KRÜPTOLOOGIA

Helger Lipmaa

26. juuni 2000. a.

Sisukord

Peatükk 1

Sissejuhatus

Kõige üldisemalt öeldes, krüptoloogia on teadus sellest kuidas tulla toime oma virtuaalse (digitaal-)eluga vaenlase olemasolul: kuidas vahetada teiste inimestega elektroonilisi teateid, ilma et vaenlane loeks talle mitte määratud teateid, või suudaks mõjutada sinu ja su sõprade otsuseid kui tal selleks õigust pole. Krüptoloogia on sisult matemaatiline teadus, kuid tema eesmärgid on praktilised. Jällegi veidi üldistades, võiksime krüptoloogiat defineerida kui abimeest infoühiskonda suundumisel, nii et meie tulevased digitaaltoimingud oleksid sama turvalised kui need mida me oleme aastasadu teostanud ilma Internetti ja arvuteid kasutamata.

Siinkohas on paslik tuua paar konkreetset näidet: krüptoloogias uuritakse hetkel aktiivselt *e-raha*, ehk sularaha elektroonilist ekvivalenti, ning *e-valimisi*, ehk valimisi üle Interneti. Mõlemal juhul on realistlik arvestada potentsiaalse vaenlase olemasoluga: esimesel juhul salakaval pankur Andres, kes soovib sinu raha enda arvele kanda, teisel juhul veelgi kavalam poliitik Edgar, kes sinu häält endale haarata himustab. Siiani on nii rahavahetus kui ka valimised olnud turvatud füüsiliste meetmetega (turvariba sajakroonisel, privaatsed valimiskabiinid), digitaalinformatsiooni vahetamisel kaotavad aga senised meetodid oma jõu.

Krüptograafilised meetodid erinevad 'tavameetoditest' eeskätt kuna digitaalinfo on lihtsalt kopeeritav ehk informatsioon on sõltumatu oma kandjast. Kui näiteks siiani tähendas signeerimine (allkirjastamine) sisuliselt paari

$$(\text{informatsioon}, \text{kandja})$$

signeerimist, sõltub digitaalsignatuuri väärtus vaid informatsioonist ning signeerijast. Ehk: digitaalmaailmas ei ole signatuur enam füüsiline ese (tükike tinti paberil), vaid bitistring, mis sõltub funktsionaalselt saadetavast teatest M ning signeerijast S.

Kuidas tagada seda, et digitaalekvivalendid on vähemalt sama turvalised kui vastavad füüsilised toimingud? Selleks tuleb defineerida 'turvalisus' antud toimingu (*protokolli* või *primitiivi*) kontekstis, ehk turvaeesmärk, mida antud protokoll peab rahuldama. Nii näiteks on signeerimise puhul turvaeesmärgiks vaenlase võimetus võltsida kellegi teise signatuure. Väga oluline on täielikult formaliseerida turvaeesmärk, ehk anda turvaeesmärgi matemaatiline definitsioon: ainult nii on võimalik tõestada, et mingi primitiiv on turvaline.

Krüptograafia tähtsust ei ole vaja ilmselt rõhutada. Enamus inimesi siiski ei tea, et justnimelt krüptograafia meetodite kasutamine Lääneriikide (eeskätt Suurbritannia ja USA) poolt — aga mitte kangelaslikud Nõukogude partisanid — otsustas määravalt Teise Maailmasõja saatuse. Juba tollal saadeti peaaegu 100% teateid edasi raadio teel. Et teha raskeks salajase informatsiooni pealtkuulamine, kasutasid erinevad sõdivad riigid erinevaid šifreid oma teadete kodeerimiseks. Lääneriikide spetsialistid (keda koordineeris Alan Turing) murdsid lahti sakslaste šifrimasina Enigma sel määral, et olid võimelised operatiivselt ennetama sakslaste olulisemaid lahinguaktsioone. Samas olid sakslased võimetud: nemad ei suutnud lahti murda Lääneriikide šifreid.

Tekib küsimus: 'kuidas mõõta võimetust?' Siin on mitu võimalikku lähenemisviisi, kõige standardsem on informatsiooniteoreetiline lähenemine, kus nõutakse et vaenlase jaoks peaks olema *põhimõtteliselt* võimatu väärata turvaeesmärki (ehk murda vastavat primitiivi/protokolli). Kahjuks on informatsiooniteoreetiliselt turvalised vaid mõned üksikud primitiivid, ja needki kipuvad olema väga ebapraktilised (näitena võiks tuua ühekordse numbrijadaga krüptimise, kus võtmepikkus on võrdne edastatava teksti pikkusega).

Järgmine samm on arvutuslik turvalisus: vaenlase jaoks peab olema *arvutuslikult* võimatu murda antud primitiivi. See võib tähendada näiteks vastava polünomiaalse algoritmi mitteleidumist (selle poole püüeldakse avaliku võtme krüptograafias). Kahjuks aga ei osata siiani tõestada, et selliseid algoritme ei ole: väga paljudel juhtudel osatakse primitiivi turvalisust taandada mingile eeldatavalt raskele arvutuslikule ülesandele (näiteks suurte arvude tegurdamisele), mille enda arvutuslik raskus pole aga siiani tõestatud.

Tihti on aga krüptograafide lähenemine hoopis maisem, seda eriti siis kui ei osata mingit *väga* vajalikku turvaeesmärki rahuldavaid primitiive koostada, või osatakse konstrueerida vaid ebaefektiivseid primitiive. Klassikaliseks näiteks võiks tuua plokkšifrid: ehkki tegemist on kõige kauem tuntud krüptograafia osana, ei ole siiani teada ühtegi efektiivset tõestatavalt turvalist plokkšifri konstruktsiooni. Samas on aga olemas kümneid väljapakutud plokkšifreid, mis on ülimalt efektiivsed ning mida ei osata murda (nii töötavad erinevad kaasaegsed plokkšifrid 300

MHz Pentium II peal kiirusega 50...180 Mbit/s).

Siit on lühike maa, et jõuda tänapäeva tüüpkrüptograafide tööpäeva kirjelduseni. Krüptograaf Ron pakub välja uue primitiivi (näiteks RC42), mille vastu pole teada ühtegi rünnet, ning mis on eelmistest mingi omaduse poolest parem (kiirem, või turvalisem teadaolevate rünnete vastu). Ron publitseerib RC42 kirjelduse mõnes respektaablis ajakirjas. Tema artiklit loeb krüptograaf Don, kes leiutab uue ründe nii RC42 kui ka mõne varem tuntud primitiivi vastu, ning publitseerib ründed sama ajakirja järgmises numbris. Ron ei heitu, vaid teeb oma primitiivi Doni ründe vastu turvalisemaks, publitseerides uue šifri RC42′ sama ajakirja jõulueris. Uut šifrit Don ei oska rünnata, ning seega standardiseeritakse see lähima kolme aasta jooksul, publitseeritakse kolm Interneti RFC-d, seatakse RC42′ uueks USA (vastava ala) primitiivistandardi kandidaadiks. Alles kuue aasta pärast tulevad uued krüptograafid, Bart ja Lars, ning näitavad et RC42′ on täiesti ebaturvaline uue, ennenägematult keerulise, ründe vastu. Nüüd on pall jälle Roni käes ...

Samas ei tööta sugugi kõik krüptograafid nii. Enamus neist baseerub eeldusel et mingi primitiiv (näiteks plokkšiffer) on "ideaalselt" (sõna "ideaalselt" sobiva definitsiooni korral) turvaline ning üritavad sellele üles ehitada uusi praktilise elu rakendusi, mis on samuti "ideaalselt" turvalised (sõna "ideaalselt" mõneti erineva definitsiooni korral). Näiteks võiks tuua nn *juhusliku oraakli* lähenemist, kus eeldatakse et protokollis kasutatav räsifunktsioon on juhuslik funktsioon (juhuslik oraakel), ning seejärel tõestatakse, et kogu protokoll ise on sellisel juhul turvaline. Pärast sellise tõestuse (reduktsiooni) andmist asendatakse juhuslik oraakel mõne konkreetse, kiire, räsifunktsiooniga ning eeldatakse et saadud kokteil on ka kokku turvaline: kõik ründed protokolli vastu võib nüüd taandada rünneteks räsifunktsiooni vastu.

Käesolev konspekt annab ülevaate tähtsamatest krüptograafilistest primitiividest ja protokollidest, sh tähtsamatest rakendustest, ning samuti ka krüptograafi tööpäevast. Muuhulgas kirjeldame nii seda, kuidas konstrueeritakse ja rünnatakse RC42′ laadseid primitiive kui ka seda, kuidas tõestada reduktsioonide abil protokollide turvalisust.

Konspekt eeldab Jan Willemsoni õppematerjali "Sissejuhatus krüptoloogiasse" mahus krüptoloogia tundmist, ning keskmisest paremaid eelteadmisi algebrast, keerukusteooriast ning arvuteooriast. Kursuse kuulajatel on kohustuslik kuulata (kui nad seda juba teinud pole) samaaegselt Valdis Laane kursust arvuteooriast.

Konspekt ise täieneb aja jooksul, täiustatud versioonid ilmuvad pidevalt lehele http://home.cyber.ee/helger/teaching/crypto2000.

1.1 Viited

1. Paul Johnson, “Modern Times. A History of the World from the 1920s to the 1990s”, 1992.

2. Phil Rogaway, “ECS 227 — Modern Cryptography”, 1999.

3. Jan Willemson, “Sissejuhatus krüptoloogiasse”, 1999. Kättesaadaval autori koduleheküljelt http://home.cyber.ee/jan

Peatükk 2

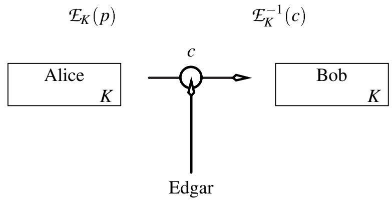
Plokkšifrid

2.1 Formaalne definitsioon. Ründed

Pöörame kõigepealt oma tähelepanu *plokkšifritele* (block cipher), mille käsitlemisega algavad kõik viimasel kümnel aastal välja antud krüptoloogiaõpikud. Selle peamiseks põhjuseks on nö "ajalooline tuntus", kuna ka kõige esimesed šifrid (permutatsiooni- ja substitutsioonišifrid) kuuluvad siia kategooriasse. Samuti on DES (peatükk 4.1), USA krüptograafiastandard just plokkšiffer; DES-i standardimine 70ndatel oli aga koos avaliku võtmega krüptograafia leiutamisele üks krüptoloogia kui akadeemilise teadusharu tekkimise peamisi põhjustajaid. DES-i ennast kui standardit on uuritud väga põhjalikult, aga tänu DES-i uurimisele on hakanud välja kujunema ka üldine plokkšifrite teooria.

Definitsioon 2.1.1 Fikseerime arvud b ja k — vastavalt ploki- ja võtmepikkuse. Plokkšiffer $\mathcal{E} : \{0,1\}^k \times \{0,1\}^b \to \{0,1\}^b$ on võtmega $K \in \{0,1\}^k$ indekseeritud permutatsioonide pere hulgal $\{0,1\}^b$. Kui arvud b ja k pole selged kontekstist, nimetame šifrit $\mathcal{E}$ ka (b,k)-plokkšifriks.

Nii siin kui hiljem nimetame sõnumit M *avatekstiks* ning selle krüptitud versiooni $\mathcal{E}(K,M)$ *krüptogrammiks*. On oluline aru saada, et plokkšifrite (ning üldse sümmeetrilise krüptograafia) puhul on võti K jagatud kahe või enama kõrgema taseme protokolliga määratud legitiimse osapoole vahel. Saatja rakendab mingile avatekstile $p \in \{0,1\}^b$ permutatsiooni $\mathcal{E}_K = \mathcal{E}(K,\cdot)$ ning edastab saadud tulemuse $c = \mathcal{E}_K(p)$ vastuvõtjale. Vastuvõtja rakendab väärtusele c pöördpermutatsiooni $\mathcal{E}_K^{1}$ ning saab tagasi algse suuruse p (joonis 2.1). Vaenlaseks loetakse suvalist osapoolt, kelle jaoks protokoll ei näe ette võtme omamist. Vaenlane võib olla pas-



Joonis 2.1: Kahe osapoole vaheline suhtlus vaenlase olemasolul

siivne (ainult pealt kuulata edastatavaid teateid) või aktiivne (lisaks olemasolevaid teateid muuta, uusi edastada, või mõningaid teateid kustutada).

Näide 2.1.1 Vaatleme nihkešifrit $\mathcal{E}$ parameetritega $k = b$ (seega on tähestikuks $\{0,1\}^8$ ehk kõik väärtused vahemikust $0\dots255$). Nihkešiffer kujutab paari (K,M) väärtuseks $K+M \mod 256$. Fikseeritud võtme K korral on ilmselgelt tegu hulga $\{0,1\}^8$ permutatsiooniga ning seega on $\mathcal{E}$ plokkšiffer.

Ülesanne 2.1.1 Miks on vajalik, et kujutused $\mathcal{E}(K,\cdot)$ oleksid permutatsioonid? Kui palju permutatsioone on hulgal $\{0,1\}^b$? Mitmebitine peaks olema võti k, et (b,k)-plokkšiffer hõlmaks endas kõiki antud hulga permutatsioone? Arvutage k väärtus $b = 64$ ja $b = 128$ puhul.

On selge, et (b,k)-plokkšiffer hõlmab endas ülimalt 2^k permutatsiooni. Permutatsioone võib olla vähem, kui erinevatele võtmetele vastavad identsed permutatsioonid, sh ka siis kui k väärtus on “liiga suur” plokipikkuse b jaoks. Üldjuhul on aga soovitav, et võtmepikkus oleks maksimaalselt väike, näiteks 128-bitine, et vähendada võtmehoidmisel ja -transportimisel tekkivaid kulusid.

Plokkšifrilt eeldatakse, et šifri “lahtimurdmiseks” kuluv aeg oleks 2^k. On selge, et selliseks “ideaalseks” turvalisuseks peab šifrisse $\mathcal{E}$ kuuluma 2^k erinevat permutatsiooni. Mitteformaalselt (me formaliseerime plokkšifrite turvalisuse mõiste hiljem peatükis 3) tähendab see seda, et võtme K juhuslikul valimisel saadud permutatsioone $\mathcal{E}(K,\cdot)$ ei ole “võimalik” eristada lihtsalt juhuslikult valitud permutatsioonidest P. Seega, võtmega $K \in \{0,1\}^k$ parametriseeritud permutatsioonide pere $\mathcal{E}$ peab välja nägema “juhuslik” (mitteformaalselt, $\mathcal{E}$ peab olema *pseudojuhuslik permutatsioon*, pseudorandom permutation ehk PRP).

Ülesanne 2.1.2 *Afiinse šifri* korral $K = (A,B)$, kus $B \in \{0,1\}^b$ ning $\gcd(A,2) = 1$. Krüptimisreegliks on $\mathcal{E}_{(A,B)}(M) := AM + B \mod 2^b$. Miks ei ole afiinne šiffer pseudojuhuslik permutatsioon?[1]

Kuna valitavate võtmete arv on kaduvväike võrreldes kõikide permutatsioonide arvuga, on sedalaadi “ideaalse” turvalisuse saavutamine plokkšifrite puhul ülimalt raske; kaasaegseid plokkšifreid disainitakse eelkõige turvaline olema juba tuntud rünnete vastu, mitte aga kõikvõimalike tulevikurünnete vastu. Sealjuures peavad plokkšifrid olema ka ülimalt efektiivsed, mis peamiselt tähendab et: 1) kõik permutatsioonid perest $\mathcal{E}$ peavad olema kiiresti arvutatavad, 2) permutatsiooni valik antud perest peab olema kiire. Kuna lisaks kõigele muule on oluline ka ruumiefektiivsus, nõutakse et eri permutatsioonidele vastaks võimalikult sarnane arvutiprogramm: kujutagem ette C-keelset programmi, milles vaid mõned üksikud osad on võtmest sõltuvad, kõik ülejäänu on aga võtmest sõltumatu.

Antud paragrahvi lõppu toome veel lühidalt tüüprünnete ja plokkšifri turvamudelite klassifikatsiooni. Esmalt, ütleme et plokkšiffer on *täielikult murtud*, kui mingi ründe abil on võimalik leida krüptimisvõti K. Plokkšiffer on *osaliselt murtud*, kui ründajal õnnestub leida vähemalt ühele krüptogrammile vastav avatekst.

Vaenlase tegevusvabadusele vastavad järgmised neli ründetüüpi:

1. *Teadaoleva avatekstiga rünne*. Vaenlasel on teada $n \geq 1$ sama võtmega krüptitud krüptogrammi $c_1, \ldots, c_n$. Sedatüüpi rünne õnnestub vaid siis, kui plokkšiffer on statistiliselt nõrk: krüptogrammide jaotusest saab tuletada vähese hulga arvutuste tagajärjel vastavad avatekstid.

2. *Teadaoleva krüptogrammiga rünne*. Vaenlasel on teada $n \geq 1$ paari (p_i, c_i), kus mingi (salajase) võtme K korral $c_i = \mathcal{E}(K, p_i)$. Mudel on realistlik, nii näiteks algavad kõik IP paketid “peaaegu” sama informatsiooniga, samuti paljud standardkirjad.

3. *Valitud avatekstiga ründe* puhul võib vaenlane ise valida $n \geq 1$ avateksti p_i ning omandada nendele vastavad krüptogrammid $c_i = \mathcal{E}_K(p_i)$. Tema ülesandeks on leida $\mathcal{E}_K(p)$ mingi $p \notin \{1, \ldots, n\}$ jaoks. Ka see mudel on realistlik, kuna paljud protokollid sisaldavad endast juhuslike väärtuste krüptimist. Lisaks võib eeldada, et vaenlane on ajutiselt saanud ligipääsu krüptivale tarkvara/riistvara jupile (näiteks kiipkaardile) nn “musta kasti režiimis,” kus ta saab mustale kastile ette anda suvalisi sisendeid, näeb nendele vastavaid väljundeid, kuid mitte seda mis on musta kasti sees (st võtit ennast).

[1] Soovitav vaadata Jan Villemsoni ainekonspekti!

Analoogiline on *valitud krüptogrammiga ründe* puhul saab vaenlane ise valida $n > 1$ krüptogrammi c_i ning omandada nendele vastavad avatekstid $p_i = \mathcal{E}_k(c_i)$.

4. *Adaptiivne valitud avatekstiga (krüptogrammiga) rünne*, kus valitud avatekstid/ krüptogrammid võivad sõltuda mustast kastist saadud eelnevatest väljunditest.

On selge, et iga järgmine ründetüüp siin on eelmise ründetüübi laienduseks. Kuigi näiteks adaptiivne valitud krüptogrammiga rünne on rakendatav vaid spetsiifiliste tingimuste täidetud olemisel, tuleb plokkšifri konstrueerimisel selle ründega arvestada.

Märgime mitteformaalselt (formaalne käsitlus tuleb hiljem), et ründe edukust parametriseeritakse selle ründe läbiviimiseks vajalike resurssidega: ründeaeg, ründe õnnestumise tõenäosus, valitud avatekstide arv n (valitud avatekstiga ründe puhul) jne.

Ülesanne 2.1.3 Oletagem, et käesoleval hetkel oleks rentaabel ehitada ASIC kiip, mis teostab "mõistliku" aja jooksul 2^{72} krüptimist šifriga $\mathcal{E}$. Millise ründe vältimiseks peaks plokipikkus olema vähemalt 144? Kui palju vajaks kujuteldav kiip mälu selle ründe läbiviimiseks juhul $b < 144$?

Ülesanne 2.1.4 Pakkuge välja paar lihtsat abinõud 1) teadaoleva avatekstiga ründe ja 2) teadaoleva krüptogrammiga ründe kirjelduses toodud situatsiooni ning ründe enda vältimiseks.

Ülesanne 2.1.5 Millistel tingimustel ja milliseid ründeid kasutades saab igat plokkšifrit lahti murda ajas 2^k? Pakkuge välja ideid, kuidas seda siiski vältida.

2.2 Plokkšifri töörežiimid

Eelmine paragrahv rääkis plokkšifritest, ehk ühest oluliseimast primitiivist keeruliste krüptograafiliste protokollide koostamisel. Käesolevas paragrahvis tutvustame meetodeid (*plokkšifri töörežiime*, block cipher mode), kuidas b-bitise plokipikkusega šifrit kasutades krüptida peaaegu piiramatu pikkusega bitistringe

$$p_1||p_2||\dots||p_n \quad ,$$

kus $p_i \in \{0,1\}^b$. Töörežiimide ohjeldamatu kasutamine on seletatav nende efektiivsusega: erinevates protokollides ja rakendustes saadetakse edasi väga erineva pikkusega bitistringe. On selge, et eraldi plokkšifri konstrueerimine iga erineva pikkusega stringi jaoks on nonsenss. Soovitav oleks omada konstruktsiooni, mille abil turvalisest (b,k)-plokkšifriks saaks konstrueerida turvalise meetodi suvalise nb-bitise, $n \in \mathbb{N}$, bitistringi krüptimiseks.

Esmalt kirjeldame ära üldise skeemi, mida järgivad kõik plokkšifrite töörežiimid. Üks osapool (rahvalegendidest tuntud ka kui Alice) soovib krüptida k-bitise võtmega K bitistringi $p = p_1||\ldots||p_n$, ning edastada krüptitud stringi $c = c_1||\ldots c_n$ teisele osapoolele, tuntud ka kui Bob (maskiga mees). Bob dekrüptib saadud stringi, kasutades sama võtit K. Nii krüptimine kui ka dekrüptimine toimuvad tavaliselt jadarežiimis, st et c_i väärtust peab olema võimalik välja arvutada väärtustest $(p_1,\ldots,p_i)$. Lisaks võib c_i arvutamisel kasutada mõnda avalikku (stringiga kaasa tulevat) konstanti iv $\in \{0,1\}^b$ (*initial value*, algväärtus). Seega, eeldame et mingite funktsiooniperede (f_i) ja (g_i) korral

$$\begin{aligned} c_i &= f_i(K,\mathrm{iv},p_1,\ldots,p_i) \ , \\ p_i &= g_i(K,\mathrm{iv},c_1,\ldots,c_i) \ . \end{aligned}$$

Kogu bitistring c on omakorda funktsioon võtmest K ning avatekstist p:

$$c = f(K,p) \ .$$

Analoogiliselt plokkšifritele, võime ka siinkohas vaadelda permutatsioonide peresid $f_{K,\mathrm{iv}}(\cdot) = f(K,\mathrm{iv},\cdot)$. Nagu ka plokkšifrite korral, on siingi turvaeelduseks, et permutatsioonide hulk f_K on vaenlase jaoks eristamatu kõikide permutatsioonide hulgast nb-bitistel stringidel. Järgnevalt tutvustame neid meetodeid, konspekti tulevikuosas toome formaalselt ära plokkšifrite töörežiimide turvaeesmärgid, ning tõestame nende meetodite turvalisuse määra juhul, kui allolev plokkšiffer on pseudojuhuslik permutatsioon.

2.2.1 Koodiraamat

Koodiraamat (*electronic codebook*, ECB) on ilmselt kõige lihtsam võimalik töörežiim. Siin $f_i(K,\mathrm{iv},p_1,\ldots,p_i) = \mathcal{E}_K(p_i)$, ehk iga plokk krüptitakse eraldi ning tulemused konkateneeritakse:

$$c = \mathcal{E}_K(p_1)||\mathcal{E}_K(p_2)||\ldots||\mathcal{E}_K(p_n) \ .$$

Dekrüptija tegevus on analoogiline, iga saadud plokk dekrüptitakse eraldi ning tulemused konkateneeritakse:

$$p = \mathcal{E}_K^{-1}(c_1)||\mathcal{E}_K^{-1}(c_2)||\ldots||\mathcal{E}_K^{-1}(c_n) \ .$$

Koodiraamatu režiimi omadused:

1. Identsed avatekstid: Avateksti identsed plokid krüptuvad sama võtme abil identseteks krüptogrammi plokkideks.

2. Ahelsõltuvused: plokid krüptitakse eraldi, sõltumatult teistest plokkidest. Avateksti plokkide ümberjärjestamisel järjestuvad vastavalt ümber krüptogrammi plokid.

3. Paralleelsus: nii krüptimist kui ka dekrüptimist saab täielikult paralleliseerida.

4. Vealevi: ühe või enama biti viga ühesainsas krüptogrammi plokis mõjutab ainult sellesama krüptogrammi dekrüptimist. "Ideaalse" šifri $\mathcal{E}$ korral dekrüptub vigane plokk juhuslikult (50% saadud avateksti bittidest on valed).

Kuna krüptogrammi plokid on sõltumatud, siis krüptogrammide plokkide kuritahtlikud asendused (näiteks tihti esineva ploki lisamised) ei mõjuta kõrvalolevate plokkide dekrüptimist. Kuna avateksti identsed plokid krüptuvad identseteks krüptogrammi plokkideks, siis ei soovitata koodiraamatu režiimi kasutada pikemate kui üheplokiste teadete saatmiseks (samuti ei soovitata koodiraamatu režiimis ühe võtmega saata rohkem kui ühte üheplokist teadet). Turvalisust saab mõnevõrra tõsta lisades igasse plokki mõned juhuslikud bitid.

2.2.2 Ahelrežiim

Ahelrežiimis (*cipher-block chaining*) töödeldakse bitistringi

$$p = p_1||\ldots||p_n$$

plokkhaaval, igal sammul kasutatakse võtit K, antud plokki p_i ning algväärtuse iv uuendatud väärtust:

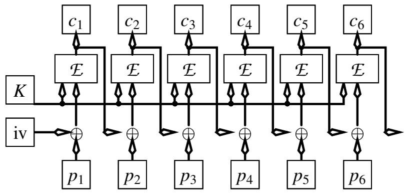
$$c_i = \mathcal{E}(K, p_i \oplus \mathrm{iv}) \ ,$$

ning seejärel omistatakse $\mathrm{iv} = c_i$ (joonis 2.2. Vastuvõtjale edastatakse muutuja iv esialgne versioon (enne krüptimise alustamist) ning string c. Vastuvõtja dekrüptib stringi c plokkhaaval:

$$p_i = \mathrm{iv} \oplus \mathcal{E}^{-1}(K, c_i)$$

ning omistab seejärel $\mathrm{iv} = c_i$.

Ahelrežiimi omadusi:



Joonis 2.2: Krüptimine ahelrežiimis

1. Identsed avatekstid: krüptogrammi identsete plokkide dekrüptimisel saadakse identne tulemus, kui krüptimine toimub sama võtme ja algväärtusega. Algväärtuse, võtme või avateksti esimese ploki muutmisel saadakse erinev krüptogramm.

2. Ahelsõltuvused: ahelrežiimis sõltub c_i p_i-st ning kõigist sellele eelnenud avateksti plokkidest (kõikidest eelnenud plokkidest sõltuvus on küll 'peidetud" sõltuvusse krüptogrammi eelnevast plokist). Seega, krüptogrammi plokkide ümber järjestamine mõjutab oluliselt dekrüptimise tulemust; krüptogrammi korrektse ploki õige dekrüptimine eeldab krüptogrammi eelnenud ploki korrektsust.

3. Vealevi: ühebitine viga krüptogrammi plokis c_i mõjutab plokkide c_i ja c_{i+1} dekrüptimist (kuna p_i sõltub c_i-st ning c_{i-1}-st). Vigasest krüptogrammist $c_i^* := c_i \oplus e_s$ dekrüptimisel saadud avateksti plokk p_i^* on ideaalse plokkšifri korral täiesti juhuslik (50% bittidest vigased), kuid samas $p_{i+1}^* = p_{i+1} \oplus e_s$. Seega muutes vajalikke bitte c_i-s, saavutab vastane p_{i+j}-s ennustatavad bitimuutused.

4. Veaparandus: CBC režiim on isesünkroniseeruv. Vea korral, mis leiab aset plokis c_i (k.a. ploki kaotsiminek), kuid mitte plokis c_{i+1} dekrüptub c_{i+2} korrektselt x_{i+2}-ks. Samas, avateksti x_i bittide muutmine muudab krüptogrammi kõiki järgnevaid plokke (see mõjutab ahelrežiimi kasutatavust rakendustes, milles läheb vaja juhupöördust andmete poole).

Märkus 2.2.1 Plokkide piire muutvat bittide kaotust ei võimalda sünkroniseerida ükski töörežiim.

Algväärtus iv ei pea olema salajane, kuid tema autentsus peab olema tagatud: iv muutmine võimaldab vastasel teha dekrüptitud avateksti plokis ennustatavaid muudatusi. Autentsuse tagamiseks tuleb kasutada vastavaid krüptograafilisi primitiive.

Ülesanne 2.2.1 Pakkuge välja vastav rünne.

Algväärtus genereeritakse juhuslikult, kuid ka see ei vajalik. Sama turvalisuse astme saab, kui kasutada eelmise paketi krüptimisel kasutatud muutuja iv lõppväärtust järgmise paketi krüptimisel. Põhjuseks on juhuarvude genereerimise raskus.

2.2.3 Loendurrežiim

Loendurrežiimis töötab algväärtus loendurina: selle väärtust suurendatakse ühe võrra iga järgneva ploki juures. Seega, i-ndal sammul tehakse järgmised kaks sammu:

$$\begin{aligned} c_i &= \mathcal{E}(K, \mathrm{iv}) \oplus p_i \quad , \\ \mathrm{iv} &= \mathrm{iv} + 1 \quad . \end{aligned}$$

Saatja edastab seejärel vastuvõtjale loenduri algväärtuse iv koos stringiga c. Vastuvõtja rakendab täpselt analoogilist protsessi avateksti p kättesaamiseks:

$$\begin{aligned} p_i &= \mathcal{E}(K, \mathrm{iv}) \oplus c_i \quad , \\ \mathrm{iv} &= \mathrm{iv} + 1 \quad . \end{aligned}$$

Loendurrežiimi headeks omadusteks on:

1. Võimalus teha eelkalkulatsiooni: väärtused $E_k(\mathrm{iv})$ saab välja arvutada sobival ajal, tegeliku krüptimise juures võib eelnevalt valmis pandud krüptogrammid c_i' mälust lihtsalt ekstraheerida ning teostada elementaarne xor-tehe $c_i := c_i' \oplus p_i$.

2. Nii eelkalkulatsiooniga kui ka ilma on loendurrežiim ideaalselt paralleliseeritav: suvalise paralleelsusastmega $n \geq 1$ arvuti võib paralleelselt krüptida suurused iv, $\mathrm{iv} + 1, \ldots, \mathrm{iv} + n - 1$ ning seejärel paralleelselt xor'ida saadud tulemused avateksti n järjestikkuse plokiga.

3. Loendurrežiim ei nõua, et $\mathcal{E}(k, \cdot)$ oleks pööratav. See tähendab, et tegu ei pea olema permutatsiooniga, piisab ka "piisavalt heast" pseudojuhuslikust *funktsioonist*.

Kui plokkšiffer $\mathcal{E}$ on ise nõrk, võib loendurrežiim olla ebaturvalisem kui ahelrežiim: põhjuseks on väike Hammingi kaugus suuruste iv ja $\mathrm{iv}+1$ vahel. Algväärtust ei pea valima juhuslikult: tavaliselt suurendab iga osavõtja oma personaalloenduri väärtust ühe võrra iga saadetud ploki korral. Sellisel juhul oleks siiski kasulik kõige esimese ploki loendurit juhuslikult valida, kuna vastasel juhul saab vaenlane triviaalselt teada, mitu plokki konkreetne saatja on oma elutegevuse jooksul krüptinud. See aga on nii mõnelgi juhul mittesoovitav.

Ülesanne 2.2.2 Vaatleme juhtu, kus osavõtja saadab ühe paketi korraga, ning valib iga kord loenduri juhuslikult. Mis on sellise talitlusviisi nõrkusteks võrreldes loenduri lihtsa suurendamisega? [efektiivsus, turvalisus]

Millega võib inkrementeerimise loendurrežiimis asendada, et selliseid nõrkusi ei esineks?

Peatükk 3

Pseudojuhuslikud funktsioonid

3.1 Funktsioonide ja permutatsioonide pered

Käesolevas peatükis vaatleme plokkšifreid formaalselt, defineerides “ideaalse” plokkšifri vastena *pseudojuhuslikud permutatsioonid* ning viimastega lähedases suguluses olevad *pseudojuhuslikud funktsioonid*.

Definitsioon 3.1.1 *Funktsioonide pere* on kujutus

$$F : \mathrm{Keys}(F) \times \mathrm{Dom}(F) \to \mathrm{Range}(F) \enspace ,$$

kus $\mathrm{Keys}(F)$ on F-i võtmete hulk, $\mathrm{Dom}(F)$ — originaal ning $\mathrm{Range}(F)$ — kujutis. Funktsioonide pere F väärtust kohal (K,p) tähistatakse kui $F(K,p)$. Iga võtme K jaoks võime vaadelda kujutust $F_K = F(K,\cdot)$, seega on F kujutuste hulk, kus igale võtmele vastab üks kujutus.

Tavaliselt $\mathrm{Keys}(F) = \{0,1\}^k$, $\mathrm{Dom}(F) = \{0,1\}^\ell$ ja $\mathrm{Range}(F) = \{0,1\}^L$ mingite täisarvude k, ℓ ja L jaoks. Võtmehulgaga K seotakse tavaliselt mingit tõenäosusjaotust. Sümboliga $K \stackrel{R}{\leftarrow} \mathrm{Keys}(F)$ tähistame juhusliku võtme valikut hulgast $\mathrm{Keys}(F)$ vastava tõenäosusjaotuse järgi. Sümboliga $f \stackrel{R}{\leftarrow} F$ tähistame järgmise kahe operatsiooni järjestikust täitmist: $K \stackrel{R}{\leftarrow} \mathrm{Keys}(F)$; $f := F_K$, sellisel juhul on f võrdne kujutusega F_K juhuslikult valitud võtme K korral.

Definitsioon 3.1.2 Kujutus $f : D \to R$ on *permutatsioon*, kui 1) $D = R$ ning 2) f on bijektsioon. Funktsioonide pere F on *permutatsioonide pere* kui suvalise K korral F_K on permutatsioon.

Tavaliselt eeldatakse, et võtmeid valitakse ühtlase jaotuse järgi, sel juhul vastab võtme K valikule juhusliku k-bitise stringi valik. Nagu eelnevast teame, on plokkšiffer permutatsioonide pere.

Edasises mängivad otsustavat rolli järgmised kaks funktsioonide peret: $\mathrm{Func}_{\ell,L}$ — kõigi funktsioonide $f : \{0,1\}^{\ell} \to \{0,1\}^{L}$ pere, ning Perm_{ℓ} — kõigi hulga $\{0,1\}^{\ell}$ permutatsioonide pere. Esimese pere juhuslikult valitud elementi nimetatakse ka *juhuslikuks funktsiooniks*, teise pere juhuslikult valitud elementi — *juhuslikuks permutatsiooniks*.

Vaatleme järgnevalt lähemalt hulki $\mathrm{Func}_{\ell,L}$ ning Perm_{ℓ}. Juhusliku funktsiooni

$$f = \mathrm{Func}_{\ell,L}(K,\cdot)$$

võtmeks on f-i täielik kirjeldus. Seega,

$$\mathrm{Keys}(\mathrm{Func}_{\ell,L}) = \{\vec{y} : |y| = 2^{\ell} \wedge y_i \in \{0,1\}^{L}\} \ ,$$

ning $|\mathrm{Keys}(\mathrm{Func}_{\ell,L})| = 2^{L2^{\ell}}$. Kui interpreteerida elementi $x \in \{0,1\}^{\ell}$ kui täisarvu vahemikust $[1,2^{\ell}]$, siis saame, et $\mathrm{Func}_{\ell,L}(\vec{y},x) = y_x$. Kuna võtmeruum on varustatud ühtlase jaotusega, siis tähistab $f \xleftarrow{R} \mathrm{Func}_{\ell,L}$ juhusliku kujutuse valikut ℓ-bitiste stringide hulgast L-bitiste stringide hulka.

Analoogiliselt, permutatsioonide pere Perm_{ℓ} võtmeruumiks on

$$\mathrm{Keys}(\mathrm{Perm}_{\ell}) = \{\vec{y} : |y| = 2^{\ell} \wedge y_i \in \{0,1\}^{L} \wedge (i \neq j \Rightarrow y_i \neq y_j)\} \ ,$$

ning $f \xleftarrow{R} \mathrm{Perm}_{\ell}$ tähistab juhusliku ℓ-bitiste stringide permutatsiooni valikut.

Ülesanne 3.1.1 Kui mitu elementi on hulgas $\mathrm{Keys}(\mathrm{Perm}_{\ell})$?

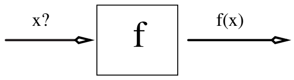
Definitsioon 3.1.3 Ülaltoodud peresid nimetatakse ka *lõplikeks*. *Lõpmatu funktsioonide pere* on jada $F = \{F^n\}_{n\geq 1}$, kus F^n on lõplik funktsioonide pere

$$F^n : \mathrm{Keys}(F^n) \times \mathrm{Dom}(F^n) \to \mathrm{Range}(F^n) \ ,$$

kus harilikult $\mathrm{Keys}(F^n) = \{0,1\}^{k(n)}$, $\mathrm{Dom}(F^n) = \{0,1\}^{\ell(n)}$ ning

$$\mathrm{Range}(F^n) = \{0,1\}^{L(n)}$$

mingite kasvavate funktsioonide k, ℓ ning L jaoks. Suurust n nimetatakse ka *turvaparameetriks*.



Joonis 3.1: Musta kasti mudel

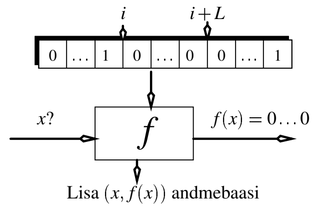
3.2 Juhuslikud funktsioonid ja permutatsioonid

Võibolla kõik tudengid ei saanud veel aru, mis on juhuslik funktsioon. Seletame siis lähemalt. Me defineerisime eelnevas juhuslikud funktsioonid kui hulgast $\mathrm{Func}_{\ell,L}$ juhuslikulkt valitud elemendid. Kuna iga selline funktsioon on identifitseeritud oma võtmega ehk oma väärtustabeliga (graafikuga), tähendab kujutuse f juhuslik valik selle kujutuse graafiku juhuslikku valikut, mis omakorda tähendab kujutuse f erinevate väärtuste $f(i)$ teineteisest sõltumatut valikut hulgast $\{0,1\}^L$. Viimane tähendab aga seda, et juhusliku funktsiooni korral väärtused $f(i)$, $i \neq j$, on sõltumatud väärtusest $f(j)$: ehk väärtuse $f(j)$ mõõtmisel ei saada mingisugust informatsiooni väärtuse $f(i)$ kohta.

Toodud argumentatsioon peaks selgitama, miks juhuslikke funktsioone vaadeldakse tihti "musta kasti" (ehk *oraakel*-) mudelis. Mainitud mudelis eeldatakse, et mingi kujutus (mis ei pea sugugi olema juhuslik funktsioon) f on paigutatud musta kasti, millele saab sisendiks anda suvalise väärtuse $x \in \{0,1\}^\ell$, ning mis seejärel tagastab väärtuse $f(x) \in \{0,1\}^L$. Eeldatakse, et ainuke viis kujutuse f kohta informatsiooni saada ongi sellise *päringu* (joonis 3.1) esitamine. Juhusliku funktsiooni korral tähendab musta kasti mudel seda, et valitakse kujutus f juhuslikult hulgast $\mathrm{Func}_{\ell,L}$ ning paigutatakse see musta kasti. Ainuke võimalus kujutuse f kohta informatsiooni saamiseks on esitada päringuid kujul x?. Kuna f on juhuslik funktsioon, tähendab see et iga järgmise päringu korral saadakse mustast kastist uus, juhuslikult genereeritud, L-bitine string. Ainsaks nõudeks on see, et kui päring x oli juba korra mustale kastile esitatud, peab kasti vastus $f(x)$ olema sama, mis eelmisel korral (f on deterministlik funktsioon!).

Seega, juhuslikku funktsiooni $f : \{0,1\}^\ell \to \{0,1\}^L$ võib alternatiivselt defineerida kui kasti, mis peab meeles andmebaasi paaridest (x,y) ning mis päringul x tagastab elemendi y, kui andmebaasis leidub paar (x,y) või vastasel juhul (kui andmebaasis ei ole paari $(x,\star)$, suvalise $\star$ jaoks) genereerib juhusliku L-bitise stringi y, tagastab selle, ning seejärel lisab paari (x,y) andmebaasi.

