Zaključaj podatke javnim ključem

DRAGAN PLESKONJIĆ, dipl.ing.

Ovaj tekst predstavlja logički nastavak teksta iz prošlog broja Mog mikra pod naslovom »Zaštita tajnih podataka i programa«. Članak se bavi problemom šifrovanja podataka odnosno poruka koji se šalju na daljinu nepoverljivim kanalom. Nepoverljivim kanalima možemo danas praktično smatrati sve osim ličnog prenosa poruke. Razrađuje se pristup prenosu podataka korištenjem šifrovanja prema tkzv. kriptosistemima sa javnim ključevima (public key systems).

Zamislite da trebate rešiti sledeći problem:

Morate poslati kofer u kom su dragocene stvari (ili poruka, podaci, programi), svom prijatelju (prijateljici) koji živi u udaljenom gradu. Niko ne sme znati sadržaj. Imate na raspolaganju kurira ili prenosni put kome previše ne verujete. Naime, ako kurir sazna šta je u koferu (poruci) svima će izbrbljati. Vi to, naravno, ne smete dozvoliti. Šta ćete učiniti? Možete zaključati kofer. Ali kako da ga primalac otključa kad je ključ kod vas? Ne smete ključ poslati istim kurirom jer mu ne verujete. Znači, treba vam i drugi kurir koji će nositi ključ. A šta ako se kuriri usput nađu i pogledaju kofer? Ne dolazi u obzir, a još je i skupo.

Šta je rešenje? Možda:

1. Kurir prvo odnese kofer (razume se, zaključan), preda ga, a onda se vrati po ključ.

2. Zaključate kofer, kurir ga odnese. Primalac zaključa kofer svojim katancem. Onda kurir vrati kofer vama. Otključate svoj katanac (kofer je sada zaključan katancem primaoca). Kurir odnese kofer primaocu, on ga otključa i uzme ono što je za njega.

Malo komplikovano, zar ne? I skupo istovremeno.

A šta ako postoji brava koja se jednim ključem zaključava, a drugim otključava? Neuobičajeno, ključ koji je bravu zaključao, ne može je i otključati. Ta dva ključa su inverzna. Dakle, na bravu primenite transformaciju T_1 koja je iz stanja otključano (S_o) prevodi u stanje zaključano (S_z). Ovo se obavlja uz pomoć ključa K_1. Da biste bravu otključali koristite transformaciju T_2 uz pomoć ključa K_2. Svaki od ključeva (K_1,K_2) poništava transformaciju koju izvede onaj drugi. Dakle važi (u pojednostavljenom zapisu):

$T_2 = (T_1)$ tj. transformacije T_1 i T_2 su međusobno inverzne

$K_1 * K_2 = 1$ tj. ključevi K i K su međusobno inverzni (znak * ne označava prosto množenje)

$T_1 (S_o,K_1) = S_z$ tj. T_1 prevodi poruku iz stanja S_o u stanje S_z uz pomoć ključa K_1

$T_2 (S_z,K_2) = S_o$ tj. T_2 prevodi poruku iz stanja S_z u stanje S_o uz pomoć ključa K_2.

Uzastopna primena transformacija T_1 i T_2 daje identitet, tj. ostavlja poruku u stanju u kom je prethodno bila. Dakle, onaj ko želi da mu se dragocene poruke šalju i ostanu tajne kod prenosa, treba proizvesti dva inverzna ključa. Jedan od ta dva ključa pošalje onima koji mu poruke šalju i objavi:

»Sve što mi šaljete zaključajte (šifrujte) ovim ključem (K_1).«

Niko, naravno, ne može otključati (dešifrovati) poruku, jer niko nema ključ koji za to služi (K_2). To može učiniti samo onaj ko je generisao par ključeva K_1 i K_2, pod uslovom da se ne nađe neka oštroumna glava koja će na osnovu javnog ključa K_1 izračunati i tajni K_2 (tj. inverzni). Dakle, tajni ključ mora biti jednosmerna funkcija javnog ključa, tj. ne sme se moći, u raspoloživom vremenu i raspoloživim sredstvima, izračunati tajni ključ na osnovu poznatog javnog. Tada se ne može izvršiti ni transformacija šifrovane (zaključane) poruke u dešifrovanu (otvorenu), tj. šifrovani tekst je takođe jednosmerna funkcija otvorenog teksta.

