Gestiunea Securitatii

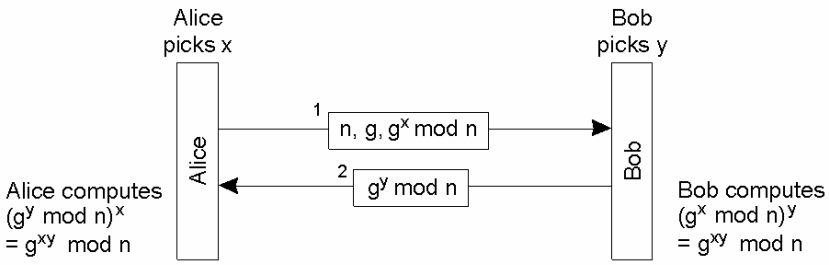
- Gestiunea cheilor
  - Stabilirea cheilor
  - Distributia cheilor
- Gestiunea sigura a grupurilor (Secure Group Management)
  - Adaugarea unui nou membru al grupului
- Gestiunea autorizarii
  - delegarea

Stabilirea cheilor

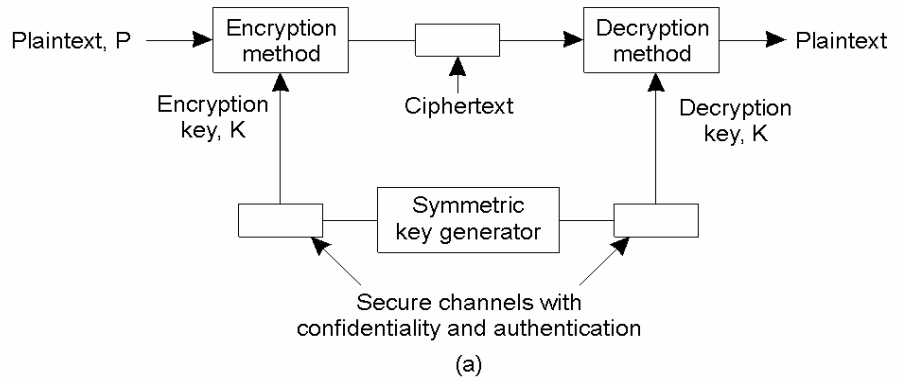
1. Chei secrete:
   - De catre o a treia parte, de incredere (KDC)
   - Fara a treia parte; ex Diffie-Hellman
2. Chei publice:
   - De catre o autoritate de certificare (CA = Certificate Authority)

Ex. 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



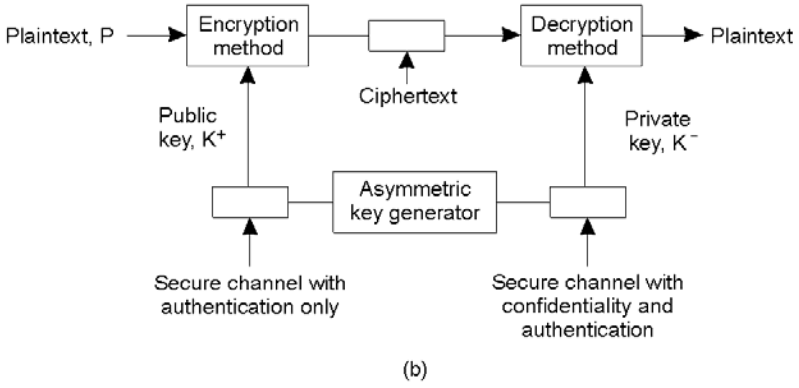
Distributia cheilor secrete



Foloseste cai autentificate si confidentiale

- Telefon, posta, valiza diplomatica, contact direct

Distributia cheilor publice

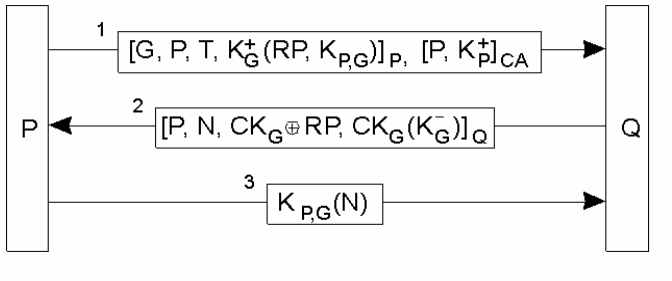


Cheia publica – prin canale autentificate

- Certificate cu cheie publica

Cheia privata – prin canale confidentiale si autentificate

Gestiunea sigura a grupurilor (1)



