Vello Hanson, Ahto Buldas, Helger Lippmaa,

Arne Ansper, Viljar Tulit

# Infosüsteemide turve II

# **Turbetehnoloogia**

Cybernetica AS

Tallinn 1998

Infosüsteemide turve:

I Turvarisk (1997 Vello Hanson; 2009 Vello Hanson, Märt Laur, Kristjan Alliksoo, Monika Oit)

II Turbetehnoloogia (1998 Vello Hanson, Ahto Buldas, Helger Lippmaa, Arne Ansper, Viljar Tulit)

## Teise osa saateks

Turberaamatu teine osa tutvustab turvamehhanismide teoreetilisi ja tehnilisi aluseid. Pääslaväravast kvantkrüptograafiani – sellise ulatusega ülevaadet ei ole maailma turvakirjanduses seni pakkunud ükski raamat. Kõige selle mahutamiseks tuli muuta raamatu ülesehitust. Algselt kavandatud organisatsiooniliste meetmete peatükk jäi välja ning laieneb raamatu neljandaks osaks pealkirjaga "Turbe korraldus". Niisiis puudutab käesolev osa peamiselt infotehnilisi mehhanisme; erandiks on viimane peatükk, mis annab lühiülevaate peamistest füüsilistest turvameetmetest.

Meie maa hakkab väljuma infoturbe lastehaiguste staadiumist. Meie ajalehed ekspluateerivad küll mõnuga andmebaaside vargusi, pangakaartide võltsimisi ja muid turvaintsidente, märkamatuks jääb aga turvateadlikkuse ja -teadmuse tunduv kasv paaril viimasel aastal. Aktiivselt tegutseb andmekaitse inspektsioon, revideerimisel on andmekaitseseaduste turvaaugud, Küberneetika AS initsiatiivil koostatud digitaalsignatuuri seaduseelnõuga on alustatud elektroonilise asjaajamise õiguslikku tagamist – niisiis alustatud turvameetmetest. Nende ridade kirjutamise ajal naasis Kesk-Euroopast Küberneetika AS kõrgtehnoloogiliste turvatoodete välisturgu laiendamast meie firma turundusjuht. USAs on menetluses Küberneetika AS töötajate (A. Buldas, P. Laud, H. Lipmaa, J. Villemson) krüptograafilise ajatemplisüsteemi patenditaotlus. Meil on edumaa oma postsotsialistlike (ja mitte ainult) naabrite ees. Eestil on eeldusi saada kõrge infoturbekultuuriga maaks ja me loodame sellele kaasa aidata oma raamatuga.

Teine osa sisaldab suhteliselt püsivat alusteavet ning on pigem õpik kui käsiraamat. Kui ei ole öeldud teisiti, on firma- ja tootenimed ning tehnilised andmed mõeldud ainult hetkenäidetena. Sihilikult on välditud selliseid tehnilisi üksikasju ja andmeid, mis võiksid tekitada ebatervet huvi turvamehhanismide vastu. Lugejalt eeldatakse vähemalt keskharidust, krütograafiapeatükkide mõistmine nõuab põhjalikumat matemaatika tundmist.

***Autorid***

## Tänuavaldused

Autorite tänu on pälvinud Alar Leibak, kes abistas 9. peatüki kirjutamisel, ja Helen Oit, kes tegi rohkesti jooniseid, tabeleid ja muud vormistuslikku.

# SISUKORD

# 6 TURVAMEETMED

Varade kaitseks rakendatavad turvameetmed võivad mõjutada turvaülesande kõiki komponente:

- ohte (mitmeid sisemisi ohuallikaid saab kõrvaldada),
- turvaauke (peaaegu kõiki turvaauke saab sulgeda või kahandada),
- toimet varadele (näiteks võib loobuda kulukatest ja vähetulusatest varadest),
- oodatavat kahju (näiteks saab riski "edasi müüa" kindlustusseltsile),
- teisi turvameetmeid (neid tugevdades või nõrgendades).

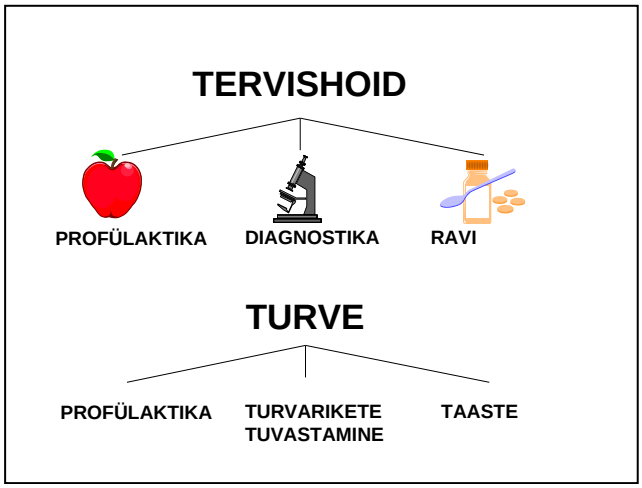
## 6.1 Turvameetmete funktsioonid

Tüüpilistes etalonturbe metoodikate turvameetmete spetsifikatsioonides loetletakse (sõltuvalt detailsustasemest) kümneid või sadu kohustuslikke meetmeid, individuaallahenduste korral võib nende arv olla veelgi suurem. Meetmete otstarbekaks valimiseks, objekti kaitsetaseme hindamiseks ja eri turvalahenduste võrdlemiseks tuleks turvameetmeid vaadelda süstemaatiliselt.

Lähtuda tuleks turbeabinõude põhiatribuudist – temalt saadavast turvateenusest, st turbeotstarbest, mida ta täidab. Esmase jämeda liigituse saamiseks võib infosüsteemi turvalisust võrrelda tervisega, turvarikkeid haigustega. Haiguste tõrjes võib eristada kolme faasi – profülaktikat, diagnoosimist ja ravi. Analoogiliselt jagunevad turvameetmete funktsioonid kolmeks põhirühmaks:

- profülaktika,
- turvarikete tuvastamine,
- infoobjekti turvalisuse ja rikke-eelse oleku taaste.

Paljud turvameetmed on polüfunktsionaalsed. Näiteks veaparanduskoodid täidavad nii tuvastuse kui ka taaste ülesandeid.




**Joonis 1. Turve kui infoobjekti tervishoid**

### 6.1.1 Profülaktika

Profülaktilised meetmed moodustavad turbearsenali suurima osa. Nad võimaldavad sulgeda turvaauke, ära hoida ründeid, vähendada ohtude realiseerumise tõenäosust, kahandada turvarikete toimet infovaradele ja hõlbustada objekti taastet.

#### *6.1.1.1 Tugevdusmeetmed*

Need on peamiselt organisatsioonilised abinõud kaitstava objekti kõige levinumate, peamiselt stiihilistel ohtudel toimimist võimaldavate turvaaukude sulgemiseks või kahandamiseks.

Kindel ***kord***, süstemaatilisus, kindlaksmääratud protseduurid igapäevases tööelus on peamine vahend stiihiliste ohtude tõrjeks. Kõik asutuse töö, sealhulgas infotehnoloogia sujuvat kulgu soodustavad abinõud tõstavad ka turvalisust: sisekorra eeskirjad, täpsed ametijuhendid, standardite järgimine, infrastruktuuri hooldus, kindlaksmääratud hankeprotseduurid, töövahendite dokumenteerimine, andmekandjate ja kaabelduse märgistus, versioonihaldus, ressursivarude käigushoid jne. Alles seejärel lisanduvad turvaspetsiifilised meetmed, alates üldisest turvapoliitikast ja turvaplaanist ning lõpetades üksikobjekte puudutavate konkreetsete turvajuhistega.

Kõigi töötajate ***turvateadlikkus*** ja -motivatsioon on tähtis tugevdav tegur. Infoturbe aluste tundmine on tänapäeval niisama vajalik kui infotehnoloogia enda tundmine. See saavutatakse töötajate sobiva valimise, regulaarse koolituse, teavitusürituste, auditeerimiste, proovihäirete ja muude turbehalduslike meetmetega. Turvaprobleemi ja esmaste turvameetmete tutvustus peaks sisalduma igas arvutiõppe programmis.

Normaalsed ***töötingimused*** tõstavad eelkõige süsteemide käideldavust ja terviklust. Sobiv mikrokliima (temperatuur, õhuniiskus, õhu puhtus) kahandab nii tehniliste komponentide tõrkeid kui ka personali vigu ja tõstab tööviljakust. Inimeste puhul on niisama tõhus ka töökoha ergonoomiline ehitus ja kujundus. Veelgi olulisem on asutuse sotsiaalne kliima; positiivsed inimsuhted, objektiivne edutamis- ja ergutuspoliitika loovad aluse töötajate turvamotivatsioonile ning vähendavad siserünnete tõenäosust.

Süstemaatiline ***kontroll*** võimaldab õigel ajal avastada seni märkamatuks jäänud või uusi turvaauke. Infotehniliste toodete ja turvamehhanismide verifitseerimine ja testimine toob defektid nähtavale. Regulaarne turbealase operatiivteabe jälgimine (näiteks vastavate Interneti uudisegruppide lugemine) võimaldab kõrvaldada või kompenseerida mujal avastatud defekte. Turvamehhanismide usaldatavuse kontrolliks võib proovida neist läbi murda või mööduda. Süsteemide auditeerimine standardmetoodikate alusel annab kompleksse pildi kogu tegelikust turvatasemest.

#### *6.1.1.2 Peletusmeetmed*

Hoiatava loomuga abinõud kahandavad rünnete üritamise tõenäosust. Peletav toime on üldiselt enamiku turvameetmete kasulik lisaomadus; ainuüksi teadmine turvameetmete käigushoiust või nende tajumine vähendab ründeindu, eriti kui oodatav saak ei korva ründaja riski. Mida laiem on mingi turvameetme mõju, seda suurem on ta peletav toime. Kui on teada, et asutuse turvapoliitika on formuleeritud, et ta on range ja et teda järgitakse rangelt, ei paku konkreetsed turvamehhanismid mõnelegi potentsiaalsele ründajale enam huvi.

Kõige laiema ja tugevama peletustoimega on rünnete kohta kehtestatud ***sanktsioonid***, mis on sätestatud mitmel tasemel, alates andmekaitseseadustest ja muudest asjakohastest õigusaktidest ning lõpetades asutusesiseste distsiplinaarmeetmetega.

Tahtmatult sooritatud turvarikkeid väldib ***hoiatav märgistus*** dokumentidel, andmekandjatel, kuvadel, ruumide ustel jne. Turvaintsidentide uurimisel välistab see katsed esitada sihilikku rünnet eksitusena. Märgistust rakendatakse peamiselt konfidentsiaalsuse kaitseks. Mitme konfidentsiaalsusastme puhul kasutatakse astmete eristamiseks lisaks erinevale hoiatustekstile ("Salajane", "Ainult tööalaseks kasutamiseks" jne) enamasti ka erinevaid standardseid tähistusvärvusi.

Peletavat toimet avaldavad paljud ***nähtavad turvameetmed*** – valvur, telekaamera, territooriumi valgustatus, turvauksed, kaartjuhtimisega lukusüsteemid jne.

#### *6.1.1.3 Tõkestus- ja eraldusmeetmed*

Subjektide (inimeste või protsesside) juurdepääsu varadele kitsendavad meetmed tõrjuvad peamiselt ründeid ning kaitsevad turvalisuse kõiki põhiaspekte (käideldavust, terviklust, konfidentsiaalsust).

***Ruumiline*** isoleerimine on lihtsaim ja levinuim eraldamisprintsiip. Erineva tundlikkusastmega andmeid võib töödelda mitmel eraldi arvutil, igal neist oma kasutaja(d) ja eri rangusega turvameetmed. Need arvutid võivad ka füüsiliselt paikneda üksteisest lahus, kaitsetasemelt erinevates ruumides. Andmekandjate puhul tähendab ruumiline eraldamine, et ühel andmekandjal (näiteks disketil) asuvad ainult võrdse tundlikkusega või sama(de)le kasutaja(te)le määratud andmed; tundlikkuselt erinevaid andmekandjaid võidakse säilitada eri kohtades ja erinevatel tingimustel. Sidekanalite puhul tähendab ruumiline isoleerimine eraldi füüsiliste sideliinide sisseseadmist erineva tundlikkusega teabe edastuseks.

***Ajaline isoleerimine*** on rakendusvõimalustelt piiratum ja seda kasutatakse nappide ressursside korral. Näiteks võidakse mingil arvutil töödelda kuus tundi päevas ühe tundlikkusklassi andmeid, kaks tundi aga töötleb teine kasutaja teistsuguse klassi andmeid; sellisel juhul kõrvaldab kumbki kasutaja töö lõpetamisel kõvakettalt oma andmekogumid ja vajaduse korral ka neid töötleva rakendusprogrammi. Igapäevasem näide on ühe ja sama ruumi kasutamine eri aegadel erineva tundlikkusastmega üritusteks.

***Loogiline isoleerimine*** hõlmab suurimat infotehniliste meetodite ja vahendite kogu. Rakendusvõimalused on peaaegu piiramatud ning objektide kasutamist saab reguleerida väga diferentseeritult, eriti kui seda rakendada koos objektide ***granuleerimisega*** (näiteks andmete tükeldamisega piisavalt väikesteks elementideks, mida saab eraldi töödelda või suvaliselt rühmitada). Meetmed jagunevad kolme suurde rühma:

- pääsu reguleerimine,
- teenusevahendus,
- salastamine.

***Pääsu reguleerimine*** (*access control*) tähendab objektide (andmeüksuste, seadmete, ruumide jne) kasutamise valikulist võimaldamist subjektidele (isikutele või neid esindavatele protsessidele). Levinuimad näited on kaartjuhtimisega ukselukusüsteem ja paroolkaitsega sisselogimissüsteem. Detailsemalt vaadeldakse pääsu reguleerimist 7. peatükis.

***Teenusevahendus*** on olemuselt sarnane pääsu reguleerimisele, kuid erinevalt sellest ei kasuta üldjuhul subjekti identsuse kontrolli (autentimist), vaid põhineb lubamatute teenuste ja operatsioonide tuvastusel nende blokeerimiseks. Pääsu reguleerimisega võrreldes rakendatakse märksa keerukamat eraldusloogikat. Näiteid: kohtvõrku avalikust kaugvõrgust isoleeriv ja lubatavaid võrguteenuseid vahendav tulemüür, ühiskasutusfailide terviklust säilitav sünkroniseerimis- ja lukustusmehhanism, andmebaasi terviklust ja konfidentsiaalsust kaitsev päringuprotsessor, programmeerijat masinakäskudest isoleeriv süsteemiarhitektuur.

***Salastamine*** kaitseb peamiselt konfidentsiaalsust, teatud määral ka terviklust. Peamised salastusviisid on teabe

- krüpteerimine,
- peitmine,
- hävitamine.

***Krüpteerimist*** vaadeldakse lähemalt peatükkides 9–13. Rakendusnäited ulatuvad väikeste andmeüksuste (näiteks paroolifailide) salastamisest kuni krüptomüüride tehnikani, mis võimaldab avalikus laivõrgus luua turvalisi sisevõrke. Krüpteerimise üks alaliike on telefonside salastuseks kasutatav skrambleerimine.

***Peitmine*** infotehnilises mõttes tähendab näiteks steganograafiliste meetodite rakendamist (vt 14. ptk), andmeliikluse täidistamist (sõnumite täiendamist pseudoteabega nende pikkuse võrdsustamiseks, pseudosõnumite saatmist) liiklusvoogude analüüsi tõrjeks, subjekti jaoks keelatud andmeüksuste (või operatsioonide) nimede esitamata jätmist liidestuskuvadel jne. Infrastruktuuri puhul tähendab peitmine näiteks tundlike ruumide tähistamata jätmist uksesiltidel ja viitadel, nähtava kaabelduse vältimist jms.

***Hävitamine*** tähendab eelkõige aegunud või muul põhjusel tarbetuks muutunud, kuid ründajaile huvipakkuva teabe usaldusväärset kõrvaldamist ja on konfidentsiaalsust kaitsev vahend. Näiteid: faili kustutamine kettalt andmejälgi jätmata, paberdokumendi purustamine paberihundis. Hävitamise spetsiifilisemaid näiteid on kiipide või muude konfidentsiaalset teavet sisaldavate komponentide purustamise vahendid, mis käivituvad komponendi hermeetilisuse rikkumisel.

### 6.1.2 Turvarikete tuvastamine

Objekti sajaprotsendiline turve ei ole võimalik. Maksimaalne võimalik turve ei ole enamasti majanduslikult õigustatud (vt 4. ptk). Optimaalse turbe korral jääb aktsepteeritud jääkriski tõttu alati

turvarikete võimalus, kuid need rikked ei tohi jääda märkamatuks. Turvarikkest tekkiva kahju minimeerimise seisukohalt on turvameetmete pingerida selline: *rikke vältimine – rikke kohene tuvastamine – rikke kohene registreerimine ja hilisem tuvastamine – rikke tõestamine hiljem*. Kui rikke vältimine osutub põhjendamatult kulukaks, tuleb tagada vähemalt mingil tasemel tuvastus.

#### *6.1.2.1 Operatiivtuvastus*

See töötermin tähistab meetmeid, mis võimaldavad turvaintsidente tuvastada kohe nende tekkimisel ja neile kohe reageerida. Selliste vahendite hulka kuuluvad paljud infrastruktuuri üldturbe abinõud: valvur, sisetelevisioon, tuletõrje- ja valvesignalisatsiooni süsteemid, keskkonnaseire süsteemid jms. Infotehnikas on enamasti kombineeritud pääsu reguleerimisega ja muude blokeerimismehhanismidega; turvarikke katse signaliseerimise kõige tavalisem näide on keelatud operatsiooni blokeerimisele kaasnev vea- või hoiatusteade. Tarkvara väljatöötajatele pakuvad operatiivtuvastuse funktsiooni mitmesugused silumisvahendid.

#### *6.1.2.2 Järeltuvastus*

Otseselt või kaudselt turvariketega seotud sündmusi registreerivad paljud infotehnilised süsteemid; levinuimad näited on arvutite ja lukusüsteemide logifailid. Registreeritavate sündmuste suurest arvust ja automaatanalüüsi puudumisest tingituna ei teavita need vahendid sündmustest kohe, kuid võimaldavad regulaarsete või intsidendijärgsete läbivaatuste käigus analüüsida turvarikkeid või rikkekatseid. Üks süsteemiülemate igapäevaseid töövahendeid. Järeltuvastuse instrumentide hulka kuuluvad ka mitmesugused diagnostika- ja testimisvahendid ning läbivaatuse, verifitseerimise ja auditeerimise meetodid.

#### *6.1.2.3 Tõendtuvastus*

Eelmistes alajaotistes loetletud vahendid kasutavad turvarikete tuvastuseks objektide, protsesside ja sündmuste loomulikke atribuute. Paljudel juhtudel ei piisa neist aga turvarikete rangeks ja täpseks tõendamiseks, muuhulgas näiteks juriidiliste menetluste tarbeks või turvarikke automaatseks kõrvaldamiseks. Sellistel juhtudel varustatakse andmekogumid mitmesuguste terviklust ja/või konfidentsiaalsust kontrollida võimaldavate turvaelementidega.

Andmekogumi sisu terviklust võimaldavad kontrollida mitmesugused kogumis sisalduvatest andmetest moodustatud kontrollkoodid ja -kogumid: paarsusbitid, kontrollsummad, tsükkelkoodid, krüptograafilised sõnumilühendid (*message digest*). Selline kontrollkomponent luuakse andmekogumi genereerimisel, salvestamisel või saatmisel, seejärel aga veel kord andmekogumi lugemisel või vastuvõtul; kui saadud tulemus erineb algsest kontrollkomponendist, on andmete sisu terviklus rikutud.

Andmeallika ja edastusprotsessi terviklust saab kontrollida digitaalsignatuuri ning digitaalsete ajatemplite abil. Need vahendid võimaldavad autentida suhtluse osapooli ja edastatavaid sõnumeid, pakkudes kaitset näiteks teeskluse, saatmise salgamise, sõnumi taasesituse jt sedalaadi rünnete eest.

Andmete konfidentsiaalsuse tagamiseks kasutatavad tõendmärgised võivad olla füüsilised (nähtavad või vähemärgatavad turvakiled, -niidid, -pitserid, värvust muutvad märgised jms) või infotehnilised (näiteks steganograafiline eksemplarimärgistus, vt 14. ptk).

Infomaterjali legaalsust aitavad tagada piraatkopeerimise tõendamiseks määratud nähtavad/kuuldavad või steganograafilised (vt 14. ptk) vesimärgid (*watermark*) ja "sõrmejäljed" (*fingerprint*), millega märgistatakse vastavalt originaal või (kopeerimisprotsessis, automaatselt) koopia. Steganograafilisi märke saab lisada paberdokumentidele, slaididele, raalgraafikale, tarkvarale, heli- ja videosalvestistele.

### 6.1.3 Taastemeetmed

Objekti turvalisust kahjustanud intsidendi järel tuleb taastada objekti normaalne talitlus seda kiiremini ja seda suuremas ulatuses, mida olulisem ja tundlikum on objekt.

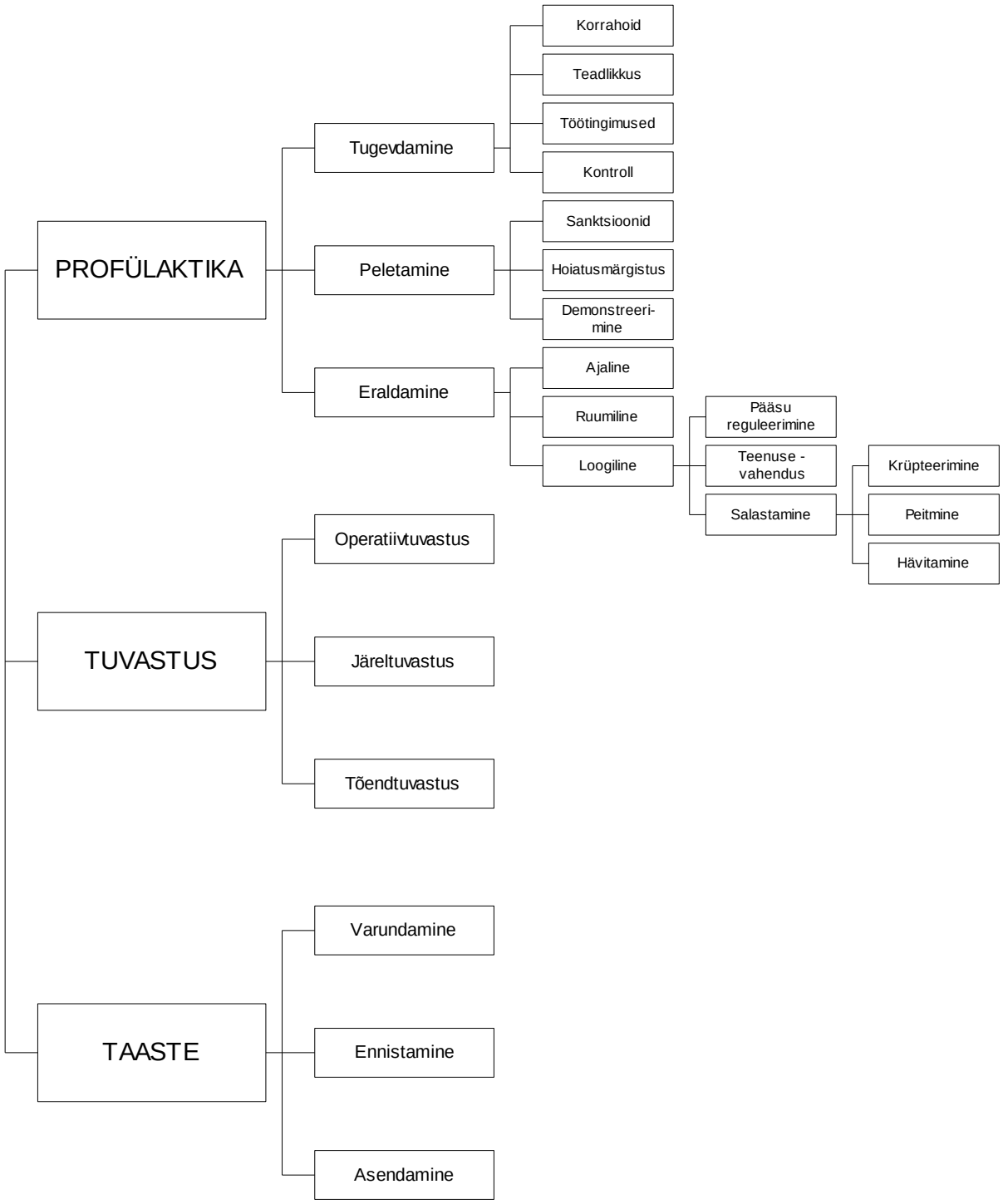
Taaste eelduseks on ***varundamine*** - varusüsteemide, -komponentide, -andmete, -protseduuride, -ruumide jne loomine/soetamine ning nende kasutuselevõtu tagamine nõutava ajaga. Lihtsaim näide on andmete regulaarne (tavaliselt mitte harvemini kui kord nädalas) varukopeerimine. Kõrgete käideldavusnõuete puhul kasutatakse automatiseeritud kuumvarundust, mis võimaldab varuressursi kasutusele võtta minimaalse seisuajaga; näiteid: paralleelselt töös hoitav arvutisüsteem, RAID-kettasüsteem.

Komponendi funktsionaalsuse ***ennistamine*** hõlmab rikete, tõrgete ja defektide kõrvaldamist: aparatuuri remonti, tarkvara parandamist ja modifitseerimist, sh versioonihalduse meetmeid rakendades ja valmistaja või tarnijaga konsulteerides, operatsioonide tagasivõttu rakendustes, infrastruktuuri remonti jms. Automaatse ennistuse vahendid on näiteks viirusetõrjeprogrammid ja andmevahetusel kasutatavad veaparanduskoodid.

Parandamatute kahjustuste puhuks peab olema ette valmistatud komponendi **asendamine**, sh aegsasti sõlmitud kiirtarne- või üürilepingute abil. Asendusplaanid peavad hõlmama ka töötajate võimalikke ootamatuid ajutisi väljalangemisi või alaliseks lahkumist.

### 6.1.4 Turbefunktsioonide kokkuvõte

Turvameetmete funktsioonide lühiülevaate annab Joonis 2.




**Joonis 2. Turvameetmete funktsioonid**

## 6.2 Turvameetmete teostus

Eelmises jaotises loetletud turbefunktsioone saab luua

- organisatsiooniliste,
- füüsiliste või
- infotehniliste

turvameetmetega või nende kombinatsioonidega.

Näiteid:

| *Funktsiooni tüüp:* | *Organisatsiooniliselt* | *Füüsiliselt* | *Infotehniliselt* |
|---|---|---|---|
| Eraldamine | Kohustuste lahusus | Lukustamine | Biomeetriline pääs |
| Tuvastus | Valvur | Valvesignalisatsioon | Sündmuste logimine |
| Taaste | Asendusost | Remont | Veaparanduskood |

**Organisatsiooniliste** meetmete hulka kuuluvad töökorralduse, turbesüsteemide kavandamise, plaanimise, halduse, turvaintsidentide käsitluse tegevused ja toimingud, sealhulgas üldise turvapoliitika ja turbeplaani koostamine, süsteemide ja turvamehhanismide turvapoliitikate formuleerimine ja järgimine, kaasa arvatud tehnilis-organisatsioonilised projekteerimis- ja häälestusotsused ning turvamehhanismide kasutamise protseduurid. Organisatsioonilist laadi abinõude valik on suurim, neid tuleb rakendada esmajärjekorras, alates turvapoliitika sõnastamisest, riskianalüüsist ja turbeplaani koostamisest. Enamasti on nad muude võimalustega võrreldes ökonoomsemad ja paljudel juhtudel ka tõhusamad (näiteks tulemusliku viirusetõrje sisseseadmisel). Ilma toetavate organisatsiooniliste meetmeteta ei toimi ka füüsilised ega infotehnilised vahendid. Seetõttu sõltub nõutava turvataseme saavutamine ja säilitamine eelkõige turbe organisatsioonilisest korraldusest. Organisatsioonilised meetmed põhinevad üldisel töökorraldusõpetusel (*management science*).

**Füüsilised** meetmed hõlmavad eelkõige objekti infrastruktuuri: ehituslikke piirdeid, kommunikatsioone, kütte- ja kliimaseadmeid, üldotstarbelist sisseseadet, turvauksi ja -aknaid, seife, barjääre, tõkkepuid, pöördväravaid jms ning ka mitmesuguseid mehaanilisi komponente (lukud, sildid, viidad, pakendid, märgised). Traditsiooniliselt loetakse infrastruktuuri osaks ja seetõttu füüsilisteks meetmeteks ka näiteks tuletõrje- ja valvesignalisatsiooni, sisetelevisiooni, automaatväravaid jm oma loomult infotehnilisi süsteeme, mis tänapäeval üha sagedamini integreeritakse arvutitega ja neil põhinevate turvamehhanismidega. On ilmne, et nende liigitamine ei ole puhtformaalne küsimus, seetõttu vajaksid praegu käibel olevad turbemetoodikad mõnevõrra korrigeerimist. Füüsilised meetmed põhinevad vastavatel tehnikadistsipliinidel.

**Infotehnilised** meetmed on kasutusel peamiselt loogilise eraldamise ja turvarikete tuvastuse funktsioonide teostamiseks. Nende suur mitmekesisus tuleneb kaitstava infotehnilise objekti konkreetsetest iseärasustest (eriti võrgukeskkonnas) ja eri funktsioonide kombineerimisest. Meetmete aluseks on mõned üldised infotehnilised ja turbedistsipliinid (formaalne spetsifitseerimine, autentimistehnika, krüptograafia, steganograafia jt), mida on kohati püütud viia turvatehnika (*security technology*, *security engineering*) koondnime alla. Niisuguseid üldisi teoreetilisi ja tehnilisi aluseid, mis võimaldavad luua uusi turvamehhanisme ja mõista käibelolevate turvatoodete talitlust, tutvustatakse detailsemalt järgmistes peatükkides.

## 6.3 Turvameetmete valimine

### 6.3.1 Üldprintsiipe

Turvameetmed valitakse lähtudes riskianalüüsiga väljaselgitatud kriitilistest ohtudest ja nõrkustest ning kahjustatavatest turvalisuse aspektidest (käideldavus, terviklus, konfidentsiaalsus).

Etalonturbe rakendamisel valitakse meetmed lihtsalt tüüpmoodulite turvaspetsifikatsioonide alusel vastavast kataloogist.

Individuaallahenduse korral on otstarbekas tükeldada organisatsioon erineva turvatarbega aladeks, kohaldades madala ja keskmise turvatarbe korral etalonlahendust. Sellised alad võivad kattuda organisatsiooni talitluslike allüksustega. Alade piiritlemisel tuleb arvestada järgmisi tegureid:

- kasutatava informatsiooni tundlikkusaste,
- sooritatavad operatsioonid ja infopääsu tüüp,
- kasutajaskond,
- seosed teiste alade ja keskkondadega.

Turbesüsteemi individuaallahenduse korral tuleb iga meetme valimisel lähtudes järgmistest asjaoludest:

- turvameetme funktsioonid (vt 6.1, üldiselt tuleks eelistada polüfunktsionaalseid),
- toime turvaaspektidele (käideldavus, terviklus, konfidentsiaalsus, vt 6.3.2)
- meetme suhteline tugevus (vt 6.3.2),
- läbipaistvus kasutajale, st võimalikult vähene häiriv toime,
- meetme kasutamise hõlpsus.

### 6.3.2 Turvameetmete tugevus ja toime turvaaspektidele

Meetmete tugevuse hindamiseks pole mingit objektiivset skaalat. Teatavaid orientiire võivad pakkuda mitmeastmelise turbe tunnustatud tüüplahendused. Ühe sellise lahenduse näide on esitatud tabelis 28 (andmed on pärit NASA infoturbe teatmikust). Sõltumata turvatasemete täpsest määratlusest annab tabel aluse turvameetmete suhtelise tugevuse jämedaks hindamiseks.

Enamasti tekitavad teatava turbevajaduse mitte kõik turvalisuse aspektid (käideldavus, terviklus, konfidentsiaalsus) võrdselt, vaid üks neist, konkreetses (alam)süsteemis kõige olulisem. Seetõttu tuleb turvameetmete valimisel arvestada ka nende aspektispetsiifikat. Teatava ettekujutuse meetmete toimest turvalisuse üksikkomponentidele annab Tabel 1 (andmed on pärit Kanada politsei turvastandardist).

**Tabel 1. Turvameetmete valimise kaalutlusi sõltuvalt nõutavast kaitsetasemest**

| Meetmetüüp | Tase 0 | Tase 1 | Tase 2 | Tase 3 |
|---|---|---|---|---|
| Pääsu reguleerimine | Kasutajate ühesed kasutajanimed ja paroolid.<br>Kõigi pääsujuhtude logimine. | Volitamist ja pääsu reguleerivad mehhanismid (org./füüsil./infotehnilised). | Iga kasutaja identimine ja autentimine.<br>Õiguste individuaalne kitsendamine.<br>Kasutajatevaheline ressursside kaitse.<br>Andmete krüpteerimine valikuliselt. | Pidev individuaalne pääsu reguleerimine ja logimine ressursside, rakenduste, failide haaval.<br>Volitused kinnitab ressursi omanik vähemalt kaks korda aastas.<br>Krüpteerimisest loobumine valikuliselt ja põhjendusega. |
| Võrgupääs | | Pääs parooliga.<br>Failiedastus veaavastuse/-parandusega | Valitud süsteemidel:<br>Enne iga siseneva ühenduse aktiveerimist kirjalik konkreetse välispöörduse allikat aktsepteeriv luba. | Kõigil süsteemidel:<br>Enne iga siseneva ühenduse aktiveerimist kirjalik konkreetse välispöörduse allikat aktsepteeriv luba. |
| Füüsiline pääs | Füüsiline turve äraoleku ajal. | Füüsiline kaitse volitamatu pääsu, varguse, hävimise eest. Aparatuurilukud. | | |
| Revisjonipäevik | | | Automaatne individuaalne süsteemi-, andme-, rakendusepöörduste logimine. | |
| Desaktiveerimine | | | Tööjaamade automaatne väljalogimine pikematel passiivsusperioodidel. | |
| Konfiguratsiooni-haldus | Kõigi failide kataloog.<br>Kogu kasutatava tarkvara litsentsid. | Haldusprotsess kõigi tundlike ja turbega seotud komponentide muutuste reguleerimiseks. | | Vahendid andmebaaside salvestuseks andmekandjatele. |
| Tarkvara varukoopiad | Rakendustarkvara vähemalt ühe versiooni varukoopia.<br>Muutunud andmefailide varukoopiad vähemalt kord kuus. | Vähemalt kahe versiooni varukoopiad.<br>Vanima versiooni ladustus süsteemi asukohast eemal. | | |
| Rakenduste turve | Süsteemifailide juhuläbivaatused etteteatamiseta. | | | Kõik süsteemi ressursid alati varustatud tundlikkustaseme märgistega. |
| Andmebaasiturve | | | Süsteemid kogu baasi sisu tervikluse, käideldavuse ja konfidentsiaalsuse kaitseks.<br>Baasihaldur peab omanikku teavitama kõigist pääsuõigustest ja andmete kasutamisest. | |
| Andmekandjate turve | Sobivad ladustuskarbid | | | |

| | | | | |
|---|---|---|---|---|
| Side turve | Sidekanalite turvakinnitus enne kasutuselevõttu. | | Määratletud ja kirjeldatud trakt kasutajate tuvastuse ja autentimise protseduurideks. Andmete krüpteerimine valikuliselt. | Tõkestada kontrollimatu sissehelistuspääs ja volitamatud ühendused välisvõrkudega. Krüpteerimisest loobumine valikuline ja põhjendusega. |
| Personaliturve | Kõigi kasutajate koolitus kasutatava rakenduse, tarkvaraprotseduuride ja turbemiinimumi alal. | Tausta kontroll töölevõtul. | | Tugev tausta kontroll töölevõtul. |
| Keskkonnaturve | Mikrokliima reguleerimine (tolm, temperatuur, niiskus, tuulutus). Toitepingetõugete tõrje. | | | |
| Taasteplaanid | | Koostada vastavalt standardjuhistele. | | |

**Tabel 2. Turvameetmete toime turvaaspektidele**

| KÄIDELDAVUS | TERVIKLUS | KONFIDENTSIAALSUS |
|---|---|---|
| | | |
| **ORGANISATSIOONILISED MEETMED** | | |
| **Haldus. Töökorraldus** | | |
| ***1. tase:***<br>Logide läbivaatus<br>Varundamine ja taaste<br>Dokumenteeritud protseduurid<br>Süsteemiarenduse elutsükkel<br>Hankelepingud:<br>• riistvara<br>• tarkvara<br>• side<br>Spetsifitseerida:<br>• maksimaalne seisuaeg<br>• kriitilised miinimumid<br>Talitluse pidevuse plaanimine<br>Talitluse taasalustusplaan | Muutuste haldus<br>Infokandjate märgistamine<br>Logiprotseduurid ja läbivaatus<br>Verifitseerimine<br>Turvaaudit<br>Testimine | Vastutuste määramine<br>Kohustuste lahutamine<br>Liigitusprotseduurid<br>Süsteemiarenduse elutsükkel<br>Standardid, poliitikad<br>Talitluse taasalustusplaan<br>Tundlikkuse määrangud<br>Turvasätted lepingutes |
| **Personal** | | |
| ***1. tase:***<br>Koolitus<br><br>Spetsifitseeritud varutöötajad | Koolitus<br>Ametijuhend<br>Töökohustused<br>Vallandamisprotseduurid | Turvateadlikkuse koolitus<br>Tausta kontroll<br>Vallandamisprotseduurid<br>Turvasätted lepingutes |
| ***2. tase:***<br>Avariirühm | | Kohustuste lahutamine<br>Tundmistarbe printsiip |
| ***3. tase:***<br>Taasterühm | Pääsu autentimine | Vastastikune aktsepteeritavus<br>Pääsu verifitseerimine |
| | | |
| **FÜÜSILISED MEETMED JA INFRASTRUKTUUR** | | |
| ***1. tase:***<br>Keskkonnameetmed<br>Tuletõrjeseadmed | Keskkonnameetmed | Pääsu reguleerimine<br>• füüsiline<br>• loogiline |
| ***2. tase:***<br>Ladustus väljaspool objekti | Füüsiline pääsu reguleerimine<br>Andmekandjate ohutu transport | Uste turve<br>Kapitaalseinad<br>Jäätmete hävitamine |
| ***3. tase:***<br>Alternatiivne asukoht | | Valvesignalisatsioon<br>Volituste verifitseerimine |

| KÄIDELDAVUS | TERVIKLUS | KONFIDENTSIAALSUS |
|---|---|---|
| | | |
| **INFOTEHNIKA: ARVUTISÜSTEEMID** | | |
| ***1. tase:***<br>Regulaarne hooldus<br>Muudatuste haldus<br>Vahendite inventuur<br>Tarkvara ja andmete varukoopiad<br>Minimaalkonfiguratsioon | Muudatuste haldus<br>Õiguste piiramine<br>Konfiguratsioonihaldus | Pääsu reguleerimine:<br>• süsteemipääs<br>• failipääs<br>Kohustuste lahusus:<br>• väljatöötamine<br>• testimine<br>• tootmistöö<br>Turvaklass C1/C2 |
| ***2. tase:***<br>Puhvertoiteallikad<br>Varuriistvara | Vahemike kontroll<br>Väärtuste kontroll<br>Veaavastus<br>Veaparandus | Andmekandjate füüsil. eraldamine<br>Tehingute logimine<br>Audit<br>Õiguste piiramine<br>Turvaklass B1/B2 |
| ***3. tase:***<br>Varuasukoht<br>Talitluse pidevuse plaanimine | Kontrollkoodid<br>Logimisvigade arvestus<br>Tehingupäevikud<br>Autentimine | Krüpteerimine<br>Turvaklass B3/A1<br>Kiirguse piiramine |
| | | |
| **INFOTEHNIKA: SIDE** | | |
| ***1. tase:***<br>Konfigureerimine<br>Muudatuste haldus<br>Logide läbivaatus<br>Spetsifitseerida<br>• maksimaalne seisuaeg<br>• kriitilised miinimumid | Konfigureerimine<br>Muudatuste haldus<br>Järelevalve<br>Veaavastus<br>Saate kordamine<br>Logide läbivaatus | Konfigureerimine<br>Järelevalve<br>Logide läbivaatus<br>Muudatuste haldus |
| **2. tase:**<br>Varumarsruudid | | Pääsu reguleerimine<br>Autentimine<br>Lihtkrüpteerimine<br>Kiirguse piiramine |
| **3. tase:**<br>Teenuste dubleerimine | Autentimine | Tugev krüpteerimine |

### 6.3.3 Kitsendused

Lisaks meetmete turbeomadustele tuleb arvestada ka nende valimist piiravaid kitsendusi. Tüüpilised kitsendused on alljärgnevad.

