Informacija in kodi

UN2-1-AV 2024/2025

Tajno kodiranje II

Simon Dobrišek

november 2024

Teme predavanja

Kriptografski sistemi z javnim ključem

Kriptografski postopek RSA

Overjanje sporočil in uporabnikov

Overjanje datuma in časa podpisa sporočila

Kriptografska zgoščevalna funkcija SHA-1

Kriptografski sistemi z javnim ključem

Kriptografski sistemi s tajnim ključem niso primerni za množično komunikacijo, ker zahtevajo za vsak par uporabnikov, ki želi medsebojno komunicirati, svoj ključ, o katerem se morata uporabnika z osebnim ali kakšnim drugačnim zaupnim stikom sporazumeti že pred oddajo sporočil.

Diffie in Hellman sta leta 1976 predlagala kriptografski sistem, ki ne zahteva tajne izmenjave ključev pred oddajo sporočil.

Sistem, ki sta ga predlagala, ne uporablja več enega, temveč dva med seboj odvisna, vendar različna ključa: *ključ za šifriranje* in *ključ za dešifriranje sporočil*.

Predlagani sistem temelji na *navidezno enosmernih funkcijah* in je znan pod imenom *kriptografski sistem z javnim ključem*.

Kriptografski sistemi z javnim ključem

Vsak uporabnik komunikacijskega sistema U, ki želi tajno komunicirati, mora razpolagati z:

a) šifrirnim algoritmom E,

b) dešifrirnim algoritmom D in

c) lastnim parom ključev $(\mathrm{K}_e, \mathrm{K}_d)$.

Ključ K_e javno objavi (zapiše na kakšen javni seznam), ključ K_d pa drži v tajnosti.

Če želi poslati uporabnik U_i tajnopis uporabniku U_j, šifrira svoje sporočilo M tako, da vstavi v šifrirni algoritem E javni šifrirni ključ K_e^j uporabnika U_j. Tako dobi tajnopis

$$C = E(M, \mathrm{K}_e^j),$$

ki ga pošlje po (nezaščitenem) komunikacijskem kanalu uporabniku U_j.

Kriptografski sistemi z javnim ključem

Uporabnik U_j dešifrira sprejti tajnopis C z dešifrirnim algoritmom D in s svojim tajnim ključem K_d^j. Tako odpre sporočilo

$$M = D\Big(E(M, \mathrm{K}_e^j), \mathrm{K}_d^j\Big) = D(C, \mathrm{K}_d^j),$$

ki mu je bilo namenjeno.

Varnost šifrirnega sistema z javnim ključem temelji na naslednjih lastnostih funkcij (algoritmov) E in D:

1. za vsak M mora biti $C = E(M, \mathrm{K}_e)$ preprosto izračunljiv,
2. za vsak E, D, K_e in K_d mora biti računsko nemogoče določiti K_d iz znanih E, D, K_e, C in M,
3. za vsak C mora biti $M = D\Big(E(M, \mathrm{K}_e), \mathrm{K}_d\Big)$ preprosto izračunljiv ter
4. ne sme biti težko "narediti"para ključev $(\mathrm{K}_e, \mathrm{K}_d)$.

Kriptografski sistemi z javnim ključem

Prva in druga lastnost zahtevata, da je šifrirni algoritem z javnim ključem *enosmerna funkcija*, tretja lastnost pa zahteva, da ima enosmerna funkcija še dodatno lastnost, imenujemo jo *past*, ki iz enosmerne funkcije naredi *navidezno enosmerno funkcijo*.

V nadaljevanju bomo podrobno opisali kriptografski sistem z javnim ključem, ki uporablja šifrirni in dešifrirni postopek RSA (Rivest - Shamir - Adleman).

Postopka temeljita na uporabi eksponentne funkcije kot enosmerne funkcije z vgrajeno pastjo ter na težavnosti naloge faktorizacije števila, ki je zmnožek dveh velikih praštevil.

Seznanimo se najprej z najnujnejšimi pojmi iz teorije števil.

O deljivosti celih števil

Celo število u je *deljivo* s celim številom v, če lahko zapišemo število u kot zmnožek števila v s celim številom k:

$$u = kv.$$

Če je v naravno število, lahko zapišemo vsako celo število u kot vsoto nekega večkratnika števila v in nekega nenegativnega celega števila R (ostanek), ki je manjše od v:

$$u = Tv + R;\ 0 \leq R < v.$$

Deljivost števil in največja skupna mera

Primer 6.9

Na primer:

a) za dani števili $u = 104$ in $v = 10$ lahko pišemo
$104 = 10 \cdot 10 + 4$,

b) za dani števili $u = -53$, $v = 10$ pa
$-53 = -6 \cdot 10 + 7$ □.

