Sheme digitalnog potpisa

Hunjadi, Jelena

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Jelena Hunjadi

SHEME DIGITALNOG POTPISA

Diplomski rad

Voditelj rada:
izv. prof. dr. sc. Zrinka Franušić

Zagreb, 2016.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Sadržaj

Uvod

U današnje vrijeme život više nije moguće zamisliti bez komunikacije i poslovanja na globalnoj računalnoj mreži, odnosno online. Internet je omogućio brže i jednostavnije poslovanje, sklapanje ugovora, elektroničke transakcije i mnoge druge aktivnosti koje smanjuju operativne troškove i poboljšavaju kvalitetu usluge. Komunicirajući preko interneta, različiti poslovni procesi susreću se s raznim problemima, od kojih je najveći problem sigurnosti. Povjerljive informacije koje se razmjenjuju putem računalne mreže mogu imati dalekosežne nepovoljne posljedice za rad poslovne organizacije ako dođu u posjed neovlaštenim korisnicima. Stoga je od velike važnosti uspostaviti mehanizme koji će osigurati da povjerljive informacije ostanu povjerljive te da su dostupne samo ovlaštenim osobama.

Zbog toga, u poslovnom i ICT svijetu sve češće susrećemo pojam *digitalni potpis*. Digitalni potpis je digitalna verzija vlastoručnog potpisa, koja uz odgovarajuće zakone vrijedi jednako kao i vlastoručno potpisan dokument. On omogućuje utvrđivanje autentičnosti elektroničkog dokumenta, npr. elektroničkog pisma, web stranice ili slikovne datoteke, te na taj način bitno ubrzava i pojednostavljuje poslovanje uz ogromnu uštedu vremena.

Kako je danas rad bez računala i interneta gotovo nezamisliv u obavljanju bilo kakvog posla, potreba za očuvanjem autentičnosti i sigurnosti dokumenta postaje sve veća. Iz navedenog razloga digitalni potpis poprima sve širu uporabu te predstavlja ključ sigurnosti i povjerenja u suvremenom poslovanju.

Valjani digitalni potpis mora zadovoljiti nekoliko uvjeta:

1. Vjerodostojnost (engl. authenticity): osoba B zna da je samo osoba A mogla poslati poruku koju je ona upravo primila;
2. Netaknutost (engl. integrity): osoba B zna da poruka koju je poslala osoba A nije promijenjena prilikom slanja;
3. Nepobitnost (engl. non-repudiation): osoba A ne može kasnije zanijekati da je poslala poruku;

Ako se nastave današnji trendovi u korištenju digitalnog potpisa, oni će s vremenom potpuno zamijeniti pisane potpise. Mnogi računalni stručnjaci slažu se kako će, kroz određeno vrijeme, za pokretanje svih programskih paketa i pristup svim datotekama biti potreban odgovarajući digitalni potpis.

U ovom radu opisat će se neke od shema digitalnog potpisa. Rad se sastoji od tri poglavlja. U prvom poglavlju dan je pregled osnovnih pojmova i teorema iz teorije brojeva bitnih za razumijevanje kriptosustava s javnim ključem i digitalnog potpisa.

U drugom poglavlju govori se o kriptografiji javnog ključa ili asimetričnoj kriprografiji. Kao što i sam naziv kaže, u kriptosustavima s javnim ključem je ključ za šifriranje javan, odnosno dostupan svima. No, svaki sudionik komunikacije posjeduje i tajni ključ koji je sastavni dio postupka dešifriranja. Važno je reći da iz poznavanja javnog ključa za šifriranje nije moguće lako spoznati tajni ključ za dešifriranje. U tu svrhu koriste se tzv. *jednosmjerne funkcije*, funkcije čiji jedan smjer računamo lako, a suprotan (inverz) teško. Među najpoznatije kriptosustave s javnim ključem spadaju sustav *RSA* koji za zasniva na teškoći faktorizacije velikih složenih brojeva i *ElGamalov* sustav koji se zasniva na teškoći računanja *diskretnog* logaritma.

Digitalni potpis može se konkretno realizirati pomoću kriptosustava s javnim ključem. U trećem poglavlju opisuju se sheme digitalnog potpisa koje se temelje na spomenutim kriptosustavima. Bit će govora o njihovoj efikasnosti te o njihovim mogućim slabim točkama koje bi omogućile krivotvorenje potpisa.

Poglavlje 1

Pojmovi i tvrdnje iz teorije brojeva

1.1 Kongruencije

Definicija 1.1.1. *Ako cijeli broj $m \neq 0$ dijeli razliku $a-b$, onda kažemo da je a* kongruentan *b modulo m i pišemo $a \equiv b \pmod{m}$. U protivnom, kažemo da a nije kongruentan b modulo m i pišemo $a \not\equiv b \pmod{m}$.*

Definicija 1.1.2. *Neka su a i m prirodni brojevi, te b cijeli broj. Kongruencija oblika $ax \equiv b \pmod{m}$ se naziva* linearna kongruencija.

Rješenje kongruencije $ax \equiv b \pmod{m}$, gdje su a i m prirodni brojevi i b cijeli broj, je svaki cijeli broj x koji je zadovoljava. Ako je x_1 neko rješenje te kongruencije i $x_2 \equiv x_1 \pmod{m}$, onda je i x_2 također njeno rješenje. Za dva rješenja x i x' kongruencije $ax \equiv b \pmod{m}$ kažemo da su ekvivalentna ako je $x \equiv x' \pmod{m}$. Pod brojem rješenja kongruencije podrazumijevamo broj neekvivalentnih rješenja.

Teorem 1.1.3. *Neka su a i m prirodni, te b cijeli broj. Kongruencija $ax \equiv b \pmod{m}$, ima rješenja ako i samo ako najveći zajednički djelitelj od a i m, tj. $d = (a, m)$ dijeli b. Ako je ovaj uvjet zadovoljen, onda gornja kongruencija ima točno d rješenja modulo m.*

Iz ovog teorema slijedi da ako je p prost broj i a nije djeljiv s p, onda kongruencija $ax \equiv 1 \pmod{p}$ uvijek ima rješenje. Štoviše, to rješenje je jedinstveno i ono je multiplikativni inverz broja a modulo p. To je jedan od razloga što je skup ostataka pri dijeljenju s p, $\{0, 1, \ldots, p-1\}$, uz zbrajanje i množenje modulo p, čini polje. To polje se označava sa $\mathbb{Z}_p$.

Teorem 1.1.4 (Kineski teorem o ostacima)**.** *Neka su $m_1, m_2, \ldots, m_r$ u parovima relativno prosti realni brojevi, te neka su $a_1, a_2, \ldots, a_r$ cijeli brojevi. Tada sustav kongruencija*

$$x \equiv a_1 \pmod{m_1}, \quad x \equiv a_2 \pmod{m_2}, \quad \ldots, \quad x \equiv a_r \pmod{m_r} \tag{1.1}$$

ima rješenja. Ako je x_0 jedno rješenje, tada su sva rješenja od (1.1) dana s

$$x \equiv x_0 \pmod{m_1 m_2 \cdots m_r}.$$

Dokaz. Neka je $m = m_1 m_2 \cdots m_r$, te neka je $n_j = \frac{m}{m_j}$, za $j = 1, \ldots, r$. Tada je $(m_j, n_j) = 1$, pa postoji cijeli broj x_j takav da je $n_j x_j \equiv a_j \pmod{m_j}$. Promotrimo broj

$$x_0 = n_1 x_1 + \cdots + n_r x_r.$$

Za njega vrijedi $x_0 \equiv 0 + \cdots + 0 + n_j x_j + 0 + \cdots + 0 \equiv a_j \pmod{m_j}$. Prema tome, x_0 je rješenje od (1.1). Ako su sada x, y dva rješenja od (1.1), onda je $x \equiv y \pmod{m_j}$ za sve $j = 1, \ldots, r$. Jer su m_j u parovima relativno prosti, dobivamo da je $x \equiv y \pmod{m}$.

□

Definicija 1.1.5. Reducirani sustav ostataka modulo m *je skup cijelih brojeva* $\{r_1, \ldots, r_k\}$ *sa sljedećim svojstvima:*

- $(r_i, m) = 1$, *za* $i = 1, \ldots, k$,
- $r_i \not\equiv r_j \pmod{m}$ *za sve* $i \neq j$, $i, j \in \{1, \ldots, k\}$
- *za svaki cijeli broj* x *takav da je* $(x, m) = 1$ *postoji* r_i, $i \in \{1, \ldots, k\}$ *takav da je*

$$x \equiv r_i \pmod{m}.$$

Jedan reducirani sustav ostataka modulo m je skup svih brojeva $a \in \{1, \ldots, m\}$ takvih da je $(a, m) = 1$. Jasno je da svi reducirani sustavi ostataka modulo m imaju isti broj elemenata. Stoga sljedeća definicija ima smisla.

Definicija 1.1.6. *Broj elemenata u reduciranom sustavu ostataka modulo m označava se s* $\varphi(m)$. *Funkcija* $\varphi : \mathbb{N} \to \mathbb{N}$ *zove se* Eulerova funkcija.

Drugim riječima, $\varphi(m)$ je broj brojeva u nizu $1, \ldots, m$ koji su relativno prosti sa m.

Teorem 1.1.7 (Eulerov teorem)**.** *Ako je* $(a, m) = 1$, *onda je* $a^{\varphi(m)} \equiv 1 \pmod{m}$.

Teorem 1.1.8 (Mali Fermatov teorem)**.** *Neka je p prost broj. Ako* $p \nmid a$, *onda je* $a^{p-1} \equiv 1 \pmod{p}$. *Za svaki cijeli broj a vrijedi* $a^p \equiv a \pmod{p}$.

1.2 Primitivni korijen

Definicija 1.2.1. *Neka su a i n relativno prosti prirodni brojevi. Najmanji prirodni broj d sa svojstvom da je $a^d \equiv 1 \pmod{n}$ zove se* red *od a modulo n. Još se kaže da a* pripada eksponentu *d modulo n.*

Teorem 1.2.2. *Neka je d red od a modulo n. Tada za prirodan broj k vrijedi $a^k \equiv 1 \pmod{n}$ ako i samo ako $d|k$. Posebno, $d|\varphi(n)$.*

Dokaz. Ako $d|k$, recimo $k = d \cdot l$, onda je $a^k \equiv (a^d)^l \equiv 1 \pmod{n}$.

Obratno, neka je $a^k \equiv 1 \pmod{n}$. Podijelimo k sa d, pa dobivamo $k = q \cdot d + r$, gdje je $0 \leq r < d$. Sada je

$$1 \equiv a^k \equiv a^{qd+r} \equiv (a^d)^q \equiv a^r \pmod{n},$$

pa zbog minimalnosti od d slijedi da je $r = 0$, tj. $d|k$. □

Korolar 1.2.3. *Neka je d red od a modulo n. Ako $a^l \equiv a^k \pmod{n}$, onda vrijedi $l \equiv k \pmod{d}$.*

Definicija 1.2.4. *Ako je red od a modulo n jednak $\varphi(n)$, onda se a zove* primitivni korijen *modulo p.*

Ako postoji primitivni korijen modulo n, onda je grupa reduciranih ostataka modulo n ciklička.