Proširimo ovaj problem:

Recimo da postoje dvije osobe, A i B, koje žele međusobno slati poruke. Nisu baš poverljivi prema drugima. Žele biti sigurni u sledeće:

1. Onaj ko šalje poruku želi biti siguran da niko drugi neće pročitati poruku osim onoga kome je ona namenjena.

2. Onaj ko prima poruku želi biti siguran od koga ona stvarno potiče, tj. želi biti siguran da mu možda neko ne šalje poruku u tuđe ime (lažno predstavljanje).

Šta učiniti? Rešenje je u sledećem:

Osobe A i B generišu po par inverznih ključeva. Tako osoba A generiše dva inverzna ključa Ka i Ka', pri čemu je Ka javni, a Ka' tajni ključ. Na sličan način osoba B generiše ključeve Kb i Kb'.

Sada A i B javno obznane ključeve Ka i Kb, dok Ka' i Kb' čuvaju u tajnosti. Pretpostavimo da niko nije u stanju iz javnih ključeva izvesti tajne (kasnije će biti objašnjeno kako se takvi ključevi stvarno generišu i koriste). Kako osoba A šalje poruku osobi B? Prvo poruku zaključa, tj. transformiše korištenjem svog tajnog ključa Ka' (postupak ove transformacije biće takođe objašnjen kasnije). Zatim tako zaključanu poruku zaključa još jednom, i to javnim ključem osobe B, tj. sa Kb koji je svima poznat. I takvu poruku može bez brige poslati na bilo koji način osobi B. Niko je neće moći dešifrovati osim osobe B.

Kako osoba B dešifruje primljenu poruku? Vrlo jednostavno. Prvo primeni javni ključ osobe A, tj. Ka. Time »poništi« efekat koji je osoba A proizvela svojim tajnim ključem Ka' (ovo može učiniti bilo ko). Uspešan ishod ove operacije potvrđuje osobi B da je stvarni izvor poruke osoba A, jer samo ona (osoba A) zna svoj tajni ključ. Poruka još nije dešifrovana. Sledeći korak koji preduzima osoba B je transformacija uz pomoć ključa Kb', tj. svog tajnog ključa (ovo može učiniti jedino osoba B jer jedino ona zna svoj tajni ključ). Time je osoba B došla do otvorenog (dešifrovanog) oblika poruke koju joj je poslala osoba A.

Na analogan način osoba B moći će slati poruke osobi A. Ovo se može proširiti na još više osoba koje žele na sličan način komunicirati. Znači, svaka osoba koja želi učestvovati, određuje par ključeva (svoj javni i svoj tajni ključ). Javni ključ obznani i zahteva da svi koji mu šalju poruke obave šifrovanje uz pomoć njenog javnog ključa. Ukoliko još želi i proveru verodostojnosti (autentičnosti) poruke, onda zahteva da oni koji šalju poruku vrše šifrovanje svojim tajnim ključem, kao što je objašnjeno.

Ovakvi sistemi se u kriptografiji nazivaju kriptosistemi sa javnim ključevima (public key systems). Problemi koje treba rešiti su sledeći:

– određivanje para ključeva (javnog i tajnog) tako da se iz javnog ključa ne može izvesti tajni

– određivanje transformacije (funkcije preslikavanja) poruke (otvorenog teksta, programa, podataka) u šifrovani tako da se ne može izvršiti inverzija bez poznavanja tajnog ključa.

Znači, tajni ključ je jednosmerna funkcija javnog ključa i šifrovani tekst je jednosmerna funkcija otvorenog teksta. Mehanizam koji ove probleme rešava, iza sebe ima dosta složen matematički aparat koji će ovde biti kratko prikazan. Cilj je da se omogući razumevanje postupka određivanja ključeva prema zahtevima koji se postavljaju i da se objasni princip šifrovanja i dešifrovanja.

Šta je to jednosmerna funkcija

Jednosmerne funkcije imaju sledeću karakteristiku: za datu vrednost X lako je izračunati vrednost funkcije Y=F(X), ali ako je dato Y=F(X), onda ne postoji lak put za izračunavanje X. Drugim rečima, ne postoji izvodljiv računski metod za određivanje $F^{-1}(Y)$. Za kontinualne i analitičke funkcije lako je izvršiti numeričku inverziju. Zato se kao jednosmerne funkcije koriste diskontinualne i proizvoljne funkcije.

