Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Peter Nose

Varnostna analiza protokolov za overjen dogovor o ključu in shem za digitalni podpis

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Aleksandar Jurišić

Ljubljana, 2014

Povzetek

Dogovor o ključu in digitalni podpis sta glavna in najbolj uporabna prispevka moderne kriptografije. Takšni protokoli in sheme omogočajo dvema ali več osebam varno izmenjavo skupnega sejnega ključa v prisotnosti zlonamernega prisluškovalca in se uporabljajo za zagotavljanje pristnosti in celovitosti podatkov ter za preprečevanje zanikanja. Zato so varni protokoli za dogovor o ključu in sheme za digitalni podpis osnovni gradniki za sestavo zapletenih, višje nivojskih protokolov.

V tej doktorski disertaciji se ukvarjamo z varnostno analizo že obstoječih protokolov za dogovor o ključu in shem za digitalni podpis. Konkretno preučujemo deset protokolov za dogovor o ključu brez možnosti potrditve, od katerih je pet dvostranskih in pet tristranskih ter eno deterministično shemo za digitalni podpis na osnovi identitete z možnostjo združevanja. S konkretnimi napadi pokažemo, da ti protokoli in shema nimajo želenih varnostnih lastnosti ali varnostnih lastnosti, ki jih trdijo njihovi avtorji. Hkrati predstavimo tudi varno deterministično shemo za digitalni podpis na osnovi identitete, ki podpira delno združevanje.

Najprej pokažemo, da Okamotov in Chenov protokol nista odporna na napad lažnega predstavljanja z razkritim ključem, ter da Tanov, Limov in dva Hölblova protokola niso varni pred napadom lažnega predstavljanja ter napadom vmesne osebe. Nato dokažemo, da je en Hölblov protokol ranljiv za napad notranje osebe, en za deljenje ključa z neznano osebo, v enem protokolu lahko napadalec izračuna zasebni ključ vsakega uporabnika in v enem lahko izračuna skupni sejni ključ. Pokazali bomo tudi, da Selvijeva deterministična shema na osnovi identitete z možnostjo združevanja ni odporna na univerzalne poneverbe, tj. napadalec lahko preprosto sestavi veljavne podpise poljubnih sporočil po svoji izbiri, če ima v povprečju na voljo dvanajst

pristnih digitalnih podpisov.

Na koncu predstavimo še novo deterministično shemo za digitalni podpis na osnovi identitete in njeno varnost formalno dokažemo v modelu naključnega preroka. Slednja je velika izboljšava sheme za digitalni podpis, ki so jo predlagali Selvi in sodelavci, hkrati pa nudi enako tesno varnostno prevedbo na vmesni Diffie-Hellmanov problem. Ker je novo predlagana shema deterministična, omogoča delno združevanje, tj. vse digitalne podpise istega podpisnika je možno združiti v en sam kratek podpis.

Ključne besede: kriptografija na osnovi identitete, dokazljiva varnost, overjen dogovor o ključu, dvostranski, tristranski, tripartitni, digitalni podpis, deterministični podpis, delno združevanje, bilinearno parjenje, model naključnega preroka

Abstract

Key agreement and digital signature are two significant and most useful contributions of modern cryptography. Such protocols and schemes allow two or more parties to establish a common session key securely in the presence of a malicious adversary and provide means of ensuring data origin authentication, data integrity and non-repudiation. Thus, secure key agreement protocols and signature schemes are fundamental building blocks for constructing complex higher-level protocols.

In this thesis, we deal with security analysis of existing key agreement protocols and digital signature schemes. We examine ten authenticated key agreement protocols without key confirmation, out of which half are two-party and the other half are three-party, and one deterministic identity-based aggregate signature scheme. By giving concrete attacks, we show that these protocols and the scheme do not possess the desirable security attributes or security attributes claimed by their authors. We also present a secure deterministic identity-based signature scheme that supports partial aggregation.

First, we show that Okamoto's and Chen et al.'s protocol cannot withstand the key-compromise impersonation attack and that Tan et al.'s, Lim et al.'s and two protocols of Hölbl et al. are insecure against the impersonation attack and the man-in-the-middle attack. Next, we prove that one protocol of Hölbl et al. is vulnerable to the insider attack, one to the unknown key-share attack, one protocol allows an adversary to compute the private key of any user and one protocol allows her to compute the shared session key. We also show that Selvi et. al.'s deterministic identity-based aggregate signature scheme is universally forgeable, i.e., anyone is able to easily generate valid signatures on any messages of his choice having on average twelve genuine digital signatures.

In the end, we introduce a new deterministic identity-based signature scheme and formally prove its security in the random oracle model. The scheme is a major improvement of the signature scheme proposed by Selvi et. al. and offers the same tight security reduction to the underlying gap Diffie-Hellman problem. Because the newly proposed scheme is deterministic it allows partial aggregation, i.e., all the signatures from the same signer can be aggregated into one single short signature.

Keywords: identity-based cryptography, provable security, authenticated key agreement, two-party, three-party, tripartite, signature scheme, deterministic signature, partial aggregation, bilinear pairing, random oracle model

Izjava o avtorstvu doktorske disertacije

Spodaj podpisani **Peter Nose**, z vpisno številko **63040285**, sem avtor doktorske disertacije z naslovom:

Varnostna analiza protokolov za overjen dogovor o ključu in shem za digitalni podpis

S svojim podpisom zagotavljam, da:

- sem doktorsko disertacijo izdelal samostojno pod vodstvom mentorja prof. dr. Aleksandra Jurišića,
- so elektronska oblika doktorske disertacije, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko doktorske disertacije,
- soglašam z javno objavo elektronske oblike doktorske disertacije v zbirki "Dela FRI".

V Ljubljani, dne 17. marec 2014 Podpis avtorja:

Zahvaljujem se mentorju, mami in očetu, vsem sodelavcem v Laboratoriju za kriptografijo in računalniško varnost ter vsem ostalim, ki so kakorkoli pripomogli k izdelavi pričujoče doktorske disertacije.

— Peter Nose, Ljubljana, 2014

Kazalo

Seznam protokolov

Seznam shem

Seznam uporabljenih kratic in simbolov

$\mathbb{N}$	množica naravnih števil
$\mathbb{Z}$	množica celih števil
$\mathbb{R}$	množica realnih števil
$D(a, b)$	največji skupni delitelj števil a in b
$v(a, b)$	najmanjši skupni večkratnik števil a in b
$\varphi(n)$	Eulerjeva φ funkcija
$\mathbb{G}$, $\mathbb{H}$	grupa, podgrupa
$\mathbb{K}$, $\mathbb{L}$	kolobar ali končni obseg
$\overline{\mathbb{K}}$	algebraično zaprtje obsega $\mathbb{K}$
$\mathbb{Z}_n$	kolobar ostankov pri deljenju z $n \in \mathbb{N}$
$\mathbb{Z}_n^*$	grupa obrnljivih elementov kolobarja $\mathbb{Z}_n$
$\mathbb{F}_q$	končen obseg moči $q = p^n$, kjer je p praštevilo in $n \in \mathbb{N}$
$\lvert\mathbb{G}\rvert$, $\lvert p\rvert$	red grupe, dolžina dvojiškega zapisa števila p
$\mathrm{red}(a)$	red elementa a
$\langle a \rangle$	množica vseh potenc elementa a
$\chi_{\mathbb{K}}$	karakteristika kolobarja $\mathbb{K}$
$\#E$	število točk na eliptični krivulji E
$\hat{e}$	bilinearno parjenje
P	verjetnost
κ	varnostni parameter
sk	zasebni ključ (angl. private key)
pk	javni ključ (angl. public key)
msk	glavni tajni ključ (angl. master secret key)
mpk	glavni javni ključ (angl. master public key)

K	sejni ključ
σ	digitalni podpis, skupna skrivnost
Cert	digitalno potrdilo
kdf	funkcija za izpeljavo ključa (angl. key derivation function)
E_k/D_k	šifrirna in odšifrirna funkcija simetrične šifre
$\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, ...	oznake za udeležence protokola (Anita, Bojan, Cene, ...)
$\mathcal{A} \rightarrow \mathcal{B} : m$	Anita pošlje Bojanu sporočilo m
$\mathcal{A} : m \xleftarrow{\$} M$	Anita naključno izbere element m iz množice M
$\mathcal{A} : x \stackrel{?}{=} y$	Anita preveri, če sta vrednosti x in y enaki
$A \leq_p B$	problem A je v polinomskem času prevedljiv na problem B
$A \equiv_p B$	problema A in B sta računsko ekvivalentna v polinomskem času
RSA	kriptosistem, ki so ga razvili Rivest, Shamir in Adleman
DLP	problem diskretnega logaritma
CDH	računski Diffie-Hellmanov problem
DDH	odločitveni Diffie-Hellmanov problem
BDHP	bilinearni Diffie-Hellmanov problem
gapCDH	vmesni Diffie-Hellmanov problem

Poglavje 1

Uvod

"Anyone, from the most clueless amateur to the best cryptographer, can create an algorithm that he himself cannot break."

— *Bruce Schneier*

V zadnjih desetletjih sta razvoj tehnologije in širitev interneta bistveno poenostavila izmenjavo podatkov in komunikacijo med oddaljenimi uporabniki ter napravami. Ob vse večjem številu računalnikov, pametnih telefonov in tablic so zahteve po računalniški in informacijski varnosti postale še toliko bolj pomembne. Med glavna orodja za doseganje slednjih zagotovo sodijo tudi kriptografski protokoli za overjen dogovor o ključu in sheme za digitalni podpis. Brez njih si varne programske opreme in sodobnih internetnih storitev, kot so npr. spletno nakupovanje, e-bančništvo, varna e-pošta in računalništvo v oblaku, sploh ne bi mogli predstavljati.

V okviru te disertacije se bomo osredotočili na varnostno analizo protokolov za overjen dogovor o ključu in shem za digitalni podpis. S konkretnimi napadi na obstoječe protokole in sheme bomo razkrili njihove varnostne pomanjkljivosti ter predlagali možne izboljšave za odpravo le-teh. Hkrati bomo pokazali, da nekatere konstrukcije protokolov in shem niso varne, zato se jih je potrebno v prihodnje izogibati oz. jih ustrezno spremeniti. Razvili bomo tudi novo shemo za digitalni podpis, vpeljali varnostni model in v njem dokazali njeno varnost.

Novim razvijalcem kriptografskih protokolov in shem lahko napadi, opi-

sani v tej disertaciji, služijo kot zgled, kako težko je v resnici sestaviti varen kriptografski protokol oz. shemo. Pri tem se morajo zavedati splošno znanega pravila, ki ga je že davnega leta 1864 postavil Charles Babbage [7], svojo pravo veljavo pa je dobil šele stoletje kasneje, ko ga je v zgoraj zapisani obliki postavil priznani varnostni strokovnjak Bruce Schneier.

1.1 Opis znanstvenega področja

Razvoj protokolov za dogovor o ključu in shem za digitalni podpis sega v leto 1976, ko sta Whitfield Diffie in Martin Hellman objavila revolucionaren koncept kriptografije z javnimi ključi [48]. Le-ta za razliko od simetričnih kriptosistemov ne uporablja enega ključa, temveč dva po funkciji različna, zasebnega in javnega. Prvega vsak uporabnik obdrži zase in ga varno hrani v tajnosti, medtem ko je drugi praviloma javno objavljen in znan vsem uporabnikom sistema. Pri tem velja, da se iz javnega ključa ne da učinkovito izračunati zasebnega. Kadar želimo varno poslati sporočilo nekemu uporabniku, sporočilo zašifriramo z njegovim javnim ključem. Dobljeni tajnopis bo lahko odšifriral le on, saj je edini, ki pozna pripadajoči zasebni ključ. Zaradi razlikovanja med zasebnim in javnim ključem takšno kriptografijo včasih imenujemo tudi asimetrična kriptografija.

Poleg kriptografije z javnimi ključi sta Diffie in Hellman predstavila tudi idejo digitalnega podpisa, s katerim je možno zagotoviti pristnost podatkov in preprečiti možnost njihovega zanikanja. V osnovi je digitalni podpis zelo podoben lastnoročnemu, saj avtor sporočilu doda še nek podatek oz. podpis, s katerim dokazuje, da se strinja z vsebino sporočila. Danes poznamo več vrst digitalnih podpisov, kot so skupinski podpisi [35], slepi podpisi [33], enkratni podpisi [92, 128], fail-stop podpisi [121], prazni podpisi [64], podpisi brez možnosti zanikanja [34], podpisi z možnostjo združevanja [25, 67] itd.

Diffie in Hellman v svojem članku [48] nista podala nobenega praktičnega modela za kriptosistem z javnimi ključi ali za digitalni podpis. Prvi se je pojavil šele dve leti kasneje, ko so Rivest, Shamir in Adleman objavili sistem RSA [129]. Sta pa v svojem delu objavila prvi protokol za dogovor o ključu, s katerim se lahko dve osebi, ki se nista še nikoli srečali, dogovorita za skupni sejni ključ. Kasneje se je izkazalo, da je leta 1974 podobno, vendar nekoliko

manj učinkovito idejo dobil že Ralph Merkle, ko je kot doktorski študent predlagal svojo konstrukcijo, danes znano pod imenom Merklove uganke [106].

Osnovni Diffie-Hellmanov dogovor o ključu je varen pred pasivnimi napadalci (prisluškovalci), ni pa varen pred aktivnimi, ki lahko sporočila prestrezajo, spreminjajo, brišejo in vstavljajo nova. Že dolgo časa je namreč znano, da je protokol ranljiv za napad vmesne osebe, saj ne vsebuje overjanja udeležencev. Za odpravo te pomanjkljivosti so bili v literaturi predlagani različni pristopi. Najpogosteje se protokoli poslužujejo certifikatnih agencij, ki z izdajo digitalnega potrdila jamčijo, da neki javni ključ pripada točno določeni osebi. V takšnih protokolih mora vsak udeleženec pred pričetkom protokola pridobiti digitalna potrdila vseh udeležencev in preveriti njihovo veljavnost. Med takšne protokole sodijo vsi protokoli iz družine MTI [99], trenutno najbolj učinkovit protokol za overjen dogovor o ključu MQV [103] in njegovi različici HMQV [88] ter CMQV [149].

Ideja kriptosistemov z javnimi ključi in certifikatnimi agencijami je požela veliko uspeha, hkrati pa je na mizo prinesla nove probleme, kot je upravljanje s ključi in izdajanje, podaljševanje ter preklicevanje digitalnih potrdil. Da bi se znebili teh težav, je Shamir predlagal nov koncept kriptografije na osnovi identitete [136]. V slednjem se za javni ključ uporabnika uporabi kar njegova identifikacijska informacija, kot je npr. naslov e-pošte ali telefonska številka, zato digitalna potrdila niso več potrebna. Na žalost pa imajo takšni sistemi tudi svoje pomanjkljivosti, saj zasebnih ključev uporabniki ne morejo določiti sami. Te lahko iz njihove identitete in glavnega tajnega ključa izračuna le zaupanja vredna tretja oseba, imenovana generator zasebnih ključev, ki jim jih nato posreduje preko varnega kanala.

Kriptografija javnih ključev in kriptografija na osnovi identitete uporabljata dolge zasebne ključe. Slednji so nepriročni, saj si povprečen človek lahko zapomni le ključe z majhno entropijo. Zato je za shranjevanje le-teh potrebno imeti posebno namensko kriptografsko napravo, kot je npr. pametna kartica ali mobilni telefon. Protokoli za dogovor o ključu na osnovi osebnih gesel teh problemov nimajo, saj si mora vsak uporabnik zapomniti zgolj lažje zapomljivo geslo. In ker vemo, da si uporabniki radi izbirajo šibka gesla, morajo biti ti protokoli odporni na napade s slovarjem. Prvi takšen protokol, imenovan šifrirana izmenjava ključa, sta leta 1992 prestavila Bellovin in

Merritt [19].

Po objavi Diffie-Hellmanovega protokola se je v literaturi pojavilo mnogo novih protokolov za dogovor o ključu. Glede na število aktivnih udeležencev jih v grobem delimo na dvostranske in večstranske protokole. Poseben primer slednjih so tristranski protokoli, katere je možno učinkovito sestaviti z uporabo bilinearnih parjenj. Prvi takšen protokol je leta 2000 predstavil Joux [76]. Slednji je strukturno zelo podoben Diffie-Hellmanovem dogovoru o ključu, zato ni overjen in je ranljiv za napad vmesne osebe.

V razvoju kriptografije z javnimi ključi je bilo predlaganih že veliko kriptosistemov, shem in protokolov, med katerimi so nekateri v uporabi še danes. Po drugi strani pa so nekateri predlogi skozi čas pokazali svoje ranljivosti in so bili posledično razbiti. Da se takšne stvari v prihodnje ne bi ponavljale, je zaželeno, da vsak nov predlog, poleg natančnega opisa in analize učinkovitosti, vsebuje tudi nekakšno zagotovilo o njegovi varnosti. Vsak avtor naj bi zato v svojem predlogu natančno opredelil, katerim varnostnim kriterijem kriptosistem, shema ali protokol zadošča in na kakšne vrste napadov je odporen.

Sodobna kriptografija temelji na dokazljivi varnosti, s katero želimo varnostne lastnosti kriptografskih rešitev utemeljiti z dokazom [82, 18, 60]. Najpogostejši način za dokazovanje slednjih je znanstveno korekten dokaz, ki varnost rešitve znotraj nekega varnostnega modela prevede na nek težak računski problem. S tem dokažemo, da če napadalec odkrije pomanjkljivost kriptografske rešitve in s tem razbije njeno varnost, potem lahko isti napadalec reši izbran računski problem. Vendar, ker naj slednji ne bi bil rešljiv v doglednem času, štejemo rešitev med varne.

Dokazi varnosti so odličen pripomoček za zagotavljanje varnosti novih kriptografskih predlogov, vendar pa imajo tudi ti svoje pomanjkljivosti. Običajno so dokazi dolgi in težko berljivi, zato le redko kdo preverja njihovo pravilnost. Tudi napake v njih so pogoste, povrhu vsega pa jih je običajno še zelo težko odkriti. Naslednji problem so pomanjkljive definicije varnostnih modelov, v katerih dokazujemo varnost. Velikokrat se je namreč že zgodilo, da je bil dokaz znotraj modela pravilen, vendar slednji ni zajemal vseh možnih napadov, ki jih lahko v praksi uporabi napadalec. Najpogostejši primer takšnih napadov so napadi s stranskim kanalom, saj je slednje v splošnem zelo težko

matematično formulirati. Zato je potrebno dokaze varnosti pazljivo obravnavati in dobro razumeti njihove prednosti ter slabosti. Nekatere izmed njih sta z malce drugačnim pogledom na dokaze varnosti izpostavila Koblitz in Menezes [87, 86, 85].

1.2 Prispevki k znanosti

V doktorski disertaciji predstavimo naslednja izvirna prispevka k znanosti.

- *Kriptoanaliza obstoječih protokolov in shem.* Uspešno smo opravili kriptoanalizo petih dvostranskih in petih tristranskih protokolov za overjen dogovor o ključu (na osnovi identitete), ter ene sheme za digitalni podpis. S konkretnimi napadi znotraj ustreznih varnostnih modelov smo dokazali, da predlagani protokoli in shema ne izpolnjujejo želenih varnostnih kriterijev oz. kriterijev, ki so jih zastavili njihovi avtorji, in zato niso varni. S tem smo hkrati tudi avtorje v splošnem opozorili, na kakšne napake morajo biti pozorni pri sestavi novih predlogov. Varnostne pomanjkljivosti smo podrobno preučili in predlagali izboljšave za odpravo le-teh.

- *Predlog sheme za digitalni podpis.* Sestavili smo novo, izboljšano deterministično shemo za digitalni podpis na osnovi identitete z možnostjo delnega združevanja. Njeno učinkovitost smo analizirali na podlagi računske, prostorske in komunikacijske zahtevnosti ter jo primerjali z osnovno shemo. Nova shema je varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom in identiteto. Varnost smo dokazali v modelu naključnega preroka s prevedbo na vmesni Diffie-Hellmanov problem. To pomeni, da shema ustreza najvišjim varnostnim kriterijem, ki jih običajno zahtevamo od shem za digitalni podpis.

Prvi prispevek je bil objavljen v dveh znanstvenih člankih z naslovom *Security weaknesses of authenticated key agreement protocols* [112] in *Security weaknesses of a signature scheme and authenticated key agreement protocols* [114], medtem ko je članek *Improved deterministic identity-based signature scheme* [113] z drugim navedenim prispevkom v pripravi.

1.3 Pregled disertacije

Doktorska disertacija je organizirana na naslednji način. V drugem poglavju predstavimo teoretične osnove, potrebne za razumevanje protokolov in shem obravnavanih v tem delu. Definicije in izreke iz tega poglavja bomo uporabljali pri opisu napadov na protokole ter shemo in za dokazovanje varnosti predlagane sheme za digitalni podpis. Poglavje začnemo z matematičnim razdelkom 2.1, v katerem predstavimo osnovne pojme iz teorije računske zahtevnosti, kot sta polinomski algoritem in zanemarljiva funkcija, ter na kratko predstavimo teorijo števil, algebraične strukture in njihove lastnosti, eliptične krivulje ter bilinearna parjenja. Sledi razdelek 2.2, v katerem predstavimo težke računske probleme, na katerih naj bi temeljila varnost obravnavanih protokolov in shem. Poglavje nadaljujemo s predstavitvijo kriptografije javnih ključev v razdelku 2.3 in ga zaključimo z dokazi varnosti v razdelku 2.4. V slednjem vpeljemo tri kriptografske gradnike, ki se običajno uporabljajo pri konstrukciji varnih protokolov in shem, ter opišemo dva varnostna modela za dokazovanje varnosti.

Tretje poglavje je namenjeno predstavitvi protokolov za overjen dogovor o ključu, katero začnemo z definicijo osnovnih pojmov v razdelku 3.1. Nadalje razdelamo varnostne cilje in lastnosti, ki naj bi jih varni protokoli dosegali oz. imeli, ter opišemo osnovne napade na njih v razdelku 3.2. Sledi opis različnih vrst protokolov za dogovor o ključu, kot so dvostranski, tristranski, večstranski protokoli in protokoli na osnovi identitete, ter predstavitev njihovih najpomembnejših predstavnikov v razdelku 3.3.

V četrtem poglavju se spoznamo s shemami za digitalni podpis, ki jih najprej definiramo v razdelku 4.1. Nato predstavimo različne vrste poneverb digitalnih podpisov in možne napade na sheme, med katerimi bo za nas najpomembnejši prilagodljiv napad z izbranim sporočilom in identiteto, saj bomo z njim določili zmogljivost napadalca v dokazu varnosti naše sheme. Razdelek 4.2 zaključimo z definicijo različnih varnostnih modelov shem za digitalni podpis. Na koncu si v razdelku 4.3 ogledamo še različne vrste shem, kot so verjetnostne in deterministične sheme, sheme na osnovi identitete ter sheme z možnostjo združevanja.

V petem poglavju predstavimo deset protokolov za dogovor o ključu in

eno shemo za digitalni podpis. Za vsak predlog izvedemo varnostno analizo in s konkretnimi napadi pokažemo, da ne ustreza varnostnim zahtevam ter zato ni varen. Začnemo z opisom Okamotovega dvostranskega in Chenovega tristranskega protokola za dogovor o ključu v razdelku 5.1 in 5.2. Za oba protokola dokažemo, da nista varna pred napadom lažnega predstavljanja z razkritim ključem. Nato v razdelkih 5.3–5.5 opišemo Tanov in Limov tristranski protokol ter dva Hölblova dvostranska protokola in razkrijemo napad lažnega predstavljanja ter napad vmesne osebe na vsakega izmed njih. Sledijo štirje razdelki 5.6–5.9, v katerih predstavimo še dva dvostranska in dva tristranska Hölblova protokola za dogovor o ključu. Tudi ti protokoli niso varni, saj lahko v prvem napadalec izračuna zasebni ključ vsakega uporabnika, drugi protokol ni odporen na napad deljenja ključa z neznano osebo, v tretjem lahko napadalec izračuna skupni sejni ključ in četrti protokol ni odporen na napade lažnega predstavljanja, če je napadalec notranja oseba. Poglavje zaključimo z opisom Selvijeve sheme za digitalni podpis in s predstavitvijo napada, s katerim lahko napadalec ustvari univerzalne poneverbe.

Šesto poglavje je v celoti namenjeno predstavitvi nove sheme za digitalni podpis na osnovi identitete z možnostjo delnega združevanja. Shemo gradimo postopoma, tako da najprej v razdelku 6.1 predstavimo verjetnostno shemo za digitalni podpis. Slednjo v razdelku 6.2 nadgradimo v shemo na osnovi identitete in jo v razdelku 6.3 spremenimo v deterministično. Poglavje zaključimo z razdelkom 6.4, v katerem opišemo postopek, s katerim lahko deterministično shemo dopolnimo do sheme z možnostjo delnega združevanja. Za vsako verzijo sheme dokažemo tudi njeno pravilnost in varnost v modelu naključnega preroka.

V zadnjem poglavju na kratko povzamemo vsebino disertacije, predstavimo možnosti za nadaljnje delo, podamo sklepne misli in zaključimo disertacijo.

Poglavje 2

Osnovni pojmi

Glavni cilj tega poglavja je predstaviti osnovne pojme, potrebne za razumevanje doktorske disertacije. Začnemo z malce bolj obsežno predstavitvijo matematičnih osnov s področij teorije računske zahtevnosti, teorije števil, abstraktne algebre in do neke mere tudi s področja algebraične geometrije. Nadaljujemo z vpeljavo težkih računskih problemov in predstavitvijo kriptografije javnih ključev. V zaključku poglavja se dotaknemo še področja dokazljive varnosti, kjer predstavimo tri pogosta kriptografska orodja za sestavo protokolov in shem ter dva varnostna modela za dokazovanje njihove varnosti.

2.1 Matematično ozadje

V tem razdelku bomo predstavili matematično ozadje, potrebno za razumevanje kriptografskih protokolov in shem, obravnavanih v tem delu ter njihove varnostne analize. Navedli bomo glavne pojme iz teorije števil, različne algebraične strukture in njihove osnovne lastnosti. Definirali bomo tudi eliptične krivulje in vpeljali bilinearna parjenja, ki v sodobni kriptografiji igrajo zelo pomembno vlogo.

2.1.1 Teorija računske zahtevnosti

Klasična teorija računske zahtevnosti nam omogoča, da lahko računske probleme klasificiramo glede na njihovo težavnost in razrede med seboj pri-

merjamo. Za določitev težavnosti problema običajno uporabimo količino računskih virov in sredstev, ki jih potrebujemo za njegovo rešitev (npr. porabljen čas, prostor, energijo). Klasifikacija ni odvisna od izbranega računskega modela temveč zgolj od velikostnega reda problema.

V nadaljevanju bomo definirali osnovne termine, ki jih bomo pogosto uporabljali v prihodnjih poglavjih, in kriterije za primerjavo učinkovitosti protokolov ter shem. Snov je delno povzeta po delu avtorjev Menezes, Oorschot in Vanstone [105] ter Hölbl [69].

Definicija 2.1. *Algoritem* je končno zaporedje natančno določenih pravil, operacij oz. ukazov, ki sprejme vhodne podatke in ob koncu izvajanja vrne rezultat.

Algoritem je *deterministično*, če ob sprejetju poljubnih vhodnih podatkov vedno izvede enako zaporedje ukazov in vrne enak rezultat. Nasprotno je algoritem *verjetnosten*, če se izvajanje algoritma spremeni skorajda vsakič, ko ga poženemo z istimi vhodnimi podatki. Verjetnostni algoritmi za svoje delovanje uporabljajo naključne bite, ki vplivajo na potek izvajanja in posledično na vrnjen rezultat.

Definicija 2.2. *Velikost* vhodnih podatkov je število bitov, ki jih potrebujemo za njihovo predstavitev v binarnem zapisu.

Definicija 2.3. *Čas izvajanja* algoritma ob sprejetju nekih vhodnih podatkov je število osnovnih operacij oz. korakov, ki jih algoritem izvede, preden zaključi svoje izvajanje.

Definicija 2.4. *Najslabši čas izvajanja* algoritma je zgornja meja za čas izvajanja glede na poljubne vhodne podatke določene velikosti, izražena kot funkcija velikosti vhodnih podatkov.

Definicija 2.5. *Pričakovani čas izvajanja* algoritma je povprečni čas izvajanja za vse možne vhodne podatke določene velikosti, izražen kot funkcija velikosti vhodnih podatkov.

Običajno je težko natančno določiti čas izvajanja algoritma, zato si v takšnih primerih raje pomagamo z njegovo približno oceno. Ker nas običajno

zanima zgolj velikostni red izvajanja operacij, tj. za koliko se poveča čas izvajanja algoritma, če povečamo velikost vhodnih podatkov, si lahko pomagamo z asimptotičnim časom izvajanja in z O-notacijo.

Definicija 2.6. Za funkciji f in g velja $f(k) = O(g(k))$, če obstaja taka pozitivna konstanta c in naravno število k_0, da za vsa števila $k > k_0$ velja $0 \leq f(k) \leq cg(k)$.

Definicija 2.7. Algoritem je *polinomski*, če je njegova časovna zahtevnost v najslabšem primeru enaka $O(k^c)$, kjer je k velikost vhodnih podatkov in c neka konstanta. V nasprotnem primeru je algoritem nepolinomski oz. *eksponenten*.

Definicija 2.8. Funkcija $\varepsilon(k) : \mathbb{N} \to \mathbb{R}$ je *zanemarljiva* glede na parameter k, če za vsako realno število $c \geq 0$ obstaja naravno število k_c, da za vsak $k > k_c$ velja $\varepsilon(k) < k^{-c}$.

Asimptotično štejemo polinomske algoritme za učinkovite in eksponentne za neučinkovite. Za problem, ki ga ne znamo rešiti v polinomskem času, pravimo, da je *težko rešljiv* oz. *nerešljiv v doglednem času*. Podobno štejemo dogodke, katerih verjetnost je določena z zanemarljivo funkcijo, za malo verjetne oz. skorajda *nemogoče*.

Časovno zahtevnost algoritmov merimo na podlagi nekega parametra κ. V kriptografiji slednjemu pravimo *varnosti parameter*, saj z njim običajno določimo dolžino ključev in velikosti algebrskih grup. Z večanjem varnostnega parametra lahko torej vplivamo na varnost kriptosistema, sheme ali protokola. Kriptografski algoritmi lahko varnostni parameter kot vhodni podatek sprejmejo v različnih oblikah. Najpogosteje parameter κ zapišemo v obliki niza enic dolžine κ, ki ga označimo z 1^κ, medtem ko algoritmi manj pogosto sprejmejo varnosti parameter κ direktno kot število. V tem delu bomo uporabljali prvi zapis.

Za primerjavo učinkovitosti protokolov za dogovor o ključu in shem za digitalni podpis je potrebno definirati ustrezne kriterije. Pri shemah je pomembna predvsem računska in prostorska zahtevnost, saj nas najbolj zanima hitrost izračuna ter preverjanja digitalnega podpisa in njegova dolžina. Pri primerjavi protokolov za dogovor o ključu pa je dodatno potrebno upoštevati še komunikacijsko zahtevnost.

- *Računska zahtevnost.* V kriptografiji računske zahtevnosti protokolov in shem običajno ne merimo glede na čas izvajanja na določenem procesorju, temveč nas zanima zgolj število računskih operacij, ki jih je potrebno izvesti v posamezni fazi. V tej disertaciji bomo operacije grobo razdelili v tri razrede. Med manj zahtevne, ki jih lahko običajno kar zanemarimo, bomo uvrstili modularno seštevanje, odštevanje in množenje. Med srednje zahtevne operacije po naraščajoči zahtevnosti sodijo modularno potenciranje in deljenje, seštevanje in odštevanje točk na eliptični krivulji ter izračun zgoščevalne funkcije. Zelo zahtevne operacije, ki se jih v praksi želimo čimbolj izogibati, pa so množenje točke s skalarjem in izračun bilinearnega parjenja.

- *Prostorska zahtevnost.* Tudi pri prostorski zahtevnosti nas ne zanima konkretna poraba prostora v bitih, saj je ta močno odvisna od izbranega varnostnega parametra. Zato običajno velikosti javnih, zasebnih in glavnih tajnih ključev ter digitalnih podpisov podamo s številom shranjenih elementov grupe oz. z velikostnim redom shranjenih bitnih nizov in števil. Pri porabi prostora nas včasih zanima tudi število dodatnih funkcij, npr. zgoščevalnih ali psevdonaključnih funkcij, ki jih protokol oz. shema uporablja za svoje delovanje.

- *Komunikacijska zahtevnost.* Pri protokolih za dogovor o ključu je pomembna tudi komunikacijska zahtevnost. Tu nas za vsako fazo zanima število izmenjanih sporočil in njihova velikost, koliko sporočil je bilo poslanih vsem udeležencem hkrati in v koliko krogih se lahko zaključi posamezna faza. Pri tem en krog sestavljajo vsa sporočila, ki so lahko poslana in prejeta vzporedno znotraj neke časovne enote.

2.1.2 Teorija števil

Nekoč je znani matematik Carl Friedrich Gauss dejal, da je matematika kraljica znanosti in teorija števil kraljica matematike. Imel je prav. To področje, ki se v glavnem ukvarja s preučevanjem lastnosti celih števil, je tako lepo, preprosto in koristno, da se zdi njegova raba skorajda brezmejna. Dosežki iz teorije števil se množično uporabljajo tudi v kriptografiji, zato bomo ne-

katere izmed njih predstavili v nadaljevanju. V podrobno analizo izrekov in trditev se ne bomo spuščali, raje bomo dokaze kar preskočili. Zainteresiran bralec jih lahko poišče v standardnih učbenikih teorije števil avtorjev Grasselli [62], Vidav [152] in Plemelj [122] ter kriptografski knjigi avtorjev Menezes, Vanstone in Oorschot [105], od koder je povzeta tudi snov.

Množico vseh celih števil $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$ bomo označili z $\mathbb{Z}$ in množico naravnih števil $\{z \in \mathbb{Z} \mid z > 0\}$ z $\mathbb{N}$.

Definicija 2.9. Število $a \in \mathbb{Z}$ *deli* število $b \in \mathbb{Z}$, če obstaja tak $k \in \mathbb{Z}$, da velja $b = ka$. Pišemo $a \mid b$. Število a, ki deli b, je *faktor* števila b, če je $a > 0$ in $a \notin \{1, b\}$.

Izrek 2.1 (Deljenje z ostankom)**.** *Naj bo $a \in \mathbb{Z}$ in $b \in \mathbb{N}$. Potem obstajata enolični števili $k, r \in \mathbb{Z}$, za kateri velja $a = kb + r$ in $0 \leq r < b$.*

Naj bodo a, b, k in r oznake iz prejšnjega izreka. Število a imenujemo *deljenec*, b *delitelj*, k *kvocient* in r *ostanek*. Ostanek pri deljenju označimo z $a \bmod b$ in kvocient z a/b.

Definicija 2.10. Število a je *praštevilo*, če se ga ne da zapisati kot produkt faktorjev, tj. če se ga ne da faktorizirati. Sicer je *sestavljeno* število.

Definicija 2.11. Število $d \in \mathbb{N}$ je *največji skupni delitelj* števil $a, b \in \mathbb{Z}$, če $d \mid a$, $d \mid b$ in za vsako število $z \in \mathbb{Z}$, ki deli a in b, velja $z \mid d$. Največji skupni delitelj označimo z $D(a, b)$.

Definicija 2.12. Število $v \in \mathbb{N}$ je *najmanjši skupni večkratnik* števil $a, b \in \mathbb{Z}$, če $a \mid v$, $b \mid v$ in za vsako število $z \in \mathbb{Z}$, ki je deljivo z a in b, velja $v \mid z$. Najmanjši skupni večkratnik označimo z $v(a, b)$.

Največji skupni delitelj in najmanjši skupni večkratnik sta povezana z enačbo $ab = D(a, b)\, v(a, b)$.

Izrek 2.2. *Na bo $a, b \in \mathbb{N}$. Potem obstajata celi števili x in y, da velja $ax + by = D(a, b)$. Največji skupni delitelj je najmanjše naravno število, ki se ga da zapisati v tej obliki.*

Algoritem 1 Razširjen Evklidov algoritem

Vhod: $a \geq b$

Izhod: $d = D(a,b),\ ax + by = d$

procedura EVKLID(a,b)
  $u = a,\ v = b$
  $x_1 = 1,\ x_2 = 0,\ y_1 = 0,\ y_2 = 1$
  dokler $v > 0$ **ponavljaj**
    $k = \lfloor u/v \rfloor,\ r = u - kv,\ x = x_1 - kx_2,\ y = y_1 - ky_2$
    $u = v,\ v = r,\ x_1 = x_2,\ x_2 = x,\ y_1 = y_2,\ y_2 = y$
  $d = u,\ x = x_1,\ y = y_1$
  vrni (d, x, y)

Za poljubni celi števili a in b lahko njun največji skupni delitelj poiščemo z *razširjenim Evklidovim algoritmom*, ki ima kvadratno časovno zahtevnost. Z njim lahko izračunamo tudi vrednosti x in y iz prejšnjega izreka. Algoritem 1 prikazuje psevdokodo razširjenega Evklidovega algoritma.

Definicija 2.13. Celi števili a in b sta *tuji*, če velja $D(a,b) = 1$.

Izrek 2.3 (Osnovni izrek aritmetike)**.** *Vsako naravno število $n > 1$ lahko razcepimo v produkt oblike*

$$n = p_1^{e_1}\, p_2^{e_2}\, \cdots\, p_k^{e_k},$$

kjer so p_i različna praštevila in e_i naravna števila. Razcep je do vrstnega reda faktorjev enoličen.

Definicija 2.14. Naj bo $n \in \mathbb{N}$ in $\varphi(n)$ število naravnih števil manjših od n, ki so tuja številu n. Preslikavo $\varphi : \mathbb{N} \to \mathbb{N}$ imenujemo *Eulerjeva funkcija*.

Izrek 2.4 (Lastnosti Eulerjeve funkcije)**.**

1. *Naj bo p praštevilo. Potem je $\varphi(p) = p - 1$.*
2. *Če sta a in b tuji naravni števili, je $\varphi(ab) = \varphi(a)\varphi(b)$.*
3. *Naj bo $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ praštevilski razcep števila n. Potem je*

$$\varphi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_k}\right).$$

Definicija 2.15. Celi števili a in b sta *kongruentni* po modulu $n \in \mathbb{N}$, če dajeta enak ostanek pri deljenju z n, kar zapišemo kot $a \equiv b \pmod{n}$.

Izrek 2.5. *Relacija $\equiv$ je ekvivalenčna relacija.*

Definicija 2.16. Ekvivalenčni razred števila a je množica vseh celih števil, ki so kongruentna a po modulu n. Množico predstavnikov ekvivalenčnih razredov $\{0, 1, 2, \ldots, n-1\}$ označimo z $\mathbb{Z}_n$.

Definicija 2.17. *Multiplikativni inverz* števila $a \in \mathbb{Z}_n$ po modulu n je število $x \in \mathbb{Z}_n$, za katerega velja $ax \equiv 1 \pmod{n}$. Če tak x obstaja, potem pravimo, da je število a *obrnljivo*, in inverz označimo z a^{-1}.

Izrek 2.6. *Multiplikativni inverz števila $a \in \mathbb{Z}_n$ po modulu n je enoličen.*

Definicija 2.18. Množico obrnljivih elementov po modulu n označimo z $\mathbb{Z}_n^*$. Velja $\mathbb{Z}_n^* = \{a \in \mathbb{Z}_n \mid D(a, n) = 1\}$.

Izrek 2.7 (Euler)**.** *Naj bo $n \geq 2$ celo število. Če $a \in \mathbb{Z}_n^*$, potem*

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

Eulerjev izrek je pravzaprav posplošitev Fermatovega malega izreka, ki se ga pogosto izpelje že v srednji šoli. Ta pravi, da za vsako celo število a in praštevilo p velja $a^p \equiv a \pmod{p}$.

2.1.3 Abstraktna algebra

Naslednje področje matematike, ki je zelo uporabno v kriptografiji, je algebra. Njene abstraktne strukture s svojimi lepimi lastnostmi so idealne za sestavo varnih kriptosistemov, protokolov in shem. Mednje sodita tudi grupa in obseg, ki ju bomo tudi bolj podrobno predstavili. Snov tega podrazdelka je povzeta iz učbenikov za algebro avtorjev Vidav [152] in Plemelj [122], iz najbolj brane kriptografske knjige avtorjev Menezes, Vanstone ter Oorschot [105] in iz dveh knjig o končnih obsegih, ki so ju izdali Lidl ter Niederreiter [95] in Menezes s sodelavci [102]. V njih lahko bralec najde tudi vse dokaze izrekov, ki jih bomo v nadaljevanju preskočili.

Definicija 2.19. Naj bo M neka neprazna množica. *Binarna operacija* na tej množici je preslikava $\circ : M \times M \to M$, ki vsakemu urejenemu paru $(a, b) \in M \times M$ priredi natanko en element $a \circ b \in M$.

Če je nad množico M definirana vsaj ena operacija, pravimo da ima množica *algebraično strukturo*.

Definicija 2.20. Naj bo $\circ : \mathbb{G} \times \mathbb{G} \to \mathbb{G}$ binarna operacija na množici $\mathbb{G}$. Algebraična struktura $(\mathbb{G}, \circ)$ je *grupa*, če:

- je $\circ$ *notranja* operacija, torej za poljubna $a, b \in \mathbb{G}$ velja $a \circ b \in \mathbb{G}$,
- je $\circ$ *asociativna* operacija, torej za $a, b, c \in \mathbb{G}$ velja $(a \circ b) \circ c = a \circ (b \circ c)$,
- obstaja *enota* e, da za poljuben $a \in \mathbb{G}$ velja $a \circ e = e \circ a = a$,
- za vsak $a \in \mathbb{G}$ obstaja *inverz* b, da velja $a \circ b = b \circ a = e$.

Grupa je komutativna oziroma *Abelova*, če:

- je $\circ$ *komutativna* operacija, torej za vse $a, b \in \mathbb{G}$ velja $a \circ b = b \circ a$.

Kadar je binarna operacija na množici $\mathbb{G}$ seštevanje, pravimo da je grupa *aditivna*. Operacijo označimo s $+$, enoto za seštevanje z 0 in inverz elementa $a \in \mathbb{G}$ z oznako $-a$. Podobno notacijo lahko uvedemo tudi za *multiplikativne* grupe, na katerih je definirana operacija množenje. Pri slednjih operacijo označimo z $*$, enoto za množenje z 1 in inverz elementa $a \in \mathbb{G}$ z $a^{-1} = \frac{1}{a}$.

V nadaljevanju bomo splošno znana dejstva o grupah vedno zapisali v multiplikativni notaciji, čeprav slednja, z ustrezno zamenjavo notacije, veljajo za poljubno grupo. Kadar bo binarna operacija $\circ$ jasna iz konteksta, pa bomo grupo $(\mathbb{G}, \circ)$ pogosto skrajšano označili z $\mathbb{G}$.

Definicija 2.21. Število elementov v grupi $\mathbb{G}$, oznaka $|\mathbb{G}|$, imenujemo tudi *red grupe*. Grupa je *končna*, če ima končni red.

V multiplikativni grupi lahko dodatno definiramo še operacijo potenciranja. Naj bo $k \in \mathbb{Z}$ in $a \in \mathbb{G}$. Potem je k-ta potenca elementa a definirana kot

$$a^k = \underbrace{a * a * \cdots * a}_{k-\text{krat}}.$$

Množico vseh potenc elementa a označimo z $\langle a \rangle = \{a^k : k \in \mathbb{N}\}$.

Definicija 2.22. Element $a \in \mathbb{G}$ ima *končni red*, če je množica $\langle a \rangle$ končna. V tem primeru je red elementa a enak moči te množice, tj. najmanjše naravno število n, za katerega velja $a^n = 1$. Če množica $\langle a \rangle$ ni končna, pravimo da je element a *reda nič*. Red elementa a označimo z $\mathrm{red}(a)$.

Definicija 2.23. Grupa $\mathbb{G}$ je *ciklična*, če obstaja tak element $g \in \mathbb{G}$, da velja $\langle g \rangle = \mathbb{G}$. Takemu elementu pravimo *generator* grupe.

Definicija 2.24. Naj bo $\mathbb{H}$ neprazna podmnožica grupe $\mathbb{G}$, ki vsebuje enoto. Potem je $\mathbb{H}$ *podgrupa* grupe $\mathbb{G}$, če:

- za poljubna $a, b \in \mathbb{H}$ velja $a * b \in \mathbb{H}$,
- za vsak $a \in \mathbb{H}$ velja $a^{-1} \in \mathbb{H}$.

Izrek 2.8 (Lagrange)**.** *Naj bo $\mathbb{G}$ končna grupa in $\mathbb{H}$ njena podgrupa. Potem red grupe $\mathbb{H}$ deli red grupe $\mathbb{G}$. Posledično, red vsakega elementa $a \in \mathbb{G}$ deli red grupe $\mathbb{G}$.*

Vse grupe, obravnavane v tem delu, bodo končne in komutativne. Najpogosteje bomo uporabljali grupe $\mathbb{Z}_n$, $\mathbb{Z}_n^*$ in $\mathbb{E}$, opisane v nadaljevanju. Grupa $\mathbb{Z}_n = (\mathbb{Z}, +_n)$ označuje množico celih števil z operacijo seštevanja po modulu n. Red takšne grupe je n, saj grupo sestavljajo elementi $0, 1, \ldots, n-1$. Če operacijo seštevanja zamenjamo z operacijo množenja po modulu n, potem ni nujno, da imajo vsi elementi svoj inverz. Množico vseh obrnljivih elementov označimo z $\mathbb{Z}_n^* = (\mathbb{Z}^*, *_n)$ in jo sestavljajo vsi elementi, ki so tuji številu n. Iz definicije 2.14 sledi, da je red takšne grupe enak $\varphi(n)$. Zadnja pomembna grupa je aditivna grupa točk na eliptični krivulji $\mathbb{E}$, katero bomo bolj podrobno spoznali v nadaljevanju.

Definicija 2.25. Algebraična struktura $(\mathbb{K}, +, *)$ je *kolobar*, če:

- je $(\mathbb{K}, +)$ Abelova grupa,
- je $*$ asociativna operacija z enoto 1, ki se razlikuje od enote 0 za operacijo $+$,
- za poljubne elemente $a, b, c \in \mathbb{K}$ velja *distributivnost*, torej

$$a * (b + c) = a * b + a * c,$$
$$(a + b) * c = a * c + b * c.$$

Kolobar $(\mathbb{K}, +, *)$ bomo v nadaljevanju skrajšano označevali s $\mathbb{K}$, operaciji $+$ in $*$ pa bomo poimenovali seštevanje in množenje. Kolobar $\mathbb{K}$ je *komutativen*, če je množenje komutativno.

Naj bo $\mathbb{K}$ kolobar. Pravimo, da je element $a \in \mathbb{K}$ *obrnljiv*, če obstaja element $b \in \mathbb{K}$, da velja $a * b = b * a = 1$. Takemu elementu pravimo *inverz* in ga označimo z oznako a^{-1}. Množico vseh obrnljivih elementov iz kolobarja $\mathbb{K}$ označimo z $\mathbb{K}^*$.

Če v kolobarju $\mathbb{K}$ za neničelna elementa $a, b \in \mathbb{K}$ velja $a * b = 0$, potem je a *levi delitelj niča* in b *desni delitelj niča*. V komutativnem kolobarju so zaradi komutativnosti operacije $*$ levi in desni delitelji niča enaki in jim pravimo *delitelji niča*.

Definicija 2.26. *Karakteristika* kolobarja $\mathbb{K}$, oznaka $\chi(\mathbb{K})$, je najmanjše število $n \in \mathbb{N}$, za katerega velja $\underbrace{1 + 1 + \ldots + 1}_{n\text{-krat}} = 0$, in 0, če tak n ne obstaja.

Definicija 2.27. Komutativnemu kolobarju z enoto 1, kjer je vsak neničeln element obrnljiv, pravimo *obseg*.

Izrek 2.9. *V obsegu ni deliteljev niča.*

Izrek 2.10. *Karakteristika obsega je lahko* 0 *ali p, kjer je p praštevilo.*

Obseg je *končen*, če vsebuje končno mnogo elementov. Končni obseg sestavlja množica elementov $\mathbb{K}$ in binarni operaciji seštevanje ter množenje, ki sta definirani na njej. Z njima in inverzom lahko definiramo še dve novi operaciji.

- *Odštevanje.* Elementa $a, b \in \mathbb{K}$ odštejemo tako, da elementu a prištejemo *negativni element* od b. Nasprotni element $-b \in \mathbb{K}$ je enoličen element, za katerega velja $b + (-b) = 0$. Operacijo odštevanja zapišemo kot $a - b = a + (-b)$.

- *Deljenje.* Elementa $a, b \in \mathbb{K}$, kjer $b \neq 0$, delimo tako, da element a pomnožimo z inverznim elementom od b. Recipročni element $b^{-1} \in \mathbb{K}$ je enoličen element, za katerega velja $b * b^{-1} = 1$. Operacijo deljenja zapišemo kot $a/b = a * b^{-1}$.

V nadaljevanju bomo končni obseg s q elementi označevali s $\mathbb{F}_q$.

Izrek 2.11. *Naj bo p praštevilo, $\mathbb{Z}_p = \{0, 1, 2, \ldots, p-1\}$, $+_p$ operacija seštevanja in $*_p$ operacija množenja po modulu p. Potem je $(\mathbb{Z}_p, +_p, *_p)$ končni obseg. Označili ga bomo z $\mathbb{Z}_p$.*

Definicija 2.28. Naj bosta $(\mathbb{K}, +, *)$ in $(\mathbb{K}', \oplus, \otimes)$ dva kolobarja. Preslikava $\psi : \mathbb{K} \to \mathbb{K}'$ je *homomorfizem kolobarjev*, če za vse elemente $a, b \in \mathbb{K}$ velja:

- $\psi(a + b) = \psi(a) \oplus \psi(b)$,
- $\psi(a * b) = \psi(a) \otimes \psi(b)$,
- $\psi(1) = 1$.

Če sta $\mathbb{K}$ in $\mathbb{K}'$ obsega, preslikavo ψ imenujemo *homomorfizem obsegov*.

Definicija 2.29. Naj bosta $\mathbb{K}$ in $\mathbb{L}$ dva obsega. Pravimo, da je $\mathbb{L}$ *razširitev obsega* $\mathbb{K}$, oznaka $\mathbb{L}/\mathbb{K}$, če obstaja homomorfizem obsegov $\psi : \mathbb{L} \to \mathbb{K}$.

Definicija 2.30. Obseg $\mathbb{K}$ je *algebraično zaprt*, če je v njem možno vsak nekonstantni polinom $p(x) \in \mathbb{K}[x]$ zapisati kot produkt samih linearnih faktorjev. To pomeni, da obstajajo elementi $k, a_1, \ldots, a_n \in \mathbb{K}$, da velja:

$$p(x) = k \prod_{i=1}^{n} (x - a_i).$$

Izrek 2.12 (Steinitz)**.** *Obstaja enolična razširitev obsega $\mathbb{K}$, ki je algebraično zaprta. Takšni razširitvi pravimo algebraično zaprtje in jo označimo s $\overline{\mathbb{K}}$.*

2.1.4 Eliptične krivulje

Teorija eliptičnih krivulj je eno izmed bolj razvitih področij matematike. Eliptične krivulje so predmet raziskovanja na mnogih področjih, kot so algebraična geometrija, teorija števil, teorija kodiranja in kriptografija. Uporabljene so v različnih algoritmih za praštevilski razcep števil [94], za preverjanje praštevilskosti [6], za konstrukcijo psevdonaključnih števil [80] in enosmernih permutacij [81], uporabljajo pa se tudi pri kodah za odpravljanje

napak [55] in v kriptografiji z javnimi ključi [78]. V slednjo sta jih leta 1985 neodvisno vpeljala Neal Koblitz [84] in Victor Miller [108].

Večino kriptografskih protokolov in shem je možno definirati nad poljubno grupo. Vendar pa je Abelova grupa točk na eliptični krivulji primernejša od ostalih, saj trenutno ni znan še noben algoritem za reševanje problema diskretnega logaritma, ki ne bi imel eksponentne časovne zahtevnosti.

V nadaljevanju bomo na kratko opisali eliptične krivulje in nad njimi definirali grupo. Snov tega podrazdelka je povzeta po knjigi o eliptičnih krivuljah avtorja Vidava [153] in po kriptografskih knjigah s tega področja avtorjev Menezes, Vanstone ter Oorschot [105], Cohen ter Frey [44], Menezes [101] in Hankerson, Menezes ter Vanstone [63].

Definicija 2.31. Naj bo $\mathbb{K}$ obseg in $\overline{\mathbb{K}}$ njegovo algebraično zaprtje. *Afina Weierstrassova enačba* eliptične krivulje nad obsegom $\mathbb{K}$, označimo jo z $E/\mathbb{K}$, je enačba oblike

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, \tag{2.1}$$

kjer so $a_1, a_2, a_3, a_4, a_6 \in \mathbb{K}$ takšni koeficienti, da je v vsaki točki $(x_1, y_1) \in \overline{\mathbb{K}} \times \overline{\mathbb{K}}$, ki je rešitev enačbe (2.1), vsaj eden od parcialnih odvodov po x in y neničelen, tj. $2y_1 + a_1x_1 + a_3 \neq 0$ ali $3x_1^2 + 2a_2x_1 + a_4 - a_1y_1 \neq 0$.

Zadnji pogoj pravi, da je eliptična krivulja *nesingularna* oz. *gladka*. To pomeni, da krivulja v nobeni točki nima dveh ali več tangent. Če takšna točka obstaja, pravimo da je krivulja *singularna*. Nesingularnost lahko definiramo tudi z diskriminanto eliptične krivulje.

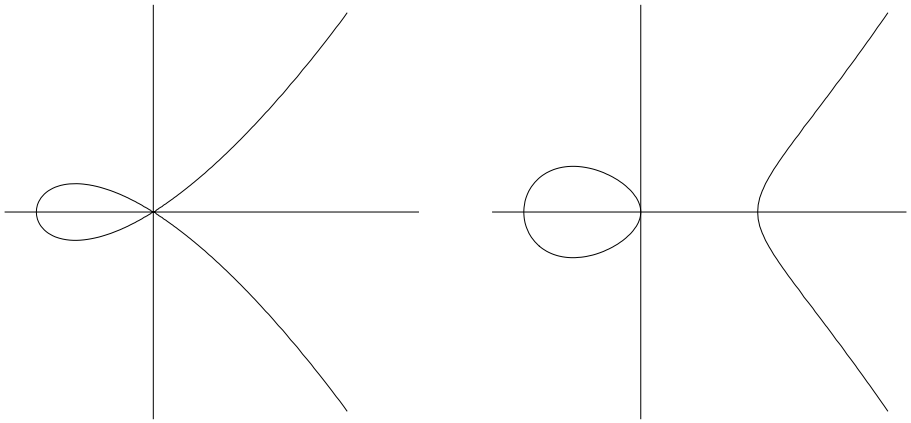
Definicija 2.32. Naj bo E eliptična krivulja definirana z enačbo (2.1) in

$$\begin{aligned} d_2 &= a_1^2 + 4a_2, & d_3 &= 2a_4 + a_1a_3, \\ d_6 &= a_3^2 + 4a_6, & d_8 &= a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^2. \end{aligned}$$

Tedaj vrednosti

$$\Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6$$

pravimo *diskriminanta eliptične krivulje* E. Eliptična krivulja je nesingularna, če je $\Delta \neq 0$.



(a) Singularna krivulja (b) Nesingularna krivulja

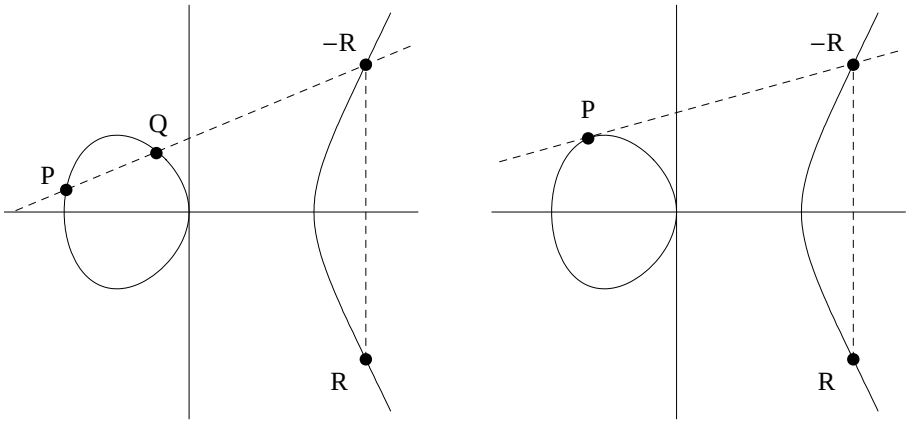
Slika 2.1: Eliptični krivulji $y^2 = x^3 + x^2$ in $y^2 = x^3 - x$ nad $\mathbb{R}$

Množico točk na eliptični krivulji $E/\mathbb{K}$ lahko spremenimo v Abelovo grupo, če na njej definiramo komutativno operacijo seštevanja. Slednjo bomo definirali s pravilom sekant in tangent ter jo označili s $+$. Operacijo seštevanja si najlažje prestavljamo geometrijsko na eliptični krivulji nad realnimi števili.

Različni točki $P = (x_1, y_1)$ in $Q = (x_2, y_2)$ na eliptični krivulji E seštejemo tako, da naprej skozi njiju potegnemo premico. Slednja seka krivuljo E še v natanko eni točki $-R = (x_3, -y_3)$. Če to točko preslikamo čez os x, dobimo vsoto $R = P + Q = (x_3, y_3)$. Podoben pristop lahko uporabimo tudi za *podvajanje* točke, tj. primer, ko je $P = Q$, le da namesto sekante vzamemo tangento na krivuljo v tej točki.

Zgoraj opisani postopek seštevanja je rahlo pomanjkljiv, saj z njim ne moremo sešteti dveh različnih točk $P = (x_1, y_1)$ in $Q = (x_1, y_2)$ z enako koordinato x. Namreč, premica skozi njiju je vzporedna osi y in zato ne seka eliptične krivulje v nobeni drugi točki. Ta problem rešimo s točko v neskončnosti $\mathcal{O}$, ki bo enota za seštevanje. Predstavljali si bomo, da leži neskončno daleč na y-osi in da se vse navpične premice sekajo v njej. Premica skozi točki P in Q torej poteka skozi točko $\mathcal{O}$ in zato velja $P + Q = \mathcal{O}$. Od tod tudi sledi, da je točka Q inverz oz. nasprotni element točke P.

Na prvi pogled zgleda, da geometrijska definicija seštevanja velja samo za eliptične krivulje nad $\mathbb{R}$, vendar pa se da seštevanje zapisati tudi z algebraičnimi formulami za poljuben obseg $\mathbb{K}$. Naj bosta $P = (x_1, y_1)$ in



(a) Seštevanje $P + Q = R$ (b) Podvajanje $P + P = R$

Slika 2.2: Geometrijska predstavitev seštevanja in podvajanja točk na eliptični krivulji

$Q = (x_2, y_2)$ točki na eliptični krivulji. Potem velja

$$P + \mathcal{O} = \mathcal{O} + P = P, \qquad P + (-P) = \mathcal{O}$$

in

$$\begin{aligned} -P &= (x_1, -y_1 - a_1x_1 - a_3) \\ P + Q &= (\lambda^2 + a_1\lambda - a_2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1 - a_1x_3 - a_3), \end{aligned}$$

kjer je

$$\lambda = \begin{cases} \dfrac{y_1 - y_2}{x_1 - x_2}, & \text{če } P \neq \pm Q, \\ \dfrac{3x_1^2 + 2a_2x_1 + a_4 - a_1y_1}{2y_1 + a_1x_1 + a_3}, & \text{če } P = Q. \end{cases}$$

Definicija 2.33. *Množenje točke* P s skalarjem n je preslikava $\mathbb{N} \times E \to E$, ki je definirana kot

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

Definicijo lahko posplošimo tudi na vsa števila $n \in \mathbb{Z}$, če privzamemo $0P = \mathcal{O}$ in $nP = (-n)(-P)$ za $n < 0$.

Definicija 2.34. Eliptični krivulji E_1 in E_2 nad obsegom $\mathbb{K}$, definirani z Weierstrassovima enačbama

$$E_1 : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,$$
$$E_2 : y^2 + a_1'xy + a_3'y = x^3 + a_2'x^2 + a_4'x + a_6',$$

sta *izomorfni*, če obstajajo $u, r, s, t \in \mathbb{K}$, $u \neq 0$, da zamenjava spremenljivk

$$(x, y) \to (u^2x + r, u^3y + u^2sx + t)$$

spremeni enačbo E_1 v E_2.

Naj bo $E/\mathbb{K}$ eliptična krivulja definirana z enačbo (2.1). Tedaj lahko z zamenjavo spremenljivk Weierstrassovo enačbo poenostavimo v

- $y^2 + xy = x^3 + ax^2 + b$ ali $y^2 + xy = x^3 + ax + b$, če je $\chi(\mathbb{K}) = 2$,
- $y^2 = x^3 + ax^2 + b$ ali $y^2 = x^3 + ax + b$, če je $\chi(\mathbb{K}) = 3$ in v
- $y^2 = x^3 + ax + b$ sicer.

Hkrati lahko poenostavimo tudi formule za seštevanje in podvajanje točke.

Definicija 2.35. Naj bo $\mathbb{L}$ razširitev obsega $\mathbb{K}$. Tedaj so $\mathbb{L}$-*racionalne točke* na eliptični krivulji urejeni pari $(x, y) \in \mathbb{L} \times \mathbb{L}$, kjer x in y rešita enačbo (2.1), skupaj s točko v neskončnosti $\mathcal{O}$. Množico vseh takšnih točk označimo z $E(\mathbb{L})$.

Naslednji izrek nam približno oceni število vseh točk na eliptični krivulji.

Izrek 2.13 (Hasse)**.** *Naj bo E eliptična krivulja nad obsegom $\mathbb{F}_q$ in $\#E(\mathbb{F}_q)$ oznaka za število njenih točk. Potem za neki $t \in \mathbb{Z}$, kjer je $|t| \leq 2\sqrt{q}$, velja*

$$\#E(\mathbb{F}_q) = q + 1 - t.$$

Definicija 2.36. Naj bo P točka na eliptični krivulji E nad obsegom $\mathbb{F}_q$ reda n. Potem število $k = \#E(\mathbb{F}_q)/n$ imenujemo *kofaktor* eliptične krivulje.

V kriptografiji običajno uporabljamo eliptične krivulje, ki vsebujejo točke velikega praštevilskega reda in imajo zelo majhen kofaktor, običajno 1, 2 ali 4. S tem pospešimo računske operacije na krivulji in preprečimo napade, ki izkoriščajo obstoj majhnih podgrup.

2.1.5 Bilinearna parjenja

Bilinearne preslikave oz. parjenja igrajo v moderni kriptografiji zelo pomembno vlogo, saj lahko z njimi ciklične grupe opremimo z dodatnimi lastnostmi. Te nam omogočajo sestavo elegantnih protokolov in shem, kot so tristranski dogovor o ključu, šifriranje in dogovor o ključu na osnovi identitete, ter digitalni podpis z možnostjo združevanja.