Edasises on väga oluline aru saada, mida juhuslik funktsioon endast õieti kujutab, seetõttu soovitame tudengil uuesti läbi mõelda omandatud teadmus, ning seetõttu toome siinjärel koheselt veel ühe juhusliku funktsiooni definitsiooni. Nimelt,



Joonis 3.2: Musta kasti mudel: bitistringiga juhtum

juhuslikku funktsiooni võib vaadelda musta kastina, millele on kaasa antud juhuslikult valitud bitistring $K \overset{R}{\leftarrow} \{0,1\}^{L2^\ell}$ (või üldiselt string $K \overset{R}{\leftarrow} \{0,1\}^\infty$). Saadud must kast peab meeles loendurit i (algselt null) ning andmebaasi (algselt tühi). Päringul x on mustal kastil kaks valikut (joonis 3.2):

1. Kui (x,y) on andmebaasis mingi y korral, siis tagasta y.
2. Vastasel juhul: $y := K[i \dots i+L]$; $i := i+L$; lisa (x,y) andmebaasi; tagasta y.

Üks neist kolmest mudelist peaks tudengile mõistetav olema. Märgime, et juhuslikku funktsiooni kutsutakse tihti ka *juhuslikuks oraakliks*. Kui ℓ ja L on kontekstist selged, tähistatakse hulka $\mathrm{Func}_{\ell,L}$ ka sümboliga $\{0,1\}^\infty = 2^\infty$.

Lemma 3.2.1 $\mathrm{Prob}[f(x) = y : f \overset{R}{\leftarrow} \mathrm{Func}_{\ell,L}] = 2^{-L}$.

Tõestus. Juhusliku funktsiooni f väärtustamine kohal x tähendab, et must kast loeb lindilt L-bitise juhusliku stringi. Tõenäosus, et see string on võrdne stringiga y on 2^{-L}. □

Ülesanne 3.2.1 Tõestada, et

$$\mathrm{Prob}[f(x_1) = f(x_2) = y : f \overset{R}{\leftarrow} \mathrm{Func}_{\ell,L}] = \begin{cases} 2^{-2L} & \text{kui } x_1 \neq x_2 \ , \\ 2^{-L} & \text{kui } x_1 = x_2 \ . \end{cases}$$

Tõestada, et

$$\mathrm{Prob}[f(x_1) \oplus f(x_2) = y : f \overset{R}{\leftarrow} \mathrm{Func}_{\ell,L}] = \begin{cases} 2^{-L} & \text{kui } x_1 \neq x_2 \ , \\ 0 & \text{kui } x_1 = x_2 \text{ ja } y \neq 0^L \ , \\ 1 & \text{kui } x_1 = x_2 \text{ ja } y = 0^L \ . \end{cases}$$

3.3 Pseudojuhuslikud funktsioonid: motivatsioon

Enne pseudojuhuslike funktsioonide formaalse definitsiooni andmist üritame esmalt lühidalt motiveerida nende vajalikkust. Sümmeetrilises (=salajase võtmega) krüptograafias jagavad kaks osapoolt võtit mida vaenlane ei tea. Jagatud salajast võtit kasutakse erinevateks otstarveteks, sh krüptimiseks (salastuseks) ning autentimiseks.

Jagatud juhusliku funktsiooni mudelis on jagatud võtmeks juhusliku funktsiooni kirjeldus, pikkusega $2^{L2^{\ell}}$ bitti. Suuruste L ja ℓ praktiliste väärtuste juures on selline võti liiga pikk isegi selleks, et seda salvestada. Seega on tegemist pigem kontseptsionaalse, kuid nagu hiljem näeme, väga kasuliku mudeliga. Nimelt on jagatud juhusliku funktsiooni mudelis väga mugav uusi krüptograafilisi skeeme defineerida ning neid analüüsida, kus eeldatakse et mingi skeemis kasutatud sümmetriline primitiiv on ideaalne (ehk juhuslik funktsioon) ning tõestatakse et sel juhul ka kogu skeem on ideaalselt turvaline. Märgime, et jagatud juhusliku funktsiooni mudelis on turvatõestused absoluutsed: vaenlase arvutuslikule võimsusele ei säeta mingeid tõkkeid; turvatõestus annab vaenlase õnnestumistõenäosusele lihtsalt ülemtõkke.

Kuid mis kasu on saadud turvatõestusest, kui skeem on liiga ebapraktiline selle realiseerimiseks? Siinkohas tulevadki mängu *pseudo*juhuslike funktsioonide ja permutatsioonide pered: sellised funktsioonide pered F, mille korral võtmeruum $\mathrm{Keys}(F)$ on väike (näiteks võimsusega $\leq 2^{128}$). Lisaks nõutakse neilt funktsioonidelt, et kui võti $K \in \mathrm{Keys}(F)$ on jagatud kahe osapoole vahel, ning seejärel kasutatakse funktsiooni F_K juhusliku funktsiooni f asemel mõnes jagatud juhusliku funktsiooni mudelis disainitud turvalises skeemis, on resulteeruv skeem endiselt turvaline *eeldusel*, et vaenlase arvutusvõimsus on piiratud. Seega, pseudojuhuslik funktsioon on juhusliku funktsiooni praktiline analoog: praktika vajaduste tõttu eeldatakse, et kasutatavad võtmed on lühikesed, selle nimel ohverdatakse absoluutne turvalisus ning räägitakse vaid *arvutuslikust turvalisusest*. Nagu näeme hiljem selles kursuses, on absoluutne (informatsiooniteoreetiline) turvalisus saavutatav vaid väga pikkade võtmete korral: seetõttu tegeleb suurem osa krüptograafiast arvutuslikult turvaliste skeemidega.

Märgime, et on krüptograafilisi skeeme, mille turvalisust saab tõestada jagatud juhusliku funktsiooni mudelis, kuid mis muutuvad ebaturvaliseks kui juhuslik funktsioon asendada pseudojuhuslikuga. Samas enamikel juhtudel selline asendus toimub. Krüptograafilises praktikas tõestatakse esmalt, et mingi skeem on turvaline jagatava juhusliku funktsiooni mudelis ning seejäre, et sama skeem jääb turvaliseks ka pärast kasutatud juhuslike funktsioonide asendamist pseudojuhuslikega.

On üks oluline piirang: avalikult teadaolevaid juhuslikke funktsioone ei saa asendada pseudojuhuslike funktsioonidega; pseudojuhuslikud funktsioonid on kasulikud vaid olukorras kus võti K on salajane: vaenlane seda võtit ei tea, ning ei saa funktsiooni ennast ka otseselt välja arvutada.

Pseudojuhuslike funktsioonide ja permutatsioonide üheks esmaseks motivatsiooniks on plokkšifrite modelleerimine ning seeläbi plokkšifreid kasutavate protokollide turvaanalüüsi. Klassikaliselt käsitletakse plokkšifri turvalisust piiratud ulatuses: näiteks öeldakse, et plokkšiffer F on turvaline, kui teades vaid väärtusi $(x_1, F_K(x_1))$, ..., $(x_q, F_K(x_q))$, ei leidu kiiret algoritmi, mis arvutaks välja võtme K.

On selge, et selline turvadefinitsioon ei ole täiuslik: on kindlasti võimalik ette kujutada olukorda, kus sellist algoritmi ei leidu, aga leidub teine algoritm mis sisenditel x, $F_K(x)$ ning $F_K(x+1)$ leiab efektiivselt suuruse $F_K(x+2)$. On selge, et sellise algoritmi leidumise korral oleks antud plokkšiffer täielikult ebaturvaline loendurrežiimis. Võiksime siis defineerida, et plokkšiffer on turvaline kui ka viimasena mainitud algoritme ei leidu.

Samas võib ette kujutada, et plokkšifri kasutamiseks (näiteks) ahelrežiimis on vajalik veel mingite turvatingimuste täidetus. Nii võiksime aga turvatingimusi muudkui lisada ja lisada, ning plokkšifrite formaalne definitsioon muutuks järjest keerulisemaks ja keerulisemaks — muuhulgas muutuks hõlmamatult keeruliseks ka erinevate turvatõestuste andmine. On selge, et palju eelistatavam on omada lühikest ja lihtsat turvadefinitsiooni, mis võimaldaks anda võimalikult lihtsaid turvatõestusi. Üheks ilusaks plokkšifreid modelleerivaks vahendiks ongi pseudojuhuslikud funktsioonid ja permutatsioonid.

3.4 PRF-d ja PRP-d

Seega on pseudojuhuslike funktsioonide (PRF-d) ja permutatsioonide (PRP-d) põhiliseks motivatsiooniks nende kasutamine ebapraktiliste juhuslike funktsioonide asemel krüptograafilistes skeemides kus osapooled jagavad lühikest salajast võtit K ning funktsiooni $f = F_K$ nii et olemasolev skeem oleks endiselt “turvaline” suvalise arvutuslikult piiratud vaenlase vastu. Sellist turvalisust kõikidel või enamus juhtudes on võimalik saavutada siis, kui arvutuslikult piiratud vaenlane ei ole võimeline aru saama, kas antud skeemis on kasutatud juhuslikku või pseudojuhuslikku funktsiooni kui funktsioonile ligipääs on lubatud vaid musta kasti (oraakli) mudelis.

3.4.1 Pseudojuhuslikud funktsioonid

Asume järgnevas formaalse definitsiooni andmise poole. Olgu

$$F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L \ .$$

PRF-ide korral vaadeldakse järgnevat kahte maailma:
Maailm 0. Kasutajale antakse musta kastina ette hulgast $\mathrm{Func}_{\ell,L}$ juhuslikult valitud funktsioon f: $f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,L}$. Seega f on lihtsalt juhuslik funktsioon.
Maailm 1. Kasutajale antakse musta kastina ette funktsioon $f \stackrel{R}{\leftarrow} F$. Seega sellisel juhul valitakse juhuslikult võti K ning omistatakse seejärel $f = F_K$. (Võtit K ei anta kasutajale!)

Seejärel antakse kasutajale ette üks kahest maailmast (mustast kastist), ning küsitakse, kumbas maailmas ta on. Vastamise jooksul on kasutajale lubatud mustale kastile esitada päringuid enda valitud kohtadel (see on ainukeseks lubatud meetodiks f-i kohta informatsiooni saamiseks!). Mingi arvu q päringute järel teeb kasutaja oletuse b maailma kohta. Pseudojuhusliku pere F kvaliteeti mõõdetakse kasutaja raskusega ülaltoodud mängus korrektselt ära arvata Maailma b.

Seega, intuitiivselt modelleerib toodud mäng funktsiooni f kasutamist erinevates rakendustes. Kui vaenlasel pole võimalik eristada pere F juhuslikult valitud liikme sisend-väljund käitumist juhusliku funktsiooni käitumisest, käitub tema jaoks ka kogu rakendus ligikaudu samal viisil, olenemata sellest kas $f \stackrel{R}{\leftarrow} F$ või $f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,L}$. Nagu eelpool mainitud, ei ole sellist asendust võimalik alati teha: pseudojuhuslikke funktsioone saab kasutada enamikes, kuid mitte kõikides rakendustes.

Ülaltoodud mängus korrektset maailma oletavat algoritmi formaliseeritakse *eraldaja* (distinguisher) mõistega. Eraldajale antakse oraakelpääs kujutusele f ning ta üritab seejärel aru saada, kas f on juhuslik või pseudojuhuslik. Intuitiivselt, pere F on pseudojuhuslik kui tõenäosus et eraldaja vastab "0" on ligikaudu võrdne, sõltumatult sellest, kummas maailmas ta tegelikult on.

Definitsioon 3.4.1 Olgu

$$F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^L$$

funktsioonide pere, ning olgu D algoritm (*eraldaja*), mis võib teha oraakelpäringuid kujutusele

$$f : \{0,1\}^\ell \to \{0,1\}^L$$

ning väljastab seejärel nulli või ühe. Suurust

$$\mathrm{Adv}_F^{\mathrm{prf}}(D) := |\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} F] - \mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,L}]|$$

nimetatakse eraldaja *eduks* (*advantage*). Olgu

$$\mathrm{InSec}_F^{\mathrm{prf}}(q,t) = \max_D \mathrm{Adv}_F^{\mathrm{prf}}(D) \ ,$$

kus maksimum on võetud üle algoritmide D mis töötavad ajas t ja teevad ülimalt q musta kasti päringut. Funktsioonide peret F nimetatakse (q,t,ε)-*turvaliseks pseudojuhuslike funktsioonide pereks*, kui $\mathrm{InSec}_F^{\mathrm{prf}}(q,t) \leq \varepsilon$.

Turvalisuse juures on oluline vaid suuruse $\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} F] - \mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,L}]$ absoluutväärtus: kui leidub algoritm, mille puhul see suurus on $\varepsilon < 0$, siis selle algoritmi väljundite ümberpööramine annab triviaalselt algoritmi eduga $-e > 0$. Seega, algoritmid, mille puhul see suurus on lähedane -1-le, on sama edukad kui algoritmid, mille edu on lähedane 1-le: nad väljastavad vaid (meelega?!) valesid vastuseid.

Näide 3.4.1 Nihkešifrite pere $F = \{f : f(x) \equiv x+K \mod 2^\ell\}$ ei ole pseudojuhuslik funktsioon, kuna seda saab juhuslikust funktsioonist eraldada järgneva lihtsa algoritmi D^f abil:

1. $x \stackrel{R}{\leftarrow} \{0,1\}^\ell$; $K := f(x) - x$.
2. Korrata $q-1$ korda:
   $x \stackrel{R}{\leftarrow} \{0,1\}^\ell$.
   Kui $f(x) \neq x+K$ siis vasta 0.
3. Vasta 1.

On selge, et $\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} F] = 1$, samas juba juhul $q = 1$

$$\begin{aligned}
\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,\ell}] &= \mathrm{Prob}[f(x) - x = f(y) - y : x,y \stackrel{R}{\leftarrow} \{0,1\}^\ell] \\
&= \mathrm{Prob}[f(x) - x = f(y) - y : x,y \stackrel{R}{\leftarrow} \{0,1\}^\ell \wedge x = y] \cdot \\
&\quad \mathrm{Prob}[x = y : x,y \stackrel{R}{\leftarrow} \{0,1\}^\ell] + \\
&\quad \mathrm{Prob}[f(x) - x = f(y) - y : x,y \stackrel{R}{\leftarrow} \{0,1\}^\ell \wedge x \neq y] \cdot \\
&\quad \mathrm{Prob}[x \neq y : x,y \stackrel{R}{\leftarrow} \{0,1\}^\ell] \\
&= 1 \cdot 2^{-\ell} + 2^{-\ell} \cdot \frac{2^\ell - 1}{2^\ell} = \frac{2 - 2^{-\ell}}{2^\ell} = 2^{1-\ell} - 2^{-2\ell} \ ,
\end{aligned}$$

($q > 1$ korral on saadud tõenäosus veelgi väiksem) ning seega

$$\mathrm{InSec}_F^{\mathrm{prf}}(2, O(1)) = 1 - 2^{1-\ell} + 2^{-2\ell} = 1 - o(1) \ .$$

Ülesanne 3.4.1 Olgu F — kõigi afiinsete šifrite pere. Leida võimalikult hea eraldaja D, ning määrata suurus $\mathrm{InSec}_F^{\mathrm{prf}}(q,t)$.

Toodud näitest on näha, et suurus $\varepsilon = \varepsilon(q,t)$ on (kasvav) funktsioon arvudest q ja t, ehk üks ja seesama funktsioon on $(q,t,\varepsilon(q,t))$-turvaline PRF suvaliste q ja t jaoks. Negatiivsetes turvatõestustes leitakse hea eraldaja, mille korral $\varepsilon(q,t)$ oleks võimalikult suur. Positiivsetes turvatõestustes tõestatakse, et ei leidu eraldajat, mille korral $\varepsilon(q,t) > g(q,t)$ mingi aeglaselt kasvava funktsiooni g korral. Paneme tähele, et iga funktsioonide pere on $(q,t,1)$-turvaline PRF-ide pere. Samas mida väiksem on $\varepsilon(q,t)$, seda "juhuslikum" on antud funktsioon.

Miks on vaja pseudojuhuslikust nii täpselt kvantiseerida? Sest täpne kvantitatiivne hinnang võimaldab konstrueerida praktilisi primitiive ja protokolle, mille puhul erinevad turvaparameetrid (näiteks võtme pikkus) on nii minimaalsed kui võimalik. Me toome selle kohta külluslikult näiteid hiljem. Märgime siinkohas, et $q = t = \infty$ korral saab eraldada suvalist funktsioonide peret $F \neq \mathrm{Func}_{\ell,L}$ perest $\mathrm{Func}_{\ell,L}$. Oluline on aru saada, et PRF-i tugevuse definitsioonis kasutatatud eraldajad on limiteeritud ainult aja ja päringute arvu poolest: eraldaja võib kasutada suvalist strateegiat, seni kuni ta ei ületa oma ressursse.

Kuidas sõltub $\mathrm{InSec}_F^{\mathrm{prf}}(q,t)$ võtmepikkusest k? On selge, et $\mathrm{InSec}_F^{\mathrm{prf}}(q,t) \geq t/2^k$, kuid ilma perest F rohkem teadmata on sellele küsimusele raske vastata. Üldjuhul ei loe mitte võtme pikkus, vaid eraldaja poolt saavutatav edu. Näiteks võib tuua funktsioonide pere $\{F_K\}$, mis väljastab oma teise argumendi, sõltumata võtme pikkusest k: sellise funktsioonide pere puhul on $\mathrm{InSec}_F^{\mathrm{prf}}(2, O(1)) = 1 - o(1)$.

3.4.2 Pseudojuhuslikud permutatsioonid

Definitsioonis 2.1.1 olid plokkšifrid defineeritud kui permutatsioonide pere. On selge, et plokkšifrilt ei saa seetõttu ka nõuda juhusliku funktsioonina käitumist: pigem käitub ta kui juhuslik permutatsioon. Alljärgnevas toome ära pseudojuhuslike permutatsioonide definitsiooni, mis erineb pseudojuhuslike funktsioonide omast vaid hulga Perm_ℓ kasutamise poolest hulga $\mathrm{Func}_{\ell,L}$ asemel.

Olgu $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$. PRP-de korral vaadeldakse järgnevat kahte maailma:

Maailm 0. Kasutajale antakse musta kastina ette hulgast Perm_ℓ juhuslikult valitud funktsioon f: $f \xleftarrow{R} \mathrm{Perm}_\ell$. Seega f on lihtsalt juhuslik permutatsioon.
Maailm 1. Kasutajale antakse musta kastina ette funktsioon $f \xleftarrow{R} F$.

Definitsioon 3.4.2 Olgu $F : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ funktsioonide pere, ning olgu D algoritm (*eraldaja*), mis võib teha oraakelpäringuid kujutusele

$$f : \{0,1\}^\ell \to \{0,1\}^\ell$$

ning väljastab seejärel nulli või ühe. Suurust

$$\mathrm{Adv}_F^{\mathrm{prp}}(D) := \mathrm{Prob}[D^f = 1 : f \xleftarrow{R} F] - \mathrm{Prob}[D^f = 1 : f \xleftarrow{R} \mathrm{Perm}_\ell]$$

nimetatakse eraldaja *eduks* (advantage). Olgu $\mathrm{InSec}_F^{\mathrm{prp}}(q,t) = \max_D Adv_F^{\mathrm{prp}}(D)$, kus maksimum on võetud üle algoritmide D mis töötavad ajas t ja teevad ülimalt q musta kasti päringut. Funktsioonide peret F nimetatakse (q,t,ε)-*turvaliseks pseudojuhuslike permutatsioonide pereks*, kui $\mathrm{InSec}_F^{\mathrm{prp}}(q,t) \leq \varepsilon$.

3.4.3 Sünnipäevarünne

Selgitame järgnevalt lühidalt, kuidas eraldada permutatsioonide perest valitud permutatsiooni f juhuslikust funktsioonist. Üks võimalus on esitada oraaklile päringuid juhuslikult valitud kohtadel x_1, x_2, ... Kui f on permutatsioon, siis peavad kõik väärtused $y_i = f(x_i)$ olema erinevad. Kui f on juhuslik funktsioon, siis need väärtused võivad olla samad või erinevad. Kui väärtused y_i on paarikaupa erinevad, siis tõenäoliselt tegu permutatsiooniga. Üllataval kombel töötab selline eraldaja väga efektiivselt, selle põhjuseks on *sünnipäeva paradoks*.

Teoreem 3.4.1 *Olgu $P : \{0,1\}^k \times \{0,1\}^\ell \to \{0,1\}^\ell$ mingi permutatsioonide pere. Sellisel juhul leidub eraldaja D, mis teeb q, $2 \leq q \leq 2^{(\ell+1)/2}$ päringut ja töötab ajas $t = O(q\ell)$, nii et $\mathrm{Adv}_P^{\mathrm{prf}}(D) \geq 0.3 \cdot \frac{q(q-1)}{2^\ell}$. Seega,*

$$\mathrm{InSec}_P^{\mathrm{prf}}(q,t) \geq 0.3 \cdot \frac{q(q-1)}{2^\ell} \quad .$$

Tõestus. Ehitame järgmise eraldaja D^f:

1. Iga $i = 1, \dots, q$ jaoks tee:
   Olgu x_i arvu i ℓ-bitine esitus.
   Olgu $y_i = f(x_i)$.

2. Kui suurused y_i on paarikaupa erinevad, väljasta 1, muidu väljasta 0.

Lihtne on näha, et $\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} P] = 1$. Kui f on juhuslik funktsioon, tuleb tõenäosuse $\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,\ell}]$ leidmiseks esmalt leida tõenäosus, et juhuslikult valitud suurused $y_1, \ldots, y_q \in 2^\ell$ on paarikaupa erinevad. Viimane tõenäosus on

$$1 \cdot \left(1 - \frac{1}{2^\ell}\right) \cdots \left(1 - \frac{q-1}{2^\ell}\right) = \prod_{i=1}^{1-1} \left(1 - \frac{i}{2^\ell}\right) \approx \prod_{i=1}^{q-1} e^{-i/2^\ell} = e^{-q(q-1)/2^\ell} \quad .$$

Nüüd,

$$\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,\ell}] = 1 - e^{-q(q-1)/2^\ell} \quad .$$

Kuna suvalise reaalarvu $x \in [0,1]$ korral $(1 - 1/e)x \leq 1 - e^{-x} \leq x$, siis juhul $1 \leq q \leq \sqrt{2^{\ell+1}}$ saame, et

$$\mathrm{Prob}[D^f = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,\ell}] \leq 1 - \left(1 - \frac{1}{e}\right) \frac{q(q-1)}{2^{\ell+1}} \quad .$$

Seega $\mathrm{Adv}_P^{\mathrm{prf}}(D) \geq |1 - (1 - \left(1 - \frac{1}{e}\right) \frac{q(q-1)}{2^{\ell+1}})| = \left(1 - \frac{1}{e}\right) \frac{q(q-1)}{2^{\ell+1}}$. □

3.5 Erinevad turvadefinitsioonid

Eelmise loengu temaatika lõpetuseks märgime, et toodud lähenemine baseerub, kuid erineb mitmes tähtsas punktis, tüüpilisest keerukusteoreetilisest lähenemisest kus on tavaliselt oluline vaid algoritm polünomiaalsus. Kui erinevate algoritmide vahel on võimalik ära tuua polünomiaalse keerukusega reduktsioon, siis tavaliselt sellega ka rahuldutakse. "Täpse turvalisuse" paradigmat järgides oleme me huvitatud eksaktsetest reduktsioonidest, mille tulemusel saadud algoritmide efektiivsuse (=kuluvate ressursside) ülem- ja alamtõkkeid saab täpselt hinnata. Täpsed (st mitte ainult asümptootilised) hinnangud on hädavajalikud sümmeetrilise krüptograafia puhul, kus näiteks plokkšifreid (DES, IDEA, ...) disainitakse fikseeritud võtme- ja plokipikkuste k ja ℓ jaoks. Turvatõestustes antud täpne hinnang reduktsioonide perele varieeruvate k ja ℓ puhul resulteerub täpses hinnangus suuruste k ja ℓ (näiteks $k = \ell = 128$) fikseerimisel.

Käesolevas ja järgnevas loengus vaatleme (sümmeetriliste) krüptosüsteemide turvalisust. Mitteformaalselt, krüptosüsteem on võtmega indekseeritud funktsioonide pere avatekstide hulgast krüptogrammide hulka. Krüptosüsteemiks on näiteks fikseeritud režiimis töötav plokkšiffer.

Definitsioon 3.5.1 Olgu $\mathrm{Coins} = 2^\infty$ — lõpmatute stringide hulk. *Randomiseeritud, mäluta, sümmeetriline) krüptosüsteem* $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ on järgmiste algoritmide kolmik:

$$\begin{aligned}
\mathcal{E} &: \mathrm{Keys}(\Pi) \times \mathrm{Dom}(\Pi) \times \mathrm{Coins} \to \mathrm{Range}(\Pi) \ , \\
\mathcal{D} &: \mathrm{Keys}(\Pi) \times \mathrm{Range}(\Pi) \to \mathrm{Dom}(\Pi) \cup \{\perp\} \ , \\
\mathcal{K} &: \mathrm{Coins} \to \mathrm{Keys}(\Pi) \ ,
\end{aligned}$$

tingimusel $p \in \mathrm{Dom}(\Pi) \Rightarrow (\forall p')|p'| = |p| \Rightarrow p' \in \mathrm{Dom}(\Pi)$. Hulga $\mathrm{Dom}(\Pi)$ elemente nimetatakse *avatekstideks*, hulga $\mathrm{Range}(\Pi)$ elemente *krüptogrammideks* ja hulga $\mathrm{Keys}(\Pi)$ elemente *võtmeteks*. Algoritmi $\mathcal{E}$ kutsutakse *krüptimisalgoritmiks*, algoritmi $\mathcal{D}$ — *dekrüptimisalgoritmiks* ning algoritmi $\mathcal{K}$ — *võtmegenereerijaks*.

Nii nagu plokkšifritegi puhul vaadeldakse algoritme $\mathcal{E}$ ja $\mathcal{D}$ kui võtmega indekseeritud funktsioonide peresid. Harilikult on võtmegenereerijaks $\mathcal{K}$ mingi tõenäosusjaotuse alusel võtme $K \stackrel{R}{\leftarrow} \mathrm{Keys}(\Pi)$ valimise algoritm. Sellisel juhul ei mainita algoritmi $\mathcal{K}$. Samuti jäetakse tavaliselt mainimata algoritmi $\mathcal{E}_K(p, r)$ teine argument: sellisel juhul öeldakse lihtsalt, et $\mathcal{E}_K(\cdot)$ on randomiseeritud algoritm. $\mathcal{D}_K(c) = \perp$ kui $c \neq \mathcal{E}_K(p)$ ühegi avateksti p jaoks.

Nagu tavaliselt nõutakse ka siin, et $\mathcal{E}$, $\mathcal{D}$ ja $\mathcal{K}$ oleks efektiivselt arvutatavad funktsioonid.

Formaalseks turvalisuseks on vaja, et $\mathcal{E}_K$ oleks randomiseeritud algoritm: vastasel korral krüptuks avatekst p alati krüptogrammiks c. Sellisel juhul saaks vaenlane iga krüptogrammi puhul teada, kas seda teksti on varem saadetud, ja kui on siis millises kontekstis. Üks võimalusi randomiseerituse vältimiseks on kasutada *mäluga krüptosüsteeme* (stateful encryption schemes), mis peavad meeles (näiteks) loendurit, mida suurendatakse iga uue paketi krüptimisel. Sel juhul sõltub krüptogramm $\mathcal{E}_K(p)$ turvalisel moel avatekstist p ning eelnevalt krüptitud avatekstide arvust: see välistab eelnevalt mainitud informatsioonilekke. Loendur ei ole aga sugugi ainus mäluliik. Formaalselt defineeritakse mäluga krüptosüsteem samuti kui mäluta süsteem, v.a. krüptimisalgoritm $\mathcal{E}$, mis saab lisasisendiks ja lisaväljundiks oleku. Kõik järgnevad turvadefinitsioonid on lihtsalt ülekantavad mäluga süsteemidele. Vastandina mäluga algoritmidele nimetatakse ülejäänud krüptosüsteeme *mäluta krüptosüsteemideks* (stateless encryption schemes).

Üks esimesi turvatõestuste juures tekkivaid küsimusi on turvadefinitsioonid ise: kas mingis mõttes turvalise krüptosüsteemi vastu ei leidu efektiivseid ründeid mõnes teises praktikas olulises tähenduses? Defineerime esmalt mitteformaalselt neli erinevat turvalisuse mõistet:

tõeline-või-juhuslik (*real-or-random*), kus eraldaja püüab aru saada, kas mustas kastis olev funktsioon krüptib eraldaja poolt ette antud avateksti või juhuslikku sama pikka stringi.[1]

vasak-või-parem (*left-or-right*), kus mängu eri maailmates krüptitakse eri võtmetega ning eraldaja eesmärk on aru saada, kummas maailmas ta on (eristada ühe võtmega krüptitud krüptogramme teise võtmega krüptitutest),

semantiline (*semantic*), kus krüptogramme nägeva ründaja ülesandeks on krüptitud avateksti mingi fikseeritud omaduse (näiteks paarsuse) leidmine ning nõutakse, et selle omaduse leidmine oleks "sama raske" kui vastavaid krüptogramme nägemata,

arvuta-ja-arva (*find-then-guess*), kus vaenlane valib ise kaks avateksti, laseb mustal kastil need krüptida juhuslikult valitud võtmega ning endale tagasi saata ühe (juhuslikult valitud) krüptogrammi. Ründaja ülesanne on ära arvata, kumba avateksti krüptitud versiooni ta näeb.

Semantiline turvalisus on võibolla kõige intuitiivsem, kuid see-eest raskeimini defineeritav ning seetõttu ka raskeimini rakendatav turvatõestustes. Tänu sellele tõestatakse krüptosüsteemide turvalisust tavaliselt tõeline-või-juhuslik või vasak-või-parem tähenduses. Samas poleks saadud turvatõestustest kasu, kui selguks et tõeline-või-juhuslik mõistes turvaline funktsioonide pere pole semantiliselt turvaline (näiteks kui oleks ikkagi lihtne krüptogrammi teades välja arvutada sellele vastava avateksti bittide paarsust)! Järgnevas tõestame, et asjalood ei ole siiski nii hullud: nimelt näitame, et kui krüptosüsteem on turvaline tõeline-või-juhuslik või vasak-või-parem tähenduses, siis on ta ka turvaline ülejäänud kolmes tähenduses.

Ütleme, et oraakliga algoritm D^f on (q,t,μ)-algoritm, kui ta töötab ajas t ning teeb oraaklile ülimalt q päringut mis ei ületa oma pikkuselt kokku μ bitti. "Ajaks" t võetakse tavaliselt kuluva aja ning kasutatava ruumi summa fikseeritud RAM-mudelis, μ on vajalik kuna erinevad päringud võivad olla erineva pikkusega. Sümboolika lihtsustamiseks defineerime $\mathrm{left}(x_0,x_1) := x_0$, $\mathrm{right}(x_0,x_1) := x_1$ ning

$$\mathrm{rand}_{|\cdot|}(x) := (x' \xleftarrow{R} \{0,1\}^{|x|})$$

[1]Paneme tähele, et enamuses praktilistest krüptosüsteemidest on krüptogrammi ja avateksti pikkus üksüheselt seotud ning seega on avateksti pikkus lihtsalt leitav kui teame vähemalt ühte krüptogrammi. Siin me nõuame, et mitte midagi muud peale avateksti pikkuse eraldaja avateksti kohta teada ei saaks.

(st $\mathrm{rand}_{|\cdot|}(x)$ on randomiseeritud algoritm, mis valib juhuslikult ühtlase jaotuse järgi $|x|$-bitise stringi). Kujutuste f ja g korral defineerime kujutuse $f \times g$ järgnevalt: $(f \times g)(x_1, x_2) := (f(x_1), g(x_2))$. Edasises tähistame $\Sigma = \{0,1\}$.

3.5.1 Turvadefinitsioonid

Erinevate turvalisuse mõistete defineerimisel kasutame analoogilist kahe maailma mudelit nagu PRF-ide ning PRP-de defineerimisel.

Definitsioon 3.5.2 (Tõeline-või-juhuslik) On antud krüptosüsteem

$$\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K}) \ ,$$

kus

$$\mathcal{E} : \Sigma^k \times \Sigma^\ell \times \Sigma^r \to \Sigma^L \ .$$

Maailm 0. Valitakse juhuslikult $f \stackrel{R}{\leftarrow} \mathcal{E}$; musta kasti paigutatakse funktsioon f.
Maailm 1. Valitakse $f \stackrel{R}{\leftarrow} \mathcal{E}$; musta kasti paigutatakse kujutus $f \circ \mathrm{rand}_{|\cdot|}$.
Vaenlase ülesandeks on ära arvata, kumbas maailmas ta on. Krüptosüsteemi ebaturvalisust defineeritakse kui maksimumi üle (q,t,μ)-vaenlaste edu:

$$\begin{aligned} \mathrm{Adv}^{\mathrm{ror}}_{\Pi}(D) &:= \mathrm{Prob}[A_1^f = 1 : f \stackrel{R}{\leftarrow} \mathcal{E}] - \mathrm{Prob}[A_1^{f \circ \mathrm{rand}_{|\cdot|}} = 1 : f \stackrel{R}{\leftarrow} \mathcal{E}] \ , \\ \mathrm{InSec}^{\mathrm{ror}}_{\Pi}(q,t,\mu) &:= \max_D \{\mathrm{Adv}^{\mathrm{ror}}_{\Pi}(D)\} \ . \end{aligned}$$

Definitsioon 3.5.3 (Vasak-või-parem) Olgu $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$. Vaatleme järgmist kahte maailma:
Maailm 0. Valitakse juhuslikult $f \stackrel{R}{\leftarrow} \mathcal{E}$; musta kasti paigutatakse kujutus $f \circ \mathrm{left}$.
Maailm 1. Valitakse juhuslikult $f \stackrel{R}{\leftarrow} \mathcal{E}$; musta kasti paigutatakse kujutus $f \circ \mathrm{right}$.
Päringu (x_0, x_1) pikkuseks loetakse väärtust $|x_0| = |x_1|$. Ka siin on vaenlase ülesandeks ära arvata, kumbas maailmas ta on. Krüptosüsteemi ebaturvalisust defineeritakse kui maksimumi üle (q,t,μ)-vaenlaste edu:

$$\begin{aligned} \mathrm{Adv}^{\mathrm{lor}}_{\Pi}(D) &:= \mathrm{Prob}\left[D^{f \circ \mathrm{left}} = 1 : f \stackrel{R}{\leftarrow} \mathcal{E}\right] - \mathrm{Prob}\left[D^{f \circ \mathrm{right}} = 1 : f \stackrel{R}{\leftarrow} \mathcal{E}\right] \ , \\ \mathrm{InSec}^{\mathrm{lor}}_{\Pi}(q,t,\mu) &:= \max_D \{\mathrm{Adv}^{\mathrm{lor}}_{\Pi}(D)\} \ . \end{aligned}$$

Definitsioon 3.5.4 (Arvuta-ja-arva) Antud juhul töötab eraldaja D kahes etapis. Esimeses etapis genereerib (arvutab) D kaks võrdse pikkusega stringi x_0 ja x_1, millele vastavaid krüptogramme ta hiljem soovib eristada. Lisaks salvestab D oleku s, mida ta kasutab hilisemas "arva" faasis, kus talle lisaks s-le antakse ka juhuslikult valitud stringi $x \in \{x_0, x_1\}$ krüptogramm y. Eraldaja eesmärgiks on leida, kumb string valiti:

$$\begin{aligned}\mathrm{Adv}^{\mathrm{fg}}_{\Pi}(D) = 2 \cdot \mathrm{Prob}[D^f(\mathrm{guess}, y, s) = b : f \xleftarrow{R} \mathcal{E}, (x_0, x_1, s) \xleftarrow{R} D^f(\mathrm{find}),\\ b \xleftarrow{R} \Sigma, y \xleftarrow{R} f(x_b)] - 1 \quad ,\\ \mathrm{InSec}^{\mathrm{fg}}_{\Pi}(q, t\mu) = \max_D \{\mathrm{Adv}^{\mathrm{fg}}_{\Pi}(D)\}\end{aligned}$$

(korrutamine kahega ja ühe lahutamine toimub selleks, et 0 vastaks täielikult turvalisele süsteemile, 1 aga täielikult mitteturvalisele). Siin eeldatakse, et $|x_0| = |x_1|$.

Semantilise turvalisuse mõistet ei hakka me loengus täpselt defineerima. Tudeng loeb ise artiklit [BDJR97]. ning defineerib formaalselt erinevad turvalisuse mõisted mäluga krüptosüsteemide jaoks.

3.5.2 Turvatõestused

Teoreemid 3.5.2 ja 3.5.1 näitavad, et vasak-või-parem turvalisus tähendab peaaegu täpselt sedasama, mis tõeline-või-juhuslik turvalisus.

Teoreem 3.5.1 *Kui krüptosüsteem $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ on (q, t, μ, ε)-turvaline tõeline-või-juhuslik tähenduses, siis on ta $(q, t - c \cdot \mu, \mu, 2\varepsilon)$-turvaline vasak-või-parem tähenduses, kus c on valitud masinamudelist sõltuv konstant.*

Tõestus. Tõestus on vastuväiteline. Oletame, et leidub (t', q, μ)-algoritm D_{lor}, nii et $\mathrm{Adv}^{\mathrm{lor}}_{\Pi}(D_{\mathrm{lor}}) = \varepsilon_2$. On selge, et joonisel 3.1 kujutatud algoritm D_{ror} on $(q, t + c \cdot \mu, \mu)$-algoritm. Tõestuse lõpetuseks on vaja leida väärtuse $\mathrm{Adv}^{\mathrm{ror}}_{\Pi}(D_{\mathrm{ror}})$ alamtõke. Kui oraakliks on $f = f' \circ \mathrm{rand}_{|\cdot|}$ (maailm 1), siis on kujutuste $f \circ \mathrm{left}$ ja $f \circ \mathrm{right}$ väljundid võrdse tõenäosusjaotusega, seega $\mathrm{Prob}[D_{\mathrm{ror}}^{f \circ \mathrm{rand}_{|\cdot|}} = 1 : f \xleftarrow{R} \mathcal{E}] = \frac{1}{2}$. Siit saame, et

$$\begin{aligned}\mathrm{Adv}^{\mathrm{ror}}_{\Pi}(D_{\mathrm{ror}}) &= \mathrm{Prob}\left[D^f_{\mathrm{ror}} = 1 : f \xleftarrow{R} \mathcal{E}\right] - \mathrm{Prob}\left[D_{\mathrm{ror}}^{f \circ \mathrm{rand}_{|\cdot|}} = 1 : f \xleftarrow{R} \mathcal{E}\right]\\ &= \mathrm{Prob}\left[D^f_{\mathrm{ror}} = 1 : f \xleftarrow{R} \mathcal{E}\right] - \frac{1}{2} \quad .\end{aligned}$$

Algoritm 3.1 Algoritm $D_{\mathrm{ror}}^{f(\cdot)}$.

1. $b \xleftarrow{R} \Sigma$.
2. Kui $b = 0$ siis $d \xleftarrow{R} D_{\mathrm{lor}}^{f\circ\mathrm{left}}$ muidu $d \xleftarrow{R} D_{\mathrm{lor}}^{f\circ\mathrm{right}}$.
3. Kui $b = d$, tagasta 0 muidu tagasta 1.

