţie

- Caracteristici:
  - Cifrul permută caracterele cu o perioadă fixă d.
  - Sunt d! permutări posibile.
  - Toate sunt echiprobabile.
- H(K) = log d!
  - N = H(K) / D = log d! / D
  - N = d log (d/e) / D
- Pentru d = 27 şi D = 3.2 rezultă:
  - N = 27.9