V kriptografiji so bilinearna parjenja prvi uporabili Fray, Menezes, Okamoto in Vanstone za izvedbo MOV napada [104], ki problem diskretnega logaritma nad določenimi razredi eliptičnih krivulj prenese na problem diskretnega logaritma v podgrupi multiplikativne grupe $\mathbb{Z}_n^*$. Pravo veljavo v kriptografiji pa so bilinearna parjenja dobila šele leta 2000, ko sta Boneh in Franklin objavila prvo šifrirno shemo na osnovi identitete, ki uporablja Weilovo parjenje. Od takrat je bilo predlagano mnogo kriptosistemov, protokolov in shem na osnovi parjenj, tako da je danes to področje izjemno aktivno in razširjeno.

Definicija 2.37. Naj bo $\mathbb{G}_1 = \langle P \rangle$ aditivna grupa praštevilskega reda p in $\mathbb{G}_2$ multiplikativna grupa istega reda. Simetrično *bilinearno parjenje* na $(\mathbb{G}_1, \mathbb{G}_2)$ je preslikava

$$\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$$

z naslednjimi lastnostmi.

- *Bilinearnost*: $\hat{e}(aQ, bR) = \hat{e}(Q, R)^{ab}$ za vse elemente $Q, R \in \mathbb{G}_1$ in za vsa števila $a, b \in \mathbb{Z}_p^*$.
- *Neizrojenost*: $\hat{e}(P, P) \neq 1$, kjer smo z 1 označili identiteto grupe $\mathbb{G}_2$.
- *Izračunljivost*: za vsaka elementa $P, Q \in \mathbb{G}_1$ lahko učinkovito izračunamo $\hat{e}(P, Q)$.

Bilinearno parjenje je natančno določeno z vrednostjo $\hat{e}(P, P)$, zato obstaja natanko $p - 1$ parjenj. Slednja so enaka do konstante natančno, tj. če sta $\hat{e}_1$ in $\hat{e}_2$ parjenji, potem obstaja nek $c \in \mathbb{Z}_p^*$, da velja $\hat{e}_1(P, Q) = \hat{e}_2(P, Q)^c$ za vse $P, Q \in \mathbb{G}_1$. To pa pomeni, da v resnici obstaja natanko eno parjenje na $(\mathbb{G}_1, \mathbb{G}_2)$.

Poiskati učinkovito bilinearno parjenje je težko, ni pa nemogoče. Trenutno sta znani dve konstrukciji, ki sta ju predlagala Tate [146, 54, 53] in Weil [154, 109]. V obeh primerih za $\mathbb{G}_1$ vzamemo grupo točk na ustrezni eliptični krivulji in za $\mathbb{G}_2$ multiplikativno grupo končnega obsega.

2.2 Težki računski problemi

Varnost protokolov in shem, obravnavanih v tej disertaciji, naj bi temeljila na težkih računskih problemih, ki jih bomo spoznali v tem razdelku. Ti problemi so znani že dolgo časa in so bili intenzivno preučeni s strani raziskovalcev na področju kriptografije, matematike ter računalništva in ostalih strokovnjakov širom sveta. Trenutno najboljši znani algoritmi za njihovo reševanje imajo eksponentno časovno zahtevnost, zato jih štejemo med težke, kljub temu da njihove prave časovne zahtevnosti v resnici sploh ne poznamo. Lahko pa vse te težke probleme med seboj primerjamo. V ta namen so bile razvite in predlagane posebne tehnike, s katerimi lahko algoritem, ki reši prvi problem, pretvorimo v algoritem, ki reši drugega. Vse to je pripeljalo do pojma *polinomska prevedba*, katerega opišemo v naslednji definiciji.

Definicija 2.38. Naj bosta A in B računska problema. Pravimo, da se problem A *polinomsko prevede* na problem B, oznaka $A \leq_p B$, če obstaja polinomski algoritem za reševanje problema A, ki kot vhod sprejme hipotetičen algoritem za reševanje problema B. Če velja $A \leq_p B$ in $B \leq_p A$, pravimo, da sta problema *računsko ekvivalentna* in to označimo z $A \equiv_p B$.

Zgornja definicija nam pravzaprav pove, da če se problem A polinomsko prevede na problem B, potem je problem B vsaj tako težek kot problem A oz. problem A ni težji od problema B. Če je problem A težko rešljiv, potem lahko s polinomsko prevedbo dokažemo, da je težko rešljiv tudi problem B. Polinomske prevedbe so zato zelo uporabne pri dokazovanju varnosti shem in protokolov, saj lahko varnost slednjih prevedemo na nek težko rešljiv računski problem. Tako dobimo zagotovilo, da če lahko napadalec razbije shemo oz. protokol, potem lahko slednji reši tudi izbran računski problem.

2.2.1 Problem RSA

Najprej bomo spoznali probleme, ki so povezani z razcepom števil na prafaktorje in z iskanjem e-tih korenov po modulu nekega sestavljenega števila. Na teh problemih temelji tudi kriptosistem RSA, ki je danes še vedno najpogosteje uporabljani kriptosistem, čeprav ga počasi zamenjujejo kriptosistemi z eliptičnimi krivuljami. Leta 1978 so ga predlagali Rivest, Shamir in Adleman [129], uporablja pa se ga lahko tako za šifriranje sporočil kot za digitalno podpisovanje.

Definicija 2.39 (Problem razcepa)**.** Naj bo n naravno število. Poišči praštevilski razcep števila n, tj. razcepi n v produkt oblike $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$, kjer so p_i različna praštevila in e_i naravna števila.

Definicija 2.40 (Problem RSA)**.** Naj bo n sestavljeno naravno število, c celo število in e celo število tuje $\varphi(n)$. Poišči celo število m, za katerega velja $m^e \equiv c \pmod{n}$.

Zadnji problem lahko malce poenostavimo, če reševalcu dovolimo prosto izbiro javnega eksponenta e [8].

Definicija 2.41 (Krepak problem RSA)**.** Naj bo n sestavljeno naravno število in c celo število. Poišči liho število $e \geq 3$ in celo število m, za kateri velja $m^e \equiv c \pmod{n}$.

V kriptografski skupnosti velja splošno prepričanje, da naj bi bil problem razcepa računsko ekvivalenten problemu RSA, čeprav za to nimamo nobenega dokaza. Trenutno znamo dokazati le naslednjo povezavo med zgoraj naštetimi problemi:

$$\text{Krepak problem RSA} \leq_p \text{Problem RSA} \leq_p \text{Problem razcepa.}$$

2.2.2 Problem diskretnega logaritma

V tem podrazdelku bomo definirali problem diskretnega logaritma in Diffie-Hellmanov problem ter njegove različice, na katerih temelji varnost mnogih kriptografskih protokolov in shem. Med njimi je najbolj znan Diffie-Hellmanov dogovor o ključu [48] in njegove inačice, ter ElGamalova shema za šifriranje ter podpisovanje [50].

Problemi predstavljeni v nadaljevanju so definirani, če ni drugače rečeno, nad končnimi cikličnimi grupami $\mathbb{G}$ (pisanimi multiplikativno) praštevilskega reda n z generatorjem α. V kriptografiji se običajno uporabljajo podgrupe multiplikativne grupe končnega obsega $\mathbb{F}_q$ ali aditivne podgrupe točk na eliptični krivulji $E(\mathbb{F}_q)$.

Definicija 2.42 (Problem diskretnega logaritma - DLP)**.** Naj bo $\mathbb{G}$ končna ciklična grupa reda n in α njen generator. Za naključno izbrani element $\beta \in \mathbb{G}$ poišči število $x \in \mathbb{Z}_n$, za katerega velja $\beta = \alpha^x$.

Število x je enolično določeno, saj sta ciklični grupi $\mathbb{G}$ in $\mathbb{Z}_n$ istega reda in zato izomorfni [31, Thm. 9.5.5]. Pravimo mu *diskretni logaritem* elementa β z osnovo α v grupi $\mathbb{G}$, kar zapišemo z oznako $x = \log_\alpha \beta$. Izračun diskretnega logaritma v $\mathbb{Z}_n^*$ je tesno povezan s problem razcepa, kar prikazuje tudi neenakost

$$\text{Problem razcepa} \leq_p \text{DLP v } \mathbb{Z}_n^*.$$

Pravimo, da ima algoritem A *prednost* ε pri reševanju problema diskretnega logaritma v $\mathbb{G}$, če velja

$$P[A(\alpha, \beta) = \log_\alpha \beta] \geq \varepsilon,$$

kjer verjetnost izračunamo glede na vse možne izbire elementov $\alpha, \beta \in \mathbb{G}$ in vse možne naključne bite algoritma A, ki jih slednji uporablja.

Predpostavka 2.1 (Predpostavka diskretnega logaritma)**.** Noben polinomski algoritem A nima nezanemarljive prednosti glede na parameter $\kappa = \log n$ pri reševanju problema diskretnega logaritma.

Za problem diskretnega logaritma je preprosto pokazati, da je njegova težavnost neodvisna od izbire generatorja. To pomeni, da algoritem, ki zna izračunati diskretni logaritem z osnovo α, lahko izračuna tudi diskretni logaritem z osnovo γ, če je γ generator grupe $\mathbb{G}$.

Problem diskretnega logaritma lahko posplošimo tudi za poljubno neciklično grupo, kjer element α ni nujno generator.

Definicija 2.43 (Problem diskretnega logaritma - DLP)**.** Naj bo $\mathbb{G}$ končna grupa in $\alpha, \beta \in \mathbb{G}$ dva njena elementa. Poišči celo število x, za katerega velja $\beta = \alpha^x$, če tak x obstaja.

Učinkovito izračunljive grupe, v katerih je problem diskretnega logaritma težak, igrajo zelo pomembno vlogo v kriptografiji. Kljub temu pa do danes še ni bilo dokazano, ali takšne grupe sploh obstajajo. In ker to vprašanje ostaja težak odprt problem v kriptografiji, se v praksi uporabljajo grupe, v katerih s trenutnimi najboljšimi algoritmi v doglednem času ni možno rešiti problema diskretnega logaritma.

Algoritme za reševanje problema diskretnega logaritma delimo v dve kategoriji. V prvo sodijo splošni algoritmi, ki znajo rešiti problem diskretnega logaritma v poljubni grupi. Victor Shoup [140] je dokazal, da imajo takšni algoritmi v najboljšem primeru časovno zahtevnost enako $O(\sqrt{p})$, kjer je p največje praštevilo, ki deli red grupe. Najbolj znane takšne metode so Pollardova ρ in λ (kenguru) metoda [125, 126], ter Shanksova metoda mali-veliki korak [137]. V drugo kategorijo uvrščamo ne splošne algoritme, ki pri reševanju problema diskretnega logaritma upoštevajo konkretno predstavitev elementov grupe. Takšni algoritmi imajo načeloma boljšo časovno zahtevnost od splošnih, vendar pa je njihovo delovanje omejeno na točno določene grupe. Dobro poznani primeri iz te kategorije so Pohlig–Hellmanov algoritem [123], metoda Index calculus [1] in številsko sito [2, 3].

2.2.3 Diffie-Hellmanov problem in njegove različice

Predpostavka diskretnega logaritma običajno ni dovolj za dokazovanje varnosti kriptografskih protokolov in shem. Zato so se pojavili tudi drugi, malce lažji problemi. Med slednje sodita računski in odločitveni Diffie-Hellmanov problem [48, 23], ki sta tesno povezana s problemom diskretnega logaritma.

Definicija 2.44 (Računski Diffie-Hellmanov problem - CDH)**.** Naj bo $\mathbb{G}$ končna ciklična grupa praštevilskega reda n, α njen generator in $\alpha^a, \alpha^b \in \mathbb{G}$ dva njena elementa. Poišči element $\alpha^{ab} \in \mathbb{G}$.

Definicija 2.45 (Odločitveni Diffie-Hellmanov problem - DDH)**.** Naj bo $\mathbb{G}$ končna ciklična grupa praštevilskega reda n, α njen generator in $\alpha^a, \alpha^b, \beta \in \mathbb{G}$ trije njeni elementi. Ugotovi ali velja $\beta = \alpha^{ab}$.

Za omenjene tri probleme je že dolgo časa znana povezava

$$\text{DDH} \leq_p \text{CDH} \leq_p \text{DLP}.$$

Pri tem moramo poudariti, da neenakosti v obratni smeri (vedno) ne veljajo. Obstajajo namreč grupe, v katerih je DDH problem lahek, medtem ko sta problema DLP in CDH še vedno težka. Primer takšne grupe je grupa, katere velikost je deljiva z nekim majhnim praštevilom. V takšni grupi je lahko izračunati diskretni logaritem po modulu tega praštevila, vendar pa nam to nič ne pomaga pri reševanju DLP in CDH problema. Po drugi strani pa lahko s tem znanjem rešimo DDH problem z občutno večjo verjetnostjo kot 50%, katero dosežemo z naivnim ugibanjem [77]. Naj omenimo še, da obstajajo primeri grup, v katerih sta DLP in CDH problema računsko ekvivalentna. V modelu splošne grupe, tj. v grupi, kjer ne obstaja noben nesplošen algoritem za reševanje DLP problema, pa je bilo dokazano, da so vsi trije problemi računsko ekvivalentni [140].

Naslednji problem, ki ga bomo definirali, je osnova za dokazovanje varnosti mnogih shem in protokolov, ki temeljijo na bilinearnih parjenjih. Med slednje sodijo (hierarhične) sheme za šifriranje in digitalni podpis na osnovi identitete, protokoli za tristranski dogovor o ključu itd.

Definicija 2.46 (Bilinearni Diffie-Hellmanov problem - BDHP)**.** Naj bo $\mathbb{G}_1 = \langle P \rangle$ aditivna in $\mathbb{G}_2$ multiplikativna ciklična grupa praštevilskega reda n ter $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ bilinearno parjenje. Za naključno izbrane elemente $aP, bP, cP \in \mathbb{G}_1$ poišči element $\hat{e}(P, P)^{abc} \in \mathbb{G}_2$.

Bilinearni Diffie-Hellmanov problem [76] se pojavi pri Weilovem ali Tatovem parjenju, ki ju lahko konstruiramo iz super-singularnih eliptičnih krivulj ali Abelovih algebrskih varietet. S temi parjenji pa lahko konstruiramo tudi nove grupe, v katerih je problem DDH lahek, medtem ko je CDH težak in verjetno še vedno računsko ekvivalenten problemu DLP [77]. Takšnim grupam pravimo *vmesne grupe* in na slednjih lahko definiramo naslednji računski problem.

Definicija 2.47 (Vmesni Diffie-Hellmanov problem - gapCDH)**.** Če imamo na voljo hipotetičen polinomski algoritem A za reševanje odločitvenega Diffie-Hellmanovega problema v grupi $\mathbb{G}$, reši računskega.

Za vmesni Diffie-Hellmanov problem [120] ni težko pokazati, da ni težji od računskega, oz. da velja neenakost

$$\text{gapCDH} \leq_p \text{CDH}.$$

2.3 Kriptografija javnih ključev

Kriptologija je znanstvena veda, ki jo v splošnem delimo na medsebojno odvisni področji kriptografijo in kriptoanalizo. *Kriptografija* preučuje matematične tehnike, povezane s cilji informacijske varnosti, in se ukvarja z načrtovanjem ter razvojem varnih kriptosistemov, protokolov in shem, medtem ko se *kriptoanaliza* osredotoča na razbijanje le-teh. Med glavne cilje kriptografije sodijo:

- *zaupnost* (angl. confidentiality), s katero želimo ohraniti podatke tajne pred nepooblaščenimi osebami,
- *celovitost* (angl. data integrity), ki nudi zagotovilo, da podatki niso bili spremenjeni,
- *overjanje* (angl. authentication), kjer preverjamo verodostojnost (izvor) in pristnost (identiteto pošiljatelja) podatkov,
- *preprečitev tajenja* (angl. non-repudiation), s katero lahko preprečimo neizpolnitev sprejetih obvez ali dejanj.

Ravno nasprotno želi kriptoanaliza z ustreznim napadom dosego omenjenih ciljev preprečiti. Najpogosteje si prizadeva poiskati napad, s katerim je možno iz tajnopisov razbrati čistopise, poneveriti digitalni podpis poljubnega sporočila, razkriti dogovorjeni sejni ključ itd. Z uspešnim napadom tako dokaže, da kriptografska rešitev ne zadošča zastavljenim varnostim kriterijem in zato ni varna.

V splošnem kriptografijo delimo na *simetrično* in *asimetrično*. Simetrični običajno pravimo *klasična kriptografija*, medtem ko asimetrično imenujemo tudi *kriptografija javnih ključev*. Prva veja kriptografije je starejša in zato tudi bolje poznana. V njej si par ali skupina oseb deli *tajni* oz. *skrivni ključ*, ki pa ne sme biti razkrit drugim osebam. Za ključ si lahko izberejo število, besedo ali niz naključnih znakov in ga uporabijo v različnih kriptografskih orodjih. Med njimi so najbolj znane simetrične šifre (npr. DES, AES), tokovne šifre (npr. RC4, Trivium) in kode za overjanje podatkov (npr. HMAC).

2.3.1 Terminologija in notacija

Naslednje razlikovanje je pogosto med kriptografskim algoritmom, protokolom in shemo.

Definicija 2.48. *Kriptografski algoritem* je končno zaporedje natančno določenih pravil, operacij oz. ukazov, ki sprejme vhodne podatke in ob koncu izvajanja vrne rezultat ter pri tem doseže določen varnostni cilj v končnem številu korakov.

Definicija 2.49. *Kriptografski protokol* je porazdeljen algoritem sestavljen iz zaporedja korakov, ki natančno opisujejo ukrepe, ki jih morajo udeleženci protokola izvesti za dosego določenih varnostnih ciljev.

Definicija 2.50. *Kriptografska shema* je zbirka sorodnih kriptografskih algoritmov in/ali protokolov, s katero je možno doseči določene varnostne cilje.

Primer kriptografskega algoritma je algoritem za šifriranje sporočil. Ta kot vhod sprejme čistopis in tajni ključ, izvede končno število korakov ter ob koncu izvajanja vrne tajnopis. Tega lahko odšifriramo z drugim algoritmom, ki poleg tajnopisa sprejme tudi tajni ključ ter vrne pripadajoči čistopis. Če oba algoritma združimo, dobimo šifrirno shemo, s katero je možno doseči tajnost sporočil, enega izmed glavnih kriptografskih ciljev. Med kriptografske protokole pa na primer uvrščamo protokol SSL/TLS [47], ki se uporablja za varno komunikacijo preko spleta. Ta algoritem je namreč porazdeljen, saj natančno določa, katere korake mora opraviti odjemalec na eni strani in strežnik na drugi za vzpostavitev varne povezave.

V kriptografiji udeležence protokola običajno počlovečimo in jih poimenujemo z imeni, ki se začnejo s črkami iz začetka abecede, kot so Anita, Bojan, Cene itd. V opisih protokolov se na njih nanašamo, kot da so dejansko ljudje, čeprav v resnici ni nujno, da so. Udeleženec protokola je lahko med drugimi tudi mobilni telefon, računalnik, program, ki teče na nekem računalniku, brezžični usmerjevalnik ali TV sprejemnik. Pri tem se morda lastnik naprave sploh ne zaveda, kaj se v resnici dogaja v ozadju oz. vsaj ne vseh korakov protokola. Osebi, ki želi udeležencem preprečiti dosego varnostnega cilja, pravimo napadalec in jo bomo poimenovali Oskar. Tudi napadalec ni nujno oseba in je v resnici lahko samodejni program, kriminalna organizacija

ali obveščevalna agencija. Kadar je napadalec hkrati tudi udeleženec protokola, mu pravimo *notranji napadalec*. Napadalec ima lahko pod nadzorom nekatere udeležence protokola, katere lahko na svojo stran pridobi npr. s podkupnino ali izsiljevanjem. Takšnim udeležencem pravimo, da so *zlonamerni*, preostalim pa, da so *pošteni*. Slednji strogo sledijo korakom protokola, saj v njem sodelujejo, ker želijo doseči neki varnostni cilj. Pri dokazovanju varnosti protokolov in shem lahko nastopi še dodatna oseba, ki v varnostnem modelu preko igre da napadalcu izziv. Izzivalca v varnostni igri bomo imenovali Iztok, njegova naloga pa je, da uporabi Oskarjevo znanje pri reševanju nekega težkega računskega problema.

Nekateri protokoli za svoje delovanje uporabljajo *zaupanja vredno osebo*, ki je običajno vnaprej določena s strani vseh udeležencev in lahko tudi aktivno sodeluje v protokolu. Zaupanja vredna oseba nima nobenega osebnega interesa v protokolu in naj ne bi bila zvesta nobenemu udeležencu. Zaupajo ji vsi udeleženci protokola, zato je vse, kar reče, resnično, vse, kar izračuna, pravilno in vedno opravi svojo nalogo v protokolu. Zaupanja vredna oseba je v kriptografskih protokolih nadomestek za odvetnike, notarje, sodnike, učitelje, zdravnike itd. Razdelimo jih lahko v več razredov glede na njihovo vlogo v protokolu. *Certifikatne agencije* so zaupanja vredne osebe, ki osebam izdajajo digitalna potrdila oz. certifikate. Kadar zaupanja vredna oseba v protokolu izdaja in vzdržuje mehanizem za preklic zasebnih ključev uporabnikov, ji pravimo *generator zasebnih ključev*. Zadnji razred zaupanja vrednih oseb so *overitveni strežniki*, katere protokoli običajno uporabljajo za preverjanje pristnosti podatkov in identifikacijo oseb.

Udeleženci protokola si med seboj izmenjujejo sporočila preko komunikacijskega kanala, ki omogoča pošiljanje sporočil enemu udeležencu ali vsem hkrati. Kanal ni zaščiten, če lahko vsakdo bere, vstavlja in briše pošiljke ali spreminja njihov vrstni red. Takšnemu kanalu pravimo *javni kanal*. Nasprotno nudi *zasebni kanal* zaščito pred prisluškovanjem, zaradi katere napadalec ne more niti brati niti spreminjati sporočil. Zasebni kanal lahko takšno zaščito doseže z uporabo kriptografije (npr. sporočila so šifrirana, digitalno podpisana) ali pa je kanal sam po sebi fizično zaščiten (npr. sporočila so izmenjana osebno na nekem varnem mestu, sporočila prenašajo zanesljivi kurirji).

Napadalec lahko protokol poskuša razbiti na različne načine. Napadom, v katerih zgolj prisluškuje izmenjanim sporočilom, pravimo *pasivni napadi*.

V slednjih napadalec ne vpliva na izvedbo protokola, temveč želi s prisluškovanjem pridobiti čim več podatkov za kriptoanalizo. Ker je pasivne napade zelo težko odkriti, protokoli takšne napade namesto odkrivanja raje preprečijo. *Aktivni napadi* so veliko bolj nevarni, saj pri slednjih napadalec aktivno sodeluje v protokolu. Napadalec lahko prestreza, spreminja in briše izmenjana sporočila, se predstavlja za neko drugo osebo, pošilja lažna sporočila, deloma ali v celoti ne sledi protokolu itd. Nadzira lahko tudi zlonamerne udeležence in njihove informacije uporabi za razbijanje protokola.

Protokoli obravnavani v tem delu so zapisani kot zaporedje korakov. Izvajanje protokola poteka po korakih, kjer vsak korak opisuje izračune, ki jih morajo opraviti udeleženci, ali sporočila, ki si jih morajo med seboj izmenjati. Udeleženci v protokolu, torej Anita, Bojan in Cene, so skrajšano označeni z velikimi črkami $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$, medtem ko so zaupanja vreden strežnik, generator zasebnih ključev ter napadalec Oskar označeni s črkami $\mathcal{S}$, $\mathcal{G}$ in $\mathcal{O}$. V opisih protokolov in shem se uporablja naslednja notacija:

- $\mathcal{A} \to \mathcal{B} : m$ – Anita pošlje Bojanu sporočilo m,
- $\mathcal{A} : x \stackrel{?}{=} y$ – Anita preveri, če sta vrednosti x in y enaki,
- $\mathcal{A} : x \stackrel{\$}{\leftarrow} M$ – Anita naključno izbere element x iz množice M z enakomerno porazdelitvijo.

2.3.2 Infrastruktura javnih ključev

Največja težava simetričnih kriptosistemov je izmenjava tajnega ključa preko varnega kanala. V praksi je običajno to težko storiti, predvsem, če se osebi nahajata na zelo oddaljenih lokacijah. Poleg tega pa mora vsaka oseba po izmenjavi ključa pri sebi varno hraniti veliko število ključev, tj. enega za vsako osebo, s katero želi komunicirati. Na primer, osebni zdravnik bi moral na svoji pametni zdravstveni kartici hraniti tajne ključe vseh svojih sedanjih pacientov kot tudi bodočih.

Leta 1976 sta Diffie in Hellman predstavila idejo kriptografije javnih ključev [48], ki odpravi veliko težav simetrične kriptografije. Za razliko od

slednje, kriptografija javnih ključev uporablja dve vrsti ključev, *javne* in *zasebne ključe*. Zasebni ključ vsaka oseba varno hrani pri sebi, medtem ko lahko pripadajoči javni ključ javno objavi, saj naj njegova objava ne bi ogrožala varnosti zasebnega. V nekaterih sistemih je lahko zasebni ključ uporabnika znan tudi tretji osebi, ki ji vsi uporabniki zaupajo.

Glavna prednost kriptografije javnih ključev je, da lahko osebe med seboj varno komunicirajo, ne da bi si predhodno izmenjale skupni tajni ključ oz. se kadarkoli tajno srečale. Za varno komunikacijo je dovolj, da vsaka oseba pridobi zgolj javne ključe ostalih oseb. Kljub temu pa je potrebno javne ključe pred uporabo ustrezno preveriti. Kriptografija javnih ključev tako problem izmenjave tajnih ključev prenese na problem izmenjave overjenih javnih ključev.

Varnostno infrastrukturo, ki je namenjena distribuciji in upravljanju z javnimi ključi, imenujemo *infrastruktura javnih ključev*. V nadaljevanju bomo spoznali dve takšni infrastrukturi. Prva temelji na uporabi digitalnih potrdil, medtem ko druga za svoje delovanje uporablja uporabnikovo identiteto.

2.3.3 Kriptografija na osnovi digitalnih potrdil

V kriptografiji javnih ključev se izmenjava overjenih ključev najpogosteje opravi z uporabo *verodostojne agencije* (angl. trusted authority), pri kateri uporabniki registrirajo svoje javne ključe. Verodostojna agencija preveri identiteto vsakega uporabnika in mu nato izda digitalno potrdilo, ki vsebuje njegov javni ključ, identifikacijsko informacijo in overitvene podatke (tj. običajno digitalni podpis vsebine potrdila), s katerimi zagotavlja njegovo verodostojnost ter pristnost. Digitalna potrdila lahko vsebujejo tudi dodatne podatke, kot je serijska številka, datum veljavnosti, namen uporabe (npr. šifriranje, podpisovanje, izdajanje potrdil), ime izdajatelja itd. Agencijo, ki izdaja digitalna potrdila, imenujemo *certifikatna agencija* (angl. certification authority).

2.3.4 Kriptografija na osnovi identitete

Ideja o kriptografiji javnih ključev je bila revolucionaren dosežek, vendar pa je s seboj prinesla tudi nove probleme. Težave so se začele pojavljati pri upra-

vljanju s ključi in pri izdajanju, podaljševanju ter preklicevanju digitalnih potrdil. Da bi odpravili nekatere izmed njih, je Shamir leta 1984 predlagal nov koncept kriptografije, imenovan *kriptografija na osnovi identitete* [136]. V slednji se javni ključ osebe izračuna kar iz njegove identifikacijske informacije, kot je npr. naslov e-pošte, telefonska številka, davčna številka, IP naslov itd. S tem se znebimo odvečnega dela z upravljanjem digitalnih potrdil, saj so javni ključi implicitno overjeni. Zato lahko uporabniki šifrirajo podatke in preverjajo digitalne podpise brez predhodnega razdeljevanja ključev ali digitalnih potrdil. Poleg tega lahko samo uporabniki, ki imajo ustrezno identiteto, odšifrirajo tajnopise in v njenem imenu podpisujejo sporočila. To storijo z zasebnim ključem, ki jim ga po preverjanju identitete preko varnega kanala izda zaupanja vredna tretja oseba. Slednjo običajno imenujemo *generator zasebnih ključev*.

Shamir je v svojem članku [136] predstavil tudi prvo shemo za digitalni podpis na osnovi identitete. Kmalu po njegovi objavi pa so se pojavili tudi prvi protokoli za dogovor o ključu [115, 57, 145], katerih varnost je temeljila na problemu RSA, Diffie-Hellmanovem problemu in problemu diskretnega logaritma. Z odkritjem učinkovitih bilinearnih parjenj je to področje še bolj zacvetelo in posledično se je v literaturi pojavilo veliko novih protokolov [76, 141, 131, 37, 36]. Leta 2001 sta se pojavili tudi prvi shemi za šifriranje na osnovi identitete, ki so ju predstavili Boneh in Franklin [24], ter Cocks [43]

Kriptografija na osnovi identitete je dobra rešitev za odpravo digitalnih potrdil, vendar pa ima tudi svoje pomanjkljivosti. Tako lahko generator zasebnih ključev odšifrira in digitalno podpisuje vsa sporočila, saj ima v posesti glavni tajni ključ, s katerim lahko izračuna zasebni ključ vsakega uporabnika. Zaradi tega dejstva mu morajo vsi uporabniki brezpogojno zaupati, sistemi na osnovi identitete pa se ne morejo uporabljati za preprečevanje zanikanja. Poleg tega je generator zasebnih ključev privlačna tarča napadalcev, saj edini pozna glavni tajni ključ. Če upoštevamo še, da so kriptografske rešitve na osnovi identitete običajno računsko zahtevnejše od ostalih, ni presenetljivo, da se ta pristop do danes še ni dobro uveljavil.

2.4 Dokazovanje varnosti

V razvoju kriptografije je bilo predlaganih in razbitih že veliko predlogov kriptosistemov, protokolov ter shem. Razloge za to lahko najdemo v dejstvu, da je bilo nekoč sestavljanje novih predlogov osredotočeno na preprečevanje znanih napadov. Pri tem niso bili upoštevani morebitni novi, še nepoznani napadi oz. možne izboljšave obstoječih. Zato so z leti nove varnostne pomanjkljivosti predlogov slej ko prej prišle na dan. Da se to v prihodnje ne bi več dogajalo, se v moderni kriptografiji poslužujemo drugačnega pristopa.

Reševanje novega kriptografskega problema pričnemo z njegovo natančno definicijo. Nato določimo varnostni model, v katerem jasno opredelimo zmogljivosti napadalca, tj. kakšno računsko moč ima, kakšna sredstva ima na voljo, kako sodeluje s poštenimi udeleženci itd. Sledi definicija varnosti, s katero točno določimo, kdaj je nek predlog rešitve varen in kakšne cilje ima napadalec pri izvedbi napada. Šele nato iščemo rešitev problema in njeno varnost utemeljimo z dokazom. S slednjim običajno želimo dokazati, da če lahko napadalec v varnostnem modelu doseže svoj cilj, potem zna rešiti tudi težko rešljiv problem (glej §2.2).

V nadaljevanju bomo predstavili tri osnovne gradnike za sestavo varnih kriptografskih orodij in dva varnostna modela za dokazovanje varnosti le-teh.

2.4.1 Psevdonaključne funkcije

V tem podrazdelku bomo predstavili psevdonaključne funkcije, ki so jih v kriptografijo leta 1985 uvedli Goldreich, Goldwasser in Micali [59]. Na področju simetrične kriptografije, kjer si vsak par oseb deli skupni tajni ključ, se uporabljajo predvsem za varno izmenjavo podatkov, za overjanje sporočil in za medsebojno identifikacijo. V moderni kriptografiji z njimi modeliramo bločne šifre, dokazujemo varnost protokolov, ki takšne šifre uporabljajo, itd.

Naj bo D definicijsko območje oz. domena in Z končna zaloga vrednosti. Pri tem privzamemo, da sta množici D in Z vedno neprazni. Domena D je lahko neskončna, saj bomo nad tema množicama kasneje definirali tudi zgoščevalne funkcije, ki lahko kot vhod sprejmejo poljubno dolga sporočila. Množico vseh preslikav, ki slikajo iz domene D v zalogo vrednosti Z, bomo označili z $\mathcal{R}$. Moč slednje je enaka $|Z|^{|D|}$, saj lahko vsakemu izmed $|D|$ ele-

mentov domene priredimo poljuben element iz zaloge vrednosti Z, neodvisno od ostalih priredb.

Definicija 2.51. *Naključna funkcija* je preslikava $r : D \to Z$, ki je izbrana slučajno z enakomerno porazdelitvijo iz množice $\mathcal{R}$.

Naključne funkcije ne moremo implementirati z nobeno konkretno funkcijo, saj so izhodi le-teh deterministični. Zato bi bilo potrebno v definicijo uvesti dodaten parameter, s katerim bi lahko slučajno izbrali eno izmed možnih preslikav iz množice $\mathcal{R}$. Vendar, ker je pri smiselni izbiri domene in zaloge vrednosti teh preslikav ogromno, takšna implementacija ne bo nikoli učinkovita. S praktičnega vidika bi bilo zato veliko bolje, če bi lahko preslikavo izbrali iz neke manjše podmnožice $\mathcal{F} \subseteq \mathcal{R}$ tako, da bi njeno "obnašanje" navzven še vedno izgledalo naključno. To lahko storimo tako, da iz množice $\mathcal{R}$ najprej izberemo le nekaj preslikav, jih indeksiramo s ključem in nato s slučajno izbiro ključa določimo eno izmed njih. Množico teh preslikav lahko formalno zapišemo z *družino funkcij*

$$\mathcal{F} = \{F_k : D \to Z \mid k \in K\},$$

kjer smo s K označili končno neprazno množico ključev. V računalništvu lahko družino funkcij $\mathcal{F}$ predstavimo preprosto kar s tabelo, v kateri nam vsaka vrstica določa en *primerek* F_k iz te družine, matematično pa jo lahko zapišemo tudi kot preslikavo $\mathcal{F} : K \times D \to Z$.

V kriptografiji množico ključev K, domeno D in zalogo vrednosti Z določimo glede na nek varnosti parameter $\kappa \in \mathbb{N}$. Pogosto izberemo množice $K = \{0,1\}^\kappa$, $D = \{0,1\}^{\ell_1(\kappa)}$ in $Z = \{0,1\}^{\ell_2(\kappa)}$, kjer sta ℓ_1 in ℓ_2 polinoma z realnimi koeficienti. Število κ imenujemo *velikost ključa*, $\ell_1(\kappa)$ *velikost vhoda* in $\ell_2(\kappa)$ *velikost izhoda*. Preprost izračun nam v tem primeru razkrije, da množica $\mathcal{R}$ vsebuje natanko $2^{\ell_2(\kappa)2^{\ell_1(\kappa)}}$ različnih funkcij. To pa ni edina uporabna izbira množic K, D in Z. V nekaterih kriptosistemih lahko definicijsko območje oz. zalogo vrednosti sestavljajo elementi grupe, nizi različnih dolžin itd.

Neformalno je psevdonaključna funkcija učinkovito izračunljiva družina funkcij, za katero ne obstaja polinomski algoritem, ki bi lahko naključno izbrani primerek iz te družine ločil od dejanske naključne funkcije. Njena formalna definicija pa je sledeča:

Definicija 2.52. Družina funkcij $\mathcal{F} = \{F_k : D \to Z \mid k \in K\}$ je *psevdonaključna*, če

- obstaja deterministično polinomski algoritem, ki za vsak ključ $k \in K$ in vhod $x \in D$ učinkovito izračuna vrednost funkcije $F_k(x)$,
- je za vse polinomsko časovno omejene napadalce $\mathcal{O}_f$ z dostopom do preroka za neko funkcijo f, ki kot izhod vrnejo zgolj `true` ali `false`, vrednost razlike pogojnih verjetnosti

$$\left| P[\mathcal{O}_{F_k}(D, Z) = \texttt{true} \mid k \xleftarrow{\$} K] - P[\mathcal{O}_r(D, Z) = \texttt{true} \mid r \xleftarrow{\$} \mathcal{R}] \right|$$

  zanemarljiva v parametru κ, glede na vse izbire ključa k in funkcije r.

Pri tem je potrebno poudariti, da napadalcu $\mathcal{O}_f$ nikoli ni razkrit izbran ključ k in le-ta ne ve, do preroka katere funkcije f ima dostop. Prerok je za njega zgolj črna škatla, katere notranjega delovanja ne pozna.

Brezpogojen obstoj psevdonaključne funkcije bi pomenil P $\neq$ NP, zato njihovo varnost običajno dokažemo s privzetjem različnih predpostavk. Najšibkejša izmed njih je obstoj enosmerne funkcije, tj. funkcije, ki je lahko izračunljiva in težko obrnljiva. Znano je namreč, da je iz enosmernih funkcij možno sestaviti psevdonaključne generatorje [66], iz njih pa nato še psevdonaključne funkcije [59].

Leta 1997 sta Naor and Reingold [110] predstavila preprosto psevdonaključno družino funkcij in njeno varnost dokazala s prevedbo na računski Diffie-Hellmanov problem. Za njeno konstrukcijo si je potrebno izbrati praštevili p in q, tako da velja $p \mid q - 1$, ter element $g \in \mathbb{Z}_q^*$ reda p. Za poljubno število $n \in \mathbb{N}$ lahko nato določimo množico ključev $K = (\mathbb{Z}_p)^{n+1}$, domeno $D = \{0,1\}^n$ in zalogo vrednosti $Z = \mathbb{Z}_q^*$. Psevdonaključno družino funkcij $\mathcal{F}$ v tem primeru sestavljajo preslikave $F_k : D \to Z$, ki so za vsak ključ $k = (k_0, k_1, \ldots, k_n) \in K$ in binarni niz $x = (x_1, \ldots, x_n) \in D$ definirane kot

$$F_k(x) = g^{k_0 \prod_{i=1}^{n} k_i^{x_i}} \bmod q.$$

Pri tem je potrebno poudariti, da je izračun te psevdonaključne funkcije zelo preprost. Najprej je potrebno izračunati produkt $e = k_0 \prod_{i=1}^{n} k_i^{x_i} \bmod p$ in nato še potenco $g^e \bmod q$. Za izračun je zato potrebnih le n množenj po modulu p in eno potenciranje po modulu q.

2.4.2 Kriptografske zgoščevalne funkcije

Kriptografska zgoščevalna funkcija je preslikava, ki preslika podatke iz večje domene v manjšo. Za razliko od navadnih zgoščevalnih funkcij morajo kriptografske imeti še dodatne varnostne lastnosti, ki jih bomo podrobno spoznali v nadaljevanju. V računalništvu se uporabljajo predvsem v podatkovnih strukturah, kot so npr. razpršene tabele, za vstavljanje in iskanje elementov v konstantnem času, za odkrivanje podvojenih podatkov in datotek, ter za njihovo enolično identifikacijo. Na področju računalniške varnosti z njimi shranjujemo uporabniška gesla za varno prijavo v sistem in preverjamo pristnost ter verodostojnost podatkov. Njihova uporaba v kriptografiji je zelo široka, najbolj pogosto pa se uporabljajo v protokolih za izmenjavo ključa, v shemah za digitalni podpis in v shemah za overjanje podatkov.

Neformalno lahko zgoščevalno H definiramo kot enostavno izračunljivo preslikavo, ki sprejme vhodne podatke x poljubne dolžine in jih stisne v običajno dosti krajši niz $H(x)$ fiksne dolžine. Tako pripravi "prstni odtis", s katerim lahko morebitne spremembe podatkov zaznamo. Vrednost $H(x)$ imenujemo *povzetek* ali *zgostitev* (angl. message digest, hash) in je običajno kratek binaren niz. Njegova dolžina v današnjih aplikacijah znaša vsaj 160 bitov, kar zaradi rojstnodnevnega napada nudi zadovoljivo 80 bitno varnost.

Paru različnih elementov x in x', za katera velja $H(x) = H(x')$, pravimo *trk*. Vsaka zgoščevalna funkcija vsebuje mnogo trkov, saj vedno slika iz večje domene v manjšo in zato po Dirichletovem principu vedno obstajajo elementi z isto zgostitvijo. Vendar kljub temu od "varne" kriptografske zgoščevalne funkcije zahtevamo, da je trke težko učinkovito najti. Zato pravimo, da je zgoščevalna funkcija *odporna na trke*, če noben polinomski algoritem v doglednem času ne more najti trka. Na žalost pa takšna definicija ni ustrezna, saj za fiksno zgoščevalno funkcijo vedno obstajajo zelo kratki in preprosti algoritmi, ki najdejo trke. Takšni algoritmi imajo v svoji kodi shranjen en primer trka, ki ga vrnejo kot izhod. Edina težava teh algoritmov je, da jih ne poznamo oz. bi potrebovali eksponentno mnogo časa, da bi jih poiskali.

Da lahko matematično formalno definiramo odpornosti na trke, je potrebno zgoščevalno funkcijo definirati kot množico indeksiranih funkcij. V nadaljevanju bomo podali definicijo zgolj za zgoščevalne funkcije brez ključa,

saj slednje zadoščajo za razumevanje opisov protokolov in shem, predstavljenih v tem delu.

Definicija 2.53. Družina *zgoščevalnih funkcij* je množica

$$\mathcal{H} = \{H_s : D \to Z \mid s \in K\},$$

kjer je domena D po moči večja od zaloge vrednosti Z, tj. $|D| > |Z|$.

Pri definiciji je možno opaziti, da smo ključ v *primerku* zgoščevalne funkcije $H_s(x)$ zapisali z oznako s in ne s k, kot bilo to v navadi pri psevdonaključnih funkcijah. Razlog za to se skriva v dejstvu, da $s \in K$ ni običajni tajni ključ, temveč je javen. Takšen ključ v kriptografiji imenujemo *sol*. Za razliko od pravega tajnega ključa, ta služi zgolj za označitev enega primerka zgoščevalne funkcije. Zato ni potrebno, da je tajen, niti da je izbran naključno z enakomerno porazdelitvijo iz množice K. Posledično tudi zgoščevalna funkcija $H_s(x)$ ne rabi biti definirana za vse možne vrednosti iz množice K.

Na prvi pogled zgleda, da zgoščevalne funkcije, uporabljene v praksi ne ustrezajo zgoraj navedeni definiciji. V resnici to ni čisto res, saj nekatere zgoščevalne funkcije uporabljajo vnaprej določeno začetno verižno vrednost ali začetni inicializacijski vektor. Med slednje sodi tudi SHA1, zato jo lahko obravnavamo kot en primerek iz družine funkcij. V podano definicijo sodijo tudi vse zgoščevalne funkcije, ki uporabljajo novejšo verzijo Merkle–Damgårdove konstrucije [46, 107], imenovano HAIFA [21]. Ta v kompresijski funkciji, poleg števca trenutno zgoščenih bitov, dodatno uporablja še sol. Z uporabo soli lahko tako sestavimo celotno družino zgoščevalnih funkcij, oz. če želimo imeti samo en njen primerek, sol vedno nastavimo na neko privzeto vrednost. Omenjeno konstrukcijo uporablja zgoščevalna funkcija BLAKE[1], ki se je na SHA3 natečaju, na katerem je zmagala zgoščevalna funkcija Keccak[2], uvrstila med štiri finaliste. Njeno varnost smo skupaj s sodelavcema J. Vidalijem in E. Pašalićem preučili tudi mi in zaključili, da obe njeni oslabljeni različici, ki so ju avtorji zastavili kot izziv, nista varni [151].

V literaturi obstaja veliko različnih formalnih definicij varnosti zgoščevalnih funkcij. Kratek pregled slednjih sta pripravila Rogaway in Shrimpton [130], ki sta hkrati preučila tudi njihove medsebojne relacije in razlike

[1] `http://131002.net/blake/`

[2] `http://keccak.noekeon.org/`

med njimi. Za potrebe te disertacije, bomo definirali naslednje varnostne zahteve.

Definicija 2.54. Družina zgoščevalnih funkcij $\mathcal{H} = \{H_s \mid s \in K\}$ je *odporna na praslike* (angl. preimage resistant), če je za vse verjetnostne polinomske napadalce $\mathcal{O}$, verjetnost

$$P\big[H_s(x') = y \mid s \xleftarrow{\$} K, x \xleftarrow{\$} D, y = H_s(x), x' = \mathcal{O}(1^\kappa, s, y)\big]$$

zanemarljiva v parametru κ, glede na vse izbire soli s in vhodne podatke x.

Definicija 2.55. Družina zgoščevalnih funkcij $\mathcal{H} = \{H_s \mid s \in K\}$ je *odporna na druge praslike* (angl. second-preimage resistant), če je za vse verjetnostne polinomske napadalce $\mathcal{O}$, verjetnost

$$P\big[x \neq x', H_s(x) = H_s(x') \mid s \xleftarrow{\$} K, x \xleftarrow{\$} D, x' = \mathcal{O}(1^\kappa, s, x)\big]$$

zanemarljiva v parametru κ, glede na vse izbire soli s in vhodne podatke x.

Zgoščevalne funkcije z odpornostjo praslik in drugih praslik imenujemo *enosmerne zgoščevalne funkcije.*

Definicija 2.56. Družina zgoščevalnih funkcij $\mathcal{H} = \{H_s \mid s \in K\}$ je *odporna na trke* (angl. collision resistant), če je za vse verjetnostne polinomske napadalce $\mathcal{O}$, verjetnost

$$P\big[x \neq x', H_s(x) = H_s(x') \mid s \xleftarrow{\$} K, (x, x') = \mathcal{O}(1^\kappa, s)\big]$$

zanemarljiva v parametru κ, glede na vse izbire soli s.

Zgoščevalne funkcije, ki so odporne na trke, so odporne tudi na druge praslike. Ni pa nujno, da so odporne na praslike, čeprav v praksi od njih pričakujemo da so.

V članku [111], sta Naor in Yung ugotovila, da v mnogih kriptografskih aplikacijah zadostuje preprostejša lastnost kot odpornost na trke. To lastnost sta poimenovala *odpornost na ciljne trke*, zgoščevalno funkcijo s to lastnostjo pa *univerzalna enosmerna zgoščevalna funkcija.*

Definicija 2.57. Družina zgoščevalnih funkcij $\mathcal{H} = \{H_s \mid s \in K\}$ je *odporna na ciljne trke* (angl. target collision resistant), če je za vse verjetnostne polinomske napadalce $\mathcal{O}$, vrednost

$$\max_{x \in D} \left\{ P\big[x \neq x', H_s(x) = H_s(x') \mid s \xleftarrow{\$} K, x' = \mathcal{O}(1^\kappa, s)\big] \right\}$$

zanemarljiva v parametru κ, glede na vse izbire soli s.

Kriptografske zgoščevalne funkcije običajno slikajo iz množice $D = \{0,1\}^*$ v množico $Z = \{0,1\}^n$, kjer je n fiksno naravno število. Včasih pa od njih zahtevamo, da je njihovo definicijsko območje ali zaloga vrednosti kakšna bolj kompleksna struktura, kot je npr. končna grupa. V tem primeru je lahko zgoščevalna funkcija zapletena, saj je najprej potrebno elemente preslikati v neko vmesno množico in jih šele nato z ustrezno transformacijo preslikati v grupo. Na primer, primerek zgoščevalne funkcije $H : \{0,1\}^* \to \mathbb{G}$, kjer je $\mathbb{G} = \langle P \rangle$ grupa točk eliptične krivulje reda p, bi lahko sestavili z uporabo zgoščevalne funkcije SHA1. Z njo bi najprej vhod $x \in \{0,1\}^*$ preslikali v bitni niz $(b_0, b_1, \ldots, b_k) = \mathrm{SHA1}(x)$, ki bi nam predstavljal celo število $m = b_0 + 2b_1 + \cdots + 2^k b_k$. To število bi nato reducirali po modulu p in dobili skalar $n = m \bmod p$, s katerim bi izračunali zgostitev $H(x) = nP$.

V nadaljevanju teze bomo, če ni drugače rečeno, z besedo zgoščevalna funkcija označevali izbran primerek iz družine kriptografske zgoščevalne funkcije, ki je odporna na trke.

2.4.3 Funkcije za izpeljavo ključa

Funkcije za izpeljavo ključa (angl. key derivation functions) so sestavni del mnogih kriptografskih sistemov. Njihov cilj je iz vhodnih podatkov, ki vsebujejo dovolj veliko stopnjo naključnosti, izračunati enega ali več tajnih ključev. Izpeljani ključi morajo biti kriptografsko varni, kar pomeni, da jih računsko omejeni napadalec ne more ločiti od naključno izbranih ključev enake dolžine. Od tod sledi, da če napadalec pozna del izpeljanega ključa, potem mu ti podatki ne razkrijejo nič informacij o preostalem delu. To velja tudi v primeru, če vhodni podatki niso porazdeljeni z enakomerno porazdelitvijo, ali če jih napadalec deloma pozna. Funkcije za izpeljavo ključa običajno kot vhod sprejmejo osebna gesla, tajne ključe, skupne skrivnosti izračunane v

protokolih za dogovor o ključu, izhodna zaporedja nepopolnih generatorjev naključnih bitov itd.

Najbolj pogost pristop za sestavo funkcij za izpeljavo ključa se imenuje *izvleci-in-razširi*. Ta upošteva dejstvo, da kadar so vhodni podatki enakomerno porazdeljeni ali morda celo psevdonaključni, takrat jih lahko uporabimo za seme generatorja psevdonaključnih števil ali kot ključ psevdonaključne funkcije, s katero nato izračunamo nove ključe. Če pa vhodni podatki ne zadoščajo omenjenim pogojem, potem je najprej potrebno iz njih izvleči psevdonaključen ključ in šele nato iz njega izpeljati nove tajne ključe. Takšen pristop uporablja tudi varna in učinkovita funkcija HKDF, ki jo je leta 2010 na osnovi HMAC kode za overjanje podatkov [9] sestavil Krawczyk [89, 90].

2.4.4 Model naključnega preroka

Zgoščevalne funkcije igrajo v kriptografiji zelo pomembno vlogo, saj lahko zaradi njihove preprostosti sestavimo hitrejše in učinkovitejše protokole ter sheme. Vendar takšne konstrukcije s seboj prinesejo tudi težave, saj je njihovo varnost v standardnem modelu, v katerem je napadalec omejen le s časom in računsko močjo, ki jo ima na voljo, zelo težko dokazati. Zato se je v kriptografiji uveljavil nov pristop za dokazovanje varnosti, pri katerem določena kriptografska orodja nadomestimo z njihovimi idealiziranimi različicami.

Najbolj popularen primer takšnega pristopa je *model naključnega preroka* (angl. random oracle model), ki sta ga leta 1993 predstavila Bellare in Rogaway [13]. V slednjem zgoščevalne funkcije nadomestimo z naključnimi preroki. Te si predstavljamo kot črne škatle, ki na vsako novo poizvedbo odgovorijo z naključno vrednostjo, izbrano enakomerno iz zaloge vseh možnih vrednosti, medtem ko na že opravljene poizvedbe vedno odgovorijo z istim odgovorom. Naključni preroki so pravzaprav naključne funkcije (glej def. 2.51), ki vsako vhodno vrednost preslikajo v neko naključno izbrano vrednost.

Dokazi varnosti v modelu naključnega preroka so običajno lažji kot v standardnem modelu. Posledično so zato tudi protokoli in sheme, katerih varnost je bila dokazana v slednjem, manj učinkoviti od tistih, ki uporabljajo zgoščevalne funkcije in so dokazano varni v modelu naključnega preroka. To

pa je tudi eden izmed razlogov, zakaj je ta model postal tako zelo popularen za dokazovanje varnosti različnih protokolov in shem.

Kadar dokazujemo varnost v modelu naključnega preroka, moramo upoštevati tudi nekatere njegove pomanjkljivosti in omejitve. Zavedati se moramo, da naključnega preroka ne moremo implementirati z nobeno konkretno funkcijo, in da obstajajo določene "umetno" sestavljene sheme za šifriranje in digitalno podpisovanje, ki so varne v tem modelu, a jih lahko preprosto razbijemo, če preroka zamenjamo z neko konkretno funkcijo [29]. Kljub omenjenim pomanjkljivostim je model naključnega preroka dobro uveljavljen in splošno sprejet model, saj mora napadalec za razbitje protokola oz. sheme najti varnostno pomanjkljivost v uporabljeni zgoščevalni funkciji. V tem primeru lahko slednjo zamenjamo z novo in s tem zopet zagotovimo varnost kriptografske rešitve.

V literaturi lahko najdemo veliko protokolov in shem, katerih varnost je bila dokazana v modelu naključnega preroka [13, 17, 51, 117, 124]. Med njimi najbolj izstopa shema OAEP [15], ki se pogosto skupaj s šifro RSA uporablja za šifriranje sporočil.

2.4.5 Varnosti model eCK

Varnostni model eCK (extended Canetti-Krawczyk) je trenutno eden izmed najmočnejših modelov za dokazovanje varnosti dvostranskih protokolov za overjen dogovor o ključu. Leta 2007 so ga predstavili LaMacchia, Lauter in Mityagin [91] kot nadgradnjo osnovnega Canetti-Krawczyk modela [30]. Z njim naj bi zajeli vse možne napade, ki jih lahko napadalec izvede zaradi razkritja začasnih ali trajnih zasebnih ključev. Med zajete napade sodijo tudi napad z znanim ključem, napad lažnega predstavljanja z razkritim ključem in napad deljenja ključa z neznano osebo. Avtorji modela tudi poudarjajo, da je v njihovem modelu možno dokazati prihodnjo in delno prihodnjo varnost protokolov (glej §3.2.2). Zato ima napadalec v njem občutno večjo moč, kot jo ima v starejših varnostnih modeli [14, 16, 22, 10].

V dvostranskem protokolu za overjen dogovor o ključu se skupna skrivnost, iz katere se izpelje sejni ključ, običajno izračuna iz štirih podatkov. Če v protokolu nastopata Anita in Bojan, potem se skrivnost izračuna iz njunih

trajnih zasebnih ključev in iz dveh začasnih zasebnih ključev, ki si ju Anita in Bojan izbereta med izvajanjem protokola. V modelu eCK ima napadalec izjemno moč, saj lahko razkrije katerokoli podmnožico teh štirih podatkov, s katero ni možno trivialno razbiti protokola. Napadalec lahko tako razkrije trajni ključ Anite in začasni ključ Bojana, in še vedno ne bo mogel izračunati skupne skrivnosti. Ni pa mu dovoljeno razkriti trajnega in začasnega ključa Anite, saj mu s tema podatkoma ni težko izračunati skrivnosti in iz nje izpeljati sejni ključ. Dodatno lahko v modelu eCK napadalec zamenjuje javne ključe uporabnikov, razkriva stare sejne ključe itd.

Model je definiran kot eksperiment, v katerem poleg napadalca nastopajo tudi poštene osebe. Vsa komunikacija poteka preko javnega kanala in je pod nadzorom napadalca. Ta sam določi, kateri pari oseb naj izvedejo dogovor o ključu in v kakšnem vrstnem redu naj dogovori potekajo. Zato v neki seji udeleženec nikoli ne more biti čisto prepričan, s kom v resnici komunicira. Napadalec ima v eksperimentu možnost opraviti različne poizvedbe, s katerimi lahko spreobrne določene poštene osebe in prevzame njihov nadzor ter razkrije sejne ali začasne ključe poljubnih sej. Na koncu eksperimenta si izbere poljubno sejo in kot odgovor na testno poizvedbo prejme njen sejni ključ ali naključno izbran niz. Cilj napadalca je nato ugotoviti, kaj je prejel v odgovoru. Pri tem mora za testno sejo veljati, da se je ta v eksperimentu uspešno zaključila in da je nedotaknjena, tj. da napadalec preko poizvedb ni razkril sejnega ključa ali vseh njenih začasnih ključev, ter da nima pod nadzorom nobenega njenega udeleženca.

Podroben opis varnostnega modela eCK, ki ga še danes štejemo med najmočnejše varnostne modele, presega obseg te disertacije in ni potreben za razumevanje varnostne analize protokolov za dogovor o ključu, predstavljene v petem poglavju. Zainteresirani bralci lahko več podrobnosti preberejo v izvirnem članku [91].

2.4.6 Dokazi s prevedbami

V sodobni kriptografiji varnostne lastnosti novih kriptografskih orodij, protokolov in shem najpogosteje dokazujemo s prevedbami. Z njimi želimo pokazati, da če lahko napadalec v varnostnem modelu doseže svoj cilj, potem

ga lahko uporabimo za reševanje težkega problema. Če pri tem predpostavimo, da izbran problem ni rešljiv v doglednem času, potem nam dokaz s prevedbo zagotavlja računsko varnost. To pomeni, da noben napadalec z omejeno računsko močjo ne more z nezanemarljivo verjetnostjo ogroziti varnosti kriptografske rešitve. Ne zagotavlja pa nam popolne varnosti, saj bi za dokaz slednje bilo potrebno narediti časovno analizo vseh možnih napadov in najti spodnjo mejo za čas njihovega izvajanja. Pravzaprav bi bilo potrebno dokazati, da noben polinomski algoritem ne more izvesti uspešnega napada. To pa je zelo malo verjetno, saj bi pri dokazovanju varnosti nekaterih protokolov in shem to pomenilo, da bi morali rešiti težak problem iz teorije izračunljivosti, kot je na primer vprašanje, ali je P = NP. Dokazi s prevedbami so se v kriptografiji prvič pojavili leta 1982, ko sta Shafi Goldwasser in Silvio Micali[3] objavila shemo za šifriranje z javnimi ključi ter dokazala njeno varnost s prevedbo na problem kvadratnih ostankov v standardnem varnostnem modelu [61].

Pri dokazih s prevedbami predpostavimo, da so določeni problemi težki, čeprav za to zaenkrat nimamo še nobenega zagotovila. Njihova točna časovna zahtevnost nam običajno ni znana, trenutno poznamo le zahtevnost najboljših algoritmov za njihovo reševanje. Ker pa bi kljub temu radi imeli zagotovilo o njihovi težavnosti, pogosto spodnjo mejo za čas reševanja dokažemo v idealnih modelih. Na primer, za problem diskretnega logaritma je bilo dokazano, da ga je v modelu splošne grupe najhitreje možno rešiti v času $O(\sqrt{n})$, kjer je n velikost grupe [140]. Nekaj najbolj pogostih težkih problemov smo že predstavili v razdelku 2.2, obstaja pa še mnogo drugih[4].

Pri dokazovanju varnosti določene kriptografske rešitve želimo poiskati polinomsko prevedbo, s katero lahko napadalca pretvorimo v algoritem za reševanje nekega težkega problema. Takšen dokaz se običajno prične s predpostavko, da izbran računski problem ni rešljiv v polinomskem času z nezanemarljivo verjetnostjo. Nato sledijo trije koraki.

1. Najprej določimo učinkovitega napadalca, tj. verjetnostni polinomski algoritem, ki zna razbiti varnost kriptografske rešitve z verjetnostjo ε.

[3] Shafi Goldwasser in Silvio Micali, raziskovalca na inštitutu za tehnologijo MIT, sta za leto 2012 prejela Turingovo nagrado za napredek v kriptografiji.

[4] `http://www.ecrypt.eu.org/wiki/index.php`

2. Nato sestavimo nov algoritem, ki poskuša rešiti težak problem tako, da uporabi napadalca kot podprogram. Pri tem je pomembno, da algoritem ne pozna notranjega delovanja napadalca. Ve le to, da napadalec želi napasti izbrano kriptografsko rešitev. Ko algoritem kot vhod prejme primerek problema, simulira rešitev napadalcu tako, da se ta ne zaveda, da je uporabljen kot podprogram. Če napadalec uspešno razbije varnost kriptografske rešitve, potem algoritem to izkoristi in z nezanemarljivo verjetnostjo $1/p$ reši problem.

3. Iz prejšnjih dveh korakov sledi, da če verjetnost ε ni zanemarljiva, potem algoritem reši problem v polinomskem času z nezanemarljivo verjetnostjo ε/p. Od tod lahko zaključimo, da če je izbran računski problem težak, potem noben napadalec ne more razbiti kriptografske rešitve z nezanemarljivo verjetnostjo in rešitev je računsko varna.

Pravimo, da je prevedba *tesna*, če je verjetnost uspeha napadalca ε, ki teče v nekem času t, približno enaka verjetnosti ε/p, s katero algoritem v približno istem času reši izbran problem. Tesne prevedbe so največ, kar lahko dosežemo, saj intuitivno shemo, ki temelji na enem samem primerku težkega problema, ne more biti težje razbiti kot pa rešiti problem sam.

Dokazi s prevedbami imajo tudi svoje slabosti, saj običajno varnostni modeli ne zajemajo vseh možnih napadov, ki se lahko zgodijo pri implementaciji kriptografske rešitve v praksi. Med takšne napade sodijo napadi s stranskim kanalom, saj je slednje težko oz. skorajda nemogoče formalno opisati v varnostnem modelu.

Poglavje 3

Protokoli za overjen dogovor o ključu

V tem poglavju bomo sistematično razdelali protokole za dogovor o ključu, ki sodijo med najpomembnejša kriptografska orodja. Definirali bomo varnostne cilje, ki naj bi jih ti protokoli dosegali, in varnostne lastnosti, ki naj bi jih imeli. Protokole za dogovor o ključu lahko razdelimo v različne skupine, zato bomo nekatere izmed skupin tudi na kratko opisali. Predstavili bomo dvostranske, tristranske in večstranske protokole za dogovor o ključu, protokole na osnovi identitete in gesel ter podali nekatere njihove glavne predstavnike.

Gradivo tega poglavja je povzeto po dveh splošnih kriptografskih knjigah avtorjev Menezes, Vanstone in Oorschot [105] ter Stinson [143], po pregledni knjigi o protokolih za overjanje in vzpostavitev ključa avtorjev Boyd in Mathuria [27] ter po člankih na temo overjenega dogovora o ključu avtorjev Blake-Wilson, Johnson in Menezes [22] ter Diffie, Oorschot in Wiener [49].

3.1 O protokolih in ključih

Protokolu, v katerem želita vsaj dve osebi vzpostaviti skupno skrivnost, namenjeno za nadaljnjo kriptografsko uporabo, pravimo *protokol za vzpostavitev ključa*. Udeleženci protokola iz skupne skrivnosti zelo pogosto izračunajo *sejni ključ* in ga uporabijo za šifriranje ter odšifriranje podatkov, ki si jih želijo izmenjati med seboj. Ker se sejni ključ uporablja zgolj za krajša

časovna obdobja oz. seje, mu včasih pravimo tudi *začasna skrivnost*.

Od protokolov za vzpostavitev ključa običajno zahtevamo, da se skupna skrivnost razlikuje ob vsaki novi izvedbi. Hkrati pa od udeležencev pričakujemo, da se ob poteku seje znebijo sejnega ključa in ga varno izbrišejo. Ker je priprava in vzpostavitev ključa možna na različne načine, lahko protokole v grobem razdelimo v dve skupini: protokole za prenos ključa in protokole za dogovor o ključu.

Definicija 3.1. *Protokol za prenos ključa* je tehnika za vzpostavitev ključa, pri kateri en udeleženec izbere ali na nek način pridobi skupno skrivnost in jo varno posreduje ostalim udeležencem.

To lahko stori z uporabo simetrične kriptografije tako, da skrivnost zašifrira s tajnim ključem, ki si ga predhodno deli z vsakim udeležencem protokola. Če je uporabnikov v sistemu veliko, potem takšen pristop ni najbolj učinkovit, saj mora vsaka oseba pri sebi hraniti veliko simetričnih ključev. V tem primeru je varen prenos bolje opraviti z uporabo kriptografije javnih ključev. Tu oseba, ki izbere skrivnost, pridobi javne ključe vseh udeležencev, s katerimi želi komunicirati, jih preveri in z njimi zašifrira skupno skrivnost.

