Digitalni potpis pomoću eliptičkih krivulja

Zima, Mihaela

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Mihaela Zima

DIGITALNI POTPIS POMOĆU ELIPTIČKIH KRIVULJA

Diplomski rad

Voditelj rada:
prof. dr. sc. Andrej Dujella

Zagreb, rujan 2023.

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 ovom radu ćemo najprije definirati osnovne pojmove kriptografije koje ćemo koristiti kroz ostala poglavlja. Od velike važnosti će nam biti pojam kriptografije javnog ključa jer će se na njoj temeljiti metoda za generiranje digitalnih potpisa pomoću eliptičkih krivulja (ECDSA). Literatura za ovo poglavlje je [6], [11] i [4].

U drugom poglavlju ćemo uvesti pojam digitalnog potpisa. Objasnit ćemo što je digitalni potpis i kako funkcionira, te ćemo govoriti o sigurnosti digitalnog potpisa i potencijalnim problemima s kojima se možemo susresti pri napadu. Literatura za ovo poglavlje je [2].

U trećem poglavlju ćemo definirati hash funkcije. Uvest ćemo dvije podjele hash funkcija, te ćemo govoriti o njihovoj sigurnosti. Prva podjela će nam biti na hash funkcije bez ključa i s ključem. Hash funkcija bez ključa nam je sigurna ako je teško riješiti probleme jednosmjernosti, jednoznačnosti i jake otpornosti na koliziju. Druga podjela će nam biti na MDC i MAC, te ćemo navesti njihova osnovna svojstva i veze između svih definiranih svojstava. Literatura za ovo poglavlje je [11] i [8].

Zatim ćemo uvesti pojam eliptičke krivulje nad proizvoljnim poljem $\mathbb{K}$, te ćemo, u ovisnosi o karakteristici polja $\mathbb{K}$, općeniti oblik eliptičke krivulje svesti na kraći oblik. Ono što će nama biti od velike važnosti su eliptičke krivulje nad konačnim poljem. Pretpostavit ćemo da polje $\mathbb{F}$ ima q elemenata i uvest ćemo dva najbitnija slučaja: $q = p$ prost broj i $q = 2^m$. Poglavlje ćemo završiti uvođenjem osnovnih operacija na eliptičkim krivuljama kao što su zbrajanje, oduzimanje i dupliciranje točaka. Literatura za ovo poglavlje je [6], [5], [8], [12] i [11].

Sljedeći korak nam je reći nešto o problemu diskretnog logaritma nad eliptičkim krivuljama (ECDLP). To je teško rješivi problem pa će nam biti bitan za sigurnost ECDSA. ECDLP je zapravo problem pronalaska broja n takvog da $Q = nP$, gdje su P i Q točke na eliptičkoj krivulji. n nam zapravo označava eliptički diskretni logaritam. Definirat ćemo "Dupliciraj i zbroji" algoritam koji se koristi za računanje nP. Na kraju poglavlja ćemo spomenuti index calculus metodu i Shanksovu "baby step-giant step" (BSGS) metodu. Literatura za ovo poglavlje je [7] i [5].

Na kraju ćemo napokon uvesti ECDSA algoritam u kojem ćemo koristiti sve što smo uveli u prethodnim poglavljima. ECDSA je generalizacija algoritma digitalnog potpisa

(DSA) pa ćemo, radi boljeg razumijevanja, prvo proučiti potpisivanje poruke u DSA te provjeru vjerodostojnosti potpisa. Nakon što uvedemo algoritme za potpisivanje poruke i provjeru vjerodostojnosti potpisa za ECDSA, proučit ćemo njegovu sigurnost. Uvest ćemo definicije različitih krivotvoritelja te pronaći nužne i dovoljne uvjete pod kojima je ECDSA najsigurniji. Za kraj ćemo proučiti različite varijante ECDSA, te njegove napadačke osobine. Literatura za ovo poglavlje je [1] i [6].

Diplomski rad napravljen je u sklopu aktivnosti Projekta KK.01.1.1.01.0004 - Znanstveni centar izvrsnosti za kvantne i kompleksne sustave te reprezentacije Liejevih algebri.

Poglavlje 1

Osnovni pojmovi u kriptografiji

Prisjetimo se definicije kriptografije, te nekih osnovnih pojmova.

Kriptografija (grč. kryptos - tajanstven, graphy- pisati) je znanstvena disciplina koja se bavi proučavanjem metoda koje nam omogućuju prijenos osjetljivih podataka preko nesigurnih komunikacijskih kanala tako da ih može pročitati samo onaj kome su namijenjene.

Zadatak je omogućiti komunikaciju između dvije osobe, u literaturi obično susrećemo pod nazivom Alice i Bob, tako da treća osoba, nazovimo je Eve, ne može razumjeti njihove poruke. Alice i Bob ćemo nazivati *pošiljateljem* i *primateljem*, dok je Eve njihov *protivnik*.

Otvoreni tekst je smislena i razumljiva poruka koju pošiljatelj želi poslati primatelju. Metoda transformiranja otvorenog teksta na takav način da se sakrije njegovo pravo značenje naziva se *šifriranje* i odvija se uz pomoć *ključa* K koji znaju i pošiljatelj i primatelj. Šifriranu poruka koju pošiljatelj šalje primatelju nazivamo *šifrat*. Proces vraćanja šifriranog teksta, uz pomoć ključa, u izvorni otvoreni tekst naziva se *dešifriranje*. Primatelj zna ključ K, pa može pomoću njega dešifrirati šifrat, tj. odrediti otvoreni tekst. Protivnik može otkriti sadržaj šifrata prisluškujući komunikacijski kanal, ali kako ne zna ključ, ne može dešifrirati šifrat i odrediti otvoreni tekst.

Definicija 1.0.1. *Kriptosustav je uređena petorka* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, gdje je*

- $\mathcal{P}$ *konačan skup svih otvorenih tekstova,*
- $\mathcal{C}$ *konačan skup svih šifrata,*
- $\mathcal{K}$ *konačan skup svih mogućih ključeva,*
- $\mathcal{E}$ *skup svih funkcija šifriranja,*
- $\mathcal{D}$ *skup svih funkcija dešifriranja.*
- *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* $x \in \mathcal{P}$*.*

Funkcija e_K je funkcija šifriranja, a funkcija d_K je funkcija dešifriranja i obadvije ovise o ključu K.

Kriptosustavi se mogu općenito podijeliti na dvije vrste prema vrsti ključa: kriptosustavi s tajnim ključem (simetrični kriptosustavi) i kriptosustavi s javnim ključem (asimetrični kriptosustavi). Kod simetričnih kriptosustava pošiljatelj i primatelj izabiru tajni ključ i pomoću njega se generiraju funkcije šifriranja i dešifriranja. Ključ za dešifriranje lako se izvodi iz ključa za šifriranje i obratno. Kod ovih sustava, za šifriranje i dešifriranje najčešće se koristi isti ključ. Sigurnost ovih kriptosustava leži u tajnosti ključa. Neki važniji simetrični kriptosustavi su DES, 3DES, AES, Blowfish i TwoFish.

Revolucionarnu ideju kriptosustava s javnim ključem uveli su 1976. Diffie i Hellman kako bi riješili nedostatke kriptografije sa simetričnim ključem, tj. nedostatak sigurnog komunikacijskog kanala. Oni su ponudili jedno moguće rješenje problema razmjene ključeva, zasnovano na činjenici da je u nekim grupama potenciranje puno jednostavnije od logaritmiranja. Diffieja i Hellmana smatramo začetnicima kriptografije javnog ključa.

Kod kriptosustava s javnim ključem ključ za dešifriranje se ne može izračunati iz ključa za šifriranje. Svaki korisnik ima *javni* i *tajni* ključ. Kod takvih kriptosustava je funkcija e_K javna i nazivamo ju javni ključ, dok je d_K tajni ključ. Svaka osoba koja ima javni ključ može šifrirati poruku pomoću njega, ali samo osoba koja ima tajni ključ može dešifrirati tu poruku. Neki važniji kriptosustavi s javnim ključem su RSA, ElGamal i Menezes-Vanstone. Sigurnost RSA kriptosustava se zasniva na problemu faktorizacije. Sigurnost ElGamal kriptosustava se temelji na problemu diskretnog logaritma, dok se sigurnost Menezes-Vanstone na problemu eliptičkog diskretnog logaritma.

Kriptosustavi s javnim ključem su puno sporiji od simetričnih kriptosustava. U primjenama se često ova dva kriptosustava koriste zajedno, tako da se pomoću javnog ključa šifrira nasumično generiran ključ za šifriranje, a taj ključ šifrira poruku koristeći simetrični kriptosustav.

Glavna prednost kriptografije s javnim ključem je u tome što pruža metodu za korištenje *digitalnih potpisa*. Potreba za digitalnim potpisom je nastala iz želje da znamo tko nam je poslao poruku. Točnije, ako pošiljatelj (Alice) šalje poruku primatelju (Bobu), Bob želi biti siguran da mu je tu poruku poslala upravo Alice, a ne netko drugi. Također, digitalni potpis omogućuje primatelju da potvrdi da informacija nije promijenjena tijekom prijenosa. Još jedna bitna karakteristika digitalnog potpisa je neporecivost, što znači da pošiljatelj ne može poreći da je upravo on poslao poruku.

Poglavlje 2

Digitalni potpisi

Digitalni potpis koristi se za potpisivanje elektroničkih dokumenata. Takvi potpisi imaju svojstva slična vlastoručnim potpisima. Ako Alice potpiše dokument vlastoručnim potpisom, tada svatko tko vidi dokument i tko zna Alicein potpis može potvrditi da je dokument potpisala upravo Alice. Digitalni potpisi se koriste kod npr. elektroničkih ugovora, elektroničkih bankovnih transakcija, koji moraju biti potpisani. Objasnimo kako digitalni potpisi funkcioniraju. Pretpostavimo da Alice želi potpisati dokument m. Ona tada koristi tajni ključ d i izračunava potpis s. Koristeći javni ključ e, Bob može potvrditi da je s potpis od dokumenta m. Svaka shema digitalnog potpisa se sastoji od tri dijela:

1. algoritma generiranja javnog i tajnog ključa,
2. algoritma za generiranje digitalnog potpisa (koristi tajni ključ potpisa)
3. algoritma za provjeru digitalnog potpisa (koristi javni verifikacijski ključ).

2.1 Sigurnost sheme digitalnog potpisa

Shema digitalnog potpisa je sigurna ako je gotovo nemoguće konstruirati tajni ključ iz javnog ključa. Sheme potpisa koje se koriste u današnje vrijeme imaju ovo svojstvo. Temelje se na računskim problemima iz teorije brojeva koji se smatraju nerješivima. Ne postoje dokazi za nerješivost tih problema.

Pronalaženje tajnog ključa potpisa nije jedini mogući cilj napadača. Napadač može pokušati generirati nove valjane potpise bez znanja tajnog ključa potpisa. Ovo nazivamo *egzistencijalnom krivotvorinom*.

Točnije, napadač postupa na sljedeći način:

1. Napadač dobiva Alicein javni verifikacijski ključ.

2. Napadač izračuna poruku x i potpis za x koji se može provjeriti s Alicinim verifikacijskim ključem.

Napadač može izračunati dokument x kao funkciju javnog verifikacijskog ključa. U sljedećem paragrafu ćemo opisati napad koji također koristi znanje o valjanim potpisima drugih dokumenata. Ovdje je situacija jednostavnija jer se ne koriste valjani potpisi drugih dokumenata. Stoga se ovaj napad naziva *napad bez poruke*. Jasno je da napadač može jednostavno pogoditi potpis. Tada postoji mala vjerojatnost da će taj potpis biti valjan. Shema potpisa naziva se sigurnom protiv napada bez poruke ako napadač s polinomijalnim vremenom ne može pokrenuti takav napad koji je uspješan s nezanemarivom vjerojatnošću.

Nije dovoljno da je shema potpisa sigurna protiv napada bez poruke. Moguće je da napadač zna valjane potpise i pomoću njih konstruira nove potpise. Čak je moguće da napadač može dobiti potpise po svom izboru prije nego što generira novi potpis. Kod napada izabrane poruke, napadaču je poznat javni verificirani ključ pošiljatelja. Napadač izračunava poruku x i potpis za x koji se može provjeriti pomoću Alicinog verifikacijskog ključa. Tijekom izračuna, napadač uvijek može dobiti potpise za dokumente po svom izboru.

Poglavlje 3

Hash funkcija

Hash funkcija u kriptografiji je funkcija koja preslikava proizvoljno duge nizove u nizove fiksne duljine. Ideja hash funkcije u kriptografiji je da osigurava integritet podataka. Pomoću hash funkcije konstruiramo kratki "fingerprint" nekih podataka. Ako se podaci izmijene, tada "fingerprint" (s velikom vjerojatnošću) više neće biti valjan. Pretpostavimo da je "fingerprint" pohranjen na sigurno mjesto. Tada, čak i ako su podaci pohranjeni na nesigurnom mjestu, njihov se integritet može provjeriti ponovnim izračunavanjem "fingerprinta" i provjerom da se "fingerprint" nije promijenio.

Neka je h hash funkcija i neka je x neki podatak, tj. neka je x binaran niz proizvoljne duljine. Tada je odgovarajući "fingerprint" definiran sa $y = h(x)$. "Fingerprint" se često naziva "sažetkom poruke". Pretpostavimo da je y spremljen na sigurno mjesto, a da x nije. Dakle, može se dogoditi da se x promijeni. Označimo promijenjeni x sa x'. Tada želimo da "sažetak poruke" y, nije "sažetak poruke" od x'. Da je x promijenjen otkrivamo tako da računamo "sažetak poruke" $y' = h(x')$ i provjerimo da $y' \neq y$.

Postoje dvije vrste hash funkcija: hash funkcije s ključem i hash funkcije bez ključa. Definirajmo hash familiju koja je zapravo familija hash funkcija s ključem.

Definicija 3.0.1. *Hash funkcija je četvorka* $(\mathcal{X}, \mathcal{Y}, \mathcal{K}, \mathcal{H})$*, gdje su zadovoljeni sljedeći uvjeti:*

1. $\mathcal{X}$ *je skup mogućih poruka*
2. $\mathcal{Y}$ *je konačan skup mogućih "sažetaka poruka" ili oznaka za provjeru autentičnosti*
3. $\mathcal{K}$ *je konačan skup mogućih ključeva*
4. *Za svaki* $K \in \mathcal{K}$ *postoji hash funkcija* $h_K \in \mathcal{H}$ *takva da* $h_K : \mathcal{X} \to \mathcal{Y}$.

Svaki ključ ima hash funkciju koja je specifična za taj ključ. Ako je K tajni ključ, za svaku poruku x odgovarajuća oznaka za provjeru autentičnosti je $y = h_K(x)$. Oznaku za provjeru autentičnosti kraće nazivamo oznakom.

