Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Diplomski studij, smjer: Financijska matematika i statistika

Sanja Rendulić

Sheme digitalnog potpisa

Diplomski rad

Osijek, 2018.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku
Diplomski studij, smjer: Financijska matematika i statistika

Sanja Rendulić

Sheme digitalnog potpisa

Diplomski rad

Mentor: izv.prof.dr.sc. Ivan Matić

Osijek, 2018.

Sadržaj

1 UVOD

Sheme šifriranja, bez obzira jesu li simetrične ili asimetrične, riješavaju problem sigurne komunikacije preko nesigurnog komunikacijskog kanala. Digitalni potpisi rješavaju drugačiji problem, analogan svrsi potpisa olovkom i tintom na fizičkom dokumentu. Ono što je zanimljivo jest da je konstrukcija digitalnih potpisa veoma slična konstrukciji asimetričnih šifri. Recimo, imamo računalnu datoteku kao nekakav (digitalni) dokument. Želimo stvoriti dodatni podatak koji se može koristiti kao dokaz da odobravamo taj dokument. Tako dolazimo do digitalnog potpisa koji je analogan našem stvarnom potpisu na običnom papirnatom dokumentu. Unatoč njihovim različitim namjenama, sheme digitalnog potpisa su slične asimetričnim kriptosustavima po tome što uključuju javne i privatne ključeve te pozivaju algoritme koji ih koriste.

Digitalni potpisi i kriptosustavi javnog ključa su od jednake važnosti za vođenje poslovanja. Primjerice, naše računalo redovno prima obavijesti o ažuriranju programa i sustava putem interneta. Kako naše računalo može reći da obavijest o ažuriranju dolazi iz legitimnog izvora, odnosno tvrtke koja je stvarni vlasnik programa? Odgovor je digitalni potpis. Originalni program dolazi s javnim ključem verifikacije koji pripada tvrtki. Tvrtka koristi svoj privatni ključ za potpisivanje kojim potpisuje ažuriranje i šalje našem računalu novi program i pripadni potpis. Naše računalo može koristiti javni ključ verifikacije potpisa, čime se potvrđuje da program dolazi iz pouzdanog izvora prije instalacije na našem sustavu. Moramo, međutim, naglasiti da je primjena digitalnog potpisa ovim primjerom znatno pojednostavljena.

Stvarne aplikacije digitalnih potpisa zahtijevaju mnogo veću brigu oko izbjegavanja fatalnih sigurnosnih problema. Problem nam može predstavljati pitanje hoćemo li biti u potpunosti sigurni da navodni javni ključ verifikacije pripada pravom vlasniku. Ukoliko se dogodi to da nesvjesno upotrijebimo protivnikov ključ verifikacije umjesto stvarnog, potvrdit ćemo sve krivotvorene dokumente. U prvom dijelu rada detaljnije ćemo se baviti ovakvim problemima i načinima kako spriječiti iste. Nadalje, većina shema digitalnog potpisa potpisuje samo malu količinu podataka, tj. između 80 i 1000 bita. Vrlo je neučinkovito potpisati veliki digitalni dokument jer, osim što je potrebno mnogo vremena za potpisivanje svakog bita, potpisivanje rezultira digitalnim potpisom veličine izvornog dokumenta. Standardno rješenje ovog problema je korištenje hash funkcija kojima ćemo se također baviti u ovom radu. U drugom dijelu rada upoznat ćemo se s ElGamalovom shemom potpisa objavljenom 1985. te diskutirati o njenim varijantama, Schonorrovom shemom i Algoritmom digitalnog potpisa.

2 OSNOVE KRIPTOGRAFIJE

Za početak ponovit ćemo neke bitne pojmove iz teorije brojeva koje će nam biti potrebne za bolje razumijevanje ne samo uvodnog dijela nego i ostatka rada.

Definicija 2.1. *Neka je n prirodan broj veći od 1. Skup $S = \{a_1, a_2, \ldots, a_n\}$ se naziva potpuni sustav ostataka modulo n ako za svaki cijeli broj b postoji jedinstveni $a_i \in S$ za koji vrijedi $b \equiv a_i \pmod{n}$.*

Primjer 2.1. Nekoliko potpunih sustava ostataka modulo 5 su: $\{0, 1, 2, 3, 4\}$, $\{1, 2, 3, 4, 5\}$, $\{15, 26, 32, -12, 44\}$, $\{-10, -8, -4, 13, 39\}$. Očigledno ih postoji beskonačno mnogo.

Svaki potpuni sustav ostataka modulo n ima točno n elemenata. Također, svaki n - člani skup koji se sastoji od cijelih brojeva međusobno nekongruentnih modulo n predstavlja jedan potpuni sustav ostataka modulo n. Najčešće korišten potpun sustav ostataka modulo n je skup $\{0, 1, 2, \ldots, n-1\}$.

Definicija 2.2. *Neka je n prirodan broj veći od 1. Skup $S = \{a_1, a_2, ..., a_k\}$ se naziva reducirani sustav ostataka modulo n ako za svaki cijeli broj b, koji je relativno prost s n, postoji jedinstveni $a_i \in S$ za koji vrijedi $b \equiv a_i \pmod{n}$.*

Primjer 2.2. Reducirani sustavi ostataka modulo 5 su: $\{1, 2, 3, 4\}$ i $\{-2, -6, 6, 7\}$.

Postoji beskonačno mnogo reduciranih sustava ostataka modulo n. Također, svaki reducirani sustav ostataka modulo n ima jednako mnogo elemenata.

Definicija 2.3. *Neka je n prirodan broj. Broj prirodnih brojeva u nizu $1, 2, \ldots, n$ koji su relativno prosti s n se označava s $\varphi(n)$; ovim je definirana funkcija $\varphi : \mathbb{N} \to \mathbb{N}$ koja se naziva Eulerova funkcija.*

Primjer 2.3. $\varphi(1) = 1$, $\varphi(5) = 4$, $\varphi(9) = 6$.

Ako je p prost broj, tada je $\varphi(p) = p - 1$. Također, ako za neki prirodan broj n vrijedi $\varphi(n) = n - 1$, možemo zaključiti kako je n relativno prost sa svakim manjim prirodnim brojem. Prema tome, n nema djelitelja većeg od 1 i manjeg od n pa je prost.

Lema 2.1. *Neka je $S = \{a_1, a_2, \ldots, a_{\varphi(n)}\}$ potpuni sustav ostataka modulo n. Tada je i $\{b \cdot a_1, b \cdot a_2, \ldots, b \cdot a_{\varphi(n)}\}$ potpuni sustav ostataka modulo n, za svaki cijeli broj b za koji vrijedi $(b, n) = 1$.*

Teorem 2.1. (Eulerov teorem). *Neka je a cijeli broj te n prirodan broj. Ako su brojevi a i n relativno prosti, tada je $a^{\varphi(n)} \equiv 1 \pmod{n}$.*

Ako je p prost broj i a cijeli broj koji nije djeljiv s p, tada su a i p relativno prosti. Idući rezultat je direktna posljedica Eulerova teorema:

Korolar 2.1. (Mali Fermatov teorem). *Neka je p prost broj i a cijeli broj. Tada je $a^p \equiv a \pmod{p}$ te ako p ne dijeli a vrijedi i $a^{p-1} \equiv 1 \pmod{p}$.*

Definicija 2.4. *Funkciju $f : \mathbb{N} \to \mathbb{C}$ za koju vrijedi:*

1. $f(1) = 1$,
2. $f(mn) = f(m)f(n)$ *za sve m, n takve da je $(m, n) = 1$,*

zovemo multiplikativna funkcija.

Teorem 2.2. *Eulerova funkcija je multiplikativna.*

Definicija 2.5. *Neka je p prost broj i $a < p$ prirodan broj. Tada postoji prirodan broj b za koji vrijedi $a \cdot b \equiv 1 \pmod{p}$ i takav broj b se naziva multiplikativni inverz od a modulo p.*

Definicija 2.6. *Neka su $n \in \mathbb{N}$ i $a \in \mathbb{Z}$ takvi da je $(a, n) = 1$. Red od a modulo n je najmanji $e \in \mathbb{N}$ takav da je $a^e \equiv 1 \pmod{n}$. Red od a modulo n najčešće označavamo s $ord_n(a)$.*

Primjer 2.4. Neka je $a = 2$ i $n = 9$. Nadalje, vrijedi $(2, 9) = 1$. Iz kongruencija

$$\begin{aligned}
2^0 &\equiv 1 \pmod{9},\\
2^1 &\equiv 2 \pmod{9},\\
2^2 &\equiv 4 \pmod{9},\\
2^3 &\equiv 8 \pmod{9},\\
2^4 &\equiv 7 \pmod{9},\\
2^5 &\equiv 5 \pmod{9},\\
2^6 &\equiv 1 \pmod{9},
\end{aligned}$$

slijedi da je red od 2 modulo 9 jednak 6.

Definicija 2.7. *Neka je $a \in \mathbb{Z}$, $n \in \mathbb{N}$ i $(a, n) = 1$. Kažemo da je a primitivni korijen modulo n ako je $ord_n(a) = \varphi(n)$.*

Uočimo da svaki prost broj p ima primitivni korijen.

Primjer 2.5. Neka je $a = 2$ i $n = 11$. Najmanja vrijednost od k, $1 \leq k \leq 10$ za koju je $2^k \equiv 1 \pmod{11}$ je 10. Stoga je 2 primitivni korijen modulo 11.

2.1 Kriptosustavi

Kriptografija (od grčki krypto - skrivati te grafo - pisati) je znanstvena disciplina koja se bavi analiziranjem i pronalaženjem metoda kako uspostaviti sigurnu komunikaciju preko nesigurnog komunikacijskog kanala. Zadatak je kriptografije omogućavanje sigurne komunikacije između *pošiljatelja* i *primatelja* (u ovom radu zvat ćemo ih Ana i Luka) na način da njihov protivnik (Filip) ne može odgonetnuti sadržaj poruke koju pošiljaoc šalje primaocu.