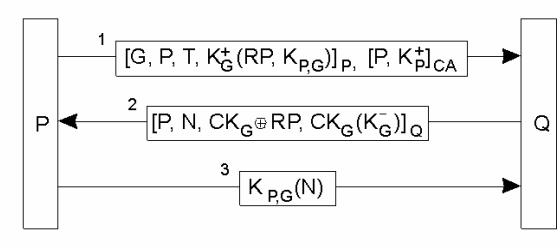
Grupul foloseste K^{+}_{G} / K^{-}_{G} pentru a comunica cu alte procese din afara

CK_{G} (secreta) in interiorul grupului

Problema: admiterea unui nou membru in grup: P trimite:

- o cerere de alaturare la grup, semnata de el
  - T – timpul curent local lui P
  - RP – Reply Pad – folosit la alcatuirea raspunsului
  - $K_{P,G}$ – cheie secreta partajata intre P si G
- si un certificat semnat de CA cu cheia sa publica K^{+}_{P}

Gestiunea sigura a grupurilor (2)



Q autentifica P si

- Verifica daca P poate fi admis ca membru
- Transmite mesajul semnat 2 de admitere in grup
  - N – Nonce
  - CK_{G} – cheia secreta a grupului G combinata cu RP
  - K^{-}_{G} – cheia privata a grupului criptata cu cheia secreta a grupului CK_{G}
    - Obs: in mesaj 2, CK_{G} este criptat cu RP, nu cu K^{+}_{P}; motiv - daca K^{-}_{P} este compromisa atunci CK_{G} poate fi compromisa.

P autentifica Q si

- Transmite inapoi N criptat cu cheia secreta partajata $K_{P,G}$

Managementul cheilor pentru comunicarea de grup

Neajuns metoda prezentata: P poate vedea mesaje anterioare admiterii sale in grup

Uzual, cheia de grup se schimba la modificarea componentei grupurilor:

- Un nou membru nu poate intelege mesajele anterioare intrarii sale in grup
- Un fost membru (care a iesit sau a fost exclus din grup) nu poate intelege mesajele transmise dupa ce a parasit grupul

Distribuirea unei noi chei de grup

- Triviala inainte de alaturarea unui nou membru
  - Noua cheie se transmite multicast criptata cu cheia veche
- Foarte complicata la iesirea unui membru – schema nescalabila
  - un KDC are o cheie secreta cu fiecare membru din grup
  - Alcatuieste un mesaj lung de O(n): fiecare inregistrare este noua cheie criptata cu cheia secreta partajata cu un membru din grup (face n criptari)
  - Transmite multicast mesajul (mesajul este mare)
  - Fiecare membru isi ia partea criptata cu cheia sa

A Survey of Key Management for Secure Group Communication
SANDRO RAFAELI AND DAVID HUTCHISON, ACM Computing Surveys, No. 3, September 2003

Protocoale pentru managementul cheilor de grup

Protocoale

- Centralizate
  - Un singur KDC
- Descentralizate
  - Mai multe KDC pentru subgrupuri
- Distribuite
  - Nu exista un KDC

Rol management chei

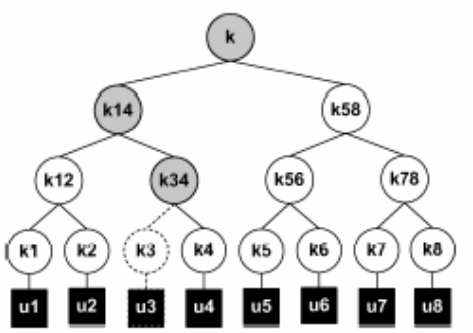
- Identificarea si autentificarea membrilor
- Controlul accesului (operatia de alaturare trebuie validata)
- Generarea, distributia si instalarea cheilor (noile chei independente de cele vechi)