Vzemimo, da so dana cela števila $u_1, u_2, \ldots, u_n$. Če celo število m deli vsako izmed celih števil $u_1, u_2, \ldots, u_n$, pravimo, da je skupni delitelj števil $u_1, u_2, \ldots, u_n$.

Deljivost števil in največja skupna mera

Med skupnimi delitelji celih števil $u_1, u_2, ..., u_n$ je eden največji in tega imenujemo *največja skupna mera* števil $u_1, u_2, ..., u_n$. Največjo skupno mero števil $u_1, u_2, ..., u_n$ označimo z $NSM(u_1, u_2, ..., u_n)$.

Primer 6.10

Na primer, števili 27 in 18 imata skupne delitelje 1, 3 in 9, zato je $NSM(27, 18) = 9$ □.

Cela števila $u_1, u_2, ..., u_n$ so med seboj *tuja*, če je

$$NSM(u_1, u_2, ..., u_n) = 1.$$

Kongruenca celih števil

Če je razlika $u - v$ celih števil u in v deljiva s celim številom m, to je $u = k \cdot m + v$, kjer je k celo pozitivno število, pravimo, da sta števili u in v *kongruentni* po modulu m, in pišemo:

$$u \equiv v \mod m,$$

ali enakovredno

$$u \mod m = v.$$

Kongruenca $u \equiv 0 \mod m$ pomeni, da je celo število u deljivo s celim številom m

Primer 6.11

Na primer, ker je razlika števil (27-18) deljiva s 3, sta števili 27 in 18 kongruentni po modulu 3, torej je

$$27 \equiv 18 \mod 3.$$ □

Praštevila in sestavljena števila

Praštevilo je od 1 različno naravno število q, ki nima drugih deljiteljev kot 1 in q. Koliko je praštevil med 1 in x, pove funkcija $\Pi(x)$.

Primer 6.12

Na primer, $\Pi(3) = \{2, 3\} = 2$, $\Pi(7) = \{2, 3, 5, 7\} = 4$, $\Pi(10) = \{2, 3, 5, 7\} = 4$ itn. □

Celo število, ki ni praštevilo in ne 1, imenujemo *sestavljeno* in ga lahko zapišemo kot zmnožek vsaj dveh faktorjev, od katerih nobeden ni 1.

Vzemimo, da je u poljubno celo število, q pa praštevilo. Največja skupna mera $NSM(u, q)$ je ali 1 ali q. Praštevilo deli celo število ali pa mu je tuje. Iz tega tudi sledi, da je vsako število, večje od 1, možno razcepiti v zmnožek praštevil.

Eulerjeva funkcija

Če je u pozitivno celo število, je vsako izmed števil $0, 1, 2, ..., u-1$ ali tuje številu u ali pa ima z u kakšen skupni delitelj.

Primer 6.13

Vzemimo $u = 8$. Med števili $0, 1, 2, 3, 4, 5, 6, 7$ imajo števila $0, 2, 4, 6$ skupne delitelje s številom 8, števila $1, 3, 5, 7$ pa so tuja številu 8. □

S $\varphi(u)$ označimo število tistih števil med $0, 1, ..., u-1$, ki so tuja u. Funkciji $\varphi(u)$ pravimo *Eulerjeva funkcija*.

Primer 6.14

Za primer 6.13 je $\varphi(8) = 4$, ker imamo med števili $0, 1, 2, 3, 4, 5, 6, 7$ štiri števila, ki so tuja številu 8. □

Eulerjeva funkcija

Za vsako praštevilo q velja

$$\varphi(q) = q - 1,$$

saj je med števili $0, 1, ..., q-1$ le število 0, ki ni tuje praštevilu q, drugih $q-1$ števil pa je praštevilu q tujih.

Prav tako bi lahko ugotovili, da velja za tuji števili u in v:

$$\varphi(u \cdot v) = \varphi(u) \cdot \varphi(v).$$

Eulerjeva funkcija

Primer 6.15

Vzemimo $u = 7$ in $v = 8$. Ker je 7 praštevilo, je $\varphi(7) = 6$, iz primera 7.16 pa sledi $\varphi(8) = 4$.