Poruku koju pošiljatelj želi poslati primatelju zovemo *otvoreni tekst*. *Šifriranje* je transformacija otvorenog teksta koristeći unaprijed dogovoreni ključ, a dobiveni rezultat nazivamo *šifrat*. *Dešifriranje* je postupak kojim se šifrat transformira u izvorne podatke, odnosno u otvoreni tekst. I otvoreni tekst i šifrat se sastoje od elemenata određenih, ne nužno jednakih, alfabeta. Najčešće se alfabet otvorenog teksta sastoji od slova abecede i znamenki, ponekad i interpunkcijskih znakova, dok se alfabet šifrata često sastoji samo od znamenki.

Šifra je uređen par dvije funkcije, od kojih prva služi za šifriranje, a druga za dešifriranje. Ove funkcije često ovise o nekom unaprijed zadanom parametru (*ključu*), poznatom pošiljaocu i primaocu poruke. Ključ je uglavnom jednak nekom odabranom slovu abecede, broju ili nekoj ključnoj riječi.

Definicija 2.1.1. *Kriptosustav je uređena petorka* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, gdje su zadovoljeni sljedeći uvjeti:*

1. $\mathcal{P}$ *je konačan skup svih mogućih osnovnih elemenata otvorenog teksta*
2. $\mathcal{C}$ *je konačan skup svih mogućih osnovnih elemenata šifrata*
3. $\mathcal{K}$ *je konačan skup svih mogućih ključeva*
4. $\mathcal{E}$ *je skup svih funkcija šifriranja*
5. $\mathcal{D}$ *je skup svih funkcija dešifriranja*
6. *Za svaki* $K \in \mathcal{K}$ *postoji* $e_K \in \mathcal{E}$ *i odgovarajući* $d_K \in \mathcal{D}$*. 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* $x \in \mathcal{P}$*.*

Kriptosustave dijelimo na one s tajnim i one s javnim ključem. Kriptosustavi s tajnim ključem koriste isti ključ za šifriranje i dešifriranje poruka (ili se ključ za dešifriranje može lako odrediti poznavajući ključ za šifriranje i obratno). Ovakvi kriptosustavi se nazivaju *simetrični kriptosustavi*. U slučaju kriptosustava s javnim ključem praktički je nemoguće odrediti ključ za dešifriranje (u nekom razumnom vremenu), usprkos tomu što poznajemo ključ za šifriranje. Ovakve kriptosustave nazivamo *asimetrični kriptosustavi*.

2.2 RSA kriptosustav

RSA kriptosustav je nastao 1977. i nazvan je prema inicijalima trojice njegovih tvoraca, matematičara Rona Rivesta, Adia Shamira i Lena Adlemana. Osnovne sastavnice RSA

kriptosustava su multiplikativni inverzi modulo neki prirodan broj te Eulerov teorem. Alfabet otvorenog teksta se sastoji od slova engleske abecede, no čitav postupak šifriranja i dešifriranja se obavlja nad cijelim brojevima te možemo smatrati kako se od njih sastoji i polazni alfabet.

RSA kriptosustav sadrži sljedeće parametre:

- modul n koji je produkt neka dva prosta broja p i q, po mogućnosti što veća
- eksponent e koji se koristi za šifriranje
- eksponent d koji se koristi za dešifriranje.

Definicija 2.2.1. *Neka je $n = p \cdot q$, gdje su p i q prosti brojevi. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}/n\mathbb{Z} = \{0, 1, \ldots, n-1\}$, te*

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

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

$$e_K(x) = x^e \mod n,$$
$$d_K(y) = y^d \mod n.$$

Računski najkompliciraniji koraci prilikom korištenja RSA kriptosustava su određivanje izraza x^e i i y^d modulo n. Kako ovaj kriptosustav često koristi velike eksponente, prethodni izrazi mogu poprimiti vrijednosti koje su krajnje nepogodne za računanje. Nama u stvari nisu potrebni prirodni brojevi x^e i y^d, već samo njihovi ostatci pri dijeljenju s n. Računanje ostataka pri dijeljenju koje daju kvadrati je standardno najlakše izvediv zadatak od svih potencija. Osnovni korak je prikazati eksponent u obliku sume potencija broja 2 te iskoristiti dobivene ostatke u odgovarajućoj kongruenciji.

Algoritam:

- Tajno odaberemo dva velika prosta broja p i q.
- Izračunamo $n = p \cdot q$ i $\varphi(n) = (p-1)(q-1) = n - p - q + 1$. Ovdje je $\varphi(n)$ Eulerova funkcija.
- Odabiremo enkripcijski eksponent e takav da je $(e, \varphi(n)) = 1$.
- Iz $(e, \varphi(n)) = 1$ slijedi da postoji multiplikativni inverz d od e modulo $\varphi(n)$, tj. $e \cdot d \equiv 1 \pmod{\varphi(n)}$. Broj d se naziva dekripcijski eksponent. Multiplikativni inverz d se može odrediti iz Euklidovog algoritma jer ima svojstvo da postoji neki cijeli broj u za koji je $ed + u\varphi(n) = 1$.
- Neka je x dio otvorenog teksta koji treba šifrirati. Uzimamo da je $x < n$. Odgovarajući dio šifrata dobivamo pomoću $f_t(x) = x^e \mod n$. No, ako je y dio šifrata dekripcija se obavlja pomoću $g_t(y) = y^d \mod n$. U oba slučaja t je jednak (n, e).

Parametar (n, e) je javni ključ poznat svima, dok su faktorizacija $n = p \cdot q$ i eksponent d tajni i poznati samo pošiljatelju i primatelju poruke.

3 DIGITALNI POTPIS

Svrha konvencionalnog rukopisnog potpisa, priloženog dokumentu, jest identifikacija odgovorne osobe. Potpis se koristi u svakodnevnim situacijama kao što su pisanje pisma, podizanje novca iz banke, potpisivanje ugovora i slično. Shema potpisa je metoda potpisivanja poruka pohranjenih u elektroničkom obliku te kao takva, potpisana poruka se može prenositi računalnim mrežama. U nastavku rada predstavit ćemo nekoliko shema potpisa, ali tek nakon razmatranja nekih temeljnih razlika između konvencionalnih i digitalnih potpisa.

Prvo pitanje koje se nameće jest potpisivanja dokumenta. Konvencionalni potpis je potpis koji je ujedno i dio dokumenta koji se potpisuje. Međutim, digitalni potpis nije u fizičkom smislu pričvršćen na poruku koja je potpisana, tako da algoritam koji se koristi mora nekako "vezati" potpis na poruku. Drugo pitanje je pitanje potvrde. Konvencionalni potpis potvrđen je usporedbom s drugim, autentičnim potpisima. Primjerice, kada osoba potpiše kupnju kreditnom karticom, prodavač treba usporediti potpis na prodajnom listu s potpisom na poleđini kreditne kartice. Naravno, ovo nije vrlo sigurna metoda jer je relativno lako krivotvoriti nečiji potpis. S druge strane, digitalne potpise bilo tko može potvrditi pomoću javnog algoritma verifikacije. Korištenjem osigurane sheme potpisa spriječava se mogućnost krivotvorenja.

Još jedna temeljna razlika između konvencionalnih i digitalnih potpisa jest da je kopija potpisane digitalne poruke identična originalu. S druge strane, primjerak potpisanog papirnatog dokumenta obično se može razlikovati od originalnog. Dakle, potrebno je obrati pažnju na sigurnost potpisane digitalne poruke kako bih se spriječila ponovna upotreba iste. Na primjer, ako Ana potpiše digitalnu poruku kojom odobrava Luki da podigne 1000 dolara s bankovnog računa (tj. potpisuje ček), ona mu daje mogućnost da to učini samo jednom. Poruka treba sadržavati informacije koje onemogućuju ponovnu upotrebu kao što je, primjerice, datum.

Shema potpisa se sastoji od dvije komponente: algoritma potpisivanja i algoritma verifikacije. Ana može potpisati poruku x pomoću tajnog (privatnog) algoritma potpisivanja sig_K koji ovisi o tajnom ključu K. Zatim se dobiveni potpis $sig_K(x)$ može potvrditi korištenjem javnog algoritma verifikacije ver_K. S obzirom na par (x, y), gdje je x poruka i y navodni potpis na x, algoritam verifikacije vraća odgovor "true" ili "false", ovisno o tome je li y valjani potpis za poruku x.

Definicija 3.1. *Shema potpisa je uređena petorka $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, gdje su zadovoljeni sljedeći uvjeti:*

1. *$\mathcal{P}$ je konačan skup svih mogućih poruka*
2. *$\mathcal{A}$ je konačan skup svih mogućih potpisa*
3. *$\mathcal{K}$, prostor ključeva, je konačan skup svih mogućih ključeva*
4. *Za svaki $K \in \mathcal{K}$, postoji algoritam potpisivanja $sig_K \in \mathcal{S}$ i odgovarajući algoritam verifikacije $ver_K \in \mathcal{V}$. Svake $sig_K : \mathcal{P} \to \mathcal{A}$ i $ver_K : \mathcal{P} \times \mathcal{A} \to \{true, false\}$ funkcije su takve da su za svaku poruku $x \in \mathcal{P}$ i za svaki potpis $y \in \mathcal{A}$ zadovoljene sljedeće jednakosti:*

$$ver_K(x,y) = \begin{cases} true, ako\ y = sig_K(x) \\ false, ako\ y \neq sig_K(x). \end{cases}$$

Par (x, y), gdje je $x \in \mathcal{P}$ i $y \in \mathcal{A}$, se naziva potpisana poruka.

Za svaki $K \in \mathcal{K}$, funkcije sig_K i ver_K trebaju biti polinomijalno vremenske funkcije, tj. funkcije čija se vrijednost na elementu domene može odrediti u polinomijalnom vremenu. Funkcija ver_K je javna, dok je sig_K tajna. Za danu poruku x trebalo bi biti računski nemoguće da bilo koja druga osoba, osim Ane, izračuna potpis y tako da je $ver_K(x, y) = true$ (napominje se da bi moglo biti više od jednog takvog y za određeni x, ovisno o tome kako je definirana funkcija ver). Ako Filip može izračunati par (x, y) tako da je $ver_K(x, y) = true$, gdje Ana nije prethodno potpisala poruku x, tada se potpis y naziva krivotvorina. Neformalno, krivotvoreni potpis je važeći potpis koji pripada drugoj osobi, a ne Ani.