Hash funkcija bez ključa je funkcija $h : \mathcal{X} \to \mathcal{Y}$, gdje su $\mathcal{X}$ i $\mathcal{Y}$ kao u Definiciji 3.0.1. Hash funkciju bez ključa gledat ćemo kao hash familiju sa samo jednim ključem, tj. $|\mathcal{K}| = 1$.

Obično "sažetak poruke" nazivamo izlaznu vrijednost hash funkcije bez ključa, dok "oznakom" nazivamo izlaznu vrijednost hash funkcije s ključem.

Za par $(x, y) \in \mathcal{X} \times \mathcal{Y}$ kažemo da je *valjan par* pod hash funkcijom h ukoliko $h(x) = y$.

3.1 Sigurnost hash funkcije

Neka je $h : \mathcal{X} \to \mathcal{Y}$ hash funkcija bez ključa, $x \in \mathcal{X}$ i definiramo $y = h(x)$. Da bismo izračunali valjani par (x, y) prvo izabiremo x, pa $y = h(x)$ računamo pomoću hash funcije h i x.

Smatramo da je hash funkcija sigurna, ako je teško riješiti sljedeća tri problema:

1. Jednosmjernost: Zadani su $h : \mathcal{X} \to \mathcal{Y}$ i "sažetak poruke" $y \in \mathcal{Y}$. Tražimo $x \in \mathcal{X}$ takav da $h(x) = y$. Hash funkciju za koju nije moguće riješiti ovaj problem nazivamo jednosmjernom.

2. Jednoznačnost ili slaba otpornost na koliziju: Zadani su $h : \mathcal{X} \to \mathcal{Y}$ i $x \in \mathcal{X}$. Tražimo $x' \in \mathcal{X}$ takav da $x' \neq x$ i $h(x') = h(x)$. Hash funkciju za koju nije moguće efektivno riješiti ovaj problem zovemo jednoznačnom ili slabo otpornom na kolizije.

3. Općenita jednoznačnost ili jaka otpornost na koliziju: Zadana je $h : \mathcal{X} \to \mathcal{Y}$. Tražimo $x, x' \in \mathcal{X}$ takve da $x' \neq x$ i $h(x') = h(x)$. Hash funkciju za koju nije moguće efektivno riješiti ovaj problem zovemo općenito jednoznačnom ili jako otpornom na kolizije.

3.2 Druga klasifikacija hash funkcija

Osim podjele hash funkcija na funkcije bez ključa i funkcije s ključem, postoji još jedna klasifikacija hash funkcija koju ćemo spomenuti, a to je podjela na:

1. kodove za otkrivanje izmjena (MDC)

2. kodove autentičnosti poruke (MAC)

Kodovi za otkrivanje izmjena

MDC je podklasa hash funkcija bez ključa. Svrha MDC-a je izrada sažetka kojima se osigurava integritet podataka koji je zahtjevan od različitih aplikacija. MDC-ovi se mogu podijeliti na sljedeće dvije podvrste:

1. jednosmjerne hash funkcije (OWHF) - za njih je teško pronaći ulaz koji će hashiran dati unaprijed određenu hash-vrijednost
2. hash funkcije otporne na koliziju (CRHF) - za njih je teško pronaći dvije ulazne vrijednosti koje imaju istu hash-vrijednost

Kodovi autentičnosti poruke

MAC je podklasa hash funkcija s ključem. Kao ulaz ima dva parametra - ulaznu poruku i tajni ključ. Svrha MAC-a je olakšati, bez korištenja nekih dodatnih mehanizama, sigurnost uzimajući u obzir izvornu poruku i njezin integritet.

Osnovna svojstva i definicije

Sada možemo navesti formalne definicije sljedećih pojmova.

Definicija 3.2.1. *Jednosmjerna hash funkcija (OWHF) je hash funkcija h koja ima sljedeća svojstva:*

1. *kompresija - h preslikava ulaznu vrijednost x koja je proizvoljne konačne duljine u izlaznu vrijednost $h(x)$ koja je fiksne duljine n,*
2. *jednostavnost izračuna - dani su h i ulazna vrijednost x, a $h(x)$ je lako izračunati,*

i svojstva jednosmjernosti i jednoznačnosti.

Definicija 3.2.2. *Hash funkcija otporna na koliziju (CRHF) je hash funkcija h koja ima sljedeća svojstva:*

1. *kompresija - h preslikava ulaznu vrijednost x koja je proizvoljne konačne duljine u izlaznu vrijednost $h(x)$ koja je fiksne duljine n,*
2. *jednostavnost izračuna - dani su h i ulazna vrijednost x, a $h(x)$ je lako izračunati,*

i svojstva jednoznačnosti i općenite jednoznačnosti.

Iako se ne navodi u definiciji, u praksi CRHF gotovo uvijek ima svojstvo jednosmjernosti.

Definicija 3.2.3. *MAC algoritam je familija funkcija h_k s parametrom k koji je tajni ključ, sa sljedećim svojstvima:*

1. *jednostavnost izračuna: za poznatu funkciju h_k, danu vrijednost k i ulaznu vrijednost x, lako je izračunati $h_k(x)$. Ovaj rezultat zovemo MAC vrijednost ili MAC.*

2. *kompresija - h_k preslikava ulaznu vrijednost x koja je proizvoljne konačne duljine u izlaznu vrijednost $h_k(x)$ koja je fiksne duljine n.*

3. *računska otpornost - ako nam je dano* 0 *ili više MAC parova $(x_i, h_k(x_i))$, računski je neizvedivo izračunati bilo koji MAC par $(x, h_k(x))$ za bilo koji novi ulaz $x \neq x_i$.*

Ako računska otpornost nije ispunjena, MAC algoritam je podložan MAC krivotvorenju. Računalna otpornost povlači svojstvo nemogućnosti otkrivanja ključa (računski mora biti nemoguće otkriti k), obratno ne vrijedi.

Napomena 3.2.4. *Veze između svojstava:*

1. *Hash funkcija sa svojstvom jake otpornosti na koliziju je ujedno i funkcija slabo otporna na koliziju.*

2. *Jaka otpornost na koliziju ne garantira jednosmjernost.*

3.3 Ostala svojstva hash funkcija i primjene

Većina hash funkcija bez ključa u praksi je obično originalno dizajnirana u svrhu pružanja integriteta podataka, uključujući i potvrdu identiteta pomoću digitalnog potpisa. Većina tih funkcija su ustvari MDC-ovi dizajnirani da imaju svojstvo jednosmjernosti, slabe otpornosti na koliziju ili jake otpornosti na koliziju. MDC-ovi se koriste u raznim primjenama kao što su potvrda znanja određenih podataka, izvođenje ključa i generiranje pseudoslučajnih brojeva. Hash funkcije koje se koriste za potvrdu znanja određenih podataka olakšavaju predaju vrijednosti podataka ili pokazuju posjedovanje podataka bez da otkriju te podatke. Hash funkcije za izvođenje ključa se koriste za izračunavanje nizova novih ključeva iz prethodnih ključeva. Hash funkcije za generiranje pseudoslučajnih brojeva koriste se za generiranje nizova brojeva koji imaju različita svojstva slučajnosti.

Mnogi MDC-ovi koji se koriste u praksi mogu zadovoljavati dodatne zahtjeve osim onih za koje su izvorno dizajnirani. Unatoč tome, korištenje proizvoljnih hash funkcija ne preporuča se ni za jednu primjenu bez pažljive analize koja precizno identificira i kritična svojstva koja zahtijeva primjena i ona koja pruža predmetna funkcija.

Dodatna svojstva jednosmjernih hash funkcija

Dodatna svojstva jednosmjernih hash funkcija koja zahtijevaju gore navedene primjene su sljedeća:

1. međusobna nepovezanost - ulazni i izlazni podaci ne bi trebali biti u korelaciji
2. otpornost na blisku koliziju - Trebalo bi biti teško naći bilo koje dvije ulazne vrijednosti x i x' takve da se $h(x)$ i $h(x')$ razlikuju samo u malom broju bitova.
3. djelomična jednosmjernost ili lokalna jednosmjernost - Trebalo bi biti jednako teško obnoviti bilo koji podniz kao i obnoviti cijelu ulaznu vrijednost. Štoviše, čak i ako je dio ulaza poznat, trebalo bi biti teško pronaći ostatak.

Mnoga od ovih svojstava mogu se sažeti kao zahtjevi da ne postoje ni lokalne ni globalne statističke slabosti. Hash funkcija ne smije biti slabija u odnosu na neke dijelove svog ulaza ili izlaza od drugih, a svi bitovi trebaju biti jednako teški. Neka od njih mogu se nazvati *certifikacijskim svojstvima* – svojstvima koja se intuitivno čine poželjnima, iako se ne može pokazati da su izravno potrebna.

Poglavlje 4

Eliptičke krivulje

4.1 Uvod u eliptičke krivulje

Definirajmo opći oblik eliptičke krivulje, nad proizvoljnim poljem.

Definicija 4.1.1. *Neka je $\mathbb{K}$ proizvoljno polje. Eliptička krivulja E nad $\mathbb{K}$ je nesingularna krivulja oblika*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$

gdje su $a_1, a_2, a_3, a_4, a_6 \in \mathbb{K}$.

Ova se jednadžba naziva *Weierstrassova forma*. Nesingularnost znači da je u svakoj točki na barem jedna od parcijalnih derivacija različita od 0.

Definicija 4.1.2. *Neka je $\mathbb{K}$ polje. Karakteristika polja $\mathbb{K}$ je najmanji prirodni broj n takav da je $1 + 1 + \ldots + 1 = n \cdot 1 = 0$, gdje je 0 neutralni elementi za zbrajanje, a 1 neutralni element za množenje u $\mathbb{K}$. Ako je $n \cdot 1 \neq 0$ za svaki prirodan broj n, onda se kaže da je $\mathbb{K}$ polje karakteristike 0.*

Weierstrassova forma može biti pojednostavljena, ovisno o karakteristici polja $\mathbb{K}$.

Ako je karakteristika polja $\mathbb{K}$ različita od 2 i 3, onda nadopunjavanjem do potpunog kvadrata i potpunog kuba, gornja jednadžba se može svesti na sljedeći oblik

$$y^2 = x^3 + ax + b$$

gdje su $a, b \in \mathbb{K}$. Ovu jednadžbu nazivamo *kratka Weierstrassova forma*. Uvjet nesingularnosti je da je diskriminanta ove krivulje $D = -16(4a^3 + 27b^2) \neq 0$.

Ako je karakteristika polja $\mathbb{K}$ jednaka 2, onda postoje dva slučaja.

1. Ako je $a_1 \neq 0$, onda Weierstrassovu formu možemo svesti na oblik

$$y^2 + xy = x^3 + ax^2 + b,$$

gdje su $a, b \in \mathbb{K}$. Diskriminanta ove krivulje je $D = b$.

2. Ako je $a_1 = 0$, onda Weierstrassovu formu možemo svesti na oblik

$$y^2 + cy = x^3 + ax + b,$$

gdje su $a, b \in \mathbb{K}$. Diskriminanta ove krivulje je $D = c^4$.

Ako je karakteristika polja $\mathbb{K}$ jednaka 3, onda postoje dva slučaja.

1. Ako je $a_1^2 \neq -a_2$, onda Weierstrassovu formu možemo svesti na oblik

$$y^2 = x^3 + ax^2 + b,$$

gdje su $a, b \in \mathbb{K}$. Diskriminanta ove krivulje je $D = -a^3 b$.

2. Ako je $a_1^2 = -a_2$, onda Weierstrassovu formu možemo svesti na oblik

$$y^2 = x^3 + ax + b,$$

gdje su $a, b \in \mathbb{K}$. Diskriminanta ove krivulje je $D = -a^3$.

4.2 Eliptičke krivulje nad konačnim poljem

Kod primjena u kriptografiji, eliptičke krivulje se promatraju nad konačnim poljima.

Definicija 4.2.1. *Neprazan skup $G = (G, \cdot)$, gdje je $\cdot : G \times G \to G$ binarna operacija, zove se grupa ako vrijede sljedeća svojstva:*

1. $(x \cdot y) \cdot z = x \cdot (y \cdot z)$, $\forall x, y, z \in G$ *(asocijativnost),*
2. $(\exists e \in G)\ e \cdot x = x \cdot e = x, \forall x \in G$ *(neutralni element),*
3. $(\forall x \in G)(\exists x^{-1} \in G)\ x \cdot x^{-1} = x^{-1} \cdot x = e$ *(inverzni element).*

Ako još vrijedi i svojstvo

4. $x \cdot y = y \cdot x$, $\forall x, y \in G$ *(komutativnost),*

onda kažemo da je G komutativna ili Abelova grupa. Inače je G nekomutativna ili ne-Abelova grupa.

Definicija 4.2.2. *Za proizvoljan podskup S neke grupe G, definirajmo*

$$\langle S \rangle := \bigcap_{\substack{H \leq G \\ S \subseteq H}} H$$

To je podgrupa od G koju zovemo *grupa generirana* s S, a sam skup S zovemo *skup generatora*. Kažemo da je grupa G konačno generirana ako postoji konačan podskup $S = \{x_1, \dots x_k\}$ takav da je $G = \langle S \rangle$ i pišemo $G = \langle x_1, \dots, x_k \rangle$. Grupa G je *ciklička* ako se može generirati jednim elementom g, tj. ako postoji $g \in G$ takav da je $G = \langle g \rangle$. Takav g nazivamo *generator* cikličke grupe G.