***Ajalised***. Näiteks võib juhtkond nõuda, et meetmed peavad olema rakendatud mingiks kindlaks tähtpäevaks, või vastupidi, neid ei saa evitada enne teatud hetke. Evitusaega võib piirata ka infosüsteemi enda eluiga. Evitamist võib edasi lükata uue turvatoote ilmumisaeg.

***Rahalised***. Turvameetmed ei tohi üldiselt olla kulukamad nendega kaitstavate varade väärtusest (vt 5.1). Kui aga kõrge turvatarve on tingitud mitterahalistest teguritest, ei saa selle kitsendusega sõnasõnalt arvestada ning lõppotsuse peab tegema juhtkond.

***Tehnilised***. Eeskätt riistvara ja tarkvara ühilduvusega seotud piirangud. Enamasti on neid kergem vältida, kui infosüsteem kavandatakse ja teostatakse algusest peale koos vajalike turbevahenditega.

***Sotsiaalsed ja kultuurilised***. Turvameetmed vajavad personali aktiivset tuge. Et personal nad omaks võtaks, ei tohi nad oma otstarbelt ega olemuselt olla vastuolus kohalike, sh organisatsioonis valitsevate kultuuritavadega. Kultuuriliselt vastuvõtmatuid turvameetmeid hakatakse ignoreerima.

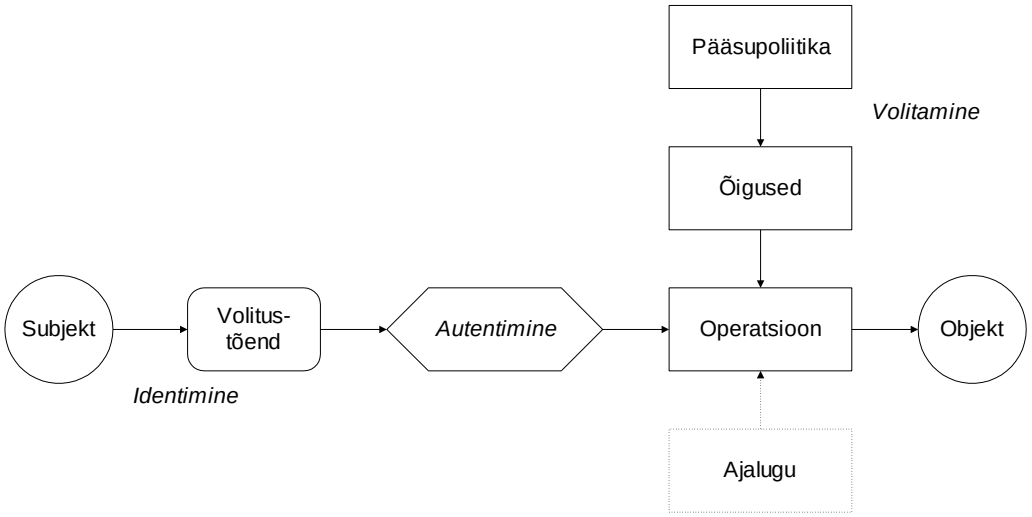
***Keskkondlikud***. Valikut võivad mõjutada kohalik kliima, lähiümbruse geograafilised iseärasused, kasutadaolev ruum jne.

***Õiguslikud***. Turvameetmete valikut võivad mõjutada seadused ja eeskirjad, mis puudutavad infotöötlust otseselt (isikuandmete seadus, andmekogude seadus, riigisaladuse seadus jt) või kaudsemalt (töökaitseseadused, tuletõrje-eeskirjad jms).

# 7 PÄÄSU REGULEERIMINE

## 7.1 Pääsumehhanism

Pääsu reguleerimine on loogilise eraldamise protsess, mille otstarve on tagada, et juurdepääs kaitstavale infoobjektile oleks ainult volitatud subjektidel ja et see toimuks turvalisuse seisukohalt lubataval viisil. Pääsu reguleerimine kaitseb kõiki turvalisuse aspekte, eelkõige konfidentsiaalsust. Tüüpilist pääsu reguleerimise protsessi illustreerib Joonis 3.




**Joonis 3. Pääsu reguleerimise lihtsustatud üldskeem**

***Objekt*** Joonis 3 on infosüsteemi või ta infrastruktuuri suvaline loogiliselt eraldatav komponent või komponentide kogum, näiteks andmefail, -kirje, -väli, muutuja, failikataloog, operatsioonisüsteem, programm, personaalarvuti, välisseade, võrguteenus, tööruum, sidekanal.

***Subjekt*** on objekti potentsiaalne kasutaja (isik või protsess). Subjekte võidakse pääsuhalduse lihtsustamiseks ühendada rühmadeks, mida saab käsitleda kollektiivsubjektina.

***Operatsioon*** on objekti kasutamise spetsiifiline toiming, mille sisu sõltub objekti tüübist. Andmeüksuse puhul võivad operatsioonid olla näiteks objekti loomine, lugemine, muutmine, kopeerimine, kustutus; või: lugemine, kirjutus. Ruumide lukusüsteemi puhul võivad operatsioonid olla näiteks sissepääs, väljapääs, läbipääs mõlemas suunas.

***Pääsupoliitika*** on turvanõudeid väljendav reeglistik, mis määrab objektide lubatava kasutamise: millistel subjektidel on õigus mingeid objekte kasutada ja mil viisil, st milliste operatsioonidega. Niisiis määratleb pääsupoliitika lubatavate kolmikute (subjekt, objekt, operatsioon) hulga. Pääsupoliitika põhiolemust väljendab maatriks Joonis 4. Objekti loomusest tingituna ja mitmesuguste spetsiifiliste rünnete tõrjeks on tegelik pääsupoliitika enamasti keerukam, sisaldades mitmesuguseid ajalisi ja muid lisatingimusi (vt 7.2). Näiteks võib lubatav operatsioon andmebaaside puhul sõltuda ka päringu ajaloost (vt kriipsjoonega esitatud haru Joonis 4), st sellest, milliste objektide poole pöördus subjekt enne käsilolevat pääsutaotlust ja milliseid operatsioone ta nendega sooritas.

***Volitamine*** (*authorization*) on pääsupoliitika rakendamise halduslik komponent, mis seob pääsupoliitika mudelisse kuuluvad olemid vastavusse konkreetsete objektide ja subjektidega. Kuna hajusas võrgukeskkonnas ei saa luua ühtset tsentraliseeritud haldussüsteemi, algatatakse seal volitusprotsess enamasti vahetult subjekti pääsukatsel ning on tihedamalt seotud pääsu võimaldava autentimismehhanismiga; seetõttu nimetatakse võrguteenuste puhul pääsu reguleerimist sageli lihtsalt volitamiseks.

***Pääsumehhanism*** on pääsupoliitikat realiseerivate infotehniliste vahendite kogum. Pääsupoliitika tagamiseks peavad pääsumehhanismi alati toetama vastavad organisatsioonilised ja sageli ka füüsilised meetmed. Pääsu automatiseerimiseks peab pääsumehhanism saama subjekti identsust automaatselt kontrollida. Selleks kasutab ta abiprotsessina ***autentimist***. Pääsu taotlemiseks peab subjekt ennast või oma

volitusi tõendama. Selleks ***idendib*** ta end, st esitab mingi identsus- või ***volitustõendi***. Autentimine on subjekti väidetava identsuse tõesuse verifitseerimine volitustõendi alusel. Autentimine on iseseisev turvatehnika haru, millel on teisigi rakendusi, seetõttu käsitletakse teda eraldi ja detailsemalt järgmises peatükis.

***Pääsumudel*** on pääsupoliitika formaliseeritud esitus. Süsteemide turvalisuse modelleerimisel on valdav osa senistest uuringuist seotud just pääsumehhanismide formaalse analüüsiga, eeskätt andmebaasisüsteemide ja ühiskasutuslike operatsioonisüsteemide tarbeks. Selleks on loodud kümneid pääsupoliitikate mudeleid. Viimastel aastatel on uuringute keskmesse tõusnud hajuskeskkonnad, eriti pääsuprobleemid kaugvõrkudes.

## 7.2 Pääsupoliitikad ja pääsumudelid

### 7.2.1 Pääsupoliitika

Pääsupoliitika reeglistik koostatakse teatavatest tüüpilistest komponentidest. Näiteks määratlevad USA (NIST ja NSA) kaitseprofiili föderaalkriteeriumid alljärgnevad pääsupoliitika komponendid.

***Atribuutide määratlemine*** reeglite koostamiseks.

Subjekti atribuudid:
- kasutaja rekvisiidid (nt kasutaja/grupi/rolli identifikaator(id), konfidentsiaalsus- või terviklustasemed, pääsuperioodid, pääsukoha identifikaator),
- privileegsubjektide rekvisiidid (nt süsteemiprivileegid).

Objekti atribuudid:
- kasutajaspetsiifilised (nt konfidentsiaalsus- või terviklustasemed, pääsuaja ja -koha piirangud),
- muud (nt erinevad objektiõigused eri kasutajaile).

Kontekstiatribuudid, nt
- rühmade määratlused,
- jooksev ajahetk,
- eriolukordade indikaatorid.

***Atribuutide haldus***: reeglid, mille alusel mingi subjekt saab muuta enda, teiste subjektide ja objektide atribuute, ning impordi- ja ekspordioperatsioonide atribuutide määratlemise reeglid. Atribuutide halduse määravad näiteks atribuutide jaotamise ja kõrvaldamise parameetrite valimine järgmiste hulgast:

- selektiivsus: jaotamine üksikatribuutide tasemel (kasutaja, rühm, roll, luba, privileeg, turvatase);
- transitiivsus: edasivolitatud luba kaotab kehtivuse algse loa tühistamisel;
- viivitamatus: atribuudi kinnistamine või kõrvaldamine peab toimuma etteantud ajaga;
- sõltumatus: sama subjekti atribuute saavad anda või ära võtta mitu subjekti sõltumatult;
- ajastatus: atribuudi andmise või äravõtmise toime algab teatud ajal ja kestab teatud aja;
- kohtkindlus: atribuute saab anda või ära võtta teatud kohas.

***Objektipöörduste volitamine***: reeglid, millega määratakse

- subjekti volitus sooritada mingit toimingut,
- toimingu volitus sooritamiseks õhe või mitme objektiga,
- subjektide ja objektide granulaarsus, sh ajaline,
- subjekti volituste edasidelegeerimine.

***Subjektide ja objektide loomine ja kõrvaldamine***: reeglid, millega määratakse

- volitused teatud atribuutidega subjektide ja objektide loomiseks ja kõrvaldamiseks,
- objektide korduskasutus (nt ei tohi järgmisele subjektile siirduda teave eelmise kasutamise kohta),
- vaikeatribuudid ja võimalik pärilikkus.

***Objektide kapseldus***. Reeglid objektipääsu kitsendavate subjektide kohta lähtuvad järgmistest printsiipidest:

- kõik objektipöördused toimuvad neid objekte üldisema pääsu eest kaitsva isoleeritud subjektihulga kaudu, kusjuures igal subjektil on ühene kaitstud sisendpunkt;
- kaitsesubjektidel puudub juurdepääs muudele objektidele ja nad ei saa ära anda juurdepääsu kaitstavatele objektidele.

### 7.2.2 Diskretsionaarsed pääsupoliitikad

See ***informatsiooni omandusel*** ja ***õiguste delegeerimisel*** põhinev poliitikate tüüp (*discretionary access policy*, *DAC*) on levinuim ja kommertsrakendustes (eriti andmebaasisüsteemides) peaaegu ainuvaldav.

***Kinnise*** DAC-poliitika puhul on objektipääs vaikimisi keelatud ning subjekt-objekt-paari määratlemisega antakse pääsuluba, st kõik, mis pole lubatud, on keelatud.

***Lahtise*** DAC-poliitika puhul on objektipääs vaikimisi lubatud ning subjekt-objekt-paari määratlemisega keelatakse pääs, st kõik, mis pole keelatud, on lubatud.

Ressursi omaniksubjektil on maksimaalsed õigused talle kuuluva ressursi kasutamiseks ning ta saab ressurssi kaitsta otseste volitamata pöörduste eest. Diskretsionaarne tähendab meelevaldset ning meelevaldsus avaldub siin selles, et omaniksubjekt (nt faili looja) saab oma pääsuõigusi otseselt või kaudselt edasi anda teisele subjektile, puuduvad vahendid, mis sunniksid rakendama ühtseid pääsureegleid kogu süsteemi ulatuses. Omandusprintsiip kõrgematel hierarhia tasemetel tavaliselt ei kehti, näiteks on süsteemiülemal juurdepääs kõigile andmetele, sõltumata nende omandusest.
.
Diskretsionaarne poliitika realiseeritakse sageli iga objekti juurde kuuluva ***pääsuloendiga*** (*access control list*, *ACL*), mis sisaldab subjektiatribuudi ja lubatava (või keelatud) pöörduse paare. Teine analoogiline vahend on iga subjektiga seotud ***voliloend*** (*capability list*), mis sisaldab objekti ja lubatava pöörduse paare. Levinud on ka B. Lampsoni 70il aastail loodud ja formaliseeritud ***maatriksmudel*** (vt Joonis 4), mida arendasid edasi G. Graham, D. Denning ja M. Harrison. Viimasel kümnendil on uusi DAC-mudeleid loodud andmebaasisüsteemide tarbeks (E. Bertino jt).

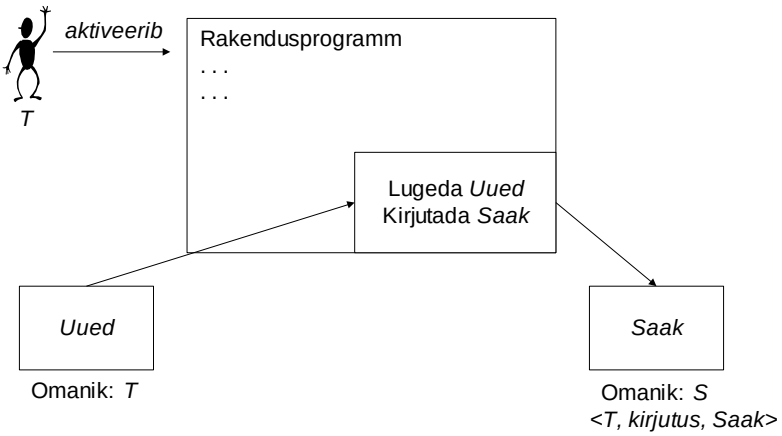
| **Objektid:** | $O_1$ | $O_2$ | $O_3$ | $O_4$ | ... | ... | ... | ... | $O_n$ |
|---|---|---|---|---|---|---|---|---|---|
| **Subjektid:** | | | | | | | | | |
| $S_1$ | - | L | LK | L | ... | ... | ... | ... | - |
| $S_2$ | L | - | - | L | ... | | | | |
| $S_3$ | L | L | - | L | | | | | |
| ... | ... | ... | ... | .. | | | | | |
| ... | | | | | | | | | |
| ... | | | | | | | | | |
| .. | | | | | | | | | |
| $S_m$ | - | - | - | L | ... | | ... | ... | L |

**Joonis 4. Pääsumaatriks (näide)**

Kuna kitsendused objektide ühiskasutusele on nõrgad ning infovoogu subjektide/objektide ja neid mitte esindavate süsteemimuutujate (näiteks operatsioonisüsteemi sisemuutujate) vahel ei reguleerita, on DAC puhul võimalikud volitamata pöördused, objektide ja subjektide volitamata loomine ja hävitamine ning pääsuõiguste volitamata levi. Seetõttu ei paku DAC kaitset salakanalite, trooja hobuste, viiruste ega surrogaatpöörduste eest. Konfidentsiaalsust kaitsvaid kitsendusi ei saa selgekujuliselt määratleda.

Näide 1. Mingi andmekogumi omanik võib anda selle kogumi lugemise volituse kasutajale A, mitte aga kasutajale B. Kasutaja A võib aga poliitika seisukohalt legaalsete protsessidevahelise suhtluse vahenditega edastada andmekogumi sisu kasutajale B.

Näide 2. Turundusdirektoril *T* on arvutis tabel *Uued*, mis sisaldab uudistoodetele kavandatud hindu. Pääsumehhanism annab lugemispääsu ainult tabeli omanikule *T*. Turundusdirektori alluvuses töötab konkurendi spioon *S*, kes loob tabeli *Saak* ning omanikuna määrab kirjutusõiguse subjektile *T*. Seejärel sokutab ta direktori tabeliprogrammi trooja hobuse, mis sisaldab kaht operatsiooni: lugeda tabel *Uued* ja kirjutada see tabelisse *Saak* (vt Joonis 5).




**Joonis 5. Trooja hobune diskretsionaarpääsuga süsteemis**

Diskretsionaarsel pääsupoliitikal põhinevad näiteks USA kaitseministeeriumi senise turvanormistiku TCSEC madalaima, C-taseme (D-tase tähendab sisuliselt turbeta süsteeme) turvaklassid C1 ja C2. Klass C2 tõhustab kaitset pääsuolemite peenema granulaarsusega ning mõnede lisameetmetega, mis ei puuduta pääsu reguleerimist. Klassile C2 vastavad näiteks Unixi uuemad versioonid ja Windows NT. Rahvusvahelises ulatuses on diskretsionaarse poliitika osaliselt standardinud ISO (SQLi standardis) ja ECMA.

Diskretsionaarset poliitikat on varieeritud lisakitsendustega ja kombineeritud muude pääsupoliitikatega. Mittediskretsionaarsed on kõik pääsupoliitikad, mis sisaldavad vahendeid ülalnimetatud turvaaukude kõrvaldamiseks, eeskätt infovoo reguleerimiseks.

### 7.2.3 Mandatoorsed pääsupoliitikad

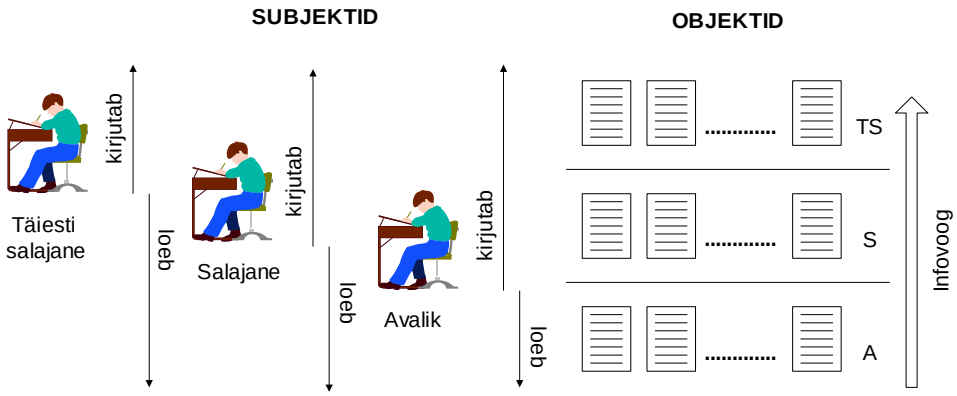
Infovoogude reguleerimiseks süsteemis kinnistab mandatoorne pääsupoliitika (*mandatory access policy*, *MAC*) subjektidele ja objektidele ***turvaklassid***, mida kirjeldavad ***märgendid***. Mandatoorsed poliitikad töötati välja militaarkeskkonnas, kus märgendamist rakendati juba enne automatiseeritud infosüsteemide teket. Iga märgend koosneb kahest komponendist: olemi tüüpi kirjeldavast mittehierharhilisest (järjestamata) kategooriast ning hierarhilist turvataset määravast objekti tundlikkusastmest (näiteks: täiesti salajane > salajane > konfidentsiaalne > avalik) või subjekti lojaalsusastmest. Ühe komponendi järjestamatuse tõttu on turvamärgendid ainult osaliselt järjestatud ja moodustavad matemaatilises mõttes võre. Selles võres on klass $c_1$ võrreldav klassiga $c_2$ ja sellest kõrgem (≥), kui ta turvatase on suurem $c_2$ omast või sellega võrdne ning ta kategooriates sisalduvad $c_2$ omad.

Levinuimad MAC kirjeldamise formaalsed vahendid on Bell-LaPadula mudel ja Biba mudel. Andmeüksuste puhul taandub Bell-LaPadula mudeli keskne sisu kahele reeglile:

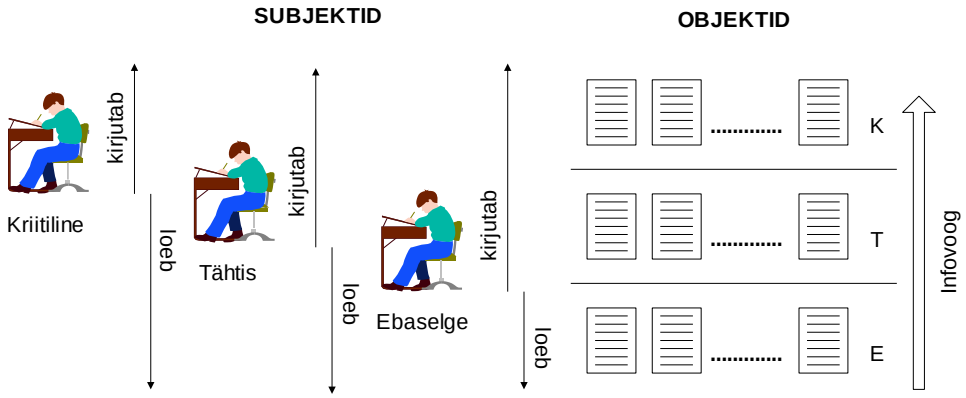
(1) Subjekt *S* tohib lugeda andmeüksust *A*, kui $c_S \geq c_A$.
(2) Subjekt *S* tohib kirjutada andmeüksust *A*, kui $c_S = c_A$.

Andmebaasirakendusteks on MAC-mudelit edasi arendanud S. Jajodia, R. Sandhu jt. Andmebaaside puhul tähendab mandatoorse poliitika rakendamine, et eri turvatasemega subjektidele esitatakse andmebaasi kohta erinevaid vaateid. Tsiviilsfääris on loodud mitmeid MAC-poliitikal põhinevaid pilootsüsteeme ja ka ärilisi tooteid (Ingres, Oracle, Sybase, Trudata jt). Seni puudub üksmeel optimaalse granulaarsuse suhtes; teostuste skaala ulatub kogu andmebaasi kaitsest kuni üksikatribuutideni ja isegi teatud atribuudiväärtusteni.

Mandatoorne mudel võimaldab infovoogude juhtimisega otseselt ette anda salastus- ja tervikluskitsendusi (vt Joonis 6 ja Joonis 7, kus struktuur erineb ainult voogude suuna poolest).




**Joonis 6. Konfidentsiaalsuse kaitse infovoogude reguleerimisega**




**Joonis 7. Tervikluse kaitse infovoogude reguleerimisega**

Kuna kasutajad ei saa ise anda teistele pöördusvolitusi, on andmed kaitstud DAC-poliitikale omaste lekete ja trooja hobuste eest. Lõplikult lahendamata on mõningad salakanalitega seotud probleemid. MAC-poliitika puudusteks on liiga jäik staatilisus, subjektide turvataseme muutusi ei saa automaatselt arvestada, ei saa rakendada võimalikku kollektiivpääsu nõuet, tsentraliseeritud ja käsitsi sooritatav märgendamine on kohmakas. Nende puuduste tõttu ei sobi ta paljudele ärirakendustele.

Mandatoorse pääsupoliitika rakendamist nõuab TCSEC tase B (klassid B1, B2, B3).

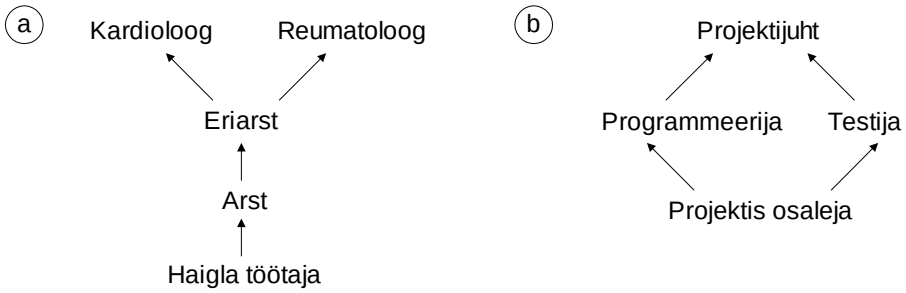
### 7.2.4 Rollipõhised pääsupoliitikad

Seni vaadeldud poliitikatüübid oma traditsioonilisel kujul ei rahulda paljusid tegeliku elu vajadusi. Koostöises ja autonoomiat taotlevas akadeemilises keskkonnas tekkinud diskretsionaarsed poliitikad on äriettevõtete infovarade kaitseks liiga nõrgad, militaarkeskkonnast pärit kohmakad ja jäigad mandatoorsed poliitikad on suunatud eeskätt valitsuse salastuseeskirjade täitmisele. Kumbki neist poliitikatüüpidest ei arvesta volituste andmisel pääsuobjektide infosisu. Seetõttu on eriti viimasel kümnendil pakutud rohkeid ärimaailmale paremini sobivaid alternatiivmudeleid. Viimastel aastatel on uuringute (R. Sandhu jt) ja väljatöötuse keskmes olnud rollipõhised pääsupoliitikad, mille alged tekkisid

juba 70tel aastatel. Tuntud formaliseering on näiteks Clark-Wilsoni mudel (1987), mis on suunatud eelkõige andmetervikluse tagamisele.

Rollipõhised poliitikad (*role-based access policy*, *RBAC*) reguleerivad kasutajate pääsu infoobjektide juurde vastavalt nende tööalasele funktsioonile süsteemis. Seda funktsiooni väljendab kasutaja ***roll*** – tema töötoimingute ja -ülesannete kogum. Kasutajaile volitatakse nende rollid, rollidele aga juurdepääs objektidele. Üldjuhul võib ühel ja samal kasutajal olla erinevates olukordades eri rolle, üks ja sama roll võib aga olla mitmel kasutajal. Eri poliitikad võimaldavad seda asjaolu arvestada erineval määral, näiteks lubades kasutajale korraga ainult üht rolli.

Rollidel võib olla kattuvaid ülesandeid ja privileege, seetõttu tuleb eri rollidel kohati sooritada samu toiminguid; enamasti on asutuses teatavaid üldtoiminguid, mida tuleb sooritada kõigil. Neid asjaolusid arvestab poliitikamudelis ***rollihierarhia***, milles rollid võivad sisaldada teisi rolle. Sellises hierarhias kehtib ***õiguste päritavus***, st iga roll saab lisaks oma eriõigustele kõigi temas sisalduvate rollide õigused. Rollihierarhiad on loomulik viis vastutus-, võimu- ja pädevussuhete kirjeldamiseks (vt Joonis 8). Rollihierarhia võimaldab senivaadeldud poliitikatega võrreldes lihtsalt ja ökonoomselt rakendada piisava pääsu printsiipi: iga roll saab juurdepääsu teabele ainult oma tööks vajalikus ulatuses.




*Nooled näitavad pärilussuunda*

**Joonis 8. Rollihierarhiate näiteid**

Rollipõhise poliitika sisu määratlevad järgmised reegliteks formaliseeritavad põhiprintsiibid, mida võidakse täiendada lisatingimustega või kombineerida muude pääsupoliitikatega. DAC- ja MAC-mudelitega võrreldes on RBAC õieti "poliitiliselt neutraalne", ta on vahend, mis võimaldab realiseerida etteantud poliitikat.

***Staatiline kohustuste lahusus***. Määratletakse üksteist välistavad rollid (näiteks panga teller ja audiitor), mida ei tohi volitada ühele ja samale subjektile.

***Rollikandjate arvu piiramine***. Iga rolli saab kanda teatav etteantud maksimaalarv subjekte. Näiteks saab asutuse peadirektori rollis olla korraga ainult üks isik, ehkki alalise peadirektori rolli võivad tema äraolekul kanda teised.

***Dünaamiline kohustuste lahusus***. Mõnedel juhtudel võib teatud kohustuste alaline (staatiline) lahusus olla tööülesannete seisukohalt ebaotstarbekas, küll aga on vaja vältida niisuguste kohustuste üheaegsust. Näiteks võib üks ja sama isik olla mingi kulutuse algataja, ühtlasi aga mingi teise kulutustaotluse kinnitaja; loomulikult ei saa ta korraga olla ühe ja sama kulutuse algataja ja kinnitaja ning see võimalus välistatakse vastavate reeglitega.

***Protseduuriline kohustuste lahusus*** on vajalik näiteks võltsingute tõrjeks. Mitmesuguste kuritarvituste võimaluse tekitaks ühe ja sama isiku juurdepääs mingi kriitilise funktsiooni järjestikustele faasidele. Näiteks võib mingi kauba ostmine koosneda järgmistest operatsioonidest: ostutellimuse kinnitamine, arve saabumise registreerimine, kauba saabumise registreerimine, maksekorralduse kinnitamine. Vastav reegel lubab mingil rollil operatsiooni sooritada ainult siis, kui samale rollile pole kinnistatud ühtki käsiloleva protseduuri eelmist operatsiooni.

Rollipõhiste poliitikate puhul on volituste haldus traditsiooniliste poliitikatega võrreldes tunduvalt hõlpsam, poliitika on paindlikum, sobib paremini organisatsiooni loomuliku struktuuriga, võimaldab arvestada objektide infosisu ja selle konteksti ning teda saab ühendada teist tüüpi turvapoliitikatega (vt 7.2.5) . Seetõttu on teda rakendatud mitmetes viimaste aastate standardiprojektides ja kommertstoodetes. RBAC on lülitatud viie riigi ühisstandardi *Common Criteria* äriettevõtetele mõeldud turvaprofiili, mitmesse NIST projekti, süsteemidesse CORBA, Kerberos, Sesame, DCE ning andmebaaside halduseks määratud SQL3 standardisse (tuginedes teostusnäitele baasisüsteemis Oracle 7). Rolle kasutatakse ka uuemate võrguoperatsioonisüsteemide (Novelli NetWare, Microsofti Windows NT) turvahalduses.

**Halduslik rollipõhine pääsu reguleerimine** (*administrative role-based access control, ARBAC*) on RBAC-mudeli edasiarendus, mis rakendab RBAC-meetodit RBAC-poliitika halduseks.

### 7.2.5 Variandid ja hübriidpoliitikad

***Hiina müüri poliitika***

on diskretsionaarse ja mandatoorse poliitika kombinatsioon, mille 1989. a. tuletasid briti rahanduseteenuste seadusest D.Brewer ja M.Nash ja mis taotleb siseringi teabe kuritarvituse tõkestamist näiteks konsultatsioonifirmades. Kui konsultandil on juurdepääs mingi firma siseteabele, ei tohi ta nõustada konkureerivat firmat. Konsultatsioonifirma arvutis olev teave firmade kohta tükeldatakse kogumiteks, mis vastavad konkureerivatele huvigruppidele. Konsultandist kasutaja võib vabalt valida suvalisse (näiteks pankade) gruppi kuuluva suvalise firma andmed, kuid pärast esimest pöördust muutub ta turvaklass, nii et ta ei saa enam valida ühtki teist samasse huvigruppi kuuluvat firmat. Selline mandatoorne kitsendus ei puuduta aga teisi huvigruppe (näiteks kindlustusseltse või veondusfirmasid), mida ta pole veel valinud. "Hiina müür" ehitatakse ainult juba valitud firma teabe ja teiste sama grupi firmade teabe vahele.

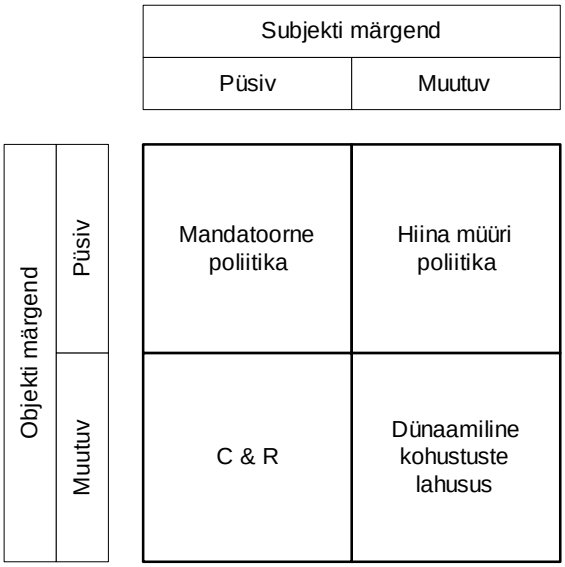
Hiina müüri poliitika oluline komponent on niisiis muutuv subjekti turvamärgend. Märgendite dünaamika järgi liigitades jagunevad pääsupoliitikad nelja klassi. Seda jaotust illustreerib Joonis 9. Muutuvate objektimärgenditega poliitikate olemus sellel skeemil on lühidalt järgmine.

***Kontrollitud väljastuse poliitika*** (*check and release, C & R*)

vastab näiteks traditsioonilistele pangaprotseduuridele, kus makseandmete sisestus, kontroll ja kinnitamine hoitakse lahus. Subjektimärgendid on püsivad (näiteks teller ja kontrolör), kuid makseandmete märgend muutub: kinnitamiseelne kasutamise keeld (volitamatus) muutub pärast kinnitamist loaks (volitatuseks).

***Dünaamiline kohustuste lahusus***

on näiteks üks võimalik rollipõhise poliitika komponent (vt 7.2.4). Elektronarvelduses on üks tüüpilisi rakendusi juhtudel, kus tehinguid kinnitavatele juhtidele on antud õigus teha tehingutesse muudatusi, kuid pärast muudatuste tegemist ei tohi nad tehingut kinnitada kohe, enne ümbervormistust.




**Joonis 9. Pääsupoliitikate liigitus märgendite dünaamika järgi**

***"Kohandatud mandatoorne pääsu reguleerimine"*** (*adapted mandatory access control*, *AMAC*)

on näide rollipõhise poliitika kombineerimisest mandatoorsega. Mudeli töötas P.Cheni relatsioonskeemist lähtudes välja G.Pernul 90te alguses ja ta on mõeldud eeskätt turvaliste mitmekihiliste andmebaaside väljatöötamiseks. Iga rolli jaoks koostatakse eraldi kontseptuaalskeem ja sellest tuletatakse vastav andmebaasivaade. Mandatoorse poliitika sisseviimiseks tuleb defineerida objektid ja subjektid. AMAC-mudeli turvaobjektid on andmebaasi fragmendid, subjektid aga vaated. Fragment on suurim kahele või mitmele vaatele kättesaadav andmebaasi osa. AMAC toetab objektide automaatset turvamärgendamist, mis põhineb eeldusel, et objekti tundlikkusaste on seda kõrgem, mida vähemas arvus vaadetes ta sisaldub. Mandatoorsed kitsendused realiseeritakse andmebaasisüsteemis sisalduvate kaitsemehhanismidega. Puhtmandatoorse poliitikaga võrreldes on AMAC märksa paindlikum, peegeldab paremini organisatsiooni struktuuri ja on kergem hallata. Ta toetab kõiki infosüsteemi analüüsi, modelleerimise ja projekteerimise faase. AMAC-poliitika puudusteks on piiratud võimalused konfidentsiaalsuskitsenduste seadmiseks ja dünaamilise volitamise toe puudumine.

***"Isikuteadmuse meetod"*** (*personal knowledge approach*, *PKA*)

on rollipõhise ja diskretsionaarse poliitika kombinatsioon, mis keskendub isikuandmete privaatsusele. Mudeli töötasid välja J.Biskup ja H.Brüggemann. Mudeli põhieesmärk on anda tuge inimeste teabelise enesemääramise õigusele, mis on paljudes maades kaitstud vastavate seadustega: isikul on õigus valida, milliseid oma eraelu elemente ta nõustub avalikustama. Mudeli põhiolemid on isik ja ta teadmus. Iga andmebaasis esindatud inimene teab enda kohta kõike; kui ta aga soovib midagi teada kellegi teise kohta, peab ta seda temalt küsima.

Mudel tugineb järgmistele elementidele.

1. *Isikud* – infosüsteemi kasutajad ja baasis oleva teabega esindatud inimesed. Iga isik on kapseldatud objekt, mis sisaldab kahesugust teadmust: kõiki andmeid iseenda kohta ja seoseid teiste süsteemis olevate isikutega. Midagi muud objekt püsivalt ei tohi "mäletada".

2. *Tuttavad* – isiku keskkonda kuuluvad objektid, millega isik tohib suhelda päringute ja värskendustaotluste saatmise teel. Tuttavate hulk võib muutuda dünaamiliselt.

3. *Rollid ja volitused*. Sõnumi saaja reaktsioon sõltub saatja volitustest, isiku volitused oma tuttava suhtes sõltuvad aga tema hetkerollist. Volitused ja rollid on süsteemis deklareeritud staatiliselt.

4. *Mälu*. Iga isik "mäletab" saadetud ja vastuvõetud sõnumeid. See võimaldab teabe revisjoni ning tehingute jälitust vastava isikuni.