$uv = 7 \cdot 8 = 56$. Med števili $0, 1, 2, \ldots, 55$ imajo števila $0, 2, 4, 6, 7, 8, 10, 12, 14, 16, 18, 20, 21, 22, 24, 26, 28, 30, 32, 34, 35, 36, 38, 40, 42, 44, 46, 48, 49, 50, 52, 54$ skupne delitelje s številom 56, vsa ostala pa so številu 56 tuja.

Ker je le-teh 24, je $\varphi(56) = 24$. Vidimo, da je Eulerjeva funkcija multiplikativna, oziroma da je

$$\varphi(56) = \varphi(7 \cdot 8) = \varphi(7) \cdot \varphi(8) = 6 \cdot 4 = 24. \qquad \square$$

Fermatov izrek

Če je celo število u tuje praštevilu q, velja kongruenca

$$u^{q-1} \equiv 1 \mod q. \tag{1}$$

Vsako praštevilu q tuje število, potencirano s $q-1$, da ob deljenju s q ostanek 1.

Primer 6.16

Na primer:

$$1^6 \equiv 1 \mod 7, 2^6 \equiv 1 \mod 7, ..., 6^6 \equiv 1 \mod 7. \qquad \square$$

Eulerjev izrek

Vsakemu celemu številu u, ki je tuje številu m, ustreza kongruenca

$$u^{\varphi(m)} \equiv 1 \mod m. \tag{2}$$

Pri tem je $\varphi(m)$ vrednost Eulerjeve funkcije števila m.

Vidimo, da je v Eulerjevem izreku zajet tudi Fermatov izrek, saj je v primeru, ko je m enak praštevilu q, Eulerjeva funkcija $\varphi(q) = q - 1$, iz kongruence (2) pa dobimo kongruenco (1).

Enosmerne in navidezno enosmerne funkcije

Šifrirni sistemi z javnim ključem temeljijo na uporabi *navideznih* enosmernih funkcij, to je enosmernih funkcij z *vgrajeno pastjo*, zato se bomo najprej seznanili z njimi.

Enosmerne funkcije

V kriptografske sisteme vgrajujemo NP-težke naloge z *enosmernimi* funkcijami[1]. To so funkcije, od katerih zahtevamo, da:

1. za vsako vrednost x na definicijskem območju funkcije f lahko enolično določimo vrednost y,
2. pri dani vrednosti x izračun vrednosti y ni časovno zahteven,
3. je pri dani vrednosti y izračun vrednosti x časovno zahteven.

[1] Funkcija f je enosmerna funkcija, če je pri vsakem argumentu x v zalogi vrednosti f izračun vrednosti $f(x)$ mogoč v polinomskem času. Hkrati mora veljati za vse vrednosti y v zalogi vrednosti f, da za rešitev enačbe $x = f^{-1}(y)$ ne sme obstajati algoritem, ki določi x v polinomskem času.

Enosmerne in navidezno enosmerne funkcije

Primer 6.17

Vzemimo funkcijo

$$f(u, v) = uv, \tag{3}$$

kjer sta u in v poljubni celi števili dolžine k dvojiških znakov.

Funkcija f je enosmerna, ker če poznamo u in v, njun zmnožek izračunamo v polinomskem času. Inverz f^{-1} izračunamo s faktorizacijo števila uv dolžine $2k$. To pa je naloga, za katero je znano, da je ne moremo rešiti v polinomskem času.

Vzemimo sedaj v funkciji (3) namesto u in v praštevili q in q' dolžine k dvojiških znakov. Izračun inverza funkcije je sedaj časovno še zahtevnejša naloga. Je takšna, da jo uvrščamo med NP-težke naloge. □

Navidezno enosmerne funkcije

Te funkcije imajo naslednje lastnosti:

1. $f^{-1}(f(x)) = x$,
2. za izračun f in f^{-1} obstajajo učinkoviti algoritmi,
3. pri znanem f je računsko nemogoče izračunati f^{-1}, če ne poznamo "ključa" za odprtje enosmerne pasti.

Vidimo, da to v resnici niso prave enosmerne funkcije, saj je mogoče preprosto izračunati inverz, to je f^{-1}, vendar le v primeru, da vemo, kako odpreti enosmerno past.

Navidezno enosmerne funkcije - primer

Primer 6.18

Oglejmo si eksponentno funkcijo

$$y = a^x \mod q, \tag{4}$$

kjer je q praštevilo, a pa primitivni element[2] v obsegu $\mathrm{GO}(q)$.