Napomena 4.2.3. *Kažemo da je $(G, \cdot)$ grupoid ako za proizvoljne $x, y \in G$ vrijedi $x \cdot y \in G$. Grupoid u kojemu vrijedi i asocijativnost zove se polugrupa. Polugrupa koja ima jedinstveni neutralni element zove se monoid. Monoid u kojem postoji inverz svakog elementa je grupa.*

Definicija 4.2.4. *Prsten je uređena trojka $(R, +, \cdot)$ takva da je*

1. *$(R, +)$ Abelova grupa,*
2. *$(R, \cdot)$ polugrupa,*
3. *$(a + b) \cdot c = a \cdot c + b \cdot c$, $\forall a, b, c \in R$,*
   $a \cdot (b + c) = a \cdot b + a \cdot c$, $\forall a, b, c \in R$

Primjer 4.2.5. *Grupa $(\mathbb{Z}/n\mathbb{Z}, +)$ je konačna ciklička grupa. Tu grupu nazivamo grupa ostataka modulo n i pišemo $\mathbb{Z}/n\mathbb{Z} = \{0, 1, \dots n - 1\}$.*

Primjer 4.2.6. *$\mathbb{Z}_p = (\mathbb{Z}/p\mathbb{Z}, +, \cdot)$ je prsten uz operacije zbrajanja i množenja modulo p. Prsten $\mathbb{Z}_p$ je polje ako i samo ako je p prirodan prost broj. Za bilo koji prost broj p definiramo konačno polje cijelih brojeva modulo p u oznaci $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z} = \{0, 1, 2, \dots, p - 1\}$.*

Neka je q prirodan broj i neka je $\mathbb{F}$ konačno polje s q elemenata. Tada kažemo da je q *red* konačnog polja $\mathbb{F}$. Konačno polje $\mathbb{F}$ reda q postoji ako i samo ako je q potencija prostog broj, tj. $q = p^m$ gdje je p prost broj koji je karakteristika polja $\mathbb{F}_q$, a m je prirodan broj. Ako je $m = 1$, onda F_p nazivamo *prostim poljem*. Za bilo koju prostu potenciju q, postoji samo jedno konačno polje reda q, tj. bilo koja dva konačna polja reda q su izomorfna i označavamo takvo polje s $\mathbb{F}_q$. Elementi polja $\mathbb{F}_q$ različiti od nule tvore Abelovu grupu s obzirom na množenje. Označimo tu grupu sa $\mathbb{F}_q^*$. Ta grupa je ciklička uz operaciju množenja. Postoje generatori g grupe $\mathbb{F}_q^*$ takvi da

$$\mathbb{F}_q^* = \{g^j : 0 \leq j \leq q - 2\}.$$

Red elementa $a \in \mathbb{F}_q^*$ je najmanji prirodan broj i takav da $a^i = 1$. U primjeni eliptičkih krivulja u kriptografiji, od velike su važnosti sljedeća dva slučaja

1. $q = p$ prost broj
2. $q = 2^m$ potencija broja 2

Neka je sada p prost broj. Cijeli brojevi modulo p činit će konačno polje reda p, gdje će se polje zapravo sastojati od cijelih brojeva $\{0, 1, \ldots, p-1\}$ s operacijama zbrajanja i množenja modulo p. Ovo polje ćemo označiti sa $\mathbb{F}_p$ i to je polje ostataka modulo p.

Konačno polje reda 2^m zovemo *binarno polje* ili *konačno polje karakteristike dva* i označavamo ga $\mathbb{F}_{2^m}$. To polje se može promatrati kao vektorski prostor dimenzije m nad poljem $\mathbb{F}_2$ koje se sastoji od dva elementa 0 i 1. Dakle, svaki element $x \in \mathbb{F}_{2^m}$ se može prikazati na jedinstven način u sljedećem obliku

$$x = a_0 x_0 + a_1 x_1 + \ldots + a_{m-1} x_{m-1}$$

gdje su $a_i \in \{0, 1\}$, a $x_0, \ldots, x_{m-1} \in \mathbb{F}_2$.

Skup $\{x_0, \ldots, x_{m-1}\}$ nazivamo bazom $\mathbb{F}_{2^m}$ nad poljem $\mathbb{F}_2$. Postoji više različitih baza $\mathbb{F}_{2^m}$ nad $\mathbb{F}_2$, ali mi ćemo spomenuti dvije takve baze: polinomijalnu bazu i normalnu bazu.

Neka je $f(x) = x^m + f_{m-1}x^{m-1} + \ldots + f_2 x^2 + f_1 x + f_0$, gdje su $f_i \in \{0, 1\}$ za $i = 0, 1, \ldots, m-1$, ireducibilni polinom stupnja m nad $\mathbb{F}_2$. To znači da se $f(x)$ ne može faktorizirati kao produkt dva polinoma nad $\mathbb{F}_2$ čiji je stupanj manji od m. Konačno polje $\mathbb{F}_{2^m}$ čine svi polinomi nad $\mathbb{F}_2$ čiji je stupanj manji od m

$$\mathbb{F}_{2^m} = \{a_{m-1}x^{m-1} + \ldots + a_2 x^2 + a_1 x + a_0 : a_i \in \{0, 1\}\}$$

Ovo nazivamo reprezentacija pomoću *polinomijalne baze*.

Odaberimo ireducibilan binarni polinom $f(x)$ stupnja m. Zbrajanje elemenata polja $\mathbb{F}_{2^m}$ je uobičajeno zbrajanje polinoma pri čemu se aritmetika koeficijenata računa modulo 2. Množenje elemenata polja $\mathbb{F}_{2^m}$ izvodi se tako da prvo pomnožimo polinome, a zatim računamo umnožak modulo redukcijski polinom $f(x)$. Redukcija modulo $f(x)$ je operacija takva da svaki binarni polinom $a(x)$, $a(x) \mod f(x)$ će označavati jedinstveni polinom ostatka $r(x)$ stupnja manjeg od m dobivenog dugim dijeljenjem $a(x)$ s $f(x)$.

Normalna baza od $\mathbb{F}_{2^m}$ nad $\mathbb{F}_2$ je baza oblika

$$\{\beta, \beta^2, \beta^{2^2}, \ldots, \beta^{2^{m-1}}\}$$

gdje je $\beta \in \mathbb{F}_{2^m}$. Takva baza uvijek postoji. Svaki element $a \in \mathbb{F}_{2^m}$ se može prikazati pomoću normalne baze kao $a = \sum_{i=0}^{m-1} a_i \beta^{2^i}$. Reprezentacija pomoću normalne baze ima računsku prednost jer kvadriranje elemenata u polju postaje trivijalno: ako je $a = (a_0, a_1, \ldots a_{m-1})$, onda je $a = (a_{m-1}, a_0, a_1, \ldots a_{m-2})$. Za općenitu normalnu bazu, množenje

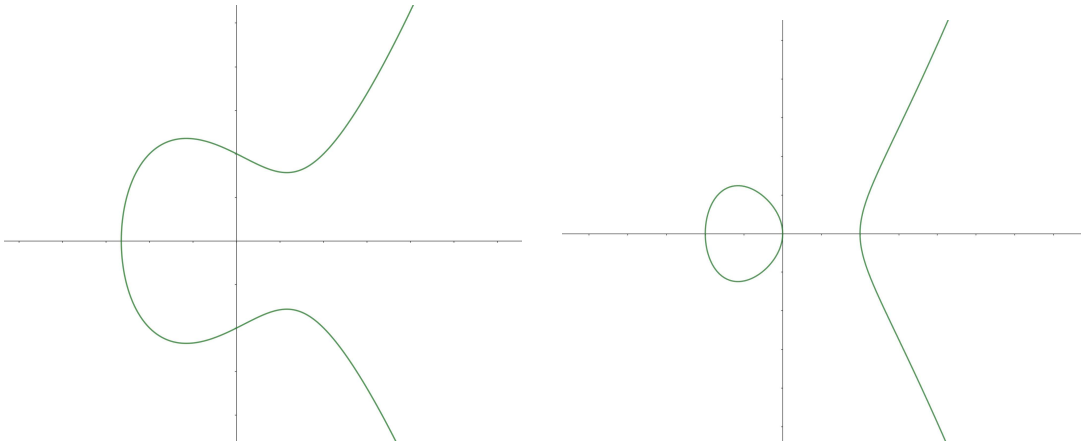
u polju je znatno kompliciranije pa su od velikog značaja *optimalne normalne baze*. To su normalne baze kod kojih je množenje jednostavnije. Alternativna karakterizacija kaže da b generira optimalnu normalnu bazu ako i samo ako za sve k_1, k_2, $0 \leq k_1 < k_2 \leq m-1$, postoje cijeli brojevi l_1, l_2 takvi da vrijedi

$$b^{2^{k_1}+2^{k_2}} = b^{2^{l_1}} + b^{2^{l_2}}$$

Optimalna normalna baza ne mora postojati. Nužan uvjet za njihovo postojanje je da je ili $m+1$ ili $2m+1$ prost. Na primjer, ako je $m+1$ prost broj i 2 primitivni korijen modulo $m+1$, tada m netrivijalnih $(m+1)$-vih korijena iz jedinice tvore optimalnu normalnu bazu od $\mathbb{F}_{2^m}$ nad $\mathbb{F}_2$.

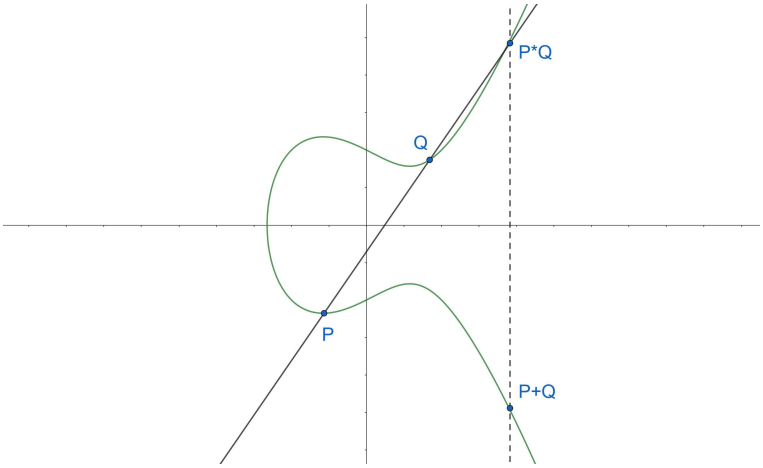
Osnovne operacije na eliptičkim krivuljama

Neka je E eliptička krivulja definirana nad poljem K. Jedno od važnijih svojstava eliptičkih krivulja jest da se na njima može uvesti operacija uz koju točke na eliptičkoj krivulji čine Abelovu grupu. Polinom trećeg stupnja može imati jedan ili tri realna korijena. U ovisnosti o tome, graf eliptičke krivulje ima jednu ili dvije komponente kao što vidimo na sljedećim slikama.



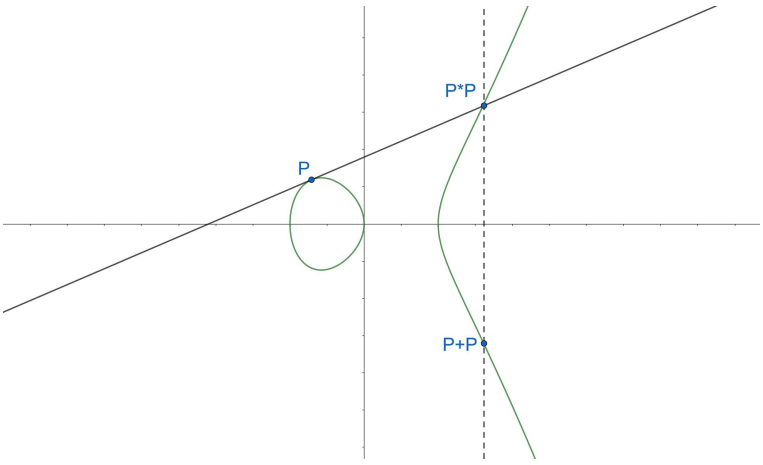
(a) jedna komponenta (b) dvije komponente

Definirajmo operaciju zbrajanja na $E(K)$. Pravilo zbrajanja je najbolje objasniti geometrijski. Neka su $P = (x_1, y_1)$ i $Q = (x_2, y_2)$ dvije različite točke na eliptičkoj krivulji E. Tada je zbroj P i Q, u oznaci R, definiran na sljedeći način. Prvo povucimo pravac kroz točke P i Q. Ovaj pravac siječe krivulju u trećoj točki koju označavamo s $P * Q$. Sada definiramo zbroj $P + Q$ kao osnosimetričnu točku točki $P * Q$ obzirom na os x kao što vidimo na 4.1.



Slika 4.1: Zbrajanje točaka

Ako je $P = Q$ točka na krivulji, onda $P + P = 2P$ nazivamo dupliciranjem točake. Označimo dupliciranu točku s R. Objasnimo pravilo dupliciranja geometrijski. Povucimo tangentu na eliptičku krivulju kroz točku P. Ovaj pravac siječe eliptičku krivulju u drugoj točki koju označavavamo s $P * P$. Sada definiramo zbroj $P + P$ kao osnosimetričnu točku točki $P * P$ obzirom na os x kao što vidimo na 4.2.



Slika 4.2: Dupliciranje točke

Definicija 4.2.7. *Neka je $p > 3$ prost broj. Neka je E eliptička krivulja oblika $y^2 = x^3 + ax + b$ takva da je $D = -16(4a^3 + 27b^3) \neq 0$. Eliptička krivulja E nad konačnim poljem $\mathbb{F}_p$,*

u oznaci $E(\mathbb{F}_p)$ je skup točaka takvih da

$$E(\mathbb{F}_p) = \{(x, y) : x, y \in \mathbb{F}_p, y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

gdje su $a, b \in \mathbb{F}_p$, a $\mathcal{O}$ označava "točku u beskonačnosti".

Teorem 4.2.8 (svojstva operacije zbrajanja)**.** *Neka je E eliptička krivulja definirana na konačnom polju $\mathbb{F}_p$ kao gore. Neka su $P = (x_1, y_1)$, $Q = (x_2, y_2)$ dvije točke na krivulji E. Tada vrijedi*

1. $-\mathcal{O} = \mathcal{O}$,
2. $-P = (x_1, -y_1)$,
3. $P + \mathcal{O} = \mathcal{O} + P = P$, *za sve* $P \in E(\mathbb{F}_p)$,
4. $P + (-P) = \mathcal{O}$, *za sve* $P \in E(\mathbb{F}_p)$,
5. *(zbrajanje točaka) Ako je* $P \neq \pm Q$, *onda je* $P + Q = (x_3, y_3)$, *gdje je*

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = -y_1 + \lambda(x_1 - x_3)$$

gdje je

$$\lambda = \frac{y_2 - y_1}{x_2 - x_1} \tag{4.1}$$

6. *(dupliciranje točaka) Ako je* $P = Q$, *onda je* $2P = P + Q = (x_3, y_3)$, *gdje je*

$$x_3 = \lambda^2 - x_1 - x_2, \quad y_3 = -y_1 + \lambda(x_1 - x_3)$$

$$\lambda = \frac{3x_1^2 + a}{2y_1} \tag{4.2}$$

Za proučavanje algoritama koje ćemo uvesti kasnije, potrebno nam je i množenje točke s cijelim brojem kP. Postoji više načina na koje možemo izračunati kP ovisno o tome je li P fiksna ili nije, te zbrajamo li $kP + lQ$ dvije točke. Ono što će nama biti dovoljno u ovom trenutku je znati da je množenje točke cijelim brojem zapravo uzastopno zbrajanje točke

$$kP = \underbrace{P + P + \ldots + P}_{\text{k-puta}}$$

Poglavlje 5

Problem diskretnog logaritma nad eliptičkim krivuljama

Problem diskretnog logaritma nad eliptičkim krivuljama uvodimo jer nam je bitan za sigurnost ECDSA o kojemu ćemo pričati u sljedećem poglavlju.

Prvo uvodimo pojam primitivnog korijena u polju $\mathbb{F}_p$.