Pojam jednosmernam nije apsolutan, već zavisi od toga za koliko mnogo izračunavanja kažemo da je »nemoguće velik broj«. Kaže se da je, za datu vrijednost funkcije F(X), izračunavanje vrijednosti za X »računskim putem neizvodljivo« ako ovo izračunavanje zahteva onoliko vremena i sredstava koliko se ne može izdvojiti. Takođe se zna da sa vrlo srećnim pogađanjem ili biranjem vrednosti funkcije F(X) kojima odgovara poznato X, ovo izračunavanje postaje nepotrebno.

Jedan sistem sa javnim ključem napravljen je tako da koristi očevidno težak problem računanja logaritama u aritmetici po modulu q (q je prost broj i polje GF(q) ima q elemenata (0, 1, ..., q−1; GF je oznaka za Galo-ovo polje od q elemenata).

Neka je:

$$Y = a^x \bmod q \quad 1 \leq X \leq q-1 \qquad (1)$$

Ovde se sva računanja obavljaju u aritmetici po modulu q (npr. 5^3 mod 11 = 125 mod 11 = 4). Pri tome je u gornjoj formuli *a* fiksirani primitivni element polja GF(q) (gde su stepeni broja *a* nenula elementi 1, 2, ..., q−1 od GF(q)). Tada se X računa kao logaritam od Y u bazi *a* preko GF(q):

$$X = \log_a Y \text{ preko } GF(q) \text{ za } 1 \leq Y \leq q-1 \qquad (2)$$

Računanje Y iz X je lako i zahtjeva najviše $2*\log_2$ množenja. Na primer:

$$a^{18} = (((a^2)^2)^2)^2 \quad * \quad a^2 \qquad (3)$$

Računanje X iz Y je mnogo teže i za određene pažljivo izabrane vrednosti q zahteva operacija reda $q^{1/2}$.

Svaki korisnik generira jedan nezavisan slučajni broj izabran iz intervala (1, 2, ..., q−1). On drži X_i u tajnosti, ali $Y_i = a^{Xi} \bmod q$ **(4)**b

i objavljuje ga kao javni ključ. Kada korisnici i i j žele komunicirati tajno, oni kao svoj ključ koriste:

$$K_{ij} = a^{XiXj} \bmod q \qquad (5)$$

Korisnik i računa K_{ij} iz Y_j na sledeći način:

$$K_{ij} = Y_j^{Xi} \bmod q \qquad (6)$$

$$= (a^{Xj})^{Xi} \bmod q \qquad (7)$$

$$= a^{XjXi} \bmod q = a^{XiXj} \bmod q \qquad (8)$$

Korisnik j dobija K_{ij} na jednostavan način:

$$K_{ij} = Y_i^{Xj} \bmod q \qquad (9)$$

Drugi korisnik može izračunati K_{ij} iz Y_i i Y_j, na primer, računanjem:

$$K_{ij} = Y_i^{(\log_a Y_j)} \bmod q \qquad (10)$$

Dakle, ako su logaritmi preko GF(q) (u aritmetici po modulu q) lako izračunljivi, sistem će biti provaljen. Ako nema načina da se izračuna K_{ij} iz Y_i i Y_j, a da se prvo ne dobije X_i ili X_j, sistem je siguran.

Ako je q prosti broj mnogo manji od 2^b, sve veličine se mogu predstaviti kao b−bitni brojevi. Stepenovanje onda zahteva najviše 2b množenja u aritmetici po modulu q, dok logaritmiranje zahteva $q^{1/2} = 2^{b/2}$ operacija, koristeći najbolji poznati algoritam. Ako b=200, potrebno je najviše 400 množenja da se izračuna Y_i iz X_i ili K_{ij} iz Y_i i X_j, iz Y i X, međutim, izračunavanje K_{ij} iz Y_i i Y_j zahteva najmanje 2^{100} ili oko 10^{30} operacija.

Opis RSA algoritma sa javnim ključem

Eksponencijalna funkcija poslužila je na poseban način **R**ivestu, **S**hamiru i **A**dlemanu da stvore (RSA) kriptosistem sa javnim ključem. Oni su koristili činjenicu da je nalaženje velikih (npr. 100-cifrenih) prostih brojeva računarski dosta lako, ali faktorizacija proizvoda dva takva broja je, čini se, računarski praktično nemoguća.

Opisaćemo ukratko postupak određivanja ključeva i šifrovanja po ovom algoritmu.