Očitna slabost protokolov za prenos ključa je, da lahko na skupno skrivnost in posledično na sejni ključ vpliva le oseba, ki je skrivnost izbrala. Protokoli za dogovor o ključu te slabosti nimajo, saj noben udeleženec ne more vnaprej predvideti skupne skrivnosti ali se do nje dokopati pred začetkom protokola.

Definicija 3.2. *Protokol za dogovor o ključu* je tehnika za vzpostavitev ključa, pri kateri udeleženci izračunajo skupno skrivnost iz podatkov, ki jih sami prispevajo ali so na nek način z njimi povezani.

V tem delu se bomo ukvarjali zgolj z varnostno analizo protokolov za dogovor o ključu.

3.2 Napadi in varnostne zahteve

Preden bolj podrobno razdelamo protokole za dogovor o ključu, pojasnimo katere varnostne cilje naj bi varni protokoli dosegali in kakšne lastnosti naj

bi imeli. Hkrati predstavimo tudi nekatere osnovne napade, ki bodo služili za razumevanje varnostne analize, opisane v petem poglavju tega dela.

3.2.1 Varnostni cilji

Začnemo z definicijo osnovnih varnostnih ciljev protokolov za dogovor o ključu.

Definicija 3.3. *Overitev ključa* (angl. key authentication) je lastnost, ki udeležencu protokola zagotavlja, da lahko do sejnega ključa pridejo zgolj natančno opredeljene osebe (sem štejemo tudi določene zaupanja vredne osebe).

V protokolu za dogovor o ključu, v katerem nastopata Anita in Bojan, overitev ključa Aniti zagotavlja, da lahko sejni ključ izračuna le Bojan in nobena druga oseba. Pri tem je potrebno opozoriti, da ta lastnost Aniti ne zagotavlja, da ima Bojan sejni ključ v svoji posesti, niti da je bil na kakršenkoli način vključen v protokol. Zato tej lastnosti običajno pravimo tudi *implicitna* overitev ključa.

Definicija 3.4. *Potrditev ključa* (angl. key confirmation) je lastnost, ki udeležencu protokola zagotavlja, da ima drugi (morda neznan) udeleženec v posesti sejni ključ.

Pri potrditvi ključa običajno en udeleženec prejme sporočilo od drugega, ki vsebuje nekakšen dokaz, da ima slednji v svoji posesti sejni ključ. V praksi lahko to storimo na različne načine, npr. uporabimo dokaz brez razkritja znanja, pošljemo s kriptografsko zgoščevalno funkcijo izračunano zgostitev sejnega ključa ali s sejnim ključem zašifriramo neko vnaprej znano vrednost. Pri zadnjih dveh načinih je potrebno paziti, da pri tem ne razkrijemo kakšne dodatne informacije o ključu, medtem ko pri dokazih brez razkritja znanja za to ni potrebno skrbeti.

Nekatere kriptografske aplikacije, ki uporabnikom omogočajo šifriranje izmenjanih sporočil, uporabljajo zgolj protokole brez potrditve ključa. Takšne aplikacije običajno potrditev preložijo v fazo šifriranja, v kateri najprej zašifrirajo vnaprej znane podatke. Če se dobljeni tajnopis na drugi strani odšifrira pravilno, je to impliciten dokaz, da pošiljatelj pozna sejni ključ.

Definicija 3.5. Protokol ima lastnost *eksplicitne overitve ključa*, če implicitno overi ključ in vsebuje potrditev ključa.

Glavni poudarek te disertacije je na analizi protokolov za dogovor o ključu, zato bomo pojem overjanja definirali zgolj za slednje. Sicer pa lahko naslednjo definicijo posplošimo tudi na protokole za vzpostavitev ključa.

Definicija 3.6. Protokol za *overjen dogovor o ključu* (angl. authenticated key agreement) je protokol za dogovor o ključu, ki implicitno overi ključ.

Protokol za overjen dogovor o ključu, ki je odporen na aktivne napadalce, je možno sestaviti le, če si udeleženci vnaprej delijo določene informacije. Namreč, če se dogovor začne iz "nič" in je v protokolu udeležencem dovoljena le izmenjava sporočil, potem ne obstaja noben postopek, s katerim bi lahko Anita ločila Bojana od napadalca Oskarja. Zato protokoli za overjen dogovor o ključu običajno zahtevajo pripravljalno fazo, v kateri se razdelijo oz. izmenjajo določeni verodostojni in včasih še tajni podatki. Primer slednjih so digitalna potrdila, ki vsebujejo overjene trajne javne ključe uporabnikov.

Pri protokolih za overjen dogovor o ključu je potrebno ločiti med *trajnimi ključi* in *sejnimi ključi*. Uporabniki sistema imajo lahko v posesti trajne ključe, ki so izračunani vnaprej in jih morajo varno hraniti. To najlaže storijo z uporabo varne kriptografske naprave (npr. pametne kartice), na katero shranijo takšne ključe in jih nato uporabljajo preko ustreznega vmesnika. Na takšen način trajni ključi nikoli ne zapustijo naprave in so težje dosegljivi napadalcem. V nekaterih protokolih so trajni ključi izračunani s strani zaupanja vredne tretje osebe, npr. generatorja zasebnih ključev, ki za izračun uporablja t. i. *glavni tajni ključ*. V takšnem primeru zasebni ključ uporabnika pozna tudi zaupanja vredna oseba. Tega problema se lahko znebimo, če si uporabniki sami izberejo zasebni ključ in pripadajoči javni ključ objavijo v digitalnem potrdilu. Trajni ključi se lahko uporabljajo tudi v kodah za overjanje podatkov, v shemah za digitalni podpis itd.

Z uporabo trajnih ključev se lahko udeleženci protokola za overjen dogovor o ključu dogovorijo za skupni sejni ključ, ki je namenjen za uporabo zgolj v eni seji. Ko se slednja zaključi, ga morajo vsi udeleženci zavreči in

varno izbrisati. Začasni sejni ključi se običajno uporabljajo v drugih kriptografskih orodjih, kot so npr. simetrične šifre in kode za overjanje. Razlogov za njihovo uporabo je več. Če se sejni ključ uporablja za šifriranje sporočil s simetrično šifro, potem je šifriranje hitrejše, kakor če bi šifrirali s trajnimi. Ker se sejni ključi uporabljajo zgolj v eni seji in se zato redno spreminjajo, s tem tudi zmanjšamo število tajnopisov, do katerih lahko dostopa napadalec. Še ena prednost sejnih ključev je v tem, da je v primeru njihovega razkritja ogrožena zgolj varnost ene seje. Zato jih lahko uporabljamo tudi v bolj tveganih okoljih, v katerih je verjetnost razkritja ključa večja, oz. v katerih ne zaupamo popolnoma vsem uporabljenim kriptografskim orodjem. Na koncu omenimo še, da se z uporabo sejnih ključev zmanjša tudi količina podatkov, ki jih mora posamezni uporabnik hraniti, saj ključe izračunamo šele, ko jih potrebujemo.

Definicija 3.7. Protokol za dogovor o ključu temelji *na osnovi identitete*, če se identifikacijska informacija (npr. ime, naslov, e-poštni naslov) uporabnikov uporablja kot njihov javni ključ.

Protokoli za dogovor o ključu na osnovi identitete igrajo v današnjem času zelo pomembno vlogo, saj se z njimi znebimo težav, ki nastopijo zaradi potrebe po uporabi digitalnih potrdil.

3.2.2 Varnostne lastnosti

Do danes je bilo v kriptografiji predlaganih že veliko protokolov za dogovor o ključu in za mnoge izmed njih se je izkazalo, da niso varni. Pri nekaterih so varnostne pomanjkljivosti odkrili šele mnogo let po njihovi objavi. Eden izmed razlogov za to je, da pri sestavi protokola ni popolnoma jasno, kaj vse lahko napadalci izkoristijo za izvedbo napada. Zato se je v kriptografiji uveljavilo pravilo, da je pri objavi protokola potrebno natančno opredeliti, na katere napade je protokol odporen, katere varnostne cilje dosega in kakšne varnostne lastnosti ima. Razvoj protokolov za dogovor o ključu je pripeljal do naslednjih varnostnih lastnosti, ki jih pričakujemo od takšnih protokolov [22, 105, 103].

Najprej bomo definirani želene varnostne lastnosti, ki so povezane z razkritjem sejnih in trajnih ključev.

Definicija 3.8. Protokol je varen pred *napadom z znanim ključem* (angl. known-key attack), če pasivni napadalec z razkritjem prejšnjih sejnih ključev ne more ogrožati prihodnjih sejnih ključev oz. se aktivni napadalec v prihodnosti ne more lažno predstavljati za neko osebo.

Napad z znanim ključem je primerljiv napadu z znanim čistopisom na simetrične šifre. Pri prvem napadu si lahko napadalec pomaga z razkritimi sejnimi ključi prejšnjih sej, medtem ko pri drugem pozna tajnopise nekaterih čistopisov. Realizacija napada z znanim ključem je možna, saj so sejni ključi bolj ranljivi za razkritje kot trajni. Na primer, napadalec lahko pride do sejnega ključa, če je bil uporabljen v ogroženem okolju, ali pa je bil nekoč uporabljen v varnem kriptografskem orodju, ki pa je danes zaradi napredka kriptoanalize uspešno razbit.

Definicija 3.9. Protokol ima lastnost *prihodnje varnosti* (angl. forward secrecy), če razkritje trajnih ključev enega ali več udeležencev ne ogroža predhodnih sejnih ključev.

Zgornjo definicijo lahko malce omilimo, če dovolimo, da lahko napadalec razkrije trajne ključe le nekaterih udeležencev.

Definicija 3.10. Protokol ima lastnost *delne prihodnje varnosti* (angl. partial forward secrecy), če razkritje trajnih ključev določenih udeležencev ne ogroža prejšnjih sejnih ključev.

Delno prihodnjo varnost želimo doseči v protokolih, ki uporabljajo zaupanja vredne tretje osebe, kot je npr. zaupanja vreden strežnik. V tem primeru nam ta lastnost zagotavlja, da je varnost sejnih ključev ogrožena le, če napadalec razkrije zasebni ključ strežnika in ne, če pride do razkritja zasebnega ključa odjemalca.

Definicija 3.11. Protokol je varen pred *napadom lažnega predstavljanja z razkritim ključem* (angl. key-compromise impersonation attack), če razkrit trajni zasebni ključ neke osebe ne omogoča napadalcu lažnega predstavljanja tej osebi kot neka druga oseba.

Za izvedbo napada lažnega predstavljanja z razkritim ključem, s katerim sta se prva ukvarjala Just in Vaudenay [79], mora napadalec priti do trajnega

ključa uporabnika. Če ima ta svoj ključ shranjen na osebnem računalniku, potem lahko napadalec pride do njega z vdorom v njegov računalnik ali pa slednjega okuži z ustreznim računalniškim virusom. Ko napadalec enkrat dobi trajni ključ uporabnika, se lahko očitno drugim osebam lažno predstavlja v njegovem imenu. Če protokol ni varen pred omenjenim napadom, pa se lahko napadalec dodatno temu uporabniku predstavlja kot neka druga oseba. Pri tem se slednji sploh ne bo zavedal, da se pogovarja z napadalcem, niti da ima ta v posesti njegov zasebni ključ. Takšen napad lahko vodi do resnih posledic, predvsem če se napadalec izdaja kot zaupanja vredna oseba. V tem primeru lahko uporabnik napadalcu razkrije zelo občutljive in zaupne informacije.

Definicija 3.12. Protokol je varen pred *napadom deljenja ključa z neznano osebo* (angl. unknown key-share attack), če napadalec ne more prepričati neke osebe, da si deli ključ z drugo osebo, čeprav si ga v resnici deli s tretjo.

Napadalec torej ne more prepričati Anite, da si deli ključ s Cenetom, medtem ko si ga v resnici deli z Bojanom. Posledice napada deljenja ključa z neznano osebo, ki so ga prvi predstavili Diffie, Oorschot in Wiener [49], si lahko ogledamo na naslednjem primeru. Denimo, da želi Bojan na bankomatu z avtomatskim pologom gotovine položiti neko količino denarja na svoj bančni račun. Potem v bankomat vstavi bankovce in svojo bančno kartico, ter pošlje zaupno sporočilo banki Aniti, v katerem ji sporoča, da bo ta denar naložil na svoj račun. Če poslano sporočilo ne vsebuje dodatnega overjanja pošiljatelja, npr. zaradi manjše velikosti sporočila in hitrejšega pologa gotovine, potem lahko napadalec z uspešnim napadom prepriča Anito, da se pogovarja s Cenetom. Zato bo denar naložila na njegov in ne na Bojanov račun.

Uporaba napada je za nekatere kriptografe vprašljiva, saj obstajajo metode, s katerimi lahko napad preprečimo. V nekaterih primerih je dovolj, da certifikatne agencije pred izdajo digitalnega potrdila preverijo, ali uporabnik pozna zasebni ključ. V ostalih pa lahko protokolu dodamo dodaten krog, v katerem udeleženca izmenjata svoji identiteti, ali pa slednji vključimo v izračun sejnega ključa iz skupne skrivnosti. V zadnjem primeru tako Anita in Bojan ne bosta izračunala istega sejnega ključa. Ker nekatere rešitve vpli-

vajo na učinkovitost, bi jih avtorji morali upoštevati že v definiciji protokola. Zato protokole, ki so ranljivi za ta napad, ne uvrščamo med varne.

Definicija 3.13. Protokol ima lastnost *nadzor ključa* (angl. key control), če je sejni ključ določen s strani vseh udeležencev.

Z nadzorom ključa je tesno povezana tudi naslednja lastnost.

Definicija 3.14. V protokolu se udeleženci dogovorijo za *svež ključ* (angl. fresh key), če protokol nudi zagotovilo, da je ključ nov in da se udeleženci zaradi vmešavanja napadalca ali neke pooblaščene osebe niso dogovorili za star ključ, ki je bil nekoč že uporabljen v seji.

V splošnem od varnih protokolov za overjen dogovor o ključu zahtevamo, da imajo vse zgoraj naštete lastnosti. Kadar pa je protokol namenjen izključno za uporabo v določeni aplikaciji, lahko katero izmed njih zaradi učinkovitosti tudi izpustimo, odvisno od zastavljenih varnostnih zahtev.

3.2.3 Napadi na dogovore o ključu

Dokončen seznam napadov na protokole za dogovor o ključu je nemogoče sestaviti, saj se vedno pojavljajo novi napadi, medtem ko se znani le izboljšujejo. Zato se moramo pri sestavi novih protokolov zavedati, da nikoli ne bomo poznali vseh napadov ali imeli na voljo dokazov, da drugačni napadi ne obstajajo.

Znane napade lahko glede na njihove lastnosti razdelimo v različne razrede, med katerimi so za nas najbolj pomembni naslednji.

- *Napad s ponavljanjem* (angl. replay attack). Pri tem napadu ima napadalec na voljo vsa sporočila, ki so bila poslana v prejšnjih sejah protokola. Le-ta lahko kasneje v celoti ali deloma ponovno uporabi pri razbijanju protokola.

- *Napad lažnega predstavljanja* (angl. impersonation attack). S tem napadom se lahko napadalec ostalim udeležencem v protokolu predstavi kot neka legitimna oseba. Pri dvostranskih dogovorih o ključu te napade delimo v dva razreda, glede na vlogo napadalca v protokolu. Pri

napadu *lažnega predstavljanja pobudnika* napadalec začne protokol s pošiljanjem prvega sporočila, medtem ko se pri napadu *lažnega predstavljanja naslovnika* napadalec odzove na takšno sporočilo. V protokolih za dogovor o ključu imajo lahko ti napadi zelo resne posledice, predvsem če se napadalec lažno predstavi kot sodnik, revizor ali skrbnik tajnih ključev.

- *Napad vmesne osebe* (angl. man-in-the-middle attack). S tem napadom želi napadalec prisluškovati in nadzorovati pogovor med dvema osebama. To stori tako, da izmenjana sporočila prestreže in jih nadomesti s svojimi. Z zamenjavo sporočil obe osebi prepriča, da komunicirata preko varnega kanala, čeprav v resnici njun pogovor poteka preko napadalca. V dvostranskem protokolu za dogovor o ključu je napad vmesne osebe možen le, če se lahko napadalec lažno predstavlja kot pobudnik na eni strani in naslovnik na drugi.

- *Napad notranje osebe* (angl. insider attack). V ta razred sodijo vsi napadi izvedeni s strani notranjega napadalca. Slednji ima lahko pod nadzorom zlonamerne udeležence protokola in dostop do nekaterih dodatnih informacij, kot so npr. sejni in zasebni ključi. Notranji napadalec lahko te pridobi na privilegiran način, običajno z zaroto ali izsiljevanjem, s fizičnim dostopom do tajnih virov podatkov itd.

- *Napad s slovarjem* (angl. dictionary attack). Pri tem napadu želi napadalec najti zasebni ali sejni ključ z zaporednim preskušanjem vseh ključev iz seznama, ki ga imenujemo slovar. Za razliko od napada z grobo silo, kjer napadalec sistematično preveri velik del vseh možnih ključev, pri tem napadu preveri le najbolj verjetne. Takšni napadi so pogosto uspešni pri razkrivanju osebnih gesel uporabnikov, saj si ti običajno izbirajo kratka in šibka gesla, katere je možno enostavno predvideti.

3.3 Delitev dogovorov o ključu

Protokole za dogovor o ključu lahko razdelimo v dve skupini glede na število aktivnih udeležencev, v dvostranske in večstranske protokole. Pri slednjih so zanimivi tudi tristranski protokoli, katere je možno učinkovito sestaviti z uporabo bilinearnih parjenj. Poseben primer protokolov za dogovor o ključu so tudi protokoli na osnovi identitete, ki odpravijo potrebo po uporabi digitalnih potrdil, in protokoli na osnovi gesel. Vseh pet vrst protokolov in njihove glavne predstavnike bomo podrobneje spoznali v nadaljevanju.

3.3.1 Dvostranski protokoli

Kadar govorimo o protokolih za dogovor o ključu, imamo v mislih dvostranske protokole, s katerimi se lahko dve osebi dogovorita za skupni sejni ključ. To lahko storita kljub temu, da se nista še nikoli srečali in si zato ne delita nobene skupne skrivnosti. Ti protokoli igrajo danes ključno vlogo pri zagotavljanju varnosti na spletu, saj so glavni gradniki višje nivojskih protokolov, kot so npr. protokoli SSL/TLS [47], SSH [156] in IPSec [83].

Diffie-Hellmanov protokol

Prvi in najbolj znan dvostranski protokol za dogovor o ključu sta leta 1976 predstavila Diffie in Hellman [48]. Pri slednjem lahko izmenjava sporočil poteka preko javnega kanala, saj tudi če obstaja možnost, da napadalec prisluškuje pogovoru, še zmeraj ne bo mogel v doglednem času izračunati sejnega ključa. Osnovna verzija protokola uporablja računanje v multiplikativni grupi $\mathbb{Z}_q^*$ in je izjemno učinkovita, saj je za dogovor o ključu potrebno izmenjati le dve kratki sporočili.

Protokol 1 Diffie-Hellmanov dogovor o ključu

1. *Priprava (izbira javnih parametrov).* Naj bosta p in q veliki praštevili, tako da velja $p \mid q-1$, in $g \in \mathbb{Z}_q^*$ element reda p.

2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}: &\quad A \quad \text{(1a)} \\ \mathcal{B} \to \mathcal{A}: &\quad B \quad \text{(1b)} \end{aligned}$$

3. *Koraki v protokolu.* Anita in Bojan se dogovorita za sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere svoj začasni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A = g^a \bmod q$ in ga v sporočilu (1a) pošlje Bojanu.

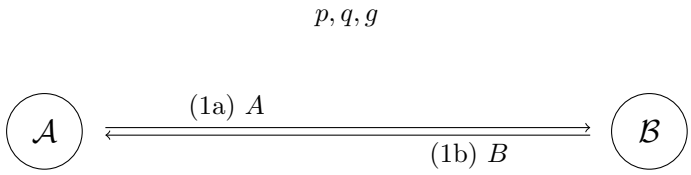
   Podobno Bojan izbere svoj začasni zasebni ključ $b \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $B = g^b \bmod q$ in ga v sporočilu (1b) pošlje Aniti.

   (b) Po izmenjavi sporočil Anita izračuna skupno skrivnost po enačbi $\sigma_{\mathcal{A}} = B^a \bmod q$ in Bojan po enačbi $\sigma_{\mathcal{B}} = A^b \bmod q$. Če sta izračuna pravilna, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in oba udeleženca lahko izpeljeta skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.

Ob zaključku protokola Anita in Bojan varno izbrišeta začasni vrednosti a in b, tako da jima ostane le še skupni sejni ključ $K = \mathrm{kdf}(g^{ab} \bmod q)$.

Varnost opisanega protokola temelji na računskem Diffie-Hellmanovem problemu in sorodnem problemu diskretnega logaritma (glej §2.2). Protokol nudi zaščito pred pasivnimi napadalci (prisluškovalci), ne nudi pa zaščite pred aktivnimi, ki lahko sporočila prestrezajo, spreminjajo in vstavljajo nova. Že dolgo časa je namreč znano, da je protokol ranljiv za napad vmesne osebe, saj ni overjen. Tako se lahko napadalec Aniti lažno predstavi kot Bojan in obratno. S tem oba prepriča, da se pogovarjata drug z drugim, čeprav v resnici njun pogovor poteka preko napadalca.

Diffie-Hellmanov dogovor o ključu lahko izvedemo v poljubni grupi, če v njej problem diskretnega logaritma ni preprosto rešljiv. V kriptografiji zato



(a)

$$\begin{aligned} \mathcal{A}: &\quad a \xleftarrow{\$} \mathbb{Z}_p^* \\ \mathcal{B}: &\quad b \xleftarrow{\$} \mathbb{Z}_p^* \\ \mathcal{A} \to \mathcal{B}: &\quad A = g^a \bmod q \qquad (1a) \\ \mathcal{B} \to \mathcal{A}: &\quad B = g^b \bmod q \qquad (1b) \end{aligned}$$

(b)

$$\begin{aligned} \mathcal{A}: &\quad \sigma = B^a \bmod q \\ \mathcal{B}: &\quad \sigma = A^b \bmod q \\ \mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma) \end{aligned}$$

Slika 3.1: Diffie-Hellmanov dogovor o ključu

zaradi učinkovitosti običajno grupo $\mathbb{Z}_q^*$ zamenjamo z grupo točk na eliptični krivulji $E(\mathbb{F}_q)$. Takšnemu protokolu pravimo Diffie-Hellmanov dogovor o ključu nad eliptično krivuljo.

MQV protokol

Protokol MQV za overjen dogovor o ključu so leta 1995 predstavili Menezes, Qu in Vanstone [103], leta 1998 pa sta mu nekaj popravkov dodala še Law in Solinas [93]. Trenutno obstaja več verzij tega protokola, nekatere so celo patentirane s strani podjetja Certicom Corp.[1], ki je sedaj v lasti kanadskega proizvajalca pametnih telefon in tablic BlackBerry Limited. Ker je protokol overjen, ne nudi zaščite samo pred pasivnimi napadalci, temveč tudi pred aktivnimi. Zasnovan je na Diffie-Hellmanovem protokolu in je trenutno še vedno najbolj učinkovit protokol za overjen dogovor o ključu. Zato je tudi priporočen s strani inštituta za standardizacijo in tehnologijo NIST ter ameriške agencije za nacionalno varnost NSA (suite B) in vključen v mnoge standarde, kot so IEEE [74], ANSI [4, 5] ter ISO/IEC [75].

Podobno kot Diffie-Hellmanov protokol je tudi MQV protokol možno defi-

[1] `http://www.certicom.com/`

nirati nad poljubno končno grupo. Vendar pa se v praksi zaradi učinkovitosti uporabljajo zgolj grupe točk na eliptični krivulji. Prav tako je izjemno učinkovit, saj je overjen dogovor o ključu možno opraviti z izmenjavo dveh sporočil v enem samem krogu protokola.

Protokol 2 MQV dogovor o ključu

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bo E eliptična krivulja nad končnim obsegom $\mathbb{F}_q$, kjer je q neko veliko praštevilo ali potenca števila 2, ki vsebuje aditivno grupo točk $\mathbb{G} = \langle P \rangle$ velikega praštevilskega reda p in k njen kofaktor.

   (b) Anita izbere svoj trajni zasebni ključ $d_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$ in izračuna pripadajoči javni ključ $Q_{\mathcal{A}} = d_{\mathcal{A}}P$. Zasebni ključ varno shrani, medtem ko javni ključ overi pri certifikatni agenciji in objavi v digitalnem potrdilu $\mathrm{Cert}_{\mathcal{A}}$.

2. *Izmenjava sporočil.*

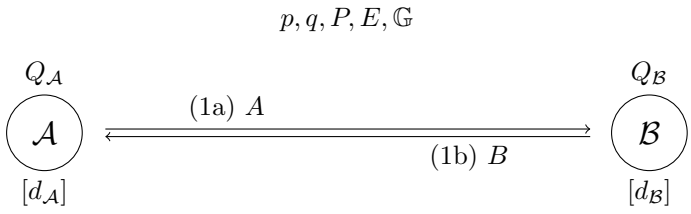
$$\begin{aligned} \mathcal{A} \to \mathcal{B}: &\quad A \quad (1a) \\ \mathcal{B} \to \mathcal{A}: &\quad B \quad (1b) \end{aligned}$$

3. *Koraki v protokolu.* Anita in Bojan se lahko po uspešno izvedeni pripravi, v kateri si izbereta trajna zasebna ključa $d_{\mathcal{A}}$ in $d_{\mathcal{B}}$ ter svoja javna ključa $Q_{\mathcal{A}}$ in $Q_{\mathcal{B}}$ objavita v digitalnih potrdilih $\mathrm{Cert}_{\mathcal{A}}$ ter $\mathrm{Cert}_{\mathcal{B}}$, dogovorita za sejni ključ. To storita z izvedbo naslednjih korakov, v katerih smo s $\bar{T}$ označili prvih $L = \lceil (\log_2 p + 1)/2 \rceil$ bitov koordinate x točke T.

   (a) Anita izbere svoj začasni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A = aP$ in ga v sporočilu (1a) pošlje Bojanu.

   Podobno Bojan izbere svoj začasni zasebni ključ $b \xleftarrow{\$} \mathbb{Z}_p^*$ in začasni javni ključ $B = bP$ v sporočilu (1b) pošlje Aniti.

   (b) Po izmenjavi sporočil Anita iz Bojanovega digitalnega potrdila $\mathrm{Cert}_{\mathcal{B}}$ prebere njegov trajni javni ključ $Q_{\mathcal{B}}$ in izračuna skupno skrivnost $\sigma_{\mathcal{A}} = k(a + \bar{A}d_{\mathcal{A}})(B + \bar{B}Q_{\mathcal{B}})$. To stori tudi Bojan, le da on iz Anitinega potrdila $\mathrm{Cert}_{\mathcal{A}}$ prebere javni ključ $Q_{\mathcal{A}}$ in ga uporabi



(a)

$$\begin{aligned} \mathcal{A}: &\quad a \xleftarrow{\$} \mathbb{Z}_p^* \\ \mathcal{B}: &\quad b \xleftarrow{\$} \mathbb{Z}_p^* \\ \mathcal{A} \to \mathcal{B}: &\quad A = aP \qquad (1a) \\ \mathcal{B} \to \mathcal{A}: &\quad B = bP \qquad (1b) \end{aligned}$$

(b)

$$\begin{aligned} \mathcal{A}: &\quad \sigma = k(a + \bar{A}d_{\mathcal{A}})(B + \bar{B}Q_{\mathcal{B}}) \\ \mathcal{B}: &\quad \sigma = k(b + \bar{B}d_{\mathcal{B}})(A + \bar{A}Q_{\mathcal{A}}) \\ \mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma) \end{aligned}$$

Slika 3.2: MQV dogovor o ključu

za izračun skupne skrivnosti $\sigma_{\mathcal{B}} = k(b + \bar{B}d_{\mathcal{B}})(A + \bar{A}Q_{\mathcal{A}})$. Če sta izračuna pravilna, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in oba udeleženca lahko izpeljeta skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.

Ni se težko prepričati, da si na koncu protokola Anita in Bojan delita isto skrivnost $\sigma = k(a + \bar{A}d_{\mathcal{A}})(b + \bar{B}d_{\mathcal{B}})P$ in posledično isti sejni ključ.

3.3.2 Tristranski protokoli

Tristranski oz. tripartitni protokoli za dogovor o ključu omogočajo trem osebam varno vzpostavitev skupnega sejnega ključa, tako da lahko vse tri hkrati sodelujejo v isti seji. V praksi so uporabni predvsem v primerih, ko želita dve osebi varno komunicirati v prisotnosti tretje. Slednja lahko v pogovoru nastopa kot voditelj, sodnik, revizor ali skrbnik tajnih ključev. Tristranske protokole običajno ločimo od večstranskih, saj jih je možno učinkovito sestaviti z uporabo bilinearnih parjenj.

Jouxov protokol

Najučinkovitejši tristranski protokol za dogovor o ključu je leta 2000 predstavil Joux [76]. Z uporabo bilinearnega parjenja je sestavil preprost protokol, katerega varnost temelji na bilinearnem Diffie-Hellmanovem problemu (glej def. 2.46). Ta je podobno kot Diffie-Hellmanov protokol neoverjen in zato ranljiv za napad vmesne osebe. Protokol je izjemno učinkovit, saj se izmenjava sporočil med osebami zgodi hkrati v enem samem krogu. Leto dni po njegovi objavi je Verheul [150] predlagal majhne izboljšave, z upoštevanjem katerih je pri izmenjavi sporočil dovolj poslati le eno točko eliptične krivulje. Izboljšano verzijo protokola lahko predstavimo v naslednji obliki.

Protokol 3 Jouxov tristranski dogovor o ključu

1. *Priprava (izbira javnih parametrov).* Naj bo $\mathbb{G}_1$ grupa točk na eliptični krivulji E, točka P njen generator velikega praštevilskega reda p, $\mathbb{G}_2$ multiplikativna grupa in $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ bilinearno parjenje.

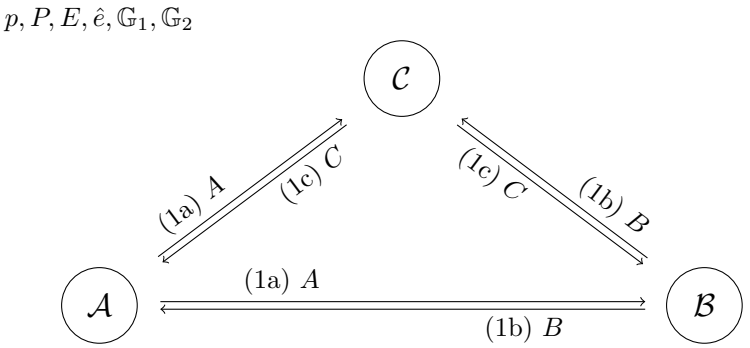
2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}, \mathcal{C} : &\quad A \quad \text{(1a)} \\ \mathcal{B} \to \mathcal{A}, \mathcal{C} : &\quad B \quad \text{(1b)} \\ \mathcal{C} \to \mathcal{A}, \mathcal{B} : &\quad C \quad \text{(1c)} \end{aligned}$$

3. *Koraki v protokolu.* Anita, Bojan in Cene se dogovorijo za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere svoj začasni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A = aP$ in ga v sporočilu (1a) pošlje Bojanu ter Cenetu. Podobno Bojan in Cene izbereta svoja začasna ključa $b \xleftarrow{\$} \mathbb{Z}_p^*$ oz. $c \xleftarrow{\$} \mathbb{Z}_p^*$ in začasna javna ključa $B = bP$ oz. $C = cP$ pošljeta v sporočilih (1b) in (1c) vsem udeležencem protokola.

   (b) Po izmenjavi sporočil Anita izračuna skrivnost $\sigma_{\mathcal{A}} = \hat{e}(B, C)^a$, Bojan $\sigma_{\mathcal{B}} = \hat{e}(A, C)^b$ in Cene $\sigma_{\mathcal{C}} = \hat{e}(A, B)^c$. Če so izračuni pravilni, potem velja enakost $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}} = \sigma_{\mathcal{C}}$ in vsi trije lahko izpeljejo skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.



(a)

$$\begin{aligned}
\mathcal{A}: &\quad a \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{B}: &\quad b \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{C}: &\quad c \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{A} \to \mathcal{B}, \mathcal{C}: &\quad A = aP \qquad (1a) \\
\mathcal{B} \to \mathcal{A}, \mathcal{C}: &\quad B = bP \qquad (1b) \\
\mathcal{C} \to \mathcal{A}, \mathcal{B}: &\quad C = cP \qquad (1c)
\end{aligned}$$

(b)

$$\begin{aligned}
\mathcal{A}: &\quad \sigma = \hat{e}(B, C)^a \\
\mathcal{B}: &\quad \sigma = \hat{e}(A, C)^b \\
\mathcal{C}: &\quad \sigma = \hat{e}(A, B)^c \\
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad K = \mathrm{kdf}(\sigma)
\end{aligned}$$

Slika 3.3: Jouxov tristranski dogovor o ključu

Zaradi lastnosti bilinearnih parjenj Anita, Bojan in Cene ob koncu protokola iz skupne skrivnosti $\sigma = \hat{e}(P, P)^{abc}$ izračunajo isti sejni ključ.

Jouxovemu protokolu so sledili številni novi tristranski protokoli za (overjen) dogovor o ključu, vendar pa do danes protokola še nihče ni posplošil na štiri ali več oseb. Zato je ta protokol še vedno edina posplošitev Diffie-Hellmanovega dogovora o ključu z enim samim krogom, v katerem vsak udeleženec javno objavi le eno sporočilo. Posplošitev na več oseb pa še vedno ostaja odprt kriptografski problem.

3.3.3 Večstranski protokoli

Protokole za dogovor o ključu, v katerih nastopajo vsaj tri osebe, imenujemo večstranski oz. konferenčni protokoli. Za njih je značilno, da jih lahko posplošimo na poljubno število oseb. Zaradi svoje splošnosti so zato tudi manj učinkoviti kot prej omenjeni dvostranski in tristranski protokoli.

Diffie-Hellmanov večstranski protokol

Preprost večstranski protokol za dogovor o ključu lahko sestavimo s posplošitvijo Diffie-Hellmanovega protokola. V nadaljevanju bomo predstavili posplošitev na n oseb, ki jih bomo označili z $\mathcal{A}_0$, $\mathcal{A}_1$, $\ldots$, $\mathcal{A}_{n-1}$. Pri tem bomo upoštevali, da se indeksi v oznakah računajo po modulu števila n.

Protokol 4 Diffie-Hellmanov večstranski dogovor o ključu

1. *Priprava* (glej protokol 1).

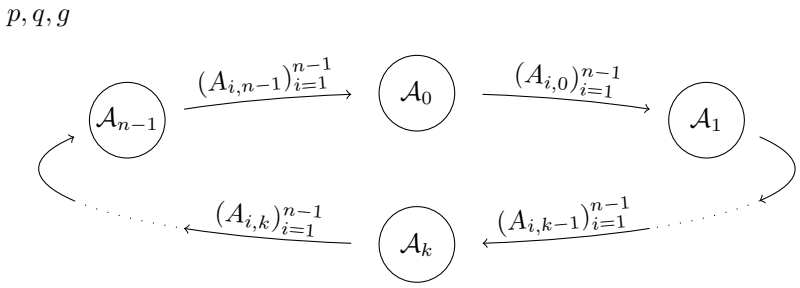
2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A}_k \to \mathcal{A}_{k+1}: &\quad A_{1,k} \quad (1) \\ \mathcal{A}_k \to \mathcal{A}_{k+1}: &\quad A_{2,k} \quad (2) \\ &\quad \vdots \\ \mathcal{A}_k \to \mathcal{A}_{k+1}: &\quad A_{n-1,k} \quad (n-1) \end{aligned}$$

3. *Koraki v protokolu.* Vsak udeleženec protokola $\mathcal{A}_k$ za dogovor o ključu izvede naslednje korake.

   (a) V prvem krogu izbere svoj začasni zasebni ključ $a_k \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A_{1,k} = g^{a_k} \bmod q$ in ga v sporočilu (1) pošlje osebi $\mathcal{A}_{k+1}$.

   (b) V i-tem krogu, $2 \leq i \leq n-1$, od osebe $\mathcal{A}_{k-1}$ prejme sporočilo z vrednostjo $A_{i-1,k-1}$, ki je bilo poslano v prejšnjem krogu protokola. Nato izračuna vrednost $A_{i,k} = (A_{i-1,k-1})^{a_k} \bmod q$ in jo v sporočilu (i) pošlje osebi $\mathcal{A}_{k+1}$.



(a)
$$\mathcal{A}_k: \quad a_k \xleftarrow{\$} \mathbb{Z}_p^*$$
$$\mathcal{A}_k \to \mathcal{A}_{k+1}: \quad A_{1,k} = g^{a_k} \bmod q \qquad (1)$$

(b)
$$\mathcal{A}_k \to \mathcal{A}_{k+1}: \quad A_{2,k} = (A_{1,k-1})^{a_k} \bmod q \qquad (2)$$
$$\vdots$$
$$\mathcal{A}_k \to \mathcal{A}_{k+1}: \quad A_{n-1,k} = (A_{n-2,k-1})^{a_k} \bmod q \qquad (n-1)$$

(c)
$$\mathcal{A}_k: \quad \sigma = (A_{n-1,k-1})^{a_k} \bmod q$$
$$\mathcal{A}_k: \quad K = \mathrm{kdf}(\sigma)$$

Slika 3.4: Diffie-Hellmanov večstranski dogovor o ključu

(c) Ko so vsa sporočila izmenjana, oseba $\mathcal{A}_k$ izračuna skupno skrivnost po enačbi $\sigma_{\mathcal{A}_k} = (A_{n-1,k-1})^{a_k} \bmod q$, kjer je vrednost $A_{n-1,k-1}$ prejela v zadnjem sporočilu od osebe $\mathcal{A}_{k-1}$. Če so izračuni pravilni, potem vse osebe izračunajo isto skrivnost $\sigma = \sigma_{\mathcal{A}_0} = \cdots = \sigma_{\mathcal{A}_{n-1}}$ in iz nje izpeljejo sejni ključ $K = \mathrm{kdf}(\sigma)$.

Ni se težko prepričati, da na koncu protokola vsi udeleženci iz skupne skrivnosti $\sigma = g^{a_0 a_1 \cdots a_{n-1}} \bmod q$ izračunajo isti sejni ključ.

3.3.4 Protokoli na osnovi identitete

Poseben primer protokolov za dogovor o ključu so protokoli na osnovi identitete, pri katerih se za javni ključ uporabnika vzame kar njegovo identifikacijsko informacijo, kot je npr. ime, naslov, davčna številka ali e-poštni

naslov. S takšnimi protokoli se znebimo potrebe po digitalnih potrdilih, saj so javni ključi implicitno overjeni. Na žalost pa imajo ti protokoli tudi svoje pomanjkljivosti, saj je v sistem potrebno vključiti zaupanja vreden generator zasebnih ključev, ki uporabnikom preko varnega kanala izdaja zasebne ključe.

Generator zasebnih ključev ima v lasti glavni tajni ključ, s katerim lahko izračuna vse zasebne ključe uporabnikov. Ta ključ mu tudi omogoča, da v primeru konfliktov med uporabniki iz izmenjanih sporočil razkrije dogovorjeni sejni ključ. To pa pravzaprav pomeni, da celotna varnost sistema temelji na eni sami osebi. Za odpravo te težave nekateri protokoli glavni tajni ključ razdelijo med različne generatorje. To običajno storijo z uporabo shem za deljenje skrivnosti, ki jih je prvi predlagal Shamir [135]. Takšen pristop je seveda boljši, saj zaupanje porazdelimo med različne osebe in tako nihče nima dostopa do celotnega tajnega ključa.

Prvi protokol za dogovor o ključu na osnovi identitete je leta 1987 predstavil Okamoto [116]. Varnost slednjega temelji na problemu razcepa (sestavljenih) števil in problemu RSA. Tri leta pozneje je Günther objavil dva nova protokola [57], ki ju je zasnoval na idejah Diffie-Hellmanovega dogovora o ključu in ElGamalove sheme za digitalni podpis. Z odkritjem učinkovitih bilinearnih parjenj je področje kriptografije na osnovi identitete zelo zaživelo, zato lahko danes v literaturi najdemo veliko novih protokolov.

Smartov protokol

Leta 2002 je Smart objavil prvi protokol za dogovor o ključu na osnovi identitete, ki izkorišča lastnosti bilinearnih parjenj [141]. Slednji temelji na Jouxovem tristranskem dogovoru o ključu (glej protokol 3) in šifrirni shemi na osnovi identitete, ki sta jo leta 2001 predstavila Boneh in Franklin [24]. Protokol uporablja generator zasebnih ključev, ki uporabnikom preko varnega kanala izdaja trajne zasebne ključe.

Protokol 5 Smartov dogovor o ključu na osnovi identitete

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bosta $\mathbb{G}_1$ in $\mathbb{G}_2$ grupi, $P \in \mathbb{G}_1$ element reda p, preslikava $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ bilinearno parjenje in $H : \{0,1\}^* \to \mathbb{G}_1$ kriptografska zgoščevalna funkcija.

   (b) Generator zasebnih ključev izbere glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna glavni javni ključ $S = sP$ in ga javno objavi.

   (c) Aniti z identiteto $\mathrm{ID}_{\mathcal{A}}$ generator zasebnih ključev izračuna zgostitev $Q_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}})$ in njen trajni zasebni ključ $S_{\mathcal{A}} = sQ_{\mathcal{A}}$. Slednega ji preko varnega kanala pošlje v trajno last.

2. *Izmenjava sporočil.*

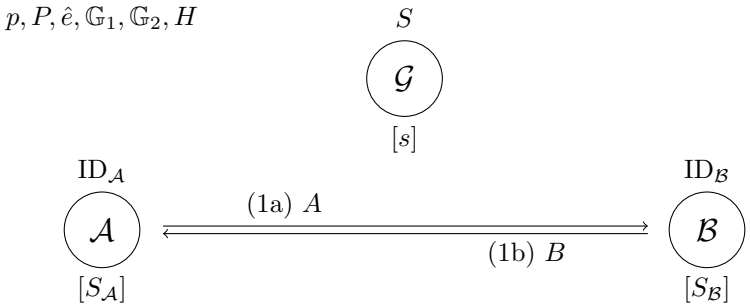
$$\mathcal{A} \to \mathcal{B} : \quad A \quad \text{(1a)}$$
$$\mathcal{B} \to \mathcal{A} : \quad B \quad \text{(1b)}$$

3. *Koraki v protokolu.* Anita in Bojan se z javno znanima identitetama $\mathrm{ID}_{\mathcal{A}}$ in $\mathrm{ID}_{\mathcal{B}}$ po uspešno izvedeni pripravi, v kateri jima generator zasebnih ključev pošlje zasebna ključa $S_{\mathcal{A}}$ in $S_{\mathcal{B}}$, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere svoj začasni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A = aP$ in ga v sporočilu (1a) pošlje Bojanu.
   Podobno Bojan izbere začasni zasebni ključ $b \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $B = bP$ in ga v sporočilu (1b) pošlje Aniti.

   (b) Po izmenjavi sporočil Anita izračuna zgostitev Bojanove identitete $Q_{\mathcal{B}} = H(\mathrm{ID}_{\mathcal{B}})$ in skupno skrivnost $\sigma_{\mathcal{A}} = \hat{e}(aQ_{\mathcal{B}}, S)\,\hat{e}(S_{\mathcal{A}}, B)$. To stori tudi Bojan, tako da najprej izračuna zgostitev $Q_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}})$ in nato še skrivnost $\sigma_{\mathcal{B}} = \hat{e}(bQ_{\mathcal{A}}, S)\,\hat{e}(S_{\mathcal{B}}, A)$. Če sta izračuna pravilna, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in oba udeleženca lahko izpeljeta skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.

Iz opisa protokola je očitno, da Anita in Bojan izračunata isto skupno skrivnost $\sigma = \hat{e}(Q_{\mathcal{A}}, P)^{bs}\,\hat{e}(Q_{\mathcal{B}}, P)^{as}$ in posledično isti sejni ključ.



(a)

$$\begin{aligned}
\mathcal{A}: &\quad a \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{B}: &\quad b \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{A} \to \mathcal{B}: &\quad A = aP \qquad (1a) \\
\mathcal{B} \to \mathcal{A}: &\quad B = bP \qquad (1b)
\end{aligned}$$

(b)

$$\begin{aligned}
\mathcal{A}: &\quad Q_\mathcal{B} = H(\mathrm{ID}_\mathcal{B}) \\
\mathcal{B}: &\quad Q_\mathcal{A} = H(\mathrm{ID}_\mathcal{A}) \\
\mathcal{A}: &\quad \sigma = \hat{e}(aQ_\mathcal{B}, S)\,\hat{e}(S_\mathcal{A}, B) \\
\mathcal{B}: &\quad \sigma = \hat{e}(bQ_\mathcal{A}, S)\,\hat{e}(S_\mathcal{B}, A) \\
\mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma)
\end{aligned}$$

Slika 3.5: Smartov dogovor o ključu na osnovi identitete

3.3.5 Protokoli na osnovi gesel

Uporabniška imena in gesla so še vedno najbolj pogost način za preverjanje identitete uporabnikov. Z njimi se prijavljamo v operacijske sisteme, brezžična in družbena omrežja, dostopamo do elektronskega bančnega računa, beremo e-pošto itd. V kriptografiji javnih ključev osebna gesla uporabljamo za shranjevanje trajnih ključev v šifrirani obliki in za njihovo zaščito na kriptografskih napravah, kot so npr. pametne kartice. Popularna so predvsem zato, ker si jih je lažje zapomniti kot naključno izbrane ključe z zelo visoko entropijo. Na žalost pa to s seboj prinese tudi mnoge nevšečnosti, saj si uporabniki pogosto izbirajo šibka gesla in ista gesla uporabljajo v različnih aplikacijah.

Protokoli za dogovor o ključu na osnovi gesel omogočajo dvema osebama, ki si delita skupno geslo, vzpostavitev sejnega ključa preko javnega kanala. Pri tem upoštevajo, da gesla niso enakomerno izbrana iz množice vseh možnih gesel in da je le-ta pogosto majhna, kot je to v primeru štirimestne osebne številke PIN. Zato so ti protokoli odporni na napade s slovarjem, v katerih napadalec z grobo silo preveri gesla iz neke majhne množice oz. slovarja.

Šifrirana izmenjava ključa

Prve korake na tem področju sta naredila Bellovin in Merritt, ko sta leta 1992 predstavila protokol za šifrirano izmenjavo ključa (angl. encrypted key exchange) [19]. Le-ta za varen prenos podatkov preko javnega kanala, za njihovo overitev in za dogovor o ključu uporablja orodja simetrične ter asimetrične kriptografije. Zato obstaja tudi več različic protokola, med katerimi nekatere temeljijo na kriptosistemih RSA [129] in ElGamal [50]. Predlagani protokol naj bi bil odporen na napade s slovarjem in napade s ponavljanjem (glej §3.2.3) ter varen, tudi če si uporabniki izberejo šibka gesla.

V nadaljevanju bomo predstavili protokol za šifrirano izmenjavo ključa, ki je zasnovan na Diffie-Hellmanovem dogovoru o ključu.

Protokol 6 Šifrirana izmenjava ključa (Diffie-Hellman)

1. *Priprava (izbira javnih parametrov).* Naj bosta p in q praštevili, tako da velja $p \mid q-1$, nadalje $g \in \mathbb{Z}_q^*$ element reda p in E_k šifrirna ter D_k odšifrirna funkcija varne simetrične šifre (npr. AES), kjer smo s k označili simetrični ključ.

2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}: &\quad X & (1)\\ \mathcal{B} \to \mathcal{A}: &\quad Y, C_{\mathcal{B}} & (2)\\ \mathcal{A} \to \mathcal{B}: &\quad C_{\mathcal{AB}} & (3)\\ \mathcal{B} \to \mathcal{A}: &\quad C_{\mathcal{A}} & (4) \end{aligned}$$

3. *Koraki v protokolu.* Anita in Bojan, ki si delita tajni simetrični ključ (geslo), za dogovor o ključu opravita naslednje korake.

   (a) Anita izbere svoj začasni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A = g^a \bmod q$ in tajnopis $X = E_{\text{geslo}}(A)$ v sporočilu (1) pošlje Bojanu.

   (b) Ko Bojan prejme sporočilo, tudi sam izbere svoj začasni zasebni ključ $b \xleftarrow{\$} \mathbb{Z}_p^*$ in izračuna začasni javni ključ $B = g^b \bmod q$. Iz prejetega sporočila prebere tajnopis X, ga odšifrira in Anitin javni ključ $A = D_{\text{geslo}}(X)$ uporabi za izračun skrivnosti $\sigma_{\mathcal{B}} = A^b \bmod q$ ter izpeljavo sejnega ključa $K = \text{kdf}(\sigma_{\mathcal{B}})$. Nato pripravi tajnopisa $Y = E_{\text{geslo}}(B)$ in $C_{\mathcal{B}} = E_K(\text{izziv}_{\mathcal{B}})$ ter ju v sporočilu (2) pošlje nazaj Aniti.

   (c) Iz Bojanovega sporočila Anita prebere tajnopis Y, ga odšifrira in iz njegovega javnega ključa $B = D_{\text{geslo}}(Y)$ izračuna skupno skrivnost $\sigma_{\mathcal{A}} = B^a \bmod q$, katero uporabi za izpeljavo sejnega ključa $K = \text{kdf}(\sigma_{\mathcal{A}})$. Če so bili vsi izračuni pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in oba udeleženca sta izračunala isti sejni ključ. S slednjim lahko Anita odšifrira še drugi tajnopis iz Bojanovega sporočila in čistopis $\text{izziv}_{\mathcal{B}} = D_K(C_{\mathcal{B}})$ skupaj s svojim izzivom zašifrira s sejnim ključem K. Tajnopis $C_{\mathcal{AB}} = E_K(\text{izziv}_{\mathcal{A}}, \text{izziv}_{\mathcal{B}})$ nato pošlje v sporočilu (3) Bojanu.

(d) Po prejetju Anitinega sporočila Bojan odšifrira tajnopis $C_{\mathcal{AB}}$ in preveri, ali se njegov izziv v čistopisu $(\text{izziv}_{\mathcal{A}}, \text{izziv}_{\mathcal{B}}) = D_K(C_{\mathcal{AB}})$ ujema s tistim, ki ga je v drugem krogu poslal Aniti. Če preverjanje ni uspešno, prekine izvajanje protokola, sicer sprejme sejni ključ K in tajnopis $C_{\mathcal{A}} = E_K(\text{izziv}_{\mathcal{A}})$ pošlje v sporočilu (4) Aniti.

(e) Ko Anita prejme Bojanov tajnopis $C_{\mathcal{A}}$, ga odšifrira in preveri, ali se $\text{izziv}_{\mathcal{A}} = D_K(C_{\mathcal{A}})$ ujema z njenim izzivom, ki si ga je izbrala v tretjem krogu protokola. Če je preverjanje uspešno, sprejme sejni ključ K in zaključi izvajanje protokola.

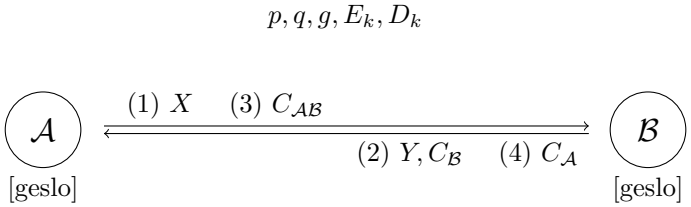
V opisu protokola smo z oznako $\text{izziv}_{\mathcal{A}}$ označili podatke, s katerimi Anita preveri, ali Bojan v resnici pozna sejni ključ K. Če oba udeleženca sledita korakom protokola, potem izračunata isto skupno skrivnost $\sigma = g^{ab} \bmod q$ in posledično isti sejni ključ. Zato so vsa preverjanja uspešna in protokol se pravilno zaključi.

Pri šifrirani izmenjavi ključa morata oba udeleženca poznati skupno geslo. Takšen pristop ni primeren za komunikacijo po modelu odjemalec/strežnik, v katerem si odjemalec deli osebno geslo z zaupanja vrednim strežnikom. Strežniki so namreč pogosta tarča napadalcev, zato obstaja velika verjetnost za razkritje gesel, če so le-ta shranjena na strežniku v obliki čistopisa. Za odpravo te težave sta Bellovin in Merritt leto dni kasneje objavila nov protokol, v katerem strežnik pri sebi hrani le zgostitve uporabniških gesel [20].

Čez čas se je izkazalo, da oba predlagana protokola in tudi nekatere njune izboljšave niso varne. Zato je bilo potrebno narediti korak naprej in varnost novih predlogov utemeljiti z dokazom. To so prvi storili Bellare, Pointcheval in Rogaway [12] ter Boyko, MacKenzie in Patel [28] leta 2000, ko so varnost svojih protokolov za dogovor o ključu na osnovi gesel dokazali v modelu naključnega preroka.

Tristranska šifrirana izmenjava ključa

Pri protokolih na osnovi gesel nastopijo težave, kadar imamo veliko število uporabnikov. Takrat si mora vsak uporabnik za komunikacijo z ostalimi zapomniti mnogo gesel. Da bi zmanjšali število slednjih, so Steiner, Tsudik



(a)
$$\begin{aligned} \mathcal{A}: &\quad a \xleftarrow{\$} \mathbb{Z}_p^*,\ A = g^a \bmod q \\ \mathcal{B}: &\quad b \xleftarrow{\$} \mathbb{Z}_p^*,\ B = g^b \bmod q \\ \mathcal{A} \to \mathcal{B}: &\quad X = E_{\text{geslo}}(A) \end{aligned} \tag{1}$$

(b)
$$\begin{aligned} \mathcal{B}: &\quad A = D_{\text{geslo}}(X),\ \sigma = A^b \bmod q,\ K = \text{kdf}(\sigma) \\ \mathcal{B} \to \mathcal{A}: &\quad Y = E_{\text{geslo}}(B),\ C_{\mathcal{B}} = E_K(\text{izziv}_{\mathcal{B}}) \end{aligned} \tag{2}$$

(c)
$$\begin{aligned} \mathcal{A}: &\quad B = D_{\text{geslo}}(Y),\ \sigma = B^a \bmod q,\ K = \text{kdf}(\sigma), \\ &\quad \text{izziv}_{\mathcal{B}} = D_K(C_{\mathcal{B}}) \\ \mathcal{A} \to \mathcal{B}: &\quad C_{\mathcal{AB}} = E_K(\text{izziv}_{\mathcal{A}}, \text{izziv}_{\mathcal{B}}) \end{aligned} \tag{3}$$

(d)
$$\begin{aligned} \mathcal{B}: &\quad (\text{izziv}_{\mathcal{A}}, \text{izziv}_{\mathcal{B}}) = D_K(C_{\mathcal{AB}}),\ \text{preveri}(\text{izziv}_{\mathcal{B}}) \\ \mathcal{B} \to \mathcal{A}: &\quad C_{\mathcal{A}} = E_K(\text{izziv}_{\mathcal{A}}) \end{aligned} \tag{4}$$

(e)
$$\mathcal{A}: \quad \text{izziv}_{\mathcal{A}} = D_K(C_{\mathcal{A}}),\ \text{preveri}(\text{izziv}_{\mathcal{A}})$$

Slika 3.6: Šifrirana izmenjava ključa (Diffie-Hellman)

in Waidner objavili predlog tristranskega protokola za dogovor o ključu na osnovi gesel [142], ki so ga zasnovali na protokolu za šifrirano izmenjavo ključa. V njem si vsak uporabnik deli le eno geslo s spletnim strežnikom, preko katerega nato poteka dogovor o ključu in overjanje udeležencev. To pa prinese nezaželene posledice, saj je potrebno strežniku popolnoma zaupati, poleg tega pa mora biti navzoč pri vsaki izvedbi protokola.

Protokol 7 Tristranska šifrirana izmenjava ključa (Diffie-Hellman)

1. *Priprava* (glej protokol 6).

2. *Izmenjava sporočil.*

$$\begin{aligned}
\mathcal{A} \to \mathcal{B}: &\quad \mathrm{ID}_{\mathcal{A}}, X && (1)\\
\mathcal{B} \to \mathcal{S}: &\quad \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, X, Y && (2)\\
\mathcal{S} \to \mathcal{B}: &\quad S_{\mathcal{A}}, S_{\mathcal{B}} && (3)\\
\mathcal{B} \to \mathcal{A}: &\quad S_{\mathcal{A}}, C_{\mathcal{B}} && (4)\\
\mathcal{A} \to \mathcal{B}: &\quad C_{\mathcal{A}} && (5)
\end{aligned}$$

3. *Koraki v protokolu.* Anita in Bojan, ki delita tajni simetrični ključ ($\mathrm{geslo}_{\mathcal{A}}$ in $\mathrm{geslo}_{\mathcal{B}}$) z zaupanja vrednim strežnikom, za dogovor o ključu opravita naslednje korake.

   (a) Anita izbere začasni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $A = g^a \bmod q$ in tajnopis $X = E_{\mathrm{geslo}_{\mathcal{A}}}(A \oplus \mathrm{ID}_{\mathcal{B}})$ skupaj s svojo identiteto $\mathrm{ID}_{\mathcal{A}}$ pošlje v sporočilu (1) Bojanu.

   (b) Ko Bojan prejme sporočilo, tudi sam izbere začasni zasebni ključ $b \xleftarrow{\$} \mathbb{Z}_p^*$ in izračuna začasni javni ključ $B = g^b \bmod q$. Nato pripravi tajnopis $Y = E_{\mathrm{geslo}_{\mathcal{B}}}(B \oplus \mathrm{ID}_{\mathcal{A}})$ in ga skupaj s svojo identiteto $\mathrm{ID}_{\mathcal{B}}$ ter prejetimi Anitinim vrednostmi pošlje v sporočilu (2) zaupanja vrednemu strežniku.

   (c) Po prejetju Bojanovega sporočila strežnik izbere število $s \xleftarrow{\$} \mathbb{Z}_p^*$ in odšifrira Anitin ter Bojanov javni ključ $A = D_{\mathrm{geslo}_{\mathcal{A}}}(X) \oplus \mathrm{ID}_{\mathcal{B}}$ oz. $B = D_{\mathrm{geslo}_{\mathcal{B}}}(Y) \oplus \mathrm{ID}_{\mathcal{A}}$. Nato izračuna vrednosti $S_{\mathcal{B}} = A^s \bmod q$ in $S_{\mathcal{A}} = B^s \bmod q$ ter ju v sporočilu (3) pošlje nazaj Bojanu.

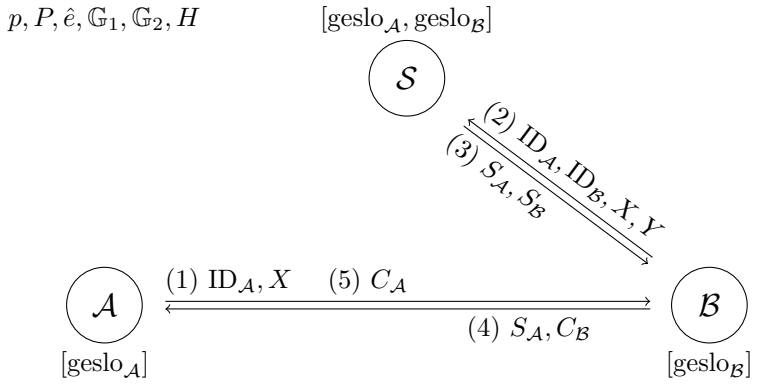
(d) Iz prejetega sporočila Bojan izračuna skrivnost $\sigma_{\mathcal{B}} = S_{\mathcal{B}}^b \bmod q$ in iz nje izpelje sejni ključ $K = \mathrm{kdf}(\sigma_{\mathcal{B}})$. Z njim nato zašifrira Anitin tajnopis X, ki ga je prejel v prvem krogu protokola, in tajnopis $C_{\mathcal{B}} = E_K(X)$ skupaj z vrednostjo $S_{\mathcal{A}}$ pošlje v sporočilu (4) Aniti.

(e) Anita iz Bojanovega sporočila prebere vrednost $S_{\mathcal{A}}$ in jo uporabi za izračun skupne skrivnosti $\sigma_{\mathcal{A}} = S_{\mathcal{A}}^a \bmod q$, iz katere nato izpelje sejni ključ $K = \mathrm{kdf}(\sigma_{\mathcal{A}})$. Če so bili vsi izračuni pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in oba udeleženca sta izračunala isti sejni ključ. Anita nato preveri, ali se tajnopis $X' = D_K(C_{\mathcal{B}})$ ujema s tistim, ki ga je v prvem krogu poslala Bojanu. Če preverjanje ni uspešno, zaključi protokol, sicer sprejme sejni ključ K in tajnopis $C_{\mathcal{A}} = E_K(C_{\mathcal{B}})$ pošlje v sporočilu (5) nazaj Bojanu.

(f) Podobno Bojan po prejetju tajnopisa $C_{\mathcal{A}}$ preveri, ali se odšifriran tajnopis $C'_{\mathcal{B}} = D_K(C_{\mathcal{A}})$ ujema s tistim, ki ga je v četrtem krogu poslal Aniti. Če je preverjanje uspešno, sprejme sejni ključ K, sicer ga zavrne in zaključi izvajanje protokola.

Tudi pri tristranski šifrirani izmenjavi ključa ni težko preveriti, da Anita in Bojan izračunata isto skupno skrivnost $\sigma = g^{abs} \bmod q$ in posledično isti sejni ključ. Dodatno lahko tudi opazimo, da strežnik ob zaključku protokola ne more izračunati sejnega ključa. Kljub temu pa mu moramo v protokolu brezpogojno zaupati, saj pozna osebna gesla vseh uporabnikov in se lahko zato lažno predstavlja v njihovem imenu.

Varnost tristranskih protokolov za dogovor o ključu na osnovi gesel je odvisna od zahtevnosti osebnega gesla. Ker si uporabniki radi izbirajo šibka gesla, je prišla potreba po odpravi le-teh. V literaturi so se zato pojavili novi protokoli brez gesel, ki za overjanje uporabnikov preko strežnika uporabljajo digitalne podpise ali simetrične šifre, kot sta npr. DES in AES.