Teorem 5.0.1 (Teorem o primitivnom korijenu)**.** *Neka je p prost broj. Tada postoji $g \in \mathbb{F}_p^*$ čije potencije daju sve elemente iz $\mathbb{F}_p^*$, tj.*

$$\mathbb{F}_p^* = \{1, g, g^2, \ldots, g^{p-2}\}.$$

Elemente s ovakvim svojstvom nazivamo primitivnim korijenima od $\mathbb{F}_p$ ili generatorima od $\mathbb{F}_p^*$. To su elementi skupa $\mathbb{F}_p^*$ reda $p-1$.

Prije nego definiramo problem diskretnog logaritma, objasnimo što znači izraz g^x.

Neka je $(G, \cdot)$ grupa. Neka je g element grupe G i neka je x pozitivan cijeli broj. Tada g^x znači da smo x puta primijenili grupovnu operaciju $\cdot$ na element g,

$$g^x = \underbrace{g \cdot g \cdot \ldots \cdot g}_{x \text{ puta}}.$$

Posebno, za $x = 0$, tada je $g^0 = e$, gdje je e neutralni element grupe G.

Definicija 5.0.2. *Neka je g primitivan korijen za $\mathbb{F}_p$ i neka je h nenul element iz $\mathbb{F}_p$. Problem diskretnog logaritma je problem pronalaska eksponenta x takvog da*

$$g^x \equiv h \pmod{p}$$

Broj x se naziva diskretan logaritam od h s bazom g.

Uvedimo problem diskretnog logaritma u kriptografiju kroz sljedeći primjer.

Alice objavljuje dva broja g i h, a njezin tajni ključ je eksponent x koji je rješenje

$$h \equiv g^x \pmod{p}.$$

Zanima nas može li Alice učiniti nešto slično s eliptičkom krivuljom E nad poljem $\mathbb{F}_p$. Ako sada Alice promatra g i h kao elemente grupe $\mathbb{F}_p^*$, problem diskretnog logaritma zahtjeva od Aliceine protivnice Eve da pronađe x takav da

$$h \equiv \underbrace{g \cdot g \cdot \ldots \cdot g}_{x \text{ puta}} \pmod{p}$$

Dakle, Eve mora odrediti koliko puta je g pomnožen sa samim sobom kako bi dobila h.

Odavde zaključujemo da Alice može ovaj postupak provesti i u slučaju grupe točaka $E(\mathbb{F}_p)$ eliptičke krivulje E nad konačnim poljem $\mathbb{F}_p$. Odabire i objavljuje dvije točke P i Q iz $E(\mathbb{F}_p)$, te je njezin tajni ključ cijeli broj n takav da

$$Q = nP$$

gdje je

$$nP = \underbrace{P + P + \ldots + P}_{\text{n puta}}.$$

Zbrajanje točaka eliptičke krivulje E je jako kompliciranа operacija, tako da Eve nije jednostavno pronaći tajni ključ n. Iz ovog razloga je problem diskretnog logaritma na eliptičkim krivuljama teško riješiti.

Definicija 5.0.3. *Neka je E eliptička krivulja nad konačnim poljem $\mathbb{F}_p$ i neka su P i Q točke na $E(\mathbb{F}_p)$. Problem diskretnog logaritma eliptičke krivulje (ECDLP) je problem pronalaska cijelog broja n takvog da je $Q = nP$. Ovaj n označavamo sa*

$$n = \log_P(Q)$$

i zovemo eliptički diskretni logaritam od Q u odnosu na P.

Prije komentiranja ove definicije, uvedimo dva rezultata koja će nam biti potrebna za bolje razumijevanje.

Propozicija 5.0.4. *Neka je G konačna grupa i neka je $a \in G$. Tada red od a dijeli red grupe G. Drugim riječima, neka je $n = |G|$ red grupe G i neka je d red od a, tj. a^d je najmanja pozitivna potencija od a takva da $a^d = e$. Tada*

$$a^n = e \quad i \quad d|n.$$

Teorem 5.0.5 (Lagrangeov teorem)**.** *Neka je G konačna grupa. Tada svaki element grupe G ima konačan red. Nadalje, ako $a \in G$ ima red d i ako je $a^k = e$, onda d dijeli k.*

Napomena 5.0.6. *Postoji par problema s definicijom* $\log_P(Q)$.

Prvi problem je da se može dogoditi da za neke dvije točke $P, Q \in E(\mathbb{F}_p)$, Q zapravo nije višekratnik od P. U tom slučaju $\log_P(Q)$ *nije definiran. No, u kriptografiji, Alice na početku ima javnu točku P i cijeli broj n, koji je privatan, te računa i objavljuje vrijednost $Q = nP$. To znači, da u praktičnoj primjeni,* $\log_P(Q)$ *postoji i njegova vrijednost je Alicein tajni ključ.*

Drugi problem da ako postoji jedna vrijednost n koja zadovoljava $Q = nP$, onda takvih vrijednosti ima mnogo. Da bismo ovo vidjeli, prvo primijetimo da postoji prirodan broj s takav da $sP = O$. Ovo slijedi iz Propozicije 5.0.4. S obzirom na to da je $E(\mathbb{F}_p)$ konačan, to znači da točke P, $2P$, $3P$, ... ne mogu biti sve različite. Dakle, postoje $k, l \in \mathbb{Z}$ takvi da $k > l$ za koje vrijedi $kP = lP$. Uzmimo sada da je $s = k - l$. Najmanji takav $s \geq 1$ naziva se red od P. Iz Teorema 5.0.5 slijedi da red od P dijeli red od $E(\mathbb{F}_p)$. Dakle, ako je s red od P i $n_0 \in \mathbb{Z}$ takav da je $Q = n_0P$, onda su rješenja izraza $Q = nP$ cijeli brojevi $n = n_0 + js$ gdje $j \in \mathbb{Z}$. To znači da je vrijednost $\log_P(Q)$ *element $\mathbb{Z}/s\mathbb{Z}$, odnosno* $\log_P(Q)$ *je cijeli broj modulo s, gdje je s red od P. Možemo staviti $n_0 = \log_P(Q)$. Međutim, prednost definiranja vrijednosti u $\mathbb{Z}/s\mathbb{Z}$ je ta da eliptički diskretni logaritam tada zadovoljava*

$$\log_P(Q_1 + Q_2) = \log_P(Q_1) + \log_P(Q_2)$$

za sve $Q_1, Q_2 \in E(\mathbb{F}_p)$. Možemo primijetiti analogiju s običnim logaritmom $\log(ab) = \log(a) + \log(b)$ *i diskretnog logaritma za $\mathbb{F}_p^*$. Iz gornjeg izraza zaključujemo da diskretni logaritam za $E(\mathbb{F}_p)$ poštuje pravilo zbrajanja kada se $E(\mathbb{F}_p)$ preslika u grupu $\mathbb{Z}/s\mathbb{Z}$. Dakle, preslikavanje* $\log_P$ *definira homomorfizam grupa*

$$\log_p : E(\mathbb{F}_p) \to \mathbb{Z}/s\mathbb{Z}.$$

5.1 "Dupliciraj i zbroji" algoritam

Problem diskretnog logaritma nad eliptičkim krivuljama je izrazito teško riješiti, tj. ako su nam poznate točke P i $Q = nP$ u $E(\mathbb{F}_p)$, teško je odrediti vrijednost n. Da bismo mogli koristiti funkciju

$$\mathbb{Z} \ni n \mapsto nP \in E(\mathbb{F}_p)$$

u kriptografiji, moramo na učinkoviti način izračunati nP iz poznatih vrijednosti n i P. Ako je n velik, ne želimo računati nP tako da računamo P, $2P$, ... Za računanje nP koristimo algortam "dupliciraj i zbroji". Algoritam se još naziva i "binarne ljestve" jer koristi binarni zapis broja n. Ideja je sljedeća: prvo zapišemo broj n u binarnoj formi

$$n = n_0 + n_1 \cdot 2 + n_2 \cdot 2^2 + \ldots + n_k \cdot 2^k$$

gdje su $n_0, n_1, \ldots, n_k \in \{0, 1\}$.

Pretpostavimo da je $n_r = 1$. Računamo sljedeće vrijednosti:

$$Q_0 = P, \quad Q_1 = 2Q_0, \quad Q_2 = 2Q_1, \quad \ldots \quad Q_k = 2Q_{k-1}.$$

Odavde zaključujemo da vrijedi $Q_i = 2Q_{i-1}$ za svaki i, pa vrijedi

$$Q^i = 2^i P.$$

Ukupno vrijeme koje nam je potrebno da bismo izračunali nP je najviše $2k$ operacija u $E(\mathbb{F}_p)$. Dakle, $n \geq 2r$ iz čega slijedi da za računanje nP nam je potrebno najviše $2\log_2(n)$ operacija u $E(\mathbb{F}_p)$. Iz toga zaključujemo da je računanje nP izvedivo i za velike n.

Algoritam 1 "Dupliciraj i zbrajaj" algoritam

Ulaz: $P \in E(\mathbb{F}_p)$ i cijeli broj $n \geq 1$.

```
Q = P i R = O
while n > 0 do
    if n ≡ 1 (mod 2) then
        R = R + Q
    end if
    Q = 2Q i n = ⌊n/2⌋
end while
return Vrati točku R koja je jednaka nP.
```

Zanima nas koliko smo uštedjeli ovim algoritmom. Neka je n velik broj i neka je $k = \lfloor \log_n \rfloor + 1$. U najgorem slučaju, n ima oblik $2^k - 1$, te tada računanje nP zahtjeva $2k$ operacija.

Zapisivanje broja n kao zbroja pozitivnih i negativnih potencija broja 2 naziva se trojni prikaz broja n. Ako dozvolimo trojni prikaz broja n, onda za računanje nP treba najviše $\frac{3}{2} + 1$ operacija što slijedi iz sljedeće propozicije.

Propozicija 5.1.1. *Neka je n pozitivan cijeli broj i neka je* $k = \lfloor \log_n \rfloor + 1$*, što znači da je* $2^k > n$*. Onda uvijek možemo pisati*

$$n = u_0 + u_1 \cdot 2 + u_2 \cdot 2^2 + \ldots + u_k \cdot 2^k,$$

gdje su $u_0, u_1, \ldots u_k \in \{-1, 0, 1\}$ *i najviše* $\frac{1}{2}k$ *takvih* u_i *je različito od* 0*.*

Ovo je najgori mogući scenarij, ali također je važno znati što se događa u prosjeku. Binarni prikaz slučajnog broja ima približno isti broj 1 i 0, tako da za većinu n izračunavanje nP u tom slučaju zahtjeva $\frac{3}{2}k$ koraka. Ako dopustimo zbrojeve i razlike potencija od 2, tada za većinu n možemo izračunati nP u otprilike $\frac{4}{3}k + 1$ koraka.

5.2 Zahtjevnost ECDLP-a

Problem diskretnog algoritma se može brže riješiti od ECDLP. Glavni razlog zašto se eliptičke krivulje koriste u kriptografiji je činjenica da ne postoje efikasni (polinomijalni; čak niti subeksponencijalni) algoritmi izračuna za ECDLP, i doista, nema poznatih općih algoritama koji rješavaju ECDLP u manje od $O\left(\sqrt{n}\right)$ koraka, gdje je n red grupe $E(\mathbb{F}_p)$.

Index calculus metoda

Spomenimo subeksponencijalni algoritam za problem diskretnog logaritma u $\mathbb{F}_p^*$.

Definicija 5.2.1. *Kažemo da je prirodan broj n B-gladak ako su mu svi prosti faktori manji ili jednaki B.*

Metoda za rješavanje problema diskretnog logaritma u konačnom polju $\mathbb{F}_p$ zove se *index calculus*. Ideja iza *index calculus* metode je jednostavna. Želimo riješiti problem diskretnog logaritma

$$g^x \equiv h \pmod{p},$$

gdje je p prost broj, a cijeli brojevi g i h su dani. Radi jednostavnosti, pretpostavimo da je g primitivni korijen modulo p, tako da njegove potencije daju sve članove $\mathbb{F}_p^*$.

Umjesto da računamo gornji izraz direktno, radije izaberemo vrijednost B i rješavamo problem diskretnog logaritma

$$g^x \equiv l \pmod{p} \quad \text{za sve proste } l \leq B.$$

Drugim riječima, računamo diskretni logaritam $\log_g(l)$ za svaki prosti broj $l \leq B$.

Sada računamo vrijednosti

$$h \cdot g^{-k} \pmod{p} \quad za\ k = 1, 2, \ldots$$

sve dok ne pronađemo k takav da je $h \cdot g^{-k} \pmod{p}$ B-gladak. Za taj k imamo

$$h \cdot g^{-k} \equiv \prod_{l \leq B} l^{e_l} \pmod{p}$$

za određene eksponente e_l. Zapišimo ovaj izraz u terminima diskretnog logaritma

$$\log_g(h) \equiv k + \sum_{l \leq B} e_l \cdot \log_g(l) \pmod{p-1},$$

gdje se prisjetimo da su diskretni logaritmi definirani samo modulo $p-1$. Mi pretpostavljamo da smo već izračunali vrijednosti $\log_g(l)$ za sve proste brojeve $l \leq B$, tako da nam gornji izraz daje vrijednost $\log_g(h)$.

Još ćemo objasniti kako pronaći $\log_g(l)$ za male proste brojeve l. Za nasumični odabir eksponenata i računamo

$$g_i \equiv g^i \pmod{p} \quad \text{sa } 0 < g_i < p.$$

Ako g_i nije B-gladak, odbacujemo ga, a ako je B-gladak, možemo ga faktorizirati kao

$$g_i = \prod_{l \leq B} l^{u_l(i)}.$$

U terminima diskretnog logaritma, ovo nam daje relaciju

$$i \equiv \log_g(g_i) \equiv \sum_{l \leq B} u_l(i) \cdot \log_g(l) \pmod{p-1}$$

Jedine nepoznanice u ovom izrazu su vrijednosti diskretnog logaritma $\log_g(l)$, za proste brojeve $l \leq B$.

Ako uspijemo dobiti jednadžbi koliko imamo nepoznanica, onda možemo očekivati da će sustav imati jedinstveno rješenje. Tada odgovarajućim metodama za rješavanje sustava možemo efikasno pronaći rješenje. Što je B veći, veća je i šansa da će g_i biti B-gladak, ali veći B znači da će rješavanje sustava biti zahtjevnije. Pokazuje se da je optimalan izbor za B subeksponencijalna funkcija od p, pa je zato i složenost ovog algoritma subeksponencijalna.

Index calculus metoda je subeksponencijalni algoritam za rješavanje problema diskretnog logaritma u $\mathbb{F}_p^*$. Ovo je u značajnoj suprotnosti s problemom diskretnog logaritma u grupama eliptičkih krivulja.

Trenutačno su najpoznatiji algoritmi za rješavanje problema općeg diskretnog logaritma u skupinama eliptičkih krivulja potpuno eksponencijalni.

