Teorija kodiranja in kriptografija 2013/2014

Kriptografija v praksi

Arjana Žitnik

Univerza v Ljubljani, Fakulteta za matematiko in fiziko

Ljubljana, 22. 4. 2014

Vsebina

- Upravljanje ključev pri simetričnih kriptosistemih
- Infrastruktura javnih ključev (PKI)
- Avtentikacija
- Protokoli za uskladitev ključev
- Primeri uporabe

Vprašanja

- Od kod dobimo ključe?
- Zakaj zaupamo ključem?
- Kako vemo čigav ključ imamo?
- Kaj se zgodi, če je kompromitiran (izgubljen) zasebni ali tajni ključ? Kdo je odgovoren?
- Kako preklicati ključ?
- Kako omogočimo servis preprečitve zanikanja?

Ta vprašanja veljajo tako za simetrične (tajne) ključe kakor tudi za javne in zasebne ključe.

Ključi pri simetričnem kriptosistemu

- Izmenjava ključa zahteva varen kanal (osebno, kurir,...).
- V skupini n uporabnikov mora vsak uporabnik deliti različen ključ z vsakim uporabnikom. Vseh tajnih ključev je $\binom{n}{2}$.
- Vsak uporabnik mora hraniti $n-1$ različnih tajnih ključev.
- Verodostojnost uporabnika in sporočila sta povezana.

Center zaupanja

V omrežju, ki ni varno, se v nekaterih shemah pojavi agencija, ki je odgovorna za

- potrjevanje identitete,
- izbiro in prenos ključev,
- ...

Imenovali jo bomo **center zaupanja** ali **verodostojna agencija** (angl. Trusted Authority – TA).

Distribucija ključev

Sistem za distribucijo ključev je mehanizem, pri katerem verodostojna agencija (trusted authority) v naprej generira in distribuira tajne podatke uporabnikom. Kasneje lahko vsak par uporabnikov izračuna skupen ključ, ki je nepoznan ostalim.

Distribucija ključev po Diffie-Hellmanovi shemi

Naj bo p veliko praštevilo in α generator grupe $\mathbb{Z}_p^*$.
Vsak uporabnik U ima

- zasebni ključ $a_U \in \{0, 1, \ldots, p-2\}$ in
- javni ključ $b_U = \alpha^{a_U} \pmod{p}$.

Uporabnika U in V lahko izračunata skupen ključ

$$K_{U,V} = \alpha^{a_U a_V}.$$

Varnost temelji na težavnosti Diffie-Hellmanovega problema: nasprotnik ne more hitro izračunati ključa $\alpha^{a_U a_V}$, če pozna le α^a_U, α^a_V in α.

Sejni ključi

Če ima vsak par uporabnikov fiksen ključ, ki se ne spreminja, je preveč izpostavljen nasprotnikom. Zato se več uporablja tako imenovan sejni ključ, ki se oblikuje le takrat, ko dva uporabnika želita komunicirati.

- Omeji se velikost šifriranega teksta z istim ključem
- Če se razkrije sejni ključ, naj ne bi bilo mogoče s tem izvedeti nobene informacije o tajnem ključu osebe.
- Omeji se količina informacij, ki jih je treba dolgoročno hraniti, saj se sejni ključi generirajo samo takrat, ko jih potrebujemo.

Center za distribucijo ključev

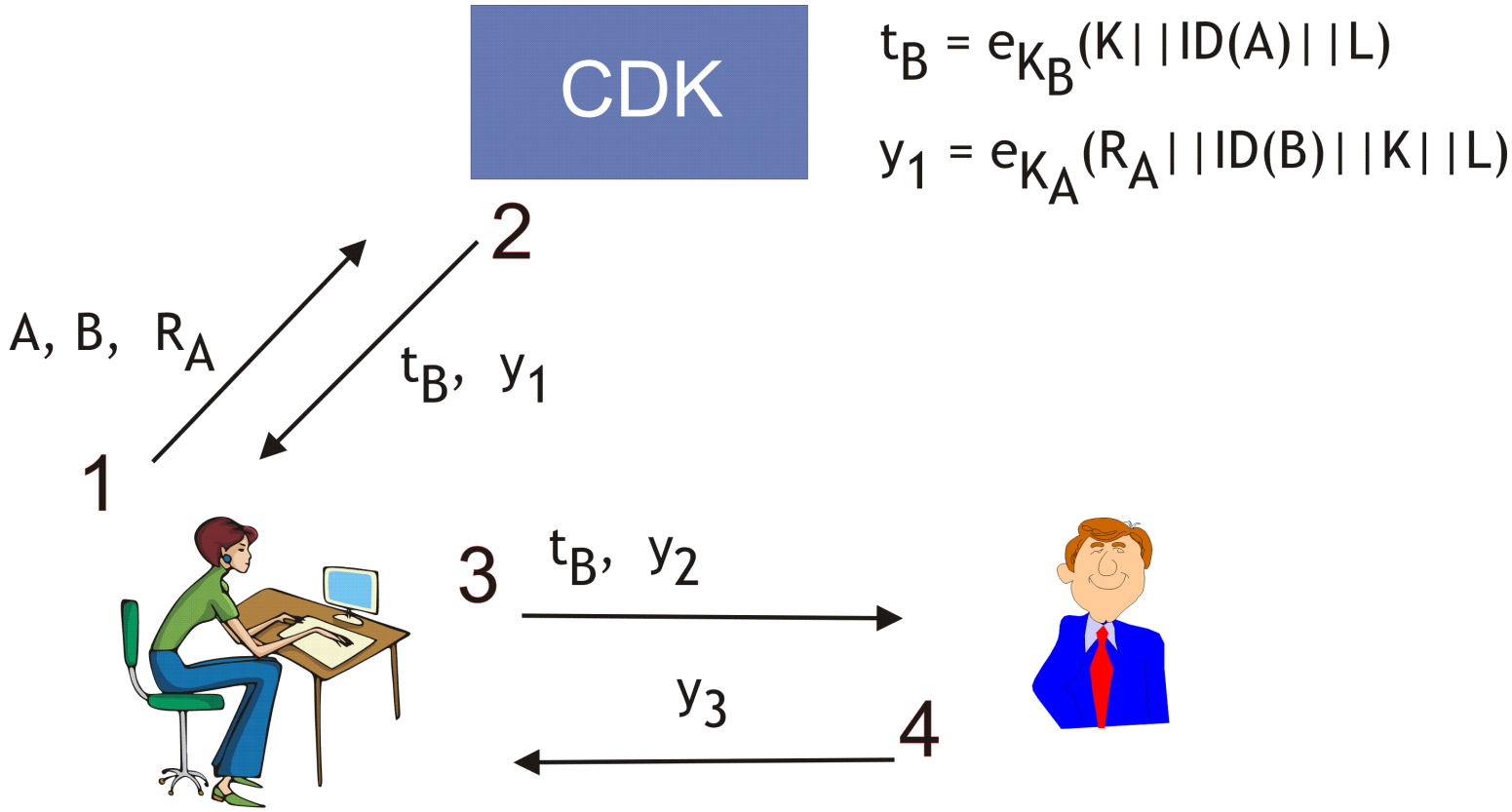
- Vsak uporabnik deli ključ s Centrom za distribucijo ključev (CDK)
- Uporabnika zahtevata od CDK, da jima dodeli sejni ključ
  CDK pošlje obema uporabnikoma sejni ključ
- Primer: Kerberos, MIT
  zagotovi avtentikacijo in sejne ključe
  posodobljena verzija lahko uporablja AES.