Kasutades tõenäosuste liitmislauset saame järgnevalt, et

$$
\begin{aligned}
\mathrm{Adv}_{\Pi}^{\mathrm{ror}}(D_{\mathrm{ror}}) =& \mathrm{Prob}\,[b=1]\,\mathrm{Prob}\left[D_{\mathrm{ror}}^{f}=1 : b=1, f\xleftarrow{R}\mathcal{E}\right]+\\
&\mathrm{Prob}\,[b=0]\,\mathrm{Prob}\left[D_{\mathrm{ror}}^{f}=1 : b=0, f\xleftarrow{R}\mathcal{E}\right]-\frac{1}{2}\\
=&\frac{1}{2}\left(1-\mathrm{Prob}\left[D_{\mathrm{ror}}^{f}=0 : b=1, f\xleftarrow{R}\mathcal{E}\right]\right)+\\
&\frac{1}{2}\mathrm{Prob}\left[D_{\mathrm{ror}}^{f}=1 : b=0, f\xleftarrow{R}\mathcal{E}\right]-\frac{1}{2}\\
=&\frac{1}{2}\left(1-\mathrm{Prob}\left[D_{\mathrm{lor}}^{f\circ\mathrm{right}}=1 : f\xleftarrow{R}\mathcal{E}\right]\right)+\\
&\frac{1}{2}\mathrm{Prob}\left[D_{\mathrm{lor}}^{f\circ\mathrm{left}}=1 : f\xleftarrow{R}\mathcal{E}\right]-\frac{1}{2}\\
=&\frac{1}{2}\left(\mathrm{Prob}\left[D_{\mathrm{lor}}^{f\circ\mathrm{left}}=1 : f\xleftarrow{R}\mathcal{E}\right]-\mathrm{Prob}\left[D_{\mathrm{lor}}^{f\circ\mathrm{right}}=1 : f\xleftarrow{R}\mathcal{E}\right]\right)\\
=&\frac{1}{2}\mathrm{Adv}_{\Pi}^{\mathrm{lor}}(D_{\mathrm{lor}}) \quad .
\end{aligned}
$$

Siin kasutasime algoritmi D_{ror}^{f} järgmist kahte omadust: 1) kui $b=1$ siis $D_{\mathrm{ror}}^{f} = 0 \Rightarrow D_{\mathrm{lor}}^{f\circ\mathrm{right}} = 1$ ning 2) kui $b = 0$ siis $D_{\mathrm{ror}}^{f} = 1 \Rightarrow D_{\mathrm{lor}}^{f\circ\mathrm{left}} = 1$. $\square$

Teoreem 3.5.2 *Kui krüptosüsteem $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ on (t, q, μ, ε)-turvaline vasak-või-parem tähenduses, siis on ta $(t - c\mu, q, \mu, \varepsilon)$-turvaline tõeline-või-juhuslik tähenduses, kus c on masinamudelist sõltuv konstant.*

Tõestus. Tõestus on vastuväiteline. Olgu D_{ror} mingi (t', q, μ)-algoritm. Näitame, et sel juhul leidub $(q, t' + c\cdot\mu, \mu)$-algoritm D_{lor}, nii et $\mathrm{Adv}_{\Pi}^{\mathrm{lor}}(D_{\mathrm{lor}}) = \mathrm{Adv}_{\Pi}^{\mathrm{ror}}(D_{\mathrm{ror}})$. Tõepoolest, analüüsime joonisel 3.2 kujutatud algoritmi D_{lor}. Lihtne on näha, et

Algoritm 3.2 Algoritm $D_{\mathrm{lor}}^{f(\cdot,\cdot)}$ teoreemist 3.5.2.

1. Tagasta $D_{\mathrm{ror}}^{f\circ(I\times \mathrm{rand}_{|\cdot|})}$.

D_{lor} on $(q, t' + c\cdot\mu, \mu)$-algoritm. Tõenäosuse $\mathrm{Adv}_{\Pi}^{\mathrm{lor}}(D_{\mathrm{lor}})$ arvutame järgnevalt:

$$\begin{aligned}
\mathrm{Adv}_{\Pi}^{\mathrm{lor}}(D_{\mathrm{lor}}) =& \mathrm{Prob}\left[D_{\mathrm{lor}}^{f\circ\mathrm{left}} = 1 : f \overset{R}{\leftarrow} \mathcal{E}\right] - \mathrm{Prob}\left[D_{\mathrm{lor}}^{f\circ\mathrm{right}} = 1 : f \overset{R}{\leftarrow} \mathcal{E}\right]\\
=& \underbrace{\mathrm{Prob}\left[D_{\mathrm{ror}}^{f\circ\mathrm{left}\circ(I\times\mathrm{rand}_{|\cdot|})} = 1 : f \overset{R}{\leftarrow} \mathcal{E}\right]}_{\text{Lihtsustame: } \mathrm{left}\circ(I\times\mathrm{rand}_{|\cdot|}) = I} -\\
& \underbrace{\mathrm{Prob}\left[D_{\mathrm{ror}}^{f\circ\mathrm{right}\circ(I\times\mathrm{rand}_{|\cdot|})} = 1 : f \overset{R}{\leftarrow} \mathcal{E}\right]}_{\text{Lihtsustame: } \mathrm{right}\circ(I\times\mathrm{rand}_{|\cdot|}) = \mathrm{rand}_{|\cdot|}}\\
=& \mathrm{Prob}\left[D_{\mathrm{ror}}^{f} = 1 : f \overset{R}{\leftarrow} \mathcal{E}\right] - \mathrm{Prob}\left[D_{\mathrm{ror}}^{f\circ\mathrm{rand}_{|\cdot|}} = 1 : f \overset{R}{\leftarrow} \mathcal{E}\right]\\
=& \mathrm{Adv}_{\Pi}^{\mathrm{ror}}(D_{\mathrm{ror}}) \ .
\end{aligned}$$

□

Teoreem 3.5.3 *Olgu $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$ mingi krüptosüsteem. Kui Π on (q,t,μ,ε)-turvaline vasak-või-parem tähenduses, siis on ta $(q, t - c\mu, \mu, \varepsilon)$-turvaline arvuta-ja-arva tähenduses, kus c on masinamudelist sõltuv konstant.*

Tõestus. Toodud artiklis [BDJR97]. □

Teoreem 3.5.4 *Olgu $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$. Kui Π on (q,t,μ,ε)-turvaline arvuta-või-arva tähenduses, siis on ta $(q, t - c\mu, \mu, q\varepsilon)$-turvaline vasak-või-parem tähenduses, kus c on masinamudelist sõltuv konstant.*

Tõestus. Vastuväiteline. Olgu D_{ror} mingi (q,t',μ)-algoritm eduga $\mathrm{Adv}_{\Pi}^{\mathrm{lor}}(D_{\mathrm{ror}})$. Vaatleme algoritmi 3.3. On lihtne näha, et D_{fg} on $(q, t' + c\mu, \mu)$-algoritm. Näitame järgnevas, et $\mathrm{Adv}_{\Pi}^{\mathrm{fg}}(D_{\mathrm{fg}}) = \mathrm{Adv}_{\Pi}^{\mathrm{lor}}(D_{\mathrm{ror}})/q$.

Defineerimise selleks mängud $G_0, \ldots, G_q$, kus mäng G_i valib esmalt $f \overset{R}{\leftarrow} \mathcal{E}$ ning seejärel käivitab D_{ror}, vastates esimesele i päringule oraakliga $f\circ\mathrm{left}$ ning ülejäänud päringutele oraakliga $f\circ\mathrm{right}$. Olgu $\mathrm{Prob}_i[D_{\mathrm{ror}} = 1]$ tõenäosus, et D_{ror}

Algoritm 3.3 Teoreemis 3.5.4 kasutatav algoritm D_{fg}.

Algoritm $D^{f}_{\mathrm{fg}}(\mathrm{find})$

1. $i \xleftarrow{R} \{1,\dots,q\}$.
2. Algoritmi $D^{f\circ\mathrm{left}}_{\mathrm{ror}}$ täidetakse punktini, kus D_{ror} teeb oma $i$nda oraakelpäringu (x^i_0, x^i_1) (st et D_{ror} on selleks hetkeks teinud oma $i$nda päringu ning ootab oraaklilt vastust). Olgu s algoritmi D_{ror} olek sel hetkel.
3. Tagasta (x^i_0, x^i_1, s).

Algoritm $D^{f}_{\mathrm{fg}}(\mathrm{guess}, y, s)$

1. Jätkata algoritmi D_{ror} olekust s, vastates $i$ndale oraakelpäringule stringiga y. Peatu enne kui D_{ror} teeb järgmise päringu.
2. Jätka algoritmi D_{ror} täitmist, andes järelejäänud oraaklipäringutele (x,y) vastuse $(f\circ\mathrm{right})(x,y)$, kuni D_{ror} peatub.
3. Kui D_{ror} vastab 1, siis tagasta 0, vastasel korral 1.

vastab 1 mängus G_i. Vaatleme järgnevas eksperimenti, mida sooritatakse suuruse $\mathrm{Adv}^{\mathrm{fg}}_{\Pi}(D_{\mathrm{fg}})$ leidmiseks definitsioonis 3.5.4:

$$\mathrm{Adv}^{\mathrm{fg}}_{\Pi}(D_{\mathrm{fg}}) = 2\cdot \mathrm{Prob}[D^{f}_{\mathrm{fg}}(\mathrm{guess},y,s) = b : f \xleftarrow{R} \mathcal{E}, (x_0,x_1,s) \xleftarrow{R} D^{f}_{\mathrm{fg}}(\mathrm{find}),$$
$$b \xleftarrow{R} \Sigma, y \xleftarrow{R} f(x_b)] - 1 \ .$$

Selles eksperimendis

$$y = f(x^i_b) = \begin{cases} f\circ\mathrm{left}(x_0,x_1), & b=0 \ , \\ f\circ\mathrm{right}(x_0,x_1), & b=1 \ . \end{cases}$$

Seega juhul $b=0$ on antud eksperiment võrdne mänguga G_i, juhul $b=1$ on aga tegemist mänguga G_{i+1} ning järelikult fikseeritud i korral on algoritmi D_{fg} edu

$\mathrm{Prob}_i[D_{\mathrm{lor}} = 1] - \mathrm{Prob}_{i+1}[D_{\mathrm{lor}} = 1]$. Kuna i valiti juhuslikult, siis

$$\begin{aligned}
\mathrm{Adv}^{\mathrm{fg}}_{\Pi}(D) =& \frac{1}{q} \cdot \sum_{i=0}^{q-1} (\mathrm{Prob}_i[D_{\mathrm{lor}} = 1] - \mathrm{Prob}_{i+1}[D_{\mathrm{lor}} = 1]) \\
=& (\mathrm{Prob}_0[D_{\mathrm{lor}} = 1] - \mathrm{Prob}_q[D_{\mathrm{lor}} = 1])/q \\
=& (\mathrm{Prob}\left[D^{f\circ\mathrm{left}}_{\mathrm{lor}} = 1\right] - \mathrm{Prob}\left[D^{f\circ\mathrm{left}}_{\mathrm{lor}} = 1\right])/q \\
=& \mathrm{Adv}^{\mathrm{lor}}_{\Pi}(D_{\mathrm{lor}})/q \quad .
\end{aligned}$$

□

Järgmine teoreem näitab, et toodud reduktsioon on peaaegu optimaalne.

Teoreem 3.5.5 *Leidub selline konstant $c > 0$, nii et järgmine väide on tõene. Leidugu mäluta krüptosüsteem mis on (q,t',μ,ε')-turvaline arvuta-ja-arva tähenduses. Siis leidub mäluta krüptosüsteem Π', nii et 1) leidub (q,cq,q)-algoritm, mis murrab selle lahti vasak-või-parem tähenduses eduga $\varepsilon' + 1/q$ ning 2) ei leidu (q,t',μ)-algoritmi, mis murraks selle lahti arvuta-ja-arva tähenduses eduga* 0.632.

Tõestus. Toodud artiklis [BDJR97]. □

Artiklis [BDJR97] on seotud erinevad turvalisuse mõisted. Kõige huvitavamad on seal teoreemid 7 ja 8, mis näivad väitvat, et avateksti mingi "keerulise" omaduse leidmine võib ründajal kergemini õnnestuda kui mingi "lihtsa" omaduse leidmine. Nendest teoreemidest järeldub, et vasak-või-parem tähenduses turvaline (ning ka tõeline-või-juhuslik tähenduses turvaline) krüptosüsteem on ka teiste definitsioonide järgselt turvaline, kusjuures reduktsioonid on täpsed (v.a. kui semantilise turvalisuse juures leitav omadus ja/või kasutatava tõenäosusjaotuse järgi elemendi valimine on raske). Seega turvatõestus vasak-või-parem-turvalisust kasutades annab ka täpse tõestuse semantilise turvalisuse jaoks.

Ülesanne 3.5.1 Lugeda läbi artikli [BDJR97] peatükid 1–3 ning lisa A.

3.6 Plokkšifri töörežiimide turvalisuse analüüs

Rakendame toodud teadmisi konkreetsel näitel. Peatükis 2.2 kirjeldasime erinevaid plokkšifrite töörežiime ning tõime natuke mitteformaalseid argumente nende turvalisuse kohta. On selge, et mitteformaalsetest argumentidest ei piisa aga turvalisuseks (sinu boss võib sind küll mitteformaalselt vallandada, kui sinu valitud ebaturvalise krüptosüsteemiga kaitstud salajased firma dokumendid avalikuks tulevad!).

Järgnevalt tõestame (toetudes artiklile [BDJR97]) mõned täpsed tõkked ahelning loendurrežiimide turvalisuse kohta eeldusel, et kasutatav plokkšiffer on pseudojuhuslik permutatsioon. Teame juba eelnevalt, et PRP-d ei ole "täpselt" PRF-d (tänu sünnipäevaründele, paragrahv 3.4.3). Eelsoojenduseks järgnevale aga selline kodune ülesanne, mis annab vastupidise seose.

Ülesanne 3.6.1 Olgu F mingi fikseeritud (q,t,ε)-turvaline PRP-de pere. Tõestada, et F on $(q,t,e-q^2 2^{-\ell-1})$-turvaline PRF-ide pere.

Konkreetse plokkšifri (näiteks DES-i) tugevus krüptoanalüütiliste rünnete vastu annab meile ligikaudsed väärtused (q,t,ε), mille puhul see šiffer on (q,t,ε)-turvaline PRP. Toodud ülesandest saame hinnangu selle šifri turvalisusele PRF-ina.

3.6.1 Loendurrežiimi turvalisus

Definitsioon 3.6.1 Olgu $F:\Sigma^k\times\Sigma^\ell\to\Sigma^L$. Krüptosüsteem

$$\mathrm{Counter\$}(F)=(\mathcal{E},\mathcal{D},\mathcal{K})$$

on defineeritud järgnevalt. Võtmegenereerija $\mathcal{K}$ valib juhuslikult k-bitise võtme ning spetsifitseerib sellega funktsiooni $f=F_K$. Avateksti

$$p=\Sigma^{L\cdot n}=p[1]||\ldots||p[n]$$

vaadeldakse L-bitiste plokkide jadana. Algoritm $\mathcal{E}_K(p)$ valib esmalt juhuslikult ℓ-bitise stringi r ning seejärel tagastab paari (r,c), kus

$$c[i]=f(r+i \mod 2^\ell)\oplus p[i] \enspace ,$$

$i=1,\ldots,n$. Sisendil (r,c) tagastab $\mathcal{D}_K$ avateksti p, kus $p[i]=f(r+i \mod 2^\ell)\oplus c[i]$, $i=1,\ldots,n$.

Värskelt genereeritud ja ühekordselt kasutatavat juhuslikku suurust r nimetatakse nii siin kui ka mujal krüptograafias *nonsiks*.

Loendurrežiimist on olemas ka mäluga versioon.

Definitsioon 3.6.2 Olgu $F : \Sigma^k \times \Sigma^\ell \to \Sigma^L$. Mäluga krüptosüsteem

$$\mathrm{Counter}+(F) = (\mathcal{E}, \mathcal{D}, \mathcal{K})$$

on defineeritud järgnevalt. Krüptosüsteemi initsialiseerimisel väärtustakse spetsiaalne muutuja $\mathtt{ctr} = -1$, mida suurendatakse edasises krüptija poolt iga järgneva ploki krüptimisel. Võtmegenereerija $\mathcal{K}$ valib juhuslikult k-bitise võtme ning spetsifitseerib sellega funktsiooni $f = F_K$. Avateksti

$$p = \Sigma^{L\cdot n} = p[1]||\ldots||p[n]$$

vaadeldakse L-bitiste plokkide jadana. Algoritm $\mathcal{E}_K(p)$ tagastab paari $(\mathtt{ctr}, c)$, kus $c[i] = f(\mathtt{ctr} + i) \oplus p[i]$, $i = 1, \ldots, n$, ning suurendab seejärel muutuja $\mathtt{ctr}$ väärtust n-i võrra. Sisendil (r, c) tagastab $\mathcal{D}_K$ avateksti p, kus $p[i] = f(r+i) \oplus c[i]$, $i = 1, \ldots, n$.

Järgnev teoreem annab alamtõkke krüptosüsteemi Counter\$ turvalisusele juhusliku funktsiooni mudelis, ehk juhul $F = \mathrm{Func}_{\ell,L}$. Mudeli valiku tõttu ei spetsifitseeri me teoreemis eraldaja tööaega, ehkki see on lihtsasti välja arvutatav.

Teoreem 3.6.1 *Leidub algoritm D mis teeb q päringut kogupikkusega μ bitti, nii et* $\mathrm{Adv}^{\mathrm{lor}}_{\mathrm{Counter\$(Func}_{\ell,L})}(D) \geq 0.316 \cdot \frac{\mu(q-1)}{L\cdot 2^\ell}$.

Tõestus. Toodud artiklis [BDJR97]. Iseseisvalt läbi töötada ja võrrelda teoreemi 3.4.1 tõestusega! Töö hõlpsustamiseks märgime, et kui $\overline{n} = \mu/(Lq)$ on plokkide keskmine arv päringu kohta, siis $\mathrm{Adv}^{\mathrm{lor}}_{\mathrm{Counter\$(Func}_{\ell,L})}(D) = \Theta(q^2/2^\ell) \cdot \overline{n}$. □

Toodud teoreemist saame, et isegi juhul kui plokkšiffer on väga hea (tõeliselt juhuslikust funktsioonist paremat šifrit ei saa konstrueerida!) lekib Counter\$ režiim krüptitud andmehulga kasvades üha enam informatsiooni. Järgneva teoreemiga näitame, et sünnipäevarünne on antud juhul ka sisuliselt parim võimalik rünne: sünnipäevarünnet ei saa parandada rohkem kui konstantse kordaja võrra.

Toodud teoreem koosneb kahest osast: osa üks on sõnastatud juhusliku funktsiooni mudelis, ning seega seal toodud ülemtõke on informatsiooniteoreetiliselt tõene kui plokkšiffer on juhuslik funktsioon. Kuidas on lugu aga praktikas, juhul kui plokkšiffer F ei ole juhuslik funktsioon? Intuitiivselt väidab osa üks, et kui Counter\$$(F)$ ei ole turvaline, siis F ei ole juhuslik. Täpse kvantitatiivse vastuse küsimusele "kui kaugel juhuslikust on F" annab osa kaks. Märgime et teise osa tõestus tugineb esimeses osas saadud ülemtõkkele. Seega ei ole juhusliku funktsiooni mudelis (ehk juhusliku oraakli mudelis, *random oracle model*) vaja teoreeme tõestada mitte ainult siis kui “ilma ei osata”: juhusliku funktsiooni mudelis saadud ülemtõkked on tihti orgaaniliselt seotud PRF-ide jaoks kehtivate alamtõketega.

Teoreem 3.6.2
1) Olgu $\delta_{\mathrm{Counter\$}} := \frac{\mu(q-1)}{L2^{\ell}}$. *Olgu D algoritm, mis teeb ülimalt q päringut kogupikkusega ülimalt* μ *bitti. Siis* $\mathrm{Adv}^{\mathrm{lor}}_{\mathrm{Counter\$}(\mathrm{Func}_{\ell,L})}(D) \leq \delta_{\mathrm{Counter\$}}$.
2) Olgu $F : \Sigma^k \times \Sigma^\ell \to \Sigma^L$ — (q', t', ε')*-turvaline pseudojuhuslik funktsioon. Siis leidub konstant* $c \geq 0$*, nii et krüptosüsteem* $\mathrm{Counter\$}(F)$ *on suvalise q jaoks* $(q, t' - c \cdot \frac{\mu}{L}(\ell + L), \mu, 2\varepsilon' + \frac{q\mu}{L \cdot 2^\ell})$*-turvaline vasak-või-parem tähenduses, kus* $\mu = q'\ell$.

Tõestus.
1)[Osa 1]
Olgu (M_i, N_i), $i \in \{1, \ldots, q\}$, algoritmi D oraaklipäringud. Need päringud on juhuslikud muutujad, mis sõltuvad D mündivisetest (st valitud juhuarvudest) ning oraakli vastustest eelmistele päringutele. Olgu r_i i-nda päringu ajal oraakli poolt valitud nonss. Definitsiooni tõttu $|M_i| = |N_i|$, $\forall i$. Olgu $n_i = |N_i|/\ell$ ühe päringu pikkus plokkides. Seega i-nda päringu ajal rakendab oraakel funktsiooni f stringidele $r_i + 1, \ldots, r_i + n_i \in \Sigma^\ell$.

Tuletame meelde, et vasak-või-parem turvalisuse definitsioonis on kaks maailma: maailmas 0 antakse eraldajale must kast funktsiooniga $f \circ \mathrm{left}$, maailmas 1 aga funktsiooniga $f \circ \mathrm{right}$. Tähistagu $\mathrm{Prob}_i[A]$ tõenäosust, et sündmus A on tõene maailmas i, seega näiteks

$$\mathrm{Prob}_1[D = 1] = \mathrm{Prob}[D^{f \circ \mathrm{right}} = 1 : f \stackrel{R}{\leftarrow} \mathrm{Func}_{\ell,L}] \ .$$

Tähistagu järgnevas X sündmust, et kui $(j_1, k_1) \neq (j_2, k_2)$, siis $r_{j_1} + k_1 \neq r_{j_2} + k_2$, kõigi $j_1, j_2 \in \{0, \ldots, q\}$ ja $k_m \in \{0, \ldots, n_{j_m}\}$ jaoks. On selge, et $\mathrm{Prob}_0[X] = \mathrm{Prob}_1[X]$, kuna X ei sõltu oraaklist, vaid ainult nonsside r_i valikutest.

Kuna sündmuse X kehtimisel rakendatakse mõlemas maailmas kujutust f uuele sisendväärtusele igal uuel väljakutsel, siis on ka mõlemas maailmas selle kujutuse väärtus juhuslikult ja ühtlaselt jaotatud üle hulga Σ^L. Seega on krüptogrammi iga plokk võrdne avateksti vastava ploki ja juhusliku suuruse XOR’iga, millest tuleneb et krüptogrammi iga ploki jaotus on sõltumatu eelmistest krüptogrammi plokkidest kui ka avatekstist. Siit saame, et

$$\mathrm{Prob}_0[D = 1|X] = \mathrm{Prob}_1[D = 1|X]$$

ning seega

$$
\begin{aligned}
\mathrm{Adv}^{\mathrm{lor}}_{\mathrm{Counter\$(Func}_{\ell,L})}(D) =& \mathrm{Prob}_0[D=1]-\mathrm{Prob}_1[D=1]\\
=& \mathrm{Prob}_0[D=1|\overline{X}]\mathrm{Prob}_0[\overline{X}]+\mathrm{Prob}_0[D=1|X]\mathrm{Prob}_0[X]-\\
& \mathrm{Prob}_1[D=1|\overline{X}]\mathrm{Prob}_1[\overline{X}]-\mathrm{Prob}_1[D=1|X]\mathrm{Prob}_1[X]\\
=& (\mathrm{Prob}_0[D=1|\overline{X}]-\mathrm{Prob}_1[D=1|\overline{X}])\mathrm{Prob}_0[\overline{X}]+\\
& \underbrace{(\mathrm{Prob}_0[D=1|X]-\mathrm{Prob}_1[D=1|X])}_{=0}\mathrm{Prob}_0[X]\\
=& \underbrace{(\mathrm{Prob}_0[D=1|\overline{X}]-\mathrm{Prob}_1[D=1|\overline{X}])}_{\leq 1}\mathrm{Prob}_0[\overline{X}]\\
\leq& \mathrm{Prob}_0[\overline{X}]\ .
\end{aligned}
$$

Seega on vaja osa üks tõestuseks anda veel ülemtõke suurusele

$$\mathrm{Prob}[\overline{X}] := \mathrm{Prob}_0[\overline{X}] = \mathrm{Prob}_1[\overline{X}]$$

ehk kollisioonide tõenäosusele erinevate r_i+k vahel, kus kõik r_i-d on juhuslikult valitud. Märgime, et kollisioon i-nda ja j-nda päringu vahel tekib siis kui $r_i \in [r_j+1-n_i, r_j+n_j] =: I_j$. Olgu p_i tõenäosus, et i-nda päringu tegemisel tekib kollisioon. Eelnevast saame, et p_i on maksimaalne juhul kui $I_i \cap I_j = \emptyset$, $\forall i \neq j$. Järelikult

$$p_i \leq \frac{\sum_{j=1}^{i-1}(n_j+n_i-1)}{2^\ell} = \frac{(i-1)(n_i-1)+\sum_{j=1}^{i-1} n_j}{2^\ell}$$

ning seega

$$
\begin{aligned}
\mathrm{Prob}_0[\overline{X}] \leq \sum_{i=1}^{q} p_i \leq & \sum_{i=1}^{q} \frac{(i-1)(n_i-1)+\sum_{j=1}^{i-1} n_j}{2^\ell}\\
=& \frac{\sum_{i=1}^{q}(in_i-n_i-i+1)+\sum_{i=1}^{q-1}(qn_i-in_i)}{2^\ell}\\
=& \frac{(q-1)\frac{\mu}{L}-\frac{q(q-1)}{2}}{2^\ell} \leq \frac{\mu(q-1)}{L\cdot 2^\ell}
\end{aligned}
$$

ning seega $\mathrm{Adv}^{\mathrm{lor}}_{\mathrm{Counter\$(Func}_{\ell,L})}(D) \leq \frac{\mu(q-1)}{L\cdot 2^\ell}$.

2) [Osa 2]

Vastuväiteline. Oletame, et leidub (q,t,μ)-algoritm D_i, mille edu krüptosüsteemi

$$\Pi(F) = \mathrm{Counter\$}(F)$$

Algoritm 3.4 Teoreemis 3.6.2 kasutatav eraldaja D_F^f.

1. $b \xleftarrow{R} \{0,1\}$.
2. Olgu $f' := \begin{cases} f \circ \text{left}, & b=0 \\ f \circ \text{right}, & b=1 \end{cases}$.
3. Olgu $d \xleftarrow{R} D_{\text{Counter\$}}^{f'}$.
4. Kui $d=b$, väljasta 1, muidu 0.

vastu vasak-või-parem tähenduses on ε. Vaatleme algoritmis 3.4 kirjeldatud eraldajat D_F, mis üritab murda pseudojuhuslike funktsioonide peret F. Vastates oraaklipäringule (x_0,x_1) teeb eraldaja D_F $n := |M_1|/L = |M_2|/L$ oraaklipäringut oraaklile f. Seega on D_F-i poolt tehtud oraaklipäringute arv ülimalt μ/L. Algoritmi D_F käitamisaeg on $t + c\cdot(\mu/L)(\ell+L)$ (oraaklipäringud võtavad ühikaja!). Oraakli D_F eduks $\text{Adv}_F^{\text{prf}}(D_F)$ on järgmise summa absoluutväärtus:

$$
\begin{aligned}
&\text{Prob}[D_F^f = 1 : f \xleftarrow{R} F] - \text{Prob}[D_F^f = 1 : f \xleftarrow{R} \text{Func}_{\ell,L}] \\
&\quad= \frac{1}{2}\text{Prob}[D_F^f = 1 | b=1 : f \xleftarrow{R} F] + \frac{1}{2}\text{Prob}[D_F^f = 1 | b=0 : f \xleftarrow{R} F] - \\
&\qquad \frac{1}{2}\text{Prob}[D_F^f = 1 | b=1 : f \xleftarrow{R} \text{Func}_{\ell,L}] - \frac{1}{2}\text{Prob}[D_F^f = 1 | b=0 : f \xleftarrow{R} \text{Func}_{\ell,L}] \\
&\quad= \frac{1}{2}\Big(\text{Prob}[D_i^{f\circ\text{right}} = 1 : f \xleftarrow{R} F] + 1 - \text{Prob}[D_i^{f\circ\text{left}} = 1 : f \xleftarrow{R} F]\Big) - \\
&\qquad \frac{1}{2}\Big(\text{Prob}[D_i^{f\circ\text{right}} = 1 : f \xleftarrow{R} \text{Func}_{\ell,L}] + 1 - \text{Prob}[D_i^{f\circ\text{left}} = 1 : f \xleftarrow{R} \text{Func}_{\ell,L}]\Big) \\
&\quad= \frac{1}{2}(\text{Prob}[D_i^{f\circ\text{right}} = 1 : f \xleftarrow{R} F] - \text{Prob}[D_i^{f\circ\text{left}} = 1 : f \xleftarrow{R} F] - \\
&\qquad \text{Prob}[D_i^{f\circ\text{right}} = 1 : f \xleftarrow{R} \text{Func}_{\ell,L}] + \text{Prob}[D_i^{f\circ\text{left}} = 1 : f \xleftarrow{R} \text{Func}_{\ell,L}]) \\
&\quad= \frac{1}{2}(\text{Adv}_{\Pi(F)}^{\text{lor}}(D_i) - \text{Adv}_{\Pi(\text{Func}_{\ell,L})}^{\text{lor}}(D_i))
\end{aligned}
$$

Osast 1 saame, et $\text{Adv}_{\Pi(\text{Func}_{\ell,L})}^{\text{lor}}(D_i) \le \delta_{\text{Counter\$}}$. Eelduse põhjal

$$\text{Adv}_{\Pi(F)}^{\text{lor}}(D_i) > \varepsilon$$

ning seega

$$\mathrm{Adv}^{\mathrm{prf}}_F(D_F) > |\varepsilon/2 - \delta_{\mathrm{Counter\$}}/2| \ .$$

□

Selgub, et mäluga versioon loendurrežiimist on tunduvalt parem: juhusliku funktsiooni mudelis on eraldaja eduks null. Ka PRF-ide korral on eraldaja edu tunduvalt väiksem.

Teoreem 3.6.3 *1) Olgu D suvaline oraakliga algoritm mis teeb ülimalt q päringut kogupikkusega ülimalt $\mu < L \cdot 2^\ell$ bitti. Siis* $\mathrm{Adv}^{\mathrm{lor}}_{\mathrm{Counter}+(\mathrm{Func}_{\ell,L})}(D) = 0$.
2) Olgu $F : \Sigma^k \times \Sigma^\ell \to \Sigma^L$ — (q',t',ε')-turvaline pseudojuhuslike funktsioonide pere. Siis suvalise q jaoks $\mathrm{Counter}+(F)$ *on $(q, t' - c\frac{\mu}{L}(\ell + L), \mu, 2\varepsilon')$-turvaline, kus $\mu = \min(q'L, L \cdot 2^\ell)$.*

Tõestus. Tõestus on artiklis [BDJR97]. □

Tudengile on jäetud ülesandeks kõik tõestused läbi närida.

3.6.2 Ahelrežiimi turvalisus

Konspekt — lühiversioon. Artikli [BDJR97] paragrahvis 4.3 on toodud järgmine teoreem.

Teoreem 3.6.4 *Olgu $F : \Sigma^k \times \Sigma^\ell \to \Sigma^\ell$ (t',q',ε')-turvaline pseudojuhuslik funktsioon. Siis leidub konstant $c \geq 0$, nii et krüptosüsteem CBC(F) on (t,q,μ,ε)-turvaline vasak-või-parem tähenduses, kus $t = t' - c \cdot \mu$, $\mu = q'\ell$ ning $\varepsilon = \varepsilon' + \frac{2\mu^2}{\ell^2 2^\ell}$.*

Artiklis on toodud ka ründed, mis on küllalt täpselt vastavuses toodud turvalisuse ülemtõketega: täielik võtmeotsing loendurrežiimi ning sünnipäevarünne ahelrežiimi korral.

Ülesanne 3.6.2 Töötada iseseisvalt läbi artikli [BDJR97] peatükk neli ja vastavad tõestused lisas B.

Ülesanne 3.6.3 Vaatame teoreemi 14 [BDJR97]. Oletame, et F pole mitte PRF-ide pere, vaid (q',t',ε')-turvaline PRP-de pere (plokkšiffer). Mis kitsendused peab peale panema erinevatele F-i parameetritele, et tõestus läheks endiselt läbi?

Eelnevatest tulemustest võime järeldada, et mäluga loendurrežiim on oluliselt turvalisem kui ahelrežiim. Loendurrežiim on ka tunduvalt efektiivsem kui ahelrežiim, seega võiks ja tuleks kõikides rakendustes just seda kasutada — kahjuks pole seda režiimi aga standarditud, seega kannataks sellisel juhul ühilduvus. Kuna loendurrežiimi turvalisus tuleneb just PRF-ide (suure tõenäosusega mittepööratavate funktsioonide!) kasutamisest, tekib siinkohas küsimus plokkšifrite mõttekusest üldse. Siiski: toodud tõestused baseeruvad eeldusel et F on ideaalselt turvaline, ning kaitsevad ainult üldiste, šifrite konstruktsiooni mittekasutavate, rünnete (nagu näiteks sünnipäevarünne) vastu. Plokkšifreid on aga piisavalt palju uuritud, neid osatakse tänapäeval teha turvaliseks ka krüptoanalüütiliste rünnete vastu. Üheks võimaluseks oleks mittepööratavate PRF-ide konstrueerimine plokkšifritest: on olemas mitmeid efektiivseid konstruktsioone, mis teevad turvalisest PRP-de perest turvalisuse kadudeta PRF-ide pere (teame, et iga PRP-de pere on automaatselt PRF-ide pere, aga teatud turvakaoga — seesama lause 9). Huvitatud tudeng võib lugeda artiklit [BKR98]. (mittekohustuslik!)

Peatükk 4

Praktiliste plokkšifrite koostamine

4.1 DES

Põhineb raamatul [Sti95].

DES kui standard. Ajalugu. Feisteli šifrid. DESi enda lühikirjeldus. Miks selline?

Loengus anti paljundatud materjal raamatust [Sti95]. Põhjendasin jupp-jupi haaval, miks võib üks plokkšiffer selline olla. Lisaks rääkisin DES-i ajaloost ning selle loomise ümber valitsevast müstikast.

Järgnevalt toome edaspidi tihti vajamineva DES-i üldskeemi (joonis 4.1). Lühike sisuseletus:

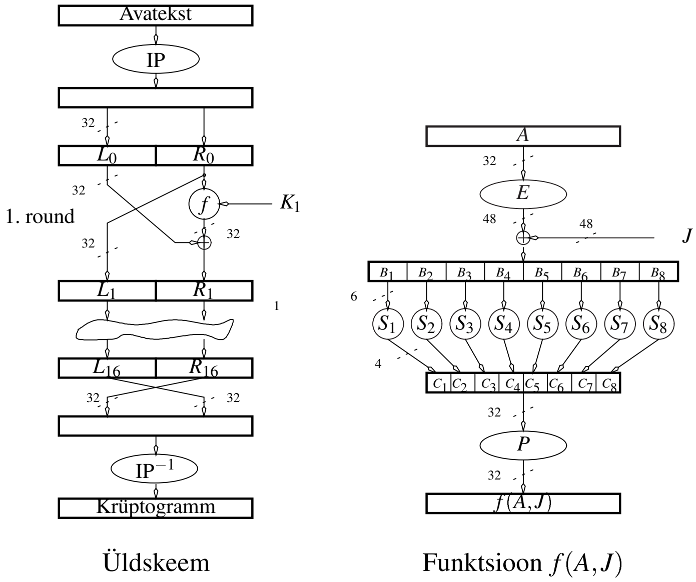
1. DES-i 56-bitisest võtmest genereeritakse 16 48-bitist raundivõtit k_1, ..., k_{16}, kasutades võtmelaiendusalgoritmi.

2. Krüptimise alguses rakendatakse 64-bitisele avatekstile permutatsiooni IP, mis kopeerib avateksti bitid mingitesse bitipositsioonidesse. Saadud 64-bitine tulemus $\mathrm{IP}(X)$ kirjutatakse lahti kui kahe 32-bitise arvu konkatenatsioon: $\mathrm{IP}(X) = L_0||R_0$.

3. Seejärel täidetakse 16 roundi, $i \in [1, 16]$. $i$nda raundi sisendiks on konkatenatsioon $L_{i-1}||R_{i-1}$ ning raundivõti k_i. Raundis täidetakse järgmised operatsioonid:

   (a) Olgu $L_i := R_{i-1}$.

   (b) Olgu $R_i := f(R_{i-1}, k_i) \oplus k_i$, kus f on *raundifunktsioon*.



Joonis 4.1: DES-i üldskeem.

4. Krüptogrammiks on bitistring $\mathrm{IP}^{-1}(R_{16}||L_{16})$.

Funktsioon f sisendipaaril (A,J) on defineeritud järgnevalt:

1. Olgu $B = E(A) \oplus J$, kus E on fikseeritud laiendav bitipermutatsioon (B on 48-bitine arv, E on bitipermutatsioon, mis permuteerib kõik 32 sisendbitti, kopeerides neist pooled kahekordselt).

2. Kirjutame $B = B_1||B_2||B_3||B_4||B_5||B_6||B_7||B_8$, kus B_i on 6-bitine arv.

3. Iga $i \in [1,8]$ korral olgu $C_i = S_i(B_i)$, kus S_i on is S-box (mittelineaarne funktsioon $S_i : 2^6 \to 2^4$, määratud enamasti oma väärtuste tabeliga). Olgu
$$C = C_1||C_2||\dots||C_8 \ .$$
Edasises kirjutame ka tihti $C = S(B)$.

4. Olgu $f(A,J) := P(C)$, kus P on fikseeritud bitipermutatsioon.

(kõik puuduvad detailid leiduvad raamatus [Sti95]. Oluline on veel teada, et raundivõtmed k_i genereeritakse võtmelaiendusalgoritmi poolt, sh on k_i võtme k bittide permuteeritud alamhulga konkatenatsioon).

DES on äärmiselt efektiivne riistvaras, kuna ta sisaldab riistvaras kiiresti implementeeritavaid operatsioone: bitipermutatsioonid, "XOR"-tehted ning S-boxid (viimased on oma väiksuse tõttu realiseeritavad massiivist väärtuse lugemise abil — ehk nn *table-lookup*). See-eest tarkvaras on DES-i realiseerimine tunduvalt ebamugavam: "triviaalse lahenduse" korral kulub (näiteks) permutatsiooni P tulemi leidmiseks $32 \cdot c$ sammu, kus c on mingi masinast sõltuv konstant. Õnneks saab DES-is esinevaid permutatsioone tunduvalt kiiremini välja arvutada, kasutades tänapäevastel masinatel olemasolevaid 64-bitiseid tehteid (liitmine, lahutamine, Boole'i tehted, nihked ja rotatsioonid). Nii näiteks on võimalik permutatsiooni E välja arvutada C keeles kasutades (ülimalt) kaheksat 64-bitist "AND" ja "OR"-tehet, ülimalt 10 nihutamist ning ülimalt kümmet omistamist.

Ülesanne 4.1.1 Leida kiire implementatsioon kõikide ülejäänud DESis esinevate permutatsioonide (PC1, PC2, IP, IP^{-1}, P) jaoks, st igat permutatsiooni tuleb implementeerida eraldi. Kasutada võib mainitud 64-bitiseid tehteid, ≤ 256-baidiseid massiive ning muid C keeles olemasolevaid vahendeid.

4.2 Mõned DES-i omadused

Eelmistes peatükkides rääkisime sellest, kuidas konstrueerida PRP-dest ja PRF-idest lähtuvalt krüptosüsteeme. Selline käsitlus on küllaltki teoreetiline, kuna sellisel juhul eeldatakse alati mingi turvalisuse astmega PRF-i (PRP) olemasolu. Kuigi teoreetiliselt leiduvad ka $(\infty, \infty, 0)$-turvalised PRP-d (nimelt juhuslikud permutatsioonid), ei piisa nendest praktikas, kus on vaja lühikese võtmega, kiireid ja turvalisi plokkšifreid: seega on vaja kandidaate, mille peale saab hiljem ehitada muid primitiive. Paragrahvis 4.1 kirjeldasime ühe sellise kandidaatplokkšifri, DES-i, kuid jätsime täielikult käsitlemata tema turvalisuse. Osaliselt teeme seda käesolevas paragrahvis: nimelt näitame, et leiduvad ründed < 16-*raundiste* DES-i versioonide vastu, mis töötavad kiiremini kui täielik läbivaatus — sellega saab hakkama erinevustega krüptoanalüüs (*differential cryptanalysis*). Kuid esmalt sõnastame mõned triviaalsed DES-i nõrgad kohad.