Shanksova "baby step-giant step" (BSGS) metoda

Spomenimo algoritam za ECDLP čija je složenost $O\left(\sqrt{n}\right)$.

Teorem 5.2.2 (Teorem o dijeljenju s ostatkom)**.** *Za proizvoljan prirodan broj a i cijeli broj b postoje jedinstveni cijeli brojevi $q \in \mathbb{Z}$ i $r \in \mathbb{N}_0$ takvi da je $b = qa + r$, $0 \leq r < a$.*

Neka su P i Q elementi grupe G, te neka je $Q = mP$. Koristeći teorem o dijeljenju s ostatkom, m možemo zapisati u sljedećem obliku

$$m = \lceil \sqrt{n} \rceil a + b, \quad \text{gdje je } 0 \leq a, b < \sqrt{n}.$$

Trebamo odrediti vrijednosti a i b. Ako uvrstimo gornji izraz u jednadžbu $Q = mP$, dobit ćemo sljedeće

$$Q - bP = a(\lceil \sqrt{n} \rceil P)$$

Najprije računamo tablicu "baby stepova" koja se sastoji od vrijednosti

$$R_b = Q - bP, \qquad \text{za } b = 0, 1, \ldots, \lceil \sqrt{n} \rceil - 1.$$

Zatim sortiramo tu tablicu i spremimo u memoriju tako da ju se može efikasno pretraživati. Sljedeći korak je računanje "giant stepova"

$$S_a = a(\lceil \sqrt{n} \rceil P), \qquad \text{za } a = 0, 1, \ldots, \lceil \sqrt{n} \rceil - 1.$$

Nakon svakog računanja "giant stepa", provjeravamo pojavljuje li se S_a u tablici. Ako se pojavljuje, onda smo pronašli vrijednosti a i b. Ovaj postupak mora završiti prije nego a dosegne vrijednost $\lceil \sqrt{n} \rceil$.

Poglavlje 6

ECDSA algoritam

6.1 Protokoli

Postoje različite vrste kriptografskih protokola koje koriste eliptičke krivulje. Oni se razlikuju po svojim kriptografskim svojstvima. Mi se usredotočujemo na tri područja: potpise, šifriranje i dogovor oko ključeva.

Standardizacija kriptografskih protokola, a posebno protokola eliptičkih krivulja, u posljednjih je nekoliko godina znatno napredovala. Standardizacija je važna za implementaciju sustava u velikim razmjerima. Rad u skladu s dobro definiranim standardom za bilo koju tehnologiju pomaže interoperabilnosti.

U kontekstu kriptografije eliptičke krivulje, standardi su definirani tako da se zna ne samo točan rad svakog algoritma, već i format prenesenih podataka.

Standard odgovara na pitanja kao što su

- U kojem se formatu prenose elementi konačnog polja i točke eliptičke krivulje?
- Kako se javni ključevi formatiraju prije potpisivanja u certifikat?
- Kako će se izvršiti pretvorbe između proizvoljnih nizova bitova u elemente konačnih polja, ili iz elemenata konačnih polja u cijele brojeve, i obrnuto?
- Kako se opcije kao sto je upotreba kompresije točke ili izbor krivulje signaliziraju korisniku?

Došlo je do brojnih pokušaja standardizacije, a u mnogima su ograničili izbor parametara zahtijevajući određene krivulje ili određena konačna polja. Ovo pomaže u interoperabilnosti, ali i znači da postoje dobro definirani skupovi izbora parametara koji pružaju određenu razinu sigurnosti.

Od posebne važnosti za kriptografiju eliptičke krivulje su sljedeći standardi:

- IEEE 1363
- ANSI X9.62 i X9.63
- FIPS 186.2
- SECG
- ISO

Više o ovim standardima možete pronaći u [1].

6.2 ECDSA

ECDSA (eng. *Elliptic Curve Digital Signature Algorithm*) je varijanta algoritma digitalnog potpisa (eng. *Digital Signature Algorithm* ili DSA) koji koristi eliptičke krivulje. To je najraširenija standardizirana shema potpisa temeljena na eliptičkim krivuljama.

Prvo opišimo DSA kako bismo vidjeli da je ECDSA njegova generalizacija. U DSA prvo biramo hash funkciju H čija je izlazna vrijednost niz bitova duljine m. Zatim se definira prosti broj q duljine m ili više bitova i prosti broj p od n bitova tako da vrijedi

- q dijeli $p-1$, tj. $p = qk + 1$ za neki k cijeli broj.
- Problem diskretnog logaritma u podgrupi od $\mathbb{F}_p$ reda q je nerješiv.

S trenutačnim tehnikama i računalnim tehnologijama, n bi trebao biti veći ili jednak 1024 bita, dok bi m trebao biti veći od 160 bitova.

Sada trebamo pronaći generator g za podgrupu reda q u $\mathbb{F}_p^*$. To radimo tako što generiramo slučajni element $h \in \mathbb{F}_p^*$ i računamo

$$g = h^{(p-1)/q} \pmod{p}$$

sve dok ne dobijemo vrijednost g koja nije jednaka 1. Postoji $\frac{1}{q}$ šansa da nećemo dobiti $g \neq 1$ s prvim h kojeg odaberemo. Dakle, generator g je jednostavno pronaći.

Četvorku (H, p, q, g) nazivamo skup parametara domene sustava. Dakle, parametri domene definiraju hash funkciju, grupu reda q i generator te grupe.

DSA koristi funkciju $f : \mathbb{F}_p^* \to \mathbb{F}_q$ koja preslikava $x \mapsto x \pmod{q}$. Drugim riječima, $x \in \mathbb{F}_p^*$ interpretiramo kao cijeli broj kada se provodi redukcija modulo q. Funkcija f se koristi za preslikavanje elemenata grupe u cijele brojeve modulo q i često se naziva *funkcija pretvorbe*.

Kao par javni-tajni ključ u DSA sustavu koristi se (y, x) gdje

$$y = g^x \pmod{p}$$

Potpisivanje poruke u DSA se odvija na sljedeći način:

Algoritam 2 Potpisivanje poruke

Ulaz: Poruka m i tajni ključ x.
Izlaz: Vrati potpis (r, s) poruke m.

1: Izaberi $k \in_R \{1, \ldots, q-1\}$.
2: Izračunaj $t = g^k \pmod{p}$.
3: Izračunaj $r = f(t)$.
4: Ako je $r = 0$ idi na korak 1.
5: Izračunaj $e = H(m)$.
6: Izračunaj $s = k^{-1}(e + xr) \pmod{q}$.
7: Ako je $s = 0$ idi na korak 1.
8: Vrati (r, s).

Izraz $k \in_R \{1, \ldots, q-1\}$ nam označava da k biramo *nasumično* iz skupa $\{1, \ldots, q-1\}$.
Algoritam za provjeru vjerodostojnosti potpisa je tada zadan sa

Algoritam 3 Provjera vjerodostojnosti potpisa

Ulaz: Poruka m, javni ključ y i potpis (r, s).
Izlaz: Prihvati ili odbij potpis.

1: Odbij potpis ako $r, s \notin \{1, \ldots, q-1\}$.
2: Izračunaj $e = H(m)$.
3: Izračunaj $u_1 = \frac{e}{s} \pmod{q}$ i $u_2 = \frac{r}{s} \pmod{q}$.
4: Izračunaj $t = g^{u_1} y^{u_2} \pmod{p}$.
5: Prihvati potpis ako i samo ako je $r = f(t)$.

Sada kada smo objasnili DSA algoritam, možemo uvesti ECDSA algoritam.

Za ECDSA algoritam, parametri domene su dani s (H, K, E, q, G), gdje je H hash funkcija, E je eliptička krivulja nad konačnim poljem K, a G je točka na krivulji prostog reda q. Točke eliptičke krivulje označavamo velikim tiskanim slovima.

Javni-tajni ključ je par dan sa (Y, x) gdje

$$Y = [x]G,$$

a funkcija $f : E \to \mathbb{F}_q$ preslikava $P \mapsto x(P) \pmod{q}$. $x(P)$ označava x-koordinatu točke P i tumačimo je kao cijeli broj kada izvodimo redukciju modulo q. Ova interpretacija se provodi čak i kada je krivulja definirana preko polja karakteristike dva.

U slučaju polja karakteristike 2, želimo pretvoriti element polja $g(x)$, koji je binarni polinom, u cijeli broj. Gotovo svi standardi prihvaćaju konvenciju da se $g(2)$ procjenjuje

preko cijelih brojeva. Dakle, ako imamo polinom

$$x^5 + x^2 + 1$$

njega interpretiramo na sljedeći način

$$2^5 + 2^2 + 1 = 32 + 4 + 1 = 37$$

Potpisivanje poruke u ECDSA se odvija na sljedeći način:

Algoritam 4 Potpisivanje poruke

Ulaz: Poruka m i tajni ključ (r, s).
Izlaz: Vrati potpis (r, s) poruke m.

1: Izaberi $k \in_R \{1, \ldots, q-1\}$.
2: Izračunaj $T = [k]G$.
3: Izračunaj $r = f(T)$.
4: Ako je $r = 0$ idi na korak 1.
5: Izračunaj $e = H(m)$.
6: Izračunaj $s = k^{-1}(e + xr) \pmod{q}$.
7: Ako je $s = 0$ idi na korak 1.
8: Vrati (r, s).

Algoritam za provjeru vjerodostojnosti potpisa je tada dan sa:

Algoritam 5 Provjera vjerodostojnosti potpisa

Ulaz: Poruka m, javni ključ Y i potpis (r, s).
Izlaz: Prihvati ili odbij potpis.

1: Odbij potpis ako $r, s \notin \{1, \ldots, q-1\}$.
2: Izračunaj $e = H(m)$.
3: Izračunaj $u_1 = \frac{e}{s} \pmod{q}$ i $u_2 = \frac{r}{s} \pmod{q}$.
4: Izračunaj $T = [u_1]G + [u_2]Y$.
5: Prihvati potpis ako i samo ako je $r = f(T)$.

Dokažimo da provjera vjerodostojnosti potpisa funkcionira.

Dokaz. Ako je potpis (r, s) na poruci m doista generiran pomoću legitimnog potpisnika, tada je $s \cong k^{-1}(e + dr) \pmod{n}$. Sređivanjem dobijemo

$$\begin{aligned} k &\cong s^{-1}(e + dr) \\ &\cong s^{-1}e + s^{-1}rd \\ &\cong we + wrd \\ &\cong u_1 + u_2 d \pmod{n} \end{aligned}$$

Stoga je $X = u_1P + u_2Q = (u_1 + u_2d)P = kP$, pa je $v = r$ kako se traži. □

Može se pokazati da je ECDSA dokazivo siguran, uz dodatne pretpostavke koje ćemo obraditi u idućem poglavlju.

Efemerni tajni ključ k mora biti slučajan. Neka su m i m' dvije različite poruke s istom vrijednošću k. Tada su njihovi potpisi (r, s) i (r', s'), gdje su

$$r = r' = f([k]G);$$

$$s = \frac{e + xr}{k} \pmod{q}, \text{ gdje } e = H(m);$$

$$s' = \frac{e' + xr}{k} \pmod{q}, \text{ gdje } e' = H(m');$$

Tada imamo

$$\frac{e + xr}{s} = k = \frac{e' + xr}{s'} \pmod{q}$$

U tom slučaju možemo zaključiti

$$xr(s' - s) = se' - s'e,$$

odakle slijedi

$$x = \frac{se' - s'e}{r(s' - s)} \pmod{q}.$$

Od sada pretpostavljamo da je svaka vrijednost k odabrana nasumično.

6.3 Sigurnost ECDSA

ECDSA se koristi u mnogim standardima jer se smatra da je njegova reputacijska sigurnost dovoljna, na temelju toga da

a) je vrlo prirodan analogon DSA za eliptičke krivulje,

b) se smatra da i kriptografija eliptičke krivulje i DSA imaju dovoljno visoku reputacijsku sigurnost.

Pod određenim pretpostavkama, pronađeni su dokazi sigurnosti za sheme digitalnog potpisa slične DSA i ECDSA. Tehnike dokazivanja u tim početnim dokazima nisu primjenjive na DSA i ECDSA. No, s vremenom su pronađene nove dokazne tehnike i pretpostavke koje su prevladale ili izbjegle poteškoće u primjeni početnih tehnika na ECDSA. U ovom poglavlju ćemo spomenuti neke od ovih rezultata i skice njihovih dokaza.

U nekim slučajevima se nove tehnike dokazivanja nisu odnosile na DSA, što bi mogao biti pokazatelj da bi ECDSA mogao imati bolju sigurnost od DSA. Nadalje, neke od novih tehnika dokazivanja ne funkcioniraju za modificirane verzije ECDSA, dok su početne tehnike dokazivanja primjenjive. Stoga kažemo da modificirane verzije imaju dokazivu sigurnost neusporedivu s ECDSA.

Rezultati kriptoanalize komplementarni su dokazivim sigurnosnim rezultatima i jednako su važni. U ovom poglavlju ćemo uključiti uvjetne rezultate jer nije poznata uspješna praktična kriptoanaliza ECDSA. Hipoteza o dokazivom sigurnosnom rezultatu dovoljan je uvjet sigurnosti, dok rezultat kriptoanalize uspostavlja nužan uvjet sigurnosti.

Primarna svrha dokazivih sigurnosnih rezultata je ispitivanje sigurnosti ECDSA. Svrha nije ispitivati sigurnost primitiva koje koristi ECDSA. Čak i ako pretpostavimo da su primitivi sigurne, ne znači da će digitalni potpis, izrađen od tih primitiva, biti siguran.

6.4 Definicije i uvjeti

Navedimo formalne definicije za sheme potpisa i njihove sigurnosti.

Definicija 6.4.1 (Shema potpisa)**.** *Shema potpisa je uređena trojka vjerojatnosnih algoritama* $\Sigma = (K, G, V)$ *takva da*

- *K nema ulaznu vrijednost (osim nasumične vrijednosti) i izlazne vrijednosti su mu javni ključ Y i privatni ključ x,*
- *G kao ulazne vrijednosti ima privatni ključ x i proizvoljnu poruku m, a izlazna vrijednost mu je potpis S,*
- *V kao ulazne vrijednosti ima javni ključ Y, poruku m i potpis S, a izlazna vrijednost mu je valjano ili nevaljano.*

Kažemo da je shema potpisa *ispravna* ako za bilo koju poruku m i bilo koju nasumičnu vrijednost, izračunavanjem $K :\mapsto (x, Y)$, a zatim $G : (x, m) \mapsto S$ će osigurati da je $V : (Y, m, S) \mapsto$ valjano. Ako G ne koristi nasumičnost ulazne vrijednosti, onda kažemo da je Σ *deterministički*. Ako za svaku poruku m i javni ključ Y, najviše jedan potpis S zadovoljava $V(Y, m, S) =$ valjano, onda kažemo da je Σ *provjerljivo deterministička*.