Kerberos (poenostavljeno)

Vsak uporabnik U deli z agencijo TA tajni DES ključ K_U.
ID(U) so podatki, ki enolično identificirajo osebo U

- Anita izbere naključno število R_A in
  pošlje TA podatke $ID(Anita)$, $ID(Bojan)$ in R_A.
- TA izbere naključni ključ K in čas veljavnosti ključa L.
  TA izračuna vstopnico $t_{Bojan} = e_{K_{Bojan}}(K||ID(Anita)||L)$ in
  $y_1 = e_{K_{Anita}}(R_A||ID(Bojan)||K||L)$.
  TA pošlje Aniti t_B in y_1.
- Anita določi točen čas $time$ in
  izračuna $y_2 = e_K(ID(Anita)||time)$.
  Anita pošlje Bojanu t_{Bojan} in y_2.
- Bojan izračuna $y_3 = e_K(time + 1)$.
  Bojan pošlje Aniti y_3.

Kerberos - slika



Problemi CDK

- Verodostojnost CDK – CDK moramo brezpogojno zaupati; očitna tarča napadalca
- CDK lahko prisluškuje vsem komunikacijam med uporabniki
- Zahteva stalno zvezo s centrom (ozko grlo)

Digitalno potrdilo ali certifikat

Digitalno potrdilo ali certifikat je digitalno podpisan dokument, ki poveže uporabnika U z njegovim javnim ključem.

Certifikat izda in podpiše certifikatna agencija (CA).

Certifikat $C(U)$ osebe U je sestavljen iz:

- podatkovnega dela $D(U)$: uporabnikova identifikacija, njegov javni ključ in druge informacije kot npr. veljavnost,
- podpisanega dela $\mathrm{sig}_{CA}(D(U))$: CA-jev podpis podatkovnega dela.

Certifikatna agencija

Certifikatna agencija je center zaupanja, ki digitalno podpiše certifikat in

- jamči za pristnost vsebine certifikatov,
- vezanost javnega ključa na certifikat in
- lastništvo podpisanega certifikata.

Dodatno CA opravlja še druge funkcije, kot so na primer

- določanje in objavljanje svojo politiko delovanja,
- upravljanje z glavnimi ključi,
- preklic certifikatov,
- distribucijo seznama preklicanih certifikatov,

CA najprej pridobi svoj certifikat: generira in podpiše ga lahko kar sama ali pa ga pridobi od kakšne druge CA.

Pridobitev certifikata

Uporabnika A pridobi certifikat certifikatne agencije CA po naslednjem postopku:

1. Generiranje para ključev uporabnika A.
2. Uporabnik poda zahtevo za svoj certifikat na CA.
3. Preverjanje identitete uporabnika A.
4. Preverjanje para ključev uporabnika A, če uporabnik sam generira svoj par ključev, sicer par ključev generira CA.
5. CA naredi A-jev certifikat.
6. CA posreduje uporabniku A njegov certifikat.
7. A preveri, da je certifikat pravilen.

Pridobitev certifikata - nadaljevanje

Uporabnik lahko sam generira svoj par ključev (če to zna) in posreduje CA v podpis samo svoj javni ključ, lahko pa prepusti generiranje para ključev certifikatni agenciji.

Preverjanje identitete je v praksi lahko zamudno in drago; pogosto CA preloži to delo na registracijski urad (RA), npr. pošto ali banko.