Management centralizat – LKH (Logical Key Hierarchy)

Propus Wong, Wallner

KDC pastreaza un arbore de chei

- Fiecare nod tine o KEK (Key Encryption Key) cu care se cripteaza alte chei
- Radacina tine cheia de grup (cu care se cripteaza mesajele)
- Fiecare frunza este asociata cu un membru
- Fiecare membru are cheia KEK din frunza si cheile KEK din calea de la el la radacina (in total $\log_2 n + 1$ chei, n = adancime arbore balansat)
- Ex: u_1 cunoaste k_1, k_{12}, k_{14} si k



LKH (Logical Key Hierarchy) – nou membru

Membrul nou este asociat cu o frunza si inclus in arbore

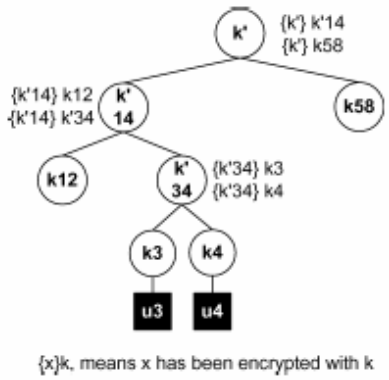
Toate nodurile de la parintele noului nod la radacina sunt compromise (fig pentru nodul u3) si trebuie schimbate

Se compune mesaj cu noile KEK; un KEK este criptat cu fiecare din cheile fiilor nodului respectiv (lungime mesaj $2\log_2 n$)

Ex: nodul u_3 primeste cheia K_3 si este adaugat la nodul k_{34}.

Nodurile compromise sunt k_{34}, k_{14} si k

- Se genereaza k'_{34}, k'_{14} si k'
- Se cripteaza fiecare cu cheile fiilor lor
- Se compune mesaj de lungime 2*3
- Se transmite multicast
- Fiecare membru calculeaza noul set de chei



LKH (Logical Key Hierarchy) – eliminare membru

Ex: eliminare u_4

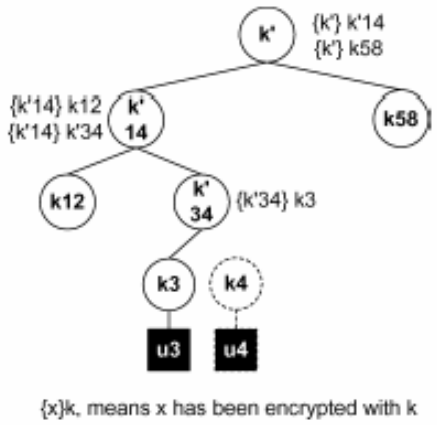
Noile chei sunt criptate cu KEK ale fiilor

Exceptie k'_{34} criptat doar cu cheia fiului ramas, k_3

Imbunatatire Waldvogel – protocol LKH+

Foloseste o functie one-way

Fiecare membru calculeaza noile chei aplicand functia one-way pentru cheile compromise



OFT - One-way Function Tree

Propus McGrew si Sherman

Schema reduce lungimea mesajelor la $\log_2 n$

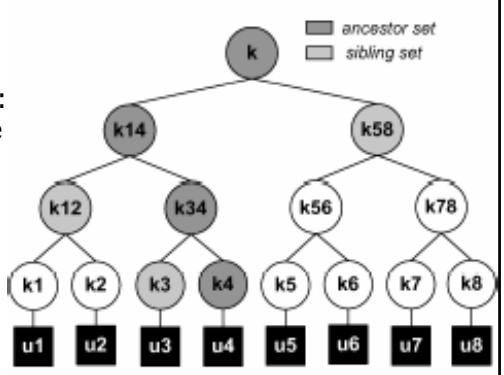
Cheia unui nod este generata prin "orbirea" cheilor fiilor si mixarea dupa formula:

$$K_i = f(g(k_{left(i)}, g(k_{right(i)})$$

- **left(i), right(i) – fiii lui i**
- **g – functie one-way de "orbire"**
- **f – functie de mixare**