Številu (inverzu funkcije (4))

$$x = \log_a y \mod q \qquad 1 \leq x \leq q-1 \tag{5}$$

pravimo *diskretni logaritem* števila y po modulu q pri osnovi a.

[2] Primitivni elementi so tisti elementi $\alpha \in \mathrm{GO}(q)$, za katere velja $\alpha^{q-1} \equiv 1 \mod q$. Če je na primer $q = 11$, so primitivni elementi 2,6,7,8.

Navidezno enosmerne funkcije - primer

Opomba: Za dana a in y v (5) je x enolično definiran le, če je a primitivni element v obsegu $\mathrm{GO}(q)$.

Če je $q \leq 2^r$, lahko vse nastopajoče količine predstavimo kot števila dolžine r dvojiških znakov.
Izračun enačbe (4) v $\mathrm{GO}(q)$ zahteva največ $2r$ operacij, medtem ko zahteva enačba (5) $2^{r/2}$ operacij.

Na primer, za $r = 200$ je potrebnih za izračun (4) največ 400 operacij, za izračun (5) pa $2^{100} \approx 10^{30}$ operacij.

Eksponentna funkcija torej predstavlja enosmerno funkcijo, ker je izračun inverza težaven.

Navidezno enosmerne funkcije - primer

Vendar pa ima eksponentna funkcija tudi naslednjo lastnost

$$(a^x)^z = a^{xz},$$

ki v primeru, da sta si x in z inverzna po modulu q, to je $xz \equiv 1 \mod q$, da

$$(a^x)^z = (a^z)^x = a^{xz} \equiv a \mod q. \tag{6}$$

Iz enačbe (6) vidimo, kako nam z pomaga, da lažje izračunamo inverz od a^x. Namesto, da računamo celo število x po enačbi (5), ga iščemo s postopnim potenciranjem a^z, dokler ni $(a^z)^x \equiv a \mod q$.

Celo število z predstavlja "ključ" za odprtje *enosmerne pasti*, eksponentna funkcija pa se zato izkaže kot navidezno enosmerna funkcija. □

Kriptografski postopek RSA

Varnost kriptografskega postopka RSA (Rivest - Shamir - Adleman), ki ga vgrajujemo v kriptografske sisteme z javnim ključem, temelji na zahtevnosti naloge faktorizacije števila, ki je zmnožek dveh velikih praštevil, denimo q in q'.

Uporabnik U_i, ki želi tajno komunicirati z uporabnikom U_j, izbere dve taki števili in izračuna njun zmnožek $n = qq'$ ter Eulerjevo funkcijo

$$\varphi(n) = (q-1)(q'-1).$$

Nato naključno izbere število d tako, da velja

$$NSM(d, \varphi(n)) = 1.$$

Števili d in n tvorita njegov tajni ključ K_d.

Kriptografski postopek RSA

Uporabnik U_i izračuna še število e iz razmika $1 < e \leq \varphi(n)$ tako, da velja

$$e \cdot d \equiv 1 \mod \varphi(n), \tag{7}$$

kar lahko pišemo tudi kot

$$e \cdot d = k\varphi(n) + 1, \tag{8}$$

kjer je k celo število.

Števili e in n tvorita njegov javni ključ K_e.

Kriptografski postopek RSA

Uporabnik U_j zapiše sporočilo M, ki ga želi posredovati uporabniku U_i, z "bloki- nizi nenegativnih celih števil $m_i < n;\ i = 1, 2, \ldots$ in ga šifrira po obrazcu

$$E:\ c_i = m_i^e \mod n;\ i = 1, 2, \ldots \tag{9}$$

Omejitev $m_i < n$ zagotavlja bijektivnost preslikave E. Uporabnik U_i

dešifrira tajnopis, ki ga je sprejel od uporabnika U_j, po obrazcu

$$D:\ m_i = c_i^d \mod n;\ i = 1, 2, \ldots \tag{10}$$

Kriptografski postopek RSA

IZREK: Za poljubno nenegativno celo število, manjše od n, pravimo mu sporočilo in ga označimo z M, sta šifriranje E in dešifriranje D inverzni transformaciji, če sta e in d izbrana tako, da zadoščata enačbi (7) oziroma (8).