Uporaba certifikata

B pridobi avtentično kopijo *A*-jevega javnega ključa na naslednji način:

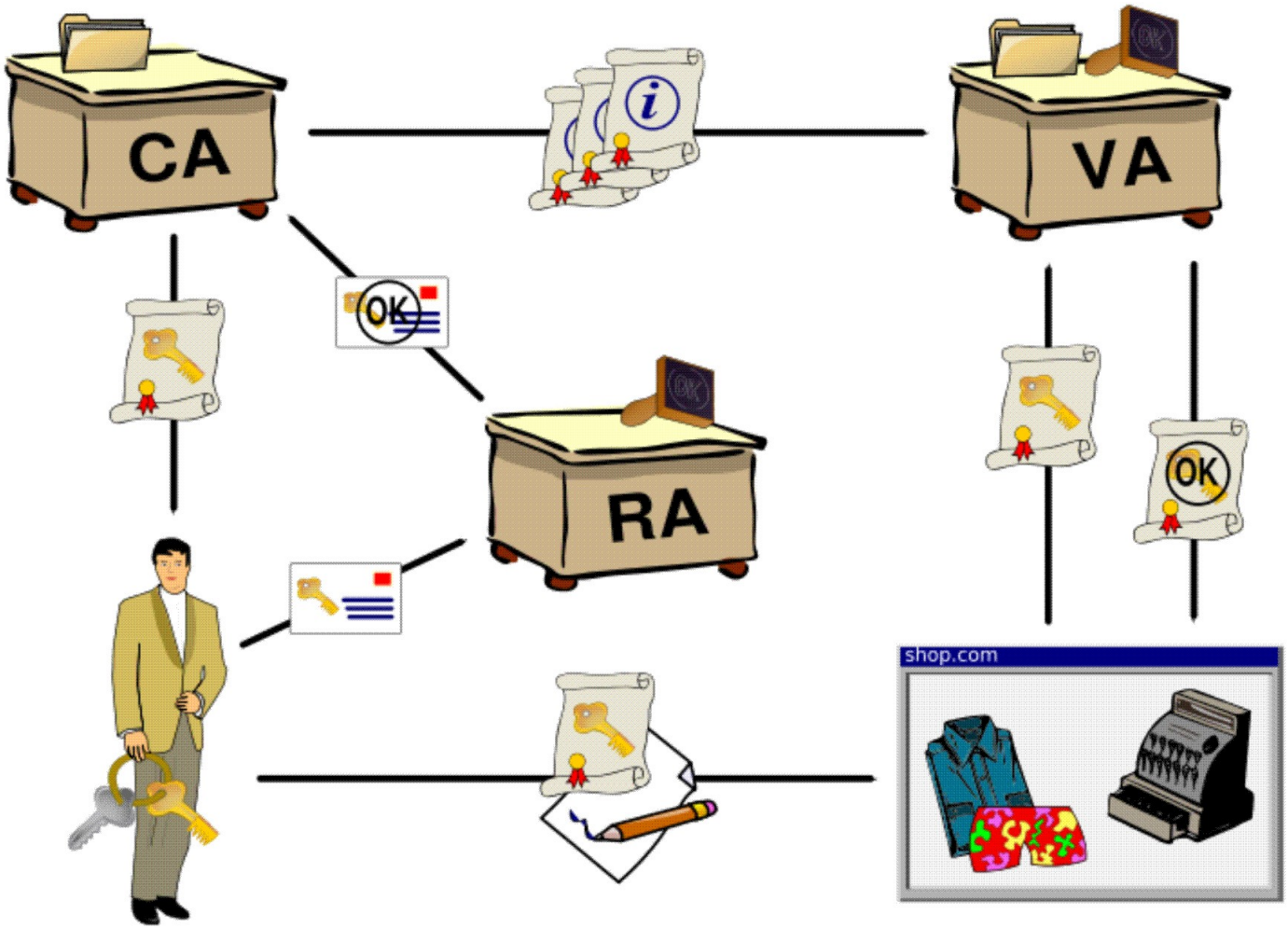
- pridobi avtentično kopijo javnega ključa CA (npr.dobljenega z brskalnikom ali operacijskim sistemom),
- pridobi $C(A)$ (preko nezavarovanega kanala),
- preveri podatke (veljavnost...) in podpis $\mathrm{sig}_{CA}(D(A))$.

Infrastruktura javnih ključev

Podporni servisi, ki so potrebni, da lahko tehnologijo javnih ključev uporabimo za večje projekte.

- Korenska certifikatna agencija (root CA), ki izdaja certifikate ostalim certifikatnim agencijam,
- certifikatne agencije, ki izdajajo certifikate registracijskim uradom in končnim uporabnikom ter skrbijo za objavo certifikatov ter seznamov preklicanih certifikatov v javnih direktorijih,
- registracijskih uradov (RA), ki so zadolženi za registracijo končnih uporabnikov na podlagi dokazila o njihovi identiteti,
- direktorijev, ki hranijo certifikate ter sezname preklicanih certifikatov,
- uporabnikov,
- ponudnikov storitev, kot so e-trgovina, e-uprava . . .

Infrastruktura javnih ključev - slika



Komponente PKI

- format certifikata,
- proces certificiranja,
- razdeljevanje certifikatov,
- modeli zaupanja,
- preklic certifikatov,
- politika certificiranja: podrobnosti o namenu in obsegu uporabe določenega certifikata,
- postopki in politike CA.