Fiecare nod are asociate doua seturi:
- Stramosi (ancestors) – nodurile de la el la radacina
- Rude (siblings) – fratii nodurilor din setul Stramosi

Ex: seturile pentru u4



OFT (2)

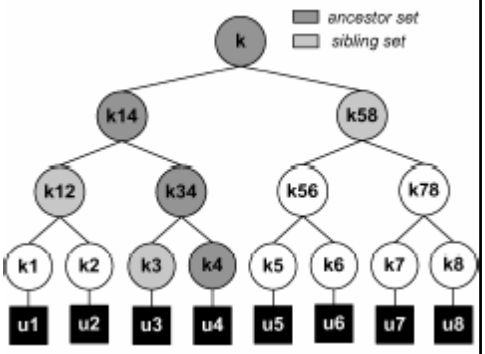
Fiecare membru pastreaza $\log_2 n+1$ chei

Ex: u_4 pastreaza 4 chei

- Cheia sa (k_4)
- Cheia orbita a fratelui $g(k_3)$
- Cheile orbite ale Rudelor $g(k_{12})$, $g(k_{58})$

Din aceste info, u_4 poate genera:

- $k_{34} = f(g(k_3), g(k_4))$
- $k_{14} = f(g(K_{12}), g(k_{34}))$
- $k = f(g(k_{58}), g(k_{14}))$



OFT – nod nou

Ex: u_3 se alatura grupului

Trebuie modificate k_{34}, k_{14}, si k

Trebuie transmise 3 chei orbite

$g(k_3)$, $g(k'_{34})$, $g(k'_{14})$ criptate

respectiv cu k_4, k_{12} si k_{58} adica

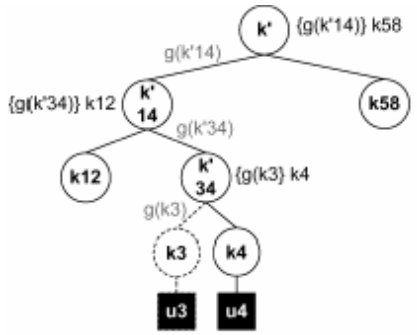
- $\{g(k_3)\}k_4$, $\{g(k'_{34})\}k_{12}$, $\{g(k'_{14})\}k_{58}$

Noile chei pot fi calculate de fiecare membru al grupului (care are cheia respectiva ptr decriptare)

$k'_{34} = f(g(k_3), g(k_4))$

$k'_{14} = f(g(k_{12}), g(k'_{34}))$

$k' = f(g(k'_{14}), g(k_{58}))$



Protocoale descentralizate - Iotus

Propus de Mittra

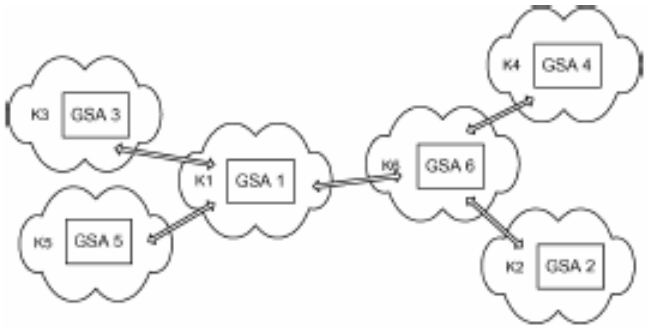
Grupul impartit in sub-grupuri

Fiecare gestionat de un GSA – Group Security Agent

Formeaza un grup de nivel inalt gestionat de Group Security Controller

Sub-grupurile au chei diferite, independente

Necesita translatarea mesajelor transmise de la un grup la altul



DEP - Dual-Encryption Protocol

Propus de Dondeti

Sub-grupuri, fiecare controlat de un SGM – Sub-Group Manager

Trei grupuri de KEK

- KEK_{i1} – partajata de SGM_i si membrii grupului i
- KEK_{i2} – partajata de **Group Controller - GC** si membrii grupului i fara SGM_i
- KEK_{i3} – partajat de GC si SGM_i