Definicija 6.4.2. *Krivotvoritelj sheme potpisa* (K, G, V) *je vjerojatnosni algoritam F, koji kao ulaznu vrijednost ima ili javni ključ Y ili potpis S i unutarnje stanje X, i ima izlazne vrijednosti: poruku m, stanje X i R, što je ili potpis ili zahtjev za potpisom poruke* m_i.

Krivotvoritelj F mjeri se svojom sposobnošću da pobijedi u sljedećoj igri.

Definicija 6.4.3. *Igra krivotvorenja za krivotvoritelja F sheme potpisa $\Sigma = (K, G, V)$ ima više rundi, od kojih se svaka sastoji od dvije igre, prvu od strane potpisnika, a drugu od strane krivotvoritelja.*

- *U nultom koraku, potpisnik koristi K kako bi ispisao javni ključ Y i privatni ključ x.*
- *Zatim, krivotvoritelj dobiva kao ulaznu vrijednost javni ključ Y i fiksno početno stanje X_0, a on šalje kao izlazne vrijednosti poruku m_i, stanje X_1 i zahtjev ili potpis R_1.*
- *Za $i \geq 1$, korak i funkcionira na sljedeći način:*
  - *Ako je R_i zahtjev za potpis, tada potpisnik koristi G s ulaznom vrijednosti x i porukom m_i za ispis potpisa S_i. Zatim se ponovno poziva krivotvoritelj s potpisom S_i kao ulaznom vrijednosti i stanjem X_i. Zatim će vratiti novu poruku m_{i+1}, novo stanje X_{i+1} i novi zahtjev ili potpis R_{i+1}.*
  - *Ako je R_i potpis, a ne zahtjev, onda je igra gotova.*

Kada igra završi, u rundi i, krivotvoritelj je pobijedio ako su oba $m_{i+1} \neq m_1, \ldots, m_i$ i $V(Y, m_{i+1}, R_{i+1}) =$ valjano; inače je krivotvoritelj izgubio.

Uvedimo definicije (p, Q, t)-krivotvoritelja i selektivnog krivotvoritelja.

Definicija 6.4.4 (Krivotvoritelj)**.** *Krivotvoritelj F je (p, Q, t)- krivotvoritelj sheme potpisa (K, G, V) ako je njegova vjerojatnost pobjede u igri krivotvorenja u najviše Q koraka, koristeći računski napor, najviše t, a najmanje p. Potpis Σ je (p, Q, t)-siguran ako nema (p, Q, t)-krivotvoritelja.*

$(p, 0, t)$-krivotvoritelj naziva se *pasivnim krivotvoriteljem*, a krivotvoritelj koji nije pasivan naziva se *aktivnim*. Tako definirani krivotvoritelji su uspješni bez obzira na kvalitetu ili smisao krivotvorene poruke.

Kako bi se naglasio ovaj ograničavajući aspekt, takvi se krivotvoritelji nazivaju *egzistencijalnim krivotvoriteljima*. Svaki krivotvoritelj koji nije egzistencijalan je *selektivan*.

Definicija 6.4.5 (Selektivni krivotvoritelj)**.** *Neka je U vjerojatnosni algoritam, bez ulaznih vrijednosti osim nasumičnosti i poruke kao izlaza. Selektivni krivotvoritelj je krivotvoritelj F sa sljedećim razlikama. Ulazna vrijednost javnog ključa uključuje poruku. Igra selektivnog krivotvorenja za selektivnog krivotvoritelja F sheme potpisa (K, G, V), s potprogramom U koji odabire poruke, je igra krivotvorenja sa sljedećim razlikama. U koraku 0, U se poziva kako bi generirao poruku m_0, koja se daje kao ulazna vrijednost krivotvoritelju F. Krivotvoritelj pobjeđuje u igri u rundi i, samo ako je zadovoljeno $m_0 = m_{i+1}$. Selektivni krivotvoritelj F je (p, Q, t, U)-krivotvoritelj sheme potpisa (K, G, V).*

Selektivni krivotvoritelj može krivotvoriti bilo koju poruku iz određene klase poruka. Selektivni krivotvoritelj općenito je mnogo štetniji od egzistencijalnog krivotvoritelja, ovisno o distribuciji poruka koje daje U.

Općenito, s istim p i t, pasivni krivotvoritelj je štetniji od aktivnog, a selektivni je štetniji od egzistencijalnog. Općenito, pasivnost i selektivnost su kvalitativne osobine krivotvoritelja, a njihova važnost ovisi o upotrebi potpisa.

Definicija 6.4.6. *Shema potpisa je (p, Q, t)-sigurna protiv egzistencijalnog krivotvoritelja ako ne postoji (p, Q, t)-krivotvoritelj.*

Shema potpisa je (p, Q, t, U)-sigurna protiv selektivnog krivotvoritelja ako ne postoji (p, Q, t, U)-selektivni krivotvoritelj.

Nužni uvjeti

Navesti ćemo i objasniti nužne uvjete za sigurnost ECDSA algoritma.

Nerješivi polulogaritam

Za funkciju pretvorbe f i grupu $\langle G\rangle$, *polulogaritam* elementa $P \in \langle G\rangle$ po bazi G je par cijelih brojeva (t, u) takav da

$$t = f([u^{-1}](G + [t]P)).$$

Problem pronalaženje polulogaritma mora biti nerješiv. U suprotnom se krivotvorine mogu pronaći postavljanjem $P = [H(m) - 1]Y$, gdje je Y javni ključ, a $(t, H(m)u)$ je krivotvorina poruke m. Ovakvo krivotvorenje je i pasivno i selektivno, što je najozbiljniji oblik krivotvorine. Stoga je nerješivost polulogaritamskog problema nužan uvjet za sigurnost ECDSA.

Nerješivi diskretni logaritam

Za grupu $\langle G\rangle$, (diskretni) logaritam elementa $P \in \langle G\rangle$ po bazi G je cijeli broj x takav da je $P = [x]G$.

Pronalaženje diskretnog logaritma od P omogućuje pronalaženje njegovog polulogaritma, pomoću

$$(t, u) = (f([k]G), k^{-1}(1 + td)),$$

prema tome dopušta krivotvorenje ECDSA potpisa. Krivotvoritelj je i pasivan i selektivan.

Gotovo bijektivna funkcija pretvorbe

Funkcija pretvorbe f je α-grozdasta ako neki element t^* iz slike funkcije ima prasliku veličine barem α puta veličina domene. Ako f nije α-grozdasta, tada se kaže da je gotovo bijektivna snage $\frac{1}{\alpha}$. α-grozdasta funkcija pretvorbe znači da su slučajni (t^*, u) polulogaritmi

s vjerojatnošću najmanje α. Stoga je u prosjeku potrebno oko $\frac{1}{\alpha}$ pokušaja da se dobije polulogaritam.

"Nepogodivo" i aritmetički nepristrano generiranje privatnog ključa

Ako se generator za statički privatni ključ x može pogoditi, u smislu da protivnik prilično lako može pogoditi njegove vrijednosti, tada je moguće pasivno selektivno krivotvorenje. Ako je efemerni privatni ključ k moguće pogoditi, tada je moguće aktivno selektivno krivotvorenje, jer je privatni ključ x određen iz k i potpisa (r, s) formulom $x = r^{-1}(ks - H(m)) \pmod q$.

Nadalje, objavljen je niz napada koji pokazuju da se privatni ključ x može dobiti ako generator slučajnih brojeva koji se koristi za k pokazuje određene vrste pristranosti. Ako se k ikada ponovi za različite poruke m i m', tada se privatni ključ može riješiti iz dva odgovarajuća potpisa (r, s) i (r', s') pomoću $x = \frac{se' - s'e}{s'r - sr'} \pmod q$.

Ispravno implementirana provjera raspona

Ako implementacija ne provjerava da je $r \neq 0$, tada je moguće krivotvorenje. Krivotvoritelj treba odabrati parametre iz domene eliptičke krivulje na takav način da je $G = [t]Z$, gdje je Z element grupe koji zadovoljava $f(Z) = 0$ i $t \in Z$. Za ECDSA funkciju pretvorbe, takve točke Z, ako postoje, mogu se pronaći na sljedeći način. Neka je x binarna reprezentacija od qu za neki cijeli broj u i pokušajte riješiti za odgovarajući y tako da (x, y) leži na krivulji (ovdje x ne treba brkati s privatnim ključem). Ponavljajte dok se ne pronađe točka ili dok se ne iscrpe sve legalne vrijednosti x. Krivotvoreni potpis je $(0, t^{-1}H(m))$.

Ova krivotvorina je pasivna i selektivna. Dva ograničenja umanjuju ozbiljnost ovakve krivotvorine. Prvo, potrebna je pogreška implementacije, a moguće je da pouzdana treća strana koristiti ispravnu implementaciju za rješavanje valjanosti potpisa. Drugo ograničenje je da je krivotvorina napad na parametre domene, ali obično pouzdani autoritet treće strane generira i distribuira parametre domene.

Rijetki nula hash

Ako *efektivna* hash funkcija, koja je sirovi hash skraćeni i reducirani po modulu q, poprima vrijednost 0 s vjerojatnošću p, tada je moguće pasivno selektivno krivotvorenje, kako je dano u nastavku. Krivotvoritelj odabire potpis $(r, s) = (f([t]Y), t^{-1}r)$, za neki $t \in Z$. Ako je odabrana poruka nula hash vrijednosti, što se događa s vjerojatnošću p, tada je krivotvoreni potpis valjan jer

$$f([s^{-1}]([H(m)]G + [r]Y)) = f([tr^{-1}]([0]G + [r]Y)) = f([t]Y) = r$$

Ovaj i drugi uvjeti za hash funkciju odnose se na efektivnu hash funkciju. Ova klasifikacija je važna za sigurnosnu analizu jer redukcija modulo q može uzrokovati neuspjeh uvjeta ako je q odabrao protivnik. Ako je protivnik odabrao parametre domene eliptičke krivulje, tada je moguće da je q odabran kao izlazna vrijednost, ili razlika između dvije izlazne vrijednosti, nereducirane hash funkcije, što bi omogućilo protivniku da pronađe nulu ili koliziju u efektivnoj (reduciranoj) hash funkciji.

Primijetite da klasteriranje na vrijednostima različitim od nule ne dovodi nužno do pasivne egzistencijalne krivotvorine. To može dovesti do drugih vrsta napada jer klasteriranje na određenim vrijednostima može dovesti do oslabljene otpornosti hash funkcije na jednoznačnost.

Hash otporan na nulu

Pronalazač nule hash funkcije je vjerojatnosni algoritam koji pronalazi poruku m tako da je $H(m) = 0$. Hash funkcija je otporna na nulu ako ne postoji pronalazač nule.

Pasivni egzistencijalni krivotvoritelj može se konstruirati na sljedeći način. Krivotvoritelj odabire potpis $(r, s) = (f([t]Y), t^{-1}r)$, koristeći pronalazač nule, pronalazi poruku m takvu da je $H(m) = 0$. Tada je (r, s) valjani potpis na m.

Hash funkcija koja je otporna na nulu je očito rijetko nula. Obrat ne vrijedi.

Hash otporan na jednosmjernost

Pretvarač hash funkcije je vjerojatnosni algoritam koji, ako mu je dana slučajna hash vrijednost e, pronalazi poruku m tako da je $H(m) = e$. Pretvarač se može koristiti za izgradnju pasivnog egzistencijalnog krivotvoritelja koristeći tehniku sličnu onoj za pronalazača nule. Krivotvoreni potpis je $(r, s) = (f([g]G + [y]Y), ry^{-1})$, a krivotvorena poruka m je jednosmjernost od $e = gs$. Krivotvoritelj je egzistencijalan, a ne selektivan, jer krivotvoritelj nema kontrolu nad onim što pronalazi pretvarač.

Ako hash funkcija nema pretvarač, ona je jednosmjerna. Upravo smo pokazali da, kako bi shema potpisa bila sigurna, efektivni hash mora biti jednosmjeran.

Jednosmjernost i otpornost na nulu su neovisna svojstva hasheva. Osobito su oba svojstva nužna za otpor pasivnom egzistencijalnom krivotvorenju.

Otpornost na jednoznačnost

Pronalazač jednoznačnosti hash funkcije je vjerojatnosni algoritam koji, ako mu je dana nasumična poruka m iz distribucije U, pronalazi drugu poruku m' tako da je $H(m) = H(m')$. Pronalazač jednoznačnosti može se koristiti za izgradnju aktivnog selektivnog krivotvoritelja. Krivotvoritelj dobiva potpis (r, s) od m' od potpisnog potprograma i vraća (r, s) kao

krivotvorinu od m. Krivotvoritelj je selektivan jer može krivotvoriti poruku m, a aktivan je jer zahtijeva pristup potprogramu za potpisivanje.

Pronalazač jednoznačnosti može se konstruirati iz hash-pretvarača pod uvjetom da distribucija U ima distribuciju slike $H(U)$ koja pokriva većinu slike od H s dovoljno velikom vjerojatnošću. Pod ovim uvjetima, pronalazač jednoznačnosti implicira postojanje pasivnog selektivnog krivotvoritelja.

(Jednosmjeran) hash otporan na kolizije

Pronalazač kolizije hash funkcije je vjerojatnosni algoritam koji pronalazi dvije poruke m i m' tako da je $H(m') = H(m)$. Pronalazač kolizije rezultira aktivnim egzistencijalnim krivotvoriteljem na sljedeći način. Krivotvoritelj dobiva potpis (r, s) od m' od potpisnog potprograma i zatim ispisuje (r, s) kao krivotvorinu od m, gdje su m i m' dobiveni od pronalazača kolizije. Ovaj krivotvoritelj je egzistencijalan jer nema kontrolu nad porukama m i m' koje je pronašao pronalazač kolizije, a aktivan je jer postavlja upit potprogramu za potpisivanje.

Treba imati na umu da se pronalazač kolizije može konstruirati iz pronalazača jednoznačnosti, pod uvjetom da je distribuciju U lako uzorkovati. Također, općenito, pronalazač kolizije može se konstruirati iz pronalazača nule, osim ako pronalazač nule uvijek ne daje kao izlaznu vrijednost jednu određenu nulu hash funkcije.

Dodatni uvjeti i modeli za dovoljnost

Uvjeti i modeli koje ćemo spomenuti u ovom odjeljku koriste se u hipotezama različitih dokazivih sigurnosnih rezultata za ECDSA. Nije poznato da su ovi uvjeti i modeli potrebni za sigurnost ECDSA. Za razliku od gore navedenih nužnih uvjeta, moguće je tvrditi da su ovi dodatni uvjeti i modeli toliko jaki da sigurnosni rezultati imaju malu vrijednost. Doista, idealno bi bilo ne koristiti nikakve uvjete ili modele osim poznatih potrebnih uvjeta.

