UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika - uporabna smer (UNI)

Mitja Korče

WEILOVO PARJENJE V SHEMAH ZA ŠIFRIRANJE

Diplomsko delo

Ljubljana, Januar 2006

Zahvala

Zahvaljujem se mentorju doc. dr. Aleksandru Jurišiću in dr. Arjani Žitnik za vso pomoč pri pisanju tega diplomskega dela.
Posebna zahvala gre moji Nataši za vso podporo in lektoriranje besedila in mojim domačim.

Kazalo

PROGRAM DIPLOMSKEGA DELA

Delo naj predstavi matematične osnove, potrebne za razumevanje digitalnih podpisov in certifikatov. Eden izmed osrednjih problemov uporabe certifikatov je preverjanje njihove veljavnosti. Glavni cilji so:

(a) Šifriranje na osnovi identititete (angl. IBE Identity-based encryption)

(b) Šifriranje na osnovi certifikatov (angl. CBE Certificate-based encryption)

(c) Weilovo parjenje, ki je eno izmed pomembnih orodij pri študiju eliptičnih krivulj in je bilo uporabljeno tudi za rešitev problema diskretnega logaritma na supersingularni eliptični krivulji.

Osnovna literatura:

1. C. Gentry, *Certificate-based encryption and the certificate revocation problem*, Advances in Cryptology – EUROCRYPT'03, LNCS **2656**, Springer-Verlag, 2003, 272–293.

2. D. Boneh in M. Franklin, *Identity-based encryption from the Weil pairing,* Advances in Cryptology – CRYPTO'01, LNCS **2139**, Springer-Verlag, 2001, 213–229.

3. Douglas R. Stinson, *Cryptography – Theory and Practice*, CRC Press, 1995.

4. A. Enge, *Elliptic Curves and Their Applications to Cryptography - An Introduction*, Kluwer Academic Publishers, 1999.

Ljubljana, september 2005

POVZETEK

V tem diplomskem delu je poudarek na bilinearni preslikavi, imenovani Weilovo parjenje, ki se uporablja v številnih shemah za šifriranje z eliptičnimi krivuljami in tudi za reševanje problema diskretnega algoritma na eliptičnih krivuljah. Za izpeljavo definicije Weilovega parjenja je potrebno poznati osnove teorije deliteljev, zato so v diplomsko delo vključene tudi te. Predstavljen je tudi konkreten primer sheme, kjer je uporabljeno takšno parjenje. Poimenovana je shema za šifriranje s certifikati, saj v njej certifikat uporabljamo tudi kot odšifrirni ključ.

Na začetku je predstavljen osnovni model sheme za šifriranje s certifikati in po definiciji Weilovega parjenja so podrobneje opisani še algoritmi, v katerih se parjenje uporablja. Na koncu je podan opis razširjene sheme, kjer se z uporabo pokritja množice uporabnikov z veljavnim certifikatom zmanjša računska zahtevnost na strani certifikatne agencije.

Ključne besede: kriptografija, sheme za šifriranje, IBE, CBE, preklic certifikata, Weilovo parjenje, teorija deliteljev

ABSTRACT

The main topic of this thesis is a billinear map called Weil pairing. It is used in many elliptic curve cryptosystems. The definiton of Weil pairing cannot be given without some knowledge of the divisor theory. I also present an example of an encryption scheme, in which Weil pairing is used. It is called certificate-based encryption by its author Craig Gentry. In this scheme a certificate acts also as a decryption key.

First, the basic model of certificate-based encryption is presented. After the definiton of the Weil pairing the algorithms with Weil pairing are described in detail. Finally, the incremental scheme is described, in which computation costs of the certificate authority are dramatically improved with the use of subset covers.

Keywords: cryptography, encryption shemes, IBE, CBE, certificate revocation, Weil pairing, divisor theory

Math. Subj. Class. (2000): 94A60, 11T71, 14G50, 14H52

Poglavje 1

Uvod

Pred množično uporabo računalnikov, elektronske pošte in modernih načinov komuniciranja so ljudje shranjevali zaupne podatke v skrbno varovanih prostorih ali pa v trezorjih. Pri teh podatkih in njihovem pošiljanju je bila največja nevarnost, da jih v roke dobi sovražnik oz. oseba, ki naj ne bi imela dostopa do njih. S podpisom ali žigom so avtorji jamčili, da je dokument njihov in veljaven. Dandanes je večina podatkov, naj bodo zaupne ali pa javne narave, v digitalni obliki, shranjena na diskih osebnih računalnikov, strežnikih ali kakih drugih modernih medijih. V tej obliki ne gre več le za nevarnost odkritja zaupnih podatkov, gre tudi za celovitost podatkov, originalnost. Pri pošiljanju digitalnega sporočila mora biti prejemnik prepričan, da je dobljeni dokument enak, kot je bil v času pošiljanja. Poleg tega mora biti tudi prepričan, da je pošiljatelj res oseba, za katero se izdaja. Veda, ki išče rešitve teh problemov, je kriptografija. Dobri kriptosistemi naj bi se držali Kerckhoffovega principa, ki pravi, da nasprotnik pozna kriptosistem oziroma algoritme, ki jih uporabljamo, ne pa tudi ključe, ki nam zagotavljajo varnost. Žal pa se ta princip ne upošteva vedno in se varnost kriptosistemov med drugim gradi tudi na skrivanju uporabljenih algoritmov.

Šifriranje sporočil sega že daleč nazaj. Špartanci so na primer na valj navili ozek trak in pravokotno nanj napisali sporočilo. Potem so trak odvili in poslali. Prejemnik je moral imeti valj enakega premera, da je lahko prebral sporočilo. Julij Cezar je uporabljal pomično šifro. Vsako črko je zamenjal s črko, ki je v abecedi nekaj mest za njo. Postopek lahko matematično opišemo kot $c = a + k \bmod n$, kjer je k ključ, n pa število črk v abecedi. Prejemnik je sporočilo odšifriral tako, da je izračunal $a = c - k \bmod n$. Podoben postopek so ob začetku svetovnega spleta uporabljali v Usenetu za šifriranje neprimernih

šal in podobnih sporočil. Takšno šifriranje seveda ne predstavlja nobene varnosti, saj lahko z malce truda iz tajnopisa dobimo čistopis. Seveda pa se je z nadaljnjim razvojem povečala tudi želja in nuja po varnosti in zasebnosti. Prvotni primeri šifriranja spadajo med t. im. simetrične algoritme oziroma algoritme s skritim ključem. Pri teh algoritmih sta si morali osebi pred varnim komuniciranjem izmenjati skriti ključ, katerega sta potem obe uporabljali za šifriranje in odšifriranje. Protokol, ki skrbi za avtentikacijo osebe, se imenuje *kerberos*. Gre za to, da za avtentikacijo skrbi nek strežnik, imenovan tudi center zaupanja, kateremu morajo uporabniki brezpogojno zaupati. Uporabnika pred komunikacijo od centra pridobita skriti ključ za trenutno sejo, katerega potem uporabita za šifriranje in odšifriranje sporočil. Najbolj znana in razširjena simetrična šifra je DES.

Leta 1976 sta Whitfield Diffie in Martin Hellman predstavila pojem kriptografije z javnimi ključi. Kriptografija z javnimi ključi za razliko od kriptografije s simetričnimi ključi uporablja par ključev, javni ter skriti ključ. Pri Diffie-Hellmanovem protokolu za izmenjavo ključev gre za to, da si lahko osebi, poimenovali ju bomo Bojan in Anita, preden želita varno komunicirati, javno med sabo izmenjata neke parametre oz.podatke. Osnovna ideja protokola je preprosta. Imamo dva parametra, p in g, ki sta javna in ju lahko uporabljajo vsi uporabniki sistema. Parameter p je veliko praštevilo, medtem ko je g število manjše od p, za katerega velja, da za vsako število n med vključno 1 in $p-1$ obstaja k, tako da je $g^k = n \mod p$. Tak g ponavadi imenujemo generator. Anita si izbere neko skrito celo število a, izračuna vrednost $g^a \mod p$ in dobljeni rezultat pošlje Bojanu. Podobno si Bojan izbere svoje skrito celo število b, izračuna $g^b \mod p$ in rezultat pošlje Aniti. Oba nato izračunata $k = g^{ab} \mod p$, kar je sedaj njun skriti ključ. Varnost protokola sloni na t. im. problemu diskretnega logaritma. Predpostavljamo namreč, da tudi z zelo veliko računsko močjo iz g^a in g^b ni mogoče izračunati g^{ab}. Diffie-Hellmanova izmenjava ključev ni varna za napade, imenovane *napadi s prestrezanjem* (ang. man in the middle attack). Napadalec pri komunikaciji med Anito in Bojanom prestreže vrednosti g^a in g^b in se z njima dogovori za drugačna ključa. Ko si Bojan in Anita, nevedoč za človeka v sredini pošiljata šifrirana sporočila, jih napadalec prestreže, odšifrira, prebere, lahko tudi spremeni, šifrira nazaj s svojim ključem in pošlje naprej. Takšni napadi na protokol so možni, saj v protokolu ni nobene avtentikacije, torej pri potrjevanja identitete uporabnikov.

Problem avtentikacije pri Diffie- Hellmanovi izmenjavi ključev rešimo z uporabo digitalnih podpisov in certifikatov. Tu nastopi certifikatna agencija. Pred izmenjavo ključev

Anita in Bojan pridobita svoj par ključev. Javni ključ nato certifikatna agencija potrdi in vključi v certifikat, katerega potem izda uporabniku. Seveda mora biti certifikatna agencija pri izdaji certifikata, torej potrjevanju javnega ključa prepričana, da je oseba z javnim ključem res tista, za katero se izdaja. Ponavadi je potrebno za pridobitev certifikata fizično oditi na certifikatno agencijo, kjer lahko potrdijo identiteto uporabnika. Med pošiljanjem sporočil Anita podpiše sporočilo, ki vsebuje tudi $g^a \mod p$, podobno stori tudi Bojan. Kljub temu, da napadalec lahko prestreže šifrirana sporočila, pa ne more ponarediti podpisa, ne da bi vedel vrednost Anitinega ali Bojanovega skritega ključa.

Pri kriptosistemih z javnim ključem se znebimo zahteve po varnem kanalu med pošiljateljem in prejemnikom, vendar pa se poveča časovna in računska zahtevnost algoritmov. Poleg tega se poveča tudi velikost samih ključev. Ravno zato veliko shem za šifriranje uporablja kombinacijo obeh sistemov.

V zadnjem času se pojavlja vedno več primerov shem za šifriranje. Namen tega diplomskega dela je bralca seznaniti z dvema vrstama teh, t. im. shemo za šifriranje s certifikati in pa shemo za šifiranje na osnovi identitete. Bistvo prve je, da uporabnik svoj certifikat ne uporablja le za podpisovanje sporočila ali avtentikacijo, temveč tudi za šifriranje sporočil. Pri slednji pa gre za to, da je uporabnikov javni ključ zgrajen iz niza, ki uporabnika identificira (npr. elektronski naslov). Poleg tega bi rad v tem diplomskem delu predstavil vsaj delček matematike, ki se skriva v ozadju kriptografije.

Delo je razdeljeno na štiri glavne dele. Za uvodnim poglavjem je podan opis infrastrukture javnih ključev in nekaj trenutno najbolj uporabljenih metod za preklic certifikata. To je en izmed glavnih problemov infrastrukture javnih ključev. V opisanih shemah je ta problem rešen na eleganten in morda malce prikrit način, saj prejemnik šifriranega sporočila ne more odšifrirati, če nima potrjene veljavnosti svojega certifikata. Naslednje poglavje je matematično poglavje, v katerem si najprej ogledamo osnove eliptičnih krivulj. Sledi teorija deliteljev, ki so matematične strukture, s katerimi si pomagamo pri preučevanju ničel in polov racionalnih funkcij, njej pa sledi definicija bilinearne preslikave, imenovane Weilovo parjenje. Poleg Tateovega parjenja je to najbolj uporabna bilinearna preslikava v prej omenjenih shemah. Za poglavjem o matematičnih pojmih pride na vrsto še konkreten opis osnovne sheme z uporabo Weilovega parjenja, njene izboljšave in v zadnjem poglavju nekaj poslošitev in razširitev te sheme.

Poglavje 2

PKI in metode preklica certifikata

V sistemih z uporabo javnih ključev in certifikatov predstavlja vzdrževanje veljavnosti certifikatov pomemben problem. Certifikate namreč certifikatna agencija izdaja za daljša obdobja, npr. za eno leto, lahko tudi pet ali deset let. Seveda lahko v takem obdobju pride do preklica veljavnosti javnega ključa iz več razlogov. Najpomembnejša sta odkritje pripadajočega skritega ključa ali pa odhod zaposlenega z institucije, kjer je bil zaposlen in v kateri so uporabljali certifikate za pregledovanje in podpisovanje zaupnih sporočil. V obeh primerih mora certifikatna agencija zagotoviti, da lastnik certifikata oz. para ključev ne more tega več uporabljati.

V tem poglavju je najprej opisana zgradba infrastrukture javnih ključev **PKI** (ang. Public Key Infrastructure), nato pa še najbolj pogoste metode za reševanje problema vzdrževanja veljavnosti certifikatov.

2.1 Infrastruktura javnih ključev

Infrastruktura javnih ključev je sestavljena iz naslednjih elementov:

1.) *Certifikatne agencije* (CA - ang. Certificate Authority), ki izdajajo in vzdržujejo veljavnost javnih ključev v obliki certifikatov. Certifikatna agencija je lahko ena, lahko pa jih je tudi več in so drevesno razporejene. Certifikatni agenciji v korenu drevesa pravimo *korenska certifikatna agencija* (ang. Root CA).

2.) *Registratorji certifikatov* (RA - ang. Registration Authority), ki so zadolženi za potrjevanje identitete uporabnika. Pri preprostejših infrastrukturah je lahko registrator

kar certifikatna agencija sama.

3.) *Direktoriji* so nekakšne shrambe certifikatov ter njihovih statusov. Če se preklic certifikatov v sami infrastrukturi vodi preko seznama preklicanih certifikatov (CRL), so le ti tudi shranjeni v direktorijih.

4.) *Uporabniki* hranijo svoje javne ključe v certifikatih, podpisanih s strani izdajatelja. Struktura certifikata je javno poznana, kot tudi protokoli, s katerimi je bil certifikat (javni ključ) izdelan. Skriti ključ uporabniki hranijo na varnem mestu.

2.2 Nekatere že znane metode preklica certifikatov

Ob večjem številu uporabnikov s certifikati postaja problem vzdrževanja veljavnosti oziroma preklic certifikatov vse bolj pomemben del infrastrukture. Glede na to, da naj bi bila približno ena desetina izdanih certifikatov z enoletno veljavnostjo preklicana še preden jim poteče obdobje veljavnosti ([24]), nas ne preseneča, da se pojavlja vedno več bolj ali manj uspešnih metod za preklicevanje certifikatov. V nadaljevanju si bomo ogledali nekatere do sedaj predstavljene metode za preklic certifikatov oziroma za vzdrževanje veljavnosti le teh.

Seznam preklicanih certifikatov

Najbolj preprosta, a tudi najmanj učinkovita rešitev problema je seznam preklicanih certifikatov (**CRL** - ang. Certificate Revocation List). Izdajatelj certifikatov (certifikatna agencija) periodično (urno, dnevno, tudi tedensko) obnavlja podpisan seznam. V seznamu imamo serijske številke preklicanih certifikatov, dejanski datum preklica in razlog preklica certifikata pred potekom veljavnosti. Poleg podatkov certifikatov pa seznam vsebuje tudi datum trenutne ter datum naslednje objave. Problem takšnega preklica certifikata je v tem, da je seznam lahko dolg, če izdajatelj, certifikatna agencija, izda veliko število certifikatov.

Protokol za sprotno preverjanje veljavnosti

Naslednja metoda preklica je protokol za sprotno preverjanje veljavnosti (**OCSP** - ang. Online Certificate Status Protocol). OCSP metoda nudi manjši časovni interval med

dejanskim preklicem certifikata in distribucijo zainteresiranim strankam. OCSP odpravi največjo slabost metode seznama preklicanih certifikatov, to je veliko število prenosov seznama s strani uporabnika. Po drugi strani pa zahteva generiranje digitalno podpisanega odgovora na vsako poizvedbo posebej. Ko želi uporabnik dobiti podatke s strežnika, OCSP pošlje zahtevo za pridobitev statusa certifikata. Strežnik lahko odgovori s tremi možnimi odgovori: 'veljaven', 'preklican' ali 'neznan'. Odgovor 'veljaven' ima lahko tri pomene:

- Dejansko pomeni, da je certifikat veljaven, torej še ni bil preklican.
- Sprašujemo o veljavnosti certifikata v trenutku izven obdobja njegove vnaprej določene veljavnosti.
- Certifikat sploh še ni bil izdan.

Protokol torej definira komunikacijo med uporabnikom in strežnikom. Slabost te metode je odprtost na **DoS** (ang. Denial of Service) napade. DoS napad je napad na računalniški sistem ali omrežje. Napadalec poskuša zasesti celotno širino mrežne oz. internetne povezave, ali pa z velikim število računsko zahtevnih poizvedb ohromi računalniški sistem.

Status preklica certifikata

Zanimivo rešitev je predstavil Micali [2] z metodo **CRS** (ang. Certificate Revocation Status). CRS vsebuje podatke enega samega certifikata in je precej krajši kot CRL. Generiranje CRS je hitrejše kot samo podpisovanje dokumenta in tudi njegova dolžina je krajša od digitalnega podpisa. Glavna razlika med CRS in ostalimi metodami je že v obliki certifikata, saj sta poleg standardnih elementov, kot je javni ključ uporabnika, ime ali elektronska pošta, serijska številka certifikata in postopki za generiranje certifikata, v njem vključena še dva elementa. Prvi je 100-bitni Y, ki predstavlja oznako za 'da', drugi pa 100-bitni N, ki predstavlja 'ne'. Ti dve vrednosti sta z veliko verjetnostjo unikatni za vsak certifikat. Certifikatna agencija zgenerira Y, tako da si izbere naključno 100-bitno število Y_0 in na njem 365-krat izvede enosmerno funkcijo F, N pa dobi tako, da izbere naključni 100-bitni N_0 in na njem enkrat izvede F. Y_0 in N_0 sta skriti števili in certifikatna agencija ju hrani na varnem.

Posodabljanje CRS-a poteka na naslednji način. Dnevno certifikatna agencija pošlje direktoriju naslednje informacije:

- Svež in overjen seznam serijskih številk izdanih in še ne preklicanih certifikatov.
- Za vsak izdan in še ne preklican certifikat pošlje 100 bitno zaporedje, izračunano na naslednji način: predpostavimo, da smo na i-tem dnevu (i-ti dan v letu, i-ti dan po izdaji certifikata ali podobno). Če je certifikat še veljaven, potem certifikatna agencija pošlje vrednost $Y_{365-i} = F^{365-i}(Y_0)$. Ker je funkcija F enosmerna, ne more nihče iz Y_{365-i} izračunati Y_0. Če je bil certifikat na ta dan preklican, potem pošlje vrednost N_0. Če želi, lahko k temu pridruži še dodatne podatke o samem preklicu (datum, razlog, itd.).
- Certifikate, ki so bili na novo zgenerirani ta dan.

Direktorij za vsako certifikatno agencijo v infrastrukturi hrani njene še nepreklicane certifikate, urejene po serijskih številkah, poleg tega pa še njihove zadnje vrednosti Y, če je certifikat še veljaven, sicer pa $N_0 = F^{-1}(N)$. Ker je F enosmerna funkcija, vrednosti N_0 ni mogoče itračunati in jo pozna le certifikatna agencija, ki si ga je izbrala na začetku. Poleg tega direktorij izvaja osnovna preverjanja podatkov. Če uporabnik dobi javni ključ prejemnika kar preko njega, potem mora od direktorija dobiti le potrditev veljavnosti za znani certifikat. Direktorij uporabniku na njegovo zahtevo pošlje zadnjo 100-bitno vrednost, ki pripada temu certifikatu. Direktorij, kateremu se pri ostalih shemah ne zaupa, tudi tu nima povečane stopnje zaupanja, saj preklicanega certifikata ne more spremeniti v veljavnega. To je res, saj bi moral na dan i za certifikat, ki je bil preklican na dan $j < i$, znati izračunati vrednost $Y_{365-i} = F^{-(i-(j-1))}(Y_{365-(j-1)})$, torej izračunati inverz enosmerne funkcije F. Podobno direktorij ne more preklicati veljavnega certifikata, saj bi moral znati izračunati vrednost $F^{-1}(N)$.

Uporabnik, ki na poizvedbo o statusu certifikata dobi odgovor, da certifikat ne obstaja, lahko to preveri, saj certifikatna agencija vedno poda dokaz o tem. Sicer pa na dan i uporabnik preveri, ali je $F^i(V) = Y$ ali $F^i(V) = N$, kjer je V vrednost, katero dobi od direktorija. V primeru, ko dobljena vrednost ne ustreza nobenemu pogoju lahko sklepa, da mu direktorij namerno ne vrne pravega podatka.