Edasises kasutame tihti vaikimisi järgmist tähistust. Olgu (p, k) mingi fikseeritud avateksti ja võtme paar. Olgu p^* ja k^* mingi teine avateksti paar. Vaatleme DES-i struktuuri sisemisi muutujaid (IP(p), L_0, $\ldots$) nii sisendite (p, k) kui ka (p^*, k^*) puhul: viimase sisendi korral märgime vastavad muutujad alati tärniga (st $\mathrm{IP}(p)^* = \mathrm{IP}(p^*)$, $L_0^* = \mathrm{IP}(p)^*/2^{32}$, $\ldots$). Suvalise muutuja x korral olgu $\Delta x := x \oplus x^*$ (näiteks $\Delta L_0 := L_0 \oplus L_0^*$).

Lemma 4.2.1 (DES-i täiendiomadus) *Tähistagu $\overline{x}$ bitistringi x pöördväärtust (st $x \oplus \overline{x} = 1 \ldots 1$). Tähistagu* $\mathrm{DES}(p,k)$ *avateksti k DES-iga krüptimisel võtmega k saadud krüptogrammi. Siis suvalise p ja k korral* $\overline{\mathrm{DES}(p,k)} = \mathrm{DES}(\overline{p},\overline{k})$.

Tõestus. Olgu (p,k) fikseeritud ning olgu $(p^*,k^*) = (p,k)$. Tõestame järgnevalt induktsiooniga i järgi, et $(L_i^*,R_i^*) = (\overline{L_i},\overline{R_i})$ ehk

$$(\Delta L_i, \Delta R_i) = \mathtt{0xFFFFFFFF}, \mathtt{0xFFFFFFFF}) \ .$$

Sellest piisab ka lemma tõestuseks: tõepoolest, saame, et

$$(L_{16}^*, R_{16}^*) = (\overline{L_{16}}, \overline{R_{16}})$$

ning kuna IP^{-1} on bitipermutatsioon, siis

$$\mathrm{DES}(p^*,k^*) = \mathrm{IP}^{-1}(R_{16}^*, L_{16}^*) = \mathrm{IP}^{-1}(\overline{R_{16}}, \overline{L_{16}}) = \overline{\mathrm{IP}^{-1}(R_{16}, L_{16})} = \overline{\mathrm{DES}(p,k)} \ .$$

<u>Baas</u>. Kuna IP on bitipermutatsioon, siis

$$\mathrm{IP}(p)^* = \mathrm{IP}(p^*) = \overline{\mathrm{IP}(p)}$$

ning seega ka $(L_0^*, R_0^*) = (\overline{L_0}, \overline{R_0})$.
<u>Samm</u>. Eeldame, et $(L_{i-1}^*, R_{i-1}^*) = (\overline{L_{i-1}}, \overline{R_{i-1}})$. Triviaalselt $L_i^* = R_{i-1}^* = \overline{R_{i-1}} = \overline{L_i}$. Kuna $R_i^* = \overline{R_i}$, siis (vt joonis 4.1, parem, tähistusi) $A^* = R_i^* = \overline{R_i} = \overline{A}$. Kuna E on bitipermutatsioon, siis $E(A)^* = \overline{E(A)}$. Võtmelaiendusalgoritmi järgi ([Sti95, lk 75]) on raundivõti J võtme k mingi bittide alamhulga permutatsiooni. Seega $J^* = \overline{J}$, ning järelikult on S-boxide sisend $B^* = E(A)^* \oplus J^* = \overline{E(A)} \oplus \overline{J} = E(A) \oplus J = B$. Seega $B = B^*$ (ehk $\Delta B = 0$ — ütleme, et S-boxide *sisend-xor* on 0), millest järeldub et ka

$$\Delta C = S(B) \oplus S(B^*) = S(B) \oplus S(B) = 0$$

(ütleme, et S-boxide *väljund-xor* on 0). Kuna P on jällegi bitipermutatsioon, siis

$$\Delta P(C) = P(C) \oplus P(C^*) = P(C) \oplus P(C) = 0$$

ning seega

$$\Delta f(R_{i-1}, k_i) = f(R_{i-1}, k_i) \oplus f(R_{i-1}^*, k_i^*) = 0 \ .$$

Kuna $L_{i-1} = \overline{L_{i-1}^*}$, siis

$$R_i^* = L_{i-1}^* \oplus f(R_{i-1}, k_i)^* = \overline{L_{i-1}} \oplus f(R_{i-1}, k_i) = \overline{L_{i-1} \oplus f(R_{i-1}, k_i)} = \overline{R_i} \ .$$

□

Antud lemmast järeldub järgmine tähelepanek (oluline edaspidi mõistmaks erinevustega krüptoanalüüsi ideed). Teadku ründaja avateksti ja krüptogrammi paari (p,c), nii et $c = \mathrm{DES}(p,k)$ mingi ründajale tundmatu võtme k korral. Eeldame nagu alati, et ründaja teab DES-i siseehitust, mh oskab ta alati välja arvutada suurust $\mathrm{DES}(p_1,k_1)$ enda poolt valitud suuruste p_1 ja k_1 korral. Kui DES oleks juhuslik permutatsioon, peaks ründaja tegema halvimal juhul kuni 2^{56} päringut DES-i poole (st sooritama täieliku läbivaatuse), leidmaks võtit k. Täiendiomadus aitab ründajat siis, kui ta teab mingi fikseeritud avateksti p korral avateksti-krüptogrammi paare (p,c_1) ning $(\overline{p},c_2)$: sellisel juhul piisab sellest, kui ründaja otsib läbi $\leq 2^{55}$ võtit, iga võtme k_1 korral kontrollides, kas $\mathrm{DES}(p,k_1) \in \{c_1,c_2\}$. Tegemist on puhtalt teoreetilise (aga siiski!) nõrkusega, kuna ründaja peab kaks korda vähem DES-iga krüptima, kuid samas kulub tal lisaaeg muude operatsioonide tegemiseks ning ta peab teadma paari (p,c_1) ja $(\overline{p},c_2)$.

Teine DES-i üldtuntud nõrkus on (enese-)duaalsete võtmete olemasolu. Nagu loengutes korduvalt märgitud, kasutatakse turvalisuse tagamiseks tihti kordšifreid (DES ise on selline!), nii et (konkreetselt DES-i korral)

$$F(p,k_1||k_2) = \mathrm{DES}(\mathrm{DES}(p,k_1),k_2) \ .$$

Ütleme, et võtmed k_1 ja k_2 on *duaalsed*, kui $F(p,k_1||k_2) = p$ suvalise p korral (st $\mathrm{DES}(\cdot,k_2) = \mathrm{DES}^{-1}(\cdot,k_1)$). Võti k on *eneseduaalne*, kui ta on iseenda duaalne võti. Kuna duaalse võtmepaari kasutamisel on krüptogramm võrdne avatekstiga, muutub kordšiffer äärmiselt ebaturvaliseks.

Ülesanne 4.2.1 1) (vt [Sti95, lk 75]!). Tõesta, et kui $C_0, D_0 \in \{0\ldots0, 1\ldots1\}$, siis K on eneseduaalne.
2) Tõestada, et järgmised võtmed on eneseduaalsed:

$$\texttt{0x0101010101010101},\ \texttt{0xFEFEFEFEFEFEFEFE}\ ,$$
$$\texttt{0x1F1F1F1F0E0E0E0E},\ \texttt{0xE0E0E0E0F1F1F1F1}\ .$$

3) Tõestada, et kui $C_0 \in \{\texttt{0x55555555}, \texttt{0xAAAAAAAA}\}$, siis

$$C_i \oplus C_{17-i} = \texttt{0xFFFFFFFF}\ ,$$

$1 \leq i \leq 16$. (Sama on tõene D_i-de kohta).
4) Tõestada, et järgmised võtmepaarid on duaalsed:

$$(\texttt{E001E001F101F101},\ \texttt{01E001E001F101F1})$$
$$(\texttt{FE1FFE1FFE0EFE0E},\ \texttt{1FFE1FFE0EFE0EFE})$$
$$(\texttt{E01FE01FF10EF10E},\ \texttt{1FE01FE00EF10EF1})$$

4.3 Erinevustega krüptoanalüüs

Erinevustega krüptoanalüüs (differential cryptanalysis) on täiendiomadust kasutavale ründele mõneti sarnane ründetüüp: teades (suurt hulka) avateksti-krüptogrammi paare (p_i, c_i) ja (p_i^*, c_i^*), nii et $(\Delta p_i, \Delta c_i) = (\Delta p, \Delta c)$ on sobivalt valitud, vähendatakse läbivaatamiseks vajalike võtmete hulka oluliselt. DES-i enda puhul ei ole erinevustega krüptoanalüüs väga praktiline, kuna ta nõuab 2^{47} *valitud* avateksti-krüptogrammi paari ning ka 2^{47} päringut DES-ile. Samas on tulemus oluliselt parem, kui täiendiomadusest järelduv (keskväärtus) 2^{54}. Samuti on erinevustega krüptoanalüüsi kasutatud mõnede teiste šifrite ning DES-i variatsioonide lahtimurdmiseks. Tänu erinevustega krüptoanalüüsi edukusele on tänapäeval uute plokkšifrite väljatöötamisel üheks oluliseks kriteeriumiks plokkšifri turvalisus selle ründetüübi vastu. Teine, DES-i vastu veelgi edukam, ründetüüp on lineaarne krüptoanalüüs, mida me selle loengukursuse jooksul ei jõua vaadelda.

Käesolevas paragrahvis käsitleme erinevustega krüptoanalüüsi kolmeraundise DESi näitel, kus n-raundine DES on defineeritud järgmiselt: $\mathrm{DES}_n(p,k) = \mathrm{IP}^{-1}(R_n, L_n)$. Kui täiendiomadust ärakasutav rünne ei sõltunud konkreetsete permutatsioonide valikust ega ka S-boxidest, on erinevustega krüptoanalüüs (lühend EK) väga intiimselt seotud kõigi DES-i komponentidega, eriti aga just S-boxide, DES-i ainsate mittelineaarsete aluskomponentide, ehitusega.

Nagu ennegi tähistab muutuja x korral x^* teist samapikka muutujat ning $\Delta x := x \oplus x^*$. Edasises eeldame, et erinevate krüptogrammide moodustamiseks kasutatud võti k on alati sama (st $k = k^*$ ning $\Delta k = 0$). Fikseeritud S-boxi S_j ning kahe kuuebitise sisendi (B_j, B_j^*) korral ütleme, et (S_j-i) *sisend-xor* on $\Delta B_j = B_j \oplus B_j^* \in \mathbb{Z}_2^6$ ning *väljund-xor* on $S_j(B_j) \oplus S_j(B_j^*) \in \mathbb{Z}_2^4$. Suvalise $\Delta B_j \in \mathbb{Z}_2^6$ korral olgu

$$\Delta(\Delta B_j) := \{(B_j, B_j \oplus \Delta B_j : B_j \in \mathbb{Z}_2^6\} \enspace .$$

Seega $|\Delta(\Delta B_j)| = 2^{64}$. Iga sisend-xori ΔB_j jaoks on saame leida kõikide seda sisend-xori omavate paari (B_j, B_j^*) väljund-xorid[1] ning tabuleerida saadud jaotuse. Olgu

$$IN_j(\Delta B_j, \Delta C_j) := \{B_j \in \mathbb{Z}_2^6 : S_j(B_j) \oplus S_j(B_j \oplus \Delta B_j) = \Delta C_j\}$$

ning olgu $N_j(\Delta B_j, \Delta C_j) := |IN_j(\Delta B_j, \Delta C_j)|$. Kuna $\Delta B_j \in \mathbb{Z}_2^6$ ning $j \in [1,8]$, saame 512 jaotust ning $512 \cdot 16 = 8192$ erinevat hulka $IN_j(\Delta B_j, \Delta C_j)$.

kujul “fikseeritud S-boxi S_j ning sisend-xori ΔB_j korral on $N_j(\Delta B_j, \Delta C_j)$ paari hulgas $\Delta(\Delta B_j)$. Alati kehtib, et $2 \mid N_j(\Delta B_j, \Delta C_j)$, kuna iga $B_j \in IN_j(\Delta B_j, \Delta C_j)$

[1] Vastavad alamprogrammid olid olemas eelmisel loengul kättejaotatud programmis `des.C`.

jaoks ka $B_j \oplus \Delta B_j \in IN_j(\Delta B_j, \Delta C_j)$, ning $B_j = B_j \oplus \Delta B_j \iff \Delta B_j = 0$, viimasel juhul aga $N_j(\Delta B_j, \Delta C_j) = 64$. Väärtusi $N_j(\Delta B_j, \Delta C_j)$ analüüsides on lihtne leida, et

$$N_j(\Delta B_j, \Delta C_j) \leq 16$$

kui $\Delta B_j \neq 0$. Kokku saame, et $\approx 22.7\%$ hulkadest IN_j on tühihulgad, st 22.7% juhtudel mingile S-boxile mingi sisend-xori sisseandmisel ei saa me kunagi mingit väljund-xori. Järgmised mõned leheküljed sisaldavad arvutiprogrammi väljundit, kus on toodud järjest $N_j(\Delta B_j, \Delta C_j)$ väljundid $\Delta C_j \in [0, 15]$ jaoks.