Standard X.509

X.509 je trenutno najbolj razširjeni standard za delo s certifikati.

Originalni standard X.509 je leta 1988 izdala International Telecommunications Union (X.509 v1). Od takrat je bil dvakrat posodobljen: leta 1993 (X.509 v2) in leta 1995 (X.509 v3).

Standard določa:

- Format certifikata
- Postopek preverjanja veljavnosti certifikata
- CRL

Dokumenti IETF: RFC3280, RFC2560, RFC3279, RFC3647...
Glej `http://www.ietf.org`.

Polja v certifikatu X.509

- Verzija,
- serijska številka certifikata,
- algoritem podpisovanja, ki ga je CA uporabila za podpis certifikata,
- izdajatelj,
- veljavnost,
- lastnik,
- javni ključ lastnika.

Dodatna polja pri verziji 3

V verzija 3 standarda X.509 so predvidena še nekatera dodatna polja. Dodana polja so sestavljena iz treh delov:

- iz tipa dodanega polja,
- kritičnosti dodanega polja in
- vrednosti dodanega polja.

Kritičnost dodanega polja pove aplikaciji, ki certifikat preverja, ali lahko to dodano polje ignorira ali ne.

Preklic certifikata

Razlogi za preklic certifikata:

- zasebni ključ je ukraden ali izgubljen, preden poteče veljavnost certifikata,
- podatki v certifikatu se spremenijo (na primer priimek, delovno mesto, ...),
- stopnja zaupanja do imetnika se je spremenila (prekinitev delovnega razmerja, kršitev pogodbe, ...),
- certifikat za svojo nalogo ni več potreben,
- ključ CA je kompromitiran (lahko tudi razbit),
- certifikat od kakšne nadrejene CA je preklican,
- izguba gesla ali PIN.

Preklic lahko zahteva imetnik certifikata ali certifikatna agencija.

Metode preklica certifikatov

Metode preklica certifikatov delimo glede na

- Način preverjanja: sprotno (on-line) preverjanje preko aktivne povezave z direktorijem, ki mu zaupamo, ali off-line preverjanje, kjer so podatki izračunani v naprej in postavljeni na mesto, ki ni nujno zanesljivo.
- Po tipu seznamov certifikatov: bele/črne liste. V bele liste postavimo veljavne certifikate, v črne liste postavimo neveljavne certifikate.
- Način dokazovanja: direkten dokaz (najdemo na listi) ali indirekten dokaz (ne najdemo na listi).
- Način razširjanja informacije o preklicu: strežnik lahko periodično pošilja podatke o preklicih vsem registriranim odjemalcem (model “push”), lahko pa si odjemalci podatke priskrbijo sami, ko jih potrebujejo (model “pull”).

Seznami preklicanih certifikatov

Certifikatna agencija lahko poskrbi za preklic certifikatov z izdajanjem seznama preklicanih certifikatov (*angl.* certificate revocation list CRL).

- CRL vsebuje digitalno podpisan seznam certifikatov, ki so preklicani, a jim še ni potekel rok veljavnosti.
- CRL se izdaja periodično, po potrebi pa tudi bolj pogosto.
- uporabnik mora pred uporabo certifikata vedno preveriti, da certifikat ni na seznamu CRL.

Slabosti seznamov preklicanih certifikatov

- Aplikacije pogosto ne preverjajo CRL.
- V velikem omrežju ja CRL težko najti.
- Čas med preklicem certifikata in objavo v CRL.
- Velikost CRL: zamudno generiranje in podpisovanje, prenos med strežniki, iskanje posameznega certifikata v seznamu (to lahko rešujemo z izpeljankami, a niso vedno podprte: Delta CRL, delitvene točke...)

Sprotno preverjanje statusa cetifikatov

Protokol za sprotno preverjanje statusa certifikatov (angl. *Online Certificate Status Protocol*) omogoča, da pridobimo čisto svežo informacijo, da dani certifikat še ni bil preklican.

Aplikacija sestavi zahtevek, v katerem navede serijsko številko certifikata (ali več certifikatov) in ga pošlje ustreznemu servisu. To je lahko CA sama, lahko pa je to pooblaščeni strežnik, ki mu pravijo *Certificate Status Responder*. Vse transakcije se ustavijo, dokler aplikacija ne dobi odgovora.

Odgovor je digitalno podpisan in vsebuje številko certifikata, interval veljavnosti ob času izdaje, čas veljavnosti odgovora ter status, ki je lahko “dober”, “preklican” ali “neznan”.
Odgovor “dober” pomeni samo, da certifikat ni bil preklican.

Slabosti OCSP

- Za vsak certifikat potrebno najti ustrezen strežnik, ki bo dal odgovor. Če odgovarja pooblaščeni strežnik, je tudi zanj potrebno preveriti, da ima veljaven certifikat. Lahko pa imajo pooblaščeni strežniki certifikate z zelo kratkim rokom veljavnostjo in jih ne preverjamo.
- Podpisovanje vsakega odgovora posebej je zelo zamudno, zato so strežniki za OCSP lahko žrtve napada “denial of service”: uporabnik ne more preveriti statusa certifikatov zaradi prevelikega hkratnega števila zahtevkov za preverjanje na strežniku.

Kako do certifikata

- Naredimo sami (npr. z OpenSSL, odprtokodnim orodjem, ki ustreza standardu FIPS 140-2.)
- Naročimo certifikat pri kakšni CA, ki jih izda brezplačno (Thawte, CaCert) ali komercialni CA.
- Naročimo certifikat pri slovenski CA.