Poglavje 3

Šifriranje s certifikati

V infrastrukturi javnih ključev je eno pomembnejših vprašanj vodenje veljavnosti izdanih certifikatov. Kot smo si ogledali že v poglavju 2, obstaja že kar nekaj rešitev tega problema, vendar pa rešitve, ki bi pritegnila izdajatelje certifikatov še ni. Najbolj razširjena rešitev je seznam preklicanih certifikatov, ki pa je pri večjem številu uporabnikov neuporaben. V shemi za šifriranje s certifikati bo ta problem rešila sama zasnova sheme. Pošiljatelju sporočila ne bo potrebno poizvedovati o veljavnosti certifikata prejemnika, saj shema omogoča prejemniku odšifriranje sporočila le, če ima veljaven oziroma potrjen certifikat v trenutku, ko sporočilo prejme.

V tem poglavju si bomo ogledali pojem sheme za šifriranje s ceritikati **CBE** (ang. Certificate-Based Encryption). Na začetku bomo definirali nekatere pojme v povezavi z varnostjo in slabostmi ostalih shem, katere CBE shema odpravi. Nadaljevali bomo z opisom modela osnovne sheme. Končali bomo s kratko oceno varnosti.

3.1 Poizvedbe tretje osebe

S poizvedbami tretje osebe (ang. Third-Party Queries) mislimo na poizvedbe oseb, ki niso lastniki certifikatov, za katere bi radi izvedeli stanje oz. potrdili veljavnost. Predpostavimo, da uporabnik svojega certifikata ne bo uporabljal za šifriranje in odšifriranje, temveč le za podpisovanje dokumentov in preverjanje podpisov. V tem primeru ne potrebujemo nobenih poizvedb prejemnika o statusu certifikata podpisnika, saj lahko podpisnik pri pošiljanju podpisanega sporočila pošlje tudi potrdilo o veljavnosti certifikata

v trenutku podpisovanja. Tu je tretja oseba prejemnik, saj bi potreboval informacijo o veljavnosti certifikata pošiljatelja, ne svojega certifikata. V tem primeru potrebujemo tako infrastrukturo, ki uporabniku omogoča le pridobitev potrdila o veljavnosti svojega certifikata.

Situacija se spremeni, če uporabniki uporabljajo certifikate tudi za šifriranje in odšifriranje. V tem primeru mora pošiljatelj imeti podatke o statusu prejemnikovega certifikata preden mu pošlje šifrirano sporočilo. Lahko bi ga zahteval od prejemnika samega, vendar to ni dobra rešitev, saj ni nujno, da bo le ta lahko odgovoril takoj (npr. elektronska pošta). Lahko bi dobil potrdilo z nekega strežnika, povezanega s prejemnikom in ne s certifikatno agencijo, vendar je to na silo dobljena rešitev in še vedno uporabimo dodatno poizvedbo.

Poizvedb tretje osebe bi se radi znebili ali pa vsaj čimbolj zmanjšali iz več razlogov. Prvi je ta, da mora imeti vsak strežnik v infrastrukturi podatke o statusu certifikatov vseh uporabnikov, ne le tistih, katerim streže sam, kot bi bilo v primeru, če bi uporabnik zahteval potrdila o statusu le za svoj certifikat. V tem primeru bi lahko strežnik periodično pošiljal statuse certifikatov vsem svojim uporabnikom, kar bi vidno zmanjšalo stroške pošiljanja certifikatne agencije. Drugič, poizvedbe tretje osebe povečajo obdelavo podatkov certifikatne agencije ali njenih strežnikov. Če želi npr. vsak uporabnik dobiti podatke o statusu certifikata desetih drugih uporabnikov, mora sistem obdelati $10N$ poizvedb, kjer je N število uporabnikov. Tretjič, zakaj bi certifikatna agencija izdajala sveže informacije o statusih njenih uporabnikov osebam, ki niso njeni uporabniki? Nazadnje še, če mora certifikatna agencija izdajati informacije osebam, ki niso njeni uporabniki, s tem postane precej bolj dovzetna za DoS napade. Če torej odstranimo poizvedbe tretjih oseb, s tem zmanjšamo stroške celotne infrastrukture, poenostavimo njeno poslovanje in povečamo varnost.

Kako se torej znebimo poizvedb tretjih oseb, ne da bi s tem omejili uporabo ključev? V nadaljevanju opisan pristop temelji na *implicitnem potrjevanju*. Z implicitnim mislimo predvsem na to, da pošiljatelju ni potrebno eksplicitno pridobiti podatkov o statusu certifikata prejemnika po vsaki posodobitvi certifikatov, ampak za pošiljanje šifriranega sporočila potrebuje le prejemnikov javni ključ ter parametre prejemnikove certifikatne agencije. Prejemnik bo lahko to sporočilo odšifriral le v primeru, ko je njegov certifikat še veljaven.

3.2 Šifriranje na osnovi identitete

Shamir [6] je leta 1984 postavil temelje šifriranja na osnovi identitete oziroma niza, ki določenega uporabnika identificira **IBE** (ang. Identity-Based Encryption). Kasneje so na njegovih temeljih odkrili bolj uporabne sheme za šifriranje na osnovi identitete. V tem poglavju bodo opisani osnovni algoritmi takšne sheme ter njihova uporaba.

Osnovna motivacija za IBE shemo je bila pomoč razvoja infrastrukture javnih ključev. IBE shema uporablja center zaupanja oz. generator skritih ključev, kateremu morajo uporabniki brezpogojno zaupati, imenovan **PKG** (ang. Private Key Generator). Na začetku ta generator generira glavni skriti ključ s in objavi nekaj parametrov, označimo jih s `param`, ki vključujejo javni ključ, povezan z glavnim skritim ključem s. Vsak uporabnik generatorja skritih ključev ima neko šifro (`ID`), ki je lahko njegov elektronski naslov ali kateri drug niz, ki uporabnika identificira. V primeru, ko na koncu unikatnega niza dodamo še trenutno leto, dobimo par ključev z veljavnostjo tisto leto. V primeru, ko ključe izdaja neko podjetje za svoje zaposlene, lahko k nizu dodamo trenutni datum. S tem imamo dnevno vzdrževane ključe. V primeru, ko Bojan zapusti podjetje, mu le to preneha izdajati dnevne skrite ključe in Bojan ne more več dešifrirati sporočil. Takšen pristop omogoča tudi pošiljanje sporočil vnaprej, saj jih bo lahko Bojan odšifriral le na dan, ki je bil vključen v niz za javni ključ in to le v primeru, ko bo od generatorja skritih ključev prejel ustrezni ključ. IBE shema je sestavljena iz štirih osnovnih algoritmov:

Nastavitve: Vzame varnostni parameter k in vrne `param` (sistemski parametri) ter glavni skriti ključ s. Sistemski parametri vsebujejo podatke o prostoru sporočil $\mathcal{M}$ in prostoru tajnopisov $\mathcal{C}$. Ti parametri so javni, medtem ko je skriti ključ s znan le generatorju skritih ključev.

Generiranje ključa: Za vhodne parametre vzame `param`, s in poljuben niz $\texttt{ID} \in \{0,1\}^*$ in vrne skriti ključ d. `ID` se uporablja kot javni ključ, medtem ko je d ustrezen skriti ključ.

Šifriranje: Za vhodne parametre vzame `ID`, `param` ter $M \in \mathcal{M}$ in vrne tajnopis $C \in \mathcal{C}$.

Odšifriranje: Za vhodne parametre vzame `param`, `ID`, C ter skriti ključ d in vrne sporočilo M.

Ti algoritmi morajo zadoščati osnovnim pogojem šifriranja, kajti če je d skriti ključ, zge-

neriran iz parametra $\texttt{ID}$, potem mora za vsak $M \in \mathcal{M}$ veljati: $\mathsf{Dec}(\texttt{param}, \texttt{ID}, C, d) = M$, kjer je $C = \mathsf{Enc}(\texttt{param}, \texttt{ID}, M)$, Dec odšifrirni algoritem in Enc šifrirni. Če je javni ključ uporabnika zgrajen po javno znanem postopku iz javno znanega niza, potem ga pošiljatelju ni potrebno zahtevati od uporabnika oz. direktorija, ampak ga lahko sam izračuna. Največja slabost takega sistema je, da lahko uporabnik odšifrira sporočila le takrat, ko od generatorja skritih ključev prejme osebni skriti ključ. Tu sta še dve slabosti, in sicer generator skritih ključev lahko z lahkoto odšifrira sporočila svojih uporabnikov, saj ima vse njihove skrite ključe. Poleg tega pa mora iti prenos uporabnikovega skritega ključa po zavarovanem kanalu, kar precej oteži samo distribucijo ključa.

Varnost pred napadi na izbrani tajnopis

Nerazločnost za (prilagodljive) napade na izbrani tajnopis, z okrajšavo **IND-CCA** (ang. Indistinguishability against (adaptive) chosen ciphertext attack) je standard za dokazovanje varnosti shem za šifriranje. Glede na to je potrebno pokazati, da tudi šifriranje na osnovi identitete ustreza temu pojmu varnosti. Vendar pa bomo pojem IND-CCA malce razširili, saj je lahko napadalec na javni ključ (niz) $\texttt{ID}$ v shemi za šifriranje na osnovi identitete pred tem že pridobil skrite ključe, ki pripadajo ključem $\texttt{ID}_1$, $\texttt{ID}_2$, ..., $\texttt{ID}_n$. Sistem naj bi ostal varen tudi pred takim napadom. Definicija varnosti pred napadi na izbrani tajnopis mora torej omogočati napadalcu pridobitev kateregakoli skritega ključa, ki pripada javnemu $\texttt{ID}_i$, razen tistega, nad katerim trenutni napad poteka. Pravimo, da je shema varna, če noben napadalec $\mathcal{A}$ s polinomsko časovno zahtevnostjo nima nezamerljivo majhne prednosti pred tekmecem (v našem primeru je tekmec kar generator skritih ključev) v naslednji igri:

Nastavitve: Generator skritih ključev vzame varnostni parameter k in dobi sistemske parametre $\texttt{param}$ ter glavni skriti ključ s. Parametre pošlje napadalcu, medtem ko s obdrži varno skrit.

1. faza: Napadalec sestavi zaporedje poizvedb $q_1, q_2, ..., q_m$, kjer je q_i ena izmed:

- Poizvedba za pridobivanje skritega ključa, ki pripada javnemu $\texttt{ID}_i$. Generator skritih ključev na to poizvedbo odgovori tako, da napadalcu vrne skriti ključ d_i.
- Poizvedba za odšifriranje, pri kateri generator skritih ključev sprejme parametra $\texttt{ID}_i$ in C_i, zgenerira pripadajoči skriti ključ d_i ter pošlje napadalcu dobljeni čistopis.

Katero poizvedbo napadalec pošlje na i-tem koraku je odvisno od odgovorov na poizvedbe $q_1, q_2, ..., q_{i-1}$.

Napad: Ko se napadalec odloči, da je 1. faza končana, poda dve sporočili $M_0, M_1 \in \mathcal{M}$ in identifikacijski niz $\texttt{ID}$. Edina omejitev pri tej izbiri je, da se $\texttt{ID}$ ni pojavil v eni izmed poizvedb v 1. fazi. Generator skritih ključev si izbere bit $b \in \{0, 1\}$ in izračuna $C = \mathsf{Enc}(\texttt{param}, \texttt{ID}, M_b)$. Dobljeni tajnopis nato pošlje napadalcu.

2. faza: Napadalec izda še dodatne poizvedbe $q_{m+1}, ..., q_n$, na katere generator skritih ključev odgovarja kot v 1. fazi.

Ugibanje: Na koncu napadalec izbere bit $b' \in \{0, 1\}$. Napadalec zmaga, če je $b' = b$. Prednost napadalca $\mathcal{A}$ v napadu na shemo definiramo kot $\mathsf{Adv}(A) = P(b = b') - 1/2$.

Bilinearne preslikave in BDH

Definicija 3.1. Naj bosta $\mathbb{G}_1$ in $\mathbb{G}_2$ ciklični grupi reda q, kjer je q neko veliko praštevilo. Preslikava $\hat{e}$ je *bilinearna*, če $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ za vsak $P, Q \in \mathbb{G}_1$ in $a, b \in \mathbb{Z}$.

V nadaljevanju opisana IBE shema bo uporabljala preslikavo $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$, katera ustreza naslednjim lastnostim:

1. $\hat{e}$ je bilinearna.
2. $\hat{e}$ je nedegenerirana, torej ne preslika vsakega para iz $\mathbb{G}_1 \times \mathbb{G}_1$ v identiteto grupe $\mathbb{G}_2$. Ker sta $\mathbb{G}_1$ in $\mathbb{G}_2$ praštevilskega reda, iz tega sledi naslednje: če je P generator grupe $\mathbb{G}_1$, potem je $\hat{e}(P, P)$ generator grupe $\mathbb{G}_2$.
3. Obstaja učinkovit algoritem za izračunanje $\hat{e}(P, Q)$ za vsak $P, Q \in \mathbb{G}_1$.

Preslikavi, ki ustreza tem trem predpostavkam, pravimo *dopustna* bilinearna preslikava.

MOV posplošitev: Menezes, Okamoto in Vanstone [16] so pokazali, da problem diskretnega logaritma v $\mathbb{G}_1$ ni nič težji kot v $\mathbb{G}_2$. Naj bosta $P, Q \in \mathbb{G}_1$ primer problema diskretnega logaritma v $\mathbb{G}_1$, kjer sta P, Q praštevilskega reda q. Radi bi našli tak $\alpha \in \mathbb{Z}_q$, da je $Q = \alpha P$. Naj bo $g = \hat{e}(P, P)$ in $h = \hat{e}(Q, P)$. Potem je zaradi bilinearnosti $h = g^\alpha$ in zaradi nedegeneriranosti imata g in h red q v $\mathbb{G}_2$. Problem diskretnega logaritma v $\mathbb{G}_1$ smo torej posplošili na problem diskretnega logaritma v $\mathbb{G}_2$. Če torej želimo, da je

problem diskretnega logaritma težek v $\mathbb{G}_1$, moramo izbrati varnostni parameter tako, da je ta problem težek v $\mathbb{G}_2$.

Odločitveni Diffie-Hellmanov problem je lahek: Odločitveni Diffie-Hellmanov problem **DDH** (ang. Decision Diffie-Hellman problem) v $\mathbb{G}_1$ je razlikovanje med distribucijama $\langle P, aP, bP, abP\rangle$ in $\langle P, aP, bP, cP\rangle$, kjer so a, b, c naključna števila iz $\mathbb{Z}_q^*$ in P naključen iz $\mathbb{G}_1^*$. Joux in Nguyen [17] sta izpostavila, da je DDH lahek v $\mathbb{G}_1$. Da to vidimo, vzamemo $P, aP, bP, cP \in \mathbb{G}_1^*$. Velja:

$$c \equiv ab(\bmod q) \Leftrightarrow \hat{e}(P, cP) = \hat{e}(aP, bP).$$

Diffie-Hellmanov problem izračunljivosti **CDH** (ang. Computational Diffie-Hellman problem) je lahko kljub temu težek v $\mathbb{G}_1$. Pri CDH-ju gre za iskanje abP, kjer imamo podano naključno trojico $\langle P, aP, bP\rangle$. V [17] imamo tudi primere grup in preslikav, za katere naj bi bil CDH težek, čeprav je DDH lahek.

Diffie-Hellmanova bilinearnostna predpostavka

Ker je DDH lahek v $\mathbb{G}_1$, ga ne moremo uporabiti za kriptosisteme v $\mathbb{G}_1$. Namesto tega vzamemo varianto CDH-ja, katero bomo imenovali Diffie-Hellmanova bilinearnostna predpostavka **BDH** (ang. Bilinear Diffie-Hellman Assumption).

Diffie-Hellmanov bilinearni problem: Naj bo q praštevilo in naj bosta grupi $\mathbb{G}_1$, $\mathbb{G}_2$ reda q. Naj bo $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$ dopustna bilinearna preslikava in P generator grupe $\mathbb{G}_1$. BDH problem v $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$ je definiran kot:
Za dane $\langle P, aP, bP, cP\rangle$, kjer so $a, b, c \in \mathbb{Z}$ izračunaj $W = \hat{e}(P, P)^{abc} \in \mathbb{G}_2$. Algoritem $\mathcal{A}$ ima prednost ε pri reševanju BDH-ja v $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$, če velja

$$P(\mathcal{A}(P, aP, bP, cP) = \hat{e}(P, P)^{abc}) \geq \varepsilon.$$

Verjetnost računamo glede na naključnost a, b, c, P ter bitov $\mathcal{A}$.

BDH generator parametrov: Naključnemu algoritmu pravimo BDH generator parametrov, če zanj veljajo naslednje stvari:

(1) Uporabi varnostni parameter $k \in \mathbb{Z}^+$.

(2) Ima polinomsko časovno zahtevnost v k.

(3) Vrne praštevilo q, opis grup $\mathbb{G}_1$ in $\mathbb{G}_2$ reda q ter opis dopustne bilinearne preslikave $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \mapsto \mathbb{G}_2$.

BDH generator parametrov označimo z $\mathcal{G}$. Njegovo delovanje označimo z $\mathcal{G}(1^k) = \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$. Varnostni parameter k se uporablja za določanje velikosti q, za q lahko na primer vzamemo k-bitno praštevilo. Opis grup $\mathbb{G}_1$ in $\mathbb{G}_2$ vsebuje algoritme polinomske časovne zahtevnosti za računanje operacij v grupah, kot tudi generatorja obeh grup. Podobno opis preslikave $\hat{e}$ vsebuje algoritem polinomske časovne zahtevnosti za računanje vrednosti $\hat{e}$.

BDH predpostavka: Naj bo $\mathcal{G}$ BDH generator parametrov. Pravimo, da ima algoritem $\mathcal{A}$ prednost $\varepsilon(k)$ pri reševanju BDH problema za $\mathcal{G}$, če za dovolj velik k velja:

$$\mathsf{Adv}_{\mathcal{G},\mathcal{A}}(k) = P[\mathcal{A}(q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, aP, bP, cP) = \hat{e}(P,P)^{abc} \mid$$

$$\mid \langle q, \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle \in \mathcal{G}(1^k),\ P \in \mathbb{G}_1^*,\ a, b, c \in \mathbb{Z}_q^*] \geq \varepsilon(k).$$

Pravimo, da $\mathcal{G}$ zadošča BDH predpostavki, če je za kak algoritem $\mathcal{A}$ polinomske časovne zahtevnosti prednost $\mathsf{Adv}_{\mathcal{G},\mathcal{A}(k)}$ nezamerljivo majhna. Če $\mathcal{G}$ zadošča BDH predpostavki, potem pravimo, da je BDH problem težek v grupah, generiranih z $\mathcal{G}$.

Zanimivo je pogledati povezavo med BDH problemom in drugimi težkimi problemi v kriptografiji. Zaenkrat lahko rečemo le, da BDH problem v $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$ ni nič težji od CDH problema v $\mathbb{G}_1$ ali $\mathbb{G}_2$. Z drugimi besedami, algoritem za rešitev CDH problema v $\mathbb{G}_1$ ali $\mathbb{G}_2$ je zadosten tudi za rešitev BDH problema v $\langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}\rangle$. Ali velja tudi obratno, je zaenkrat še odprt problem.

V nadaljevanju tega diplomskega dela bomo spoznali pojem sheme za šifriranje s certifikati, ki je nekakšna kombinacija ravnokar opisane IBE sheme ter šifriranja z javnimi ključi **PKE** (ang. Public Key Encryption). Združevala naj bi dobre lastnosti obeh shem in hkrati odpravila njune slabosti. Kot pri šifriranju z javnimi ključi si vsak uporabnik zgenerira svoj par ključev (javni in skriti) in zahteva certifikat od certifikatne agencije. Ta uporabi IBE shemo za generiranje certifikata, ki ima enake lastnosti kot certifikat v infrastrukturi javnih ključev. Na ta način se znebimo poizvedb tretjih oseb, znebimo pa se tudi potrebe po zavarovanem kanalu med certifikatno agencijo in uporabnikom, saj lahko agencija pošlje uporabniku certifikat po javnem kanalu. S tem, ko si uporabnik sam zgenerira tudi skriti ključ in ta ni znan certifikatni agenciji odpravimo tudi možnost

certifikatne agencije za odšifriranje uporabnikovih sporočil. CBE shema postane neuporabna za agencije z zelo velikim številom uporabnikov, recimo 250 milijonov, saj potrebuje ogromno računsko moč, če želi pogosto posodabljati certifikate oz. njihovo veljavnost.

3.3 Opis modela

V tem poglavju bomo spoznali formalno definicijo modela CBE sheme. V Poglavju 5 si bomo nato ogledali CBE shemo z uporabo Weilovega parjenja, v Poglavju 6 pa bomo videli, kako lahko z uporabo pokritja podmnožic računsko zahtevnost občutno zmanjšamo in naredimo CBE shemo uporabnejšo v raznih aplikacijah.

Glavna akterja v CBE modelu sta overitelj (certifikatna agencija) in uporabnik. Opisani model ne uporablja oz. ne potrebuje zavarovanega kanala med njima.