$N_1(00,\cdot) = 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0$

$N_1(01,\cdot) = 0, 0, 0, 6, 0, 2, 4, 4, 0, 10, 12, 4, 10, 6, 2, 4$

$N_1(02,\cdot) = 0, 0, 0, 8, 0, 4, 4, 4, 0, 6, 8, 6, 12, 6, 4, 2$

$N_1(03,\cdot) = 14, 4, 2, 2, 10, 6, 4, 2, 6, 4, 4, 0, 2, 2, 2, 0$

$N_1(04,\cdot) = 0, 0, 0, 6, 0, 10, 10, 6, 0, 4, 6, 4, 2, 8, 6, 2$

$N_1(05,\cdot) = 4, 8, 6, 2, 2, 4, 4, 2, 0, 4, 4, 0, 12, 2, 4, 6$

$N_1(06,\cdot) = 0, 4, 2, 4, 8, 2, 6, 2, 8, 4, 4, 2, 4, 2, 0, 12$

$N_1(07,\cdot) = 2, 4, 10, 4, 0, 4, 8, 4, 2, 4, 8, 2, 2, 2, 4, 4$

$N_1(08,\cdot) = 0, 0, 0, 12, 0, 8, 8, 4, 0, 6, 2, 8, 8, 2, 2, 4$

$N_1(09,\cdot) = 10, 2, 4, 0, 2, 4, 6, 0, 2, 2, 8, 0, 10, 0, 2, 12$

$N_1(0a,\cdot) = 0, 8, 6, 2, 2, 8, 6, 0, 6, 4, 6, 0, 4, 0, 2, 10$

$N_1(0b,\cdot) = 2, 4, 0, 10, 2, 2, 4, 0, 2, 6, 2, 6, 6, 4, 2, 12$

$N_1(0c,\cdot) = 0, 0, 0, 8, 0, 6, 6, 0, 0, 6, 6, 4, 6, 6, 14, 2$

$N_1(0d,\cdot) = 6, 6, 4, 8, 4, 8, 2, 6, 0, 6, 4, 6, 0, 2, 0, 2$

$N_1(0e,\cdot) = 0, 4, 8, 8, 6, 6, 4, 0, 6, 6, 4, 0, 0, 4, 0, 8$

$N_1(0f,\cdot) = 2, 0, 2, 4, 4, 6, 4, 2, 4, 8, 2, 2, 2, 6, 8, 8$

$N_1(10,\cdot) = 0, 0, 0, 0, 0, 0, 2, 14, 0, 6, 6, 12, 4, 6, 8, 6$

$N_1(11,\cdot) = 6, 8, 2, 4, 6, 4, 8, 6, 4, 0, 6, 6, 0, 4, 0, 0$

$N_1(12,\cdot) = 0, 8, 4, 2, 6, 6, 4, 6, 6, 4, 2, 6, 6, 0, 4, 0$

$N_1(13,\cdot) = 2, 4, 4, 6, 2, 0, 4, 6, 2, 0, 6, 8, 4, 6, 4, 6$

$N_1(14,\cdot) = 0, 8, 8, 0, 10, 0, 4, 2, 8, 2, 2, 4, 4, 8, 4, 0$

$N_1(15,\cdot) = 0, 4, 6, 4, 2, 2, 4, 10, 6, 2, 0, 10, 0, 4, 6, 4$

$N_1(16,\cdot) = 0, 8, 10, 8, 0, 2, 2, 6, 10, 2, 0, 2, 0, 6, 2, 6$

$N_1(17,\cdot) = 4, 4, 6, 0, 10, 6, 0, 2, 4, 4, 4, 6, 6, 6, 2, 0$

$N_1(18,\cdot) = 0, 6, 6, 0, 8, 4, 2, 2, 2, 4, 6, 8, 6, 6, 2, 2$

$N_1(19,\cdot) = 2, 6, 2, 4, 0, 8, 4, 6, 10, 4, 0, 4, 2, 8, 4, 0$

$N_1(1a,\cdot) = 0, 6, 4, 0, 4, 6, 6, 6, 6, 2, 2, 0, 4, 4, 6, 8$

$N_1(1b,\cdot) = 4, 4, 2, 4, 10, 6, 6, 4, 6, 2, 2, 4, 2, 2, 4, 2$

$N_1(1c,\cdot) = 0, 10, 10, 6, 6, 0, 0, 12, 6, 4, 0, 0, 2, 4, 4, 0$

$N_1(1d,\cdot) = 4, 2, 4, 0, 8, 0, 0, 2, 10, 0, 2, 6, 6, 6, 14, 0$

$N_1(1e,\cdot) = 0, 2, 6, 0, 14, 2, 0, 0, 6, 4, 10, 8, 2, 2, 6, 2$

$N_1(1f,\cdot) = 2, 4, 10, 6, 2, 2, 2, 8, 6, 8, 0, 0, 0, 4, 6, 4$

$N_1(20,\cdot) = 0, 0, 0, 10, 0, 12, 8, 2, 0, 6, 4, 4, 4, 2, 0, 12$

$N_1(21,\cdot) = 0, 4, 2, 4, 4, 8, 10, 0, 4, 4, 10, 0, 4, 0, 2, 8$

$N_1(22,\cdot) = 10, 4, 6, 2, 2, 8, 2, 2, 2, 2, 6, 0, 4, 0, 4, 10$

$N_1(23,\cdot) = 0, 4, 4, 8, 0, 2, 6, 0, 6, 6, 2, 10, 2, 4, 0, 10$

$N_1(24,\cdot) = 12, 0, 0, 2, 2, 2, 2, 0, 14, 14, 2, 0, 2, 6, 2, 4$

$N_1(25,\cdot) = 6, 4, 4, 12, 4, 4, 4, 10, 2, 2, 2, 0, 4, 2, 2, 2$

$N_1(26,\cdot) = 0, 0, 4, 10, 10, 10, 2, 4, 0, 4, 6, 4, 4, 4, 2, 0$

$N_1(27,\cdot) = 10, 4, 2, 0, 2, 4, 2, 0, 4, 8, 0, 4, 8, 8, 4, 4$

$N_1(28,\cdot) = 12, 2, 2, 8, 2, 6, 12, 0, 0, 2, 6, 0, 4, 0, 6, 2$

$N_1(29,\cdot) = 4, 2, 2, 10, 0, 2, 4, 0, 0, 14, 10, 2, 4, 6, 0, 4$

$N_1(2a,\cdot) = 4, 2, 4, 6, 0, 2, 8, 2, 2, 14, 2, 6, 2, 6, 2, 2$

$N_1(2b,\cdot) = 12, 2, 2, 2, 4, 6, 6, 2, 0, 2, 6, 2, 6, 0, 8, 4$

$N_1(2c,\cdot) = 4, 2, 2, 4, 0, 2, 10, 4, 2, 2, 4, 8, 8, 4, 2, 6$

$N_1(2d,\cdot) = 6, 2, 6, 2, 8, 4, 4, 4, 2, 4, 6, 0, 8, 2, 0, 6$

$N_1(2e,\cdot) = 6, 6, 2, 2, 0, 2, 4, 6, 4, 0, 6, 2, 12, 2, 6, 4$

$N_1(2f,\cdot) = 2, 2, 2, 2, 2, 6, 8, 8, 2, 4, 4, 6, 8, 2, 4, 2$

$N_1(30,\cdot) = 0, 4, 6, 0, 12, 6, 2, 2, 8, 2, 4, 4, 6, 2, 2, 4$

$N_1(31,\cdot) = 4, 8, 2, 10, 2, 2, 2, 2, 6, 0, 0, 2, 2, 4, 10, 8$

$N_1(32,\cdot) = 4, 2, 6, 4, 4, 2, 2, 4, 6, 6, 4, 8, 2, 2, 8, 0$

$N_1(33,\cdot) = 4, 4, 6, 2, 10, 8, 4, 2, 4, 0, 2, 2, 4, 6, 2, 4$

$N_1(34,\cdot) = 0, 8, 16, 6, 2, 0, 0, 12, 6, 0, 0, 0, 0, 8, 0, 6$

$N_1(35,\cdot) = 2, 2, 4, 0, 8, 0, 0, 0, 14, 4, 6, 8, 0, 2, 14, 0$

$N_1(36,\cdot) = 2, 6, 2, 2, 8, 0, 2, 2, 4, 2, 6, 8, 6, 4, 10, 0$

$N_1(37,\cdot) = 2, 2, 12, 4, 2, 4, 4, 10, 4, 4, 2, 6, 0, 2, 2, 4$

$N_1(38.\cdot) =$ 0, 6, 2, 2, 2, 0, 2, 2, 4, 6, 4, 4, 4, 6,10,10

$N_1(39.\cdot) =$ 6, 2, 2, 4,12, 6, 4, 8, 4, 0, 2, 4, 2, 4, 4, 0

$N_1(3a.\cdot) =$ 6, 4, 6, 4, 6, 8, 0, 6, 2, 2, 6, 2, 2, 6, 4, 0

$N_1(3b.\cdot) =$ 2, 6, 4, 0, 0, 2, 4, 6, 4, 6, 8, 6, 4, 4, 6, 2

$N_1(3c.\cdot) =$ 0,10, 4, 0,12, 0, 4, 2, 6, 0, 4,12, 4, 4, 2, 0

$N_1(3d.\cdot) =$ 0, 8, 6, 2, 2, 6, 0, 8, 4, 4, 0, 4, 0,12, 4, 4

$N_1(3e.\cdot) =$ 4, 8, 2, 2, 2, 4, 4,14, 4, 2, 0, 2, 0, 8, 4, 4

$N_1(3f.\cdot) =$ 4, 8, 4, 2, 4, 0, 2, 4, 4, 2, 4, 8, 8, 6, 2, 2

$N_2(00.\cdot) =$ 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

$N_2(01.\cdot) =$ 0, 0, 0, 4, 0, 2, 6, 4, 0,14, 8, 6, 8, 4, 6, 2

$N_2(02.\cdot) =$ 0, 0, 0, 2, 0, 4, 6, 4, 0, 0, 4, 6,10,10,12, 6

$N_2(03.\cdot) =$ 4, 8, 4, 8, 4, 6, 4, 2, 4, 2, 2, 4, 6, 2, 0, 4

$N_2(04.\cdot) =$ 0, 0, 0, 0, 0, 6, 0,14, 0, 6,10, 4,10, 6, 4, 4

$N_2(05.\cdot) =$ 2, 0, 4, 8, 2, 4, 6, 6, 2, 0, 8, 4, 2, 4,10, 2

$N_2(06.\cdot) =$ 0,12, 6, 4, 6, 4, 6, 2, 2,10, 2, 8, 2, 0, 0, 0

$N_2(07.\cdot) =$ 4, 6, 6, 4, 2, 4, 4, 2, 6, 4, 2, 4, 4, 6, 0, 6

$N_2(08.\cdot) =$ 0, 0, 0, 4, 0, 4, 0, 8, 0,10,16, 6, 6, 0, 6, 4

$N_2(09.\cdot) =$ 14, 2, 4,10, 2, 8, 2, 6, 2, 4, 0, 0, 2, 2, 2, 4

$N_2(0a.\cdot) =$ 0, 6, 6, 2,10, 4,10, 2, 6, 2, 2, 4, 2, 2, 4, 2

$N_2(0b.\cdot) =$ 6, 2, 2, 0, 2, 4, 6, 2,10, 2, 0, 6, 6, 4, 4, 8

$N_2(0c.\cdot) =$ 0, 0, 0, 4, 0,14, 0,10, 0, 6, 2, 4, 4, 8, 6, 6

$N_2(0d.\cdot) =$ 6, 2, 6, 2,10, 2, 0, 4, 0,10, 4, 2, 8, 2, 2, 4

$N_2(0e.\cdot) =$ 0, 6,12, 8, 0, 4, 2, 0, 8, 2, 4, 4, 6, 2, 0, 6

$N_2(0f.\cdot) =$ 0, 8, 2, 0, 6, 6, 8, 2, 4, 4, 4, 6, 8, 0, 4, 2

$N_2(10.\cdot) =$ 0, 0, 0, 8, 0, 4,10, 2, 0, 2, 8,10, 0,10, 6, 4

$N_2(11.\cdot) =$ 6, 6, 4, 6, 4, 0, 6, 4, 8, 2,10, 2, 2, 4, 0, 0

$N_2(12.\cdot) =$ 0, 6, 2, 6, 2, 4,12, 4, 6, 4, 0, 4, 4, 6, 2, 2

$N_2(13.\cdot) =$ 4, 0, 4, 0, 8, 6, 6, 0, 0, 2, 0, 6, 4, 8, 2,14

$N_2(14.\cdot) =$ 0, 6, 6, 4,10, 0, 2,12, 6, 2, 2, 2, 4, 4, 2, 2

$N_2(15.\cdot) =$ 6, 8, 2, 0, 8, 2, 0, 2, 2, 2, 2, 2, 2,14,10, 2

$N_2(16.\cdot) =$ 0, 8, 6, 4, 2, 2, 4, 2, 6, 4, 6, 2, 6, 0, 6, 6

$N_2(17.\cdot) =$ 6, 4, 8, 6, 4, 4, 0, 4, 6, 2, 4, 4, 4, 2, 4, 2

$N_2(18.\cdot) =$ 0, 6, 4, 6,10, 4, 0, 2, 4, 8, 0, 0, 4, 8, 2, 6

$N_2(19.\cdot) =$ 2, 4, 6, 4, 4, 2, 4, 2, 6, 4, 6, 8, 0, 6, 4, 2

$N_2(1a.\cdot) =$ 0, 6, 8, 4, 2, 4, 2, 2, 8, 2, 2, 6, 2, 4, 4, 8

$N_2(1b.\cdot) =$ 0, 6, 4, 4, 0,12, 6, 4, 2, 2, 2, 4, 4, 2,10, 2

$N_2(1c.\cdot) =$ 0, 4, 6, 6,12, 0, 4, 0,10, 2, 6, 2, 0, 0,10, 2

$N_2(1d.\cdot) =$ 0, 6, 2, 2, 6, 0, 4,16, 4, 4, 2, 0, 0, 4, 6, 8

$N_2(1e.\cdot) =$ 0, 4, 8, 2,10, 6, 6, 0, 8, 4, 0, 2, 4, 4, 0, 6

$N_2(1f.\cdot) =$ 4, 2, 6, 6, 2, 2, 2, 4, 8, 6,10, 6, 4, 0, 0, 2

$N_2(20.\cdot) =$ 0, 0, 0, 2, 0,12,10, 4, 0, 0, 0, 2,14, 2, 8,10

$N_2(21.\cdot) =$ 0, 4, 6, 8, 2,10, 4, 2, 2, 6, 4, 2, 6, 2, 0, 6

$N_2(22.\cdot) =$ 4,12, 8, 4, 2, 2, 0, 0, 2, 8, 8, 6, 0, 6, 0, 2

$N_2(23.\cdot) =$ 8, 2, 0, 2, 8, 4, 2, 6, 4, 8, 2, 2, 6, 4, 2, 4

$N_2(24.\cdot) =$ 10, 4, 0, 0, 0, 4, 0, 2, 6, 8, 6,10, 8, 0, 2, 4

$N_2(25.\cdot) =$ 6, 0,12, 2, 8, 6,10, 0, 0, 8, 2, 6, 0, 0, 2, 2

$N_2(26.\cdot) =$ 2, 2, 4, 4, 2, 2,10,14, 2, 0, 4, 2, 2, 4, 6, 4

$N_2(27.\cdot) =$ 6, 0, 0, 2, 6, 4, 2, 4, 4, 4, 8, 4, 8, 0, 6, 6

$N_2(28.\cdot) =$ 8, 0, 8, 2, 4,12, 2, 0, 2, 6, 2, 0, 6, 2, 0,10

$N_2(29.\cdot) =$ 0, 2, 4,10, 2, 8, 6, 4, 0,10, 0, 2,10, 0, 2, 4

$N_2(2a.\cdot) =$ 4, 0, 4, 8, 6, 2, 4, 4, 6, 6, 2, 6, 2, 2, 4, 4

$N_2(2b.\cdot) =$ 2, 2, 6, 4, 0, 2, 2, 6, 2, 8, 8, 4, 4, 4, 8, 2

$N_2(2c.\cdot) =$ 10, 6, 8, 6, 0, 6, 4, 4, 4, 2, 4, 4, 0, 0, 2, 4

$N_2(2d.\cdot) =$ 2, 2, 2, 4, 0, 0, 0, 2, 8, 4, 4, 6,10, 2,14, 4

$N_2(2e.\cdot) =$ 2, 4, 0, 2,10, 4, 2, 0, 2, 2, 6, 2, 8, 8,10, 2

$N_2(2f.\cdot) =$ 12, 4, 6, 8, 2, 6, 2, 8, 0, 4, 0, 2, 0, 8, 2, 0

$N_2(30.\cdot) =$ 0, 4, 0, 2, 4, 4, 8, 6,10, 6, 2,12, 0, 0, 0, 6

$N_2(31.\cdot) =$ 0,10, 2, 0, 6, 2,10, 2, 6, 0, 2, 0, 6, 6, 4, 8

$N_2(32.\cdot) =$ 8, 4, 6, 0, 6, 4, 4, 8, 4, 6, 8, 0, 2, 2, 2, 0

$N_2(33.\cdot) =$ 2, 2, 6,10, 2, 0, 0, 6, 4, 4,12, 8, 4, 2, 2, 0

$N_2(34.\cdot) =$ 0,12, 6, 4, 6, 0, 4, 4, 4, 0, 4, 6, 4, 2, 4, 4

$N_2(35.\cdot) =$ 0,12, 4, 6, 2, 4, 4, 0,10, 0, 0, 8, 0, 8, 0, 6

$N_2(36.\cdot) =$ 8, 2, 4, 0, 4, 0, 4, 2, 0, 8, 4, 2, 6,16, 2, 2

$N_2(37.\cdot) =$ 6, 2, 2, 2, 6, 6, 4, 8, 2, 2, 6, 2, 2, 2, 4, 8

$N_2(38.\cdot) =$ 0, 8, 8,10, 6, 2, 2, 0, 4, 0, 4, 2, 4, 0, 4,10

$N_2(39.\cdot) =$ 0, 2, 0, 0, 8, 0,10, 4,10, 0, 8, 4, 4, 4, 4, 6

$N_2(3a.\cdot) =$ 4, 0, 2, 8, 4, 2, 2, 2, 4, 8, 2, 0, 4,10,10, 2

$N_2(3b.\cdot) =$ 16, 4, 4, 2, 8, 2, 2, 6, 4, 4, 4, 2, 0, 2, 2, 2

$N_2(3c.\cdot) =$ 0, 2, 6, 2, 8, 4, 6, 0,10, 2, 2, 4, 4,10, 4, 0

$N_2(3d.\cdot) =$ 0,16,10, 2, 4, 2, 4, 2, 8, 0, 0, 8, 0, 6, 2, 0

$N_2(3e.\cdot) =$ 4, 4, 0,10, 2, 4, 2,14, 4, 2, 6, 6, 0, 0, 6, 0

$N_2(3f.\cdot) =$ 4, 0, 0, 2, 0, 8, 2, 4, 0, 2, 4, 4, 4,14,10, 6

$N_3(00.\cdot) =$ 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

$N_3(01.\cdot) =$ 0, 0, 0, 2, 0, 4, 2,12, 0,14, 0, 4, 8, 2, 6,10

$N_3(02.\cdot) =$ 0, 0, 0, 2, 0, 2, 0, 8, 0, 4,12,10, 4, 6, 8, 8

$N_3(03.\cdot) =$ 8, 6,10, 4, 8, 6, 0, 6, 4, 4, 0, 0, 0, 4, 2, 2

$N_3(04,\cdot) =$ 0, 0, 0, 4, 0, 2, 4, 2, 0,12, 8, 4, 6, 8,10, 4

$N_3(05,\cdot) =$ 6, 2, 4, 8, 6,10, 6, 2, 2, 8, 2, 0, 2, 0, 4, 2

$N_3(06,\cdot) =$ 0,10, 6, 6,10, 0, 4,12, 2, 4, 0, 0, 6, 4, 0, 0

$N_3(07,\cdot) =$ 2, 0, 0, 4, 4, 4, 4, 2,10, 4, 4, 8, 4, 4, 4, 6

$N_3(08,\cdot) =$ 0, 0, 0,10, 0, 4, 4, 6, 0, 6, 6, 6, 6, 0, 8, 8

$N_3(09,\cdot) =$ 10, 2, 0, 2,10, 4, 6, 2, 0, 6, 0, 4, 6, 2, 4, 6

$N_3(0a,\cdot) =$ 0,10, 6, 0,14, 6, 4, 0, 4, 6, 6, 0, 4, 0, 2, 2

$N_3(0b,\cdot) =$ 2, 6, 2,10, 2, 2, 4, 0, 4, 2, 6, 0, 2, 8,14, 0

$N_3(0c,\cdot) =$ 0, 0, 0, 8, 0,12,12, 4, 0, 8, 0, 4, 2,10, 2, 2

$N_3(0d,\cdot) =$ 8, 2, 8, 0, 0, 4, 2, 0, 2, 8,14, 2, 6, 2, 4, 2

$N_3(0e,\cdot) =$ 0, 4, 4, 2, 4, 2, 4, 4,10, 4, 4, 4, 4, 4, 2, 8

$N_3(0f,\cdot) =$ 4, 6, 4, 6, 2, 2, 4, 8, 6, 2, 6, 2, 0, 6, 2, 4

$N_3(10,\cdot) =$ 0, 0, 0, 4, 0,12, 4, 8, 0, 4, 2, 6, 2,14, 0, 8

$N_3(11,\cdot) =$ 8, 2, 2, 6, 4, 0, 2, 0, 8, 4,12, 2,10, 0, 2, 2

$N_3(12,\cdot) =$ 0, 2, 8, 2, 4, 8, 0, 8, 8, 0, 2, 2, 4, 2,14, 0

$N_3(13,\cdot) =$ 4, 4,12, 0, 2, 2, 2,10, 2, 2, 2, 2, 4, 4, 4, 8

$N_3(14,\cdot) =$ 0, 6, 4, 4, 6, 4, 6, 2, 8, 6, 6, 2, 2, 0, 0, 8

$N_3(15,\cdot) =$ 4, 8, 2, 8, 2, 4, 8, 0, 4, 2, 2, 2, 2, 6, 8, 2

$N_3(16,\cdot) =$ 0, 6,10, 2, 8, 4, 2, 0, 2, 2, 2, 8, 4, 6, 4, 4

$N_3(17,\cdot) =$ 0, 6, 6, 0, 6, 2, 4, 4, 6, 2, 2,10, 6, 8, 2, 0

$N_3(18,\cdot) =$ 0, 8, 4, 6, 6, 0, 6, 2, 4, 0, 4, 2,10, 0, 6, 6

$N_3(19,\cdot) =$ 4, 2, 4, 8, 4, 2,10, 2, 2, 2, 6, 8, 2, 6, 0, 2

$N_3(1a,\cdot) =$ 0, 8, 6, 4, 4, 0, 6, 4, 4, 8, 0,10, 2, 2, 2, 4

$N_3(1b,\cdot) =$ 4,10, 2, 0, 2, 4, 2, 4, 8, 2, 2, 8, 4, 2, 8, 2

$N_3(1c,\cdot) =$ 0, 6, 8, 8, 4, 2, 8, 0,12, 0,10, 0, 4, 0, 2, 0

$N_3(1d,\cdot) =$ 0, 2, 0, 6, 2, 8, 4, 6, 2, 0, 4, 2, 4,10, 0,14

$N_3(1e,\cdot) =$ 0, 4, 8, 2, 4, 6, 0, 4,10, 0, 2, 6, 4, 8, 4, 2

$N_3(1f,\cdot) =$ 0, 6, 8, 0,10, 6, 4, 6, 4, 2, 2,10, 4, 0, 0, 2

$N_3(20,\cdot) =$ 0, 0, 0, 0, 0, 4, 4, 8, 0, 2, 2, 4,10,16,12, 2

$N_3(21,\cdot) =$ 10, 8, 8, 0, 8, 4, 2, 4, 0, 6, 6, 6, 0, 0, 2, 0

$N_3(22,\cdot) =$ 12, 6, 4, 4, 2, 4,10, 2, 0, 4, 4, 2, 4, 4, 0, 2

$N_3(23,\cdot) =$ 2, 2, 0, 6, 0, 2, 4, 0, 4,12, 4, 2, 6, 4, 8, 8

$N_3(24,\cdot) =$ 4, 8, 2,12, 6, 4, 2,10, 2, 2, 2, 4, 2, 0, 4, 0

$N_3(25,\cdot) =$ 6, 0, 2, 0, 8, 2, 0, 2, 8, 8, 2, 2, 4, 4,10, 6

$N_3(26,\cdot) =$ 6, 2, 0, 4, 4, 0, 4, 0, 4, 2,14, 0, 8,10, 0, 6

$N_3(27,\cdot) =$ 0, 2, 4,16, 8, 6, 6, 6, 0, 2, 4, 4, 0, 2, 2, 2

$N_3(28,\cdot) =$ 6, 2,10, 0, 6, 4, 0, 4, 4, 2, 4, 8, 2, 2, 8, 2

$N_3(29,\cdot) =$ 0, 2, 8, 4, 0, 4, 0, 6, 4,10, 4, 8, 4, 4, 4, 2

$N_3(2a,\cdot) =$ 2, 6, 0, 4, 2, 4, 4, 6, 4, 8, 4, 4, 4, 2, 4, 6

$N_3(2b,\cdot) =$ 10, 2, 6, 6, 4, 4, 8, 0, 4, 2, 2, 0, 2, 4, 4, 6

$N_3(2c,\cdot) =$ 10, 4, 6, 2, 4, 2, 2, 2, 4,10, 4, 4, 0, 2, 6, 2

$N_3(2d,\cdot) =$ 4, 2, 4, 4, 4, 2, 4,16, 2, 0, 0, 4, 4, 2, 6, 6

$N_3(2e,\cdot) =$ 4, 0, 2,10, 0, 6,10, 4, 2, 6, 6, 2, 2, 0, 2, 8

$N_3(2f,\cdot) =$ 8, 2, 0, 0, 4, 4, 4, 2, 6, 4, 6, 2, 4, 8, 4, 6

$N_3(30,\cdot) =$ 0,10, 8, 6, 2, 0, 4, 2,10, 4, 4, 6, 2, 0, 6, 0

$N_3(31,\cdot) =$ 2, 6, 2, 0, 4, 2, 8, 8, 2, 2, 2, 0, 2,12, 6, 6

$N_3(32,\cdot) =$ 2, 0, 4, 8, 2, 8, 4, 4, 8, 4, 2, 8, 6, 2, 0, 2

$N_3(33,\cdot) =$ 4, 4, 6, 8, 6, 6, 0, 2, 2, 2, 6, 4,12, 0, 0, 2

$N_3(34,\cdot) =$ 0, 6, 2, 2,16, 2, 2, 2,12, 2, 4, 0, 4, 2, 0, 8

$N_3(35,\cdot) =$ 4, 6, 0,10, 8, 0, 2, 2, 6, 0, 0, 6, 2,10, 2, 6

$N_3(36,\cdot) =$ 4, 4, 4, 4, 0, 6, 6, 4, 4, 4, 4, 4, 0, 6, 2, 8

$N_3(37,\cdot) =$ 4, 8, 2, 4, 2, 2, 6, 0, 2, 4, 8, 4,10, 0, 6, 2

$N_3(38,\cdot) =$ 0, 8,12, 0, 2, 2, 6, 6, 2,10, 2, 2, 0, 8, 0, 4

$N_3(39,\cdot) =$ 2, 6, 4, 0, 6, 4, 6, 4, 8, 0, 4, 4, 2, 4, 8, 2

$N_3(3a,\cdot) =$ 6, 0, 2, 2, 4, 6, 4, 4, 4, 2, 2, 6,12, 2, 6, 2

$N_3(3b,\cdot) =$ 2, 2, 6, 0, 0,10, 4, 8, 4, 2, 4, 8, 4, 4, 0, 6

$N_3(3c,\cdot) =$ 0, 2, 4, 2,12, 2, 0, 6, 2, 0, 2, 8, 4, 6, 4,10

$N_3(3d,\cdot) =$ 4, 6, 8, 6, 2, 2, 2, 2,10, 2, 6, 6, 2, 4, 2, 0

$N_3(3e,\cdot) =$ 8, 6, 4, 4, 2,10, 2, 0, 2, 2, 4, 2, 4, 2,10, 2

$N_3(3f,\cdot) =$ 2, 6, 4, 0, 0,10, 8, 2, 2, 8, 6, 4, 6, 2, 0, 4

$N_4(00,\cdot) =$ 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

$N_4(01,\cdot) =$ 0, 0, 0, 0, 0,16,16, 0, 0,16,16, 0, 0, 0, 0, 0

$N_4(02,\cdot) =$ 0, 0, 0, 8, 0, 4, 4, 8, 0, 4, 4, 8, 8, 8, 8, 0

$N_4(03,\cdot) =$ 8, 6, 2, 0, 2, 4, 8, 2, 6, 0, 4, 6, 0, 6, 2, 8

$N_4(04,\cdot) =$ 0, 0, 0, 8, 0, 0,12, 4, 0,12, 0, 4, 8, 4, 4, 8

$N_4(05,\cdot) =$ 4, 2, 2, 8, 2,12, 0, 2, 2, 0,12, 2, 8, 2, 2, 4

$N_4(06,\cdot) =$ 0, 8, 8, 4, 8, 8, 0, 0, 8, 0, 8, 0, 4, 0, 0, 8

$N_4(07,\cdot) =$ 4, 2, 6, 4, 6, 0,16, 6, 2, 0, 0, 2, 4, 2, 6, 4

$N_4(08,\cdot) =$ 0, 0, 0, 4, 0, 8, 4, 8, 0, 4, 8, 8, 4, 8, 8, 0

$N_4(09,\cdot) =$ 8, 4, 4, 4, 4, 0, 8, 4, 4, 0, 0, 4, 4, 4, 4, 8

$N_4(0a,\cdot) =$ 0, 6, 6, 0, 6, 4, 4, 6, 6, 4, 4, 6, 0, 6, 6, 0

$N_4(0b,\cdot) =$ 0,12, 0, 8, 0, 0, 0, 0,12, 0, 0,12, 8,12, 0, 0

$N_4(0c,\cdot) =$ 0, 0, 0, 4, 0, 8, 4, 8, 0, 4, 8, 8, 4, 8, 8, 0

$N_4(0d,\cdot) =$ 8, 4, 4, 4, 4, 0, 0, 4, 4, 8, 0, 4, 4, 4, 4, 8

$N_4(0e,\cdot) =$ 0, 6, 6, 4, 6, 0, 4, 6, 6, 4, 0, 6, 4, 6, 6, 0

$N_4(0f,\cdot) =$ 0, 6, 6, 4, 6, 4, 0, 6, 6, 0, 4, 6, 4, 6, 6, 0

$N_4(10.\cdot) =$ 0, 0, 0, 0, 0, 8,12, 4, 0,12, 8, 4, 0, 4, 4, 8
$N_4(11.\cdot) =$ 4, 2, 2,16, 2, 4, 0, 2, 2, 0, 4, 2,16, 2, 2, 4
$N_4(12.\cdot) =$ 0, 0, 0, 8, 0, 4, 4, 8, 0, 4, 4, 8, 8, 8, 8, 0
$N_4(13.\cdot) =$ 8, 2, 6, 0, 6, 4, 0, 6, 2, 8, 4, 2, 0, 2, 6, 8
$N_4(14.\cdot) =$ 0, 8, 8, 0, 8, 0, 8, 0, 8, 8, 0, 0, 0, 0, 0,16
$N_4(15.\cdot) =$ 8, 4, 4, 0, 4, 8, 0, 4, 4, 0, 8, 4, 0, 4, 4, 8
$N_4(16.\cdot) =$ 0, 8, 8, 4, 8, 8, 0, 0, 8, 0, 8, 0, 4, 0, 0, 8
$N_4(17.\cdot) =$ 4, 6, 2, 4, 2, 0, 0, 2, 6,16, 0, 6, 4, 6, 2, 4
$N_4(18.\cdot) =$ 0, 8, 8, 8, 8, 4, 0, 0, 8, 0, 4, 0, 8, 0, 0, 8
$N_4(19.\cdot) =$ 4, 4, 4, 0, 4, 4,16, 4, 4, 0, 4, 4, 0, 4, 4, 4
$N_4(1a.\cdot) =$ 0, 6, 6, 4, 6, 0, 4, 6, 6, 4, 0, 6, 4, 6, 6, 0
$N_4(1b.\cdot) =$ 0, 6, 6, 4, 6, 4, 0, 6, 6, 0, 4, 6, 4, 6, 6, 0
$N_4(1c.\cdot) =$ 0, 8, 8, 8, 8, 4, 0, 0, 8, 0, 4, 0, 8, 0, 0, 8
$N_4(1d.\cdot) =$ 4, 4, 4, 0, 4, 4, 0, 4, 4,16, 4, 4, 0, 4, 4, 4
$N_4(1e.\cdot) =$ 0, 6, 6, 0, 6, 4, 4, 6, 6, 4, 4, 6, 0, 6, 6, 0
$N_4(1f.\cdot) =$ 0, 0,12, 8,12, 0, 0,12, 0, 0, 0, 0, 8, 0,12, 0
$N_4(20.\cdot) =$ 0, 0, 0, 8, 0, 0, 0,12, 0, 0, 0,12, 8,12,12, 0
$N_4(21.\cdot) =$ 0, 4, 8, 0, 8, 4, 8, 8, 4, 0, 4, 4, 0, 4, 8, 0
$N_4(22.\cdot) =$ 8, 2, 2, 0, 2, 4, 8, 6, 2, 8, 4, 6, 0, 6, 6, 0
$N_4(23.\cdot) =$ 4, 6, 2, 8, 2, 4, 0, 2, 6, 0, 4, 6, 8, 6, 2, 4
$N_4(24.\cdot) =$ 0, 6, 6, 4, 6, 4, 0, 6, 6, 0, 4, 6, 4, 6, 6, 0
$N_4(25.\cdot) =$ 0, 8, 4, 4, 4, 0, 0, 4, 8, 8, 0, 8, 4, 8, 4, 0
$N_4(26.\cdot) =$ 0, 6, 6, 0, 6, 4, 8, 2, 6, 8, 4, 2, 0, 2, 2, 8
$N_4(27.\cdot) =$ 4, 6, 2, 8, 2, 4, 0, 2, 6, 0, 4, 6, 8, 6, 2, 4
$N_4(28.\cdot) =$ 16, 4, 4, 0, 4, 4, 4, 4, 4, 4, 4, 4, 0, 4, 4, 0
$N_4(29.\cdot) =$ 0, 6, 2, 8, 2, 4, 0, 2, 6, 8, 4, 6, 8, 6, 2, 0
$N_4(2a.\cdot) =$ 0, 2, 2,16, 2, 4, 4, 2, 2, 4, 4, 2,16, 2, 2, 0
$N_4(2b.\cdot) =$ 8, 0, 4, 0, 4, 8,16, 4, 0, 0, 8, 0, 0, 0, 4, 8
$N_4(2c.\cdot) =$ 8, 4, 4, 4, 4, 0, 8, 4, 4, 8, 0, 4, 4, 4, 4, 0
$N_4(2d.\cdot) =$ 4, 2, 6, 4, 6, 8, 0, 6, 2, 0, 8, 2, 4, 2, 6, 4
$N_4(2e.\cdot) =$ 16, 0, 0, 0, 0,16, 0, 0, 0, 0,16, 0, 0, 0, 0,16
$N_4(2f.\cdot) =$ 16, 0, 0, 0, 0, 0,16, 0, 0,16, 0, 0, 0, 0, 0,16
$N_4(30.\cdot) =$ 0, 6, 6, 4, 6, 4, 0, 6, 6, 0, 4, 6, 4, 6, 6, 0
$N_4(31.\cdot) =$ 0, 8, 4, 4, 4, 0, 0, 4, 8, 8, 0, 8, 4, 8, 4, 0
$N_4(32.\cdot) =$ 16, 6, 6, 4, 6, 0, 4, 2, 6, 4, 0, 2, 4, 2, 2, 0
$N_4(33.\cdot) =$ 0, 2, 6, 4, 6, 8, 8, 6, 2, 0, 8, 2, 4, 2, 6, 0
$N_4(34.\cdot) =$ 0,12,12, 8,12, 0, 0, 0,12, 0, 0, 0, 8, 0, 0, 0
$N_4(35.\cdot) =$ 0, 4, 8, 0, 8, 4, 8, 8, 4, 0, 4, 4, 0, 4, 8, 0
$N_4(36.\cdot) =$ 0, 2, 2, 4, 2, 0, 4, 6, 2, 4, 0, 6, 4, 6, 6,16
$N_4(37.\cdot) =$ 0, 2, 6, 4, 6, 8, 8, 6, 2, 0, 8, 2, 4, 2, 6, 0
$N_4(38.\cdot) =$ 0, 4, 4, 0, 4, 4, 4, 4, 4, 4, 4, 4, 0, 4, 4,16
$N_4(39.\cdot) =$ 0, 6, 2, 8, 2, 4, 0, 2, 6, 8, 4, 6, 8, 6, 2, 0
$N_4(3a.\cdot) =$ 0, 4, 4, 0, 4, 8, 8, 4, 4, 8, 8, 4, 0, 4, 4, 0
$N_4(3b.\cdot) =$ 16, 4, 4, 0, 4, 0, 0, 4, 4, 0, 0, 4, 0, 4, 4,16
$N_4(3c.\cdot) =$ 0, 4, 4, 4, 4, 0, 8, 4, 4, 8, 0, 4, 4, 4, 4, 8
$N_4(3d.\cdot) =$ 4, 2, 6, 4, 6, 8, 0, 6, 2, 0, 8, 2, 4, 2, 6, 4
$N_4(3e.\cdot) =$ 0, 2, 2, 8, 2,12, 4, 2, 2, 4,12, 2, 8, 2, 2, 0
$N_4(3f.\cdot) =$ 8, 4, 0, 8, 0, 0, 0, 0, 4,16, 0, 4, 8, 4, 0, 8
$N_5(00.\cdot) =$ 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
$N_5(01.\cdot) =$ 0, 0, 0, 4, 0,10, 8, 6, 0, 4, 2, 2,12,10, 2, 4
$N_5(02.\cdot) =$ 0, 0, 0, 4, 0,10, 6, 4, 0, 6, 4, 2, 4, 8,10, 6
$N_5(03.\cdot) =$ 8, 2, 4, 6, 4, 4, 2, 2, 6, 8, 6, 4, 4, 0, 2, 2
$N_5(04.\cdot) =$ 0, 0, 0, 8, 0, 4,10, 6, 0, 6, 6, 4, 8, 6, 0, 6
$N_5(05.\cdot) =$ 12, 2, 0, 4, 0, 4, 8, 2, 4, 0,16, 2, 0, 2, 0, 8
$N_5(06.\cdot) =$ 0, 8, 4, 6, 4, 6, 2, 2, 4, 4, 6, 0, 6, 0, 2,10
$N_5(07.\cdot) =$ 2, 0, 4, 8, 4, 2, 6, 6, 2, 8, 6, 2, 2, 0, 6, 6
$N_5(08.\cdot) =$ 0, 0, 0, 2, 0, 8,10, 4, 0, 4,10, 4, 8, 4, 4, 6
$N_5(09.\cdot) =$ 8, 6, 0, 4, 0, 6, 6, 2, 2,10, 2, 8, 6, 2, 0, 2
$N_5(0a.\cdot) =$ 0, 6, 8, 6, 0, 8, 0, 0, 8,10, 4, 2, 8, 0, 0, 4
$N_5(0b.\cdot) =$ 4, 2, 2, 4, 8,10, 6, 4, 2, 6, 2, 2, 6, 2, 2, 2
$N_5(0c.\cdot) =$ 0, 0, 0,10, 0, 2,10, 2, 0, 6,10, 6, 6, 6, 2, 4
$N_5(0d.\cdot) =$ 10, 4, 2, 2, 0, 6,16, 0, 0, 2,10, 2, 2, 4, 0, 4
$N_5(0e.\cdot) =$ 0, 6, 4, 8, 4, 6,10, 2, 4, 4, 4, 2, 4, 0, 2, 4
$N_5(0f.\cdot) =$ 4, 4, 0, 8, 0, 2, 0, 2, 8, 2, 4, 2, 8, 4, 4,12
$N_5(10.\cdot) =$ 0, 0, 0, 0, 0, 4, 4,12, 0, 2, 8,10, 4, 6,12, 2
$N_5(11.\cdot) =$ 6, 6,10,10, 4, 0, 2, 6, 2, 4, 0, 6, 2, 4, 2, 0
$N_5(12.\cdot) =$ 0, 2, 4, 2,10, 4, 0,10, 8, 6, 0, 6, 0, 6, 6, 0
$N_5(13.\cdot) =$ 0, 0, 6, 2, 8, 0, 0, 4, 4, 6, 2, 8, 2, 8,10, 4
$N_5(14.\cdot) =$ 0,12, 2, 6, 4, 0, 4, 4, 8, 4, 4, 4, 6, 2, 4, 0
$N_5(15.\cdot) =$ 4, 8, 0, 2, 8, 0, 2, 4, 2, 2, 4, 2, 4, 8, 8, 6
$N_5(16.\cdot) =$ 0, 6,10, 2,14, 0, 2, 2, 4, 4, 0, 6, 0, 4, 6, 4
$N_5(17.\cdot) =$ 0, 6, 8, 4, 8, 4, 0, 2, 8, 4, 0, 2, 2, 8, 6, 2
$N_5(18.\cdot) =$ 0,10, 8, 0, 6, 4, 0, 4, 4, 4, 6, 4, 4, 4, 0, 6
$N_5(19.\cdot) =$ 0, 4, 6, 2, 4, 4, 2, 6, 4, 2, 2, 4,12, 2,10, 0
$N_5(1a.\cdot) =$ 0, 2,16, 2,12, 2, 0, 6, 4, 0, 0, 4, 0, 4, 4, 8
$N_5(1b.\cdot) =$ 2, 8,12, 0, 0, 2, 2, 6, 8, 4, 0, 6, 0, 0, 8, 6