Vprašanje: kdo bo našemu certifikatu zaupal?

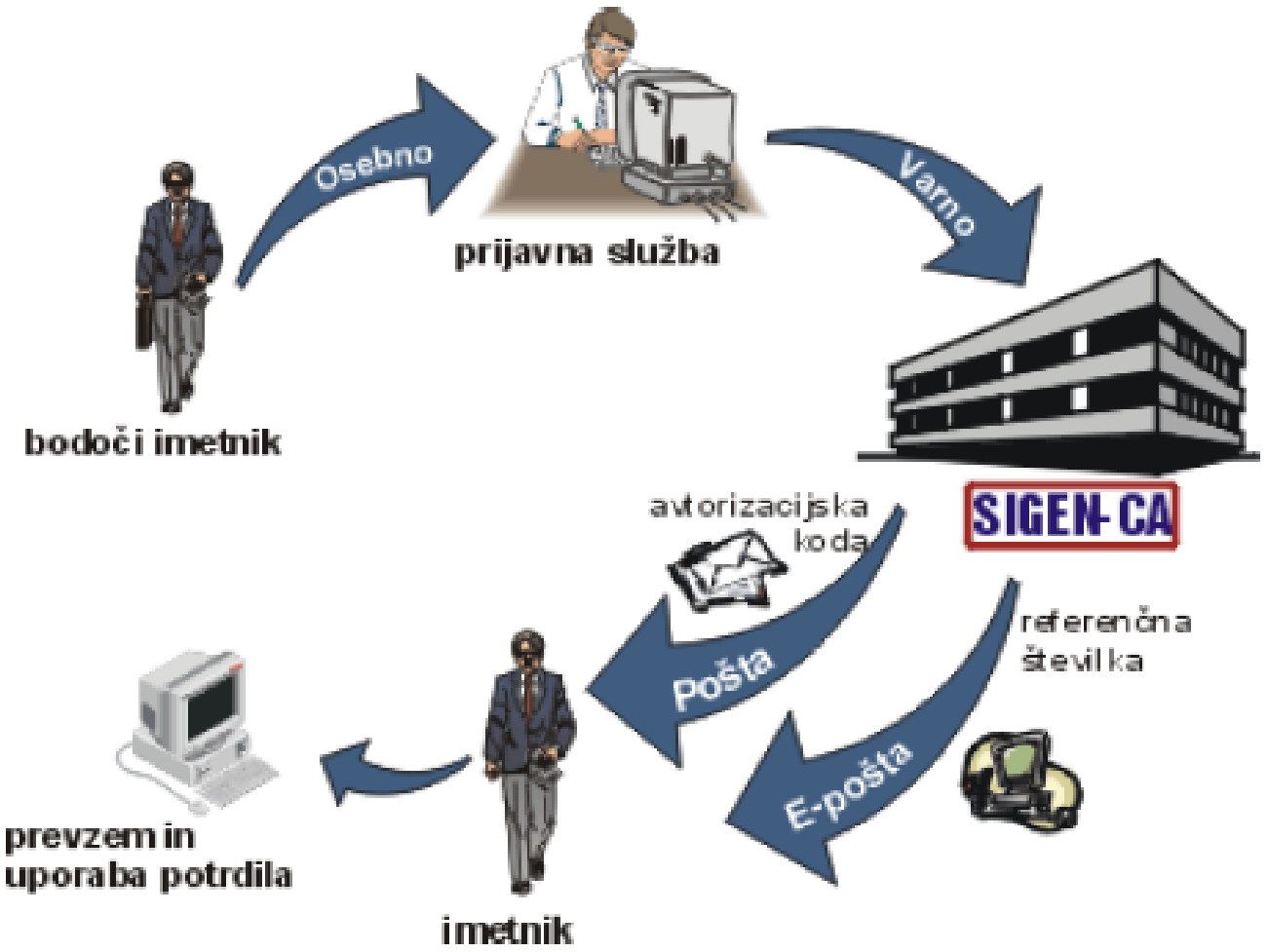
Uporaba certifikatov v Sloveniji

V Sloveniji je osnova za poslovanje s certifikati Zakon o elektronskem poslovanju in elektronskem podpisu ZEPEP in na njegovi osnovi izdana Uredba o elektronskem poslovanju in elektronskem podpisovanju.

Kvalificirana digitalna potrdila izdajajo

- Overitelj na Ministrstvu za javno upravo: SIGEN-CA in SIGOV-CA
- HALCOM-CA
- NLB-CA
- POŠTA®CA

Pridobitav potrdila SIGEN-CA



Vir: MJU, 2006

Avtentikacija

Kako dokažemo svojo identiteto tretji osebi?

Tehnike: predstavimo

- nekaj, kar smo (videz, prstni odtis...)
- nekaj, kar vemo (geslo, PIN...)
  Če to izve nekdo drug, se lahko izdaja za nas!
- nekaj, kar imamo (dokumenti, certifikat,...)

Identifikacijska shema je postopek za potrditev identitete.

Cilji identifikacijskih shem

Anita se predstavi Bojanu. Zahteve:

- Priča Anitine predstavitve Bojanu se ne more kasneje lažno predstaviti za Anito,
- tudi Bojan se ne more po Anitini predstavitvi lažno predstaviti za Anito,
- Anito s predstavitvijo ne izda nobene informacije, ki jo identificira/predstavlja.
- Shema je enostavna.

Opazimo: predstavitev ne sme biti vedno enaka! Zato je nujno vnesti naključnost.