Nadalje, RSA kriptosustav se može koristiti za osiguravanje digitalnih potpisa te je u ovom kontekstu poznat kao RSA shema potpisa. Dakle, Ana potpisuje poruku x koristeći pravilo RSA dešifriranja d_K. Ana je jedina osoba koja može stvoriti potpis jer je $d_K = sig_K$ tajno. Algoritam verifikacije koristi pravilo RSA šifriranja e_K, gdje svatko može potvrditi potpis jer je e_K javno. Imajte na umu da bilo tko može krivotvoriti Anin RSA potpis slučajnim odabirom y i računanjem $x = e_K(y)$. U tom slučaju je $y = sig_K(x)$ važeći potpis na poruci x. Međutim, ne možemo najprije odabrati x te računati odgovarajući potpis y jer bi tada RSA kriptosustav bio nesiguran.

Kriptosustav 3.1. (RSA shema potpisa). Neka je $n = p \cdot q$, gdje su p i q prosti brojevi. Neka je $\mathcal{P} = \mathcal{A} = \mathbb{Z}_n$ i definiramo

$$\mathcal{K} = \{(n, p, q, a, b) : n = p \cdot q, p, q \text{ prosti brojevi}, a \cdot b \equiv 1 \pmod{\varphi(n)}\}.$$

Vrijednosti n i b su javne, dok su vrijednosti p, q i a tajne.

Za $K = (n, p, q, a, b)$ definiramo

$$sig_K(x) = x^a \mod n$$

i

$$ver_K(x, y) = true \Leftrightarrow x \equiv y^b \pmod{n}$$

gdje su $x, y \in \mathbb{Z}_n$.

$\Diamond$

Jedan od načina da se spriječi ovaj napad je zahtijevati da poruka sadrži dovoljnu količinu suvišnih informacija da krivotvoreni potpis ovog tipa, osim s vrlo malom vjerojatnošću, ne odgovara smislenoj poruci x. Alternativno, upotreba hash funkcija uz sheme potpisa eliminirat će ovakvu metodu krivotvorenja. Ovaj pristup se nastavlja u sljedećem poglavlju.

Pogledajmo ukratko kako kombinirati potpisivanje i šifriranje s javnim ključem. Pretpostavimo da Ana želi poslati potpisanu, šifriranu poruku Luki. S obzirom na dani otvoreni tekst x, Ana će izračunati svoj potpis $y = sig_{Ana}(x)$, zatim šifrirati x i y koristeći Lukinu

javnu funkciju šifriranja e_{Luka}, dobivajući time $z = e_{Luka}(x, y)$. Šifrat z će biti prenesen Luki. Kada Luka primi z, prvo ga dešifrira koristeći funkciju dešifriranja d_{Luka} kako bi dobio (x, y), zatim koristi Aninu javnu funkciju verifikacije kako bi provjerio vrijedi li $ver_{Ana}(x, y) = true$.

No, što ako je Ana prvo šifrirala x, zatim potpisala rezultat? U tom slučaju ona bi izračunala

$$z = e_{Luka}(x) \text{ i } y = sig_{Ana}(z).$$

Ana bi prenela par (z, y) Luki, Luka bi dešifrirao z dobivajući pri tome x, zatim potvrdio potpis y na z koristeći ver_{Ana}. Ukoliko Filip uspije doći do para (z, y) ovog tipa, mogao bi zamijeniti Anin potpis y vlastitim potpisom

$$y' = sig_{Filip}(z).$$

Ovo je potencijalni problem koji se može pojaviti ovakvim pristupom. Imajte na umu da Filip može potpisati šifrat $z = e_{Luka}(x)$ iako nepozna otvoreni tekst x. Ukoliko Filip prenese (z, y') Luki, Luka će potvrditi Filipov potpis koristeći ver_{Filip} te može zaključiti da otvoreni tekst x potječe od Filipa. Iz tog razloga se preporučuje potpisivanje prije šifriranja.

4 SIGURNOST DIGITALNOG POTPISA

Što znači da shema potpisa bude sigurna? Kao što je bio slučaj kod kriptosustava, potrebno je odrediti model napada, cilj protivnika i vrstu sigurnosti koju pruža shema.

U slučaju shema potpisa uobičajeno se uzimaju sljedeće vrste modela napada:

- **Napad samo pomoću ključa.** Filip posjeduje Anin javni ključ, tj. funkciju verifikacije ver_K.
- **Napad na poznatu poruku.** Filip posjeduje poruke koje je prethodno potpisala Ana, recimo (x_1, y_1), (x_2, y_2), ..., gdje su x_i poruke, a y_i Anini potpisi na tim porukama takvi da je $y_i = sig_K(x_i)$, $i = 1, 2, 3, \ldots$
- **Napad na odabranu poruku.** Filip zahtijeva Anine potpise na porukama. Stoga odabire poruke $x_1, x_2, \ldots$, zatim Ana potpisuje odabrane poruke tako da $y_i = sig_K(x_i)$, $i = 1, 2, 3, \ldots$

Uzet će se u obzir nekoliko mogućih protivničkih ciljeva:

- **Potpuno probijanje.** Protivnik je u mogućnosti odrediti Anin tajni ključ, tj. funkciju potpisivanja sig_K. Stoga on može stvoriti valjane potpise na bilo kojoj poruci.
- **Odabrana krivotvorina.** Uz nezanemarivu vjerojatnost, protivnik je u mogućnosti stvoriti valjani potpis na poruci koju je odabrao netko drugi. Drugim riječima, ukoliko je protivniku dana poruka x, on može odrediti (s nekom vjerojatnošću) potpis y tako da vrijedi $ver_K(x, y) = true$. Poruka x ne bi trebala biti ona koju je prethodno potpisala Ana.
- **Postojana krivotvorina.** Protivnik je u mogućnosti stvoriti valjani potpis za barem jednu poruku. Drugim riječima, protivnik može stvoriti par (x, y), gdje je x poruka te vrijedi $ver_K(x, y) = true$. Poruka x ne bi trebala biti ona koju je prethodno potpisala Ana.

Shema potpisa ne može biti bezuvjetno sigurna jer Filip može testirati sve moguće potpise $y \in \mathcal{A}$ za danu poruku x koristeći javni algoritam ver_K, sve dok ne pronađe valjani potpis. Dakle, ukoliko Filip bude imao dovoljno vremena, moći će krivotvoriti Anin potpis na bilo kojoj poruci. Kao što je bio slučaj kod kriptosustava s javnim ključem, cilj nam je pronaći računalno sigurne sheme potpisa.

Primjer 4.1. Filip može konstruirati valjanu potpisanu poruku na način da prvo odabire potpis y, zatim računa x tako da vrijedi $ver_K(x, y) = true$. Ovdje je riječ o postojanoj krivotvorini nastaloj napadom samo pomoću ključa.

Primjer 4.2. Sljedeća vrsta napada temelji se na multiplikativnom svojstvu RSA. Pretpostavimo da su $y_1 = sig_K(x_1)$ i $y_2 = sig_K(x_2)$ bilo koje dvije poruke koje je prethodno potpisala Ana. Tada vrijedi

$$ver_K(x_1x_2 \mod n, y_1y_2 \mod n) = true,$$

stoga Filip može stvoriti valjani potpis $y_1y_2 \mod n$ na poruci $x_1x_2 \mod n$. Ovo je primjer postojane krivotvorine nastale napadom na poznatu poruku.

Primjer 4.3. Pretpostavimo da Filip želi krivotvoriti potpis na poruci x, gdje je x vjerojatno odabrao netko drugi. Za njega je jednostavno pronaći $x_1, x_2 \in \mathbb{Z}_n$ takve da je $x \equiv x_1x_2 \pmod{n}$. Nadalje, pretpostavimo da Filip od Ane traži potpise y_1 i y_2 na porukama x_1 i x_2. Kao u prethodnom primjeru, $y_1y_2 \mod n$ je potpis za poruku $x = x_1x_2 \mod n$. U ovom slučaju riječ je o odabranoj krivotvorini nastaloj napadom na odabranu poruku.

4.1 Hash funkcije i integritet podataka

Kriptografska hash funkcija može osigurati integritet podataka te se koristi za izradu "otiska prsta" podataka. Ukoliko su se podaci izmijenili, otisak prsta neće biti važeći. U slučaju da su podaci pohranjeni na nesigurnom mjestu, njihov integritet se može s vremena na vrijeme provjeravati ponovnim uzimanjem otiska prsta i verifikacijom da se on nije promijenio.

Neka je h hash funkcija i x neki podatak. Primjerice, x može biti binarni niz proizvoljne duljine. Odgovarajući otisak prsta je definiran kao $y = h(x)$. Ovakav otisak prsta se često naziva *izmijenjena poruka*. Izmijenjena poruka bi obično bila prilično kratak binarni niz, uobičajeno 160 bita. Pretpostavimo da je y pohranjen na sigurnom mjestu, dok x nije. Ukoliko se x promijeni u x', nadat ćemo se da "stara" izmijenjena poruka y nije ujedno izmijenjena poruka za x'. Ako je to uistinu slučaj, onda činjenicu da se x promijenio možemo jednostavno otkriti računanjem izmijenjene poruke $y' = h(x')$ i potvrdom da je $y' \neq y$.

Prethodni motivacijski primjer pretpostavlja postojanje jedinstvene fiksne hash funkcije. Korisno je također promatrati familije hash funkcija s ključem. Hash funkcija s ključem često se koristi kao autentifikacijski kod poruke (eng. *message authentication code*, skraćeno MAC).

Pretpostavimo da Ana i Luka dijele tajni ključ K koji određuje hash funkciju h_K. Za poruku x, Ana i Luka mogu izračunati odgovarajuću autentifikacijsku oznaku $y = h_K(x)$ te Ana par (x, y) može prenijeti Luki preko nesigurnog komunikacijskog kanala. Kada primi par (x, y), Luka može provjeriti vrijedi li $y = h_K(x)$. Ukoliko je prethodni uvjet zadovoljen, Luka je siguran da x i y nisu izmijenjeni od strane protivnika te je uvjeren da poruka x potječe od Ane.