DEK – Data Encryptioin Key – una singura pentru grup

Distributie DEK

- Transmite pachete $\{\{DEK\}KEK_{i2}\}KEK_{i3}$
- SGMi recupereaza $\{DEK\}KEK_{i2}$; el nu cunoaste KEK_{i2}
- Cripteaza mesajul $\{\{DEK\}KEK_{i2}\}KEK_{i1}$ si-l trimite subgrupului i
- Fiecare membru al subgrupului i decripteaza de doua ori si obtine DEK

Schimbare grup i

- SGM_i schimba KEK_{i1} si-l transmite membrilor actuali ai grupului
- Urmeaza schimbarea DEK
- Membrii exclusi pot primi mesajele grupului pana la schimbarea DEK

MARKS

Propus de Briscoe

Timpul impartit in intervale

Foloseste o cheie de criptare distincta pentru fiecare interval

Cheile sunt frunze intr-un arbore hash generat dintr-o singura sursa

Notatie: $S_{i,j}$ inseamna nodul j la nivelul i

- Calculeaza adancimea D (conform numar total chei N = 2^D)
- Alege aleator radacina $S_{0,0}$
- Genereaza sursele intermediare *left* $S_{1,0}$ si *right* $S_{1,1}$

$S_{1,0} = b(ls(S_{0,0}))$

$S_{1,1} = b(rs(S_{0,0}))$

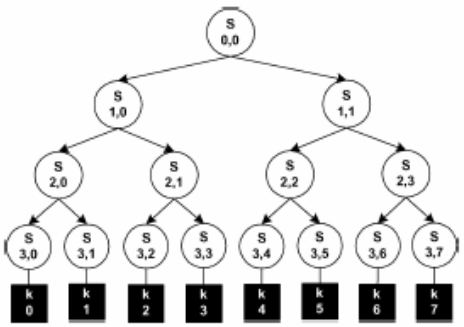
ls = left shift one bit

rs = right shift one bit

b = blinding function

- Genereaza similar celelalte nivele

Un membru care participa la grup in intervalele 3-7 primeste sursele $S_{3,3}$ si $S_{1,1}$



CS - Cipher Sequences

Propus de Moldva si Pannetrat

f(S,a) se numeste Cipher Group daca are urmatoarele caracteristici:

- Exista o secventa de n elemente $a_1,\ldots, a_n$
- Exista o secventa de n+1 elemente $S_0,\ldots S_n$

  $S_i = f(S_{i-1},a_i)$ pentru i>0 si valoarea initiala S_0 si
- Pentru fiecare cuplu (i,j) cu i<j exista o functie $h_{i,j}$ a.i.

  $S_i = h_{i,j}(S_j)$

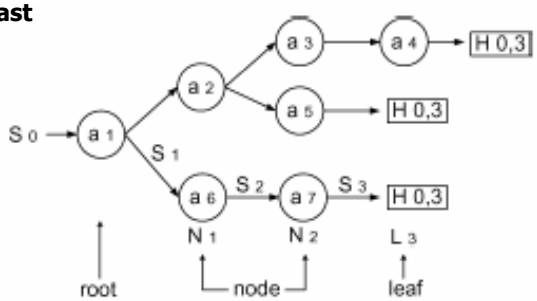
Grupul multicast este pus intr-un arbore cu sursa in radacina, membrii in frunze si nodurile interne ca elemente intermediare.

S_0 informatia de transmis multicast

a_i>0 info asociata cu nod N_i

- **Ni primeste S_j de la parinte**
- **Calculeaza $S_i = f(S_j, a_i)$**
- **Transmite S_i copiilor**
- **Frunzele cunosc $h_{0,n}$ si calculeaza**

  $S_0 = h_{0,n}(S_n)$

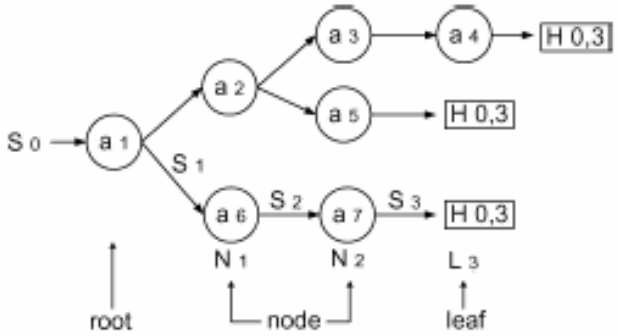


CS - 2

Frunza compusa din mai multi membri (subgrup)
Toti impart aceeasi functie $h_{0,n}$