Večina identifikacijskih shem vključuje izziv-odgovor.

Protokoli

Interaktivni protokol vključuje dve ali več entitet (osebe, računalniki,...), ki komunicirajo med seboj. Entitete si po nekem predpisanem postopku med seboj izmenjujejo sporočila.

Seja je enkratna izvedba protokola.

Korak je sestavljen iz izračunov in pošiljanja sporočila ene entitete drugi.

Ne-varen protokol z izzivom in odgovorom

Predpostavimo, da Anita in Bojan že delita tajni ključ K za simetrični kriptosistem. Anita se predstavi Bojanu.

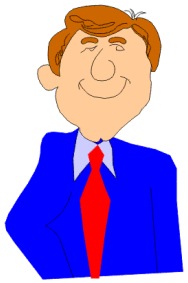
1. Bojan izbere veliko naključno število - izziv R_A in ga pošlje Aniti.
2. Anita izračuna $y = e_K(R_B)$ in ga pošlje Bojanu.
3. Bojan izračuna $y' = e_K(R_B)$ in preveri $y = y'$.

Izziv in odgovor - slika

Jaz sem Anita

R_B

$y=e_K(R_B)$



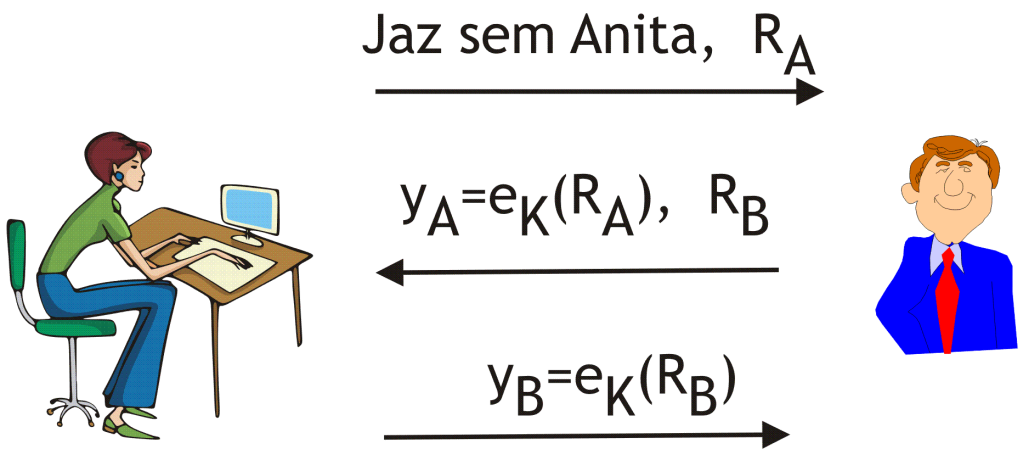
Bojan izračuna $y'=e_K(R_B)$
in preveri $y = y'$

Dvostranska predstavitev - poenostavljen primer

Predpostavimo, da Anita in Bojan že delita tajni ključ K za simetrični kriptosistem. Anita in Bojan se predstavita drug drugemu.

1. Anita izbere veliko naključno število - izziv R_A in ga pošlje Bojanu.
2. Bojan izbere veliko naključno število R_B, izračuna $y_A = e_K(R_A)$ in pošlje Aniti y_A in R_B.
3. Anita izračuna $y'_A = e_K(R_A)$ in preveri $y_A = y'_A$. Anita izračuna $y_B = e_K(R_B)$ in ga pošlje Bojanu.
4. Bojan izračuna $y'_B = e_K(R_B)$ in preveri $y_B = y'_B$.