Korisnik A bira dva vrlo velika prosta broja, P i Q, i njihovim množenjem dobija broj N. Broj N je javan, ali njegovi faktori P i Q drže se u tajnosti. Korišćenjem P i Q korisnik A može izračunati funkciju $\phi(N)$ (to je broj prirodnih brojeva manjih od N i relativno prostih sa N) kao:

$$\phi(N) = (P-1)*(Q-1) \qquad \textbf{(11)}$$

On onda bira drugi broj E iz intervala od 2 do $\phi(N)-1$. Ovaj broj je takođe javan. Poruka je prikazana kao niz brojeva M_1, M_2..., gde je svaki broj između 0 i $N-1$. Šifriranje se provodi na svakom bloku M korišćenjem javnih informacija E (to je javni ključ) i N (modulo, tj. aritmetika u kojoj se radi) kao:

$$C = M^E \bmod N \qquad \textbf{(12)}$$

gde C predstavlja šifrovani blok. Korišćenjem tajnog broja $\phi(N)$ korisnik A može lako izračunati broj D (tajni ključ) tako da:

$$(E * D) \bmod \phi(N) = 1 \qquad \textbf{(13)}$$

(ekvivalentno $E * D = k * \phi(N) + 1$).

Ovim je obezbeđena inverznost ključeva E i D a time i inverznost postupaka šifrovanja i dešifrovanja. Ako E ima zajednički faktor sa $\phi(N)$, onda D ne postoji i mora se izabrati drugi E. Onda zbog:

$$X^{k\phi(N)+1} = X \bmod N \qquad \textbf{(14)}$$

za sve cele brojeve između 0 i $N-1$ i k dešifrovanje je lako izvodljivo potenciranjem C na D-tu potenciju:

$$C^D = M^{ED} = M^{k\phi(N)+1} = M \bmod N \qquad \textbf{(15)}$$

Primer: Neka su izabrani P = 17 in Q = 31. Tada je N = PQ = 527 i $\phi(n) = (P-1)(Q-1) = 480$. Ako je E = 7, onda je D = 343 ($7 * 343 = 2401 = 5 * 480 + 1$). Ako je M = 2, onda:

$$\begin{aligned} C &= M^E \bmod N \\ &= 2^7 \bmod 527 \\ &= 128 \end{aligned}$$

Za šifrovanje bio je potreban javni ključ. Za dešifrovanje neophodan je tajni ključ:

$$\begin{aligned} M &= C^D \bmod N \\ &= 128^{343} \bmod 527 \\ &= 128^{256} * 128^{16} * 128^4 + 128^2 * 128^1 \bmod 527 \\ &= 35\ 256\ 35\ 101\ 47\ 128 \bmod 527 \\ &= 2 \bmod 527 \end{aligned}$$

Dakle, postupci šifrovanja i dešifrovanja su isti, stim što se u prvom slučaju koristi javni ključ (E) nad otvorenim tekstom (M), a u drugom tajni ključ (D) nad šifratom (C). Radi se sa numeričkim interpretacijama teksta (ASCII ili drugim).

Ilustracija šifrovanja po RSA algoritmu

Glavni problem kod realizacije šifrovanja po ovom algoritmu je izračunavanje jednosmerne funkcije koja je oblika:

m^h (mod n).

Ovde se koristi jedan dosta efikasan algoritam koji omogućuje izračunavanje eksponencijalne funkcije oblika m^h ponavljanjem kvadriranja i množenja u sledećih nekoliko koraka:

1. korak: Neka je $h_k h_{k-1} \ldots h_1 h_0$ binarna reprezentacija broja h
2. korak: c := 1
3. korak: i := k
4. korak: c := c^2 mod n
5. korak: ako je h_k=1, onda c := cm mod n
6. korak: i := i−1
7. korak: ako i<0, onda kraj, inače idi na korak 4.

Procedura za izračunavanje jednosmerne funkcije mogla bi izgledati ovako (programski jezik pascal):

Naravno, elegantnije rešenje je izdvojiti konverziju ključa u binarni oblik u posebnu proceduru. Konverziju treba obaviti samo jednom na početku šifrovanja/dešifrovanja poruke (šifrata). Rezultat se može smestiti u globalnu varijablu, tj. polje koje sadrži binarne cifre ključa. U ovom primeru A[0] je binarna cifra najmanje težine.