Teorem 1.2.5. *Ako je p prost broj, onda postoji točno $\varphi(p-1)$ primitivnih korijena modulo p.*

U polju $\mathbb{Z}_p$ vrijedi da je broj elemenata koji pripadaju eksponentu d jednak upravo $\varphi(d)$.

Neka je g primitivni korijen modulo n. Lako se vidi da tada skup brojeva $\{g^l : l = 0, 1, \ldots, \varphi(n) - 1\}$ predstavlja reducirani sustav ostataka modulo n. Stoga za svaki cijeli broj a takav da je $(a, n) = 1$ postoji jedinstven $l \in \{0, 1, \ldots, \varphi(n) - 1\}$ takav da je $g^l \equiv a \pmod{n}$. Broj l se naziva *indeks* broja a s obzirom na primitivni korijen g, odnosno *diskretni logaritam* od a.

Poglavlje 2

Kriptografija javnog ključa

2.1 Kriptografija

Kriptografija je znanstvena disciplina koja se bavi proučavanjem metoda za slanje poruka u takvom obliku da ih samo onaj kome su namijenjene može pročitati. Sama riječ kriptografija grčkog je podrijetla i mogla bi se doslovno prevesti kao ”tajnopis”.

Osnovni zadatak kriptografije je omogućiti dvjema osobama (zvat ćemo ih *pošiljalac* i *primalac* - u kriptografskoj literaturi za njih su rezervirana imena Alice i Bob) komuniciranje preko nesigurnog kanala (telefonska linija, računalna mreža, ...) na način da treća osoba (njihov protivnik - u literaturi se najčešće zove Eva ili Oscar) koja može nadzirati komunikacijski kanal, ne može razumijeti njihove poruke. Poruku koju pošiljalac želi poslati primaocu zvat ćemo *otvoreni tekst* (engl. plaintext). To može biti tekst na njihovom materinjem jeziku, numerički podaci ili bilo što drugo. Pošiljalac transformira otvoreni tekst koristeći unaprijed dogovoreni dogovoreni *ključ* (engl. key). Taj postupak se naziva *šifriranje*, a dobiveni rezultat *šifrat* (engl. ciphertext) ili *kriptogram*. Nakon toga pošiljalac pošalje šifrat preko nesigurnog komunikacijskog kanala. Protivnik prisluškujući mora doznati sadržaj šifrata, ali ne može odrediti otvoreni tekst. Za razliku od njega, primalac koji zna ključ kojim je šifrirana poruka može *dešifrirati* šifrat i odrediti otvoreni tekst.

Kriptografski algoritam ili *šifra* je matematička funkcija koja se koristi za šifriranje i dešifriranje. Općenito, radi se o dvije funkcije, jednoj za šifriranje, a drugoj za dešifriranje. Te funkcije preslikavaju osnovne elemente otvorenog teksta (najčešće su to slova, bitovi, grupe slova ili bitova) u osnovne elemente šifrata, i obratno. Funkcije se biraju iz određene familije funkcija u ovisnosti u ključu. Skup svih vrijednosti ključeva naziva se *prostor ključeva*. *Kriptosustav* se sastoji od kriptografskog algoritma, te svih mogućih otvorenih tekstova, šifrata i ključeva. Dakle, imamo sljedeću formalu definiciju:

Definicija 2.1.1. *Kriptosustav je uređena petorka $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, gdje je $\mathcal{P}$ konačan skup svih mogućih osnovnih elemenata otvorenog teksta, $\mathcal{C}$ konačan skup svih mogućih osnovnih elemenata šifrata, $\mathcal{K}$ konačan skup svih mogućih ključeva, $\mathcal{E}$ skup svih funkcija šifriranja i $\mathcal{D}$ skup svih funkcija dešifriranja. Za svaki $K \in \mathcal{K}$ postoji funkcija šifriranja $e_K \in \mathcal{E}$ i odgovarajuća funkcija dešifriranja $d_K \in \mathcal{D}$. Pritom su $e_K : \mathcal{P} \to \mathcal{C}$ i $d_K : \mathcal{C} \to \mathcal{P}$ funkcije sa svojstvom da je $d_K(e_K(x)) = x$ za svaki $x \in \mathcal{P}$.*

Postoji više podjela kriptosustava. Najčešće ih dijelimo prema tipu operacija koje se koriste, prema načinu obrade otvorenog teksta, te prema javnosti, odnosno tajnosti ključa. Prema ovoj posljednoj podjeli dijele se na *simetrične* i *asimetrične*.

Kod simetričnih kriptosustava, ključ za dešifriranje može se izračunati poznavajući ključ za šifriranje i obratno. Ustvari su ovi ključevi najčešće identični. Sigurnost ovih kriptosustava leži u tajnosti ključa. Zato se oni i zovu *kriptosustavi s tajnim ključem*.

Kod asimetričnih kriptosustava ključ za dešifriranje ne može se (barem ne u nekom razumnom vremenu) izračunati iz ključa za šifriranje. Ovdje je ključ za šifriranje *javni ključ*. Naime, bilo tko može šifrirati poruku pomoću njega, ali samo osoba koja ima odgovarajući ključ za dešifriranje (*privatni* ili *tajni* ključ) može dešifrirati tu poruku. Zbog svega ovoga se ovi kriptosustavi još i nazivaju *kriptosustavi s javnim ključem*.

2.2 Ideja javnog ključa

Problem kod simetričnih kriptosustava jest što je za njihovu sigurnost nužna tajnost ključa. To je i njihov veliki nedostatak, budući da prije šifriranja pošiljatelj i primatelj moraju izmijeniti tajni ključ preko nekog sigurnog komunikacijskog kanala. Ako stvarno imaju dostupan sigurni kanal, možemo se pitati zašto ga ne koriste za razmjenu poruka, već ih šalju preko nesigurnog kanala. Štoviše, oni bi morali često mijenjati ključeve, budući da šifriranje više poruka istim ključem smanjuje sigurnost. To može biti veliki problem ako npr. pošiljatelj i primatelj žive u udaljenim mjestima i žele sigurno komunicirati internetom.

Godine 1976. Whitfield Diffie i Martin Hellman ponudili su jedno moguće rješenje problema razmjene ključeva, zasnovano na činjenici da je u nekim grupama potenciranje puno jednostavnije od logaritmiranja.

Diffie i Hellman smatraju se začetnicima kriptografije javnog ključa. Ideja javnog ključa sastoji se u tome da se konstruiraju kriptosustavi kod kojih bi iz poznavanja funkcije šifriranja e_k bilo praktički nemoguće, to jest nemoguće u nekom razumnom vremenu,

izračunati funkciju dešifriranja d_k. Tada bi funkcija šifriranja e_k mogla biti javna. U provedbi ove ideje ključnu ulogu igraju tzv. *jednosmjerne funkcije*. Za funkciju f kažemo da je jednosmjerna ako je f lako, a f^{-1} teško izračunati. Ako je pritom f^{-1} lako izračunati ukoliko nam je poznat neki dodatni podatak (engl. trapdoor - skriveni ulaz), onda f nazivamo *osobna jednosmjerna funkcija*.

Definicija 2.2.1. *Pretpostavimo da znamo skup S_K svih mogućih korisnika K.* Kriptosustav s javnim ključem *sastoji se od dviju familija funkcija $\{e_k\}$ i $\{d_k\}$ pri čemu k odgovara korisniku K sa svojstvima:*

1. *d_k je inverz od e_k;*
2. *e_k je javan, a d_k poznat samo osobi K;*
3. *e_k je osobna jednosmjerna funkcija.*

Za svaku osobu (korisnika) $K \in S_k$, e_k predstavlja njegovu funkciju za šifriranje *koju još nazivamo i* javnim ključem. *Funkcija d_k služi za* deširiranje *i zovemo je* tajnim *ili* osobnim ključem.

Ako Alice želi poslati Bobu poruku x, onda Alice iz javnog direktorija uzima Bobov javni ključ e_B. Potom Alice šifrira svoju poruku pomoću e_B i pošalje Bobu šifrat $y = e_B(x)$. Konačno, Bob dešifrira šifrat koristeći svoj tajni ključ d_B i dobiva $d_B(y) = d_B(e_B(x)) = x$.

U konstrukciji kriptosustava s javim ključem, tj. osobnih jednosmjernih funkcija, obično se koriste neki "teški" matematički problemi, kao što su problem faktorizacije velikih prirodnih brojeva i problem diskretnog logaritma. Te probleme, kao i kriptosustave koji su na njima zasnovani, upoznat ćemo u nastavku.

2.3 RSA kriptosustav

Najpoznatiji kriptosustav s javnim ključem je RSA kriptosustav iz 1977. godine, nazvan po svojim tvorcima Ronaldu Rvestu, Adi Shamiru i Leonardu Adlermanu. RSA kriptosustav temelji se na sljedećoj pretpostavci: nije teško pomnožiti dva velika prosta broja, ali je jako teško rastaviti veliki složeni broj na njegove proste faktore. Odnosno, sigurnost RSA kriptosustava zasnovana je upravo na teškoći faktorizacije velikih prirodnih brojeva. Slijedi definicija RSA kriptosustava.

Definicija 2.3.1 (RSA kriptosustav)**.** *Neka je $n = pq$, gdje su p i q prosti brojevi. Neka je $\mathcal{P} = C = \mathbb{Z}_n$, te*

$$\mathcal{K} = \{(n, p, q, d, e) : n = pq, de \equiv 1 \pmod{\varphi(n)}\}.$$

Za $K \in \mathcal{K}$ definiramo

$$e_k(x) = x^e \mod n, \quad d_K(y) = y^d \mod n, \quad x, y \in \mathbb{Z}_n.$$

Vrijednosti n i e su javne, a vrijednosti p, q i d su tajne, tj. (n, e) je javni, a (p, q, d) je tajni ključ.

Generiranje ključeva

Objasnimo kako Bob generira svoj javni i tajni RSA ključ. Bob odabire dva velika prosta broja p i q i računa njihov produkt

$$n = pq.$$

Zatim odabire broj e takav da je

$$1 < e < \varphi(n) = (p-1)(q-1) \quad \text{i} \quad \gcd(e, (p-1)(q-1)) = 1.$$

Primijetimo da je e uvijek neparan jer je $p - 1$ paran. Nadalje, Bob računa broj d:

$$1 < d < (p-1)(q-1) \quad \text{i} \quad de \equiv 1 \pmod{(p-1)(q-1)}.$$

Kako je $\gcd(e, (p-1)(q-1)) = 1$, takav broj d postoji i, štoviše, jedinstven je u skupu $\{1, \ldots, (p-1)(q-1) - 1\}$. Može se izračunati pomoću proširenog Euklidovog algoritma. Sada je Bobov javni ključ par (n, e). Njegov tajni ključ je d. Primijetimo da se tajni ključ d može izračunati ako su poznati prosti faktori p i q. Zbog toga, ako napadač, Oscar, uspije otkriti faktorizaciju broja n, lako može izračunati Bobov tajni ključ d, računanjem $\varphi(n) = (p-1)(q-1)$ i rješavanjem linearne kongruencije

$$de \equiv 1 \pmod{\varphi(n)}.$$

Kasnije, u ovom radu, analizirat ćemo kako odabrati p i q kako bi faktorizacija od n bila što teža.