Definicija 3.2. Shema za posodabljanje certifikatov v CBE je šesterica algoritmov (Gen_{IBE},Gen_{PKE},Upd1,Upd2,Enc,Dec), za katero velja:

1) Gen_{IBE} je probabilistični algoritem IBE z varnostnim parametrom 1^{k_1} in lahko tudi s skupnim številom časovnih intervalov t. Algoritem vrne $\mathrm{SK}_{\mathrm{IBE}}$ (overiteljev glavni skriti ključ) in javne parametre, ki jih označimo s `param`. Med parametri sta med drugim tudi javni ključ $\mathrm{PK}_{\mathrm{IBE}}$ in opis prostora nizov S.

2) Gen_{PKE} je probabilistični algoritem PKE z varnostnim parametrom 1^{k_2} in lahko tudi s skupnim številom časovnih intervalov t. Algoritem vrne $\mathrm{SK}_{\mathrm{PBE}}$ in $\mathrm{PK}_{\mathrm{PBE}}$ (uporabnikova skriti in javni ključ).

3) Deterministični algoritem za posodabljanje certifikata pri izdajatelju Upd1 na začetku i-tega časovnega intervala iz parametrov $\mathrm{SK}_{\mathrm{IBE}}$, `param`, i, niz $s \in S$ in $\mathrm{PK}_{\mathrm{PBE}}$ sestavi Cert'_i in ga pošlje uporabniku.

4) Deterministični algoritem za posodabljanje certifikata pri uporabniku Upd2 na začetku i-tega časovnega intervala iz parametrov `param`, i, Cert'_i in lahko tudi Cert_{i-1} sestavi Cert_i.

5) Probabilistični algoritem Enc za šifriranje iz parametrov `param`, i, s, $\mathrm{PK}_{\mathrm{PBE}}$ in sporočila M vrne tajnopis C, ki je namenjen uporabniku s Cert_i in $\mathrm{SK}_{\mathrm{PBE}}$.

6) Deterministični algoritem Dec za odšifriranje iz parametrov $\mathtt{param}$, Cert_i, $\mathrm{SK}_{\mathrm{PBE}}$ in C vrne sporočilo M, ali pa posebni simbol $\perp$. S tem simbolom označimo, da je prišlo do napake.

Morda se nam bo zdelo nenavadno, da lahko certifikat uporabljamo kot odšifrirni ključ, vendar lahko iz vsake IBE sheme dobimo PKE na naslednji način: za podpisnikov skriti ključ vzamemo glavni skriti ključ v IBE shemi. Podpisnik podpiše sporočilo M tako, da izračuna odšifrirni ključ d za $\mathtt{ID} = M$. Prejemnik sporočila lahko torej enostavno preveri, ali je podpis veljaven. Izbere si neko sporočilo M', ga zašifrira s ključem $\mathtt{ID} = M$ in dobljeni tajnopis odšifrira z d. Če se dobljeno sporočilo ujema z M', potem je podpis veljaven. Torej je odšifrirni ključ iz IBE sheme lahko tudi podpis oz. certifikat. Ta ključ oz. certifikat lahko uporabljamo za omogočanje implicitnega potrjevanja, opisanega v uvodu, kot tudi podpis z eksplicitnim potrjevanjem veljavnosti.

Iz podane definicije mora biti razvidno, da potrebuje uporabnik svoj javni ključ kot tudi certifikat oz. skriti ključ, da lahko odšifrira sporočila. Niz s lahko vsebuje tudi sporočilo, podpisano s strani overitelja, kot npr. uporabnikovo ime, javni ključ in podobno, ni mu pa treba vsebovati statusa certifikata, saj ga pošiljatelju ni potrebno poznati.

3.4 Varnost sistema

Največjo nevarnost našega sistema predstavljata napada s strani uporabnika z neveljavnim certifikatom in pa s strani overitelja. Oba imata namreč 'polovico' informacije, ki je potrebna za odšifriranje. V ta namen definiramo IND-CCA kot dve različni igri. Na grobo lahko ocenimo, da je CBE shema varna, če napadalec ne more zmagati v nobeni izmed iger.

Igra 1: V tej igri napadalec prevzame vlogo uporabnika z neveljavnim certifikatom. Overitelj zažene $\mathsf{Gen}_{IBE}(1^{k_1}, t)$ in pošlje dobljene parametre $\mathtt{param}$ napadalcu, ki potem pomeša poizvedbe za potrjevanje in odšifriranje z eno samo lažno poizvedbo. Overitelj na te poizvedbe odgovori na naslednji način:

- Pri poizvedbi za potrjevanje (i, s, $\mathrm{PK}_{\mathrm{PBE}}$, $\mathrm{SK}_{\mathrm{PBE}}$) preveri, ali je $s \in S$ in da je $\mathrm{SK}_{\mathrm{PBE}}$ skriti ključ, ki pripada javnemu $\mathrm{PK}_{\mathrm{PBE}}$. V primeru, ko sta pogoja izpolnjena, izvede $\mathsf{Upd1}$ ter vrne Cert'_i, sicer vrne $\perp$.

- Pri poizvedbi za odšifriranje (i, s, $\mathrm{PK_{PBE}}$, $\mathrm{SK_{PBE}}$, C) preveri, ali je $s \in S$ in da je $\mathrm{SK_{PBE}}$ skriti ključ, ki pripada javnemu $\mathrm{PK_{PBE}}$. V primeru, ko sta pogoja izpolnjena, zgenerira Cert_i in vrne $\mathsf{Dec}_{\mathsf{Cert}_i,\mathrm{SK_{PBE}},s}(C)$, sicer vrne $\perp$.

- Pri napadalni poizvedbi (i', s', $\mathrm{PK'_{PBE}}$, $\mathrm{SK'_{PBE}}$, M_0, M_1) preveri, ali je $s' \in S$ in da je $\mathrm{SK'_{PBE}}$ skriti ključ, ki pripada javnemu $\mathrm{PK'_{PBE}}$. V primeru, ko sta pogoja izpolnjena, izbere naključni bit b in vrne $C' = \mathsf{Enc}_{i',s',\mathrm{PK_{IBE}},\mathrm{PK'_{PBE}}}(M_b)$, sicer vrne $\perp$.

Na koncu napadalec izbere $b' \in \{0,1\}$. Zmaga je njegova, če $b' = b$, (i', s', $\mathrm{PK'_{PBE}}$, $\mathrm{SK'_{PBE}}$, C') niso bili vključeni v nobeno veljavno poizvedbo za odšifriranje in če (i', s', $\mathrm{PK'_{PBE}}$) niso bili vključeni v nobeno veljavno poizvedbo za potrjevanje po napadu. Prednost napadalca definiramo kot absolutno vrednost razlike 1/2 in verjetnosti, da je zmaga njegova.

Igra 2: Overitelj zažene $\mathsf{Gen}_{PKE}(1^{k_2}, t)$ in da $\mathrm{PK_{PBE}}$ napadalcu, ki potem pomeša poizvedbe za odšifriranje z eno samo lažno poizvedbo. Izzivalec na te poizvedbe odgovori na naslednji način:

- Pri poizvedbi za odšifriranje (i, s, $\mathtt{param}$, $\mathrm{SK_{IBE}}$, C) preveri, ali je $s \in S$ in da je $\mathrm{SK_{IBE}}$ skriti ključ, ki pripada $\mathtt{param}$. V primeru, ko sta pogoja izpolnjena, zgenerira Cert_i in vrne $\mathsf{Dec}_{\mathsf{Cert}_i,\mathrm{SK_{PBE}},s}(C)$, sicer vrne $\perp$.

- Pri napadalni poizvedbi (i', s', $\mathtt{param}'$, $\mathrm{SK'_{IBE}}$, M_0, M_1) preveri, ali je $s' \in S$ in da je $\mathrm{SK'_{IBE}}$ skriti ključ, ki pripada $\mathtt{param}'$. V primeru, ko sta pogoja izpolnjena, izbere naključni bit b in vrne $C' = \mathsf{Enc}_{i',s',\mathrm{PK'_{IBE}},\mathrm{PK_{PBE}}}(M_b)$, sicer vrne $\perp$.

Na koncu napadalec izbere $b' \in \{0,1\}$. Zmaga je njegova, če $b' = b$ in (i', s', $\mathtt{param}'$, C') niso bili vključeni v nobeno veljavno poizvedbo za odšifriranje po napadu. Prednost napadalca definiramo kot zgoraj.

Definicija 3.3. CBE shema za posodabljanje certifikatov je *varna za prilagodljive napade z izbranim tajnopisom (IND-CBE-CCA)*, če noben napadalec nima nezamerljivo majhne prednosti v igrah 1 ali 2.

Opomba 3.1. *Od napadalca zahtevamo, da razkrije svoj skriti ključ, saj sicer izzivalec v splošnem ni sposoben odšifrirati. Z omejitvami na PKE in IBE shemi je možno to zahtevo odstraniti.*

Opomba 3.2. *Kasneje bomo sestavili takšne CBE sheme, ki uporabljajo parjenja, so varne za napade na izbrani tajnopis ter ne zahtevajo od napadalca, da razkrije svoj skriti ključ.*

Poglavje 4

Weilovo parjenje

Zadnje čase se v kriptografiji vedno bolj uporabljajo kriptosistemi z eliptičnimi krivuljami. Največja slabost kriptografije z javnimi ključi v primerjavi s kriptografijo s simetričnimi ključi je velikost samih ključev za zagotavljanje varnosti. V kriptosistemih z eliptičnimi krivuljami lahko velikost ključa v primerjavi z ostalimi kriptosistemi z javnimi ključi pomanjšamo in še vedno zagotovimo enakovredno stopnjo varnosti. Sheme za šifriranje na osnovi identitete uporabljajo posebno bilinearno preslikavo, imenovano parjenje. Najbolj pogosti takšni preslikavi sta Tateovo in Weilovo parjenje. V tem poglavju bomo spoznali slednjega.

To poglavje je razdeljeno na tri pomembnejše dele. V prvem delu se bomo posvetili osnovnim pojmom grupe na eliptični krivulji. Podrobneje se v eliptične krivulje ne bomo spuščali, saj to ni potrebno za razumevanje tega diplomskega dela. V drugem delu si bomo ogledali teorijo deliteljev, saj so v tesni povezavi z Weilovim parjenjem, katerega definicijo podamo v razdelku 4.3. Sledila bo še alternativna definicija, ki je primernejša za uporabo v raznih aplikacijah. Z uporabo Weilovega parjenja bomo dobili varnejše in učinkovitejše sheme, poleg tega pa bomo dobili bolj jasen pogled v njihovo uporabo ter implementacijo.

4.1 Eliptične krivulje

V tem poglavju bodo podane osnovne definicije in rezultati iz teorije eliptičnih krivulj, ki so pomembni za razumevanje snovi v nadaljevanju. Večina dokazov je izpuščena, saj

bi z navajanjem vseh dokazov to delo preseglo svoje okvire in si jih lahko zainteresiran bralec ogleda v literaturi, ki bo podana.
Naj bo K v nadaljevanju algebraično zaprt obseg, torej neskončen.

Definicija 4.1. Naj bo C krivulja v afini ravnini. Točka $P = (a, b)$ na krivulji je *singularna*, če je

$$\frac{\partial C}{\partial x} = \frac{\partial C}{\partial y} = 0.$$

Krivulja z vsaj eno singularno točko je *singularna*.

V geometriji nesingularnost točke na krivulji pomeni, da ima krivulja v tej točki le eno tangento.

Definicija 4.2. *Afina Weierstrassova* enačba nad K je podana z

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

kjer so koeficienti $a_1, a_2, a_3, a_4, a_6 \in K$.
Nesingularna Weierstrassova enačba predstavlja *eliptično krivuljo*.

Definicija 4.3. Naj bodo $a_1, a_2, a_3, a_4, a_6 \in K$ koeficienti iz afine Weierstrassove enačbe. Definiramo še

$$\begin{aligned}
b_2 &= a_1^2 + 4a_2 \\
b_4 &= 2a_4 + a_1a_3 \\
b_6 &= a_3^2 + 4a_6 \\
b_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2 \\
c_4 &= b_2^2 - 24b_4 \\
\Delta &= -b_2^2b_8 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 \\
j &= \frac{c_4^3}{\Delta}, \text{ če } \Delta \neq 0
\end{aligned}$$

Količino Δ imenujemo *diskriminanta* krivulje E, vrednost j pa njena *invarianta*.

Definicija 4.4. Krivulji E in E' sta *izomorfni*, če obstaja transformacija spremenljivk:

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} u^2x + r \\ u^3y + u^2sx + t \end{pmatrix} = \begin{pmatrix} u^2 & 0 \\ u^2s & u^3 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} r \\ t \end{pmatrix},$$

kjer je $u \in K^*$ in $r, s, t \in K$.

Transformaciji spremenljivk iz definicije 4.4 pravimo **dopustna** transformacija. Zgled takšne transformacije je **konjugiranje**:

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \overline{x} \\ \overline{y} \end{pmatrix} = \begin{pmatrix} x \\ -y - a_1 x - a_3 \end{pmatrix}.$$

Z dopustnimi transformacijami lahko Weierstrassovo enačbo eliptične krivulje prevedemo na normalno obliko, glej tabelo 4.5.1. Več o tem si lahko bralec pogleda v [3] ali [5].

Normalna oblika	Δ	j
$\mathrm{char}(K) \neq 2, 3$		
$y^2 = x^3 + a_4 x + a_6$	$-16(4a_4^3 + 27a_6^2)$	$1728\dfrac{4a_4^3}{4a_4^3 + 27a_6^2}$
$\mathrm{char}(K) = 3,\ j \neq 0$		
$y^2 = x^3 + a_2 x^2 + a_6$	$-a_2^3 a_6$	$\dfrac{a_2^3}{a_6}$
$\mathrm{char}(K) = 3,\ j = 0$		
$y^2 = x^3 + a_4 x + a_6$	$-a_4^3$	0
$\mathrm{char}(K) = 2,\ j \neq 0$		
$y^2 + xy = x^3 + a_2 x^2 + a_6$	a_6	$\dfrac{1}{a_6}$
$\mathrm{char}(K) = 2,\ j = 0$		
$y^2 + a_3 y = x^3 + a_4 x + a_6$	a_3^4	0

Tabela 4.5.1: Normalne oblike Weierstrasseve enačbe

Definicija 4.5. Točka $P = (a, b)$ je *reda* 2, če je $P = \overline{P}$, torej $y(P) = b = -b - a_1 a - a_3 = \overline{y}(P)$.

Točke reda 2 lahko na krivulji hitro poiščemo:

- $\mathrm{char}(K) \neq 2$: imamo natanko tri točke reda 2, katere imajo za x koordinate ravno ničle polinoma $x^3 + a_2 x^2 + a_4 x + a_6$ in y koordinato enako 0.
- $\mathrm{char}(K) = 2,\ j \neq 0$: natanko ena točka reda 2 s koordinatama $(0, \sqrt{a_6})$.
- $\mathrm{char}(K) = 2,\ j = 0$: ni točk reda 2.

Do sedaj smo o eliptičnih krivuljah govorili v afini ravnini, torej $K \times K$. Vendar pa na tako definirani eliptični krivulji ne moremo definirati operacije, s katero bi točke te eliptične krivulje tvorile grupo. Za to bi potrebovali dejstvo, da vsaka premica seka krivuljo v treh

točkah. Pri tem si pomagamo s projektivno ravnino in projektivnimi koordinatami. Tu vsakemu razredu vzporednih premic dodamo točko v neskončnosti, v kateri se te premice srečajo. Podrobneje si projektivnih eliptičnih krivulj ne bomo ogledali, zainteresiran bralec si lahko več o tej temi prebere v [4], poglavje 3.

Ponovno bomo zapisali definicijo eliptične krivulje, tokrat v poenostavljeni obliki za obsege $\mathbb{Z}_p$. Upoštevali bomo kanonično obliko iz tabele 4.5.1. Na takšno definicijo najbolj pogosto naletimo v kriptografski literaturi.

Definicija 4.6. Naj bo p liho praštevilo. *Eliptična krivulja* E nad obsegom $\mathbb{Z}_p$ je definirana s posplošeno Weierstrassovo enačbo: $y^2 = x^3 + ax + b$, torej množica rešitev kongruence $y^2 \equiv x^3 + ax + b \pmod{p}$, kjer sta $a, b \in \mathbb{Z}_p$ in velja $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$. K točkam, ki zadoščajo podanima kongruencama, dodamo še posebno točko $\boldsymbol{\mathcal{O}}$, katero imenujemo *točka v neskončnosti.*

Z definiranjem operacij med točkami krivulje lahko tako definirano eliptično krivuljo napravimo za Abelovo grupo. Vse definirane operacije bomo izvajali nad $\mathbb{Z}_p$.
Naj bosta $P = (x_1, y_1)$ in $Q = (x_2, y_2)$ točki na E. Seštevanje točk definiramo na naslednji način:

(i) Če $x_1 = x_2$ in $y_1 = -y_2$, potem $P + Q = \mathcal{O}$.

(ii) Če $P \neq \mathcal{O}$ in $Q \neq \mathcal{O}$, potem je $P + Q = (x_3, y_3)$, kjer je

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2, \\ y_3 &= \lambda(x_1 - x_3) - y_1 \\ \text{in} \qquad \lambda &= \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & \text{če } P \neq Q \\ \dfrac{3x_1^2 + a}{2y_1}, & \text{če } P = Q \end{cases} . \end{aligned}$$

(iii) $P + \mathcal{O} = \mathcal{O} + P = P$.

Točka $\mathcal{O}$ je enota v tako definirani Abelovi grupi. Dokaz lastnosti, da je to Abelova grupa z identiteto $\mathcal{O}$ je dolg, a precej naraven, le dokaz asociativnosti je težji. Obstajajo tudi lažji dokazi asociativnosti, a je njihova izpeljava dolga in mukotrpna. Opazimo pa, da je računanje inverzov enostavno, saj je inverz točke (x, y) kar $(x, -y)$.

Definicija 4.7. Naj bo E poljubna eliptična krivulja in m poljubno naravno število. Preslikavo

$$[m] : E \to E$$

$$[m] : P \mapsto mP,$$

imenujemo *množenje z* $[m]$, kjer mP predstavlja $\underbrace{P + P + ... + P}_{m\text{-krat}}$.

Definicija 4.8. Na bo $m \in \mathbb{N}$ in E eliptična krivulja. Pravimo, da je P *točka torzije* m, če $mP = \mathcal{O}$. Vse točke torzije m tvorijo podgrupo grupe E, imenovano *torzijska podgrupa reda* m. Označimo jo z $E[m]$.

4.2 Teorija deliteljev

Po krajšem pregledu eliptičnih krivulj si bomo ogledali še matematično strukturo, imenovano delitelj, s katero si pomagamo pri iskanju ničel in polov racionalnih funkcij.

Naj bo eliptična krivulja E definirana z afino Weierstrassovo enačbo. Spomnimo se, da je K algebraično zaprt kolobar.

Definicija 4.9. *Koordinatni kolobar* krivulje E nad K definiramo kot

$$\boldsymbol{K[E]} = K[x, y]/(E),$$

kjer smo z (E) označili ideal v $K[x, y]$, generiran s krivuljo E. Kolobar $K[E]$ je celostno polje, to je komutativen cel kolobar brez netrivialnih deliteljev niča. S $\boldsymbol{K(E)}$ označimo obseg polinomov iz $K[E]$, njegove elemente pa imenujemo *racionalne funkcije*.

Definicija 4.10. Naj bo P točka na krivulji E. Pravimo, da je funkcija $r \in K(E)$ *regularna* v P, če obstajata funkciji $f, g \in K[E]$, tako da je $r = f/g$ in $g(P) \neq 0$. Kolobar vseh racionalnih funkcij, ki so regularne v P je lokalni kolobar (to je kolobar z enim samim maksimalnim idealom) krivulje E v P. Označimo ga z $\boldsymbol{\mathcal{O}_P(E)}$.

Če funkcija f ni regularna v P, potem pravimo, da ima v točki P pol in pišemo $f(P) = \infty$. Če je $f(P) = 0$, pravimo, da ima f ničlo v točki P. Pri vsaki funkciji $\ell \in K[E]$ lahko y^2 zamenjamo z izrazom $y^2 - r(x, y)$, kjer je $r(x, y)$ afina Weierstrassova enačba eliptične krivulje iz definicije 4.2 in dobimo obliko: $\ell(x, y) = v(x) + yw(x)$, kjer sta $v(x), w(x) \in K[x]$.

Primer 4.1 Naj bo eliptična krivulja E podana z enačbo $y^2 = x^3 - x$ nad obsegom $K = \mathbb{F}_q$ s karakteristiko različno od 2 in 3. Naj bo $P = (1, 0)$ in funkcija $f = (x^2-1)/y \in K(E)$.
Če vzamemo funkcijo f kot kvocient polinomov, npr. $f \in K[x, y]$, potem f ni definirana v točki P. Če pa funkcijo f gledamo kot element $K(E)$, dobimo

$$f = \frac{x^2-1}{y} = \frac{(x^2-1)y}{y^2} = \frac{(x^2-1)y}{x^3-x} = \frac{y}{x}.$$

Torej ima f v točki P ničlo.