Primjetite razliku osiguravanja integriteta podataka hash funkcije bez ključa u odnosu na hash funkciju s ključem. Kod hash funkcije bez ključa izmijenjena poruka mora biti sigurno pohranjena kako se nebi mogla mijenjati. Ukoliko Ana i Luka koriste tajni ključ K za određivanje hash funkcije koju koriste, moći će prenijeti podatke i autentifikacijsku oznaku preko nesigurnog komunikacijskog kanala.

Definicija 4.1.1. *Hash familija je uređena četvorka $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$, gdje su zadovoljeni sljedeći uvjeti:*

1. *$\mathcal{X}$ je skup svih mogućih poruka*

2. *$\mathcal{Y}$ je konačan skup svih mogućih izmijenjenih poruka ili autentifikacijskih oznaka*

3. *$\mathcal{K}$, prostor ključeva, je konačan skup svih mogućih ključeva*

4. *Za svaki $K \in \mathcal{K}$ postoji hash funkcija $h_K \in \mathcal{H}$ takva da je $h_K : \mathcal{X} \to \mathcal{Y}$.*

U prethodnoj definiciji $\mathcal{X}$ može biti konačan ili beskonačan skup, dok je $\mathcal{Y}$ uvijek konačan. Ukoliko je $\mathcal{X}$ konačan skup, hash funkcija se ponekad zove *funkcija sažimanja*. U ovom slučaju uvijek ćemo pretpostaviti da je $|\mathcal{X}| \geq |\mathcal{Y}|$. Često ćemo također pretpostaviti jači uvjet, tj. da je $|\mathcal{X}| \geq 2|\mathcal{Y}|$.

Par $(x, y) \in \mathcal{X} \times \mathcal{Y}$ je *valjani par* pod ključem K ako $h_K(x) = y$. Veliki dio ovog poglavlja odnosi se upravo na metode sprječavanja konstrukcija određenih vrsta valjanih parova od strane protivnika.

Nadalje, neka $\mathcal{F}^{\mathcal{X},\mathcal{Y}}$ označava skup svih funkcija iz $\mathcal{X}$ u $\mathcal{Y}$. Pretpostavimo da je $|\mathcal{X}| = N$ i $|\mathcal{Y}| = M$. Tada je jasno da je $|\mathcal{F}^{\mathcal{X},\mathcal{Y}}| = M^N$. Bilo koja hash familija $\mathcal{F} \subseteq \mathcal{F}^{\mathcal{X},\mathcal{Y}}$ se naziva (N, M) - *hash familija*.

Hash funkcija bez ključa je funkcija $h : \mathcal{X} \to \mathcal{Y}$, gdje su $\mathcal{X}$ i $\mathcal{Y}$ određeni kao u *Definiciji 4.1.1*. Pod hash funkcijom bez ključa jednostavno možemo smatrati hash familiju u kojoj postoji samo jedan mogući ključ, tj. hash familiju za koju je $|\mathcal{K}| = 1$.

4.2 Sigurnost hash funkcija

Pretpostavimo da je $h : \mathcal{X} \to \mathcal{Y}$ hash funkcija bez ključa. Neka je $x \in \mathcal{X}$ i definiramo $y = h(x)$. U mnogim kriptografskim aplikacijama hash funkcija poželjno je da postoji samo jedan način da se dobije valjani par (x, y); prvo odabrati x, zatim izračunati $y = h(x)$ primjenom funkcije h na x. Ostali sigurnosni zahtjevi hash funkcija motivirani su njihovim primjenama u posebnim protokolima kao što su sheme potpisa. U nastavku ćemo definirati tri problema koja bi trebala biti teško rješiva u realnom vremenu ako bi se hash funkcija smatrala sigurnom.

Problem 4.2.1. (Praslika). Neka je $h : \mathcal{X} \to \mathcal{Y}$ hash funkcija i $y \in \mathcal{Y}$. Treba pronaći $x \in \mathcal{X}$ takav da je $h(x) = y$. Ukoliko se ovaj problem može riješiti za dani $y \in \mathcal{Y}$, (x, y) je valjani par. Hash funkciju za koju problem nije učinkovito rješiv nazivamo *otpornom na prasliku*.

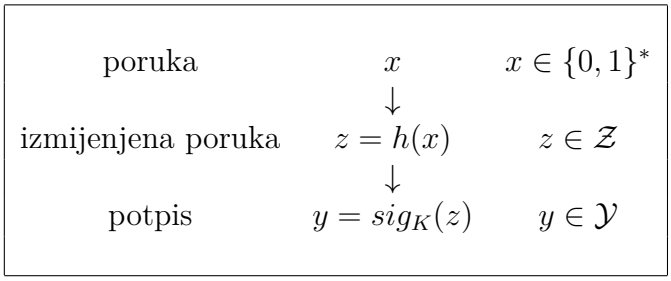
Problem 4.2.2. (Druga praslika). Neka je $h : \mathcal{X} \to \mathcal{Y}$ hash funkcija i $x \in \mathcal{X}$. Treba pronaći $x' \in \mathcal{X}$ takav da je $x' \neq x$ i $h(x') = h(x)$. Ukoliko se ovaj problem može riješiti, $(x', h(x))$ je valjani par. Hash funkciju za koju problem nije učinkovito rješiv nazivamo *otpornom na drugu prasliku*.

Problem 4.2.3. (Kolizija). Neka je $h : \mathcal{X} \to \mathcal{Y}$ hash funkcija. Treba pronaći $x, x' \in \mathcal{X}$ takve da je $x' \neq x$ i $h(x') = h(x)$. Ukoliko se ovaj problem može riješiti, (x, y) i (x', y) su

valjani parovi gdje je $y = h(x) = h(x')$. Hash funkciju za koju problem nije učinkovito rješiv možemo zvati *otpornom na koliziju.*

4.3 Potpisi i hash funkcije

Sheme potpisa se gotovo uvijek koriste zajedno s javnom kriptografskom hash funkcijom. Hash funkcija $h : \{0,1\}^* \to \mathcal{Z}$ će uzeti poruku proizvoljne duljine te stvoriti izmijenjenu poruku određene veličine (160 bita je čest izbor). Izmijenjena poruka će potom biti potpisana pomoću sheme potpisa $(\mathcal{P}, \mathcal{A}, \mathcal{K}, \mathcal{S}, \mathcal{V})$, gdje je $\mathcal{Z} \subseteq \mathcal{P}$. *Alfabet* je proizvoljan neprazan skup. Ako je sa A označen neki alfabet tada se skup svih riječi obično označava sa A^*. U našem slučaju $\{0,1\}^*$ je skup svih riječi sastavljenih od nula i jedinica. Sljedeća slika nam daje shematski prikaz korištenja hash funkcije i sheme potpisa.



Slika 4.1.: Potpisivanje izmijenjene poruke

Pretpostavimo da Ana želi potpisati poruku x, što je niz znakova proizvoljne duljine. Ona najprije konstruira izmijenjenu poruku $z = h(x)$, zatim računa potpis na z, tj. $y = sig_K(z)$. Nakon toga prenosi uređeni par (x, y) preko komunikacijskog kanala. Sada provjeru može izvršiti bilo tko tako što će prvo rekonstruirati izmijenjenu poruku $z = h(x)$ koristeći javnu hash funkciju h, zatim provjeriti vrijedi li $ver_K(z, y) = true$.

Moramo biti oprezni da upotreba hash funkcije h ne oslabi sigurnost sheme potpisa jer je izmijenjena poruka ta koja je potpisana, a ne poruka. Bit će potrebno da h zadovolji određena svojstva kako bi se spriječili različiti napadi. O željenim svojstvima hash funkcija diskutirali smo u potpoglavlju 4.2.

Primjer 4.3.1. Filip započinje s valjanom potpisanom porukom (x, y), gdje je $y = sig_K(h(x))$. Par (x, y) može biti bilo koja poruka koju je prethodno potpisala Ana. Zatim računa $z = h(x)$ i pokušava pronaći $x' \neq x$ takav da je $h(x') = h(x)$. Ukoliko je Filip u mogućnosti to napraviti, (x', y) bi bila valjana potpisana poruka takva da je y krivotvoreni potpis za poruku x'. Ovo je slučaj postojane krivotvorine nastale napadom na poznatu poruku. Da bismo spriječili ovu vrstu napada zahtijevamo da h bude otporna na drugu prasliku.

Primjer 4.3.2. Filip pronalazi dvije poruke $x \neq x'$ takve da je $h(x) = h(x')$. Filip zatim daje poruku x Ani te ju nagovara da potpiše izmijenjenu poruku $h(x)$, dobivajući time y. Tada je (x', y) valjana potpisana poruka, a y krivotvoreni potpis za poruku x'. Ovo je slučaj postojane krivotvorine nastale napadom na odabranu poruku. Napad se može spriječiti ukoliko je h otporna na koliziju.

S određenim shemama često je moguće krivotvoriti potpise na slučajno odabranim izmijenjenim porukama z (npr. koristeći RSA shemu potpisa). Pretpostavimo da je shema potpisa (bez hash funkcije) podložna postojanoj krivotvorini nastaloj napadom samo pomoću ključa.

Primjer 4.3.3. Pretpostavimo da Filip računa potpis na izmijenjenoj poruci z, zatim pronalazi poruku x takvu da je $z = h(x)$. Ukoliko je Filip u mogućnosti to napraviti, (x, y) je valjana potpisana poruka gdje je y krivotvoreni potpis za poruku x. Ovo je postojana krivotvorina nastala napadom samo pomoću ključa. Da bismo spriječili ovaj napad želimo da h bude hash funkcija otporna na prasliku.

5 PROBLEM DISKRETNOG LOGARITMA

Problem diskretnog logaritma je matematički problem koji se javlja u mnogim područjima, uključujući modulo p. Prva objavljena konstrukcija javnog ključa temelji se na problemu diskretnog logaritma u konačnom polju $\mathbb{F}_p$ s p elemenata, gdje je p prost broj, dok je $\mathbb{F}_p^* = \mathbb{F}_p \backslash \{0\}$. Radi jednostavnosti koristit ćemo oznake $\mathbb{F}_p$ i $\mathbb{Z}/p\mathbb{Z}$, gdje će se u polju $\mathbb{F}_p$ koristiti znak jednakosti, a u $\mathbb{Z}/p\mathbb{Z}$ znak kongruencije.