Primjer 2.3.2. *Bob odabire proste faktore $p = 11$ i $q = 23$. Tada je $n = 253$ i $(p-1)(q-1) = 10 \cdot 22 = 220$. Najmanji mogući e je $e = 3$ kako je $\gcd(3, 220) = 1$. Proširenim Euklidovim algoritmom dobiva se $d = 147$.*

Enkripcija

Prostor otvorenog teksta sastoji se od brojeva m takvih da je $0 \leqslant m < n$. Otvoreni tekst enkriptira se računanjem

$$c = m^e \mod n.$$

Šifrirani tekst je c. Ako Alice zna javni ključ (n, e), onda na ovaj način može šifrirati željeni otvoreni tekst.

Postupak šifriranja poruke, tj. računanja šifrata $c = m^e \mod n$ naziva se *modularno potenciranje*. Za efikasnost RSA kriptosustava vrlo je bitno da se ono može efikasno izvesti. Navedimo ovdje osnovnu metodu za računanje $c = m^e \mod n$, metodu "kvadriraj i množi". Najprije e prikažemo u bazi 2:

$$e = 2^{s-1} \cdot e_{s-1} + \ldots + 2 \cdot e_1 + e_0,$$

a potom primijenimo sljedeći algoritam:

$c = 1$
za $i = s-1, \ldots, 1, 0$ radi:
    $c = c^2 \mod n$
    ako je $e_i = 1$ tada je $c = c \cdot m \mod n$

Primjer 2.3.3. *Kao u Primjeru (2.3.2), neka je* $n = 253$ *i* $e = 3$*. Tada je prostor otvorenog teksta {0, 1, ..., 252}. Kriptirajući broj* $m = 165$*, dobivamo* $c = 165^3 \mod 253 = 110$*.*

Dekripcija

Dekriptiranje RSA temelji se na sljedećem teoremu:

Teorem 2.3.4. *Neka je* (n, e) *javni RSA ključ i* d *odgovarajući tajni RSA ključ. Tada*

$$(m^e)^d \mod n = m$$

za svaki m *takav da je* $0 \leqslant m < n$*.*

Dokaz. Kako je $ed \equiv 1 \mod (p-1)(q-1)$, postoji prirodni broj l takav da je

$$ed = 1 + l(p-1)(q-1).$$

Zato

$$(m^e)^d = m^{ed} = m^{1+l(p-1)(q-1)} = m(m^{(p-1)(q-1)})^l.$$

Slijedi

$$(m^e)^d = m(m^{(p-1)})^{(q-1)l} \equiv m \pmod{p}.$$

Ako $p \nmid m$, prethodna tvrdnja slijedi iz Malog Fermatovog teorema (1.1.8). U protivnom, tvrdnja je trivijalna jer vrijedi

$$(m^e)^d = m(m^{(p-1)})^{(q-1)l} \equiv 0 \equiv m \pmod{p}.$$

Analogno, vidimo da je

$$(m^e)^d \equiv m \pmod{q}$$

Jer su p i q različiti prosti brojevi, dobivamo

$$(m^e)^d \equiv m \pmod{n}.$$

Tvrdnja slijedi iz činjenice da je $0 \leqslant m < n$. □

Ako je šifrirani tekst c izračunat kao u Primjeru (2.3.3), tada se po Teoremu (2.3.4) otvoreni tekst m može rekonstruirati rješavanjem kongruencije

$$m = c^d \mod n.$$

Primjer 2.3.5. *U Primjerima (2.3.2) i (2.3.3) odabrali smo $n = 253$ i $e = 3$ i izračunali $d = 147$ i $c = 110$. Sada dobivamo $110^{147} \mod 253 = 165$, što je i bio originalni otvoreni tekst.*

Odabir parametara p, q i e

Danas je praktički nemoguće rastaviti na faktore pažljivo odabran prirodan broj s više od 250 znamenki. Ipak, postoje neki slučajevi u kojima je n lakše faktorizirati, pa takve n-ove treba izbjegavati.

Kako bi otežali faktorizaciju broja n, potrebno je na pravi način odabrati faktore p i q. Ovi faktori biraju se na način da najprije generiramo slučajan prirodan broj m s traženim brojem znamenki, a zatim pomoću nekog testa prostosti tražimo prvi prost broj veći ili jednak m. Po teoremu o distribuciji prostih brojeva, možemo očekivati da ćemo trebati testirati približno $\ln m$ brojeva dok ne nađemo prvi prost broj. Pritom treba paziti da broj $n = pq$ bude otporan na metode faktorizacije koje su vrlo efikasne za brojeve specijalnog oblika. Tako bi brojevi $p \pm 1$ i $q \pm 1$ trebali imati barem jedan veći prosti faktor, jer postoje efikasne metode za faktorizaciju brojeva koji imaju prosti faktor p takav da je jedan od brojeva $p - 1$, $p + 1$ "gladak", tj. ima samo male proste faktore. Također, p i q ne smiju biti jako blizu jedan drugome, jer ih se onda može naći koristeći činjenicu da su približno jednaki $\sqrt{n}$.

Javni ključ e može se birati tako da on bude najmanji mogući, kako bi enkripcija bila što efikasnija, jer broj operacija u šifriranju ovisi o veličini broja e, te o broju jedinica u binarnom zapisu od e. Izbor $e = 2$ je nemoguć jer $\varphi(n) = (p-1)(q-1)$ je paran, a moramo imati $\gcd(e, (p-1)(q-1)) = 1$. Najmanji mogući eksponent je $e = 3$. Ipak, korištenje ovako malog eksponenta e može biti opasno. Ako je neka poruka m enkriptirana e puta

s eksponentom e i napadač dozna te šifrate, on može na jednostavan način, rješavanjem sustava linearnih kongruencija, pronaći originalnu poruku m.

Pretpostavimo da imamo tri korisnika s tri različita javna ključa n_1, n_2 i n_3, te pretpostavimo da svi oni koriste isti javni eksponent za šifriranje $e = 3$. Nadalje, pretpostavimo da im netko želi poslati identičnu poruku m. Tada njihov protivnik može doznati sljedeće šifrate:

$$c_1 \equiv m^3 \pmod{n_1}, \qquad c_2 \equiv m^3 \pmod{n_2}, \qquad c_3 \equiv m^3 \pmod{n_3}$$

Koristeći Kineski teorem o ostacima (1.1.4), protivnik lako može naći rješenje sustava linearnih kongruencija

$$x \equiv c_1 \pmod{n_1}, \qquad x \equiv c_2 \pmod{n_2}, \qquad x \equiv c_3 \pmod{n_3},$$

odnosno dobiti broj x sa svojstvom

$$x \equiv m^3 \pmod{n_1 n_2 n_3}.$$

No, kako je $m^3 < n_1 n_2 n_3$, zapravo vrijedi jednakost $x = m^3$, pa protivnik može izračunati originalnu poruku m tako da nađe treći korijen iz x.

Primjer 2.3.6. *Pretpostavimo da je* $n_1 = 377, n_2 = 407, n_3 = 589$. *Pretpostavimo da je Oscar saznao šifrate* $c_1 = 285, c_2 = 214, c_3 = 202$*, te želi saznati zajednički otvoreni tekst m. Rješavanjem sustava*

$$x \equiv 285 \pmod{377}, \qquad x \equiv 214 \pmod{407}, \qquad x \equiv 202 \pmod{589}$$

pomoću Kineskog teorema o ostacima, dobiva se da je

$$x \equiv 407 \cdot 589 \cdot 225 + 377 \cdot 589 \cdot 18 + 377 \cdot 407 \cdot 503 \equiv 44738875 \pmod{377 \cdot 407 \cdot 589}.$$

To znači da je $x = 44738875$ *i* $m = \sqrt[3]{44738875} = 355$.

Zbog toga se preporuča izbor većeg eksponenta e, npr. $e = 65537$, koji je dovoljno velik da bi onemogućio sve poznate napade na RSA s malim eksponentom, a prednost mu je vrlo brzo šifriranje jer ima malo jedinica u binarnom zapisu. Naime, $65537 = 2^{16} + 1$. Budući da je broj operacija množenja u algoritmu "kvadriraj i množi" jednak broju jedinica u binarnom zapisu eksponenta, to uvelike smanjuje vrijeme potrebno za šifriranje.

Još uvijek nije pronađena metoda koja bi razbila RSA kriptosustav. Svi poznati napadi na RSA zapravo samo pokazuju na što treba paziti i što treba izbjegavati kod izbora parametara i implementacije RSA. Zasad se, uz korektnu implementaciju, RSA može smatrati sigurnim kriptosustavom.

2.4 Diffie-Hellmanov protokol za razmjenu ključeva

U ovom poglavlju opisat ću *Diffie-Hellmanov protokol za razmjenu ključeva* putem nesigurnog komunikacijskog kanala. Ovaj protokol sam po sebi nije kriptosustav s javim ključem, ali je osnova za El Gamalov kriptosustav koji će biti opisan u sljedećem poglavlju.

Pretpostavimo da Alice i Bob žele koristiti simetrični kriptosustav kako bi svoje komuniciranje putem nesigurnog kanala očuvali tajnim. Prije toga, Alice i Bob moraju razmijeniti tajni ključ. Diffie-Hellmanov protokol za razmjenu ključeva će im to omogućiti.

Diffie-Hellmanov algoritam predstavlja prekretnicu u kriptografiji javnog ključa. Njegova sigurnost zasnovana je na *problemu diskretnog logaritma* (PDL). Ovaj problem zasniva se na činjenici da su u nekim grupama operacije množenja i potenciranja jednostavne, dok bi logaritmiranje (inverzna operacija od potenciranja) trebalo biti vrlo teško.

Problem diskretnog logaritma

Neka je p prost broj. Tada je $(\mathbb{Z}_p^*, \cdot_p)$ ciklička grupa reda $p-1$. Neka je g primitivni korijen mod p. Tada za bilo koji cijeli broj $A \in \{1, 2, \ldots, p-1\}$ postoji jedinstven eksponent $a \in \{0, 1, 2, \ldots, p-2\}$ takav da je

$$A \equiv g^a \pmod{p}.$$

Eksponent a naziva se *diskretnim logaritmom* od A po bazi g. Pišemo $a = \log_g A$.

Za računanje diskretnog logaritma do danas nije poznat efikasan algoritam. S druge strane, postavlja se pitanje koliko je problem diskretnog logaritma zaista težak.

Razmjena ključeva

Diffie-Hellman protokol funkcionira na sljedeći način. Alice i Bob žele se dogovoriti oko tajnog ključa. Najprije odabiru veliki prosti broj p i primitivni korijen g modulo p. Može se odabrati i neki drugi g takav da je $2 \leq g \leq p-2$ i takav da je red od g mod p dovoljno velik. Prost broj p i primitivni korijen g su javni. Nakon toga, Alice nasumično odabire cijeli broj $a \in \{0, 1, \ldots, p-2\}$ i računa

$$A = g^a \bmod p$$

te taj rezultat šalje Bobu, ali eksponent a zadržava tajnim. Bob zatim odabire nasumični cijeli broj $b \in \{0, 1, \ldots, p-2\}$, računa

$$B = g^b \bmod p$$

i šalje rezultat Alice, zadržavajući pritom eksponent b tajnim. Kako bi dobila zajednički tajni ključ, Alice računa

$$B^a \bmod p = g^{ab} \bmod p,$$

dok Bob računa

$$A^b \bmod p = g^{ab} \bmod p.$$

Ustanovimo da su oboje, i Alice i Bob izračunali istu vrijednost

$$K = g^{ab} \bmod p$$

koja predstavljaj njihov *zajednički tajni ključ*.