Postopke kriptografskega sistema RSA (generiranje ključev, šifriranje in dešifriranje sporočil) udejanjimo s tremi algoritmi:

- algoritem za generiranje psevdonaključnih števil,
- Evklidov algoritem za iskanje *NSM* dveh števil in
- algoritem za generiranje velikih praštevil.

Kriptografski postopek RSA - primer

Primer 6.29

Vzemimo, da nam nekdo želi poslati šifrirano sporočilo. V ta namen moramo generirati svoj javni in tajni ključ[3].

Vzemimo dve majhni praštevili $q = 2$ in $q' = 5$. Izračunamo modul

$$n = qq' = 2 \cdot 5 = 10.$$

Število d (prvo število tajnega dešifrirnega ključa) mora biti tuje številu $(q-1)(q'-1) = 1 \cdot 4 = 4$ in večje kot $\max\{q+1, q'+1\} = 6$. Naj bo

$$d = 7.$$

[3] Velika praštevila, kot to zahteva sistem RSA, poiščemo z algoritmom za generiranje velikih praštevil.

Kriptografski postopek RSA - primer

Število e, kjer je $1 < e \leq 4$, mora zadoščati pogoju

$$e \cdot 7 \equiv 1 \mod 4.$$

Dobimo

$$e = 3.$$

(Pri velikih številih d in $(q-1)(q'-1)$ poiščemo e s pomočjo Evklidovega algoritma.)

Osebi, ki nam želi poslati šifrirano sporočilo, sporočimo svoj javni ključ $(e = 3, n = 10)$.

Ta šifrira svoje sporočilo po postopku RSA s številoma $e = 3$ in $n = 10$.

Kriptografski postopek RSA - primer

Vzemimo, da nam želi poslati sporočilo *JAN JE LEP*.

ASCII kode sporočila zapiše v desetiškem sistemu. Dobi

$$74\ 65\ 78\ 32\ 74\ 69\ 32\ 76\ 69\ 80.$$

Zakodirano sporočilo

$$7465783274693276698 0$$

razdeli v bloke dolžine $k = 1$. (Na koncu zakodiranega sporočila po potrebi doda ničle.)

Dobi niz enot sporočila m_i $(m_i < 10)$:

$$7\ 4\ 6\ 5\ 7\ 8\ 3\ 2\ 7\ 4\ 6\ 9\ 3\ 2\ 7\ 6\ 6\ 9\ 8\ 0$$

Kriptografski postopek RSA - primer

Enote sporočila m_i $(i = 1, ..., 20)$ potencira s potenco $e = 3$ po modulu $n = 10$:

$$c_i = m_i^3 \mod 10.$$

(Pri tem uporabi hitri algoritem za potenciranje.)
Dobi enote tajnopisa c_i $(i = 1, ..., 20)$

$$3\,4\,6\,5\,3\,2\,7\,8\,3\,4\,6\,9\,7\,8\,3\,6\,6\,9\,2\,0.$$

Po kanalu nam pošlje tajnopis C, to je

$$34653278346978366920$$

Sprejeti tajnopis dešifriramo s pomočjo dešifrirnega ključa $(7, 10)$.

Kriptografski postopek RSA - primer

Tajnopis C razdelimo na bloke takšne dolžine, da za vsak i velja $c_i < n$.

V našem primeru razbijemo tajnopis v bloke dolžine 1. Dobimo

$$3\,4\,6\,5\,3\,2\,7\,8\,3\,4\,6\,9\,7\,8\,3\,6\,6\,9\,2\,0.$$

Vsak blok tajnopisa c_i pretvorimo v bloke sporočila m_i, tako da izračunamo

$$m_i = c_i^7 \mod 10.$$

Dobimo:

$$7\,4\,6\,5\,7\,8\,3\,2\,7\,4\,6\,9\,3\,2\,7\,6\,6\,9\,8\,0.$$

Kriptografski postopek RSA - primer

Bloke sporočila m_i, zakodirane z desetiškimi števili, to je

$$7465783274693276698 0,$$

dekodiramo tako, da najprej razsekamo niz znakov v bloke po dva desetiška znaka,

$$74\ 65\ 78\ 32\ 74\ 69\ 32\ 76\ 69\ 78,$$

in končno z uporabo tabele kodov ASCII obnovimo izvirno sporočilo *JAN JE LEP*. □

Kriptografski postopek RSA

Kljub hitremu algoritmu potenciranja po modulu, ki zahteva toliko ponovitev postopka, kolikor mest zasede dvojiški zapis eksponenta (števili e in d), sta postopka šifriranja in dešifriranja RSA razmeroma počasna.