△

Izrek 4.1. *Za vsako točko $P \in E$ obstaja racionalna funkcija $u \in K(E)$, za katero velja, da je $u(P) = 0$ in lahko vsako neničelno racionalno funkcijo $s \in \mathcal{O}_P(E)$ zapišemo kot $s = u^d r$, kjer je d celo število, $r \in K(E)$, $r(P) \in \mathcal{O}_P(E)$ in $r(P) \neq 0$.*

Definicija 4.11. Funkcijo u iz izreka 4.1 imenujemo *enakomerni parameter*.

Dokaz: Enote kolobarja $\mathcal{O}_P(E)$ so funkcije, ki so regularne v točki P in so v tej točki različne od 0. Naj bo $s \in \mathcal{O}_P(E)$. Potem je $s = f/g$, kjer sta $f, g \in K[E]$ in $g(P) \neq 0$, torej je g enota. Od tod sledi, da je dovolj pokazati izrek le za polinom f. Če je $f(P) \neq 0$, potem je tudi f enota in je $d = 0$. Predpostavimo torej, da je $f(P) \neq 0$.
Naj bo $P = (a, b)$ točka, ki ni reda 2. Pokazali bomo, da je $u = x - a$ enakomerni parameter. Pišimo $f = v(x) + yw(x)$, kjer sta $v, w \in K[x]$. Iz v, w izpostavimo faktor $x - a$, tako da vsaj en izmed v in w ni več deljiv z njim. Potem lahko zapišemo $f = (x - a)^{d_1}(v_1(x) + yw_1(x))$, kjer je $v_1(x) \neq 0$ ali $w_1(x) \neq 0$. Definirajmo sedaj funkcijo $f_1 = v_1(x) + yw_1(x)$. Če je $f_1(P) \neq 0$, je f_1 enota in je $d = d_1$. Če je $\overline{f_1}(P) \neq 0$, je $\overline{f_1}$ enota in je $f_1 = f_1\ \overline{f_1}\ \overline{f_1}^{-1} = (x - a)^{d_2} f_2\ \overline{f_1}^{-1}$, kjer je $f_2 \in K[x]$ in $f_2 \neq 0$, tako da je $f_2\ \overline{f_1}^{-1}$ enota in $d = d_1 + d_2$.
Predpostavimo sedaj, da je $f_1(P) = \overline{f_1}(P) = 0$. Potem je par $(\alpha, \beta) = (v_1(x), w_1(x))$ rešitev homogenega sistema linearnih enačb

$$\begin{pmatrix} 1 & y(P) \\ 1 & \overline{y}(P) \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = 0.$$

Determinanta te matrike je $(\overline{y} - y)(P) \neq 0$, saj P ni reda 2. To pomeni, da je edina rešitev tega sistema $\alpha = \beta = 0$, kar pa je protislovje s predpostavko, da $v_1(x)$ in $w_1(x)$

nista oba nič.
Dokaz za točke reda 2 bomo izpustili. Dokaz je podoben, uporabimo le še dopustno transformacijo spremenljivk iz definicije 4.4, da dobimo enostavnejše oblike koordinat točke P in enakomernega parametra u. Zainteresiran bralec si lahko popoln dokaz ogleda v [3] na straneh 22-24. ■

Vrednost števila d je neodvisna od funkcije u, imenujemo ga *red* funkcije f v točki P, oznaka $\mathbf{ord_P(f)}$. Če ima funkcija f v točki P ničlo, potem je $\mathrm{ord}_P(f) > 0$ in pravimo, da ima ničla večkratnost $\mathrm{ord}_P(f)$. V tem primeru pišemo $f(P) = 0$. V primeru, ko ima funkcija f v P pol, je $\mathrm{ord}_P(f) < 0$ in ima pol večkratnost $-\mathrm{ord}_P(f)$. V tem primeru pišemo $f(P) = \infty$. Če P ni ne pol ne ničla funkcije f, potem je $\mathrm{ord}_P(f) = 0$. Iz definicije reda funkcije hitro vidimo, da velja $\mathrm{ord}_P(f \cdot g) = \mathrm{ord}_P(f) + \mathrm{ord}_P(g)$.

Definicija 4.12. Naj bo E eliptična krivulja nad K. *Delitelj* D je formalna vsota točk

$$D = \sum_{P \in E} n_P(P),$$

kjer je $n_P \in \mathbb{Z}$ in $n_P = 0$ za vse razen končno mnogo točk $P \in E$ (točke smo zapisali v oklepajih, da je jasno, da gre za vsoto po točkah in ne vsoto točk). *Nosilec* delitelja D je $\mathbf{supp(D)} = \{P \in E \mid n_P \neq 0\}$.

Množica vseh deliteljev, označimo jo z $\mathbf{Div(E)}$, je grupa, kjer je aditivna operacija podana z:

$$\sum_{P \in E} n_P(P) + \sum_{P \in E} m_P(P) = \sum_{P \in E} (n_P + m_P)(P).$$

$\mathrm{Div}(E)$ je prosta Abelova grupa, generirana s točkami krivulje E.

Definicija 4.13. *Stopnja* delitelja D je

$$\mathbf{deg(D)} = \sum_{P \in E} n_P.$$

Množico vseh deliteljev stopnje 0 označimo z $\mathbf{Div^0(E)}$ in je podgrupa grupe $\mathrm{Div}(E)$.

Definicija 4.14. Naj bo $f \in K(E)$. Ker ima funkcija f končno število polov in ničel, lahko definiramo *delitelj funkcije* f kot

$$\mathbf{div(f)} = \sum_{P \in E} \mathrm{ord}_P(f)(P).$$

Izrek 4.2. *Racionalna funkcija ima končno mnogo ničel in polov. To število je enako, če štejemo z večkratnostmi.*

Za dokaz tega izreka bi potrebovali bolj podroben pregled eliptičnih krivulj v projektivni ravnini, kar presega okvire tega diplomskega dela. Bralec si lahko več o tem ogleda v [3], str. 25-32, kjer je tudi dokaz omenjenega izreka.

Primer 4.2 Vzemimo eliptično krivuljo E, definirano z enačbo $y^2 = x^3 + ax + b$ nad končnim obsegom $K = \mathbb{F}_q$, $char(K) \neq 2, 3$.

- Naj bo $P = (c, d) \notin E[2]$, torej ni reda 2. Premica z enačbo $x - c = 0$ ima ničli reda 1 v točkah P in $-P = \overline{P}$ in po izreku 4.2 pol stopnje 2 v točki $\mathcal{O}$. Od tod sledi, da je delitelj $\mathrm{div}(x - c) = (P) + (-P) - 2\mathcal{O}$.
- Naj bosta P_1 in P_2 točki na E, $P_1 \neq -P_2$ in ℓ premica skozi P_1 in P_2. Potem je $\mathrm{div}(\ell) = (P_1) + (P_2) + (-P_3) - 3\mathcal{O}$, kjer je $P_3 = P_1 + P_2$.
- Naj bodo $P_1, P_2, P_3 \in E$ točke reda 2. Kot smo povedali po definiciji 4.5, so to točke na eliptični krivulji, ki imajo y koordinato enako nič. Potem je $\mathrm{div}(y) = (P_1) + (P_2) + (P_3) - 3(\mathcal{O})$.
- Vzemimo $P_4 = (0, \sqrt{b})$, $P_5 = (0, -\sqrt{b})$, $b \neq 0$. Funkcija x/y ima ničle v teh dveh točkah, pole pa v točkah P_1, P_2 in P_3. Imamo torej tri pole reda 1 in dve ničli reda 1. Po izreku 4.2 ima ta funkcija ničlo reda 1 še v točki $\mathcal{O}$. Torej je $\mathrm{div}(x/y) = (P_4) + (P_5) + (\mathcal{O}) - (P_1) - (P_2) - (P_3)$.

△

Definicija 4.15. Delitelj $D \in \mathrm{Div}^0(E)$ je *glavni delitelj*, če obstaja funkcija $f \in K(E)^*$, da je $D = \mathrm{div}(f)$.

Označimo s $\mathbf{Prin(}\boldsymbol{E}\mathbf{)}$ množico vseh glavnih deliteljev. Po izreku 4.2 je $\mathrm{Prin}(E)$ podgrupa grupe $\mathrm{Div}^0(E)$. Faktorsko grupo $\mathrm{Div}(E)/\mathrm{Prin}(E)$ imenujemo *Picardova grupa* ali *grupa razredov deliteljev*, označimo jo z $\mathbf{Pic(}\boldsymbol{E}\mathbf{)}$, faktorsko grupo $\mathrm{Div}^0(E)/\mathrm{Prin}(E)$ pa *delitelji stopnje nič Picardove grupe*, označimo jo z $\mathbf{Pic^0(}\boldsymbol{E}\mathbf{)}$.

Definicija 4.16. Delitelja $D_1, D_2 \in \mathrm{Div}^0(E)$ sta *ekvivalentna*, oznaka $D_1 \sim D_2$, če $D_1 - D_2 \in \mathrm{Prin}(E)$.

Delitelja sta torej ekvivalentna, če $D_1 = D_2 + \mathrm{div}(f)$, kjer je $f \in K(E)$.

Izrek 4.3. *Naj bo $D \in \mathrm{Div}(E)$. Potem obstaja enolično določena točka $Q \in E$, da velja*

$$D \sim (Q) + (\deg(D) - 1)(\mathcal{O}).$$

Za delitelj stopnje 0 torej obstaja enolično določena točka $Q \in E$, da velja $D \sim (Q) - (\mathcal{O})$. Za dokaz izreka bi potrebovali širše poznavanje teorije eliptičnih krivulj v projektivni ravnini, ki ni predmet tega diplomskega dela. Dokaz lahko najdemo v [3] str. 38.
Naj bo $\sigma : \mathrm{Pic}^0(E) \to E$ preslikava, ki delitelju D priredi takšno točko. Potem je σ izomorfizem grup $\mathrm{Div}^0(E)/\mathrm{Prin}(E)$ in E.

Izrek 4.4. *Naj bo $D = \sum n_P(P)$ delitelj. Potem je D glavni delitelj, če in samo če je $\sum n_P = 0$ in $\sum n_P P = \mathcal{O}$.*

Dokaz: Vemo, da ima vsak glavni delitelj D stopnjo nič. Naj bo σ zgoraj omenjeni izomorfizem. Potem velja

$$D \sim (\mathcal{O}) - (\mathcal{O}) \Leftrightarrow \sigma(D) = \mathcal{O} \Leftrightarrow \sum n_P \sigma((P) - (\mathcal{O})) = \mathcal{O}.$$

To pa je ravno željen rezultat, saj velja $\sigma((P) - (\mathcal{O})) = P$. ∎

Naj bo $f \in K(E)$ funkcija in $D = \sum_{P \in E} n_P(P)$ delitelj, za katera velja $\mathrm{supp}(\mathrm{div}(f)) \cap \mathrm{supp}(D) = \emptyset$. Potem vrednost funkcije $f(D)$ določimo kot

$$f(D) = \prod_{P \in \mathrm{supp}(D)} f(P)^{n_P}.$$

Ker imata $\mathrm{div}(f)$ in D disjunktna nosilca, je ta vrednost dobro definirana.

Izrek 4.5. *Naj bo $D \in \mathrm{Div}^0(E)$ delitelj stopnje 0 in $f_1 \in K(E)$ racionalna funkcija, tako da je $\mathrm{supp}(\mathrm{div}(f_1)) \cap \mathrm{supp}(D) = \emptyset$. Naj bo še $c \in K^*$. Potem za racionalno funkcijo $f_2 = cf_1$ velja*

$$f_2(D) = f_1(D).$$

Dokaz: Za čitljivejši opis samo za ta dokaz označimo $\mathcal{D} = \mathrm{supp}(D)$. Opazimo, da je $\mathrm{supp}(\mathrm{div}(f_1)) = \mathrm{supp}(\mathrm{div}(f_2))$, torej ima $\mathrm{div}(f_2)$ disjunktni nosilec z $\mathcal{D}$. Pišimo $D = \sum_{P \in E} n_P(P)$, kjer je $\sum_{P \in E} n_P = \sum_{P \in \mathcal{D}} n_P = 0$. Potem je

$$f_2(D) = \prod_{P \in \mathcal{D}} f_2(P)^{n_P} = \prod_{P \in \mathcal{D}} (cf_1(P))^{n_P} = c^{\sum_{P \in \mathcal{D}} n_P} \prod_{P \in \mathcal{D}} f_1(P)^{n_P} = \prod_{P \in \mathcal{D}} f_1(P)^{n_P} = f_1(D).$$

∎

Izrek 4.6 (Weilov recipročnostni zakon)**.** *Naj bosta $f, g \in K(E)$ taki, da imata njuna delitelja disjunktna nosilca. Potem velja*

$$f(\mathrm{div}(g)) = g(\mathrm{div}(f)).$$

V [20] avtorja omenjata, da žal za ta zakon ne obstaja noben preprost dokaz, bralec pa si lahko v istem delu ogleda dokaz njegove pospološitve.

Iskanje funkcije glavnega delitelja

Spomnimo se, da za vsak delitelj $D \in \mathrm{Div}^0(E)$ stopnje 0 obstaja enolično določena točka $P \in E$, da velja $D \sim (P) - (\mathcal{O})$. D lahko zapišemo kot

$$D = (P) - (\mathcal{O}) + \mathrm{div}(f), \tag{4.1}$$

kjer je funkcija $f \in K(E)$ do konstante enolično določena. Enačbo 4.1 imenujemo *kanonična forma* delitelja D. Pokazali bomo, kako za dan delitelj D stopnje 0 izračunamo točko P in funkcijo f. Najprej pa bomo podali formulo za seštevanje dveh deliteljev v kanonični formi, tako da bo tudi rezultat kanonična forma. Naj bosta $D_1, D_2 \in \mathrm{Div}^0(E)$, podana z enačbama

$$D_1 = (P_1) - (\mathcal{O}) + \mathrm{div}(f_1)$$

in

$$D_2 = (P_2) - (\mathcal{O}) + \mathrm{div}(f_2).$$

Naj bo $P_1 + P_2 = P_3$, $\ell_1 y + \ell_2 x + \ell_3 = 0$ enačba premice ℓ skozi P_1 in P_2 in $x + v_1 = 0$ enačba navpičnice v skozi točko P_3. Če je $P_1 = P_2$, za ℓ vzamemo tangento na P_1 in če je $P_3 = \mathcal{O}$ vzamemo $v = 1$. Potem je kot v primeru 4.2

$$\mathrm{div}(l) = (P_1) + (P_2) + (-P_3) - 3(\mathcal{O})$$

in

$$\mathrm{div}(v) = (P_3) + (-P_3) - 2(\mathcal{O}).$$

Vsoto deliteljev $D_1 + D_2$ zapišemo kot

$$\begin{aligned} D_1 + D_2 &= (P_1) + (P_2) - 2(\mathcal{O}) + \mathrm{div}(f_1 f_2) \\ &= (P_3) - (\mathcal{O}) + \mathrm{div}(l) - \mathrm{div}(v) + \mathrm{div}(f_1 f_2) \\ &= (P_3) - (\mathcal{O}) + \mathrm{div}(f_1 f_2 f_3), \end{aligned}$$

kjer smo definirali $f_3 = l/v$. Opazimo, da je f_3 kot element $K(x, y)$ definirana v vseh točkah, razen P_3 in $-P_3$, saj ima v teh točkah funkcija v ničle.

Vzemimo glavni delitelj $D = \sum_{i=1}^{n} a_i(P_i) \in \mathrm{Prin}(E)$. Zapisali ga bomo kot delitelj funkcije $f \in K(E)$. Ker je $\deg(D) = 0$, lahko zapišemo

$$D = \sum_{i=1}^{n} a_i(P_i) = \sum_{i=1}^{n} a_i((P_i) - (\mathcal{O})).$$

Potrebujemo naslednjo definicijo.

Definicija 4.17. Naj bo $a \in \mathbb{N}$. *Veriga sumandov* za a je zaporedje $1 = d_1, d_2, ..., d_t = a$, da za vsak d_j velja $d_j = d_k + d_l$, kjer je $2 \leq j \leq t$ in $k, l < j$.

Vzemimo sedaj $d_1^{(i)}, d_2^{(i)}, ..., d_{t_i}^{(i)}$ verigo sumandov za a_i. Potem lahko po zgornji metodi za seštevanje deliteljev v kanonični formi delitelj $d_j^{(i)}((P_i) - (\mathcal{O}))$ za $j = 1, ..., t_i$ zapišemo v kanonični formi. Na koncu seštejemo delitelje $(P_i') - (\mathcal{O}) + \mathrm{div}(f_i)$ za $1 \leq i \leq n$, kjer je $(P_i') - (\mathcal{O}) + \mathrm{div}(f_i)$ kanonična forma delitelja $a_i((P_i) - (\mathcal{O}))$.
Menezes [18] pravi, da je f polinomske velikosti, če jo vzdržujemo v faktorski obliki. Ravno tako je tudi računanje funkcije f polinomske časovne zahtevnosti, kot tudi tudi računanje $f(P)$, če je f kot element $K(x, y)$ definirana v točki P. Po tej konstrukcij f ni definirana kvečjemu v točkah $\pm Q_i$, kjer so vmesni delitelji podani kot $D_j = (Q_j) - (\mathcal{O}) + \mathrm{div}(g_j)$.

Za lažje razumevanje si bomo ogledali primer.

Primer 4.3 Vzemimo eliptično krivuljo E/K, podano z enačbo $y^2 = x^3 + 3x$, kjer je $K = \mathbb{F}_{11}$. V spodnji tabeli imamo podane točke na tej krivulji, kot tudi njihove rede.

Točka	Red	Točka	Red
$P_0 = \mathcal{O}$	1	$P_6 = (3,5)$	3
$P_1 = (0,0)$	2	$P_7 = (3,6)$	3
$P_2 = (1,2)$	6	$P_8 = (6,5)$	4
$P_3 = (1,9)$	6	$P_9 = (6,6)$	4
$P_4 = (2,5)$	12	$P_{10} = (7,1)$	12
$P_5 = (2,6)$	12	$P_{11} = (7,10)$	12

Tabela: Točke na krivulji $y^2 = x^3 + 3x$ nad $\mathbb{F}_{11}$

Delitelj $D = 6(P_2) - 6(\mathcal{O})$ je očitno glavni, saj je red točke P_2 enak 6. Veriga sumandov za 6 je 1,2,4,6. Potem racionalno funkcijo f, da bo $D = \mathrm{div}(f)$, izračunamo na naslednji

način:
Zapišemo lahko

$$(P_2) - (\mathcal{O}) \;=\; (P_2) - (\mathcal{O}) + \operatorname{div}(1).$$

Najprej moramo izračunati $2((P_2) - (\mathcal{O}))$. Po definiciji izračunamo $P_2 + P_2 = P_7$. (Pri računanju λ po definiciji seštevanja točk na eliptični krivulji upoštevamo, da je $3^{-1} = 4 \mod 11$.) Potrebujemo enačbi za tangento ℓ na krivuljo v točki P_2 ter navpičnico v skozi točko P_7. Enačbo premice v dobimo takoj, $x = 3$, torej $x - 3 = 0$ oziroma po modulu 11 dobimo enačbo $x + 8 = 0$. Enačbo za ℓ dobimo tako, da izračunamo gradient funkcije $F(x, y) = y^2 - x^3 - 3x = 0$ v točki $P_2 = (1, 2)$ modulo 11. S tem dobimo normalo na krivuljo v točki P_2, ki jo skalarno pomnožimo še z vektorjem $(x - 1, y - 2)$ in enačimo z 0. Imamo torej

$$\operatorname{grad}(F)|_{(1,2)} = (-3x^2 - 3, 2y)|_{(1,2)} = (-6, 4) = (5, 4) \bmod 11.$$

Naprej

$$\begin{aligned}(5, 4) \cdot (x - 1, y - 2) &= 5x - 5 + 4y - 8 \\ &= 5x + 4y - 2 \bmod 11 \;/ \cdot 3 \\ &= 4x + y + 5 \bmod 11.\end{aligned}$$

Torej

$$f_3 = \frac{y + 4x + 5}{x + 8}.$$

Isti postopek uporabimo tudi pri računanju $4((P_2) - (\mathcal{O}))$ in $6((P_2) - (\mathcal{O}))$.

$$\begin{aligned}2(P_2) - 2(\mathcal{O}) &= ((P_2) - (\mathcal{O})) + ((P_2) - (\mathcal{O})) \\ &= (P_7) - (\mathcal{O}) + \operatorname{div}\left(\frac{y + 4x + 5}{x + 8}\right). \\ 4(P_2) - 4(\mathcal{O}) &= (2(P_2) - 2(\mathcal{O})) + (2(P_2) - 2(\mathcal{O})) \\ &= (P_6) - (\mathcal{O}) + \operatorname{div}\left(\frac{(y + 4x + 5)^2}{(x + 8)^2}\frac{(y + 3x + 7)}{(x + 8)}\right). \\ 6(P_2) - 6(\mathcal{O}) &= (2(P_2) - 2(\mathcal{O})) + (4(P_2) - 4(\mathcal{O})) \\ &= \operatorname{div}\left(\frac{(y + 4x + 5)^3}{(x + 8)^3}\frac{(y + 3x + 7)}{(x + 8)}\frac{(x + 8)}{1}\right).\end{aligned}$$

Funkcija f je torej podana z enačbo

$$f = \frac{(y+4x+5)^3(y+3x+7)}{(x+8)^3}.$$

Funkcija f, gledana kot element $K(x,y)$ ni definirana v točkah P_6 in P_7. Vendar pa če f gledamo kot racionalno funkcijo iz $K(E)$, dobimo:

$$\begin{aligned} f &= \frac{(y+4x+5)^3(y+3x+7)}{(x+8)^3} \cdot \frac{(y-4x-5)^3}{(y-4x-5)^3} \\ &= \frac{(y^2+6x^2+4x+8)^3}{(x+8)^3} \cdot \frac{(y+3x+7)}{(y-4x-5)^3} \\ &= \frac{(x^3+6x^2+7x+8)^3}{(x+8)^3} \cdot \frac{(y+3x+7)}{(y-4x-5)^3} \\ &= \frac{(x+8)^3(x+10)^3}{(x+8)^3} \cdot \frac{(y+3x+7)}{(y-4x-5)^3} \\ &= \frac{(x+10)^3(y+3x+7)}{(y-4x-5)^3}. \end{aligned}$$

Torej je f definirana v $P_6 = (3,5)$. S podobno izpeljavo, le da namesto z $(y-4x-5)^3/(y-4x-5)^3$ množimo z $(y-3x-7)/(y-3x-7)$, dobimo, da je f definirana v $P_7 = (3,6)$. Torej je f definirana tako v P_6, kot tudi v P_7.