Primjer 2.4.1. *Neka je* $p = 23$ *i* $g = 5$. *Alice odabire* $a = 13$, *računa* $g^a \bmod p = 21$ *i šalje rezultat* $A = 21$ *Bobu. Bob odabire* $b = 4$, *računa* $g^b \bmod p = 4$ *i šalje rezultat* $B = 4$ *Alice. Alice računa* $B^a \bmod p = 16$. *Bob računa* $A^b \bmod p = 16$. *Njihov zajednički ključ tada je 16.*

Oscar-protivnik, koji može prišlukivati njihovu komunikaciju kroz nesigurni komunikacijski kanal, zna sljedeće podatke: p, g, g^a, g^b. Oscar treba iz ovih podataka izračunati g^{ab}. Kaže se da treba riješiti *Diffie-Hellmanov problem* (DHP). Ako Oscar iz poznavanja g i g^a izračuna a, to jest ako može riješiti problem diskretnog logaritma, onda može izračunati i $(g^b)^a = g^{ab}$. Vjeruje se da su za većinu grupa, koje se koriste u kriptografiji, ova dva problema, DHP i PDL, ekvivalentni, to jest da postoje polinomijalni algoritmi koji svode jedan problem na drugi.

Odabir parametra g

Kao i brojni drugi protokoli u kriptografiji, Diffie-Hellmanov protokol zahtijeva odabir cijelog broja g takvog da je njegov red modulo p dovoljno velik. Broj g može se odabrati kao primitivni korijen modulo p. Pri testiranju je li neki broj primitivni korijen potrebno je rastaviti broj $p - 1$ na proste faktore. S obzirom na veličinu broja p, problem faktorizacije broja $p-1$ je može biti jako zahtjevan. Stoga je problem pronalaženja primitivnog korijena modulo p također zahtjevan.

2.5 ElGamalov kriptosustav

ElGamalov kriptosustav usko je vezan uz Diffie-Hellmanov protokol za razmjenu ključeva. Njegova sigurnost također se temelji na teškoći rješavanja Diffie-Hellmanovog problema u $(\mathbb{Z}_p^*, \cdot_p)$.

Definicija 2.5.1 (ElGamalov kriptosustav)**.** *Neka je p prost broj i $g \in \mathbb{Z}_p^*$ primitivni korijen modulo p. Neka je*

$$\mathcal{K} = \{(p, g, a, A) : A = g^a \mod p\}$$

Vrijednosti p, g, A su javne, a vrijednost a je tajna. Za $K \in \mathcal{K}$ i tajni slučajni broj $b \in \{0, 1, \ldots, p-2\}$ definiramo

$$e_K(x, b) = (g^b \mod p,\ \ xA^b \mod p).$$

Za $y_1, y_2 \in \mathbb{Z}_p^*$ definiramo

$$d_K(y_1, y_2) = y_2(y_1^a)^{-1} \mod p.$$

Generiranje ključeva

Alice odabire prosti broj p i, kao što je objašnjeno u Potpoglavlju (2.4), primitivni korijen g mod p. Zatim odabire nasumični eksponent $a \in \{1, \ldots, p-2\}$ i računa

$$A = g^a \bmod p.$$

Njezin javni ključ tada je (p, g, A), a tajni ključ eksponent a. Cijeli broj A je Alicin dio ključa iz Diffie-Hellman protokola.

Enkripcija

Prostor otvorenog teksta je skup $\{0, 1, \ldots, p-1\}$. Kako bi kriptirao tekst m, Bob dobiva od Alice javni ključ (p, g, A). Zatim odabire nasumični cijeli broj $b \in \{0, \ldots, p-2\}$ i računa

$$B = g^b \bmod p.$$

Broj B je Bobov dio ključa iz Diffie-Hellman protokola. Bob određuje

$$c = A^b m \bmod p.$$

Drugim riječima, Bob kriptira poruku m multipliciranjem Diffie-Hellmanovim ključem. ElGamalov šifrat tada je par (B, c).

Dekripcija

Alice dobiva šifrat (B, c). Ona zna svoj tajni ključ a. Kako bi rekonstruirala tekst m, dijeli c Diffie-Hellmanovim ključem $B^a \mod p$. Kako bi izbjegla inverzije mod p, određuje eksponent

$$x = p - 1 - a.$$

Kako je $1 \leq a \leq p-2$, vrijedi $1 \leq x \leq p-2$. Zatim računa

$$m = B^x c \mod p.$$

Tada je m originalni tekst, kao što pokazuje sljedeći račun:

$$B^x c \equiv g^{b(p-1-a)} A^b m \equiv (g^{p-1})^b (g^a)^{-b} A^b m \equiv A^{-b} A^b m \equiv m \pmod p.$$

Primjer 2.5.2. *Alice i Bob komuniciraju ElGamalovim kriptosustavom s parametrima $p = 17$i $g = 3$. Alice odabire $a = 5$ i računa $A = g^a \mod p = 5$. Njezin javni ključ je $(p = 17, g = 3, A = 5)$, a njezin tajni ključ je $a = 5$. Bob želi šifrirati $m = 9$. Odabire $b = 7$ i računa $B = g^b \mod p = 11$ i $c = A^b m \mod p = 5$. Sada je šifrat $(B, c) = (11, 5)$. Alice dobiva m računajući $B^{p-1-5} c \mod p = 9$.*

Efikasnost

Dekriptiranje kod ElGamalogov kriptosustava, kao i kod RSA kriptosustava, zahtijeva jedno modularno potenciranje. Kriptiranje kod ElGamalovog kriptosustava zahtijeva dva modularna potenciranja - računanje $A^b \mod p$ i $B = g^b \mod p$. RSA kriptiranje zahtijeva samo jedno modularno potenciranje. Ali, potenciranja kod ElGamalovog kriptosustava neovisna su o otvorenom tekstu koji se kriptira. Zbog toga, ta potenciranja mogu se smatrati predračunom. Tada stvarno kriptiranje zahtijeva samo jedno modularno množenje i zbog toga je mnogo efikasnije od RSA enkriptiranja. Ali, takve prethodno izračunate vrijednosti moraju se sigurno pohraniti i čuvati tajnima.

Primjer 2.5.3. *Kao u primjeru (2.5.2), javni ključ od Alice je $(p = 17, g = 3, A = 5)$. Njezin tajni ključ je $a = 5$. Bob odabire $b = 7$ i računa $B = g^b \mod p = 11$ i $K = A^b \mod p = 10$. Ako Bob želi Alice poslati poruku $m = 9$, tada šifrat dobiva običnim množenjem $c = K \cdot m \mod 17 = 5$.*

U ElGamalovom sustavu, šifrat je dvostruko dulji od otvorenog teksta, što je njegov nedostatak. Duljina javnog ključa može se smanjiti ukoliko se isti prosti broj p koristi u svim javnim ključevima. Ipak, ukoliko se pokaže da je računanje diskretnog logaritma modulo taj specifični p jednostavno, tada cijeli sustav postaje nesiguran.

Poglavlje 3

Digitalni potpis

3.1 Ideja

Digitalni potpis koristi se za potpisivanje elektroničkih dokumenata kao što su bankovne transakcije ili elektronički ugovori. Pri tomu se želi zadržati analogija vlastoručnog, odnosno "papirnatog" potpisa. Naime, ako Alice potpiše neki dokument vlastoručno, onda će svi koji ju poznaju moći potvrditi autentičnost njenog potpisa. Vlastoručni potpis ima veliku snagu. On se, na primjer, može upotrijebiti na sudu kao dokaz da Alice zna za taj dokument i slaže se s njegovim sadržajem ukoliko je potvrđena vjerodostojnost potpisa.

Kriptosustavi s javnim ključem omogućavaju nam da ideju digitalnog potpisa i realiziramo. Prvi koji su opisali tu ideju bili su Diffie i Hellman još 1976. godine. Ona se sastoji u tome da se pomoću originalne poruke i tajnog ključa osobe A generira digitalni potpis za osobu A. Imajući na raspolaganju poruku, digitalni potpis i javni ključ osobe A, osoba B može verificirati autentičnost potpisa. Digitalni potpis mora omogućavati sljedeće:

1. *autentifikaciju* - Bob može provjeriti je li poruku koju je dobio zaista poslala Alice;
2. *nepobitnost*- Alice ne može poreći da je poslala poruku ako Bob posjeduje poruku s njezinim potpisom.

Kriptosustav digitalnog potpisa uključuje tri algoritma:

1. generiranje ključa (javnog i tajnog) za potpisivanje;
2. potpisivanje poruke - generiranje poruke koju nazivamo "potpis";
3. verifikaciju potpisa.

Kriptosustavi s javnim ključem na vrlo jednostavan način omogućuju digitalno potpisivanje. Pretpostavimo da svaki od sudionika u komunikaciji ima svoj tajni ključ t i javni ključ j sa svojstvom da je $j \circ t = t \circ j = id$. Alice posjeduje tajni ključ t_A i javni ključ j_A, a Bob ključeve t_B i j_B. Općenito digitalni potpisi funkcioniraju na sljedeći način. Pretpostavimo da Alice želi potpisati neki dokument x i poslati ga Bobu. Ona najprije koristi Bobov javni ključ j_B za kriptiranje poruke m i računa $j_B(m)$ a zatim ga *potpisuje* svojim tajnim ključem t_A te Bobu šalje poruku $x = t_A(j_B(m))$. Bob očekuje poruku od Alice i koristi najprije njen javni ključ a zatim svoj tajni ključ za dekriptiranje. Konkretno Bob računa

$$t_B(j_A(x)) = t_B(j_A(t_A(j_B(m)))) = t_B(\underbrace{j_A \circ t_A}_{=id}(j_B(m))) = \underbrace{t_B \circ j_B}_{=id}(m) = m.$$

Dobivanjem smislene poruke m, Bob bi trebao biti siguran da je poruku poslala upravo Alice.

Ovom jednostavnom shemom ostvarujemo digitalni potpis. Nedostatci su što ovakav potpis može uzeti puno vremena. Ovakvo potpisivanje je vrlo sporo budući da se radi o implementaciji kriptosustava s javnim ključem, a ti su znatno sporiji od simetričnih kriptosustava (odnosno kriptosustava s tajnim ključem). Zamislimo samo da se potpisuje neki video sadržaj. Stoga se umjesto originalne poruke često koristi *sažetak poruke* (engl. message digest) koji se dobije primjenom neke hash funkcije.

Shema digitalnog potpisa sigurna je ukoliko je problem konstrukcije tajnog ključa za potpisivanje iz javno dostupnih informacija neizvediv. Sheme koje se danas upotrebljavaju imaju ovo svojstvo. Ono se temelji na teškoći rješavanja nekih problema iz teorije brojeva. Ne tvrdimo da su ti problemi nerješivi već da ih je nemoguće riješiti u vremenu u kojem bi razbijen šifrat bio relevantan. Obično kažemo da ih nije moguće riješiti u *realnom vremenu*.