Sisselogimisel kinnistatakse igale kasutajale objektitüübi *isik* eksemplar ning ta saab endale individuaalsed tuttavad ja staatilised rollivolitused. Kui kasutaja annab päringu, saadetakse see automaatselt ainult ta tuttavatele.

Mudel on realiseeritud pilootsüsteemides ja sobib näiteks tervishoiualastele või rahvaloenduse andmebaasidele. Diskretsionaarse komponendi tõttu on ta aga rünnatav trooja hobusega.

### 7.2.6 Tehingupõhised pääsupoliitikad

Senistest pääsupoliitikatest on kõige paindlikumaks osutunud rollipõhised, kuid ka nende võimalused jäävad organisatsiooni protseduuride automatiseerimise, klient-server-süsteemide ja laiema hajuskeskkonna tingimustes ebapiisavaiks, eriti andmetervikluse tagamisel; uuringute andmetel on aga äriettevõtetes andmeterviklus märksa olulisemal kohal kui konfidentsiaalsus. Seetõttu on viimastel aastatel hakatud otsima uusi mudeleid, mis paremini sobituksid selliste tingimustega ning võimaldaksid paremini arvestada kaitstava teabe ja ta töötluse laiemat konteksti.

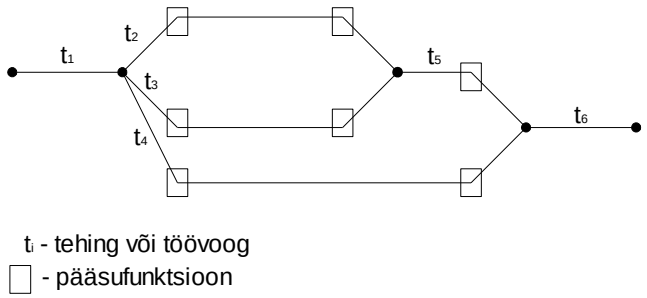
Aktuaalne ja perspektiivne uurimissuund on tehingupõhised pääsupoliitikad (*task-based/transaction-based access control*, *TBAC*), mille formaalseid mudeleid töötavad ARPA finantseerimisel välja R.Thomas, R.Sandhu jt. Niisugune nn aktiivne pääsumudel erineb oluliselt senistest, mis püüdsid formuleerida ainult volitusseoseid objektide ja subjektide vahel. TBAC on orienteeritud organisatsiooni tegevuse kontekstis sooritatavatele töödele või tehingutele ning hõlmab suurt granulaarsuse skaalat, alates kliendi ja serveri suhtlusest hajussüsteemis ning lõpetades allüksuste või asutuste vaheliste töövoogudega. Mudel sobib hajusale infotöötlussüsteemile, milles on palju pääsu-, juhtimis- ja otsustuspunkte.

***Tehing*** on TBAC-mudeli keskne olem. See on loogiline tööüksus, mis võib

- olla pikaealine,
- koosneda paljudest alamtehingutest, mis nõuavad volitamist ühelt või mitmelt subjektilt,
- hõlmata paljusid sooritajaid,
- olla hajutatud ruumis ja ajas.

***Objektid*** on määratletud analoogiliselt organisatsiooni varasemale pabertehnoloogiale ning jagunevad ajutisteks (vrd tellimused, kviitungid jms) ning alalisteks (vrd pangakonto, laoandmebaas jms). Andmetervikluse tagamise seisukohalt vajavad reguleerimist eeskätt ajutised objektid, alalised kuuluvad pääsusüsteemi kõrvalharuna.

Traditsioonilistelt pääsumudelitel tuli vastata küsimusele "kas subjektile S on lubatud pääs P objekti O juurde?" TBAC-mudeli põhiküsimust "kas tehingut T tohib sooritada?" illustreerib Joonis 10.




**Joonis 10. Tehingupõhine pääsu reguleerimine**

Niisiis arvestab TBAC tehingu paiknemist kogu tööprotsessis. Pääsufunktsioonid paiknevad arvutiprotokollide, hajusrakenduste ja tööprotseduuride kriitilistes (tervikluse seisukohalt) punktides. Volitamine on dünaamiline. Kui RBAC-mudelis võib näiteks varustusjuhi roll kord looduna kirjutada välja kuitahes palju maksekorraldusi, siis TBAC-mudelis võib maksekorralduse väljakirjutamise volitamise tarvilik, kuid mitte piisav tingimus olla samuti varustusjuhi roll, lisaks sellele võidakse aga nõuda näiteks selle konkreetse maksekorralduse viseerimist.

ARPA tellitud projekti raames on loodud prototüüp. Katsesüsteem põhineb klient-server-arhitektuuril. Poliitika spetsifitseerimiseks kasutatakse graafilisi kõrgkeeli. Ilmselt võib lähitulevikus oodata TBAC-mudeli laiemat realiseerimist ärilistes turvatoodetes.

## 7.3 Pääsuarhitektuur

Pääsumudeli realiseerimiseks süsteemis kasutatakse põhiliselt kaht lähenemisviisi turvalise süsteemi arhitektuurile: kaitstud alamsüsteemi (*protected subsystem*) ja turvatuuma (*security kernel*) kontseptsiooni.

### 7.3.1 Kaitstud alamsüsteem

Selle kontseptsiooni kohaselt koostatakse süsteem kaitstud alamsüsteemidest. Kaitstud alamsüsteem on programmide ja andmete kogum. Andmete poole pöördumiseks võib kasutada ainult sellesse kogumisse kuuluvaid programme. Ka pääs programmide juurde on piiratud, neid saab käivitada ainult etteantud sisendpunktidest. Pääsu reguleerimiseks vajalik autentimine toimub alamsüsteemis.

Tulemus on suhteliselt paindlik, kuid pääsupoliitika ei ole selgelt lahutatud pääsumehhanismidest ning seetõttu tuleb pääsupoliitika muutmisel ka reguleerimismehhanismide programme muuta.

Selline arhitektuur sai alguse 70tel aastatel ning sobivate turvamehhanismide väljatöötamise vajadus käivitas operatsioonisüsteemide turvauuringud.

### 7.3.2 Turvatuum

Uuringud ja väljatöötuskogemused näitasid, et süsteemi turvalisuse tagamiseks ja tõestamiseks peavad turvamehhanismid olema väikesed ja asuma kaitstavatele objektidele võimalikult lähedal. Turvalisus peab olema üks süsteemilt nõutavaid põhifunktsioone, mis peab olema süsteemi madalaimatesse tasemetesse sisse ehitatud. Modelleerimise tulemusena jõuti ***viitemonitori*** (*reference monitor*) kontseptsioonini. Viitemonitor on mudelis abstraktne automaat, mis vahendab kõiki objektipöördusi. Viitemonitori realiseerib süsteemi turvatuum, mis koosneb riistvara-, püsivara- ja tarkvaraelementidest. Et turvatuum saaks vahendada kõiki objektipöördusi, peab ta toimima süsteemi madalaimal tasemel.

Turvatuuma teostusele esitatakse järgmised nõuded:

1. ***Täielikkus***. Turvatuum peab vahendama pääsu kõigi objektide juurde.
2. ***Isolatsioon***. Turvatuum peab olema kaitstud kõrvalise sekkumise eest.
3. ***Verifitseeritavus***. Turvamehhanismide vastavus turvapoliitikale peab olema tõestatav.

Tavaliselt saavutatakse täielikkus ja isolatsioon mõningase aparatuurse toega (riistvara erinevate tööviisidega kasutaja tasemel ja halduri tasemel). Verifitseeritavus saavutatakse formaliseerimise ja matemaatilise tõestusega.

Turvatuuma tõhusus sõltub täielikkuse ja isolatsiooni loomise edukusest.

Turvatuuma arhitektuuri spetsifitseeris 1985. a. USA kaitseministeeriumi standard TCSEC. Viitemonitori kontseptsiooni on suuremas või vähemas ulatuses realiseeritud uuemates operatsioonisüsteemides, sealhulgas Windows NT-s.

Viitemonitori edasiarendused kulgevad pääsuhalduse hõlbustamise, automatiseerimise ja paindlikkuse suunas. Selliste tööde hulka kuulub näiteks S. Jajodia, P.Samarati jt unifitseeritud multipoliitiline raamstruktuur, mida realiseerib nn paindlik volitushaldur (*flexible authorization manager, FAM*).

## 7.4 Pääsu reguleerimise standardne raamstruktuur

ISO/IEC 10181-3:1996 ja sellega identne ITU-T X.812 määratlevad avatud süsteemide OSI-mudeli kontekstis pääsu reguleerimise üldise raamstruktuuri, mis peaks hõlmama kõiki võimalikke pääsumehhanisme, -poliitikaid ja mudeleid. Ehkki massiivne OSI-mudel on eriti TCP/IP-võrkude arengu ja muude asjaolude tõttu minetanud oma aktuaalsuse (OSI-alased tööd on ISOs peatatud) ning standardi koostajad nendivad teabelises lisas mõningaid raskusi tegelike seniste pääsupoliitikate liigitamisel ISO struktuuri lahtritesse, annab see standard kasuliku metoodilise aluse pääsumehhanismide hindamiseks, võrdlemiseks ja konstrueerimiseks. Halvemini on raamstruktuur kohandatav uuematele infovoo reguleerimisel põhinevatele pääsusüsteemidele.

### 7.4.1 Üldistatud alusmudel

Pääsu reguleerimine hõlmab olemeid, mis võivad olla
- füüsilised (nt tegelikud süsteemid),
- loogilised (nt OSI-kihi olemid, failid, organisatsioonid, ettevõtted),
- inimkasutajad.

Üldmudeli olulisemad elemendid on järgmised.

***Sihtolemid*** (*target*) on pääsu reguleerimisega kaitstavad objektid

***Initsiaatorid*** (*initiator*) on olemid (inimesed või funktsionaalüksused), mis taotlevad juurdepääsu teistele olemitele, väljastades ***pääsutaotlusi*** (*access request*).

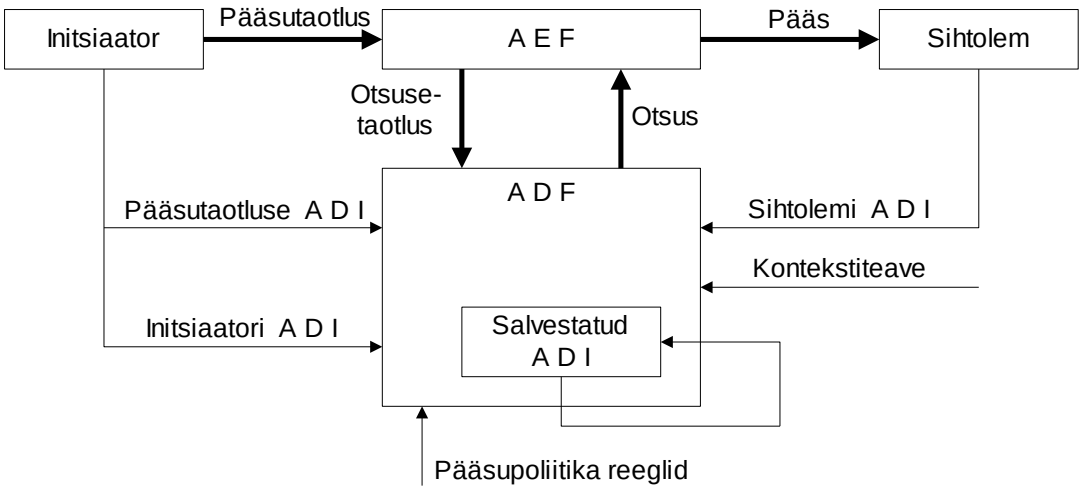
***Pääsureguleerimisteave*** (*access control information*, ACI) on olemispetsiifiline:
- initsiaatori ACI (isiku identsus, grupi/rolli identifikaatorid, tundlikkus- ja terviklusmärgised jm),
- sihtolemi ACI (olemi identsus, tundlikkus- ja terviklusmärgised, konteineri identifikaator jm),
- pääsutaotluse ACI (operatsiooni klass, operatsiooniks vajalik terviklusaste, andmetüüp jm),
- pääsutaotluse operandi ACI (tundlikkus- ja terviklusmärgised jm),
- kontekstiteave (lubatav periood, marsruut, asukoht, süsteemi olek, autentimistugevus, käimasolevad teiste initsiaatorite pöördused).

***Pääsuotsustusteave*** (*access control decision information*, ADI) on pääsuotsustusfunktsioonile kättesaadavaks tehtud ja pääsuotsuse tegemiseks kasutatav osa pääsureguleerimisteabest (ACI), erijuhul kogu ACI.

***Pääsuotsustusfunktsioon*** (*access control decision function*, ADF) on spetsialiseeritud funktsioon, mis teeb pääsuotsuseid, rakendades pääsupoliitika reegleid pääsutaotlusele, pääsuotsustusteabele ja taotluse kontekstile.

***Pääsutäiturfunktsioon*** (*access control enforcement function*, AEF) on spetsialiseeritud funktsioon, mis moodustab osa pöördusteest initsiaatori ja sihtolemi vahel ning realiseerib pääsuotsustusfunktsiooni tehtud otsuse.

Mudeli põhielementide vahelisi seoseid ja infovooge illustreerib Joonis 11. Üldstruktuurilt vastab see skeem Joonis 11 olevale, erinedes detailiseeringu ja olemite üldisusastme poolest.




**Joonis 11. Pääsu reguleerimise üldskeem**

Pääsu reguleerimise protsessid jagunevad olemuselt kaheks:

1) talitluslikud:
- pääsuotsustusteabe (ADI) andmine pääsuotsustusfunktsioonile (ADF),
- pääsu reguleerimise funktsioonide (ADF, AEF) sooritamine;

2) halduslikud:
- pääsupoliitika esituse (tüüp, süntaks, väärtused) kehtestamine,
- pääsureguleerimisteabe (ACI) esituste (tüüp, süntaks, väärtused) kehtestamine,
- ACI paigutus elementidele (initsiaator/sihtolem/pääsutaotlus),
- ACI sidumine elementidega (tervikluse tagamine jne),
- ACI modifitseerimine,
- ACI tühistamine.

### 7.4.2 Pääsupoliitikad

ISO/IEC 10181-3 viitab standardile ISO 7498-2 (= CCITT X.800), mis liigitab poliitikad kahte klassi: reeglipõhisteks ja identsuspõhisteks. Standard nendib, et tegelikes süsteemides rakendatakse nende kahe klassi kombinatsioone.

***Reeglipõhised*** poliitikad on mõeldud kohaldamiseks kõigile pöördustaotlustele, sõltumatult initsiaatorist ja sihtolemist. Ühe alamklassi moodustavad märgenditega määratletavad poliitikad, mille kohaselt initsiaatoritele ja sihtolemitele kinnistatakse turvamärgendid ning pääsuotsused põhinevad initsiaatori ja sihtolemi märgendite võrdlusel (vrd 7.2.3).

***Identsuspõhised*** poliitikad määratletakse reeglitena, mis on spetsiifilised individuaalsele initsiaatorile, initsiaatorite grupile, initsiaatorite nimel toimivatele olemitele või teatavat rolli kandvale individuaalsele olemile või olemitüübile. Gruppe ja rolle saab rakendada hierarhilistena.

***Mitmeinitsiaatorilised*** poliitikad võimaldavad pääsu ainult teatavale initsiaatorite kombinatsioonile (nt direktorile koos pearaamatupidajaga).

Ükskõik millist tüüpi pääsupoliitikat võib mõjutada ***kontekst***. Tegelikult saab kogu poliitika määratleda kontekstireeglitega (vrd 7.2.6). Poliitika võib defineerida ka sihtolemite ***granulaarsusastme*** ning olla igal granulaarsustasemel erinev. Lisaks sellele kehtestab poliitika ***pärilusreeglid*** elementide kopeerimise, modifitseerimise või ühendamise puhuks, ***ülimusreeglid*** pääsureeglite vaheliste vastuolude

lahendamiseks ning ***vaikereeglid***, mis määravad, kas vaikimisi on pääs lubatud või keelatud (vrd 7.2.2, kinnine ja lahtine poliitika)

Poliitikate haldusest lähtudes on poliitikaid kolme liiki:

1) ***püsipoliitikad*** kehtivad alati ja neid ei saa muuta (nad võivad näiteks olla süsteemi jäigalt sisse ehitatud);

2) ***halduri kehtestatavad poliitikad*** kehtivad alati ja neid saavad muuta ainult vastavate volitustega isikud;

3) ***kasutajate valitavad poliitikad*** kehtestatakse initsiaatori või sihtolemi nõudel ning neid kohaldatakse ainult seda initsiaatorit või sihtolemit (sh nende ressursse) puudutavatele pöördustaotlustele.

Koostoimes olevate turvadomeenide vahelisi piire ületavate pääsutaotluste korral tuleb üldjuhul sooritada poliitikateisendusi, sest naaberdomeenides võivad poliitikad või nende atribuudid (pääsuteabe esitusviisid, rollid ja nende atribuudid jms) erineda. Näiteks võidakse kõik privaatvõrku kuuluvad üksikkasutajad teisendada ühisesse avaliku võrgu kasutaja rolli.

### 7.4.3 Pääsumehhanismid

Pääsumehhanism koosneb pääsu reguleerimise skeemist ja abimehhanismidest ADI-teabe andmiseks sellele skeemile. Standard loetleb neli tüüpilist pääsu reguleerimise skeemi (lisaks loetletud tunnustele eeldavad nad ka sobivat tüüpi poliitikat):

1) ***volipõhises*** skeemis (*capability scheme*) moodustavad initsiaatoriga seotud ACI-teabe sihtolemi identsusest ja operatsioonitüübist koosnevad paarid, sihtolemiga seotud ACI-teabe aga sihtolemi identsus;

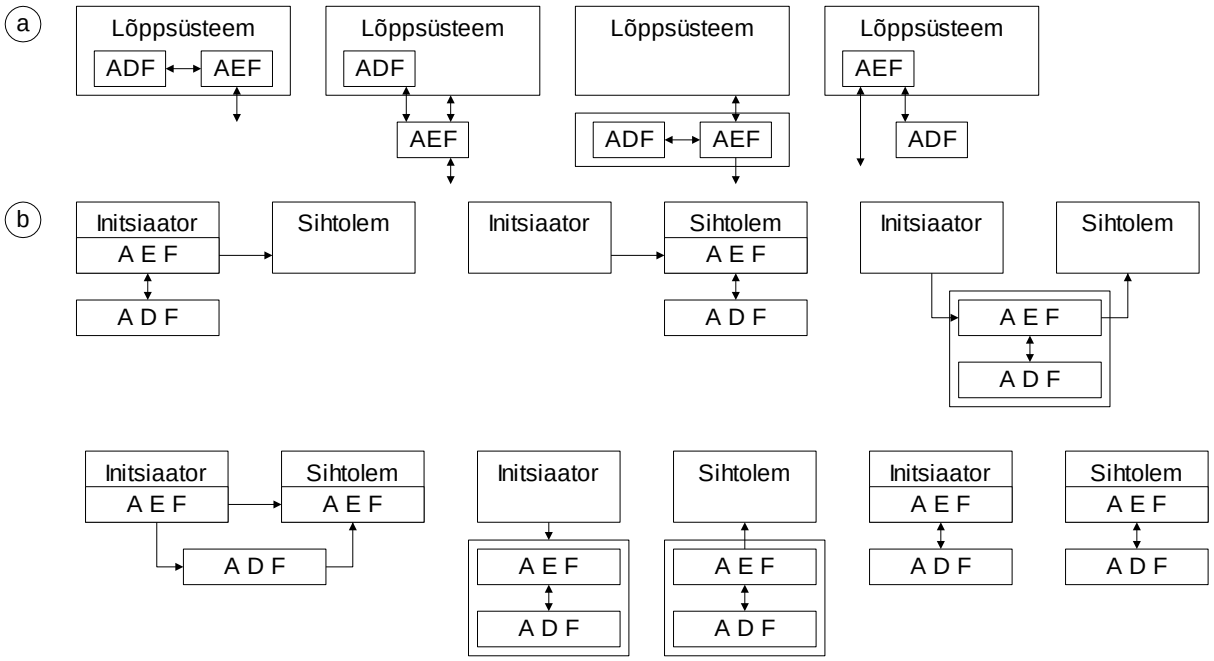
2) ***märgendipõhises*** skeemis (*label-based scheme*) moodustavad initsiaatoriga seotud ja sihtolemiga seotud ACI-teabe vastavalt initsiaatori ja sihtolemi turvamärgendid (vt 7.2.3);

3) ***pääsuloendiga*** skeemis (*access control list scheme*) on initsiaatoriga seotud ACI-teabeks initsiaatori identsus, sihtolemiga seotud ACI-teave aga koosneb initsiaatori identsuse ja operatsiooni tüübi paaridest;

4) ***kontekstipõhises*** skeemis (*context-based access control scheme*) määravad pääsu kontekstiteavet puudutavad reeglid (need võivad sisalduda ka muudes skeemides). Kontekstiteave võib olla osa initsiaatori, pääsutaotluse või sihtolemiga seotud ACI-teabest, kuid ta võidakse edastada pääsuotsustusfunktsioonile (ADF) ka sõltumatult muust ACI-teabest.

Ülaltoodud liigitus põhineb pääsuteabe tüübil ja paigutusel, lisaks sellele varieeruvad mehhanismid veel funktsioonide AEF ja ADF paigutuse ja komponentide vahelise interaktsiooni struktuuri poolest (vt Joonis 12).

Kasutatavate ***abimehhanismide*** tüüp sõltub pääsu reguleerimise skeemist ja pääsupoliitikast. Näiteks kui pääs põhineb initsiaatori identsusel, võidakse kasutada autentimist ning pääsusertifikaate või volitustõendeid. Pääsumehhanismi enda kaitseks võidakse kasutada terviklust ja konfidentsiaalsust kaitsvaid mehhanisme. Pääsu reguleerimisega on tavaliselt tihedalt seotud ka turvarevisjoni mehhanismid ning pääsuotsustusfunktsioon võib kasutada neis salvestatud teavet. Äriliste kaugteenuste puhul on pääsumehhanismidega seotud rahalise arvestuse ja arvelduse mehhanismid




**Joonis 12. Pääsufunktsioonide paigutuse (a) ja interaktsiooni (b) variandid**

### 7.4.4 Pääsuteenuse lühiülevaade

Lühiülevaate pääsu reguleerimise teenuse olemitest, funktsioonidest, teabest ja protsessidest annavad tabelidTabel 3,Tabel 4 jaTabel 5, mis põhinevad ISO/IEC 10181-3 teabelisel lisal G (ei ole standardi lahutamatu osa).

**Tabel 3. Pääsu reguleerimise teenuse elemendid**

| | |
|---|---|
| Põhiolemid | Initsiaator, sihtolem |
| Funktsioonid | Pääsuotsustusfunktsioon (ADF)<br>Pääsutäiturfunktsioon (AEF) |
| Teave | Pääsureguleerimisteave (ACI)<br>Pääsuotsustusteave (ADI)<br>Kontekstiteave<br>Poliitikareeglid |
| Olemite eesmärk | Interpreteerida teavet nii, et initsiaatorid saaksid juurdepääsu sihtolemitele ainult volitatud viisil |

**Tabel 4. Pääsu reguleerimise teenuse teave**

| | |
|---|---|
| Domeenikeskuse hallatavad andmeelemendid | Identifikaatorid (SDA, initsiaator, sihtolem, suhtluspoliitika, grupid, rollid)<br>ACI valimise kriteeriumid<br>Kehtivusaeg<br>Tundlikkusmärgistus<br>Terviklusmärgistus |
| Operatsioonides kasutatav teave | ACI/ADI (initsiaator, sihtolem, pääsutaotlus, operand, vahetus, kontekst, salvestatud)<br>Pääsuloend<br>Voli<br>Märgend<br>Pääsusertifikaat<br>Volitustõend |
| Juhtteave | Ajavahemik<br>Süsteemi olek<br>Pääsupoliitika esitus<br>Autentimise tugevus (vt 8.1)<br>Sidemarsruut |

**Tabel 5. Pääsu reguleerimise teenuse protsessid**

| | Haldusprotsessid | Talitlusprotsessid |
|---|---|---|
| **Olemid:** | | |
| Initsiaator | | Kaugkeskuse tuvastus<br>Turvalise suhtluspoliitika kehtestamine<br>ACI võtt<br>ACI genereerimine<br>ADI tühistamine |
| Sihtolem | | ACI võtt<br>ADI tühistamine |
| Turvadomeeni keskus (SDA) | ACI installeerimine<br>ACI muutmine<br>ACI tühistamine<br>ADI tühistamine<br>ACI loetlemine<br>Komponendi desaktiveerimine<br>Komponendi taasaktiveerimine | |
| **Funktsioonid:** | | |
| ADF | | ACI võtt<br>ACI verifitseerimine ja ADI tuletamine<br>Kontekstiteabe võtt<br>Pääsu otsustamine |

# 8 AUTENTIMINE

## 8.1 Autentimisprotsess

### 8.1.1 Põhimõisteid

***Autentimine*** (*authentication*) on olemi (isiku, grupi, protsessi, sõnumi vms) väidetava identsuse automaatne verifitseerimine. Autentimist rakendatakse peamiselt

- pääsu reguleerimise mehhanismides (taotleja autentimine, vt 7. ptk),
- kommunikatsiooni tervikluse tagamise mehhanismides (andmeallika autentimine, partneri autentimine).

***Identimine*** (identifitseerimine, *identification*) on autentimisele eelnev või autentimisega põimunud protsess, mille käigus subjekt esitleb end süsteemile, esitades oma ühese identifikaatori, näiteks tegeliku nime, kasutajanime, kontonumbri. Identimine on ressurssi kaitsva turvamehhanismi jaoks esmane ja tarvilik eeldus kasutamisvolituste kontrollimiseks mingite lubavate või keelavate tingimuste alusel, autentimine on täiendav turvaprotsess, mis peab vältima teeskluse, st subjekti esinemise teise subjektina.

Turvakirjanduses on identimisel ka muid tähendusi: 1) üheste identifikaatorite (nt kasutajanimede) kinnistamine olemitele; 2) passiivsete objektide tuvastamine identifikaatorite järgi; 3) identimine ülaltoodud tähenduses (mida kasutab nt NIST) koos autentimisega; jt. USA infoturbe föderaalkriteeriumid (FC ITS) annavad üsna üldise määratluse: protsess, mis võimaldab infotehnilisel tootel mingit olemit tuvastada.

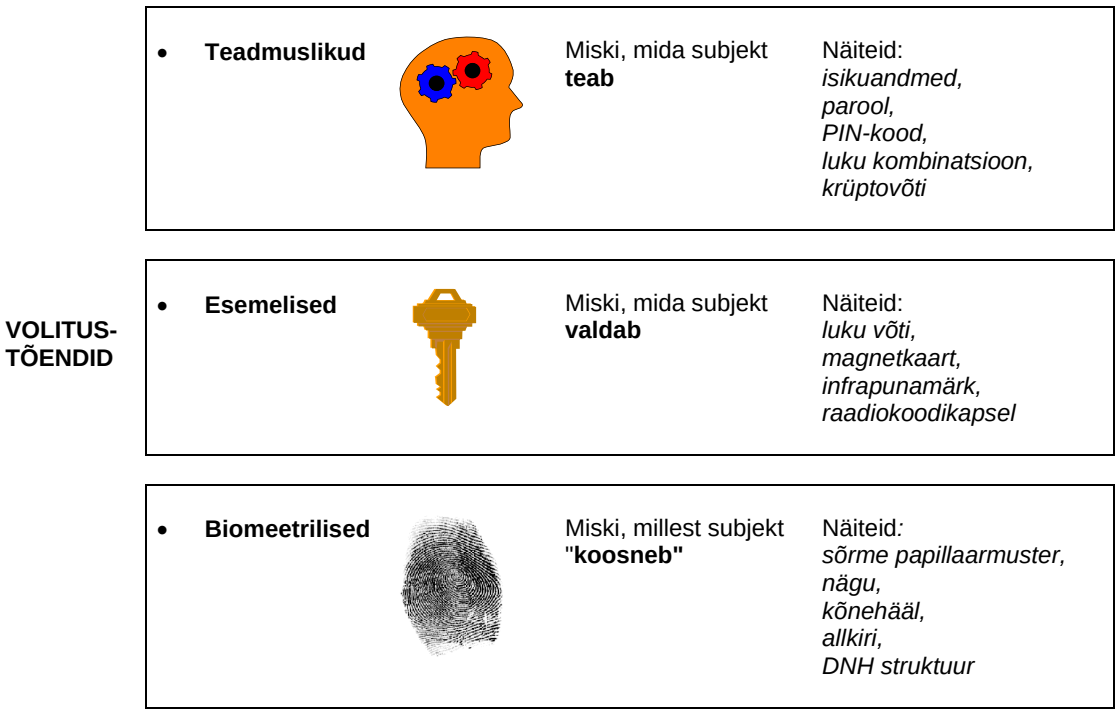
***Volitustõend*** on subjekti identsust tõendavat informatsiooni sisaldav struktuur (infokogum ja/või infokandja), mille alusel toimub väidetava identsuse automaatne verifitseerimine. Niisiis ei tohi volitustõend olla kergesti võltsitav. Volitustõend võib oma olemuselt olla teadmuslik, esemeline või biomeetriline (vt Joonis 13). Üksikasjalikumalt käsitletakse volitustõendeid jaotistes 8.2, 8.3 ja 8.4, volitustõendite võrdluse üldkriteeriumid on loetletud jaotises 8.1.5.

Selline volitustõendite traditsiooniline liigitus põhineb inimsubjekti autentimisel ja isikutuvastusel. Tegelikult on vahetult inimesega seotud ainult biomeetrilised tunnused, muude tõendite puhul kontrollitakse tegelikult inimesest suhteliselt sõltumatuid infokogumeid. Universaalsemate autentimismudelite loomise võimaldamiseks ja autentimisprotsesside formaliseeritud kirjeldamise hõlbustamiseks abstraheerib ja üldistab autentimise standardne raamstruktuur (ISO /IEC 10181-2) autentimisprotsessi, tuues muuhulgas sisse printsipaali mõiste.

***Printsipaal*** (*principal*) on olem, mille identsust saab autentida. Printsipaaliks võib olla ka (lõppkokkuvõttes inimsubjekti esindav) andmekogum, protsess, tehniline süsteem jne. Rakendatav autentimismeetod sõltub printsipaali tüübist. Printsipaalide liigitusviis sõltub konkreetsest olukorrast. Raamstruktuuri standard esitab järgmise liigitusnäite:

| **Printsipaali tüüp:** | **Autentimismeetod:** |
|---|---|
| Passiivsete (nt biomeetriliste) tunnusomadustega | Passiivsete tunnusomaduste mõõtmine |
| Infovahetus- ja -töötlusvõimeline | Kompleksne dialooghindamine |
| Infotalletusvõimeline | Salvestatud saladuse (nt parooli) võrdlus |
| Ühese kinnisasukohaga | Asukoha määramine |

Selle liigituse puhul vastab inimprintsipaal kolmele esimesele tüübile.




**Joonis 13. Volitustõendid**

Volitustõendites rakendatavale kolmele autentimisalusele (teadmus, esemesse talletatud teave, biomeetriline teave) lisab standardne raamstruktuur ülalöeldust lähtudes veel kaks:

- usaldusväärne ***kolmas osapool*** ("notariaalne" teave, vt 8.1.3),
- ***kontekstteave***, muuhulgas printsipaali aadress/asukoht (nt rahaautomaadi asukoht).

Turvalisuse tõstmiseks võidakse korraga rakendada mitut autentimisalust. Levinuim on esemelise (magnet- või kiipkaart) ja teadmusliku (parool, PIN-kood) volitustõendi kombinatsioon.

Tunduvalt võimaldab turvalisust tõsta krüptograafiliste meetodite rakendamine autentimisprotsessis, seetõttu nimetatakse vastavat protsessi ***tugevaks autentimiseks***, vastandina traditsioonilisele nõrgale ehk ***lihtautentimisele***. ISO 9594-8/CCITT X.509 määratleb tugeva autentimise nii: autentimine krüptograafiliselt tuletatud mandaadi abil. Tugeva autentimise protokolle ja protsesse kirjeldab 11. peatükk.

Lisaks võimalikult usaldusväärsetele autentimisalustele ja verifitseerimisskeemidele peab tõhus autentimissüsteem sisaldama ka autentimisteabe halduse vahendeid ning meetmeid selle teabe kaitseks rünnete eest andmevahetuse käigus. Näiteks lisatakse tavalisele autentimisteabele ***ajahetkega seotud parameetreid*** (järjenumbrid, juhuarvud, ajatemplid jm), mis aitavad tõrjuda autentimisteabe pealtkuulamisel, salvestamisel ja taasesitusel põhinevat teesklusrünnet (*masquerade*).

Eriti võrgu kaudu toimuvas suhtluses on oluline veenduda selles, et mingi andmekogum on pärit subjektilt, kes väidab end ta loojaks või saatjaks, ning et mõlemad võrgusuhtluse osapooled on need, kes nad väidavad end olevat.

***Andmeallika autentimise*** vahendid on peamiselt krüptograafilised – digitaalsignatuur, sõnumi krüpteerimine ja krüptograafilised kontrollkoodid, sh sõnumilühend (*message digest*). Ühtlasi kaitsevad need vahendid andmeterviklust.

***Partnersubjekti autentimine*** põhineb asümmeetrilisel krüptograafial ja digitaalsignatuuridel.

Kõigile krüptograafial põhinevatele autentimismehhanismidele lisandub turbesüsteemis neid toetav võtmehalduse infrastruktuur oma spetsiifiliste vahendite ja protsessidega.

### 8.1.2 Autentimisprotsessi faasid

Standardne raamstruktuur eristab alljärgnevaid faase. Need ei ole tingimata ajaliselt eraldatud, st võivad kattuda. Faaside järjestus võib eri autentimisprotsessides erineda. Igas protsessis ei tarvitse olla kõiki loetletud faase.

***1. Installeerimine***. Määratletakse taotluse autentimisinformatsioon (AI) ja verifitseerimise AI.
***2. Autentimisinformatsiooni muutmine***. Printsipaal või haldur algatab taotluse AI ja verifitseerimise AI muutmise (näiteks vahetatakse parool).
***3. Jaotamine***. Olemitele (nt taotlejaile või verifitseerijaile) jaotatakse verifitseerimise AI, näiteks autentimissertifikaatidena. Jaotamine võib toimuda enne edastusfaasi, selle ajal või pärast seda.
***4. Hankimine***. Selles faasis võib taotleja või verifitseerija saada informatsiooni, mis on vajalik spetsiifilise vahetus-AI genereerimiseks ühe autentimisjuhu (nt seansi) tarbeks. Võib sisaldada suhtlust kolmanda osapoolega. Näiteks võib taotleja või verifitseerija hankida võtmejaotuskeskusest autentimissertifikaadi.
***5. Edastus***. Taotleja ja verifitseerija vahel edastatakse vahetus-AI.
***6. Verifitseerimine***. Vahetus-AI kontrollimine verifitseerimis-AI abil. Kui olem ei ole võimeline ise verifitseerima, võib ta võtta ühendust kolmanda osapoolega, jättes verifitseerimise selle hooleks; sellisel juhul saadab kolmas osapool tagasi kinnitava või eitava vastuse.
***7. Desaktiveerimine***. Seatakse sisse olek, mille kestel printsipaali ajutiselt ei saa autentida.
***8. Taasaktiveerimine***. Kõrvaldatakse desaktiveerimisega seatud olek.
***9. Desinstalleerimine***. Printsipaal kõrvaldatakse printsipaalide kogumist.

Faasid 4, 5, 6 sisalduvad otseselt autentimisseansis, ülejäänud faasid on seotud autentimissüsteemi haldusega.

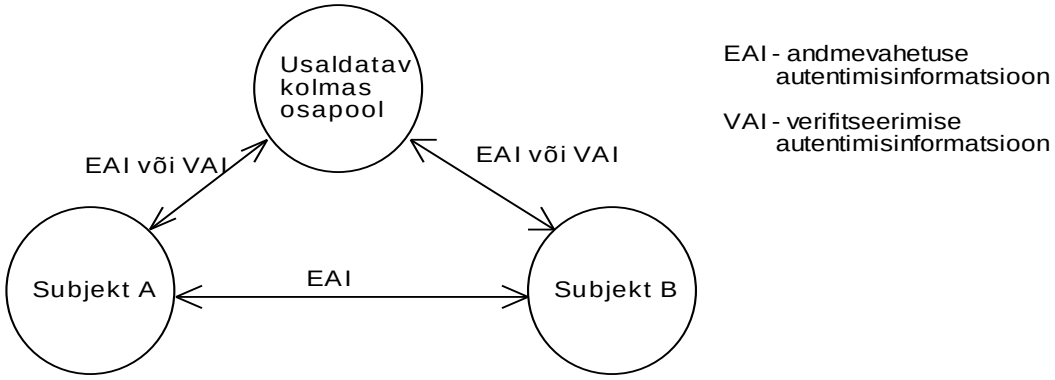
### 8.1.3 Autentimisprotsesside põhitüübid

Osapoolte arvu ja autentimissuundade järgi jagunevad autentimisprotsessid ühe-, kahe- ja kolmepoolseteks. Osapooli on autentimisprotsessis lihtsamal juhul kaks, näiteks pääsu taotleja ja autentimismehhanism või kaugsõnumi lähetaja ja saaja. Turbetaseme tõstmiseks võidakse autentimisprotsessi lülitada kolmas, verifitseerimisfunktsioonidega osapool.

***Ühepoolne*** autentimine leiab rakendamist lihtsamatel juhtudel ja ainult lokaalsüsteemides, näiteks pääsu reguleerimise mehhanismides. Mehhanism (nt magnetkaardiriider või operatsioonisüsteemi sisselogimismoodul) kontrollib pääsu taotleja autentsust, taotleja aga ei kontrolli mehhanismi autentsust. Protsessi nõrkus on selles, et taotleja usaldab mehhanismi "ametlikku ilmet", ehkki logimisakna võis ekraanile tuua mitte turbesüsteem, vaid trooja hobune.

***Kahepoolne*** autentimine aitab vältida mitmeid ründeid, milleks annab võimalusi hajus võrgukeskkond, eeskätt teesklust (volitatud subjektina esinemist) ja salgamist (sõnumi lähetamise või saamise eitamist).

***Kolmepoolne*** autentimine sisaldab usaldatavat kolmandat osapoolt, nii et üldjuhul on autentimisprotsessi üldmudel selline, nagu Joonis 14. Mõlema autentimissubjekti poolt usaldatava kolmanda osapoole juures registreeritakse andmed, nii et hiljem saab tõendada andmete sisu, allika, kohalejõudmise aja ning muude tunnusomaduste õigsust. Sellele lihtsale struktuurile saab vastavalt vajadustele rajada mitmeid erinevaid autentimisskeeme (vt 11. ptk). Kõiki lihtsamaid skeeme võib vaadelda selle skeemi erijuhtudena.