△

4.3 Definicija

Parjenje je dobilo ime po francoskem matematiku Andréu Weilu[1]. Preden podamo definicijo Weilovega parjenja, si moramo pogledati še pojem racionalne preslikave in indeksa razvejanosti.

Naj bo $k = \mathbb{F}_q$ končen obseg s q elementi in praštevilsko karakteristiko p. Naj bo $K = \overline{k}$

[1] André Weil, 6.5.1906 - 6.8.1998. Rojen v Parizu, kjer je poleg Rima in Göttingena tudi študiral. Po vojni se je preselil v Združene države Amerike, kjer je najprej predaval na Univerzi v Chicagu, kasneje pa se je ustalil na Inštitutu za napredne študije na univerzi Princeton. Mnogo je prispeval k matematiki na več področjih, najbolj pa na povezavah med algebraično ter številsko teorijo. Med največje njegove dosežke štejemo tako imenovane Weilove domneve, Riemannovo hipotezo za obsege funkcij ter Weilovo prezentacijo za razumevanje teorije kvadratičnih form.

njegovo algebraično zaprtje. Naj bo E eliptična krivulja nad k. Kot prej z E označimo grupo točk na krivulji s koordinatami v K. Grupo k-racionalnih točk, torej točk na krivulji E, ki imajo koordinate v k, označimo z E_k. Ker je k končen, je E_k končna Abelova grupa.
Z racionalno preslikavo imamo v mislih preslikavo iz eliptične krivulje E nazaj vase, za razliko od do sedaj obravnavanih funkcij, ki so slikale iz eliptične krivulje E v obseg. Oglejmo si pare racionalnih funkcij $F = (f_1, f_2) \in K(E) \times K(E)$, tako da za točko $P \in E$ leži slika $F(P) := (f_1(P), f_2(P))$ tudi na E. Formalno to pomeni

$$(E \circ F)(P) = E(F(P)) = 0 \quad \forall P \in E.$$

Od tod sledi, da je $E(F) = E \circ F$ ničelna racionalna funkcija in lahko podamo naslednjo definicijo.

Definicija 4.18. *Racionalna preslikava* je točka na eliptični krivulje $E_{K(E)}$.

Če sta $f_1(P)$ in $f_2(P)$ regularni v P, potem je $F(P) = (f_1(P), f_2(P))$, sicer je $F(P) = \mathcal{O}$. Funkciji f_1 in f_2 sta ali obe regularni ali obe neregularni v točki P. Ker so racionalne preslikave definirane kot točke na posebni eliptični krivulji, sestavljajo grupo z operacijo že definiranega seštevanja točk na krivulji.

Izrek 4.7. *Naj bosta F_1 in F_2 racionalni preslikavi. Potem velja*

$$(F_1 + F_2)(P) = F_1(P) + F_2(P) \quad \forall P \in E.$$

Dokaz izreka ni pomemben za glavne rezultate tega diplomskega dela in ga na tem mestu opuščamo. Lahko ga najdemo v [3] na straneh 47-50.

Primer 4.4 Iz trivialnih primerov racionalnih preslikav, kot sta identična preslikava $\mathsf{id} = (x, y)$ in konstantna preslikava $c_Q = (a, b)$ za $Q = (a, b) \in E$ lahko definiramo pomembnejšo racionalno preslikavo, translacijo za Q:

$$\tau_Q : P \mapsto P + Q.$$

Po prejšnjem izreku lahko translacijo zapišemo kot

$$\tau_Q = \mathsf{id} + c_Q,$$

torej je translacija za Q res racionalna preslikava.
Preslikava $[m]$ iz definicije 4.7 je tudi racionalna preslikava, saj je $[m] = [m - i] + \mathsf{id}$

△

Definicija 4.19. Naj bo $F : E \to E$ nekonstantna racionalna preslikava, P točka na E in u enakomerni parameter za $F(P)$. *Indeks razvejanosti* preslikave F v točki P definiramo kot

$$e_F(P) = \mathrm{ord}_P(u \circ F).$$

Glede na definicijo reda točke opazimo, da je $e_F(P)$ neodvisen od izbire parametra u. Ker je u enakomerni parameter funkcije F, ima $u \circ F$ ničlo v točki P in je zato $e_F(P) \geq 1$. V [19] lahko vidimo, da v primeru, ko je F endomorfizem, je $e_F(P)$ konstanta in ga označimo kar z e_F. Na splošno, če je F preslikava množenje z m iz definicije 4.7, je $e_F = 1$.

V nadaljevanju bomo potrebovali še naslednjo definicijo.

Definicija 4.20. Naj bo $F : E \to E$ nekonstantna racionalna preslikava. Homomorfizem $F^* : \mathrm{Div}(E) \to \mathrm{Div}(E)$ definiramo kot:

$$F^*((Q)) = \sum_{F(P)=Q} e_F(P)(P).$$

Preslikavo F^* smo definirali tako, da drži naslednja enakost:

$$\mathrm{div}(r \circ F) = F^*(\mathrm{div}(r)), \tag{4.2}$$

kjer je r neničelna racionalna funkcija. Enakost dokažemo s pomočjo naslednje trditve.

Trditev 4.1. *Naj bo r racionalna funkcija in F nekonstantna racionalna preslikava in $P \in E$. Potem velja*

$$\mathrm{ord}_P(r \circ F) = (\mathrm{ord}_{F(P)}(r))(e_F(P)).$$

Dokaz: Naj bo u enakomerni parameter v $F(P)$. Če je $u = r$, potem je lema ravno definicija indeksa razvejanosti. Pišimo $r = u^d r_1$, kjer je r_1 racionalna funkcija, ki je regularna in neničelna v $F(P)$. Potem je

$$\mathrm{ord}_P(r \circ F) = d\mathrm{ord}_P(u \circ F) + \mathrm{ord}_P(r_1 \circ F) = de_{F(P)}$$

■

Dokažimo še enakost 4.2. Pišimo:

$$\begin{aligned}
\operatorname{div}(r \circ F) &= \sum_{P \in E} \operatorname{ord}_P (r \circ F)(P) \\
&= \sum_{P \in E} \operatorname{ord}_{F(P)}(r) \cdot e_F(P)(P) \\
&= \sum_{Q \in E} \operatorname{ord}_Q(r) \cdot \sum_{F(P)=Q} e_F(P)(P) \\
&= \sum_{Q \in E} \operatorname{ord}_Q r \cdot F^*\big((Q)\big) \\
&= F^* \left(\sum_{Q \in E} \operatorname{ord}_Q(r)(Q) \right) \\
&= F^*(\operatorname{div}(r)).
\end{aligned}$$

Vzemimo število $m \geq 2$, tuje številu $p = char(K)$. Za točko T iz torzijske podgrupe $E[m]$ po izreku 4.4 velja, da je delitelj $m(T) - m(\mathcal{O})$ glavni. Naj bo $f \in K(E)$ takšna funkcija, da velja

$$\operatorname{div}(f) = m(T) - m(\mathcal{O}).$$

Sedaj si oglejmo delitelj $[m]^*(T) - [m]^*(\mathcal{O})$. Po definiciji 4.20 lahko ta delitelj zapišemo kot

$$\sum_{[m]P=T} e_{[m]}(P) \cdot (P) - \sum_{[m]P=\mathcal{O}} e_{[m]}(P) \cdot (P). \tag{4.3}$$

Naj bo T' takšna točka, da je $[m]T' = T$ (takšna točka vedno obstaja, saj je racionalna funkcija po trditvi 3.3 v [3] konstanta ali pa surjektivna). Ker je $e_{[m]} = 1$, lahko enačbo 4.3 zapišemo kot

$$\sum_{R \in E[m]} \left((T' + R) - (R) \right).$$

Ta delitelj ima torej stopnjo 0. Ker je število točk v torzijski podgrupi $E[m]$ enako m^2, je $[m^2]T' = \mathcal{O}$ in dobimo

$$\sum_{R \in E[m]} (T' + R - R) = [m^2]T' = \mathcal{O}.$$

Od tod sledi, da je delitelj $[m]^*(T) - [m]^*(\mathcal{O})$ glavni, torej obstaja funkcija $g_T \in K(E)$, tako da velja

$$\operatorname{div}(g_T) = [m]^*(T) - [m]^*(\mathcal{O}).$$

Definicija 4.21. Pri zgornji notaciji je *Weilovo parjenje* preslikava

$$e_m : E[m] \times E[m] \to \mu_m,$$

podana z

$$e_m(S,T) = \frac{g_T(X+S)}{g(X)}, \tag{4.4}$$

kjer je $X \in E$ poljubna točka, za katero velja $g_T(X+S) \neq 0$, $g_T(X+S) \neq \infty$, $g_T(X) \neq 0$, $g_T(X) \neq \infty$ in μ_m grupa m-tih korenov enote v K.

Pokažimo, da je Weilovo parjenje dobro definirano.

Izrek 4.8. *Weilovo parjenje je dobro definirano, torej slika v m-ti koren enote in je neodvisno od izbire funkcije g_T ali točke X.*

Dokaz: Vzemimo $S,T \in E[m]$, $f \in K(E)$, za katero velja $\mathrm{div}(f) = m(T) - m(\mathcal{O})$. Funkcija je z deliteljem do konstante natančno določena. Opazimo, da velja

$$\mathrm{div}(f \circ [m]) = [m]^*\mathrm{div}(f) = [m]^*(m((T) - (\mathcal{O}))) = m \cdot \mathrm{div}(g_T) = \mathrm{div}(g_T^m).$$

Funkcijo g_T^m lahko torej zapišemo kot $g_T^m = c \cdot (f \circ [m])$ za nek $c \in K^*$. Za vsak $X \in E$ velja

$$g_T(X+S)^m = cf([m]X + [m]S) = cf([m]X) = g_T(X)^m.$$

Od tod direktno sledi

$$e_m(S,T)^m = \frac{g_T(X+S)^m}{g_T(X)^m} = 1.$$

$e_m(S,T)$ je res m-ti koren enote. Dokazali smo tudi, da e_m ni odvisen od izbire funkcije g_T, saj je ta enolično določena do večkratnika konstante.
Dokazati moramo še, da je e_m neodvisno od izbire točke X. Naj bo $\tau_S : E \to E$ translacija za točko S, točki Q torej priredi točko $Q+S$. Potem lahko enačbo 4.4 zapišemo kot

$$e_m(S,T) = \frac{g_T \circ \tau_S}{g_T}. \tag{4.5}$$

Predpostavimo, da velja $\mathrm{div}(g_T \circ \tau_S) = \mathrm{div}(g_T)$ za točki $T,S \in E[m]$. Potem je $(g_T \circ \tau_S)/g_T$ konstanta, zato $e_m(S,T) = g_T(X+T)/g_T(X)$ ni odvisna od izbire točke X.
Dokažimo še predpostavko, da za $S,T \in E[m]$ velja $\mathrm{div}(g_T \circ \tau_S) = \mathrm{div}(g_T)$. Ker je

$S \in E[m]$, je $[m] \circ \tau_S = [m]$. Z uporabo enačbe 4.2 in dejstva, da je $(F_1 \circ F_2)^* = (F_2^* \circ F_1^*)$ dobimo

$$\begin{aligned} \operatorname{div}(g_T \circ \tau_S) &= \tau_S \circ [m]^*((T) - (\mathcal{O})) \\ &= ([m] \circ \tau_S)^*((T) - (\mathcal{O})) \\ &= [m]^*((T) - (\mathcal{O})) \\ &= \operatorname{div}(g_T) \end{aligned}$$

Dokažimo še $(F_1 \circ F_2)^* = (F_2^* \circ F_1^*)$:

$$\begin{aligned} (F_2^* \circ F_1^*)((R)) &= F_2 \left(\sum_{F_1(Q)=R} e_{F_1}(Q)(Q) \right) \\ &= \sum_{F_1(Q)=R} e_{F_1}(Q) \cdot \sum_{F_2(P)=Q} e_{F_2}(P)(P) \\ &= \sum_{(F_1 \circ F_2)(P)=R} e_{F_1}(F_2(P)) \cdot e_{F_2}(P)(P) \\ &= \sum_{(F_1 \circ F_2)(P)=R} e_{F_1 \circ F_2}(P)(P) \\ &= (F_1 \circ F_2)^*((R)). \end{aligned}$$

■

Ogledali si bomo še glavne lastnosti Weilovega parjenja, vendar moramo prej podati še eno lemo, katero potem uporabimo v dokazu.

Lema 4.1. *Naj bo r racionalna funkcija, invariantna glede na translacije za točke grupe $E[m]$. Potem obstaja racionalna funkcija s, tako da velja $r = s \circ [m]$.*

Dokaz te leme uporablja delitvene polinome, katere je obdelal že Barbič v svojem diplomskem delu [5]. Bralec si lahko dokaz ogleda v [3], str. 89 - 91.

Izrek 4.9 (Lastnosti Weilovega parjenja:)**.** *Naj bodo $S_1, S_2, S, T_1, T_2, T \in E[m]$. Za Weilovo parjenje, definirano z enačbo 4.4 veljajo naslednje lastnosti*

1) Bilinearnost:

$$\begin{aligned} e_m(S_1 + S_2, T) &= e_m(S_1, T) e_m(S_2, T) \\ e_m(S, T_1 + T_2) &= e_m(S, T_1) e_m(S, T_2) \end{aligned}$$

2) Identiteta:

$$e_m(S,S) = 1$$

3) Alternacija:

$$e_m(S,T) = e_m(T,S)^{-1}$$

4) Nedegeneriranost:

$$\begin{aligned} e_m(S,T) &= 1 \forall S \in E[m] \Leftrightarrow T = \mathcal{O} \\ e_m(S,T) &= 1 \forall T \in E[m] \Leftrightarrow S = \mathcal{O} \end{aligned}$$

Dokaz: 1) Prvi del trditve lahko dokažemo s precej naravno izpeljavo. Izberimo si točko X, da je g_T definirana v X, $X + S_1$, $X + S_1 + S_2$. Potem je

$$\begin{aligned} e_m(S_1 + S_2, T) &= \frac{g_T(X + S_1 + S_2)}{g_T(X)} \\ &= \frac{g_T(X + S_1 + S_2) g_T(X + S_1)}{g_T(X + S_1) g_T(X)} \\ &= e_m(S_2, T) e_m(S_1, T). \end{aligned}$$

Drugi del je malce bolj kompliciran. Naj bodo funkcije $f_1, f_2, f_3, g_{T_1}, g_{T_2}, g_{T_3}$ kot zgoraj za točki T_1 in T_2 in naj bo $T_3 = T_1 + T_2$. Izberimo si funkcijo $h \in K(E)$ z deliteljem $\mathrm{div}(h) = (T_1 + T_2) - (T_1) - (T_2) + (\mathcal{O})$. Potem velja

$$\mathrm{div}\left(\frac{f_3}{f_1 f_2}\right) = m \cdot \mathrm{div}(h),$$

torej

$$f_3 = c f_1 f_2 h^m \quad \text{za nek } c \in K^*.$$

Vzemimo kompozitum z množenjem z m, $f_i \circ [m] = g_{T_i}^m$, in vzamemo m-te korene enote, da dobimo

$$g_{T_3} = c' g_{T_1} g_{T_2} (h \circ [m]) \quad \text{za nek } c' \in K^*.$$

Sedaj imamo

$$\begin{aligned} e_m(S, T_1 + T_2) &= \frac{g_{T_3}(X+S)}{g_{T_3}(X)} \\ &= \frac{g_{T_1}(X+S)g_{T_2}(X+S)}{g_{T_1}(X)g_{T_2}(X)} \cdot \frac{h([m]X+[m]S)}{h([m]X)} \\ &= e_m(S,T_1)e_m(S,T_2). \end{aligned}$$

2) Za dokaz identitete si oglejmo

$$\operatorname{div}(\prod_{i=0}^{m-1} f \circ \tau_{[i]T}) = \sum_{i=0}^{m-1} \operatorname{div}(f \circ \tau_{[i]T}) = m \sum_{i=0}^{m-1}([1-i]T) - ([-i]T) = 0.$$

Torej je $\prod_{i=0}^{m-1} f \circ \tau_{[i]T}$ konstanta. Izberemo si lahko T', tak da je $[m]T' = T$ in je $\prod_{i=0}^{m-1} g_T \circ \tau_{[i]T'}$ tudi konstanta, saj je njegova m-ta potenca zgornji produkt f-ov. Z oceno produktov funkcije g_T v točkah X in $X + T'$ dobimo

$$\prod_{i=0}^{m-1} g_T(X + [i]T') = \prod_{i=0}^{m-1} g_T(X + [i+1]T').$$

Imamo

$$g_T(X) = g_T(X + [m]T') = g_T(X + T),$$

torej

$$e_m(T, T) = \frac{g_T(X+T)}{g_T(X)} = 1.$$

3) Pišimo

$$1 = e_m(S+T, S+T) = e_m(S,S)e_m(S,T)e_m(T,S)e_m(T,T) = e_m(S,T)e_m(T,S),$$

torej je res $e_m(S,T) = e_m(T,S)^{-1}$.

4) Po lastnosti alternacije je dovolj pokazati le prvo ekvivalenco. Naj bo $e_m(S,T) = 1$ za vsak $S \in E[m]$, torej je $g_T(X+S) = g_T(X)$ za vsak $S \in E[m]$. Po lemi 4.1 sledi, da je $g_T = h \circ [m]$ za neko funkcijo $h \in K(E)$. Potem imamo

$$(h \circ [m])^m = g_T^m = f \circ [m],$$

torej je $f = ch^m$ za neko konstanto $c \in K^*$. Od tod sledi, da

$$m \cdot \operatorname{div}(h) = \operatorname{div}(f) = m(T) - m(\mathcal{O}),$$

torej

$$\mathrm{div}(h) = (T) - (\mathcal{O}).$$

Torej $T = \mathcal{O}$.

■

4.4 Alternativna definicija

Poleg podane definicije 4.21 Weilovega parjenja bomo podali še alternativno definicijo, s katero se lahko potem lažje in hitreje izračuna Weilovo parjenje. Alternativno parjenje bomo označili z e'_m.
Vzemimo točki $S, T \in E[m]$, kjer je m celo število, tuje praštevilu p. Naj bosta A in B delitelja z disjunktnima nosilcema, za katera velja $A \sim (S) - (\mathcal{O})$ in $B \sim (T) - (\mathcal{O})$. Ker sta S in T točki iz m-torzije, hitro sledi, da sta mA in mB glavna delitelja. Obstajata torej funkciji f_A in f_B, da velja

$$\mathrm{div}(f_A) = mA \quad \text{in} \quad \mathrm{div}(f_B) = mB.$$

Definicija 4.22. Z zgornjo notacijo definiramo alternativno Weilovo parjenje

$$e'_m : E[m] \times E[m] \to \mu_m$$

z enačbo

$$e'_m(S, T) = \frac{f_A(B)}{f_B(A)}. \tag{4.6}$$

Izrek 4.10. *Alternativno Weilovo parjenje, definirano z enačbo 4.6, je dobro definirano, torej slika v m-te korene enote in je neodvisno od izbire deliteljev A, B ter funkcij f_A in f_B.*

Dokaz: Z uporabo Weilovega recipročnostnega zakona dobimo

$$\left(\frac{f_A(B)}{f_B(A)}\right)^m = \frac{(f_A(B))^m}{(f_B(A))^m} = \frac{f_A(mB)}{f_B(mA)} = \frac{f_A(\mathrm{div}(f_B))}{f_B(mA)} = \frac{f_B(\mathrm{div}(f_A))}{f_B(mA)} = \frac{f_B(mA)}{f_B(mA)} = 1,$$

s čimer smo pokazali, da je $e'_m(S, T)$ res m-ti koren enote.
Vzemimo sedaj tak delitelj $B' \sim (T) - (\mathcal{O})$, da imata A in B disjunktna nosilca in naj bo

$f_{B'}$ takšna funkcija, da je $\mathrm{div}(f_{B'}) = mB'$. Ker je $B \sim B'$, lahko pišemo $B' = B + \mathrm{div}(h)$ za neko $h \in K(E)$ in zato $f_{B'} = f_B h^m$. Potem velja

$$\frac{f_A(B')}{f_{B'}(A)} = \frac{f_A(B) f_A(\mathrm{div}(h))}{f_B(A) h^m(A)} = \frac{f_A(B) f_A(\mathrm{div}(h))}{f_B(A) h(mA)} = \frac{f_A(B)}{f_B(A)} \frac{f_A(\mathrm{div}(h))}{h(\mathrm{div}(f_A))} = \frac{f_A(B)}{f_B(A)},$$

kjer zadnja enakost velja po Weilovem recipročnostnem zakonu. S tem smo dokazali, da je e'_m neodvisno od izbire točke B. Analogen dokaz velja tudi za točko A.