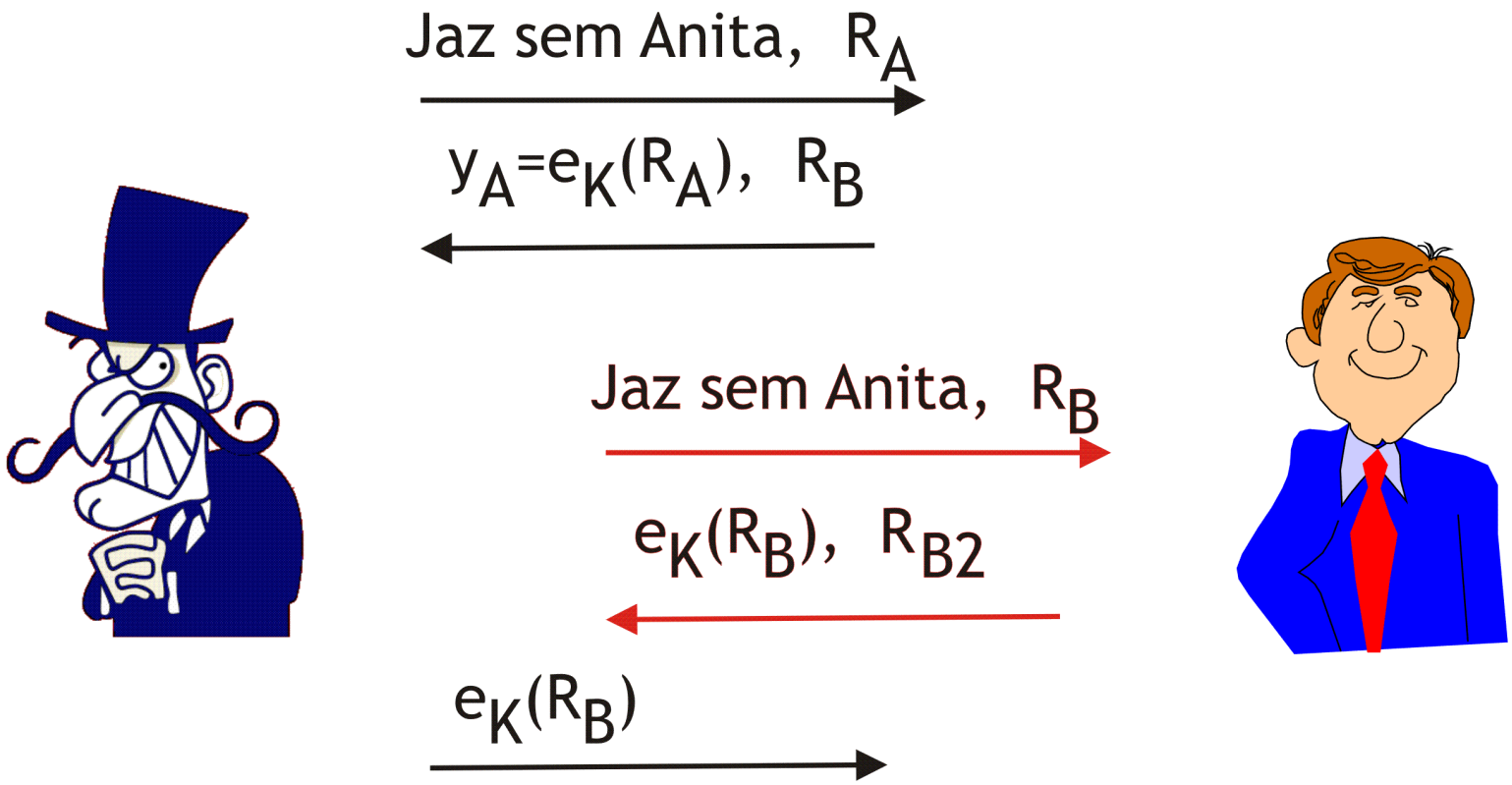
Dvostranska predstavitev - slika



Anita izračuna $y_A'=e_K(R_A)$
in preveri $y_A = y_A'$

Bojan izračuna $y_B'=e_K(R_B)$
in preveri $y_B = y_B'$

Napad z vzporedno sejo



Varen protokol izziv - odgovor

Na vsakem koraku je treba izvesti druge račune, da preprečimo napad z vzporedno sejo!

- Anita pošlje Bojanu R_A,
  Bojan pošlje Aniti $e_K(R_B, R_A)$,
  Anita pošlje Bojanu $e_K(R_A, R_B)$,
- Uporaba zgoščevalne funkcije (oziroma MAC):
  Anita pošlje Bojanu R_A,
  Bojan pošlje Aniti $R_B,\ H(R_B||R_A, ||K||ID(B))$,
  Anita pošlje Bojanu $H(R_A||R_B||K||ID(A))$.

Protokol izziv - odgovor z javnim ključem

1. Anita izbere naključen izziv R_A
   in pošlje Bojanu svoj certifikat $C(A)$ in R_A.

2. Bojan izbere naključen izziv R_B in
   izračuna podpis $y_1 = \mathrm{sig}_B(ID(A)||R_A||R_B)$ in
   pošlje Aniti svoj certifikat $C(B)$, R_B in y_1.

3. Anita preveri Bojanov javni ključ na $C(B)$ in
   preveri Bojanov podpis y_1 za $\mathrm{sig}_B(ID(A)||R_A||R_B)$.
   Če podpis ni veljaven, zavrne avtentikacijo.
   Sicer Anita izračuna podpis $y_2 = \mathrm{sig}_A(ID(B)||R_B)$ in
   pošlje Bojanu y_2.

4. Bojan preveri Anitin javni ključ na $C(A)$ in
   preveri Anitin podpis y_2 za $\mathrm{sig}_A(ID(B)||R_B)$.
   Če podpis ni veljaven, zavrne avtentikacijo.

Uskladitev ključev

Kako doseči, da se A in B dogovorita o skupnem ključu po ne-varnem kanalu za simetričen kriptosistem brez posredovanja TA?

Predpostavimo, da imata vsak svoj certifikat za javnim ključem.

Želimo:

- Noben pošten udeleženec ne sprejme ključa po koraku, v katerega je aktivno posegel napadalec.
- Če ni aktivnega napadalca, na koncu oba udeleženca izračunata isti ključ K. Pasivni napadalec ne izve nobene informacije o K.

Cilji napadalcev

Aktiven napadalec lahko

- prestreženo sporočilo spremeni in pošlje naprej,
- sporočilo shrani in uporabi kasneje,
- se izdaja za drugega uporabnika na omrežju.

Cilji napadalca so lahko naslednji.

- Prepriča A in B, ki komunicirata, da sprejmeta neveljaven ključ (ključ, ki mu je potekel rok, ključ, ki ga je izbral napadalec...)
- Prepriča A in B, da sta se dogovorila za ključ, čeprav se nista,
- Dobi delno informacijo o ključu, za katerega sta se dogovorila A in B.

Diffie-Hellmanova uskladitev ključev

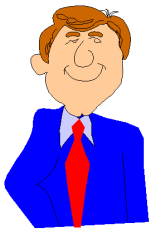
Parametri sistema so grupa G in element grupe α z velikim redom.