Velja ocena, da ponuja kriptografski sistem RSA z modulom računanja $n = qq'$ dolžine 3072 dvojiških znakov (≈ 926 desetiških števk) približno enako odpornost na napad kot kriptografski sistem AES s ključi dolžine 128 dvojiških znakov.

Ravno tako velja ocena, da sta postopka šifriranja in dešifriranja RSA več kot 1000-krat počasnejša kot primerljiva postopka šifriranja in dešifriranja AES.

Kriptografski postopek RSA

Zato je kriptografski sistem RSA, kakor tudi drugi kriptografski sistemi z javnim ključem, primeren le za zakrivanje krajših sporočil, na primer ključev za šifriranje s kriptografskimi sistemi s tajnim ključem (na primer z AES).

Oglejmo si primer pošiljanja daljših tajnopisov med krajevno oddaljenimi uporabniki telekomunikacijskega omrežja.

Uporabili bomo kriptografska sistema RSA in AES.

Šifriranje sporočil z javnim ključem

Začetek postopka

1. korak Pošiljatelj U_i izpostavi sporočilo M šifrirnemu postopku E_{AES},

$$C_M = E_{\mathrm{AES}}(M, \mathrm{K}_{\mathrm{AES}}),$$

in pri tem uporabi (tajni) šifrirni ključ $\mathrm{K}_{\mathrm{AES}}$.

2. korak Pošiljatelj U_i izpostavi šifrirni ključ $\mathrm{K}_{\mathrm{AES}}$ šifrirnemu postopku E_{RSA},

$$C_{\mathrm{K}} = E_{\mathrm{RSA}}(\mathrm{K}_{\mathrm{AES}}, \mathrm{K}_e^j),$$

in pri tem uporabi javni šifrirni ključ kriptografskega sistema RSA naslovnika U_j.

Pošiljatelj U_i pošlje tajnopisa $(C_M,\ C_{\mathrm{K}})$ po (nezaščitenem) kanalu do naslovnika U_j.

Šifriranje sporočil z javnim ključem

3. korak Naslovnik U_j izvede dešifrirni postopek kriptografskega sistema RSA na tajnopisu C_{K} z uporabo svojega tajnega ključa kriptografskega sistema RSA:

$$D_{\mathrm{RSA}}(C_{\mathrm{K}}, \mathrm{K}_d^j) = \mathrm{K}_{\mathrm{AES}}.$$

Tako dobi ključ za dešifriranje tajnopisa C_M.

4. korak Naslovnik U_j izvede dešifrirni postopek kriptografskega sistema AES na tajnopisu C_M z uporabo ključa kriptografskega sistema AES:

$$D_{\mathrm{AES}}(C_M, \mathrm{K}_{\mathrm{AES}}) = M.$$

Tako dobi sporočilo M.

Konec postopka

Overjanje sporočil in uporabnikov

Pri izmenjavi in shranjevanju informacije v elektronski obliki ni pomembna le tajnost sporočil, temveč tudi:

- *Zaščita informacije pred nepooblaščenim spreminjanjem*, kar pomeni, da je potrebno onemogočti spreminjanje delov sporočila brez vednosti pošiljatelja (podpisnika elektronskega dokumenta).

  V bančnem poslovanju je na primer možnost preprečitve nepooblaščenega spreminjanja zneskov denarnih nakazil pogosto pomembnejša kakor njihova tajnost.

Overjanje sporočil in uporabnikov

- *Zagotavljanje možnosti overjanja vira informacije* z namenom, da prepriča prejemnika sporočila, da je vir informacije dejansko tisti, za katerega se izdaja.

  Pri poslovnem dopisovanju je na primer vsak dopis opremljen z naslovom pošiljatelja, njegovim podpisom in odtisom pečata, ki prejemniku dopisa potrjujejo pristnost pošiljatelja.

- *Preprečevanje morebitne utaje avtorstva informacije*. Namen preprečitve utaje avtorstva informacije je, da razreši morebitno razpravo med poslovnima strankama z *dokazilom*, ki lahko prepriča tretjo osebo (na primer sodišče), da je bila dotična informacija dejansko ustvarjena in odposlana s strani ene izmed strank.

