SEMINAR IZ KRIPTOGRAFIJE

Porazdelitev in uskladitev ključev

Erik Schlegel

FMF

8. junij, 1998

Na zadnjem seminarju:

- Vladimir Ban je končal s 7. poglavjem
  - Hash funkcije v kriptosistemih
  - MD4 Hash funkcija
  - Časovne oznake
- rešitve nalog 6. poglavja: 3, 5 in 6

Današnji načrt:

- Porazdelitev in uskladitev ključev
  - Porazdelitev ključev
  - Kerboros
- reševanje nalog

Uvod

Sistemi z javnimi ključi imajo prednost pred sistemi s privatnimi ključi, ker varnega kanala ne potrebujemo pri izmenjavi tajnih ključev. Večina sistemov z javnimi ključi je počasnejša od sistemov s privatnimi ključi, na primer DES. V praksi se uporablja sisteme s privatnimi ključi za šifriranje daljših besedil.

Vrnimo se na problem izmenjave tajnih ključev. Obravnavali bomo več različnih protokolov za tajne ključe. Razlikovali bomo med porazdelitvijo in uskladitvijo ključev.

Definicija. *Sistem porazdelitve ključev* je mehanizem, kjer na začetni stopnji zaupljiv strežnik generira in porazdeli tajne podatke uporabnikom tako, da lahko vsak par uporabnikov kasneje izračuna ključ, ki je nepoznan ostalim.

Definicija. *Uskladitev ključev* označuje protokol, kjer dva ali več uporabnikov sestavi skupen tajni ključ, s komunikacijo po javnem kanalu. Vrednost ključa je določena s funkcijo vhodnih podatkov.

Strežnik TA
(*trusted authority*)

Imamo omrežje, ki ni varno in na katerem je n uporabnikov. V nekaterih shemah se pojavi strežnik (TA), ki je odgovoren za

- potrjevanje identifikacije
- izbiro in prenos ključev
- itd.

Potreba po zaščiti pred potencialnimi nasprotniki. Ločimo dve vrsti sovražnikov:

- pasivni
- aktivni

Pasivni sovražnik je osredotočen na prisluškovanje sporočilom, ki se pretakajo po kanalu. Več nevšečnosti nam lahko naredi aktivni sovražnik, ene od teh so:

- spreminjanje sporočil
- shranjevanje sporočil za kasnejšo uporabo
- poskuša se maskirat kot neki uporabnik omrežja

Cilj aktivnega sovražnika je lahko:

- prelisičiti U in V tako, da sprejmeta neveljaven ključ kot veljaven
- prepriča U in V, da sta si izmenjala ključ, čeprav si ga nista.

U in V sta uporabnika omrežja.

Porazdelitev ključev

Recimo, da imamo omrežje z n uporabniki. Strežnik TA generira $\binom{n}{2}$ ključev in vsak ključ pošlje enolično določenemu paru uporabnikov omrežja.

Potrebujemo varen kanal med TA in vsakim uporabnikom omrežja. Problem je vsaj $\mathcal{O}(n)$ velik, kar pomeni, da rešitev ni praktična. Želimo boljšo rešitev, kar nas pripelje do problema, ki je $\mathcal{O}(1)$ velik.

Blom-ova shema ($k = 1$)

- p praštevilo, javno
- Za vsakega uporabnika U izberemo $r_U \in \mathbb{Z}_p$ tako, da so različni, javni.
- TA izbere tri naključne elemente $a, b, c \in \mathbb{Z}_p$ (ne nujno različni) in oblikuje polinom
$$f(x, y) = a + b(x + y) + cxy \bmod p.$$
- Za vsakega uporabnika U izračuna TA polinom
$$g_U(x) = f(x, r_U) \bmod p$$
in mu ga pošlje po varnem kanalu.
Opazimo to, da je $g_U(x)$ linearen polinom tako, da ga lahko zapišemo v naslednji obliki
$$g_U(x) = a_U + b_U x,$$
kjer
$$a_U = a + br_U \bmod p$$
in
$$b_U = b + cr_U \bmod p.$$