Teorem 5.1. (o primitivnom elementu). *Neka je p prost broj. Tada postoji element $g \in \mathbb{F}_p^*$ čije potencije daju sve elemente iz $\mathbb{F}_p^*$, tj. $\mathbb{F}_p^* = \{1, g, g^2, \ldots, g^{p-2}\}$. Elemente s ovim svojstvom nazivamo primitivni elementi u $\mathbb{F}_p$. Red elemenata iz $\mathbb{F}_p^*$ jednak je $p-1$.*

Primjer 5.1. Primitivni element u polju $\mathbb{F}_{13}$ je 2, jer u $\mathbb{F}_{13}$,

$$\begin{array}{llllll} 2^0 = 1 & 2^1 = 2 & 2^2 = 4 & 2^3 = 8 & 2^4 = 3 & 2^5 = 6 \\ 2^6 = 12 & 2^7 = 11 & 2^8 = 9 & 2^9 = 5 & 2^{10} = 10 & 2^{11} = 7. \end{array}$$

S druge strane, 2 nije primitivni element u polju $\mathbb{F}_{23}$, jer u $\mathbb{F}_{23}$,

$$\begin{array}{llllll} 2^0 = 1 & 2^1 = 2 & 2^2 = 4 & 2^3 = 8 & 2^4 = 16 & 2^5 = 9 \\ 2^6 = 18 & 2^7 = 13 & 2^8 = 3 & 2^9 = 6 & 2^{10} = 12 & 2^{11} = 1, \end{array}$$

tako da se vratimo na 1 prije dobivanja svih 22 nenul vrijednosti modulo 23. Međutim, računanjem dobijemo da je 5 primitivni element u $\mathbb{F}_{23}$, jer u $\mathbb{F}_{23}$,

$$\begin{array}{llllll} 5^0 = 1 & 5^1 = 5 & 5^2 = 2 & 5^3 = 10 & 5^4 = 4 & 5^5 = 20 \\ 5^6 = 8 & 5^7 = 17 & 5^8 = 16 & 5^9 = 11 & 5^{10} = 9 & 5^{11} = 22 \\ 5^{12} = 18 & 5^{13} = 21 & 5^{14} = 13 & 5^{15} = 19 & 5^{16} = 3 & 5^{17} = 15 \\ 5^{18} = 6 & 5^{19} = 7 & 5^{20} = 12 & 5^{21} = 14. & & \end{array}$$

$\Diamond$

Neka je p (veliki) prost broj. *Teorem 5.1.* nam kaže da postoji primitivni element g. To znači da je svaki nenul element iz polja $\mathbb{F}_p$ jednak nekoj potenciji elementa g. Posebno, prema Malom Fermatovom teoremu vrijedi $g^{p-1} = 1$ i svaka veća potencija elementa g je jednaka 1. Ekvivalentno, elementi $1, g, g^2, \ldots, g^{p-2} \in \mathbb{F}_p^*$ su svi elemenati iz $\mathbb{F}_p^*$ zapisani nekim redosljedom.

Definicija 5.1. *Neka je g primitivni element u $\mathbb{F}_p$ i h nenul element iz $\mathbb{F}_p$. Problem diskretnog logaritma je problem pronalaženja eksponenta x za kojeg vrijedi $g^x \equiv h \pmod{p}$. Broj x se naziva diskretni logaritam od h s bazom g i označava se s $\log_g(h)$.*

Stariji termin za diskretni logaritam je *index* s oznakom $ind_g(h)$. Još uvijek se koristi u teoriji brojeva jer je prikladan zbog mogućnosti zabune između običnih i diskretnih logaritama. Možemo primjetiti da ako postoji jedno rješenje onda postoji beskonačno mnogo rješenja jer nam Mali Fermatov teorem kaže da je $g^{p-1} \equiv 1 \pmod{p}$. Prema tome, ako je x rješenje od $g^x = h$, tada je $x + k(p-1)$ također rješenje za svaku vrijednost k jer je

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

Tako je $\log_g(h)$ definiran samo dodavanjem ili oduzimanjem višekratnika od $p-1$, tj. $\log_g(h)$ je definiran modulo p - 1. Nadalje, $\log_g$ nam daje dobro definiranu funkciju

$$\log_g : \mathbb{F}_p^* \to \mathbb{Z}/(p-1)\mathbb{Z}.$$

Ponekad govorimo o diskretnom logaritmu kao cijelom broju x koji se nalazi između 0 i $p-2$ te zadovoljava kongruenciju $g^x \equiv h \pmod{p}$.

Napomena 5.1. *Vrijedi jednakost*

$$\log_g(ab) = \log_g(a) + \log_g(b), \forall a, b \in \mathbb{F}_p^*.$$

Iz ovog svojstva vidimo da je naziv "logaritam" za $\log_g(h)$ *opravdan jer množenje pretvara u zbrajanje poput običnog logaritma. U matematičkoj terminologiji diskretni logaritam* $\log_g$ *je izomorfizam grupa* $\mathbb{F}_p^*$ *i* $\mathbb{Z}/(p-1)\mathbb{Z}$.

Primjer 5.2. Neka je $p = 101$, prost broj. Tada je $g = 2$ primitivni element modulo p. Kako ćemo izračunati diskretni logaritam od $h = 32$? Trebamo pronaći x takav da je $g^x \equiv h \pmod{p}$. Iz kongruencija

$$\begin{aligned} 2^1 &\equiv 2 \pmod{101}, \\ 2^2 &\equiv 4 \pmod{101}, \\ 2^3 &\equiv 8 \pmod{101}, \\ 2^4 &\equiv 16 \pmod{101}, \\ 2^5 &\equiv 32 \pmod{101}, \end{aligned}$$

zaključujemo da je $\log_2(32) = 5$, odnosno diskretni logaritam od $h = 32$ je 5.

$\Diamond$

Moramo naglasiti da diskretni logaritam ima sličnosti s logaritmima koji su definirani nad skupovima realnih ili kompleksnih brojeva. Terminologija ima smisla jer je u oba slučaja potenciranje inverzna funkcija, ali ipak postoji razlika. Potenciranje modulo p varira s eksponentom na nepravilan način, što je suprotno ponašanju običnog logaritma.

Primjer 5.3. *Tablica 5.1.* sadrži prvih 20 potencija i diskretnih logaritama za prost broj $p = 941$ i bazu $g = 627$.

n	$g^n \pmod{p}$	n	$g^n \pmod{p}$
1	627	11	878
2	732	12	21
3	697	13	934
4	395	14	316
5	182	15	522
6	253	16	767
7	543	17	58
8	760	18	608
9	374	19	111
10	189	20	904

h	$\log_g(h)$	h	$\log_g(h)$
1	0	11	429
2	183	12	835
3	469	13	279
4	366	14	666
5	356	15	825
6	652	16	732
7	483	17	337
8	549	18	181
9	938	19	43
10	539	20	722

Tablica 5.1.: Potencije i diskretni logaritmi za $p = 941$ i $g = 627$

Problem diskretnog logaritma uključuje pretpostavku da je baza g primitivni element modulo p, što nije nužno. Općenito, za bilo koji $g \in \mathbb{F}_p^*$ i bilo koji $h \in \mathbb{F}_p^*$ problem diskretnog logaritma je određivanje eksponenta x koji zadovoljava $g^x \equiv h \pmod{p}$, uz pretpostavku da takav x postoji.

Nadalje, umjesto uzimanja nenul elemenata iz polja $\mathbb{F}_p$ koje bi pomnožili ili potencirali, možemo uzeti elemente iz bilo koje grupe te iskoristiti njena pravila uz zadanu operaciju. To nas dovodi do općenitije definicije problema diskretnog logaritma koja slijedi u nastavku.

Definicija 5.2. *Neka je G grupa obzirom na operaciju $*$. Problem diskretnog logaritma u G je pronaći, za bilo koja dva elementa $g, h \in G$, x koji zadovoljava*

$$\underbrace{g * g * g * \cdots * g}_{x \text{ puta}} = h.$$

6 ELGAMALOVA SHEMA POTPISA

U ovom ćemo poglavlju predstaviti ElGamalovu shemu potpisa opisanu u radu objavljenom 1985. Tijekom godina predložene su različite varijante ove sheme potpisa. Izmjena sheme prihvaćena je pod nazivom algoritam digitalnog potpisa (eng. *Digital Signature Algorithm*, skraćeno DSA) od strane Nacionalnog instituta za standarde i tehnologiju (eng. *National Institute of Standards and Technology*, skraćeno NIST). DSA također uključuje neke ideje korištene u potpisu poznatom kao Schnorrova shema potpisa. Sve ove sheme dizajnirane su za potrebe potpisa, za razliku od RSA kriptosustava, te se mogu koristiti i kao kriptosustav javnog ključa i shema potpisa.

ElGamalova shema potpisa je nedeterministička, odnosno postoji mnogo važećih potpisa za svaku danu poruku te algoritam verifikacije mora biti u mogućnosti prihvatiti bilo koji od ovih valjanih potpisa kao autentičan. Opis ElGamalove sheme potpisa je u nastavku dan kao *Kriptosustav 6.1.*

Ako je potpis ispravno konstruiran, potvrda će uspjeti jer

$$\begin{aligned}\beta^{\gamma}\gamma^{\delta} &\equiv \alpha^{a\gamma}\alpha^{k\delta} \pmod{p} \\ &\equiv \alpha^{x} \pmod{p},\end{aligned}$$

gdje koristimo činjenicu da vrijedi

$$a\gamma + k\delta \equiv x \pmod{p-1}.$$

Zapravo, puno bolje je započeti s jednadžbom verifikacije, zatim izvesti funkciju potpisivanja. Pretpostavimo da započinjemo s kongruencijom

$$\alpha^{x} \equiv \beta^{\gamma}\gamma^{\delta} \pmod{p}.$$

Zatim izvršimo supstitucije

$$\gamma \equiv \alpha^{k} \pmod{p}$$

i

$$\beta \equiv \alpha^{a} \pmod{p},$$

ali nećemo izvršiti supstituciju za γ u eksponentu od $\alpha^{x} \equiv \beta^{\gamma}\gamma^{\delta} \pmod{p}$.