Overjanje sporočil in uporabnikov

Naštete naloge lahko rešimo z uporabo:

- kriptografskih zgoščevalnih funkcij (na primer z zgoščevalno funkcijo SHA-1),
- kriptografskim sistemom z javnim ključem (na primer s kriptografskim sistemom RSA) in
- vzpostavljene mreže *"overiteljev javnih ključev'"* CA (angl. *Certification Authority*)[4].

SHA-1 (Secure Hash Algorithm 1) je algoritem, ki preslika poljuben niz dolžine do $2^{64} - 1$ dvojiških znakov ($\approx 2,3 \cdot 10^{18}$ ASCII znakov) v niz dolžine 160 dvojiških znakov (20 ASCII znakov).

Rezultatu zgoščevanja P pravimo *povzetek* sporočila.

[4] *RSA Cryptography Standard* PKCS#1 v2.1,RSA Laboratories, 2002.

Overjanje sporočil in uporabnikov

Začetek postopka

Overjanje vsebine in podpisnika sporočila

1. korak Pošiljatelj U_i izpostavi sporočilo$M$ postopku zgoščevanja z javno kriptografsko zgoščevalno funkcijo SHA-1:

$$P = SHA(M).$$

2. korak Pošiljatelj U_i izpostavi povzetek sporočila P dešifrirnemu postopku D_{RSA} in pri tem uporabi svoj tajni dešifrirni ključ K_d^i:

$$S = D_{\mathrm{RSA}}(P, \mathrm{K}_d^i).$$

Rezultatu dešifriranja S pravimo *digitalni podpis* (povzetka) sporočila.

Overjanje sporočil in uporabnikov

Pošiljatelj U_i posreduje naslovniku U_j: (1) sporočilo M, (2) digitalni podpis sporočila S in (3) digitalno potrdilo svojega javnega ključa PKC (angl. *Public-Key Certificate*).

Digitalno potrdilo javnega ključa vsebuje javni ključ in osebne podatke o njegovem imetniku, ki ju digitalno podpiše ustanova, ki ji lahko zaupamo (na primer ustanova za overjanje javnih ključev pri Ministrstvu za javno upravo Republike Slovenije SIGEN-CA[5]).

[5] Slovenian GENeral - Certification Authority

Overjanje sporočil in uporabnikov

Preverjanje vsebine in podpisnika sporočila

3. korak Naslovnik U_j izpostavi digitalni podpis S šifrirnemu postopku z uporabo javnega šifrirnega ključa K_e^i pošiljatelja U_i

$$E_{\mathrm{RSA}}(S, \mathrm{K}_e^i) = E_{\mathrm{RSA}}\left(D_{\mathrm{RSA}}(P, \mathrm{K}_d^i), \mathrm{K}_e^i\right) = P.$$

Tako dobi povzetek sporočila P.

4. korak Naslovnik U_j izpostavi sporočilo M postopku zgoščevanja z javno kriptografsko zgoščevalno funkcijo SHA-1:

$$SHA(M) = P'.$$

Tako dobi povzetek sporočila P'.

5. korak Če je $P = P'$, sta pošiljatelj in sporočilo uspešno preverjena. Avtor sporočila je pošiljatelj U_i in sporočilo, po podpisu, ni bilo spreminjano.

Konec postopka

Overjanje sporočil in uporabnikov

Za primer spora, v katerem bi pošiljatelj U_i zanikal vsebino ali avtorstvo sporočila M, mora naslovnik U_j shraniti ne samo sporočilo temveč tudi digitalni podpis sporočila in digitalno potrdilo javnega ključa pošiljatelja U_i.

S trojko (M, S, PKC) bo lahko dokazal tretji osebi celovitost in avtorstvo (spornega) sporočila M.

Celovitost sporočila bo dokazal z ujemanjem povzetkov P in P', avtorstvo pa s podpisom S, ki ga ni mogel ustvariti nihče drug kot lastnik javnega ključa K_e^i, to je pošiljatelj U_i.

Overjanje datuma in časa podpisa sporočila

Uradni dopisi navadno vsebujejo datum podpisa dopisa, ki se posreduje naslovniku.

Včasih pa je v interesu podpisnika dopisa, da pripiše starejši datum namesto pravega, na primer, ko dopis pošilja z zamudo.

V primerih, ko je pomembno, da je datum nastanka dopisa pravi, ga lahko overimo s postopkom overjanje datuma in časa podpisa sporočila.