Funkciji f_A in f_B sta do konstante enolično določeni. Izrek 4.5 pa pove, da je rezultat parjenja neodvisen od izbire teh konstant. ■

Podajmo še povezavo med obema definicijama.

Izrek 4.11. *Vzemimo $S, T \in E[m]$. Potem velja*

$$e_m(S,T) = \frac{1}{e'_m(S,T)}.$$

Dokaz: Najprej opazimo, da je funkcija f, podana pred definicijo 4.21 Weilovega parjenja, ekvivalentna funkciji f_B v alternativni definiciji 4.22. Vemo, da T' zadošča $[m]T' = T$, naj bo še $S' \in E$ taka točka, da $[m]S' = S$. Podobno kot funkcija g_T v definiciji 4.21 obstaja tudi g_A, tako da

$$\mathrm{div}(g_A) = [m]^*(S) - [m]^*(\mathcal{O}) = \sum_{R \in E[m]} \left((S' + R) - (R)\right).$$

Iz tega sledi, da je $g_A^m = f_A \circ [m]$.

Naj bo točka $X \in E$ takšna, da ima delitelj $D = (m-1)(S'+X) + (S'-S+X) - m(X)$ disjunktni nosilec s $\mathrm{supp}(\mathrm{div}(g_T))$.

Očitno je $\deg(D) = 0$ in $[m-1]S' + [m-1]X + S' - S + X - [m]X = \mathcal{O}$, torej je D glavni delitelj. Naj bo $h \in K(E)$, taka da je $\mathrm{div}(h) = D$.

Pišimo:

$$\begin{aligned}
g(\mathrm{div}(h)) &= \frac{g_T(S'+X)^{m-1} \cdot g_T(S'-S+X)}{g_T(X)^m} \\
&= \frac{g_T(S'+X)^m}{g_T(X)^m} \cdot \frac{g_T(S'-S+X)}{g_T(S'+X)} \\
&= \frac{f \circ [m](S'+X)}{f \circ [m](X)} \cdot \frac{g_T(S'')}{g_T(S''+S)} \quad \text{(kjer je } S'' = S'-S+X\text{)} \\
&= \frac{f(S+[m]X)}{f([m]X)} \cdot \frac{1}{e_m(S,T)},
\end{aligned}$$

kjer je

$$e_m(S,T) = \frac{g_T(S''+S)}{g_T(S'')},$$

kot v definiciji 4.21.
Po drugi strani je

$$h(\mathrm{div}(g_T)) = \prod_{R\in E[m]} \frac{h(T'+R)}{h(R)},$$

saj je $\mathrm{div}(g_T) = \sum_{R\in E[m]}(T'+R) - (R)$.
Definirajmo sedaj funkcijo $H \in K(E)$ kot

$$H(P) = \prod_{R\in E[m]} h(P+R) = \prod_{R\in E[m]} h\circ\tau_R(P).$$

Potem imamo

$$\begin{aligned}
\mathrm{div}(H) &= \sum_{R\in E[m]} \mathrm{div}(h\circ\tau_R)\\
&= \sum_{R\in E[m]} ((m-1)(S'+X-R)+(S'-S+X-R)-m(X-R))\\
&= \sum_{R\in E[m]} ((m-1)(S'+X+R)+(S'+X+R)-m(X+R))\\
&= m\sum_{R\in E[m]} (S'+X+R)-(X+R)\\
&= m\cdot\mathrm{div}(g_A\circ\tau_{-X})\\
&= \mathrm{div}(g_A^m\circ\tau_{-X})
\end{aligned}$$

Torej lahko pišemo $H = g_A^m\circ\tau_{-X} = f_A\circ[m]\circ\tau_{-X}$. Od tod sledi

$$\begin{aligned}
\prod_{R\in E[m]} h(T'+R) &= H(T')\\
&= f_A\circ[m]\circ\tau_{-X}(T')\\
&= f_A\circ[m](T'-X)\\
&= f_A(T-[m]X).
\end{aligned}$$

Pišimo še naprej

$$\begin{aligned}\prod_{R\in E[m]} h(R) &= H(\mathcal{O}) \\ &= f_A \circ [m] \circ \tau_{-X}(\mathcal{O}) \\ &= f_A(-[m]X).\end{aligned}$$

Po Weilovem recipročnostnem zakonu vemo, da je

$$h(\mathrm{div}(g_T)) = \prod_{R\in E[m]} \frac{h(T'+R)}{h(R)} = \frac{f_A(T-[m]X)}{f_A(-[m]X)}$$

enako

$$g_T(\mathrm{div}(h)) = \frac{f(S+[m]X)}{f([m]X)} \cdot \frac{1}{e_m(S,T)}.$$

Ker je $f = f_B$, velja

$$e_m(S,T) = \frac{f_A(-[m]X)}{f_A(T-[m]X)} \frac{f_B(S+[m]X)}{f_B([m]X)} = \frac{f_B(A)}{f_A(B)} = \frac{1}{e'_m(S,T)},$$

kjer je $A \sim (S) - (\mathcal{O})$ in $B \sim (T) - (\mathcal{O})$. ■

Ker je $e_m(S,T) = e'_m(T,S)$, potem vse lastnosti izreka 4.9 veljajo tudi za alternativno definicijo. Alternativna definicija je boljša za praktično uporabo, saj za računanje tako definiranega Weilovega parjenja obstaja učinkovit algoritem. Imenuje se Millerjev algoritem, vendar ga v tem diplomskem delu ne bomo obravnavali.

Poglavje 5

CBE shema z Weilovim parjenjem

Shemo za šifriranje s certifikati v poglavju 3 smo v grobem sestavili kot kombinacijo shem za šifriranje na osnovi identitete ter šifriranja z javnimi ključi. Glede na to, da je Boneh-Franklinova [7] IBE shema najbolj praktična, bomo podali definicijo CBE sheme na osnovi omenjene IBE. Boneh-Franklinova shema uporablja Weilovo parjenje, ki smo spoznali v prejšnjem poglavju. Ogledali si bomo dva modela CBE sheme in sicer osnovna CBE shema, katera je v bistvu prej opisan model z uporabo Weilovega parjenja in pa polna CBE shema, ki zaradi povečanja varnosti uporabi še dve dodatni zgoščevalni funkciji. Podani shemi bosta zaradi tega nekaj povečali kompleksnost šifriranja, medtem ko bosta kompleksnost odšifriranja in dolžina tajnopisa ostali enaki.

Kot smo že omenili, se v IBE shemi lahko podpisi uporabljajo kot odšifrirni ključ. V Boneh-Franklinovi shemi se v ta namen uporablja **BLS** podpis [9], imenovan po avtorjih, Bonehu, Lynnu in Shachamu. Osnovna CBE shema bo uporabljala združeni podpis, opisan v [10], zato si jih bomo najprej na kratko ogledali.

5.1 Kratki podpisi

Kratke podpise se uporablja, ko uporabnik sam vnese podpis na dokument, ali pa kjer je mrežna oziroma internetna povezava manj zmogljiva. Če na primer uporabljamo 1024-bitni modul, so RSA podpisi dolgi 1024 bitov, DSA ali ECDSA (DSA na eliptičnih krivuljah) pa 320 bitov, kar je preveč, da bi uporabnik sam vnašal podpis. Ogledali si bomo konkreten primer sheme za kratke podpise, imenovane BLS shema. BLS podpisi so

dolgi 160 bitov in zagotavljajo podobno varnost kot 320-bitni DSA podpis. BLS shema uporablja grupe, v katerih je Diffie-Hellmanov problem izračunljivosti težek, medtem ko je odločitveni Diffie-Hellmanov problem lahek. Takšne grupe bomo imenovali **GDH grupe** (ang. Gap Diffie-Hellman). Nato si bomo na kratko ogledali še združevanje več podpisov v en kratki podpis.

GDH podpisi

Podpis σ poljubnega sporočila $M \in \{0,1\}^*$ je element grupe G, ki je skupaj z generatorjem g sistemski parameter. Z G^* označimo $G \backslash \{1\}$, kjer je 1 enota grupe G. Shema za podpise vsebuje tri osnovne algoritme, KeyGen za generiranje ključev, Sign za generiranje podpisa in Verify za preverjanje podpisa.

KeyGen

```
Input:  g generator grupe G
Output: javni ključ v, skriti ključ x
 1. izberi naključni x ∈ Z_p^*
 2. v ← g^x
 3. Return v, x
```

Sign

```
Input:  skriti ključ x, sporočilo M ∈ {0,1}^*,
        zgoščevalna funkcija h : {0,1}^* ↦ G^*
Output: zgostitev h(M), podpis σ
 1. h ← h(M)
 2. σ ← h^x
 3. Return σ, h
```

Verify

```
Input:  javni ključ v, sporočilo M, podpis σ
Output: boolean
 1. If (g, v, h, σ) veljavna Diffie-Hellmanova četvorka Then
        Return True
    Else
        Return False
```

Opomba 5.1. *Z veljavno Diffie-Hellmanovo četvorko mislimo na četvorko, ki ustreza odločitvenem Diffie-Hellmanovem problemu.*

BLS podpise si bomo ogledali na konkretni GDH grupi, ki izhaja iz eliptične krivulje $E/\mathbb{F}_{3^\ell}$, kjer je ℓ neko praštevilo, definirane z $y^2 = x^3 + 2x \pm 1$. Z m označimo število točk te krivulje.

KeyGen

Input: $\ell \in \{79, 97, 149, 163, 167\}$, q največji praštevilski faktor reda krivulje E, $P \in E/\mathbb{F}_{3^l}$ točka reda q
Output: javni ključ (ℓ, q, P, R), skriti ključ x
1. izberi naključni $x \in \mathbb{Z}_q^*$
2. $R \leftarrow xP$
3. Return $(\ell, q, P, R), x$

Sign

Input: skriti ključ x, sporočilo $M \in \{0,1\}^*$, funkcija $g : \{0,1\}^* \mapsto E$
Output: zgostitev $h(M)$, podpis $\sigma \in \mathbb{F}_{3^\ell}$
1. $P_M \leftarrow g(M)$
2. $S_M \leftarrow xP_M$
3. $\sigma \leftarrow x-$koordinata točke S_M
4. Return σ

Verify

Input: javni ključ (ℓ, q, P, R), sporočilo M, podpis σ, Weilovo parjenje e na krivulji $E/\mathbb{F}_{3^{6l}}$, $\Phi : E \mapsto E$ avtomorfizem krivulje $\ni$ če P reda $q \Rightarrow \Phi(P)$ točka reda q, ki je linearno neodvisna od P
Output: boolean
1. $S \leftarrow$ točka na $E/\mathbb{F}_{3^\ell}$ reda q, ki ima x-koordinato enako σ
2. If S ne obstaja Then
   Return False
3. $u \leftarrow e(P, \Phi(S))$
4. $v \leftarrow e(R, \Phi(h(M)))$
5. If $u = v$ Or $u^{-1} = v$ Then
   Return True
   Else
   Return False

Združevanje podpisov

Iz več podpisov na več različnih sporočilih je možno zgenerirati en sam kratki podpis, ki bo osebo, ki preverja veljavnost podpisa, prepričal, da so omenjena sporočila dejansko podpisali ustrezni uporabniki. Združevanje podpisov nam torej omogoča združevanje verige podpisov v večnivojski infrastrukturi javnih ključev, kjer ima vsak uporabnik verigo certifikatov, katera vsebuje podpise ustreznih certifikatnih agencij. Vzemimo množico uporabnikov $\mathbb{U}$. Vsak uporabnik $u \in \mathbb{U}$ ima svoj par ključev (PK_u, SK_u), PK_u je javni in SK_u skriti. Združevalni algoritem je javni, torej lahko kdorkoli združuje podpise. Algoritem pa je lahko tudi takšne oblike, da lahko popis σ_{uv}, katerega smo dobili iz podpisov σ_u in σ_v združimo s podpisom σ_w in dobimo združeni podpis σ_{uvw}. Podpise lahko torej dodajamo k že združenim podpisom. Seveda pa obstaja tudi algoritem, ki iz javnih ključev $PK_u, PK_v, ..., PK_z$, sporočil $M_u, M_v, ..., M_z$ in združenega podpisa σ preveri, ali je ta podpis veljaven. Za združevanje podpisov bomo uporabili grupi G_1 ter G_2 in bilinearno preslikavo $e : G_1 \times G_1 \mapsto G_2$, kjer je CDH problem težek v G_1.

Radi bi združili podpise uporabnikov iz podmnožice $U \subseteq \mathbb{U}$. Vsak uporabnik $u \in U$ zgenerira svoj podpis σ_u na sporočilu M_u. Te podpise potem nek center za združevanje združi v enotni podpis σ. Centru za združevanje uporabnikom ni potrebno zaupati, saj nima dostopa do kakšnih skritih ključev, temveč le do javnih ključev, sporočil ter podpisov teh sporočil. Dolžina združenega podpisa je enaka dolžini posameznega podpisa sporočila. Shema za združevanje podpisov, imenovana BGLS shema po avtorjih Bonehu, Gentryju, Lynnu ter Shachamu, je sestavljena iz petih algoritmov: KeyGen, Sign, Verify, Aggregate, AggregateVerify. Prvi trije so enaki kot v shemi za kratke podpise, medtem ko sta zadnja dva dodatna algoritma za združevanje in preverjanje združenih podpisov.

KeyGen

Output: javni ključ v, skriti ključ x, g_1 generator grupe G_1
1. izberi naključni $x \in \mathbb{Z}_p^*$
2. $v \leftarrow g_1^x$
3. Return v, x

Sign

Input: skriti ključ x, sporočilo $M \in \{0,1\}^*$,
zgoščevalna funkcija $h : \{0,1\}^* \mapsto G_2$
Output: podpis σ
1. $h \leftarrow h(M)$
2. $\sigma \leftarrow h^x$
3. Return σ

Verify

```
Input:  javni ključ v, sporočilo M, podpis σ,
        bilinearna preslikava e : G1 × G1 ↦ G2
Output: boolean
 1. h ← h(M)
 2. If e(g1, σ) = e(v, h) Then
        Return True
    Else
        Return False
```

Aggregate

```
% Uporabnike iz U ⊆ 𝕌 indeksiramo z indeksi i = 1, 2, ..., k,
% kjer je k = |U|.  Uporabnik u_i podpiše sporočilo M_i in dobi podpis σ_i.
% Sporočila M_i morajo biti različna.

Input:  podpisi σ_i, množica uporabnikov U
Output: združeni podpis σ
 1. k ← |U|, σ ← 1
 2. For i = 1 To k Do
    2.1 σ ← σ · σ_i
    Return σ
```

AggregateVerify

```
Input:  javni ključi v_i, sporočila M_i, združeni podpis σ, množica
        uporabnikov U, bilinearna preslikava e : G1 × G1 ↦ G2,
        zgoščevalna funkcija h : {0,1}* ↦ G2
Output: boolean
 1. If Not M_i različna Then
        Return False
 2. k ← |U|, prod ← 1
 2. For i = 1 To k Do
    2.1 h_i ← h(M_i)
    2.2 prod ← prod · e(v_i, h_i)
 3. If e(g1, σ) = prod Then
        Return True
    Else
        Return False
```

Podpis uporabnika u_i je oblike $\sigma_i = h_i^{x_i}$, kjer je h_i zgostitev sporočila M_i. Če upoštevamo lastnosti bilinearne preslikave, lahko levo stran enačbe koraka 3. v zadnjem algoritmu

razvijemo v:

$$e(g_1, \sigma) = e(g_1, \prod_{i=1}^{k} h_i^{x_i}) = \prod_{i=1}^{k} e(g_1, h_i)^{x_i} = \prod_{i=1}^{k} e(g_1^{x_i}, h_i) = \prod_{i=1}^{k} e(v_i, h_i).$$

5.2 Osnovna CBE

Naj bo k varnostni parameter v namestitvenem algoritmu, algoritem IG pa naj bo BDH generator parametrov, ki vrne trojico $(\mathbb{G}_1, \mathbb{G}_2, e)$, kjer sta $\mathbb{G}_1$ in $\mathbb{G}_2$ praštevilskega reda q in $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ Weilovo parjenje.

Namestitev: Certifikatna agencija stori naslednje.

1) Na vhodnem parametru k izvede algoritem IG in zgenerira grupi $\mathbb{G}_1$ in $\mathbb{G}_2$ praštevilskega reda q in sprejemljivo parjenje $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

2) Izbere poljuben generator $P \in \mathbb{G}_1$.

3) Izbere naključni skriti $s_C \in \mathbb{Z}_q$ in izračuna $Q = s_C P$.

4) Izbere dve kriptografski zgoščevalni funkciji $H_1 : \{0,1\}^* \to \mathbb{G}_1$ in $H_2 : \mathbb{G}_2 \to \{0,1\}^n$, za nek n.

Parametri v sistemu so $\texttt{param} = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, P, Q, H_1, H_2)$, prostor sporočil je $\mathcal{M} = \{0,1\}^n$ in glavni skriti ključ certifikatne agencije je $s_c \in \mathbb{Z}_q$.
Certifikatna agencija na podlagi $\texttt{param}$ in s_C izdaja certifikate. Naj bo par $(s_B, s_B P)$ Bojanov par ključev (skriti, javni), kjer je $s_B P$ zgeneriran iz $\texttt{param}$, katere izda certifikatna agencija.

Izdaja certifikata (potrjevanje veljavnosti):

1) Bojan pošlje certifikatni agenciji informacijo o sebi, označimo jo z $\texttt{info}_B$. Informacija vsebuje Bojanov javni ključ $s_B P$ in še nekaj njegovih identifikacijskih podatkov (ime, elektronska pošta, ...).

2) Certifikatna agencija preveri prejete podatke.

3) Če so podatki pravilni, v i-tem časovnem intervalu certifikatna agencija izračuna $P_B = H_1(s_C P, i, \texttt{info}_B) \in \mathbb{G}_1$.

4) Certifikatna agencija na koncu izračuna še $\mathsf{Cert}_B = s_C P_B$ in ta certifikat pošlje Bojanu.

Pred odšifriranjem Bojan podpiše svoj $\texttt{info}_B$ in dobi $s_B P'_B$, kjer je $P'_B = H_1(\texttt{info}_B)$. Bojan bo torej za svoj odšifrirni ključ uporabljal $S_B = s_C P_B + s_B P'_B$, kar je ravno iz dveh podpisov združen podpis, opisan v poglavju 5.1.

Šifriranje: Za šifriranje sporočila $M \in \mathcal{M}$, Anita izvede naslednje korake.

1) Izračuna $P'_B = H_1(\texttt{info}_B) \in \mathbb{G}_1$.

2) Izračuna $P_B = H_1(Q, i, \texttt{info}_B) \in \mathbb{G}_1$.

3) Izbere naključno število $r \in \mathbb{Z}_q$.

4) Sestavi tajnopis $C = [rP, M \oplus H_2(g^r)]$, kjer je $g = \hat{e}(s_C P, P_B)\hat{e}(s_B P, P'_B) \in \mathbb{G}_2$.

Anita lahko zmanjša računsko zahtevnost postopka z uporabo vnaprej izračunanih vrednosti, kot npr. $\hat{e}$, saj je obdobje njegove veljavnosti dolgo.

Odšifriranje: Za odšifriranje sporočila $[U, V]$ (tu smo z U označili večkratnik rP in z V izraz $M \oplus H_2(g^r)$), Bojan izračuna

$$M = V \oplus H_2(\hat{e}(U, S_B)).$$

5.3 Polna CBE

Preoblikovana shema, katero poimenujemo 'polna CBE', uporabi še dve dodatni zgoščevalni funkciji H_3 in H_4 ter varno simetrično šifrirno shemo E. Namestitev in izdaja certifikata (potrjevanje veljavnosti) sta enaki kot pri osnovni CBE shemi, medtem ko šifriranje in odšifriranje potekata na naslednji način:

Šifriranje: Za šifriranje sporočila $M \in \mathcal{M}$, Anita izvede naslednje korake.

1) Izračuna $P'_B = H_1(\texttt{info}_B) \in \mathbb{G}_1$.

2) Izračuna $P_B = H_1(Q, i, \texttt{info}_B) \in \mathbb{G}_1$.

3) Izbere naključni $\sigma \in \{0,1\}^n$.

4) Izračuna $r = H_3(\sigma, M)$.

5) Sestavi tajnopis $C = [rP, \sigma \oplus H_2(g^r), E_{H_4(\sigma)}(M)]$,
kjer je $g = \hat{e}(s_C P, P_B)\hat{e}(s_B P, P'_B) \in \mathbb{G}_2$.