La schimbare componenta

- Nodul N_n primeste o noua valoare a'_n
- Toti membrii din frunza primesc o noua functie $h'_{0,n}$
- Un membru eliminat din subgrup nu primeste $h'_{0,n}$



Protocoale distribuite - Group Diffie-Hellman key exchange

Propus de Steiner

Grupul are n membri; fiecare membru i are un numar secret xi

Grupul stabileste doua numere prime (q si a) si incepe sa calculeze distribuit valorile intermediare; protocolul are n runde

- Primul membru calculeaza a^{x1} si o paseaza urmatorului
- Fiecare membru primeste setul de valori intermediare si genereaza un nou set folosind numarul sau secret
  - Un set generat de membrul **i** va avea **i** valori intermediare cu **i-1** exponenti si o valoare cardinal cu **i** exponenti
  - De ex: al patrulea membru primeste $\{a^{x2.x3}, a^{x1.x3}, a^{x1.x2}, a^{x1.x2.x3}\}$
    si genereaza $\{a^{x2.x3.x4}, a^{x1.x3.x4}, a^{x1.x2.x4}, a^{x1.x2.x3}, a^{x1.x2.x3.x4}\}$
    valoarea cardinal este $a^{x1.x2.x3.x4}$
  - Membrul n
    - calculeaza cheia k din valoarea cardinal $k = a^{x1.x2.x3...xn} \bmod q$
    - Ridica toate valorile intermediare la valoarea sa secreta
    - Multicast tot setul
  - Fiecare membru extrage valoarea intermediara corespunzatoare si calculeaza cheia k

Octopus

Propus de Becker si Wille

Bazat pe DH

Grupul impartit in patru subgrupuri

1. Fiecare subgrup calculeaza intern o valoare DH acumulata de un leader de subgrup:

   $Isg = a^{u1...un/4}$

2. Fie liderii A, B, C si D

   A si B schimba valorile intermediare (Ia si Ib) si calculeaza $a^{Ia.Ib}$

   C si D schimba valorile intermediare (Ic si Id) si calculeaza $a^{Ic.Id}$

3. A si C schimba valorile calculate si obtin $a^{Ia.Ib.Ic.Id}$

   Similar B si D

4. Fiecare trimite fiecarui membru **j** al subrgupului sau valoarea $a^{Ia.Ib.Ic.Id/uj}$

   Fiecare membru calculeaza valoarea $a^{Ia.Ib.Ic.Id}$ si cheia $k = a^{Ia.Ib.Ic.Id} \bmod q$

Numar Runde: 2(n-4)/4 + 2

CKA - Conference Key Agreement

Propus de Becker si Wille

Cheia este generata cu o functie de combinare f:

$$K = f(N1, h(N2), \ldots, h(Nn))$$

- h – functie one-way
- n – dimens grup
- Ni – contributia membrului i

Protocol

1. N-1 membri difuzeaza in clar contributiile lor Ni
2. Liderul de grup (de ex U1) cripteaza N1 cu cheile publice ale celorlalti
   Difuzeaza mesajul
3. Fiecare membru (din cei n-1) decripteaza N1 si genereaza K

Numar Runde: 3

D LKH – Distributed Logical Key Hierarchy

Propus de Rodeh

Nu are un Group Controller care sa stie toate cheile

Subarborii stabilesc cheile

Fie subarborii L si R

Liderii m_l respectiv m_r

Cheile K_L respectiv K_R

Protocol

1. m_l alege o noua cheie k_{LR} si o trimite lui m_r pe un canal securizat
2. m_l cripteaza k_{LR} cu k_L si difuzeaza rezultat tuturor membrilor subarbore L

   m_r cripteaza k_{LR} cu k_R si difuzeaza rezultat tuturor membrilor subarbore R
3. Toti membrii din R si L primesc cheia noua