Gotovo invertibilna funkcija pretvorbe

Funkcija pretvorbe f je gotovo invertibilna ako ima gotovo inverzni g, što je učinkovit vjerojatnosni algoritam koji se ponaša kao inverzna funkcija za f u sljedećem smislu. Algoritam g uzima nasumičnu ulaznu vrijednost $z \in \mathbb{Z}/q\mathbb{Z}$ i generira izlaz $P \in \langle G \rangle \bigcup \{\text{Nevaljano}\}$ tako da:

1. $P \neq$ Nevaljan s vjerojatnošću $\frac{1}{10}$ procijenjenom u odnosu na slučajne izbore z i g.
2. Ako je $P \neq$ Nevaljan, tada je $f(P) = z$.

3. Ako se neovisno slučajne ulazne vrijednosti $z \in \mathbb{Z}/q\mathbb{Z}$ opetovano daju g sve dok izlaz $P \neq$ Nevaljan, rezultirajuća distribucija takvog P ne razlikuje se od distribucije slučajnog $P \in \langle G \rangle$.

Jasno je da gotovo invertibilnost implicira gotovo bijektivnost. Konkretno, to je strogo jačanje nužnog uvjeta. Kada se koristi kao dio hipoteze o dokazivom sigurnosnom rezultatu, to implicira ovaj određeni nužni uvjet, gotovo bijektivnost.

Otpor prilagodljivog polulogaritama

Za funkciju pretvorbe f, grupu $\langle G \rangle$, i $e \in \mathbb{Z}/q\mathbb{Z}$ s $e \neq 1$, e-pomaknuti polulogaritam elementa $P \in \langle G \rangle$ po bazi G je par cijelih brojeva (t, u) tako da je $t = f([u^{-1}]([e]G + [t]P))$. Problem prilagodljivog polulogaritma je pronaći polulogaritam točke P po bazi G uz pristup potprogramu za e-pomaknute polulogaritme. Prilagodljivi polulogaritamski otpor znači da je problem prilagodljivog polulogaritma nerješiv. Ovaj uvjet je zapravo jačanje uvjeta polulogaritamskog otpora, poznatog nužnog uvjeta.

Pseudo-slučajni generator slučajnih brojeva

Sigurnosni dokazi općenito pretpostavljaju da su statički ključ x i efemerni privatni ključevi k, ako postoje, generirani uniformno i neovisno iz prostora privatnih ključeva $\mathbb{Z}/g\mathbb{Z}^*$.

U praksi to znači da se privatni ključevi mogu generirati pomoću generatora pseudoslučajnih brojeva, čiji se izlaz ne razlikuje od privatnih ključeva generiranih nasumično.

Kada se razmatra pasivno krivotvorenje, generator za efemerne ključeve se nikada ne poziva, tako da nema uvjeta kako se k generira. Generiranje x-a je još uvijek bitno.

Ako su potpisi generirani u deterministčkom načinu, što znači da je k tajna deterministička funkcija poruke koja se potpisuje, tada se pseudoslučajnost primjenjuje samo po poruci, a ne po potpisu. Neki od dokaza su primjenjivi samo u ovom slučaju.

Uniformna (glatka) hash funkcija

Neka je $H : \{0,1\}^* \to \mathbb{Z}/q\mathbb{Z}$ učinkovita hash funkcija. Neka je $U \subset \{0,1\}^*$ takav da

1. Za $m \in_R U$, $e = H(m)$ se može učinkovito generirati.

2. Za svaki $e \in \mathbb{Z}/q\mathbb{Z}$, skup $P_e = h^{-1}(e) \cap U$ je dovoljno velik tako da, čak i ako je P_e poznat, vjerojatnost $\frac{1}{|P_e|}$ je dovoljno mala da pogađanje nasumično odabranog tajnog elementa od P_e učini neizvedivim.

Neka je $b \in_R \{1,2\}$. Vjerojatnosni algoritam D_h je (ϵ_D, τ_d, U)-razlučivač za h ako D_h prihvaća kao ulaz $e \in_R \mathbb{Z}/q\mathbb{Z}$ ako $b = 1$ ili $e = H(m)$ za $m \in_R U$ ako $b = 2$, i ako D_h vraća, u vremenu izvođenja najviše τ_D, pogodak $d \in \{1,2\}$. Nadalje, $d = b$ s vjerojatnošću

najmanje $\frac{1}{2} + \epsilon_D$ procijenjenom preko slučajnih izbora b, e i D_h. Ako takav D_h ne postoji, tada je *h jednolika* ili *glatka* čvrstoće (ϵ_D, τ_d, U).

Jednolikost (ili glatkoća) se može pokazati nekim vrlo jednostavnim "hash" funkcijama koje nisu ni otporne na kolizije, niti su jednosmislene.

Ako imamo hash funkciju jednoliku, otpornu na kolizije, onda je ona jednosmjerna.

Model idealne grupe

U modelu idealne grupe, operaciji grupe pristupa se samo preko potprograma, koji prihvaća upite od dva elementa koji će biti oduzeti. Prikazi elemenata grupe slučajni su elementi podskupa. Prikazi se mogu smatrati enkripcijama. Bilo koja grupa koju je praktično moguće implementirati nije idealna grupa, budući da praktična implementacija mora izvoditi grupne operacije bez pozivanja potprograma.

Kada je grupa modelirana kao idealna, protivnički upiti i odgovori potprograma mogu se analizirati na sljedeći način. Neki će upiti biti elementi grupe koji se nisu pojavili u ranijim upitima ili odgovorima. Ovi se elementi mogu smatrati neovisnima. Svi ostali elementi mogu se smatrati cjelobrojnim kombinacijama neovisnih elemenata. Sve dok različite kombinacije cijelih brojeva predstavljaju različite elemente, potprogram nije otkrio nikakve odnose između neovisnih elemenata; inače će biti otkrivena ovisnost. Budući da su neovisni elementi zapravo nasumični elementi u grupi, moguće je definirati vjerojatnost otkrivanja ovisnosti, a ona je mala ako je broj upita mali. Prema dogovoru, točka baze G i javni ključ Y su neovisni. Ako je broj upita mali, mala je vjerojatnost da se ovisnost kao što je $Y = [x]G$, ekvivalentna pronalaženju privatnog ključa, može otkriti. Kada se grupa modelira kao idealna, moguće je dokazati da se određeni problemi vezani uz grupu (kao što je problem diskretnog logaritma) ne mogu riješiti s manje od određenog broja upita.

Algoritmi za rješavanje problema u grupi koji rade za bilo koju grupu, uključujući idealnu grupu, nazivaju se generičkim. Suprotno tome, algoritmi za rješavanje problema u određenoj skupini koji ne rade za druge skupine, uključujući idealnu skupinu, nazivaju se specifični. Generički algoritmi za rješavanje problema diskretnog logaritma ne mogu biti puno brži od kvadratnog korijena najvećeg prostog faktora reda grupe, dok specifični algoritmi potencijalno mogu raditi brže za određene grupe. Sigurnosni rezultat koji se može dokazati u modelu generičke grupe štiti samo od protivnika koji su generički u odnosu na grupu. Stoga nisu isključeni protivnici ECDSA specifični za skupine eliptičkih krivulja.

Idealni hash model

Idealni hash model je analogan modelu idealne grupe. Točnije, modelira hash funkciju slučajnom funkcijom potprograma. Svaki sigurnosni dokaz u idealnom hash modelu trebao bi se smatrati samo potvrdom ili argumentom sigurnosti s obzirom na dizajn sheme, bez

obzira na hash funkciju. Štoviše, on pati od istih ozbiljnih ograničenja kao i model idealne grupe.

6.5 Sigurnosni rezultati koji se mogu dokazati

Sljedeći dokazivi sigurnosni rezultati u suprotnom su smjeru od rezultata o nužnosti određenih uvjeta na primitivnim komponentama od ECDSA.

Teorem 6.5.1. *Ako je efektivna hash funkcija rijetko jednaka nuli, a polulogaritamska otpornost vrijedi za grupu i funkciju pretvorbe, tada ECDSA ima pasivnu selektivnu nemogućnost krivotvorenja.*

Teorem 6.5.2. *Ako je hash funkcija otporna na nulu i slabo otporna na kolizije, funkcija pretvorbe i grupa su zajedno otporne na prilagodljive polulogaritme, a generator slučajnih brojeva je pseudoslučajan (ne razlikuje se od slučajnog), tada ECDSA ima aktivnu selektivnu nemogućnost krivotvorenja.*

Teorem 6.5.3. *Ako je hash funkcija otporna na nulu i jednosmjerna, funkcija pretvorbe gotovo invertibilna, a grupa je modelirana kao idealna (tj. kao generička grupa), tada ECDSA ima pasivnu egzistencijalnu nemogućnost krivotvorenja.*

Još jedan rezultat za pasivnu egzistencijalnu nemogućnost krivotvorenja poseban je slučaj sljedećeg rezultata u nastavku za aktivnu egzistencijalnu nemogućnost krivotvorenja.

Teorem 6.5.4. *Ako je hash funkcija idealizirana kao slučajni potprogram, tada polulogaritamska otpornost implicira da ECDSA ima aktivnu egzistencijalnu nemogućnost krivotvorenja.*

Sada ćemo dati je dan rezultat u kojemu smo oslabili otpornost na prilagodljivi polulogaritam u rezultatu za aktivnu egzistencijalnu nemogućnost krivotvorenja.

Teorem 6.5.5. *Ako se grupa eliptičke krivulje modelira kao generička grupa, tada gotovo invertibilnost funkcije pretvorbe f te glatkoća, otpornost na kolizije i otpornost hash funkcije h na nulu zajedno impliciraju da ECDSA ima aktivnu egzistencijalnu nemogućnost krivotvorenja.*

6.6 Skice dokaza

Pasivna selektivna nemogućnost krivotvorenja

Pretpostavimo da je F pasivni selektivni krivotvoritelj i hash funkcija H je rijetko nula. Zatim ćemo pronaći polulogaritam baze G u nasumičnoj izazovnoj točki P kako slijedi.

Pokrenite F na nasumično odabranoj poruci m i javnom ključu $Y = [H(m)]P$ da dobijete krivotvorinu (r, s). Tada je $(r, H(m)^{-1}s)$ željeni polulogaritam.

Aktivna selektivna nemogućnost krivotvorenja

Pretpostavimo da je F aktivan selektivni krivotvoritelj. Tada ćemo ili riješiti problem prilagodljivog polulogaritma baze G na nasumičnoj izazovnoj točki P, ili pronaći nulu ili jednoznačnost slučajne izazovne poruke m za hash funkciju H, kako slijedi. Pokrenite F na nasumično odabranoj poruci m i javnom ključu $Y = [H(m)]P$. Odgovorite na upite za potpis $m_i \neq m$ postavljanjem upita potprogramu pomaknutog polulogaritma na $e_i = H(m)^{-1}H(m_i)$ ako je $e_i \neq 1$. Ako je neki $e_i = 1$, tada stanite, nakon što ste pronašli jednoznačnost m_i od m. U suprotnom, F proizvodi krivotvorinu (r, s) na poruci m. Tada je $(r, H(m)^{-1}s)$ željeni polulogaritam.

Pasivna egzistencijalna nemogućnost krivotvorenja

Pretpostavimo da je F pasivni egzistencijalni krivotvoritelj. Tada ćemo moći invertirati hash funkciju H u nasumičnoj izazovnoj $e \in \mathbb{Z}/q\mathbb{Z}$ na sljedeći način. Pokrenite F na modificiranom generičkom grupnom potprogramu. Potprogram generičke grupe je modificiran tako da je nasumično unaprijed odabrani odgovor potprograma određen da poprimi vrijednost $f^{-1}\left(\frac{ey}{q}\right)$, gdje su g i y cijeli brojevi tako da krivotvoritelj može primijetiti da je izlaz grupnog upita $[g]G + [y]Y$ (upiti koji nisu ovog oblika nisu odabrani za modifikaciju). Budući da je e slučajan, a F gotovo invertibilna, modificirani odgovor će biti efektivno slučajan u $\mathbb{Z}/q\mathbb{Z}$, pa će stoga stopa uspjeha F ostati nepromijenjena. Ako je Q broj grupnih upita koje koristi F, tada je $\frac{1}{Q}$ šansa da će krivotvorenje poruke zadovoljiti $s^{-1}H(m) = g$ i $s^{-1}r = y$, a posebno, m je praslika od e.

Aktivna egzistencijalna nemogućnost krivotvorenja s idealiziranim hashom

Pretpostavimo da je F aktivni egzistencijalni krivotvoritelj. Zatim ćemo upotrijebiti F da pronađemo polulogaritam baze G izazovne točke P, na sljedeći način. Pokrenite F na slučajnom hash potprogramu, modificiranom na sljedeći način. Nasumično unaprijed odabrani odgovor potprograma je određen da uzme vrijednost e odabranu nasumično iz $\mathbb{Z}/q\mathbb{Z}$. Budući da je e slučajan, modificirani odgovor će biti efektivno slučajan u $\mathbb{Z}/q\mathbb{Z}$, pa će stoga stopa uspjeha F ostati nepromijenjena. Pokrenite aktivnog egzistencijalnog krivotvoritelja s izazovnim javnim ključem $Y = [e]P$. Ako je h broj hash upita koje koristi F, tada je $\frac{1}{h}$ šansa da će krivotvorenje poruke zadovoljiti $H(m) = e$, u kojem slučaju imamo polulogaritam $(r, s/e)$, gdje (r, s) je potpis.

Kako bi se obradili upiti za potpise, koristi se daljnja modifikacija nasumičnog hash potprograma. Da biste odgovorili na upite za potpis za poruku m_i, odaberite nasumični par $(x_i, y_i) \in \mathbb{Z}_q^2$ i izračunajte $r_i = f([x_i]G + [y_i]Y)$, $s_i = \frac{r_i}{y_i}$ i $e_i = x_i r_i / y_i$. Ako krivotvoritelj kasnije u izvođenju postavi upit m_i hash potprogramu, potprogram odgovara s e_i. Odgovori na hash upit e_i su zapravo nasumični i stoga se slučajni potprogramski hash ne razlikuje od pravog nasumičnog potprograma, iz perspektive krivotvoritelja.

Unatoč tome, modificirani odgovori na upit imaju značajan utjecaj na odgovore na upite za potpis jer oni postaju deterministički. Prema tome, vjerojatno se ovaj sigurnosni dokaz odnosi samo na deterministički način ECDSA. Doista, intuitivni nasumični potpisi potencijalno propuštaju više informacija o privatnom ključu nego deterministički potpisi, jer bi protivnik nekako mogao iskoristiti više različitih potpisa jedne poruke za izračunavanje informacija o privatnom ključu.