(a)
$$\begin{aligned} \mathcal{A}: &\quad a \xleftarrow{\$} \mathbb{Z}_p^*, \quad A = g^a \bmod q \\ \mathcal{A} \to \mathcal{B}: &\quad \text{ID}_{\mathcal{A}},\ X = E_{\text{geslo}_{\mathcal{A}}}(A \oplus \text{ID}_{\mathcal{B}}) \end{aligned} \tag{1}$$

(b)
$$\begin{aligned} \mathcal{B}: &\quad b \xleftarrow{\$} \mathbb{Z}_p^*, \quad B = g^b \bmod q \\ \mathcal{B} \to \mathcal{S}: &\quad \text{ID}_{\mathcal{A}},\ \text{ID}_{\mathcal{B}},\ X,\ Y = E_{\text{geslo}_{\mathcal{B}}}(B \oplus \text{ID}_{\mathcal{A}}) \end{aligned} \tag{2}$$

(c)
$$\begin{aligned} \mathcal{S}: &\quad s \xleftarrow{\$} \mathbb{Z}_p^* \\ \mathcal{S}: &\quad A = D_{\text{geslo}_{\mathcal{A}}}(X) \oplus \text{ID}_{\mathcal{B}}, \quad B = D_{\text{geslo}_{\mathcal{B}}}(Y) \oplus \text{ID}_{\mathcal{A}} \\ \mathcal{S} \to \mathcal{B}: &\quad S_{\mathcal{A}} = B^s \bmod q,\ S_{\mathcal{B}} = A^s \bmod q \end{aligned} \tag{3}$$

(d)
$$\begin{aligned} \mathcal{B}: &\quad \sigma = S_{\mathcal{B}}^b \bmod q, \quad K = \text{kdf}(\sigma) \\ \mathcal{B} \to \mathcal{A}: &\quad S_{\mathcal{A}},\ C_{\mathcal{B}} = E_K(X) \end{aligned} \tag{4}$$

(e)
$$\begin{aligned} \mathcal{A}: &\quad \sigma = S_{\mathcal{A}}^a \bmod q, \quad K = \text{kdf}(\sigma), \quad \text{preveri}(D_K(C_{\mathcal{B}})) \\ \mathcal{A} \to \mathcal{B}: &\quad C_{\mathcal{A}} = E_K(C_{\mathcal{B}}) \end{aligned} \tag{5}$$

(f)
$$\mathcal{B}: \quad \text{preveri}(D_K(C_{\mathcal{A}}))$$

Slika 3.7: Tristranska šifrirana izmenjava ključa (Diffie-Hellman)

Poglavje 4

Sheme za digitalni podpis

Naslednje pomembno orodje kriptografije javnih ključev so sheme za digitalni podpis, katerim se bomo posvetili v tem poglavju. Najprej bomo predstavili osnovno idejo digitalnega podpisa in možnosti njegove uporabe. Nato bomo definirali splošni model shem za digitalni podpis in ga razširili do modela za sheme na osnovi identitete ter sheme z možnostjo združevanja. Navedli bomo tudi tri razrede poneverb digitalnih podpisov, s katerimi lahko definiramo različne nivoje varnosti shem, in pomembnejše napade na njih. Za najvišji nivo bomo podali varnostni model, katerega bomo v šestem poglavju uporabili za dokazovanje varnosti nove sheme. Poglavje bomo zaključili z opisom različnih vrst shem za digitalni podpis in njihovih glavnih predstavnikov, katere bomo pogosto omenjali ter uporabljali v nadaljevanju.

Večina snovi, obravnavane v tem poglavju, je povzeta po najbolj branih kriptografskih knjigah avtorjev Menezes, Vanstone in Oorschot [105], Stinson [143] ter Katz in Lindell [82] ter po člankih, v katerih so bile predstavljene nove ideje za sestavo shem za digitalni podpis.

4.1 O digitalnih podpisih

Digitalni podpis je nadomestilo za lastnoročni podpis pri elektronski izmenjavi in digitalnemu hranjenju podatkov. Sestavljajo ga podatki, odvisni od vsebine sporočila, ki ga podpisnik podpisuje, in od njegovega zasebnega ključa. S tem dosežemo, da že ob majhni spremembi vsebine podpis postane neve-

ljaven. Podpise se da preveriti, kar pomeni, da lahko z veliko verjetnostjo ugotovimo, ali je neka oseba v resnici podpisala sporočilo. In ker za to ne potrebujemo zasebnega ključa podpisnika, pač pa le njegov javni ključ, lahko preverjanje opravi kdorkoli.

V računalniški varnosti se digitalni podpisi uporabljajo za identifikacijo oseb, za avtorizacijo oz. nadzor nad dostopom, za overjanje sporočil in programov, za ugotavljanje pristnosti in celovitosti podatkov, za preprečevanje tajenja itd. Pomembno vlogo igrajo tudi v kriptografiji javnih ključev, saj jih certifikatne agencije uporabljajo pri izdajanju digitalnih potrdilih za povezovanje identitet uporabnikov in njihovih javnih ključev. Z digitalnim podpisom tako dosežejo, da lahko vsakdo preveri avtentičnost javnega ključa lastnika potrdila, brez njihovega vmešavanja.

Definicija 4.1. Shemo za digitalni podpis, ki pri preverjanju podpisa potrebuje tudi izvirno sporočilo, imenujemo *shema za digitalni podpis z dodatkom*. V nasprotnem primeru ji pravimo *shema za digitalni podpis z obnovo sporočila*.

V praksi se najpogosteje uporabljajo sheme z dodatkom, saj običajno izkoriščajo lastnosti zgoščevalnih funkcij in so zato bolj odporne na obstoječe poneverbe, ki jih bomo spoznali v nadaljevanju. Med sheme za digitalni podpis z dodatkom uvrščamo ElGamalovo [50], Schnorrovo [132] in DSA shemo [52], medtem ko "šolska" verzija podpisa RSA [129] sodi med sheme z obnovo sporočila. V tej tezi se bomo ukvarjali zgolj z shemami za digitalni podpis z dodatkom, katere najbolj preprosto predstavimo z naslednjo definicijo.

Definicija 4.2. *Shema za digitalni podpis* je trojica algoritmov (`KeyGen`, `Sign`, `Verify`), ki zadoščajo naslednjim pogojem.

1. `KeyGen` je verjetnostni algoritem za pripravo ključev, ki sprejme varnostni parameter 1^κ in vrne par zasebni/javni ključ $(\mathrm{sk}, \mathrm{pk})$.

2. `Sign` je deterministični ali verjetnostni algoritem za podpisovanje, ki sprejme sporočilo m in zasebni ključ sk, ter vrne digitalni podpis σ sporočila m.

3. `Verify` je deterministični algoritem za preverjanje podpisov, ki sprejme sporočilo m, digitalni podpis σ in javni ključ pk ter vrne `true`, če je podpis veljaven, sicer pa `false`.

Pri tem zahtevamo, da je shema *dosledna*, tj. da za vsako sporočilo m in za vsak par ključev $(\mathrm{sk}, \mathrm{pk})$, ki ga vrne algoritem $\texttt{KeyGen}(1^\kappa)$, velja

$$\texttt{Verify}(m, \texttt{Sign}(m, \mathrm{sk}), \mathrm{pk}) = \texttt{true}.$$

Podobno kot pri protokolih za dogovor o ključu lahko tudi pri shemah za digitalni podpis javni ključ osebe zamenjamo z njeno identifikacijsko informacijo. S tem se izognemo uporabi certifikatnih agencij in težav z upravljanjem digitalnih potrdil, hkrati pa moramo v sistem uvesti zaupanja vreden generator zasebnih ključev.

Definicija 4.3. *Shema za digitalni podpis na osnovi identitete* je četverica algoritmov (`Setup`, `KeyGen`, `Sign`, `Verify`), ki zadoščajo naslednjim pogojem.

1. `Setup` je verjetnostni algoritem, ki sprejme varnostni parameter 1^κ in vrne sistemske parametre ter par glavni tajni/javni ključ $(\mathrm{msk}, \mathrm{mpk})$. Namenjen je generatorju zasebnih ključev za pripravo sistema. Pri tem generator glavni tajni ključ varno shrani, medtem ko lahko parametre in javni ključ objavi, ali po potrebi pošlje le tistemu, ki jih bo potreboval za digitalno podpisovanje in preverjanje podpisov.

2. `Extract` je deterministični ali verjetnostni algoritem, ki sprejme identiteto ID in glavni tajni ključ msk, ter vrne zasebni ključ sk osebe s to identiteto. Požene ga lahko le generator zasebnih ključev, saj edini pozna glavni tajni ključ.

3. `Sign` je deterministični ali verjetnostni algoritem za podpisovanje, ki sprejme sporočilo m, zasebni ključ sk in identiteto ID, ter vrne digitalni podpis σ sporočila m.

4. `Verify` je deterministični algoritem za preverjanje podpisov, ki sprejme sporočilo m, digitalni podpis σ in identiteto ID ter vrne `true`, če je podpis veljaven, sicer pa `false`.

Pri tem zahtevamo, da je shema *dosledna*, tj. da za vsako sporočilo m in identiteto ID ter za vsak zasebni ključ sk, ki ga vrne algoritem $\texttt{Extract}(\mathrm{ID}, \mathrm{msk})$, velja

$$\texttt{Verify}(m, \texttt{Sign}(m, \mathrm{sk}, \mathrm{ID}), \mathrm{ID}) = \texttt{true}.$$

Nekatere sheme za digitalni podpis omogočajo, da lahko več digitalnih podpisov združimo v en kratek podpis. S tem dosežemo, da s preverjanjem enega združenega podpisa ugotovimo, ali so vsi digitalni podpisi veljavni. Sheme z možnostjo združevanja bomo bolj podrobno spoznali v nadaljevanju, zaenkrat podajmo le njihovo formalno definicijo.

Definicija 4.4. *Shema za digitalni podpis z možnostjo združevanja* dodatno vsebuje algoritma `Aggregate` in `VerifyAgg`, ki zadoščata naslednjim pogojem.

1. `Aggregate` je deterministični ali verjetnostni algoritem, ki sprejme digitalne podpise $(\sigma_i)_{i=1}^n$ in jih združi v en sam podpis σ_{agg}.

2. `VerifyAgg` je deterministični algoritem, ki sprejme sporočila $(m_i)_{i=1}^n$, združen podpis σ_{agg} in javne ključe podpisnikov $(\mathrm{pk}_i)_{i=1}^n$ ter vrne `true`, če je združen podpis veljaven, sicer pa `false`.

Pri tem zahtevamo, da je shema *dosledna*, tj. da za vsak nabor veljavnih digitalnih podpisov $(\sigma_i)_{i=1}^n$ sporočil $(m_i)_{i=1}^n$, ki so jih ustvarile osebe z javnimi ključi $(\mathrm{pk}_i)_{i=1}^n$, velja

$$\texttt{VerifyAgg}(m_1, \ldots, m_n, \texttt{Aggregate}(\sigma_1, \ldots, \sigma_n), \mathrm{pk}_1, \ldots, \mathrm{pk}_n) = \texttt{true}.$$

4.2 Napadi in varnostne zahteve

V tem razdelku bomo vpeljali nekaj zahtev, ki naj bi jih izpolnjevale varne sheme za digitalni podpis. Najprej bomo navedli tri cilje napadalca, s katerimi lahko definiramo različne nivoje varnosti. Nato bomo opisali najbolj osnovne oblike napadov, ki določajo, kakšne informacije ima napadalec na voljo za dosego svojega cilja. Na koncu bomo predstavili še splošni varnostni model, ki ga bomo v šestem poglavju uporabili za dokaz varnosti naše nove sheme za digitalni podpis.

4.2.1 Poneverbe digitalnih podpisov

Glavni cilj napadalca je običajno poneverjanje digitalnih podpisov. To pomeni, da napadalec v imenu neke druge osebe ustvari legitimen podpis nekega sporočila. Takšnemu ponarejenemu podpisu pravimo *poneverba* in jih razvrstimo v tri razrede.

- *Univerzalna poneverba.* Napadalec lahko ustvari veljavni digitalni podpis kateregakoli sporočila. Torej lahko digitalno podpiše sporočila, ki si jih izbere sam, mu jih posredujejo ostale osebe ali so izbrana naključno. Za ustvarjanje takšnih poneverb mora napadalec biti sposoben izračunati zasebni ključ podpisnika ali poiskati svoj algoritem za ustvarjanje veljavnih podpisov. Univerzalni poneverbi pravimo tudi *popolno razbitje* sheme.

- *Izbrana poneverba.* Napadalec lahko ustvari veljaven podpis sporočila oz. vrste sporočil, ki jo sam izbere pred začetkom napada. Takšna sporočila imajo običajno posebne (matematične) lastnosti povezane s strukturo sheme za digitalni podpis. Pri izbrani poneverbi pravi podpisnik ni neposredno vključen pri ustvarjanju digitalnega podpisa.

- *Obstoječa poneverba.* Napadalec lahko ustvari veljavni digitalni podpis vsaj enega sporočila. Pri tem nima oz. ima zelo malo nadzora nad izbiro sporočila za podpisovanje. Pri ustvarjanju obstoječe poneverbe je lahko zavajajoče vključen tudi pravi podpisnik.

Vsi trije razredi poneverb so med seboj povezani. Tako lahko napadalec, ki zna ustvariti univerzalno poneverbo ustvari tudi izbrano, in napadalec, ki zna ustvariti izbrano poneverbo, lahko ustvari obstoječo. Od tod sledi, da je obstoječa poneverba najšibkejša in so zato najbolj varne sheme za digitalni podpis nanjo odporne.

4.2.2 Napadi na digitalne podpise

Napadalec lahko na sheme za digitalni podpis izvede različne napade, odvisno od sredstev, ki jih ima na voljo. V nadaljevanju bomo razlikovali med naslednjimi vrstami napadov:

- *Napad s ključem.* Pri teh napadih napadalec pozna samo javni ključ podpisnika, katerega podpis želi poneveriti.

- *Napad s sporočili.* V tem primeru ima napadalec dostop do veljavnih digitalnih podpisov, katere lahko uporabi pri ustvarjanju poneverb. Napade s sporočili lahko nadalje razvrstimo v tri razrede, ki si po moči napadalca sledijo v naslednjem vrstnem redu:

  - *Napad z znanim sporočilom.* Napadalec ima na voljo množico sporočil z veljavnimi digitalnimi podpisi. Pri tem izbira sporočil ni bila prepuščena njemu.
  - *Napad z izbranim sporočilom.* Napadalec lahko pred napadom na shemo pridobi digitalne podpise sporočil po lastni izbiri. Ta napad ni prilagodljiv, saj mora napadalec določiti sporočila preden, prejme njihove digitalne podpise.
  - *Prilagodljiv napad z izbranim sporočilom.* Napadalec lahko uporabi podpisnika kot preroka za digitalno podpisovanje in tako pridobi podpise sporočil po lastni izbiri. Ta napad je prilagodljiv, kar pomeni, da lahko napadalec zahteva digitalne podpise sporočil, ki jih je sestavil na podlagi prej izbranih sporočil in njihovih podpisov.

- *Napad z identitetami.* Pri shemah za digitalni podpis na osnovi identitete si lahko napadalec na različne načine izbere identiteto osebe, katere podpis želi poneveriti. Napade delimo v dva razreda.

  - *Napad z izbrano identiteto.* Napadalec mora identiteto osebe, katere podpis želi poneveriti, določiti pred začetkom napada. Pri tem napadu ima napadalec dostop do preroka, s katerim lahko razkrije zasebne ključe vseh oseb, razen tiste, katero želi napasti.
  - *Prilagodljiv napad z izbrano identiteto.* Tudi pri tem napadu ima napadalec dostop do preroka za razkrivanje zasebnih ključev. Napad je prilagodljiv, kar pomeni, da lahko napadalec izbira osebe, katerih ključ želi razkriti, na podlagi prej pridobljenih informacij. Ko napadalec ne potrebuje več preroka, si sam izbere identiteto

osebe, katere zasebnega ključa še ni razkril, in poskuša poneveriti njen podpis.

4.2.3 Varnosti modeli

Shema za digitalni podpis zagotavlja najvišji nivo varnosti, če je odporna na obstoječe poneverbe pri prilagodljivem napadu z izbranim sporočilom. Varnostni model za takšno shemo je definiran kot igra med izzivalcem Iztokom in napadalcem Oskarjem, ki je sestavljena iz treh delov.

1. *Priprava.* Iztok si izbere varnostni parameter κ in z algoritmom `Setup` pripravi sistemske parametre ter par zasebni/javni ključ $(\mathrm{sk}, \mathrm{pk})$. Zasebni ključ varno shrani, medtem ko javnega in parametre razkrije Oskarju.

2. *Usposabljanje.* Oskar komunicira z Iztokom preko naslednjega preroka.

   (a) *Prerok za podpisovanje.* Ko prerok prejme poizvedbo s sporočilom m, vrne veljaven digitalni podpis σ.

   Iztok lahko simulira preroka, saj pozna zasebni ključ in lahko zato izračuna digitalne podpise vseh sporočil.

3. *Poneverjanje.* Ko Oskar zaključi z usposabljanjem, vrne digitalni podpis σ sporočila m, ki ga ni prejel od preroka za podpisovanje kot odgovor na poizvedbo s sporočilom m.

Napadalec Oskar zmaga v varnostni igri, če je σ veljaven digitalni podpis.

Definicija 4.5. Shema za digitalni podpis je varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom, če noben verjetnostni polinomski algoritem nima nezanemarljive prednosti za zmago v zgoraj definirani igri.

Podobno lahko definiramo tudi varnostni model sheme za digitalni podpis na osnovi identitete, le da moramo sedaj upoštevati še prilagodljiv napad z izbrano identiteto.

1. *Priprava.* Iztok si izbere varnostni parameter κ in z algoritmom `Setup` pripravi sistemske parametre ter par glavni javni/tajni ključ (mpk, msk). Tajni ključ varno shrani, medtem ko javnega in parametre razkrije Oskarju.

2. *Usposabljanje.* Oskar komunicira z Iztokom preko naslednjih dveh prerokov.

   (a) *Prerok za razkrivanje.* Ko prerok prejme poizvedbo z identiteto ID, razkrije zasebni ključ sk osebe s to identiteto.

   (b) *Prerok za podpisovanje.* Ko prerok prejme poizvedbo s sporočilom m in identiteto ID, vrne veljaven digitalni podpis σ osebe s to identiteto.

   Iztok lahko simulira oba preroka, saj pozna glavni tajni ključ in lahko zato izračuna zasebni ključ vsake osebe.

3. *Poneverjanje.* Ko Oskar zaključi z usposabljanjem, vrne digitalni podpis σ sporočila m, ki naj bi ga ustvarila oseba z identiteto ID. Pri tem mora veljati, da Oskar njenega zasebnega ključa ni razkril s prerokom za razkrivanje, in da podpisa σ ni prejel od preroka za podpisovanje, ko mu je poslal poizvedbo s sporočilom m in identiteto ID.

Napadalec Oskar zmaga v varnostni igri, če je σ veljaven digitalni podpis.

Definicija 4.6. Shema za digitalni podpis na osnovi identitete je varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom in identiteto, če noben verjetnostni polinomski algoritem nima nezanemarljive prednosti za zmago v zgoraj definirani igri.

Definicijo varnostnega modela sheme na osnovi identitete lahko posplošimo tudi na sheme z možnostjo združevanja, le da v tem primeru Oskar na koncu igre vrne združen digitalni podpis σ_{agg}. Tega je sestavil z združitvijo podpisov $(\sigma_i)_{i=1}^n$ sporočil $(m_i)_{i=1}^n$, ki naj bi jih sestavile osebe z identitetami $(\text{ID}_i)_{i=1}^n$. Pri tem mora za vsaj eno identiteto ID_k, $k \in \{1, \ldots, n\}$, veljati, da Oskar preko preroka za razkrivanje ni razkril zasebnega ključa njenega lastnika, in da podpisa σ_k ni dobil od preroka za podpisovanje kot odgovor na poizvedbo s sporočilom m_k in identiteto ID_k.

4.3 Delitev shem za digitalni podpis

Sheme za digitalni podpis lahko glede na njihove lastnosti klasificiramo v različne razrede. V nadaljevanju bomo na kratko predstavili verjetnostne in deterministične sheme, sheme na osnovi identitete in sheme z možnostjo združevanja. Opisali bomo tudi njihove glavne predstavnike, ki jih bomo omenjali oz. uporabljali v prihodnjih poglavjih.

4.3.1 Verjetnostne sheme

Med verjetnostne sheme za digitalni podpis uvrščamo sheme, ki pri ustvarjanju podpisa uporabljajo določeno stopnjo naključnosti. Ker so podpisi odvisni od naključno izbranih vrednosti, ima vsako sporočilo več veljavnih podpisov. S tem se hkrati poveča tudi odpornost na napade, saj je možno varnost shem utemeljiti s tesnejšo prevedbo na težke računske probleme, kar je dokazal Coron v [45]. Verjetnostne sheme zelo pogosto uporabljamo v praksi, med njimi pa najbolj izstopata shemi DSA in njena učinkovitejša različica ECDSA [52].

Enoličnost in tajnost naključno izbrane vrednosti sta pri verjetnostnih shemah ključnega pomena. Če se pri podpisovanju sporočila dvakrat uporabi ista vrednost ali pride do razkritja le-te, potem lahko običajno napadalec to izkoristi za izračun zasebnega ključa uporabnika [124]. Včasih je celo dovolj, da zna napadalec deloma napovedati naslednjo vrednost, ali da se pri vsakem podpisovanju razkrije njen majhen del. Zato je zelo pomembno, da takšne sheme uporabljajo varne generatorje naključnih števil. Kot zanimivost lahko še omenimo, da je bil zaradi omenjene ranljivosti leta 2010 razkrit zasebni ključ podjetja Sony, s katerim se je podpisovalo programe za igralno konzolo PlayStation 3. Vzrok za napad se je skrival v napačni implementaciji ECDSA sheme za digitalni podpis, saj naključna vrednost v resnici ni bila izbrana naključno.

ElGamalova shema

ElGamalova shema za digitalni podpis je verjetnostna shema, katere varnost naj bi temeljila na problemu diskretnega logaritma. Predstavljena je bila

leta 1985, skupaj z ElGamalovim kriptosistemom [50]. Trenutno obstaja več njenih različic, med katerimi je najbolj znana shema DSA (Digital Signature Algorithm) [52], ki jo je razvila ameriška agencija za nacionalno varnost NSA in sprejel inštitut za standardizacijo in tehnologijo NIST.

V nadaljevanju bomo opisali osnovno ElGamalovo shemo, ki pa se bolj redko uporablja v praksi. Razlog za to se skriva v dolžini digitalnega podpisa, saj je ta občutno večji kot pri DSA in Schnorrovi shemi.

Shema 1 ElGamalov digitalni podpis

1. `KeyGen` na podlagi varnostnega parametra 1^κ naključno izbere praštevilo p, generator g grupe $\mathbb{Z}_p^*$ in zgoščevalno funkcijo $H : \{0,1\}^* \to \mathbb{Z}_p$, ki je odporna na trke. Nato določi število $a \xleftarrow{\$} \mathbb{Z}_{p-1}$, izračuna vrednost $A = g^a \bmod p$ in vrne zasebni ključ (p, g, a) ter javni ključ (p, g, A).

2. `Sign` sprejme sporočilo m in zasebni ključ (p, g, a). Nato izbere tako število $k \xleftarrow{\$} \mathbb{Z}_{p-1}$, da velja $D(k, p-1) = 1$, in izračuna $r = g^k \bmod p$ ter $s = k^{-1}(H(m) - ar) \bmod (p-1)$. Na koncu vrne digitalni podpis $\sigma = (r, s)$.

3. `Verify` kot vhod sprejme sporočilo m, digitalni podpis $\sigma = (r, s)$ in javni ključ (p, g, A). Nato preveri, če je $1 \leq r \leq p-1$ in sprejme podpis kot veljaven, če in samo če velja

$$A^r r^s \equiv g^{H(m)} \pmod{p}.$$

Z naslednjim izračunom se ni težko prepričati, da je pripravljen digitalni podpis $\sigma = (r, s)$ veljaven, in da je shema dosledna

$$A^r r^s \equiv g^{ar} g^{k(k^{-1}(H(m)-ar))} \equiv g^{ar+H(m)-ar} \equiv g^{H(m)} \pmod{p}.$$

Pri ElGamalovi shemi za digitalni podpis je potrebno paziti, da se pri podpisovanju sporočila vrednost k izbere naključno z enakomerno porazdelitvijo, in da ta deloma ali v celoti ni razkrita napadalcu. V nasprotnem primeru lahko to izkoristi napadalec za hitrejši izračun zasebnega ključa (p, g, a). Prav tako je potrebno paziti, da uporabnik iste vrednosti k ne uporabi pri podpisovanju dveh sporočil, saj je potem možno iz ustreznih digitalnih podpisov izračunati celotni zasebni ključ [105, §11.5.2].

Schnorrova shema

Leta 1989 je Schnorr predstavil boljšo različico ElGamalove sheme, s katero je občutno zmanjšal velikost digitalnega podpisa [132]. Pri tej se računanje izvaja v multiplikativni podgrupi velikosti p grupe $\mathbb{Z}_q^*$, kjer sta q in p praštevili, običajno velikosti 1024 ter 160 bitov. S takšno izbiro parametrov dosežemo, da so digitalni podpisi dolgi zgolj 320 bitov. Za razliko od običajnih shem, pri katerih najprej izračunamo zgostitev sporočila in šele nato ustvarimo digitalni podpis, Schnorrova shema zgostitev izračuna v fazi podpisovanja in je zato del sheme. S tem se zagotovi večja varnost in zmanjša dolžina digitalnega podpisa.

Shema 2 Schnorrov digitalni podpis

---

1. `KeyGen` na podlagi varnostnega parametra 1^κ izbere taki praštevili p in q, da velja $p \mid q$, ter generator $g \in \mathbb{Z}_q^*$ reda p. Nato določi število $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna vrednost $A = g^a \bmod q$ in vrne zasebni ključ (p, q, g, a) ter javni ključ (p, q, g, A).

2. `Sign` kot vhod sprejme sporočilo m in zasebni ključ (p, q, g, a). Nato izbere število $k \xleftarrow{\$} \mathbb{Z}_p^*$ in izračuna $r = g^k \bmod q$, $e = H(m, r)$ ter $s = (ae + k) \bmod p$. Na koncu vrne digitalni podpis $\sigma = (s, e)$.

3. `Verify` kot vhod sprejme sporočilo m, digitalni podpis $\sigma = (s, e)$ in javni ključ (p, q, g, A). Nato izračuna $r' = g^s A^{-e} \bmod q$ in sprejme podpis kot veljaven, če in samo če velja

$$e = H(m, r').$$

Doslednost zgornje sheme lahko preverimo z izračunom

$$r' \equiv g^s A^{-e} \equiv g^{ae+k} g^{-ae} \equiv g^k \pmod{q},$$

ki nam razkrije $r' = r$, tj. da pri preverjanju podpisa izračunamo isti r, kot je bil izračunan v fazi podpisovanja, in zato preverjanje enačbe $e = H(m, r')$ uspe.

4.3.2 Deterministične sheme

Pri kriptosistemih z javnimi ključi je že dolgo časa znano, da šifriranje ne sme biti deterministično. Namreč, če bi bilo, potem bi lahko napadalec z javnim ključem zašifriral vse možne kandidate za sporočilo in preveril, pri katerem se tajnopis ujema. Razlog za to se skriva v dejstvu, da je pri šifriranju potrebno hkrati ohraniti tajnost sporočila in zasebnega ključa. Takšne omejitve pri shemah za digitalni podpis ni, saj je sporočilo vedno javno objavljeno. Zato je možno sestaviti tudi varne deterministične sheme. S tem se znebimo potrebe po generatorjih naključnih števil, kar je posebej priročno v računsko omejenih in potencialno ogroženih okoljih.

FDH shema

FDH (Full Domain Hash) shema za digitalni podpis je deterministična shema, ki za podpisovanje sporočil uporablja kriptosistem RSA in zgoščevalno funkcijo, odporno na trke. Leta 1996 sta jo predstavila Bellare in Rogaway ter dokazala, da je varna pred obstoječimi poneverbami pri napadu z izbranim sporočilom v modelu naključnega preroka [17]. Shema je zasnovana po načelu *zgosti-in-odšifriraj*, pri katerem podpis pripravimo tako, da z zasebnim ključem odšifriramo zgostitev sporočila. Preverjanje podpisa je preprosto, saj je potrebno digitalni podpis le zašifrirati s podpisnikovim javnim ključem in dobljeni tajnopis primerjati z zgostitvijo. Avtorji sheme so v modelu naključnega preroka tudi dokazali, da je shema varna pred obstoječimi poneverbami pri napadu z izbranim sporočilom.

Shema 3 FDH digitalni podpis

---

1. `KeyGen` na podlagi varnostnega parametra 1^κ izbere praštevili p in q, ter števili e in d, za kateri velja $ed \equiv 1 \pmod{\varphi(n)}$. Pri tem varnostni parameter zagotavlja, da je faktorizacija števila $n = pq$ težak problem. Kot izhod vrne zasebni ključ (d, n) in javni ključ (e, n).

2. `Sign` sprejme sporočilo m in zasebni ključ (d, n) ter vrne digitalni podpis $\sigma = H(m)^d \bmod n$.

3. `Verify` sprejme sporočilo m, digitalni podpis $\sigma = H(m)^d \bmod n$ in javni ključ (e, n) ter podpis sprejme kot veljaven, če in samo če velja

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

Algoritem za podpisovanje vrne veljavne podpise, kar dokazuje izračun

$$\sigma^e \equiv H(m)^{ed} \equiv H(m) \pmod{n}.$$

To pomeni, da je FDH shema za digitalni podpis dosledna.

BLS shema

Leta 2001 so Boneh, Lynn in Shacham predstavili BLS shemo za digitalni podpis z zelo kratkim podpisom [26]. Za 80 bitno varnost njegova dolžina znaša približno 160 bitov, kar je dvakrat manj od dolžine digitalnega podpisa ECDSA in Schnorrove sheme. Hkrati so v modelu naključnega preroka dokazali, da je shema varna pred napadom z izbranim sporočilom, če predpostavimo, da je izračunljiv Diffie-Hellmanov problem na določenih eliptičnih krivuljah težak. V splošnem je shemo možno sestaviti v vseh vmesnih grupah, tj. v grupah, kjer je odločitveni Diffie-Hellmanov problem lahek in izračunljiv težak.

Shema 4 BLS digitalni podpis

---

1. `KeyGen` na podlagi varnostnega parametra 1^κ izbere praštevilo p, bilinearno parjenje $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$, generator P reda p v grupi $\mathbb{G}_1$ in zgoščevalno funkcijo $H : \{0,1\}^* \to \mathbb{G}_1$. Nato vrne zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$ in javni ključ $A = aP$.

2. `Sign` sprejme sporočilo m in zasebni ključ a ter vrne digitalni podpis $\sigma = aH(m)$.

3. `Verify` sprejme sporočilo m, digitalni podpis $\sigma = aH(m)$ in javni ključ A ter sprejme podpis kot veljaven, natanko tedaj, ko velja

$$\hat{e}(\sigma, P) = \hat{e}(H(m), A).$$

BLS shema je dosledna, saj je zaradi lastnosti bilinearnih parjenj digitalni podpis σ vedno veljaven, kar dokazuje izračun

$$\hat{e}(\sigma, P) = \hat{e}(aH(m), P) = \hat{e}(H(m), aP) = \hat{e}(H(m), A).$$

4.3.3 Sheme na osnovi identitete

Poseben primer shem za digitalni podpis so sheme na osnovi identitete, pri katerih se za javni ključ uporabnika vzame kar njegovo identifikacijsko informacijo, kot je npr. ime, naslov, davčna številka ali e-poštni naslov. Kot smo že omenili, takšne sheme odpravijo potrebo po izmenjavi zasebnih in javnih ključev, saj so identitete znane vsem uporabnikom in implicitno overjene. Podobno kot protokoli za dogovor o ključu na osnovi identitete morajo tudi sheme v sistem vključiti zaupanja vreden generator ključev, ki uporabnikom preko varnega kanala izdaja njihove zasebne ključe.

Shamirjeva shema

Prvo shemo za digitalni podpis na osnovi identitete je leta 1985, skupaj z idejo kriptografije na osnovi identitete, predstavil Shamir [136]. Njena varnost naj bi temeljila na problemu RSA in na sorodnem problemu razcepa (sestavljenih) števil. Shema uporablja enosmerno funkcijo f in generator zasebnih ključev ter je definirana z naslednjimi štirimi algoritmi:

Shema 5 Shamirjev digitalni podpis na osnovi identitete

1. `Setup` na podlagi varnostnega parametra 1^κ izbere praštevili p in q, izračuna $n = pq$ ter določi e in d, da velja $ed \equiv 1 \pmod{\varphi(n)}$. Kot izhod vrne javni ključ (n, e) in glavni tajni ključ (n, e, d).

2. `KeyGen` sprejme identiteto ID in glavni tajni ključ (N, e, d) ter vrne zasebni ključ $a = H(\mathrm{ID})^d \bmod n$.

3. `Sign` kot vhod sprejme sporočilo m, zasebni ključ a in identiteto ID. Nato izbere naključno število $t \xleftarrow{\$} \mathbb{Z}_n^*$, izračuna vrednosti $T = t^e \bmod n$ in $s = a\, t^{f(T,m)} \bmod p$ ter vrne digitalni podpis $\sigma = (T, s)$.

4. `Verify` sprejme sporočilo m, digitalni podpis $\sigma = (T, s)$ in identiteto ID ter sprejme podpis kot veljaven, če in samo če velja

$$s^e \equiv H(\mathrm{ID})\, T^{f(T,m)} \pmod{n}.$$

Tudi Shamirjeva shema za digitalni podpis je dosledna, saj preverjanje podpisa $\sigma = (T, s)$ vedno uspe. V to se lahko prepričamo z izračunom

$$s^e \equiv a^e t^{ef(T,m)} \equiv H(\mathrm{ID})^{ed}\, T^{f(T,m)} \equiv H(\mathrm{ID})\, T^{f(T,m)} \pmod{n}.$$

4.3.4 Sheme z možnostjo (delnega) združevanja

Pri hranjenju digitalno podpisanih sporočil je poleg sporočila potrebno hraniti tudi vse pripadajoče podpise. V vsakdanjem življenju najdemo mnogo varnostnih sistemov, v katerih lahko uporabniki digitalno podpisujejo različna sporočila. Takšne sisteme najdemo v bančnih sistemih pri izvajanju transakcij, v odvetniških pisarnah za arhiviranje in posredovanje pravnih dokumentov, v računalniških sistemih za zagotavljanje pristnosti programov, pri podpisovanju peticij itd. V vseh teh sistemih se zato pojavijo težave pri pripravi, shranjevanju in pošiljanju velikega števila digitalno podpisanih sporočil ter dokumentov.

Za primer vzemimo infrastrukturo javnih ključev, v kateri veriga zaupanja vsebuje n certifikatnih agencij. V takšni infrastrukturi vsako digitalno potrdilo vsebuje n digitalnih podpisov n različnih agencij, ki uporabnikom zagotavljajo pristnost potrdila. Veliko podpisov se pojavi tudi v varnem BGB protokolu, v katerem vsak usmerjevalnik prejme seznam n digitalnih podpisov, ki potrjujejo opravljeno pot dolžine n podatkovnega paketa v omrežju. Usmerjevalnik nato doda še svoj podpis in podatke pošlje naprej po omrežju. V obeh omenjenih primerih se število digitalnih podpisov povečuje linearno z dolžino verige oz. poti, zato je potrebno pri preverjanju pristnosti preveriti mnogo podpisov, kar seveda vzame veliko časa. Kadar so sporočila majhna, pa nam digitalni podpisi zavzamejo še dodatni podatkovni prostor in tako upočasnijo hitrost prenosa sporočil. Zato bi v takšnih primerih bilo idealno, če bi lahko preverjanje vseh teh digitalnih podpisov združili v eno samo.

Leta 2003 so Boneh, Gentry, Lynn in Shacham predstavili idejo sheme za digitalni podpis z *možnostjo združevanja* [25], ki omogoča, da n digitalnih podpisov n različnih sporočil s strani n različnih oseb združimo v en kratek podpis. Tako dosežemo, da s preverjanjem enega združenega podpisa ugotovimo, ali je n oseb resnično podpisalo prvotnih n sporočil.

BGLS shema

BGLS shema za digitalni podpis z možnostjo združevanja [25] temelji na BLS shemi in vsebuje dva dodatna algoritma `Aggregate` in `VerifyAgg`. Shema je, podobno kot BLS shema, dokazano varna pred obstoječimi poneverbami v modelu naključnega preroka.

Shema 6 BGLS digitalni podpis z možnostjo združevanja

1. `KeyGen` (glej shemo 4).
2. `Sign` (glej shemo 4).
3. `Verify` (glej shemo 4).
4. `Aggregate` sprejme n digitalnih podpisov $(\sigma_i)_{i=1}^n$ in vrne združen podpis
$$\sigma_{\text{agg}} = \sum_{i=1}^{n} \sigma_i.$$
5. `VerifyAgg` sprejme sporočila $(m_i)_{i=1}^n$, združen podpis σ_{agg} in javne ključe podpisnikov $(A_i)_{i=1}^n$ ter podpis sprejme kot veljaven, če in samo če velja
$$\hat{e}(\sigma_{\text{agg}}, P) = \prod_{i=1}^{n} \hat{e}(H(m_i), A_i).$$

Tudi BGLS shema z možnostjo združevanja je dosledna, saj velja

$$\hat{e}(\sigma_{\text{agg}}, P) = \hat{e}(\sum_{i=1}^{n} \sigma_i, P) = \prod_{i=1}^{n} \hat{e}(aH(m_i), P) = \prod_{i=1}^{n} \hat{e}(H(m_i), A_i).$$

Pri idealni shemi z možnostjo združevanja naj bi bila dolžina združenega digitalnega podpisa konstantna, če pri tem zanemarimo velikost sporočil in

javnih ključev podpisnikov. Dolžina podpisa naj ne bi bila odvisna niti od števila podpisnikov niti od števila podpisanih sporočil. Leta 2006 je Herranz opazil [67], da za nobeno shemo za digitalni podpis na osnovi identitete, ki so bile do takrat objavljene v literaturi, ne zgleda, da bi bilo to lastnost možno doseči. Da bi se vsaj približal rešitvi tega problema, je predlagal *delno združevanje*, pri katerem naj bo dolžina združenega podpisa odvisna zgolj od števila podpisnikov in ne od števila podpisanih sporočil. Kmalu za tem sta Gentry in Zulfikar [56] v celoti rešila ta problem in predstavila prvo shemo za digitalni podpis na osnovi identitete z možnostjo združevanja. Njeno varnost sta tudi formalno dokazala v modelu naključnega preroka s prevedbo na vmesni Diffie-Hellmanov problem.

Poglavje 5

Varnostna analiza protokolov in shem

V uvodnem poglavju smo že omenili, kako težko je sestaviti varne kriptografske protokole in sheme. Izkušnje nas učijo, da le redki novi predlogi ostanejo varni tudi po večletni analizi. To je seveda resen problem, ki ga sodobna kriptografija poskuša rešiti z uvedbo čim bolj realističnih varnostnih modelov in z dokazovanjem varnosti znotraj njih [41]. Vendar se tudi tu pojavijo težave, saj lahko hitro pride do napak v dokazih in do pomanjkljivih definicij modelov. Zato je še vedno najboljše zagotovilo za varnost novega protokola oz. sheme varnostna analiza, opravljena s strani čim večjega števila najboljših kriptografov s celega sveta. Le-te pa še zdaleč ni lahko doseči.

To poglavje je namenjeno predstavitvi rezultatov varnostne analize desetih protokolov za overjen dogovor o ključu in ene sheme za digitalni podpis. Za vsak predlog bomo podali še neodkrite varnostne pomanjkljivosti, ki smo jih našli s skrbno analizo in podrobno študijo njegove strukture. Hkrati bomo razkrili tudi konkretne napade, ki jih lahko napadalec uporabi za izkoriščanje le-teh. Predlagani protokoli in shema zato niso varni kljub nasprotnim trditvam njihovih avtorjev. Dva izmed njih sta varnostne lastnosti svojega predloga celo "formalno" dokazala, medtem ko so ostali podali le hevristične razloge za njegovo varnost.

Varnostno analizo bomo najprej izvedli na dvostranskem protokolu za overjen dogovor o ključu avtorja Okamota [118] in na treh tristranskih pro-

		Protokoli za dogovor o ključu									
		Okamoto	Chen	Tan	Lim	Hölbl IDAK2-1	Hölbl IDAK2-2	Hölbl IDAK2-P1	Hölbl IDAK2-P2	Hölbl IDAK3-P1	Hölbl IDAK3-P2
Vrste napadov	pasivni napad							✓		✓	
	aktivni napad	✓	✓	✓	✓	✓	✓		✓		✓
	napad notranje osebe								✓		✓
	napad s ponavljanjem										✓
	razkritje zasebnega ključa							✓			
	razkritje sejnega ključa									✓	
	napad vmesne osebe			✓	✓	✓	✓				
	lažno predstavljanje			✓	✓	✓	✓				✓
	lažno predstavljanje z razkritim ključem	✓	✓								
	deljenje ključa z neznano osebo								✓		

Tabela 5.1: Pregled novih napadov na protokole za dogovor o ključu

tokolih avtorjev Chena [38], Tana [144] ter Lima [96]. Vsi štirje protokoli naj bi imeli želene varnostne lastnosti, prvi izmed njih pa naj bi bil celo dokazano varen v varnostnem modelu eCK. Nadaljevali bomo z analizo šestih protokolov za overjen dogovor o ključu na osnovi identitete, ki jih je v svoji doktorski disertaciji [69] in v treh znanstvenih člankih [70, 71, 72] predstavil Hölbl ter trdil, da ustrezajo mnogim varnostnim kriterijem. Štirje izmed njih, IDAK2-1, IDAK2-2, IDAK2-P1 in IDAK2-P2, so dvostranski protokoli za dogovor o ključu, medtem ko sta dva, IDAK3-P1 in IDAK3-P2, tristranska. Poleg analize bomo pri vsakem protokolu opisali tudi napad, s katerim bomo razkrili njegovo ranljivost. Pregled vseh napadov prikazuje tabela 5.1. Poglavje zaključimo z analizo Selvijeve sheme za digitalni podpis na osnovi identitete z možnostjo združevanja [134], katere varnost so avtorji dokazali v modelu naključnega preroka. Tudi slednja ni varna, saj smo našli postopek

za ustvarjanje univerzalnih poneverb.

Izvirni prispevki tega poglavja so bili objavljeni v dveh znanstvenih člankih z naslovom *Security weaknesses of authenticated key agreement protocols* [112] in *Security weaknesses of a signature scheme and authenticated key agreement protocols* [114].

5.1 Okamotov protokol

Na konferenci ASIACRYPT, ki sodi med tri najbolj priznane mednarodne konference s področja kriptografije, je Okamoto leta 2007 predstavil nov način sestavljanja varnih kriptografskih orodij [118]. V svojem vabljenem predavanju je predlagal tudi dvostranski overjen dogovor o ključu in trdil, da je njegovo varnost možno dokazati v varnostnem modelu eCK (glej §2.4.5). Kmalu za tem je objavil popravljeno verzijo razširjenega povzetka konference, v katerem je spremenil svoj protokol in dokazal njegovo varnost [119]. Razlogov za spremembe ni navedel, prav tako pa varnostna analiza prvotnega protokola ni bila nikoli objavljena v literaturi.

V nadaljevanju bomo pokazali, da je osnovna verzija Okamotovega protokola za overjen dogovor o ključu ranljiva za napad lažnega predstavljanja z razkritim ključem in zato nikakor ne more biti varna v modelu eCK. Hkrati bomo navedli tudi razloge, zakaj podobnega napada ni možno izvesti na popravljeno verzijo protokola.

Predstavitev protokola

Okamotov protokol je dvostranski overjen dogovor o ključu, s katerim se lahko Anita in Bojan preko javnega kanala dogovorita za skupni sejni ključ ter medsebojno overita. Temelji na kriptografiji javnih ključev, zato mora vsak udeleženec imeti v posesti svoj trajni zasebni ključ, pripadajoči javni ključ pa mora biti overjen pri certifikatni agenciji in javno objavljen v digitalnem potrdilu. Za izvedbo protokola zato ni potrebno, da se Anita in Bojan srečata vnaprej, niti da se predhodno dogovorita za skupen tajen ključ. Dovolj je le, da opravita nekaj preprostih izračunov in si izmenjata dve sporočili. Protokol je zato izjemno učinkovit in primerljiv z najučinkovitejšimi protokoli za over-

jen dogovor o ključu, kot je npr. protokol MQV. Hkrati naj bi bil tudi varen, njegova varnost pa naj bi temeljila na odločitvenem Diffie-Hellmanovem problemu, psevdonaključnih funkcijah in zgoščevalnih funkcijah, ki so odporne na ciljne trke.

Protokol 8 Okamotov dogovor o ključu

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bo κ varnosti parameter in $\mathbb{G} = \langle g_1 \rangle = \langle g_2 \rangle$ multiplikativna grupa praštevilskega reda p, tako da velja $|p| = \kappa$ in $g_1 \neq g_2$. Nadalje naj bo Π množica možnih digitalnih potrdil za trajne javne ključe, $\mathcal{H} = \{H_s : \Pi \times \mathbb{G}^2 \to \mathbb{Z}_p \mid s \in \{0,1\}^\kappa\}$ družina zgoščevalnih funkcij odporna na ciljne trke in $\mathcal{F} = \{F_k : \Pi^2 \times \mathbb{G}^4 \to \{0,1\}^\kappa \mid k \in \mathbb{G}\}$, $\hat{\mathcal{F}} = \{\hat{F}_k : \mathbb{Z}_p^4 \to \mathbb{Z}_p \mid k \in \{0,1\}^\kappa\}$ ter $\bar{\mathcal{F}} = \{\bar{F}_k : \{0,1\}^\kappa \to \mathbb{Z}_p \mid k \in \mathbb{Z}_p^4\}$ tri družine psevdonaključnih funkcij.

   (b) Anita izbere svoj trajni zasebni ključ $(a_1, a_2, a_3, a_4) \xleftarrow{\$} \mathbb{Z}_p^4$ in sol $s_\mathcal{A} \xleftarrow{\$} \{0,1\}^\kappa$, ki določa zgoščevalno funkcijo $H_{s_\mathcal{A}} \in \mathcal{H}$. Nato izračuna $A_1 = g_1^{a_1} g_2^{a_2}$ in $A_2 = g_1^{a_3} g_2^{a_4}$ ter pošlje svoj trajni javni ključ $(A_1, A_2, s_\mathcal{A})$ certifikatni agenciji v registracijo in izdajo digitalnega potrdila $\mathrm{Cert}_\mathcal{A}$.

2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}: &\quad X_1, X_2 \quad (1) \\ \mathcal{B} \to \mathcal{A}: &\quad Y_1, Y_2 \quad (2) \end{aligned}$$

3. *Koraki v protokolu.* Anita in Bojan se po uspešno izvedeni pripravi, v kateri si izbereta trajna zasebna ključa (a_1, a_2, a_3, a_4) in (b_1, b_2, b_3, b_4) ter svoja javna ključa $(A_1, A_2, s_\mathcal{A})$ in $(B_1, B_2, s_\mathcal{B})$ objavita v digitalnih potrdilih $\mathrm{Cert}_\mathcal{A}$ ter $\mathrm{Cert}_\mathcal{B}$, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere naključno število $x' \xleftarrow{\$} \{0,1\}^\kappa$, izračuna svoj začasni zasebni ključ $x = \left(\hat{F}_{x'}(a_1, a_2, a_3, a_4) + \bar{F}_{(a_1,a_2,a_3,a_4)}(x')\right) \bmod p$, začasna javna ključa $X_1 = g_1^x$ in $X_2 = g_2^x$ ter ju pošlje v sporočilu (1) Bojanu.

(b) Podobno Bojan izbere naključno število $y' \xleftarrow{\$} \{0,1\}^\kappa$, izračuna svoj začasni ključ $y = \left(\hat{F}_{y'}(b_1,b_2,b_3,b_4) + \bar{F}_{(b_1,b_2,b_3,b_4)}(y')\right) \bmod p$, začasna javna ključa $Y_1 = g_1^y$ in $Y_2 = g_2^y$ ter ju pošlje v sporočilu (2) Aniti.

(c) Po izmenjavi sporočil Anita in Bojan preverita, ali elementi X_1, X_2, Y_1 in Y_2 pripadajo grupi $\mathbb{G}$. Če preverjanje ne uspe, zaključita protokol, sicer izračunata zgostitvi $c = H_{s_\mathcal{A}}(\mathrm{Cert}_\mathcal{A}, Y_1, Y_2)$ in $d = H_{s_\mathcal{B}}(\mathrm{Cert}_\mathcal{B}, X_1, X_2)$ ter skupno skrivnost σ. Slednjo Anita izračuna po enačbi $\sigma_\mathcal{A} = Y_1^{a_1+ca_3+x}\, Y_2^{a_2+ca_4+x}\, B_1^x\, B_2^{dx}$ in Bojan po enačbi $\sigma_\mathcal{B} = X_1^{b_1+db_3+y}\, X_2^{b_2+db_4+y}\, A_1^y\, A_2^{cy}$. Če so vsi izračuni pravilni, potem velja $\sigma = \sigma_\mathcal{A} = \sigma_\mathcal{B}$ in oba udeleženca lahko izračunata sejni ključ $K = F_\sigma(\mathrm{Cert}_\mathcal{A}, \mathrm{Cert}_\mathcal{B}, X_1, X_2, Y_1, Y_2)$.

Ni se težko prepričati, da na koncu protokola Anita in Bojan izračunata isto skupno skrivnost

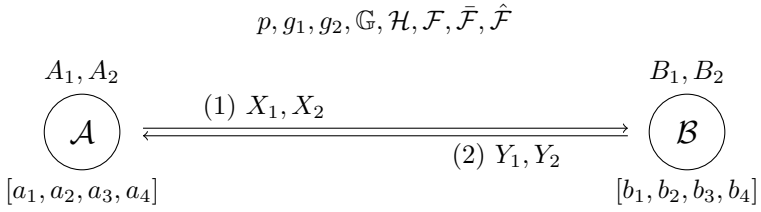
$$\sigma = g_1^{(a_1+ca_3)y+(b_1+db_3)x+xy}\, g_2^{(a_2+ca_4)y+(b_2+db_4)x+xy} \tag{5.1}$$

ter iz nje izpeljeta skupni sejni ključ.

Varnostna analiza

Varnostna pomanjkljivost Okamotovega protokola se nahaja v obeh eksponentih skupne skrivnosti (glej enačbo (5.1)). Tam trajni in začasni zasebni ključi Anite ter Bojana niso dovolj dobro prepleteni, kar lahko s pridom izkoristi napadalec Oskar. Slednji lahko z ustrezno izbiro Bojanovega začasnega javnega ključa doseže, da je skupna skrivnosti odvisna le še od trajnih zasebnih ključev. Zaradi te ranljivost je zato za izračun sejnega ključa dovolj poznati le Anitin zasebni ključ.

Izrek 5.1. *Okamotov protokol za dvostranski overjen dogovor o ključu ni odporen na napad lažnega predstavljanja z razkritim ključem. Napadalec Oskar se lahko Aniti lažno predstavi kot Bojan, če pozna njen trajni zasebni ključ* (a_1, a_2, a_3, a_4).



$$
\begin{aligned}
& \mathcal{A}: && x' \xleftarrow{\$} \{0,1\}^{\kappa} \\
\text{(a)} \quad & \mathcal{A}: && x = \left(\hat{F}_{x'}(a_1, a_2, a_3, a_4) + \bar{F}_{(a_1,a_2,a_3,a_4)}(x')\right) \bmod p \\
& \mathcal{A} \to \mathcal{B}: && X_1 = g_1^x, X_2 = g_2^x \qquad (1) \\
& \mathcal{B}: && y' \xleftarrow{\$} \{0,1\}^{\kappa} \\
\text{(b)} \quad & \mathcal{B}: && y = \left(\hat{F}_{y'}(b_1, b_2, b_3, b_4) + \bar{F}_{(b_1,b_2,b_3,b_4)}(y')\right) \bmod p \\
& \mathcal{B} \to \mathcal{A}: && Y_1 = g_1^y, Y_2 = g_2^y \qquad (2) \\
& \mathcal{A}, \mathcal{B}: && (X_1, X_2, Y_1, Y_2) \stackrel{?}{\in} \mathbb{G}^4 \\
& \mathcal{A}, \mathcal{B}: && c = H_{s_{\mathcal{A}}}(\mathrm{Cert}_{\mathcal{A}}, Y_1, Y_2) \\
& \mathcal{A}, \mathcal{B}: && d = H_{s_{\mathcal{B}}}(\mathrm{Cert}_{\mathcal{B}}, X_1, X_2) \\
\text{(c)} \quad & \mathcal{A}: && \sigma = Y_1^{a_1+ca_3+x}\, Y_2^{a_2+ca_4+x}\, B_1^x\, B_2^{dx} \\
& \mathcal{B}: && \sigma = X_1^{b_1+db_3+y}\, X_2^{b_2+db_4+y}\, A_1^y\, A_2^{cy} \\
& \mathcal{A}, \mathcal{B}: && K = F_\sigma(\mathrm{Cert}_{\mathcal{A}}, \mathrm{Cert}_{\mathcal{B}}, X_1, X_2, Y_1, Y_2)
\end{aligned}
$$

Slika 5.1: Okamotov dogovor o ključu

Dokaz. Najprej bomo dokazali, da lahko napadalec Oskar na desni strani izraza v (5.1) z ustrezno izbiro Bojanovega začasnega javnega ključa (Y_1, Y_2) popolnoma odstrani začasna zasebna ključa x in y iz obeh eksponentov skupne skrivnosti.

Naj bo (X_1, X_2) Anitin začasni javni ključ in $\text{Cert}_{\mathcal{B}}$ veljavno digitalno potrdilo, ki vsebuje Bojanov trajni javni ključ $(B_1, B_2, s_{\mathcal{B}})$. Denimo, da Oskar izračuna zgostitev $d = H_{s_{\mathcal{B}}}(\text{Cert}_{\mathcal{B}}, X_1, X_2)$ in v protokolu Bojanov začasni javni ključ (Y_1, Y_2) zamenja z vrednostmi $Y_1 = B_1^{-1}$ in $Y_2 = B_2^{-d}$. Potem nam Anitin izračun skupne skrivnosti

$$\begin{aligned}
\sigma_{\mathcal{A}} &= Y_1^{a_1+ca_3+x}\, Y_2^{a_2+ca_4+x}\, B_1^x\, B_2^{dx} \\
&= B_1^{-a_1-ca_3-x}\, B_2^{-da_2-cda_4-dx}\, B_1^x\, B_2^{dx} \\
&= B_1^{-a_1-ca_3}\, B_2^{-da_2-cda_4} \\
&= g_1^{-(a_1+ca_3)b_1-(a_2+ca_4)db_3}\, g_2^{-(a_1+ca_3)b_2-(a_2+ca_4)db_4}
\end{aligned}$$

razkrije, da le-ta ni več odvisna od začasnih zasebnih ključev x in y, ki si ju Anita in Bojan izbereta med izvajanjem protokola. To pa pomeni, da je odvisna le še od njunih trajnih zasebnih ključev. Za izračun skupne skrivnosti $\sigma_{\mathcal{A}} = B_1^{-a_1-ca_3}\, B_2^{-da_2-cda_4}$ je v tem primeru dovolj poznati le Anitin zasebni ključ (a_1, a_2, a_3, a_4), medtem ko lahko Bojanov javni ključ $(B_1, B_2, s_{\mathcal{B}})$ preberemo iz njegovega digitalnega potrdila, zgostitvi c in d pa izračunamo iz izmenjanih sporočil ter digitalnih potrdil $\text{Cert}_{\mathcal{A}}$ in $\text{Cert}_{\mathcal{B}}$, ki sta javno objavljeni. Ni pa dovolj poznati le Bojanov zasebni ključ (b_1, b_2, b_3, b_4), saj skupne skrivnosti ni možno izračunati samo iz Anitinega javnega ključa $A_1 = g_1^{a_1} g_2^{a_2}$ in $A_2 = g_1^{a_3} g_2^{a_4}$. Slednjemu namreč ustreza natanko p^2 različnih zasebnih ključev, med katerimi pa ne moremo razločiti tistih p ključev, ki vrnejo isto skupno skrivnost kot Anitin zasebni ključ.

Dokažimo sedaj, da lahko omenjeno ranljivost izkoristi Oskar in pripravi napad lažnega predstavljanja z razkritim ključem, v katerem se Aniti lažno predstavi kot Bojan, če pozna njen trajni zasebni ključ. Naj bo Anita oseba, katere zasebni ključ (a_1, a_2, a_3, a_4) je Oskar razkril. Ko se želi Anita dogovoriti za sejni ključ z Bojanom, prične protokol in v prvem krogu pošlje sporočilo s svojim začasnim javnim ključem (X_1, X_2) Bojanu, kot je to opredeljeno v definiciji protokola. V tem trenutku se v protokol aktivno vključi napadalec Oskar, ki prestreže poslano sporočilo, tako da to nikoli

ne prispe do želenega naslovnika. Nato iz javno dostopne zbirke digitalnih potrdil pridobi veljavno Bojanovo potrdilo $\mathrm{Cert}_{\mathcal{B}}$, iz njega prebere njegov javni ključ $(B_1, B_2, s_{\mathcal{B}})$ in iz prestreženega sporočila izračuna zgostitev $d = H_{s_{\mathcal{B}}}(\mathrm{Cert}_{\mathcal{B}}, X_1, X_2)$ ter vrednosti $Y_1 = B_1^{-1}$ in $Y_2 = B_2^{-d}$. Oskar v Bojanovem imenu nadaljuje s protokolom in v drugem krogu pošlje začasni javni ključ (Y_1, Y_2) nazaj Aniti. Ko ta prejme sporočilo, sledi korakom protokola in iz skupne skrivnosti $\sigma = Y_1^{a_1+ca_3+x}\, Y_2^{a_2+ca_4+x}\, B_1^{x}\, B_2^{dx}$ izpelje sejni ključ $K = F_{\sigma}(\mathrm{Cert}_{\mathcal{A}}, \mathrm{Cert}_{\mathcal{B}}, X_1, X_2, Y_1, Y_2)$. Tega lahko izračuna tudi Oskar, saj pozna Anitin zasebni ključ (a_1, a_2, a_3, a_4), s katerim lahko skupno skrivnost izračuna po enačbi $\sigma = B_1^{-a_1-ca_3}\, B_2^{-da_2-cda_4}$. Na koncu protokola si tako Anita in Oskar delita isti sejni ključ. To pa pomeni, da je napadalcu Oskarju uspela impersonacija, tj. prepričati Anito, da se pogovarja z Bojanom, čeprav v resnici komunicira z njim. □

Opomba 5.1. Napadalec Oskar lahko predstavljen napad izvede le v Bojanovem imenu, saj Anita ne more izračunati skupne skrivnosti, tudi če pozna Bojanov trajni zasebni ključ. Za njen izračun bi namreč morala še pred izbiro svojega začasnega javnega ključa (X_1, X_2) poznati zgostitev $c = H_{s_{\mathcal{A}}}(\mathrm{Cert}_{\mathcal{A}}, Y_1, Y_2)$. To pa ni možno, saj ji Bojan svoj začasni javni ključ (Y_1, Y_2) pošlje šele v drugem krogu protokola, ko že prejme njen začasni javni ključ.

V podrazdelku 2.4.5 smo omenili, da varnostni model eCK med drugim zajema tudi napad lažnega predstavljanja z razkritim ključem. Zato so vsi protokoli, katerih varnost je možno dokazati v tem modelu, odporni na ta napad. Iz zgornjega izreka lahko zato izpeljemo naslednji rezultat.

Posledica 5.1. *Okamotov protokol za dvostranski overjen dogovor o ključu ni varen v varnostnem modelu eCK.* □

Okamotov protokol torej vsebuje resno varnostno pomanjkljivost, ki krši osnovna varnostna načela protokolov za overjen dogovor o ključu, opisana v podrazdelku 3.2.2. Zato ni varen in ga ne moremo uporabljati za varen dogovor o ključu.

Za konec omenimo še, da podobnega napada ni možno izvesti na popravljeno verzijo Okamotovega protokola, ki je bila objavljena v članku Okamoto [119]. V njej je namreč avtor spremenil izračun skupne skrivnosti $\sigma_{\mathcal{A}}$ in

s tem odstranil Anitin začasni ključ x iz potenc $Y_1^{a_1+ca_3+x}$ in $Y_2^{a_2+ca_4+x}$. Zaradi te spremembe naš napad ni možen, saj napadalec tudi z ustrezno izbiro Bojanovega začasnega javnega ključa (Y_1, Y_2) ne more odstraniti začasnih zasebnih ključev x in y iz eksponenta skupne skrivnosti.

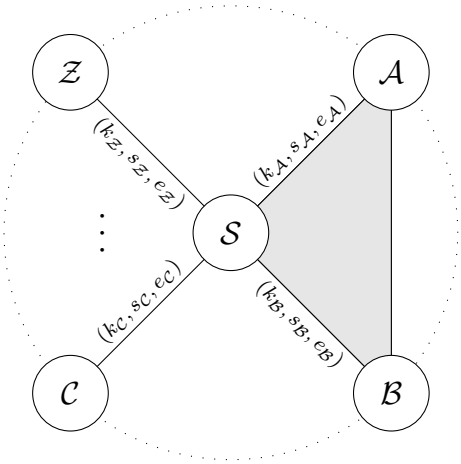
5.2 Chenov protokol

Leta 2008 so Chen, Lee in Chen predlagali tristranski protokol za overjeno izmenjavo ključa [38], v katerem si vsak uporabnik deli skrivnost z zaupanja vrednim strežnikom, preko katerega nato poteka dogovor o ključu. Za razliko od tristranske šifrirane izmenjave ključa na osnovi gesel (glej protokol 7), njihov protokol le-teh ne uporablja in zato ni ranljiv na napade s slovarjem. Poleg tega strežniku ni potrebno hraniti varnostno občutljive tabele, katero bi lahko napadalec v primeru razkritja izkoristil za izračun osebnih gesel uporabnikov. Predlagan protokol je izjemno učinkovit, saj je dogovor o ključu možno opraviti s preprostimi izračuni že v treh krogih. Zato bi bil primeren tudi za uporabo na mobilnih platformah, saj bi lahko z njim varnim aplikacijam zmanjšali začetno komunikacijo zakasnitev.

V tem razdelku bomo pokazali, da Chenov protokol ne izpolnjuje vseh varnostnih zahtev za overjen dogovor o ključu, saj lahko napadalec izvede napad lažnega predstavljanja z razkritim ključem.

Predstavitev protokola

V Chenovem protokolu se Anita in Bojan preko zaupanja vrednega strežnika dogovorita za skupni sejni ključ ter medsebojno overita. Za izvedbo dogovora mora vsak imeti v lasti zasebni ključ (t.i. overitvene podatke), ki jima ga z uporabo glavnega tajnega ključa izračuna strežnik in preko varnega kanala pošlje v trajno last. Protokol lahko v grobem razdelimo na dva dela. V prvem delu se Anita in Bojan z neoverjenim Diffie-Hellmanovim protokolom dogovorita za sejni ključ, v drugem delu pa se z uporabo zgoščevalnih funkcij in zasebnih ključev še medsebojno overita preko strežnika. Varnost protokola naj bi zato temeljila le na računskem Diffie-Hellmanovem problemu in zgoščevalnih funkcijah, ki so odporne na trke.



Slika 5.2: Razporeditev tajnih ključev v Chenovem protokolu

Protokol 9 Chenov tristranski dogovor o ključu

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bosta p in q praštevili, tako da velja $p \mid q-1$, nadalje $g \in \mathbb{Z}_q^*$ element reda p in H enosmerna zgoščevalna funkcija odporna na trke.

   (b) Zaupanja vreden strežnik izbere glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna glavni javni ključ $S = g^s \bmod q$ in ga javno objavi.