- Za komunikacijo med U in V se uporabita ključa
$$K_{U,V} = K_{V,U} = f(r_U, r_V) =$$
$$= a + b(r_U + r_V) + cr_U r_V \bmod p,$$
kjer U izračuna $K_{U,V}$ kot
$$f(r_U, r_V) = g_U(r_V)$$
in V izračuna $K_{U,V}$ kot
$$f(r_U, r_V) = g_V(r_U).$$

Izrek 1 Blom-ova shema s $k = 1$ je brezpogojno varna pred posameznimi uporabniki.

Dokaz: Recimo, da želi uporabnik W izračunati ključ
$$K_{U,V} = a + b(r_U + r_V) + cr_U r_V \bmod p.$$
Vrednosti r_U in r_V so javne, toda a, b in c so neznane. W pozna vrednosti
$$a_W = a + br_W \bmod p$$
in
$$b_W = b + cr_W \bmod p,$$
ker so to koeficienti polinoma $g_W(x)$, ki je bil poslan W-ju od strežnika TA.
Kar bomo naredili je to, da pokažemo, da je informacija poznana W-ju konsistentna s poljubno vrednostjo $l \in \mathbb{Z}_p$ za ključ $K_{U,V}$. Torej, W ne more izločiti katerekoli vrednosti za $K_{U,V}$.

Poglejmo si naslednjo matrično enačbo v ($\mathbb{Z}_p$):
$$\begin{pmatrix} 1 & r_U + r_V & r_U r_V \\ 1 & r_W & 0 \\ 0 & 1 & r_W \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} l \\ a_W \\ b_W \end{pmatrix}.$$

Prva enačba vsebuje hipotezo, da je $K_{U,V} = l$.
Drugi dve enačbi povesta, da W pozna a, b in c iz $g_W(x)$.
Determinanta koeficientov matrike je
$$r_W^2 + r_U r_V - (r_U + r_V) r_W = (r_W - r_U)(r_W - r_V),$$
kjer se računa v $\mathbb{Z}_p$. Ker $r_W \neq r_U$ in $r_W \neq r_V$ sledi, da ima matrika koeficientov determinanto različno od nič in zato ima matrična enačba enolično rešitev za a, b in c. Drugače povedano, katerakoli možna vrednost l za ključ $K_{U,V}$ je konsistentna z informacijo, ki jo ima W.

Posplošitev

Posplošena shema, ki je varna pred koalicijo velikosti k, je naslednja: Edina sprememba je na drugem koraku, kjer TA uporablja polinom $f(x, y)$, ki je oblike
$$f(x, y) = \sum_{i=0}^{k} \sum_{j=0}^{k} a_{i,j} x^i y^j \bmod p,$$
kjer je $a_{i,j} \in \mathbb{Z}_p$ ($0 \leq i \leq k$, $0 \leq j \leq k$), in $a_{i,j} = a_{j,i}$ za vsak i, j. Ostali del protokola se ne spremeni.

Diffie-Hellman-ova porazdelitev ključev

- p praštevilo, $\alpha \in \mathbb{Z}_p^*$ primitiven element, javna.
- V izračuna
$$K_{U,V} = \alpha^{a_U a_V} \bmod p = b_U^{a_V} \bmod p,$$
z uporabo javne vrednosti b_U iz U-jevega certifikata skupaj z lastno tajno vrednostjo a_V.
- U izračuna
$$K_{U,V} = \alpha^{a_U a_V} \bmod p = b_V^{a_U} \bmod p,$$
z uporabo javne vrednosti b_V iz V-jevega certifikata skupaj z lastno tajno vrednostjo a_U.

Izrek 2 Razbitje ElGamal-ovega kriptosistema je ekvivalentno reševanju Diffie-Hellman-ovega problema.