Bitno je naglasiti da pronalaženje tajnog ključa za potpisivanje nije jedini mogući cilj napadača. On također može generirati nove valjane potpise bez znanja tajnog ključa. U tom slučaju, napadač uzima javni ključ od Alice i računa poruku x i potpis za x koji se može verificirati njezinim javnim ključem. Naravno, u ovom slučaju Oscar samo pokušava pogoditi potpis. Ukoliko je tekst koji se dobije verifikacijom potpisa smislen, Oscar je na ovaj način uspio krivotvoriti Alicin potpis. Ipak, mala je vjerojatnost da će ovakav potpis biti valjan. Nadalje, napadač također može iskoristiti valjane potpise kako bi konstruirao nove potpise. O konkretnim primjerima ovih napada bit će riječi nešto kasnije.

3.2 RSA potpis

U prethodnom poglavlju opisali smo najpoznatiji kriptosustav s javnim ključem, RSA kriptosustav. Ovaj sustav može se koristiti i za generiranje digitalnih potpisa. Ideja je vrlo jednostavna. Alice potpisuje dokument m tako da izračuna potpis

$$s = s(d, m) = m^d \bmod n.$$

Ovdje je d njezin tajni eksponent, a (n, e) je njen javni ključ. Bob provjerava njezin potpis računajući

$$s^e \bmod n = m^{ed} \equiv m \pmod{n}.$$

Zašto je ovo dobra ideja za digitalni potpis? Kao prvo, Bob jednostavnim modularnim potenciranjem dolazi do dokumenta m. Nadalje, Alice je Bobu poslala e-ti korijen dokumenta m. Jedina poznata metoda za računanje e-tog korijena nekog cijelog broja m modulo n je izračunati m^d pri čemu je $ed \equiv 1 \pmod{\varphi(n)}$. Iz poznavanja javnog ključa (n, e) nije moguće u realnom vremenu izračunati d. Za to bi nam bilo nužno poznavanje dodatnog podatka, tzv. *trapdoora*, odnosno faktorizacije broja n. Stoga je Bob siguran da je Alice jedina osoba koja je izračunala s i na taj način potpisala m.

Generiranje ključeva

Generiranje ključeva za RSA potpis jednako je generiranju ključeva pri RSA enkriptiranju. Alice odabire dva velika prosta broja p i q i eksponent e takav da je

$$1 < e < (p-1)(q-1) \quad \text{i} \quad \gcd(e, (p-1)(q-1)) = 1$$

Zatim računa $n = pq$ i d takav da je

$$1 < d < (p-1)(q-1) \quad \text{i} \quad de \equiv 1 \pmod{(p-1)(q-1)}.$$

Njezin javni ključ je (n, e), a tajni ključ je d.

Generiranje potpisa

Alice sada želi potpisati poruku $m \in \{0, 1, ..., n-1\}$. Alicein digitalni potpis glasi

$$s = m^d \bmod n.$$

Verifikacija potpisa

Bob želi provjeriti potpis s. Uzima Alicin javni ključ (n, e) i rekonstruira potpisanu poruku računajući

$$m = s^e \bmod n.$$

Ova jednakost slijedi iz Teorema (2.3.4). Sada Bob zna sadržaj potpisane poruke m i siguran je da ju je poslala Alice jer pri rekonstrukciji poruke koristio njezin javni ključ (n, e) koji odgovara samo njezinom tajnom ključu d, koji posjeduje samo ona.

Primjer 3.2.1. *Alice odabire $p = 29$, $q = 17$, $e = 3$. Eksponent 3 je dobro izabran jer zadovoljva uvjet $(e, (p-1)(q-1)) = 1$. Zatim računa $n = 493$. Sada je na redu računanje tajnog ključa - eksponenta d. On zadovoljava linearnu kongruenciju*

$$3d \equiv 1 \pmod{\varphi(493)},$$

odnosno

$$3d \equiv 1 \pmod{28 \cdot 16}.$$

Budući smo eksponent, odnosno koeficijent 3 odabrali tako da bude relativno prost s modulom znamo da će ova kongruencija imati jedinstveno rješenje. Rješavamo ju primjenom proširenog Euklidovog algoritma:

$$448 = 3 \cdot 149 + 1$$

$$149 = 1 \cdot 149$$

i	-1	0	1
q_i			149
y_i	0	1	-149

Dakle, rješenje kongruencije $3d \equiv 1 \pmod{448}$ je $d \equiv -149 \equiv 299 \pmod{448}$.

Konačno, Alicein javni ključ je $(493, 3)$, a tajni ključ 299.

Pretpostavimo da Alice želi obaviti bankovnu transakciju u visini 250 $. Ona potpisuje 250 na način da računa

$$s = 250^{299} \mod 493 = 159.$$

Po primitku njezine poruke, aplikacija će izračunati $m = 159^3 \pmod{493} = 250$ i provest će transakciju.

Implementacija RSA potpisa u programskom jeziku *Python*:

```
from math import *

def prost(n):
    for i in range(2, round(n ** (1 / 2) +1)):
        if n % i == 0:
            return False
    if n > 1:
        return True
    else:
        return False

def relativno_prosti(a, b):
    while a != b:
        if a > b:
            a -= b
        else:
            b -= a
    if b == 1:
        return True
    else:
        return False


def moguci_e(p, q):
    x = (p-1)*(q-1)
    e = []
    for i in range(2, x):
        if relativno_prosti(i, x) == True:
            e = e + [i]
    return e


def izracunaj_d(k, p):
    p2 = p
    y0, y1 = 0, 1
    while k != 0:
        q, p, k = p // k, k, p % k
        y0, y1 = y1, y0 - q * y1
```

```
    while y0 < 0:
        y0 += p2

    return y0


def generiranje_potpisa(m, d, n):
    s = (m ** d) % n
    return s

def rekonstrukcija_poruke(s, e, n):
    m2 = (s ** e) % n
    return m2

def provjera_valjanosti(m, m2):
    if m == m2:
        print('Potpis je valjan!')
    else:
        print('Potpis nije valjan!')


def main():

    p = int(input('Unesite prvi prost broj: '))
    if(prost(p) == False):
        print('Nevaljan unos!')
        exit()

    q = int(input('Unesite drugi prost broj: '))
    if(prost(q) == False):
        print('Nevaljan unos!')
        exit()

    n = p * q

    print('Moguce vrijednosti od e: ', moguci_e(p, q))
    e = int(input('Unesi e: '))
```

```
    fi = (p-1) * (q-1)
    d = izracunaj_d(e, fi)
    print('d = ', d)

    print('Vas javni kljuc je: ({0}, {1})'.format(n,e))
    print('Vas tajni kljuc je: ', d)

    m = int(input('Unesite poruku: '))

    s = generiranje_potpisa(m, d, n)

    print('Vas digitalni potpis je: ', s)

    m2 = rekonstrukcija_poruke(s, e, n)
    print('Rekonstruirana poruka je: ', m2)

    provjera_valjanosti(m, m2)


main()
```

Primjer izvršavanja programa:

```
Unesite prvi prost broj: 29
Unesite drugi prost broj: 17

Moguce vrijednosti od e:  [3, 5, 9, 11, 13, 15, 17, 19, 23,
25, 27, 29, 31, 33, 37, 39, 41, 43, 45, 47, 51, 53, 55, 57,
59, 61, 65, 67, 69, 71, 73, 75, 79, 81, 83, 85, 87, 89, 93,
95, 97, 99, 101, 103, 107, 109, 111, 113, 115, 117, 121, 123,
125, 127, 129, 131, 135, 137, 139, 141, 143, 145, 149, 151,
153, 155, 157, 159, 163, 165, 167, 169, 171, 173, 177, 179,
181, 183, 185, 187, 191, 193, 195, 197, 199, 201, 205, 207,
209, 211, 213, 215, 219, 221, 223, 225, 227, 229, 233, 235,
237, 239, 241, 243, 247, 249, 251, 253, 255, 257, 261, 263,
265, 267, 269, 271, 275, 277, 279, 281, 283, 285, 289, 291,
293, 295, 297, 299, 303, 305, 307, 309, 311, 313, 317, 319,
321, 323, 325, 327, 331, 333, 335, 337, 339, 341, 345, 347,
349, 351, 353, 355, 359, 361, 363, 365, 367, 369, 373, 375,
```

```
377, 379, 381, 383, 387, 389, 391, 393, 395, 397, 401, 403,
405, 407, 409, 411, 415, 417, 419, 421, 423, 425, 429, 431,
433, 435, 437, 439, 443, 445, 447]
Unesite e: 3

d =  299

Vas javni kljuc je: (493, 3)
Vas tajni kljuc je:  299

Unesite poruku: 250

Vas digitalni potpis je:  159
Rekonstruirana poruka je:  250
Potpis je valjan!
```

Odabir p, q

Ako Oscar može faktorizirati RSA module, odnosno broj n tada može i izračunati tajni ključ d od Alice te može potpisivati dokumente u njezino ime. Zbog toga, p i q moraju biti odabrani na način da se n ne može lako faktorizirati. Za RSA kriptosustav, odabir p i q već je opisan u odsječku (2.3).

Napadi

Ako je RSA potpis implementiran kao što je opisano, postoje više vrsta mogućih napada. Kako bi provjerio potpis od Alice, Bob uzima njezin javni ključ. No, kako Bob može biti siguran da je to upravo njezin javni ključ? Ako napadač, Oscar, uspije zamijeniti javni ključ od Alice sa svojim vlastitim tako da Bob to ne primjeti, tada će on moći nešto potpisati u ime Alice. Zbog toga je bitno za Boba imati mogućnost uvjeriti se da javni ključ koji ima pripada upravo Alice. To je razlog zbog kojeg mora postojati treća, neutralna strana. U svrhu suzbijanja takve vrste napada nužno je uključivanje davatelja usluga certificiranja ili ovjerovitelja javnog ključa (engl. *certification authority*, CA). Nešto više o tome reći ću u posljednjem odjeljku.

Još jedan od mogućih napada je sljedeći. Oscar odabire cijeli broj $s \in \{0, ..., n-1\}$. On tvrdi da je s RSA potpis od Alice. Bob želi provjeriti potpis. On računa $m = s^e \bmod n$ i vjeruje da je Alice potpisala m. Ako ispada da je m smisleni tekst, tada je Oscar uspješno

krivotvorio potpis od Alice. To se naziva *egzistencijalni falsifikat* (od engl. existential forgery). To nije teško zamisliti na primjeru novčanih transakcija.

Primjer 3.2.2. *Kao u Primjeru (3.2.1) pretpostavimo da je javni ključ od Alice* $(493, 3)$*, a tajni ključ* 299*. Oscar želi povući novac s njezinog računa. On šalje potpis* $s = 139$ *bankomatu. Uređaj će zatim izračunati* $m = 139^3 \mod 493 = 248$ *te će vjerovati da Alice želi podignuti $ 248, ali to nije istina.*

Sljedeći mogući napad proizlazi iz činjenice da je šifriranje u RSA kriptosustavu multiplikativna operacija. Ako su $m_1, m_2 \in \{0, \ldots, n-1\}$ i

$$s_1 = m_1^d \mod n, \quad s_2 = m_2^d \mod n$$

potpisi od m_1 i m_2, tada je

$$s = s_1 s_2 \mod n$$

potpis od $m = m_1 m_2$. Zaista,

$$s_1 s_2 \equiv m_1^d m_2^d = (m_1 m_2)^d \pmod{n}.$$

Iz dva valjana RSA potpisa može se izračunati treći. Napadač može koristiti multiplikativnost RSA potpisa kako bi krivotvorio potpis za neki dokument $m \in \{0, ..., n-1\}$. To može napraviti na sljedeći način. Pretpostavimo da napadač ima u posjedu potpisanu poruku $m_1 \in \{0, ..., n-1\}$ različitu od m i takvu da je $\gcd(m_1, n) = 1$. Za takav m_1 postoji multiplikativni inverz modulo n. Označimo ga s m_1^{-1}. Pretpostavimo nadalje da napadač može doći u posjed digitalnog potpisa sljedeće specifične poruke:

$$m_2 = m m_1^{-1} \bmod n,$$

Znači, napadač dobiva valjane RSA potpise s_1 i s_2 za m_1 i m_2. Zatim računa potpis od m: $s = s_1 s_2 \bmod n$. Prema tome, shema RSA potpisa, ovako kako smo ju predstavili, ne štiti od napada.