   (c) Anita z identiteto $\mathrm{ID}_{\mathcal{A}}$ postane upravičen uporabnik sistema, ko zaupanja vreden strežnik opravi naslednje korake: najprej izbere število $\delta_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$ in izračuna skupni tajni ključ $k_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}, \delta_{\mathcal{A}})$. Nato uporabi glavni tajni ključ s za pripravo Schnorrovega digitalnega podpisa (glej shemo 2) identitete $\mathrm{ID}_{\mathcal{A}}$, tj. $(s_{\mathcal{A}}, e_{\mathcal{A}})$, tako da izračuna vrednost $w_{\mathcal{A}} = g^{\delta_{\mathcal{A}}} \bmod q$, zgostitev $e_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}, w_{\mathcal{A}})$ in število $s_{\mathcal{A}} = (\delta_{\mathcal{A}} - s\, e_{\mathcal{A}}) \bmod p$. Na koncu pošlje overitvene podatke $(k_{\mathcal{A}}, s_{\mathcal{A}}, e_{\mathcal{A}})$ Aniti preko varnega kanala. Ti podatki so preverljivi, zato se lahko Anita s preverjanjem podpisa prepriča, ali je bila res pooblaščena s strani zaupanja vrednega strežnika. To stori tako, da najprej pridobi glavni javni ključ S, nato izračuna

vrednost $w_{\mathcal{A}} = g^{s_{\mathcal{A}}} S^{e_{\mathcal{A}}} \bmod q$ in na koncu preveri, ali je enakost $e_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}, w_{\mathcal{A}})$ res izpolnjena.

2. *Izmenjava sporočil.*

$$\begin{aligned}
\mathcal{A} \to \mathcal{B}: &\quad \mathrm{ID}_{\mathcal{A}}, R_{\mathcal{A}}, t_{\mathcal{A}} && \text{(1a)}\\
\mathcal{A} \to \mathcal{S}: &\quad \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{A}}, t_{\mathcal{A}}, C_{\mathcal{AS}}, s_{\mathcal{A}}, e_{\mathcal{A}} && \text{(1b)}\\
\mathcal{B} \to \mathcal{A}: &\quad \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{B}}, t_{\mathcal{B}}, C_{\mathcal{BA}} && \text{(2a)}\\
\mathcal{B} \to \mathcal{S}: &\quad \mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{A}}, R_{\mathcal{B}}, t_{\mathcal{B}}, C_{\mathcal{BS}}, s_{\mathcal{B}}, e_{\mathcal{B}} && \text{(2b)}\\
\mathcal{S} \to \mathcal{A}: &\quad C_{\mathcal{SA}} && \text{(3a)}\\
\mathcal{S} \to \mathcal{B}: &\quad C_{\mathcal{SB}} && \text{(3b)}\\
\mathcal{A} \to \mathcal{B}: &\quad C_{\mathcal{AB}} && \text{(3c)}
\end{aligned}$$

3. *Koraki v protokolu.* Anita in Bojan se z javno znanima identitetama $\mathrm{ID}_{\mathcal{A}}$ ter $\mathrm{ID}_{\mathcal{B}}$ po uspešno izvedeni pripravi, v kateri od zaupanja vrednega strežnika prejmeta overitvene podatke $(k_{\mathcal{A}}, s_{\mathcal{A}}, e_{\mathcal{A}})$ in $(k_{\mathcal{B}}, s_{\mathcal{B}}, e_{\mathcal{B}})$, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov:

   (a) Anita izbere svoj začasni zasebni ključ $r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_{\mathcal{A}} = g^{r_{\mathcal{A}}} \bmod q$, prebere časovni žig $t_{\mathcal{A}}$ iz zanesljivega časovnega vira in izračuna $C_{\mathcal{AS}} = H(\mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{A}}, t_{\mathcal{A}}, k_{\mathcal{A}})$. Nato pošlje sporočilo (1a) Bojanu in sporočilo (1b) zaupanja vrednemu strežniku.

   (b) Po prejetju Anitinega sporočila tudi Bojan izbere svoj začasni zasebni ključ $r_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_{\mathcal{B}} = g^{r_{\mathcal{B}}} \bmod q$ in prebere časovni žig $t_{\mathcal{B}}$ iz zanesljivega časovnega vira. Nato izračuna skupno skrivnost $\sigma_{\mathcal{B}} = (R_{\mathcal{A}})^{r_{\mathcal{B}}} \bmod q$, zgostitvi $C_{\mathcal{BA}} = H(t_{\mathcal{A}}, R_{\mathcal{A}}, R_{\mathcal{B}}, \sigma_{\mathcal{B}})$ in $C_{\mathcal{BS}} = H(\mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{A}}, R_{\mathcal{B}}, t_{\mathcal{B}}, k_{\mathcal{B}})$ ter pošlje sporočilo (2a) Aniti in sporočilo (2b) zaupanja vrednemu strežniku.

   (c) Ko strežnik prejme sporočili, najprej preveri, ali časovna žiga $t_{\mathcal{A}}$ in $t_{\mathcal{B}}$ ležita v sprejemljivem časovnem intervalu. Nato uporabi glavni tajni ključ s in izračuna števili $\delta_{\mathcal{A}} = (s_{\mathcal{A}} + s\, e_{\mathcal{A}}) \bmod p$ ter $\delta_{\mathcal{B}} = (s_{\mathcal{B}} + s\, e_{\mathcal{B}}) \bmod p$, iz katerih izpelje ključa $k_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}, \delta_{\mathcal{A}})$ in $k_{\mathcal{B}} = H(\mathrm{ID}_{\mathcal{B}}, \delta_{\mathcal{B}})$, ki si ju v tajnosti "deli" z Anito in Bojanom. Slednja lahko izračuna le on, saj edini pozna glavni tajni ključ s.

Izračunati pa jih mora ponovno, saj si ju pri registraciji ni shranil. Sledi preverjanje pristnosti sporočil. Strežnik izračuna zgostitvi $C'_{\mathcal{AS}} = H(\mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{A}}, t_{\mathcal{A}}, k_{\mathcal{A}})$ in $C'_{\mathcal{BS}} = H(\mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{A}}, R_{\mathcal{B}}, t_{\mathcal{B}}, k_{\mathcal{B}})$ ter ju primerja s prejetima vrednostma $C_{\mathcal{AS}}$ in $C_{\mathcal{BS}}$. Če preverjanje ni bilo uspešno, zaključi protokol, sicer izračuna zgostitvi $C_{\mathcal{SA}} = H(C_{\mathcal{AS}}, R_{\mathcal{B}}, t_{\mathcal{B}}, k_{\mathcal{A}})$ in $C_{\mathcal{SB}} = H(C_{\mathcal{BS}}, R_{\mathcal{A}}, t_{\mathcal{A}}, k_{\mathcal{B}})$ ter ju v sporočilu (3a) in (3b) pošlje Aniti ter Bojanu za medsebojno overjanje.

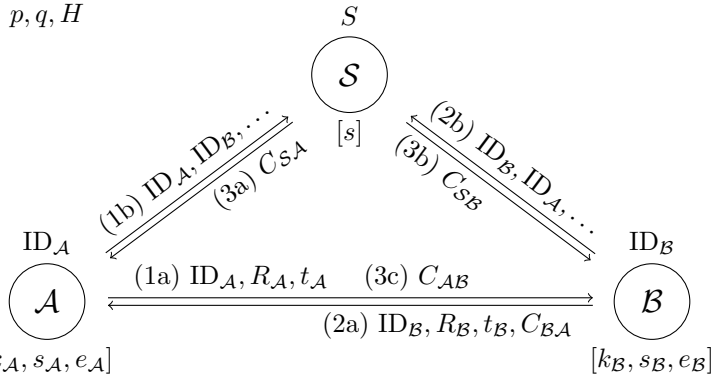
Neodvisno od strežnika, Anita preveri ustreznost časovne razlike $t - t_{\mathcal{A}}$, kjer je t čas prejetja Bojanovega sporočila. Nato izračuna skupno skrivnost $\sigma_{\mathcal{A}} = (R_{\mathcal{B}})^{r_{\mathcal{A}}} \bmod q$ in zgostitev $C'_{\mathcal{BA}} = H(t_{\mathcal{A}}, R_{\mathcal{A}}, R_{\mathcal{B}}, \sigma_{\mathcal{A}})$. Slednjo primerja z vrednostjo $C_{\mathcal{BA}}$, ki jo je Bojan poslal v svojem sporočilu. Če preverjanje uspe, nadaljuje s protokolom in zgostitev $C_{\mathcal{AB}} = H(t_{\mathcal{B}}, C_{\mathcal{BA}}, \sigma_{\mathcal{A}})$ pošlje v sporočilu (3c) Bojanu.

(d) Po izmenjavi sporočil Anita preveri dopustnost razlike $t' - t_{\mathcal{A}}$, kjer je t' čas prejetja strežnikovega sporočila. Nato preveri še njegovo pristnost, tako da izračuna zgostitev $C'_{\mathcal{SA}} = H(C_{\mathcal{AS}}, R_{\mathcal{B}}, t_{\mathcal{B}}, k_{\mathcal{A}})$ in jo primerja z vrednostjo $C_{\mathcal{SA}}$, ki jo je strežnik poslal v svojem sporočilu.

Podobno Bojan preveri časovno razliko $t'' - t_{\mathcal{B}}$, kjer je t'' čas prejetja strežnikovega in Anitinega sporočila. Nato izračuna zgostitvi $C'_{\mathcal{SB}} = H(C_{\mathcal{BS}}, R_{\mathcal{A}}, t_{\mathcal{A}}, k_{\mathcal{B}})$ in $C'_{\mathcal{AB}} = H(t_{\mathcal{B}}, C_{\mathcal{BA}}, \sigma_{\mathcal{B}})$, ter ju primerja z vrednostma $C_{\mathcal{SB}}$ in $C_{\mathcal{AB}}$, ki sta ju strežnik in Anita poslala v svojih sporočilih. S tem se prepriča, da je sporočilo od strežnika pristno in da Anita pozna skupno skrivnost.

V kolikor so bili vsi izračuni v protokolu pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in Anita ter Bojan lahko tako izpeljeta skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.

Iz opisa protokola je očitno, da Anita in Bojan iz izračunane skupne skrivnosti $\sigma = g^{r_{\mathcal{A}} r_{\mathcal{B}}} \bmod q$ izpeljeta isti sejni ključ in ga sprejmeta le, če so bila vsa preverjanja v protokolu uspešna.



(a)

$$\begin{aligned}
\mathcal{A}: &\quad r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*, \quad R_\mathcal{A} = g^{r_\mathcal{A}} \bmod q, \quad t_\mathcal{A} = \text{čas}() \\
\mathcal{A}: &\quad C_{\mathcal{AS}} = H(\mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{B}, R_\mathcal{A}, t_\mathcal{A}, k_\mathcal{A}) \\
\mathcal{A} \to \mathcal{B}: &\quad \mathrm{ID}_\mathcal{A}, R_\mathcal{A}, t_\mathcal{A} \qquad (1a) \\
\mathcal{A} \to \mathcal{S}: &\quad \mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{B}, R_\mathcal{A}, t_\mathcal{A}, C_{\mathcal{AS}}, s_\mathcal{A}, e_\mathcal{A} \qquad (1b)
\end{aligned}$$

(b)

$$\begin{aligned}
\mathcal{B}: &\quad r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*, \quad R_\mathcal{B} = g^{r_\mathcal{B}} \bmod q, \quad t_\mathcal{B} = \text{čas}() \\
\mathcal{B}: &\quad \sigma = (R_\mathcal{A})^{r_\mathcal{B}} \bmod q, \quad C_{\mathcal{BA}} = H(t_\mathcal{A}, R_\mathcal{A}, R_\mathcal{B}, \sigma) \\
\mathcal{B}: &\quad C_{\mathcal{BS}} = H(\mathrm{ID}_\mathcal{B}, \mathrm{ID}_\mathcal{A}, R_\mathcal{B}, t_\mathcal{B}, k_\mathcal{B}) \\
\mathcal{B} \to \mathcal{A}: &\quad \mathrm{ID}_\mathcal{B}, R_\mathcal{B}, t_\mathcal{B}, C_{\mathcal{BA}} \qquad (2a) \\
\mathcal{B} \to \mathcal{S}: &\quad \mathrm{ID}_\mathcal{B}, \mathrm{ID}_\mathcal{A}, R_\mathcal{B}, t_\mathcal{B}, C_{\mathcal{BS}}, s_\mathcal{B}, e_\mathcal{B} \qquad (2b)
\end{aligned}$$

(c)

$$\begin{aligned}
\mathcal{S}: &\quad \delta_\mathcal{A} = (s_\mathcal{A} + s\, e_\mathcal{A}) \bmod p, \quad k_\mathcal{A} = H(\mathrm{ID}_\mathcal{A}, \delta_\mathcal{A}) \\
\mathcal{S}: &\quad \delta_\mathcal{B} = (s_\mathcal{B} + s\, e_\mathcal{B}) \bmod p, \quad k_\mathcal{B} = H(\mathrm{ID}_\mathcal{B}, \delta_\mathcal{B}) \\
\mathcal{S}: &\quad C_{\mathcal{AS}} \stackrel{?}{=} H(\mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{B}, R_\mathcal{A}, t_\mathcal{A}, k_\mathcal{A}) \\
\mathcal{S}: &\quad C_{\mathcal{BS}} \stackrel{?}{=} H(\mathrm{ID}_\mathcal{B}, \mathrm{ID}_\mathcal{A}, R_\mathcal{B}, t_\mathcal{B}, k_\mathcal{B}) \\
\mathcal{S} \to \mathcal{A}: &\quad C_{\mathcal{SA}} = H(C_{\mathcal{AS}}, R_\mathcal{B}, t_\mathcal{B}, k_\mathcal{A}) \qquad (3a) \\
\mathcal{S} \to \mathcal{B}: &\quad C_{\mathcal{SB}} = H(C_{\mathcal{BS}}, R_\mathcal{A}, t_\mathcal{A}, k_\mathcal{B}) \qquad (3b) \\
\mathcal{A}: &\quad \sigma = (R_\mathcal{B})^{r_\mathcal{A}} \bmod q, \quad C_{\mathcal{BA}} \stackrel{?}{=} H(t_\mathcal{A}, R_\mathcal{A}, R_\mathcal{B}, \sigma) \\
\mathcal{A} \to \mathcal{B}: &\quad C_{\mathcal{AB}} = H(t_\mathcal{B}, C_{\mathcal{BA}}, \sigma) \qquad (3c)
\end{aligned}$$

(d)

$$\begin{aligned}
\mathcal{A}: &\quad C_{\mathcal{SA}} \stackrel{?}{=} H(C_{\mathcal{AS}}, R_\mathcal{B}, t_\mathcal{B}, k_\mathcal{A}) \\
\mathcal{B}: &\quad C_{\mathcal{SB}} \stackrel{?}{=} H(C_{\mathcal{BS}}, R_\mathcal{A}, t_\mathcal{A}, k_\mathcal{B}) \\
\mathcal{B}: &\quad C_{\mathcal{AB}} \stackrel{?}{=} H(t_\mathcal{B}, C_{\mathcal{BA}}, \sigma) \\
\mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma)
\end{aligned}$$

Slika 5.3: Chenov tristranski dogovor o ključu

Varnostna analiza

Preden razkrijemo varnostno pomanjkljivost Chenovega protokola, si podrobno oglejmo, kako Anita strežniku zagotovi pristnost poslanega sporočila v prvem krogu protokola. To stori tako, da sporočilu priloži zgostitev $C_{\mathcal{AS}}$ poslanih podatkov in svojega tajnega ključa $k_{\mathcal{A}}$. Ko strežnik prejme njeno sporočilo, uporabi glavni tajni ključ s in iz prejetih vrednosti $e_{\mathcal{A}}$ ter $s_{\mathcal{A}}$ izračuna Anitin ključ $k_{\mathcal{A}}$. Nato še sam izračuna zgostitev sporočila $C'_{\mathcal{AS}}$ in jo primerja s tisto, ki jo je Anita priložila sporočilu. Če sta zgostitvi enaki, potem je sporočilo pristno in strežnik ima zagotovilo, da je podatke poslala Anita. Podobno poteka tudi overjanje strežnika. Le-ta sporočilo, v katerem Aniti pošlje zgostitev Bojanovega začasnega javnega ključa $R_{\mathcal{B}}$ overi tako, da v izračun zgostitve $C_{\mathcal{SA}}$ vključi tajni ključ $k_{\mathcal{A}}$. Ko v zaključku protokola Anita prejme to sporočilo, tudi sama izračuna zgostitev $C'_{\mathcal{SA}}$ in jo primerja s priloženo. Če se ujemata, potem je strežnikovo sporočilo pristno in Anita je prepričana, da je vrednost $R_{\mathcal{B}}$ izbral Bojan.

Udeleženci protokola pristnost sporočil dokazujejo z uporabo simetrične kriptografije. Za to uporabljajo zgoščevalno funkcijo, s katero drug drugemu pošljejo dokaz, da poznajo skupni tajni ključ. Enak učinek bi lahko dosegli tudi z uporabo kod za overjanje podatkov ali s šifriranjem sporočil. Vsi omenjeni pristopi so varni in se zelo pogosto uporabljajo v protokolih, kjer mora neka oseba dokazati posest skupne skrivnosti, kot je to npr. osebno geslo ali tajni ključ. V Chenovem protokolu so se avtorji odločili za uporabo zgoščevalnih funkcij zelo verjetno zaradi njihove manjše računske zahtevnosti. Ker pa pristnost izmenjanih sporočil ni zadosten pogoj za medsebojno overitev, morajo udeleženci sproti preverjati tudi časovne žige sporočil. S tem preprečijo, da bi napadalec v protokolu pošiljal stara sporočila s prejšnjih sej.

Medsebojno overjanje Anite in strežnika torej temelji na skupni skrivnosti $k_{\mathcal{A}}$. Kdorkoli pozna to skrivnost oziroma jo zna izračunati iz vrednosti $e_{\mathcal{A}}$ in $s_{\mathcal{A}}$, lahko Aniti pošilja sporočila v imenu strežnika in obratno. Tu pa se pojavi težava, saj Anita nikoli ne preverja, ali strežnik pozna glavni tajni ključ s. Protokol bi se uspešno zaključil tudi, če bi strežnik imel pri sebi shranjene le vse zasebne ključe uporabnikov in glavnega tajnega ključa sploh ne bi poznal. V tem primeru bi bil protokol po eni strani bolj učinkovit, saj

strežniku ne bi bilo potrebno vedno znova računati tajnih ključev, po drugi strani pa bi porabil ogromno prostora za hranjenje le-teh. Overjanje v protokolu se torej izvaja zgolj z uporabo simetrične kriptografije, kar pa prinese dodatne ranljivosti. Tako se lahko napadalec, ki pozna Anitin tajni ključ $k_{\mathcal{A}}$, njej lažno predstavi kot strežnik. To lahko stori kljub temu, da ne pozna glavnega tajnega ključa s. Vendar to še ne bi bil problem, če Anita ne bi zaupala strežniku, da je poslano vrednost $R_{\mathcal{B}}$ izbral Bojan. Ker pa mu, lahko to izkoristi napadalec z izvedbo naslednjega napada, v katerem sam izbere Bojanov začasni javni ključ $R_{\mathcal{B}}$ in se tako Aniti lažno predstavi v njegovem imenu.

Izrek 5.2. *Chenov protokol za tristranski overjen dogovor o ključu ni odporen na napad lažnega predstavljanja z razkritim ključem. Napadalec Oskar se lahko Aniti lažno predstavi kot Bojan, če pozna njene overitvene podatke* $(k_{\mathcal{A}}, s_{\mathcal{A}}, e_{\mathcal{A}})$ *in obratno.*

Dokaz. Naj bo Oskar napadalec, ki pozna Anitine tajne overitvene podatke $(k_{\mathcal{A}}, e_{\mathcal{A}}, s_{\mathcal{A}})$. Če se ji želi lažno predstaviti kot Bojan, potem počaka, da Anita prične dogovor o ključu in pošlje sporočilo $(\mathrm{ID}_{\mathcal{A}}, R_{\mathcal{A}}, t_{\mathcal{A}})$ Bojanu ter sporočilo $(\mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{A}}, t_{\mathcal{A}}, C_{\mathcal{AS}}, e_{\mathcal{A}}, s_{\mathcal{A}})$ zaupanja vrednemu strežniku v prvem krogu protokola. Takrat se aktivno vključi v protokol in prestreže poslani sporočili, tako da nobeno ne prispe do svojega naslovnika. Nato v Bojanovem imenu izbere začasni zasebni ključ $r_{\mathcal{O}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna njegov začasni javni ključ $R_{\mathcal{O}} = g^{r_{\mathcal{O}}} \bmod q$, skupno skrivnost $\sigma_{\mathcal{O}} = (R_{\mathcal{A}})^{r_{\mathcal{O}}} \bmod q$, časovni žig $t_{\mathcal{O}}$ in zgostitev $C_{\mathcal{OA}} = H(t_{\mathcal{A}}, R_{\mathcal{A}}, R_{\mathcal{O}}, \sigma_{\mathcal{O}})$. Za tem sestavi sporočilo $(\mathrm{ID}_{\mathcal{B}}, R_{\mathcal{O}}, t_{\mathcal{O}}, C_{\mathcal{OA}})$ in ga pošlje nazaj Aniti, kot je to opredeljeno v drugem krogu protokola. Ker pozna njen tajni ključ $k_{\mathcal{A}}$, lahko v naslednjem krogu izračuna še zgostitev $C'_{\mathcal{OA}} = H(C_{\mathcal{AS}}, R_{\mathcal{O}}, t_{\mathcal{O}}, k_{\mathcal{A}})$ in ji jo v imenu strežnika pošlje v sporočilu. Ko Anita v zaključku protokola prejme obe sporočili, najprej preveri vse časovne žige. Nato izračuna skupno skrivnost $\sigma_{\mathcal{A}} = (R_{\mathcal{O}})^{r_{\mathcal{A}}} \bmod q$ in preveri zgostitvi $C_{\mathcal{OA}}$ in $C'_{\mathcal{OA}}$, ki naj bi ju prejela od Bojana in strežnika. Preverjanje obeh uspe, saj Oskar pozna vrednost $r_{\mathcal{O}}$ in zasebni ključ $k_{\mathcal{A}}$, ki sta bila potrebna za izračun skupne skrivnosti $\sigma_{\mathcal{A}} (= \sigma_{\mathcal{O}})$ in za preverjanje pristnosti strežnikovega sporočila. Anita tako sprejme sejni ključ $K = \mathrm{kdf}(\sigma)$ prepričana, da si ga deli z Bojanom, čeprav si ga v resnici deli z Oskarjem.

Na podoben način se lahko napadalec Oskar tudi Bojanu lažno predstavi kot Anita, če pozna njegove overitvene podatke $(k_{\mathcal{B}}, e_{\mathcal{B}}, s_{\mathcal{B}})$. Od tod lahko torej zaključimo, da Chenov protokol ni odporen na napad lažnega predstavljanja z razkritim ključem. □

Iz zgornjega izreka sledi, da Chenov protokol nima vseh želenih varnostnih lastnosti, ki smo jih omenili v podrazdelku 3.2.2. Protokol je potreben izboljšave in ga zato ne moremo uporabljati za varen overjen dogovor o ključu.

Predlog izboljšave

Chenov protokol je ranljiv za napad lažnega predstavljanja z razkritim ključem, ker sporočila, ki si ju Anita in Bojan izmenjata z zaupanja vrednim strežnikom, niso pravilno overjena. Ta pomanjkljivost je razvidna že iz opisa protokola, saj Anita in Bojan pri preverjanju pristnosti nikjer ne uporabljata strežnikov javni ključ S. Torej nimata zagotovila, da strežnik res pozna glavni tajni ključ s. Pravzaprav preverjata le, če strežnik pozna tajna ključa $k_{\mathcal{A}}$ in $k_{\mathcal{B}}$. Slednja se da sicer izračunati z uporabo glavnega ključa s, vendar to še ne pomeni, da strežnik ta ključ v resnici pozna.

Ranljivost protokola se pojavi zaradi uporabe simetrične kriptografije pri zagotavljanju pristnosti sporočil. Ta napadalcu omogoča, da se lahko Aniti predstavi kot strežnik, če pozna njen tajni ključ $k_{\mathcal{A}}$. Te težave bi se znebili, če bi pristnost preverjali z uporabo kriptografije javnih ključev. V tem primeru bi se lahko napadalec, ki je razkril Anitin tajni ključ, predstavljal le kot Anita in v nobenem primeru kot strežnik.

Preprosta rešitev za preprečitev omenjenega napada bi bila, da bi strežnik z glavnim tajnim ključem na drugačen način overil poslane podatke. To bi lahko storil z uporabo sheme za digitalni podpis, s katero bi podpisal svoja sporočila. Takšna rešitev ni najboljša, saj bi strežnik moral opraviti veliko dodatnih izračunov, zaradi katerih bi protokol hitro postal neučinkovit. Verjetno pa zelo preprost način za overjanje njegovih sporočil ne obstaja, zato bi bilo potrebno korenito spremeniti zasnovo protokola.

5.3 Tanov protokol

Kmalu po objavi Chenovega protokola so Yang in sodelavci [155] opazili, da je njegova računska zahtevnost še vedno zelo velika. V protokolu je namreč potrebno izračunati in preveriti Schnorrov digitalni podpis, kar pa je zaradi modularnega potenciranja časovno potratno. Število izmenjanih sporočil za potrebe medsebojnega overjanja ni majhno, prav tako pa ne moremo prezreti njihove velikosti. Protokol zato ni primeren za mobilne komunikacijske storitve, kot sta npr. GSM in 3GPP. Za odpravo teh slabosti so predlagali učinkovitejši tristranski protokol za overjen dogovor o ključu, ki temelji na kriptografiji z eliptičnimi krivuljami. Varnostno analizo slednjega so opravili Pu in sodelavci [127] ter zaključili, da predlagani protokol ni odporen na napad deljenja ključa z neznano osebo. Zato so poiskali izboljšave, s katerimi je možno ta napad preprečiti. To ranljivost je opazil tudi Tan [144], ki pa je hkrati odkril nove varnostne pomanjkljivosti, zaradi katerih Yangov protokol ni odporen na vzporedni napad in na napad lažnega predstavljanja. Tudi on je predlagal nov protokol, v katerem naj bi vse omenjene slabosti odpravil in pri tem ohranil učinkovitost.

V tem razdelku bomo pokazali, da je Tanov tristranski protokol za overjen dogovor o ključu ranljiv za napad lažnega predstavljanja in posledično na napad vmesne osebe. Podoben napad je možno izvesti tudi na Yangov in Pujev protokol, zato vsi trije omenjeni protokoli niso overjeni in jih ne moremo uporabljati za varen dogovor o ključu.

Predstavitev protokola

Tanov protokol je tristranski overjen dogovor o ključu, s katerim se lahko Anita in Bojan preko zaupanja vrednega strežnika dogovorita za skupni sejni ključ ter medsebojno overita. Protokol je zelo podoben Diffie-Hellmanovem dogovoru o ključu, saj se od njega razlikuje le v načinu izmenjave začasnih javnih ključev. Ta poteka posredno preko strežnika v šifrirani obliki, zato se mora vsak udeleženec med izvajanjem protokola z njim dogovoriti za tajni simetrični ključ. Tudi ta dogovor temelji na Diffie-Hellmanovem protokolu, za njegovo izvedbo pa morajo Anita, Bojan in strežnik imeti v posesti vsak svoj trajni zasebni ključ, medtem ko morajo pripadajoče javne ključe

overiti pri certifikatni agenciji in javno objaviti v digitalnih potrdilih. S šifriranjem sporočil Anita in Bojan dosežeta, da se ob izmenjavi tajnopisov hkrati tudi overita z zaupanja vrednim strežnikom. Posledično sta zato overjena tudi izmenjana začasna javna ključa, iz katerih izračunata skupno skrivnost in izpeljeta sejni ključ. Varnost protokola naj bi zato temeljila na Diffie-Hellmanovem problemu in na varnosti uporabljene simetrične šifre.

Protokol 10 Tanov tristranski dogovor o ključu

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bo E eliptična krivulja definirana z enačbo $y^2 = x^3 + ax + b$ nad končnim obsegom $\mathbb{F}_q$, ki vsebuje aditivno grupo točk $\mathbb{G} = \langle P \rangle$ velikega praštevilskega reda p, in E_k šifrirna ter D_k odšifrirna funkcija varne simetrične šifre (npr. AES), kjer smo s k označili simetrični ključ.

   (b) Zaupanja vreden strežnik izbere svoj trajni zasebni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$ in pripadajoči javni ključ $S = sP$ pošlje certifikatni agenciji v registracijo ter izdajo digitalnega potrdila $\mathrm{Cert}_{\mathcal{S}}$.

   (c) Podobno Anita izbere svoj trajni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$ in javni ključ $A = aP$ objavi v digitalnem potrdilu $\mathrm{Cert}_{\mathcal{A}}$.

2. *Izmenjava sporočil.*

$$\begin{aligned}
\mathcal{A} \to \mathcal{B}: &\quad \mathrm{ID}_{\mathcal{A}} && \text{(1a)}\\
\mathcal{A} \to \mathcal{S}: &\quad \mathrm{ID}_{\mathcal{A}}, R_{\mathcal{A}}, t_{\mathcal{A}}, C_{\mathcal{AS}} && \text{(1b)}\\
\mathcal{B} \to \mathcal{A}: &\quad \mathrm{ID}_{\mathcal{B}} && \text{(2a)}\\
\mathcal{B} \to \mathcal{S}: &\quad \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{B}}, t_{\mathcal{B}}, C_{\mathcal{BS}} && \text{(2b)}\\
\mathcal{S} \to \mathcal{A}: &\quad C_{\mathcal{SA}} && \text{(3a)}\\
\mathcal{S} \to \mathcal{B}: &\quad C_{\mathcal{SB}} && \text{(3b)}
\end{aligned}$$

3. *Koraki v protokolu.* Po uspešno izvedeni pripravi, v kateri si Anita, Bojan in zaupanja vreden strežnik izberejo trajne zasebne ključe a, b in s ter svoje javne ključe A, B in S objavijo v digitalnih potrdilih $\mathrm{Cert}_{\mathcal{A}}$, $\mathrm{Cert}_{\mathcal{B}}$ ter $\mathrm{Cert}_{\mathcal{S}}$, se lahko Anita in Bojan dogovorita za skupni sejni ključ. To storita z izvedbo naslednjih korakov, v katerih smo s $\bar{T}$ označili koordinato x točke T:

(a) Anita izbere začasna zasebna ključa $r_{\mathcal{A}}, w_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasna javna ključa $R_{\mathcal{A}} = r_{\mathcal{A}}A$ in $W_{\mathcal{A}} = w_{\mathcal{A}}P$ ter prebere časovni žig $t_{\mathcal{A}}$ iz zanesljivega časovnega vira. Nato izračuna točko $K_{\mathcal{A}} = ar_{\mathcal{A}}S$ na eliptični krivulji, katero si bo v tajnosti delila z zaupanja vrednim strežnikom, in njeno koordinato x uporabi za pripravo tajnopisa $C_{\mathcal{AS}} = E_{\bar{K_{\mathcal{A}}}}(\mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{A}}, W_{\mathcal{A}}, t_{\mathcal{A}})$. Na koncu pošlje sporočilo (1a) Bojanu in sporočilo (1b) zaupanja vrednemu strežniku.

(b) Po prejetju Anitinega sporočila tudi Bojan izbere svoja začasna zasebna ključa $r_{\mathcal{B}}, w_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna javna ključa $R_{\mathcal{B}} = r_{\mathcal{B}}B$ in $W_{\mathcal{B}} = w_{\mathcal{B}}P$ ter prebere časovni žig $t_{\mathcal{B}}$ iz zanesljivega časovnega vira. Nato izračuna točko $K_{\mathcal{B}} = br_{\mathcal{B}}S$ na eliptični krivulji, pripravi tajnopis $C_{\mathcal{BS}} = E_{\bar{K_{\mathcal{B}}}}(\mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{A}}, R_{\mathcal{B}}, W_{\mathcal{B}}, t_{\mathcal{B}})$ in pošlje sporočilo (2a) Aniti ter sporočilo (2b) zaupanja vrednemu strežniku.

(c) Ko strežnik prejme sporočili, najprej izračuna točki $K_{\mathcal{A}} = sR_{\mathcal{A}}$ in $K_{\mathcal{B}} = sR_{\mathcal{B}}$, ki si ju v tajnosti deli z Anito ter Bojanom. Nato z njunima koordinatama x odšifrira prejeta tajnopisa $C_{\mathcal{AS}}$ in $C_{\mathcal{BS}}$ ter preveri veljavnost čistopisov $(\mathrm{ID}'_{\mathcal{A}}, \mathrm{ID}'_{\mathcal{B}}, R'_{\mathcal{A}}, W_{\mathcal{A}}, t'_{\mathcal{A}}) = D_{\bar{K_{\mathcal{A}}}}(C_{\mathcal{AS}})$ in $(\mathrm{ID}''_{\mathcal{B}}, \mathrm{ID}''_{\mathcal{A}}, R'_{\mathcal{B}}, W_{\mathcal{B}}, t'_{\mathcal{B}}) = D_{\bar{K_{\mathcal{B}}}}(C_{\mathcal{BS}})$. Pri tem mora preveriti, ali so odšifrirane identitete, javni začasni ključi in časovni žigi veljavni ter se ujemajo z vrednostmi, ki jih je prejel v sporočilih. S tem dobi zagotovilo, da sta sporočili res poslala Anita in Bojan. Iz zanesljivega časovnega vira nato prebere časovni žig $t_{\mathcal{S}}$ in pripravi tajnopisa $C_{\mathcal{SA}} = E_{\bar{K_{\mathcal{A}}}}(\mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{S}}, R_{\mathcal{A}}, W_{\mathcal{B}}, t_{\mathcal{S}})$ ter $C_{\mathcal{SB}} = E_{\bar{K_{\mathcal{B}}}}(\mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{S}}, R_{\mathcal{B}}, W_{\mathcal{A}}, t_{\mathcal{S}})$. Slednja vsebujeta začasna javna ključa $W_{\mathcal{B}}$ in $W_{\mathcal{A}}$, ki ju Anita ter Bojan potrebujeta za izračun skupne skrivnosti in izpeljavo sejnega ključa. Na koncu jima tajnopisa pošlje v sporočilih (3a) in (3b).

(d) Po izmenjavi sporočil Anita odšifrira prejeti tajnopis $C_{\mathcal{SA}}$ in dobi čistopis $(\mathrm{ID}'_{\mathcal{A}}, \mathrm{ID}'_{\mathcal{S}}, R'_{\mathcal{A}}, W_{\mathcal{B}}, t_{\mathcal{S}}) = D_{\bar{K_{\mathcal{A}}}}(C_{\mathcal{SA}})$. Nato preveri ustreznost identitet $\mathrm{ID}'_{\mathcal{A}}$ in $\mathrm{ID}'_{\mathcal{S}}$, začasnega javnega ključa $R'_{\mathcal{A}}$ ter veljavnost časovnega žiga $t_{\mathcal{S}}$. Če vsa preverjanja uspejo, izračuna skupno skrivnost $\sigma_{\mathcal{A}} = w_{\mathcal{A}}W_{\mathcal{B}}$, sicer prekine izvajanje protokola.

Podobno Bojan izračuna $(\mathrm{ID}'_{\mathcal{B}}, \mathrm{ID}'_{\mathcal{S}}, R'_{\mathcal{B}}, W_{\mathcal{A}}, t_{\mathcal{S}}) = D_{\bar{K_{\mathcal{B}}}}(C_{\mathcal{SB}})$, pre-

veri identiteti $\mathrm{ID}'_{\mathcal{B}}$ in $\mathrm{ID}'_{\mathcal{S}}$, začasni javni ključ $R'_{\mathcal{B}}$ ter veljavnost časovnega žiga $t_{\mathcal{S}}$. Nato izračuna skupno skrivnost $\sigma_{\mathcal{B}} = w_{\mathcal{B}} W_{\mathcal{A}}$, če so bila vsa preverjanja uspešna, sicer tudi on zaključi izvajanje protokola.

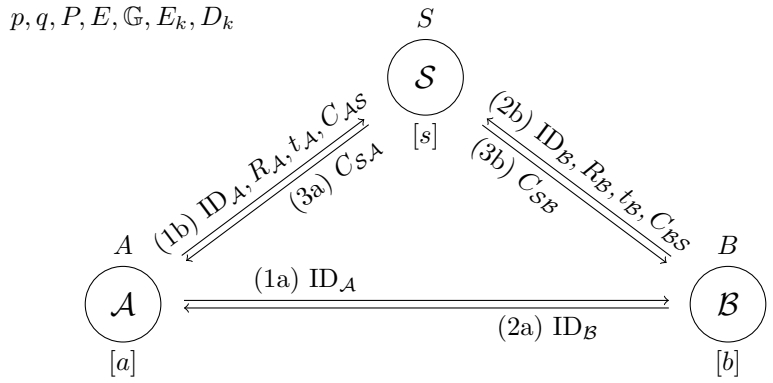
Če so bili vsi izračuni v protokolu pravilni, potem velja enakost $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in Anita ter Bojan lahko izračunata skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.

Ko se protokol zaključi, imata Anita in Bojan v posesti sejni ključ, ki sta ga izpeljala iz skupne skrivnosti $\sigma = w_{\mathcal{A}} w_{\mathcal{B}} P$. Slednjega sprejmeta le, če so bila vsa preverjanja v protokolu uspešna.

Varnostna analiza

V Tanovem protokolu se Anita in Bojan overita preko zaupanja vrednega strežnika. Ker je takšno overjanje posredno, se mora vsak izmed njiju medsebojno overiti s strežnikom. Za to se uporablja skupna skrivnost, ki jo lahko izračunata le udeleženec protokola in zaupanja vreden strežnik. Overjanje je razdeljeno na dva dela. V prvem delu se s skrivnostjo zašifrira sporočila. S tem se zagotovi njihova pristnost, saj če se tajnopisi na drugi strani pravilno odšifrirajo, to pomeni, da udeleženec protokola in strežnik poznata isto skrivnost. Hkrati se s tem zagotovi tudi njihova tajnost, saj prisluškovalci ne morejo razbrati vsebine. Nato pride drugi del, v katerem se preverijo časovni žigi izmenjanih sporočil. Ta korak je potreben, saj pristnost zagotavlja le izvor sporočila in ne, da je bil njen avtor aktiven. S tem se prepreči, da bi napadalci ponovno pošiljali sporočila, ki so že bila poslana v prejšnjih sejah protokola.

Za primer si oglejmo, kako poteka medsebojno overjanje Anite in strežnika. V prvem krogu protokola si Anita izbere začasni sejni ključ $r_{\mathcal{A}}$ in izračuna pripadajoči javni ključ $R_{\mathcal{A}} = r_{\mathcal{A}} A$. Hkrati izračuna tudi skupno skrivnost $K_{\mathcal{A}} = a r_{\mathcal{A}} S$, katero si deli z zaupanja vrednim strežnikom. Hitro lahko opazimo, da velja $K_{\mathcal{A}} = s R_{\mathcal{A}}$ oz. da lahko tudi strežnik izračuna skupno skrivnost, če pozna vrednost $R_{\mathcal{A}}$. In ker je trajni zasebni ključ s znan le njemu, nihče drug ne more izračunati te skrivnosti. Anita in strežnik



$$
\begin{aligned}
& \mathcal{A}: \quad r_\mathcal{A},\, w_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^* \quad R_\mathcal{A} = r_\mathcal{A} A, \quad W_\mathcal{A} = w_\mathcal{A} P \\
& \mathcal{A}: \quad K_\mathcal{A} = a r_\mathcal{A} S, \quad t_\mathcal{A} = \check{\mathrm{c}}\mathrm{as}() \\
\text{(a)} \quad & \mathcal{A} \to \mathcal{B}: \quad \mathrm{ID}_\mathcal{A} && \text{(1a)} \\
& \mathcal{A} \to \mathcal{S}: \quad \mathrm{ID}_\mathcal{A},\, R_\mathcal{A},\, t_\mathcal{A},\, C_{\mathcal{AS}} = E_{\bar{K_\mathcal{A}}}(\mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{B}, R_\mathcal{A}, W_\mathcal{A}, t_\mathcal{A}) && \text{(1b)} \\
& \mathcal{B}: \quad r_\mathcal{B},\, w_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*, \quad R_\mathcal{B} = r_\mathcal{B} B, \quad W_\mathcal{B} = w_\mathcal{B} P \\
& \mathcal{B}: \quad K_\mathcal{B} = b r_\mathcal{B} S, \quad t_\mathcal{B} = \check{\mathrm{c}}\mathrm{as}() \\
\text{(b)} \quad & \mathcal{B} \to \mathcal{A}: \quad \mathrm{ID}_\mathcal{B} && \text{(2a)} \\
& \mathcal{B} \to \mathcal{S}: \quad \mathrm{ID}_\mathcal{B},\, R_\mathcal{B},\, t_\mathcal{B},\, C_{\mathcal{BS}} = E_{\bar{K_\mathcal{B}}}(\mathrm{ID}_\mathcal{B}, \mathrm{ID}_\mathcal{A}, R_\mathcal{B}, W_\mathcal{B}, t_\mathcal{B}) && \text{(2b)} \\
& \mathcal{S}: \quad K_\mathcal{A} = s R_\mathcal{A}, \quad K_\mathcal{B} = s R_\mathcal{B}, \quad t_\mathcal{S} = \check{\mathrm{c}}\mathrm{as}() \\
& \mathcal{S}: \quad (\mathrm{ID}'_\mathcal{A}, \mathrm{ID}'_\mathcal{B}, R'_\mathcal{A}, W_\mathcal{A}, t'_\mathcal{A}) = D_{\bar{K_\mathcal{A}}}(C_{\mathcal{AS}}) \\
& \mathcal{S}: \quad (\mathrm{ID}''_\mathcal{B}, \mathrm{ID}''_\mathcal{A}, R'_\mathcal{B}, W_\mathcal{B}, t'_\mathcal{B}) = D_{\bar{K_\mathcal{B}}}(C_{\mathcal{BS}}) \\
\text{(c)} \quad & \mathcal{S}: \quad \mathrm{ID}_\mathcal{A} \stackrel{?}{=} \mathrm{ID}'_\mathcal{A} \stackrel{?}{=} \mathrm{ID}''_\mathcal{A}, \quad \mathrm{ID}_\mathcal{B} \stackrel{?}{=} \mathrm{ID}'_\mathcal{B} \stackrel{?}{=} \mathrm{ID}''_\mathcal{B} \\
& \mathcal{S}: \quad R_\mathcal{A} \stackrel{?}{=} R'_\mathcal{A}, \quad R_\mathcal{B} \stackrel{?}{=} R'_\mathcal{B}, \quad t_\mathcal{A} \stackrel{?}{=} t'_\mathcal{A}, \quad t_\mathcal{B} \stackrel{?}{=} t'_\mathcal{B} \\
& \mathcal{S} \to \mathcal{A}: \quad C_{\mathcal{SA}} = E_{\bar{K_\mathcal{A}}}(\mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{S}, R_\mathcal{A}, W_\mathcal{B}, t_\mathcal{S}) && \text{(3a)} \\
& \mathcal{S} \to \mathcal{B}: \quad C_{\mathcal{SB}} = E_{\bar{K_\mathcal{B}}}(\mathrm{ID}_\mathcal{B}, \mathrm{ID}_\mathcal{S}, R_\mathcal{B}, W_\mathcal{A}, t_\mathcal{S}) && \text{(3b)} \\
& \mathcal{A}: \quad (\mathrm{ID}'_\mathcal{A}, \mathrm{ID}'_\mathcal{S}, R'_\mathcal{A}, W_\mathcal{B}, t_\mathcal{S}) = D_{\bar{K_\mathcal{A}}}(C_{\mathcal{SA}}), \quad \sigma = w_\mathcal{A} W_\mathcal{B} \\
& \mathcal{A}: \quad \mathrm{ID}_\mathcal{A} \stackrel{?}{=} \mathrm{ID}'_\mathcal{A}, \quad \mathrm{ID}_\mathcal{S} \stackrel{?}{=} \mathrm{ID}'_\mathcal{S}, \quad R_\mathcal{A} \stackrel{?}{=} R'_\mathcal{A} \\
\text{(d)} \quad & \mathcal{B}: \quad (\mathrm{ID}'_\mathcal{B}, \mathrm{ID}'_\mathcal{S}, R'_\mathcal{B}, W_\mathcal{A}, t_\mathcal{S}) = D_{\bar{K_\mathcal{B}}}(C_{\mathcal{SB}}), \quad \sigma = w_\mathcal{B} W_\mathcal{A} \\
& \mathcal{B}: \quad \mathrm{ID}_\mathcal{B} \stackrel{?}{=} \mathrm{ID}'_\mathcal{B}, \quad \mathrm{ID}_\mathcal{S} \stackrel{?}{=} \mathrm{ID}'_\mathcal{S}, \quad R_\mathcal{B} \stackrel{?}{=} R'_\mathcal{B} \\
& \mathcal{A}, \mathcal{B}: \quad K = \mathrm{kdf}(\sigma)
\end{aligned}
$$

Slika 5.4: Tanov tristranski dogovor o ključu

lahko zato skupno skrivnost uporabita za zagotavljanje pristnosti in tajnosti izmenjanih sporočil.

Takšno overjanje pa vsebuje varnostno pomanjkljivost, ki se pojavi zaradi napačne izbire skupne skrivnosti $K_{\mathcal{A}} = ar_{\mathcal{A}}sP$. V njej se namreč Anitin zasebni ključ a in začasni ključ $r_{\mathcal{A}}$ zmnožita, zato je za izračun skupne skrivnosti dovolj poznati le produkt $ar_{\mathcal{A}} \bmod p$. Če si Anita za začasni ključ izbere $r_{\mathcal{A}} = ka^{-1} \bmod p$, za nek $k \in \mathbb{Z}_p^*$, potem nam preprost izračun $K_{\mathcal{A}} = ar_{\mathcal{A}}sP = aa^{-1}kS = kS$ razkrije, da Anita pri računanju skupne skrivnosti ne potrebuje več svojega zasebnega ključa.

Omenjena ranljivost je očitna že iz opisa protokola, saj strežnik v svojih izračunih nikjer ne uporablja javnih ključev udeležencev, ki so objavljeni v njihovih digitalnih potrdilih. Zato šifrirana izmenjava sporočil ne more biti overjena, kar lahko s pridom izkoristi napadalec za izvedbo naslednjega napada.

Izrek 5.3. *Tanov protokol za tristranski overjen dogovor o ključu ni odporen na napad lažnega predstavljanja. Napadalec Oskar se lahko Aniti lažno predstavi kot Bojan, in obratno.*

Dokaz. Naj bo Oskar napadalec, ki se želi Bojanu lažno predstaviti kot Anita. Za dosego tega cilja prične protokol in v prvem krogu izbere začasna zasebna ključa $r_{\mathcal{O}}, w_{\mathcal{O}} \overset{\$}{\leftarrow} \mathbb{Z}_p^*$. Zatem izračuna javna ključa $R_{\mathcal{O}} = r_{\mathcal{O}}P$ in $W_{\mathcal{O}} = w_{\mathcal{O}}P$ ter skupno skrivnost $K_{\mathcal{O}} = r_{\mathcal{O}}S$, katero uporabi za pripravo tajnopisa $C_{\mathcal{OS}} = E_{K_{\mathcal{O}}^{-}}(\mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, R_{\mathcal{O}}, W_{\mathcal{O}}, t_{\mathcal{O}})$. V Anitinem imenu nato pošlje Bojanu zahtevek za začetek dogovora o ključu skupaj z njeno identiteto $\mathrm{ID}_{\mathcal{A}}$ in sporočilo $(\mathrm{ID}_{\mathcal{A}}, R_{\mathcal{O}}, t_{\mathcal{O}}, C_{\mathcal{OS}})$ strežniku. Ko v drugem krogu Bojan prejme sporočilo, je prepričan, da se želi Anita z njim dogovoriti za skupni sejni ključ. Zato sledi korakom protokola in svojo identiteto $\mathrm{ID}_{\mathcal{B}}$ pošlje nazaj Oskarju ter sporočilo $(\mathrm{ID}_{\mathcal{B}}, R_{\mathcal{B}}, t_{\mathcal{B}}, C_{\mathcal{BS}})$ strežniku. Sledi tretji korak, v katerem strežnik izračuna skupni skrivnosti $K_{\mathcal{O}}$ in $K_{\mathcal{B}}$ ter ju uporabi za odšifriranje tajnopisov $C_{\mathcal{OS}}$ in $C_{\mathcal{BS}}$. Tu lahko opazimo, da zaupanja vreden strežnik izračuna isto skupno skrivnost $K_{\mathcal{O}} = sR_{\mathcal{O}} = r_{\mathcal{O}}sP = r_{\mathcal{O}}S$, kot jo je v prvem krogu izračunal Oskar. Tajnopis $C_{\mathcal{OS}}$ se tako uspešno odšifrira in strežnik je prepričan, da mu je sporočilo poslala Anita. Zato pripravi in pošlje tajnopisa $C_{\mathcal{SO}}$ in $C_{\mathcal{SB}}$, kot je to opredeljeno v definiciji protokola. Ko Bojan in Oskar

v zaključku protokola prejmeta sporočilo od strežnika, ga najprej odšifrirata. Bojan nato iz vrednosti $W_{\mathcal{O}}$ izračuna skupno skrivnost $\sigma = w_{\mathcal{B}} W_{\mathcal{O}}$ in iz nje izpelje sejni ključ $K = \mathrm{kdf}(\sigma)$. Ker Oskar pozna začasno vrednost $w_{\mathcal{O}}$, lahko tudi on izračuna skupno skrivnost kot $\sigma = w_{\mathcal{O}} W_{\mathcal{B}}$ in izpelje isti sejni ključ. Tako je uspel prepričati Bojana, da si deli sejni ključ z Anito, čeprav si ga v resnici deli z njim.

Podoben napad lahko napadalec Oskar izvede tudi kot Bojan. Le da mora sedaj počakati, da Anita prične protokol in pošlje zahtevek za začetek dogovora o ključu Bojanu. Takrat se aktivno vmeša, prestreže poslano sporočilo in izvede zgoraj opisani napad v Bojanovem imenu. Od tod sledi, da Tanov protokol ni odporen na napad lažnega predstavljanja. □

V podrazdelku 3.2.3 smo omenili, da je v protokolu za dvostranski dogovor o ključu možno izvesti napad vmesne osebe, če se lahko napadalec lažno predstavlja kot pobudnik protokola na eni strani in naslovnik na drugi. Podobno velja tudi za tristranske protokole, v katerih se Anita in Bojan preko zaupanja vrednega strežnika dogovorita za skupni sejni ključ. V njih napadalec Oskar preko strežnika najprej Anito lažno prepriča, da si deli sejni ključ z Bojanom, in nato še Bojana, da si deli ključ z Anito. Po uspešno izvedenem napadu si oba delita sejni ključ z Oskarjem, ki lahko izmenjana sporočila med Anito in Bojanom odšifrira, prebere, spremeni ter ponovno zašifrira, ne da bi kdorkoli to opazil.

Posledica 5.2. *Tanov protokol za tristranski overjen dogovor o ključu je ranljiv za napad vmesne osebe.* □

Tanov protokol torej omogoča aktivnemu napadalcu lažno predstavljanje v imenu poljubne osebe. Zato ni overjen, kot to trdijo avtorji, posledično pa je ranljiv še za napad vmesne osebe. Enako velja tudi za Yangov in Pujev protokol, saj imata oba zelo podobno strukturo dogovora o ključu. Ti protokoli zato niso primerni za uporabo v praksi.

Predlog izboljšave

Tanov, Yangov in Pujev protokol so ranljivi za napad lažnega predstavljanja, ker strežnik na neustrezen način preveri, če udeleženci protokola poznajo svoj

zasebni ključ. To se zgodi pri ugotavljanju pristnosti sporočil, ko strežnik s skupno skrivnostjo odšifrira prejete tajnopise. S tem naj bi dobil zagotovilo, da Anita in Bojan znata izračunati skupno skrivnosti oz. poznata svoj zasebni ključ. Vendar kot smo že pokazali v tem razdelku, temu ni tako, saj lahko Anita izračuna skupno skrivnost, ne da bi poznala svoj zasebni ključ.

Razlogi za takšno preverjanje zasebnega ključa se zelo verjetno skrivajo v dejstvu, da so avtorji za medsebojno overjanje Anite in strežnika želeli uporabiti Diffie-Hellmanovo skrivnost $K_{\mathcal{A}} = asP$. Slednjo lahko izračunata oba z uporabo svojega trajnega zasebnega ključa in javnih ključev A ter S. Vendar ima takšna skrivnost tudi svoje pomanjkljivosti, saj njen izračun ni odvisen od začasnih sejnih ključev. Zato so avtorji prišli na idejo, da naj Anita svoj javni ključ A množi z začasnim ključem $r_{\mathcal{A}}$. Skupna skrivnost se v tem primeru spremeni v $K_{\mathcal{A}} = ar_{\mathcal{A}}sP$ in je z veliko verjetnostjo različna ob vsaki novi izvedbi protokola.

Za odpravo varnostne pomanjkljivosti bi bilo potrebno spremeniti način izračuna skupne skrivnosti, ki si jo delita udeleženec protokola in strežnik. Preprosta rešitev za preprečitev omenjenih napadov bi bila, da bi uporabili trik iz MQV dogovora o ključu (glej protokol 2). V tem primeru bi Anita v prvem krogu izračunala začasni javni ključ $R_{\mathcal{A}} = r_{\mathcal{A}}P$ in skupno skrivnost $K_{\mathcal{A}} = (r_{\mathcal{A}} + \bar{R_{\mathcal{A}}}a)S$, kjer smo z $\bar{R_{\mathcal{A}}}$ označili prvih $L = \lceil(\log_2 p + 1)/2\rceil$ bitov koordinate x točke $R_{\mathcal{A}}$. Podobno bi zaupanja vreden strežnik iz prejetega javnega ključa $R_{\mathcal{A}}$ izračunal skupno skrivnost po enačbi $K_{\mathcal{A}} = s(R_{\mathcal{A}} + \bar{R_{\mathcal{A}}}A)$ v tretjem krogu protokola.

5.4 Limov protokol

Po objavi Jouxovega protokola se je v literaturi pojavilo mnogo tristranskih protokolov za overjen dogovor o ključu, ki naj bi bili odporni na napad vmesne osebe. Eden izmed teh je tudi Shimov protokol [139], ki za overjanje udeležencev uporablja kriptografijo javnih ključev. Kasneje se je izkazalo, da ta protokol ni varen, saj sta Lin in Lin [98] odkrila varnostno pomanjkljivost, na podlagi katere je možno izvesti napad notranje osebe in napad lažnega predstavljanja z razkritim ključem. Hkrati sta predstavila tudi nov protokol, v katerem sta popravila overjanje in pri tem ohranila učinkovitost. Kljub

popravkom so Lim in sodelavci [97] našli novo ranljivost, s katero je možno izvesti napad s ponarejanjem. Tudi oni so zato predstavili nov protokol za odpravo te težave, vendar pa so kasneje ugotovili, da ga lahko razbije notranji napadalec z napadom s ponavljanjem. Predlagani protokol prav tako ne nudi prihodnje varnosti, zato so kmalu objavili njegovo izboljšano verzijo [96].

V tem razdelku bomo pokazali, da je zadnja verzija tristranskega protokola za overjen dogovor o ključu Lima in sodelavcev ranljiva za napad lažnega predstavljanja in posledično za napad vmesne osebe.

Predstavitev protokola

Limov protokol je tristranski overjen dogovor o ključu, s katerim se lahko Anita, Bojan in Cene preko javnega kanala dogovorijo za skupni sejni ključ. Zasnovan je na Jouxovem dogovoru o ključu (glej protokol 3), zato podobno kot druge njegove izpeljave za izračun skupne skrivnosti uporablja učinkovita bilinearna parjenja. Za razliko od slednjega naj bi bil tudi overjen, saj temelji na kriptografiji javnih ključev. Vsak udeleženec protokola mora zato imeti v posesti svoj trajni zasebni ključ, pripadajoči javni ključ pa mora overiti pri certifikatni agenciji in objaviti v digitalnem potrdilu. Protokol je izjemno učinkovit, saj je dogovor o ključu možno izvesti v enem samem krogu, v katerem vsak udeleženec pošlje le eno sporočilo ostalim sodelujočim. Hkrati naj bi bil tudi varen, saj mora napadalec za izračun skupne skrivnosti rešiti bilinearni Diffie-Hellmanov problem (glej def. 2.46) oz. najti varnostno pomanjkljivost v izbrani zgoščevalni funkciji.

Protokol 11 Limov tristranski dogovor o ključu

---

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bosta p in q praštevili, tako da velja $q = 6p - 1 \equiv 2 \pmod 3$, E super singularna eliptična krivulja določena z enačbo $y^2 = x^3 + 1$ nad končnim obsegom $\mathbb{F}_q$ ter P točka reda p, ki določa aditivno grupo $\mathbb{G}_1$. Nadalje naj bo $\mathbb{G}_2$ podgrupa multiplikativne grupe $\mathbb{F}_{q^2}^*$, ki vsebuje vse elemente reda p, preslikava $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ Weilovo bilinearno parjenje in $H : \{0,1\}^* \to \mathbb{Z}_p^*$ kriptografska zgoščevalna funkcija.

(b) Anita izbere svoj trajni zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$ in pripadajoči javni ključ $A = aP$ pošlje certifikatni agenciji v registracijo ter izdajo digitalnega potrdila $\mathrm{Cert}_{\mathcal{A}}$.

2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}, \mathcal{C}: &\quad T_{\mathcal{A}}, W_{\mathcal{A}}, m_{\mathcal{A}}, s_{\mathcal{A}}, t_{\mathcal{A}} \quad (1a) \\ \mathcal{B} \to \mathcal{A}, \mathcal{C}: &\quad T_{\mathcal{B}}, W_{\mathcal{B}}, m_{\mathcal{B}}, s_{\mathcal{B}}, t_{\mathcal{B}} \quad (1b) \\ \mathcal{C} \to \mathcal{A}, \mathcal{B}: &\quad T_{\mathcal{C}}, W_{\mathcal{C}}, m_{\mathcal{C}}, s_{\mathcal{C}}, t_{\mathcal{C}} \quad (1c) \end{aligned}$$

3. *Koraki v protokolu.* Anita, Bojan in Cene se po uspešno izvedeni pripravi, v kateri si izberejo trajne zasebne ključe a, b in c ter svoje javne ključe A, B in C objavijo v digitalnih potrdilih $\mathrm{Cert}_{\mathcal{A}}$, $\mathrm{Cert}_{\mathcal{B}}$ ter $\mathrm{Cert}_{\mathcal{C}}$, dogovorijo za skupni sejni ključ z izvedbo naslednjih korakov.

(a) Anita izbere svoj začasni zasebni ključ $r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna pripadajoča začasna javna ključa $T_{\mathcal{A}} = r_{\mathcal{A}}A$ in $W_{\mathcal{A}} = H(r_{\mathcal{A}})T_{\mathcal{A}}$ ter prebere časovni žig $t_{\mathcal{A}}$ iz zanesljivega časovnega vira. Nato izračuna zgostitvi $m_{\mathcal{A}} = H(ar_{\mathcal{A}})$ in $n_{\mathcal{A}} = H(T_{\mathcal{A}}, W_{\mathcal{A}}, t_{\mathcal{A}})$, vrednost $s_{\mathcal{A}} = (ar_{\mathcal{A}}H(r_{\mathcal{A}}))^{-1}(m_{\mathcal{A}} + a\, n_{\mathcal{A}}) \bmod p$ ter sporočilo (1a) pošlje Bojanu in Cenetu.

Podobno Bojan in Cene izbereta začasna zasebna ključa $r_{\mathcal{B}}$ ter $r_{\mathcal{C}}$, izračunata vrednosti $(T_{\mathcal{B}}, W_{\mathcal{B}}, m_{\mathcal{B}}, s_{\mathcal{B}}, t_{\mathcal{B}})$ in $(T_{\mathcal{C}}, W_{\mathcal{C}}, m_{\mathcal{C}}, s_{\mathcal{C}}, t_{\mathcal{C}})$ ter jih v sporočilih (1b) in (1c) pošljeta vsem udeležencem protokola.

(b) Po izmenjavi sporočil Anita, Bojan in Cene preverijo, ali časovni žigi $t_{\mathcal{A}}$, $t_{\mathcal{B}}$ in $t_{\mathcal{C}}$ ležijo v sprejemljivem časovnem intervalu. Če to ne drži, zaključijo protokol, sicer izračunajo $n_{\mathcal{A}} = H(T_{\mathcal{A}}, W_{\mathcal{A}}, t_{\mathcal{A}})$, $n_{\mathcal{B}} = H(T_{\mathcal{B}}, W_{\mathcal{B}}, t_{\mathcal{B}})$ in $n_{\mathcal{C}} = H(T_{\mathcal{C}}, W_{\mathcal{C}}, t_{\mathcal{C}})$ iz vrednosti, ki so jih prejeli v sporočilih. Anita in Bojan nato preverita, ali je izpolnjena enakost $s_{\mathcal{C}}^{-1}(m_{\mathcal{C}}P + n_{\mathcal{C}}C) = W_{\mathcal{C}}$. Podobno Anita in Cene preverita enakost $s_{\mathcal{B}}^{-1}(m_{\mathcal{B}}P + n_{\mathcal{B}}B) = W_{\mathcal{B}}$ ter Bojan in Cene enakost $s_{\mathcal{A}}^{-1}(m_{\mathcal{A}}P + n_{\mathcal{A}}A) = W_{\mathcal{A}}$. Če so izpolnjene, potem so izmenjana sporočila pristna in medsebojno overjanje udeležencev je uspešno zaključeno. Anita lahko zato izračuna skupno skrivnost po enačbi $\sigma_{\mathcal{A}} = \hat{e}(B + T_{\mathcal{B}}, C + T_{\mathcal{C}})^{a + ar_{\mathcal{A}}}$, Bojan po enačbi $\sigma_{\mathcal{B}} = \hat{e}(A + T_{\mathcal{A}}, C +$

$T_{\mathcal{C}})^{b+br_{\mathcal{B}}}$ in Cene po enačbi $\sigma_{\mathcal{C}} = \hat{e}(A + T_{\mathcal{A}}, B + T_{\mathcal{B}})^{c+cr_C}$. V kolikor so vsi izračuni pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}} = \sigma_{\mathcal{C}}$ in vsi trije lahko iz izmenjanih sporočil $M_{\mathcal{A}} = (T_{\mathcal{A}}, W_{\mathcal{A}}, m_{\mathcal{A}}, s_{\mathcal{A}}, t_{\mathcal{A}})$, $M_{\mathcal{B}} = (T_{\mathcal{B}}, W_{\mathcal{B}}, m_{\mathcal{B}}, s_{\mathcal{B}}, t_{\mathcal{B}})$ in $M_{\mathcal{C}} = (T_{\mathcal{C}}, W_{\mathcal{C}}, m_{\mathcal{C}}, s_{\mathcal{C}}, t_{\mathcal{C}})$ ter javno znanih identitet $\mathrm{ID}_{\mathcal{A}}$, $\mathrm{ID}_{\mathcal{B}}$ in $\mathrm{ID}_{\mathcal{C}}$ izračunajo skupni sejni ključ $K = \mathrm{kdf}(\sigma, M_{\mathcal{A}}, M_{\mathcal{B}}, M_{\mathcal{C}}, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{C}})$.