**Joonis 14. Autentimisprotsessi üldmudel (ISO/IEC 9798-1 ja ISO/IEC 10181-2)**

ISO/IEC 10181-2 raamstruktuur esitab rea sellisel üldmudelil põhinevaid detailiseeritud autentimisskeeme. Kolmanda osapoole rakendamise viisi järgi eristab see standard kolme autentimissuhtluse mudelit.

***Vahendusautentimise*** (*in-line authentication*) korral on kolmandal osapoolel autentimisteabe vahendaja roll, kogu autentimisteabe vahetus taotleja ja verifitseerija vahel kulgeb läbi tema.

***Sidusautentimise*** (*on-line authentication*) puhul osaleb kolmas osapool aktiivselt autentimisteabe vahetuses, kuid ei paikne vahetult selle teabe vahetuse traktis. Taotleja nõudel võib ta genereerida autentimisteavet ja abistada verifitseerijat. Ta võib genereerida autentimissertifikaate autentimisprotsessi käigus. Sellise kolmanda osapoole näited on sidusad autentimisserverid ja võtmejaotuskeskused.

***Vallasautentimise*** (*off-line authentication*) puhul ei osale kolmas osapool igas autentimisseansis. Ta genereerib ja jaotab autentimissertifikaate enne tegelikku autentimisandmete vahetust. Sellise osapoole näide on tavaline vallas-sertifitseerimiskeskus.

### 8.1.4 Autentimismehhanismide klassid

ISO/IEC 10181-2 määratleb autentimismehhanismide klassid 0, 1, 2, 3, 4 ja alamklassid 4a, 4b, 4c, 4d ning jätab võimaluse vajaduse korral luua uusi klasse. ISO klasside turbenõuetest ründetüüpide tõrje järgi annab mõningase ettekujutuse Tabel 6.

**Tabel 6. Autentimismehhanismide turbenõuded**

| **Klass** | **0** | **1** | **2** | **3** | **4** |
|---|---|---|---|---|---|
| **Rünnete tõrje:** | | | | | |
| Avalikustamine | - | x | x | x | x |
| Taasesitus teisele verifitseerijale | - | - | x | - | x |
| Taasesitus samale verifitseerijale | - | - | - | x | x |
| Infopüük teesklusega | (x) | x | x | x | x |

Klassile 0 vastab nõrk autentimine, klasside 1, 2, 3, 4a ja 4b puhul on jäetud krüptograafiliste vahendite kasutamine lahtiseks, alamklassid 4c ja 4d rakendavad tugevat autentimist.

### 8.1.5 Volitustõenditele esitatavad nõuded

Autentimise tõhusus sõltub autentimisprotokolli ja selle aluseks oleva volitustõendi tõhususest.

Autentimiseks kasutatava volitustõendi hindamise ja valimise hõlbustamiseks on mitmete rahvuslike ja harukondlike standarditega püütud ühtlustada volitustõendite üldisi võrdlusaluseid ja hindamiskriteeriume. Tüüpilisi juhiseid esitavad näiteks USA Rahvusliku Standardibüroo (NBS) soovitused ( *FIPS Publication* 48, 1977), mille sisu võib kokku võtta järgmise 12 kriteeriumi kujul:

1) pettusekindlus
2) tõendi võltsimise keerukus
3) möödumiskindlus
4) verifitseerimise kestus
5) mugavus kasutajale
6) autentimisvahendi ja ta kasutamise maksumus
7) autentimisvahendi eesmärgipärane liidestamine
8) volitustõendi asendamisele kuluv aeg ja vaev
9) arvutisüsteemilt nõutav verifitseerimistöötlus
10) töökindlus ja hooldatavus
11) autentimisvahendi kaitsmise kulud
12) jaotamise ja logistilise toe kulud

Need kriteeriumid peegeldavad volitustõendi kolme põhinäitajat:

- tõendi ***tugevust*** turbevahendina,
- ***vastuvõetavust*** kasutajale,
- autentimismehhanismi ***maksumust***.

Kõik need näitajad on olulised, kuid alustada tuleb turvapoliitikast tulenevate turvateenuste andmiseks vajalikust turbetugevusest. Turvaomadused sõltuvad volitustõendi tüübist, selle tüübi konkreetsest teostusest, ta kasutamisviisist, haldusest ning integratsioonist teiste turvameetmetega.

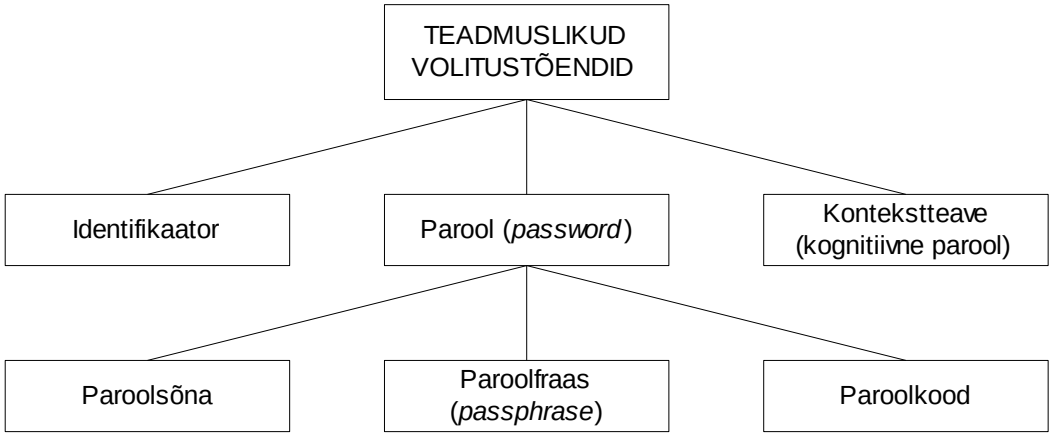
Volitustõendite põhitüüpe käsitlevad üksikasjalikumalt jaotised 8.2, 8.3 ja 8.4.

## 8.2 Teadmuslikud volitustõendid

### 8.2.1 Põhiliigid

Teadmuslik volitustõend on olemi identsuse verifitseerimist võimaldav infokogum, mida ideaaljuhul teavad ainult autentimise osapooled. "Teadmist" tuleb sõnasõnalt mõista ainult inimsubjekti puhul ja sedagi mitte alati: teave võib olla salvestatud subjektile kuuluvasse tehismällu ja subjekt ise ei tarvitse alati ta sisu otseselt teadagi.

Praktikas leiab volitustõendina rakendamist kolme liiki teave (vt Joonis 15). Lisaks spetsiaalselt turvaotstarbelisele (parool) on sageli otstarbekas ära kasutada ka sellist muu funktsiooniga teavet, mis on autentimise osapooltel niikuinii olemas ning mida peale nende võib teada ainult väike arv subjekte (identifikaatorid, kontekstteave).




**Joonis 15. Teadmuslike volitustõendite liigid**

Alljärgnevas vaatleme neid vahendeid turbe tugevuse kasvavas järjestuses.

#### *8.2.1.1 Identifikaator*

Identifikaatorite autentimisrakenduse tuntuim näide on üldised sisselogimisandmed. Sõltumata muude kaitsevahendite rakendamisest nõuavad operatsioonisüsteemid, võrgurakendused, haldusvahendid jms tööseansi alustamisel kasutajalt sisselogimist, st vähemalt kasutajanime, rollinime, kontonumbri vms sisestust. Need andmed moodustavad pääsu reguleerimisel kõige elementaarsema kaitsekihi, millele saab ehitada muid turvamehhanisme, näiteks parool- või biomeetrilise autentimise baasil. Autentimisvahendina on nad nõrgad, sest neid teab suhteliselt suur arv subjekte ja neid on üsna kerge ära arvata.

#### *8.2.1.1 Kontekstteave ehk kognitiivne parool*

See on subjekti taustandmetest valitud kogum, mis kinnitab tema identsust indiviidina või mingi rühma liikmena, kuid mida ei tea teised või rühmavälised subjektid.

Kontekstteavet on laialt kasutatud tavalises mitteautomaatses isikutuvastuses mitmesugustes konspiratiiv- ja teesklusolukordades. Tuntud näited on USA pesapalli edetabelite küsimine 2. maailmasõjas omade kätte vangi sattunud ameerika sõdureilt, keda kahtlustati kuulumises USA mundritesse riietatud SS-diversioonidiviisi; analoogiline situatsioon kordus Pärsia lahe sõjas vangistatud briti diversantidega, keda Iraagi vastuluure pidas ameeriklasteks (vahe oli oluline: Iraak kartis kogu NATO aktiivset sekkumist).

Autentimisprotsessis saab kontekstteavet kontrollida interaktiivse küsimustikuga. Sisselogimisel või kaitstud ressursi poole pöördumisel esitab süsteem pääsu taotlejale tema isikupärase teadmusega seotud küsimusi (nt tema algkoolidirektori nimi, vanaema silmade värv, lemmikautor jne). Usaldatavuse tõstmiseks võivad küsimused jätkuda ka tööseansi kestel. Kasutaja peab saama küsimusi süsteemi häälestamisel ise ette anda. "Õiged" vastused ei tarvitse olla kooskõlas tegelikkusega, sel juhul sarnaneb autentimisteave oma loomult parooliga.

Paroolkaitsega võrreldes on vahendi eeliseks parem mnemoonilisus. Peamine puudus on suurem ajakulu. Vahend ei sobi ka väga hinnaliste ressursside kaitseks, sest sel juhul tasuks rünne suunata volitatu taustandmete väljaselgitamisele. Kontekstteabe kasutamine tundub algeline ja ebapraktiline, kuid viimastel aastatel on hakanud selle vastu huvi tundma näiteks USA haldusasutused, kelle käsutuses on nagunii mitmesuguseid isikuandmeid, mida saaks kasutada autentimisteabena. Häid võimalusi pakuvad näiteks tervishoiuandmed, kinnisvaraandmed jms. Nende ärakasutamine autentimisteabe osana võimaldab suhteliselt väikeste lisakulutustega tugevdada autentimismehhanisme.

#### *8.2.1.3 Parool*

Parool on volitatud kasutaja käsutusse antud salastatud tekstijada, mis võib üldjuhul sisaldada lisaks suur- ja väiketähtedele ka numbreid ja erimärke. Parooli esitamine või täpsemalt, parooli teadmise tõestamine, on teadmuspõhise autentimise põhimeetod. Põhimõtteliselt saab paroolkaitse panna tugevuselt ületama esemelisi või biomeetrilisi volitustõendeid, kuid praktiliselt piirab seda asjaolu, et inimene ei suuda kuitahes pikki paroole meeles pidada.

Parooli levinuim vorm on ***paroolsõna***. Pikemat mitmesõnalist parooli nimetatakse praktikas ***paroolfraasiks***. Lühema, tavaliselt ainult või peamiselt numbritest koosneva ***paroolkoodi*** tuntud näited on PIN-kood ja koodluku või valvesignalisatsiooni kood.

Paroolkaitset võib rakendada kogu süsteemi hõlmavana (st lisades ta sisselogimisprotseduurile) ja/või üksikressursside (kataloog, fail, kirje väli, teatav rakendusprogramm jne) ning nendega seotud operatsioonide (nt kirjutuse) kasutamise reguleerimiseks. Lisaks sellele võib paroolidel olla erinev kehtivusulatus:

- ***rühmaparool*** on ühine mingi kasutajate rühma jaoks; kasutajate õigusi saab ka rühmaparooli puhul diferentseerida, näiteks kasutajaandmete ja ressursivolitusloendite abil;
- ***individuaalne parool*** on turvalisem ja ka paindlikum; lihtsam on korraldada mitmetasemelist pääsusüsteemi;
- ***seansiparool*** on igal süsteemi poole pöördumisel uus, sageli realiseeritakse seansiparooli sisestus aparatuurse parooligeneraatori abil.

Paroolkaitse on üks kõige levinumaid ja kesksemaid andmeturbevahendeid. Lisaks pääsu reguleerimisele rakendatakse teda muudeski autentimisprotseduurides. Detailsemalt käsitlevad paroolkaitset järgmised alajaotised.

### 8.2.2 Parooli tugevuse mõõt: parooli entroopia

Intuitiivselt võib pikemat parooli pidada lühikesest tugevamaks, st raskemini äraarvatavaks. Pikemat parooli on aga raskem meeles pidada, mistõttu on pikemad paroolid sageli tähendusega sõnad või väljendid, parooli omaniku isikuandmeid puudutav jne. See aga omakorda hõlbustab ründajal parooli ära arvamist. Parooli optimaalse pikkuse leidmiseks ja valiku üldkriteeriumide täpsustamiseks tuleb parooli turbeomadusi uurida kvantitatiivselt.

Kui ründaja teab, et levinud tavade järgi valitakse teatav (suhteliselt väike) hulk võimalikke paroole suurema tõenäosusega (sõnad, nimed, kuupäevad jne), hõlbustab see parooli äraarvamist. Täpsemalt selgitab parooli äraarvamise protsessi selle alljärgnev formaalne käsitlus informatsiooniteooria vahenditega.

Olgu $\boldsymbol{P}$ kõikvõimalike paroolide hulk. Ründaja jaotab selle hulga mõttes kaheks lõikumatuks osaks

$$\boldsymbol{P} = \boldsymbol{P}_1 \cup \boldsymbol{P}_2.$$

Oletagem, et ründaja teab, et parool on valitud hulgast $\boldsymbol{P}_1$ tõenäosusega $p_1$ ja hulgast $\boldsymbol{P}_2$ tõenäosusega $p_2$. On selge, et kui hulk $\boldsymbol{P}_1$ on suhteliselt väike ja tõenäosus $p_1$ suhteliselt suur, on ründajal kasulik oletada, et parool valiti hulgast $\boldsymbol{P}_1$. Näiteks paljudel juhtudel võib hulgaks $\boldsymbol{P}_1$ võtta pärisnimede hulga.

On selge, et hulka $\boldsymbol{P}_1$ kuulumise suurem tõenäosus ei tähenda üldse, et tegelik parool peaks sellesse hulka kuuluma. Kujutlegem mõttelist eksperimenti, milles ise oleme ründaja osas ja võime eksperimendi igal sammul mõtteliselt jaotada paroolihulka mistahes viisil, kusjuures me teame iga parooli esinemise tõenäosust. Oletagem, et jaotasime hulga $\boldsymbol{P}$ kaheks osaks $\boldsymbol{P}_1$ ja $\boldsymbol{P}_2$. Nüüd ei saa me kindlalt otsustada, millist hulka eelistada. Selleks kasutame abilist – ***oraaklit*** – kes teab alati, millisesse hulka tegelik parool kuulub, ja vastab meie igale küsimusele jah või ei. Kui oraakel vastab, et parool kuulub hulka $\boldsymbol{P}_1$, jagame hulga $\boldsymbol{P}_1$ omakorda mõtteliselt pooleks jne.

Parooli ***entroopia*** on nende küsimuste arvu matemaatiline ootus, mida on vajalik esitada oraaklile tegeliku parooli kindlakstegemiseks. Entroopia ühik on ***bitt***.

Parooli entroopiat võib seega vaadelda kui parooli tegeliku väärtuse teadmisest puudu jäävat informatsioonihulka. On võimalik näidata, et parooli entroopia on ligikaudu[1] arvutatav (nn Shannoni entroopia) avaldisega:

$$H(\boldsymbol{P}) = -[\, p(P_1)\cdot \log p(P_1) + p(P_2)\cdot \log p(P_2) + \ldots + p(P_N)\cdot \log p(P_N)\,],$$

kus $\boldsymbol{P}=\{P_1,\ldots,P_N\}$ on kõikvõimalike paroolide hulk.

Seega on turvalisuse mõttes kasulik valida parool nii, et poleks võimalik paroolide hulga selline mõtteline poolitus, kus ühel poolel on vähe paroole ja suhteliselt suur sinna kuulumise tõenäosus. Et tõenäosus $p_1$ on võrdne kõigi hulka $P_1$ kuuluvate üksikute paroolide tõenäosuste summaga, tasub valida parool nii, et mistahes kahe üksiku parooli tõenäosused oleksid võrdsed, st võrduksid $1/N$, kus $N$ on kõikvõimalike paroolide arv.

### 8.2.3 Parooli pikkus ja eluiga

Eeldame, et kõigi üksikparoolide tõenäosused on võrdsed ja võrduvad $1/N$. Kõikvõimalikke paroole on seega kokku $N$ tükki. Olgu $A$ kasutatava märgistiku maht ja $M$ märkide arv paroolis (parooli pikkus), siis

$$N = A^M.$$

**Näide 1**. Kui parool koosneb neljast numbrimärgist (0–9), on erinevate võimalike paroolide arv

$$N = A^M = 10^4 = 10{,}000.$$

**Näide 2**. Kui parool koosneb kuuest tähest (A–Z, kokku 26 märki), siis

$$N = 26^6 = 308,\, 915,\, 776.$$

Kui meil (ründajana) on võimalus juhuslikul viisil läbi proovida $K$ parooli, on õige parooli äraarvamise tõenäosus

[1] Tegelikult nimetatakse just avaldist H(P) parooli entroopiaks, kuid erinevusel pole suurte hulkade korral olulist tähtsust. Näiteks kolmeelemendilise paroolihulga ja ühtlase tõenäosusjaotuse korral saame oraakli abil defineeritud entroopiaks 5/3=1.66… ja Shannoni entroopiaks ln 3/ln 2=1.58...

$$P = K / N.$$

Kui ajaühikus on võimalik proovida $R$ erinevat parooli, on parooli eluea $T$ jooksul võimalik proovida $R \cdot T$ parooli, seega kirjeldab parooli eluea $T$, lubatava äraarvamise tõenäosuse $P$ ja parooli pikkuse M seost avaldis

$$\boldsymbol{P = R \cdot T / A^M.}$$

**Näide 3**. Leida vajalik parooli pikkus, kui soovime, et parooli eluiga oleks 6 kuud ja et selle aja jooksul oleks parooli leidmise tõenäosus alla $10^{-6}$ arvestades, et minutis on võimalik läbi proovida 8,5 parooli ja et kasutatavas tähestikus on 26 märki.
Lahendus: 6 kuud on 183 päeva ja 8,5 parooli minutis teeb 12240 parooli päevas. Seega

$$N = T \cdot R / P = (183 \cdot 12240)/10^{-6} = 2{,}23992 \cdot 10^{12} \text{ parooli}$$

ja seetõttu

$$M = \log N / \log A = 8{,}72 \text{ märki.}$$

Seega on vajalik märkide arv 9.

Parooli pikkuse ja eluea seost illustreerib järgmine tabel, mis näitab parooli kõigi võimalike kujude läbiproovimisele kuluvat aega parooli käsitsi sisestamisel (ühe märgi sisestusajaks on võetud 1 s), sõltuvalt parooli pikkusest.

| Parooli pikkus | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Märgistik: 36 | 36 s | 21 min | 13 h | 19,5 d | 1,9 a | 69 a | 2484 a | 89456 a |
| Märgistik: 256 | 4 min | 18,2 h | 194 d | 136 a | 34816 a | | | |

Üks firma Bell sooritatud uuring näitas muuhulgas, et 0,5% paroolidest kujutasid endast ühtainsat ASCII-märki, 2,2% koosnes kahest ASCII-märgist, 14,1% – kolmest ASCII-märgist, 14,5% – neljast tekstimärgist, 21,5% – viiest tähest (ainult suur- või ainult väiketähed), 18,4% – kuuest väiketähest. Niisiis saab isegi käsitsi sisestamisel peaaegu 3% paroolidest üsna lühikese ajaga leida lihtsalt kõigi kombinatsioonide läbiproovimisega.

Öeldust järeldub, et paroolkaitsesüsteemis on kasulik kehtestada parooli teatav minimaalpikkus. Ärilistes turbetoodetes on see harilikult 6–8 märki või reguleeritav (installeerimisel etteantav). Parooli seadmisel keeldub süsteem lühemat parooli vastu võtmast.

### 8.2.4 Parooli struktuur

Belli uuring näitas, et 15% paroolidest kujutas endast parooliomaniku isikuandmeid (ees- või perekonnanimi, tänav, linn, autonumber, tööruumi number, telefoninumber, isikukood) või mõnd sõnastikus leiduvat sõna (tavalisel kujul või tagurpidi). Variantide arvu kahandas veelgi märgistiku piiratus (ainult numbrid, ainult väiketähed jne, vt statistika jaotises 8.2.3).

Otstarbekas on paroolidena kasutada ebaharilikke ja häädamatuid märgikombinatsioone, mis sisaldaksid läbisegi suur- ja väiketähti ning numbreid. Mnemoonilisuse säilitamiseks võib konstrueerida lauseid, mille sõnade algustähed moodustavad parooli. Sobiva struktuuriga paroolide loomiseks võib kasutada parooligeneraatoreid; neid on saadaval iseseisvate programmidena või turbepakettide osana.

### 8.2.5 Paroolkaitse ründed

Paroolkaitse on küll vanimaid ja levinumaid andmeturbevahendeid, enamasti rakendatakse seda aga läbimõtlematult, ebatõhusalt ja kergekäeliselt. Näiteks ilmneb ülalmainitud Belli uuringust, et 86% vaadeldud tegelikest paroolidest (valim sisaldas 3289 parooli) olid liiga kergesti ennustatavad või otsitavad. Suur osa ülejäänud 14 protsendist oli ründealdis muudel põhjustel. Analoogilisi andmeid võib leida muudest uuringutest ja kontrollimisaruannetest. Muuhulgas on sellise olukorra üks peapõhjusi asjaolu, et personal ei teadvusta võimalike rünnete meetodeid ja vahendeid.

Üldmeetodilt võib ründemoodused jagada kolme rühma:

1) ründaja genereeritud paroolide katsetamine,
2) õige parooli volitamatu hankimine,
3) paroolkaitsest möödahiilimine.

#### *8.2.5.1 Paroolide genereerimine volitamatuks pääsuks*

***Mõistatamine kogemuse põhjal***, arvestades seniseid levinud parooli valimise tavasid on küll algeline, kuid liigagi tihti tulemuslik meetod (vt ülaltoodud statistika).

***Kõigi võimalike paroolide hulga skaneerimine***. Lühikese parooli ja piiratud märgistiku korral võidakse seda proovida isegi klaviatuurilt sisestamisel. Tavalisem on aga genereerimine mälus, selleks on käibel rida tarkvaratooteid (nt *PASSTEST*).

***Paroolide valimine sõnastikust***. Ründaja võib kasutada mingit süsteemis leiduvat loomuliku keele sõnastikku (näiteks tekstiprotsessori oma) või varustada "muukraua" omaenda sõnastikuga (nt programm *BRUTE*.). Sõnastikrünne on efektiivne, kui vähemalt süsteemi mõne kasutaja parool on madala entroopiaga, näiteks pärisnimi vms. Sõnastikuga rünnatakse ka paroolifailis vm krüpteeritult säilitatavaid paroole: krüpteerimisalgoritmi teadev ründaja arvutab selle algoritmiga kõigi sõnastikust võetavate sõnade kujutised. Kui mõni neist langeb kokku paroolifailis oleva kujutisega, ongi vajalik parool leitud. Praktika on näidanud seesuguste rünnete edu näiteks Unixi tüüpi operatsioonisüsteemides.

***Pöördalgoritmi rakendades*** võib teada saada mõnede rakenduspakettide paroole. Näiteks paketi *WordPerfect 5.x* üheksamärgilise failiparooli leiab programm *WPCRACK* sel meetodil 0,3 sekundiga või kiiremini, sõltuvalt arvuti jõudlusest.

#### *8.2.5.2 Õige parooli volitamatu hankimine*

Arvesse tulevad peamiselt järgmised organisatsioonilisi turvaauke ärakasutavad võimalused.

- Varem volitatu võib kasutada oma parooli, kui seda ei ole volituste lõppemisel tühistatud. Selline võimalus tekib sageli töötaja üleviimisel teisele töökohale või isegi töösuhte lõpetamisel.
- Parooli võib teada saada volitatud kasutaja läheduses selle sisestust jälgides või parooli üleskirjutatud kujul leides.
- Piisava kaitseta süsteemis võib ründaja tungida otse paroolide tabelisse.
- Trooja hobuste ning analoogiliste utiliitidega (*THIEF*, *GETIT* jt.) saab ekraanile tuua võltslogimisakna, kasutades seda sisestatava parooli püüdeks. Kui pahaaimamatu kasutaja on sinna sisestanud oma parooli, ilmub veateade ja seejärel juba tõeline logimisaken. Veateade ja kordamisnõue ei tekita tavaliselt kahtlusi, sest parooli kiirel sisestamisel on eksimine üsna tõenäoline. Trooja hobuse võib kaitsmata arvutisse paigutada teine sama või lähedalasuva arvuti kasutaja, see võidakse saada koos mingi programmiga (eriti võrgust), selle võib sinna saata koht- või laivõrgu kaudu tegutsev ründaja.

Tundlikumate süsteemide puhul tuleb arvestada luuretehniliselt professionaalsemate rünnetega:

- võrgus saab paroole püüda sideliinilt võrguanalüsaatori abil;
- parooli sisestust saab pealt kuulata akustiliselt, salvestatud klahvihelide analüüs võimaldab klahvivajutusi tuvastada;
- klaviatuuri ja arvuti vahelist suhtlust saab jälgida ja salvestada selle protsessiga kaasneva elektromagnetilise kiirguse kaudu.

#### *8.2.5.3 Paroolkaitsest möödahiilimine*

Selle tee avavad peamiselt organisatsioonilised turvaaugud.

- Ründaja ei vajagi parooli, kui seaduslik kasutaja on lahkunud arvuti juurest ja jätnud end välja logimata.
- Tihti jäetakse süsteemide installeerimisel muutmata ja parooliga kaitsmata teatavad üldised kasutajanimed, õigemini rollinimed, nt *GUEST* ja isegi väga suurte volitustega varustatud *SUPERVISOR*.
- Piisava kaitseta süsteemis võimaldavad otse ressursside poole pöörduda mitmed tavalised utiliidid või operatsioonisüsteemi funktsioonid. Näiteks saab mitmetes *PC*-de andmebaasisüsteemides lugeda faile *DOS*-i vahenditega, pöördumata üldse paroolkaitset sisaldava andmebaasiohjuri poole. Kohtvõrkudes suunatakse sellised ründed serverile, kasutades vastavaid utiliite või ründeprogramme (*Novell NetWare*'i puhul näiteks *TEMPSUP.NLM*, *NETCRACK*, *NetUtils 3* jt).

### 8.2.6 Paroolkaitse tugevdamine

#### *8.2.6.1 Parooli sisestuse ja säilituse turve*

Paroolkaitse mehhanism peab rahuldama järgmisi parooli sisestust puudutavaid turvanõudeid.

- Sisestamisel ei tohi parool olla ekraanil nähtav.
- Parooli sisestuskatsete arv peab olema tõkestatud (tavaliselt lubatakse kuni kolm katset); tõkkeni jõudmist peab süsteem signaliseerima või vähemalt registreerima. Lisaks sellele võivad mehhanismid sisaldada ka logimiskestuse tõkkeid.
- Kui süsteemis on kasutusel turvapäevik, peab see registreerima kõik sisselogimised, soovitatavalt aga ka kõik ebaõnnestunud logimiskatsed.

Kõrgendatud turvanõuetega süsteemides võidakse rakendada ka liiga kaua tegevuseta olnud kasutaja automaatset väljalogimist.

Modemi kaudu sooritatava kaugtöö korral tuleb rakendada tagasihelistusega (*dial-back*) modemeid, mis kasutajanime ja parooli vastuvõtmise järel võtavad ühenduse neile andmetele vastava saatekohaga.

Turvalisust tõstab paroolide salvestamine süsteemis mitte nende loomulikul kujul, vaid krüptograafiliselt muundatult. Levinuim on paroolide räsikuju (*hashed form*), eriti nn ***soolatud paroolid*** (*salted passwords*). Paroolifaili salvestatakse iga parooli (kasutaja, rühma, rolli) kohta kaks väärtust:

1) juhuslik arv s (nn sool) ja
2) $y = f(s,P)$,

kus *f* on ühesuunaline funktsioon ja P on parool.

Funktsiooni *f* pöördfunktsiooni leida on praktiliselt võimatu, seega ei saa kujutise järgi parooli leida lihtsalt tagasiteisenduse teel.

Sõnastikrünnet ei tee soolatud paroolid küll võimatuks, kuid muudavad ründesõnastiku vajaliku mahu niimitu korda suuremaks, kui palju on olemas erinevaid s väärtusi, st esialgne maht tuleb korrutada

suuruse s entroopiaga enne selle juhuarvu genereerimist. Kuna s on juhuslikult genereeritud ja ründaja ei tea seetõttu enne sõnastiku koostamist, millise s väärtusega on oletatavat parooli mingi süsteemi korral vaja kasutada, tuleb tal enne ründe sooritamist arvestada kõigi võimalustega ja vajalik ründesõnastik muutub seega pikemaks.

Paroolifaili kaitseks rakendatakse mitmes operatsioonisüsteemis tavalise failipöörduse blokeerimist, võimaldades juurdepääsu ainult privileegoperatsioonide ja eriutiliitidega.

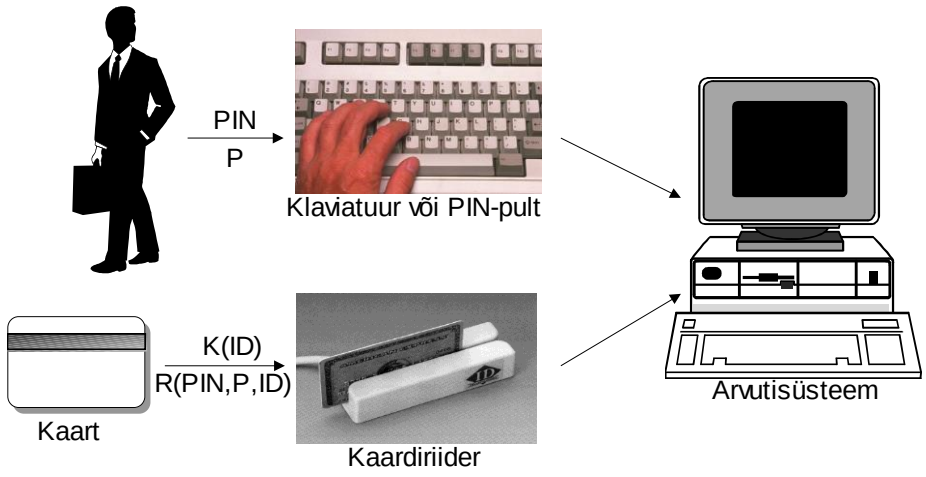
Trooja hobuste vastu tuleb rakendada eeskätt organisatsioonilisi meetmeid, muuhulgas logifailide regulaarset kontrolli ja logiandmete võrdlemist teadaoleva süsteemikasutusega.

#### *8.2.6.2 Paroolkaitse tugevdamine muu volitustõendiga*

Kõrgete turvanõuete korral võib paroolkaitset kombineerida muude volitustõenditega (vt 8.3, 8.4). Kombineeritud kaitset illustreerib alljärgnev näide.

Joonis 16 on tüüpiline paroolil ja kaardil põhinev pääsu reguleerimise süsteem. Kasutajal on kaart, millele on salvestatud järgmised andmed:

1) omaniku identifikaator *ID* (nt nimi), krüpteerituna salajase võtmega, mis on olemas kõigil volitatud agent-allsüsteemidel, nii et nad saavad *ID* dešifreerida;
2) omaniku isikliku identifitseerimisnumbri *PIN*, parooli *P* ja identifikaatori kombinatsioon räsikujul *R(PIN, P, ID)*; räsimine sooritatakse ühesuunalise funktsiooniga, nii et *PIN* ega *P* leidmine *R(PIN, P, ID)* järgi pole võimalik;
3) omaniku krüptovõtmed krüpteeritult.




**Joonis 16. Kasutaja identifitseerimine kaardi ja parooli põhjal**

Identifitseerimise protseduur on järgmine:

1. Kasutaja pistab kaardi riiderisse ning sisestab oma *PIN*-koodi ja parooli. Kahe pääsuparameetri kasutamine lisab paindlikkust; näiteks võib *PIN* olla rangelt isiklik ja salastatud, parool võib aga kõigil volitatud kasutajail olla ühine.
2. Arvutisüsteemi agent-alamsüsteem loeb kaardilt omaniku *ID* ja dešifreerib selle.
3. Agent arvutab kolme tema käsutuses oleva sisendparameetri järgi *R(PIN,P,ID)* väärtuse ning võrdleb tulemust kaardile salvestatud *R(PIN,P,ID)* väärtusega.
4. Funktsiooni *R()* väärtuse kahe eksemplari ühtumisel otsustab agent, et kasutaja on kaardi omanik, seega volitatud kasutama süsteemi.

5. Agent loeb ja dešifreerib kasutaja võtmed ning kasutab neid kaugtöös kasutaja eest.

Selle protseduuri rakendamisel ei sisalda agent mingeid volitatud kasutajate ega paroolide loendeid. Ta kontrollib lihtsalt, kas kaardi kasutaja on kaardi omanik. Ainus nõrk koht on selles, et agendi käsutuses on kasutaja *PIN* ja võtmed. Pääsu reguleeriv tarkvara peab need andmed pärast seansi lõppu agendi mälust kustutada, et neid ei saaks seal rünnata.

Samade komponentide baasil saab loomulikult konstrueerida ka teistsuguseid protseduure.

Paroolkaitset kombineeritakse ka biomeetriliste volitustõenditega (vt 8.4), näiteks dünaamilise stereotüübi analüüsiga: kasutatakse asjaolu, et klaviatuurilt sisestamise rütm on isikupärane (vrd radistide "käekiri" telegraafivõtmega töötamisel, on edukalt kasutatud isikutuvastuseks). Rahaautomaatides on hakatud rakendama silma vikerkesta analüüsil põhinevat isikutuvastust.

### 8.2.7 Paroolide haldus

Tõhusalt korraldatud paroolkaitse puhul vastutab paroolide korrakohase genereerimise, kehtivustähtaegade, paroolide jaotamise jms eest selleks volitatud töötaja, kohtvõrkudes tavaliselt võrguülem.

#### *8.2.7.1 Paroolide jaotamine*

Tavaliselt valivad üksikkasutajad endile paroolid ise. Paroolihaldur hoolitseb rühma- ja eriparoolide (sh. vaike-kasutajanimede) eest, jälgides, et süsteemi ei jääks paroolideta kasutajanimesid. Kasutajailt tuleb nõuda turvalise pikkuse ja struktuuriga (vt 8.2.3, 8.2.4) paroolide valimist ja paroolide turvalist kasutamist; selleks on otstarbekas jagada kasutajaile koos muude turvajuhistega alljärgnev meelespea.

**Parool kaitseb, kui**

- kasutad peale tähtede A-Z ka väiketähti a-z ja muid märke (#,¤,%, …)
- ta on vähemalt 6 märki pikk
- väldid pärisnimesid ja üldkasutatavaid sõnu
- vahetad teda regulaarselt
- sa ei kirjuta teda üles
- sa ei ütle oma parooli kellelegi teisele

Paljudel juhtudel nõutakse kasutajate paroolide salastamist ka paroolihalduri eest. Selleks on kasutusel muuhulgas järgmised meetodid:

1. Reas pankades genereeritakse paroolid vastava programmi abil ning prinditakse nad suletud ümbrikutele ilma värvilindita maatriks- vm löökprinteri abil; ümbrikus on kopeerleht ja paber. Ümbrikud adresseeritakse automaatselt ning jagatakse kasutajaile. Seansiparoolide rakendamisel prinditakse korraga terve rida paroole. Turvalisuse tõstmiseks jaotab nt *Society for Worldwide Interbank Financial Telecommunications* (*SWIFT*) oma võrguparoolid kaheks osaks ning saadab kumbagi poole loetelu kasutajale eraldi.

2. Kasutaja valib parooli ise, kuid see salvestatakse süsteemis mitte otseselt, vaid kombinatsioonis kasutaja kontonumbriga, kusjuures kombinatsiooni moodustamiseks kasutab süsteem ühesuunalisi räsifunktsioone, mis välistavad parooli leidmise sellest kombinatsioonist (vt 8.2.6.1). Kui kasutaja unustab parooli, peab ta pöörduma paroolihalduri poole ning aktiveerima ja ette andma uue parooli.

Paroole tuleb vahetada regulaarselt, parooli pikkusest ja konkreetsetest tingimustest sõltuva perioodilisusega (vt ka 8.2.3), kuid mitte harvemini kui kaks korda aastas. Ennetähtaegselt tuleb paroole vahetada alati, kui on tekkinud kahtlus parooli teatavaks saamise kohta. Parooli kasutava töötaja vallandamisel tuleb tema parool tühistada vahetult enne ta informeerimist vallandamisotsusest.

Teatavate ressursside ja kasutajate (nt külaliskasutajate) jaoks on otstarbekas rakendada tähtajalisi paroole, kusjuures tähtaeg on defineeritud nii kuu- või nädalapäevade kui ka kellaaegade järgi.

#### *8.2.7.2 Paroolide säilitamine ja kasutamine*

Väga tundlike failide, sh paroolide etaloneksemplare sisaldava faili kaitseks on kasulik rakendada kaksikparooli, st juurdepääs neile failidele on võimalik ainult kahe selleks volitatud isiku koostöös.

Paroolide etaloneksemplare tuleb säilitada krüpteeritult peitfailis, millele on defineeritud asjakohased kasutamisõigused. Võrgus võib paroole edastada ainult krüpteeritult.

Paroole ei ole soovitatav säilitada kirjalikul kujul. Kui seda siiski peetakse vajalikuks, tuleb iga kasutaja parool paigutada suletud ümbrikusse, millele on kirjutatud kaitstava süsteemi identifikaatorid ja mille sulgemisjoone kasutaja pitseerib või märgistab oma allkirjaga. Ümbrikut säilitatakse vastavalt salastatud dokumentide hoide eeskirjadele.

Tuleb hoolitseda selle eest, et kasutajad end välja logimata ei lahkuks arvutite juurest. Mitmed turvauuringud näitavad, et vägagi sageli jäetakse arvuti lõunatunniks ja muude ajutiste eemalolekute ajaks kaitseta. Otstarbekas on varustada pimendi (*screen saver*) parooliga.