Odšifriranje: Za odšifriranje sporočila $[U, V, W]$, Bojan stori naslednje.

1) Izračuna $\sigma = V \oplus H_2(\hat{e}(U, S_B))$.

2) Izračuna $M = E^{-1}_{H_4(\sigma)}(W)$.

3) Določi $r = H_3(\sigma, M)$ in zavrne tajnopis v primeru, ko $U \neq rP$, sicer je M pravi čistopis.

Da je polna CBE varna pred napadi na izbrani tajnopis, bomo pokazali z dokazom leme. Preden pa si jo ogledamo, si oglejmo konkreten primer kriptosistema z javnim ključem, imenovanega $BasicPub^{hy}$. Tega bomo potem uporabili v lemi ter dokazu.

Na začetku si bomo ogledali $BasicPub^{hy}$, kriptosistem z javnim ključem, ki uporablja Fujisaki-Okamotovo [8] transformacijo, da doseže varnost pred napadi na izbrani tajnopis glede na bilinearno Diffie-Hellmanovo predpostavko. Naj bo k varnostni parameter iz nastavitvenega algoritma, algoritem IG pa generator BDH parametrov.
Nastavitve: Imetnik ključa stori naslednje.

1) Na vhodnem parametru k izvede algoritem IG, da zgenerira grupi $\mathbb{G}_1$ in $\mathbb{G}_2$ praštevilskega reda q in sprejemljivo parjenje $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

2) Izbere si naključna generatorja $P, P_1 \in \mathbb{G}_1$.

3) Izbere si naključno skrito število $s_C \in \mathbb{Z}_q$ in izračuna $Q = s_C P$.

4) Izbere si kriptografsko zgoščevalno funkcijo $H_2 : \mathbb{G}_2 \mapsto \{0,1\}^n$.

5) Uporablja zgoščevalni funkciji H_3 in H_4 in varno shemo za šifriranje, kakršno sta definirala Fujisaki in Okamoto.

Javni ključ je $K_{pub} = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_1, Q, H_2, H_3, H_4, E)$, medtem ko je $s_C P_1$ skriti ključ. Prostor sporočil je še vedno $\mathcal{M} = \{0,1\}^n$.
Šifriranje: Za šifriranje sporočila $M \in \mathcal{M}$, pošiljatelj stori naslednje.

1) Izbere naključni $\sigma \in \{0,1\}^n$.

2) Izračuna $r = H_3(\sigma, M)$.

3) Zgenerira tajnopis $C = [rP, \sigma \oplus H_2(g^r), E_{H_4(\sigma)}(M)]$, kjer je $g = \hat{e}(Q, P_1)$.

Odšifriranje: Prejemnik sporočila $[U, V, W]$ za odšifriranje stori naslednje.

1) Izračuna $\sigma = V \oplus H_2(\hat{e}(U, s_C P_1))$.

2) Izračuna $M = E^{-1}_{H_4(\sigma)}(W)$.

3) Izračuna $r = H_3(\sigma, M)$. Če je $U = rP$, potem je dobljeni čistopis pravi, sicer pa prejeti tajnopis zavrne.

Lema 5.1. *Naj bo $\mathcal{A}$ IND-CCA napadalec na polno CBE s prednostjo ε. Naj bosta q_C in q_D maksimalni števili poizvedb za potrjevanje ter odšifriranje, katere izvede napadalec $\mathcal{A}$. Potem obstaja IND-CCA napadalec $\mathcal{B}$ z enako časovno zahtevnostjo kot $\mathcal{A}$, ki ima prednost vsaj $\varepsilon/e(1 + q_C + q_D)$ pred $BasicPub^{hy}$.*

Dokaz: Algoritme sheme $BasicPub^{hy}$ smo že spoznali, ogledali si bomo še, kako lahko dobimo napadalca $\mathcal{B}$, ki preko napadalca $\mathcal{A}$ dobi prednost $\varepsilon/(1+q_C+q_D)$ pred $BasicPub^{hy}$. Igra med izdajateljem certifikata (certifikatno agencijo) in napadalcem $\mathcal{B}$ se začne tako, da prvi zgenerira javni ključ $K_{pub} = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, P_1, Q, H_2, H_3, H_4, E)$ in skriti ključ $s_C P_1$. $\mathbb{G}_1$ in $\mathbb{G}_2$ sta praštevilskega reda q, $s_C \in \mathbb{Z}_q$ in $Q = s_C P$. Certifikatna agencija nato dobljeni javni ključ K_{pub} pošlje napadalcu $\mathcal{B}$, ki z napadalcem $\mathcal{A}$ sodeluje na naslednji način.

Nastavitve: Naj bo H_1 naključni orakelj, katerega ima pod kontrolo napadalec $\mathcal{B}$, ki poda parametre $\texttt{param} = (\mathbb{G}_1, \mathbb{G}_2, \hat{e}, n, P, Q, H_1, H_2, H_3, H_4, E)$

Poizvedbe H_1: Napadalec $\mathcal{A}$ lahko kadarkoli pošlje dve vrsti poizvedb H_1. Pri prvi pošlje $(i_j, s_j P, w_j)$, kjer je i_j časovni interval, $s_j P$ javni ključ, za katerega bi napadalec $\mathcal{A}$ rad potrdil veljavnost, w_j pa neka dodatna informacija iz certifikata, kot npr. ime. Te vrste poizvedb se pošlje za pridobivanje P_B v popolni CBE shemi. Druga vrsta poizvedb H_1 je oblike $(s_j P, w_j)$. Te vrste poizvedb se uporabljajo za pridobivanje P'_B v popolni CBE. Napadalec $\mathcal{B}$ si beleži vse odgovore na H_1 poizvedbe, da bo na enako poizvedbo vedno enako odgovoril. Na začetku je seznam odgovorov na poizvedbe prazen.

Ko napadalec $\mathcal{B}$ sprejme poizvedbo, ki je poslal napadalec $\mathcal{A}$, nanjo odgovori na naslednji način.

1) Če napadalec $\mathcal{A}$ pošlje poizvedbo, na katero je napadalec $\mathcal{B}$ že odgovoril, potem v seznamu odgovorov poišče pravi odgovor in mu ga pošlje.
2) Če prejme poizvedbo druge vrste, opisane zgoraj, generira naključen $b_j \in \mathbb{Z}_q$ in izračuna $P'_j = b_j Q$. $(s_j P, w_j, b_j)$ doda v seznam in pošlje P'_j nazaj napadalcu $\mathcal{A}$.
3) Če prejme poizvedbo prve vrste, potem sam izvede poizvedbo druge vrste z $(s_j P)$, da dobi b_j. Napadalec $\mathcal{B}$ generira $\texttt{coin}_j \in \{0, 1\}$ z verjetnostjo $P(\texttt{coin}_j = 0) = \delta$, kjer bomo vrednost δ določili kasneje in $c_j \in \mathbb{Z}_q$. Če je $\texttt{coin}_j = 0$, potem postavi $P_j = c_j P$, sicer pa $P_j = c_j P_1 - b_j s_j P$. Napadalec $\mathcal{B}$ nato v seznam doda $(i_j, s_j P, w_j, c_j, \texttt{coin}_j)$ in pošlje napadalcu $\mathcal{A}$ vrednost P_j.

Opazimo, da sta točki P'_j in P_j enolični v $\mathbb{G}_1$ in neodvisni od napadalca $\mathcal{A}$.

1. faza - Odgovori na poizvedbe za potrjevanje: Ko napadalec $\mathcal{B}$ prejme poizvedbo za potrjevanje veljavnosti, nanjo odgovori na naslednji način.

1) Na $(i_j, w_j, s_j P)$ izvede H_1 algoritem in dobi c_j ter $\texttt{coin}_j$. Če je $\texttt{coin}_j = 1$, potem $\mathcal{B}$ prekine svoje izvajanje, saj napad na $BasicPub^{hy}$ ni uspel.
2) Če je $\texttt{coin}_j = 0$, potem $\mathcal{B}$ pošlje napadalcu $\mathcal{A}$ njegov certifikat $s_C P_j = c_j Q$.

1. faza - Odgovori na poizvedbe za odšifriranje: Naj bo $C_j = (U_j, V_j, W_j)$ in $(i_j, s_j P, w_j, C_j)$ poizvedba za šifriranje, ki je poslal napadalec $\mathcal{A}$. Napadalec $\mathcal{B}$ na tako poizvedbo odgovori na naslednji način.

1) Na $(i_j, w_j, s_j P)$ izvede H_1 algoritem in dobi b_j, c_j ter $\texttt{coin}_j$.
2) Če je $\texttt{coin}_j = 0$, potem $\mathcal{B}$ izračuna odšifrirni ključ $s_C P_j + s_j P'_j = c_j Q + b_j (s_j P)$ in ga uporabi za odšifriranje C_j.
3) Če je $\texttt{coin}_j = 1$, potem $\mathcal{B}$ vzame $C'_j = (c_j U, V, W)$ in ga pošlje izdajatelju certifikata (certifikatni agenciji). Ko od le tega prejme odgovor, ga posreduje naprej napadalcu $\mathcal{A}$.

Spomnimo se, da je izdajateljev skriti ključ $s_C P_1$. Opazimo, da je $\hat{e}(c_j U, s_C P_1) = \hat{e}(rP, c_j s_C P_1) = \hat{e}(rP, s_C(c_j P_1 - b_j s_j P) + s_j b_j Q) = \hat{e}(rP, s_C P_j + s_j P'_j)$, kjer je $s_C P_j + s_j P'_j$

odšifrirni ključ iz polne CBE sheme. Izdajatelj torej poda pravilni odgovor glede na C_j.

Napad: Ko se napadalec $\mathcal{A}$ odloči, da je 1. faza zaključena, pošlje napadalno poizvedbo za pridobitev tajnopisa sporočil M_0 ali M_1 na $(i_z, s_z P, w_z)$. Napadalec $\mathcal{B}$ odgovori na naslednji način.

1) Izdajatelju pošlje sporočili M_0 in M_1 kot sporočili, ki sta uporabljeni za napad. Izdajatelj pošlje nazaj tajnopis $C = [U, V, W]$, zgeneriran v $BasicPub^{hy}$, kateri pripada čistopisu M_x, kjer je $x \in \{0, 1\}$.
2) Na $(i_z, s_z P, w_z)$ izvede H_1 algoritem, da dobi c_z in $\texttt{coin}_z$. Če je $\texttt{coin}_z = 0$, potem napad na $BasicPub^{hy}$ ni uspel in $\mathcal{B}$ konča.
3) Če pa je $\texttt{coin}_z = 1$, potem $\mathcal{B}$ pošlje napadalcu $\mathcal{A}$ tajnopis $C' = [c_j^{-1}U, V, W]$.

Ker je $\hat{e}(c_j^{-1}U, s_C P_j, +s_j P'_j) = \hat{e}(c_j^{-1}U, c_j s_{CP_1}) = \hat{e}(U, s_C P_1)$, kjer je $s_C P_1$ skriti ključ izdajatelja, je tajnopis, vrnjen pri napadalni poizvedbi res tajnopis, ki ustreza čistopisu M_x v popolni CBE sheme, kot smo zahtevali.

2. faza - Odgovori na poizvedbe za potrjevanje: Napadalec $\mathcal{B}$ odgovori na poizvedbe enako kot v 1. fazi.

2. faza - Odgovori na poizvedbe za odšifriranje: Napadalec $\mathcal{B}$ odgovori nanje kot v 1. fazi, le da preneha, če je poizvedba za odšifriranje, katero naj bi posredoval naprej izdajatelju enaka C.

Ugibanje: Napadalec $\mathcal{A}$ za x izbere x', katerega uporabi tudi napadalec $\mathcal{B}$.

Trditev 5.1. *Če med simulacijo napada napadalec $\mathcal{B}$ ne preneha z izvajanjem, potem je pogled napadalca $\mathcal{A}$ identičen, kot bi bil pri resničnem napadu in je verjetnost $P(M = M') \geq \varepsilon$.*

Dokaz: Vsi odgovori na H_1 poizvedbe so enaki kot pri dejanskem napadu, saj je vsak odgovor enolično in neodvisno razporejen v $\mathbb{G}_1$. Ravno tako so vsi odgovori na poizvedbe za šifriranje in odšifriranje veljavni. Vidimo tudi, da je tajnopis C', vrnjen pri napadalni poizvedbi res tajnopis, ki v popolni CBE shemi ustreza čistopisu M_x z javnim ključem, ki ga je izbral napadalec $\mathcal{A}$. Po naši definiciji algoritma bo za rezultat vrnil $x' = x$ z verjetnostjo vsaj ε. ∎

Verjetnost: Verjetnost, da bo $\mathcal{B}$ prenehal s svojim izvajanjem napada je identična kot v Boneh-Franklinovi IBE shemi, o kateri si lahko bralec več prebere v [7].
S tem smo končali z dokazom leme 5.1. ■

Poglavje 6

Zmanjševanje računske zahtevnosti

V primeru, ko ima certifikatna agencija veliko število uporabnikov z veljavnimi certifikati in pogosto izdaja potrditve teh certifikatov, mora izvesti precej računskih operacij. Če ima izdanih npr. 225 milijonov veljavnih certifikatov in potrjuje njihovo veljavnost vsako uro, mora potrditi 62500 certifikatov na sekundo. Z uvedbo pokritja podmnožic bomo to zahtevnost občutno zmanjšali.

6.1 Splošni pristop

Predpostavimo, da certifikatna agencija izda dolgo veljavne certifikate svojim uporabnikom (ne kot pri splošni CBE shemi), katerim potem periodično potrjuje veljavnost. Anita mora, preden želi poslati Bojanu šifrirano sporočilo, dobiti in potrditi veljavnost Bojanovega certifikata, kar pa ni zahtevna operacija, saj lahko to stori istočasno s pridobivanjem Bojanovega javnega ključa. (Spomnimo se, da je glavna prednost CBE sheme v tem, da Aniti ni potrebno dobiti sveže informacije o veljavnosti Bojanovega certifikata.)

Certifikatna agencija izbere N svojih uporabnikov, kjer je $N < 2^m$ in jih postavi kot liste m-nivojskega dvojiškega drevesa in vsakemu izmed dolgo veljavnih certifikatov dodeli m-bitno unikatno serijsko številko. V vsakem časovnem intervalu vsako vozlišče drevesa ustreza neki identiteti. Certifikatna agencija izračuna odšifrirni ključ po IBE shemi za vsakega izmed njih. V i-tem časovnem intervalu lahko vozlišče, ki pripada serijski številki $b_1 \dots b_k$ ($k \leq m$), preslikamo v $\langle i, b_1 \dots b_k \rangle$. Odšifrirni ključ, povezan s tem vozliščem, je odšifrirni ključ iz IBE sheme za $\langle i, b_1 \dots b_k \rangle$.

$\langle i, b_1 \dots b_k \rangle$ imenujemo prednik $\langle i, b_1 \dots b_m \rangle$, $k \leq m$. V i-tem časovnem intervalu certifikatna agencija poišče množico vozlišč S, za katere velja: vsak izmed $N-R$ uporabnikov z veljavnim certifikatom (N je število vseh uporabnikov, R je število uporabnikov s preklicanim certifikatom) ima prednika v tej množici, vendar pa nobeden izmed R uporabnikov ni potomec katerega izmed izbranih vozlišč. To množico bomo imenovali pokritje za veljavne certifikate. Takšna množica vsebuje največ $R \log N/R$ vozlišč. Postopek za iskanje takšne množice si lahko bralec ogleda v [11]. Certifikatna agencija potem objavi odšifrirne ključe vseh teh vozlišč iz množice S. Imenujemo jih *potrditveni certifikati*.

Torej, ko želi Anita poslati Bojanu šifrirano sporočilo, ne potrebuje njegovega potrditvenega certifikata. (Ker ima Anita Bojanov dolgo veljavni certifikat, ima tudi njegov javni ključ in serijsko številko.) Sporočilo šifrira $(m+1)$-krat. Na vsakem koraku uporabi poleg Bojanovega javnega ključa še identiteto vsakega izmed $(m+1)$ Bojanovih prednikov iz drevesa. Bojan bo lahko odšifriral eno izmed sporočil le, če je certifikatna agencija objavila potrditveni certifikat katerega izmed vozlišč, ki so njegovi predniki. Kljub temu, da opisana shema poveča kompleksnost šifriranja in dolžino tajnopisa, mora certifikatna agencija izračunati le $R \log N/R$ potrditvenih certifikatov, namesto $N-R$ (splošna CBE). Če predpostavimo, da je veljavnost certifikatov eno leto in jih je od tega 10 odstotkov preklicanih pred koncem veljavnosti, smo na ta način računsko zahtevnost zmanjšali za faktor 3.

6.2 CBE shema z uporabo parjenja in pokritja podmnožic

Zgoraj opisan splošni pristop oteži dokazljivost varnosti sheme pred napadi na izbrani tajnopis, saj isto sporočilo šifriramo z različnimi ključi. Na kratko bomo opisali splošno CBE shemo, ki izhaja iz omenjenega splošnega pristopa. Zopet privzamemo, da ima Bojan svoj dolgo veljavni certifikat s serijsko številko $b_1 \dots b_m$.

Izdaja certifikata (potrjevanje veljavnosti): Certifikatna agencija izvede enake namestitve kot pri splošni CBE shemi in uporabi še $H_5 : \{0,1\}^n \rightarrow \mathbb{G}_1$ za preslikavo časovnih intervalov v točke oziroma elemente grupe $\mathbb{G}_1$. Na začetku i-tega časovnega intervala certifikatna agencija izbere naključni $x \in \mathbb{Z}_q$ in poišče pokritje S za uporab-

nike z nepreklicanim certifikatom. Bojanov potrditveni certifikat (če obstaja) ima obliko $S_i = s_C T_i + xP_k$ skupaj s $Q = xP$, kjer je $T_i = H_5(Q, i)$, $P_k = H_1(b_1 \dots b_k)$ in $b_1 \dots b_k \in S$ serijska številka prednika vozlišča s serijsko številko $b_1 \dots b_m$.

Šifriranje: Anita je že preverila Bojanov osnovni certifikat in pozna $\texttt{info}_B$. Izbere si naključni $r \in \mathbb{Z}_q$. Bojanu pošlje tajnopis $C = [rP, rP_1, ..., rP_m, V]$, kjer je $V = M \oplus H_2(g^r)$ in $g = \hat{e}(Q, T_i)\hat{e}(s_B P, P'_B)$.

Odšifriranje: Bojan izračuna

$$M = V \oplus H_2\left(\frac{\hat{e}(rP, S_i + s_B P'_B)}{\hat{e}(xP, rP_k)}\right).$$

Šifriranje vsebuje torej $m + 1$ množenj točk, odšifriranje pa le dve računanji parjenja. Bojanov potrditveni certifikat je kratek, sestavljata ga le dva elementa iz $\mathbb{G}_1$. (V bistvu le en, saj je xP enak pri vseh uporabnikih.)

Poglavje 7

Razširitev CBE sheme z uporabo pokritja

V shemi, opisani v prejšnjem poglavju, mora certifikatna agencija narediti približno $R \log (N/R)$ potrditev certifikatov na uro, kjer je R število vseh preklicanih certifikatov. Število potrditev je torej enako, ne glede na to, ali je bil v trenutni uri (časovnem intervalu) preklican kak certifikat. V tem poglavju bomo opisali razširjeno shemo CBE, v kateri bo število potrditev odvisno od tega, koliko certifikatov je bilo preklicanih v trenutni uri (oziroma časovnem intervalu, na katere certifikatna agencija obnavlja veljavnosti). Pri predpostavkah, katere smo uporabili že prej, to je, da bo preklicanih nekje deset odstotkov vseh certifikatov z veljavnostjo eno leto in številom uporabnikov 250 milijonov, bo morala certifikatna agencija izvesti približno 13 potrditev veljavnosti na sekundo. (To je drastično zmanjšanje v primerjavi s prejšnjim 62500.)

7.1 Osnovna razširjena CBE shema

Zmanjšanje operacij, ki jih mora izvesti certifikatna agencija na vsakem časovnem intervalu posodabljanja certifikatov ima tudi svojo ceno. Uporabnika, ki ni bil potrjen pri zadnjem potrjevanju, ne moremo smatrati za veljavnega (oz. z veljavnim certifikatom) in torej ne sme biti sposoben odšifrirati sporočil, razen če ima nepretrgano verigo potrditvenih certifikatov od izdaje njegovega certifikata do tega trenutka. To bi lahko pomenilo, da mora biti kompleksnost šifriranja in odšifriranja proporcionalna s številom časovnih intervalov, vendar se tega znebimo z uporabo parjenja. Poleg tega lahko vsak uporabnik

združi svoje periodične certifikate v en sam odšifrirni ključ (sestavljen iz $(\log N + 1)$ elementov iz $\mathbb{G}_1$). Pri tem so kompleksnosti šifriranja in odšifriranja ter dolžina tajnopisa $(\log N)$-krat daljši, kot pri splošni CBE shemi, ne glede na to, koliko časovnih intervalov je minilo od izdaje certifikata.