Overjanje datuma in časa podpisa sporočila

Začetek postopka

Overjanje datuma in časa podpisa sporočila

1. korak Pošiljatelj U_i izpostavi sporočilo M postopku zgoščevanja s SHA-1:

$$P = SHA(M),$$

2. korak Pošiljatelj U_i posreduje povzetek sporočila P (državni) ustanovi za *"časovno žigosanje elektronskih dokumentov"* TSA[6] (angl: *Timestamping Authority*), ki spne povzetek sporočila P in uradni časovni žig T ter izpostavi niz $P \| T$ postopku zgoščevanja s SHA-1:

$$P' = SHA(P \| T),$$

kjer je P' povzetek *šporočila"* $P \| T$.

[6] SI-TSA na Ministrstvu za javno upravo RS.

Overjanje datuma in časa podpisa sporočila

3. korak TSA izpostavi povzetek P' dešifrirnemu postopku D_{RSA}:

$$S' = D_{\mathrm{RSA}}(P', \mathrm{K}_d^{\mathrm{TSA}})$$

in pri tem uporabi svoj tajni dešifrirni ključ $\mathrm{K}_d^{\mathrm{TSA}}$. TSA posreduje pošiljatelju U_i: (1) digitalni podpis S' spetja povzetka sporočila P in časovnega žiga T ter (2) časovni žig T.

Pošiljatelj U_i posreduje naslovniku U_j: (1) sporočilo M, (2) digitalni podpis S' in (3) časovni žig T.

Overjanje datuma in časa podpisa sporočila

Preverjanje datuma in časa podpisa sporočila

4. korak Naslovnik U_j izpostavi digitalni podpis S' šifrirnemu postopku z uporabo javnega šifrirnega ključa $\mathrm{K}_e^{\mathrm{TSA}}$ ustanove TSA:

$$E_{\mathrm{RSA}}(S', \mathrm{K}_e^{\mathrm{TSA}}) = E_{\mathrm{RSA}}\left(D_{\mathrm{RSA}}(P', \mathrm{K}_d^{\mathrm{TSA}}), \mathrm{K}_e^{\mathrm{TSA}}\right) = P'.$$

Tako dobi povzetek sporočila, ki mu je pripet časovni žig, P'.

5. korak Naslovnik U_j izpostavi sporočilo M postopku zgoščevanja s SHA-1:

$$SHA(M) = P.$$

Tako dobi povzetek sporočila P.

Overjanje datuma in časa podpisa sporočila

6. korak Naslovnik U_j spne povzetek P in časovni žig T in izpostavi niz $P \| T$ postopku zgoščevanja s SHA-1:

$$SHA(P \| T) = P''.$$

Tako dobi povzetek P''.

7. korak Če je $P' = P''$, sta datum in čas podpisa sporočila uspešno preverjena. Časovni trenutek, ki je pripet sporočilu, je resnični datum in čas podpisa sporočila.

Konec postopka

Overjanje datuma in časa podpisa sporočila

Če kombiniramo opisana postopka overjanja sporočil, lahko dosežemo hkratno overovitev vsebine, podpisnika ter datuma in časa podpisa sporočila.

Z uporabo postopka za zakrivanje sporočil, ki smo ga opisali, pa lahko zagotovimo tudi tajnost overjenih sporočil.

Kriptografska zgoščevalna funkcija SHA-1

Ameriški inštitut za standardizacijo NIST (National Institute of Standards and Technology) je vključil SHA-1 med standardizirane kriptografske zgoščevalne funkcije (Secure Hash Standard (SHS))

Poleg SHA-1, ki se trenutno najbolj uporablja, SHS vsebuje še naslednje zgoščevalne funkcije: SHA-256, SHA-384 in SHA-512, ki ustvarijo povzetke dolžin 256, 384 in 512 dvojiških znakov.

Zgostitev niza dolžine do $2^{64} - 1$ dvojiških znakov, to je *sporočila* M, v niz dolžine 160 dvojiških znakov, ki pravimo *povzetek* P, opravimo korakih, ki so opisani v učbeniku na strani 141.

Sporočilu najprej dodamo dvojiške znake, da je njegova dolžina večkratnik 512 dvojiških znakov, in nato z iterativno funkcijo obdelujemo bloke po 512 dvojiških znakov.

Vprašanja

- Opišite kriptografskim sistem z javnim ključem!
- Kaj je navidezno enosmerna funkcija?
- Opišite kriptografski postopke RSA!
- Kako se izvede overjanje sporočil in uporabnikov?
- Kako se izvede overjanje datuma in časa podpisa sporočila?