Potpis s redundancijom

Dva od tri napada iz prethodne sekcije nemoguća su ukoliko poruka $m \in \{0, 1, \ldots, n-1\}$ koju treba potpisati ima oblik $w \circ w$ u binarnom zapisu. Dakle, $w \in \{0,1\}^*$ i m u binarnom zapisu ima dvije identične polovice. U ovom slučaju, zapravo se potpisuje w, ali tehnički potpisat će se $w \circ w$. Ovakav potpis nazivamo *potpis s redundancijom* ili *potpis sa zalihom*. Pri verifikaciji potpisa, Bob će izračunati $m = s^e \mod n$ i provjeriti je li dobivena poruka oblika $w \circ w$. Ako nije, Bob će odbaciti potpis.

Ako se potpisuju samo dokumenti oblika $w \circ w$, onda se uvelike smanjuje mogućnost za napad "egzistencijalni falsifikat". Oscar bi u tom slučaju morao poslati lažni potpis $s \in \{0, 1, \ldots, n-1\}$ takav je binarni zapis od $m = s^e \mod n$ oblika $w \circ w$. Nije poznato kako bi se mogao konstruirati takav potpis bez poznavanja tajnog ključa.

Multiplikativnost RSA također se ne može iskoristiti jer je vrlo malo vjerojatno da je $m = m_1 m_2 \mod n$ oblika $w \circ w$ ukoliko su i m_1 i m_2 tog oblika.

Funkcija

$$R : \{0,1\}^* \to \{0,1\}^*, w \mapsto R(w) = w \circ w,$$

koja se upotrebljava za generiranje specijalne strukture dokumenta koji se potpisuje naziva se *redundantna funkcija*.

Potpis primjenom hash funkcije

Pri generiranju RSA potpisa mogu se koristiti kriptografske hash funkcije. Hash funkcija H je funkcija koja za ulazni podatak x (datoteku, poruku ili slično) proizvoljne veličine računa vrijednost unaprijed određene veličine, obično izražene u bitovima. Vrijednost $H(x)$ kratko nazivamo *hash* od x. Uobičajeno je vrijednosti hash funkcija zapisivati u heksadecimalnom obliku.

Često se koristi usporedba *hasha* s otiscima prstiju i ta se ideja koristi u raznim kriptografskih shemama, ponajprije kod digitalnog potpisa gdje hash funkcije igraju ulogu "vezivnog tkiva" između poruke i potpisa.

U kriptografiji se koriste hash funkcije H koje moraju zadovoljiti sljedeća svojstva:

1. H radi s blokovima proizvoljne veličine;
2. izlaz od H je fiksne duljine (u bitovima);
3. za svaki x je relativno lako izračunati $H(x)$;
4. za zadanu vrijednost h efektivno je nemoguće naći x takav da je
$$H(x) = h;$$
5. za zadni x efektivno je nemoguće naći y takav da je
$$H(x) = H(y);$$

6. efektivno je nemoguće naći par (x, y) tako da je

$$H(x) = H(y).$$

Hash funkcije koje zadovoljavaju svojstva (1)-(6) nazivamo *kriptografskim hash funkcijama*.

Primjer 3.2.3. *Pretpostavimo da su naše poruke oblika cjelobrojne uređene trojke čija je svaka koordinata veća od* 10^6. *Nadalje, pretpostavimo da takvu trojku ne možemo pohraniti u memoriju te ju moramo smanjiti na neku prihvatljivu veličinu, na primjer do približne veličine* 10^6. *Funkcija koja će odigrati ulogu hash funkcije je*

$$h : \mathbb{Z}^3 \to \mathbb{Z}_p$$

$$h(x, y, z) = x \cdot a^2 + y \cdot a + z \mod p,$$

pri čemu je p prost. Na primjer, $p = 1000003$. *Koeficijent a je slučajno odabran cijeli broj iz* $\mathbb{Z}_p$, *preciznije iz skupa* $\{2, \ldots, p-1\}$ *jer* $a = 0$ *i* $a = 1$ *nisu povoljno odabrane vrijednosti.*

Ustanovimo da je $h(x_0, y_0, z_0) \neq h(x_1, y_1, z_1)$ *s prilično velikom vjerojatnošću, ako je* $(x_0, y_0, z_0) \neq (x_1, y_1, z_1)$. *Pretpostavimo suprotno, to jest* $h(x_0, y_0, z_0) = h(x_1, y_1, z_1)$. *Tada je*

$$(x_0 - x_1) \cdot a^2 + (y_0 - y_1) \cdot a + (z_0 - z_1) \mod p = 0.$$

Dakle, $a \in \mathbb{Z}_p$ *je rješenje polinomijalne kongruencije*

$$f(t) = (x_0 - x_1)t^2 + (y_0 - y_1)t + (z_0 - z_1) \equiv 0 \pmod{p}.$$

Prema Lagrangeovom teoremu, ako vodeći koeficijent od f nije djeljiv s p, onda kongruencija $f(t) \equiv 0 \pmod{p}$ *ima najviše 2 rješenja modulo p. Stoga je vjerojatnost da je proizvoljno odabrani a rješenje ove konguencije,odnosno nultočka polinoma f manja od* $2 \cdot 10^{-6}$.

Ako Alice želi potpisati proizvoljno dugačak dokument x, tada će upotrijebiti javno poznatu hash funkciju

$$h : \{0, 1\}^* \to \{0, \ldots, n-1\}.$$

Potpis dokumenta x tada je

$$s = h(x)^d \mod n.$$

Iz ovog potpisa može se rekonstruirati samo vrijednost $h(x)$, ali ne i dokument x. Bob će moći provjeriti potpis od x samo ako mu je poznat dokument x. Nakon što Alice izračuna

potpis od x, šalje ga Bobu zajedno s dokumentom x. Bob zatim računa $m = s^e \mod n$ i uspoređuje ga s hash vrijednošću od x. Kako je hash funkcija javna, Bob tu vrijednost lako može izračunati. Ako su m i x jednaki, Bob prihvaća potpis. U suprotnom ga odbacuje.

Korištenje hash funkcije u sklopu generiranja digitalnog potpisa onemogućava neke od napada. Multiplikativnost više nije slaba točka jer je uglavnom nemoguće naći x takav da je $h(x) = m = m_1 m_2 \mod n$. Nadalje, nije moguće Alice podvaliti potpisivanje dokumenta x' umjesto x jer $h(x) \neq h(x')$.

3.3 ElGamalov potpis

Shema ElGamalovog potpisa slična je ElGamalovom kriptosustavu. Njegova sigurnost temelji se na teškoći računanja diskretnog logaritma u $(\mathbb{Z}_p^*, \cdot_p)$, gdje je p prost broj.

Generiranje ključeva

Generiranje ključeva jednako je kao u ElGamalovom kriptosustavu. Alice odabire veliki prosti broj p i primitivni korijen g modulo p. Također, odabire $a \in \{1, 2, \ldots, p-2\}$ i računa $A = g^a \bmod p$. Njezin privatni ključ je a, a javni ključ (p, g, A).

Generiranje potpisa

Alice potpisuje dokument $x \in \{0,1\}^*$. Pritom koristi javno poznatu, na kolizije otpornu, hash funkciju

$$h : \{0,1\}^* \to \{1, 2, \ldots, p-2\}.$$

Alice odabire nasumični broj $k \in \{1, 2, \ldots, p-2\}$ koji je relativno prost sa $p-1$. Zatim računa

$$r = g^k \mod p, \quad s = k^{-1}(h(x) - ar) \mod (p-1), \tag{3.1}$$

gdje je k^{-1} multiplikativni inverz od k modulo $p-1$. Potpis dokumenta x tada je par (r, s). Kako je Alice koristila hash funkciju, osoba koja će provjeravati potpis neće moći iz tog potpisa doći do dokumenta x. Njega joj Alice također treba proslijediti.

Verifikacija potpisa

Bob pri verifikaciji potpisa koristi javni ključ (p, g, A) od Alice. Kao i kod RSA sheme potpisa, mora se uvjeriti u autentičnost ovog javnog ključa. Provjerava je li zadovoljeno

$$1 \leq r \leq p-1.$$

Ako ovaj uvjet nije zadovoljen, tada odbacuje potpis. Nadalje, provjerava se vrijedi li kongruencija

$$A^r r^s \equiv g^{h(x)} \pmod p.$$

Bob prihvaća potpis ukoliko je ova kongruencija zadovoljena. U protivnom ga odbacuje.

Pokažimo da je ova verifikacija zaista valjana. Ako su r i s izračunati prema (3.1), tada vrijedi

$$A^r r^s \equiv g^{ar} g^{kk^{-1}(h(x)-ar)} = g^{ar+h(x)-ar} = g^{h(x)} \pmod{p}$$

Obratno, ako je za par (r, s) zadovoljeno

$$A^r r^s \equiv g^{h(x)} \pmod{p}$$

i ako je k diskretni logaritam od r po bazi g, tada

$$g^{ar+ks} \equiv g^{h(x)} \pmod{p}.$$

Kako je g primitivni korijen modulo p, slijedi

$$ar + ks \equiv h(x) \pmod{p-1}.$$

Ako su k i $p-1$ relativno prosti, to povlači

$$s = k^{-1}(h(x) - ar) \pmod{p-1}.$$

Primjer 3.3.1. *Kao u Primjeru (2.5.2), Alice odabire* $p = 17, g = 3, a = 5$ *i računa* $A = g^a \mod p = 5$. *Njezin javni ključ je* $(p = 17, g = 3, A = 5)$. *Njezin tajni ključ je* $a = 5$. *Alice želi potpisati dokument x, čija je vrijednost* $h(x) = 9$. *Odabire* $k = 7$ *i dobiva* $r = 11$. *Inverz od k modulo* $p - 1 = 16$ *je* $k^{-1} = 7$. *Stoga je*

$$s = k^{-1}(h(x) - ar) \mod (p-1) = 7 \cdot (9 - 5 \cdot 11) \mod 16 = 14.$$

Tada je potpis (11, 14). Bob želi provjeriti potpis. Računa

$$A^r r^s \mod p = 5^{11} \cdot 11^{14} \mod 17 = 14,$$

te

$$g^{h(x)} \mod p = 3^9 \mod 17 = 14,$$

pa zaključuje da je potpis valjan.