Kriptosustav 6.1. (ElGamalova shema potpisa). Neka je p prost broj takav da problem diskretnog logaritma u $\mathbb{Z}_p$ nije riješiv u realnom vremenu, i neka je $\alpha \in \mathbb{Z}_p^*$ primitivni element. Neka $\mathcal{P} = \mathbb{Z}_p^*, \mathcal{A} = \mathbb{Z}_p^* \times \mathbb{Z}_{p-1}$, i definiramo

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^{a} \pmod{p}\}.$$

Vrijednosti p, α i β su javni ključ, dok je a privatni ključ.

Za $K = (p, \alpha, a, \beta)$, i za (tajni) slučajno odabrani broj $k \in \mathbb{Z}_{p-1}^*$, definiramo

$$sig_K(x,k) = (\gamma, \delta),$$

gdje

$$\gamma = \alpha^k \mod p$$

i

$$\delta = (x - a\gamma)k^{-1} \mod (p-1).$$

Za $x, \gamma \in \mathbb{Z}_p^*$ i $\delta \in \mathbb{Z}_{p-1}$, definiramo

$$ver_K(x,(\gamma,\delta)) = true \Leftrightarrow \beta^\gamma \gamma^\delta \equiv \alpha^x \pmod p.$$

$\Diamond$

Dobivamo sljedeće:

$$\alpha^x \equiv \alpha^{a\gamma + k\delta} \pmod p.$$

Sada je α primitivni element modulo p. Dobivena kongruencija je istinita ako i samo ako su eksponenti kongruentni modulo $p-1$, tj. ako i samo ako vrijedi

$$x \equiv a\gamma + k\delta \pmod{p-1}.$$

S obzirom na dane x, a, γ i k, ova kongruencija se može riješiti po δ, dajući formulu korištenu u funkciji potpisivanja *Kriptosustava 6.1.*

Ana računa potpis koristeći privatni ključ a i tajni slučajno odabrani broj k (korišten za potpisivanje poruke x). Provjera se provodi korištenjem samo javnih informacija.

Primjer 6.1. Pretpostavimo da je $p = 317$, $\alpha = 2$, $a = 131$; tada

$$\begin{aligned} \beta &= \alpha^a \mod p \\ &= 2^{131} \mod 317 \\ &= 295. \end{aligned}$$

Pretpostavimo da Ana želi potpisati poruku $x = 150$ i slučajno odabire vrijednost $k = 253$ (napominjemo da je $(253, 316) = 1$ i $253^{-1} \mod 316 = 5$). Tada

$$\gamma = 2^{253} \mod 317 = 48$$

i

$$\delta = (150 - 131 \cdot 48) \cdot 5 \mod 316 = 278.$$

Svatko može potvrditi taj potpis provjerom

$$295^{48} 48^{278} \equiv 26 \pmod{317}$$

i

$$2^{150} \equiv 26 \pmod{317}.$$

Stoga, potpis je valjan.

$\Diamond$

6.1 Sigurnost ElGamalove sheme potpisa

Pretpostavimo da Filip pokušava krivotvoriti potpis za danu poruku x, ne poznavajući pri tome a. Ukoliko Filip odabere vrijednost γ te pokuša pronaći odgovarajući δ, mora izračunati diskretni logaritam $\log_\gamma \alpha^x \beta^{-\gamma}$. S druge strane, ako prvo odabere δ te traži γ, on pokušava riješiti jednadžbu

$$\beta^\gamma \gamma^\delta \equiv \alpha^x \pmod{p}$$

po "nepoznatom" γ. Ovo je problem za kojeg nije poznato izvedivo rješenje te nam se čini da nije povezano s nekim dobro proučenim problemom poput problema diskretnog logaritma. I dalje postoji mogućnost da se istovremeno izračunaju γ i δ tako da par (γ, δ) bude potpis. Do sada nitko nije otkrio način kako to učiniti, ali također nitko nije dokazao da se to ne može učiniti.

Ako Filip odabere γ i δ te računa po x, opet je suočen sa slučajem problema diskretnog logaritma. Prema tome, Filip mora izračunati $\log_\alpha \beta^\gamma \gamma^\delta$. Sada vidimo da on neće moći potpisati danu poruku x ovakvim pristupom. Međutim, postoji metoda kojom Filip može potpisati slučajno odabranu poruku istovremenim odabirom γ, δ i x. Ovakva postojana krivotvorina je moguća napadom samo pomoću ključa (uz pretpostavku da se hash funkcija ne koristi). U nastavku slijedi opis kako se to može učiniti.

Pretpostavimo da su i i j cijeli brojevi takvi da $0 \leq i \leq p-2$, $0 \leq j \leq p-2$, i da je $\gamma = \alpha^i \beta^j \mod p$. Tada je uvjet verifikacije

$$\alpha^x \equiv \beta^\gamma (\alpha^i \beta^j)^\delta \pmod{p}.$$

Što je ekvivalentno

$$\alpha^{x-i\delta} \equiv \beta^{\gamma + j\delta} \pmod{p}.$$

Kongruencija će biti zadovoljena ako vrijedi

$$x - i\delta \equiv 0 \pmod{p-1}$$

i

$$\gamma + j\delta \equiv 0 \pmod{p-1}.$$

Za dane i i j lako možemo riješiti prethodne dvije kongruencije modulo $p-1$ po δ i x, pod uvjetom da je $(j, p-1) = 1$. Dobivamo sljedeće:

$$\begin{aligned} \gamma &= \alpha^i \beta^j \mod p, \\ \delta &= -\gamma j^{-1} \mod (p-1), \quad i \\ x &= -\gamma i j^{-1} \mod (p-1). \end{aligned}$$

Po načinu na koji smo konstruirali (γ, δ) jasno nam je da je riječ o valjanom potpisu za poruku x.

Primjer 6.1.1. Kao u *Primjeru 6.1.* pretpostavimo da su $p = 317$, $\alpha = 2$ i $\beta = 295$. Nadalje, pretpostavimo da Filip odabere $i = 172$ i $j = 291$; tada je $j^{-1} \mod (p-1) = 139$. Računamo sljedeće:

$$\begin{aligned}\gamma &= 2^{172}295^{291} \mod 317 = 262\\ \delta &= -262 \cdot 139 \mod 316 = 238\\ x &= 238 \cdot 172 \mod 316 = 172.\end{aligned}$$

Tada je $(262, 238)$ valjani potpis poruke 172, što se može potvrditi provjerom

$$295^{262}262^{238} \equiv 100 \pmod{317}$$

i

$$2^{172} \equiv 100 \pmod{317}.$$

$\Diamond$

U sljedećoj vrsti krivotvorenja Filip započinje s porukom koju je prethodno potpisala Ana. Radi se o postojanoj krivotvorini nastaloj napadom na poznatu poruku. Pretpostavimo da je par (γ, δ) valjani potpis za poruku x. Tada Filip ima mogućnost potpisati mnoge druge poruke. Pretpostavimo da su h, i i j cijeli brojevi takvi da $0 \leq h, i, j \leq p-2$, i $(h\gamma - j\delta, p-1) = 1$. Računamo sljedeće:

$$\begin{aligned}\lambda &= \gamma^h \alpha^i \beta^j \mod p\\ \mu &= \delta\lambda(h\gamma - j\delta)^{-1} \mod (p-1), \quad i\\ x' &= \lambda(hx + i\delta)(h\gamma - j\delta)^{-1} \mod (p-1).\end{aligned}$$

Zatim, potrebno je provjeriti vrijedi li uvjet verifikacije

$$\beta^\lambda \lambda^\mu \equiv \alpha^{x'} \pmod{p}.$$

Ukoliko vrijedi, par (λ, μ) je valjani potpis za poruku x'.

Obje metode su postojane krivotvorine te se ne mogu izmijeniti kako bi nastala odabrana krivotvorina. Prema tome, one ne predstavljaju prijetnju sigurnosti ElGamalove sheme potpisa, pod uvjetom da se sigurna hash funkcija koristi na način opisan u potpoglavlju 4.3.

Postoji nekoliko načina na koje se sigurnost ElGamalove sheme potpisa može probiti ako se neoprezno koristi. Prvo, slučajno odabrana vrijednost k korištena u računanju potpisa se ne bi trebala otkriti. Ukoliko je k poznat i vrijedi $(\gamma, p-1) = 1$, jednostavno je izračunati

$$a = (x - k\delta)\gamma^{-1} \mod (p-1).$$

Jednom kada se a otkrije, sigurnosni sustav je potpuno probijen i Filip može po želji krivotvoriti potpise.