Exemplu:

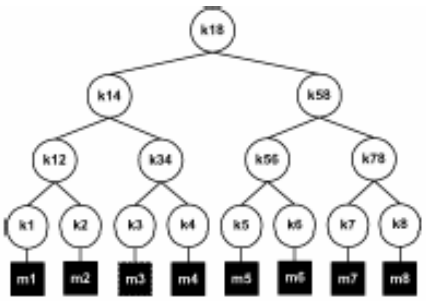
Membrii 1,2 stabilesc cheia **k12**

3,4 stabilesc cheia **k34**

1,2 si 3,4 stabilesc cheia **k14**

....

1,2,3,4 si 5,6,7,8 stabilesc cheia **k18**



D OFT – Distributed One-way Function Tree

Propus de Dondeti

Foloseste OFT – One-way Function Tree

Diferente:

- Nu exista un Group Controller
- Fiecare membru face
  - Control acces
  - Generare cheie proprie
  - Transmitere cheie proprie "orbita" catre fratele sau

Similitudini

- Fiecare membru cunoaste toate cheile din calea spre radacina si toate cheile "orbite" ale fratilor acestor noduri

Numar runde: $\log_2 n$

DH-LKH - Diffie-Hellman Logical Key Hierarchy

Propus de Perrig si Kim

Foloseste LKH

Diferente

- Membrii genereaza cheile in nivelul superior folosind algoritmul Diffie-Hellman (si nu one-way function)

  $k = a^{kl.kr} \bmod p$

Exemplu

$k_{12} = a^{k1.k2} \bmod p$

$k_{34} = a^{k3.k4} \bmod p$

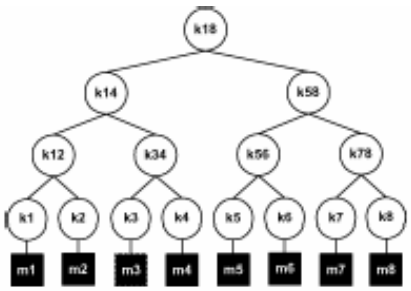
$k_{14} = a^{k12.k34} \bmod p$

...

$k_{58} = a^{k56.k78} \bmod p$

$k_{18} = a^{k14.k58} \bmod p$

Numar runde: $\log_2 n$



Delegare (1)

Abordari

- Certificate nominale
- Certificate anonime (necesita protectie contra copierii ilegale) – ex. proxy

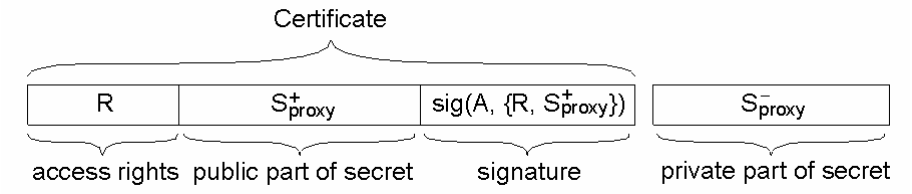
Proxy – token care permite posesorului sa opereze cu aceleasi drepturi (sau drepturi mai restranse) ca garantul

A creaza proxy

R – drepturi de acces

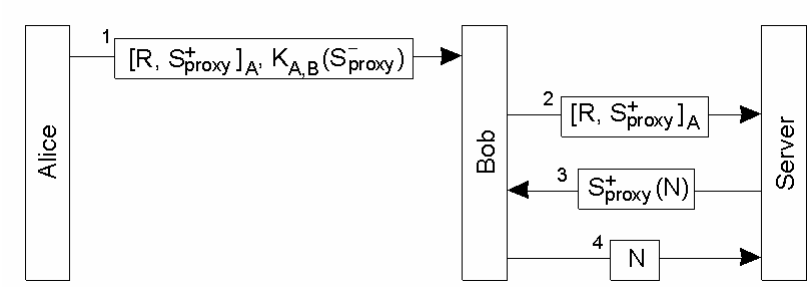
S^{+}_{proxy} – partea publica a secretului folosit pentru a autentifica detinatorul certificatului ("intrebarea")