$N_5(\mathtt{1c},\cdot) =$ 0,10, 2, 6, 6, 6, 6, 4, 8, 2, 0, 4, 4, 4, 2, 0

$N_5(\mathtt{1d},\cdot) =$ 4, 6, 2, 0, 8, 2, 4, 6, 6, 0, 8, 6, 2, 4, 2, 4

$N_5(\mathtt{1e},\cdot) =$ 0, 2, 6, 2, 4, 0, 0, 2,12, 2, 2, 6, 2,10,10, 4

$N_5(\mathtt{1f},\cdot) =$ 0, 6, 8, 4, 8, 8, 0, 6, 6, 2, 0, 6, 0, 6, 2, 2

$N_5(\mathtt{20},\cdot) =$ 0, 0, 0, 8, 0, 8, 2, 6, 0, 4, 4, 4, 6, 6, 8, 8

$N_5(\mathtt{21},\cdot) =$ 0, 0, 0, 6, 6, 2, 6, 4, 6,10,14, 4, 0, 0, 4, 2

$N_5(\mathtt{22},\cdot) =$ 14, 4, 0,10, 0, 2,12, 2, 2, 2,10, 2, 0, 0, 2, 2

$N_5(\mathtt{23},\cdot) =$ 2, 0, 0, 4, 2, 2,10, 4, 0, 8, 8, 2, 6, 8, 0, 8

$N_5(\mathtt{24},\cdot) =$ 6, 2, 8, 4, 4, 4, 6, 2, 2, 6, 6, 2, 6, 2, 2, 2

$N_5(\mathtt{25},\cdot) =$ 6, 0, 0, 8, 2, 8, 2, 6, 6, 4, 2, 2, 4, 2, 6, 6

$N_5(\mathtt{26},\cdot) =$ 12, 0, 0, 4, 0, 4, 4, 4, 0, 8, 4, 0,12, 8, 0, 4

$N_5(\mathtt{27},\cdot) =$ 12, 2, 0, 2, 0,12, 2, 2, 4, 4, 8, 4, 8, 2, 2, 0

$N_5(\mathtt{28},\cdot) =$ 2, 8, 4, 6, 2, 4, 6, 0, 6, 6, 4, 0, 2, 2, 2,10

$N_5(\mathtt{29},\cdot) =$ 6, 4, 6, 8, 8, 4, 6, 2, 0, 0, 2, 2,10, 0, 2, 4

$N_5(\mathtt{2a},\cdot) =$ 4, 4, 0, 2, 2, 4, 6, 2, 0, 0, 6, 4,10, 4, 4,12

$N_5(\mathtt{2b},\cdot) =$ 4, 6, 2, 6, 0, 0,12, 2, 0, 4,12, 2, 6, 4, 0, 4

$N_5(\mathtt{2c},\cdot) =$ 8, 6, 2, 6, 4, 8, 6, 0, 4, 4, 0, 2, 6, 0, 6, 2

$N_5(\mathtt{2d},\cdot) =$ 4, 4, 0, 4, 0, 6, 4, 2, 4,12, 0, 4, 4, 6, 4, 6

$N_5(\mathtt{2e},\cdot) =$ 6, 0, 2, 4, 0, 6, 6, 4, 2,10, 6,10, 6, 2, 0, 0

$N_5(\mathtt{2f},\cdot) =$ 10, 4, 0, 2, 2, 6,10, 2, 0, 2, 2, 4, 6, 2, 2,10

$N_5(\mathtt{30},\cdot) =$ 0, 4, 8, 4, 6, 4, 0, 6,10, 4, 2, 4, 2, 6, 4, 0

$N_5(\mathtt{31},\cdot) =$ 0, 6, 6, 4,10, 2, 0, 0, 4, 4, 0, 0, 4, 6,12, 6

$N_5(\mathtt{32},\cdot) =$ 4, 6, 0, 2, 6, 4, 6, 0, 6, 0, 4, 6, 4,10, 6, 0

$N_5(\mathtt{33},\cdot) =$ 8,10, 0,14, 8, 0, 0, 8, 2, 0, 2, 4, 0, 4, 4, 0

$N_5(\mathtt{34},\cdot) =$ 0, 4, 4, 2,14, 4, 0, 8, 6, 8, 2, 2, 0, 4, 6, 0

$N_5(\mathtt{35},\cdot) =$ 0, 4,16, 0, 8, 4, 0, 4, 4, 4, 0, 8, 0, 4, 4, 4

$N_5(\mathtt{36},\cdot) =$ 4, 4, 4, 6, 2, 2, 2,12, 2, 4, 4, 8, 2, 4, 4, 0

$N_5(\mathtt{37},\cdot) =$ 4, 2, 2, 2, 4, 2, 0, 8, 2, 2, 2,12, 6, 2, 8, 6

$N_5(\mathtt{38},\cdot) =$ 0, 4, 8, 4,12, 0, 0, 8,10, 2, 0, 0, 0, 4, 2,10

$N_5(\mathtt{39},\cdot) =$ 0, 8,12, 0, 2, 2, 2, 2,12, 4, 0, 8, 0, 4, 4, 4

$N_5(\mathtt{3a},\cdot) =$ 0,14, 4, 0, 4, 6, 0, 0, 6, 2,10, 8, 0, 0, 4, 6

$N_5(\mathtt{3b},\cdot) =$ 0, 2, 2, 2, 4, 4, 8, 6, 8, 2, 2, 2, 6,14, 2, 0

$N_5(\mathtt{3c},\cdot) =$ 0, 0,10, 2, 6, 0, 0, 2, 6, 2, 2,10, 2, 4,10, 8

$N_5(\mathtt{3d},\cdot) =$ 0, 6,12, 2, 4, 8, 0, 8, 8, 2, 2, 0, 2, 2, 4, 4

$N_5(\mathtt{3e},\cdot) =$ 4, 4,10, 0, 2, 4, 8, 8, 2, 2, 0, 2, 6, 8, 4, 0

$N_5(\mathtt{3f},\cdot) =$ 8, 6, 6, 0, 4, 2, 2, 4, 4, 2, 8, 6, 2, 4, 6, 0

$N_6(\mathtt{00},\cdot) =$ 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

$N_6(\mathtt{01},\cdot) =$ 0, 0, 0, 6, 0, 2, 6, 2, 0, 4, 2, 4, 6,16,14, 2

$N_6(\mathtt{02},\cdot) =$ 0, 0, 0, 2, 0,10, 6,10, 0, 2, 4, 8, 6, 6, 8, 2

$N_6(\mathtt{03},\cdot) =$ 0, 8, 0, 8, 0, 6, 4, 6, 4, 4, 4,12, 2, 4, 2, 0

$N_6(\mathtt{04},\cdot) =$ 0, 0, 0, 8, 0, 0, 8, 0, 0, 6, 8,10, 2, 4,10, 8

$N_6(\mathtt{05},\cdot) =$ 10, 2, 4, 4, 4, 8, 8, 4, 2, 2, 0, 4, 0, 8, 0, 4

$N_6(\mathtt{06},\cdot) =$ 0, 8, 4, 4, 8, 4, 2, 2,12, 0, 2, 6, 6, 2, 2, 2

$N_6(\mathtt{07},\cdot) =$ 6, 6, 4, 0, 2,10, 2, 2, 2, 2, 6, 6, 8, 0, 6, 2

$N_6(\mathtt{08},\cdot) =$ 0, 0, 0, 6, 0, 2,16, 4, 0, 2, 6, 2, 4,12, 6, 4

$N_6(\mathtt{09},\cdot) =$ 10, 4, 2, 6, 0, 2, 6, 2, 4, 0, 8, 6, 4, 4, 2, 4

$N_6(\mathtt{0a},\cdot) =$ 0,14, 4, 4, 0, 2, 2, 2,10, 4, 4, 4, 6, 4, 2, 2

$N_6(\mathtt{0b},\cdot) =$ 4, 6, 2, 0, 2, 2,12, 8, 2, 2, 2, 6, 8, 2, 0, 6

$N_6(\mathtt{0c},\cdot) =$ 0, 0, 0,12, 0,10, 4, 6, 0, 8, 4, 4, 2,12, 2, 0

$N_6(\mathtt{0d},\cdot) =$ 12, 0, 2,10, 6, 4, 4, 2, 4, 2, 6, 0, 2, 6, 0, 4

$N_6(\mathtt{0e},\cdot) =$ 0, 6, 4, 0, 4, 4,10, 8, 6, 2, 4, 6, 2, 0, 6, 2

$N_6(\mathtt{0f},\cdot) =$ 2, 2, 2, 2, 6, 2, 6, 2,10, 4, 8, 2, 6, 4, 4, 2

$N_6(\mathtt{10},\cdot) =$ 0, 0, 0, 8, 0, 8, 0,12, 0, 4, 2, 6, 8, 4, 6, 6

$N_6(\mathtt{11},\cdot) =$ 6, 2, 6, 4, 6, 2, 6, 4, 6, 6, 4, 2, 4, 0, 6, 0

$N_6(\mathtt{12},\cdot) =$ 0, 8, 4, 2, 0, 4, 2, 0, 4,10, 6, 2, 8, 6, 4, 4

$N_6(\mathtt{13},\cdot) =$ 6, 6,12, 0,12, 2, 0, 6, 6, 2, 0, 4, 0, 2, 4, 2

$N_6(\mathtt{14},\cdot) =$ 0, 4, 6, 2, 8, 6, 0, 2, 6,10, 4, 0, 2, 4, 6, 4

$N_6(\mathtt{15},\cdot) =$ 2, 2, 6, 6, 4, 4, 2, 6, 2, 6, 8, 4, 4, 0, 4, 4

$N_6(\mathtt{16},\cdot) =$ 0, 4,14, 6, 8, 4, 2, 6, 2, 0, 2, 0, 4, 2, 0,10

$N_6(\mathtt{17},\cdot) =$ 2, 6, 8, 0, 0, 2, 0, 2, 2, 6, 0, 8, 8, 2,12, 6

$N_6(\mathtt{18},\cdot) =$ 0, 4, 6, 6, 8, 4, 2, 2, 6, 4, 6, 4, 2, 4, 2, 4

$N_6(\mathtt{19},\cdot) =$ 2, 6, 0, 2, 4, 4, 4, 6, 4, 8, 6, 4, 2, 2, 6, 4

$N_6(\mathtt{1a},\cdot) =$ 0, 6, 6, 0, 8, 2, 4, 6, 4, 2, 4, 6, 2, 0, 4,10

$N_6(\mathtt{1b},\cdot) =$ 0, 4,10, 2, 4, 4, 2, 6, 6, 6, 2, 2, 6, 6, 2, 2

$N_6(\mathtt{1c},\cdot) =$ 0, 0, 8, 2,12, 2, 6, 2, 8, 6, 6, 2, 4, 0, 4, 2

$N_6(\mathtt{1d},\cdot) =$ 2, 4, 0, 6, 8, 6, 0, 2, 6, 8, 6, 0, 2, 4, 0,10

$N_6(\mathtt{1e},\cdot) =$ 0,10, 8, 2, 8, 2, 0, 2, 6, 4, 2, 4, 6, 4, 2, 4

$N_6(\mathtt{1f},\cdot) =$ 0, 6, 6, 8, 6, 4, 2, 4, 4, 2, 2, 0, 2, 4, 2,12

$N_6(\mathtt{20},\cdot) =$ 0, 0, 0, 0, 0, 6, 6, 4, 0, 4, 8, 8, 4, 6,10, 8

$N_6(\mathtt{21},\cdot) =$ 2, 8, 6, 8, 4, 4, 6, 6, 8, 4, 0, 4, 0, 2, 2, 0

$N_6(\mathtt{22},\cdot) =$ 16, 2, 4, 6, 2, 4, 2, 0, 6, 4, 8, 2, 0, 2, 2, 4

$N_6(\mathtt{23},\cdot) =$ 0, 4, 0, 4, 4, 6,10, 4, 2, 2, 6, 2, 4, 6, 6, 4

$N_6(\mathtt{24},\cdot) =$ 10, 8, 0, 6,12, 6,10, 4, 8, 0, 0, 0, 0, 0, 0, 0

$N_6(\mathtt{25},\cdot) =$ 0, 2, 4, 2, 0, 4, 4, 0, 4, 0,10,10, 4,10, 6, 4

$N_6(\mathtt{26},\cdot) =$ 2, 2, 0,12, 2, 2, 6, 2, 4, 4, 8, 0, 6, 6, 8, 0

$N_6(\mathtt{27},\cdot) =$ 8, 4, 0, 8, 2, 4, 2, 4, 0, 6, 2, 4, 4, 8, 2, 6

$N_6(28,\cdot) =$ 6, 8, 4, 6, 0, 4, 2, 2, 4, 8, 2, 6, 4, 2, 2, 4

$N_6(29,\cdot) =$ 2, 4, 4, 0, 8, 8, 6, 8, 6, 4, 0, 4, 4, 4, 2, 0

$N_6(2a,\cdot) =$ 6, 0, 0, 6, 6, 4, 6, 8, 2, 4, 0, 2, 2, 4, 6, 8

$N_6(2b,\cdot) =$ 12, 0, 4, 0, 0, 4, 2, 2, 2, 6,10, 6,10, 2, 4, 0

$N_6(2c,\cdot) =$ 4, 2, 6, 0, 0, 6, 8, 6, 4, 2, 2, 8, 4, 6, 4, 2

$N_6(2d,\cdot) =$ 6, 2, 2, 6, 6, 4, 4, 2, 6, 2, 4, 8, 4, 2, 4, 2

$N_6(2e,\cdot) =$ 4, 6, 2, 4, 2, 4, 4, 2, 4, 2, 4, 6, 4,10, 4, 2

$N_6(2f,\cdot) =$ 10, 0, 4, 8, 0, 6, 6, 2, 0, 4, 4, 2, 6, 2, 2, 8

$N_6(30,\cdot) =$ 0,12, 8, 2, 0, 6, 0, 0, 6, 6, 0, 2, 8, 2, 6, 6

$N_6(31,\cdot) =$ 2, 6,10, 4, 2, 2, 2, 4, 6, 0, 2, 6, 0, 2, 4,12

$N_6(32,\cdot) =$ 4, 2, 2, 8,10, 8, 8, 6, 0, 2, 2, 4, 4, 2, 2, 0

$N_6(33,\cdot) =$ 4, 2, 2, 2, 6, 0, 4, 0,10, 6, 6, 4, 0, 4, 8, 6

$N_6(34,\cdot) =$ 0, 4, 4, 2, 6, 4, 0, 4, 6, 2, 6, 4, 2, 8, 0,12

$N_6(35,\cdot) =$ 6,12, 4, 2, 4, 2, 2, 4, 8, 2, 2, 0, 6, 4, 4, 2

$N_6(36,\cdot) =$ 0, 2, 2, 4, 4, 4, 4, 0, 2,10,12, 4, 0,10, 4, 2

$N_6(37,\cdot) =$ 10, 2, 2, 6,14, 2, 2, 6, 2, 0, 4, 6, 2, 0, 4, 2

$N_6(38,\cdot) =$ 0, 4,14, 0, 8, 2, 0, 4, 4, 4, 2, 0, 8, 2, 4, 8

$N_6(39,\cdot) =$ 2, 4, 8, 0, 6, 2, 0, 6, 2, 6, 4, 2, 8, 6, 2, 6

$N_6(3a,\cdot) =$ 8, 4, 0, 4, 6, 2, 0, 4, 6, 8, 6, 0, 6, 0, 4, 6

$N_6(3b,\cdot) =$ 0, 4, 6, 6, 2, 2, 2,14, 0,12, 0, 4, 2, 2, 8, 0

$N_6(3c,\cdot) =$ 0, 6,16, 0, 2, 2, 2, 8, 4, 2, 0,12, 6, 2, 2, 0

$N_6(3d,\cdot) =$ 0, 6, 2, 2, 2, 6, 8, 2, 4, 2, 6, 2, 6, 2, 4,10

$N_6(3e,\cdot) =$ 4, 2, 2, 4, 4, 0, 6,10, 4, 2, 4, 6, 6, 2, 6, 2

$N_6(3f,\cdot) =$ 0, 4, 6, 6, 4, 8, 4, 0, 4, 8, 4, 0, 4, 8, 2, 2

$N_7(00,\cdot) =$ 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

$N_7(01,\cdot) =$ 0, 0, 0, 2, 0, 4, 4,14, 0,12, 4, 6, 2, 6, 6, 4

$N_7(02,\cdot) =$ 0, 0, 0, 0, 0,12, 2, 2, 0, 4, 0, 4, 8,12, 6,14

$N_7(03,\cdot) =$ 8, 2,12, 2, 6, 8, 6, 0, 6, 4, 4, 2, 2, 0, 0, 2

$N_7(04,\cdot) =$ 0, 0, 0, 8, 0, 4, 4, 8, 0, 8, 8,12, 2, 6, 2, 2

$N_7(05,\cdot) =$ 6, 0, 0, 2, 8, 0, 8, 4, 0, 2, 6, 0,10, 6, 6, 6

$N_7(06,\cdot) =$ 0, 2,12, 0, 8, 4, 8, 2, 4, 4, 4, 2, 6, 0, 6, 2

$N_7(07,\cdot) =$ 4, 6, 4,12, 0, 4, 2, 0, 0,14, 2, 6, 4, 0, 0, 6

$N_7(08,\cdot) =$ 0, 0, 0, 8, 0, 0, 6,10, 0, 4,12, 4, 6, 6, 0, 8

$N_7(09,\cdot) =$ 10, 8, 4, 8, 6, 2, 2, 0, 2, 6, 8, 2, 0, 6, 0, 0

$N_7(0a,\cdot) =$ 0,10, 6, 2,12, 2, 4, 0, 4, 4, 6, 4, 4, 0, 0, 6

$N_7(0b,\cdot) =$ 0, 2, 2, 2, 4, 8, 6, 4, 4, 0, 4, 2, 6, 4, 2,14

$N_7(0c,\cdot) =$ 0, 0, 0, 4, 0, 4, 8, 4, 0, 2, 6, 0,14,12, 8, 2

$N_7(0d,\cdot) =$ 6, 6, 2, 4, 2, 6, 4, 6, 6, 4, 8, 8, 0, 2, 0, 0

$N_7(0e,\cdot) =$ 0,12,10,10, 0, 2, 4, 2, 8, 6, 4, 2, 0, 0, 2, 2

$N_7(0f,\cdot) =$ 2, 0, 0, 0, 6, 8, 8, 0, 6, 2, 4, 6, 8, 0, 6, 8

$N_7(10,\cdot) =$ 0, 0, 0, 4, 0, 2, 8, 6, 0, 6, 4,10, 8, 4, 8, 4

$N_7(11,\cdot) =$ 6,10,10, 4, 4, 2, 0, 4, 4, 0, 2, 8, 4, 2, 2, 2

$N_7(12,\cdot) =$ 0, 0, 8, 8, 2, 8, 2, 8, 6, 4, 2, 8, 0, 0, 8, 0

$N_7(13,\cdot) =$ 4, 4, 2, 2, 8, 6, 0, 2, 2, 2, 0, 4, 6, 8,14, 0

$N_7(14,\cdot) =$ 0, 8, 6, 2, 8, 8, 2, 6, 4, 2, 0, 2, 8, 6, 0, 2

$N_7(15,\cdot) =$ 4, 4, 8, 2, 4, 0, 4,10, 8, 2, 4, 4, 4, 2, 0, 4

$N_7(16,\cdot) =$ 0, 6,10, 2, 2, 2, 2, 4,10, 8, 2, 2, 0, 4,10, 0

$N_7(17,\cdot) =$ 8, 2, 4, 2, 6, 4, 0, 6, 4, 4, 2, 2, 0, 4, 8, 8

$N_7(18,\cdot) =$ 0,16, 2, 2, 6, 0, 6, 0, 6, 2, 8, 0, 6, 0, 2, 8

$N_7(19,\cdot) =$ 0, 8, 0, 2, 4, 4,10, 4, 8, 0, 6, 4, 2, 6, 2, 4

$N_7(1a,\cdot) =$ 0, 2, 4, 8,12, 4, 0, 6, 4, 4, 0, 2, 0, 6, 4, 8

$N_7(1b,\cdot) =$ 0, 6, 2, 6, 4, 2, 4, 4, 6, 4, 8, 4, 2, 0,10, 2

$N_7(1c,\cdot) =$ 0, 8, 4, 4, 2, 6, 6, 6, 6, 4, 6, 8, 0, 2, 0, 2

$N_7(1d,\cdot) =$ 4, 4, 4, 0, 0, 2, 4, 2, 4, 2, 2, 4,10,10, 8, 4

$N_7(1e,\cdot) =$ 0, 0, 2, 2,12, 6, 2, 0,12, 2, 2, 4, 2, 6, 8, 4

$N_7(1f,\cdot) =$ 2, 2,10,14, 2, 4, 2, 4, 4, 6, 0, 2, 4, 8, 0, 0

$N_7(20,\cdot) =$ 0, 0, 0,14, 0, 8, 4, 2, 0, 4, 2, 8, 2, 6, 0,14

$N_7(21,\cdot) =$ 4, 2, 6, 2,12, 2, 4, 0, 6, 4,10, 2, 4, 2, 2, 2

$N_7(22,\cdot) =$ 10, 6, 0, 2, 4, 4,10, 0, 4, 0,12, 2, 8, 0, 0, 2

$N_7(23,\cdot) =$ 0, 6, 2, 2, 2, 4, 6,10, 0, 4, 8, 2, 2, 6, 0,10

$N_7(24,\cdot) =$ 4, 2, 0, 6, 8, 2, 6, 0, 8, 2, 2, 0, 8, 2,12, 2

$N_7(25,\cdot) =$ 2, 0, 2,16, 2, 4, 6, 4, 6, 8, 2, 4, 0, 6, 0, 2

$N_7(26,\cdot) =$ 6,10, 0,10, 0, 6, 4, 4, 2, 2, 4, 6, 2, 4, 2, 2

$N_7(27,\cdot) =$ 4, 0, 2, 0, 2, 2,14, 0, 4, 6, 6, 2,12, 2, 4, 4

$N_7(28,\cdot) =$ 14, 4, 6, 4, 4, 6, 2, 0, 6, 6, 2, 2, 4, 0, 2, 2

$N_7(29,\cdot) =$ 2, 2, 0, 2, 0, 8, 4, 2, 4, 6, 4, 4, 6, 4,12, 4

$N_7(2a,\cdot) =$ 2, 4, 0, 0, 0, 2, 8,12, 0, 8, 2, 4, 8, 4, 4, 6

$N_7(2b,\cdot) =$ 16, 6, 2, 4, 6,10, 2, 2, 2, 2, 2, 2, 4, 2, 2, 0

$N_7(2c,\cdot) =$ 2, 6, 6, 8, 2, 2, 0, 6, 0, 8, 4, 2, 2, 6, 8, 2

$N_7(2d,\cdot) =$ 6, 2, 4, 2, 8, 8, 2, 8, 2, 4, 4, 0, 2, 0, 8, 4

$N_7(2e,\cdot) =$ 2, 4, 8, 0, 2, 2, 2, 4, 0, 2, 8, 4,14, 6, 0, 6

$N_7(2f,\cdot) =$ 2, 2, 2, 8, 0, 2, 2, 6, 4, 6, 8, 8, 6, 2, 0, 6

$N_7(30,\cdot) =$ 0, 6, 8, 2, 8, 4, 4, 0,10, 4, 4, 6, 0, 0, 2, 6

$N_7(31,\cdot) =$ 0, 8, 4, 0, 6, 2, 2, 6, 6, 0, 0, 2, 6, 4, 8,10

$N_7(32,\cdot) =$ 2, 4, 0, 0, 6, 4,10, 6, 6, 4, 6, 2, 4, 6, 2, 2

$N_7(33,\cdot) =$ 0,16, 6, 8, 2, 0, 2, 2, 4, 2, 8, 4, 0, 4, 6, 0

$N_7(34,\cdot) = 0, 4, 14, 8, 2, 2, 2, 4, 16, 2, 2, 2, 0, 2, 0, 4$

$N_7(35,\cdot) = 0, 6, 0, 0, 10, 8, 2, 2, 6, 0, 0, 8, 6, 4, 4, 8$

$N_7(36,\cdot) = 2, 0, 2, 2, 4, 6, 4, 4, 2, 2, 4, 2, 4, 16, 10, 0$

$N_7(37,\cdot) = 6, 6, 6, 8, 4, 2, 4, 4, 4, 0, 6, 8, 2, 4, 0, 0$

$N_7(38,\cdot) = 0, 2, 2, 2, 8, 8, 0, 2, 2, 2, 0, 6, 6, 4, 10, 10$

$N_7(39,\cdot) = 4, 4, 16, 8, 0, 6, 4, 2, 4, 4, 2, 6, 0, 2, 2, 0$

$N_7(3a,\cdot) = 16, 6, 4, 0, 2, 0, 2, 6, 0, 4, 8, 10, 0, 0, 4, 2$

$N_7(3b,\cdot) = 2, 0, 0, 2, 0, 4, 4, 4, 2, 6, 2, 6, 6, 12, 12, 2$

$N_7(3c,\cdot) = 0, 0, 8, 0, 12, 8, 2, 6, 6, 4, 0, 2, 2, 4, 6, 4$

$N_7(3d,\cdot) = 2, 4, 12, 2, 2, 2, 0, 4, 6, 10, 2, 6, 4, 2, 0, 6$

$N_7(3e,\cdot) = 4, 6, 6, 6, 2, 0, 4, 8, 2, 10, 4, 6, 0, 4, 2, 0$

$N_7(3f,\cdot) = 14, 0, 0, 0, 8, 0, 6, 8, 4, 2, 0, 0, 4, 8, 4, 6$

$N_8(00,\cdot) = 64, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0$

$N_8(01,\cdot) = 0, 0, 0, 6, 0, 16, 10, 0, 0, 0, 6, 0, 14, 6, 2, 4$

$N_8(02,\cdot) = 0, 0, 0, 8, 0, 10, 4, 2, 0, 10, 2, 4, 8, 8, 6, 2$

$N_8(03,\cdot) = 6, 0, 2, 8, 2, 6, 4, 0, 6, 6, 6, 2, 2, 0, 8, 6$

$N_8(04,\cdot) = 0, 0, 0, 2, 0, 4, 6, 12, 0, 6, 8, 4, 10, 4, 8, 0$

$N_8(05,\cdot) = 4, 10, 6, 0, 0, 2, 6, 0, 4, 10, 4, 6, 8, 2, 0, 2$

$N_8(06,\cdot) = 0, 0, 10, 4, 6, 4, 4, 8, 2, 6, 4, 2, 4, 2, 2, 6$

$N_8(07,\cdot) = 6, 2, 8, 2, 8, 10, 6, 6, 4, 2, 0, 4, 0, 0, 0, 6$

$N_8(08,\cdot) = 0, 0, 0, 4, 0, 6, 4, 2, 0, 8, 6, 10, 8, 2, 2, 12$

$N_8(09,\cdot) = 8, 4, 0, 6, 0, 4, 4, 6, 2, 4, 6, 2, 12, 2, 0, 4$

$N_8(0a,\cdot) = 0, 0, 16, 4, 6, 6, 4, 0, 4, 6, 4, 2, 2, 0, 0, 10$

$N_8(0b,\cdot) = 2, 8, 0, 6, 2, 6, 0, 4, 4, 10, 0, 2, 10, 2, 6, 2$

$N_8(0c,\cdot) = 0, 0, 0, 2, 0, 10, 10, 6, 0, 6, 6, 6, 2, 6, 10, 0$

$N_8(0d,\cdot) = 6, 0, 4, 10, 2, 0, 8, 6, 2, 2, 6, 10, 2, 2, 2, 2$

$N_8(0e,\cdot) = 0, 0, 6, 8, 4, 8, 0, 2, 10, 6, 2, 4, 6, 2, 4, 2$

$N_8(0f,\cdot) = 8, 0, 4, 2, 2, 4, 2, 2, 2, 6, 4, 6, 0, 2, 14, 6$

$N_8(10,\cdot) = 0, 0, 0, 4, 0, 0, 8, 12, 0, 0, 8, 8, 2, 10, 6, 6$

$N_8(11,\cdot) = 0, 6, 4, 6, 2, 2, 6, 6, 4, 6, 4, 6, 0, 4, 4, 4$

$N_8(12,\cdot) = 0, 4, 0, 8, 6, 2, 8, 4, 2, 4, 4, 6, 2, 4, 10, 0$

$N_8(13,\cdot) = 4, 2, 2, 6, 8, 6, 2, 2, 14, 2, 2, 4, 2, 2, 2, 4$

$N_8(14,\cdot) = 0, 16, 4, 2, 6, 0, 2, 6, 4, 0, 4, 6, 4, 6, 4, 0$

$N_8(15,\cdot) = 0, 10, 6, 0, 6, 0, 2, 8, 2, 2, 0, 8, 2, 6, 6, 6$

$N_8(16,\cdot) = 0, 12, 6, 4, 6, 0, 0, 0, 8, 6, 6, 2, 2, 6, 4, 2$

$N_8(17,\cdot) = 0, 6, 8, 0, 6, 2, 4, 6, 6, 0, 2, 6, 4, 4, 2, 8$

$N_8(18,\cdot) = 0, 12, 2, 2, 8, 0, 8, 0, 10, 4, 4, 2, 4, 2, 0, 6$

$N_8(19,\cdot) = 6, 4, 8, 0, 8, 0, 4, 2, 0, 0, 12, 2, 4, 6, 2, 6$

$N_8(1a,\cdot) = 0, 4, 6, 2, 8, 8, 0, 4, 8, 0, 0, 0, 6, 2, 0, 16$

$N_8(1b,\cdot) = 2, 4, 8, 10, 2, 4, 2, 8, 2, 4, 8, 2, 0, 2, 4, 2$

$N_8(1c,\cdot) = 0, 12, 6, 4, 6, 4, 2, 2, 6, 0, 4, 4, 2, 10, 2, 0$

$N_8(1d,\cdot) = 8, 6, 0, 0, 10, 0, 0, 8, 10, 4, 2, 2, 2, 8, 4, 0$

$N_8(1e,\cdot) = 0, 4, 8, 6, 8, 2, 4, 4, 10, 2, 2, 4, 2, 0, 6, 2$

$N_8(1f,\cdot) = 4, 2, 4, 2, 6, 2, 4, 0, 2, 6, 2, 2, 2, 16, 8, 2$

$N_8(20,\cdot) = 0, 0, 0, 16, 0, 4, 0, 0, 0, 14, 6, 4, 2, 0, 4, 14$

$N_8(21,\cdot) = 0, 0, 2, 10, 2, 8, 10, 0, 0, 6, 6, 0, 10, 2, 2, 6$

$N_8(22,\cdot) = 8, 0, 6, 0, 6, 4, 10, 2, 0, 6, 8, 0, 4, 4, 2, 4$

$N_8(23,\cdot) = 4, 8, 0, 6, 0, 4, 8, 6, 2, 2, 10, 4, 8, 0, 0, 2$

$N_8(24,\cdot) = 4, 0, 4, 8, 4, 6, 2, 4, 8, 6, 2, 0, 0, 4, 4, 8$

$N_8(25,\cdot) = 0, 4, 6, 8, 2, 8, 8, 0, 4, 2, 4, 4, 2, 2, 6, 4$

$N_8(26,\cdot) = 2, 6, 0, 6, 4, 4, 4, 6, 6, 0, 4, 4, 10, 4, 2, 2$

$N_8(27,\cdot) = 6, 6, 0, 0, 2, 2, 6, 2, 4, 4, 6, 10, 2, 6, 2, 6$

$N_8(28,\cdot) = 10, 2, 6, 2, 4, 12, 12, 0, 2, 2, 4, 0, 0, 0, 2, 6$

$N_8(29,\cdot) = 4, 0, 0, 14, 2, 10, 4, 2, 8, 6, 4, 0, 4, 2, 2, 2$

$N_8(2a,\cdot) = 8, 8, 0, 2, 0, 2, 4, 0, 2, 6, 8, 14, 2, 8, 0, 0$

$N_8(2b,\cdot) = 2, 2, 0, 0, 4, 2, 10, 4, 6, 2, 4, 0, 6, 4, 8, 10$

$N_8(2c,\cdot) = 2, 6, 6, 2, 4, 6, 2, 0, 2, 6, 4, 0, 6, 4, 10, 4$

$N_8(2d,\cdot) = 8, 0, 4, 4, 6, 2, 0, 0, 6, 8, 2, 4, 6, 4, 4, 6$

$N_8(2e,\cdot) = 6, 2, 2, 4, 2, 2, 6, 12, 4, 0, 4, 2, 8, 8, 0, 2$

$N_8(2f,\cdot) = 8, 12, 4, 6, 6, 4, 2, 2, 2, 2, 4, 2, 2, 4, 0, 4$

$N_8(30,\cdot) = 0, 4, 6, 2, 10, 2, 2, 2, 4, 8, 0, 0, 8, 4, 6, 6$

$N_8(31,\cdot) = 4, 6, 8, 0, 4, 6, 0, 4, 4, 6, 10, 2, 2, 4, 4, 0$

$N_8(32,\cdot) = 6, 6, 6, 2, 4, 6, 0, 2, 0, 6, 8, 2, 2, 6, 6, 2$

$N_8(33,\cdot) = 6, 6, 4, 2, 4, 0, 0, 10, 2, 2, 0, 6, 8, 4, 0, 10$

$N_8(34,\cdot) = 0, 2, 12, 4, 10, 4, 0, 4, 12, 0, 2, 4, 2, 2, 2, 4$

$N_8(35,\cdot) = 6, 4, 4, 0, 10, 0, 0, 4, 10, 0, 0, 4, 2, 8, 8, 4$

$N_8(36,\cdot) = 4, 6, 2, 2, 2, 2, 6, 8, 6, 4, 2, 6, 0, 4, 10, 0$

$N_8(37,\cdot) = 2, 2, 8, 2, 4, 4, 4, 2, 6, 2, 0, 10, 6, 10, 2, 0$

$N_8(38,\cdot) = 0, 4, 8, 4, 2, 6, 6, 2, 4, 2, 2, 4, 6, 4, 4, 6$

$N_8(39,\cdot) = 4, 4, 4, 8, 0, 6, 0, 6, 4, 8, 2, 2, 2, 4, 8, 2$

$N_8(3a,\cdot) = 8, 8, 0, 4, 2, 0, 10, 4, 0, 0, 0, 4, 8, 6, 8, 2$

$N_8(3b,\cdot) = 8, 2, 6, 4, 4, 4, 4, 0, 6, 4, 4, 6, 4, 4, 4, 0$

$N_8(3c,\cdot) = 0, 6, 6, 6, 6, 0, 0, 8, 8, 2, 4, 8, 4, 2, 4, 0$

$N_8(3d,\cdot) = 2, 2, 8, 0, 10, 0, 2, 12, 0, 4, 0, 8, 0, 2, 6, 8$

$N_8(3e,\cdot) = 6, 4, 0, 0, 4, 4, 0, 10, 6, 2, 6, 12, 2, 4, 0, 4$

$N_8(3f,\cdot) = 0, 6, 6, 0, 4, 4, 6, 10, 0, 6, 8, 2, 0, 4, 8, 0$

```
1866 zeros, 6326 non-zeros, 8192 total (22.778320% of all are zeros)
```

Vaatleme järgnevas, kuidas S-boxide jaotuse ebaühtlust saab kasutada suuremate DES-i komponentide analüüsimisel. Kuna S-boxide sisend i-ndas raundis on $B = E \oplus J$, kus $E = E(R_{i-1})$ ning $J = k_i$, siis

$$\Delta B = B \oplus B^* = (E \oplus J) \oplus (E \oplus \underbrace{J^*}_{=J}) = E \oplus E^* \quad .$$

Seega *sisend-xor ei sõltu võtmebittidest* samas kui välis-xor tavaliselt sõltub. Tähistame järgnevas $B = B_1||\ldots||B_8$, $E = E_1||\ldots||E_8$ ning $J = J_1||\ldots||J_8$, kus B_i, E_i ja J_i on 6-bitised arvud. Samuti kirjutame lahti suurused B_i^* ja E_i^*, ning nagu alati, olgu $\Delta B_i = B_i \oplus B_i^*$ ning $\Delta E_i = E_i \oplus E_i^*$.

Oletame, et me teame mingi $j \in [1,8]$ jaoks suurusi E_j ja E_j^* ning väljundxori $\Delta C_j = S_j(B_j) \oplus S_j(B_j \oplus B_j^*)$. Me soovime leida tundmatuid $J_j = B_j \oplus E_j = B_j^* \oplus E_j^*$. Selleks paneme esmalt tähele, et $E_j \oplus J_j \in IN_j(\Delta E_j, \Delta C_j)$ ehk

$$J_j \in \mathrm{test}_j(E_j, E_j^*, \Delta C_j) \quad ,$$

kus hulk $\mathrm{test}_j(\cdot,\cdot,\cdot)$ on defineeritud järgmiselt:

$$\mathrm{test}_j(E_j, E_j^*, \Delta C_j) := \{B_j \oplus E_j : B_j \in IN_j(\Delta E_j, \Delta C_j)\} \quad .$$

Sellega oleme tõestanud järgmise teoreemi.

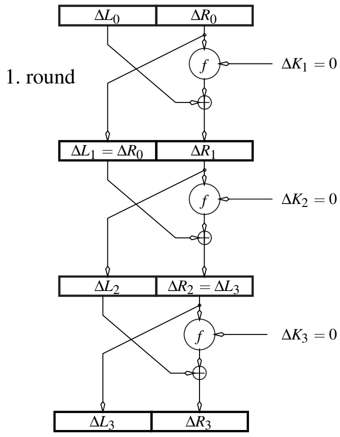
Teoreem 4.3.1 *Olgu E_j ja E_j^* S-boxi S_j kaks sisendit, ning olgu ΔC_j vastav väljund-xor. Siis $J_j \in \mathrm{test}_j(E_j, E_j^*, \Delta C_j)$.*

Kuna

$$|\mathrm{test}_j(E_j, E_j^*, \Delta C_j)| = N_j(\Delta E_j, \Delta C_j) \quad ,$$

siis oleme juhul $E_j \neq E_j^*$ piiranud alamvõtme J_j võimalike väärtuste hulka *vähemalt* neli korda (64-st võimalikust väärtusest on alles jäänud $N_j(\Delta E_j, \Delta C_j) \leq 16$ kandidaatväärtust).

Paneme nüüd tähele, et kui me teame ka mõnda teist $(E_{1,j}, E_{1,j}^*, \Delta C_{1,j})$, siis $J_j \in \mathrm{test}_j(E_j, E_j^*, \Delta C_j) \cap \mathrm{test}_j(E_{1,j}, E_{1,j}^*, \Delta C_{1,j})$. "Piisava arvu" erinevate kolmikute teadmiste järel jääb järele vaid üks alamvõti J_j. Üks võimalus selleks on pidada kõikvõimalike $J_j \in \mathbb{Z}_2^6$ väärtuste kohta loendurite tabelit, mille i-ndat elementi suurendatakse iga kord kui uue kolmiku $(E_j, E_j^*, \Delta C_j)$ puhul $i \in \mathrm{test}_j(E_j, E_j^*, \Delta C_j)$.



Joonis 4.2: Kolmeraundise DES-i erinevustega krüptoanalüüs.

Pärast t kolmiku nägemist on ideaalsel juhul vaid üks loendur, mille väärtus on t. Demonstreerime seda lähenemist kolmeraundise DES-i puhul.

Märgime esiteks, et permutatsioonid IP ja IP^{-1} on erinevustega krüptoanalüüsi seisukohast ebavajalikud (kui soovime, et $\mathrm{IP}(p) \oplus \mathrm{IP}(p^*) = x$, siis valime p ja p^* nii, et $p \oplus p^* = \mathrm{IP}^{-1}(x)$) ja seega jätame nad selles paragrahvis edaspidi vaatluse alt välja. Seega alustame olukorrast, kus meil on teada kaks avatekst-krüptogrammi paari: (L_0R_0, L_3R_3) ning $(L_0^*R_0^*, L_3^*R_3^*)$ (joonis 4.2. Märgime, et siin ja järgmistel joonistel on kastikesse väärtuste L_i ja R_i asemel märgitud vastavad xorid ΔL_i ja ΔR_i).

Jooniselt saame, et

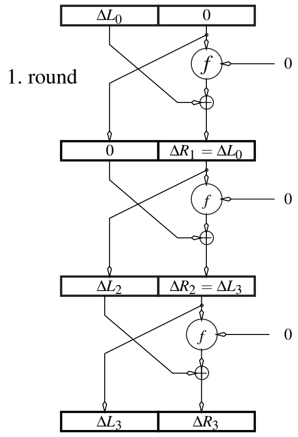
$$\Delta R_3 = \Delta L_0 \oplus f(R_0, K_1) \oplus f(R_0^*, K_1) \oplus f(R_2, K_3) \oplus f(R_2^*, K_3) \quad .$$

Tuletame nüüd meelde, et suuruseid ΔR_0 ja ΔL_0 saab ründaja ise valida. Olgugi siis $\Delta R_0 = 0$ (joonis 4.3). Kui funktsiooni sisend-xor on null, on ka tema väljund-xor null, seega $\Delta f(R_0, K_1) = 0$ ning järelikult

$$\Delta R_3 \oplus \Delta L0 = f(R_2, K_3) \oplus f(R_2^*, K_3) \quad .$$

Viimase võrduse vasak pool on ründajale teada, ning seetõttu saab ta ka leida parema poole. Vastavalt funktsiooni f ehitusele $f(R_2, K_3) = P(C)$ ning $f(R_2^*, K_3) = P(C^*)$, kus C ja C^* on S-boxide kaks väljundit. Seega $P(C) \oplus P(C^*) = \Delta R_3 \oplus \Delta L0$ ning järelikult

$$\Delta C = C \oplus C^* = P^{-1}(\Delta R_3 \oplus \Delta L0) \quad .$$



Joonis 4.3: Kolmeraundise DES-i erinevustega krüptoanalüüs. Juht $\Delta R_0 = 0$.

Kuna $R_2 = L_3$ ja $R_2^* = L_3^*$ on ründajale teada, siis saab ta leida ka suurused $E = E(L_3)$ ning $E^* = E(L_3^*)$ ehk S-boxide sisendid raundis 3. Seega teab ründaja kolmikut $(E, E^*, \Delta C)$ ning saab konstrueerida vastavad hulgad $\text{test}_j(E, E^*, \Delta C)$, mille üheks elemendiks on ka alamvõti J_j.

4.4 DC 6-raundise DES-i korral

Vaatleme nüüd erinevustega krüptoanalüüsi (lühendatult DC) viimistletumat versiooni, mis on edukas 6-raundise DES-i vastu. Ründega kolmeraundise DES-i vastu võrreldes tulevad siin mängu mitmed uued nüansid. Nagu ka paragrahvis 4.3, valitakse ka siin erinevustega krüptoanalüüsi korral (aga tunduvalt hoolikamalt) DES-i sisend-xor (ehk fikseeritud erinevusega sisendid) ning uuritakse erinevuse (täpselt määratud tõenäosusega) propageerumist krüptimise jooksul.

Definitsioon 4.4.1 Olgu $n \geq 1$. Fikseerime mingi (otsitava) võtme. Korteež

$$(\Delta L_0, \Delta R_0), (\Delta L_1, \Delta R_1; p_1), \ldots, (\Delta L_n, \Delta R_n; p_n) \tag{4.1}$$

on DES-i *n-raundine karakteristik* kui järgmised kaks omadust on täidetud:

1. $\Delta L_i = \Delta R_{i-1}$, $1 \leq i \leq n$.

2. Olgu $1 \leq i \leq n$ ja olgu L_{i-1}, R_{i-1}, L_{i-1}^* ja R_{i-1}^* valitud nii, et $L_{i-1} \oplus L_{i-1}^* = \Delta L_{i-1}$ ja $R_{i-1} \oplus R_{i-1}^* = \Delta R_{i-1}$. Olgu (L_i, R_i) ja (L_i^*, R_i^*) vastavalt DES-i ühe

raundi väljundid sisenditel (L_{i-1}, R_{i-1}) ja (L^*_{i-1}, R^*_{i-1}). Siis

$$\mathrm{Prob}_{J\in\mathbb{Z}_2^{48}}[L_i \oplus L_i^* = \Delta L_i \wedge R_i \oplus R_i^* = \Delta R_i] = p_i \ ,$$

kus J on vastavas raundis kasutatav raundivõti.

Karakteristiku tõenäosuseks nimetatakse suurust $p = \prod_{i=1}^{n} p_i$.

Karakteristik annab *lähenduse* $p = \prod_{i=1}^{n} p_i$ tõenäosusele, et lähtudes avatekstidest erinevustega $(\Delta L_0, \Delta R_0)$ saame DES-i raundifunktsiooni järjest rakendades raundide väljund-xorideks paarid $(\Delta L_i, \Delta R_i)$. p on vaid (loodetavasti täpne) lähendus viimasele tõenäosusele, kuna DES-i raundivõtmed ei ole teineteisest sõltumatud (kui oleks sõltumatud, oleks p tõepoolest täpne tõeväärtus). Karakteristik ei vasta tegelikkusele veel ühes punktis: nimelt on võetud karakteristiku tõenäosus üle suvalise (kuid fikseeritud) avateksti paari ning muutuva raundivõtme; samas ründaja üritab mõistatada fikseeritud raundivõtit enda poolt vabalt valitud (fikseeritud erinevusega) avatekstide korral. Ründaja loodab sealjuures, et n-raundi krüptimise jooksul saadud erinevused on ligilähedaselt võrdsed suurustega $p_1, \ldots, p_n$.

Paragrahvis 4.3 toodud ründes kolmeraundise DES-i vastu kasutati järgmist 1-raundist karakteristikut:

$$(x, 0), (0, x; 1) \ , \tag{4.2}$$

kus x võib olla suvaline 32-bitine arv.

Lemma 4.4.1 *Veel üheks karakteristikuks on*

$$(0, \mathtt{04000000}_{16}), (\mathtt{04000000}_{16}, \mathtt{40080000}_{16}, 0.25) \ , \tag{4.3}$$

Tõestus. Tõepoolest, $E(\mathtt{04000000}_{16}) = \mathtt{008000000000}_{16}$, ning seega on teise S-boxi sisend-xoriks $\mathtt{001000}_2$ ning ülejäänud S-boxide sisend-xoriks on 0. Eelnevalt toodud tabelit kasutades võib kontrollida, et

$$N_2(\mathtt{001000}_2, \mathtt{1010}_2) = \max_{i,o} N_2(i, o) = 16 \ .$$

Seega on teise S-boxi väljund-xoriks $\mathtt{1010}_2$ tõenäosusega $p = 0.25$ ning ülejäänud S-boxide väljund-xoriks 0 tõenäosusega 1. Seega on S-boxide väljund-xoriks arv

$$\Delta C = \mathtt{0A000000}_{16}$$

tõenäosusega 0.25, ning raundi väljund-xoriks suurus

$$\Delta L_0 \oplus P(C) \oplus P(C^*) = \Delta L_0 \oplus P(\Delta C) = \mathtt{40080000}_{16}$$

tõenäosusega p. Kuna $\mathrm{Prob}[\Delta L_1 = \Delta R_0] = 1$, siis oleme lemma tõestanud. □

Analoogiliselt saab tõestada, et ka

$$(\texttt{40080000}_{16},\texttt{04000000}_{16}),(\texttt{04000000}_{16},0,0.25) \tag{4.4}$$

on üheraundine karakteristik. Kombineerides toodud üheraundiseid karakteristikuid (4.4), (4.2) ja (4.3), saame järgmise kolmeraundise karakteristiku

$$\begin{aligned}&(\texttt{40080000}_{16},\texttt{04000000}_{16}),\\&(\texttt{04000000}_{16},\texttt{00000000}_{16},0.25),\\&(\texttt{00000000}_{16},\texttt{04000000}_{16},1),\\&(\texttt{04000000}_{16},\texttt{40080000}_{16},0.25)\end{aligned} \tag{4.5}$$

tõenäosusega $p = 0.25^2 = 1/16$.

Ülesanne 4.4.1 Leida DESi kõik kolmeraundised karakteristikud, mille tõenäosus on $\geq 1/16$. Leida maksimaalne neljaraundine karakteristik. Miks see on maksimaalne? Põhjendada (kasutamata argumente, et "arvuti leidis")!

Toodud kolmeraundisel karakteristikul baseerub järgmine valitud avatekstiga rünne kuueraundise DES-i vastu. Ründaja alustab suurustega (L_0,R_0), (L_0^*,R_0^*), (L_6,R_6) ja (L_6^*,R_6^*), kus ründaja on valinud avatekstid erinevusega

$$(\Delta L_0,\Delta R_0) = \texttt{40080000}_{16},\texttt{04000000}_{16}) \tag{4.6}$$

(joonis 4.4). Lähtudes DES-i struktuurist, saame et

$$R_6 = L_5 \oplus f(R_5,K_6) = R_4 \oplus f(R_5,K_6) = L_3 \oplus f(R_3,K_4) \oplus f(R_5,K_6)$$

ning analoogiliselt $R_6^* = L_3^* \oplus f(R_3^*,K_4) \oplus f(R_5^*,K_6)$. Seega

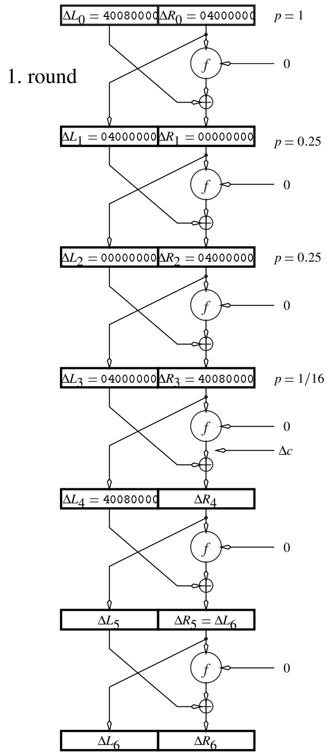
$$\Delta R_6 \oplus \Delta L_3 = f(R_3,K_4) \oplus f(R_5,K_6) \oplus f(R_3^*,K_4) \oplus f(R_5^*,K_6) \ . \tag{4.7}$$

Siin ΔR_6 on teada. Karakteristiku (4.5) ning valitud sisend-xor (4.6) tõttu saame, et tõenäosusega $1/16$ kehtib

$$(\Delta L_3,\Delta R_3) = (\texttt{04000000}_{16},\texttt{40080000}_{16}) \ . \tag{4.8}$$

Oletame, et (4.8) on tõene. Siis neljanda raundi S-boxide sisend-xoriks on

$$\Delta b = E(\Delta R_3) = \texttt{200050000000}_{16} \ .$$



Joonis 4.4: Kuueraundise DES-i erinevustega krüptoanalüüs.

Kirjutades jällegi $\Delta b = (\Delta b_1, \Delta b_2, \ldots, \Delta b_8)$, saame et $\Delta b_2 = \Delta b_5 = \Delta b_6 = \Delta b_7 = \Delta b_8 = 0$ ning seega ka

$$c_i = S_i(b_i) \oplus S_i(b_i \oplus \Delta b_i) = 0, \quad i \in \{2,5,6,7,8\} \ . \tag{4.9}$$

Vaatleme järgnevalt kuuenda raundi S-boxide sisendeid ning väljund-xore. Sisendid $E = E(L_6)$ ja $E^* = E(L_6^*)$ on ründajal teada. Kuuenda raundi väljund-xor on võrdne järgmise väärtusega:

$$\begin{aligned}\Delta C &= P^{-1}(\Delta R_6 \oplus \Delta L_5) = P^{-1}(\Delta R_6 \oplus \Delta R_4) = P^{-1}(\Delta R_6 \oplus \mathtt{04000000}_{16} \oplus P(c)) \\ &= P^{-1}(\Delta R_6 \oplus \mathtt{04000000}_{16}) \oplus c \ .\end{aligned}$$

Olgu $\Delta C' = P^{-1}(\Delta R_6 \oplus \mathtt{04000000}_{16})$. Olgu $\Delta C = \Delta C_1 \Delta C_2 \ldots \Delta C_8$ ning

$$\Delta C' = \Delta C'_1 \Delta C'_2 \ldots \Delta C'_8 \ .$$

Siis (võrdusest (4.9)) saame, et tõenäosusega $1/16$, $\Delta C_i = \Delta C_i'$, $i \in \{2,5,6,7,8\}$.

Seega, tõenäosusega $1/16$ teab ründaja viimase raundi S-boxide sisendeid E_i ja E_i^* ning väljund-xori ΔC_i, kus $i \in \{2,5,6,7,8\}$. Sellest informatsioonist soovib ründaja välja tuletada kuuenda raundi võtme $J = J_1 J_2 \ldots J_8$ alamvõtmeid J_i, $i \in \{2,5,6,7,8\}$ (ehk 30 bitti võtmest k). Nende alamvõtmete leidmist raskendab asjaolu, et kuuenda raundi väljund-xor on korrektne vaid tõenäosusega $1/16$, mis tähendab et $15/16$ ajast saadakse vastuseks "prügi", aga mitte võtmebitte.

Ründe lõpetamiseks on vaja veel kirjeldada protseduur, mis leiab korrektsed võtmebitid andmehulgast, millest enamus ei ole korrektne. Defineerime selleks *õige* ja *vale paari* mõiste.

Definitsioon 4.4.2 Olgu $L_0 \oplus L_0^* = \Delta L_0$ ning $R_0 \oplus R_0^* = \Delta R_0$. Ütleme et avatekstide paar (L_0, R_0) ja (L_0^*, R_0^*) on *õige* karakteristiku (4.1) suhtes kui $L_i \oplus L_i^* = \Delta L_i$ ning $R_i \oplus R_i^* = \Delta R_i$, $\forall i \in [1,n]$. Vastasel korral on tegu *vale paariga*.

Kui ründaja poolt valitud avatekstide paar oli õige (seega umbes $1/16$ndik juhtudest), siis leiab ta soovitud 30 võtmebitti. Seega oleme taandanud küsimuse õigete avatekstipaaride leidmisele, mille hõlbustamiseks on olemas mitmed filtreerimisoperatsioonid.

Üks neist on järgmine. Ründaja leiab kolmikud $(E_i, E_i^*, \Delta C_i)$ ning leiab hulgad $\mathrm{test}_i(E_i, E_i^*, \Delta C_i)$, $i \in \{2,5,6,7,8\}$. Kui ründaja alustas õige võtmepaariga, siis $J_i \in \mathrm{test}_i(E_i, E_i^*, \Delta C_i)$. Kui ründaja alustas vale paariga, siis on väärtus ΔC_i mittekorrektne ning tundub mõistlik oletada, et hulk test_i on lähedane juhuslikule. Seega, kui mingi $i \in \{2,5,6,7,8\}$ korral $|\mathrm{test}_i(E_i, E_i^*, \Delta C_i)| = 0$, siis on tegemist vale võtmega. Kui on antud vale võti siis paragrahvis 4.3 toodud tabelit kasutades võime eeldada, et

$$p := \mathrm{Prob}[|\mathrm{test}_i(E_i, E_i^*, \Delta C_i)| = 0] \approx \mathrm{Prob}_{j,E^1,E^2,C}[N_j(E^1, E^2, C) = 0] \approx 0.23 \ .$$

Tõenäosusega

$$1 - (1-p)^5 \approx 0.73$$

on vähemalt üks hulkadest test_i, $i \in \{2,5,6,7,8\}$, tühi. Seega antud filtreerimisoperatsiooniga elimineeritakse ligikaudu 73% valedest paaridest ning pärast sellist filtreerimist on õigete paaride suhteline hulk ligikaudu

$$\frac{\frac{1}{16}}{\frac{15}{16} + 0.27 \cdot \frac{15}{16}} \approx 0.2 \ .$$

Pärast filtreerimist kehtib kõigi järelejäänud sisendpaaride kohta, et $|\text{test}_i| > 0$, $i \in \{2,5,6,7,8\}$. Ütleme, et 30-bitine string $J_2J_5J_6J_7J_8$ on *soovitatud* (hulkade test_i poolt) kui $J_i \in \text{test}_i$, $i \in \{2,5,6,7,8\}$. Soovitatud bitistringide arv on

$$\prod_{i\in\{2,5,6,7,8\}} |\text{test}_i|$$

(päris sageli on soovitatud bitistringide hulk väga suur, näiteks > 80000). Loendame nüüd, mitu korda esinevad soovitatud bitistringid hulkades test_i. Kui on tegu õige paariga ($\approx 20\%$ juhtudest), on $J_2J_5J_6J_7J_8$ alati soovitatud bitistring ning seega on antud stringile vastav loendur ligikaudu $0.2N$, kus N on valitud avatekstipaaride arv. Kui tegu on vale paariga, soovitatakse sisuliselt juhuslikult valitud stringi $x \in \mathbb{Z}_2^{30}$ ($2^3 0$ võimalust x-i valikuks) ning seega on kõikide mittekorrektsete bitistringide loendurid oluliselt väiksemad.

Loendurite mälushoidmiseks ning alamvõtme $J_2J_5J_6J_7J_8$ leidmiseks on olemas kiired algoritmid. Ülejäänud $56-30=26$ võtmebitti leitakse täieliku läbivaatuse meetodil, selleks kuluv aeg $\Theta(2^{26})$ on täiesti realistlik.

4.5 Algebraline vahepala: lõplikud korpused

Kõik mäletavad, mis asi on korpus: kahe algebralise tehtega $\cdot$ ja $+$ hulk $\mathbb{F}$, nii et

- $(\mathbb{F},+)$ on Abeli rühm,
- kehtib distributiivsuse seadus: $a(b+c) = ab+ac$ ning $(a+b)c = ac+bc$ suvaliste $a,b,c \in \mathbb{F}$ korral.
- $(\mathbb{F}\setminus\{0\},\cdot)$ on rühm.

Korpus on lõplik kui tema elementide arv on lõplik. q-elemendilist lõplikku korpust tähistame $\mathrm{GF}(q)$. Minimaalset elementi p, mille korral

$$p = \underbrace{1+1+\ldots+1}_{p \text{ korda}} = 0$$

nimetatakse *korpuse karakteristikuks* (characteristic). Kui sellist elementi ei leidu, loetakse et korpuse karakteristik on null. On selge, et juhul $p \neq 0$ on p algarv ning et lõpliku korpuse karakteristik ei ole null. Alamkorpus $\mathrm{GF}(p) = \{0,1,\ldots,p-1\} \in \mathrm{GF}(q)$ on isomorfne jäägiklassikorpusega $\mathbb{Z}_p$. Iga korpus on vektorruum üle oma alamkorpuse. Kuna GF(q) on lõplik, siis on tema mõõde n vektorruumina üle korpuse GF(p) lõplik. Kuna baasi valik ruumis GF(q) võimaldab seada üksüheselt vastavusse selle n-mõõtmelise vektorruumi ning kõigi pikkusega n korteežide $\vec{x} \in \mathrm{GF}(p)^n$ vahel, saame et $|\mathrm{GF}(q)| = p^n$ ehk $q = p^n$. Järelikult, *lõpliku korpuse elementide arv on tema karakteristiku (algarvu!) mingi aste*.

Vaatleme nüüd $d$nda astme polünoomi $f(X) \in \mathrm{GF}(q)[X]$. Minimaalset korpust $\mathbb{K}$, $\mathrm{GF}(q) \subseteq \mathbb{K}$, mille korral polünoomil $f(X)$ leidub d juurt korpuses $\mathbb{K}$ nimetatakse polünoomi $f(X)$ *lahutuskorpuseks* üle korpuse GF(q).

Lemma 4.5.1 (Hulga GF(q) **eksisteerimine ja ühesus.)**
1) (Ühesus) Olgu GF(q) *suvaline korpus* $q = p^e$ *elemendiga. Siis iga* $X \in \mathrm{GF}(q)$ *korral* $X^q - X = 0$ *ning* GF(q) *on täpselt võrdne polünoomi* $X^q - X$ *juurte hulgaga.*
2) (Leidumine) Iga algarvu astme $q = p^e$ *jaoks on polünoomi* $X^q - X = 0$ *korpuse* GF(p) *lahutuskorpuses* q *elementi.*

Tõestus.
1) Olgu GF(q) suvaline lõplik korpus q elemendiga. Kuna iga rühma $\mathrm{GF}(q)^* = \mathrm{GF}(q) \setminus 0$ elemendi X järk jagub rühma järguga $q-1$, siis iga $X \neq 0$ korral $X^{q-1} = 1$ ning seega $X^q = X$ ehk $X^q - X = 0$. Triviaalselt $0^q - 0 = 0$. Seega GF(q) kõik elemendid on polünoomi $X^q - X = 0$ juured. Kuna aga igal polünoomil on täpselt q juurt, siis on GF(q) võrdne selle polünoomi juurde hulgaga ning seega on GF(q) võrdne polünoomi $X^q - X = 0$ lahutuskorpusega ning järelikult alati üheselt määratud.
2) Olgu $q = p^e$ algarvu aste ning olgu $\mathbb{K}$ — polünoomi $X^q - X = 0$ lahutuskorpus üle korpuse $\mathrm{GF}(p) = \mathbb{Z}_p$. Kuna polünoomi $f(X) = X^q - X$ formaalne tuletis on $\frac{df}{dX} = qX^{q-1} - 1 = -1$ ($q = p^e = 0$ korpuses GF(p)) ning viimasel ei ole juuri, ei ole ka polünoomil ning tema tuletisel ühiseid juuri ning seega on polünoomi $X^q - X$ kõik juured erinevad. Seega sisaldab $\mathbb{K}$ kõiki qd erinevat polünoomi $X^q - X$ juurt. Samas on juurte hulk juba korpus, kuna kahe juure X_0 ja X_1 summa ja korrutis on samuti juur: $(X_0 + X_1)^q - X_0 - X_1 = X_0^q - X_0 + X_1^q - X_1 = 0$ (vt lemmat 4.5.2) ning $(X_0X_1)^q - X_0X_1 = X_0^qX_1^q - X_0X_1 = 0$. Seega on juurte hulk korpus ning seetõttu ka polünoomi $X^q - X = 0$ lahutuskorpus üle korpuse GF(p). □

Seega oleme näidanud, et iga $q = p^e$ korral leidub üheselt määratud lõplik korpus GF(p^e) ning iga muu q korral ei leidu q-elemendilist lõplikku korpust. Keskendume edaspidi juhule $q = p^e$. Algebrast on teada, et suvalise sellise q korral saab polünoomi $X^q - X = 0$ lahutada hulgas GF$(p)[X]$ kõigi selliste taandatud (*monic*) taandumatute (*irreducible*) polünoomide korrutiseks, mille järk d jagab arvu e.

Definitsioon 4.5.1 Olgu $q = p^e$ ning olgu $\sigma : \mathrm{GF}(q) \to \mathrm{GF}(q)$ järgmine kujutus: iga $x \in \mathrm{GF}(q)$ korral $\sigma(x) = x^p$. Kujutust σ nimetatakse *Frobeniuse kujutuseks*.

Lemma 4.5.2 *Frobeniuse kujutus σ on korpuse $\mathrm{GF}(q)$ automorfism (st bijektsioon, mis säilitab liitmise ja lahutamise), kusjuures $\sigma(x) = x \iff x \in \mathrm{GF}(p)$. Kujutuse σ järk (st vähim arv f, nii et σ^f on samasuskujutus) on e.*

Tõestus. Kujutus σ säilitab triviaalselt korrutamist:

$$\sigma(X_0X_1) = (X_0X_1)^p = X_0^pX_1^p = \sigma(X_0)\sigma(X_1) \enspace .$$

Liitmise säilimise tõestamiseks märgime, et

$$\sigma(X_0+X_1) = (X_0+X_1)^p = \sum \binom{p}{j} X_0^j X_1^{p-j} = X_0^p + X_1^p \enspace .$$

Nüüd, kujutus $\sigma^j : X \mapsto X^{p^j}$ ning seega on selle kujutuse püsipunktideks polünoomi $X^{p^j} - X$ juured. Kui $j = 1$, siis lemmast 4.5.1 saame, et juurte hulk on võrdne korpusega $\mathrm{GF}(p)$. Kujutuse σ^e püsipunktideks on polünoomi $X^q - X$ juurte hulk, st kogu korpus $\mathrm{GF}(q)$, seega on σ^e samasuskujutus. Seetõttu on σ bijektsioon pöördkujutusega $\sigma^1 : X \mapsto X^{p^{e-1}}$. Suvalise $1 \le j < e$ korral σ^e ei ole samasuskujutus, kuna kõik $\mathrm{GF}(q)$ elemendid ei saa olla polünoomi $X^{p^j} - X$ juured. □

Kuidas konstrueerida korpust $\mathrm{GF}(p^e)$? Edasisse algebrasse süüvimata märgime mitteformaalselt, et (suvalise) fikseeritud taandumatu $e$nda astme polünoomi $f(X)$ korral on $\mathrm{GF}(p^e)$ esituv korpusena $\mathrm{GF}(p)/f(X)$ ehk kõikide polünoomide modulo $f(X)$ korpusena, mille kordajad on võetud hulgast $\mathrm{GF}(p)$.

Näide 4.5.1 Konstrueerime korpuse $\mathrm{GF}(2^8)$. Valime selleks esmalt 8nda astme taandatud taandumatu polünoomi

$$f(X) = X^8 + X^4 + X^3 + X + 1 \enspace .$$

Iga korpuse $\mathrm{GF}(2^8)$ liige A vastab üksüheselt korteežile $(a_0, \dots, a_7)$, $a_i \in \mathrm{GF}(2)$, mida saab esitada polünoomina $A = \sum_{i=0}^{7} a_iX^i$ aga ka 8-bitise arvuna $\sum a_i 2^i$. Muuhulgas $f(X) = \mathtt{11B}_{16}$.