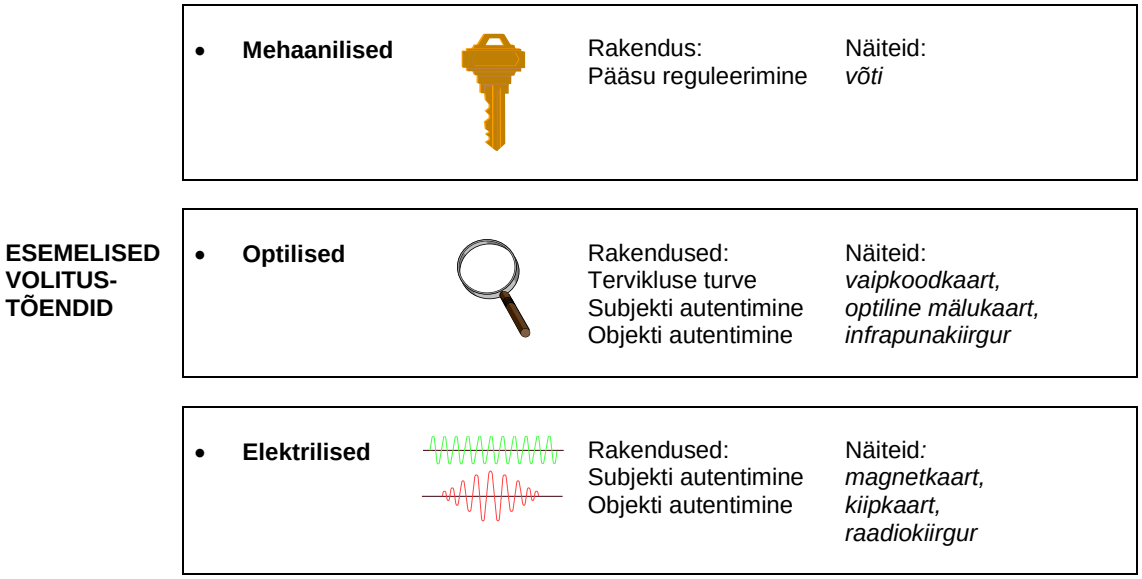
Süsteemi turvalisuse eest vastutajad peavad hoolitsema selle eest, et kasutajate käsutuses ei oleks ohtlikke üldotstarbelisi utiliite ega spetsiaalseid ründeprogramme. Võlts-logimisprogrammide avastamiseks peavad kasutajad regulaarselt kontrollima, kas kataloogide algusesse pole tekkinud logimisprogrammi nime (*Login* või vastav).

## 8.3 Esemelised volitustõendid

### 8.3.1 Põhiliigid

Esemeliste volitustõendite eelis on kasutamise hõlpsus ja kiirus. Mnemoonikaprobleemide puudumise tõttu on võimalik kasutada tunduvalt pikemaid pääsukoode, st suuremat kombinatsioonide hulka. Põhipuudused on

- kaotamise või varguse oht; vastumeetmena võib kasutada ühendamist paroolkaitse ja krüptotehnikaga;
- (levinumate) suhteliselt kerge kopeeritavus; vastumeetmed on volituseseme keerukam tehnoloogia, kombineerimine paroolkaitse ja krüptotehnikaga;
- kõrgem maksumus, eriaparatuuri vajadus.




**Joonis 17. Esemeliste volitustõendite põhiliigid**

Mittemehaaniliste esemeliste tõendite põhiline teostusviis on mitmesugused taskuformaadis (vt Tabel 7) plastikkaardid. Niisuguste kaartide ja nendega seotud aparatuuri standardimisega tegeleb ISO/IEC alamkomitee JTC1/SC17. Optilised ja elektrilised tõendid võivad aga olla kujundatud ka rinnamärgina, käekellana, taskus kantava karbikesena, mehaanilise võtme ühe osana jne.

**Tabel 7. Identifitseerimiskaartide nimimõõtmed (ISO/IEC 7810:1995)**

| Kaardi tüüp | Laius, mm | Kõrgus, mm | Paksus, mm | Nurga ümardusraadius, mm |
|---|---|---|---|---|
| ID-1 | 85,60 | 53,98 | 0,76 | 3,18 |
| ID-2 | 105,00 | 74,00 | 0,76 | 3,18 |
| ID-3 | 125,00 | 88,00 | 0,76 | 3,18 |

Kaardi materjal peab ISO/IEC järgi olema polüvinüülkloriid (PVC) ja/või PVC-atsetaat või neile vastavate või paremate omadustega materjal, näiteks polüester, polüetüleen, polükarbonaat.

### 8.3.2 Mehaanilised volitustõendid

Levinuim liik on lukkude ja lukklülitite võtmed. Odavaimad ja lihtsaimad vahendid, nende koodikombinatsioonide arv on suhteliselt väike. Kopeerimine on ülilihtne, koodi saab asjatundja lugeda juba kiire välise vaatlusega ning sobiva mnemoonilise teisenduse abil meelde jätta. Täiuslikumad süsteemid on kallid, kuid ei suuda infoturbes võistelda elektrooniliste vahenditega. Hinnaliste inforessursside kaitseks mehaanilised võtmed ei sobi, nad kuuluvad kaitsevahendite elementaarkihti.

Lisaks traditsioonilistele rakendustele ruumide sissepääsude sulgemiseks, aparatuuriosade mehaaniliseks lukustamiseks ning seadmete sisse- või ümberlülituseks (sh arvutite süsteemiplokkide ja klaviatuuride blokeerimiseks) kasutatakse lukumehhanisme muude kaitsevahenditega kombineeritult ka näiteks arvuti pordiga ühendatavates turbeplokkides.

### 8.3.3 Optilised turvaelemendid ja volitustõendid

#### *8.3.3.1 Optilised turbemeetodid*

Optikal põhinevad turvameetmed on mitmekesisuselt ja rakendusvõimalustelt suurim tõendesemete grupp. Kasutatavate meetodite arv ulatub kümnetesse ning neid realiseerivate vahendite teostusviise on sadu. Erinevalt mehaanilistest vahenditest, mis on juba algselt välja töötatud turvaotstarbeks, kasutatakse optilistes vahendites meetodeid, mis on algselt välja töötatud muuks otstarbeks (vt Joonis 18). Erandi moodustavad ehk dokumentide turbe meetodid, kuid needki on algselt olnud tihedalt põimunud muude rakendusaladega (teaduslik, tehniline ja meditsiiniline visualiseerimine, tarbegraafika, disain, meelelahutus).

| | | Algseid rakendusi | Vahendeid | Infoturberakendusi |
|---|---|---|---|---|
| **OPTILISED TURBE-MEETODID** | • **Andmete visuaal-kodeerimise meetodid** | *Toodete märgistus*<br>*Hindade kodeerimine* | Vöötkood<br>Vaipkood | Autentimiskaardi põhi- või abielement |
| | • **Dokumentide turbe meetodid** | *Väärtpaberid*<br>*Raha*<br>*Passid* | Vesimärk<br>Reljeef<br>Hologramm<br>Kinegramm | Autentimisteabe kodeerimine<br>Autentimiskaardi lisaturve<br>Dokumendivälja turve |
| | • **Optilise salvestuse meetodid** | *Laserketas*<br>*Laserplaat* | WORM-salvestus | Laserkaart<br>(optiline mälukaart) |
| | • **Optilise side meetodid** | *Telerite jms juhtimispuldid* | Infrapunaside | Infrapunamärk |

**Joonis 18. Optilised turbemeetodid**

Turbe tugevuselt on optiliste meetodite skaala väga lai. Mitmed meetodid on autentimisteabe ainsa kandjana rakendamiseks liiga nõrgad, kuid enamasti on optilisi meetodeid tehnoloogiliselt lihtne kombineerida ning sellega turbetugevust tõsta. Iseseisva autentimisvahendina tuleb arvesse peamiselt

laserkaart, teatud olukordades ka infrapunamärk. Muid meetodeid rakendatakse tavaliselt põhitõendi (nt magnet- või kiipkaardi) sekundaarse turvaelemendina.

### *8.3.3.2 Vööt- ja vaipkoodid*

**Vöötkood** (*bar code*, *1D bar code*) ja vaipkood (*2D bar code*) on kõige lihtsamad ja odavamad, kuid turbetugevuselt kõige nõrgemad optilised turvaelemendid. Neid rakendatakse laialdaselt seal, kus autentimisele kuuluvate subjektide arv on väga suur, kuid riskid on ohtude suhteliselt piiratud toime tõttu väikesed ning seetõttu peab turve olema odav. Tüüpiline on kasutamine juhilubadel, õpilaspiletitel, lugejakaartidel jms dokumentidel, aga ka lukukaartidel. Turbetehniline nõrkus on kopeerimise ja võltsimise hõlpsus.

Vöötkoodid võeti kasutusele juba 60il aastail, kuid nende lai levik sai alguse 1973. a, mil USA kaubahallides võeti kaupade märgistuseks kasutusele 12-kohaline numberkood UPC (Universal Product Code, "universaalne tootekood"), USA sõjajõudude ja autotööstuse standardiks aga sai Code 39 (numbrid, suurtähed, 6 erimärki). Need kaks koodi on oma rakendusulatuselt tänini kõige massilisemad. Vöötkoodisüsteemide arv ulatub praegu kümnetesse, paljud neist on välja töötatud mingeid spetsiifilisi rakendusi või eriomadusi silmas pidades.

Vöötkood (vt Joonis 19 ,a) kujutab endast vööti, mis koosneb teatava (ühe või mitme) paksusega püstkriipsudest ning sisemistest (kriipse eraldavatest) ja elementidevahelistest tühikutest. Kriipsu minimaalne jämedus on enamasti vähemalt mitu mikromeetrit. Kodeeritav märgistik sõltub konkreetsest koodisüsteemist ning ulatub kümnendnumbritest ASCII-märgistikuni. Kogu kood on enamasti püsiva pikkusega ning võib jaguneda kindla tähendusega väljadeks.

UPC rahvusvahelise analoogi EAN 13 (*European Article Number*, "Euroopa artiklinumber", joonis 27,a) struktuur on järgmine:

1) valmistajamaa kood (2 kümnendkohta),
2) valmistaja või tarnija kood (5 kümnendkohta),
3) valmistaja tootekood (5 kümnendkohta),
4) kontrollnumber (1 kümnendkoht).

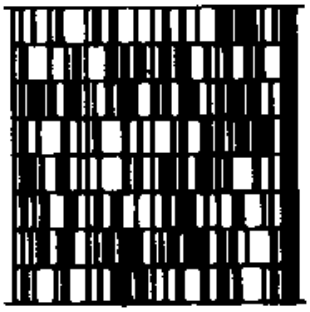
Näiteks tähistab 50 18374 49130 5 briti firma Tesco valmistatud dieetkuivikuid.

Levinumaid vöötkoode lisaks juba nimetatutele:

ISBN on raamatute märgistuseks kasutatav EAN 13 alamhulk prefiksiga 978;
ISSN on ISBN analoog ajakirjade jaoks (EAN 13 prefiksiga 977);
EAN 8 on EAN 13 alamhulk, 8-kohaline kood pisikaupadele, kuhu pikk kood ei mahu;
Code 128 on Code 39 analoog, millega saab esitada 128 ASCII-märki.

Kuna vöötkoode saab printida peaaegu igasuguse printeriga, koodiriiderid on ehituselt lihtsad ja odavad (alla 300 dollari) ning standardid on stabiliseerunud, on nad levinuim esemete automaatse identifitseerimise vahend. Paljudes rakendustes jääb aga vajaka nii koodi pikkusest kui ka märkidest. See sundis otsima võimalusi koodi edasiarenduseks.


**a** - vöötkood EAN 13

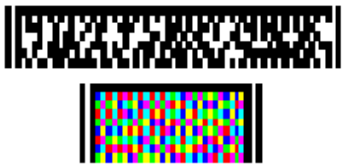


**b** - virnkood Code 49

**c** - virnkood PDF417

**d** - maatrikskood QR

**e** - maatrikskood Aztec



**f** - maatrikskood UltraCode

**Joonis 19. Vööt- ja vaipkoodide näiteid**

**Vaipkood** on vöötkoodi kahemõõtmeline vaste, mis võeti esmakordselt kasutusele 1984. a.

Esimesed vaipkoodid olid **virnkoodid** (*stacked code*, *multi-row code*), st lihtsalt mitmerealised vöötkoodid (vt Joonis 19, b). Code 49 meenutab vöötkoodi Code 39, on 2- kuni 8-realine (igas reas 18 kriipsu) ja võimaldab kodeerida 7-bitist ASCII-märgistikku. Code 16K on 2- kuni 16-realine, kompaktsemate teisendustabelitega ning mahutab kuni 8025 kirjamärki või kuni 16050 kümnendnumbrit. Mõlemat koodi saab lugeda laser- või CCD-riideriga.

Levinumaid virnkoode on PDF 417 (Joonis 19, c), mida muuhulgas kasutatakse USA sõjaväepiletitel. Üks koodisümbol mahutab 1850 ASCII-märki või 2729 kümnendnumbrit. Äärmiselt kõrge on koodi veaparandusvõime: kood taastatakse täielikult ka siis, kui sümbolist hävib 50%. Miniatuurse variandi Micro PDF 417 sümbolisse mahub 250 kirjamärki või 366 numbrit.

Tõeliselt kahemõõtmelised on **maatrikskoodid** (*matrix code*), mille elemendiks on kriipsu asemel ruudu- või muukujuline piksel (vt Joonis 19, d-f).

Ruudukujulise sümboliga tihe kood QR (Joonis 19,d) on määratud jaapani hieroglüüfistike ja ladina tähestiku kodeerimiseks. Sümboli maht on 4464 kirjamärki või 7366 numbrit. Koodisümbol orienteeritakse kolmes nurgas asuvate tugiruudukeste järgi.

1995. a patenditud ja ISO-s standardimisel oleva koodi Aztec (Joonis 19, e) moodulsümbol on keskme ümber kontsentriliselt laiendatav. Vähim sümbol (15×15) mahutab 14 kümnendnumbrit, suurim (151×151) mahutab 3067 kirjamärki. Märgistik on täielik ISO 8859-1 ladina tähestik nr.1.

UltraCode'i (Joonis 19, f) mõõduka mahuga sümbol laieneb vajaduse korral. Ta ladina, kreeka, vene, hiina ja jaapani tähestikkudega mitmeid loomulikke keeli. Neljavärviline variant (alumine koodiriba joonisel) kahekordistab kodeerimistihedust.

Vaipkoode on praegu kasutusel üle kahekümne. Lisaks oma põhirakendustele pakuvad nad kombineeritult muude optiliste vahenditega odavat ja vähemalt keskmise turbetugevusega autentimistõendit. Kasvanud infomaht võimaldab kasutada ka krüptotehnilisi vahendeid. Üks selliste kombinatsioonide näiteid on briti firma Electronic Automation loodud masinloetav hologramm, mis sisaldab kolmemõõtmelist maatrikskoodi.

#### *8.3.3.3 Dokumentide optilised turvaelemendid*

Paber- või plastikdokumentide (paberraha, väärtpaberid, passid, viisad, isikutunnistused, juhiload, äridokumendid jne) kaitse moodustab omaette valdkonna, mida tavaliselt nimetatakse turvatrükinduseks (*security printing*). Ainuüksi turbemeetodite aluseks olevate füüsikaliste ja keemiliste nähtuste arv ulatub kümnetesse, meetodite teostusviiside arv aga sadadesse. Optiliste turvaelementide detailide mõõtmed ulatuvad visuaalsetest submikromeetrilisteni (vt Tabel 8). Areng üha väiksemate mõõtmete suunas peegeldab taotlust asendada kaitsva *kujutise* keerukust (giljošš jms) *struktuurse* keerukusega, mida on tunduvalt raskem võltsida; kujutis ja tema muutused turvarikke korral peavad olema nii visuaalselt kui ka automaatselt kergesti tuvastatavad.

**Tabel 8. Optiliste turbedetailide mõõtmeklassid**

| Klass | Mõõtmed, mm | Näiteid |
|---|---|---|
| Nähtav | > 0,1 | Erivärvuskujutis, luminestsentskujutis, vesimärk |
| Makroskoopiline | 0,1...0,01 | Giljošš, turvaniit, mikrotekst, latentkujutis |
| Mikroskoopiline | 0,01...0,001 | Rastri modulatsioon, metalne läige, pärlmutter, steganograafia |
| Submikromeetriline | < 0,001 | Iridestsentsed muutoptilised elemendid (hologramm, kinegramm jt) |

Optilised turvaelemendid jagunevad otstarbe järgi kolme kihti: abivahenditeta kontrollitavad, lihtsate abivahenditega (mikroskoop, UV-lamp vms) kontrollitavad ning eriseadmete ja -teadmistega kontrollitavad, sageli salastatud elemendid.

Algselt töötati enamik meetodeid välja visuaalseks verifitseerimiseks, valdav enamik neist on aga rakendatav ka masinloetavana. Paberdokumentide turbevahendite algne ja peamine otstarve oli võltsimisrünnete tõrje (dokumendi autentsuse ja tervikluse tagamine), seetõttu jagunevad meetmed kolme rühma: (1) dokumendimaterjalile suunatud meetmed, (2) trükitehnilised meetmed, (3) optilised turvaelemendid (vt ka Tabel 9).

**Tabel 9. Võltsingutõrjemeetmed**

| Turvameetmed | Võltsimismeetod | | | |
|---|---|---|---|---|
| | Volitamatu originaal | Koopia | Imitatsioon | Muutmine |
| Organisatsioonilised | x | | | |
| Paberi tehnoloogia | x | x | x | |
| Trüki tehnoloogia | x | x | x | x |
| Muutoptilised elemendid | x | x | x | x |
| Lamineerimine | | | | x |

***Paberi tehnoloogia***

võimaldab rakendada mitmesuguseid võtteid dokumendi tervikluse kaitseks:

- mehaanilise kustutuse väldib trükivärvi või tinti imav või sellega reageeriv paber, mis "imeb" kirja kogu paberi paksusesse;
- mehaanilised kustutused toob ilmsiks kromogeenikiht, mis värvub kraapimisel;
- keemilist kustutust inditseerib lahustitega reageeriv lisand;
- kustutust väldib lamineerimine hapra, kergesti rebeneva kattekilega.

Autentimismeetmete valik on veelgi ulatuslikum:

- vesimärk;
- metalliseeritud, värvilised või fotokromaatilised turvakiud või -niidid (turvaniit võidakse ka katta mikrotrükiga), sh punktiirniit (*window thread*), mis kulgeb vaheldumisi paberi pinnal ja sisemuses;
- planšetid (1–2 mm läbimõõduga litrid), mis võivad kanda luminofoori või muutoptilisi elemente;
- foto-, termo-, bio-, elektro- või kemoluminofoor kiudude, terade või turvaniitidega sisseviidult (luminofoore võib sisaldada ka trükivärv, tint või templivärv);
- iridestsentsed mitmekihilised või mikroosakestest koosnevad struktuurid;
- keemilised latentkujutised ("salatint");
- tagasipeegeldusega (retrofleksiivsed) pinnad: läige valguse suunas varjab kujutise.

***Trükitehnoloogia***

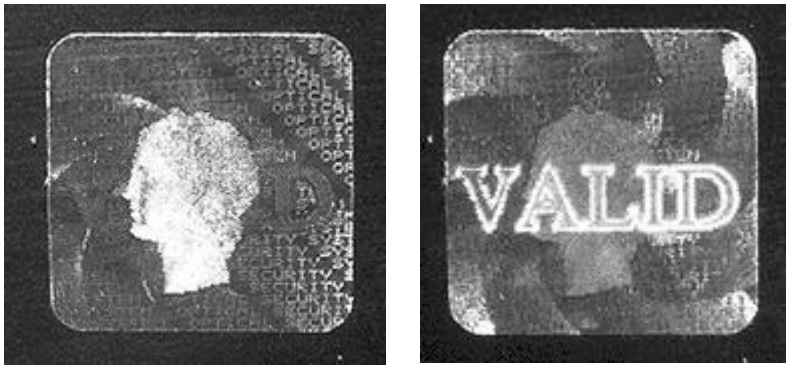
turbemeetoditest on suur osa alati olnud rajatud turvatrüki tehnilisele üleolekule imiteerija või kopeerija käsutuses olevast. Edasine areng tugineb sellele üha rohkem.

Traditsioonilinegi trükitehnoloogia pakub kümneid vahendeid imiteerimise, kopeerimise ja muutmise tõrjeks, näiteks:

- peened joonestikud, giljošš, mikrokujutised;
- kujutise jaotamine paberi kahele küljele, täpse ühitamisega, mida näitab läbivalgustus;
- reljeefkiri, reljeefpitser;
- perforatsioon, sh erikujuliste avadega ja laserperforatsioon;
- värvuste valimine väljastpoolt paljundivärvuste piirkonda;
- värvipaarid, mille värvuskontrast sõltub valguse tüübist;
- rasterlõks – korrapärane kujund, mille interfereeriv ruumisagedus tekitab kopeerimisel muareemustri;
- rastrinurga modulatsioon (SAM) tekitab kopeerimisel steganograafilise sõrmejälje (nt tekstina "KOOPIA", vt Joonis 20);
- steganograafiline kodeerimine paljundi või skanneri ruumsageduse diskreetimisribas (SABIC).


**Joonis 20. Kopeerimisel või skaneerimisel tekkiv sõrmejälgtekst**

Autentimisotstarbelistest vahenditest vastavad kontrollimise hõlpsuse ja võltsimise keerukuse nõudele eriti nn ***muutoptilised elemendid*** (OVD, *opically variable devices*). Need on elemendid, mille kuju, värvus, asukoht vm tunnus muutub vaatlustingimuste (valguse tüüp, vaatlusnurk, temperatuur jne) muutumisel. Eriti hõlpsalt jälgitavad on optokineetilised (liikuvad või kippavad) elemendid, mille abil saab luua isegi animatsioone (vt Joonis 21) .




**Joonis 21. Muutoptiline element: kallutamisel kujutis muutub**

Turbe seisukohalt on oluline eristada mitteiridestsentseid ja iridestsentseid muutoptilisi elemente.

***Iridestsents*** on värvitu pinna optiline värvumine või helkimine pinna korrapärases mikrostruktuuris tekkivate difraktsiooni- või interferentsinähtuste tõttu. Piltlikult öeldes: juhuslikkus tekitab hajusa peegelduse, korrastatus tekitab iridestsentsi. Looduses esineb ta näiteks mõnede liblikaliikide tiibade mikrolibledel. Trükitehnikas võimaldavad teda rakendada nüüdisaegsed tehnilised vahendid (lasersöövitus, vaakuumsadestus, elektronkiirlitograafia).

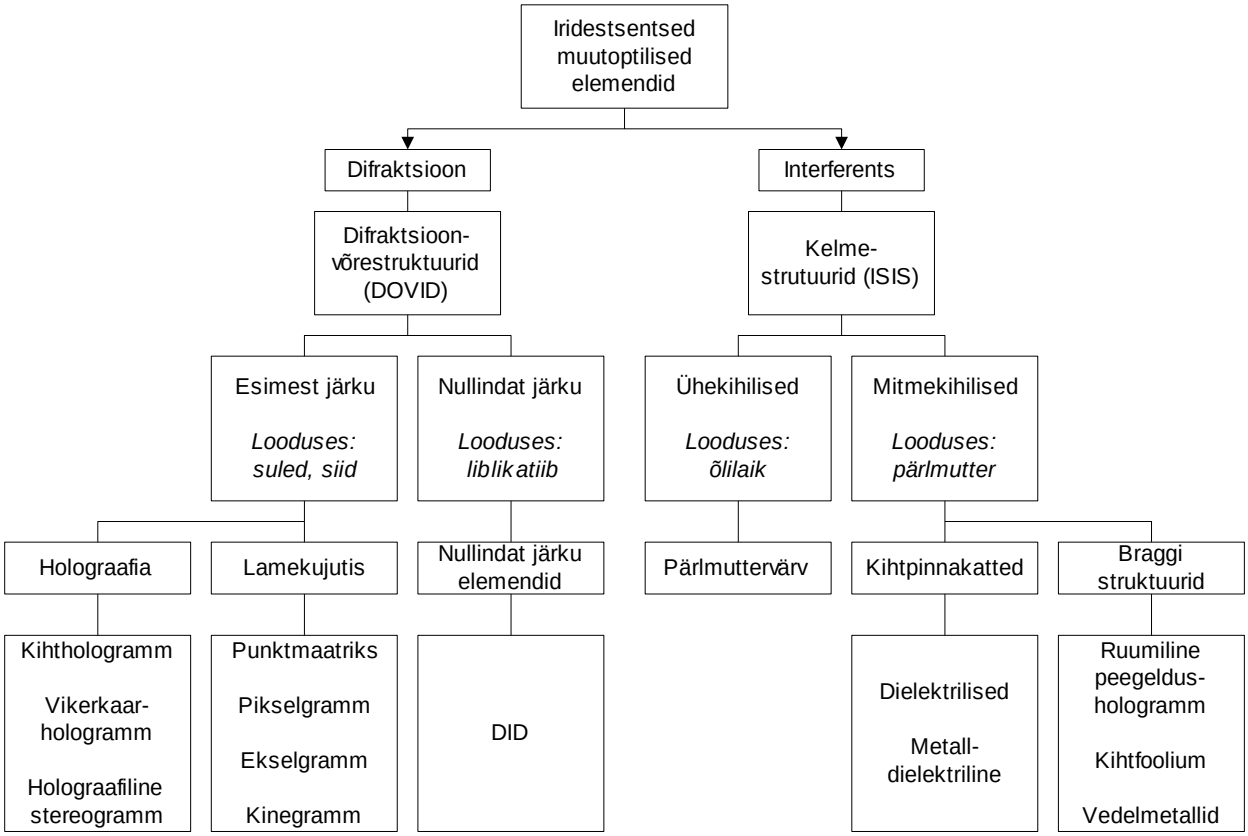
***Mitteiridestsentsed muutoptilised elemendid***

põhinevad peamiselt järgmistel optilistel nähtustel:

- hajus peegeldus (nt metameervärvid: värvus sõltub vaatlusnurgast);
- suundpeegeldus (nt metallfooliumid);
- hajus läbistus (vesimärk);
- läbistus (läbipaistev aken, nt Austraalia plastikrahas);
- parallaks (kallutamisel kaduvad või tekkivad ajutised kujutised, läätsmaatrikselemendid);
- fotoluminestsents (valgusest sõltuv helendus);
- pööratav fotokromatism (värvusemuutus teatava, nt aktiinse valguse toimel);
- pööratav termokromatism (värvusemuutus temperatuuri toimel);
- elektrokromatism, nt paberis, kaitsekiles või trükivärvis sisalduvate vedelkristallide baasil.

***Iridestsentsed muutoptilised elemendid***

on tunduvalt suurema turbetugevusega kõrgtehnoloogilised vahendid. Kuna käesoleva raamatu trükitehnoloogia ei võimalda neid tehniliselt ja esteetiliselt huvitavaid elemente illustreerida, tasub asjahuvilistel tutvuda tekstis viidatud rakenduskohtadega ning väga efektsete näidistega R.L.v.Renesse raamatus ja Aestroni turvatrükinduse entsüklopeedias. Iridestsentsed elemendid jagunevad difraktsioonil ja interferentsil põhinevateks (vt Joonis 22). Difraktsioonelementide üldnimetus on DOVID (*diffractive optically variable image device*, "difraktiivne muutoptilise kujutisega element"), interferentselemente tähistab ISIS (*interference security image structure*, "interferentne turbekujutise struktuur").




**Joonis 22. Iridestsentsed muutoptilised elemendid**

Jämedamatel, esimest järku difraktsioonstruktuuridel on iridestsentsed ja kinemaatilised ilmingud suhteliselt nõrgad ning valgustamisel tugevalt hajusa valgusega nõrgenevad veelgi. Esimest järku struktuuride tuntud näide on hologramm.

***Hologramm*** on mikroskoopiline (1000–2000 joont millimeetril) difraktsioonstrukruur; ta jäädvustab interferentsimustri, mille tekitavad objektilt peegelduva laservalguse kimp ja samalt laserilt saadav tugikimp. Hologramm "dekodeeritakse" difraktsiooni kaudu. Nn vikerkaarhologrammi (S.Benton, 1969) saab vaadelda tavalises valguses. Et põhimikule paigutatud hologrammi saaks vaadelda läbivas valguses, metalliseeritakse ta tagakülg. Näivalt ruumiline kihthologramm sisaldab kaht või mitut kujutisetasandit; tagumisel tasandil on tavaliselt mingi korduv kujund, näiteks logo. Hologramme rakendatakse laialdaselt juhilubadel, kaubapakenditel (sh näiteks viskisiltidel). Tavaline, lisameetmeteta hologramm on tänapäeval suhteliselt kergesti võltsitav. Ida-Aasia maades tegutseb mitu suurt võltsimiskeskust, mille toodangut kasutatakse mitmesuguse piraatkauba, sh tarkvarapakettide märgistamiseks.

***Punktimaatriks*** on üks lihtsamaid ja levinumaid difraktsioonelemente. Tasapinnalise kujutise moodustab korrapärane punktimaatriks, tavaliselt tihedusega 400 joont tollil. Iga punkt koosneb püsiva suuna ja sammuga difraktsioonjoontest. Meetodit kasutab terve rida firmasid, sh briti Applied Holographics. Näiteid võib leida Microsofti pakenditelt.

***Pikselgramm*** (1988, CSIRO, Austraalia) on kujutis, mille luubiga nähtavad pikselid koosnevad muutuva suuna ja sammuga difraktsioonjoontest. Pikselgrammi tüüpiline optiline efekt on kujutise muutumine negatiivseks ta kallutamisel. Ei ole võltsitav tavalise laser-interferentstehnikaga.

***Ekselgramm*** on pikselgrammi edasiarendus struktuuri peenenduse suunas. Positiiv-negatiiv-kippamine on sihilikult maha surutud, üleminekud on sujuvamad. Rakendusnäiteid: Austraalia margil 1995, Vietnami pangatšekkidel, American Expressi toodetel.

***Kinegramm*** (Landis & Gyr, Šveits) loodi spetsiaalselt dokumentide kõrgturbeks. Ei jäädvusta loomulikke objekte, vaid genereeritakse arvutiga. Erinevalt nt hologrammist ei ole visuaalefektiks ruumilisus, vaid väga selge ja terava kujutise animatsioon. Meetodi aluseks on difraktsioonvagude ristlõike

ebasümmeetria. Võrreldes muude esimest järku elementidega peab optiline efekt veidi rohkem vastu hajusale valgusele. Rakendusnäiteid: Austria 5000-šillingine (millel Mozarti pea vaatab vaheldumisi paremale ja vasakule), Hollandi pangakaardid ja -tšekid, Šveitsi ID-kaart, 1998. a Saksa pangatähed. Kinegrammi on paigutatud isegi metallmündile (1998. a Hollandis valmistatud euro).

***Nullindat järku elemendid*** (Paul Scherrer Institut, Šveits). Aluseks on korrapärane suure murdumisnäitajaga (nt n=2) ruumiline submikromeeterstruktuur, mis on paigutatud väikese murdumisnäitajaga (nt n=1,5) maatriksisse. Turberakendust tähistatakse lühendiga DID (*diffractive identification device*, "difraktiivne identimisvahend"). Pööramisel muutub intensiivselt värv ja tekib tugevalt polariseeritud peegeldus. Ei ole tundlik valguse hajususele. Ärilised rakendused on alles väljatöötamisel.

***Kelmestruktuur*** koosneb ühest või mitmest pealistikku õhukesest (paksusega alla mikromeetri) kelmest, mis tekitavad interferentsiefekte. Kasutatakse ka trükivärvi (OVI, *optically variable ink*, "muutoptiline trükivärv") kujul. Ei ole tundlik valguse hajususele. Kujutis ei ole kopeeritav. Rakendusnäiteid: Kanada pangatähed, paljude pangatähtede trükivärv.

***Ruumiline peegeldushologramm*** (1962, J.Denisjuk, N.Liit). Vastav turvaelement luuakse kümnetest pisut varieeruva murdumisnäitajaga (nt 1,52 kuni 1,54) kelmetest koosneva kihiga. Hologrammi kallutamisel muutub värvus ruugest roheliseni või rohelisest siniseni, sõltuvalt kasutatud jäädvustuslaseri valguse värvusest.

Loetletud vahendid on määratud paber- ja plastikdokumentide turbeks. Mitmed neist rakendavad aga hoopis universaalsema toimega meetodeid, mida põhimõtteliselt võib kasutada igasuguse meediumi ja teabetüübi puhul. Eriti puudutab see steganograafilisi vahendeid. Steganograafiast annab lühiülevaate käesoleva raamatu 14. peatükk.

**Autentimisrakendused**. Paljusid vaadeldud optilistest meetmetest kasutatakse täiendava visuaalse või masinloetava kaitsevahendina nt magnetkaartide puhul; nad tagavad seal kaardi kuulumise teatavasse autentsesse sarja või dubleerivad autentimisteavet. Joonisel 31 on näide masinloetava hologrammi kasutamisest magnetkaardi lisaturbeks: hologramm sisaldab magnetribal olevate andmete krüptograafilist kontrollkoodi.

**Joonis 23. Hologrammiga magnetkaart. Masinloetav hologrammivööt on magnetriba all**

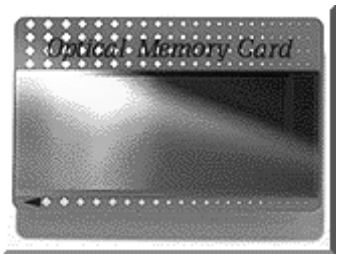
Iseseisva autentimisvahendina on seni leidnud laiemat kasutamist peamiselt hologramm.

**Hologrammkaardid** on magnetkaardiga võrreldes kopeerimiskindlamad, nende valmistuse ja lugemisaparatuuri hind on kõrgem. Standardid puuduvad. Praegu kasutatakse neid peamiselt maksuliste side- ja leviteenuste pääsusüsteemides (nt. British Telecomi telefonimaksesüsteemis Cardphone, maksuliste televisioonikanalite puhul jm.). Iga teenuseühiku tarbimisel kustutab riider vastava osa hologrammist. Kogu hologrammi ammendumisel tuleb osta uus kaart.

#### *8.3.3.4 Optiline mälukaart*

WORM-salvestusel põhinev optiline mälukaart (laserkaart) loodi 1980il aastail (Drexler Technology, USA). Ehituselt ja tööpõhimõttelt on tal sugulus laserplaadi (CD) ja laserkettaga (CD-ROM). Tüüpilist välisilmet (salvestisega külg) näitab Joonis 24. Nagu laserplaadilgi, jäetakse salvestist kandev peegelpind

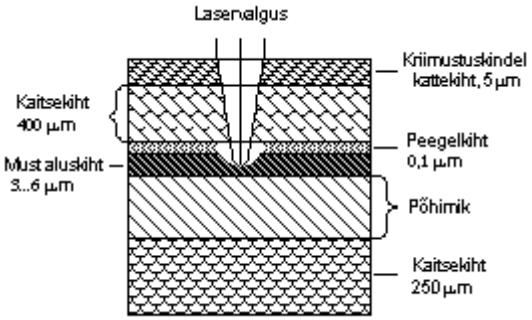
tavaliselt tühjaks, kaardi teisel küljel aga võib lisaks tekstile olla visuaalseid või masinloetavaid turvaelemente ning autentimisteavet nt vöötkoodina või magnetribal.




**Joonis 24. Optiline mälukaart**

Optilise mälukaardi ehitust näitab Joonis 25. Salvestuskandja koosneb peegelkihist mustal aluskihil. Mõlemad kihid valmistatakse hõbedast, kuid nende metallograafiline struktuur on erinev. Salvestamisel põletatakse laserikiirega peegelkihti bittide väärtusi esindavad avad läbimõõduga 2,25 μm (need on nähtamatud, sest silma eraldusvõime on 20 μm). Salvestuskandja koos põhimikuga suletakse läbipaistvate polükarbonaadist kaitsekihtide vahele; nende kihtide vastupidavuse iseloomustuseks: sama polükarbonaati kasutatakse näiteks hävitajate kabiinikupli materjalina.

Kaardi salvestusmaht ulatub 6,6 megabaidini. Tüüpiline on 4,1 MB, millest jääb rakenduse käsutusse 2,8 MB. Võrdluseks: kiipkaardi tüüpiline mälumaht on 16K, magnetkaart mahutab alla 1K.




**Joonis 25. Optilise mälukaardi ehitus**

Suure mälumahu tõttu kasutatakse laserkaarti peale tavaliste isikuautentimisrakenduste (juhiluba, pääsukaart, deebetkaart, ID-kaart) paljude selliste dokumentidena, kus lisaks autentimisteabele on hulgaliselt muid andmeid: haiguslugusid sisaldava patsiendikaardina, auto remondikaardina (auto "patsiendikaart"), Pentagoni merekonteinerite saatelehena jne. Kaardile on loodud hea liidestus PC-ga: süsteem näitab kaartsalvestit loogilise kettadraivina; seadmedraiverid on olemas DOS-i ja Windowsi keskkondadele.

Turbevahendina võimaldab kaart ulatuslikult rakendada krüptotehnikat. Laserkaarti on kombineeritud ka magnetribaga, kiibiga, hologrammiga jm vahenditega. ID-kaardina on ta kõrgete turbevõimaluste tõttu muutunud üha populaarsemaks. Kaardi levikut soodustab standardimine (üldnäitajad, radade paigutuse, kodeerimisviisi jms määravad ISO/IEC 11693:1995 ja 11694:1995).

Hea näite tugevate turvaomadustega autentimistõendist pakub USA Maine'i osariigi püsielaniku kaart. See on laserkaart, millel on muuhulgas järgmised turvaelemendid:

- peegelkihile on trükitud osariigi kaardi ja pitsati täpselt valitud fragmendid;
- salvestusala üla- ja alaserva piirab täpirida, ülarea täppides on kõigi 42 presidendi mikroportreed, alareas aga kõigi osariikide lippude värvilised mikrokujutised;
- kaardil on optiline vesimärk;

- kaardi pöördel on lasergraveeritud number, omaniku foto ja allkiri ning individuaalhologramm; kõik need on digitaalkujul ka kaardi mälus.

Analoogiliselt, kuigi mõnevõrra tagasihoidlikumalt on üles ehitatud ka muude nüüdisaegsete olulisemate isikut tõendavate kaartdokumentide turve.

### *8.3.3.5 Infrapunatõend*

Infrapunaside on lisaks tõsisematele tehnilistele rakendustele juba pikemat aega laialt kasutusel koduaparatuuri (telerid, videomagnetofonid, stereoseadmed, kardinate elektriajamid jms) distantsjuhtimise süsteemides. Turbes hakati teda kasutama üsna hiljuti, kuid ta osatähtsus kasvab kiiresti: mõne aasta eest valmistas vastavaid tooteid ainult üks firma, praegu on neid juba mitu (sh näiteks Olivetti ja Hewlett-Packard).

Nõudluse tekitasid kõigepealt suured haiglad, kus vajati distantstoimega pääsusüsteeme, raadiosagedusseadmeid (vt 8.3.4.4) aga ei saa neis kasutada meditsiiniaparatuuri tööd häiriva kiirguse tõttu (pealegi häirib aparatuur omakorda nende seadmete tööd). Asja uurinud Precision Tracking FM (Texas) lahendas ülesande edukalt: pääsutõendiks olevad infrapunakiirguriga varustatud rinnasildid võimaldavad ühtlasi automaatselt lokaliseerida töötajaid ja neid kiiresti kohale kutsuda (varem püüdsid neid funktsioone täita häiriv valjuhääldite süsteem või primitiivsed piiparid), samuti telefonikõnesid automaatselt ümber suunata.