Dokaz: Najprej obnovimo kako ElGamal-ovo šifriranje in dešifriranje poteka. Ključ je $K = (p, \alpha, a, \beta)$, kjer $\beta = \alpha^a \bmod p$ (a je tajni in p, α in β so javni). Za tajno naključno število $k \in \mathbb{Z}_{p-1}$ je
$$e_K(x, k) = (y_1, y_2),$$
kjer
$$y_1 = \alpha^k \bmod p$$
in
$$y_2 = x\beta^k \bmod p.$$
Za $y_1, y_2 \in \mathbb{Z}_p^*$ je
$$d_K(y_1, y_2) = y_2 (y_1^{\alpha})^{-1} \bmod p.$$

Recimo, da imamo algoritem A, ki reši Diffie-Hellman-ov problem in podano ElGamal-ovo šifriranje (y_1, y_2). Z Uporabo algoritma A na podatkih p, α, y_1 in β dobimo vrednost

$$A(p, \alpha, y_1, \beta) = A(p, \alpha, \alpha^k, \alpha^a) =$$

$$= \alpha^{ka} \bmod p = \beta^k \bmod p.$$

Potem dešifriranje (y_1, y_2) lahko enostavno izračunamo kot

$$x = y_2(\beta^k)^{-1} \bmod p.$$

Recimo, da imamo še algoritem B, ki izvrši ElGamal-ovo dekripcijo. Torej B vzame podatke p, α, β, y_1 in y_2 in izračuna

$$x = y_2(y_1^{\log_\alpha \beta})^{-1} \bmod p.$$

Sedaj je za dane podatke p, α, β in γ Diffie-Hellman-ovega problema enostavno videti, da je

$$B(p, \alpha, \beta, \gamma, 1)^{-1} = 1((\gamma^{\log_\alpha \beta})^{-1})^{-1} \bmod p =$$

$$= \gamma^{\log_\alpha \beta} \bmod p,$$

kar smo želeli.

Kerboros

Do sedaj smo imeli sisteme, kjer vsak par uporabnikov izračuna fiksen ključ, ki se ne spreminja. Zaradi tega je preveč izpostavljen nasprotnikom. Pojavi se želja po oblikovanju tako imenovanega sejnega ključa, ki se oblikuje brž ko se pojavita dva z željo po komunikaciji. Gre za obnovitev ključa. Kerboros je tak sistem, ki uporablja privatno kriptografijo.

Slabost tega sistema je zahteva po sinhronizaciji ur uporabnikov omrežja. Določena časovna variacija je dovoljena.

Prenos sejnega ključa z uporabo Kerboros-a

- U zahteva od TA sejni ključ za komunikacijo z V-jem.

- Strežnik TA izbere naključni sejni ključ K, časovno oznako T in življensko dobo L.

- TA izračuna
  $$m_1 = e_{K_U}(K, ID(V), T, L)$$
  in
  $$m_2 = e_{K_V}(K, ID(U), T, L)$$
  in pošlje m_1 in m_2 U-ju.

- U uporabi dešifrirno funkcijo d_{K_U} za izračun K, T, L in ID(V) iz m_1. Potem izračuna
  $$m_3 = e_K(ID(U), T)$$
  in pošlje m_3 V-ju skupaj s sporočilom m_2, ki ga je dobil od strežnika TA.

- V uporabi dešifrirno funkcijo d_{K_V} za izračun K, T, L in ID(U) iz m_2. Potem uporabi d_K za izračun T in ID(U) iz m_3. Preveri da sta tako dobljeni vrednosti za T in ID(U) enaki prejšnjim. Če je tako, potem izračuna V
  $$m_4 = e_K(T + 1)$$
  in ga pošlje U-ju.

- U dešifrira m_4 z uporabo e_K in se prepriča, da je rezultat T+1.

V tem protokolu se prenašajo različne funkcije sporočil. Sporočili m_1 in m_2 poskrbita za tajnost pri prenosu sejnega ključa K. m_3 in m_4 se uporabljata kot potrdilo sejnega ključa K tako, da se U in V prepričata, da imata isti sejni ključ K.