Kahe polünoomi $A = \sum_{i=0}^{7} a_iX^i$ ning $B = \sum_{i=0}^{7} b_iX^i$ liitmine toimub koordinaatkaupa:

$$A + B = \sum_i a_iX^i + \sum_i b_iX^i = \sum_i (a_i \oplus b_i)X^i \enspace .$$

(kuna koordinaadid kuuluvad hulka $\mathrm{GF}(2) = \mathbb{Z}_2$, siis on liitmistehteks xor). Kuna $A + B < f(X)$, siis ei ole liitmise juures jagamine modulo $f(X)$ vajalik.

Korrutamisel võib tekkida olukord, kus $A \cdot B > f(X)$, sellisel juhul tuleb ka jagada. Näiteks:

$$\begin{aligned}
\mathtt{57}_{16} \cdot \mathtt{83}_{16} &= (X^6 + X^4 + X^2 + X + 1)(X^7 + X + 1) \mod f(X) \\
&= X^{13} + X^{11} + X^9 + X^8 + X^6 + X^5 + X^4 + X^3 + 1 \mod f(X) \\
&= X^7 + X^6 + 1 = \mathtt{C1}_{16} \ .
\end{aligned}$$

Kahe polünoomi (ehk kaheksabitise arvu) $A(X)$ ja $B(X)$ korrutamist modulo $f(X)$ saab efektiivselt teostada, kasutades järgmist algoritmi. Esmalt paneme tähele, et $X \cdot A(X)$ on võrdne jäägiga, mis saadakse polünoomi

$$a_7X^8 + a_6X^7 + \ldots + a_0X$$

jagamisel polünoomiga $f(X)$. Kui $a_7 = 0$, ei tule redutseerimiseks midagi teha. Kui $a_7 = 1$, siis tuleb toodud suurusest lahutada $f(X)$. Olgu A ja f vastavalt polünoomide $A(X)$ ja $f(X)$ esitused 16ndsüsteemis mod 2^8 (st $f = \mathtt{1B}_{16}$). Siis

$$X \cdot A(X) = \begin{cases} A \ll 1, & a_7 = 0 \ , \\ (A \ll 1) \oplus f, & a_7 = 1 \ . \end{cases}$$

Korrutise $A(X) \cdot B(X) = A(X) \cdot \sum_i b_i X^i$ saab leida nüüd juba tavalist korrutamisalgoritmi kasutades. Näiteks

$$A(X) \cdot (X^4 + X) = ((((A(X) \cdot X) \cdot X) \cdot X) \cdot X) \oplus (A(X) \cdot X) \ .$$

4.6 Tõestatav turvalisus DC vastu

Paragrahvis 4.4 kirjeldatud rünne põhines kolmeraundisel karakteristikul (4.5). Ründe efektiivsuse juures oli üks olulisemaid faktoreid selle karakteristiku tõenäosus $1/16$: kui leiduks mõni suurema tõenäosusega karakteristik, siis oleks õigete paaride arvu suhe valede paaride arvusse suurem ning siis kuluks ka vähem valitud avatekstide paare, et tuvastada õige alamvõti. Seega, suure tõenäosusega $(r-3)$-raundised karakteristikud aitavad r-raundise šifri vastu (me ei piirdu järgnevas ainult DES-iga) erinevustega krüptoanalüüsi (DC-d) teostada. Samas erinevustega krüptoanalüüsi läbiviimine on raske, kui ei leidu suure tõenäosusega karakteristikuid.

Täpsustame seda intuitiivset tähelepanekut. Esmalt märgime, et ründe teostamise edukus ei sõltu otseselt maksimaalse karakteristiku tõenäosusest (4.1), kuna me ründes ei kasutanud suurusi $(\Delta L_1, \Delta R_1)$ ning $(\Delta L_2, \Delta R_2)$ üldse: oluline oli vaid tõenäosus

$$\mathrm{Prob}[(\Delta L_3, \Delta R_3)|(\Delta L_0, \Delta R_0)] \ .$$

Mitteformaalselt, paari

$$((\Delta L_0, \Delta R_0), (\Delta L_r, \Delta R_r)) \ ,$$

kus $(\Delta L_0, \Delta R_0) \neq 0$, nimetatakse r-raundiseks *diferentsiaaliks*. Märgime et paragrahvis 4.4 toodud rünne kasutas kolmeraundist diferentsiaali

$$(\mathtt{40080000}_{16}, \mathtt{04000000}_{16}), (\mathtt{04000000}_{16}, \mathtt{40080000}_{16}) \ ,$$

mille tõenäosus on $1/16$. Seega, erinevustega krüptoanalüüs on edukas mingi šifri vastu siis kui selles leidub r-raundine, r on piisavalt suur, suure tõenäosusega diferentsiaal. Järelikult, et muuta šifrit erinevustega krüptoanalüüsi vastu turvaliseks, tuleb leida ülemtõke r-raundiste diferentsiaalide tõenäosusele. Järgnevalt vaatleme kuidas selliseid ülemtõkkeid anda, ehk kuidas konstrueerida erinevustega krüptoanalüüsi vastu tõestatavalt turvalisi “DESi-sarnaseid šifreid”. Käsitlus põhineb artiklil [NK95].

Definitsioon 4.6.1 (DESi-sarnane šiffer) Olgu $2n$ ja r fikseeritud arvud (plokipikkus ning raundide arv). Olgu $m \geq n$. Olgu

$$f : \mathrm{GF}(2^m) \to \mathrm{GF}(2^n)$$

ja

$$E : \mathrm{GF}(2^n) \to \mathrm{GF}(2^m)$$

kaks kujutust, kus E on afiinne laienduskujutus. Olgu

$$K = (K_1, K_2, \dots, K_r) \ ,$$

$K_i \in \mathrm{GF}(2^m)$, raundivõtmed. Raundifunktsioon

$$F : \mathrm{GF}(2)^n \times \mathrm{GF}(2)^m \to \mathrm{GF}(2)^n$$

on defineeritud järgnevalt: $F(X, K_i) = f(E(X) + K_i)$.

Võtme $K = (K_1, \ldots, K_r)$ ning avateksti $X = (X_L, X_R)$ korral leitakse krüptogramm $Y = (Y_L, Y_R)$ järgmiselt. Olgu $X_L(0) = X_L$ ning $X_R(0) = X_R$, ning iga $i \in [1, r]$ jaoks olgu

$$\begin{aligned} X_L(i) &= X_R(i-1) \ , \\ X_R(i) &= F(X_R(i-1), K_i) + X_L(i-1) \ , \\ X(i) &= (X_L(i), X_R(i)) \ . \end{aligned}$$

Kahe n-bitise ploki erinevus ΔX on defineeritud kui $\Delta X = X + X^*$. Ütleme, et raundifunktsioon $F(X, K)$ on *krüptograafiliselt nõrk*, kui üksikuid paare

$$(X, X^*, \Delta F(X))$$

(kus $\Delta F(X) = F(X) \oplus F(X^*)$) teades on võimalik (enamasti) välja arvutada alamvõtit K.

Märkus 4.6.1 NB! Eeldame, et raundivõtmed on sõltumatult valitud! Märgime, et DES-i raundifunktsioon on nõrk.

4.6.1 DC formaliseering

Mida tähendab sellise šifri korral turvalisus erinevustega krüptoanalüüsi vastu? Erinevustega krüptoanalüüs r-raundise plokkšifri vastu põhineb üldjuhul järgmisel protseduuril [LMM91]:

1. Leitakse $(r-1)$-raundine diferentsiaal (α, β), nii et suuruse $\mathrm{Prob}[\Delta X(r-1) = \beta | \Delta X(0) = \alpha]$ tõenäosus on (peaaegu) maksimaalne.

2. Valitakse juhuslikult avatekst X ning leitakse $X^* := X \oplus \alpha$. Krüpteeritakse X ja X^* tundmatu võtme K korral. Saadud krüptogrammidest $(X(r), X^*(r))$ leida viimase raundi raundivõtme K_r kõik võimalikud väärtused, mis vastavad erinevusele $\Delta X(r-1) = \beta$. Lisada üks kõigi selliste alamvõtmete K_r väärtuste loendurile.

3. Korrata eelmist sammu kuni ühele (või rohkematele) alamvõtme K_r väärtus(t)ele vastavate loendurid on oluliselt suuremad kui teistele väärtustele vastavad loendurid. Fikseerime kõige rohkem loendatud alamvõtme (või väikese hulga alamvõtmeid) kui krüptoanalüütiku otsustuse alamvõtme K_r kohta.

Nagu juba eelnevalt märgitud, on erinevustega krüptoanalüüsi (DC) korral kõik raundivõtmed on fikseeritud ja ainult avatekste saab juhuslikult valida. Samas diferentsiaali tõenäosuse arvutamisel eeldatakse, et ka alamvõtmed on juhuslikult valitud (ühtlase jaotuse järgi). DC-ks valmistumisel kasutatakse nii arvutatud diferentsiaali tõenäosust, et otsustada millist diferentsiaali kasutada ründe käigus. Seega tehakse DCs alati vaikimisi järgmine hüpotees:

Eeldus 4.6.1 (Stohhastilise samaväärsuse hüpotees) *Suvalise* $(r-1)$*-raundise diferentsiaali* (α,β) *korral*

$$\begin{aligned}&\mathrm{Prob}[\Delta X(r-1)=\beta|\Delta X(0)=\alpha]\approx\\&\qquad\mathrm{Prob}[\Delta X(r-1)=\beta|\Delta X(0)=\alpha,K_1=\kappa_1,\dots,K_{r-1}=\kappa_{r-1}]\end{aligned}$$

peaaegu kõigi alamvõtmeväärtuste $(\kappa_1,\dots,\kappa_{r-1})$ *korral.*

Eeldusel 4.6.1 ning eeldusel, et raundifunktsioon on nõrk, saame et r-raundine DESi-sarnane šifrit saab murda DCd kasutades parajasti siis kui leidub selline $(r-1)$-raundine diferentsiaal (α,β), et

$$\mathrm{Prob}[\Delta X(r-1)=\beta|\Delta X=\alpha]\gg 2^{-m}\quad.$$

Lemma 4.6.1 *Fikseerime DESi-sarnase šifri* $\mathcal{E}$*. Kehtigu eeldus 4.6.1. Siis on DC jaoks vaja teha vähemalt*

$$C(\mathcal{E})=\frac{2}{p_{max}(r-1)-\frac{1}{2^m-1}}$$

$\mathcal{E}$*-ga krüpteerimist, kus*

$$p_{max}(r-1)=\max_{\beta}\max_{\alpha\neq 0}\mathrm{Prob}[\Delta X(r-1)=\beta|\Delta X(0)=\alpha]\quad.$$

Tõestus. DC õnnestumiseks peab valitud erinevuse $\Delta(r-1)=\beta$ väärtusele vastava loenduri suurus peab olema keskmiselt vähemalt ühe võrra suurem kui juhuslikult valitud väärtuse β' oma. Seega T katse järgi õnnestumiseks peab olema täidetud $Tp_{\max}(r-1)\geq\frac{T}{2^m-1}+1$, kus katse seisneb erinevusega α avatekstipaari valimises ning nendele vastavate krüptogrammide arvutamises. □

4.6.2 Markovi šifrid

Erinevustega krüptoanalüüsis on eriti huvitavad *Markovi šifrid*, kus jada

$$(\Delta X(0), \Delta X(1), \ldots, \Delta X(r))$$

moodustab Markovi ahela.

Definitsioon 4.6.2 Diskreetsete juhuslike suuruste $v_0, v_1, \ldots, v_r$ jada on *Markovi ahel*, kui iga $0 \leq i < r$ jaoks

$$\begin{aligned}\mathrm{Prob}[v_{i+1} = \beta_{i+1} | v_i = \beta_i, v_{i-1} = \beta_{i-1}, \ldots, v_0 = \beta_0] \\ = \mathrm{Prob}[v_{i+1} = \beta_{i+1} | v_i = \beta_i] \quad .\end{aligned}$$

Markovi ahel on *homogeenne*, kui $\mathrm{Prob}[v_{i+1} = \beta | v_i = \alpha]$ on ist sõltumatu kõigi α ja β väärtuste korral.

Definitsioon 4.6.3 DESi-sarnane šiffer on *Markovi šiffer*, kui suvaliste suuruste $\alpha \neq 0$ ja $\beta \neq 0$ korral

$$\mathrm{Prob}[\Delta X(1) = \beta | \Delta X(0) = \alpha, X(0) = \gamma]$$

on sõltumatu suurusest γ kui alamvõti J on valitud ühtlase jaotuse järgi ehk kui

$$\mathrm{Prob}[\Delta X(1) = \beta | \Delta X(0) = \alpha, X(0) = \gamma] = \mathrm{Prob}[\Delta X(1) = \beta | \Delta X(0) = \alpha] \quad .$$

Kehtib järgmine teoreem [LMM91], mille tõestust me siinkohas ei esita:

Teoreem 4.6.1 *Kui r-raundine DESi-sarnane šiffer on Markovi šiffer ja kõik r raundivõtit on valitud ühtlase jaotuse järgi ja sõltumatult, siis on jada*

$$(\Delta X(0), \Delta X(1), \ldots, \Delta X(r))$$

homogeenne Markovi ahel.

Lemma 4.6.2 *DES on Markovi šiffer.*

Tõestus. Olgu $\alpha \neq 0$, $\beta \neq 0$ ning $p = \mathrm{Prob}[\Delta X(1) = \beta | \Delta X(0) = \alpha]$. Teame, et alati (sõltumatult X-i valikust) $\Delta X_L(1) = \Delta X_R(0)$ ning $\Delta X_R(1) = \Delta X_L(0) + \Delta F(X(0), J)$. Seega peame näitama veel, et

$$\begin{aligned}\mathrm{Prob}[\Delta F(X(0), J) = \beta | \Delta X(0) = \alpha, X(0) = \gamma] \\ = \mathrm{Prob}[\Delta F(X(0), J) = \beta | \Delta X(0) = \alpha] \quad ,\end{aligned}$$

kus F on DESi raundifunktsioon. Vastavalt DESi ehitusele,

$$\Delta F(X(0),J) = \Delta P(S(E(X(0)\oplus J)) = P(S(E(X(0)\oplus J))\oplus P(S(E(X^*(0)\oplus J)) \ ,$$

mis on sõltumatu avateksti $X(0)$ konkreetsest valikust parajasti siis kui seda on suurus $\Delta S(E(X(0)\oplus J)$. Me teame viimaste väärtust (ehk S-boxide väljund-xori) $P^{-1}(\beta_R)$. Igale sellisele väljund-xorile vastavad teatud sisend-xorid.

Näitame, et $\mathrm{Prob}[S(\Delta E) = \Delta C] = \prod_i \frac{N_i(\Delta E_i,\Delta C_i)}{64}$ ei sõltu avateksti $X(0)$ valikust. Tõepoolest, iga sisendi $X(0) = \alpha_R$ ja S-boxide sisend-xori ΔE korral saab valida täpselt $\prod_i N_i(\Delta E_i,\Delta C_i)$ võtit J, nii et iga J jaoks, S-boxide sisendpaari $(E(\alpha_R)\oplus J, E^*(\alpha_R)\oplus J)$ korral S-boxide väljund-xor on ΔC. □

Toodud tõestus laieneb kõigi DESi-sarnaste šifrite jaoks ning seega on kõik DESi-sarnased šifrid Markovi šifrid. Järelikult, tänu Chapman-Kolmogorovi võrdusele Markovi ahelate jaoks, kehtib kõigi DESi-sarnaste šifrite jaoks

$$\begin{aligned}\mathrm{Prob}[\Delta X(1) = \beta_1,\ldots,\Delta X(r) = \beta_r|\Delta X(0) = \beta_0]\\ = \prod_{i=1}^{r}\mathrm{Prob}[\Delta X(1) = \beta_i|\Delta X(0) = \beta_{i-1}]\end{aligned}$$

ning seega $(r-1)$-raundise diferentsiaali (β_0,β_r) tõenäosus on

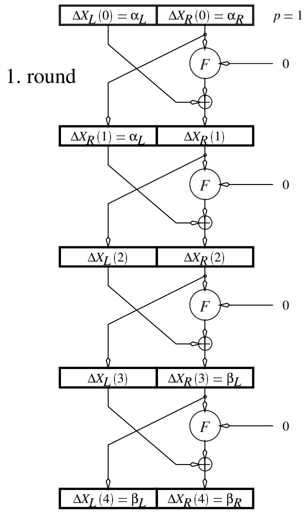
$$\begin{aligned}&\mathrm{Prob}[\Delta X(r) = \beta_r|\Delta X(0) = \beta_0]\\ &= \sum_{\beta_1}\cdots\sum_{\beta_r}\prod_{i=1}^{r}\mathrm{Prob}[\Delta X(1) = \beta_i|\Delta X(0) = \beta_{i-1}]\end{aligned}$$

Tuletame meelde, et lemmast 4.6.1 järeldub, et r-raundine DESi-sarnane šiffer on DC vastu (tõestatavalt) turvaline kui $p_{\max}(r-1)\approx\frac{1}{2^m-1}$. Näitame järgnevalt, et sellise šifri korral väärtus $p_{\max}(r-1)$ on ülevalt piiratud ruutfunktsiooniga väärtusest

$$p_{\max} = \max_{\beta}\max_{\alpha_R\neq 0}\mathrm{Prob}[\Delta X(1) = \beta|\Delta X(0) = \alpha] \ .$$

4.6.3 Ülemtõke diferentsiaalidele neljaraundise šifri korral

Teoreem 4.6.2 *Olgu $\mathcal{E}$ — r-raundine, $r\geq 4$ DESi-sarnane šiffer juhuslikult ja sõltumatult valitud raundivõtmetega. Siis $p_{max}(s)\leq 2p_{max}^2$, kui $r\geq s\geq 4$.*



Joonis 4.5: Erinevuste propageerumine neljaraundises DESi-sarnases šifris.

Tõestus. Anname esmalt tõestuse juhul $s = 4$, st näitame et

$$\mathrm{Prob}[\Delta X(4) = \beta | \Delta X(0) = \alpha] \leq 2p_{\max}^2$$

suvaliste $\alpha = (\alpha_L, \alpha_R) \neq 0$ ja $\beta = (\beta_L, \beta_R) \neq 0$ jaoks. (Joonisel 4.5 on toodud erinevuste propageerumine neljaraundises DESi-sarnases šifris.) Tähistagu $\delta \to \varepsilon$ fakti, et s-raundise diferentsiaali (α, β) esinemiseks peavad funktsiooni F sisendid erinevusega δ alati endaga kaasa tooma väljunderinevuse ε. Märgime, et kui $\beta_L = 0$ siis $\beta_R \neq 0$ (sest muidu $\beta = 0$) ning kui $\alpha_L = 0$ siis $\alpha_R \neq 0$.

Juht $\beta_L = 0$. Siis ka $\Delta_R(3) = 0$ ning seega ka $\Delta F(\Delta_R(3), J) = 0$, mistõttu $\Delta X_R(2) = \Delta_L(3) = \Delta X_R(4) \oplus 0 = \beta_R \neq 0$ ning teises raundis $\Delta F(X, K_2) = \beta_R \oplus \alpha_R$. Jaotame selle juhu omakorda kaheks alamjuhuks. 1) Kui $\Delta X_R(1) = 0$, siis $\Delta X_R(2) = \beta_R = \alpha_R \oplus \Delta F(X, K_2) = \alpha_R \neq 0$. Sellisel juhul esimeses raundis $\alpha_R = \beta_R \to \alpha_L$ ning kolmandas raundis $\Delta X_R(2) = \beta_R \to 0$ (tuletame meelde, et mõlema sündmuse tõenäosus on $\leq p_{\max}$). 2) Kui $\Delta X_R(1) \neq 0$, siis fikseeritud $\Delta X_R(1)$ korral teises raundis $\Delta X_R(1) \to \alpha_R \oplus \beta_R$ ja kolmandas raundis $\Delta X_R(2) = \beta_R \to \Delta X_R(1) \beta_L = \Delta X_R(1)$ (jällegi on mõlema sündmuse tõenäosus $\leq p_{\max}$). Seega Markovi šifri omaduste põhjal

$$
\begin{aligned}
&\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha]\\
&\quad=\sum_{\Delta X_R(1)}\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha]\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha,\Delta X_R(1)]\\
&\quad=\mathrm{Prob}[\Delta X_R(1)=0|\Delta X(0)=\alpha]\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha,\Delta X_R(1)=0]+\\
&\qquad\sum_{\Delta X_R(1)\neq 0}\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha]\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha,\Delta X_R(1)]\\
&\quad\le p_{\max}^2+\sum_{\Delta X_R(1)\neq 0}\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha]\cdot p_{\max}^2\\
&\quad\le 2p_{\max}^2\quad.
\end{aligned}
$$

Juht $\beta_L\neq 0$. Vaatleme esmalt kolmeraundist diferentsiaali, mis saadakse suuruse $\Delta X_R(1)$ fikseerimisel:

$$
\begin{aligned}
&\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha,\Delta X_R(1)]\\
&\quad=\sum_{\Delta X_R(2)}\mathrm{Prob}[\Delta X_R(2)|\Delta X(0)=\alpha,\Delta X_R(1)]\cdot\\
&\qquad\qquad\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha,\Delta X_R(1),\Delta X_R(2)]\\
&\quad=\mathrm{Prob}[\Delta X_R(2)=0|\Delta X(0)=\alpha,\Delta X_R(1)]\cdot\\
&\qquad\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha,\Delta X_R(1),\Delta X_R(2)=0]+\\
&\qquad\sum_{\Delta X_R(2)\neq 0}\mathrm{Prob}[\Delta X_R(2)|\Delta X(0)=\alpha,\Delta X_R(1)]\cdot\\
&\qquad\qquad\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha,\Delta X_R(1),\Delta X_R(2)]\\
&\quad\le p_{\max}^2+\sum_{\Delta X_R(2)\neq 0}\mathrm{Prob}[\Delta X_R(2)|\Delta X(0)=\alpha,\Delta X_R(1)]\cdot p_{\max}\\
&\quad\le 2p_{\max}^2\quad.
\end{aligned}
$$

(Siin kasutasime, et kui $\Delta X_R(2)=0$ siis $\Delta X_R(1)\neq 0$, sest muidu $\alpha_L=\alpha_R=0$). Siit järeldub, et antud teoreem on juhul $\beta_L\neq 0$ tõene ka kolmeraundiste diferentsiaalide jaoks. Nüüd

$$\begin{aligned}
&\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha]\\
&=\sum_{\Delta X_R(1)}\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha]\mathrm{Prob}[\Delta X(4)=\beta|\Delta X(0)=\alpha,\Delta X_R(1)]\\
&\leq\sum_{\Delta X_R(1)}\mathrm{Prob}[\Delta X_R(1)|\Delta X(0)=\alpha]\cdot 2p_{\max}^2\\
&\leq 2p_{\max}^2 \quad .
\end{aligned}$$

Juhul $s>4$

$$\begin{aligned}
&\mathrm{Prob}[\Delta X(s)=\beta|\Delta X(0)=\alpha]\\
&=\sum_{\Delta X_R(s-4)}\mathrm{Prob}[\Delta X_R(s-4)|\Delta X(0)=\alpha]\cdot\\
&\qquad\mathrm{Prob}[\Delta X(s)=\beta|\Delta X(0)=\alpha,\Delta X_R(s-4)] \quad .
\end{aligned}$$

Kuna võtmed on juhuslikult ja sõltumatult valitud ja tegu on Markovi šifriga, siis saame juhu $s=4$ tõestusest, et

$$\mathrm{Prob}[\Delta X(s)=\beta|\Delta X(0)=\alpha,\Delta X_R(s-4)]$$
$$=$$
$$Prob\Delta X(s)=\beta|\Delta X_R(s-4)\leq 2p_{\max}^2$$

ning seega $\mathrm{Prob}[\Delta X(s)=\beta|\Delta X(0)=\alpha]\leq 2p_{\max}^2$. □

Kui f on permutatsioon, siis kehtib teoreemi 4.6.2 tulemus $s\geq 3$ korral.

Teoreem 4.6.3 *Olgu antud DESi-sarnane šiffer, milles f on permutatsioon, ning raundivõtmed on valitud juhuslikult ja sõltumatult. Siis s-raundise, $s\geq 3$, diferentsiaali tõenäosus on $\leq 2p_{max}^2$.*

Tõestus. Anname tõestuse $s=3$ jaoks. Üldjuhtu tõestatakse nagu teoreemis 4.6.2.

Juht $\beta_L=0$. Siis $\Delta_R(0)=\alpha_R\neq 0$, sest vastasel korral oleks teises raundis erinevatel sisenditel sama väljund, mis pole võimalik kuna f on permutatsioon. Sisendite erinevus esimeses raundis on $\alpha_R\neq 0$ ning teises raundis $\beta_r\neq 0$, seega $\mathrm{Prob}[\Delta X(3)=\beta|\Delta X(0)=\alpha]\leq p_{\max}^2$.

Juht $\beta_L \neq 0$. Kui $\Delta X_R(1) = 0$ siis $\alpha_L \neq 0$, vastasel korral $\alpha_R \to \alpha_L = 0$, millest järelduks $\alpha_R = 0$. Seega

$$\begin{aligned}
&\mathrm{Prob}[\Delta X(3) = \beta | \Delta X(0) = \alpha] \\
&\quad = \sum_{\Delta X_R(1)} \mathrm{Prob}[\Delta X_R(1) | \Delta X(0) = \alpha] \cdot \mathrm{Prob}[\Delta X(3) = \beta | \Delta X(0) = \alpha, \Delta X_R(1)] \\
&\quad = \mathrm{Prob}[\Delta X_R(1) = 0 | \Delta X(0) = \alpha] \cdot \\
&\qquad \mathrm{Prob}[\Delta X(3) = \beta | \Delta X(0) = \alpha, \Delta X_R(1) = 0] + \\
&\qquad \sum_{\Delta X_R(1) \neq 0} \mathrm{Prob}[\Delta X_R(1) | \Delta X(0) = \alpha] \cdot \\
&\qquad \mathrm{Prob}[\Delta X(3) = \beta | \Delta X(0) = \alpha, \Delta X_R(1)] \\
&\quad \leq p_{\max}^2 + \sum_{\Delta X_R(1) \neq 0} \mathrm{Prob}[\Delta X_R(1) | \Delta X(0) = \alpha] \cdot p_{\max}^2 \\
&\quad \leq p_{\max}^2 \quad .
\end{aligned}$$

□

4.7 Kvaasitäiuslikult mittelineaarsed permutatsioonid

4.7.1 Algmõisted

Tõestasime, et kolmeraundise DESi-sarnase šifri korral kui f on permutatsioon, siis $p_{\max}(3) \leq 2p_{\max}^2$. Järgnevas leiame mõned kandidaatpermutatsioonid f, mille korral $p_{\max}$ on võimalikult väike. Selleks märgime, et

$$\Delta X(1) = (\Delta\alpha_R, \Delta\alpha_L + f(E(\alpha_R), J) + f(E(\alpha_R^*), J))$$

ning seega

$$\begin{aligned}
p_{\max} &= \max_{\beta} \max_{\alpha_R \neq 0} \mathrm{Prob}[\Delta X(1) = \beta | \Delta X(0) = \alpha] \\
&= \max_{\beta} \max_{\alpha_R \neq 0} \mathrm{Prob}[\Delta X_R(1) = \beta_R | \Delta X(0) = \alpha] \\
&= \max_{\beta} \max_{\alpha_R \neq 0} \mathrm{Prob}[\alpha_L + f(E(X) + J) + f(E(X + \alpha_R) + J) = \beta_R] \leq p_f \quad ,
\end{aligned}$$

kus

$$\begin{aligned} p_f &:= \max_u \max_{w \neq 0} \mathrm{Prob}[f(x+w) + f(x) = u] \\ &= \max_u \max_{w \neq 0} \mathrm{Prob}[f(x+w) - f(x) = u] \end{aligned}$$

(kuna korpuses GF(2) '+'='$-$'='$\oplus$'). Siin me eeldasime, et E on afiinne.

Definitsioon 4.7.1 Kujutus $g : \mathrm{GF}(q)^n \to \mathrm{GF}(q)$ on *afiinne* kui $g(x) = u^t x + v$, kus $u \in \mathrm{GF}(q)^n$, $v \in \mathrm{GF}(q)$.

Suvalise kujutuse $f : \mathrm{GF}(2)^m \to \mathrm{GF}(2)^n$ (märgime, et $\mathrm{GF}(2)^n = \mathbb{Z}_2^n$, $\forall n$) korral on $p_f \geq |\mathrm{GF}(2)^n| = 2^{-n}$ (valime lihtsalt suuruse x juhuslikult!). Kujutusi f, mille korral $p_f = 2^{-n}$, nimetatake *täiuslikult mittelineaarseteks* (*perfect nonlinear mapping*). On teada järgmine tulemus, mida me siinkohal ei tõesta:

Teoreem 4.7.1 *Olgu* $f : \mathrm{GF}(2)^m \to \mathrm{GF}(2)^n$ *— täiuslikult mittelineaarne kujutus. Siis* $2 \mid m$ *ning* $m \geq 2n$ *[Nyb91].*

Juhul kui DESi-sarnase plokkšifri raundifunktsioon ei sisalda laienduskujutust (muuhulgas ka siis kui f on permutatsioon), on triviaalseks alamtõkkeks $p_f \geq 2^{1-n}$, kuna sellisel juhul omandab erinevus $f(x+w) + f(x)$, $w \neq 0$, täpselt pooled väärtustest $\in \mathrm{GF}(2)^n$ kaks korda ning ülejäänusid mitte kordagi. Permutatsioone f, mille korral $p_f = 2^{1-n}$, nimetatakse *kvaasitäiuslikult mittelineaarseteks*. Edaspidi tegelemegi kvaasitäielikult mittelineaarsete permutatsioonide otsimisega, kuna sel juhul, erinevalt juhust $m \geq 2n$, on tõkestatud juba kolmeraundiste diferentsiaalide tõenäosus.

Lemma 4.7.1 *Olgu* $f : \mathrm{GF}(2)^n \to \mathrm{GF}(2)^n$ *kvaasitäiuslikult mittelineaarne permutatsioon ning olgu* $\mathcal{E}$ *neljaraundine DES-i sarnane šiffer raundifunktsiooniga* f*. Siis on vaja edukaks erinevustega krüptoanalüüsi soorituseks šifri* $\mathcal{E}$ *vastu teostada vähemalt* 2^{n-2} *šifriga* $\mathcal{E}$ *krüptimist.*

Tõestus. Teoreemist 4.6.3 saame, et šifri $\mathcal{E}$ kõigi 3-raundiste diferentsiaalide tõenäosus on $\leq 2p_f^2 = 2^{3-2n}$. Lemmast 4.6.1 saame siis omakorda, et neljaraundise DESi-sarnase šifri erinevustega krüptoanalüüsiks on vaja teha vähemalt

$$\frac{2}{2^{3-2n} - \frac{1}{2^n - 1}} \approx 2^{2n-2}$$

šifriga $\mathcal{E}$ krüptimist. $\square$

Seega, konstrueerides kvaasitäiuslikult mittelineaarse permutatsiooni, saame tõepoolest šifri, mille (tõsi küll, mitte täiuslikku) turvalisust DC vastu me oskame tõestada. Järgnevas tegelemegi sellise permutatsiooni konstrueerimisega, kasutades seekord laialdaselt lineaaralgebra vahendeid.

4.7.2 Ruutvormid

Definitsioon 4.7.2 Vaatleme hulka $\mathrm{GF}(p)^n$ vektorruumina üle korpuse $\mathrm{GF}(p)$ ja fikseerime mingi baasi e. Seega $x = (x_1, \ldots, x_n) \in \mathrm{GF}(p)^n$. Olgu $f : \mathrm{GF}(p)^n \to \mathrm{GF}(p)$,

$$f(x) = f(x_1, x_2, \ldots, x_n) = \sum_{i,j} a_{ij} x_i x_j = x^t A x$$

n-muutuja ruutvorm üle lõpliku korpuse $\mathrm{GF}(p)$ (baasi e suhtes), kus $A = (a_{ij})$. Juhul kui korpuse karakteristik $\neq 2$ eeldatakse, et A on sümmeetriline maatriks.

Kaks ruutvormi $f(x) = x^t A x$ ja $g(x) = x^t B x$ on *sarnased*, kui nad esitavad sama ruutvormi, st kui leidub selline lineaarne permutatsioon (baasiteisendus, üleminekumaatriks) C, nii et $A = C^t B C$, ehk

$$g(Cx) = (Cx)^t B (Cx) = x^t (C^t B C) x = f(x) \ .$$

Paneme tähele et iga ruutvormi f korral $f(0) = 0$.

Definitsioon 4.7.3 Olgu $f : \mathrm{GF}(p)^n \to \mathrm{GF}(p)$. Ütleme, et vektor $w \in \mathrm{GF}(p)^n$ on kujutuse f *lineaarne struktuur*, kui $\forall x, y \in \mathrm{GF}(p)^n$,

$$f(x+w) - f(x) = f(y+w) - f(y) = f(w) \ .$$

Lemma 4.7.2 *Olgu $f : \mathrm{GF}(p)^n \to \mathrm{GF}(p) : x \mapsto x^t A x$, n-muutuja ruutvorm. Ruutvormi f lineaarsed struktuurid moodustavad $\mathcal{X}(f)$-mõõtmelise lineaarse alamruumi, kus $\mathcal{X}(f) := n - \mathrm{rank}(A + A^t)$.*

Tõestus. Paneme tähele, et

$$\begin{aligned} f(x+w) - f(x) &= (x+w)^t A (x+w) - x^t A x \\ &= x^t A w + w^t A x + w^t A w = x^t (A + A^t) w + f(w) \ . \end{aligned}$$

Seega w on f-i lineaarne struktuur $\iff$ $f(x+w) - f(x)$ on konstant $\iff$ $x^t (A + A^t) w$ on konstant $\iff$ $(A + A^t) w = 0$. □

Märgime, et kujul $w^t f(x)$, kus $w \in \mathrm{GF}(q)^m$, esituvad parajasti kõik vektorkujutuse f komponentide lineaarsed kombinatsioonid. Vaatleme edasises juhtu, kus põhikorpuseks on $\mathrm{GF}(2^d)$ (seega korpus karakteristikuga kaks). Sellise korpuse korral defineeritakse nii mõnedki mõisteid teistmoodi kui siis kui karakteristik ei ole kaks (üks näide on järgnevalt toodud mittekiduvuse definitsioon).

Definitsioon 4.7.4 Kujutus $f : \mathrm{GF}(2^m) \to \mathrm{GF}(2^d)^n$ on *mittekidunud*, kui f ei ole sarnane ühelegi sellisele ruutvormile, mis sõltub vähem kui n muutujast.

Ütleme, et permutatsioon $f : \mathrm{GF}(2^m) \to \mathrm{GF}(2^d)^n$ on *hea*, kui iga funktsiooni f komponentide lineaarkombinatsioon $w^t f$, $w \neq 0$, on mittekidunud n-muutuja ruutvorm $x^t Ax$ üle korpuse $\mathrm{GF}(2^d)$, kus $\mathrm{rank}(A + A^t) = n - 1$.

Antud paragrahvi põhitulemuse jaoks peame eelnevalt ära tõestama järgmised kaks lemmat.

Lemma 4.7.3 *Olgu $f : \mathrm{GF}(2)^n \to \mathrm{GF}(2)$, $f(x) = x^t Ax$, selline ruutvorm, et $2 \not| n$ ning $\mathrm{rank}(A + A^t) = n - 1$. Siis f on mittekidunud parajasti siis kui $f(w) \neq 0$ suvalise f-i lineaarse struktuuri $w \neq 0$ korral.*

Tõestus. Algebrast on teada, et suvaline ruutvorm, mille korral $\mathrm{rank}(A + A^t) = n - 1$, on sarnane ruutvormile kujul

$$\varphi(x_1, \ldots, x_n) = x_1 x_2 + \ldots + x_{n-2} x_{n-1} + \delta x_n^2 \ ,$$

kus $\delta \in \{0, 1\}$. Seega leidub selline koordinaatide lineaarne teisendus T, et $f(x) = \varphi(Tx)$. Nüüd, w on kujutuse f lineaarne struktuur $\iff$ kui $f(x + w) - f(w) = \varphi(Tx + Tw) - \varphi(Tw)$ on konstantne kujutus. Olgu $Tx = (X_1, \ldots, X_n)$ ning $Tw = (W_1, \ldots, W_n)$. Kuna

$$\begin{aligned}
\varphi(Tx + Tw) - \varphi(Tw) &= \sum_{i=1}^{n-2} (X_i + W_i)(X_{i+1} + W_{i+1}) + \delta (X_n + W_n)^2 - \\
&\quad \sum_{i=1}^{n-2} X_i X_{i+1} - \delta X_n^2 \\
&= \sum_{i=1}^{n-2} (X_{i+1} W_i + X_i W_{i+1} + W_i W_{i+1}) + \delta (2 X_n W_n + W_n^2) \ ,
\end{aligned}$$

siis $\varphi(Tx + Tw) - \varphi(Tw)$ on konstantne $\iff$

$$\sum_{i=1}^{n-2} (X_{i+1} W_i + X_i W_{i+1}) = W_2 X_1 + \sum_{i=2}^{n-2} (W_{i-1} + W_{i+1}) X_i + W_{n-2} X_{n-1}$$

(kasutasime mh seda, et $2X_nW_n = 0$, kuna korpuse karakteristik on 2) on konstantne $\iff$ $W_1 = W_2 = \ldots = W_{n-1} = 0$ (siin läks meil vaja seda, et $2 \not| n!$) $\iff$ $Tw = (0,\ldots,0,a)^t$ mingi $a \in \mathrm{GF}(2^d)$ korral.

Samas, f on mittekidunud $\iff$ (kuna T on regulaarne) φ on mittekidunud $\iff$ $\delta = 1$ (see fakt on ära tõestatud näiteks raamatus [LN97]). Paneme tähele, et kujutuse f suvalise lineaarse struktuuri w korral

$$f(w) = \varphi(Tw) = \varphi(0,\ldots,0,a) = \delta a^2 \ ,$$

Seega $\delta = 1 \iff f(w) \neq 0$ suvalise lineaarse struktuuri $w \neq 0$ korral. □

Lemma 4.7.4 *Olgu $f : \mathrm{GF}(2^d)^n \to \mathrm{GF}(2^d)^n$ hea permutatsioon. Siin on suvaline $w \in \mathrm{GF}(2^d)^n$ mingi $u \in \mathrm{GF}(2^d)^n \setminus 0$ jaoks kujutuse $u^t f$ lineaarne struktuur.*

Tõestus. Kuna f on hea permutatsioon siis järeldub lemmast 4.7.2 iga $u \neq 0$ jaoks, et ruutvormi $u^t f$ lineaarsed struktuurid moodustavad ühemõõtmelise alamruumi V_u vektorruumis $\mathrm{GF}(2^d)^n$ ning seega $|V_u| = |\mathrm{GF}(2^d)|$. Oletame esmalt, et kui $u_1 \neq cu_2$ mingi $c \in \mathrm{GF}(2^d)$ korral, siis $V_{u_1} \cap V_{u_2} = \emptyset$. Vektorruumis $\mathrm{GF}(2^d)^n$ on 2^{dn-d} vektorit u_1, u_2, $\ldots$, $u_{2^{dn-d}} \neq 0$ nii et $u_i \neq cu_j$ iga $c \in \mathrm{GF}(2^d)$ korral (fikseerime vektori u_1 esimese koordinaadi $u_{11} \neq 0$. Valime ülejäänud $2^{dn-d} - 1$ vektorit hulgast $u_{11} \times \mathrm{GF}(2^d)^n$. On selge, et sellisel juhul $\forall c$, $u_i neq cu_2$ ja samas kõik vektorid hulgast $\mathrm{GF}(2^d)^n \setminus \{u_1,\ldots,u_{2^{dn-n}}\}$ on esituvad kujul cu_i mingi i korral), ning seega ka $|\bigcup_{i=1}^{2^{dn-d}} V_{u_i}| = \sum_{i=1}^{2^{dn-d}} |V_{u_i}| = 2^{dn-d}|\mathrm{GF}(2^d)| = |\mathrm{GF}(2^d)^n|$ ning järelikult on *iga* $w \in \mathrm{GF}(2^d)^n$ ühes hulkadest V_{u_i}.