Lokaliseerimiseks on kõigi ruumide lakke kinnitatud infrapunaside vastuvõtjad, mis on telefoniliinide või bifilaarjuhtme kaudu ühendatud arvutiga. Lokaliseerimissignaali väljastab rinnasilt iga 3–4 sekundi järel. Nupuke sildi tagaküljel võimaldab väljastada erisignaali, nt luku avamiseks. Infrapunasiltidega varustati ka patsiendid ja mobiilsed seadmed (vt Joonis 26, mis näitab ühe teise firma analoogilise süsteemi komponente), niisiis saab arvuti reaalajas jälgida kõige olulise paiknemist. Kuigi süsteem on suhteliselt kallis (iga silt umbes 100 dollarit ja iga ruumi seadmed umbes 60 dollarit), tasus ta end kiiresti; ühtlasi saavutati ajasääst umbes 10 minutit iga operatsiooni kohta.

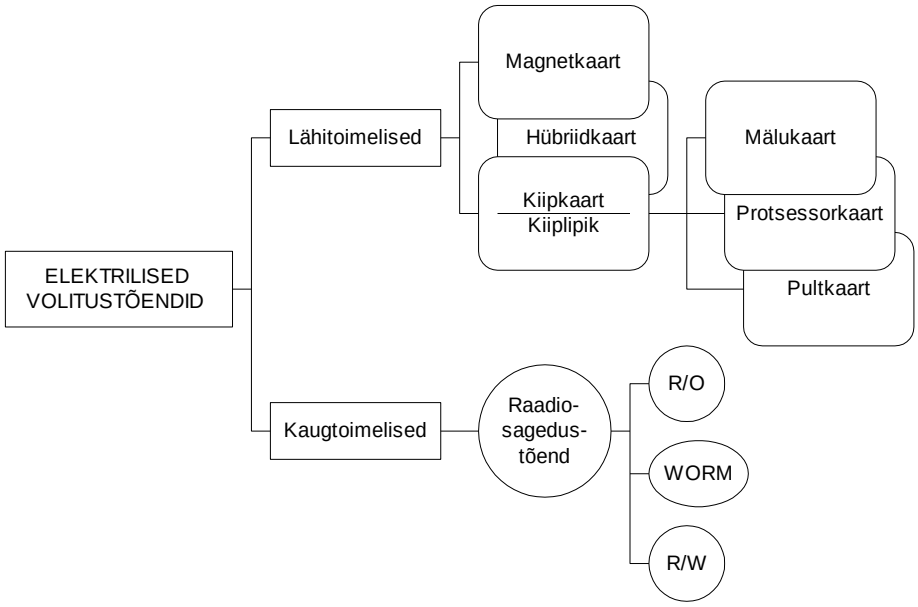

**Joonis 26. Infrapuna-autentimine. Vasakul vastuvõtja, paremal ID-kaardiga IP-rinnasilt, personali IP-rinnasilt ja patsiendi randmesilt**

Ülalkirjeldatule üsna sarnaseid süsteeme on võetud kasutusele farmaatsia- ja elektroonikatehastes, tarkvarafirmade klienditeeninduses, laomajanduses jm. Automaatseks lokaliseerimiseks on hakatud IP-siltidega märgistama üha väiksemaid objekte, näiteks kiipe. Infrapunatehnika on oma rakendustelt võrreldav raadiosagedussüsteemidega ning asendab neid kõikjal, kus valitsevad kitsendused raadiokiirgusele.

### 8.3.4 Elektrilised ja elektromagnetilised volitustõendid

#### *8.3.4.1 Elektriliste volitustõendite liigid*

Elektriliste volitustõendite jäme liigitus on Joonis 27. Kaks peamist käibelolevat vahendit on magnetkaart ja kiipkaart, viimastel aastatel on neile lisandunud raadiosageduslikud identsustõendid, mida seni kasutati laialdaselt passiivsete objektide tuvastuseks ja autentimiseks.

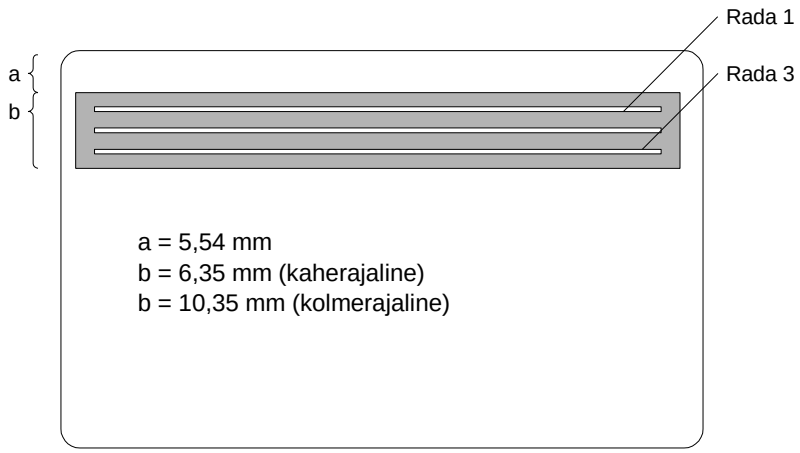



**Joonis 27. Elektrilised volitustõendid**

Alljärgnev lühiülevaade lähtub sellest liigitusest, puudutades ka mõningaid lisa- ja vahevariante, mida skeemil pole selguse säilitamiseks näidatud. Detailsemalt vaadeldakse kiipkaarti tema aktuaalsuse ja perspektiivsuse tõttu.

#### *8.3.4.2 Magnetkaart*

Suhtelise odavuse tõttu levinuim (aastas lastakse käibele üle kümne miljardi) autentimisvahend pääsu reguleerimiseks, rahalisteks operatsioonideks ja paljudeks muudeks rakendusteks. Levikut soodustavad stabiliseerunud standardid. Formaadi määrab ISO/IEC 7810 (vt Tabel 7 ülal), kaardile kantava reljeefkirja kuju ja mõõtmed ISO/IEC 7811-1 ja paigutuse 7811-3. Standardi 7811 osad 2, 4 ja 5 määravad magnetriba paigutuse ja parameetrid ning radade (kuni 3) paigutuse magnetribal (vt Joonis 28 ja Tabel 10). Levinumad on kaherajalise magnetribaga kaardid.




**Joonis 28. Magnetriba paigutus kaardil ID-1**

Rajad 1 ja 2 on määratud kasutamise käigus ainult lugemiseks, kolmas rada ka kirjutuseks. Magnetsalvestus on kahesageduslik. Veatõrjeks kasutatakse paarsusbitti koos liias-pikikontrolliga (LRC).

**Tabel 10. Magnetradade salvestustihedus ja salvestusmaht (ISO/IEC 7811)**

| | Andmete tüüp | Operatsioonid | Salvestustihedus, bit/mm | Salvestusmaht, märki * |
|---|---|---|---|---|
| Rada 1 | Tekstandmed | R/O | 8,27 | 79 |
| Rada 2 | Numberandmed | R/O | 2,95 | 40 |
| Rada 3 | Numberandmed | R/W | 8,27 | 107 |

* Koos juht- ja kontrollmärkidega.

Salvestis kaardi magnetribal on tundlik väliste magnetväljade, muuhulgas mitmesuguste sulgurites jm kasutatavate püsimagnetite väljade toime suhtes. Kaardi häirekindluse tõstmiseks valmistatakse kaarte ka tugevamalt magneeditud kõrgkoertsitiivse baariumferriidistmagnetribaga. ISO/IEC 7811-6 määratleb seda tüüpi magnetribade normparameetrid, nõudes koertsitiivsust 2500...4200 Oe (200...335 kA/m). Tavalist tüüpi raudoksiid-magnetriba koertsitiivsus on umbes 300 Oe, Euroopa standarditele vastava pangakaardi magnetriba koertsitiivsus aga 2750 Oe.

Tavalise magnetkaardi põhipuudus on liiga hõlbus kopeeritavus. Selle nõrkuse kõrvaldamiseks on välja töötatud keerukamaid mittestandardseid magnetriba struktuure.

**Vesimärgiga magnetribad.** Selle meetodi näidete hulka kuulub Emidata/Malco neljarajaline magnetriba, mille kihi magnetosakesed (tavaliselt on need nõeljad ja orienteeritud pikisuunas) orienteeritakse valmistusprotsessis pikisihist vaheldumisi kahele poole 45° kallutatud vöötidena, nii et magnetkihis tekib kaldtriibuline "vesimärkmuster". Radade 1–3 kirjutust ja lugemist niisugune kihi struktuur ei mõjuta. Rajale 0 kantakse valmistusprotsessis 50–100 bitti püsiinformatsiooni, mis on määratud osakeste püsiva vahelduvorientatsiooniga sellel rajal. Kaardi lugemiseks kasutatakse erikonstruktsiooniga riiderit; see rakendab rajale 0 kõigepealt püsimagnetvälja, mis hävitab võimaliku (tavalise salvestusmeetodiga tehtud) võltssalvestise ning alles siis loeb seda rada. Eritehnoloogia ja aparatuuri tõttu on meetod kallis.

**Kahekihiline magnetriba.** Alumise kihi koertsitiivsus on kõrge, pealmise kihi oma madal. Kirjutus sooritatakse tavalisest tugevama magnetväljaga, mis salvestab informatsiooni mõlemasse kihti. Riider sisaldab kustutuspea, mis enne lugemist hävitab salvestise pealmises kihis, nii et tavalisele ühekihilisele ribale tehtud koopia hävib. Protsess on mittestandardne ja kallim.

**Wiegandi kaart** loodi 70te lõpul turvarakendusteks; see on magnetkaardi erivorm, mille puhul magnetsalvestise lugemine põhineb Wiegandi efektil. Sellisel kaardil on magnetiliseks salvestuskandjaks mitte õhuke magnetkelme riba, vaid kaardi plastikusse pressitud magnetmaterjalist traadid. Wiegandi kaarti ei saa üle kirjutada ega kopeerida. Tavalise magnetkaardiga võrreldes on ta tunduvalt vähem tundlik väliste magnetväljade, temperatuuri ja muude välismõjude tõttu. Süsteem on mõnevõrra kallim

tavalisest ega ühildu sellega. Kuna lugemine on kontaktivaba, peavad aga nii kaart kui ka riider kauem vastu. Vaatamata suhteliselt piiratud rakendusvaldkonnale on Wiegandi kaarte kasutusel miljoneid, paigaldatud riiderite arv ulatub sadadesse tuhandetesse.

Ühelgi neist kallimatest erilahendustest ei ole suuri tulevikuperspektiive. Neid turbetugevuselt ületavate kiipkaartide ja biomeetriliste vahendite hind on kiirelt langemas ja peatselt võrdne magnetkaardisüsteemi omaga. Magnetkaart püsib just tänu oma standardsusele, seetõttu on otstarbekam säilitada standardsus, kuid lisada kaardile näiteks optilisi kaitseelemente.

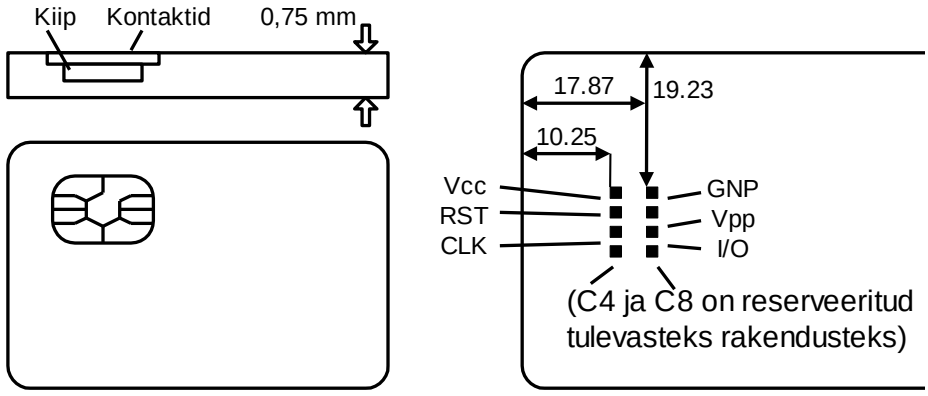
Odavuse tõttu on hakatud magnetkaarti kasutama üha väiksemate objektide kaitseks. Mitmed lauaarvutite turbetooted sisaldavad arvuti pordiga ühendatud kaardiriiderit, mis pakub tavalise paroolkaitsega võrreldes tunduvalt kõrgemat turvataset.

**Pääsudiskett** on magnetkaardi veelgi odavam aseaine. Ta ei vaja lugemiseks ega salvestuseks eriseadmeid, kuid võib täita samu funktsioone, eriti kombinatsioonis paroolkaitsega. On kasutusel mõnedes turustatavates turbepakettides.

Kuigi magnetkaarti on hakanud välja tõrjuma täiuslikumad autentimisvahendid (eeskätt kiipkaart ja laserkaart), jääb ta spetsialistide hinnangul oma hiigelleviku ja väljaarendatud infrastruktuuri tõttu käibele veel vähemalt 20-30 aastaks.

### *8.3.4.3 Kiipkaart*

Kiipkaart on üht või mitut integraallülitust sisaldav plastikkaart (vt Joonis 29). Ehkki magnetkaardiga võrreldes vähem levinud (1997. a lasti käibele 900 miljonit tk), on tal juba üsna pikk ajalugu (K.Arimura patent Jaapanis 1970, mitme rakenduse patendid Euroopas 1974), ees aga kiire arengu perspektiiv (prognoos aastaks 2003 on 6,3 miljardit tk). Arengut kiirendab praeguse rakendussfääri paiknemine kiirelt arenevates valdkondades (tervishoid, side, satelliit-TV, elektronarveldus).




**Joonis 29. Standardne kiipkaart (ISO 7816)**

Arengut pidurdab standardite rohkus ja kirevus. ISO 7816 (ei defineeri funktsioone) jms on mõnevõrra elust maha jäänud, nende nõuded tulevad arvesse ainult miinimumina, paljude valmistusfirmade sisestandardid on tunduvalt rangemad. Tunda annab ka konkurents tehniliselt tunduvalt nõrgema, kuid odavama ja arenenud infrastruktuurile toetuva magnetkaardiga, mille püsimajäämist ennustatakse veel 20–30 aastaks. Üleminekut magnetkaardilt kiipkaardile soodustab ***hübriidkaart*** – magnetribaga varustatud kiipkaart.

Magnetkaardist tehniliselt täiuslikuma ja tunduvalt turvalisema kiipkaardi peamine tehniline puudus praeguse komponenditaseme puhul autentimisprotsessi aeglus, eriti krüptoprotseduuride rakendamisel: 1024-bitise RSA-ga krüpteerimiseks (vt 9. ptk) kulub praegu kuni 500 ms. See võib osutuda takistuseks mitmete rakenduste puhul ning sundida eelistama biomeetrilisi meetodeid. Teisalt, komponentide tehnika kiire areng lisab iga aastaga ressursse (näiteks kiibi mälumaht kahekordistub Moore'i seaduse järgi 18 kuuga).

Kiipkaardi põhiliigid on mälukaart, protsessorkaart ja pultkaart.

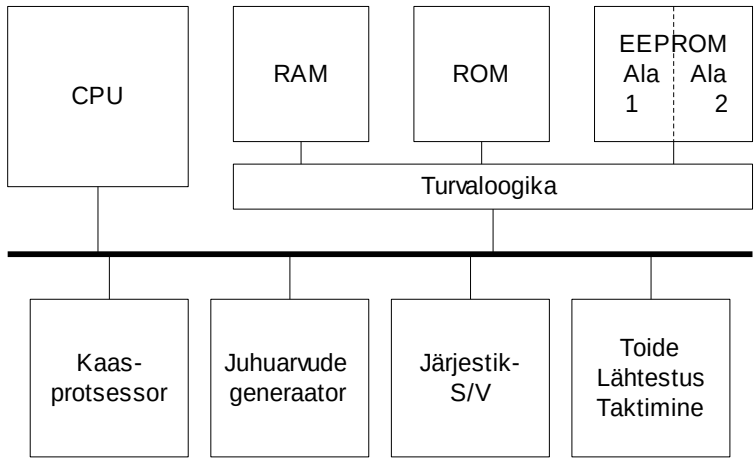
***Kaitseta mälukaart***
on kiipkaardi kõige algelisem vorm, mis hakkab käibelt kaduma. Kiibis asuv mälu on kontaktide kaudu otseselt kättesaadav, ta sisu saab lugeda sünkroon-andmevahetuse protokolliga. Turvaomadustelt nõrk.

***Kaitsega mälukaart***
on levinumaid praegu kasutusel olevaid vorme. Kaardi EEPROM-mälu jaguneb kaheks või enamaks eraldatud alaks, otsepöördus kontaktide kaudu on võimalik ainult ühte neist. Kui osi on mitu, võib üks neist olla kättesaadav ainult valmistajale, tema erivahendite abil. Mälumahud on tüüpiliselt 12 kuni 512 baiti, maksimaalne maht praegu 8K baiti. On kasutusel telefoni- ja automaadikaardina ning lihtsa pääsukaardina, mis sisaldab ühtainsat numbervõtit või PIN-koodi. On suhteliselt kergesti võltsitav.

***Turvaloogikaga mälukaart***
Mälu- ja protsessorkaardi vahevorm. Sisaldab autentimismehhanisme, elementaarset tehinguloogikat (maksesumma ei saa ületada jääki jne) ning andmetervikluse kaitset poolelijäänud tehingu puhuks (see võib juhtuda kaardi liiga kiirel eemaldamisel terminalist). Sellist tüüpi on enamik praegusi telefoni- ja transpordikaarte. Odav (1-5 dollarit). Suhteliselt võltsimiskindel.

***Protsessorkaart***

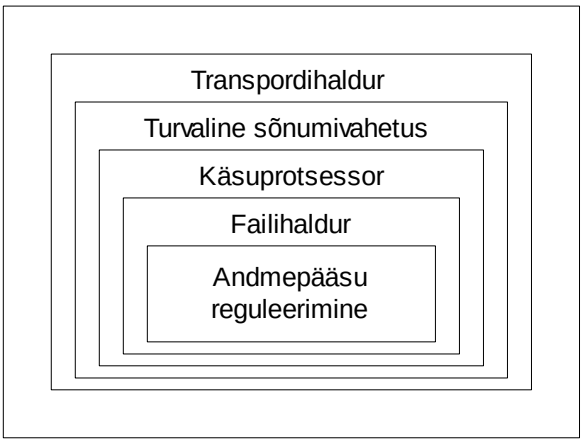
Nimetus on tinglik, ehkki peamine vorm sisaldab mikroprotsessoril ja von Neumanni arhitektuuril põhinevat mikroarvutit (vt Joonis 30). Ingliskeelne termin *smart card* seevastu hõlmab küll ka programmeeritavatel ventiilimaatriksitel (PGA, FPGA) põhinevat arhitektuuri, milles protsessorit asendavad olekumuutusseadised (*state change device*), kuid on laialt käibel ka kõigi kiipkaardiliikide üldnimena. Olekumuutusarhitektuur on kiirem, eriti krüptotehnilistes rakendustes, kuid kergemini analüüsitav ja emuleeritav. Traditsioonilise arhitektuuriga kaardi võimalusi piirab lisaks aeglusele ka mälumaht (tavaliselt mitte üle 16K baidi). Ressursside suurendamist piiravad transistoride soojaeraldused ning mehaanilise tugevuse nõuetele vastavad kiibi mõõtmed.




**Joonis 30. Protsessorkaardi mikroarvuti**

Protsessor on enamasti 8-bitine (nt Motorola 6805 või Intel 8051), kuni 5 MHz taktsagedusega; uuemates ka 16-bitine, sh RISC-protsessor. Krüptofunktsioone kiirendab aritmeetika-kaasprotsessor.

ROM-mälu sisaldab kaardi operatsioonisüsteemi, nn maski (*mask*); lisaks sellele võib seal paikneda rakendus. Opsüsteem on kihilise turbestruktuuriga (vt Joonis 31): transpordihaldur korraldab sidefunktsioone (ISO 7816-3), turvalise sõnumivahetuse kiht (ISO 7816-4) kaitseb autentsust ja konfidentsiaalsust, käsuprotsessor interpreteerib ja kontrollib käske.




**Joonis 31. Kaardi operatsioonisüsteemi turbestruktuur**

Kaardi numbrit jms sisaldava PROM-mälu maht ei ületa tavaliselt 32 baiti. Lisaprogrammide ja andmeplokkide mahutamiseks võidakse kasutada välkmälu (*flash memory*). EEPROM-mälu on kaardi jaoks kõvaketta vaste, ta võib hõlmata poole kogu kiibi mahust. Kaardi eluea määrab EEPROM-i lugemis-kirjutustsüklite maksimaalarv (tavaliselt 10000). Muutmälu (RAM) kasutatakse töömäluna, ta maht on 128-512 baiti. Kaardiarvuti töö iseärasuste tõttu on eeliseid säillugemisega muutmälul; selline struktuur on loodud ferroelektrilise muutmälu kujul (FERAM, FRAM). See lisakihiga pooljuhtmälu oleks sobiv kogu kaardi mälu teostuseks, osalt ROM- ja osalt RAM-kujul; praegu on ta aga firmaomanduslik ja kasutusel piiratult.

Sisend-väljundosa on kahesuunaline kahejuhtmeline järjestikliides. ISO 7816-3 annab mitu varianti, neist on levinuim asünkroonne märgipõhine protokoll. Andmevahetuskiiruse määravad kaarditerminalilt saadav taktsagedus ja kaardi kiibis asuvad sagedusjagurid.

Kaardiriideri või -terminali (vt joonised Joonis 32 ja Joonis 33) ehitus sõltub konkreetsest rakendusalast, seade võib olla integreeritud rakendusseadmega (telefon, TV-tuuner, pangaautomaat jne).

***Kontaktivaba kiipkaart***
on kiipkaardi ja raadiosagedustõendite (vt 8.3.4.4) hübriid. Valmistatakse ka kaksikliidestusega kaarte, millel on nii standardkontaktid kui ka siseantenniga raadioside. Kontaktivabade kaartide nimi-töökaugused ulatuvad praegu nullist (vahetu kokkupuude terminaliga) meetrini. Kaugus ja edastuskiirus sõltuvad sagedusest; näiteks sagedusel 13,56 MHz on edastuskiirus 100 Kbit/s. Kontaktivaba kaardi võimalusi piirab toitetarve: tüüpiline protsessorkaart tarbib 5–8 mW, kaitsega mälukaart 1–1,5 mW, vajalik võimsus aga kasvab kaardi ja terminali vahelise kauguse kuubiga.

**Joonis 32. Makseotstarbeline kiipkaarditerminal**

Kiipkaardi praegused peamised rakendusvaldkonnad on side (telefonikaart, satelliit-TV dekrüpteerimiskaart), pangateenused (panga-, krediit-, deebetkaart, elektronrahakott, kaugarveldused), tervishoid (kus puudub vanemate vahendite konkurents), transport (ühissõidukid, takso, raudtee, lennuliinid, parkimine, maanteemaks), pääsu reguleerimine, polüfunktsionaalsed üliõpilas- ja õpilaspiletid (pääs, transport, toitlustus jm).


**Joonis 33. Kiipkaardi kasutamine mobiiltelefoniga**

Lähitulevikus lisandub rakendussfäärile Internet kahel eri tasemel: ühendusteenuse pääsu reguleerimise ja arveldusega ning võrgurakenduste (elektronpost, pangateenused, kaubandus) turbe tasemel. Laieneb ka kasutamine üksikarvutite turbeks. Peamine hetketakistus on üldtunnustatud rakendusstandardite puudumine. PC jaoks töötab rida nimekaid firmasid (Bull, Hewlett-Packard, Microsoft, Schlumberger, Siemens Nixdorf) koostöös välja PC/SC standardeid. Selle standardisarjaga võidakse ühendada IBM-i väljatöötatud raamstruktuur OpenCard võrguarvutite tarbeks.

Isegi standardite ühtlustumisel jääb probleemiks rakenduste ja tarkvara ühilduvus mitmesuguste kaardivariantide riistvaraga. Sõltumatuse võivad anda interpretaatorid, mis võimaldavad programmeerida rakendusi sõltumatult platvormist. Seda ideed realiseerib Java-keelt kasutav JavaCard, mille töötasid välja Gempus ja kiipkaardipioneer Schlumberger. Praegustel kaartidel ei piisa sellise lahenduse praktiliseks rakendamiseks jõudlust, kuid olukord võib mõne aastaga muutuda.

***Pultkaart***

on kaardiformaadis autentimisotstarbeline taskuarvuti. Ta on varustatud näidikuga ning sensorsõrmistega, mis võimaldavad sisestada isikunumbrit, parooli, šifri võtit vms. Tavaliselt sisaldavad parooligeneraatoreid ja krüptotehnilisi vahendeid. Liidesed on firmapärased.


**Joonis 34. Pultkaardi näide**

Joonis 34 kujutatud seadet ei saa õieti enam nimetada kaardiks, ta sarnaneb pigem märkmikarvutiga. Analoogiliste funktsioonidega vahendeid valmistatakse siiski ka kaardikujulistena. Tüüpilisi funktsioone:

- pääsukoodi genereerimine ühenduseks serveriga,
- serveri autentimine,
- teise kaugkasutaja autentimine,
- dokumendi sertifitseerimine,
- tundliku dokumendi signeerimine ja verifitseerimine,
- krüptovõtmete haldus.

Andmevahetus kaarditerminaliga toimub infrapuna- või raadioside kaudu.

***Diskettkaart***

on katse rakendada kiipkaarti PC turbeks, säästes kulutusi suhteliselt kallist kaarditerminalist loobumisega. Fischer International Systems Corp. valmistab kaardikiibi baasil disketikujulist kaarti, mille saab pista tavalisse disketidraivi. Kaardi toidet ja andmesidet vahendab draivi magnetväli.

**Kiipkaardi valmistuslikud turvameetmed**

***Optilise ja füüsikalise analüüsi tõrje***

1. Püsimälus olev kood on nähtamatu. Varasemat transistoridevaheliste ühendustega kodeeritud struktuuri oli lihtne optiliselt lugeda. Praegu toimub kodeerimine lisandite tiheduse muutmisega transistorides. Transistorstruktuur kaetakse optilist analüüsi takistava erikihiga.
2. Kiibi kompositsioon on hajus. Varasemates kiipides olid funktsionaalüksused (protsessor, püsimälu, muutmälu jne) visuaalselt selgesti eristatavad. Praegustes tõendikiipides on need üksused hajutatud üle kogu kiibi ja nii-öelda "läbi segatud".
3. Juhtmestik kiibil on kahekihiline ning optiliselt raskesti jälgitav. Analüüsi muudavad veelgi tülikamaks "tühjad" pettejuhtmed.

***Elektrilise analüüsi tõrje***

1. Elektrilise analüüsi käigus pannakse kiip pingete ja voolude mõõtmiseks tööle väga madalal sagedusel. Nüüdisaegne kiip on konstrueeritud nii, et ei hakka madalatel sagedustel tööle.
2. Püsimällu kirjutatakse andmed topograafiliselt hajutatult, nii et on raske kindlaks teha, milline on bittide järjestus.
3. Valmistamisel kiibi testimiseks kasutatavad eriviigud desaktiveeritakse füüsiliselt, nii et nende kaudu puudub pääs kiibi sisemusse.

4. Kiibi juhtmestik on väga täpselt arvestatud teatavale tööpingele. Kui analüüsija kasutab veidi kõrgemat pinget, põleb juhtmestik läbi.

### *Võltskiipide valmistuse tõrje*

1. Valmistustehnoloogia nõuab sadade miljonite dollarite suurust investeeringut seadmetesse ning töötajate väga kõrget kvalifikatsiooni.
2. Spetsiifilise rakenduse jaoks määratud kood integreeritakse kiibi valmistusmaski väljatöötajate ja valmistajate koostöös. Seega ei ole mask tüüpne.
3. Kõik tõendkiipide operatsioonisüsteemid on firmapärased, aktsepteeritav käsustik on piiratud. Valmistajad opsüsteemi lähtekoodi ei avalda.
4. Andmete paigutuse ja pea-krüptovõtmed määrab kaardi väljaandja kiibi initsialiseerimise faasis ning neid teab ainult tema.
5. Ranged organisatsioonilised meetmed väldivad valmistuseks vajaliku teabe lekked valmistusfirmast.

### *Kiibi sisu muutmise tõrje*

1. EEPROM-mälu sisu kaitseb modifitseeriva ultraviolett-, röntgen- ja elektromagnetilise kiirguse eest spetsiaalne kattekiht.
2. EEPROM-mälu sisu muutmiseks tuleb anda mitu teatavas järjestuses käsku. Käsujada äraarvamine on võrreldav dekrüpteerimisülesandega.
3. Teatavate EEPROM-mällu salvestatud andmete tervikluse kontrolliks arvutatakse nende räsikuju, mis salvestatakse kontrollregistrisse. Kontrollkoodide lahknemisel andmepöördus blokeeritakse.

## Kiipkaardi kasutuslikud turvameetmed

Kaardile paigutatavad andmed, nende turbe moodused, kaardi infrastruktuuri turve jms sõltuvad kaardi rakendusest ja nõutavast turbetasemest. Lühiülevaate annab Tabel 11.

**Tabel 11. Kiipkaardisüsteemi turbetasemed**

| Parameeter | Turbetase | | |
|---|---|---|---|
| | **Madal** | **Keskmine** | **Kõrge** |
| Konfidentsiaalsus | Andmed salvestatakse avatekstina | Kaitsega mälukaart pääsu reguleerimisega | Andmed krüpteeritud või turvalisel mälukaardil |
| Kaardi valdaja autentimine | Puudub (volituseks on kaardi valdamine) | PIN | Biomeetriline |
| Kaardi autentimine | Kaardi identifikaator PROM-is | Individuaalne krüptograafiline kontrollkood kaardil | Kahepoolne dünaamiline autentimine (nullteadmusprotokoll) |
| Terminali autentimine | Puudub (valdaja usaldab terminali või mitte) | Krüptovõtme sertifikaat terminalis | Kahepoolne autentimine |
| Side terviklus | Sõnumi CRC | MAC sõnumi numbri, kuupäeva, kellaaja ja võtmeandmetega | MAC, kviteerimistega |
| Salvestuse terviklus | Arvutifailide kontrollkoodid, igapäevane varukopeerimine | Kõik andmed kontrollkoodidega, täielik tehingute logi | Kriitiliste väljadega seotud MAC; tähtsad andmed ka varifailides |

**Kiiplipik**

Kiiplipik (*contact memory tag*, "kontaktmälulipik") on kiiptõendi konstruktsioonivariant, mis põhimõtteliselt ei erine kiipkaardist. Algselt töötati ta välja (1991, Dallas Semiconductor) mälu sisaldava, mehaaniliselt ja termiliselt väga vastupidava kapslikesena, millega sai märgistada masinaid, detaile, kariloomi, postisaadetisi, lennupagasit jms automaatse andmehõive eesmärgil.

Eksootiline rakendusnäide on Pariisi prügiurnide märgistamine, mis võimaldab prügivedu arveldada kaalu järgi (kaalu teatab prügiautosse ehitatud automaatkaal).

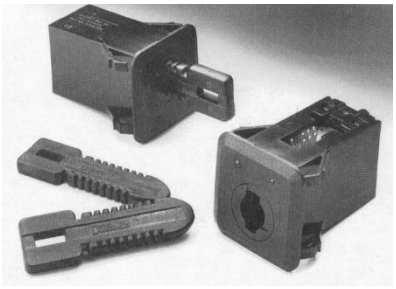
Elemendi tüüpiline kuju on 6 mm paksune metallkapslike läbimõõduga 16 mm. Väiksemate näide on ∅3,5×0,8 mm (firma Valgay). Tüüpiline mälumaht on 1K, suurim kuni 2 MB. Mälu võib oma tüübilt olla ROM, RAM või WORM (viimane võimaldab andmeid lisada, mitte aga muuta). Lipik võib sisaldada oma toiteallikat (millest piisab kümneks aastaks) või saada toidet lugemis-kirjutusvahendilt.

**Joonis 35. Kiiplipiku lugemine välitingimustes**

Pliiatsikujuline lugemisseade on lihtne ja odav, kirjutuseks valmistatakse lihtsaid pihupulte (vt Joonis 35) ning kandearvuti liideseid. Andmevahetus toimub otse kesta kaudu; kui see on kinnitatud masinale või metallkonteinerile, võib lugeda ja kirjutada suvalise punktiga ühendudes. Lipikuid saab isegi ühendada Internetiga.

Viimastel aastatel on valik laienenud, on ilmunud protsessoriga ja koguni kontaktivaba tööd võimaldava optilise sidega variandid. Rakendused on nihkunud ka turbe alale.

- Lipikuid kasutatakse registreerivate kontrollkelladena, märgistades nendega territooriumil kohad, mida valvur peab regulaarselt läbima.
- Dallas Semiconductor valmistab krüptofunktsioonidega lipikuid lauaarvutite ja kaugarvelduse turbeks. Kaotamise puhuks on lipikusse programmeeritud valvetaimer, mis tühjendab SRAM-mälu.
- Autovõtmesse paigutatud lipik välistab käivituse valevõtmega (vt ka Joonis 36).
- Analoogilist süsteemi kasutatakse seifides: lisaks kombinatsiooni valimisele tuleb kasutada lipikuga võtit. seifi elektronlukk registreerib kõik avamis- ja sulgemisajad ning vastavate võtmelipikute andmed.
- Valgay pisilipikuid on ehitatud krediitkaardi suurustesse rinnasiltidesse, millele võib anda kõik tavalise kiipkaardi funktsioonid. Pääsulipikuid on ehitatud ka sõrmustesse.

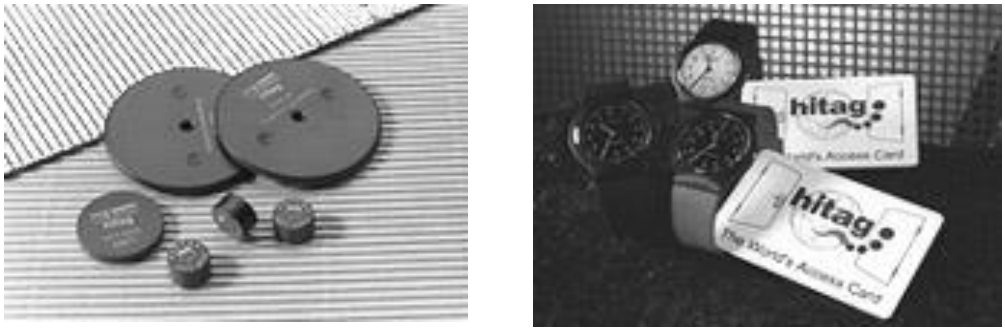


**Joonis 36. Kiipvõtmega lukklülitid**

Kiiplipikute laiemat levikut autentimisvahendina takistab ühtsete standardite puudumine. Eri firmade tooted ei ühildu omavahel ega ka traditsioonilisemate kasutuselolevate vahenditega.

### *8.3.4.4 Raadiosagedustõendid*

Ka 1980il aastail alguse saanud raadiosagedustõendid (RFID, *radio-frequency identification*, "raadiosageduslik identimine") siirdusid turbe valdkonda objektide automaatse identimise alalt. Erinevalt kiiplipikutest olid nad siiski algusest peale orienteeritud ka turbele, sest õieti kasvasid nad välja lihtsatest kaupade turvamärgistuse metall-lipikutest, mis on meie kauplustes tänini kasutusel. RF-tõendite klassikaline väliskuju on tableti- või münditaolised metallkapslid (tüüpilised mõõtmed on näiteks ∅12×6 ja ∅25×3 mm), kuid neid paigutatakse ka käekelladesse ja standardformaadis kaartidesse (vt Joonis 37). Gabariidid ulatuvad riisitera suurusest (kasutatakse näiteks loomade rände uurimiseks) mobiiltelefoni suuruseni (merekonteinerite "saateleht", integreeritud GPS-lokaliseerimissüsteemiga).




**Joonis 37. Raadiosagedustõendite teostusvariante**

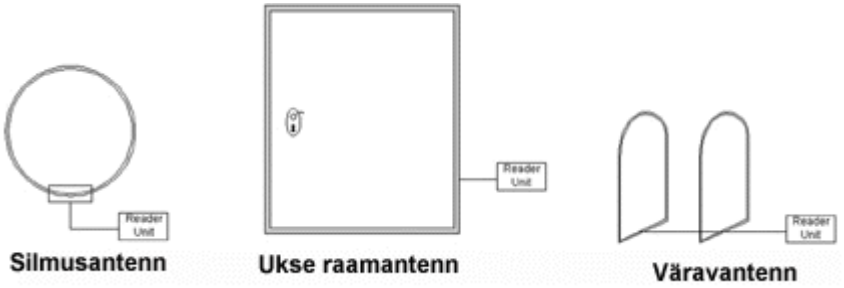
Kiip sisaldab lisaks mälule ja loogikale (on küll ka protsessoriga variante) analoog-digitaalmuunduri, RF-transiiveri ja antennisilmuse. Õigupoolest eristavad RF-vahendeid kiipkaardist või -lipikust vaid kiibile lisatud raadiosidevahendid. Töökaugus sõltub tüübist (vt Tabel 12) ja vastuvõtjast; tavaline maksimaalkauguste skaala ulatub mõnest sentimeetrist 80 meetrini.

**Tabel 12. Raadiosagedustõendite andmeid**

| Parameeter | Variandid | Omadused |
|---|---|---|
| Toite olemasolu | Aktiivsed | Mälu kuni 1 MB (R/W)<br>Suuremad, kallimad<br>Suurem töökaugus, tööiga 3 kuni 10 aastat |
| | Passiivsed | Mälu tavaliselt 4-16 baiti (R/O)<br>Väiksem töökaugus, pikem tööiga |

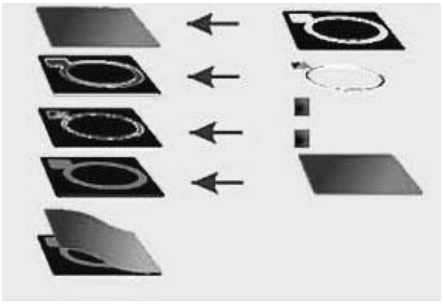
| | | |
|---|---|---|
| Mälu tüüp | R/O<br>R/W<br>WORM | Levinuimad mälumahud on<br>8 baidist 512 baidini |
| Sagedusala | HF: 0,85-0,95 ja 2,4-5 GHz | Töökaugus võib ulatuda üle 100 m<br>Suur andmevahetuskiirus<br>Tõkketundlik, nõuab "silmsidet" vastuvõtjaga<br>Kallis |
| | MF: 10-15 MHz | Kasutatakse ka kaardina ja pääsu reguleerimiseks |
| | LF: 100-500 kHz | Väike võimsustarve<br>Vähem suuna- ja tõkketundlik<br>Kasutatakse ka pääsu reguleerimiseks |

Vastuvõtuantenni (vt Joonis 38) ja vastuvõtuaparatuuri ehitus sõltub konkreetsest rakendusest ja süsteemi funktsioonidest.