Implementacija ElGamalovog potpisa u programskom jeziku *Python*:

```
from math import *

def prost(n):
    for i in range(2, round(n ** (1 / 2) +1)):
        if n % i == 0:
            return False
    if n > 1:
        return True
    else:
        return False

def relativno_prosti(a, b):
    while a != b:
        if a > b:
            a -= b
        else:
            b -= a
    if b == 1:
        return True
    else:
        return False

def moguci_g(p):
    g = []
    fi = p - 1
    nasao = 0
    for i in range(2, p):
        if relativno_prosti(i, p) == True:
            for j in range(1, p):
                if (i**j) % p == 1:
                    nasao = 1
                    break
            if nasao == 1:
                if j == fi:
                    g = g + [i]
            nasao = 0
    return g
```

```
def moguci_k(p):
    k = []
    for i in range(1, p-1):
        if relativno_prosti(i, p-1) == True:
            k = k + [i]
    return k

def mult_inverz(k, p):
    p2 = p
    y0, y1 = 0, 1
    while k != 0:
        q, p, k = p // k, k, p % k
        y0, y1 = y1, y0 - q * y1

    while y0 < 0:
        y0 += p2

    return y0



def main():

    p = int(input('Unesite prosti broj: '))
    if(prost(p) == False):
        print('Nevaljan unos!')
        exit()

    print('Moguce vrijednosti od g: ', moguci_g(p))
    g = int(input('Unesi g: '))

    a = int(input('Unesite prirodan broj manji ili
                jednak {0}: '.format(p-2)))

    A = (g ** a) % p

    print('Vas javni kljuc je: ({0}, {1}, {2})'.format(p,g,A))
    print('Vas tajni kljuc je: ', a)
```

```
    h = int(input('Unesite vrijednost h(x) poruke x: '))

    print('Moguce vrijednosti od k: ', moguci_k(p))
    k = int(input('Unesite k: '))

    r = (g ** k) % p

    inverz = mult_inverz(k, p-1)
    s = inverz * (h - a*r) % (p-1)

    print('Vas potpis je: ({0}, {1})'.format(r, s))

    if 1 > r or r > p-1:
        print('Potpis nije valjan!')

    if ((A ** r) * (r ** s) - (g ** h)) % p == 0:
        print('Potpis je valjan!')
    else:
        print('Potpis nije valjan!')

main()
```

Primjer izvršavanja programa:

```
Unesite prosti broj: 17

Moguce vrijednosti od g:  [3, 5, 6, 7, 10, 11, 12, 14]
Unesite g: 3

Unesite prirodan broj manji ili jednak 15: 5

Vas javni kljuc je: (17, 3, 5)
Vas tajni kljuc je:  5

Unesite vrijednost h(x) poruke x: 9

Moguce vrijednosti od k:  [1, 3, 5, 7, 9, 11, 13, 15]
Unesite k: 7
```

```
Vas potpis je: (11, 14)

Potpis je valjan!
```

Odabir p, k

Ako napadač, Oscar, može izračunati diskretni logaritam modulo p, tada može odrediti tajni ključ od Alice i generirati potpise u njezino ime. Zato p mora biti odabran na način da je računanje diskretnog logaritma mod p neisplativo. S obzirom na danas poznate algoritme za računanje diskretnog logaritma, to znači da bi p trebao imati više od 200 znamenki. Također, neke proste brojeve specijalnih formi za koje neki algoritmi mogu biti djelomično efikasni također treba izbjegavati. Najbolja strategija je koristiti nasumične proste brojeve.

Za svaki novi potpis potrebno je odabrati novi eksponent k. Pretpostavimo da su potpisi s_1 i s_2 dokumenata x_1 i x_2 generirani korištenjem istog k. Tada je broj $r = g^k \bmod p$ jednak za oba potpisa. Zato vrijedi

$$s_1 - s_2 \equiv k^{-1}(h(x_1) - h(x_2)) \pmod{(p-1)}.$$

Ako je $h(x_1) - h(x_2)$ invertibilan modulo $p - 1$, iz ove kongruencije može se odrediti k. Iz $k, s_1, r, h(x_1)$, tajni ključ a od Alice može se odrediti jer vrijedi

$$s_1 = k^{-1}(h(x_1) - ar) \bmod (p-1)$$

te je zato

$$a \equiv r^{-1}(h(x_1) - ks_1) \pmod{(p-1)}.$$

Efikasnost

Generiranje ElGamalovog potpisa zahtijeva jednu primjenu proširenog Euklidovog algoritma za izračunavanje $k^{-1} \bmod (p-1)$ i jedno modularno potenciranje modulo p za računanje $r = g^k \bmod p$. Ovo su mogući predračuni. Oni ne ovise o dokumentu koji se potpisuje. Vrlo je važno ove predračune sigurno spremiti i čuvati tajnima. Stvarni potpis u tom slučaju zahtijeva samo dva modularna multipliciranja te je vrlo brz. Verifikacija ElGamalovog potpisa zahtijeva tri modularna potenciranja, što je znatno dulje nego RSA verifikacija potpisa.

Napadi

Ako se pri generiranju ElGamalovog potpisa ne koristi hash funkcija, tada je moguće krivotvoriti potpis. Bez hash funkcije, kongruencija za verifikaciju potpisa je

$$A^r r^s \equiv g^x \pmod{p}.$$

U nastavku pokazujemo kako se mogu odabrati r, s, x kako bi ova kongruencija bila zadovoljena. Kako bi krivotvorio potpis, Oscar odabire dva cijela broja u, v sa svojstvom $\gcd(v, p-1) = 1$. Zatim računa

$$r = g^u A^v \bmod p, \qquad s = -rv^{-1} \bmod (p-1), \qquad x = su \bmod (p-1).$$

Tada vrijedi kongruencija

$$A^r r^s \equiv A^r g^{su} A^{sv} \equiv A^r g^{su} A^{-r} \equiv g^x \pmod{p}.$$

Ova procedura funkcionira i u slučaju kada je upotrijebljena hash funkcija otporna na kolizije. Ali kako je hash funkcija jednosmjerna, Oscaru je nemoguće naći dokument x takav da generirani potpis bude potpis od x.

Uvjet $1 \leq r \leq p-1$ je također vrlo važan. U protivnom, moguće je generirati nove potpise koristeći stare potpise, kao što ću sada pokazati. Neka je (r, s) ElGamalov potpis dokumenta x. Neka je x' neki drugi dokument. Kako bi potpisao x', Oscar računa

$$u = h(x')h(x)^{-1} \bmod (p-1).$$

Ovdje pretpostavljamo da je $h(x)$ invertibilan modulo $p-1$. Nadalje, Oscar računa

$$s' = su \bmod (p-1)$$

i koristeći Kineski teorem o ostacima određuje r' pomoću

$$r' \equiv ru \pmod{(p-1)}, \qquad r' \equiv r \pmod{p}.$$

Potpis od x' sada je (r', s'). Verifikacija ovog potpisa funkcionira jer

$$A^{r'}(r')^{s'} \equiv A^{ru} r^{su} \equiv g^{u(ar+ks)} \equiv g^{h(x')} \pmod{p}.$$

Nadalje, pokazujem da je $r' \geq p$, što znači da je uvjet $1 \leq r' \leq p-1$ narušen. S jedne strane, imamo

$$1 \leq r \leq p-1, \qquad r \equiv r' \pmod{p}, \tag{3.2}$$

a s druge strane

$$r' \equiv ru \not\equiv r \pmod{p-1}. \tag{3.3}$$

Ovo slijedi iz $u \equiv h(x')h(x)^{-1} \not\equiv 1 \pmod{p-1}$ i iz činjenice da je h otporna na kolizije, što znači da je za zadani x efektivno nemoguće naći y takav da je $h(x) = h(y)$. Sada (3.3) implicira $r \neq r'$ a (3.2) povlači da je $r' \geq p$.

3.4 Digital Signature Algorithm

Digital Signature Algorithm predložen je i standardiziran od strane NIST-a (National Institute of Standards and Technology). DSA je algoritam za digitalni potpis razvijen po uzoru na ElGamalov kriptosustav digitalnog potpisa, odnosno temelji se na teškoći računanja diskretnog logaritma. Ovaj algoritam također umjesto cijele poruke m za šifriranje koristi samo hash vrijednost $h(m)$.

DSA funkcionira na sljedeći način. Za neki dokument računa se vrijednost hash funkcije i zatim kriptira privatnim ključem pošiljatelja. Nakon toga, dokument i potpis prenose se primatelju koji će verificirati potpis pomoću javnog ključa pošiljatelja.

Generiranje ključeva

Alice odabire prosti broj q takav da je

$$2^{159} < q < 2^{160}.$$

Prema tome, q ima binarni zapis duljine 160. Nadalje, Alice odabire veliki prosti broj p sa sljedećim svojstvima:

- $2^{511+64t} < p < 2^{512+64t}$ za neki $t \in \{0, 1, \ldots, 8\}$;
- prosti broj q, koji je prethodno odabran, dijeli $p - 1$.

Duljina binarnog zapisa broja p je između 512 i 1024 i višekratnik je broja 64. Prema tome, binarni zapis broja p je slijed od 8 do 16 bitstringova duljine 64. Uvjet $q|(p-1)$ povlači da grupa $(\mathbb{Z}_p^*, \cdot_p)$ sadrži elemente reda q.

Zatim Alice odabire primitivni korijen x modulo p i računa

$$g = x^{(p-1)/q} \mod p.$$

Tada je $g + p \cdot k$ reda q u $(\mathbb{Z}_p^*, \cdot_p)$, za sve $k \in \mathbb{Z}$. Zaista,

$$(g + p \cdot k)^q = (x^{(p-1)/q} + p \cdot k)^q \equiv (x^{(p-1)/q})^q \equiv x^{p-1} \equiv 1 \pmod p.$$

Konačno, Alice odabire nasumični broj $a \in \{1, 2, \ldots, q-1\}$ i računa

$$A = g^a \mod p.$$

Njezin javni ključ tada je (p, q, g, A), a tajni ključ a.

Generiranje potpisa

Alice želi potpisati dokument x. Koristi javno poznatu hash funkciju

$$h : \{0,1\}^* \to \{1,2,\ldots,q-1\}.$$

Odabire nasumični broj $k \in \{1,2,\ldots,q-1\}$, računa

$$r = (g^k \mod p) \mod q, \tag{3.4}$$

i zatim

$$s = k^{-1}(h(x) + ar) \mod q \tag{3.5}$$

Ovdje je k^{-1} multiplikativni inverz od k modulo q. Sada je potpis (r, s).

Verifikacija potpisa

Bob želi provjeriti potpis (r, s) dokumenta x. Uzima javni ključ od Alice (p, q, g, A) i javno poznatu hash funkciju i provjerava je li zadovoljeno

$$1 \le r \le q-1 \quad i \quad 1 \le s \le q-1. \tag{3.6}$$

Ukoliko ovaj uvjet nije zadovoljen, Bob odbacuje potpis. U protivnom, provjerava jednakost

$$r = ((g^{(s^{-1}h(x)) \mod q} A^{(rs^{-1}) \mod q}) \mod p) \mod q. \tag{3.7}$$

Ukoliko je potpis konstruiran prema (3.4) i (3.5), tada je (3.7) zadovoljeno. Zaista,

$$g^{(s^{-1}h(x)) \mod q} A^{(rs^{-1}) \mod q} \equiv g^{s^{-1}(h(x)+ra) \mod q} \equiv g^k \mod p,$$

što implicira (3.7).

Efikasnost

DSA je vrlo sličan shemi ElGamalovog digitalnog potpisa. Kao i kod ElGamala, predračuni čine generiranje potpisa mnogo bržim.