Tõestame nüüd, et kui $u_1 \neq cu_2$, $\forall c$, siis $V_{u_1} \cap V_{u_2} = \emptyset$. Olgu siis $u_1, u_2 \neq 0$ sellised vektorid, et mingi $w \neq 0$ korral on nii $u_1^t f$ kui ka $u_2^t f$ lineaarne struktuur ja seega on w ka ruutvormi $(c_1u_1 + c_2u_2)^t f$, kus $c_1, c_2 \in \mathrm{GF}(2^d)$, lineaarne struktuur. Kuna f ei ole kidunud ning $u_1, u_2 \neq 0$, siis ka $u_1^t f$ ja $u_2^t f$ on mittekidunud. Nüüd, $u_1^t f(w) \neq 0$ ning $u_2^t f(w) \neq 0$ ning järelikult leidub selline $c \neq 0$, et $(cu_1)^t f(w) = u_2^t f(w)$ ehk $(cu_1 + u_2)^t f(w) = 0$. Kui $cu_1 \neq u_2$ siis on lemma 4.7.3 põhjal $(cu_1 + u_2)^t f$ kiduv, mis on vastuolus faktiga et nii $u_1^t f$ kui ka $u_2^t f$ on mittekiduvad. Seega $cu_1 = u_2$ ning järelikult u_1 ja u_2 on lineaarselt sõltuvad. □

Teoreem 4.7.2 *Olgu $f : \mathrm{GF}(2^d)^n \to \mathrm{GF}(2^d)^n : x \mapsto x^t Ax$, $2 \not| n$, hea permutatsioon. Siis $p_f = 2^{d(1-n)}$.*

Tõestus. Tõestame, et kui $f = (f_1,\ldots,f_n) : \mathrm{GF}(2^d)^n \to \mathrm{GF}(2^d)^n$ on hea permutatsioon, siis iga $w \neq 0$ ja a korral on võrrandil $f(x+w) + f(x) = a$ kas 0 või 2^d

lahendit. (Selle fakti tõestusest piisaks teoreemi tõestuseks, kuna siis iga a korral

$$\mathrm{Prob}_x[f(x+w)+f(x)=a] \leq \frac{2^d}{|\mathrm{GF}(2^d)^n|} = 2^{d(1-n)} \quad .)$$

Olgu $w \neq 0$. Siis lemmast 4.7.4 saame, et w on mingi $v \neq 0$ jaoks ruutvormi $v^t f$ lineaarne struktuur. Kuna hea permutatsiooni definitsiooni kohaselt $v^t f$ on mittekidunud, siis saame lemmast 4.7.3, et

$$v^t f(x+w) + v^t f(x) = v^t f(w) \neq 0 \quad ,$$

suvalise $x \in \mathrm{GF}(2^d)^n$ korral. Tähistame $b_0 := v^t f(w)$.

Olgu nüüd $u_1, \ldots, u_{n-1} \in \mathrm{GF}(2^d)^n$ sellised lineaarselt sõltumatud vektorid, et $v \notin \mathrm{span}(u_1, \ldots, u_{n-1})$, kus $\mathrm{span}(\{v_i\})$ tähistab nagu alati vektorihulga $\{v_i\}$ lineaarset katet. Olgu $u \in \mathrm{span}(u_1, \ldots, u_{n-1}) \setminus 0$ (seega muuhulgas $u \neq cv$, $\forall c$) ning olgu g järgmine kujutus:

$$g : x \mapsto u^t f(x+w) + u^t f(x) \quad .$$

Paneme tähele, et suvalise ruutvormi h, $h(x) = x^t Bx$, korral ja iga fikseeritud w jaoks on kujutus

$$h'(x) := h(x+w) + h(x) = x^t (B + B^t) w + w^t Bw$$

kas afiinne või konstantne. Juhul kui w ei ole kujutuse h lineaarne struktuur, siis on h' afiinne, kuna aga iga afiinne kujutus on balanseeritud siis ka h' on balanseeritud. (Tuletame meelde, et kujutus h on balanseeritud, kui $\forall u \in \mathrm{GF}(2^d)$, $\mathrm{Prob}_{x \in \mathrm{GF}(2^d)^n}[h(x) = u] = \frac{1}{|\mathrm{GF}(2^d)|}$.)

Kuna w on ruutvormi $v^t f$ lineaarne struktuur ning $u \neq cv$, $\forall c$, siis lemma 4.7.4 tõestusest saame, et w ei ole ruutvormi $u^t f$ lineaarne struktuur. Seega on g afiinne ning järelikult ka balanseeritud. Nüüd, kuna u on hulga $\mathrm{span}(u_1, \ldots, u_{n-1}) \setminus 0$ suvaline element, siis on ka kujutused $u_i^t f(x+w) + u_i^t f(x)$ balanseeritud ning järelikult on võrrandisüsteemil

$$u_i^t f(x+w) + u_i^t f(x) = b_i \quad i \in \{1, \ldots, n-1\}$$

iga $(b_1, \ldots, b_{n-1})^t \in \mathrm{GF}(2^d)^{n-1}$ korral 2^d erinevat lahendit $x \in \mathrm{GF}(2^d)^n$. Seega on täiendatud võrrandisüsteemil

$$\begin{aligned} &v^t f(x+w) + v^t f(x) = b \\ &u_i^t f(x+w) + u_i^t f(x) = b_i \qquad\qquad i \in \{1, \ldots, n-1\} \end{aligned}$$

2^d erinevat lahendit kui $b = b_0$ ning 0 lahendit kui $b \neq b_0$. Nüüd, suvaline võrrandisüsteem

$$f_i(x+w)+f_i(x) = a_i, \quad i \in \{1,\dots,n\}$$

(ehk suvaline võrrand $f(x+w)+f(x) = a$) on eelmise võrrandisüsteemi lineaarteisendus ning järelikult on sellel võrrandil sõltuvalt a valikust kas 0 või 2^d erinevat lahendit $x \in \mathrm{GF}(2^d)^n$. $\square$

Eelmisest teoreemist saame, et kui permutatsioon f on "hea", n on paaritu ning $d = 1$, siis on f ka kvaasitäiuslikult mittelineaarne. Esineb permutatsioone, mille korral on lihtsam tõestada, et nad on "head" kui otseselt, et nad on kvaasitäiuslikult mittelineaarsed. Samas on ka permutatsioone, mille korral on "lihtne" otseselt tõestada nende kvaasitäiuslikust. Demonstreerime seda järgnevalt.

Teoreem 4.7.3 *Olgu n paaritu, ja olgu $f : \mathrm{GF}(2^n) \to \mathrm{GF}(2^n) : x \mapsto x^{2^k+1}$ ning olgu $s = \gcd(k,n)$. Siis $p_f \leq 2^{s-n}$. Muuhulgas, kui $\gcd(k,n) = 1$ siis f on kvaasitäiuslikult mittelineaarne permutatsioon.*

Tõestus. Olgu $w \in \mathrm{GF}(2^n) \setminus 0$ ja $a \in \mathrm{GF}(2^n)$. Meil on vaja ülalt tõkestada polünoomi

$$(x+w)^{2^k+1} + x^{2^k+1} = a \tag{4.10}$$

lahendite hulka: näidata, et lahendeid on $\leq 2^s$, sõltumata w ja a valikust. Teisendame alguses võrrandit (4.10), kasutades seda, et korpuse karakteristik on 2 (sellest lähtuvalt ka lemma 4.5.2 tulemust):

$$\begin{aligned}(x+w)^{2^k+1} + x^{2^k+1} &= (x^{2^k} + w^{2^k})(x+w) + x^{2^k+1} = \\ &= wx^{2^k} + w^{2^k}x + w^{2^k+1} \quad .\end{aligned}$$

Kui lahendeid on vähem kui 2 (sellisel juhul pole võrrandil ühtegi lahendit), siis oleme teoreemi tõestanud. Olgu nüüd x_1 ja x_2 kaks erinevat lahendit. Siis kehtib

$$w(x_1+x_2)^{2^k} + w^{2^k}(x_1+x_2)x = 0 \quad ,$$

ehk

$$(x_1+x_2)^{2^k-1} = w^{2^k-1} \quad . \tag{4.11}$$

Olgu g multiplikatiivse rühma $\mathrm{GF}(2^n)^*$ moodustaja (selline on olemas iga lõpliku korpuse multiplikatiivsel rühmal) ning olgu $w = g^j$ ning $x_1 + x_2 = g^i$. Võrrandi (4.11) lahendite arvu leidmine on samaväärne järelikult võrrandi

$$g^{(2^k-1)x} = 1$$

(siin tähistasime $x = i - j$) lahendite arvu leidmisega. Viimane võrrand toimib rühmas järguga $2^n - 1$, ning seega on see samaväärne võrrandi

$$(2^k - 1)x = 0 \mod 2^n - 1$$

lahendite arvu leidmisega. Viimaseid on aga $\gcd(2^k - 1, 2^n - 1) = \gcd(2^s - 1)$ tükki. Seega, kuna võrrandil (4.11 on $2^s - 1$ lahendit, leidub võrrandi (4.10) fikseeritud lahendi x_1 korral veel $2^s - 1$ erivat selle võrrandi lahendit. Seega on lahendite arv kokku 2^s. □

Toodud teoreemis kasutasime järgmist tulemust arvuteooriast.

Lemma 4.7.5 *Olgu* $a > 1$*, ning* $s = \gcd(k, n)$*. Siis* $\gcd(a^k - 1, a^n - 1) = a^s - 1$.

Tõestus. On lihtne näha, et $a^s - 1 \mid \gcd(a^k - 1, a^n - 1)$, kuna juhul kui $k = sd$, siis $a^k - 1 = (a^s - 1)\sum_{i=0}^{k} x^{si}$ ning analoogiliselt $a^s - 1 \mid a^n - 1$. Tõestamaks, et $\gcd(a^k - 1, a^n - 1) \mid a^s - 1$ tuletame elementaarsest arvuteooriast meelde, et kui $s = \gcd(k, n)$ siis mingite täisarvude a ja b korral $s = bk + cn$, kusjuures kas $b > 0$ ja $c \leq 0$ või $c > 0$ ja $b \leq 0$. Oletame üldsust kitsendamata, et $b > 0$. Olgu $m = \gcd(a^k - 1, a^n - 1)$, mis on samaväärne sellega, et $a^k \equiv 1 \pmod{m}$ ning $a^n \equiv 1 \pmod{m}$. Võtame neist esimese kongruentsi astmesse b, saame, et $a^{bk} \equiv 1 \pmod{m}$. Võttes teise kongruentsi astmesse $-c$, saame, et $a^{-cn} \equiv 1 \pmod{m}$, millest saame kokku, et $a^{bk+cn} \equiv 1 \pmod{m}$ ehk $a^s \equiv 1 \pmod{m}$ ning seega

$$m = \gcd(a^k - 1, a^n - 1) \mid a^s - 1 \ .$$

4.7.3 Kokkuvõte

Kokkuvõttes oleme saanud, et sõltumatute raundivõtmetega DESi-sarnane šiffer, kus raundifunktsiooniks on (näiteks) $f : \mathrm{GF}(2^{33}) \to \mathrm{GF}(2^{33}) : x \mapsto x^3$, on peaaegu täiuslikult turvaline erinevustega krüptoanalüüsi vastu. Kuna 33 on ebamugav arv, soovitasid Nyberg ja Knudsen [NK95] kasutada mingit afiinset laienduskujutust $E : \mathrm{GF}(^{)}2^{32} \to \mathrm{GF}(2^{33})$ enne kuupfunktsiooni f ning pärast kujutuse f rakendamist üks 33st koordinaadist ära keelata. Saab tõestada, et ka saadud uue raundifunktsiooni F korral p_F on väike. Kuna aga F pole enam permutatsioon, tuleb sama turvalisuse tagamiseks kasutada vähemalt nelja raundi, ideaalse turvalisuse jaoks vähemalt kuut raundi, millisel juhul toodud šiffer vajab 198-bitist võtit. Seega on saadud šifril juba üks väga halb omadus — liiga pikk võti. Lisaks sellele on ka näidatud (näiteks artiklis [JK97]), et kirjeldatud šiffer (tuntud kui Nyberg-Knudseni šiffer, $\mathcal{NK}$) on lahti murtav erinevustega krüptoanalüüsi üldistusi kasutades. Seega, kehtib juba eelnevas rõhutatud tõde et plokkšifrite disainimisel ei tohi ainuüksi lähtuda turvalisusest mingi konkreetse ründe vastu, plokkšiffer peab olema turvaline kõigi mõeldavate rünnete vastu — ideaalsel juhul on meil aga tegu juhusliku permutatsiooniga.

Peatükk 5

Informatsiooniteoreetiline turvalisus

5.1 Täiuslik turvalisus

Enamuses selles krüptoloogiakursuses tegeleme me arvutusliku turvalisusega, kus kasutajate ressursid on piiratud (tuletage meelde pseudojuhuslikke funktsioone). Käesolevas peatükis vaatleme lähemalt aga informatsiooniteoreetilist turvalisust, ehk turvalisust ründaja vastu, kelle ressursid *ei ole* piiratud (tuletagem meelde, et informatsiooniteoreetiliselt turvalised on juhuslikud funktsioonid). Kuigi informatsiooniteoreetilisel turvalisusel ei ole väga palju praktilisi rakendusi, on tegemist ajalooliselt esimese “range” krüptograafia haruga [Sha49], ning peale selle võimaldab informatsiooniteoreetiline turvalisus tõestada ennekõike just negatiivseid tulemusi: näiteks, et “ideaalset” (ehk täiuslikku) turvalisust on praktikas võimatu saavutada piiratud võtmepikkuse korral. Samas, juba kõige lihtsamad šifrid — sh ka asendusšiffer ning ühekordne numbrijada — on informatsiooniteoreetiliselt turvalised, kui me iga võtmega krüptime vaid ühe korra. Järgnevas me eeldamegi, et iga võtit kasutatakse krüptimiseks vaid ühe korra ning vaatame, mida saab sellisel juhul šifri turvalisuse kohta tõestada.

Toome esmalt mõned elementaarsed mõisted tõenäosusteooriast. Olgu X ja Y juhuslikud suurused, mõlema suuruse jaoks on fikseeritud oma tõenäosusjaotus. Nagu tõenäosusteoorias tavaks, tähistame $p(x) := \mathrm{Prob}_X[X = x]$ ning $p(y) := \mathrm{Prob}_Y[Y = y]$. Sündmuste X ja Y ühistõenäosus (ehk nende sündmuste korrutise tõenäosus) $p(x,y) := \mathrm{Prob}_{X,Y}[X = x \wedge Y = y]$. Tähistame lühidalt ka järgmisi tinglikke tõenäosusi $p(x|y) := \mathrm{Prob}_{X,Y}[X = x|Y = y]$ ja $p(y|x) := \mathrm{Prob}_{X,Y}[Y = y|X = x]$.

Tuletame meelde, et juhuslikud suurused X ja Y on *sõltumatud*, kui $p(x,y) =$

$p(x)p(y)$ suvalise paari (x,y) korral. Tõenäosusteooriast on teada, et

$$p(x,y) = p(x|y)p(y) = p(y|x)p(x) \quad ,$$

mistõttu X ja Y on sõltumatud parajasti siis kui $p(x|y) = p(y)$ (ehk samaväärselt $p(y|x) = p(x)$) suvalise paari (x,y) korral.

Eeltoodust tuleneb automaatselt Bayes'i teoreem:

Teoreem 5.1.1 (Bayes'i teoreem) *Kui* $p(y) > 0$*, siis*

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \quad .$$

Fikseerime nüüd mingi konkreetse šifri $\mathcal{E}$ (mis ei pruugi olla permutatsioon, kuid peab olema injektsioon), nii et $\mathcal{E} : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$, kus $\mathcal{K}$ on võtmete hulk, $\mathcal{P}$ avatekstide hulk ja $\mathcal{C}$ krüptogrammide hulk. Seostame hulgaga $\mathcal{P}$ mingi tõenäosusjaotuse (edasises eeldame, et antud jaotus on ründajale teada); tähistagu $p_{\mathcal{P}}(x)$ tõenäosust, et selle jaotuse järgi valitakse x. Lisaks eeldame, et võti K on valitud kõigi võtmete hulgast $\mathcal{K} = \{0,1\}^k$, kasutades mingit (potentsiaalselt) erinevat tõenäosusjaotust; tähistagu $p_{\mathcal{K}}(k)$ tõenäosust, et valiti võti k (ka see tõenäosus on ründajale teada: väga paljudes rakendustes kasutatakse avalikult teadaolevaid algoritme juhuarvude genereerimiseks). Me eeldame, et võti valiti enne avateksti krüptimist; kuna iga võtmega krüptitakse vaid üks avatekst, võime eeldada, et võti ja avatekst on sõltumatud muutujad.

Tõenäosusjaotused avatekstide hulgal ja võtmete hulgal indutseerivad jaotuse krüptogrammide hulgal; tähistame tõenäosust et krüptogramm on y järgmiselt: $p_{\mathcal{C}}(y)$. Olgu $C(k) := \{\mathcal{E}_k(x) : x \in \mathcal{P}\}$ (me ei eelda, et $\mathcal{E}$ on sürjektsioon!). Siis iga $y \in \mathcal{C}$ jaoks

$$p_{\mathcal{C}}(y) = \sum_{\{k: y \in C(k)\}} p_{\mathcal{K}}(k) p_{\mathcal{P}}(\mathcal{E}_k^{-1}(y)) \quad .$$

Lihtsalt saab leida ka tingliku tõenäosuse, et krüptogramm on y kui avatekst on x:

$$p_{\mathcal{C}}(y|x) = \sum_{\{k: x = \mathcal{E}_k^{-1}(y)\}} p_{\mathcal{K}}(k) \quad .$$

Kasutades Bayes'i teoreemi, saame välja arvutada tingliku tõenäosuse $p(x|y)$ (ehk tõenäosuse, et avatekst oli x kui krüptogramm on y):

$$p_{\mathcal{P}}(x|y) = \frac{p_{\mathcal{P}}(x) \sum_{\{k: x = \mathcal{E}_k^{-1}(y)\}} p_{\mathcal{K}}(k)}{\sum_{\{k: y \in C(k)\}} p_{\mathcal{K}}(k) p_{\mathcal{P}}(\mathcal{E}_k^{-1}(y))} \quad .$$

Seega jagatise $p_{\mathcal{P}}(x|y)/p_{\mathcal{P}}(x)$ arvutamiseks on vaid vaja teada jaotusi $p_{\mathcal{P}}$ ning $p_{\mathcal{K}}$, mis on aga eelduse tõttu avalikult teada.

Definitsioon 5.1.1 Šiffer $\mathcal{E}$ on täiuslikult turvaline, kui $p_{\mathcal{P}}(x|y) = p_{\mathcal{P}}(x)$ kõigi $x \in \mathcal{P}$ ja $y \in \mathcal{C}$ jaoks. See on samaväärne väitega, et suuruse y teadmine ei aita (ka teoreetiliselt) kaasa suuruse x teadmisele (ehk juhuslikud suurused X ja Y on sõltumatud).

Näitame järgnevalt, et täiuslikult turvalised krüptosüsteemid on olemas.

Teoreem 5.1.2 *Fikseerime mingi lõpliku (multiplikatiivse) rühma $(G,\cdot)$. Olgu $\mathcal{E}$ järgmine šiffer: $\mathcal{E} : G\times G \to G$, $\mathcal{E}(k,x) := k\cdot x$. Eeldame, et võti k valitakse ühtlase jaotuse järgi ($p_{\mathcal{K}}(k) = |G|^{-1}$, $\forall k$). Siis iga avatekstide jaotuse $p_{\mathcal{P}}$ korral on šiffer $\mathcal{E}$ täiuslikult turvaline.*

Tõestus. Märgime, et siin $|\mathcal{K}| = |\mathcal{P}| = |\mathcal{C}| = |G|$ ning seetõttu on $\mathcal{E}$ pealekujutus nii fikseeritud $k \in \mathcal{K}$ kui ka fikseeritud $x \in \mathcal{P}$ jaoks. Seega $C(k) = \mathcal{C}$, iga k jaoks, ning $\{k : y \in C(k)\} = \mathcal{K}$. Arvutame välja tõenäosused:

$$p_{\mathcal{C}}(y) = \sum_{k\in\mathcal{K}} p_{\mathcal{K}}(k)p_{\mathcal{P}}(k^{-1}y) = \sum_{k\in\mathcal{K}} |G|^{-1}p_{\mathcal{P}}(k^{-1}y) = |G|^{-1}\sum_{k\in\mathcal{K}} p_{\mathcal{P}}(k^{-1}y) \quad .$$

Kuna $\mathcal{E}(k,\cdot)$ on permutatsioon kõigi $k \in \mathcal{K}$ jaoks, siis

$$p_{\mathcal{C}}(y) = |G|^{-1}\sum_{k\in\mathcal{K}} p_{\mathcal{P}}(y) = |G|^{-1} \quad .$$

Samas kuna iga x,y korral leidub parajasti üks k, nii et $x = k^{-1}y$, siis

$$\begin{aligned} p_{\mathcal{C}}(y|x) = & \sum_{\{k:x=\mathcal{E}_k^{-1}(y)\}} p_{\mathcal{K}}(k) = \sum_{\{k:x=\mathcal{E}_k^{-1}(y)\}} |G|^{-1} \\ = & |G|^{-1}|\{k : x = \mathcal{E}_k^{-1}(y)\}| = |G|^{-1} \end{aligned}$$

ning järelikult $p_{\mathcal{C}}(y) = p_{\mathcal{C}}C(y|x)$, $\forall x$. □

See teoreem katab endaga mitut tuntud šifrit. Juhul $G = \mathbb{Z}_2$, on tegu *Vernami šifriga* ehk *ühekordse numbrijadaga*, kus iga saadetav bitt $b \in \{0,1\}$ krüptitakse juhuslikult valitud võtmebitiga $k \in \{0,1\}$ järgmiselt: $c_i = k_i \oplus p_i$. Samasse valdkonda käib ka asendusšiffer, kus rühmaks on $(\mathbb{Z}_n,+)$.

Järgmise teoreemi jaoks eeldame, et iga y korral $p_{\mathcal{C}}(y) > 0$ (vastasel korral võime elemendi y hulgast $\mathcal{C}$ lihtsalt välja visata). Fikseerime mingi $x \in \mathcal{P}$. Kuna iga $y \in \mathcal{C}$ jaoks $p_{\mathcal{C}}(y) > 0$, siis juhul kui $\mathcal{E}$ on täiuslikult turvaline, siis ka

$p_{\mathcal{C}}(y|x) = p_{\mathcal{C}}(y) > 0$ ning seega leidub vähemalt üks võti k, nii et $\mathcal{E}_k(x) = y$. Seega $|\mathcal{K}| \geq |\mathcal{C}|$. Iga šifri jaoks peab kehtima, et $|\mathcal{C}| \geq |\mathcal{P}|$, kuna šifrid on injektsioonid. Piirjuhul $|\mathcal{P}| = |\mathcal{K}| = |\mathcal{C}|$ saab anda järgmise tingimuse selle kohta, millal täielik turvalisus kehtib.

Teoreem 5.1.3 (Claude Shannon, 1949) *Olgu $|\mathcal{P}| = |\mathcal{K}| = |\mathcal{C}| = n$. Siis šiffer $\mathcal{E} : \mathcal{K} \times \mathcal{P} \to \mathcal{C}$ on täiuslikult turvaline avatekstide suvalise jaotuse korral parajasti siis, kui*

1. *$\forall k$, $p_{\mathcal{K}}(k) = 1/n$ (st et võtit k valitakse ühtlase jaotuse järgi).*
2. *iga x ja y korral leidub parajasti üks k, nii et $\mathcal{E}(k,x) = y$.*

Tõestus. $\Leftarrow$. Olgu $\mathcal{E}$ täiuslikult turvaline, st $p_{\mathcal{C}}(x|y) = p_{\mathcal{C}}(x)$ ja $p_{\mathcal{C}}(y|x) = p_{\mathcal{C}}(y)$, $\forall x, y$. Eelnevast arutelust teame, et iga $x \in \mathcal{P}$ ja $y \in \mathcal{C}$ korral leidub vähemalt üks võti k, nii et $\mathcal{E}_k(x) = y$. Seega

$$|\mathcal{C}| = |\{\mathcal{E}_k(x) : k \in \mathcal{K}\}| \leq |\mathcal{K}| \ ,$$

aga samas $|\mathcal{C}| = |\mathcal{K}|$ ning seega $|\{\mathcal{E}_k(x) : k \in \mathcal{K}\}| = |\mathcal{K}|$. Järelikult ei leidu kahte erinevat võtit k_1 ja k_2, nii et $\mathcal{E}(k_1,x) = \mathcal{E}(k_2,x) = y$. Seega olema tõestanud teise väite. Esimese väite tõestamiseks fikseerime ühe suvalise elemendi $y \in \mathcal{C}$ ning nummerdame hulgad $\mathcal{K} = \{k_1, \ldots, k_n\}$ ja $\mathcal{P} = \{x_1, \ldots, x_n\}$ nii, et $\mathcal{E}(k_i, x_i) = y$, $\forall i$ (selline numeratsioon leidub ja on ühene tänu väitele 2). Bayes'i teoreemist saame, et

$$p_{\mathcal{P}}(x_i|y) = \frac{p_{\mathcal{C}}(y|x_i)p_{\mathcal{P}}(x_i)}{p_{\mathcal{C}}(y)} = \frac{p_{\mathcal{K}}(k_i)p_{\mathcal{P}}(x_i)}{p_{\mathcal{C}}(y)} \ .$$

Kuna $\mathcal{E}$ on täiuslikult turvaline, siis $p_{\mathcal{P}}(x_i|y) = p_{\mathcal{P}}(x_i)$ ning seega $p_{\mathcal{K}}(k_i) = p_{\mathcal{C}}(y)$, iga $1 \leq i \leq n$ jaoks, millest järeldub, et kõiki võtmeid kasutatakse sama tõenäosusega $p_{\mathcal{C}}(y)$, ning kuna võtmeid on $|\mathcal{K}| = n$ tükki, on selleks tõenäosuseks $1/n$.

$\Rightarrow$. Analoogne teoreemi 5.1.2 tõestusega. □

Märgime, et praktikas tähendab nõue $|\mathcal{K}| = |\mathcal{P}|$ seda, et iga järgnevalt saadetava avateksti x korral tuleb genereerida uus võti k pikkusega $\log_2 |\mathcal{P}|$, ning see ka kuidagi edasi toimetada. Kui seda ei tehta, langevad paljud täiuslikult turvalised šifrid väga kergesti valitud avatekstiga ründe all. Näiteks kui $\mathcal{E}(k,x) = k \cdot x$ rühmatehte $\cdot$ korral, siis valides avateksti x ja nähes krüptogrammi y saab ründaja siit tuletada võtme $k = y \cdot x^{-1}$ ning sellega lahti krüptida suvalisi järgnevaid sama võtmega krüptitud krüptogramme.

Käesolev peatükk on lihtne ning oleks loogiliselt pidanud asume õppematerjali päris alguses.

Peatükk 6

Jadašifrid

Peatükk 7

Diskreetsel logaritmil põhinevad krüptosüsteemid

7.1 Sissejuhatus

Kursuse jooksul oleme siiani vaadelnud sümmeetrilisi krüptosüsteeme (sh PRFd, PRPd, plokkšifrid, jadašifrid, ühekordne numbrijada, ...), mille puhul suhtluspartnerid on enne suhtluse algust vahetanud salajase võtme, mis on teada *täpselt neile osapooltele, kes selle võtmega krüpteeritud krüptogramme tohivad lugeda*. Peale selle, et võtmeid peab eelnevalt vahetama (kui on näiteks n osapoolt, kes kõik soovivad omavahel paarikaupa turvaliselt suhelda, on seega vaja vahetada $\frac{n}{2}$ võtit), on sümmeetrilistel krüptosüsteemidel ka see halb omadus, et nendega saab tagada vaid konfidentsiaalsust, mitte aga teisi võimalikke turvaeesmärke. Saab küll tagada osalist autentsust, kasudades nn MACe (*message authentication code*), kuid sellistes süsteemides peab dokumendi signatuuri kontrollija teadma ka salajast signeerimisvõtit — mis tähendab muuhulgas, et verifitseerija võib signatuuri sama hästi ka ise moodustada. Väga paljudes situatsioonides (näiteks digitaalallkirja seaduse kontekstis) on vaja aga üheselt tuvastada, kes antud dokumendi signeeris.

Nii autentsust kui ka väga paljusid muid turvaeesmärke rahuldavad avaliku võtmega krüptograafia vahendid, kus iga osapool genereerib endale avaliku ja salajase võtme. Väga üldiselt, avalik võti publitseeritakse avalikus kataloogis (kasutades PKI-d, aga ka näiteks oma veebilehel), salajast võtit teab aga vaid kasutaja ise. Nüüd, kui Alice soovib edastada teadet M, teeb ta päringu avalikku kataloogi ning omandab sealt Bobi avaliku võtme P_B. Seejärel krüptib ta teate M omandatud avaliku võtmega ning saadab krüptogrammi C Bobile. Bob dekrüptib saadud teade, kasutades oma salajast võtit S_A. Analoogiliselt käib ka signeerimine: Alice signeerib teate M oma salajase võtmega S_A ning saadab signatuuri C Bobile. Nüüd teeb Bob päringu avalikku kataloogi, kust ta saab kätte Alice'i avaliku võtme P_A ning seejärel verifitseerib signatuuri, kasutades saadud võtit. Tegelikus elus on need protseduurid muidugi märksa keerulisemad, ning edasises tegelemegi nende spetsifitseerimisega.

Märgime sissejuhatuse lõpetuseks, et kui sümmeetriline on olnud inimkonnale teada aastatuhandeid (räägitakse, et seda kasutanud nii spartalased Kreekas kui ka Julius Caesar isiklikult), siis avaliku võtmega krüptograafia leiutasid Whitfield Diffie ja Martin Hellman aastal 1976, seega siis lähiminevikus. Üks põhjusi miks see nii on, on avaliku võtmega krüptograafia taga seisneva matemaatika keerukus: kui ka nihkešiffer (sisuliselt modulaarne liitmine!) on sümmeetrilise krüptograafia näide, siis avaliku võtmega krüptograafia baseerub (näiteks) diskreetse logaritmi ja arvude tegurdamise raskusel. Teine põhjus, miks avaliku võtmega krüptograafia alles nii hiljuti avastati, on ka nõudluse järsk kasv: aastasadu kasutati krüptograafiat põhiliselt sõjaväelistel otstarvetel, ning sõjaväes on fikseeritud hierarhia ning ka eelteadmus, kes kellega suhtleb: sellises süsteemis on võtmehaldus palju lihtsam.

Seitsmekümnendateks oli aga juba arvutustehnika arenenud piisavalt kaugele ning saavutanud piisava populaarsuse selleks, et äri hakati ajama elektroonilisi kanaleid kasutades inimeste ja firmade vahel, kes teineteist kunagi enne näinud ei olnud. Võib aga väita ka vastupidist: avaliku võtmega krüptograafia lõi eelduse (tehnoloogilise baasi) nn e-äri laialdaseks levikuks. Enne avaliku võtmega krüptograafia avastamist ei olnud olemas võimalust digitaaldokumente signeerida, ei olnud digitaalset sularaha ega elektroonilisi valimisi jne.

7.2 Formaalne definitsioon

Järgnevas eeldame, et tudengitel on olemas baasteadmus keerukusteooriast. Nagu ka keerukusteoorias, parametriseeritakse ka avaliku võtmega krüptograafias algoritmide keerukust mingi parameetri järgi — milleks on tavaliselt otseselt algoritmide sisendi(te) pikkus. Me eeldame, et selleks parameetriks on algoritmide esimese sisendina antud turvaparameeter 1^k, mille suhtes väljendatakse nii "positiivsete" (krüptimiseks kasutatavate) kui ka "negatiivsete" (lahtimurdmiseks kasutatavate) algoritmide keerukust.

Paneme tähele seda, et erinevalt sümmeetrilisest krüptograafiast on avaliku võtmega krüptograafia parametriseeritav: parameetrit k suurendades saadakse sama krüptosüsteem, millega krüptimiseks ja mille lahtimurdmiseks kuluvad ajad suurenevad. Eriti heal juhul on krüptosüsteem esitatav paari ilusa matemaatilise valemi abil, mida on võimalik rakendada erinevate sisendipikkuste korral. Järgnevas eeldamegi, et krüptosüsteemides kasutatakse uniformseid algoritme, mis lõpetavad oma töö polünomiaalses ajas oma esimese sisendi 1^k pikkusest, väljastades (olenevalt oma funktsioonile) uue võtme pikkusega k või näiteks verifitseerivad pikkusega k võtmega signeeritud teateid.

Definitsioon 7.2.1 *Avaliku võtmega krüptosüsteem* on kolmik (G,E,D) algoritme, mis rahuldavad järgmisi tingimusi

1. G on *võtmegenereerimisalgoritm*: randomiseeritud polünomiaalses ajas töötav algoritm G, mis sisendil 1^k (turvaparameeter) väljastab paari (e,d), kus e on avalik ning d salajane võti. Tähistame seda $(e,d) \stackrel{R}{\leftarrow} G(1^k)$. Paari (e,d) nimetatakse ka võtmepaariks.

2. Krüptimisalgoritm E: polünomiaalses ajas töötav randomiseeritud algoritm E, mis saab sisendiks turvaparameetri 1^k, avaliku võtme e, mis kuulub $G(1^k)$ kujutisse, ning teate $M \in \{0,1\}^k$. E väljastab krüptogrammi $c \in \{0,1\}^k$. Tähistame $c \stackrel{R}{\leftarrow} E(1^k,e,M)$.

3. Dekrüptimisalgoritm E: polünomiaalses ajas töötav deterministlik algoritm D, mis saab sisendiks turvaparameetri 1^k, salajase võtme d, mis kuulub $G(1^k)$ kujutisse, ning krüptogrammi $c \in \{0,1\}^k$. E väljastab teate $M' \in \{0,1\}^k$, nii et iga $(e,d) \stackrel{R}{\leftarrow} G^k$ korral $D(1^k,d,E(1^k,e,m)) = m$. Tähistame $M' \stackrel{R}{\leftarrow} D(1^k,d,c)$.

Krüptosüsteemilt nõutakse harilikult veel mingis hästi defineeritud mõistes turvaline olemist. Täpse turvalisuse mõiste toome hiljem, mitteformaalselt ei tohi kasutaja kes ei tea salajast võtit d saada lugeda selle võtmega krüptitud teateid.

Praktikas elus kasutatakse avaliku võtme krüptosüsteeme järgmiselt:

1. Kasutaja Bob valib endale sobiva turvaparameetri 1^k (soovi korral võib neid mitmeid olla), ning genereerib võtmepaari $(e,d) \stackrel{R}{\leftarrow} G(1^k)$. Seejärel edastab Bob paari (k,e) avalikku kataloogi.

2. Kui Alice soovib teadet M Bobile saata, küsib ta kataloogist Bobi võtme (k,e). Juhul kui k on Alice'i meelest liiga väike, teeb ta midagi mittekrüptograafilist

(näiteks palub Bobil suurem võti genereerida). Juhul kui k on paras, arvutab Alice suuruse $c \stackrel{R}{\leftarrow} E(1^k, e, M)$ ning edastab selle Bobile.

3. Bob dekrüptib saadud suuruse: $M' = D(1^k, d, c)$. Vastavalt avaliku võtmega krüptosüsteemi definitsioonile $M = M'$.

7.3 ElGamal

Järgnevas vaatleme lähemalt diskreetsel logaritmil põhinevaid krüptosüsteeme. Põhjused nende krüptosüsteemide eelistamiseks RSA-le on lihtne: a) Willemson juba vaatles RSA-d põhjalikult b) diskreetsel logaritmil põhinevad krüptosüsteemid töötavad üle paljude erinevate rühmade ning seega saab juhul kui mingi rühm halvaks osutub sama süsteemi mujale üle kanda c) on olemas formaalsed tõestused ElGamali ja seotud krüptosüsteemide turvalisuse kohta nn geneerilises mudelis.

Definitsioon 7.3.1 Olgu G suvaline rühm algarvulise elementide arvuga, olgu $g \in G_q \setminus \{1\}$. Elemendi $h \in G$ diskreetseks logaritmiks alusel g nimetatakse üheselt määratud arvu a, $0 \leq a \leq |G| - 1$, nii et $g^a = h$. Tähistame edasise $a = \log_g h$. Diskreetse logaritmi probleemiks nimetatakse ülesannet leida $\log_g h$ etteantud g ning juhuslikult valitud $h \stackrel{R}{\leftarrow} G$ korral. Rühmade pere $\mathcal{G}$ on algarvulise elementide arvuga rühmade jada $(G_1, \dots)$. Edasises eeldame, et rühmaoperatsioon, rühma elemendi juhuslik valik ning rühmaelementide võrdlus on efektiivselt teostatavad (parameetri 1^k suhtes) operatsioonid. Eeldame et rühmade pere iga liikme jaoks on fikseeritud ka moodustaja g_k, mille suhtes diskreetset logaritmi võetakse.

Definitsioon 7.3.2 ElGamali krüptosüsteem on avaliku võtmega krüptosüsteem (G, E, D), kus esmalt fikseeritakse rühmade pere ning siis (tähistame lihtsuse mõttes $g := g_k$)

1. $G(1^k)$ leiab juhusliku arvu $2 \leq x \leq |G_k| - 1$, ning tagastab paari $(h = g^x, x)$.

2. $E(1^k, e, M)$ genereerib juhusliku arvu r, ning tagastab paari (Mg^r, h^r).

3. $D(1^k, x, y_1, y_2)$ leiab suuruse $M' = y_1^x / y_2$.

On lihtne näha, et $D(1^k, x, E(1^k, g^x, M)) = D(1^k, x, Mg^r, g^{rx}) = M$ ning seega on ElGamali krüptosüsteem korrektselt defineeritud.

7.4 Ründed ElGamali krüptosüsteemi vastu

On selge, et kui ründaja oskab diskreetset logaritmi leida, siis suudab ta lahti murda ka ElGamali krüptosüsteemi. Tõepoolest, teadku ründaja arvupaari (Mg^r, g^{rx}). Kuna ta oskab diskreetset logaritmi leida, siis suudab ta arvust g^{rx} leida arvu g^r. Kuna rühma pöördelemendi leidmine on kerge, leiab nüüd ründaja teate M väga lihtsasti.

Register

Kirjandus

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phil Rogaway. A concrete security treatment of symmetric encryption: Analysis of the des modes of operation. In *Proceedings of 38th Annual Symposium on Foundations of Computer Science*, 1997.

[BKR98] Mihir Bellare, Ted Krovetz, and Phillip Rogaway. Luby-Rackoff Backwards: Increasing Security by Making Block Ciphers Non-Invertible. In Kaisa Nyberg, editor, *Advances on Cryptology — EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 266–280, Helsinki, Finland, June 1998. Springer-Verlag.

[Dav91] D. W. Davies, editor. *Advances on Cryptology — EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, Brighton, UK, April 1991. Springer-Verlag.

[JK97] Thomas Jakobsen and Lars Knudsen. The Interpolation Attack on Block Ciphers. In Eli Biham, editor, *Fast Software Encryption '97*, volume 1267 of *Lecture Notes in Computer Science*, pages 28–40, Haifa, Israel, January 1997. Springer-Verlag.

[LMM91] Xuejia Lai, James L. Massey, and Sean Murphy. Markov ciphers and differential cryptanalysis. In Davies [Dav91], pages 17–38.

[LN97] Rudolf Lidl and Harald Niederreiter. *Finite Fields (Encyclopedia of Mathematics and Its Applications, Vol 20).* Cambridge University Press, 2 edition, 1997.

[NK95] Kaisa Nyberg and Lars Knudsen. Provable Security Against A Differential Attack. *Journal of Cryptology*, 8(1):27–37, 1995.

[Nyb91] Kaisa Nyberg. Perfect Nonlinear S-Boxes. In Davies [Dav91], pages 378–386.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:657–715, 1949.

[Sti95] Douglas Stinson. *Cryptography Theory and Practice*. CRC Press, 1995.