**Joonis 38. Raadiosagedusautentimise antennid**

Identimiskaardi valmistus raadiosageduselementi pooltootena kasutades on väga lihtne. Selleks on mitu tehnoloogiat, sealhulgas külmlamineerimine (vt Joonis 39), mis on jõukohane ka väikefirmadele. Tavalise kiipkaardiga võrreldes on selline kaart mehaaniliselt märksa vastupidavam.



**Joonis 39. RF-kaardi valmistus külmlamineerimisega**

Traditsioonilised RF-elementide rakendusalad on kaupade, pagasi, sõidukite, loomade jne märgistamine nende liikumise jälgimiseks, detailide automaattöötlus tootmisel ja ladustusel, kaupade turve, autode turve (süütevõtmesse ehitatud immobilisaator). Elemente on paigutatud isegi golfipallidesse nende leidmise hõlbustamiseks.

Pääsu reguleerimise rakendustes on RF-tehnika oluline eelis selles, et vastuvõtja saab korraga töödelda mitme tõendi signaali. Seetõttu sobib ta suurte inimhulkade pääsu reguleerimiseks (ettevõtete pääslad, metroo). Tõend on väga raskesti võltsitav ning suhteliselt odav (märgise hinnaskaala on 10 kuni 100 dollarit). Riiderite hinnad (tarkvarata) on 1000–4000 dollarit.

Puuduseks on tundlikkus häiringutele, mida võivad tekitada metallesemed, keskkonna kiirgusallikad ning läheduses töötavad teised RF-süsteemid.

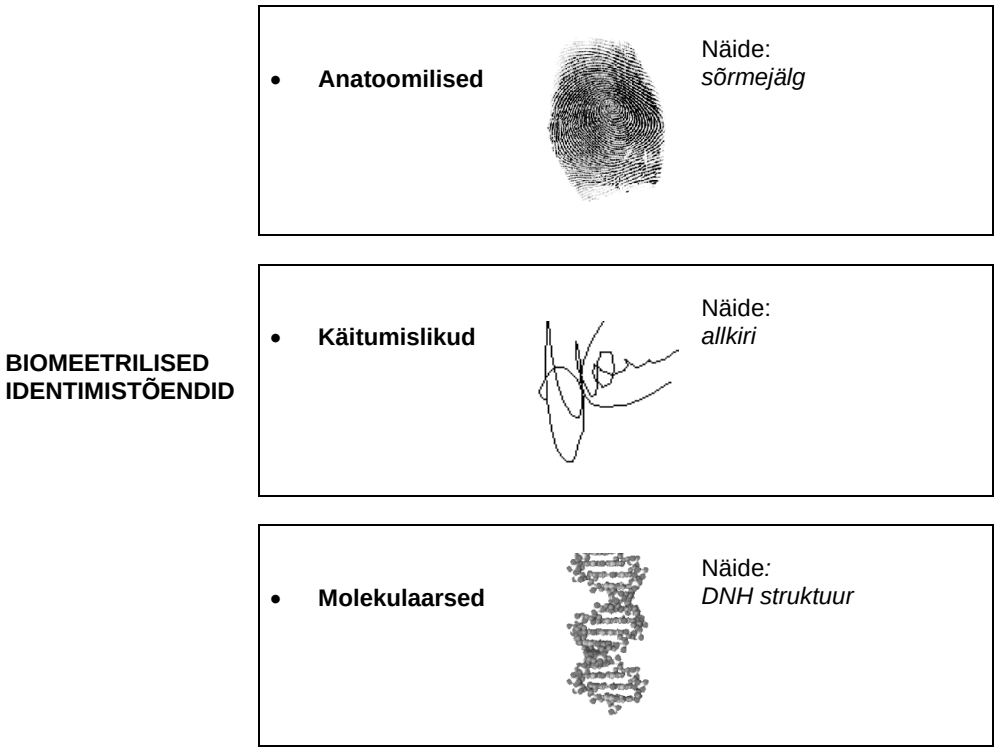
Valmistajaid on praegu paarkümmend, süsteemid on firmapärased, üldisemad standardid on alles koostamisel. ISO seniilmunud standardid (11784:1996 ja 11785:1996) puudutavad rakendust loomade jälgimiseks.

## 8.4 Biomeetrilised autentimistõendid

### 8.4.1 Biomeetriline autentimine

#### *8.4.1.1 Olemus*

Biomeetriline autentimine on erinevalt seni vaadelduist ainult isiku autentimine. Ta põhineb inimese organismi või käitumise individualiseerivatel elementidel. Biomeetriline identimistõend on inimese isikuomane praktiliselt kordumatu bioloogiliste või psühholoogiliste tunnuste kogum, mida saab kasutada tema väidetava identiteedi verifitseerimiseks (vt Joonis 40). Autentimiseks kasutatavad biomeetrilised tunnused erinevad näiteks isegi identsetel kaksikutel.




**Joonis 40. Biomeetriliste identimistõendite liigid**

Biomeetrilise autentimise protsess koosneb järgmistest faasidest:

1) autentimistõendi etalonnäidise sisestus,
2) eristavate tunnuste väljaeraldamine näidisest, tõendi digitaalse malli loomine,
3) autentimistõendi hetkenäidise sisestus,
4) hetkenäidise verifitseerimine malli alusel.

Biomeetrilist autentimist tuleb eristada *biomeetrilisest tuvastusest*, st hetkenäidise alusel sooritatavast otsingust näidiste andmebaasis. Algselt rakendati biomeetrilisi meetodeid ainuüksi tuvastuseks, peamiselt kriminalistikarakendustes. Praegugi on mitme vahendi puhul võimalikud mõlemad rakendused, mõne tõendiliigi puhul aga võib vahendite keerukus, maksumus ja juriidiline jõud tugevalt sõltuda sihtrakendusest.

#### *8.4.1.2 Täpsus*

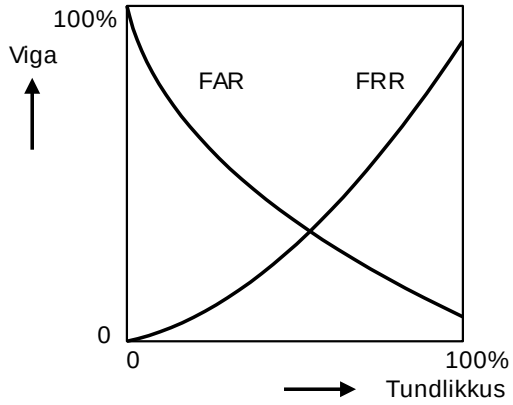
Erinevalt eespool vaadeldud autentimismeetoditest toimib biomeetrilisel autentimisel juhuslikkus, biomeetrilise tõendi eri eksemplaride täpse ühtumise tõenäosus on üliväike. (Sellel asjaolul on ka

positiivne külg: ta võimaldab tõrjuda kopeerimisründeid.) Niisiis on verifitseerimisotsused loomult statistilised ning saavad olla seda täpsemad, mida püsivamaid tunnuseid õnnestub tõendis leida.

Iga statistiline otsustusprotsess annab mingi tõenäosusega väärtulemeid, kusjuures viga saab olla üks kahest:

1) tõendi ***väär mitteaktsepteerimine*** (*false rejection*) ehk esimest tüüpi viga;
2) tõendi ***väär aktsepteerimine*** (*false acceptance*) ehk teist tüüpi viga.

Biomeetrilise autentimissüsteemi täpsust iseloomustavad nende vigade tõenäosused (tavaliselt esitatakse protsentides), vastavalt ***FRR*** (*false rejection rate*) ja ***FAR*** (*false acceptance rate*). Need vead on pöördsõltuvuses (vt Joonis 41).




**Joonis 41. Verifitseerimisvigade sõltuvus eristusmeetodi tundlikkusest**

Erinõuete puudumisel püütakse verifitseerimisalgoritm häälestada nii, et FRR=FAR. Tegelikult sõltuvad nõuded rakendusest.

Näiteks sõnastas briti pangandusorganisatsioon 1993. a nõuded, mille kohaselt FRR on mitte üle 0,001% ja FAR mitte üle 5%. Nõuded võivad esmapilgul tunduda kummalised, kuid FRR minimeerimine tähendab kliendi solvamise või ärritamise riski kahandamist; niisiis peetakse seda briti panganduses oluliseks. Suhteliselt lõtva FAR väärtust seletab kompleksse turbe rakendamine: selline FAR on näiteks kiipkaardi lisameetmena küllaldane üldist turvet tunduvalt ja pangarakenduste jaoks piisavalt tugevdama. Seevastu nõutakse USA tuumaenergeetiliste objektide pääsusüsteemides kasutatavailt vahendeilt FAR väärtust mitte üle 0,1%.

1997. a sooritatud tootevõrdlused näitasid, et biomeetrilised süsteemid suudavad saavutada FAR väärtusi 0,0001%...0,1% ja FRR väärtusi 0,0007%...1,0%.

### *8.4.1.3 Identimistõendi digitaalmall*

Identimistõendi näidise (nt käelaba kujutise) sisestamisel ei salvestata ta kujutist harilikult tervikuna. See raiskaks liigselt mälu ja aeglustaks verifitseerimist, pealegi salvestataks juhuslikkust sisaldavad muutuvad elemendid, mis toimiksid autentimisprotsessis mürana.

Iga identimistõendite tüübi kohta on välja selgitatud püsivad individualiseerivad tunnused, mille arv oleks verifitseerimiseks piisav. Sellist etalontunnuste kogumit nimetatakse biomeetriliseks ***malliks***. Mida väiksem on mall, seda kiirem ja ökonoomsem on meetod. Malli tüüpiline maht on mõnikümmend kuni mõnisada baiti. Malli moodustamisel võidakse kasutada pakkimis- ja krüptoalgoritme.

Mallide loomiseks ja nende alusel verifitseerimiseks rakendatakse kujutuvastuses kasutatavaid statistilise modelleerimise, dünaamilise plaanimise, hägusa loogika ja neurovõrkude analüüsi meetodeid. Nende meetodite areng tekitaski viimasel kümnendil biomeetriliste autentimisvahendite järsu läbimurde, nii et

nad saavad hakata hinnalt ja turbe tugevuselt võistlema esemelistega. Lisaks malli kompaktsusele on oluline ka algoritmide, eriti verifitseerimisalgoritmide kompaktsus ja kiirus.

### 8.4.2 Biomeetrilistele vahenditele esitatavad nõuded

Üleeuroopalise standardimise käigus on loetletud alljärgnevaid esialgseid nõudeid, neid täpsemalt spetsifitseerimata. Neid nõudeid saab kasutada toodete või meetodite võrdlemiseks.

#### *8.4.2.1 Kasutuslikud nõuded*

1. Kasutamise mugavus, st võimalikult minimaalne

- ajakulu registreerimisele, identimisele ja verifitseerimisele,
- kasutaja toimingute arv,
- kasutaja väljaõpe,
- mõõdetav ja salvestatav andmestik.

2. Sobivus keskkonda:

- kasutajasõbralikkus, harjumuslikkus, hõlpsus,
- kasutaja turvalisus ja privaatsus,
- sobivus konkreetseks rakenduseks (nt ei sobi sõrmejälg seal, kus käed peavad olema vabad),
- eetilisus, sotsiaalne ja kultuuriline vastuvõetavus,
- ühilduvus,
- töökindlus ja hooldatavus.

3. Psühholoogiline vastuvõetavus:

- pealetungimatus (nt füüsilise kontakti puudumine mehhanismiga),
- diskrimineerimatus (soo, vanuse, füüsilise seisundi, ametikoha jne järgi).

4. Turbetugevus:

- võltsimatus ja pettusekindlus,
- füüsiline ja juriidiline tugevus,
- ühesus (tulem peab olema ühene),
- püsivus (mehhanism ei tohi muutuda ega olla muudetavus),
- välistavus (ühegi teise autentimisvahendi kasutamine ei tohi olla tingimata vajalik).

#### *8.4.2.2 Tehnilised nõuded*

1. Võimalikult minimaalne autentimisaeg:
- kasutaja ja süsteemi ettevalmistusaeg,
- bioandmete võtu aeg,
- verifitseerimisaeg,
- mõõte- ja salvestusaeg,
- malli mälutarve.

2. Täpsus:
- aktsepteerimise ja mitteaktsepteerimise lävede reguleeritavus sõltuvalt nõutavast turbetasemest,
- FAR ja FRR võimalikult väikesed väärtused,
- adapteeruvus.

3. Paindlikkus
4. Toimivus
5. Jõudlus
6. Ühilduvus, interoperaablus
7. Lihtsus
8. Sõltumatus keskkonnatingimustest (müra, valgustus, kiirgus, niiskus, tolm, temperatuur jne)

#### *8.4.2.3 Majanduslikud nõuded*

1. Aparatuuri maksumus
2. Paigalduse maksumus
3. Koolituskulud
4. Värskenduskulud ja -aeg
5. Autentimiseks vajalikud arvutiressursid
6. Vahendi kaitse kulud
7. Liidestuskulud
8. Haldus- ja hooldekulud

#### *8.4.2.4 Nõuded meetoditele*

1. Algoritmide ja nende teostuse õigsus
2. Algoritmide turvalisus:
- matemaatiline (murdmismeetodite puudumine piiramatute arvutusressursside puhuks),
- arvutustehniline (piisavate murdmisressursside puudumine praegu ja tulevikus).

3. Turvaline andmebaas: mallide turvaline salvestus ja haldus
4. Turvalised protokollid
5. Turvalised võrgud ja muud jaotussüsteemid

### 8.4.3 Anatoomilised tõendid

Põhinevad inimese organismi mitmesuguste osade individualiseerivatel püsitunnustel. Moodustavad suurima biomeetriliste tõendite ja neil põhinevate toodete rühma.

#### *8.4.3.1 Sõrmejälg*

Üks vanimaid biomeetrilisi tõendeid, praegugi levinuim. Staatilisi sõrmejälje kujutisi võrreldi tuvastuse eesmärgil arvuti abil juba 1960il aastail. Tuvastusrakendusele spetsialiseeritud vahendid on tänini keerulised, kallid ja suhteliselt aeglased. Autentimisrakenduste alal saavutati arvestatavaid tulemusi alles viimasel kümnendil.

Tunnustik on stabiilne, korduvuse tõenäosus on väiksem kui $10^{-9}$. Traditsiooniliselt on digitaalmall olnud teiste tõenditega võrreldes suur (sajad baidid või isegi üle 1K), alles 1994. a jõuti 60-baidise mallini. Meetodid on kahesugused, ühed neist opereerivad papillaarjoonte spetsiifiliste punktide (joonte otspunktid ja hargmikud) asukohtadega, teised (nt Identixi toodetes, Šveits) rakendavad kogu sõrmejälje kujutisele tervikuna kujutuvastuse meetodeid. Ühildamiseks seniste sõrmejäljearhiividega kasutavad tuvastusotstarbelised süsteemid ka klassikalisi daktüloskoopilisi tunnuseid (kaar, aas, keeris jms, nt Henry süsteem toetub kaheksale põhikujundile). Papillaarjooni analüüsitakse enamasti Fourier' teisenduste abil sagedusmõõtmes, mitte ruumimõõtmes.

Peamised skaneerimismeetodid on seni olnud optilised; on kasutatud CCD-pildiandurit, aga ka ühemõõtmelist kiirega skaneerimist. Teravaimad kujutised on saadud sõrme surumisega kolmnurkprisma ühele tahule (teise tahu kaudu suunatakse sõrmele valgus ja kolmandal tahul on andur). Viimasel ajal püütakse ka optilise skaneerimisega saavutada ruumilist kujutist; selleks on mõnedes süsteemides skaneerimisaknake kaetud läbipaistva elastomeeriga, mis väldib papillaarjoonte laiakslitsumist. Ruumilist kujutist on hõlpsam saada ultraheliskaneerimisega; esimene selline süsteem oli UltraScan (Niagara Technology Laboratories). Skannerid on piisavalt väikesed kasutamiseks näiteks lauaarvuti välisseadmena (vt Joonis 42).

**Joonis 42. PC järjestikpordiga ühendatav sõrmejäljeskanner**

Revolutsiooniliseks võib lugeda 1998. a väljalastud elementi FingerLock (Harris Semiconductor), miniatuurse skanneriga integreeritud töötluskiipi, mis vabalt mahub tikutoosi (elemendi maht on 7,5 $cm^3$). Ruumiline skaneerimine toimub elekriväljaga, mille tekitab sõrme all mikroelektroodide maatriks. Ära jäävad optikamoonutused, masstootmine viib hinna sajast dollarist allapoole. Sellist elementi on paigutatud juba isegi kiipkaardile, kaitsevahendina kaotamise või varguse puhuks.

Täpsus on eri toodetel äärmiselt erinev. Parimad saavutatud tulemused on FAR < $10^{-6}$ ja FRR < $10^{-9}$. Keskmised näitajad on veidi tagasihoidlikumad, nt 60-baidise malliga Thorni süsteemil on FAR=0,01%, FRR=0,1%.

Verifitseerimisaeg on eri toodetel 0,1 kuni 1 sekundi piires.

Praktikas on autentimisrakendused juba üsna levinud. Näiteks Los Angelesi sotsiaalosakond kasutab sõrmejälge juba 1991. aastast alates abirahade korduva väljamaksmise vältimiseks.

Rakendamist piiravaid tegureid: sõrmevigastused, kinnastatult töötamise vajadus, töölistel saastumine, psühholoogiline vastumeelsus (sõrmejälge seostatakse eelkõige kriminalistika, kuritegevuse ja politseiarhiividega). Mõnedel aasia rahvastel on papillaarjooned liiga vähe reljeefsed.

### *8.4.3.2 Sõrm. Käsi*

Käelaba ja sõrmede geomeetria rakendusvõimaluste uurimine algas 1971. a ning vastavaid süsteeme on kasutatud juba üle 20 aasta. Mõõdetud on sõrmede pikkust ja proportsioone, käelaba paksust, peopesa kuju ja joonestikku jt elemente.

***Sõrme***, õigemini sõrmepaari ruumilisel geomeetrial põhineb näiteks süsteem Digi-2 (BioMet Partners, Šveits), mis opereerib 20-baidise malliga; verifitseerimisele kulub alla sekundi, tuvastus 100 näidise hulgas kestab alla kahe sekundi.

***Nahapooride*** topograafiat on uuritud peamiselt sõrmeotstel. Pooride arv sõrmetipul on umbes 2000, tuvastuse ja verifitseerimise aluseks võetakse 20–50. Meetodit on kasutatud kombineeritult sõrmejälgede analüüsiga; muuhulgas väldib selline lisatunnustik näiteks sõrmejälje imiteerimise.

***Käelaba***

Juba üsna laialt kasutusel olev ning perspektiivne biomeetriline tõend. Peamised vastunäidustused on sõrmuste, vigastuste, paistetuse häiriv mõju, Parkinsoni tõbi jms. On ka vastuseisu kultuuritraditsioonide pinnal, näiteks Jaapanis.

Meetodeid on juba üsna mitu, neist ja nendega saavutatud tulemustest võib saada mingi ülevaate turustatavate autentimistoodete alusel.

Edukaim on olnud firma Recognition Systems, kelle süsteemi ID-3D HandKey (skaneerib kujutise ruumilisena) müüdi 1996. a 5000 tk. Selles süsteemis on realiseeritud kõige väiksem teadaolev biomeetriline mall (9 baiti), ka muud näitajad on piisavalt head: FAR<0,1%, FRR<0,1%, verifitseerimisaeg 1 sekund, tuvastusaeg võrdlemisel 11000 näidisega alla 5 sekundi. On kasutusel muuhulgas tervishoiusüsteemis, USA ja Kanada lennujaamades, vanglates jm.

**Joonis 43. Kontrollkellaga integreeritud käelabaskanner**

Palmetrics kasutab ühemõõtmelist optilist skaneerimist, mis hõlmab umbes 10000 punkti. FAR =0,00025%.

HandMark XO (PIDEAC) skaneerib IP-kiirgusega rusika liigesenukke, tuvastuseks piisab kolmest liigesest. Skaneerimise ajal peab käsi vajaliku asendi saavutamiseks pigistama ümarat pidet, mis käivitab autentimise. Eeliseks on sõrmuste jms häiriva toime puudumine.

***Käe veenid***

Käeselja veenide mustri analüüs on üks uuemaid biomeetrilisi meetodeid. Süsteem Veincheck (British Technology Group) registreerib CCD-pildianduri abil käeselja kujutise. Käsi peab naha pingutamiseks pigistama ümarpidet, kasutatava valguse lainepikkus on IP-kiirguse lähedal (sellel lainepikkusel on valguse neeldumine hemoglobiinis maksimaalne). Kujutis sisestatakse hallskaalas, seejärel muudetakse kahetasemeliseks kontrastpildiks (vt Joonis 44).

**Joonis 44. Käeveenide skaneerimine: täielik kujutis, kontrastpilt, XOR-võrdlus**

Veenide eeliseks on see, et muster on suur, stabiilne, vigastusohutu, muutumatu; piisab väikesest eraldusvõimest. Algoritmid on lihtsad, seetõttu verifitseerimine kiire (alla 0,2 s). Häirida võivad naha anomaaliad (armid, soolatüükad). Meetod on seni olnud orienteeritud pigem tuvastusele kui autentimisele.

### *8.4.3.3 Silm*

Silmal on teiste identimistõendite ees mitmeid eeliseid: erinevalt keha pealispinnast on ta paremini kaitstud vigastuste, saastumise jms eest, on ajas tunduvalt püsivam ega ole kunstlikult muudetav. Puudub kontakt autentimisseadmega, seetõttu on psühholoogiliselt vastuvõetavam. Reaktsioon valgusele võimaldab eristada elutust koopiast. On üks suurimat täpsust võimaldavaid biomeetrilisi vahendeid. Verifitseerimist ei mõjuta prillid ega kontaktläätsed.

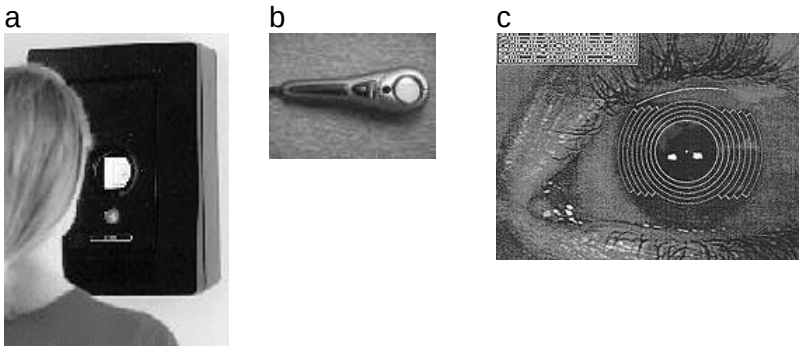
Kasutatud on kaht tunnustikku: silma võrkkesta veresoonte mustrit ja silma vikerkesta mustrit.

***Võrkkesta*** soonemustri kasutamine verifitseerimiseks patenditi 1978. a (R.Hill). Verifitseeritav suunab pilgu okulaari ja vaatleb silma asendi fikseerimiseks värvilist täppi. Silmapõhja valgustamiseks kasutati algul väikese võimsusega laserikiirt, psühholoogilise vastuseisu tõttu siirduti hiljem välk- ja siis hõõglambile. Kasutatav spekter on IP-lähedane.

Ainsa tootena on saadaval süsteem EyeDentify samanimeliselt firmalt. Mall on 48 baiti, FAR ja FRR praktiliselt nulli lähedased. Verifitseerimisaeg on alla 1,5 sekundi, tuvastusotsing 300000 hulgast alla 15 sekundi, 4000 hulgast umbes 3 sekundit.

***Vikerkest*** on keeruka struktuuriga ja sisaldab tuhandeid individualiseerivaid elemente. Mustri korduvuse tõenäosus on $1^{-78}$. Meetodile panid 50il aastail aluse oftalmoloogid F.Adler, L.Flom ja A.Safir, automaatse verifitseerimiseni jõuti 80il aastail.

Edukaim toode on sari IriScan samanimeliselt firmalt. Silm skaneeritakse 15-25 cm kauguselt, kaugust võib suurendada meetrini. Skaneeritakse CCD-pildianduriga, kasutada võib paljusid videokaamera tüüpe või spetsialiseeritud skannerit (vt Joonis 45). Analüüsimeetod töötati välja 1995. a ja põhineb Gabori lainekestel. Vikerkesta kujutisest eraldatakse 256 muutujat, mis kodeeritakse 512-baidisesse vahevormingusse (värvikoodiriba Joonis 45, c). Pärast eeltöötlust ja pakkimist ei ületa malli maht 35 baiti. FRR=FAR<0,0001%. Koodi arvutus kestab sekundi, kogu verifitseerimisprotsess alla 3 sekundi. Tuvastusotsing 100000 hulgast kestab 2 sekundit.




**Joonis 45. Vikerkesta skaneerimine: a - pääsujaama integreeritud skanner, b - käsiskanner, c - skaneeritud tunnuste eeltöötlus**

Vikerkestal põhinev autentimine on esimene biomeetriline meetod, mis võeti kasutusele pangaautomaatide turbeks (esimene toode 1998).

### *8.4.3.4 Nägu*

Näo staatilise kujutise automaatse tuvastusega tegeldi juba 60il aastail, peamiselt fotokartoteekide automaattöötluse eesmärgil ning tuvastavate fotorobotite loomiseks. Peamisi tunnustikke on valitud lähtudes kriminalistikas väljaarendatud süstemaatikast (nina suurus, silmade, lõua, kulmude, suu kuju

jms). Seadmed on seni olnud suhteliselt kallid, kuid multimeediumi areng on muutmas videokaameraid lauaarvutite odavaks välisseadmeks, seetõttu on tooted omandanud tarkvarapakettide kuju, skaneerimisaparatuur on teisejärguline, põhiliselt on tooted orienteeritud tavalistele video- ja termokaameratele. Kui arvuti monitorile on alaliselt paigutatud kaugkonverentside pidamiseks mõeldud videokaamera, saab seda edukalt kasutada arvuti turbeks: pettuste vältimiseks on eelistatavad just sellised biomeetrilised vahendid, mis võimaldavad kasutajat autentida mitte ainult sisselogimisel, vaid ka töö käigus.


**Joonis 46. Näo tuvastus: a - skaneerimine videokaameraga, b - ruumiline skaneerimine pääsujaama ehitatud kahe pildianduriga, c - tunnustike eraldamine töötluseks**

Täpsus ja pettusekindlus on esialgu muude meetoditega võrreldes suhteliselt madalad, kuid olukord võib kiiresti muutuda. Arendustöö on käimas MIT-s, Harvardi ülikoolis, Pentagoni tellimusi täitvates firmades, mitmes Jaapani suurfirmas. Meetodid jagunevad peamiselt kaheks: ühed põhinevad statistilisel korrelatsioonianalüüsil, teised õppivatel neurovõrkmudelitel.

Näol on teiste biomeetriliste tõenditega võrreldes mitmeid puudusi: tuvastus nõuab kindlat pea asendit, nägu muutub vananemisel, omandab väga erinevaid ilmeid, on muudetav grimeerimisega.

Tooteid on siiski juba üsna mitmelt firmalt.

Vision Systems (Neuromatics) põhineb neurovõrkudel, teda saab rakendada ka fotodele ja videosalvestistele. Süsteem suudab välja filtreerida väikesed pea asendist, valgustusest, näoilmest, soengust, meigist jne tingitud muutused.

Viisage Gallery (Viisage Technology) põhineb MIT patendil. Süsteem loob 128-st täisarvust koosneva malli, lisaks sellele salvestab andmebaasi kogu pildi bmp- või jpg-vormingus. Süsteemis registreerimine koos testiga kestab 3 sekundit. Iga subjektiindeksiga saab siduda mitu sama subjekti malli.

TrueFace Access (Miros) rakendab neurovõrke. Malli maht on 750 baiti, verifitseerimisaeg on 1 sekund. Pääsupunkt sisaldab kaks CCD-andurit (fotoga petmise vältimiseks) ning PIN-klaviatuuri või magnetkaardiriideri. Süsteem töötab ka suvalise videokaameraga. Teda saab rakendada lauaarvuti turbeks, PC tarkvarapakett maksab 60 dollarit.

***Kõrv*** on näo analüüsi kõrvalsaadusena hakanud muutuma iseseisvaks identimisvahendiks. Kõrva geomeetria analüüsimiseks sisestatakse telefonitorutaolise skanneriga kogu kujutis. Tuvastusrakendustes on saavutatud positiivseid tulemusi. Ärilisi tooteid turul veel ei ole.

### 8.4.4 Käitumislikud tõendid

Käitumislikud (*behavioral*) tõendid on sellised tunnustikud, mis ei ole kaasasündinud, vaid põhinevad mingil arenemise ja õppimise käigus omandatud individualiseerival stereotüübil. Peamised uurimisobjektid on allkiri ja kõne.

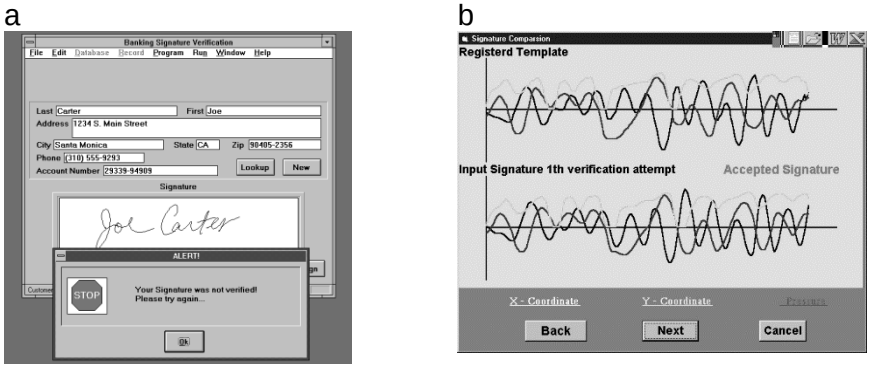
### *8.4.4.1 Allkiri*

Allkirja staatilise kujutise analüüs tuvastuseesmärgil ja eeskätt kriminalistikarakendusteks on juba üsna pika ajalooga valdkond. Autentimisrakendusteks tegeldakse aga eranditult dünaamilise allkirjaga, niisiis õieti mitte allkirja, vaid tema kirjutamise protsessiga. Lisaks staatilise allkirja tunnustele (mõõtmed, ristumiste arv, rõht- ja püstpöördepunktide arv, kirjutusvahendi langetuste ja tõstete arv, joone jämedus jms) mõõdetakse dünaamikaparameetreid (kirjutusaeg, pöörete summaarne aeg, kiirusevektorid, kiirendused, rõhu muutumine ajas jne). Mida rohkem arvestab meetod dünaamilisi tunnuseid, seda pettusekindlam ta on: võltsida saab allkirja välisilmet, mitte aga tema kirjutamise protsessi (vt joonis 55, b). Etalonallkirjaga väliselt täielikult identne allkiri lükatakse tagasi kui koopia.

Dünaamilise allkirja verifitseerimine areneb kiiresti, registreeritud on üle 100 patendi, mh sellistelt firmadelt nagu IBM, NCR, VISA. Tooteid on juba üsna mitmelt firmalt ja nende hind langeb. Esimene alla 1000 dollari maksev biomeetriline vahend üldse oli allkirjasüsteem Sign/On (Electronic Signatures, 1986).

Esimesed süsteemid põhinesid tavalistel graafikalaudadel (mille hind langes 1998. aastaks alla 100 dollari, prognoos näitab hinna peatset langust 50%), hilisematel jagunevad kasutatav skaneerimisvahend peamiselt kahte liiki: (1) raadio- või IP-sidestatud andurpliiats, mis tööpõhimõttelt sarnaneb hiirele, (2) tavaline kirjutusvahend puutelaual. Ainulaadne erand on Rolls-Royce Aerospace Groupi väljatööde, mida 1990te algul edukalt katsetati pangas: süsteem analüüsib kirjutamisel paberi alla levivaid helilaineid, mis salvestatakse metallist kirjutusaluse all asuvate anduritega (seade on tunduvalt odavam tavalistest); kirjutada võib suvalise peenejoonelise kirjutusvahendiga (lai viltpliiats tekitab paberi konarustel liikudes ülemäärast lisamüra).

Analüüsiks kasutatakse peamiselt tunnuste hierarhiaga opereerivaid kujutuvastuse meetodeid ja viimastel aastatel neurovõrke. Teistest erineb 1997. a patenditud Gupta ja Joyce'i (AT&T Bell Laboratories) meetod, mis põhineb lainekeste dünaamikal ning mõõdab 7 parameetrit; saavutatud on FRR=0,5%, kuid FAR on esialgu üle 10%.




**Joonis 47. Allkiriautentimise liidestus: a - allkirja välisilmet ja keeldumisteadet väljastav kasutajaliides, b - allkirja ja malli sisestruktuure esitav haldusliides**

Meetodi võimalustest annavad ettekujutuse mõnede turustatavate toodete andmed.

Countermatch (AEA Tchnology) töötab suvalise sisendseadmega, mis väljastab koordinaatide muutusi ajas. Analüüs põhineb neurovõrgul. Tarkvara verifitseerib ka nt araabia, hiina ja kanji kirjas allkirju. FAR<5%, FRR<2%.

Cyber-SIGN samanimeliselt USA firmalt on tarkvarapakett, mis arvestab rõhu, kuju, suuna ja kiiruse muutusi. Sõltuvalt litsentsist on hind ühe kasutaja kohta 10 kuni 50 dollarit.

PenOp on graafikalauaga töötav tarkvara samanimeliselt firmalt. 42 tunnust sisaldav mall (maht alla 1K) moodustatakse registreerimisel võetava 5-8 allkirjaproovi alusel. Mall salvestatakse krüpteeritud biotõendis (maht 1–2K), mis sisaldab väidetava identifikaatori (nt nime, kontonumbri vms), ajatempli, rakendusest sõltuva kuvatava hoiatusteate, allkirja staatilise kujutise ja MD5-kontrollkoodi.

Verifitseerimisaeg on 0,05 s. FRR ja FAR väärtustest ei saa selle süsteemi puhul kõnelda, sest tulem on mitte üks kahest, vaid autentsuse tõenäosus (0...100%).

Allkirja biomeetriaga tegeldakse isegi endistes sotsmaades. Rumeenias on loodud neurovõrgul põhinev puutelaua ja 8-bitise A/D-muunduriga töötav süsteem, mida õpetatakse 4-10 näidisega (õppeaeg 2...25 s). Mall on 400-baidine, verifitseerimisaeg alla 0,05 ms.

Allkirja kasutamise vastunäidustusi: aparatuur on veel suhteliselt kallis; mõnedel inimestel muutub allkiri kiiresti; probleeme võib olla füüsiliste puuetega (Parkinsoni tõbi) ja kirjaoskamatusega.

#### *8.4.4.2 Kõne*

Kõnehääle tuvastusega on tegeldud juba alates 60ist aastaist. Muude biotõenditega võrreldes on mall mahukas (1K kuni mitukümmend K) ning autentimise täpsus väike. kasutamine on õigustatud ainult keskmiste turbenõuete või suure autenditavate arvuga situatsioonis. Kõnehääl on siiski ainus biomeetriline vahend, mida saab kasutada kaugautentimiseks.

Analüüsimeetodeid on välja töötanud peamiselt side- ja helitehnikaga tegelevad suurfirmad (AT&T, ITT, France Telecom, Bellcore, Texas Instruments, Siemens jt). Kasutatakse nii statistilisi kui ka diskriminantmeetodeid. 1978. a alates rakendatud kõne aegkodeerimine (TES, *time encoded speech*) on verifitseerimisotstarbeks liiga arvutusmahukas. Meetodid on võrdlemisi pettusekindlad: matkimine ei toimi (on orienteeritud inimesele ega saa arvestada olulisi sisetunnuseid, vrd dünaamiline allkiri), salvestise suudavad süsteemid eristada ka DAT-kasseti puhul; peale selle võib süsteem esitada ootamatuid kontrollküsimusi.

Süsteemide hind sõltub kasutatavast aparatuurist: pääsupunkti paigutatud eriaparatuuri maksumus on umbes 1000 dollarit ühe ukse kohta; odavam on lauaarvuti ja tavalise telefoni kombinatsioon, kuid sel juhul on kanali sagedusriba kitsam.

Kõneautentimise peamised takistused on hääle muutumine koos eaga, kõnedefektid, traumad, alkoholi või hambatuimestuse mõju, taustmüra. Seevastu suudetakse välja filtreerida külmetuste, stressi, emotsionaalsete seisundite mõju.

Kõigile puudustele vaatamata on spetsiifiline nõudlus olemas ja rakendusi üsna palju. Pääsusüsteemides kasutatakse kõneautentimist uste, telefoniteenuste, sõidukite jm turbeks. Üks omalaadseid rakendusi on süsteemil VoiceTrack (Norment Industries), millega kontrollitakse vabakäiguvangide kodusolekut automaatse helistamise teel, mis võib toimuda kuni 30 korda päevas. Lähiaastatel võib oodata arvestatavat rakenduste kasvu side- ja võrguteenuste turbes. Laiemat levikut soodustab esimene laia kandepinnaga biomeetriastandard SVAPI (*Speech Verification API*, kõne verifitseerimise rakendusprogrammiliides), mille algvariandi kinnitas 1997. a kaalukaid firmasid ja organisatsioone ühendav erialaliit.

Tüüpilisi võimalusi ja andmeid võib illustreerida paari tootenäitega.

VoiceKey (International Electronics) täpsus lõikepunktitundlikkusel on FRR=FAR=8%, seetõttu tugevdatakse pääsupunkti PIN-koodi ja (suulise) parooliga.

SpeakEZ (T-NETIX) salvestab 5K baiti ühe kõnesekundi kohta. Registreerimisel ütleb isik 3–4 korda parooli; andmed töödeldakse kolme sekundiga ning moodustatakse malli sisaldav 30K suurune kirje. Autentimisel salvestatakse alla 20K, verifitseerimisaeg on 100...200 ms. Süsteem kasutab statistiliste ja diskriminantmeetodite kombinatsiooni.

VOCAL SCW1 (ABS, Saksamaa) on kõneparooliga immobilisaator kuni viiele sama auto kasutajale.

#### *8.4.4.3 Tippimisrütm*

Sai alguse teises maailmasõjas radistide "käekirja" tuvastusega saadud kogemuste rakendamisest klaviatuursisestuse analüüsimisele. Uuringuid on muuhulgas sooritanud NSF ja NIST.