Predpostavimo, da je Bojan že dobil svoj certifikat, ki vključuje serijsko številko $b_1 \dots b_m$ ter čas izdaje certifikata t_0.

Izdaja certifikata (potrjevanje veljavnosti): Certifikatna agencija izvede enake namestitve kot pri splošni CBE shemi in uporabi še $H_5 : \{0,1\}^n \to \mathbb{G}_1$ za preslikavo časovnih intervalov v točke oziroma elemente grupe $\mathbb{G}_1$. Na začetku i-tega časovnega intervala certifikatna agencija izbere naključni element $x \in \mathbb{Z}_q$ in poišče pokritje S za uporabnike, kateri niso imeli preklica certifikata v prejšnjem časovnem intervalu. Bojanov potrditveni certifikat ima obliko $S'_i = s_C(T_i - T_{i-1}) + xP_k$ skupaj s $Q = xP$, kjer je $T_i = H_5(Q, i)$, $P_k = H_1(b_1 \dots b_k)$ in $b_1 \dots b_k \in S$ serijska tevilka prednika vozlišča s serijsko številko $b_1 \dots b_m$.

Združitev: Če je certifikatna agencija na vsakem časovnem intervalu (od izdaje do i-tega časovnega intervala) potrdila Bojanov certifikat, potem bi radi, da si Bojan lahko izračuna svoj združeni certifikat, ki bo naslednje oblike:

$$S_i = s_C(T_i - T_{t_0}) + x_{i,1}P_1 + ... + x_{i,m}P_m,$$

skupaj s $Q_{i,j} = x_{i,j}P$ za $1 \leq j \leq m$, kjer je $x_{i,j} \in \mathbb{Z}_q$ in $P_j = H_1(b_1 \dots b_j)$. Recimo, da ima Bojan združeni certifikat S_{i-1} pravilne oblike s prejšnjega časovnega intervala. Po pridobitvi S'_i in xP, si izračuna nov združeni certifikat na naslednji način: $S_i = S_{i-1} + S'_i$, $Q_{i,j} = Q_{i-1,j}$, kjer $j \neq k$ in $Q_{i,k} = Q_{i-1,k} + xP$.

Šifriranje: Anita je že preverila Bojanov osnovni certifikat in pozna $\texttt{info}_B$. Izbere si naključni $r \in \mathbb{Z}_q\mathbb{Z}$. Bojanu pošlje tajnopis $C = [rP, rP_1, ..., rP_m, V]$, kjer je $V = M \oplus H_2(g^r)$ in $g = \hat{e}(Q, T_i - T_{t_0})\hat{e}(s_B P, P'_B)$.

Odšifriranje: Bojan izračuna

$$M = V \oplus H_2\left(\frac{\hat{e}(rP, S_i + s_B P'_B)}{\prod_{j=1}^{m} \hat{e}(rP_j, Q_{i,j})}\right).$$

Opomba 7.1. *V primeru, ko certifikatna agencija ob izdaji certifikata pošlje Bojanu certifikat $s_C T_{t_0} + x_{t_0} P_m$ skupaj s $Q_{t_0,m} = x_{t_0} P$ in Bojan to vključi pri svojem združevanju, potem odštevanje "$-T_{t_0}$ "pri šifriranju ni potrebno.*

7.2 Varnost razširjene CBE sheme

Certifikat S_i vsebuje časovno komponento oblike $s_C(T_i - T_{t0})$ in identifikacijsko komponento oblike $x_1P_1 + ... + x_mP_m$. Če bi hotel Bojan združiti svoje certifikate brez potrditvenega certifikata na dan z, $t_0 \leq z \leq i$, bi imel rezultat obliko $s_C(T_i - T_z + T_{z-1} - T_{t_0}) + x_1P_1 + ... + x_mP_m$. Časovna komponenta certifikata S_i bi bila torej napačne oblike. Če pa bi Bojan na časovnem intervalu z želel zamenjati svojo potrditev s potrditvijo nekoga drugega, bi imel S_i identifikacijsko komponento napačne oblike.

7.3 Zahtevnost in zmogljivost opisane sheme

Pri vseh opisanih CBE shemah smo odstranili poizvedbe tretje osebe. Pri razširjeni CBE shemi smo računsko zahtevnost zmanjšali na potrjevanje veljavnosti trinajstih certifikatov na sekundo. Glede na to, da je potrjevanje veljavnosti glede na računsko zahtevnost ekvivalentno generiranju BLS podpisa, katerega zgeneriramo na računalniku s procesorjem *Pentium III, 1 GHz* v 3.75 ms [12], lahko torej tako potrjevanje izvajamo z osebnim računalnikom.

Izdajanje potrditvenih certifikatov lahko poteka na različne načine. Zanimiv način je uporaba direktorijev in pošiljanje certifikatov uporabnikom na vsakem časovnem intervalu (metoda 'push'). S tem se znebimo vsakršnih poizvedb.[1] V primeru, ko certifikatna agencija pošilja potrditvene certifikate vsakemu uporabniku posebej, mora imeti povezavo z zmogljivostjo 20 Mbit/s (320 bitov na certifikat $\times$ 225 milijonov certifikatov na uro / 3600 sekund). To je seveda spodnja meja, saj nismo upoštevali dodatnih podatkov, kot npr. glave paketov, katere pošiljamo. Seveda pa lahko zmogljivost povezave, katero certifikatna agencija potrebuje, zmanjšamo s pošiljanjem podatkov več prejemnikom hkrati (ang. multicast). Spomnimo se, da potrditveni certifikat za vozlišče $b_1b_2 \ldots b_k$ velja tudi za vse njegove potomce. S postavitvijo hkratnih naslovov za vsako notranje vozlišče drevesa lahko certifikatna agencija precej zmanjša zahtevo po zmogljivosti povezave. S prejšnjih 20 Mbit/s s tem zmanjšamo zmogljivost povezave na približno 2.1 Kbit/s (160 bitov na

[1] V praksi lahko certifikatna agencija dovoli nekaj poizvedb s strani uporabnikov, kot tudi s strani tretjih oseb. Vendar pa CBE shema omogoča certifikatni agenciji omejevanje teh poizvedb po svoji želji. Certifikatna agencija lahko na primer to stori z zaračunavanjem vsake poizvedbe.

certifikat * 13 certifikatov na sekundo). Tu računamo 160 bitov na certifikat, ker lahko 160 bitov za xP pošljemo vsem uporabnikom z enim samim hkratnim pošiljanjem.

Za uporabnike je razširjena CBE shema morda neuporabna, saj je računska zahtevnost šifriranja in odšifriranja približno $\lceil \log(N) \rceil$-krat dražja od zahtevnosti že omenjene Boneh-Franklinove sheme. Zaradi tega je lahko opisana shema nepraktična za nekatere aplikacije, vendar pa lahko shemo uporabljamo za aplikacije, kjer hitro računanje prek internetne ali mrežne povezave (ang. on-line) ni problem, npr. elektronska pošta. V prihodnosti lahko pričakujemo, da bo pomen računske zahtevnosti padel v primerjavi zahtevnosti po hitri internet oziroma mrežni povezavi in tu bo lahko CBE shema pokazala vse svoje prednosti. Če želimo, lahko še vedno povečamo računsko zahtevnost s strani certifikatne agencije, tako da uporabljamo drevesa, kjer imajo vozlišča več potomcev, ali pa da certifikatna agencija vzdržuje več dreves hkrati.

Dolžina združenega potrditvenega certifikata, katerega si uporabnik sestavi, nikoli ne preseže dolžine $160(\lceil log(N) \rceil + 1)$ bitov. Eksplicitno potrjevanje veljavnosti certifikata je v razširjeni shemi bolj zgoščeno kot v shemi, katero so opisali avtorji v [13], čeprav CBE shema ni nikoli bila mišljena kot shema za izboljšanje eksplicitnega potrjevanja.

Poglavje 8

Posplošitve in razširitve CBE sheme

V tem poglavju bomo opisali dva posebna primera CBE sheme in na grobo opisali še nekaj njenih posplošitev.

8.1 CBE shema s sprotnim potrjevanjem

Včasih potrjevanje veljavnosti certifikatov vsako uro ni zadovoljivo in bi si želeli nekakšno sprotno potrjevanje. Anita ima možnost šifriranja z uporabo ključa iz $(i+1)$-ega intervala na i-tem časovnem intervalu, vendar bi moral potem Bojan čakati eno uro, da bi lahko sporočilo odšifriral. Ena izmed rešitev za sprotno potrjevanje veljavnosti se imenuje SEM. To je on-line strežnik, kateremu lahko delno zaupamo, imenovan tudi posrednik za varnost (ang. Security Mediator). Pri SEM je preklic certifikata nemuden, tako kot tudi Bojanova sposobnost odšifriranja. Vendar pa mora Bojan dobiti potrdilo za veljavnost njegovega certifikata od posrednika za varnost, če želi odšifrirati sporočila. V tem poglavju bomo opisali takšno CBE shemo, pri kateri poteka potrjevanje veljavnosti certifikatov neprestano, recimo vsako sekundo in kjer komunikacija med Bojanom in certifikatno agencijo ne raste s številom sporočil, katere Bojan šifrira, ampak raste s številom Bojanovih sej.

Preden opišemo shemo, si moramo ogledati pojem vnaprej varne sheme za šifriranje, imenovane **FSE** (ang. forward-secure encryption). Pri vnaprej varnem protokolu za izmenjavo skritih ključev gre v bistvu za to, da se v primeru odkritja skritega ključa ne poruši varnost skritih ključev, zgeneriranih v prejšnjih sejah. Takšen protokol je potem osnova za vnaprej varne sheme, v kateri pošiljatelj in prejemnik najprej zgenerirata skriti

ključ K, pošiljatelj ga uporabi za šifriranje, nato ga takoj oba izbrišeta. Več o FSE shemi lahko najdete v [15]. CBE shema z zgoščenim potrjevanjem bo uporabila FSE shemo, vendar v obratnem vrstnem redu. To pomeni, da lahko uporabnik z odšifrirnim ključem za časovni interval i izračuna odšifrirni ključ za vsak časovni interval j, kjer je $j < i$, ne pa za $j > i$. Bojan lahko takšen certifikat pridobi večkrat dnevno, odvisno od tega, kako pogosto pregleda svoja sporočila. Takšno razširitev sheme, ki sprotno potrjuje veljavnost certifikatov lahko izvedemo, ne da bi s tem občutno pomanjšali njeno zmogljivost, saj je velikost takih certifikatov le logaritemska glede na število časovnih intervalov. Uporabimo pa lahko tudi kombinacijo opisane sheme in prej opisane razširjene CBE sheme. Pri tem certifikatna agencija potrdi veljavnost certifikatov vsako uro, vendar pa uporablja poddrevo s 3600 sekundami za uporabnike, ki želijo shemo z zgoščenim potrjevanjem.

8.2 Hierarhična CBE shema

V tem poglavju bomo opisali, kako lahko CBE shemo uporabimo v hierarhičnem sistemu certifikatnih agencij. Do sedaj opisane sheme so vse uporabljale le eno certifikatno agencijo, vendar je posplošitev na več hierarhično razporejenih certifikatnih agencij precej enostavna. Najbolj očiten pristop je uporaba hierarhične IBE sheme v kombinaciji s PKE shemo, vendar pa pri takem pristopu povečamo tako dolžino tajnopisa, kot tudi kompleksnost šifriranja in odšifriranja za faktor t glede na te lastnosti pri Boneh-Franklinovi shemi, kjer t označuje nivo prejemnika v hierarhičnem drevesu. Namesto tega lahko uporabimo že prej omenjeni združeni BGLS podpis. Kompleksnost šifriranja sporočil je tako še vedno proporcionalna t, vendar pa sta dolžina tajnopisa in kompleksnost odšifriranja enaka kot pri Boneh-Franklinovi shemi. Naj t predstavlja nivo Bojana, kateri ima javni ključ s_tP in naj imajo certifikatne agencije na nivojih višje od njega javne ključe s_jP, kjer je $0 \leq j \leq t-1$. Postopki v tej shemi potekajo na naslednji način:

Potrjevanje certifikatne agencije: CA_j podpiše javni ključ $s_{j+1}P$ certifikatne agencije CA_{j+1} s podpisom oblike s_jP_{j+1}, kjer je $p_{j+1} = H_1(s_jP, \texttt{info}_{CA_{j+1}})$ in $\texttt{info}_{CA_{j+1}}$ vsebuje ključ $s_{j+1}P$.

Potrjevanje Bojana: Podobno kot pri potrjevanju certifikatne agencije Bojanu certifikatna agencija nad njim zgenerira certifikat oblike $s_{t-1}P_t$, kjer je $P_t = H_1(s_{t-1}P, \texttt{info}_B)$ in $\texttt{info}_B$ vsebuje s_tP.

Združevanje: Bojan podpiše svoj ključ, da dobi $s_t P'_B$ in združi dobljeni certifikat s certifikati v verigi pred njim. To stori tako, da jih enostavno sešteje:

$$S_{Agg} = s_t P'_B + \sum_{j=1}^{t} s_{j-1} P.$$

Šifriranje: Predpostavimo, da Anita pozna $\texttt{info}_B$ in $\texttt{info}_{CA_j}$ za $0 \leq j \leq t-1$. Izbere si naključni $r \in \mathbb{Z}_q$ in pošlje tajnopis $C = [rP, V]$, kjer je $V = M \oplus H_2(g^r)$ in $g = \hat{e}(P'_B, s_t P) \prod_{j=1}^{t} \hat{e}(P_j, s_{j-i} P)$.

Odšifriranje: Bojan izračuna $M = V \oplus H_2(\hat{e}(rP, S_{Agg}))$.

Opomba 8.1. *Za preklic certifikatov lahko časovne intervale vključimo v certifikate. Vendar pa mora Anita poznati vrstni red potrjevanja certifikatov za vsako certifikatno agencijo, katera je Bojanov prednik v hierarhiji. To precej oteži implementacijo sheme v praksi.*

Opomba 8.2. *Opazimo, da je druga shema uporabna zunaj nastavitev infrastrukture javnih ključev. Imetniku ključa namreč omogoča sposobnost odšifriranja neodvisno od tega, ali je ta imetnik ključa pridobil večkratne podpise oz. je bila veljavnost njegovega certifikata večkrat potrjena.*

Poglavje 9

Zaključek

Kriptografski sistemi z eliptičnimi krivuljami bodo kmalu prevladali na področju kriptografije z javnimi ključi, saj kot sem že omenil uporabljajo za podobno stopnjo varnosti manjše velikosti ključev od ostalih kriptosistemov. Varnost temelji na problemu diskretnega logaritma na eliptični krivulji, torej pri danih točkah P in Q najdi takšno število k, da velja $kP = Q$. Pri velikem številu točk na eliptični krivulji tega problema tudi z največjo računsko močjo trenutno na voljo ni možno rešiti v zadovoljivem času.

V tem delu sem opisal posebno bilinearno preslikavo, imenovano Weilovo parjenje, ki se uporablja v modernih shemah za šifriranje. Eno od teh smo si tudi malce podrobneje ogledali. Poleg Weilovega se v modernejših shemah za šifriranje uporablja tudi Tateovo parjenje. S tema in podobnimi matematičnimi prijemi avtorji shem poskušajo zmanjšati računske zahtevnosti in povečati njihovo varnost. Opisana shema je poimenovana shema za šifriranje s certifikati, saj se certifikat uporablja kot javni ključ. Shema je v bistvu nekakšna nadgraditev infrastrukture javnih ključev.

Glavna prednost CBE sheme je omogočanje implicitnega potrjevanja veljavnost certifikatov brez težav in pomanjkljivosti, s katerimi se sooča šifriranje na osnovi identitete (IBE). Poleg tega smo se znebili poizvedb o statusih certifikatov tretje osebe in s tem precej zmanjšali stroške same sheme oziroma infrastrukture. Pri opisu razširjene CBE sheme smo zmanjšali računsko zahtevnost kot tudi zahtevo po zmogljivi mrežni oziroma internetni povezavi certifikatne agencije. Pri tem nismo uporabili zgoščevalnih verig ali dreves, ki jih uporabljajo pred to shemo objavljeni predlogi za infrastrukturo javnih ključev.

Morda se zdi, da smo na problem preklica certifikata in vzdrževanje njegove veljavnosti malce pozabili, pa ni tako. Spomnimo se, da Bojan sporočila ne more dešifrirati, če

nima veljavnega certifikata v trenutnem časovnem intervalu. Torej smo uporabili boljšo in hitrejšo metodo kot pa seznam preklicanih certifikatov CRL, ki se še vedno uporablja v največ shemah.

Literatura

[1] C. Gentry, *Certificate-Based Encription and the Certificate Revocation Problem*, Advances in Cryptology - EUROCRYPT 2003, LNCS 2656, str. 272–293, Springer, 2003.

[2] S. Micali, *Efficient Certificate Revocation*, Technical Report TM-542b, MIT Laboratory for Computer Science, 1996.
(dostopno na http://portal.acm.org/citation.cfm?id=889659)

[3] A. Enge, *Elliptic Curves and Their Applications to Cryptography - An Introduction*, Kluwer Academic Publishers, 1999.

[4] I. Vidav, *Eliptične krivulje in eliptične funkcije*, Drutvo matematikov, fizikov in astronomov Slovenije, 1991.

[5] J. Barbič, *Schoofov algoritem*, Diplomsko delo, Univerza v Ljubljani, Fakulteta za matematiko in fiziko, 2000.

[6] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*, Advances in Cryptology, Proceedings of CRYPTO '84, LNCS 196, str. 47–53, Springer, 1984.

[7] D. Boneh in M. Franklin, *Identity-Based Encryption from the Weil pairing*, Advances in Cryptology - CRYPTO 2001, LNCS 2139, str. 213–229, Springer, 2001.

[8] E. Fujisaki in T. Okamoto, *Secure integration of asymmetric and symmetric encryption*, Advances in Cryptology - CRYPTO '99, LNCS 1666, str. 537–554, Springer, 1999.

[9] D. Boneh, B. Lynn in H. Shacham, *Short Signatures from the Weil Pairing*, Advances in Cryptology - ASIACRYPT 2001, LNCS 2248, str. 514–532, Springer, 2001.

[10] D. Boneh, C. Gentry, B. Lynn in H. Shacham, *Aggregate and Verifiably Encrypted Signatures from Bilinear Maps*, Advances in Cryptology - EUROCRYPT 2003, LNCS 2656, str. 416–432, Springer, 2003.

[11] D. Naor, M. Naor in J. Lotspicch, *Revocation and Tracing Schemes for Stateless Receivers*, Advances in Cryptology - CRYPTO 2001, LNCS 2139, str. 41–62, Springer, 2001.

[12] P.S.L.M. Barreto, H.Y. Kim, B. Lynn in M. Scott, *Efficient Algorithms for Pairing-Based Cryptosystems*, Advances in Cryptology - CRYPTO 2002, LNCS 2442, str. 354–368, Springer, 2002.

[13] W. Aiello, S. Lodha in R. Ostrovsky, *Fast Digidal Identity Revocation*, Advances in Cryptology - CRYPTO '98, LNCS 1462, str. 137–152, Springer, 1998.

[14] D. Boneh, X. Ding, G. Tsudik in M. Wong, *A Method for Fast Revocation of Public Key Certificates and Security Capabilities*, Proc. of 10th Annual USENIX Security Symposium, 2001.
(dostopno na http://crypto.stanford.edu/~dabo/pubs.html)

[15] R. Canetti, S. Halovi in J. Katz, *A Forward-Secure Public-Key Encryption Scheme*, Advances in Cryptology - EUROCRYPT 2003, LNCS 2656, str. 255–271, Springer, 2003.

[16] A. Menezes, T. Okamoto in S. Vanstone, *Reducing elliptic curve logarithms to logarithms in a fnite field*, IEEE Tran. on Info. Th., Vol. 39, str. 1639–1646, 1993.

[17] A. Joux in K. Nguyen, *Separating Decision Diffe-Hellman from Diffe-Hellman in cryptographic groups*, Journal of Cryptology 16, str. 239–247, 2003.

[18] A. J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, 1993.

[19] L.S Charlap in D. P. Robbins, *An Elementary Introduction to Elliptic Curves*, CRD Expository Report No. 31, 1988.
(dostopno na: http://www.idaccr.org/reports/reports.html)

[20] L.S Charlap in R. Coley, *An Elementary Introduction to Elliptic Curves II*, CRD Expository Report 34, 1990.
(dostopno na: http://www.idaccr.org/reports/reports.html)

[21] M. Maas, *Pairing-Based Cryptography*, Master's Thesis, Technische Universiteit Eindhoven, 2004.
(dostopno na: http://paginas.terra.com.br/informatica/paulobarreto/pblounge.html)

[22] J.H. Silverman, *The Arithmetic of Elliptic Curves*, Springer-Verlag, New York, 1986.

[23] J. D. van den Heiden, *Weil Pairing and the Drinfeld Modular Curve*, Rijksuniversiteit Groningen, 2003.
(dostopno na: http://dissertations.ub.rug.nl/faculties/science/2003/g.j.van.der.heiden/)

[24] S. Micali, *Novomodo: Scalable Certificate Validation and Simplified PKI Management*, Proc. of 1st Annual PKI Research Workshop, 2002.
(dostopno na: http://www.cs.dartmouth.edu/ pki02/Micali/)