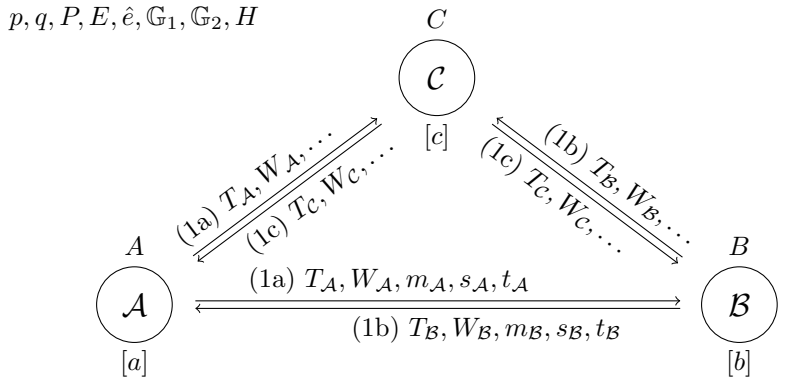
Iz opisa protokola je očitno, da na koncu Anita, Bojan in Cene izračunajo isto skupno skrivnost $\sigma = \hat{e}(P, P)^{(a+ar_{\mathcal{A}})(b+br_{\mathcal{B}})(c+cr_C)}$ in iz nje izpeljejo skupni sejni ključ.

Varnostna analiza

Limov protokol za dogovor o ključu vsebuje dve resni varnostni pomanjkljivosti. Prva se nahaja v izračunu skupne skrivnosti $\sigma = \hat{e}(A + T_{\mathcal{A}}, B + T_{\mathcal{B}})^{c+cr_C}$, iz katere udeleženci izpeljejo sejni ključ. Tu se Anitin trajni javni ključ A in njen začasni javni ključ $T_{\mathcal{A}} = r_A A$ seštejeta, kar lahko privede do resnih varnostnih problemov. Ti se pojavijo, če si Anita za svoj začasni zasebni ključ izbere vrednost $r_{\mathcal{A}} = -1 \bmod p$ in s tem izda oba sogovornika. V tem primeru se skupna skrivnost spremeni v $\sigma = \hat{e}(A - A, B + T_{\mathcal{B}})^{c+cr_C} = 1$ in tako ni več odvisna od zasebnih ter začasnih sejnih ključev.

Preden predstavimo drugo pomanjkljivost, si oglejmo, kako Bojan in Cene preverita pristnost Anitinega sporočila. To storita z izračunom zgostitve $n_{\mathcal{A}} = H(T_{\mathcal{A}}, W_{\mathcal{A}}, t_{\mathcal{A}})$ in s preverjanjem enačbe $s_{\mathcal{A}}^{-1}(m_{\mathcal{A}}P + n_{\mathcal{A}}A) = W_{\mathcal{A}}$, kjer sta vrednosti $T_{\mathcal{A}}$, $W_{\mathcal{A}}$, $m_{\mathcal{A}}$ ter $s_{\mathcal{A}}$ prejela v prvem krogu protokola. Če preverjanje uspe, je poslano sporočilo zagotovo sestavila Anita. In ker je v izračun zgostitve $n_{\mathcal{A}}$ vključen tudi časovni žig sporočila, je sporočilo novo in ni bilo sestavljeno na eni izmed prejšnjih sej. Takšen način medsebojnega overjanja udeležencev na prvi pogled zgleda ustrezen, v resnici pa vsebuje kar nekaj napak, ki jih bomo razkrili v nadaljevanju.

Z vrednostmi $W_{\mathcal{A}}$, $m_{\mathcal{A}}$ in $s_{\mathcal{A}}$ naj bi Anita ostalim udeležencem v protokolu dokazala, da pozna zasebni ključ a. Vendar, če Anita ne sledi protokolu in si izbere vrednosti $W_{\mathcal{A}} = P + A$, $m_{\mathcal{A}} = n_{\mathcal{A}}$ ter $s_{\mathcal{A}} = n_{\mathcal{A}} \bmod p$, potem preprost



(a)

$$
\begin{aligned}
\mathcal{A}: &\quad r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*, \quad T_\mathcal{A} = r_\mathcal{A} A, \quad W_\mathcal{A} = H(r_\mathcal{A}) T_\mathcal{A}, \quad t_\mathcal{A} = \text{čas}() \\
\mathcal{A}: &\quad m_\mathcal{A} = H(a r_\mathcal{A}), \quad n_\mathcal{A} = H(T_\mathcal{A}, W_\mathcal{A}, t_\mathcal{A}) \\
\mathcal{A}: &\quad s_\mathcal{A} = (a r_\mathcal{A} H(r_\mathcal{A}))^{-1} (m_\mathcal{A} + a n_\mathcal{A}) \bmod p \\
\mathcal{A} \to \mathcal{B}, \mathcal{C}: &\quad T_\mathcal{A}, W_\mathcal{A}, m_\mathcal{A}, s_\mathcal{A}, t_\mathcal{A} \qquad (1a) \\
\mathcal{B}: &\quad r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*, \quad T_\mathcal{B} = r_\mathcal{B} B, \quad W_\mathcal{B} = H(r_\mathcal{B}) T_\mathcal{B}, \quad t_\mathcal{B} = \text{čas}() \\
\mathcal{B}: &\quad m_\mathcal{B} = H(b r_\mathcal{B}), \quad n_\mathcal{B} = H(T_\mathcal{B}, W_\mathcal{B}, t_\mathcal{B}) \\
\mathcal{B}: &\quad s_\mathcal{B} = (b r_\mathcal{B} H(r_\mathcal{B}))^{-1} (m_\mathcal{B} + b n_\mathcal{B}) \bmod p \\
\mathcal{B} \to \mathcal{A}, \mathcal{C}: &\quad T_\mathcal{B}, W_\mathcal{B}, m_\mathcal{B}, s_\mathcal{B}, t_\mathcal{B} \qquad (1b) \\
\mathcal{C}: &\quad r_\mathcal{C} \xleftarrow{\$} \mathbb{Z}_p^*, \quad T_\mathcal{C} = r_\mathcal{C} C, \quad W_\mathcal{C} = H(r_\mathcal{C}) T_\mathcal{C}, \quad t_\mathcal{C} = \text{čas}() \\
\mathcal{C}: &\quad m_\mathcal{C} = H(c r_\mathcal{C}), \quad n_\mathcal{C} = H(T_\mathcal{C}, W_\mathcal{C}, t_\mathcal{C}) \\
\mathcal{C}: &\quad s_\mathcal{C} = (c r_\mathcal{C} H(r_\mathcal{C}))^{-1} (m_\mathcal{C} + c n_\mathcal{C}) \bmod p \\
\mathcal{C} \to \mathcal{A}, \mathcal{B}: &\quad T_\mathcal{C}, W_\mathcal{C}, m_\mathcal{C}, s_\mathcal{C}, t_\mathcal{C} \qquad (1c)
\end{aligned}
$$

(b)

$$
\begin{aligned}
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad n_\mathcal{A} = H(T_\mathcal{A}, W_\mathcal{A}, t_\mathcal{A}), \quad M_\mathcal{A} = (T_\mathcal{A}, W_\mathcal{A}, m_\mathcal{A}, s_\mathcal{A}, t_\mathcal{A}) \\
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad n_\mathcal{B} = H(T_\mathcal{B}, W_\mathcal{B}, t_\mathcal{B}), \quad M_\mathcal{B} = (T_\mathcal{B}, W_\mathcal{B}, m_\mathcal{B}, s_\mathcal{B}, t_\mathcal{B}) \\
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad n_\mathcal{C} = H(T_\mathcal{C}, W_\mathcal{C}, t_\mathcal{C}), \quad M_\mathcal{C} = (T_\mathcal{C}, W_\mathcal{C}, m_\mathcal{C}, s_\mathcal{C}, t_\mathcal{C}) \\
\mathcal{A}, \mathcal{B}: &\quad s_\mathcal{C}^{-1} (m_\mathcal{C} P + n_\mathcal{C} C) \stackrel{?}{=} W_\mathcal{C} \\
\mathcal{A}, \mathcal{C}: &\quad s_\mathcal{B}^{-1} (m_\mathcal{B} P + n_\mathcal{B} B) \stackrel{?}{=} W_\mathcal{B} \\
\mathcal{B}, \mathcal{C}: &\quad s_\mathcal{A}^{-1} (m_\mathcal{A} P + n_\mathcal{A} A) \stackrel{?}{=} W_\mathcal{A} \\
\mathcal{A}: &\quad \sigma = \hat{e}(B + T_\mathcal{B}, C + T_\mathcal{C})^{a + a r_\mathcal{A}} \\
\mathcal{B}: &\quad \sigma = \hat{e}(A + T_\mathcal{A}, C + T_\mathcal{C})^{b + b r_\mathcal{B}} \\
\mathcal{C}: &\quad \sigma = \hat{e}(A + T_\mathcal{A}, B + T_\mathcal{B})^{c + c r_\mathcal{C}} \\
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad K = \text{kdf}(\sigma, M_\mathcal{A}, M_\mathcal{B}, M_\mathcal{C}, \text{ID}_\mathcal{A}, \text{ID}_\mathcal{B}, \text{ID}_\mathcal{C})
\end{aligned}
$$

Slika 5.5: Limov tristranski dogovor o ključu

izračun

$$s_{\mathcal{A}}^{-1}(m_{\mathcal{A}}P + n_{\mathcal{A}}A) = n_{\mathcal{A}}^{-1}(n_{\mathcal{A}}P + n_{\mathcal{A}}A) = P + A = W_{\mathcal{A}}$$

razkrije, da za zagotavljanje pristnosti Anita ne rabi poznati svojega zasebnega ključa a. To je druga resna pomanjkljivost protokola, katero lahko napadalec skupaj s prvo izkoristi za pripravo naslednjega napada.

Izrek 5.4. *Limov protokol za tristranski overjen dogovor o ključu ni odporen na napad lažnega predstavljanja. Napadalec Oskar se lahko lažno predstavi kot Anita, Bojan ali Cene ostalima udeležencema v protokolu.*

Dokaz. Naj bo Oskar napadalec, ki se želi Bojanu in Cenetu lažno predstaviti kot Anita. Potem si izbere naključna števila $i, j, k \xleftarrow{\$} \mathbb{Z}_p^*$, prebere časovni žig $t_{\mathcal{O}}$ iz zanesljivega časovnega vira in izračuna vrednosti $T_{\mathcal{O}} = -A + kP$, $W_{\mathcal{O}} = iP + jA$, $n_{\mathcal{O}} = H(T_{\mathcal{O}}, W_{\mathcal{O}}, t_{\mathcal{O}})$, $m_{\mathcal{O}} = ij^{-1}n_{\mathcal{O}} \bmod p$ ter $s_{\mathcal{O}} = j^{-1}n_{\mathcal{O}} \bmod p$. Nato v Anitinem imenu pošlje sporočilo $M_{\mathcal{O}} = (T_{\mathcal{O}}, W_{\mathcal{O}}, m_{\mathcal{O}}, s_{\mathcal{O}}, t_{\mathcal{O}})$ ostalim udeležencem protokola. Bojan in Cene, ki sta tudi vključena v protokol, slednjemu zvesto sledita in pošljeta svoji sporočili $M_{\mathcal{B}} = (T_{\mathcal{B}}, W_{\mathcal{B}}, m_{\mathcal{B}}, s_{\mathcal{B}}, t_{\mathcal{B}})$ in $M_{\mathcal{C}} = (T_{\mathcal{C}}, W_{\mathcal{C}}, m_{\mathcal{C}}, s_{\mathcal{C}}, t_{\mathcal{C}})$. Ko so sporočila izmenjana, se prične medsebojno overjanje. Bojan in Cene overita Oskarjeva sporočila tako, da izračunata $n_{\mathcal{O}} = H(T_{\mathcal{O}}, W_{\mathcal{O}}, t_{\mathcal{O}})$ in preverita enakost $s_{\mathcal{O}}^{-1}(m_{\mathcal{O}}P + n_{\mathcal{O}}A) = W_{\mathcal{O}}$. Ker velja

$$s_{\mathcal{O}}^{-1}m_{\mathcal{O}} \equiv (j^{-1}n_{\mathcal{O}})^{-1}ij^{-1}n_{\mathcal{O}} \equiv i \pmod{p}$$

in

$$s_{\mathcal{O}}^{-1}n_{\mathcal{O}} \equiv (j^{-1}n_{\mathcal{O}})^{-1}n_{\mathcal{O}} \equiv j \pmod{p},$$

se leva stran enakosti poenostavi v $iP + jA$ in je enaka desni strani $W_{\mathcal{O}}$. Bojan in Cene sta tako prepričana, da je sporočilo poslala Anita. Napadalec Oskar mora sedaj izračunati le še skupno skrivnost in sejni ključ, kar lahko stori zaradi enakosti

$$\hat{e}(A + T_{\mathcal{O}}, C + T_{\mathcal{C}})^{b+br_{\mathcal{B}}} = \hat{e}(kP, (c + cr_{\mathcal{C}})P)^{b+br_{\mathcal{B}}} = \hat{e}(P, P)^{(b+br_{\mathcal{B}})(c+cr_{\mathcal{C}})k}$$

z naslednjima izračunoma:

$$\sigma = \hat{e}(B + T_{\mathcal{B}}, C + T_{\mathcal{C}})^k,$$

$$K = \mathrm{kdf}(\sigma, M_{\mathcal{O}}, M_{\mathcal{B}}, M_{\mathcal{C}}, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{C}}).$$

S tem mu je uspelo prepričati Bojana in Ceneta, da si delita sejni ključ z Anito, čeprav si ga v resnici delita z njim.

Podobno se lahko napadalec Oskar lažno predstavi tudi v Bojanovem in Cenetovem imenu. Od tod sledi, da Limov protokol ni odporen na napad lažnega predstavljanja. □

Opisani napad lahko napadalec Oskar izkoristi za izvedbo napada vmesne osebe, v katerem hkrati prične tri seje protokola. V prvi prepriča Anito, da si deli sejni ključ z Bojanom in Cenetom, v drugi prepriča Bojana, da si deli ključ z Anito in Cenetom, v tretji seji pa prepriča Ceneta, da si deli sejni ključ z Anito in Bojanom. Tako bodo vse tri osebe, torej Anita, Bojan in Cene, prepričane, da si delijo isti sejni ključ, medtem ko si ga v resnici vsak izmed njih deli zgolj z Oskarjem. Ta lahko sedaj bere in prenaša sporočila med posameznimi sejami.

Posledica 5.3. *Limov protokol za tristranski dogovor o ključu je ranljiv za napad vmesne osebe.* □

Iz varnostne analize lahko torej zaključimo, da Limov protokol v resnici ni overjen in ga zato ne moremo uporabljati za varen tristranski dogovor o ključu.

5.5 Hölblova IDAK2-1 in IDAK2-2 protokola

Protokola IDAK2-1 in IDAK2-2 za overjen dogovor o ključu na osnovi identitete je leta 2009 predlagal Hölbl [69, 70]. Oba sta dvostranska in izvirata iz Güntherjevega protokola [57] oz. njegovih kasnejših izboljšav. Prvi protokol IDAK2-1 odpravlja varnostno pomanjkljivost Hsiehovega dogovora o ključu [73], saj ta ni varen pred Tsengovim napadom lažnega predstavljanja z razkritim ključem [148]. Isti avtor je kasneje predlagal nov protokol, ki naj bi to ranljivost odpravil [147]. Računsko učinkovitost slednjega je dodatno izboljšal Hölbl v svojem protokolu IDAK2-2.

Varnostno analizo protokolov IDAK2-1 in IDAK2-2 so naredili tudi Chen, Chou in Huang [39] ter zaključili, da sta oba protokola ranljiva za napad

notranje osebe. Tako lahko zlonamerni legitimen uporabnik sistema izračuna glavni tajni ključ generatorja zasebnih ključev kot $s = \alpha\beta$, kjer je α funkcija zasebnega/javnega ključa uporabnika in njegove identitete ter β rešitev določene diofantske enačbe. V resnici opisani napad ne deluje, saj diofantska enačba v splošnem nima enolične rešitve.

Skoraj eno leto po objavi naših rezultatov varnostne analize obeh omenjenih Hölblovih protokolov [112] je svoje napade na njiju razkril tudi Shim [138]. Slednji je našel napad lažnega predstavljanja na oba protokola in napad vmesne osebe le na protokol IDAK2-1. Vsi predstavljeni napadi so enaki našim, ki si jih bomo podrobno ogledali v nadaljevanju.

Predstavitev protokolov

Hölblova protokola IDAK2-1 in IDAK2-2 sta dvostranska protokola za overjen dogovor o ključu, s katerima se lahko Anita in Bojan preko javnega kanala dogovorita za skupni sejni ključ. Ker temeljita na kriptografiji na osnovi identitete, predhodna izmenjava trajnih javnih ključev ali objava njihovih overjenih kopij v digitalnih potrdilih ni potrebna. Za dogovor o ključu je dovolj poznati le splošno znano identifikacijsko informacijo soudeleženca, kot je npr. naslov njegove e-pošte ali telefonska številka. Oba protokola za svoje delovanje uporabljata generator zasebnih ključev, ki mu vsi uporabniki zaupajo. Le-ta ima v lasti glavni tajni ključ, ki je namenjen izdajanju trajnih zasebnih ključev preko varnega kanala. Strukturno sta si protokola zelo podobna, saj oba temeljita na idejah Diffie-Hellmanovega dogovora o ključu in ElGamalovega digitalnega podpisa ter opravljata računanje v multiplikativni grupi $\mathbb{Z}_p^*$. Po trditvah avtorja naj bi bila overjena in imela veliko želenih varnostnih lastnosti, kot so nadzor ključa, prihodnja varnost, odpornost na napad deljenja ključa z neznano osebo in na napad lažnega predstavljanja z razkritim ključem (glej §3.2.2). Nujna varnost naj bi temeljila na Diffie-Hellmanovem problemu in na varnosti uporabljene kriptografske zgoščevalne funkcije. Oba protokola sta tudi izjemno učinkovita, saj si morata Anita in Bojan za dogovor o ključu izmenjati le dve sporočili.

Protokol 12 Hölblov dogovor o ključu IDAK2-1

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bo p veliko praštevilo, $g \in \mathbb{Z}_p^*$ primitivni koren in preslikava $H : \{0,1\}^* \to \mathbb{Z}_p^*$ varna kriptografska zgoščevalna funkcija.

   (b) Generator zasebnih ključev izbere glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna glavni javni ključ $S = g^s \bmod p$ in ga javno objavi.

   (c) Aniti z identiteto $\mathrm{ID}_\mathcal{A}$ generator izbere število $t_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna zgostitev $I_\mathcal{A} = H(\mathrm{ID}_\mathcal{A})$, njeno javno vrednost $T_\mathcal{A} = g^{t_\mathcal{A}} \bmod p$ in trajni zasebni ključ $a = (t_\mathcal{A} I_\mathcal{A} + s T_\mathcal{A}) \bmod (p-1)$. Zadnji dve vrednosti ji preko varnega kanala dostavi v trajno last.

2. *Izmenjava sporočil.*

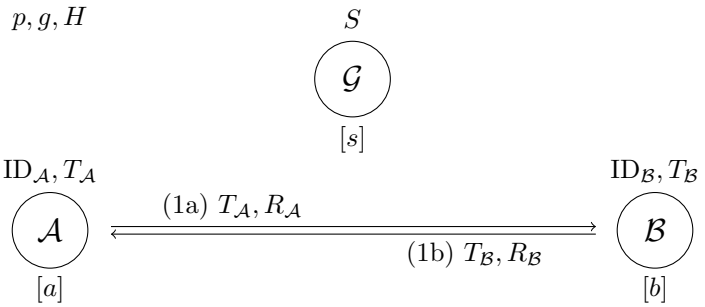
$$\mathcal{A} \to \mathcal{B}: \quad T_\mathcal{A}, R_\mathcal{A} \quad \text{(1a)}$$
$$\mathcal{B} \to \mathcal{A}: \quad T_\mathcal{B}, R_\mathcal{B} \quad \text{(1b)}$$

3. *Dogovor o ključu.* Anita in Bojan se z javno znanima identitetam $\mathrm{ID}_\mathcal{A}$ ter $\mathrm{ID}_\mathcal{B}$ po uspešno izvedeni pripravi, v kateri jima generator zasebnih ključev pošlje javni vrednosti $T_\mathcal{A}$ in $T_\mathcal{B}$ ter zasebna ključa a ter b, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere svoj začasni zasebni ključ $r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_\mathcal{A} = g^{r_\mathcal{A}} \bmod p$ in ga skupaj s svojo javno vrednostjo $T_\mathcal{A}$ pošlje v sporočilu (1a) Bojanu.

   Podobno Bojan izbere začasni zasebni ključ $r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_\mathcal{B} = g^{r_\mathcal{B}} \bmod p$ in ga skupaj z vrednostjo $T_\mathcal{B}$ pošlje v sporočilu (1b) Aniti.

   (b) Po izmenjavi sporočil Anita izračuna zgostitev $I_\mathcal{B} = H(\mathrm{ID}_\mathcal{B})$ in skupno skrivnost $\sigma_\mathcal{A} = (T_\mathcal{B}^{I_\mathcal{B}} S^{T_\mathcal{B}} R_\mathcal{B})^{a+r_\mathcal{A}} \bmod p$. To stori tudi Bojan, tako da izračuna zgostitev $I_\mathcal{A} = H(\mathrm{ID}_\mathcal{A})$ in nato še skrivnost $\sigma_\mathcal{B} = (T_\mathcal{A}^{I_\mathcal{A}} S^{T_\mathcal{A}} R_\mathcal{A})^{b+r_\mathcal{B}} \bmod p$. Če so vsi izračuni pravilni, potem velja $\sigma = \sigma_\mathcal{A} = \sigma_\mathcal{B}$ in Anita ter Bojan lahko izpeljeta skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.



(a)

$$\begin{aligned}
\mathcal{A}: &\quad r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{B}: &\quad r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{A} \to \mathcal{B}: &\quad T_\mathcal{A},\ R_\mathcal{A} = g^{r_\mathcal{A}} \bmod p \qquad (1a) \\
\mathcal{B} \to \mathcal{A}: &\quad T_\mathcal{B},\ R_\mathcal{B} = g^{r_\mathcal{B}} \bmod p \qquad (1b)
\end{aligned}$$

(b)

$$\begin{aligned}
\mathcal{A}, \mathcal{B}: &\quad I_\mathcal{A} = H(\mathrm{ID}_\mathcal{A}),\ I_\mathcal{B} = H(\mathrm{ID}_\mathcal{B}) \\
\mathcal{A}: &\quad \sigma = (T_\mathcal{B}^{I_\mathcal{B}}\, S^{T_\mathcal{B}}\, R_\mathcal{B})^{a+r_\mathcal{A}} \bmod p \\
\mathcal{B}: &\quad \sigma = (T_\mathcal{A}^{I_\mathcal{A}}\, S^{T_\mathcal{A}}\, R_\mathcal{A})^{b+r_\mathcal{B}} \bmod p \\
\mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma)
\end{aligned}$$

Slika 5.6: Hölblov dogovor o ključu IDAK2-1

Pri dogovoru o ključu je Aniti in Bojanu potrebno le prvič poslati javni vrednosti $T_\mathcal{A}$ in $T_\mathcal{B}$, saj se ob ponovni izvedbi protokola ne spremenita. Preprost izračun nam pokaže, da v protokolu oba udeleženca izračunata isto skupno skrivnost $\sigma = g^{(a+r_\mathcal{A})(b+r_\mathcal{B})} \bmod p$ in posledično isti sejni ključ.

Glavna razlika med protokoloma IDAK2-1 in IDAK2-2 se nahaja v pripravi, saj generator na malce drugačen način izračuna trajne zasebne ključe uporabnikov. To sicer ne vpliva na izmenjavo sporočil, vpliva le na izračun skupne skrivnosti pri dogovoru o ključu.

Protokol 13 Hölblov dogovor o ključu IDAK2-2

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bo p veliko praštevilo, $g \in \mathbb{Z}_p^*$ primitivni koren in preslikava $H : \{0,1\}^* \times \mathbb{Z}_p^* \to \mathbb{Z}_p^*$ varna kriptografska zgoščevalna funkcija.

   (b) Generator zasebnih ključev izbere glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna glavni javni ključ $S = g^s \bmod p$ in ga javno objavi.

   (c) Aniti z identiteto $\mathrm{ID}_\mathcal{A}$ generator izbere število $t_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna njeno javno vrednost $T_\mathcal{A} = g^{t_\mathcal{A}} \bmod p$, zgostitev $I_\mathcal{A} = H(\mathrm{ID}_\mathcal{A}, T_\mathcal{A})$ in trajni zasebni ključ $a = (t_\mathcal{A} + sI_\mathcal{A}) \bmod (p-1)$. Zadnji dve vrednosti ji preko varnega kanala dostavi v trajno last.

2. *Izmenjava sporočil.*

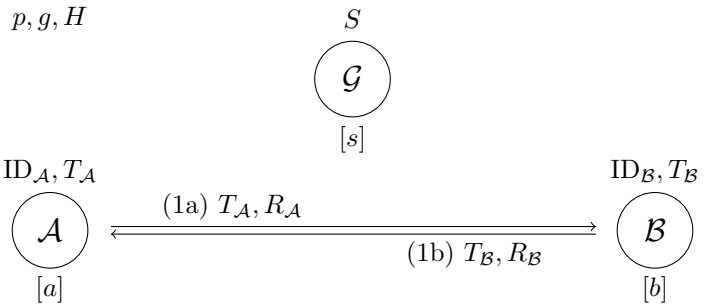
$$\mathcal{A} \to \mathcal{B}: \quad T_\mathcal{A}, R_\mathcal{A} \quad \text{(1a)}$$
$$\mathcal{B} \to \mathcal{A}: \quad T_\mathcal{B}, R_\mathcal{B} \quad \text{(1b)}$$

3. *Dogovor o ključu.* Anita in Bojan se z javno znanima identitetam $\mathrm{ID}_\mathcal{A}$ ter $\mathrm{ID}_\mathcal{B}$ po uspešno izvedeni pripravi, v kateri jima generator zasebnih ključev pošlje javni vrednosti $T_\mathcal{A}$ in $T_\mathcal{B}$ ter zasebna ključa a ter b, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere svoj začasni zasebni ključ $r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_\mathcal{A} = g^{r_\mathcal{A}} \bmod p$ in ga skupaj s svojo javno vrednostjo $T_\mathcal{A}$ pošlje v sporočilu (1a) Bojanu.

   Podobno Bojan izbere začasni zasebni ključ $r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_\mathcal{B} = g^{r_\mathcal{B}} \bmod p$ in ga skupaj z vrednostjo $T_\mathcal{B}$ pošlje v sporočilu (1b) Aniti.

   (b) Po izmenjavi sporočil Anita izračuna zgostitev $I_\mathcal{B} = H(\mathrm{ID}_\mathcal{B}, T_\mathcal{B})$ in skupno skrivnost $\sigma_\mathcal{A} = (T_\mathcal{B}\, S^{I_\mathcal{B}}\, R_\mathcal{B})^{a+r_\mathcal{A}} \bmod p$. To stori tudi Bojan, tako da izračuna zgostitev $I_\mathcal{A} = H(\mathrm{ID}_\mathcal{A}, T_\mathcal{A})$ in nato še skrivnost $\sigma_\mathcal{B} = (T_\mathcal{A}\, S^{I_\mathcal{A}}\, R_\mathcal{A})^{b+r_\mathcal{B}} \bmod p$. Če so vsi izračuni pravilni, potem velja $\sigma = \sigma_\mathcal{A} = \sigma_\mathcal{B}$ in Anita ter Bojan lahko izpeljeta skupni sejni ključ $K = \mathrm{kdf}(\sigma)$.



(a)

$$\begin{aligned}
\mathcal{A}: &\quad r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{B}: &\quad r_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{A} \to \mathcal{B}: &\quad T_{\mathcal{A}},\ R_{\mathcal{A}} = g^{r_{\mathcal{A}}} \bmod p \qquad \text{(1a)} \\
\mathcal{B} \to \mathcal{A}: &\quad T_{\mathcal{B}},\ R_{\mathcal{B}} = g^{r_{\mathcal{B}}} \bmod p \qquad \text{(1b)}
\end{aligned}$$

(b)

$$\begin{aligned}
\mathcal{A}, \mathcal{B}: &\quad I_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}, T_{\mathcal{A}}),\ I_{\mathcal{B}} = H(\mathrm{ID}_{\mathcal{B}}, T_{\mathcal{B}}) \\
\mathcal{A}: &\quad \sigma = (T_{\mathcal{B}}\, S^{I_{\mathcal{B}}}\, R_{\mathcal{B}})^{a+r_{\mathcal{A}}} \bmod p \\
\mathcal{B}: &\quad \sigma = (T_{\mathcal{A}}\, S^{I_{\mathcal{A}}}\, R_{\mathcal{A}})^{b+r_{\mathcal{B}}} \bmod p \\
\mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma)
\end{aligned}$$

Slika 5.7: Hölblov dogovor o ključu IDAK2-2

Tudi v tem protokolu je Aniti in Bojanu potrebno le enkrat poslati vrednosti $T_{\mathcal{A}}$ in $T_{\mathcal{B}}$. In zopet se lahko prepričamo, da oba udeleženca izračunata isto skrivnost $\sigma = g^{(a+r_{\mathcal{A}})(b+r_{\mathcal{B}})} \bmod p$ in zato tudi isti sejni ključ.

Varnostna analiza

Varnost Hölblovih protokolov IDAK2-1 in IDAK2-2 temelji na skupni skrivnosti $\sigma = g^{(a+r_{\mathcal{A}})(b+r_{\mathcal{B}})} \bmod p$, iz katere Anita ter Bojan izpeljeta sejni ključ. Na prvi pogled sicer zgleda, da s skrivnostjo ni nič narobe, saj sta v izračun vključena oba njuna trajna in začasna zasebna ključa, kot je to običajno v navadi pri protokolih za dogovor o ključu. Na žalost pa temu ni tako, saj se varnostna pomanjkljivost nahaja ravno v eksponentu skupne skrivnosti. Tam se Anitin trajni zasebni ključ a in njen naključno izbran začasni ključ $r_{\mathcal{A}}$ seštejeta, kar s seboj prinese cel kup ranljivosti.

Za primer si oglejmo, kaj se v obeh protokolih zgodi s skupno skrivnostjo,

če si Anita izbere število $r_{\mathcal{A}} = -a \bmod (p-1)$. Preprost izračun nam v tem primeru razkrije, da se njena zasebna ključa a in $r_{\mathcal{A}}$ v eksponentu odštejeta, skupna skrivnost pa se spremeni v $\sigma = g^{(a-a)(b+r_{\mathcal{B}})} \bmod p = 1$. To predstavlja veliko ranljivost, katero lahko izkoristi napadalec Oskar. Ta sicer v Anitinem imenu ne more izbrati števila $r_{\mathcal{O}} = -a \bmod (p-1)$, saj ne pozna njenega zasebnega ključa a, pravzaprav pa mu to tudi ni potrebno. Dovolj je le, da se število $-a$ nahaja v eksponentu vrednosti $R_{\mathcal{O}}$. To pa lahko doseže, če $R_{\mathcal{O}}$ na ustrezen način izračuna iz Anitine identitete $\mathrm{ID}_{\mathcal{A}}$, njene javne vrednosti $T_{\mathcal{A}}$ in glavnega javnega ključa S.

Izrek 5.5. *Hölblova protokola IDAK2-1 in IDAK2-2 za dvostranski overjen dogovor o ključu na osnovi identitete nista odporna na napad lažnega predstavljanja. Napadalec Oskar se lahko Aniti lažno predstavi kot Bojan in obratno.*

Dokaz. Naj bo Oskar napadalec, ki je na eni izmed prejšnjih sej protokola, na kateri je sodelovala tudi Anita, prisluškoval izmenjanim sporočilom in si shranil njeno javno vrednost $T_{\mathcal{A}}$. Ko se želi Bojanu lažno predstaviti kot Anita, najprej izbere število $r_{\mathcal{O}} \xleftarrow{\$} \mathbb{Z}_p^*$. Za tem izračuna zgostitev njene identitete $I_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}})$ in vrednost $R_{\mathcal{O}} = (T_{\mathcal{A}}^{I_{\mathcal{A}}} S^{T_{\mathcal{A}}})^{-1} g^{r_{\mathcal{O}}} \bmod p$, če želi za dogovor o ključu uporabiti protokol IDAK2-1, oziroma zgostitev $I_{\mathcal{A}} = H(\mathrm{ID}_A, T_{\mathcal{A}})$ in vrednost $R_{\mathcal{O}} = (T_{\mathcal{A}} S^{I_{\mathcal{A}}})^{-1} g^{r_{\mathcal{O}}} \bmod p$, če se želi za sejni ključ dogovoriti s protokolom IDAK2-2. Ni se težko prepričati, da si v obeh primerih Oskar za začasni javni ključ izbere vrednost $R_{\mathcal{O}} = g^{-a+r_{\mathcal{O}}} \bmod p$. Slednjo v Anitinem imenu skupaj z njeno javno vrednostjo $T_{\mathcal{A}}$ pošlje Bojanu, kot je to opredeljeno v definiciji protokola. Ko Bojan prejme sporočilo je prepričan, da se želi Anita z njim dogovoriti za sejni ključ. Zato sledi korakom protokola in izračuna svoj začasni javni ključ $R_{\mathcal{B}}$ ter pošlje sporočilo $(T_{\mathcal{B}}, R_{\mathcal{B}})$ nazaj Oskarju. Hkrati izpelje tudi sejni ključ iz skupne skrivnosti $\sigma = (T_{\mathcal{A}}^{I_{\mathcal{A}}} S^{T_{\mathcal{A}}} R_{\mathcal{O}})^{b+r_{\mathcal{B}}} \bmod p$, če nastopa v protokolu IDAK2-1, oz. iz skrivnost $\sigma = (T_{\mathcal{A}} S^{I_{\mathcal{A}}} R_{\mathcal{O}})^{b+r_{\mathcal{B}}} \bmod p$, če nastopa v protokolu IDAK2-2. Ker je v obeh primerih skupna skrivnost enaka $\sigma = g^{r_{\mathcal{O}}(b+r_{\mathcal{B}})} \bmod p$, jo lahko izračuna tudi Oskar po enačbi $\sigma = (T_{\mathcal{B}}^{I_{\mathcal{B}}} S^{T_{\mathcal{B}}} R_{\mathcal{B}})^{r_{\mathcal{O}}} \bmod p$ oz. $\sigma = (T_{\mathcal{B}} S^{I_{\mathcal{B}}} R_{\mathcal{B}})^{R_{\mathcal{O}}} \bmod p$ in iz nje izpelje isti sejni ključ. S tem mu je uspelo prepričati Bojana, da si deli sejni ključ z Anito, čeprav si ga v resnici deli z njim.

Protokola IDAK2-1 in IDAK2-2 sta simetrična, saj imata Anita in Bojan v njih enako vlogo. Napadalec Oskar lahko zato enak napad izvede tudi v Bojanovem imenu. Od tod sledi, da protokola nista odporna na napad lažnega predstavljanja. □

Ker se lahko napadalec Oskar v obeh protokolih lažno predstavlja v imenu pobudnika in naslovnika, lahko iz zgornjega izreka izpeljemo tudi naslednji rezultat.

Posledica 5.4. *Hölblova protokola IDAK2-1 in IDAK2-2 za dvostranski overjen dogovor o ključu na osnovi identitete nista odporna na napad vmesne osebe.* □

Iz varnostne analize lahko zaključimo, da protokola IDAK2-1 in IDAK2-2 nista overjena. Poleg tega nimata vseh varnostnih lastnosti, kot trdi avtor, zato ju ne moremo uporabljati za varen dogovor o ključu.

Predlog izboljšave

Za odpravo varnostnih pomanjkljivosti obeh protokolov bi bilo potrebno v eksponentu skupne skrivnosti drugače kombinirati trajna in začasna zasebna ključa Anite ter Bojana in nato ustrezno popraviti vse vmesne izračune. Nekaj teh kombinacij so preučili že avtorji družine MTI protokolov [99], dodatne, kot so npr. $ab + r_{\mathcal{A}} r_{\mathcal{B}}$, $ar_{\mathcal{B}} + br_{\mathcal{A}}$ in $ab + ar_{\mathcal{B}} + br_{\mathcal{A}}$, pa bi bilo potrebno podrobno preučiti ter nato preveriti učinkovitost nastalega protokola.

Za konec lahko omenimo še, da nekatere spremembe vodijo nazaj k osnovnim protokolom za dogovor o ključu, katerih pomanjkljivosti naj bi protokola IDAK2-1 in IDAK2-2 odpravila. Na primer, kombinacijo $ab + r_{\mathcal{A}} r_{\mathcal{B}}$ uporablja Hsiehov dogovor o ključu [73], katerega varnost naj bi izboljšal Hölbl v protokolu IDAK2-1. Zato je morda bolje razmišljati o povsem novih pristopih za čimbolj učinkovito in varno sestavo protokolov za overjen dogovor o ključu na osnovi identitete.

5.6 Hölblov IDAK2-P1 protokol

Po objavi Smartovega protokola leta 2002 [141] so se v literaturi množično začeli pojavljati novi protokoli za dvostranski dogovor o ključu na osnovi identitete. Eden izmed njih je bil tudi protokol avtorjev Choie, Jeong in Lee [40], katerega pomanjkljivosti je našel Shim [139]. Za odpravo le-teh je Hölbl predlagal majhne spremembe in objavil protokol IDAK2-P1 [69], ki naj bi obdržal vse želene lastnosti. V tem razdelku bomo pokazali, da temu ni tako, saj lahko napadalec izračuna zasebni ključ vsakega uporabnika.

Predstavitev protokola

Hölblov protokol IDAK2-P1 je dvostranski overjen dogovor o ključu na osnovi identitete, s katerim se lahko Anita in Bojan preko javnega kanala dogovorita za sejni ključ. Za dogovor o ključu morata imeti v posesti trajni zasebni ključ, ki jima ga z uporabo glavnega tajnega ključa izračuna generator zasebnih ključev in preko varnega kanala dostavi v trajno last. Hkrati morata poznati tudi identifikacijsko informacijo soudeleženca, ki pa naj bi bila javno znana vsem uporabnikom. Protokol lahko uvrstimo med overjene različice Diffie-Hellmanovega dogovora o ključu saj udeleženca izmenjane začasne javne ključe medsebojno overita z uporabo učinkovitih bilinearnih parjenj. Varnost protokola naj bi zato temeljila na vmesnem Diffie-Hellmanovem problemu (glej def. 2.47) in na varnosti uporabljene zgoščevalne funkcije. Protokol ima tudi majhno komunikacijsko zahtevnost, saj je možno izmenjavo sporočil opraviti v enem samem krogu.

Protokol 14 Hölblov dogovor o ključu IDAK2-P1

---

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bosta $\mathbb{G}_1$ in $\mathbb{G}_2$ grupi, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ bilinearno parjenje, $P \in \mathbb{G}_1$ element praštevilskega reda p in $H_1 : \{0,1\}^* \to \mathbb{G}_1$ ter $H_2 : \mathbb{G}_1 \to \mathbb{Z}_p^*$ kriptografski zgoščevalni funkciji.

   (b) Generator zasebnih ključev izbere glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna glavni javni ključ $S = sP$ in ga javno objavi.

(c) Aniti z identiteto $\mathrm{ID}_{\mathcal{A}}$ generator zasebnih ključev izračuna zgostitev $Q_{\mathcal{A}} = H_1(\mathrm{ID}_{\mathcal{A}})$ in iz nje trajni zasebni ključ $S_{\mathcal{A}} = sQ_{\mathcal{A}}$, ki ji ga preko varnega kanala dostavi v trajno last.

2. *Izmenjava sporočil.*

$$\mathcal{A} \to \mathcal{B}: \quad R_{\mathcal{A}}, V_{\mathcal{A}} \quad \text{(1a)}$$
$$\mathcal{B} \to \mathcal{A}: \quad R_{\mathcal{B}}, V_{\mathcal{B}} \quad \text{(1b)}$$

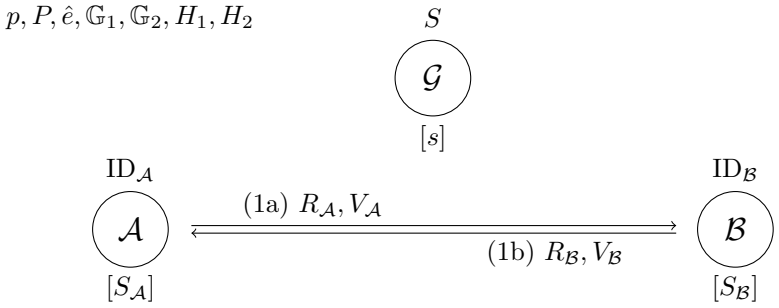
3. *Dogovor o ključu.* Anita in Bojan se z javno znanima identitetama $\mathrm{ID}_{\mathcal{A}}$ ter $\mathrm{ID}_{\mathcal{B}}$ po uspešno izvedeni pripravi, v kateri jima generator zasebnih ključev pošlje trajna zasebna ključa $S_{\mathcal{A}}$ ter $S_{\mathcal{B}}$, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere začasni zasebni ključ $r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasna javna ključa $R_{\mathcal{A}} = r_{\mathcal{A}}P$ in $V_{\mathcal{A}} = H_2(R_{\mathcal{A}})S_{\mathcal{A}}$ ter ju v sporočilu (1a) pošlje Bojanu.

   Podobno Bojan izbere začasni zasebni ključ $r_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^*$ in začasna javna ključa $R_{\mathcal{B}} = r_{\mathcal{B}}P$ ter $V_{\mathcal{B}} = H_2(R_{\mathcal{B}})S_{\mathcal{B}}$ pošlje v sporočilu (1b) Aniti.

   (b) Po izmenjavi sporočil Anita izračuna zgostitev $Q_{\mathcal{B}} = H_1(\mathrm{ID}_{\mathcal{B}})$ in preveri, ali je enakost $\hat{e}(Q_{\mathcal{A}}, V_{\mathcal{B}}) = \hat{e}(S_{\mathcal{A}}, Q_{\mathcal{B}})^{H_2(R_{\mathcal{B}})}$ izpolnjena. Podobno stori tudi Bojan, le da on izračuna $Q_{\mathcal{A}} = H_1(\mathrm{ID}_{\mathcal{A}})$ in preveri enakost $\hat{e}(Q_{\mathcal{B}}, V_{\mathcal{A}}) = \hat{e}(S_{\mathcal{B}}, Q_{\mathcal{A}})^{H_2(R_{\mathcal{A}})}$. Če sta obe enakosti izpolnjeni, potem sta poslani sporočili pristni in medsebojno overjanje je s tem uspešno zaključeno. Anita nato izračuna skupno skrivnost σ po enačbi $\sigma_{\mathcal{A}} = r_{\mathcal{A}}R_{\mathcal{B}}$, Bojan pa po enačbi $\sigma_{\mathcal{B}} = r_{\mathcal{B}}R_{\mathcal{A}}$. V kolikor so vsi izračuni pravilni, potem velja enakost $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in Anita ter Bojan lahko izračunata skupni sejni ključ $K = \mathrm{kdf}(\sigma, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}})$.

Ni se težko prepričati, da na koncu protokola Anita in Bojan izračunata isto skupno skrivnost $\sigma = r_{\mathcal{A}}r_{\mathcal{B}}P$ in iz nje izpeljeta skupni sejni ključ.



(a)

$$
\begin{aligned}
\mathcal{A}: &\quad r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{B}: &\quad r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{A} \to \mathcal{B}: &\quad R_\mathcal{A} = r_\mathcal{A} P,\ V_\mathcal{A} = H_2(R_\mathcal{A}) S_\mathcal{A} && \text{(1a)} \\
\mathcal{B} \to \mathcal{A}: &\quad R_\mathcal{B} = r_\mathcal{B} P,\ V_\mathcal{B} = H_2(R_\mathcal{B}) S_\mathcal{B} && \text{(1b)}
\end{aligned}
$$

(b)

$$
\begin{aligned}
\mathcal{A}, \mathcal{B}: &\quad Q_\mathcal{A} = H_1(\mathrm{ID}_\mathcal{A}), \quad Q_\mathcal{B} = H_1(\mathrm{ID}_\mathcal{B}) \\
\mathcal{A}: &\quad \hat{e}(Q_\mathcal{A}, V_\mathcal{B}) \stackrel{?}{=} \hat{e}(S_\mathcal{A}, Q_\mathcal{B})^{H_2(R_\mathcal{B})} \\
\mathcal{B}: &\quad \hat{e}(Q_\mathcal{B}, V_\mathcal{A}) \stackrel{?}{=} \hat{e}(S_\mathcal{B}, Q_\mathcal{A})^{H_2(R_\mathcal{A})} \\
\mathcal{A}: &\quad \sigma = r_\mathcal{A} R_\mathcal{B} \\
\mathcal{B}: &\quad \sigma = r_\mathcal{B} R_\mathcal{A} \\
\mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma, \mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{B})
\end{aligned}
$$

Slika 5.8: Hölblov dvostranski dogovor o ključu IDAK2-P1

Varnostna analiza

Varnostna pomanjkljivost opisanega protokola se skriva v prvem koraku dogovora o ključu. V njem Anita svoj trajni zasebni ključ $S_{\mathcal{A}}$ množi z zgostitvijo $H_2(R_{\mathcal{A}})$ in nato dobljeno vrednost $V_{\mathcal{A}} = H_2(R_{\mathcal{A}})S_{\mathcal{A}}$ skupaj z $R_{\mathcal{A}}$ pošlje preko javnega kanala. Poslani vrednosti lahko zato izkoristi prisluškovalec Oskar za izračun Anitinega zasebnega ključa.

Izrek 5.6. *Hölblov protokol IDAK2-P1 za dvostranski overjen dogovor o ključu na osnovi identitete ni varen v prisotnosti pasivnega napadalca. Slednji lahko iz izmenjanih sporočil izračuna zasebna ključa Anite in Bojana.*

Dokaz. Naj bo Oskar pasivni napadalec, ki v protokolu prisluškuje pogovoru med Anito in Bojanom. Če želi razkriti njuna trajna zasebna ključa, potem najprej iz Anitinega sporočila prebere vrednosti $R_{\mathcal{A}}$ in $V_{\mathcal{A}}$. Nato izračuna zgostitev $H_2(R_{\mathcal{A}}) \in \mathbb{Z}_p^*$ in z Evklidovim algoritmom še njen inverz. Ta vedno obstaja, saj je $\mathbb{Z}_p^*$ grupa in so vsi njeni elementi obrnljivi. Anitin zasebni ključ lahko sedaj izračuna z enačbo

$$S_{\mathcal{A}} = H_2(R_{\mathcal{A}})^{-1}V_{\mathcal{A}} = H_2(R_{\mathcal{A}})^{-1}H_2(R_{\mathcal{A}})S_{\mathcal{A}}.$$

Podobno lahko Oskar izračuna tudi Bojanov zasebni ključ, če iz njegovih poslanih vrednosti $R_{\mathcal{B}}$ in $V_{\mathcal{B}}$ izračuna $S_{\mathcal{B}} = H_2(R_{\mathcal{B}})^{-1}V_{\mathcal{B}}$. Z razkritjem zasebnih ključev obeh udeležencev je protokol očitno popolnoma razbit in zato ni varen. □

Zgornji izrek pravi, da protokol IDAK2-P1 ni varen niti v prisotnosti pasivnega napadalca. Zato lahko zaključimo, da ne nudi popolnoma nobene varnosti in ga je potrebno temeljito spremeniti.

5.7 Hölblov IDAK2-P2 protokol

V znanstvenem članku [72] in v svoji doktorski disertaciji [69] je Hölbl predstavil učinkovit dvostranski protokol za overjen dogovor o ključu na osnovi identitete IDAK2-P2, ki ga je sestavil s pomočjo bilinearnega parjenja in z uporabo Hessove sheme za digitalni podpis [68]. Hkrati je podal tudi hevristične razloge, zakaj naj bi protokol imel vse želene varnostne lastnosti,

naredil analizo njegove učinkovitosti in primerjavo s sorodnimi protokoli. V nadaljevanju bomo razkrili, da predlagani protokol v resnici nima vseh želenih lastnosti, saj ni odporen na napad deljenja ključa z neznano osebo.

Predstavitev protokola

Protokol IDAK2-P2 je dvostranski overjen dogovor o ključu, s katerim se lahko Anita in Bojan preko javnega kanala dogovorita za sejni ključ in medsebojno overita. Ker uporablja overjanje na osnovi identitete, v njem nastopa tudi generator zasebnih ključev, ki uporabnikom preko varnega kanala izdaja trajne zasebne ključe. Podobno kot protokol IDAK2-P1 je zasnovan na Diffie-Hellmanovem dogovoru o ključu, kateremu je dodano overjanje sporočil z digitalnim podpisom začasnih javnih ključev. Ker protokol uporablja učinkovita bilinearna parjenja, naj bi njegova varnost temeljila na vmesnem Diffie-Hellmanovem problemu. Tudi ta Hölblov protokol ima zelo majhno komunikacijsko zahtevnost, saj je dogovor možno opraviti z izmenjavo dveh sporočil v enem samem krogu.

Protokol 15 Hölblov dogovor o ključu IDAK2-P2

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   Ta faza je podobna prvi fazi protokola 14, le da je zgoščevalna funkcija H_2 definirana kot preslikava $H_2 : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{Z}_p^*$.

2. *Izmenjava sporočil.*

$$\mathcal{A} \to \mathcal{B}: \quad R_{\mathcal{A}}, v_{\mathcal{A}}, U_{\mathcal{A}} \quad \text{(1a)}$$
$$\mathcal{B} \to \mathcal{A}: \quad R_{\mathcal{B}}, v_{\mathcal{B}}, U_{\mathcal{B}} \quad \text{(1b)}$$

3. *Dogovor o ključu.* Anita in Bojan se z javno znanima identitetama $\mathrm{ID}_{\mathcal{A}}$ ter $\mathrm{ID}_{\mathcal{B}}$, po uspešno izvedeni pripravi, v kateri jima generator zasebnih ključev pošlje trajna zasebna ključa $S_{\mathcal{A}}$ ter $S_{\mathcal{B}}$, dogovorita za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere začasni zasebni ključ $r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasni javni ključ $R_{\mathcal{A}} = r_{\mathcal{A}} P$ in ga s trajnim zasebnim ključem $S_{\mathcal{A}}$ digitalno

podpiše tako, da izračuna $t_{\mathcal{A}} = \hat{e}(S_{\mathcal{A}}, P)^{r_{\mathcal{A}}}$, $v_{\mathcal{A}} = H_2(R_{\mathcal{A}}, t_{\mathcal{A}})$ in $U_{\mathcal{A}} = (v_{\mathcal{A}} + r_{\mathcal{A}})S_{\mathcal{A}}$. Nato ga skupaj s podpisom $(v_{\mathcal{A}}, U_{\mathcal{A}})$ pošlje Bojanu v sporočilu (1a).

Podobno Bojan izbere začasni zasebni ključ $r_{\mathcal{B}} \overset{\$}{\leftarrow} \mathbb{Z}_p^*$, izračuna javni ključ $R_{\mathcal{B}} = r_{\mathcal{B}}P$, vrednosti $t_{\mathcal{B}} = \hat{e}(S_{\mathcal{B}}, P)^{r_{\mathcal{B}}}$, $v_{\mathcal{B}} = H_2(R_{\mathcal{B}}, t_{\mathcal{B}})$ in $U_{\mathcal{B}} = (v_{\mathcal{B}} + r_{\mathcal{B}})S_{\mathcal{B}}$ ter pošlje sporočilo (1b) Aniti.

(b) Po izmenjavi sporočil Anita izračuna zgostitev $Q_{\mathcal{B}} = H_1(\mathrm{ID}_{\mathcal{B}})$ in preveri Bojanov podpis z izračunom $t'_{\mathcal{B}} = \hat{e}(U_{\mathcal{B}}, P)\,\hat{e}(Q_{\mathcal{B}}, -S)^{v_{\mathcal{B}}}$ ter preverjanjem enakosti $v_{\mathcal{B}} = H_2(R_{\mathcal{B}}, t'_{\mathcal{B}})$. Hkrati tudi Bojan preveri Anitin digitalni podpis z izračunom zgostitve $Q_{\mathcal{A}} = H_1(\mathrm{ID}_{\mathcal{A}})$ in vrednosti $t'_{\mathcal{A}} = \hat{e}(U_{\mathcal{A}}, P)\,\hat{e}(Q_{\mathcal{A}}, -S)^{v_{\mathcal{A}}}$ ter s preverjanjem enakosti $v_{\mathcal{A}} = H_2(R_{\mathcal{A}}, t'_{\mathcal{A}})$. Če digitalna podpisa nista veljavna, prekineta protokol, sicer nadaljujeta z njegovim izvajanjem in izračunata skupno skrivnost. Anita to stori z izračunom $\sigma_{\mathcal{A}} = r_{\mathcal{A}}R_{\mathcal{B}}$ in Bojan z izračunom $\sigma_{\mathcal{B}} = r_{\mathcal{B}}R_{\mathcal{A}}$. V kolikor so vsi izračuni pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}}$ in oba udeleženca lahko izračunata sejni ključ $K = \mathrm{kdf}(\sigma)$.

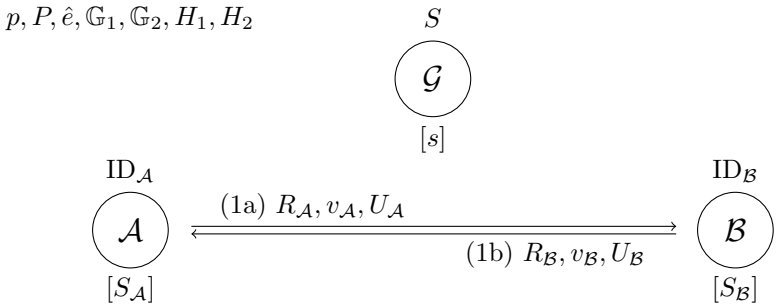
Na koncu protokola Anita in Bojan iz skupne skrivnosti $\sigma = r_{\mathcal{A}}r_{\mathcal{B}}P$ izračunata skupni sejni ključ ter ga sprejmeta le, če preverjanje podpisov v prejšnjih korakih uspe. Iz enačbe

$$t'_{\mathcal{A}} = \hat{e}(U_{\mathcal{A}}, P)\,\hat{e}(Q_{\mathcal{A}}, -S)^{v_{\mathcal{A}}} = \hat{e}(Q_{\mathcal{A}}, P)^{s(v_{\mathcal{A}}+r_{\mathcal{A}})}\,\hat{e}(Q_{\mathcal{A}}, P)^{-sv_{\mathcal{A}}} = \hat{e}(Q_{\mathcal{A}}, P)^{sr_{\mathcal{A}}}$$

sledi, da je izračunana vrednost $t'_{\mathcal{A}}$ enaka prejeti $t_{\mathcal{A}}$. Enakost $v_{\mathcal{A}} = H_2(R_{\mathcal{A}}, t'_{\mathcal{A}})$ je torej v celoti izpolnjena in Bojan Anitin podpis sprejme kot veljaven. Podobno velja tudi za Anito, zato oba udeleženca sprejmeta podpis in posledično tudi skupni sejni ključ.

Varnostna analiza

Varnostna pomanjkljivost Hölblovega protokola IDAK2-P2 se skriva v napačni uporabi digitalnega podpisa $(v_{\mathcal{A}}, U_{\mathcal{A}})$. S slednjim naj bi Anita poslala zagotovilo Bojanu, da edina pozna začasni zasebni ključ $r_{\mathcal{A}}$ in da je ona izračunala



(a)

$$
\begin{aligned}
\mathcal{A}: &\quad r_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*,\ t_\mathcal{A} = \hat{e}(S_\mathcal{A}, P)^{r_\mathcal{A}} \\
\mathcal{B}: &\quad r_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*,\ t_\mathcal{B} = \hat{e}(S_\mathcal{B}, P)^{r_\mathcal{B}} \\
\mathcal{A} \to \mathcal{B}: &\quad R_\mathcal{A} = r_\mathcal{A} P,\ v_\mathcal{A} = H_2(R_\mathcal{A}, t_\mathcal{A}),\ U_\mathcal{A} = (v_\mathcal{A} + r_\mathcal{A}) S_\mathcal{A} \qquad \text{(1a)} \\
\mathcal{B} \to \mathcal{A}: &\quad R_\mathcal{B} = r_\mathcal{B} P,\ v_\mathcal{B} = H_2(R_\mathcal{B}, t_\mathcal{B}),\ U_\mathcal{B} = (v_\mathcal{B} + r_\mathcal{B}) S_\mathcal{B} \qquad \text{(1b)}
\end{aligned}
$$

(b)

$$
\begin{aligned}
\mathcal{A}, \mathcal{B}: &\quad Q_\mathcal{A} = H_1(\mathrm{ID}_\mathcal{A}),\ Q_\mathcal{B} = H_1(\mathrm{ID}_\mathcal{B}) \\
\mathcal{A}: &\quad t'_\mathcal{B} = \hat{e}(U_\mathcal{B}, P)\, \hat{e}(Q_\mathcal{B}, -S)^{v_\mathcal{B}} \\
\mathcal{A}: &\quad v_\mathcal{B} \stackrel{?}{=} H_2(R_\mathcal{B}, t'_\mathcal{B}) \\
\mathcal{B}: &\quad t'_\mathcal{A} = \hat{e}(U_\mathcal{A}, P)\, \hat{e}(Q_\mathcal{A}, -S)^{v_\mathcal{A}} \\
\mathcal{B}: &\quad v_\mathcal{A} \stackrel{?}{=} H_2(R_\mathcal{A}, t'_\mathcal{A}) \\
\mathcal{A}: &\quad \sigma = r_\mathcal{A} R_\mathcal{B} \\
\mathcal{B}: &\quad \sigma = r_\mathcal{B} R_\mathcal{A} \\
\mathcal{A}, \mathcal{B}: &\quad K = \mathrm{kdf}(\sigma)
\end{aligned}
$$

Slika 5.9: Hölblov dogovor o ključu IDAK2-P2

vrednost $R_{\mathcal{A}} = r_{\mathcal{A}}P$. Vendar, ker $r_{\mathcal{A}}$ v digitalnem podpisu nastopa zgolj kot naključna vrednost, je možno $R_{\mathcal{A}}$ podpisati, ne da bi poznali začasni ključ $r_{\mathcal{A}}$.

Izrek 5.7. *Hölblov protokol IDAK2-P2 za dvostranski overjen dogovor o ključu na osnovi identitete ni odporen na napad deljenja ključa z neznano osebo. Notranji napadalec Oskar lahko prepriča Bojana, da se pogovarja z njim, čeprav v resnici komunicira z Anito, in obratno.*

Dokaz. Najprej bomo pokazali, da za pripravo digitalnega podpisa vrednosti $R_{\mathcal{A}}$ ne potrebujemo zasebnega ključa $r_{\mathcal{A}}$. To bomo storili tako, da bomo začetni protokol malce spremenili. Predstavljamo si, kaj se zgodi, če Anitin začasni ključ $r_{\mathcal{A}}$ v vrednostih $t_{\mathcal{A}}$ in $U_{\mathcal{A}}$ zamenjamo z $s^{-1} \bmod p$. V tem primeru dobimo nov protokol, v katerem Anita za sestavo podpisa opravi naslednje izračune:

$$t_{\mathcal{A}} = \hat{e}(Q_{\mathcal{A}}, P), \quad v_{\mathcal{A}} = H_2(R_{\mathcal{A}}, t_{\mathcal{A}}), \quad U_{\mathcal{A}} = v_{\mathcal{A}}S_{\mathcal{A}} + Q_{\mathcal{A}}.$$

Ni težko preveriti, da kljub spremembam Bojan sprejme Anitin podpis kot veljaven in da se protokol uspešno zaključi. Poleg tega lahko hitro opazimo, da v izračunih ne nastopa več začasni ključ $r_{\mathcal{A}}$ in je zato digitalni podpis odvisen le še od Anitinega zasebnega ključa $S_{\mathcal{A}}$. Od tod sledi, da lahko vsakdo sestavi digitalni podpis vrednosti $R_{\mathcal{A}}$. To lahko izkoristi napadalec za pripravo napada deljenja ključa z neznano osebo.

Naj bo Oskar zlonamerni notranji napadalec, ki ima v lasti trajni zasebni ključ $S_{\mathcal{O}}$. Če se želi Anita dogovoriti za sejni ključ z Bojanom, potem v prvem krogu protokola pošlje sporočilo z vrednostmi $R_{\mathcal{A}}$, $v_{\mathcal{A}}$ in $U_{\mathcal{A}}$. Te Oskar prestreže in s svojim zasebnim ključem $S_{\mathcal{O}}$ sestavi lasten podpis $(v_{\mathcal{O}}, U_{\mathcal{O}})$ vrednosti $R_{\mathcal{A}}$, tako da izračuna $t_{\mathcal{O}} = \hat{e}(Q_{\mathcal{O}}, P)$, $v_{\mathcal{O}} = H_2(R_{\mathcal{A}}, t_{\mathcal{O}})$ in $U_{\mathcal{O}} = v_{\mathcal{O}}S_{\mathcal{O}} + Q_{\mathcal{O}}$. Nato pošlje sporočilo $(R_{\mathcal{A}}, v_{\mathcal{O}}, U_{\mathcal{O}})$ Bojanu kot legitimen uporabnik Oskar. Ko ta prejme sporočilo, misli, da se želi Oskar dogovoriti za sejni ključ. Bojan zato sledi protokolu in pošlje vrednost $R_{\mathcal{B}}$ ter ustrezen digitalni podpis $(v_{\mathcal{B}}, U_{\mathcal{B}})$ nazaj Oskarju, ki prejeto sporočilo posreduje Aniti. Ker sta digitalna podpisa vrednosti $R_{\mathcal{A}}$ in $R_{\mathcal{B}}$ veljavna, Anita in Bojan izračunata isto skupno skrivnost $\sigma = r_{\mathcal{A}}r_{\mathcal{B}}P$. Slednjo lahko izračunata oba, saj Anita pozna svoj zasebni ključ $r_{\mathcal{A}}$ in Bojan pozna svojega $r_{\mathcal{B}}$. Iz skupne skrivnosti nato izpeljeta sejni ključ in protokol se uspešno zaključi.

S tem je napadalec Oskar uspešno izvedel napad deljenja ključa z neznano osebo, saj je na koncu protokola Anita prepričana, da si deli sejni ključ z Bojanom, medtem ko ta napačno verjame, da si ga deli z Oskarjem. □

Predlagani protokol torej ni varen pred aktivnim notranjim napadalcem, zato je potrebno protokol ustrezno spremeniti oz. ga uporabljati le v okoljih, kjer takšni napadi ne pridejo v poštev.

Predlog izboljšave

Za vse protokole, ki so ranljivi za napad deljenja ključa z neznano osebo, obstaja preprosta rešitev, s katero lahko napad preprečimo. Vse, kar je potrebno storiti, je, da se pri izpeljavi sejnega ključa upoštevata tudi identiteti obeh udeležencev. V protokolu IDAK2-P2 bi morali tako sejni ključ izračunati po formuli $K = \mathrm{kdf}(\sigma, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}})$. S takšnim pristopom naš napad ne bi uspel, saj Bojan in Anita ne bi izračunala istega sejnega ključa. Bojan bi bil namreč prepričan, da se pogovarja z Oskarjem, zato bi v izpeljavo ključa vključil njegovo identiteto $\mathrm{ID}_{\mathcal{O}}$ in ne Anitine $\mathrm{ID}_{\mathcal{A}}$.