S^{-}_{proxy} – "raspunsul"



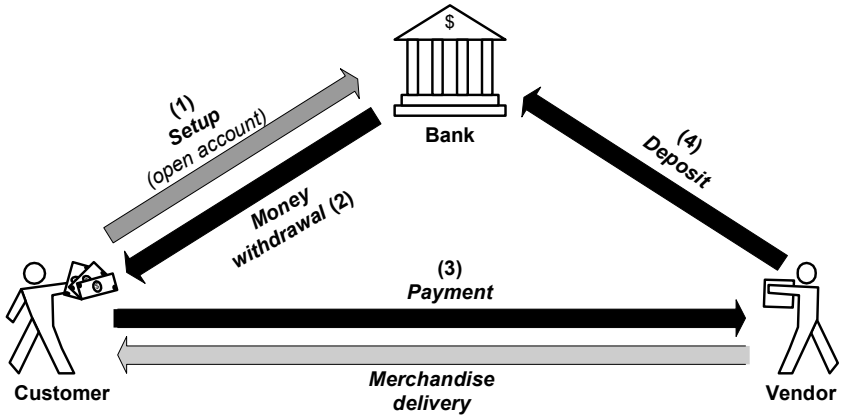
Distributed Systems, Tanenbaum

Utilizare proxy pentru a delega si proba proprietatea asupra drepturilor de acces



Ciclul de viata al cache-ului electonic

1. Inregistrare (deschidere cont)
2. Extragere
3. Cheltuire (or plata)
4. Depunere



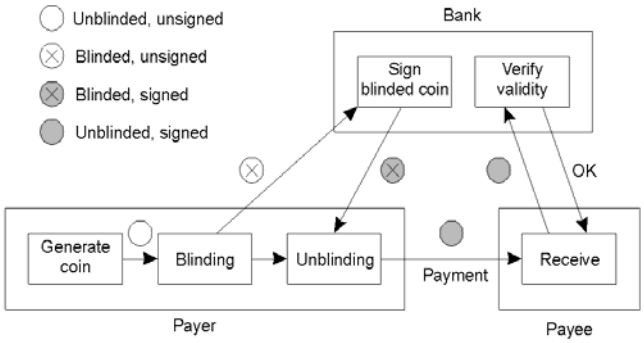
Semnaturi oarbe (Chaum 1982)

Notatie: Cheile RSA ale bancii (publica = (n,e); privata = (n,d))
k este un "secret" ales aleator de Alice

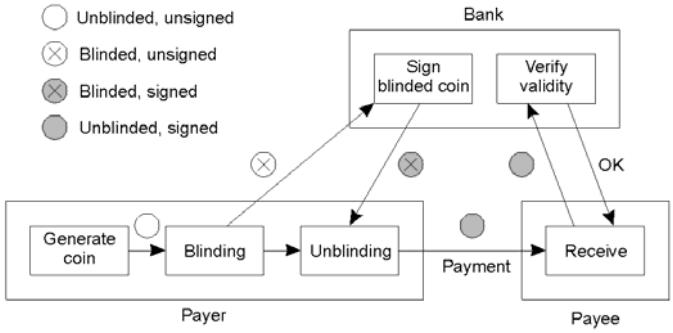
Protocol: Alice genereaza o moneda de $10 notata m si cere bancii sa o semneze (fara ca banca sa o poata urmari ulterior)

Orbire — Alice genereaza k aleator, $0 \le k \le n-1$, GCD(n, k) =1
Calculeaza $t = mk^e \bmod n$

Semnare — Banca semneaza t cu cheia privata d, obtinand t^d



Protocoale: E-cash



- "Dez"-orbire Alice anuleaza orbirea prin calculul
  $s = t^d / k \bmod n = (mk^e)^d / k \bmod n$
  $s = m^d k^{ed-1} \bmod n = m^d \bmod n$
- Alice foloseste m^d ca moneda de $10 pentru plata lui Bob

Problema: nu releva identitatea in cazul dublei cheltuiri

Alin cumpara o carte de la Crypto Store