DSA verifikacija mnogo je efikasnija od ElGamalove verifikacije. S jedne strane, kod DSA potrebna su samo dva modularna potenciranja, dok su kod ElGamalovog digitalnog potpisa potrebna tri modularna potenciranja. No, mnogo je važnija činjenica da su eksponenti u DSA 160-bitni brojevi, dok su kod ElGamala mnogo veći (najmanje 768-bitni brojevi). Ovo skraćuje račun za više od 700 kvadriranja i množenja modulo p.

Napadi

Kao i kod ElGamalove sheme, potrebno je odabrati novi eksponent k za svaki novi potpis. Štoviše, upotreba hash funkcije i provjera uvjeta $1 \leq r \leq q-1$ i $1 \leq s \leq q-1$ nužna je kako bi se izbjegle krivotvorine.

Ako Oscar može izračunati diskretni logaritam u podgrupi H grupe $(\mathbb{Z}_p^*, \cdot_p)$ generirane sa $g + p\mathbb{Z}$, tada može izračunati Alicin tajni ključ a iz njezinog javnog ključa i moći će potpisivati dokumente u njezino ime. Ovo je za sada jedini poznati napad na DSA. Stoga sigurnost leži na određivanju diskretnog logaritama u podgrupi H.

3.5 Problem identiteta

Komunikacija kojom se razmjenjuju poruke povjerljivog sadržaja kao što su bankovne transakcije ili razmjena osobnih podataka mora zadovoljavati nekoliko osnovnih zahtjeva. Prvi zahtjev je vjerodostojnost, odnosno mogućnost da primatelj poruke pouzdano utvrdi identitet pošiljatelja poruke. Drugi zahtjev je netaknutost, odnosno mogućnost da primatelj poruke utvrdi da se poruka koju je poslao pošiljatelj nije promijenila prilikom slanja. Treći zahtjev je neporecivost, odnosno sprečavanje pošiljatelja poruke u opovrgavanju činjenice da je on poslao poruku. Četvrti zahtjev je povjerljivost kojom se osigurava tajnost sadržaja poruke za svakoga kome poruka nije namijenjena.

Kriptosustavi s javnim ključem imaju nedostatak koji smo dosad ignorirali - problem povezivanja ključa i osobe, odnosno pitanje identiteta ključa, što je u praksi problematično. Dosad smo u kriptografskim shemama prešutno poistovjećivali ključeve s osobama. Nismo razmatrali da se Oscar lažno predstavi s pseudonimom Bob. U takvoj situaciji Alice može misliti da je šifrirala poruku za Boba, no tu poruku može pročitati jedino Oscar. Bob je čak ne može ni dešifrirati budući da ne posjeduje pripadni tajni ključ.

Slično, uspješna provjera digitalnog potpisa nas još uvijek ne uvjerava da je poruku potpisala Alice, već samo da je poruka potpisana tajnim ključem koji odgovara javnom ključu za koji vjerujemo da pripada Alice. Od kriptosustava s javnim ključem imamo koristi tek ako smo uspješno razmjenili ključeve i imamo povjerenja u identitet osobe koja posjeduje ključ.

Jedno rješenje ovog problema daje sustav protokola koji nazivamo *infrastruktura javnog ključa* (engl. Public Key Infrastructure, kratko PKI) u kojem treća osoba od povjerenja jamči za identitete osoba, odnosno ključeva. Ovakav autoritet koji jamči za identitete klijenata naziva se *središnji autoritet*, odnosno engl. *certificate authority*, što se kratko označava s CA. U duhu usporedbe papirnatih i digitalnih potpisa, možemo reći da CA za-

mjenjuje ulogu javnog bilježnika (notara) u digitalnom svijetu.

CA djeluje izdavanjem *certifikata* - digitalno potpisanih poruka u kojoj se nalaze osobni podaci nositelja javnog ključa i sam javni ključ. Certifikat potpisuje CA svojim ključem Javni ključ od CA jedini je javni ključ u koji trebamo imati povjerenja. U identitete ostalih ključeva imamo povjerenja ako za njih postoji certifikat.

Prema ovome, za pokretanje sustava dovoljno je da CA izda certifikat za javni ključ svakog korisnika sustava. Uobičajeno je, iz praktičnih razloga, uspostaviti više CA-ova, ponekad i u hijerarhiji. Javni ključ se smatra valjanim ako postoji *lanac certifikata* koji doseže do CA.

Npr. Bob uz ključ od Alice dobije certifikat koji je potpisao CA_1. Za CA_1, Bob može pronaći certifikat koji je potpisao CA_0 kojem on vjeruje. U konkretnim PKI sustavima pravo izdavanja certifikata nije dano svim korisnicima već samo nekolicini, baš kao što ni u "papirnatom" svijetu nije svatko javni bilježnik.

Upravo postojanje lanca certifikata omogućuje da korisnici koji nisu poznati jedni drugima uspostave povjerljivu vezu sa CA. CA obavlja autentifikaciju korisnika, objavljuje njegov certifikat koji je potpisao svojim tajnim ključem te time jamči vjerodostojnost identiteta vlasnika tih certifikata.

Nakon što dvije strane razmijene certifikate, slijedi provjera njihove valjanosti. Ukoliko su oni valjani, tada su i informacije koje oni sadrže valjane. Valjanost certifikata provjerava se tako da se provjeri digitalni potpis koji je stvorio CA, a kojem oba korisnika vjeruju, na način da se javnim ključem od CA dekriptira digitalni potpis. Dobiveni tekst uspoređuje se s izračunatim sažetkom ostatka certifikata. Ukoliko se dobivene vrijednosti podudaraju, tada je certifikat valjan te možemo biti sigurni da nitko nije mijenjao njegov sadržaj. Kako nitko osim CA ne zna njegov privatni ključ, samo je CA mogao napraviti taj digitalni potpis te su stoga informacije koje sadrži certifikat vjerodostojne.

PKI ima raširenu primjenu u povezivanju ključeva i identiteta, te se koristi u raznim sustavima:

1. za šifriranje, dešifriranje i potpisivanje e-mail poruka;
2. za provjeru identiteta (autentifikaciju) korisnika (npr. kod prijava u sustav).

Alternativno rješenje problema identiteta je *mreža povjerenja* (engl. web of trust) kakvu koriste OpenPGP kompatibilni sustavi. U mreži povjerenja korisnici jamče jedni za druge.

Možemo ugrubo reći da Alice može vjerovati Bobu ako Alice i Bob imaju zajedničkog prijatelja Charlesa koji je potpisao Bobov ključ. Može se reći da se u mreži povjerenja povjerenje širi ”prijateljski”, a ne ”službeno” kao u PKI infrastrukturi.

Iako digitalni certifikati imaju unaprijed definirani vremenski interval valjanosti, postoje situacije kada se više u njihovu valjanost ne može vjerovati te se vjerodajnice moraju proglasiti nevaljanim. Budući da se sve operacije obavljaju na računalu, moguće je da uporaba ključa na ”provaljenom” računalu kompromitira javni ključ. Uz viruse, crve, trojanske konje i bug-ove u operativnim sustavima, vjerojatnost za takvo što je stvarna. U tom slučaju, nakon što se otkrije da je tajni ključ kompromitiran, CA povlači svoj certifikat. U PKI infrastrukturi za to postoje *Certificate Revocation Lists* (kratko CRL) - lista povučenih i nevažećih certifikata. Nju generira CA, a sadrži jedinstvene informacije o opozvanim certifikatima što omogućuje entitetima da provjere je li vjerodajnica valjana ili nije.

Uz korištene liste opozvanih digitalnih certifikata vezani su određeni problemi. Kad god neki korisnik želi koristiti certifikat, mora se uvjeriti da taj certifikat nije opozvan. Ako iz nekog razloga ne uspije izvršiti tu provjeru, vjerodajnica može biti protumačena kao valjana. To znači da se za efektivno korištenje infrastrukture javnih ključeva mora imati stalna veza s listom opozvanih digitalnih certifikata. Ovaj zahtjev kontradiktoran je sa svojstvom certifikata da su ”samoautentificirajući”.

Nadalje, ako netko slučajno opozove krivi certifikat, valjani certifikat postaje neupotrebljiv. Kako se konstantno mora provjeravati lista opozvanih digitalnih certifikata, javlja se potencijalna opasnost od uskraćivanja usluge ako netko onesposobi listu opozvanih digitalnih certifikata.

Bibliografija

[1] J. A. Buchmann, *Introduction to cryptography*, Springer-Verlag, New York, 2004.

[2] A. Dujella, M. Maretić,, *Kriptografija*, Element, Zagreb, 2007.

[3] D. R. Stinson, *Cryptography: Theory and Practice*, CRC Press, Boca Raton, 1995.

[4] A. Y. Yan, *Number Theory for Computing*, Springer-Verlag, Berlin, 2002.

[5] D. Čuljak, Infrastruktura javnih ključeva u prividnoj mreži računalnih sustava zasnovanih na uslugama, *Diplomski rad* `https://bib.irb.hr/datoteka/330872.DiplomskiRad_DavorCuljak.pdf`, (svibanj, 2016.)

[6] Department of Mathematics University of Zagreb, A. Dujella: `http://web.math.pmf.unizg.hr/~duje/utb/utblink.pdf`, (lipanj, 2016.)

Sažetak

U današnje vrijeme život više nije moguće zamisliti bez poslovanja na globalnoj razini, odnosno online. Tvrtke usvajaju nove tehnologije, kao što su rad na daljinu, internet bankarstvo i elektroničke dokumente. Poslovanje preko interneta i potpisivanje elektroničkih dokumenata zahtijeva sve više sigurnosti.

Shema digitalnog potpisa metoda je potpisivanja dokumenata u elektroničkoj formi. Kao takav, potpisani dokument može se na siguran način prenijeti preko računalne mreže. Digitalni potpis temelji se na kriptografiji javnog ključa, to jest, na teškoći rješavanja nekih matematičkih problema, kao što su faktorizacija velikih prirodnih brojeva i računanje diskretnog logaritma u $(\mathbb{Z}_p^*, \cdot_p)$, gdje je p prost broj.

Summary

Nowadays, the speed of business connections is increasing rapidly. Companies are adopting new technologies such as distant work places, internet banking and usage of electronic documents. Doing business via internet or signing e-documents requires more security.

A digital signature sheme is a method of signing a message stored in an electronic form. As such, a signed message can be transmitted over a computer network. The digital signature is based on a public key cryptography, that is, on the difficulty of solving some mathematical problems, such as finding the factorization of a composite positive integer that is the product of two large primes, or computing discrete logarithms in $(\mathbb{Z}_p^*, \cdot_p)$, where p is a prime number.

Životopis

Rođena sam 16.07.1992. godine u Čakovcu. Godine 1999. upisana sam u Osnovnu školu Donja Dubrava, a 2007. godine upisujem Gimnaziju u Čakovcu. Godine 2011. završavam Gimnaziju s izvrsnim uspjehom na državnoj maturi i upisujem sveučilišni preddiplomski studij Matematika; smjer: nastavnički na Prirodoslovno-matematičkom fakultetu u Zagrebu. Godine 2014. stječem naziv sveučilišna prvostupnica (baccalaurea) edukacije matematike. Iste godine, na istom fakultetu upisujem sveučilišni diplomski studij Matematika i informatika; smjer: nastavnički. Godine 2016. dobivam nagradu Fakultetskog vijeća PMF-a za izuzetan uspjeh u studiju.