S prvo rešitvijo lahko preprečimo predlagan napad, ne vemo pa, ali lahko Oskar omenjene pomanjkljivosti izkoristi kako drugače. Za odpravo slednjih bi morala Anita na drugačen način Bojanu dokazati, da pozna začasni zasebni ključ $r_{\mathcal{A}}$. Tega ni možno storiti z uporabo Hessove sheme za digitalni podpis, saj v njej $r_{\mathcal{A}}$ nastopa kot naključna vrednost in nima nobene povezave s sporočilom $R_{\mathcal{A}}$, ki ga podpisujemo. Zato je potrebno uporabiti drugačne pristope, kot so npr. dokazi brez razkritja znanja, ki pa lahko drastično vplivajo na učinkovitost protokola.

Po našem mnenju je v tem primeru najboljša rešitev, da Anita poleg digitalnega podpisa Bojanu pošlje še vrednost $R'_{\mathcal{A}} = r_{\mathcal{A}}^{-1}P$. Slednji ob prejetju sporočila preveri veljavnost enačbe $\hat{e}(R_{\mathcal{A}}, R'_{\mathcal{A}}) = \hat{e}(P, P)$ in tako dobi zagotovilo, da Anita v resnici pozna zasebni ključ $r_{\mathcal{A}}$. Takšna rešitev na prvi pogled zgleda zelo preprosto, vendar pa ima tudi svoje posledice. Tako mora Anita dodatno izračunati en inverz v grupi $\mathbb{Z}_p^*$, eno množenje s skalarjem v grupi $\mathbb{G}_1$ in v poslano sporočilo vključiti še en element grupe $\mathbb{G}_1$. Na drugi strani, pa mora Bojan pri preverjanju enačbe dodatno izračunati eno bilinearno parjenje. Vse te operacije močno vplivajo na računsko zahtevnost protokola, zato

bi bilo pametneje razmišljati v smeri popolne spremembe protokola.

5.8 Hölblov IDAK3-P1 protokol

Protokol IDAK3-P1 je učinkovit tristranski protokol za dogovor o ključu na osnovi identitete [71, 69]. Idejo za slednjega je Hölbl dobil v dvostranskem protokolu avtorjev McCullagh in Barreto [100] ter v njegovi izboljšavi avtorjev Choo, Boyd in Hitchcock [42]. Podobno kot njegovi ostali protokoli naj bi tudi ta protokol imel mnoge varnostne lastnosti, kot je varnost pred napadom z znanim ključem, prihodnja varnost itd. V tem razdelku bomo pokazali, da to ni res, saj lahko pasivni napadalec izračuna skupni sejni ključ.

Predstavitev protokola

Hölblov protokol IDAK3-P1 je tristranski protokol za overjen dogovor o ključu, s katerim se lahko Anita, Bojan in Cene preko javnega kanala dogovorijo za skupni sejni ključ in medsebojno overijo. Temelji na kriptografiji na osnovi identitete, zato mora generator zasebnih ključev vsakemu uporabniku izračunati trajni zasebni ključ in mu ga dostaviti preko varnega kanala. Za svoje delovanje uporablja bilinearna parjenja, zato njegova varnost temelji na bilinearnem in vmesnem Diffie-Hellmanovem problemu (glej §2.2.3). Protokol je tudi izjemo učinkovit, saj si lahko vsi trije udeleženci med seboj izmenjajo sporočila v enem krogu.

Protokol 16 Hölblov tristranski dogovor o ključu IDAK3-P1

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   (a) Naj bosta $\mathbb{G}_1$ in $\mathbb{G}_2$ grupi, $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ bilinearno parjenje, $P \in \mathbb{G}_1$ element praštevilskega reda p in $H : \{0,1\}^* \to \mathbb{Z}_p^*$ kriptografska zgoščevalna funkcija.

   (b) Generator zasebnih ključev izbere glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna glavni javni ključ $S = sP$ in ga javno objavi.

   (c) Aniti z identiteto $\mathrm{ID}_\mathcal{A}$ generator zasebnih ključev izračuna zgostitev $I_\mathcal{A} = H(\mathrm{ID}_\mathcal{A})$, vrednost $Q_\mathcal{A} = (I_\mathcal{A} + s)P$ in njen trajni zasebni

ključ $S_{\mathcal{A}} = (I_{\mathcal{A}} + s)^{-1}P$, ki ji ga preko varnega kanala dostavi v trajno last.

2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}, \mathcal{C}: &\quad T_{\mathcal{AB}},\, T_{\mathcal{AC}} \quad (1a) \\ \mathcal{B} \to \mathcal{A}, \mathcal{C}: &\quad T_{\mathcal{BA}},\, T_{\mathcal{BC}} \quad (1b) \\ \mathcal{C} \to \mathcal{A}, \mathcal{B}: &\quad T_{\mathcal{CA}},\, T_{\mathcal{CB}} \quad (1c) \end{aligned}$$

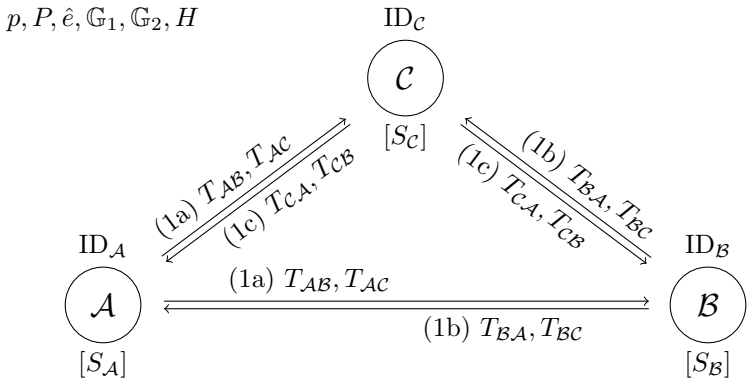
3. *Dogovor o ključu.* Anita, Bojan in Cene se z javno znanimi identitetami $\mathrm{ID}_{\mathcal{A}}$, $\mathrm{ID}_{\mathcal{B}}$ ter $\mathrm{ID}_{\mathcal{C}}$ po uspešno zaključeni pripravi, v kateri so od generatorja zasebnih ključev prejeli svoje trajne zasebne ključe $S_{\mathcal{A}}$, $S_{\mathcal{B}}$ in $S_{\mathcal{C}}$, dogovorijo za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita, Bojan in Cene najprej iz njihovih identitet izračunajo vrednosti $Q_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}) + S$, $Q_{\mathcal{B}} = H(\mathrm{ID}_{\mathcal{B}}) + S$ in $Q_{\mathcal{C}} = H(\mathrm{ID}_{\mathcal{C}}) + S$. Anita nato izbere začasni zasebni ključ $r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna začasna javna ključa $T_{\mathcal{AB}} = r_{\mathcal{A}}Q_{\mathcal{B}}$ in $T_{\mathcal{AC}} = r_{\mathcal{A}}Q_{\mathcal{C}}$ ter ju pošlje v sporočilu (1a) Bojanu in Cenetu.

   Podobno Bojan in Cene izbereta svoja začasna zasebna ključa $r_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^*$ oz. $r_{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_p^*$ in pošljeta začasna javna ključa $T_{\mathcal{BA}} = r_{\mathcal{B}}Q_{\mathcal{A}}$ ter $T_{\mathcal{BC}} = r_{\mathcal{B}}Q_{\mathcal{C}}$ oz. $T_{\mathcal{CA}} = r_{\mathcal{C}}Q_{\mathcal{A}}$ ter $T_{\mathcal{CB}} = r_{\mathcal{C}}Q_{\mathcal{B}}$ v sporočilih (1b) in (1c) ostalima udeležencema protokola.

   (b) Po izmenjavi sporočil vsi trije udeleženci uporabijo svoj trajni in začasni zasebni ključ za izračun skupne skrivnosti σ. Anita jo izračuna po enačbi $\sigma_{\mathcal{A}} = \hat{e}(T_{\mathcal{BA}}, S_{\mathcal{A}})\,\hat{e}(T_{\mathcal{CA}}, S_{\mathcal{A}})\,\hat{e}(P,P)^{r_{\mathcal{A}}}$, Bojan po enačbi $\sigma_{\mathcal{B}} = \hat{e}(T_{\mathcal{AB}}, S_{\mathcal{B}})\,\hat{e}(T_{\mathcal{CB}}, S_{\mathcal{B}})\,\hat{e}(P,P)^{r_{\mathcal{B}}}$ in Cene po enačbi $\sigma_{\mathcal{C}} = \hat{e}(T_{\mathcal{AC}}, S_{\mathcal{C}})\,\hat{e}(T_{\mathcal{BC}}, S_{\mathcal{C}})\,\hat{e}(P,P)^{r_{\mathcal{C}}}$. Če so vsi izračuni pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}} = \sigma_{\mathcal{C}}$ in vsi trije lahko iz izmenjanih sporočil $M_{\mathcal{A}} = (T_{\mathcal{AB}}, T_{\mathcal{AC}})$, $M_{\mathcal{B}} = (T_{\mathcal{BA}}, T_{\mathcal{BC}})$ ter $M_{\mathcal{C}} = (T_{\mathcal{CA}}, T_{\mathcal{CB}})$ izpeljejo skupni sejni ključ $K = \mathrm{kdf}(\sigma, M_{\mathcal{A}}, M_{\mathcal{B}}, M_{\mathcal{C}}, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{C}})$.

Hitro lahko preverimo, da na koncu protokola vsi trije udeleženci izračunajo isto skupno skrivnost $\sigma = \hat{e}(P,P)^{r_{\mathcal{A}}+r_{\mathcal{B}}+r_{\mathcal{C}}}$ in iz nje, izmenjanih sporočil ter identitet izpeljejo skupni sejni ključ.



(a)

$$
\begin{aligned}
\mathcal{A}: &\quad r_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{B}: &\quad r_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{C}: &\quad r_{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_p^* \\
\mathcal{A},\mathcal{B},\mathcal{C}: &\quad Q_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}) + S, \quad Q_{\mathcal{B}} = H(\mathrm{ID}_{\mathcal{B}}) + S, \\
&\quad Q_{\mathcal{C}} = H(\mathrm{ID}_{\mathcal{C}}) + S \\
\mathcal{A} \to \mathcal{B},\mathcal{C}: &\quad T_{\mathcal{AB}} = r_{\mathcal{A}} Q_{\mathcal{B}},\ T_{\mathcal{AC}} = r_{\mathcal{A}} Q_{\mathcal{C}} \qquad \text{(1a)} \\
\mathcal{B} \to \mathcal{A},\mathcal{C}: &\quad T_{\mathcal{BA}} = r_{\mathcal{B}} Q_{\mathcal{A}},\ T_{\mathcal{BC}} = r_{\mathcal{B}} Q_{\mathcal{C}} \qquad \text{(1b)} \\
\mathcal{C} \to \mathcal{A},\mathcal{B}: &\quad T_{\mathcal{CA}} = r_{\mathcal{C}} Q_{\mathcal{A}},\ T_{\mathcal{CB}} = r_{\mathcal{C}} Q_{\mathcal{B}} \qquad \text{(1c)}
\end{aligned}
$$

(b)

$$
\begin{aligned}
\mathcal{A}: &\quad \sigma = \hat{e}(T_{\mathcal{BA}}, S_{\mathcal{A}})\, \hat{e}(T_{\mathcal{CA}}, S_{\mathcal{A}})\, \hat{e}(P,P)^{r_{\mathcal{A}}} \\
\mathcal{B}: &\quad \sigma = \hat{e}(T_{\mathcal{AB}}, S_{\mathcal{B}})\, \hat{e}(T_{\mathcal{CB}}, S_{\mathcal{B}})\, \hat{e}(P,P)^{r_{\mathcal{B}}} \\
\mathcal{C}: &\quad \sigma = \hat{e}(T_{\mathcal{AC}}, S_{\mathcal{C}})\, \hat{e}(T_{\mathcal{BC}}, S_{\mathcal{C}})\, \hat{e}(P,P)^{r_{\mathcal{C}}} \\
\mathcal{A},\mathcal{B},\mathcal{C}: &\quad M_{\mathcal{A}} = (T_{\mathcal{AB}}, T_{\mathcal{AC}}), \quad M_{\mathcal{B}} = (T_{\mathcal{BA}}, T_{\mathcal{BC}}), \\
&\quad M_{\mathcal{C}} = (T_{\mathcal{CA}}, T_{\mathcal{CB}}) \\
\mathcal{A},\mathcal{B},\mathcal{C}: &\quad K = \mathrm{kdf}(\sigma, M_{\mathcal{A}}, M_{\mathcal{B}}, M_{\mathcal{C}}, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{C}})
\end{aligned}
$$

Slika 5.10: Hölblov tristranski dogovor o ključu IDAK3-P1

Varnostna analiza

Varnostna pomanjkljivost opisanega protokola je bolj algebraične narave, zato jo bomo razkrili v dokazu naslednjega izreka.

Izrek 5.8. *Hölblov protokol IDAK3-P1 za tristranski overjen dogovor o ključu na osnovi identitete ni varen v prisotnosti pasivnega napadalca. Slednji lahko iz izmenjanih sporočil izračuna skupni sejni ključ.*

Dokaz. Naj bo Oskar pasivni napadalec, ki je prisluškoval dogovoru o ključu med Anito, Bojanom in Cenetom. Potem lahko iz Anitinega poslanega sporočila prebere vrednosti $T_{\mathcal{AB}}$ in $T_{\mathcal{AC}}$, iz Bojanovega sporočila $T_{\mathcal{BA}}$ in $T_{\mathcal{BC}}$ ter na koncu še vrednosti $T_{\mathcal{CA}}$ in $T_{\mathcal{CB}}$, ki ju je v svojem sporočilu poslal Cene. Ker Oskar pozna njihove identitete $\mathrm{ID}_{\mathcal{A}}$, $\mathrm{ID}_{\mathcal{B}}$ in $\mathrm{ID}_{\mathcal{C}}$, lahko najprej izračuna zgostitve

$$I_{\mathcal{A}} = H(\mathrm{ID}_{\mathcal{A}}), \quad I_{\mathcal{B}} = H(\mathrm{ID}_{\mathcal{B}}) \quad \text{in} \quad I_{\mathcal{C}} = H(\mathrm{ID}_{\mathcal{C}})$$

ter nato še vrednosti

$$\begin{aligned} R_{\mathcal{A}} &= (I_{\mathcal{B}} - I_{\mathcal{C}})^{-1}(T_{\mathcal{AB}} - T_{\mathcal{AC}}) \\ &= (I_{\mathcal{B}} - I_{\mathcal{C}})^{-1}(r_{\mathcal{A}}(I_{\mathcal{B}} + s)P - r_{\mathcal{A}}(I_{\mathcal{C}} + s)P) \\ &= (I_{\mathcal{B}} - I_{\mathcal{C}})^{-1}(I_{\mathcal{B}} - I_{\mathcal{C}})r_{\mathcal{A}}P = r_{\mathcal{A}}P, \\ R_{\mathcal{B}} &= (I_{\mathcal{A}} - I_{\mathcal{C}})^{-1}(T_{\mathcal{BA}} - T_{\mathcal{BC}}) = r_{\mathcal{B}}P, \\ R_{\mathcal{C}} &= (I_{\mathcal{A}} - I_{\mathcal{B}})^{-1}(T_{\mathcal{CA}} - T_{\mathcal{CB}}) = r_{\mathcal{C}}P. \end{aligned}$$

Te uporabi za izračun skupne skrivnosti

$$\sigma = \hat{e}(R_{\mathcal{A}}, P)\,\hat{e}(R_{\mathcal{B}}, P)\,\hat{e}(R_{\mathcal{C}}, P) = \hat{e}(P, P)^{r_{\mathcal{A}}+r_{\mathcal{B}}+r_{\mathcal{C}}},$$

iz katere lahko izpelje skupni sejni ključ, ki pa bi moral biti v protokolu za dogovor o ključu znan le udeležencem protokola. Od tod lahko zaključimo, da Hölblov protokol IDAK3-P1 ni varen. □

Zgornji izrek nam pravzaprav pravi, da protokol IDAK3-P1 ne zadošča niti osnovnim varnostnim zahtevam protokolov za overjen dogovor o ključu in zato ni primeren za uporabo v praksi.

5.9 Hölblov IDAK3-P2 protokol

Zadnji protokol za dogovor o ključu, ki ga bomo obravnavali v tej disertaciji, je protokol IDAK3-P2 [71, 69]. Tudi ta sodi med tristranske protokole za overjen dogovor o ključu na osnovi identitete in za vzpostavitev ključa izkorišča lastnosti bilinearnih parjenj. Za overjanje izmenjanih sporočil uporablja malce prirejeno verzijo digitalnega podpisa, ki sta ga predstavila Cha in Cheon [32]. Po učinkovitosti naj bi bil primerljiv s sorodnimi protokoli, hkrati pa naj bi imel tudi vse želene varnostne lastnosti. Kot bomo videli v nadaljevanju, to ne drži, saj se lahko notranji napadalec z napadom s ponavljanjem lažno predstavlja kot neka druga oseba.

Predstavitev protokola

Hölblov protokol IDAK3-P2 je tristranski overjen dogovor o ključu, s katerim se lahko Anita, Bojan in Cene preko javnega kanala dogovorijo za skupni sejni ključ in medsebojno overijo. Protokol za svoje delovanje uporablja generator zasebnih ključev, ki ima v lasti glavni tajni ključ in uporabnikom preko varnega kanala izdaja trajne zasebne ključe. Protokol je zasnovan na Jouxovem protokolu, kateremu je dodano overjanje izmenjanih sporočil s prirejeno shemo za digitalni podpis. Ker protokol uporablja bilinearna parjenja, naj bi njegova varnost temeljila na bilinearnem in vmesnem Diffie-Hellmanovem problemu. Protokol je tudi učinkovit, saj je izmenjava sporočil možna v enem samem krogu.

Protokol 17 Hölblov tristranski dogovor o ključu IDAK3-P2

1. *Priprava (izbira javnih parametrov in generiranje ključev).*

   Ta faza je podobna prvi fazi protokola 14, le da je zgoščevalna funkcija H_2 definirana kot preslikava $H_2 : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{Z}_p^*$.

2. *Izmenjava sporočil.*

$$\begin{aligned} \mathcal{A} \to \mathcal{B}, \mathcal{C} : &\quad R_{\mathcal{A}}, V_{\mathcal{A}}, U_{\mathcal{A}} \quad (1a) \\ \mathcal{B} \to \mathcal{A}, \mathcal{C} : &\quad R_{\mathcal{B}}, V_{\mathcal{B}}, U_{\mathcal{B}} \quad (1b) \\ \mathcal{C} \to \mathcal{A}, \mathcal{B} : &\quad R_{\mathcal{C}}, V_{\mathcal{C}}, U_{\mathcal{C}} \quad (1c) \end{aligned}$$

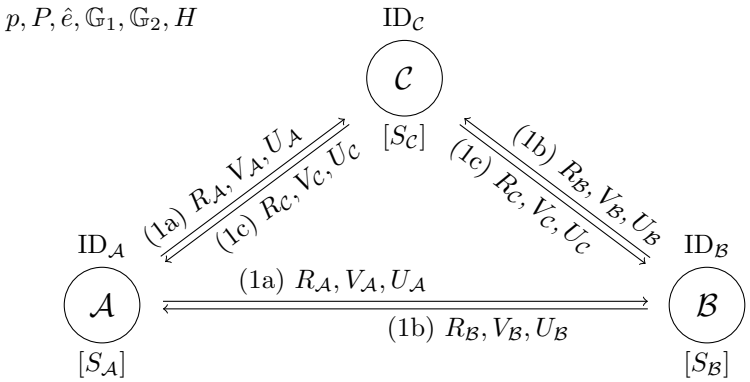
3. *Dogovor o ključu.* Anita, Bojan in Cene se z javno znanimi identitetami $\mathrm{ID}_{\mathcal{A}}$, $\mathrm{ID}_{\mathcal{B}}$ ter $\mathrm{ID}_{\mathcal{C}}$ po uspešno zaključeni pripravi, v kateri so od generatorja zasebnih ključev prejeli svoje trajne zasebne ključe $S_{\mathcal{A}}$, $S_{\mathcal{B}}$ in $S_{\mathcal{C}}$, dogovorijo za skupni sejni ključ z izvedbo naslednjih korakov.

   (a) Anita izbere svoja začasna zasebna ključa $r_{\mathcal{A}}, v_{\mathcal{A}} \xleftarrow{\$} \mathbb{Z}_p^*$ in izračuna začasna javna ključa $R_{\mathcal{A}} = r_{\mathcal{A}}P$ ter $V_{\mathcal{A}} = v_{\mathcal{A}}Q_{\mathcal{A}}$, katera skupaj z vrednostjo $U_{\mathcal{A}} = (v_{\mathcal{A}} + H_2(R_{\mathcal{A}}, V_{\mathcal{A}}))S_{\mathcal{A}}$ pošlje v sporočilu (1a) Bojanu in Cenetu.

   Podobno Bojan in Cene izbereta svoja začasna zasebna ključa $r_{\mathcal{B}}, v_{\mathcal{B}} \xleftarrow{\$} \mathbb{Z}_p^*$ oz. $r_{\mathcal{C}}, v_{\mathcal{C}} \xleftarrow{\$} \mathbb{Z}_p^*$, izračunata javna ključa $R_{\mathcal{B}} = r_{\mathcal{B}}P$ in $V_{\mathcal{B}} = v_{\mathcal{B}}Q_{\mathcal{B}}$ oz. $R_{\mathcal{C}} = r_{\mathcal{C}}P$ in $V_{\mathcal{C}} = v_{\mathcal{C}}Q_{\mathcal{C}}$ ter ju skupaj z vrednostjo $U_{\mathcal{B}} = (v_{\mathcal{B}} + H_2(R_{\mathcal{B}}, V_{\mathcal{B}}))S_{\mathcal{B}}$ oz. $U_{\mathcal{C}} = (v_{\mathcal{C}} + H_2(R_{\mathcal{C}}, V_{\mathcal{C}}))S_{\mathcal{C}}$ pošljeta v sporočilih (1b) in (1c) ostalima udeležencema.

   (b) Po izmenjavi sporočil vsi izračunajo zgostitve $h_{\mathcal{A}} = H_2(R_{\mathcal{A}}, V_{\mathcal{A}})$, $h_{\mathcal{B}} = H_2(R_{\mathcal{B}}, V_{\mathcal{B}})$ in $h_{\mathcal{C}} = H_2(R_{\mathcal{C}}, V_{\mathcal{C}})$. Anita nato preveri pristnost prejetih sporočil s preverjanjem enakosti $\hat{e}(U_{\mathcal{B}} + U_{\mathcal{C}}, P) = \hat{e}(V_{\mathcal{B}} + h_{\mathcal{B}}Q_{\mathcal{B}} + V_{\mathcal{C}} + h_{\mathcal{C}}Q_{\mathcal{C}}, S)$, Bojan s preverjanjem enakosti $\hat{e}(U_{\mathcal{A}} + U_{\mathcal{C}}, P) = \hat{e}(V_{\mathcal{A}} + h_{\mathcal{A}}Q_{\mathcal{A}} + V_{\mathcal{C}} + h_{\mathcal{C}}Q_{\mathcal{C}}, S)$ in Cene s preverjanjem enakosti $\hat{e}(U_{\mathcal{A}} + U_{\mathcal{B}}, P) = \hat{e}(V_{\mathcal{A}} + h_{\mathcal{A}}Q_{\mathcal{A}} + V_{\mathcal{B}} + h_{\mathcal{B}}Q_{\mathcal{B}}, S)$. Če vse enakosti veljajo, je medsebojno overjanje uspešno zaključeno in Anita, Bojan ter Cene lahko izračunajo skupno skrivnost σ. Anita jo izračuna po enačbi $\sigma_{\mathcal{A}} = \hat{e}(R_{\mathcal{B}}, R_{\mathcal{C}})^{r_{\mathcal{A}}}$, Bojan po enačbi $\sigma_{\mathcal{B}} = \hat{e}(R_{\mathcal{A}}, R_{\mathcal{C}})^{r_{\mathcal{B}}}$ in Cene po enačbi $\sigma_{\mathcal{C}} = \hat{e}(R_{\mathcal{A}}, R_{\mathcal{B}})^{r_{\mathcal{C}}}$. V kolikor so vsi izračuni pravilni, potem velja $\sigma = \sigma_{\mathcal{A}} = \sigma_{\mathcal{B}} = \sigma_{\mathcal{C}}$ in vsi trije lahko iz izmenjanih sporočil $M_{\mathcal{A}} = (R_{\mathcal{A}}, V_{\mathcal{A}}, U_{\mathcal{A}})$, $M_{\mathcal{B}} = (R_{\mathcal{B}}, V_{\mathcal{B}}, U_{\mathcal{B}})$ in $M_{\mathcal{C}} = (R_{\mathcal{C}}, V_{\mathcal{C}}, U_{\mathcal{C}})$ izpeljejo skupni sejni ključ $K = \mathrm{kdf}(\sigma, M_{\mathcal{A}}, M_{\mathcal{B}}, M_{\mathcal{C}}, \mathrm{ID}_{\mathcal{A}}, \mathrm{ID}_{\mathcal{B}}, \mathrm{ID}_{\mathcal{C}})$.

Na koncu protokola vsi trije udeleženci izračunajo isto skupno skrivnost $\sigma = \hat{e}(P, P)^{r_{\mathcal{A}} r_{\mathcal{B}} r_{\mathcal{C}}}$. Iz nje nato izpeljejo sejni ključ pod pogojem, da so bila preverjanja v prejšnjih korakih uspešno izvedena.



(a)

$$
\begin{aligned}
\mathcal{A}: &\quad r_\mathcal{A}, v_\mathcal{A} \xleftarrow{\$} \mathbb{Z}_p^*, \quad R_\mathcal{A} = r_\mathcal{A} P, \quad V_\mathcal{A} = v_\mathcal{A} Q_\mathcal{A} \\
\mathcal{B}: &\quad r_\mathcal{B}, v_\mathcal{B} \xleftarrow{\$} \mathbb{Z}_p^*, \quad R_\mathcal{B} = r_\mathcal{B} P, \quad V_\mathcal{B} = v_\mathcal{B} Q_\mathcal{B} \\
\mathcal{C}: &\quad r_\mathcal{C}, v_\mathcal{C} \xleftarrow{\$} \mathbb{Z}_p^*, \quad R_\mathcal{C} = r_\mathcal{C} P, \quad V_\mathcal{C} = v_\mathcal{C} Q_\mathcal{C} \\
\mathcal{A} \to \mathcal{B}, \mathcal{C}: &\quad R_\mathcal{A},\, V_\mathcal{A},\, U_\mathcal{A} = (v_\mathcal{A} + H_2(R_\mathcal{A}, V_\mathcal{A}))S_\mathcal{A} \qquad (1a) \\
\mathcal{B} \to \mathcal{A}, \mathcal{C}: &\quad R_\mathcal{B},\, V_\mathcal{B},\, U_\mathcal{B} = (v_\mathcal{B} + H_2(R_\mathcal{B}, V_\mathcal{B}))S_\mathcal{B} \qquad (1b) \\
\mathcal{C} \to \mathcal{A}, \mathcal{B}: &\quad R_\mathcal{C},\, V_\mathcal{C},\, U_\mathcal{C} = (v_\mathcal{C} + H_2(R_\mathcal{C}, V_\mathcal{C}))S_\mathcal{C} \qquad (1c)
\end{aligned}
$$

(b)

$$
\begin{aligned}
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad h_\mathcal{A} = H_2(R_\mathcal{A}, V_\mathcal{A}),\, h_\mathcal{B} = H_2(R_\mathcal{B}, V_\mathcal{B}) \\
&\quad h_\mathcal{C} = H_2(R_\mathcal{C}, V_\mathcal{C}) \\
\mathcal{A}: &\quad \hat{e}(U_\mathcal{B} + U_\mathcal{C}, P) \stackrel{?}{=} \hat{e}(V_\mathcal{B} + h_\mathcal{B} Q_\mathcal{B} + V_\mathcal{C} + h_\mathcal{C} Q_\mathcal{C}, S) \\
\mathcal{B}: &\quad \hat{e}(U_\mathcal{A} + U_\mathcal{C}, P) \stackrel{?}{=} \hat{e}(V_\mathcal{A} + h_\mathcal{A} Q_\mathcal{A} + V_\mathcal{C} + h_\mathcal{C} Q_\mathcal{C}, S) \\
\mathcal{C}: &\quad \hat{e}(U_\mathcal{A} + U_\mathcal{B}, P) \stackrel{?}{=} \hat{e}(V_\mathcal{A} + h_\mathcal{A} Q_\mathcal{A} + V_\mathcal{B} + h_\mathcal{B} Q_\mathcal{B}, S) \\
\mathcal{A}: &\quad \sigma = \hat{e}(R_\mathcal{B}, R_\mathcal{C})^{r_\mathcal{A}} \\
\mathcal{B}: &\quad \sigma = \hat{e}(R_\mathcal{A}, R_\mathcal{C})^{r_\mathcal{B}} \\
\mathcal{C}: &\quad \sigma = \hat{e}(R_\mathcal{A}, R_\mathcal{B})^{r_\mathcal{C}} \\
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad M_\mathcal{A} = (R_\mathcal{A}, V_\mathcal{A}, U_\mathcal{A}), \quad M_\mathcal{B} = (R_\mathcal{B}, V_\mathcal{B}, U_\mathcal{B}), \\
&\quad M_\mathcal{C} = (R_\mathcal{C}, V_\mathcal{C}, U_\mathcal{C}) \\
\mathcal{A}, \mathcal{B}, \mathcal{C}: &\quad K = \mathrm{kdf}(\sigma, M_\mathcal{A}, M_\mathcal{B}, M_\mathcal{C}, \mathrm{ID}_\mathcal{A}, \mathrm{ID}_\mathcal{B}, \mathrm{ID}_\mathcal{C})
\end{aligned}
$$

Slika 5.11: Hölblov tristranski dogovor o ključu IDAK3-P2

Varnostna analiza

Varnostna pomanjkljivost opisanega protokola se nahaja v napačni uporabi digitalnega podpisa za overjanje izmenjanih sporočil. Podpisi namreč uporabnikom ne zagotavljajo, da so izbrani začasni zasebni ključi $r_{\mathcal{A}}$, $r_{\mathcal{B}}$ in $r_{\mathcal{C}}$ sveži. Ravno nasprotno, v digitalnih podpisih so običajno te vrednosti bile izbrane enkrat daleč nazaj v preteklosti. V tristranskem protokolu za dogovor o ključu lahko to izkoristi napadalec, če v imenu tretje osebe pošlje njeno staro sporočilo, ki ga je poslala v eni izmed prejšnjih izvedb protokola.

Izrek 5.9. *Hölblov protokol IDAK3-P2 za tristranski overjen dogovor o ključu na osnovi identitete ni odporen na napad lažnega predstavljanja. Notranji napadalec Oskar lahko Anito lažno prepriča, da si deli sejni ključ z njim in Bojanom, čeprav si ga v resnici deli samo z njim.*

Dokaz. Naj bo Oskar notranji napadalec, ki je v eni izmed prejšnjih sej protokola, v kateri je med drugimi sodeloval tudi Bojan, prisluškoval izmenjanim sporočilom in si shranil Bojanove poslane vrednosti $R_{\mathcal{B}}$, $V_{\mathcal{B}}$ in $U_{\mathcal{B}}$. Če se želijo Anita, Bojan in Oskar dogovoriti za skupni sejni ključ, potem lahko Oskar uporabi te podatke in se v protokolu Aniti lažno predstavi kot Bojan. To stori tako, da prestreže sporočilo $(R_{\mathcal{A}}, V_{\mathcal{A}}, U_{\mathcal{A}})$, ki ga v prvem krogu Anita pošlje ostalim udeležencem protokola. Nato ji v Bojanovem imenu pošlje shranjeno sporočilo $(R_{\mathcal{B}}, V_{\mathcal{B}}, U_{\mathcal{B}})$ in v svojem imenu sporočilo $(R_{\mathcal{O}}, V_{\mathcal{O}}, U_{\mathcal{O}})$, ki ga je sam sestavil po definiciji protokola. Pri tem se spomnimo, da je Oskar legitimen uporabnik in ima zato v lasti svoj zasebni ključ $S_{\mathcal{O}}$, s katerim lahko izračuna vse tri vrednosti v sporočilu. Ko se izmenjava sporočil zaključi, Anita in Oskar izračunata skupni sejni ključ. Na izračun slednjega ne vpliva dejstvo, da noben izmed njiju ne pozna (nekoč izbrane) Bojanove začasne vrednosti $r_{\mathcal{B}}$. Slednjo bi za izračun sejnega ključa potreboval Bojan, če bi bil v protokolu seveda aktiven. Ker pa ni, namesto njega sporočila pošilja Oskar. Tako je na koncu protokola Anita napačno prepričana, da si deli sejni ključ tudi z Bojanom. Od tod sledi, da protokol ni odporen na napad lažnega predstavljanja. □

Z opisanim napadom smo dokazali, da protokol IDAK3-P2 ni odporen na napade notranje osebe. Razlog za to se skriva v dejstvu, da avtor ni upošteval

možnosti ponovnega pošiljanja sporočil s prejšnjih sej protokola. Predlagani protokol zato ni varen in je potreben konkretnih izboljšav.

5.10 Selvijeva shema

Po pregledu shem za digitalni podpis na osnovi identitete so Selvi in sodelavci opazili, da so vse sheme, ki uporabljajo bilinearna parjenja, verjetnostne. Hkrati so tudi ugotovili, da je realizacija deterministične sheme na osnovi identitete v grupah sestavljenega reda še vedno odprt problem. Za njegovo rešitev so zato predlagali novo shemo za digitalni podpis [134], ki so jo zasnovali na krepki predpostavki RSA. Ker njihova shema ne uporablja nobenega vira naključnosti, se da digitalne podpise različnih podpisnikov združiti v en sam podpis brez njihove predhodne komunikacije (glej §4.3.4). Združevanje je možno opraviti v enem samem krogu, zato je predlagana shema izjemno učinkovita. Poleg tega naj bi bila tudi varna, saj so avtorji v modelu naključnega preroka "formalno" dokazali, da je shema odporna na obstoječe poneverbe pri prilagodljivem napadu z izbranim sporočilom in identiteto.

V tem razdelku bomo pokazali, da je dokaz varnosti napačen in da predlagana shema ni varna. Našli smo namreč učinkovit algoritem za ustvarjanje univerzalnih poneverb, s katerim je možno poneveriti podpis poljubnega sporočila. Pri tem je v povprečju potrebno poznati zgolj dvanajst pristnih digitalnih podpisov.

Predstavitev sheme

Selvijeva deterministična shema za digitalni podpis na osnovi identitete uporablja generator zasebnih ključev, ki pripravi sistemske parametre in izbere glavni tajni ključ. S slednjim lahko vsakemu uporabniku izračuna zasebni ključ in mu ga preko varnega kanala pošlje v trajno last. Shemo sestavljajo naslednji štirje algoritmi (glej def. 4.3).

Shema 7 Selvijev digitalni podpis na osnovi identitete

1. `Setup` na podlagi varnostnega parametra 1^κ izbere praštevili p in q, izračuna $n = pq$, naključno izbere eksponent e velikosti približno $\kappa/4$ ter izračuna število d, za katerega velja $ed \equiv 1 \pmod{\varphi(n)}$. Nato izbere tri kriptografske zgoščevalne funkcije $H_0 : \{0,1\}^{\ell_{\mathrm{ID}}} \times \{0,1\} \to \mathbb{Z}_n^*$ in $H_1, H_2 : \{0,1\}^{\ell_M} \times \{0,1\}^{\ell_{\mathrm{ID}}} \times \{0,1\} \to \{0,1\}^{\kappa/2}$, kjer ℓ_{ID} predstavlja bitno velikost identitete uporabnika in ℓ_M velikost sporočil. Kot izhod vrne sistemske parametre H_0, H_1 in H_2, glavni javni ključ (e, n) ter glavni tajni ključ (d, n).

2. `KeyGen` sprejme identiteto ID in glavni tajni ključ (d, n) ter vrne zasebni ključ $\mathrm{sk} = (d_0, d_1)$ uporabnika s to identiteto, kjer je $g_0 = H_0(\mathrm{ID}, 0)$, $g_1 = H_0(\mathrm{ID}, 1)$, $d_0 = (g_0)^d \bmod n$ in $d_1 = (g_1)^d \bmod n$.

3. `Sign` kot vhod sprejme sporočilo m, zasebni ključ (d_0, d_1) in identiteto ID. Nato izbere $\beta \xleftarrow{\$} \{0,1\}$ in izračuna zgostitvi $h_1 = H_1(m, \mathrm{ID}, \beta)$, $h_2 = H_2(m, \mathrm{ID}, \beta)$ ter število $\alpha = (d_0)^{h_1}(d_1)^{h_2} \bmod n$. Kot izhod vrne digitalni podpis $\sigma = (\alpha, \beta)$.

4. `Verify` kot vhod sprejme sporočilo m, digitalni podpis $\sigma = (\alpha, \beta)$ in identiteto ID. Nato izračuna vrednosti $g_0 = H_0(\mathrm{ID}, 0)$, $g_1 = H_0(\mathrm{ID}, 1)$, $h_1 = H_1(m, \mathrm{ID}, \beta)$ in $h_2 = H_2(m, \mathrm{ID}, \beta)$ ter podpis sprejme kot veljaven, če in samo če velja

$$\alpha^e \equiv (g_0)^{h_1}(g_1)^{h_2} \pmod{n}.$$

Preprost izračun

$$\alpha^e \equiv \left((d_0)^{h_1}(d_1)^{h_2}\right)^e \equiv (g_0)^{edh_1}(g_1)^{edh_2} \equiv (g_0)^{h_1}(g_1)^{h_2} \pmod{n}$$

nam razkrije, da algoritem za podpisovanje vrne veljavne podpise in je zato shema dosledna.

Iz opisa sheme lahko opazimo, da algoritem za podpisovanje sporočil naključno izbere bit β, kar namiguje, da shema ni deterministična. Avtorji

sheme opozarjajo, da temu ni tako, saj lahko podpisnik izbere bit s psevdonaključno funkcijo, kateri kot vhod poda svoj zasebni ključ (d_0, d_1) in sporočilo m. Tako pripravljeni digitalni podpisi bodo izgledali popolnoma naključno s strani tretje osebe, medtem ko bodo enolični z vidika podpisnika. Shema je torej deterministična, zato jo je možno preprosto razširiti do sheme z možnostjo združevanja. Podrobnosti razširitve za varnostno analizo niso pomembne, zato jih bomo izpustili. Zainteresiran bralec jih lahko poišče v izvirnem članku [134].

Varnostna analiza

Zgoraj opisana shema za digitalni podpis naj bi bila odporna na obstoječe poneverbe pri prilagodljivem napadu z izbranim sporočilom in identiteto. Njena varnost je bila dokazana v modelu naključnega preroka s prevedbo na krepak problem RSA (glej def. 2.41). V nadaljevanju bomo razkrili, da temu ni tako, saj je shema ranljiva ne samo za obstoječe poneverbe temveč tudi za univerzalne.

Izrek 5.10. *Sevlijeva deterministična shema za digitalni podpis na osnovi identitete ni odporna na univerzalne poneverbe. Napadalec Oskar lahko poneveri Anitin digitalni podpis poljubnega sporočila, če ima v povprečju na voljo dvanajst njenih pristnih podpisov.*

Dokaz. Naj bo Oskar napadalec, ki želi v varnostnem modelu sheme za digitalni podpis na osnovi identitete (glej §4.2.3) ustvariti univerzalno poneverbo Anitinega podpisa. Za dosego svojega cilja prične varnostno igro z izzivalcem Iztokom, v kateri z veliko verjetnostjo izračuna Anitin zasebni ključ (d_0, d_1) in ga nato uporabi za poneverjanje podpisa poljubnega sporočila.

1. *Priprava.* V prvem delu igre Iztok izbere varnostni parameter κ in pripravi javne parametre H_0, H_1 ter H_2, glavni javni ključ (e, n) in glavni tajni ključ (d, n). Slednjega varno shrani, medtem ko parametre in javni ključ razkrije Oskarju.

2. *Usposabljanje.* V tem delu ima Oskar na voljo preroka za razkrivanje zasebnih ključev in preroka za podpisovanje sporočil, s katerima si

lahko pomaga pri ustvarjanju univerzalne poneverbe. Če želi poneveriti digitalni podpis Anite, potem najprej razkrije njen zasebni ključ. To stori tako, da naključno izbere štiri sporočila

$$m^{(1)}, m^{(2)}, m^{(3)} \text{ in } m^{(4)}$$

ter jih skupaj z Anitino javno znano identiteto $\mathrm{ID}_{\mathcal{A}}$ pošlje preroku za podpisovanje. Iztok, ki simulira oba preroka, mora odgovoriti na prejeto poizvedbo. Zato z glavnim tajnim ključem (d, n) izračuna Anitin zasebni $\mathrm{sk}_{\mathcal{A}} = (d_0, d_1)$, z njim podpiše vsa štiri sporočila in njihove digitalne podpise

$$\sigma^{(i)} = (\alpha^{(i)}, \beta^{(i)}) \qquad (1 \leq i \leq 4),$$

v odgovoru preroka pošlje nazaj Oskarju. Ta nato uporabi javne parametre in izračuna vmesne vrednosti

$$h_1^{(i)} = H_1\left(m^{(i)}, \mathrm{ID}, \beta^{(i)}\right), \quad h_2^{(i)} = H_2\left(m^{(i)}, \mathrm{ID}, \beta^{(i)}\right),$$

ki jih je Iztok uporabil pri podpisovanju izbranih sporočil, najmanjša skupna večkratnika

$$L_1 = v\left(h_1^{(1)}, h_1^{(2)}\right), \quad L_2 = v\left(h_1^{(3)}, h_1^{(4)}\right),$$

naravna števila

$$D_1 = L_1/h_1^{(1)}, \quad D_2 = L_1/h_1^{(2)},$$
$$D_3 = L_2/h_1^{(3)}, \quad D_4 = L_2/h_1^{(4)},$$

in

$$T_1 = \left(\alpha^{(1)}\right)^{D_1} \bmod n, \quad T_2 = \left(\alpha^{(2)}\right)^{D_2} \bmod n,$$
$$T_3 = \left(\alpha^{(3)}\right)^{D_3} \bmod n, \quad T_4 = \left(\alpha^{(4)}\right)^{D_4} \bmod n.$$

Od tod lahko opazimo, da velja

$$T_i \equiv \left((d_0)^{h_1^{(i)}} (d_1)^{h_2^{(i)}}\right)^{D_i} \equiv \left(d_0\right)^{L_{\lceil i/2 \rceil}} \left(d_1\right)^{D_i h_2^{(i)}} \pmod{n},$$

in da je element T_i vedno obrnljiv, za $1 \leq i \leq 4$. Zadnje sledi iz dejstva, da elementa d_0 in d_1 ležita v multiplikativni grupi $\mathbb{Z}_n^*$, saj

sta izračunana z uporabo zgoščevalne funkcije H_0. Oskar lahko zato z razširjenim Evklidovim algoritmom učinkovito izračuna inverza T_2^{-1} in T_4^{-1} ter vrednosti

$$E_1 = D_1 h_2^{(1)} - D_2 h_2^{(2)}, \quad E_2 = D_3 h_2^{(3)} - D_4 h_2^{(4)}.$$

Ker je za nadaljnje izvajanje napada pomembno, da sta si števili E_1 in E_2 tuji, Oskar prekine igro, če temu ni tako. Nato jo začne znova, vse dokler ne najde štiri ustrezna sporočila. Ko jih najde, izračuna še vrednosti

$$U_1 = T_1 T_2^{-1} \bmod n, \quad U_2 = T_3 T_4^{-1} \bmod n,$$

ki ju bo v nadaljevanju potreboval za izračun Anitinega zasebnega ključa. Iz enakosti

$$U_1 \equiv \left(d_1\right)^{D_1 h_2^{(1)} - D_2 h_2^{(2)}} \equiv \left(d_1\right)^{E_1} \pmod{n},$$

$$U_2 \equiv \left(d_1\right)^{D_3 h_2^{(3)} - D_4 h_2^{(4)}} \equiv \left(d_1\right)^{E_2} \pmod{n},$$

je razvidno, da sta U_1 in U_1 potenci števila d_1, ki predstavlja del Anitinega zasebnega ključa. To lahko izkoristi Oskar, če z razširjenim Evklidovim algoritmom izračuna celi števili a in b, ki ustrezata enakosti

$$aE_1 + bE_2 = 1,$$

in nato še

$$d_1 = U_1^a \, U_2^b \bmod n.$$

Pri tem upošteva, da če je število a negativno, potem mora najprej izračunati vrednost $U_1^{|a|} \bmod n$ in šele nato z Evklidovim algoritmom njen inverz. Podobno velja tudi za število b. Ker velja relacija

$$U_1^a \, U_2^b \equiv d_1^{aE_1 + bE_2} \equiv d_1 \pmod{n},$$

je Oskar izračunal drugi del zasebnega ključa $\mathrm{sk}_{\mathcal{A}} = (d_0, d_1)$.

S podobnim postopkom nato poskusi izračunati še prvi del ključa d_0. To stori tako, da v izračunih zamenja vlogo vrednosti $h_1(i)$ in $h_2(i)$, za $1 \leq i \leq 4$. Če mu z istimi sporočili in njihovimi digitalnimi podpisi

uspe izračunati d_0, potem mu je uspelo v celoti razkriti Anitin zasebni ključ. Zato zaključi usposabljanje in prične z zadnjim delom varnostne igre.

3. *Poneverjanje.* Če Oskar v fazi usposabljanja ni prekinil igre, je uspešno izračunal Anitin zasebni ključ. Ker pri tem ni uporabil preroka za razkrivanje zasebnih ključev, lahko varnostno igro zaključi s poneverjanjem njenega podpisa. Zato si naključno izbere novo sporočilo m, ga podpiše z Anitinim zasebnim ključem in skupaj s podpisom vrne Iztoku.

Oskar zmaga v zgoraj opisani varnostni igri, če vrne veljaven digitalni podpis. To se zgodi natanko tedaj, ko v fazi usposabljanja ne prekine igre. Ker lahko do prekinitve pride le pri izračunu vrednosti d_1 in d_0, v kolikor števili E_1 in E_2 nista tuji, pred izračunom Oskarjeve verjetnosti za zmago izračunajmo verjetnost tega dogodka. Če si podrobno ogledamo izračuna teh dveh števil, lahko opazimo, da sta pri izbrani identiteti neodvisna. Število E_1 je odvisno zgolj od vrednosti $m^{(1)}$, $m^{(2)}$, $\beta^{(1)}$ in $\beta^{(2)}$, medtem ko je število E_2 odvisno od $m^{(3)}$, $m^{(4)}$, $\beta^{(3)}$ in $\beta^{(4)}$. Pri izračunu se uporablja tudi zgoščevalna funkcija H_1, ki je v varnostni igri modelirana z naključnim prerokom. To pa pomeni, da se njeni izhodi obnašajo popolnoma naključno in takšni sta pri fiksni identiteti tudi števili E_1 in E_2. Iz teorije verjetnosti nam je znano, da je verjetnost, da sta si dve naključno izbrani števili tuji, enaka $6/\pi^2$, kar je približno 61% [65, Thm. 332]. Ker morata biti števili E_1 in E_2 tuji tako pri izračunu vrednosti d_1 kot pri izračunu d_0, Oskar zmaga v igri z verjetnostjo

$$P[\text{zmaga}] = (6/\pi^2)^2 \approx 0{,}37.$$

Od tod sledi, da mora Oskar za razkritje Anitinega zasebnega ključa in za ustvarjanje univerzalnih poneverb varnostno igro v povprečju ponoviti približno trikrat, za kar potrebuje dvanajst pristnih digitalnih podpisov. Ker verjetnost 37% ni zanemarljiva, smo s tem dokazali, da Selvijeva shema ni varna pred univerzalnimi poneverbami in zato tudi ni varna pred obstoječimi. □

Poglavje 6

Predlog sheme za digitalni podpis

V konferenčnem članku [133] so Selvi in sodelavci predstavili deterministično shemo za digitalni podpis na osnovi identitete ter njeno varnost dokazali v modelu naključnega preroka s prevedbo na vmesni Diffie-Hellmanov problem. Dokaz varnosti, za razliko od mnogih obstoječih shem, ne uporablja razvejitvene leme [124] in nudi tesnejšo prevedbo varnosti na težak računski problem. Kot smo že omenili, je shema deterministična, zato je možno vse digitalne podpise enega podpisnika učinkovito združiti v en kratek podpis.

Cilj tega poglavja je predstaviti izboljšano deterministično shemo za digitalni podpis na osnovi identitete [113], ki je učinkovitejša od Selvijeve sheme, saj uporablja manjši glavni javni/tajni ključ in večina algoritmov sheme izvede manj računsko zahtevnih operacij. Primerjavo učinkovitosti nazorno prikazuje tabela 6.1. V nadaljevanju bomo shemo gradili postopoma, tako da bomo najprej predstavili verjetnostno shemo za digitalni podpis, s katero bomo sestavili shemo na osnovi identitete. Nato bomo predstavili še njeno deterministično verzijo, ki omogoča delno združevanje. Za vsako shemo bomo tudi dokazali, da je v modelu naključnega preroka varna pred obstoječimi poneverbami.

	Selvi in sodelavci	Naš predlog
Učinkovitost		
`Setup`	$2M$	M
`KeyGen`	$a + 2m + 2M + 2H$	$2M + H$
`Sign`	$M + H$	$M + H$
`Verify`	$A + M + 4P + 3H$	$4P + 2H$
Poraba prostora		
Dolžina glavnega tajnega ključa	$2\|\mathbb{G}_1\|$	$\|\mathbb{G}_1\|$
Dolžina glavnega javnega ključa	$2\|\mathbb{Z}_p^*\|$	$\|\mathbb{Z}_p^*\|$
Dolžina digitalnega podpisa	$1 + 3\|\mathbb{G}_1\|$	$1 + 3\|\mathbb{G}_1\|$
Število zgoščevalnih funkcij	3	2

Tabela 6.1: Primerjava Selvijeve in naše deterministične sheme za digitalni podpis na osnovi identitete. V njej smo z a in m označili zahtevnost seštevanja ter množenja v $\mathbb{Z}_p$, z A in M zahtevnost seštevanja ter skalarnega množenja v $\mathbb{G}_1$, s P zahtevnost bilinearnega parjenja in s H zahtevnost izračuna zgostitve.

6.1 Verjetnostna shema za digitalni podpis

Najprej bomo z uporabo kriptografije javnih ključev sestavili učinkovito verjetnostno shemo za digitalni podpis in dokazali njeno varnost s prevedbo na vmesni Diffie-Hellmanov problem. Shema je zelo preprosta in primerljiva z BLS shemo (glej shemo 4), ki so jo leta 2001 predstavili Boneh in sodelavci [26].

Shema 8 Verjetnostni digitalni podpis

1. `KeyGen` na podlagi varnostnega parametra 1^κ izbere ciklični grupi $\mathbb{G}_1$ in $\mathbb{G}_2$ praštevilskega reda p, element P grupe $\mathbb{G}_1$, bilinearno parjenje $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ in zgoščevalno funkcijo $H : \{0,1\}^* \times \mathbb{Z}_p \to \mathbb{G}_1$. Nato vrne zasebni ključ $a \xleftarrow{\$} \mathbb{Z}_p^*$ in javni ključ $A = aP$.

2. `Sign` kot vhod sprejme sporočilo m in zasebni ključ a. Nato izbere naključno število $r \xleftarrow{\$} \mathbb{Z}_p$ in izračuna vrednost $V = aH(m, r)$ ter vrne digitalni podpis $\sigma = (r, V)$.

3. `Verify` sprejme sporočilo m, digitalni podpis $\sigma = (r, V)$ in javni ključ A ter sprejme podpis kot veljaven, če in samo če velja

$$\hat{e}(V, P) = \hat{e}(H(m, r), A). \tag{6.1}$$

Naslednja dva izreka zagotavljata, da je verjetnostna shema za digitalni podpis dosledna in varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom v modelu naključnega preroka.

Izrek 6.1. *Verjetnostna shema za digitalni podpis je dosledna.*

Dokaz. Naj bo $\sigma = (r, V)$ digitalni podpis sporočila m, ki ga je ustvarila oseba z zasebnim ključem a in javnim ključem A. Potem zaradi lastnosti bilinearnih parjenj velja

$$\hat{e}(V, P) = \hat{e}(aH(m, r), P) = \hat{e}(H(m, r), aP) = \hat{e}(H(m, r), A),$$

kar pomeni, da enačba (6.1) drži za vsako sporočilo m in par zasebni/javni ključ (a, A). Torej je podpis veljaven. Od tod sledi, da je verjetnostna shema za digitalni podpis dosledna. □

Izrek 6.2. *Denimo, da lahko napadalec v modelu naključnega preroka z verjetnostjo ε ustvari obstoječo poneverbo verjetnostne sheme za digitalni podpis s prilagodljivim napadom z izbranim sporočilom. Potem je vmesni Diffie-Hellmanov problem v $\mathbb{G}_1$ in $\mathbb{G}_2$ rešljiv v polinomskem času z isto verjetnostjo ε.*

Dokaz. Naj bo napadalec Oskar polinomski algoritem, ki zna v varnostnem modelu sheme ustvariti obstoječo poneverbo z verjetnostjo ε. V nadaljevanju bomo videli, da lahko izzivalec Iztok izkoristi njegovo znanje za reševanje vmesnega Diffie-Hellmanovega problema (glej def. 2.46). Iztok bo kot vhod sprejel izziv problema $(P, aP, bP) \in \mathbb{G}_1^3$ in vrnil njegovo rešitev $abP \in \mathbb{G}_1$ v polinomskem času z verjetnostjo ε.

V varnostnem modelu Iztok in Oskar igrata igro, v kateri Oskar poskuša s prilagodljivim napadom z izbranim sporočilom poneveriti Iztokov podpis. Pri tem ima na voljo dva preroka, enega za podpisovanje in enega za izračun

zgostitve sporočil. Slednji mu je na voljo zato, ker so zgoščevalne funkcije modelirane kot naključni preroki. Simulacija prerokov je prepuščena Iztoku, ki njune odgovore izbira na konstruktiven način tako, da lahko na koncu igre uporabi Oskarjevo poneverbo za rešitev problema. Pri tem mora Iztok opraviti določeno mero knjigovodstva, saj mora hraniti seznam sporočil, za katere je Oskar zahteval digitalne podpise ali zgostitve.

Za rešitev vmesnega Diffie-Hellmanovega problema Iztok najprej izvede naslednjo igro, ki se malce razlikuje od osnovne igre. Razlika Oskarju ni opazna, zato je prepričan, da sodeluje v pravi varnostni igri.

1. *Priprava.* V prvem delu igre naj bi Iztok z algoritmom `KeyGen` pripravil svoj javni in zasebni ključ, a tega ne stori, saj želi rešiti izziv Diffie-Hellmanovega problema. Zato si raje za svoj javni ključ izbere vrednost $A = aP$, ki jo je prejel v izzivu, in jo pošlje Oskarju. Pri tem je potrebno opomniti, da Iztok ne pozna "svojega" zasebnega ključa a. To v nadaljevanju igre ne bo prestavljalo težave, saj bo Iztok odgovore prerokov izbiral tako, da bo znal podpisovati sporočila.

2. *Usposabljanje.* Ko Oskar prejme Iztokov javni ključ, se prične drugi del varnostne igre. V njem ima dostop do preroka za zgoščevalno funkcijo H in do preroka za podpisovanje. S prvim prerokom lahko izračuna zgostitve poljubnih sporočil, medtem ko z drugim prejme njihove digitalne podpise. Pri tem velja omejitev, da lahko Oskar pošlje prerokom le polinomsko omejeno število poizvedb.

   (a) *Prerok zgoščevalne funkcije H.* Za simulacijo tega preroka Iztok vzdržuje seznam L četveric $(m, r, h, H_{m,r})$, ki je na začetku prazen. Ko prerok prejme poizvedbo s sporočilom m in vrednostjo r, Iztok stori naslednje.

   - Če $(m, r, \cdot, \cdot)$ že obstaja v seznamu L, potem iz seznama prebere celotno četverico $(m, r, h, H_{m,r})$.
   - Sicer izbere število $h \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna

$$H_{m,r} = \begin{cases} h(bP), & \text{če je poizvedbo poslal Oskar,} \\ hP, & \text{sicer,} \end{cases}$$

in shrani $(m, r, h, H_{m,r})$ v seznam L. Pri tem zopet opomnimo, da Iztok ne pozna števila b, temveč le vrednost bP.

- Kot odgovor vrne zgostitev $H(m, r) = H_{m,r}$.

Opazimo lahko, da so odgovori preroka izbrani naključno z enakomerno porazdelitvijo iz grupe $\mathbb{G}_1$ in jih zato Oskar vidi kot naključno izbrane vrednosti. Od tod sledi, da se zgoščevalna funkcija H obnaša kot naključen prerok.

(b) *Prerok za podpisovanje.* Ko prerok prejme poizvedbo za digitalni podpis sporočila m, Iztok izvede naslednje korake.

- Najprej izbere število $r \xleftarrow{\$} \mathbb{Z}_p$ in preveri, če se četverica $(m, r, \cdot, \cdot)$ že nahaja v seznamu L. Če se, potem izbere novo število r, sicer nadaljuje z naslednjim korakom.
- S prerokom zgoščevalne funkcije H izračuna zgostitev $H_{m,r} = H(m, r) = hP$. Iztok število $h \in \mathbb{Z}_p^*$ pozna, saj ga lahko poišče v seznamu L.
- Na koncu izračuna vrednost $V = h(aP)$ in vrne digitalni podpis $\sigma = (r, V)$.

Preprost izračun

$$\hat{e}(V, P) = \hat{e}(ahP, P) = \hat{e}(hP, aP) = \hat{e}(H(m, r), A)$$

nam razkrije, da vrnjeni digitalni podpisi zadoščajo enačbi (6.1) in so zato veljavni.

3. *Poneverjanje.* V zadnjem delu igre Oskar kot izhod vrne sporočilo $\hat{m}$ in njegov digitalni podpis $\hat{\sigma} = (\hat{r}, \hat{V})$, ki ni bil ustvarjen s strani preroka za podpisovanje.

Če v opisani igri Oskar vrne veljaven digitalni podpis, lahko to izkoristi Iztok in reši vmesni Diffie-Hellmanov problem.

Preden opišemo postopek reševanja, se prepričajmo, da je preroka za zgoščevalno funkcijo H prvi za zgostitev $H(\hat{m}, \hat{r})$ vprašal Oskar enkrat v fazi usposabljanja. Ta pomožni rezultat bomo dokazali s protislovjem. Denimo, da je preroku poizvedbo s sporočilom $\hat{m}$ in vrednostjo $\hat{r}$ prvi poslal Iztok.

Ker v drugem delu igre Iztok pošilja poizvedbe le pri podpisovanju sporočil, je moral na Oskarjevo zahtevo podpisati sporočilo $\hat{m}$ z uporabo naključne vrednosti $\hat{r}$ in mu v odgovoru preroka vrniti digitalni podpis $\sigma = (\hat{r}, V)$. Iz enačbe (6.1) je možno razbrati, da le ena vrednost V skupaj z $\hat{r}$ tvori veljaven podpis. Od tod sledi, da sta vrednosti V in $\hat{V}$ enaki, prav tako pa digitalna podpisa σ in $\hat{\sigma}$. To pa je v protislovju z dejstvom, da je Oskar vrnil digitalni podpis, ki ga ni ustvaril prerok. Poizvedbo je torej prvi poslal Oskar ali pa ta še nikoli ni bila poslana. Zadnjo možnost lahko takoj izločimo, saj če zgostitev $H(\hat{m}, \hat{r})$ še nikoli ni bila izračunana, potem lahko Iztok pri simulaciji preroka izbere takšno njeno vrednost, da podpis $\hat{\sigma}$ ne bo veljaven. To pa bi bilo v nasprotju s predpostavko, da je Oskar vrnil veljaven digitalni podpis.

Preroka za zgoščevalno funkcijo je torej prvi za zgostitev $H(\hat{m}, \hat{r})$ vprašal Oskar. Ker je v varnostni igri izračun zgostitve odvisen od pošiljatelja poizvedbe, v tem primeru velja

$$H(\hat{m}, \hat{r}) = H_{\hat{m},\hat{r}} = bhP.$$

Pri tem je potrebno omeniti, da Iztok pozna vrednost $h \in \mathbb{Z}_p^*$, saj je le-ta shranjena v četverici $(\hat{m}, \hat{r}, h, H_{\hat{m},\hat{r}})$ seznama L. Digitalni podpis $\hat{\sigma} = (\hat{r}, \hat{V})$ je veljaven, zato iz enačbe (6.1) sledi

$$\hat{e}(\hat{V}, P) = \hat{e}(H_{\hat{m},\hat{r}}, A) = \hat{e}(bhP, aP),$$

od tod pa

$$\hat{V} = abhP.$$

Postopek reševanja vmesnega Diffie-Hellmanovega problema je preprost. Ko Iztok od Oskarja prejme digitalni podpis $\hat{\sigma} = (\hat{r}, \hat{V})$, najprej izračuna zgostitev $H_{\hat{m},\hat{r}} = H(\hat{m}, \hat{r})$ in nato iz seznama L prebere četverico $(\hat{m}, \hat{r}, h, H_{\hat{m},\hat{r}})$. Ker pozna vrednosti $\hat{V}$ in h, lahko reši problem, če izračuna

$$abP = h^{-1}\hat{V}.$$

Zaključimo lahko, da če napadalec Oskar vrne obstoječo poneverbo z verjetnostjo ε, potem je vmesni Diffie-Hellmanov problem rešljiv z isto verjetnostjo ε. □

Če predpostavimo, da noben polinomski algoritem nima nezanemarljive prednosti pri reševanju vmesnega Diffie-Hellmanovega problema, potem je verjetnost, da napadalec razbije verjetnostno shemo za digitalni podpis zanemarljiva. Shema je zato varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom v modelu naključnega preroka.

6.2 Shema za digitalni podpis na osnovi identitete

Verjetnostno shemo za digitalni podpis, predstavljeno v prejšnjem razdelku, lahko uporabimo za sestavo sheme na osnovi identitete. Osnovna ideja v ozadju je, da se verjetnostno shemo uporabi dvakrat, enkrat pri izdajanju zasebnih ključev in enkrat pri podpisovanju sporočil. V prvem primeru z njo generator zasebnih ključev podpiše identiteto uporabnika in njegov zasebni ključ, medtem ko v drugem uporabnik s svojim ključem digitalno podpiše sporočilo. Za tako sestavljeno shemo je možno dokazati, da je varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom in identiteto v modelu naključnega preroka. Njeno varnost je možno prevesti na vmesni Diffie-Hellmanov problem.

Podoben postopek za sestavo sheme na osnovi identitete iz sheme za digitalni podpis so predlagali tudi Bellare, Namprempre in Neven [11]. Njihova ideja je bila, da generator zasebnih ključev vsakem uporabniku izda digitalno potrdilo, iz katerega je možno razbrati njegov javni ključ in identiteto. Uporabnik nato pri ustvarjanju podpisov poleg sporočila priloži še digitalno potrdilo. S takšnim pristopom bi lahko tudi mi sestavili shemo na osnovi identitete z enakimi varnostnimi lastnostmi. Vendar pa je naša konstrukcija boljša, saj varnost dosežemo brez uporabe digitalnih potrdil.