Anita izbere a
in izračuna $A=\alpha^a$

Bojan izbere b
in izračuna $B=\alpha^b$

A

B



Anita izračuna

$K=B^a=\alpha^{ba}$

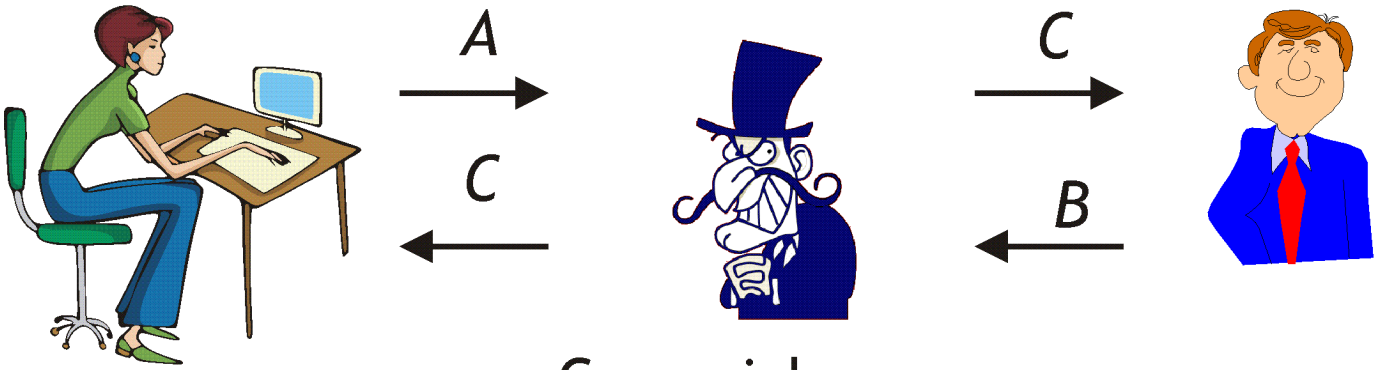
Bojan izračuna

$K=A^b=\alpha^{ba}$

Napad srednjega moža

Anita izbere a
in izračuna $A=\alpha^a$

Bojan izbere b
in izračuna $B=\alpha^b$



Cene izbere c
in izračuna $C=\alpha^c$

Anita deli skupi ključ $K1=\alpha^{ac}$ s Cenetom.
Bojan deli skupi ključ $K2=\alpha^{bc}$ s Cenetom.

Protokol station-to-station

Parametri sistema so grupa G in element grupe α reda n.
Vsak udeleženec U ima svoj certifikat $C(U)$.

1. Anita izbere naključno število $a_A \in \{2 \dots n-1\}$.
   Anita izračuna $b_A = \alpha^{a_A}$ in
   pošlje Bojanu $C(Anita)$ in b_A.

2. Bojan izbere naključno število $a_B \in \{2 \dots n-1\}$.
   Bojan izračuna $b_B = \alpha^{a_B}$ in
   izračuna ključ $K = b_A^{a_B}$.
   Bojan podpiše $y_B = \mathrm{sig}_{Bojan}(ID(Anita)||b_B||b_A)$.
   Bojan pošlje Aniti $C(Bojan)$, b_B in y_B.

3. Anita preveri podpis y_B.
   Če podpis ni veljaven, zavrne sejo in konča, sicer jo sprejme.
   Anita zračuna ključ $K = b_B^{a_A}$.
   Anita podpiše $y_A = \mathrm{sig}_{Anita}(ID(Bojan)||b_A||b_B)$.
   Anita pošlje Bojanu y_A.
4. Bojan preveri podpis y_A.
   Če podpis ni veljaven, sejo zavrne, sicer jo sprejme.

Če v dogovor ni posegel aktiven napadalec, si Anita in Bojan na koncu delita skupen ključ $K = b_A^{a_B} = b_B^{a_A} = \alpha^{a_A a_B}$.

Uporaba certifikatov

Certifikati, oziroma celotna PKI, omogočajo, da lahko naslednji servisi potekajo varno preko spleta.

- bančništvo, e-davki, e-uprava,
- e-Študent, varna domača stran,
- šifrirana pošta,
- varne povezave preko interneta (SSL/TLS),
- e-trgovina.

Secure Sockets Layer (SSL)

- SSL je razvil Netscape.
- TLS (Transport Layer Security) je IETF-ova verzija SSL.
- SSL uporabljamo v brskalnikih za zaščito mrežnih transakcij.
- Osnovni komponenti SSL/TLS sta
  - handshake protocol: overjanje in dogovor o ključu,
  - record protocol: šifriranje in overjanje poslanih podatkov.

Upravljanje z javnimi ključi v SSL/TLS

- Ključ korenske CA (root CA) je vnaprej inštaliran v brskalnik.
- Mrežnim strežnikom certificirajo javne ključe z enim izmed korenskih CA-jev.
- Klienti (uporabniki) lahko pridobijo svoje certifikate.
  Večina uporabnikov trenutno nima svojih lastnih certifikatov.
  Če klienti nimajo svojih certifikatov, potem je overjanje samo enostransko - strežnik se predstavi klientu).

SSL - slika

Uporabnik

Strežnik

Jaz sem Anita →

Jaz sem Bojan d.o.o. ←

PK, $sig_{CA}(PK)$ ←

Anita preveri podpis.

Anita generira naključen MS.

$y = e_{PK}(MS)$ →

Bojan d.o.o izračuna MS.

Iz MS oba izračunata po dva ključa:
K_1 za overjanje sporočil in K_2 za šifriranje.