Još jedna zlouporaba sustava je korištenje iste vrijednosti k u potpisivanju dvije različite poruke. To također olakšava Filipu da izračuna a te probije sustav. Kako se to može učiniti opisat će se u nastavku. Pretpostavimo da je (γ, δ_1) potpis na poruci x_1 i (γ, δ_2) potpis na poruci x_2. Tada imamo

$$\beta^{\gamma}\gamma^{\delta_1} \equiv \alpha^{x_1} \pmod{p}$$

i

$$\beta^{\gamma}\gamma^{\delta_2} \equiv \alpha^{x_2} \pmod{p}.$$

Prema tome

$$\alpha^{x_1-x_2} \equiv \gamma^{\delta_1-\delta_2} \pmod{p}.$$

Uzmemo li $\gamma = \alpha^k$, dobivamo sljedeću jednadžbu s nepoznanicom k:

$$\alpha^{x_1-x_2} \equiv \alpha^{k(\delta_1-\delta_2)} \pmod{p},$$

što je ekvivalentno

$$x_1 - x_2 \equiv k(\delta_1 - \delta_2) \pmod{p-1}.$$

Neka je $d = (\delta_1 - \delta_2, p-1)$. Budući da d dijeli $p-1$ i $\delta_1 - \delta_2$, slijedi da $d|(x_1 - x_2)$. Definiramo

$$x' = \frac{x_1 - x_2}{d}$$
$$\delta' = \frac{\delta_1 - \delta_2}{d}$$
$$p' = \frac{p-1}{d}.$$

Tada kongruencija postaje:

$$x' \equiv k\delta' \pmod{p'}.$$

S obzirom da je $(\delta', p') = 1$, možemo izračunati

$$\epsilon = (\delta')^{-1} \mod p'.$$

Vrijednost k je određena modulo p' tako da

$$k = x'\epsilon \mod p'.$$

To daje d vrijednosti koje su kandidati za k:

$$k = x'\epsilon + ip' \mod (p-1)$$

za neki i, $0 \leq i \leq d-1$. Od tih d vrijednosti, dobar (jedinstveni) kandidat se može odrediti provjerom

$$\gamma \equiv \alpha^k \pmod{p}.$$

7 VARIJANTE ELGAMALOVE SHEME POTPISA

U mnogim situacijama, poruka može biti šifrirana i dešifrirana samo jednom, tako da je dovoljno koristiti bilo koji kriptosustav za kojeg se zna da je siguran u vrijeme kada je poruka šifrirana. S druge strane, potpisana poruka može funkcionirati kao pravni dokument kao što je primjerice ugovor ili oporuka. Vrlo vjerojatno će biti potrebno provjeravati potpis dugi niz godina nakon potpisivanja poruke. Stoga je važno poduzimati mnogo veće mjere opreza glede sigurnosti digitalnog potpisivanja za razliku od kriptosustava. Budući da ElGamalova shema potpisa nije sigurnija od problema diskretnog logaritma, zahtjevano je korištenje velikog modula p. Mnogi vode raspravu o tome kako bi duljina p trebala biti najmanje 1024 bita kako bi se pružila dobra sigurnost.

1024 - bitni modul vodi do ElGamalove sheme potpisa koja sadrži 2048 bita. Za potencijalne aplikacije, od kojih mnoge uključuju upotrebu pametnih kartica, poželjan je kraći potpis. Schnorr je 1989. predložio shemu potpisa koja se može promatrati kao varijanta ElGamalove sheme potpisa u kojoj je veličina potpisa znatno smanjena. Algoritam digitalnog potpisa (eng. *Digital Signature Algorithm*, skraćeno DSA) je još jedna modifikacija ElGamalove sheme potpisa, koja uključuje neke od ideja koje se koriste u Schnorrovoj shemi potpisa. *Federal Register* je objavio DSA 19. svibnja 1994. te je usvojen kao standard 1. prosinca 1994. (prvi put je predložen u kolovozu 1991.).

Najvažnija operacija na skupu riječi je *konkatenacija*. Konkatenacija je binarna operacija na A^*, koja je definirana na sljedeći način: ako su a i b riječi (bolje reći oznake za riječi) tada kažemo da je riječ ab nastala konkatenacijom riječi a i b. Kažemo da je b podriječ riječi a ako postoje riječi c i d tako da je riječ a nastala konkatenacijom riječi c, b i d, tj. a je jednaka cbd. Oznaka za konkatenaciju u nastavku je $||$.

Podsjetimo se, hash funkcija uzima proizvoljno dugačak dokument $\mathcal{D}$ i vraća kratak niz znakova $\mathcal{H}$. U praksi, većina hash funkcija koristi algoritam, označimo ga s $\mathcal{M}$, koji transformira niz znakova duljine n u drugi niz znakova duljine n. Hash funkcija funkcionira na način da razbija neki dugačak dokument na blokove i sukcesivno, upotrebljavajući algoritam $\mathcal{M}$, kombinira svaki blok s prethodno obrađenim materijalom. Dakle, za izračunavanje $\mathcal{H}(\mathcal{D})$, prvo dodajemo dodatnih 0 bitova u $\mathcal{D}$ tako da je duljina $\mathcal{D}$ jednaka višekratniku od n bitova. To nam omogućuje da $\mathcal{D}$ zapišemo u obliku konkatenacije

$$\mathcal{D} = \mathcal{D}_1||\mathcal{D}_2||\mathcal{D}_3||\mathcal{D}_4||\dots||\mathcal{D}_k,$$

niza znakova duljine n.

7.1 Schnorrova shema potpisa

Pretpostavimo da su p i q prosti brojevi takvi da je $p - 1 \equiv 0 \pmod{q}$. Obično ćemo uzeti $p \approx 2^{1024}$ i $q \approx 2^{160}$. Schnorrova shema potpisa modificira ElGamalovu shemu na način da je $\log_2 q$ - bitna izmijenjena poruka potpisana pomoću $2\log_2 q$ - bitnog potpisa, no izračuni su izvršeni unutar skupa $\mathbb{Z}_p$. Način na koji se to postiže jest računati u podgrupi reda q od $\mathbb{Z}_p^*$. Pretpostavka sigurnosti sheme temelji se na uvjerenju da je određivanje diskretnih

logaritama u određenoj podgrupi od $\mathbb{Z}_p^*$ sigurno.

Neka je α_0 primitivni element u $\mathbb{Z}_p$ i definirajmo $\alpha = \alpha_0^{\frac{p-1}{q}} \mod p$. Tako definirani α nazivamo *q - ti korijen od 1 modulo p*. Pretpostavit ćemo da je $h : \{0,1\}^* \to \mathbb{Z}_q$ sigurna hash funkcija. Opis Schnorrove sheme potpisa u nastavku je dan kao *Kriptosustav 7.1.1.* Jednostavno je provjeriti da vrijedi $\alpha^\delta \beta^{-\gamma} \equiv \alpha^k \pmod p$. Time će Schnorrova shema potpisa biti potvrđena.

Primjer 7.1.1. Pretpostavimo da je $q = 163$ i $p = 36q + 1 = 5869$. Kako je 2 primitivni element u $\mathbb{Z}_{5869}^*$, možemo uzeti

$$\alpha = 2^{36} \mod 5869 = 1326.$$

α je q - ti korijen od 1 modulo p. Pretpostavimo da je $a = 35$; tada

$$\beta = \alpha^a \mod 5869 = 4958.$$

Sada pretpostavimo da Ana želi potpisati poruku x i slučajno odabire vrijednost $k = 23$. Prvo računa

$$\alpha^k \mod p = 1326^{23} \mod 5869 = 1729.$$

Idući korak je izračunati $h(x \parallel 1729)$, gdje je h dana hash funkcija i 1729 je prikazan binarno (kao niz znakova). U svrhu ilustracije pretpostavimo da je $h(x \parallel 1729) = 86$. Tada je δ izračunat kao

$$\delta = 23 + 35 \cdot 86 \mod 163 = 99,$$

te dobivamo potpis $(86, 99)$.

Potpis se potvrđuje računanjem

$$1326^{99} 4958^{-86} \mod 5869 = 1729,$$

te provjerom vrijedi li $h(x \parallel 1729) = 86$.

$\Diamond$

Kriptosustav 7.1.1. (Schnorrova shema potpisa). Neka je p prost broj takav da problem diskretnog logaritma u $\mathbb{Z}_p^*$ nije riješiv u realnom vremenu, i neka je q prost broj koji dijeli $p - 1$. Neka je $\alpha \in \mathbb{Z}_p^*$ q - ti korijen od 1 modulo p. Neka $\mathcal{P} = \{0,1\}^*$, $\mathcal{A} = \mathbb{Z}_q \times \mathbb{Z}_q$, i definiramo

$$\mathcal{K} = \{(p, q, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\},$$

gdje je $0 \leq a \leq q - 1$. Vrijednosti p, q, α i β su javni ključ, dok je a tajni ključ. Konačno, neka je $h : \{0,1\}^* \to \mathbb{Z}_q$ sigurna hash funkcija.

Za $K = (p, q, \alpha, a, \beta)$, i za (tajni) slučajno odabrani broj k, $1 \leq k \leq q - 1$, definiramo

$$sig_K(x, k) = (\gamma, \delta),$$

gdje

$$\gamma = h(x \parallel \alpha^k \mod p)$$

i

$$\delta = k + a\gamma \mod q.$$

Za $x \in \{0,1\}^*$ i $\gamma, \delta \in \mathbb{Z}_q$ definiramo

$$ver_K(x,(\gamma,\delta)) = true \Leftrightarrow h(x \parallel \alpha^\delta \beta^{-\gamma} \bmod p) = \gamma.$$

$\Diamond$

7.2 Algoritam digitalnog potpisa

Sada ćemo objasnit promjene koje su izvršene na funkciji verifikacije ElGamalove sheme potpisa u specifikaciji DSA. DSA koristi podgrupu reda q od $\mathbb{Z}_p^*$, kao i Schnorrova shema potpisa. Zatim, u DSA je zahtijevano da je q 160 - bitni prost broj dok je p L - bitni prost broj takav da je $L \equiv 0 \pmod{64}$ i $512 \leq L \leq 1024$. Ključ u DSA ima isti oblik kao u Schnorrovoj shemi. Prije nego što bude potpisana, poruka u DSA će biti "hashirana" korištenjem iterativne hash funkcije $SHA-1$ (*The Secure Hash Algorithm*). Rezultat je 160 - bitna izmijenjena poruka potpisana 320 - bitnim potpisom te su izračuni izvršeni unutar skupova $\mathbb{Z}_p$ i $\mathbb{Z}_q$.

U ElGamalovoj shemi potpisa pretpostavimo da mijenjamo " $-$ " u " $+$ " u definiciji δ, tako da

$$\delta = (x + a\gamma)k^{-1} \mod (p-1).$$

Time mijenjamo uvjet verifikacije na sljedeće:

$$\alpha^x \beta^\gamma \equiv \gamma^\delta \pmod{p}.$$

Kriptosustav 7.2.1. (Algoritam digitalnog potpisa). Neka je p L - bitni prost broj takav da problem diskretnog logaritma u $\mathbb{Z}_p$ nije riješiv u realnom vremenu i $L \equiv 0 \pmod{64}$ i $512 \leq L \leq 1024$, i neka je q 160 - bitni prost broj koji dijeli $p-1$. Neka je $\alpha \in \mathbb{Z}_p^*$ q - ti korijen od 1 modulo p. Neka $\mathcal{P} = \{0,1\}^*$, $\mathcal{A} = \mathbb{Z}_q^* \times \mathbb{Z}_q^*$, i definiramo

$$\mathcal{K} = \{(p,q,\alpha,a,\beta) : \beta \equiv \alpha^a \pmod{p}\},$$

gdje je $0 \leq a \leq q-1$. Vrijednosti p, q, α i β su javni ključ, dok je a privatni ključ.