Shema 9 Verjetnostni digitalni podpis na osnovi identitete

1. `Setup` na podlagi varnostnega parametra 1^κ izbere ciklični grupi $\mathbb{G}_1$ in $\mathbb{G}_2$ praštevilskega reda p, generator P grupe $\mathbb{G}_1$, bilinearno parjenje $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ in zgoščevalni funkciji $H_1 : \mathbb{G}_1 \times \{0,1\}^* \to \mathbb{G}_1$ ter $H_2 : \{0,1\}^* \times \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{G}_1$. Nato vrne glavni tajni ključ $s \xleftarrow{\$} \mathbb{Z}_p^*$ in objavi parametre p, $\hat{e}$, $\mathbb{G}_1$, $\mathbb{G}_2$, H_1 ter H_2, in glavni javni ključ $S = sP$.

2. `KeyGen` kot vhod sprejme identiteto ID in glavni tajni ključ s. Nato izbere $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna vrednosti $A = aP$ in $X = sH_1(A, \mathrm{ID})$ ter kot izhod vrne zasebni ključ (a, A, X).

3. `Sign` sprejme sporočilo m, zasebni ključ (a, A, X) in identiteto ID. Nato izbere število $r \xleftarrow{\$} \mathbb{Z}_p$ in izračuna vrednost $V = aH_2(m, r, \mathrm{ID})$ ter kot izhod vrne digitalni podpis $\sigma = (r, V, A, X)$.

4. `Verify` kot vhod sprejme sporočilo m, digitalni podpis $\sigma = (r, V, A, X)$ in identiteto ID ter sprejme podpis kot veljaven, če in samo če velja

$$\hat{e}(X, P) = \hat{e}(H_1(A, \mathrm{ID}), S), \tag{6.2}$$

in

$$\hat{e}(V, P) = \hat{e}(H_2(m, r, \mathrm{ID}), A). \tag{6.3}$$

Pri pripravi zasebnega ključa lahko opazimo, da generator izbere zasebni ključ t uporabnika z identiteto ID, izračuna pripadajoči javni ključ $T = tP$ in ga podpiše z uporabo verjetnostne sheme za digitalni podpis. Vrednost T in identiteta ID sta tako zavzeli vlogi sporočila m in naključne vrednosti r. Podobno se verjetnostna shema ponovno uporabi pri podpisovanju sporočila, le da se tu v zgostitev vključi še identiteta uporabnika.

Naslednja dva rezultata nam dokazujeta, da je shema za digitalni podpis na osnovi identitete dosledna in varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom in identiteto v modelu naključnega preroka.

Izrek 6.3. *Shema za digitalni podpis na osnovi identitete je dosledna.*

Dokaz. Naj bo $\sigma = (r, V, A, X)$ digitalni podpis sporočila m, ki ga je ustvarila oseba z zasebnim ključem (a, A, X) in identiteto ID. Zaradi lastnosti bilinearnih parjenj velja

$$\hat{e}(X, P) = \hat{e}(sH_1(A, \mathrm{ID}), P) = \hat{e}(H_1(A, \mathrm{ID}), sP) = \hat{e}(H_1(A, \mathrm{ID}), S),$$

$$\hat{e}(V, P) = \hat{e}(aH_2(m, r, \mathrm{ID}), P) = \hat{e}(H_2(m, r, \mathrm{ID}), aP) = \hat{e}(H_2(m, r, \mathrm{ID}), A),$$

kar dokazuje, da enačbi (6.2) in (6.3) držita, ter da je podpis veljaven. Od tod sledi, da je shema na osnovi identitete dosledna. □

Izrek 6.4. *Denimo, da lahko napadalec v modelu naključnega preroka z verjetnostjo ε ustvari obstoječo poneverbo sheme za digitalni podpis na osnovi identitete s prilagodljivim napadom z izbranim sporočilom in identiteto. Potem je vmesni Diffie-Hellmanov problem v $\mathbb{G}_1$ in $\mathbb{G}_2$ rešljiv v polinomskem času z verjetnostjo ε/q_{H_1}, kjer je q_{H_1} število zgostitev zgoščevalne funkcije H_1, ki jih je v času napada izračunal napadalec.*

Dokaz. Naj bo napadalec Oskar polinomski algoritem, ki zna v varnostnem modelu sheme ustvariti obstoječo poneverbo z verjetnostjo ε. Potem bomo dokazali, da lahko izzivalec Iztok izkoristi njegovo znanje za reševanje vmesnega Diffie-Hellmanovega problema. Iztok kot vhod sprejme izziv problema $(P, sP, tP) \in \mathbb{G}_1^3$ in vrne njegovo rešitev $stP \in \mathbb{G}_1$ v polinomskem času z verjetnostjo ε/q_{H_1}, kjer je q_{H_1} število zgostitev zgoščevalne funkcije H_1, ki jih tekom napada izračuna Oskar.

Za rešitev problema Iztok najprej izvede naslednjo igro, v kateri ima Oskar dostop do obeh prerokov za zgoščevalni funkciji, do preroka za razkrivanje zasebnih ključev in do preroka za podpisovanje. Igra je sestavljena iz treh delov in je podobna igri v dokazu varnosti verjetnostne sheme za digitalni podpis.

1. *Priprava.* V prvem delu igre naj bi Iztok z algoritmom `Setup` pripravil glavni javni in tajni ključ generatorja zasebnih ključev. Vendar tega ne stori, temveč za glavni tajni ključ izbere vrednost $S = sP$, ki jo je prejel v izzivu problema. Slednjo pošlje Oskarju, kljub temu, da ne pozna glavnega tajnega ključa s. Iztok nato izbere število $k \xleftarrow{\$} \{1, 2, \ldots, q_{H_1}\}$ in k-to identiteto poslano preroku za zgoščevalno funkcijo H_1 označi za

ciljno identiteto. Pri tem je potrebno omeniti, da Iztok in Oskar ciljne identitete ne poznata vnaprej.

2. *Usposabljanje.* Ko Oskar prejme glavni javni ključ, se prične drugi del varnostne igre, v katerem Iztok simulira štiri preroke. S prvima dvema lahko Oskar izračuna zgostitve poljubnih sporočil, s tretjim njihove podpise in s četrtim razkrije zasebne ključe uporabnikov. Pri tem velja omejitev, da lahko Oskar pošlje le polinomsko omejeno število poizvedb.

   (a) *Prerok zgoščevalne funkcije H_1.* Za simulacijo tega preroka Iztok vzdržuje seznam L_1 četveric (A, ID, h, H_A), ki je na začetku prazen. Ko prerok prejme i-to poizvedbo z vrednostjo A in identiteto ID, Iztok stori naslednje.

   - Če je $i = k$, potem Iztok določi ciljno identiteto $\mathrm{ID}_k = \mathrm{ID}$.
   - Če $(A, \mathrm{ID}, \cdot, \cdot)$ že obstaja v seznamu L_1, potem iz seznama prebere celotno četverico (A, ID, h, H_A).
   - Sicer izbere število $h \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna

     $$H_A = \begin{cases} h(tP), & \text{če je poizvedbo poslal Oskar,} \\ hP, & \text{sicer,} \end{cases}$$

     in shrani (A, ID, h, H_A) v seznam L_1.
   - Kot odgovor vrne zgostitev $H_1(A, \mathrm{ID}) = H_A$.

   (b) *Prerok zgoščevalne funkcije H_2.* Za simulacijo tega preroka Iztok vzdržuje seznam L_2 četveric $(m, r, \mathrm{ID}, h, H_{m,r})$, ki je na začetku prazen. Ko prerok prejme poizvedbo s sporočilom m in vrednostjo r, Iztok stori naslednje.

   - Če $(m, r, \mathrm{ID}, \cdot, \cdot)$ že obstaja v seznamu L_2, potem iz seznama prebere celotno četverico $(m, r, \mathrm{ID}, h, H_{m,r})$.
   - Sicer izbere število $h \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna

     $$H_{m,r} = \begin{cases} h(tP), & \text{če je poizvedbo poslal Oskar,} \\ hP, & \text{sicer,} \end{cases}$$

     in shrani $(m, r, \mathrm{ID}, h, H_{m,r})$ v seznam L_2.

- Kot odgovor vrne zgostitev $H_2(m, r, \mathrm{ID}) = H_{m,r}$.

Pri tem lahko opazimo, da so odgovori obeh prerokov izbrani naključno z enakomerno porazdelitvijo iz grupe $\mathbb{G}_1$ in jih zato Oskar vidi kot naključno izbrane vrednosti. Od tod sledi, da se obe zgoščevalni funkciji H_1 in H_2 obnašata kot naključen prerok.

(c) *Prerok za razkrivanje.* Pri prilagodljivem napadu z izbrano identiteto lahko Oskar razkrije zasebne ključe oseb po lastni izbiri. Da lahko Iztok simulira tega preroka, vzdržuje seznam L_3 četveric (a, A, X, ID), ki je na začetku prazen. Ko prerok prejme poizvedbo z identiteto ID, Iztok stori naslednje.

- Če $(\cdot, \cdot, \cdot, \mathrm{ID})$ že obstaja v seznamu L_3, potem iz seznama prebere celotno četverico (a, A, X, ID).
- Sicer izbere število $a \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna

$$A = \begin{cases} a(sP), & \text{če je } \mathrm{ID} = \mathrm{ID}_k, \\ aP, & \text{sicer,} \end{cases}$$

in preveri, če je bila zgostitev $H_1(A, \mathrm{ID})$ že izračunana in zato četverica $(A, \mathrm{ID}, \cdot, \cdot)$ obstaja v seznamu L_1. Če obstaja, potem izbere novo naključno vrednost a, sicer nadaljuje z naslednjim korakom.

- S prerokom zgoščevalne funkcije H_1 izračuna zgostitev $H_A = H_1(A, \mathrm{ID}) = hP$. Iztok število $h \in \mathbb{Z}_p^*$ pozna, saj ga lahko poišče v seznamu L_1. Nato izračuna vrednost $X = h(sP)$ in shrani (a, A, X, ID) v seznam L_3.
- Na koncu vrne zasebni ključ (a, A, X), če je $\mathrm{ID} = \mathrm{ID}_k$, in prekine igro sicer.

Opazimo lahko, da sta števili a in h izbrani naključno z enakomerno porazdelitvijo in zato Oskar vidi zasebne ključe uporabnikov kot naključne vrednosti, ter da Iztok ne pozna zasebnega ključa osebe z identiteto ID_k, saj shranjena vrednost a v četverici $(a, \cdot, \cdot, \mathrm{ID}_k)$ ni pravilna. Zato mora prekiniti igro, če prejme poizvedbo za razkritje zasebnega ključa osebe s to identiteto.

(d) *Prerok za podpisovanje.* Ko prerok prejme poizvedbo za digitalni podpis sporočila m osebe z identiteto ID, Iztok stori naslednje.

- Če zasebni ključ osebe z identiteto ID še ni bil izračunan, potem pošlje poizvedbo preroku za razkrivanje. Nato iz seznama L_3 prebere četverico (a, A, X, ID), ki vsebuje njen zasebni ključ.
- Preden izračuna digitalni podpis, si izbere število $r \xleftarrow{\$} \mathbb{Z}_p$ in preveri, če je bila zgostitev $H_2(m, r, \mathrm{ID})$ že izračunana in zato četverica $(m, r, \mathrm{ID}, \cdot, \cdot)$ obstaja v seznamu L_2. Če obstaja, potem si izbere novo število r, sicer nadaljuje z naslednjim korakom.
- S prerokom zgoščevalne funkcije H_2 izračuna zgostitev $H_{m,r} = H_2(m, r, \mathrm{ID}) = hP$. Iztok število $h \in \mathbb{Z}_p^*$ pozna, saj ga lahko poišče v seznamu L_2. Nato izračuna še vrednost

$$V = \begin{cases} ah(sP), & \text{če je ID} = \mathrm{ID}_k, \\ ahP, & \text{sicer,} \end{cases}$$

- Na koncu vrne digitalni podpis $\sigma = (r, V, A, X)$.

Na tem mestu bomo ustavili opis igre in se prepričali, da so vsi digitalni podpisi preroka veljavni. Najprej se spomnimo, da če $\mathrm{ID} \neq \mathrm{ID}_k$, potem velja $A = aP$, $X = aH_A$, $H_A = H_1(A, \mathrm{ID}) = h_1P$, $H_{m,r} = H_2(m, r, \mathrm{ID}) = h_2P$ in $V = aH_{m,r} = ah_2P$, za neki števili $h_1, h_2 \in \mathbb{Z}_p^*$. Preprost izračun vrednosti

$$\hat{e}(X, P) = \hat{e}(ah_1P, P) = \hat{e}(h_1P, aP) = \hat{e}(H_1(A, \mathrm{ID}), S)$$

$$\hat{e}(V, P) = \hat{e}(ah_2P, P) = \hat{e}(h_2P, aP) = \hat{e}(H_2(m, r, \mathrm{ID}), A)$$

nam razkrije, da digitalni podpisi v tem primeru zadoščajo enačbama (6.2) in (6.3). Zato jih sprejmemo kot veljavne. Če velja $\mathrm{ID} = \mathrm{ID}_k$, se preverjanje podpisa razlikuje zgolj v enačbi (6.3), saj je v tem primeru $A = asP$ in $V = asH_{m,r} = ah_2sP$. Kljub temu je podpis veljaven, kar razkriva enakost

$$\hat{e}(V, P) = \hat{e}(ah_2sP, P) = \hat{e}(h_2P, asP) = \hat{e}(H_2(m, r, \mathrm{ID}), A).$$

3. *Poneverjanje.* V zadnjem delu igre Oskar kot izhod vrne sporočilo $\hat{m}$ in njegov digitalni podpis $\hat{\sigma} = (\hat{r}, \hat{V}, \hat{A}, \hat{X})$ osebe z identiteto $\hat{\mathrm{ID}}$. Pri tem velja, da podpis ni bil ustvarjen s strani preroka za podpisovanje in zasebni ključ osebe z identiteto $\hat{\mathrm{ID}}$ ni bil razkrit preko preroka za razkrivanje.

Če v opisani igri Oskar vrne veljaven digitalni podpis in je $\hat{\mathrm{ID}} = \mathrm{ID}_k$, lahko to izkoristi Iztok in reši vmesni Diffie-Hellmanov problem, kot je to opisano v nadaljevanju.

V kolikor je v fazi usposabljanja napadalec Oskar prvi poslal poizvedbo $H_{\hat{A}} = H_1(\hat{A}, \hat{\mathrm{ID}})$ preroku za zgoščevalno funkcijo H_1, potem je $H_{\hat{A}} = htP$. Iztok število h pozna, saj ga lahko poišče v četverici $(\hat{A}, \hat{\mathrm{ID}}, h, H_{\hat{A}})$ seznama L_1. Ker je digitalni podpis $\hat{\sigma}$ veljaven, zadošča enačbi (6.2). Od tod sledi, da je $\hat{X} = hstP$ in Iztok lahko reši vmesni Diffie-Hellmanov problem, če izračuna $stP = h^{-1}\hat{X}$. Sicer je poizvedbo $H_1(\hat{A}, \hat{\mathrm{ID}})$ prvi opravil Iztok, ko je računal zasebni ključ osebe z identiteto $\hat{\mathrm{ID}}$. Ker v drugem delu Iztok izračuna največ en ključ za vsako osebo, obstaja natanko ena četverica $(a, \hat{A}, X, \hat{\mathrm{ID}})$ v seznamu L_3. Pri tem velja $\hat{A} = asP$ in $X = sH_A$, kjer Iztok pozna vrednost $a \in \mathbb{Z}_p^*$. Ker je digitalni podpis $\hat{\sigma}$ veljaven, iz enačbe (6.3) sledi $\hat{X} = sH_A = X$.

Preden nadaljujemo s postopkom rešitve, se prepričajmo, da je preroka za zgoščevalno funkcijo H_2 prvi za zgostitev $H_2(\hat{m}, \hat{r}, \hat{\mathrm{ID}})$ vprašal Oskar. To trditev bomo dokazali s protislovjem. Denimo, da je preroku poizvedbo s sporočilom $\hat{m}$, vrednostjo $\hat{r}$ in identiteto $\hat{\mathrm{ID}}$ prvi poslal Iztok. Ker v drugem delu Iztok pošilja poizvedbe le pri podpisovanju sporočil, je moral na Oskarjevo zahtevo podpisati sporočilo $\hat{m}$ z uporabo naključne vrednosti $\hat{r}$ v imenu osebe z identiteto $\hat{\mathrm{ID}}$ in vrniti digitalni podpis $\sigma = (\hat{r}, V, \hat{A}, \hat{X})$. Iz enačbe (6.3) je možno razbrati, da le ena vrednost V skupaj z $\hat{r}$ tvori veljaven podpis. Od tod sledi, da sta vrednosti V in $\hat{V}$ enaki, prav tako pa digitalna podpisa σ in $\hat{\sigma}$. To pa je v protislovju z dejstvom, da je Oskar vrnil digitalni podpis, ki ga ni ustvaril prerok.

Iz prejšnjega odstavka vemo, da je poizvedbo $H_{\hat{m},\hat{r}} = H_2(\hat{m}, \hat{r}, \hat{\mathrm{ID}})$ prvi opravil Oskar. Zato velja $H_{\hat{m},\hat{r}} = htP$ in Iztok lahko poišče število h v seznamu L_2, če poišče četverico $(\hat{m}, \hat{r}, \hat{\mathrm{ID}}, \cdot, \cdot)$. Iz enačbe (6.3) sledi $\hat{V} = ahstP$

in Iztok lahko reši vmesni Diffie-Hellmanov problem, če izračuna vrednost $stP = (ah)^{-1}\hat{V}$.

Za konec še dokažimo, da lahko Iztok z zgoraj opisanim postopkom reši izziv problema z verjetnostjo ε/q_{H_1}. To se zgodi natanko tedaj, ko se ne zgodi nobeden izmed naslednjih dveh dogodkov:

- E_1: prerok za razkrivanje prejme poizvedbo z identiteto ID_k in zato Iztok prekine igro,
- E_2: Oskar vrne veljaven digitalen podpis sporočila osebe z identiteto $\hat{\mathrm{ID}} \neq \mathrm{ID}_k$.

Z osnovnim znanjem verjetnosti lahko ugotovimo, da se dogodka E_1 in E_2 zgodita le v primeru, če Iztok izbere napačno ciljno identiteto. Verjetnost tega dogodka je $1 - 1/q_{H_1}$ in zato lahko zaključimo, da Iztok reši vmesni Diffie-Hellmanov problem z verjetnostjo ε/q_{H_1}. □

6.3 Deterministična shema za digitalni podpis na osnovi identitete

Shemo za digitalni podpis na osnovi identitete iz prejšnjega razdelka lahko z majhnimi spremembami pretvorimo v deterministično. Varnost slednje lahko prav tako dokažemo v modelu naključnega preroka s prevedbo na vmesni Diffie-Hellmanov problem.

Deterministična shema se od sheme za digitalni podpis na osnovi identitete razlikuje v načinu izbire naključne vrednosti r v fazi podpisovanja. V osnovni shemi je ta vrednost izbrana naključno iz množice $\mathbb{Z}_p$, zaradi česar je shema verjetnostna. V deterministični verziji pa podpisnik izbere vrednost r iz množice $\{0, 1\}$ tako, da je izbira deterministična le z vidika podpisnika in naključna s strani vseh ostalih. Na primer, podpisnik lahko r določi s psevdonaključno funkcijo, ki kot vhod sprejme sporočilo m, identiteto ID in zasebni ključ (a, A, X). Na ta način je r točno določen s strani podpisnika, saj poleg generatorja zasebnih ključev le on pozna svoj zasebni ključ [58].

Shema 10 Deterministični digitalni podpis na osnovi identitete

1. `Setup` (glej shemo 9).
2. `KeyGen` (glej shemo 9).
3. `Sign` kot vhod sprejme sporočilo m, zasebni ključ (a, A, X) in identiteto ID. Nato izbere naključno število $r \xleftarrow{\$} \{0,1\}$ in izračuna vrednost $V = aH_2(m, r, \mathrm{ID})$. Kot izhod vrne digitalni podpis $\sigma = (r, V, A, X)$.
4. `Verify` (glej shemo 9).

Tudi za deterministično shemo za digitalni podpis na osnovi identitete je možno dokazati, da je dosledna in varna v modelu naključnega preroka. Ker je dokaz doslednosti enak kot pri izreku 6.3, ga bomo preskočili.

Izrek 6.5. *Deterministična shema za digitalni podpis na osnovi identitete je dosledna.* □

Izrek 6.6. *Denimo, da lahko napadalec v modelu naključnega preroka z verjetnostjo ε ustvari obstoječo poneverbo deterministične sheme za digitalni podpis na osnovi identitete s prilagodljivim napadom z izbranim sporočilom in identiteto. Potem je vmesni Diffie-Hellmanov problem v $\mathbb{G}_1$ in $\mathbb{G}_2$ rešljiv v polinomskem času z verjetnostjo $\varepsilon/2q_{H_1}$, kjer je q_{H_1} število zgostitev zgoščevalne funkcije H_1, ki jih je v času napada izračunal napadalec.*

Dokaz. Preden se lotimo dokazovanja, naj omenimo, da dokaz ne sledi direktno iz izreka 6.4. Pri shemi 9 je namreč število r izbrano naključno, zato lahko Iztok v varnosti igri pri simulaciji preroka za podpisovanje to vrednost izbira toliko časa, dokler ne najde še neuporabljene. Takšna simulacija pri deterministični shemi ni možna, saj lahko bit r zavzame natanko eno vrednost. Kljub temu si lahko pri dokazovanju pomagamo z omenjeno varnostno igro, saj je v njej potrebno spremeniti le odzive preroka za zgoščevalno funkcijo H_2 in preroka za podpisovanje.

1. *Priprava* (glej dokaz izr. 6.4).

2. *Usposabljanje.*

   - *Prerok zgoščevalne funkcije* H_1 (glej dokaz izr. 6.4).
   - *Prerok zgoščevalne funkcije* H_2. Za simulacijo tega preroka Iztok vzdržuje seznam L_2 peteric $(m, r, \mathrm{ID}, h, H_{m,r})$, ki je na začetku prazen. Ko prerok prejme poizvedbo s sporočilom m in vrednostjo r, Iztok stori naslednje.
     - Če $(m, r, \mathrm{ID}, \cdot, \cdot)$ že obstaja v seznamu L_2, potem iz seznama prebere celotno četverico $(m, r, \mathrm{ID}, h, H_{m,r})$.
     - Sicer izbere bit $b \xleftarrow{\$} \{0, 1\}$ in števili $h_0, h_1 \xleftarrow{\$} \mathbb{Z}_p^*$, izračuna
       - $H_{m,b} = h_b(tP)$ in shrani $(m, b, \mathrm{ID}, h_b, H_{m,b})$ v seznam L_2,
       - $H_{m,\bar{b}} = h_{\bar{b}}P$ in shrani $(m, \bar{b}, \mathrm{ID}, h_{\bar{b}}, H_{m,\bar{b}})$ v seznam L_2.

       Pri tem smo z $\bar{b}$ označili logično negacijo bita b.
     - Kot odgovor vrne zgostitev $H_2(m, r, \mathrm{ID}) = H_{m,r}$.

     Pri tem lahko opazimo, da je prerok za zgoščevalno funkcijo H_2 še vedno predstavljen kot naključni prerok, saj je zgostitev $H_{m,r}$ izbrana naključno z enakomerno porazdelitvijo iz grupe $\mathbb{G}_1$.
   - *Prerok za razkrivanje* (glej dokaz izr. 6.4).
   - *Prerok za podpisovanje.* Ko prerok prejme poizvedbo za digitalni podpis sporočila m osebe z identiteto ID, Iztok stori naslednje.
     - Če zasebni ključ osebe z identiteto ID še ni bil izračunan, potem pošlje poizvedbo preroku za razkrivanje. Nato iz seznama L_3 prebere četverico (a, A, X, ID), ki vsebuje njen zasebni ključ. Spomnimo se, da velja

       $$A = \begin{cases} a(sP), & \text{če je } \mathrm{ID} = \mathrm{ID}_k, \\ aP, & \text{sicer,} \end{cases}$$

       $X = aH_A$ in $H_A = H_1(A, \mathrm{ID}) = h_1 P$, za neko število $h_1 \in \mathbb{Z}_p^*$.
     - Če se $(m, 0, \mathrm{ID}, \cdot, \cdot)$ ne nahaja v seznamu L_2, pošlje poizvedbo s sporočilom m, bitom 0 in identiteto ID preroku za zgoščevalno funkcijo H_2. Nato iz seznama L_2 prebere peterici

$(m, 0, \mathrm{ID}, h_0, H_{m,0})$ in $(m, 1, \mathrm{ID}, h_1, H_{m,1})$, ter nastavi vrednost r na 0, če $H_{m,0} = h_0 P$ in na 1 sicer.

– Na koncu izračuna še

$$V = \begin{cases} ah_r(sP), & \text{če je ID} = \mathrm{ID}_k, \\ ah_r P, & \text{sicer,} \end{cases}$$

in vrne digitalni podpis $\sigma = (r, V, A, X)$.

Zopet se lahko prepričamo, da so vrnjeni digitalni podpisi veljavni. Ker smo spremenili le preroka za zgoščevalno funkcijo H_2 in preroka za podpisovanje, moramo še enkrat preveriti, če enačba (6.3) drži. Za podpise osebe z identiteto ID $\neq \mathrm{ID}_k$, velja

$$\hat{e}(V, P) = \hat{e}(ah_r P, P) = \hat{e}(h_r P, aP) = \hat{e}(H(m, r, \mathrm{ID}), T),$$

medtem ko za podpis osebe z identiteto ID_k velja

$$\hat{e}(V, P) = \hat{e}(ah_r sP, P) = \hat{e}(h_r P, asP) = \hat{e}(H(m, r, \mathrm{ID}), T).$$

V obeh primerih je enačba izpolnjena in vrnjeni podpisi so veljavni.

3. *Poneverjanje* (glej dokaz izr. 6.4).

Če v opisani igri Oskar vrne veljaven digitalni podpis in je $\hat{\mathrm{ID}} = \mathrm{ID}_k$, ter $H_2(\hat{m}, \hat{r}, \hat{\mathrm{ID}}) = H_{\hat{m},\hat{r}} = bh_{\hat{r}}(tP)$ za $h_{\hat{r}}$ in $H_{\hat{m},\hat{r}}$ iz četverice $(\hat{m}, \hat{r}, h_{\hat{r}}, H_{\hat{m},\hat{r}})$ seznama L_2, potem lahko to izkoristi Iztok in reši vmesni Diffie-Hellmanov problem, kot je to opisano v nadaljevanju.

V kolikor je v fazi usposabljanja Oskar prvi poslal poizvedbo $H_1(\hat{A}, \hat{\mathrm{ID}})$ preroku za zgoščevalno funkcijo H_1, potem Iztok nadaljuje, kot je opisano v dokazu izreka 6.4 in reši vmesni Diffie-Hellmanov problem. Sicer je poizvedbo $H_A = H_1(\hat{A}, \hat{\mathrm{ID}})$ prvi opravil Iztok. Iz dokaza izreka 6.4 vemo, da se četverica $(a, \hat{A}, \hat{X}, \hat{\mathrm{ID}})$ nahaja v seznamu L_3, kjer je $\hat{A} = asP$ in $\hat{X} = aH_A$, za neko znano vrednost $a \in \mathbb{Z}_p^*$. Iz enakosti $H_{\hat{m},\hat{r}} = bh_{\hat{r}}P$ in enačbe (6.3) zato sledi, da velja $\hat{V} = ah_{\hat{r}}stP$ in Iztok lahko reši vmesni Diffie-Hellmanov problem, če izračuna $stP = (ah_{\hat{r}})^{-1}\hat{V}$.

Za konec še dokažimo, da lahko Iztok z zgoraj opisanim postopkom reši izziv problema z verjetnostjo $\varepsilon/2q_{H_1}$. To se zgodi natanko tedaj, ko se ne zgodi eden izmed naslednjih treh dogodkov:

- E_1, E_2: glej dokaz izr. 6.4,
- E_3: Oskar si izbere tak $\hat{r}$, da velja $H_{\hat{m},\hat{r}} = h_{\hat{r}}P$.

Verjetnost dogodka E_3 je enaka $1/2$, saj smo predpostavili, da je bit $\hat{r}$ izbran naključno z Oskarjevega vidika. Ker sta dogodka $E_1 \cup E_2$ in E_3 neodvisna sledi, da lahko Iztok reši vmesni Diffie-Hellmanov problem z verjetnostjo $\varepsilon/2q_{H_1}$. □

6.4 Shema z možnostjo delnega združevanja

Podpise ustvarjene z deterministično shemo za digitalni podpis na osnovi identitete lahko delno združimo v en sam podpis s postopkom, ki ga je opisal Herranz v članku [67]. Če zanemarimo velikost bita r, potem je velikost združenega podpisa odvisna samo od števila različnih podpisnikov in ne od števila podpisanih sporočil.

Shema 11 Digitalni podpis z možnostjo delnega združevanja

1. `Setup` (glej shemo 10).
2. `KeyGen` (glej shemo 10).
3. `Sign` (glej shemo 10).
4. `Aggregate` sprejme digitalne podpise $(\sigma_i)_{i=1}^n = (r_i, V_i, A_i, X_i)_{i=1}^n$ od t različnih oseb in vrne združen podpis

$$\sigma_{\text{agg}} = \left(r_1, \ldots, r_n, \sum_{i=1}^{n} V_i, A_1, \ldots, A_t, \sum_{j=1}^{t} X_j \right).$$

5. `VerifyAgg` kot vhod sprejme sporočila $(m_i)_{i=1}^n$, združen digitalni podpis $\sigma_{\text{agg}} = (r_1, \ldots, r_n, V_{\text{agg}}, A_1, \ldots, A_t, X_{\text{agg}})$ in identitete podpisnikov $(\text{ID}_i)_{i=1}^t$. Nato ugotovi, kateri identiteti ID_i pripada vrednost A_j, za $i, j \in \{1, \ldots, t\}$, in obratno. To pripadnost bomo v nadaljevanju

označili z oznako ID_{A_j} oz. A_{ID_i}. Na koncu sprejme združen podpis kot veljaven, če in samo če velja

$$\hat{e}(X_{\mathrm{agg}}, P) = \hat{e}\left(\sum_{j=1}^{t} H_1(A_j, \mathrm{ID}_{A_j}), S\right) \tag{6.4}$$

in

$$\hat{e}(V_{\mathrm{agg}}, P) = \prod_{i=1}^{n} \hat{e}\left(H_2(m_i, r_i, \mathrm{ID}_i), A_{\mathrm{ID}_i}\right). \tag{6.5}$$

Izrek 6.7. *Shema za digitalni podpis na osnovi identitete z možnostjo združevanja je dosledna.*

Dokaz. Naj bo $\sigma_{\mathrm{agg}} = (r_1, \ldots, r_n, V_{\mathrm{agg}}, A_1, \ldots, A_t, X_{\mathrm{agg}})$ združitev digitalnih podpisov $(\sigma_i)_{i=1}^n = (r_i, V_i, A_i, X_i)_{i=1}^n$ sporočil $(m_i)_{i=1}^n$, ki so jih ustvarile osebe z identitetami $(\mathrm{ID}_i)_{i=1}^t$. Potem zaradi lastnosti bilinearnih parjenj velja

$$\begin{aligned}
\hat{e}(X_{\mathrm{agg}}, P) &= \hat{e}\left(\sum_{j=1}^{t} X_j, P\right) \\
&= \hat{e}\left(\sum_{j=1}^{t} sH_1(A_j, \mathrm{ID}_{A_j}), P\right) \\
&= \hat{e}\left(\sum_{j=1}^{t} H_1(A_j, \mathrm{ID}_{A_j}), sP\right) \\
&= \hat{e}\left(\sum_{j=1}^{t} H_1(A_j, \mathrm{ID}_{A_j}), S\right)
\end{aligned}$$

in

$$\begin{aligned}
\hat{e}(V_{\mathrm{agg}}, P) &= \hat{e}\left(\sum_{i=1}^{n} V_i, P\right) \\
&= \prod_{i=1}^{n} \hat{e}\left(V_i, P\right) \\
&= \prod_{i=1}^{n} \hat{e}\left(a_{\mathrm{ID}_i} H_2(m_i, r_i, \mathrm{ID}_i), P\right) \\
&= \prod_{i=1}^{n} \hat{e}\left(H_2(m_i, r_i, \mathrm{ID}_i), A_{\mathrm{ID}_i}\right)
\end{aligned}$$

kar dokazuje, da enačbi (6.4) in (6.5) držita, ter da je združen podpis veljaven. Od tod sledi, da je shema na osnovi identitete z možnostjo združevanja dosledna. □

Poglavje 7

Zaključek

V doktorski disertaciji smo se ukvarjali z varnostno analizo protokolov za overjen dogovor o ključu in shem za digitalni podpis. Podrobno smo preučili sestavo desetih protokolov in ene sheme ter našli njihove še neodkrite varnostne pomanjkljivosti. Te smo uporabili za sestavo napadov, s katerimi smo dokazali, da predlagani protokoli in shema niso varni, prav tako pa niso varne nekatere njihove različice. Zato smo predlagali tudi nekaj izboljšav, s katerimi lahko razkrite ranljivosti odpravimo. Z napadi smo potrdili dejstvo, da je varne kriptografske rešitve zelo težko sestaviti in da je njihovo varnost potrebno utemeljiti v ustreznem varnostnem modelu. Pri dokazih varnosti pa moramo biti zelo pazljivi, saj lahko le ena majhna napaka ogrozi celotno varnost.

Predstavili smo tudi izboljšano deterministično shemo za digitalni podpis na osnovi identitete, ki omogoča delno združevanje. Pri tej lahko vse podpise enega uporabnika združimo v en sam kratek podpis in s tem zmanjšamo porabo prostora za njihovo hranjenje ter pohitrimo preverjanje podpisov. V primerjavi z osnovno shemo je naša učinkovitejša in zato bolj primerna za praktično uporabo. Hkrati je varna pred obstoječimi poneverbami pri prilagodljivem napadu z izbranim sporočilom in identitet. Slednje smo tudi formalno dokazali v modelu naključnega preroka s prevedbo na vmesni Diffie-Hellmanov problem.

Nadaljnje delo na področju disertacije bo razdeljeno na dva dela. V prvem delu se bomo usmerili v izboljšavo varnosti in učinkovitosti predlagane

sheme. Varnost bomo poskušali dokazati v standardnem modelu brez uporabe naključnih prerokov in s tesnejšo prevedbo na izbran računski problem. Pri tem se zavedamo, da bo shemo potrebno spremeniti in ustrezno dopolniti, kar bo verjetno vplivalo tudi na njeno učinkovitost. Zato ne izključujemo možnosti, da bomo morali shemo zasnovati na drugem težkem problemu. Pri tem obstaja želja, da bi njena varnost temeljila zgolj na problemu diskretnega logaritma ali na Diffie-Hellmanovem problemu.

V drugem delu se bo delo nadaljevalo na varnostni analizi novih in starih protokolov za overjen dogovor o ključu ter shem za digitalni podpis. Še vedno si namreč želimo najti ranljivosti najbolj znanih protokolov in jih izkoristiti za pripravo novih napadov. Na področju kriptografije na osnovi identitete je cilj sestaviti tudi nov protokol za dogovor o ključu, ki bi bil tako učinkovit kot MQV protokol in hkrati varen v priznanem varnostnem modelu, po možnosti v standardnem modelu. To bo sicer zelo težko doseči, vendar pa bo že podobna učinkovitost predstavljala zelo velik dosežek.

Na koncu naj še enkrat poudarimo, da je varne protokole in sheme zelo težko sestaviti, zato je v praksi zaželeno uporabljati le tiste kriptografske rešitve, ki so bile dobro preučene in odobrene s strani najboljših kriptografov s celega sveta. Napadi, opisani v tej disertaciji, pa naj bodo mladim razvijalcem dober zgled, da se v ozadju na videz preprostih protokolov in shem skriva obširna znanost.

Literatura

[1] Adleman, L.M., *A subexponential algorithm for the discrete logarithm problem with applications to cryptography.* V *20th Symp. on Foundations of Computer Science*, str. 55–60. IEEE Computer Society Press, 1979.

[2] Adleman, L.M., *The function field sieve.* V *Algorithmic Number Theory I*, del 877 iz LNCS, str. 108–121. Springer-Verlag, 1994.

[3] Adleman, L.M. in M.D.A. Huang, *Function field sieve method for discrete logarithms over finite fields.* Inform. Comput., 151(1–2):5–16, 1999.

[4] ANSI X9.42, *Public key cryptography for the financial services industry: Agreement of symmetric keys using discrete logarithm cryptography.* ANSI, mar. 2003.

[5] ANSI X9.63, *Public key cryptography for the financial services industry: Key agreement and key transport using elliptic curve cryptography.* ANSI, nov. 2001.

[6] Atkin, A.O.L. in F. Morain, *Elliptic curves and primality proving.* Math. Comput., 61:29–68, 1993.

[7] Babbage, C., *Passages from the life of a philosopher.* Longman, Green, Longman, Roberts & Green, 1864.

[8] Barić, N. in B. Pfitzmann, *Collision-free accumulators and fail-stop signature schemes without trees.* V *Advances in Cryptology – Eurocrypt '97*, del 1233 iz LNCS, str. 480–494. Springer-Verlag, 1997.

[9] Bellare, M., R. Canetti in H. Krawczyk, *Keying hash functions for message authentication.* V *Advances in Cryptology – Crypto '96*, del 1109 iz LNCS, str. 1–15. Springer-Verlag, 1996.

[10] Bellare, M., R. Canetti in H. Krawczyk, *A modular approach to the design and analysis of authentication and key exchange protocols.* V *30th Symp. on Theory of Computing*, str. 419–428. ACM Press, 1998.

[11] Bellare, M., C. Namprempre in G. Neven, *Security proofs for identity-based identification and signature schemes.* J. Cryptol., 22(1):1–61, 2008.

[12] Bellare, M., D. Pointcheval in P. Rogaway, *Authenticated key exchange secure against dictionary attacks.* V *Advances in Cryptology – Eurocrypt 2000*, del 1807 iz LNCS, str. 139–155. Springer-Verlag, 2000.

[13] Bellare, M. in P. Rogaway, *Random oracles are practical: A paradigm for designing efficient protocols.* V *1st Conf. on Computer and Communications Security*, str. 62–73. ACM Press, 1993.

[14] Bellare, M. in P. Rogaway, *Entity authentication and key distribution.* V *Advances in Cryptology – Crypto '93*, del 773 iz LNCS, str. 232–249. Springer-Verlag, 1994.

[15] Bellare, M. in P. Rogaway, *Optimal asymmetric encryption.* V *Advances in Cryptology – Eurocrypt '94*, del 950 iz LNCS, str. 92–111. Springer-Verlag, 1995.

[16] Bellare, M. in P. Rogaway, *Provably secure session key distribution: The three party case.* V *27th Symp. on Theory of Computing*, str. 57–66. ACM Press, 1995.

[17] Bellare, M. in P. Rogaway, *The exact security of digital signatures - how to sign with RSA and Rabin.* V *Advances in Cryptology – Eurocrypt '96*, del 1070 iz LNCS, str. 399–416. Springer-Verlag, 1996.

[18] Bellare, M. in P. Rogaway, *Introduction to modern cryptography*, 2005. `http://cseweb.ucsd.edu/users/mihir/cse207/classnotes.html`.

[19] Bellovin, S.M. in M. Merritt, *Encrypted key exchange: Password-based protocols secure against dictionary attacks.* V *Symp. on Research in Security and Privacy*, str. 72–84. IEEE Computer Society Press, 1992.

[20] Bellovin, S.M. in M. Merritt, *Augmented encrypted key exchange: A password-based protocol secure against dictionary attacks and password file compromise.* V *1st Conf. on Computer and Communications Security*, str. 244–250. ACM Press, 1993.

[21] Biham, E. in O. Dunkelman, *A framework for iterative hash functions: HAIFA.* V *2nd NIST Cryptographic Hash Workshop*, 2006.

[22] Blake-Wilson, S., D. Johnson in A.J. Menezes, *Key agreement protocols and their security analysis.* V *6th Int. Conf. on Cryptography and Coding*, str. 30–45. Springer-Verlag, 1997.

[23] Boneh, D., *The decision Diffie-Hellman problem.* V *Algorithmic Number Theory III*, del 1423 iz LNCS, str. 48–63. Springer-Verlag, 1998.

[24] Boneh, D. in M. Franklin, *Identity-based encryption from the Weil pairing.* V *Advances in Cryptology – Crypto 2001*, del 2139 iz LNCS, str. 213–229. Springer-Verlag, 2001.

[25] Boneh, D., C. Gentry, B. Lynn in H. Shacham, *Aggregate and verifiably encrypted signatures from bilinear maps.* V *Advances in Cryptology – Eurocrypt 2003*, del 2656 iz LNCS, str. 416–432. Springer-Verlag, 2003.

[26] Boneh, D., B. Lynn in H. Shacham, *Short signatures from the Weil pairing.* V *Advances in Cryptology – Asiacrypt 2001*, del 2248 iz LNCS, str. 514–532. Springer-Verlag, 2001.

[27] Boyd, C. in A. Mathuria, *Protocols for Authentication and Key Establishment.* Springer-Verlag, 2003.

[28] Boyko, V., P. MacKenzie in S. Patel, *Provably secure password-authenticated key exchange using Diffie-Hellman.* V *Advances in Cryptology – Eurocrypt 2000*, del 1807 iz LNCS, str. 156–171. Springer-Verlag, 2000.

[29] Canetti, R., O. Goldreich in S. Halevi, *The random oracle methodology, revisited*. J. ACM, 51(4):557–594, 2004.

[30] Canetti, R. in H. Krawczyk, *Analysis of key-exchange protocols and their use for building secure channels*. V *Advances in Cryptology – Eurocrypt 2001*, del 2045 iz LNCS, str. 453–474. Springer-Verlag, 2001.

[31] Carstensen, C., B. Fine in G. Rosenberger, *Abstract Algebra: Applications to Galois Theory, Algebraic Geometry, and Cryptography*. De Gruyter, 2011.

[32] Cha, J.C. in J.H. Cheon, *An identity-based signature from gap Diffie-Hellman groups*. V *Public Key Cryptography 2003*, del 2567 iz LNCS, str. 18–30. Springer-Verlag, 2002.

[33] Chaum, D., *Blind signatures for untraceable payments*. V *Advances in Cryptology – Crypto '82*, str. 199–203. Springer-Verlag, 1983.

[34] Chaum, D. in H. van Antwerpen, *Undeniable signatures*. V *Advances in Cryptology – Crypto '89*, del 435 iz LNCS, str. 212–216. Springer-Verlag, 1990.

[35] Chaum, D. in E. van Heyst, *Group signatures*. V *Advances in Cryptology – Eurocrypt '91*, del 547 iz LNCS, str. 257–265. Springer-Verlag, 1991.

[36] Chen, L., Z. Cheng in N.P. Smart, *Identity-based key agreement protocols from pairings*. Int. J. Inf. Secur., 6(4):213–241, 2007.

[37] Chen, L. in C. Kudla, *Identity based authenticated key agreement protocols from pairings*. V *16th Computer Security Foundations Workshop*, str. 219–233. IEEE Computer Society Press, 2003.

[38] Chen, T.H., W.B. Lee in H.B. Chen, *A round- and computation-efficient three-party authenticated key exchange protocol*. J. Syst. Software, 81(9):1581–1590, 2008.

[39] Chen, Y., J.S. Chou in C.H. Huang, *Comment on four two-party authentication protocols.* Cryptology ePrint Archive, Report 2010/165, 2010.

[40] Choie, Y.J., E. Jeong in E. Lee, *Efficient identity-based authenticated key agreement protocol from pairings.* Appl. Math. Comput., 162(1):179–188, 2005.

[41] Choo, K.K.R., *Secure Key Establishment.* Springer-Verlag, 2008.

[42] Choo, K.K.R., C. Boyd in Y. Hitchcock, *On session key construction in provably-secure key establishment protocols.* V *Progress in Cryptology – Mycrypt 2005*, del 3715 iz LNCS, str. 116–131. Springer-Verlag, 2005.

[43] Cocks, C., *An identity based encryption scheme based on quadratic residues.* V *8th IMA Int. Conf. on Cryptography and Coding*, del 2260 iz LNCS, str. 360–363. Springer-Verlag, 2001.

[44] Cohen, H., G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen in F. Vercauteren, *Handbook of Elliptic and Hyperelliptic Curve Cryptography.* Discrete Mathematics and Its Applications. Taylor & Francis, 2005.

[45] Coron, J.S., *Optimal security proofs for PSS and other signature schemes.* V *Advances in Cryptology – Eurocrypt 2002*, del 2332 iz LNCS, str. 272–287. Springer-Verlag, 2002.

[46] Damgård, I.B., *A design principle for hash functions.* V *Advances in Cryptology – Crypto '89*, del 435 iz LNCS, str. 416–427. Springer-Verlag, 1990.

[47] Dierks, T. in E. Rescorla, *The transport layer security (TLS) protocol, version 1.2.* IETF (RFC 5246), avg. 2008.

[48] Diffie, W. in M.E. Hellman, *New directions in cryptography.* IEEE T. Inform. Theory, 22(6):644–654, 1976.

[49] Diffie, W., P.C. van Oorschot in M.J. Wiener, *Authentication and authenticated key exchanges*. Design. Code. Cryptogr., 2(2):107–125, 1992.

[50] ElGamal, T., *A public key cryptosystem and a signature scheme based on discrete logarithms*. V *Advances in Cryptology – Crypto '84*, del 196 iz LNCS, str. 10–18. Springer-Verlag, 1985.

[51] Fiat, A. in A. Shamir, *How to prove yourself: Practical solutions to identification and signature problems*. V *Advances in Cryptology – Crypto '86*, del 263 iz LNCS, str. 186–194. Springer-Verlag, 1987.

[52] FIPS 186, *Digital signature standard (DSS)*. NIST, maj 1994.

[53] Frey, G., M. Muller in H.G. Rück, *The Tate pairing and the discrete logarithm applied to elliptic curve cryptosystems*. IEEE T. Inform. Theory, 45(5):1717–1719, 1999.

[54] Frey, G. in H.G. Rück, *A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves*. Math. Comput., 62(206):865–874, 1994.

[55] Geer, G. van der, *Codes and elliptic curves*. V *Effective Methods in Algebraic Geometry*, del 94 iz Progress in Mathematics, str. 159–168. Birkhäuser Boston, 1991.

[56] Gentry, C. in R. Zulfikar, *Identity-based aggregate signatures*. V *Public Key Cryptography 2006*, del 3958 iz LNCS, str. 257–273. Springer-Verlag, 2006.

[57] Günther, C.G., *An identity-based key-exchange protocol*. V *Advances in Cryptology – Eurocrypt '89*, del 434 iz LNCS, str. 29–37. Springer-Verlag, 1990.

[58] Goh, E.J., S. Jarecki, J. Katz in N. Wang, *Efficient signature schemes with tight reductions to the Diffie-Hellman problems*. J. Cryptol., 20(4):493–514, 2007.

[59] Goldreich, O., S. Goldwasser in S. Micali, *How to construct random functions*. J. ACM, 33(4):792–807, 1986.

[60] Goldwasser, S. in M. Bellare, *Lecture notes in cryptography*, 2008. `http://cseweb.ucsd.edu/users/mihir/papers/gb.pdf`.

[61] Goldwasser, S. in S. Micali, *Probabilistic encryption & how to play mental poker keeping secret all partial information*. V *14h Symp. on Theory of Computing*, str. 365–377. ACM Press, 1982.

[62] Grasselli, J., *Elementarna teorija števil*. Št. 87 v *Knjižnica Sigma*. DMFA - založništvo, 2009.

[63] Hankerson, D., A.J. Menezes in S. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2004.

[64] Hanser, C. in D. Slamanig, *Blank digital signatures*. Cryptology ePrint Archive, Report 2013/130, 2013.

[65] Hardy, G.H. in E.M. Wright, *An Introduction to the Theory of Numbers*. Oxford University Press, 6. izd., 2008.

[66] Håstad, J., R. Impagliazzo, L.A. Levin in M. Luby, *A pseudorandom generator from any one-way function*. SIAM J. Comput., 28(4):1364–1396, 1999.

[67] Herranz, J., *Deterministic identity-based signatures for partial aggregation*. Comput. J., 49(3):322–330, 2006.

[68] Hess, F., *Efficient identity based signature schemes based on pairings*. V *Selected Areas in Cryptography*, del 2595 iz LNCS, str. 310–324. Springer-Verlag, 2003.

[69] Hölbl, M., *Development of identity-based authenticated key agreement protocols*. Doktorska disertacija, Fakulteta za elektrotehniko, računalništvo in informatiko, Univerza v Mariboru, 2009.

[70] Hölbl, M. in T. Welzer, *Two improved two-party identity-based authenticated key agreement protocols*. Comp. Stand. Inter., 31(6):1056–1060, 2009.

[71] Hölbl, M., T. Welzer in B. Brumen, *Two proposed identity-based three-party authenticated key agreement protocols from pairings.* Comput. Secur., 29(2):244–252, 2010.

[72] Hölbl, M., T. Welzer in B. Brumen, *An improved two-party identity-based authenticated key agreement protocol using pairings.* J. Comput. Syst. Sci., 78(1):142–150, 2012.

[73] Hsieh, B., H. Sun, T. Hwang in C. Lin, *An improvement of Saeednia's identity-based key exchange protocol.* V *Information Security Conference 2002*, str. 41–43, 2002.

[74] IEEE P1363, *Standard specifications for public key cryptography.* IEEE, jan. 2000.

[75] ISO/IEC 15946-3, *Information technology: Security techniques: Cryptographic techniques based on elliptic curves: Part 3 - key establishment.* ISO/IEC, dec. 2002.

[76] Joux, A., *A one round protocol for tripartite Diffie-Hellman.* V *4th Int. Symp. on Algorithmic Number Theory*, del 1838 iz LNCS, str. 385–393. Springer-Verlag, 2000.

[77] Joux, A. in K. Nguyen, *Separating decision Diffie–Hellman from computational Diffie–Hellman in cryptographic groups.* J. Cryptol., 16(4):239–247, 2003.

[78] Jurišić, A. in A.J. Menezes, *Elliptic curves and cryptography.* Dr. Dobb's J., str. 26–37, 1997.

[79] Just, M. in S. Vaudenay, *Authenticated multi-party key agreement.* V *Advances in Cryptology – Asiacrypt '96*, del 1163 iz LNCS, str. 36–49. Springer-Verlag, 1996.

[80] Kaliski Jr., B.S., *A pseudo-random bit generator based on elliptic logarithms.* V *Advances in Cryptology – Crypto '86*, del 263 iz LNCS, str. 84–103. Springer-Verlag, 1987.

[81] Kaliski Jr., B.S., *One-way permutations on elliptic curves.* J. Cryptol., 3(3):187–199, 1991.

[82] Katz, J. in Y. Lindell, *Introduction to modern cryptography.* Cryptography and Network Security Series. Chapman & Hall/CRC, 2007.

[83] Kent, S. in R. Atkinson, *IP encapsulating security payload (ESP).* IETF (RFC 2406), nov. 1998.

[84] Koblitz, N., *Elliptic curve cryptosystems.* Math. Comput., 48(177):203–209, 1987.

[85] Koblitz, N. in A.J. Menezes, *Another look at provable security.* `http://anotherlook.ca/`, obiskano 1. feb. 2014.

[86] Koblitz, N. in A.J. Menezes, *Another look at "provable security". II.* V *Progress in Cryptology – Indocrypt 2006*, del 4329 iz LNCS, str. 148–175. Springer-Verlag, 2006.

[87] Koblitz, N. in A.J. Menezes, *Another look at "provable security".* J. Cryptol., 20(1):3–37, 2007.

[88] Krawczyk, H., *HMQV: A high-performance secure Diffie-Hellman protocol.* V *Advances in Cryptology – Crypto 2005*, del 3621 iz LNCS, str. 546–566. Springer-Verlag, 2005.

[89] Krawczyk, H., *Cryptographic extraction and key derivation: The HKDF scheme.* V *Advances in Cryptology – Crypto 2010*, del 6223 iz LNCS, str. 631–648. Springer-Verlag, 2010.

[90] Krawczyk, H. in P. Eronen, *HMAC-based extract-and-expand key derivation function (HKDF).* IETF (RFC 5869), maj 2010.

[91] LaMacchia, B., K. Lauter in A. Mityagin, *Stronger security of authenticated key exchange.* V *1st Int. Conf. on Provable Security*, del 4784 iz LNCS, str. 1–16. Springer-Verlag, 2007.

[92] Lamport, L., *Constructing digital signatures from a one-way function.* Teh. por. CSL-98, SRI International, okt. 1979.

[93] Law, L., A.J. Menezes, M. Qu, J. Solinas in S. Vanstone, *An efficient protocol for authenticated key agreement.* Design. Code. Cryptogr., 28(2):119–134, 2003.

[94] Lenstra Jr., H.W., *Factoring integers with elliptic curves.* Ann. Math., 126(3):649–673, 1987.

[95] Lidl, R. in H. Niederreiter, *Introduction to Finite Fields and their Applications.* Cambridge University Press, 1994.

[96] Lim, M.H., S. Lee in H. Lee, *Cryptanalysis of improved one-round Lin-Li's tripartite key agreement protocol.* V *10th Int. Conf. on Advanced Communication Technology*, del 3, str. 1916–1921, 2008.

[97] Lim, M.H., S. Lee, Y. Park in H. Lee, *An enhanced one-round pairing-based tripartite authenticated key agreement protocol.* V *Computational Science and Its Applications 2007*, del 4706 iz LNCS, str. 503–513. Springer-Verlag, 2007.

[98] Lin, C.H. in H.H. Lin, *Secure one-round tripartite authenticated key agreement protocol from Weil pairing.* V *Advanced Information Networking and Applications 2005*, del 2, str. 155–168. IEEE Computer Society Press, 2005.

[99] Matsumoto, T., Y. Takashima in H. Imai, *On seeking smart public-key distribution systems.* Trans. IECE Japan, E69(2):99–106, 1986.

[100] McCullagh, N. in P.S.L.M. Barreto, *A new two-party identity-based authenticated key agreement.* V *Topics in Cryptology – CT-RSA 2005*, del 3376 iz LNCS, str. 262–274. Springer-Verlag, 2005.

[101] Menezes, A.J., *Elliptic Curve Public Key Cryptosystems.* Springer-Verlag, 1993.

[102] Menezes, A.J., I.F. Blake, X. Gao, R.C. Mullin, S. Vanstone in T. Yaghoobian, *Applications of Finite Fields.* Springer-Verlag, 1992.

[103] Menezes, A.J., M. Qu in S. Vanstone, *Some new key agreement protocols providing mutual implicit authentication.* V *Selected Areas in Cryptography*, str. 22–32. 1995.

[104] Menezes, A.J., S. Vanstone in T. Okamoto, *Reducing elliptic curve logarithms to logarithms in a finite field.* V *23rd Symp. on Theory of Computing*, str. 80–89. ACM Press, 1991.

[105] Menezes, A.J., S.A. Vanstone in P.C. van Oorschot, *Handbook of Applied Cryptography.* Discrete Mathematics and Its Applications. CRC Press, 1996.

[106] Merkle, R.C., *Secure communications over insecure channels.* Commun. ACM, 21(4):294–299, 1978.

[107] Merkle, R.C., *One way hash functions and DES.* V *Advances in Cryptology – Crypto '89*, del 435 iz LNCS, str. 428–446. Springer-Verlag, 1990.

[108] Miller, V.S., *Use of elliptic curves in cryptography.* V *Advances in Cryptology – Crypto '85*, del 218 iz LNCS, str. 417–426. Springer-Verlag, 1986.

[109] Miller, V.S., *The Weil pairing, and its efficient calculation.* J. Cryptol., 17(4):235–261, 2004.

[110] Naor, M. in O. Reingold, *Number-theoretic constructions of efficient pseudo-random functions.* V *38th Symp. on Foundations of Computer Science*, str. 458–467. IEEE Computer Society Press, 1997.

[111] Naor, M. in M. Yung, *Universal one-way hash functions and their cryptographic applications.* V *21st Symp. on Theory of Computing*, str. 33–43. ACM Press, 1989.

[112] Nose, P., *Security weaknesses of authenticated key agreement protocols.* Inform. Process. Lett., 111(14):687–696, 2011.

[113] Nose, P., *Improved deterministic identity-based signature scheme.* V pripravi, 2014.

[114] Nose, P., *Security weaknesses of a signature scheme and authenticated key agreement protocols.* Inform. Process. Lett., 114(3):107–115, 2014.

[115] Okamoto, E., *Proposal for identity-based key distribution systems.* Electron. Lett., 22(24):1283–1284, 1986.

[116] Okamoto, E., *Key distribution systems based on identification information.* V *Advances in Cryptology – Crypto '87*, del 293 iz LNCS, str. 194–202. Springer-Verlag, 1988.

[117] Okamoto, T., *Provably secure and practical identification schemes and corresponding signature schemes.* V *Advances in Cryptology – Crypto '92*, del 740 iz LNCS, str. 31–53. Springer-Verlag, 1993.

[118] Okamoto, T., *Authenticated key exchange and key encapsulation in the standard model.* V *Advances in Cryptology – Asiacrypt 2007*, del 4833 iz LNCS, str. 474–484. Springer-Verlag, 2007.

[119] Okamoto, T., *Authenticated key exchange and key encapsulation without random oracles.* Cryptology ePrint Archive, Report 2007/473, 2007.

[120] Okamoto, T. in D. Pointcheval, *The gap-problems: A new class of problems for the security of cryptographic schemes.* V *Public Key Cryptography 2001*, del 1992 iz LNCS, str. 104–118. Springer-Verlag, 2001.

[121] Pedersen, T.P. in B. Pfitzmann, *Fail-stop signatures.* SIAM J. Comput., 26(2):291–330, 1997.

[122] Plemelj, J., *Algebra in Teorija Števil.* Št. 6 v *Dela.* Slovenska akademija znanosti in umetnosti, 1962.

[123] Pohlig, S. in M. Hellman, *An improved algorithm for computing logarithms over $GF(p)$ and its cryptographic significance.* IEEE T. Inform. Theory, 24(1):106–110, 1978.

[124] Pointcheval, D. in J. Stern, *Security proofs for signature schemes.* V *Advances in Cryptology – Eurocrypt '96*, del 1070 iz LNCS, str. 387–398. Springer-Verlag, 1996.

[125] Pollard, J.M., *A Monte Carlo method for factorization.* BIT Numer. Math., 15(3):331–334, 1975.

[126] Pollard, J.M., *Monte Carlo methods for index computation mod p.* Math. Comput., 32(143):918–924, 1978.

[127] Pu, Q., X. Zhao in J. Ding, *Cryptanalysis of a three-party authenticated key exchange protocol using elliptic curve cryptography.* V *Research Challenges in Computer Science 2009*, str. 7–10. IEEE Computer Society Press, 2009.

[128] Rabin, M.O., *Digitalized signatures.* V *Foundations of Secure Computation*, str. 155–168. Academic Press, 1978.

[129] Rivest, R.L., A. Shamir in L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems.* Commun. ACM, 21(2):120–126, 1978.

[130] Rogaway, P. in T. Shrimpton, *Cryptographic hash-function basics: Definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance*.

[131] Sakai, R. in M. Kasahara, *ID based cryptosystems with pairing on elliptic curve.* Cryptology ePrint Archive, Report 2003/054, 2003.

[132] Schnorr, C.P., *Efficient identification and signatures for smart cards.* V *Advances in Cryptology – Crypto '89*, del 435 iz LNCS, str. 239–252. Springer-Verlag, 1990.

[133] Selvi, S.S.D., S.S. Vivek in C.P. Rangan, *Identity-based deterministic signature scheme without forking-lemma.* V *Advances in Information and Computer Security*, del 7038 iz LNCS, str. 79–95. Springer-Verlag, 2011.

[134] Selvi, S.S.D., S.S. Vivek in C.P. Rangan, *Deterministic identity based signature scheme and its application for aggregate signatures.* V *Information Security and Privacy*, del 7372 iz LNCS, str. 280–293. Springer-Verlag, 2012.

[135] Shamir, A., *How to share a secret.* Commun. ACM, 22(11):612–613, 1979.

[136] Shamir, A., *Identity-based cryptosystems and signature schemes.* V *Advances in Cryptology – Crypto '84*, del 196 iz LNCS, str. 47–53. Springer-Verlag, 1985.

[137] Shanks, D., *Class number, a theory of factorization, and genera.* V *Symposia in Pure Mathematics*, del 20, str. 415—440, 1971.

[138] Shim, K.A., *Cryptanalysis of two identity-based authenticated key agreement protocols.* IEEE Commun. Lett., 16(4):554–556, 2012.

[139] Shim, K.A. in S.H. Seo, *Cryptanalysis of ID-based authenticated key agreement protocols from bilinear pairings.* V *Information and Communications Security*, del 4307 iz LNCS, str. 410–419. Springer-Verlag, 2006.

[140] Shoup, V., *Lower bounds for discrete logarithms and related problems.* V *Advances in Cryptology – Eurocrypt '97*, del 1233 iz LNCS, str. 256–266. Springer-Verlag, 1997.

[141] Smart, N., *Identity-based authenticated key agreement protocol based on Weil pairing.* Electron. Lett., 38(13):630–632, 2002.

[142] Steiner, M., G. Tsudik in M. Waidner, *Refinement and extension of encrypted key exchange.* Oper. Syst. Rev., 29(3):22–30, 1995.

[143] Stinson, D.R., *Cryptography: Theory and Practice.* Discrete Mathematics and Its Applications. Taylor & Francis, 3. izd., 2005.

[144] Tan, Z., *An enhanced three-party authentication key exchange protocol for mobile commerce environments.* J. Commun., 5(5):436–443, 2010.

[145] Tanaka, K. in E. Okamoto, *Key distribution system for mail systems using ID-related information directory.* Comput. Secur., 10(1):25–33, 1991.

[146] Tate, J., *WC-groups over p-adic fields.* Séminaire N. Bourbaki, exposé 156, 4:265–277, 1958.

[147] Tseng, Y.M., *An efficient two-party identity-based key exchange protocol.* Informatica, 18(1):125–136, 2007.

[148] Tseng, Y.M., J.K. Jan in C. Wang, *Cryptanalysis and improvement of an identity-based key exchange protocol.* J. Computers, 14(3):17–22, 2002.

[149] Ustaoglu, B., *Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS.* Design. Code. Cryptogr., 46(3):329–342, 2008.

[150] Verheul, E.R., *Evidence that XTR is more secure than supersingular elliptic curve cryptosystems.* J. Cryptol., 17(4):277–296, 2004.

[151] Vidali, J., P. Nose in E. Pašalić, *Collisions for variants of the BLAKE hash function.* Inform. Process. Lett., 110(14-15):585–590, 2010.

[152] Vidav, I. in F. Lebedinec, *Algebra.* Matematika - fizika: Zbirka univerzitetnih učbenikov in monografij. DMFA - založništvo, 2003.

[153] Vidav, I. in M. Štalec, *Eliptične krivulje in eliptične funkcije.* Matematika - fizika: Zbirka univerzitetnih učbenikov in monografij. DMFA - založništvo, 1991.

[154] Weil, A., *Sur les fonctions algébriques à corps de constantes fini.* C. R. Acad. Sci. Paris, 210:592–594, 1940.

[155] Yang, J.H. in C.C. Chang, *An efficient three-party authenticated key exchange protocol using elliptic curve cryptography for mobile-commerce environments.* J. Syst. Software, 82(9):1497–1502, 2009.

[156] Ylonen, T. in C. Lonvick, *The secure shell (SSH) authentication protocol.* IETF (RFC 4252), jan. 2006.

Stvarno kazalo