Aktivna egzistencijalna nemogućnost krivotvorenja s idealiziranom grupom

Pretpostavimo da je F aktivni egzistencijalni krivotvoritelj, da je hash funkcija H otporna na nulu i glatka, te da je funkcija pretvorbe gotovo invertibilna. Koristit ćemo F da bismo pronašli koliziju u hash funkciji H. Pokrenite F na modificiranom generičkom grupnom potprogramu. Potprogram generičke grupe modificiran je tako da je svaki odgovor potprograma određen da poprimi vrijednost $f^{-1}(H(\tilde{m}_i y/g))$, gdje je $\tilde{m}_i$ odabrano nasumično iz U gdje su g i y cijeli brojevi tako da krivotvoritelj može primijetiti da je izlaz grupnog upita $[g]G + [y]Y$. Budući da se $H(\tilde{m}_i)$ ne može razlikovati od slučajnog, a f je gotovo invertibilan, modificirani odgovor će biti efektivno slučajan u $\mathbb{Z}/q\mathbb{Z}$, pa će stoga stopa uspjeha F ostati nepromijenjena. Nadalje, budući da je H gladak, F neće moći pogoditi $\tilde{m}_i$. Stoga se upiti krivotvoritelja potpisa m_i i krivotvorena poruka m razlikuju od poruka $\tilde{m}_i$. Kako bi se potpis mogao verificirati sa šansom boljom nego slučajno, trebao bi imati jedan od upita koji uključuje $\tilde{m}_i$ te stoga $H(m) = H(\tilde{m}_i)$, što je željena kolizija.

6.7 Ostala svojstva ECDSA

Polulogaritmi i diskretni logaritmi

Problem diskretnog logaritma tradicionalno se smatra primarnom osnovom za sigurnost ECDSA. U ovom poglavlju smo razmatrali problem polulogaritma jer nije očito je li ekvivalentan ili slabiji od problema diskretnog logaritma. Sigurnost ECDSA ovisi o ovom potencijalno slabijem problemu. Sigurnost ECDSA bit će razjašnjena kada se uspostavi odnos između ova dva problema.

Ako se ikad dokaže da je ECDSA siguran ako je diskretni logaritam eliptičke krivulje nerješiv, onda bi to uspostavilo ekvivalentnost problema polulogaritma eliptičke krivulje i diskretnog logaritma. To vrijedi čak i ako sigurnosni dokaz koristi model slučajnog potprograma za hash funkciju. Također, stručnjaci za eliptičke krivulje mogu proučavati ekvivalentnost ova dva problema bez gledanja ECDSA ili hash funkcije.

Primjena rezultata na ECDSA varijante

Moguće su različite varijante ECDSA, pa razmotrimo odnose li se raniji rezultati na te varijante.

Hashiranje kG

Jedna predložena varijanta ECDSA uključuje efemerni javni ključ $[k]G$ kao dio ulaza za "sažetak poruke" hash funkcije. U nasumičnom modelu potprograma idealiziranom za hash funkcije, to doprinosi dokazivoj sigurnosti jer se aktivna egzistencijalna nemogućnosti krivotvorenja ECDSA može dokazati oslanjajući se na problem diskretnog logaritma, a ne na polulogaritam. S druge strane, pod različitim hipotezama ova varijacija umanjuje dokazivu sigurnost, kako slijedi. Neki sigurnosni dokazi za ECDSA više ne vrijede. Dokaz aktivne selektivne nemogućnosti krivotvorenja koja ne idealizira ni hash ni grupu nije primjenjiv. Čini se da prepreka ovim tehnikama dokazivanja proizlazi iz poteškoća u odvajanju uloga grupe i hash-a u ovoj varijanti. Nadalje, ovo također predstavlja primjer načela da složenost dizajna općenito ometa dokazivu sigurnost.

DSA i funkcije jednosmjerne pretvorbe

Gotovo invertibilnost ne vrijedi za DSA funkciju pretvorbe. U DSA, funkcija pretvorbe je vjerojatno dobra jednosmjerna funkcija, što je sasvim suprotno od gotovo invertibilne. Stoga dokazivi sigurnosni rezultati korištenjem gotovo invertibilnosti funkcije pretvorbe nisu dobro primjenjivi na DSA. Stoga DSA i ECDSA imaju različita dokaziva sigurnosna svojstva.

Nepseudoslučajni k

Nijedan rezultat nije pokazao da se k mora razlikovati od uniformnog slučajnog cijelog broja u $[1, q-1]$. Doista, budući da ECDSA nije namijenjen pružanju povjerljivosti, nije jasno je li nerazlučivost potrebna. Intuitivno, slabiji uvjet od pseudoslučajnosti trebao bi biti dovoljan za ECDSA.

Da bismo vidjeli zašto pseudoslučajnost možda nije potrebna za k, razmotrimo sljedeće. Odaberimo istinski nasumične privatne ključeve k pod uvjetom da njihovi hashovi prika-

zuju zadani uzorak. Takvi k nisu pseudoslučajni jer se mogu razlikovati primjenom hash funkcije, no oni nisu slabi. Malo je vjerojatno da će imati aritmetičku pristranost koju je moguće napasti. Mogu imati dovoljno entropije da ih se ne može pogoditi.

Također, neki od rezultata ne uključuju potprogram za potpisivanje i stoga ne zahtijevaju da se efemerni privatni ključevi k generiraju pseudoslučajno.

Deterministički k

U nekim dokazima, vrijednost potprograma za potpisivanje ima svojstvo da isti upit poruke uvijek daje isti odgovor potpisa. Tehnički, to znači da je dokaz primjenjiv samo na deterministički način ECDSA potpisivanja, gdje je k odabran kao tajna deterministička funkcija poruke m koja se potpisuje. Intuitivno objašnjenje da je deterministički način rada sigurniji je to što otkriva manje potpisa i time manje informacija o privatnom ključu. Vrlo oprezna implementacija ECDSA mogla bi koristiti deterministički način kako bi se primijenili ovi dokazivi sigurnosni rezultati.

Napadačke osobine ECDSA

Alternativne definicije sigurnosti za potpise u kojima se ECDSA može smatrati "nesigurnim" istražuju se i procjenjuje se njihova relevantnost. Mnoge osobine koje istražujemo ovise o određenoj funkciji pretvorbe f koja se koristi u ECDSA. Promjena f u svrhu izbjegavanja ovih osobina mogla bi potencijalno učiniti više štete nego koristi, smanjujući reputacijsku sigurnost ECDSA i dokazivu sigurnost ECDSA. U skladu s tim, pristupanje ovim osobinama najbolje je odraditi na druge načine.

Potpisi bez anonimnosti

Uz važeći ECDSA potpis (r, s) na poruci m, pridruženi javni ključ Y može se obnoviti na sljedeći način. Riješimo javni ključ kao $Y = [r^{-1}]([s]R - [H(m)]G)$, gdje je R odabran iz $f^{-1}(r)$, skup točaka u praslici od r.

Samopotpisani potpisi

Potpis poruke je samopotpisan ako poruka sadrži potpis. Samopotpisani ECDSA potpis može se generirati na sljedeći način. Odaberimo slučajno k i s. Izračunajmo $r = f([k]G)$. Oblikujmo poruku m koja sadrži potpis (r, s). Izračunajmo $e = H(m)$. Sada riješimo za privatni ključ x koji ovaj potpis čini valjanim, a koji se može pronaći kao $x = (\pm sk - e)/r \pmod{q}$.

Ovo može biti korisna osobina jer se može koristiti kako bi se osiguralo da privatni ključ nije ukraden. Također može biti korisna za generiranje ključa uz pomoć poslužitelja,

gdje poslužitelj dodaje entropiju poruci m tako da potpisnikov privatni ključ x ima dovoljno entropije. Međutim, ako se poslužitelju ne može vjerovati, a entropija potpisnika za k je slaba, potrebne su dodatne izmjene samopotpisane provjere potpisa.

Nepoznati privatni ključ

Valjani ECDSA potpis može se generirati bez poznavanja privatnog ključa, a da se ipak ne krši definicija sigurnosti potpisa, kako slijedi. To se može učiniti za bilo koje parametre domene eliptičke krivulje i bilo koju poruku m, prvo generiranjem nasumične vrijednosti potpisa (r, s) i zatim rješavanjem javnog ključa kao $Y = [r^{-1}]([s]R - [H(m)]G)$, gdje je $R \in f^{-1}(r)$, skup točaka u praslici od r. Ako je $f^{-1}(r) = \{\}$, onda samo pokušajte s drugom vrijednošću r.

Ovaj bi se "napad" mogao spriječiti ako bi f bio jednosmjeran, ali tada bi bila izgubljena dokaziva sigurnost koju nude dokazi koji pretpostavljaju gotovo invertibilnost f.

Nevažeći javni ključ

ECDSA potpis koji se može provjeriti može se generirati za onoliko poruka koliko želite nakon prethodnog povezivanja s javnim ključem Y, kako slijedi. Neka je Y nevažeći javni ključ malog reda a (u ovom slučaju, verifikator treba koristiti Y bez provjere). Neka je m zadana poruka. Generirajte slučajne s i t. Izračunajte $r = f([H(m)]G + [t]Y)$. Ako je $r = t \pmod a$, tada će (r, s) vrijediti.

Dvostruki potpisi

ECDSA potpis koji je valjan za dvije zadane poruke m_1 i m_2 može se generirati na sljedeći način. Odaberite nasumično k, zatim izračunajte $R = [k]G$, $r = f(R)$, $x = (2r) - 1(H(m_1) + H(m_2)) \pmod q$ i $Y = [x]G$ i $s = k - 1(H(m_1) + xr) \pmod q$. Ovo radi jer je $f(P) = f(-P)$.

Prilagodljivi potpisi

Valjani ECDSA potpis koji nikada nije generirao potpisnik s javnim ključem Y može se generirati za bilo koju poruku m na sljedeći način. Neka potpisnik generira potpis (r, s) poruke m, a zatim generira $(r, -s \pmod q)$ kao željeni potpis. Ovo funkcionira jer je $f(P) = f(-P)$.

Bibliografija

[1] I. F. Blake, G. Seroussi i N. P. Smart, *Advances in Elliptic Curve Cryptography*, Cambridge University Press, 2005.

[2] J. A. Buchmann, *Introduction to Cryptography*, Springer-Verlag NY, 2004.

[3] CARNet, *Algoritmi za izračunavanje sažetka*, https://www.cert.hr/wp-content/uploads/2006/08/CCERT-PUBDOC-2006-08-166.pdf.

[4] A. Dujella, *Teorija brojeva u kriptografiji*, 2003, https://web.math.pmf.unizg.hr/~duje/tbkript/tbkriptlink.pdf.

[5] A. Dujella, *Eliptičke krivulje u kriptografiji*, 2013, https://web.math.pmf.unizg.hr/~duje/elkript/elkripto2.pdf.

[6] D. Hankerson, A. Menezes i S. Vanstone, *Guide to Elliptic Curve Cryptography*, Springer-Verlag NY, 2004.

[7] J. Hoffstein, J. Pipher i J. H. Silverman, *An Introduction to Mathematical Cryptography*, Springer, 2008.

[8] A. J. Menezes, P. C. van Oorschot i S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[9] A. Menezes, S. Vanstone i D. Johnson, *The Elliptic Curve Digital Signature Algorithm (ECDSA)*, 2001, https://www.cs.miami.edu/home/burt/learning/Csc609.142/ecdsa-cert.pdf.

[10] R. Smith, *The Discrete Logarithm Problem on Supersingular Elliptic Curves*, 2020, https://fse.studenttheses.ub.rug.nl/22732/1/bMATH_2020_SmitR.pdf.

[11] D. R. Stinson i M. B. Paterson, *Cryptography: Theory and Practice*, CRC Press, 2019.

[12] B. Širola, *Algebarske strukture*, https://web.math.pmf.unizg.hr/nastava/alg/predavanja/ASpred.pdf.

Sažetak

U ovom diplomskom radu smo proučavali Digital Signatures Elliptic Curves Algorithm (ECDSA). ECDSA je varijanta algoritma digitalnog potpisa (DSA) koji koristi problem diskretnog logaritma na eliptičkim krivuljama nad konačnim poljima. To je najraširenija standardizirana shema potpisa temeljena na eliptičkim krivuljama. Sigurnost ECDSA se temelji na tome da ne postoje efikasni algoritmi izračuna za problem diskretnog logaritma na eliptičkim krivuljama.

Na početku ovog rada smo definirali osnovne pojmove kriptografije te pojmove kao što su digitalni potpisi i hash funkcije i njihovu sigurnost. Zatim smo uveli pojam eliptičke krivulje nad konačnim poljem te operacije na njoj. Sljedeći korak nam je bio uvesti problem diskretnog logaritma nad eliptičkim krivuljama (ECDLP) te "Dupliciraj i zbroji" algoritam. Spomenuli smo i dva algoritma - index calculus metodu i Shanksovu "baby step-giant step" (BSGS) metodu, od kojih jedan rješava problem diskretnog logaritma u $\mathbb{F}_p^*$, dok drugi rješava ECDLP sa složenosti $O\left(\sqrt{n}\right)$. Nakon što smo uveli sve ove pojmove, možemo uvesti ECDSA algoritam te proučiti njegovu sigurnost i skicirati dokaze sigurnosnih rezultata.

Summary

In this diploma thesis, we studied Digital Signatures Elliptic Curves Algorithm (ECDSA). ECDSA is a variant of the Digital Signature Algorithm (DSA) that uses the discrete logarithm problem on elliptic curves over finite fields. It is the most widely used standardized signature scheme based on elliptic curves. The security of ECDSA is based on the fact that there are no efficient methods known for computing the discrete logarithm problem on elliptic curves.

At the beginning of this thesis, we defined the basic terms of cryptography and concepts such as digital signatures and hash functions and studied their security. Then we introduced the concept of elliptic curves over finite fields and operations on them. The next step was to introduce the Elliptic Curve Discrete Logarithm Problem (ECDLP) and the double-and-add algorithm. We also mentioned two algorithms - the index calculus method and Shanks's baby-step/giant-step method, one of which solves the discrete logarithm problem in $\mathbb{F}_p^*$, while the other solves ECDLP with complexity $O\left(\sqrt{n}\right)$. After introducing all these concepts, we can introduce the ECDSA algorithm and study its security and proof sketches of security results.

Životopis

Mihaela Zima rođena je u Virovitici 26. siječnja 1995. godine. Završila je Osnovnu školu Ivana Nepomuka Jemeršića te smjer opće gimnazije u Srednjoj školi Bartola Kašića u Grubišnom Polju. Nakon završene srednje škole školovanje je nastavila u Zagrebu. Godine 2013. upisala je preddiplomski studij Matematika na Prirodoslovno-matematičkom fakultetu. Nakon što je 2019. završila preddiplomski studij, upisala je diplomski sveučilišni studij Primijenjena matematika na Prirodoslovno-matematičkom fakultetu u Zagrebu.