Primer šifrovanja. Recimo da je trebalo po ovom metodu šifrovati sledeći tekst:

»SNAGA RSA ALGORITMA JE U PROBLEMU FAKTORIZACIJE VELIKIH BROJEVA«

Uzmimo, za ilustraciju, proste brojeve:

P = 9 i Q = 11

Znači, radi se o aritmetici po modulu

N = P * Q = 99

Potrebo je izračunati funkciju

$\phi(N) = (P-1)*(Q-1) = 80$

Neka javni ključ bude E = 3 i tajni ključ D = 27, jer je

(3*27) mod 80 = 1

Napomena: U ovom primeru slova su interpretirana tako da je numerička interpretacija od A = 1, B = 2 itd. Šifrovani su blokovi od po jednog znaka (slova) i korišteni su mali prosti brojevi. To je učinjeno radi jednostavnije ilustracije. Ovako šifrovan tekst lako bi se, naravno, dešifrovao. Naime, u ovom primeru sistem je degradiran na prostu zamenu znakova.

Primena RSA algoritma

U stvarnim primenama RSA algoritma uzimaju se veliki prosti brojevi za generisanje ključa (recimo brojevi od oko 50–100 dekadskih cifara). Kao jedan blok poruke tada se uzima čitav niz od po 20 do 30 znakova. Recimo, u ASCII kodu, interpretacija niza znakova »ABCDEFGH« bi bila »6566676869707172«. Kod ovakvog šifrovanja ne postoji mogućnost faktorizacije broja N, niti izračunavanja tajnog ključa na bazi javnog (što se takođe svodi na faktorizaciju). Dakle, stvarna sigurnost RSA algoritma počiva na nemoći današnjih računara i algoritama da u raspoloživom vremenu izvrše faktorizaciju velikih brojeva, odnosno izračunaju inverz jednosmerne funkcije.

Prikažimo ovde neke podatke o vremenima potrebnim za faktorizaciju velikih brojeva:

Broj decimalnih cifara	Broj potrebnih operacija	Potrebno vreme
50	1,4 * 10	3,9 sati
75	9,0 * 10	104 dana
100	2,3 * 10	74 godine
200	1,2 * 10	3,8 * 10 godina
300	1,5 * 10	4,0 * 10 godina
500	1,3 * 10	4,2 * 10 godina

Kao rezultat dobija se sledeći šifrat (numerička interpretacija):

28 71 01 46 01 45 90 28 01 45 01 45 46 09 90 36 80 19 01 45
10 26 45 54 45 37 90 09 08 45 26 19 54 45 18 01 44 80 09 90
36 53 01 27 36 10 26 45 55 26 45 36 44 36 17 45 08 90 09 10
26 55 01

```
Procedure JednosmFunkcija (Poruka, Kljuc, Modulo : integer, Var Sifra
: integer);
{ Procedura dobija kao ulaz: numeričku interpretaciju bloka poruke, ključ
i modulo. Izlaz je šifrovani blok. Poruka, Kljuc, Modulo i Sifra su globalne
varijable tipa integer. }
Var A : array [1..500] of 0..1; {max. 500 binarnih cifara za predstavljanje
ključa}
I, J : integer;
Begin
  { Konverzija ključa u binarni }
  I := 0;
  While Kljuc > 0 do
  begin
    I := I+1; A[I] := Kljuc mod 2;
    Kljuc := Kljuc div 2
  end;
  { Šifrovanje jedinice teksta predstavljene numerički u varijabli Poruka }
  Sifra := 1;
  For J := I downto 1 do
  begin
    Sifra := Sifra * Sifra;
    If A[J] = 1 then Sifra := Sifra * Poruka;
    Sifra := Sifra mod Modulo
  end
End; { JednosmFunkcija }
```

RSA algoritam se smatra veoma sigurnim. Omogućava šifrovano komuniciranje velikog broja učesnika, pri čemu se može obezbediti mogućnost provere identiteta izvora poruke, ukoliko je to potrebno. Sistem sa javnim ključevima je novi koncept u kriptografiji, s obzirom na rešenje veoma osetljivog problema distribucije ključa.

Probleme, kod primene ovog algoritma, predstavljaju: dosta složen postupak šifrovanja, relativno niska brzina šifrovanja i složen postupak određivanja parova ključeva (javni i tajni). Naravno, ovi problemi se mogu ublažiti kvalitetnim algoritmima i brzim softverom i hardverom.

RSA algoritam je naročito pogodan za primenu kada više učesnika komunicira. Vrlo pogodno se može primeniti kod banaka, gde veliki broj poslovnica komunicira sa centralom ili kod sličnih ustanova kod kojih je bitna tajnost poruka (transakcija), a poslovna mreža je razgranata.