Za $K = (p,q,\alpha,a,\beta)$ i za (tajni) slučajno odabrani broj k, $1 \leq k \leq q-1$, definiramo

$$sig_K(x,k) = (\gamma,\delta),$$

gdje

$$\gamma = (\alpha^k \mod p) \mod q, \quad i$$
$$\delta = (SHA-1(x) + a\gamma)k^{-1} \mod q.$$

(Ukoliko je $\gamma = 0$ ili $\delta = 0$ potrebno je odabrati novu vrijednost k.)

Za $x \in \{0,1\}^*$ i $\gamma, \delta \in \mathbb{Z}_q^*$ definiramo

$$e_1 = SHA-1(x)\delta^{-1} \mod q$$
$$e_2 = \gamma^{\delta^{-1}} \mod q$$

$$ver_K(x,(\gamma,\delta)) = true \Leftrightarrow (\alpha^{e_1}\beta^{e_2} \mod p) \mod q = \gamma.$$

$\Diamond$

Znači, α je reda q. β i γ su također reda q jer su potencije od α. Dakle, sve potencije u $\alpha^x\beta^\gamma \equiv \gamma^\delta \pmod p$ mogu biti reducirane modulo q bez utjecaja na valjanost kongruencije. S obzirom da ćemo poruku x zamijeniti 160 - bitnom izmijenjenom porukom u DSA, pretpostavit ćemo da je $x \in \mathbb{Z}_q$. Nadalje, $\delta \in \mathbb{Z}_q$ ćemo definirati kao

$$\delta = (x + a\gamma)k^{-1} \mod q.$$

Ostaje razmotriti slučaj $\gamma = \alpha^k \mod p$. Pretpostavimo da privremeno definiramo

$$\gamma' = \gamma \mod q = (\alpha^k \mod p) \mod q.$$

Imajte na umu da

$$\delta = (x + a\gamma')k^{-1} \mod q,$$

tako da je δ nepromijenjen. Jednadžbu verifikacije možemo zapisati kao

$$\alpha^x\beta^{\gamma'} \equiv \gamma^\delta \pmod p.$$

Primijetite da ne možemo zamijeniti ostatak od γ sa γ'. Sada nastavljamo sređivati $\alpha^x\beta^{\gamma'} \equiv \gamma^\delta \pmod p$ potenciranjem obiju strana $\delta^{-1} \mod q$ (ovaj korak zahtijeva $\delta \neq 0$). Dobivamo sljedeće:

$$\alpha^{x\delta^{-1}}\beta^{\gamma'\delta^{-1}} \mod p = \gamma.$$

Nadalje, reducirat ćemo obje strane prethodne jednadžbe čime dobivamo:

$$(\alpha^{x\delta^{-1}}\beta^{\gamma'\delta^{-1}} \mod p) \mod q = \gamma'.$$

U listopadu 2001., NIST je predložio da p bude izabran kao 1024 - bitni prost broj (tj. 1024 je jedina dopuštena vrijednost za L). Ovime se ukazivala zabrinutost oko sigurnosti problema diskretnog logaritma.

Primjer 7.2.1. Pretpostavimo da su $p = 5869$, $q = 163$, $\alpha = 1326$, $a = 35$, $\beta = 4958$ i $k = 23$ (kao u *Primjeru 7.1.1.*) i pretpostavimo da Ana želi potpisati izmijenjenu poruku $SHA - 1(x) = 33$. Tada ona računa

$$k^{-1} \mod 163 = 23^{-1} \mod 163 = 78,$$

$$\begin{aligned} \gamma &= (1326^{23} \mod 5869) \mod 163 \\ &= 1729 \mod 163 \\ &= 99, \end{aligned}$$

$$\begin{aligned} \delta &= (33 + 35 \cdot 99) \cdot 78 \mod 163 \\ &= 145. \end{aligned}$$

Potpis $(99, 145)$ se na izmijenjenoj poruci 33 potvrđuje računanjem

$$\begin{aligned} \delta^{-1} = 145^{-1} &\mod 163 = 9 \\ e_1 = 33 \cdot 9 &\mod 163 = 134 \\ e_2 = 99 \cdot 9 &\mod 163 = 76 \end{aligned}$$

$$(1326^{134} 4958^{76} \mod 5869) \mod 163 = 1729 \mod 163 = 99.$$

$\Diamond$

Kada je DSA predložen 1991. postavljeno je nekoliko kritika. Jedan prigovor se odnosio na NITS - ov postupak odabira koji nije javan. Standard je razvijen od strane Nacionalne agencije za sigurnost (eng. *National Security Agency*, skraćeno NSA) bez ulaganja američke industrije. Bez obzira na zasluge dobivene sheme, mnogi su zamjerili pristup "*zatvorenih vrata*".

Od iznesenih tehničkih kritika, najozbiljnija je ona koja govori da je veličina modula p fiksirana početno na 512 bita. Mnogi su sugerirali da veličina modula ne bi trebala biti fiksirana, tako da se po želji mogu upotrijebiti veći moduli. Kao odgovor na primjedbe, NIST je izmijenio standard na način da je dopušten čitav niz veličina modula.

LITERATURA

[1] A. Dujella: *Uvod u teoriju brojeva*; skripta, PMF - Matematički odjel, Sveučilište u Zagrebu, 2002.

[2] J. Hoffstein, J. Pipher, J. H. Silverman: *An Introduction to Mathematical Cryptography*; Springer - Verlag, New York, 2008.

[3] I. Matić: *Uvod u teoriju brojeva*; Odjel za matematiku Sveučilišta J. J. Strossmayera u Osijeku, 2015.

[4] D. R. Stinson: *Cryptography: Theory and Practice, Third Edition - Solutions Manual*; Chapman & Hall/CRC, Sveučilište Waterloo, Ontario, 2006.

[5] M. Vuković: *Matematička logika 1*; skripta, PMF - Matematički odjel, Sveučilište u Zagrebu, 2007.

SAŽETAK

Glavni cilj ovog rada je predstaviti ElGamalovu shemu potpisa i neke njene varijante. Rad je podijeljen na sedam poglavlja, uključujući uvodni dio. Nakon što dobijemo kratak uvid u to što je digitalni potpis i koja je njegova primjena u našoj svakodnevnici, u drugom poglavlju ćemo se osvrniti na osnovne pojmove iz teorije brojeva i kriptografije radi boljeg razumijevanja sadržaja rada. Treće i četvrto poglavlje bazira se na ozbiljnijim problemima koji mogu narušiti sigurnost digitalnog potpisa te načinima na kojima se isti mogu spriječiti. Peto poglavlje predstavlja matematički problem koji se javlja u mnogim područjima pa tako i modulo p, a to je problem diskretnog logaritma. Kroz šesto poglavlje prvo ćemo se upoznati s ElGamalovom shemom potpisa objavljenom u radu 1985., zatim osvrnuti na njenu sigurnosti. Kako bi nam shema bila jasnija proći ćemo kroz nekoliko primjera. Posljednje poglavlje rada bavi se varijantama ElGamalove sheme potpisa, a to su Schnorrova shema potpisa predložena 1989. i Algoritam digitalnog potpisa prvi puta predložen 1991.

Ključne riječi:
digitalni potpis, kriptosustav, RSA shema potpisa, sigurnost digitalnog potpisa, hash funkcija, problem diskretnog logaritma, ElGamalova shema potpisa, Schnorrova shema potpisa, algoritam digitalnog potpisa.

TITLE AND SUMMARY

Signature Schemes

The main purpose of this paper is to present ElGamal's signature scheme and some of its variants. The work is divided into seven chapters, including the introductory part. After we get a brief insight into what the digital signature is and what its application is in our everyday life, for a better understanding of the content of the work the second chapter will cover the basic concepts from the theory of numbers and cryptography. The third and fourth chapters are based on more serious issues that can violate the security of the digital signature and the ways in which it can be prevented. The fifth chapter represents a mathematical problem that occurs in many areas, such as modulo p, which is the problem of discrete logarithm. Through the sixth chapter, we will first get acquainted with ElGamal's signature scheme published in 1985, and then consider its security. In order to make the scheme clearer, we will go through several examples. The last chapter deals with the variants of the ElGamal signature scheme, which are the Schnorra signature scheme proposed in 1989 and the Digital Signature Algorithm for the first time suggested in 1991.

Keywords:
digital signature, cryptosystem, RSA signature scheme, security requirements for digital signature, hash function, the discrete logarithm problem, the ElGamal signature scheme, the Schnorr signature scheme, the digital signature algorithm.

ŽIVOTOPIS

Sanja Rendulić rođena je 24. siječnja 1992. godine u Zagrebu. Pohađala je Osnovnu školu Vladimir Nazor u Đakovu te nakon osnovnoškolskog obrazovanja upisuje opću Gimnaziju A. G. Matoša Đakovo. Nakon polaganja državne mature daljnje obrazovanje nastavlja na Sveučilištu J. J. Strossmayera u Osijeku. Diplomirala je na Odjelu za matematiku Sveučilišni preddiplomski studij matematike i stekla akademski naziv *sveučilišna prvostupnica (baccalaurea) matematike*. Nakon toga upisuje diplomski studij matematike na Odjelu za matematiku u Osijeku, smjer Financijska matematika i statistika. Osim dugogodišnjeg aktivnog bavljenja sportom, tijekom studiranja daje poduke iz područja matematike. Tijekom završne godine diplomskog studija odradila je stručnu praksu u *Hrvatskoj agenciji za hranu* gdje joj je osnovna zadaća bila procjena izloženosti određenog kemijskog i mikrobiološkog kontaminanta hrane za odraslu populaciju u RH.