Tippimisrütmi uurimiseks skaneeritakse klaviatuuri umbes 1000 korda sekundis. Malli maht on alla 100 baidi. Meetodi eelised on ta odavus (ei nõua lisaaparatuuri) ja pideva (pealegi märkamatu) kontrolli võimalus. Kontrollimehhanismi saab realiseerida aparatuurselt (püsimälus). On rakendatav ainult professionaalse sisestuskiiruse puhul ning ka siis (madala täpsuse tõttu) ainult lisavahendina. Häirivad tegurid on klaviatuuride erinevused (isegi sama margi eri eksemplaride puhul) ja andmevahetusprotokollide erinevused.

### 8.4.5 Molekulaarsed tõendid

See on biomeetriavahendite uusim, alles kujunemisjärgus olev rühm.

#### *8.4.5.1 Lõhn*

Tõuke lõhna kasutamiseks autentimisvahendina andis umbes 10 aastat tagasi kõrgselektiivsete mitut komponenti määravate gaasianalüsaatorite areng. Bioelektrokeemial, elektrit juhtivatel polümeeridel jms kõrgtehnikal põhinevad "tehisninad" on kasutusel mitmetes keemia- ja toiduainetööstuse harudes, kosmosetehnikas jm.

30 komponendi analüüsile rajatud verifitseerimismeetod töötati välja Leedsi ülikoolis. Koostöös firmaga Mastiff Electronics jõuti 1997. a süsteemini, mis suudab tuvastada inimese kohalolu. Inimest autentiv süsteem Scentinel peab valmima 1998. a. Lõhnaproovi andmiseks paneb autenditav käe restiga kaetud avale; üle käe pinna puhutakse õhujuga. Proovivõtt kestab alla sekundi. Analüüsitavat odorogrammi ei mõjuta parfüüm, seep, määrded ega muud võõrkomponendid.

#### *8.4.5.2 DNA struktuur*

Inimest praktiliselt üheselt määrav tunnus on nukleotiidipaaride järjestus pärilikkust kandva DNA molekulis, kusjuures molekul on identne kõigis sama organismi rakkudes. Esmakordselt rakendati isiku tuvastuseks 1983. a Inglismaal. Protsess on laboratoorne, aeganõudev ja kallis, reaalajas töötavaid süsteeme veel ei ole, kuid perspektiivse täppisvahendina on DNA võetud biomeetria uurimisprogrammidesse. Lahendada tuleb ka psühholoogiliselt vastuvõetav proovivõtu meetod.

### 8.4.6 Biomeetriliste vahendite võrdlus ja perspektiivid

Biomeetriliste vahendite hetkeseisu (mis raamatu ilmumise ajaks võib olla tugevalt muutunud) peegeldab nende põhiomadusi võrdlev Tabel 13.

**Tabel 13. Levinumate biomeetriliste identimistõendite võrdlus**

| Tuvastus-alus | Rakendus-valdkond | Eelised | Puudused | Tuvastusjaama hind ($) |
|---|---|---|---|---|
| Nägu | Universaalne | Hõlbus<br>Kiire<br>Odav | Teeseldav<br>Sõltub valgustusest | 1500 |
| Sõrmejälg | Korrakaitse | Odav | Armide, saaste jms mõju | 1200 |

| | | | | |
|---|---|---|---|---|
| | Suurasutuse andmebaas | Väga turvaline | | |
| Peopesa/käeselg | Tööstusruumid | Väike mälutarve<br>Intuitiivne | Aeglane<br>Sõrmejäljest ebatäpsem | 2150 |
| Silma võrkkest | Tuumaseadmed<br>Raviasutused<br>Karistusasutused | Väga turvaline | Ebamugav | 5000 |
| Termokujutis | Tippturbeobjektid | Väga turvaline | Kallis | >5000 |
| Kõnejälg | Kaugpangandus<br>Kaugandmebaasid | Odav<br>Sobib kaugpöörduseks | Aeglane<br>Sõltub seisundist ja meeleolust | 1200 |
| Allkiri | Majandussfäär | Odav | Sõltub seisundist ja meeleolust | 1000 |

Meetodite täpsust iseloomustavad tabelid Tabel 14Tabel 15, mallide mahtu näitab Tabel 15. Biomeetrilised vahendid on võimelised raskusteta tagama paljudeks rakendusteks vajalikku täpsust; väga suur täpsus on vaid hinna küsimus. Verifitseerimisaeg on kõigil biomeetrilistel vahenditel enamasti alla 2 sekundi.

**Tabel 14. Nõutav ja tegelik täpsus**

| | FRR, % | FAR, % |
|---|---|---|
| 1998. a toodete maksimaalne täpsus | 0,007...1,0 | 0,0001...0,1 |
| Briti panganduse nõue 1993 | mitte üle 0,001% | mitte üle 5% |
| EN 50133-1 (projekt) | mitte üle 1% | mitte üle 0,001% |
| Keskmise turbetaseme nõue | mitte üle 5% | mitte üle 1% |

**Tabel 15. Biotõendite täpsus ja mälutarve**

| | FRR, % | FAR, % | Mall, baiti |
|---|---|---|---|
| Sõrmejälg | 0,1 | 0,0001 | 60...1200 |
| Käelaba | 0,1 | 0,1 | 9 |
| Käeselja veenid | Väike | Väike | 50 |
| Silmapõhi | alla 0,001 | alla 0,001 | Väike |
| Silma vikerkest | alla 0,001 | alla 0,001 | 2048 |
| Dünaamiline allkiri | alla 0,5 | alla 5 | 1000-2000 |

***Vastuvõetavus kasutajale***. Sandia 1991. a korraldatud kasutajauuring reastas biomeetrilised vahendid vastuvõetavuse järgi alljärgnevalt (arvud esitavad poolt- ja vastuhäälte suhet).

| | |
|---|---|
| Käsi | 16,5 |
| Võrkkest | 1,0 |
| Sõrmejälg | 0,8 |
| Allkiri | 0,3 |
| Kõnehääl | 0,2 |

Need tulemused näitavad peamiselt protseduuride psühholoogilist vastuvõetavust ja mugavust autenditavale subjektile. Üldiselt aga eelistaksid riigiasutused, pangad jt autentimisrakenduste tarbijad n-ö "loomulikke" traditsioonilisi identimistõendeid, mis esinevad ka paberdokumentidel: eelkõige näopilti, allkirja ja sõrmejälge. Seetõttu jätkub intensiivne arendustöö neil suundadel. Omaette koht on kõnehäälel kaugautentimise vahendina. Arvuti- ja võrgusüsteemide turbes on eelistatavad sellised vahendid, mis võimaldavad hõlpsat pidevat või regulaarset autentimist töö käigus; selleks sobib hästi näo verifitseerimine, aga ka klaviatuuri või hiirega integreeritud sõrmejäljeskanner.

***Hind*** on hetkel üks kõige otsustavamaid biomeetriliste vahendite levikut määravaid tegureid. Ajavahemikul 1991–1996 langes biomeetrilise pääsupunkti hind viis korda ja on praegu 1000 dollari ümber. Kõne- ja allkiriautentimise vahendid maksavad juba alla 1000 dollari, sõrmejälje või käe analüüsil põhinevate vahendite hind on 900...3000 dollarit. Prognooside kohaselt on pääsupunkti hind aastal 2001 võrreldav kaardiriideri omaga.

Biomeetriatoodete kasv on viimastel aastatel olnud 35% aastas. Aastal 1999 moodustavad nad elektrooniliste pääsu reguleerimise vahendite turust 3%. Kõikjal, kus volitustõend ei pea kandma muid andmeid peale isiku identimiseks vajalike, võib oodata biotõendite astumist esemeliste asemele.

## 8.5 Volitustõendite põhitüüpide võrdlus

Käesolev jaotis annab üldiseid orientiire autentimisvahendite tüübi valimiseks, võttes arvesse turbetehnilisi, kasutuslikke ja majanduslikke aspekte.

### 8.5.1 Turbeomadused

Volitustõendite turbeomadustest teeb võrdleva lühikokkuvõtte tabelTabel 16.

**Tabel 16. Volitustõendite turbeomaduste võrdlus**

| | Teadmuslikud | Esemelised | Biomeetrilised |
|---|---|---|---|
| **Ohud ja nõrkused** | Pealtkuulamine sideliinidel<br>Rünne sõnastikuga<br>Äraarvamine<br>Sihilik või kogemata edasiandmine | Vargus<br>Kaotamine<br>Sihilik edasiandmine<br>Imiteerimine<br>Võltsimine | Praktiliselt võltsimatu; mehhanismid eristavad elutu koopia tõelisest tõendist |
| **Subjekti tõendamine** | Ei tõenda rangelt subjekti | Ei tõenda rangelt subjekti | Põhimõtteliselt tugevaim subjekti identsuse tõend |
| **Vahetatavus** | Vahetatav regulaarselt või vajaduse korral | Vahetatav varguse või kaotamise korral; mehhanism seatav mitte aktsepteerima kaotatud või varastatud tõendit | Kui autentimismehhanismi õnnestub petta, ei ole tõend vahetatav |
| **Väär mitteaktsepteerimine** | Tõend muutumatu, seetõttu autentimisotsus determineeritud (jah/ei); FRR =0, kui kasutaja ei unusta parooli ja sisestab ta õigesti | Tõend muutumatu, seetõttu autentimisotsus determineeritud (jah/ei); FRR =0, kui kasutajal tõend käepärast | Tõend on stohhastiline, verifitseerimine on statistiline, seetõttu põhimõtteliselt FRR>0 |
| **Väär aktsepteerimine** | Õige halduse ja kasutamise korral FAR väga väike | Õige valiku ja halduse korral FAR väga väike | Statistilise verifitseerimise tõttu põhimõtteliselt FAR>0 |

### 8.5.2 Kasutuslikud omadused

Volitustõendite kasutusomadustest teeb võrdleva lühikokkuvõtte tabel Tabel 17.

**Tabel 17. Volitustõendite kasutuslike omaduste võrdlus**

| | Teadmuslikud | Esemelised | Biomeetrilised |
|---|---|---|---|
| **Omaksvõtt** | Laialdaselt omaks võetud | Laialdaselt omaks võetud (krediitkaardid jm) | Pole laialt kasutusel, on psühholoogilisi tõkkeid |
| **Kasutamise mugavus** | Tülikas meelde jätta, eriti kui vajalikke paroole või koode on mitu[1] | Nõuab hoolikat hoidmist ja kaasaskandmist. Vanuritel jt on raskusi kasutamisega | Takistada võivad füüsilised puuded või iseärasused |
| **Tõendi püsivus** | Mittemnemoonilised, eriti numbrilised kipuvad ununema | Kuluvad ja lagunevad (magnetriba kulumine, kaardi murdumine jne) | Mõni tõend võib muutuda vigastuste, haiguste või ea tõttu |
| | | | |

| | | | |
|---|---|---|---|
| **Ajakulu** | Verifitseerimine on lihtne, seetõttu kiire | Verifitseerimine on lihtne, seetõttu kiire | Verifitseerimine on keeruline, seetõttu mõnedes süsteemides aeglane |

[1] Üks 1995. a Inglismaal sooritatud uuring näitab, et iga kolmas PIN-koodi valdaja kirjutab koodi üles. Igal viiendal on PIN-koodi unustamise tõttu jäänud mõnikord automaadist raha saamata.

### 8.5.3 Majanduslikud näitajad

Volitustõendite majanduslikest näitajatest teeb võrdleva lühikokkuvõtte tabel Tabel 18.

**Tabel 18. Volitustõendite majanduslike näitajate võrdlus**

| | **Teadmuslikud** | **Esemelised** | **Biomeetrilised** |
|---|---|---|---|
| Alginvesteering | Võivad nõuda eriseadmeid (PIN-kood nõuab klaviatuuri) | Üldiselt nõuavad verifitseerimisseadet (riider, lukk), mis tuleb paigaldada ja süsteemiga integreerida | Vajalik on suhteliselt kallis verifitseerimisseade, mis tuleb paigaldada ja süsteemiga integreerida |
| Kasutajapõhised kulud | Iga kasutaja võib tuua kaasa kulutusi | Iga kasutajaga on seotud teatavad kulud (kaardi või isikliku võtme maksumus) | Kasutajaga seotud lisakulusid ei ole |

# 9 KRÜPTOGRAAFIA

Sõna krüptograafia (*cryptography*) pärineb kreeka keelest (*κρυπτος*– peidetud). Vahel kasutatakse samas tähenduses sõna krüptoloogia (ingl *cryptology*), mille lõpp -loogia on samuti kreeka päritoluga (*λογος*– sõna, ütlus, kõne). Otseses tõlkes tähendab krüptoloogia seega sõna (samuti keele, teadmise ja tähenduse) peitmise kunsti.

Kuni käesoleva sajandini oligi krüptograafia luureteenistuste ringides inimeselt inimesele edasiantav salakirjade koostamise ja murdmise oskus (vt N. Liventhal, "Krüptoloogia ja salaluure"). Nüüdisaegse krüptograafia eesmärgid on tunduvalt laienenud, eriti seoses arvutite laialdase kasutuselevõtuga, ja hõlmavad ka andmeterviklust. Sellegipoolest ei ole mingit vajadust teaduse nime muuta, sest (1) ka terviklust tagavad praktilised mehhanismid ei saa läbi traditsiooniliste salastusvahenditeta (paroolkaitse jms) ja (2) krüptoloogiat võib mõista kui peidetud tõe esiletoomise kunsti, sest λογοσ tähendab ka tõe väljaütlemist ja esiletoomist. Et informatsioon ise on teatavat laadi teadmus, sõltub infoturve alati sellest, kes mida ja mis ajahetkel teab või on võimeline teada saama. Õigesti tasakaalustatud teadmine ja salastus tagavad digitaalmaailmas korra ja ka võimaluse reaalse maailma tõdesid (näiteks sõnumi signeerimise fakti) päevavalgele tuua.

## 9.1 Ajalugu ja põhimõisted

### 9.1.1 Caesari šiffer

Inimestel on alati olnud saladusi. Kirja kasutuselevõtuga tekkis kohe vajadus ka salakirja järele, st sellise kirjaviisi järele, mis oleks arusaadav vaid kirja saajale ja lähetajale. On teada, et Julius Caesar kasutas salakirja, kus iga ladina tähestiku täht oli asendatud tähega, mille järjenumber oli kolme võrra suurem, ning kui sellist tähte polnud, jätkati lugemist tähestiku algusest. Näiteks asendati täht `A` tähega `D`, täht `K` tähega `N` jne. Krüpteerimist võib siin vaadelda kui alljärgneva asendustabeli kasutamist:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Näiteks sõna CAESAR oleks krüpteeritult FDHVDU. Tänapäevases matemaatilises esituses asendatakse Caesari šifri kirjeldamisel ladina tähestiku tähed arvudega (A=0,B=1,...,Z=25) ning eeldades, et *x* on sõnumiteksti täht (number), saame vastava krüpteeritud teksti ehk krüptogrammi tähe (numbri) valemiga

$$y = x \oplus 3,$$

kus $\oplus$ tähendab liitmist *modulo* 26. Näiteks $23 \oplus 3 = 0$ ja $17 \oplus 11 = 2$. Tänapäeval ei võta seesugust salakirja enam keegi tõsiselt, sest kooliõpilanegi (vt Ülo Kaasik, “Lihtsaid ja keerulisi”, Tallinn 1979, V osa “Täiesti salajane”) on võimeline krüptogrammi lugema, teadmata seejuures krüpteerimise viisi. Eriti veel siis, kui ta juhtub olema lugenud Edgar Allan Poe tuntud lugu "Kuldpõrnikas" (*The Gold Bug*).

### 9.1.2 Vernami šiffer

Salakirjatehnika arenes jõudsalt edasi XX sajandi alguse sõdade ajal. Aastal 1926 mõtles G.S. Vernam välja šifri, mis tänapäeval on tuntud kui ***ühekordne šifriplokk*** (ingl *one-time pad*). Hilisem analüüs näitas, et Vernami šiffer on teoreetiliselt turvaline; see tähendab, et tema murdmine on isegi teoreetiliselt võimatu. Vernami süsteem on lähedane Caesari poolt kasutatuga. Krüpteerimisel leitakse avateksti märgile *x* vastav krüptogrammi märk *y* valemiga

$$y = x \oplus z,$$

kus *z* on nn võtmemärk (Caesaril $z = 3$), mida vahetatakse pärast iga märgi krüpteerimist. Võtmemärkide järjendit nimetatakse salajaseks võtmeks. Loomulikult eeldatakse siin seda, et sõnumi saajal on olemas samasugune salajane võti, sest vastasel korral ei suudaks saaja sõnumit lugeda. Eeldatakse, et salajase võtme toimetamiseks saatjalt vastuvõtjale kasutatakse nn turvalist kanalit, mis võib seisneda näiteks salajase võtme isiklikus üleandmises saatjalt saajale. See üleandmine võib toimuda tükk aega enne tegelikku sideseanssi. Sajandi keskpaiga spioonid kandsidki kaasas imepisikesi tabeleid ühekordse šifriploki kasutamiseks vajalike salajaste võtmetega.

### 9.1.3 Shannoni teooria

Ehkki Vernam ja ka paljud tema šifri kasutajad olid veendunud selle murdmatuses, ei suutnud tema ega keegi teine seda formaalselt tõestada, sest polnud olemas vajalikku matemaatilist aparatuuri.

Sobiva aparatuuri lõi Shannon, kes 1949. a avaldas töö, milles defineeritakse täieliku salastuse formaalne nõue ja tõestatakse, et Vernami ühekordne šifriplokk tagab täieliku salastuse. Selle artikli ilmumist võib lugeda tänapäeva informatsiooniteooria ja samuti krüptoloogia kui teaduse sünniks.

Shannon defineeris krüptosüsteemi kui salajasest võtmest $Z$ sõltuva teisenduste paari $Y=E_Z(X)$ (krüpteerimine) ja $X=D_Z(Y)$ (dešifreerimine). Avateksti $X$ vaadeldakse kui juhuslikku suurust, mille erinevatel võimalikel väärtustel $X_1, X_2, \ldots$ on teatavad esinemise tõenäosused $p(X_1), p(X_2), \ldots$ Täielik salastus defineeritakse kui olukord, kus krüptogramm $Y$ on avatekstist statistiliselt sõltumatu, st krüptogrammi $Y$ tegeliku väärtuse $y$ teadasaamine ei võimalda mingil määral täpsustada vastava avateksti $X$ tegelikku väärtust $x$, kui ei teata võtme $Z$ tegelikku väärtust $z$. Tõenäosuste kaudu väljendatuna

$$p(X_k) = p(X_k| \ Y=y)$$

mistahes $k$ korral. Võib ka öelda, et suuruse $X$ Shannoni entroopia ründaja jaoks ei muutu, kui ta saab teada midagi krüptogrammi $Y$ kohta.

Näiteks kui kasutatakse Vernami süsteemi ja edastatav sõnum on ühebitine, st koosneb ühestainsast märgist $x \in \{0,1\}$, siis saanud teada krüptogrammi $Y$ tegeliku väärtuse $y$, ei tea ründaja midagi enamat $X$ jaotuse kohta, kui ta ei tea midagi võtme $Z$ jaotusest. Tõepoolest, kui näiteks ründaja teab, et $y=1$, siis sõltuvalt sellest, kas $z$ on 0 või 1, on $x$ vastavalt kas 1 või 0.

Shannon jõudis oma töös ülitähtsa tulemuseni, mille kohaselt peab täieliku salastuse tagamiseks salajane võti $Z$ sisaldama vähemalt niisama palju märke (st olema vähemalt niisama pikk) kui avatekst $X$. Täieliku salastuse tagamine läheb seega väga kalliks ning teda saab kasutada üksnes väga head salastust nõudvates kanalites, milles liikuv infohulk on piiratud.

### 9.1.4 Kerckhoffi eeldus

Et krüptograafia kasutamine peab ise tagama informatsiooni salastuse, ei saa krüptograafia kui meetod toetuda muudele salastusvahenditele. Seetõttu on üks krüptograafia põhieeldusi krüptogrammi Y kättesaadavus kõigile. Turvalisust ei pea tagama mitte krüptogrammi varjamine, vaid avatekstist krüptogrammini viivad matemaatilised teisendused.

Hollandlane A. Kerckhoff (1835–1903) tegi esimesena veelgi kaugemale ulatuva eelduse, et ka krüpteerimise tehnoloogia (algoritmid) on avalik ning kogu salastuse peab tagama võtme Z salastus. See eeldus võimaldab avalikult konstrueerida turvalisi šifreid, kasutades näiteks ülikoolis töötavate matemaatikute abi, kartmata, et kõrvalised isikud võiksid algoritmi kirjeldust enda valdusse saades olla võimelised krüptogramme murdma. Kerckhoffi mõtte laialdane omaksvõtt tagas seega võimaluse krüptograafial teaduseks areneda ja veel enam, ka krüptoalgoritmide standardimiseni jõuda.

### 9.1.5 Enigma

Teise maailmasõja (1939–1945) tingimustes polnud alati võimalik kasutada ühekordset šifriplokki. Salastatud side otstarbeks tuli leida kiiremaid ja odavamaid meetodeid. Ühekordse šifriplokiga krüpteerimise juures tuleb teatavasti kasutada iga sõnumimärgi kohta üht turvalise kanali kaudu edastatud võtmemärki, mistõttu krüpteerimine on liiga kulukas. Seepärast konstrueerisid sakslased erilise krüpteerimismasina ENIGMA, kus salastus tagati fikseeritud pikkusega võtme abil. Selle masinaga edastasid sakslased salajasi sõnumeid ja korraldusi allveelaevadele. Inglastel õnnestus hiljem selle masina üks eksemplar enda valdusse saada. Sõja lõpupoole suutsid nad juba kõiki sakslaste salajasi teateid lugeda, sest Allan Turingi juhtimisel konstrueeriti esimene elektronarvuti Colossus, mis oli spetsiaalselt mõeldud ENIGMA sõnumite dekrüpteerimiseks. Seega ei tulnud mitte krüptograafia arvutitesse, vaid arvuteid hakati kasutama krüptograafia murdmiseks.

### 9.1.6 DES

Aastal 1972 kuulutas USA rahvuslik standardibüroo välja krüptograafiaalgoritmide konkursi, et riigiasutustes tekiks ühtne andmeturbemetoodika. Oli lõplikult aru saadud, et krüptograafias ei saa kasutada salajasi ega n-ö omatehtud algoritme. Kõik algoritmid peavad olema avalikud, spetsialistide koostatud ja avaldatud, et nende tugevust saaksid hinnata krüptograafiaspetsialistid. Konkursi võitis IBM, kes oli esitanud H.Feisteli ja W.Tuchmanni projekteeritud algoritmi LUCIFER, mida hiljem täiustades avaldati 1975. a algoritm DES (*Data Encryption Standard,* "andmete krüpteerimise standard"), mis võeti ametliku standardina vastu 1977. aastal.

### 9.1.7 Diffie-Hellmani võtmevahetus

Salajase võtmega krüptosüsteemide põhiline probleem on vajadus turvalise kanali järele. Salajane side on aga võimalik ka turvalist kanalit kasutamata. Aastal 1976 esitasid W. Diffie ja M. Hellman nn ***avaliku võtmega krüptosüsteemi*** idee, milles esmakordselt kadus otsene vajadus turvalise kanali järele. Diffie ja Hellmani saavutused on eelkõige seotud algoritmide keerukuse teooria edusammudega sajandi teisel kolmandikul. Sai selgeks, et krüptosüsteemidelt ei olegi mõtet nõuda täielikku salastust. Piisab, kui nõuame niiöelda praktilist salastust, mis eeldab arvutusvõimsuse piiratust.

W.Diffie ja M.Hellman kasutasid nn ***ühesuunalisi funktsioone*** (*one-way functions*), mis on ise kergesti arvutatavad, kuid mille pöördfunktsiooni leidmiseks ei piisa ka kogu maailma arvutusvõimsusest. Tänapäeval usutakse, et üks sellise funktsiooni näiteid on nn diskreetne eksponentfunktsioon

$$f(x) = \alpha^x \bmod p,$$

kus $p$ on suur algarv ning $\alpha$ on positiivne täisarv, mis on valitud nii, et astmed $\alpha^1, \ldots, \alpha^{p-1}$ on paarikaupa erinevad. Kui arv $x$ on teada, on arvu $y=f(x)$ suhteliselt kerge leida, kuid suuruse $y$ abil vastava originaali $x$ leidmine on ülikeerukas kombinatoorikaülesanne, mille lahendamine on igasuguste mõistlike arvutiressursside abil praktiliselt võimatu.

Diffie ja Hellmani süsteemi igal kasutajal $i$ on oma salajane võti $x(i)$ ja nn. avalik võti $y(i)=f(x(i))$. Kui näiteks kasutaja $i$ soovib saata salastatud sõnumit kasutajale $j$, krüpteerib ta saadetava sõnumi ükskõik millise eelnevalt kokkulepitud traditsioonilise krüptoalgoritmiga, kasutades salajase võtmena arvu

$$Z(i,j) = y(j)^{x(i)} \bmod p.$$

Lihtne on kontrollida, et alati kehtib $Z(j,i) = Z(i,j)$ ja seetõttu saab kasutaja $j$ dešifreerida talle saadetud sõnumi võtmega, mis on koostatud samadel põhimõtetel.

### 9.1.8 Vahendusrünne

Avalike võtmete vahetamisel tuleb alati kontrollida partnerilt saadud võtme autentsust, st tuleb veenduda, et saadud avalik võti pärineb tõepoolest eeldatavalt adressaadilt. Avalike võtmete vahetuseks võib kasutada näiteks tavalist telefonsidet, eeldades et partneri häält on raske tõetruult imiteerida.

Kui autentsusele tähelepanu ei pöörata, on võimalik nn vahendusrünne (*man in the middle attack*). Kui kasutaja *A* soovib saata salastatud sõnumit kasutajale *B*, võib ründaja *C* vahetada kasutaja *B* avaliku võtme *y(B)* iseenda avaliku võtmega *y(C)* ja dešifreerida saadetud sõnumi.

### 9.1.9 Salaluugiga funktsioonid ja digitaalsignatuurid

Olgu *f* funktsioon, mis on kergesti pööratav ainult siis, kui on teada mingi selle funktsiooniga seotud suurus *z*; st kui *z* on teada, on kerge leida etteantud kujutisele *y* vastavat originaali *x*, nii et $y=f(x)$. Seega neile, kes vastavat *z* ei tea, on funktsioon *f* ühesuunaline. Sellist ühesuunalist funktsiooni nimetatakse salaluugiga funktsiooniks (*trapdoor one-way function*). Salaluugiga funktsiooni mõiste esitasid esmakordselt Diffie ja Hellmann 1976. aastal, kuid sel ajal polnud teada ühtki seesuguse funktsiooni kandidaati.

Salaluugiga funktsioonid võimaldavad luua digitaalsignatuure: kui A on ainus, kes teab saladust (salaluuki) z, võib sõnumile *x* suuruse $\sigma=f^{-1}(x)$ lisamist võrrelda allkirja andmisega. Igaüks on suuteline kontrollima signatuuri $\sigma$ ehtsust, kontrollides võrdust

$$x=f(\sigma).$$

Digitaalsignatuuril on oluline roll elektroonilise dokumendihalduse võimaldamises. Salaluugiga funktsioonide ja seega ka digitaalsignatuuride olemasolu ennustamine Diffie ja Hellmanni poolt oli 70te aastate üks olulisemaid teaduslikke tulemusi.

### 9.1.10 RSA

Aastal 1978 pakkusid R.Rivest, A.Shamir ja L.Adleman salaluugiga funktsiooni kandidaati. Selle funktsiooni kirjeldamiseks tuleb veidi rakendada elementaarset arvuteooriat. Kasutaja A valib kõigepealt kaks suurt algarvu *p* ja *q*. Seejärel leiab ta positiivsed arvud *e* ja *d* nii et: (1) SÜT$(e,(p-1)\cdot(q-1))=1$ ja (2) $e\cdot d \mod (p-1)(q-1)=1$. Salaluuk *z* on siin järjend $(p,q,e)$ ja funktsioon *f* defineeritakse järgmiselt:

$$f(x) = x^e \bmod n,$$

kus $n = p\cdot q$. Selgub, et tänu arvuteooriast tuntud Euleri teoreemile on sel juhul alati arvutatav pöördfunktsioon:

$$f^{-1}(y) = y^d \bmod n.$$

Rivest-Shamir-Adlemani süsteemis on igal kasutajal *i* seega oma salajane funktsioon $D_i(y) = f^{-1}(y)$ ja avalik funktsioon $E_i(x)=f(x)$, mis on teineteise pöördfunktsioonid. Teades vaid funktsiooni $E_i$ on (teadmata vastavat salaluuki) peaaegu võimatu leida pöördfunktsiooni $D_i$. Sõnumi *X* signeerimiseks teisendab kasutaja *i* sõnumit oma salajase algoritmiga, saades signatuuri $\sigma = D_i(X)$. Kui aga kasutaja *i* soovib, et sõnumit lugeda saaks ainult kasutaja *j*, krüpteerib ta sõnumi *X* kasutaja *j* avaliku algoritmiga $E_j$ ja saab krüptogrammi $\kappa=E_j(X)$.

## 9.2 Plokkšifrid

Plokkšifrid on nüüdisaja krüptosüsteemide tähtsaimaid komponente, mille põhiotstarve on andmete konfidentsiaalsuse tagamine nende edastusel ja säilitusel. Plokkšifreid kasutatakse sageli ehituskividena pseudojuhuslike arvude generaatorites, jadašifrites, sõnumiautentimise koodides (*message authentication code, MAC*) ja räsifunktsioonides. Pole olemas ideaalset plokkšifrit kõigi kasutusalade jaoks. Praktilistes rakendustes tuleb näiteks sageli teha kompromiss kiiruse ja turvalisuse vahel. Seetõttu on olemas ka palju erinevaid plokkšifreid.

### 9.2.1 Plokkšifri määratlus

Plokkšiffer on *k*-bitisest parameetrist *K* (võtmest) sõltuv funktsioon $E_K(\cdot)$, mis kujutab *n*-bitise avateksti *X* krüptogrammiks *Y*, mis samuti on *n*-bitine. Eeldatakse, et võti *K* on valitud mingist piiritletud võtmete hulgast ***K***, nn võtmeruumist. Et krüpteeritud andmeid saaks hiljem dešifreerida, peab krüpteerimisfunktsioon olema üksühene, st kui $E_K(X)=E_K(X')$, siis $X=X'$. Arvu *n* nimetatakse ploki pikkuseks ja arvu *k* võtme pikkuseks. Eeldatakse, et võti *K* on valitud võtmeruumist ***K*** juhuslikult (ühtlase jaotusfunktsiooniga). Suurust $\log_2|K| \le k$ nimetatakse efektiivseks võtmepikkuseks. Eeldades, et ründaja ei tea võtme *K* kohta mingit lisainformatsiooni, on efektiivne võtmepikkus võrdne võtme *K* kui juhusliku suuruse entroopiaga ründaja suhtes.

Universaalne *n*-bitine plokkšiffer realiseerib mistahes üksühest teisendust (substitutsiooni) kõikvõimalike *n*-bitiste järjendite hulgal $V_n$. Seega peaks universaalse plokkšifri võtmepikkus olema

$$k \ge \log_2( 2^n! ) \approx (n-1.44)\cdot 2^n$$

Loomulikult on nii pika võtme kasutamine ebapraktiline. Sellegipoolest on üks plokkšifrite projekteerimise põhinõue, et juhuslikult valitud võtmele peab vastama juhuslikult valitud substitutsioon.

Kui plokipikkus *n* on liiga väike, võib algoritm olla rünnatav statistilise analüüsi meetoditega. Statistilise ründe tõrjumiseks võib varieerida krüpteerimise režiimi. Plokipikkuse liigne suurendamine raskendab algoritmi realiseerimist ja suure tõenäosusega vähendab krüpteerimise kiirust.

### 9.2.2 Tüüpründed ja nende keerukus

Krüptoalgoritmide projekteerimisel arvestatakse alati ründaja võimalikke tegevusi, mille eesmärk on leida teatud fikseeritud avatekst *X* või krüpteerimisvõti *K*. Öeldakse, et krüptosüsteem on täielikult murtud (*totally broken*), kui krüpteermisvõti on mingi ründe abil leitav, ja osaliselt murtud (*partially broken*) kui ründajal õnnestub teada saada osa avatekstist, kuid võtit mitte.

Tüüpründeid võib jagada nelja rühma, sõltuvalt sellest, kui palju lähtematerjali on ründaja käsutuses.

- ***Teadaoleva krüptogrammiga rünne*.** Ründajale on teada ainult üks või mitu krüptogrammi $Y_1,Y_2,\ldots,Y_m$. Kui võtme pikkus on väike, on piisavalt suure *m* korral võimalik statistiline analüüs.
- ***Teadaoleva avatekstiga rünne.*** Ründajale on teada üks või mitu avatekst-krüptogramm-paari $(X_1,Y_1),(X_2,Y_2),\ldots,(X_m,Y_m)$. Sedalaadi võimalused avanevad näiteks siis, kui krüpteeritud informatsioon on loomuliku keele tekst (näiteks kiri), mis alati sisaldab kokkuleppelisi tüüpfraase. Näiteks aitas inglasi teise maailmasõja ajal sakslaste krüpteeritud sõnumite lugemisel suuresti asjaolu, et peaaegu kõik need lõppesid sõnadega "*Heil Hitler!*".
- ***Valitava avatekstiga rünne.*** Ründajal on võimalik valida teatud piiratud hulk avatekste $X_1,X_2,\ldots,X_m$ ja saada teada neile vastavad krüptogrammid. Selline võimalus võib avaneda näiteks juhul, kui poearve saadetakse panka krüpteeritud kujul. Ostes sobivaid kaupu ja kuulates samal ajal pealt sideliini võibki teada saada hulgaliselt avatekst-krüptogramm-paare.

- ***Adaptiivne valitava avatekstiga rünne.*** Ründajal on võimalik valida järgmisi avatekste sõltuvalt eelmistele avatekstidele vastavatest krüptogrammidest.

Valitava avateksti printsiip nõuab selliste krüptoalgoritmide (sh plokkšifrite) kasutamist, mis on turvalised valitava avatekstiga rünnete suhtes, vaatamata sellele, et reaalsetes rakendustes võivad seesugused ründed olla ebapraktilised.

Mis tahes krüptoalgoritmi parim tugevuse mõõt on efektiivseima teadaoleva ründe keerukus, mida võib mõõta

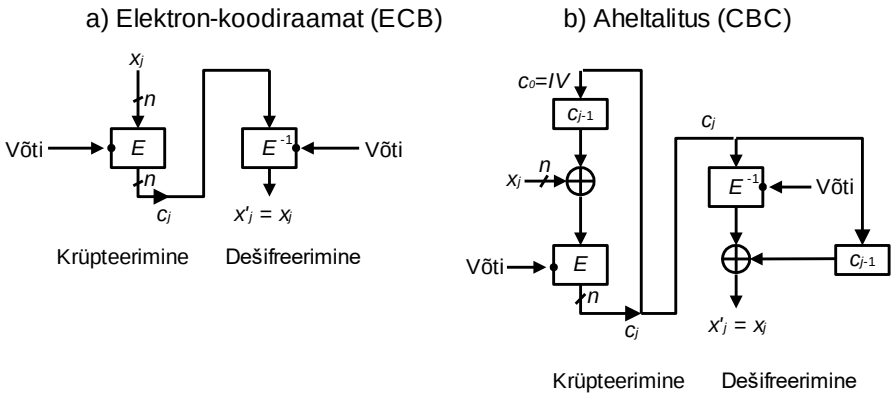
- vajaliku lähteandmete hulgaga,
- vajaliku salvestusruumiga,
- vajaliku tööajaga.

On selge, et kui lubatud lähteandmete hulk on suurusjärgus $2^n$ ja $n$ on ploki pikkus, on rünne alati võimalik. Juhul kui lubatav tööaeg on suurusjärgus $2^k$ ja $k$ on võtme pikkus, on rünne alati võimalik võtmeruumi täieliku läbivaatuse teel.

Kui plokipikkus on liiga väike, on võimalikud nii ***sõnastikrünne*** (*dictionary attack*) kui ka ***krüptogrammisobitusrünne*** (*matching ciphertext attack*). Sõnastiku moodustamine on võimalik juhul, kui on teada piisaval hulgal avatekst-krüptogramm-paare mingi kindla võtme korral. Mida suurem on sõnastik, seda suurem on tõenäosus, et seda saab kasutada juhuslikult valitud krüptogrammi dešifreerimiseks. Täielik sõnastik koosneb $2^n$ paarist. Kui on teada $2^{n/2}$ paari ja $2^{n/2}$ juhuslikku krüptogrammi, siis on tulenevalt sünnipäevaparadoksist (vt "Räsifunktsioonid") väga tõenäoline, et vähemalt ühele krüptogrammile vastav paar on sõnastikus olemas ja seega saab ühe krüptogrammi dekrüpteerida. Isegi kui sõnastik üldse puudub ja on teada $2^{n/2}$ krüptogrammi, on kahe krüptogrammi kokkulangevuse tõenäosus piisavalt suur ja seega on võimalik saada osalist informatsiooni krüpteeritud andmete kohta.

### 9.2.3 Plokkšifri tööviisid

Plokkšiffer krüpteerib avateksti $n$-bitiste plokkide kaupa krüptogrammi vastavateks $n$-bitisteks plokkideks. Mistahes ploki krüpteerimise tulemus ei sõltu ülejäänud plokkidest ja seega oleks põhimõtteliselt võimalik kõiki plokke krüpteerida korraga. Seesugusel krüpteerimisviisil, mida nimetatakse koodiraamatuks (ECB, e*lectronic codebook*), on palju puudusi (vt 9.2.2), mistõttu teda kasutatakse harva pikemate sõnumite krüpteerimisel. Teised levinumad tööviisid on ahel (CBC, *ciprer-block chaining*), tagasisidestus krüptogrammist (CFB, *cipher feedback*) ja tagasisidestus väljundist (OFB, *output feedback*).




**Joonis 48. Plokkšifrite tööviisid**

### *9.2.3.1 ECB*

SISEND: $k$-bitine võti $K$; $n$-bitised avateksti plokid $X_1,\ldots,X_t$.
VÄLJUND: krüptogrammi plokid $Y_1,\ldots,Y_t$.
KRÜPTEERIMINE: $Y_i = E_K(X_i)$
DEKRÜPTEERIMINE: $X_i = E_K^{-1}(Y_i)$

Koodiraamat-tööviisi põhiomadused on järgmised.

- Identsetele avatekstidele vastavad identsed krüptogrammid.
- Plokid on krüpteeritud sõltumatult, mistõttu krüptogrammi plokkide ümberjärjestamisel saadud uuele krüptogrammile vastab esialgne avatekst samasugusel viisil ümberjärjestatud avatekstiplokkidega.

- Üks või mitu juhuslikku arvutus- või edastusviga ühes plokis ei põhjusta vigu teistes plokkides.

Eri plokkide krüpteerimise sõltumatuse tõttu on võimalik plokkide järjekorra muutmine, mis võib olla võimalike rünnete objekt. See krüpteerimisviis on ohtlik ka seetõttu, et võimaldab sarnaste krüptogrammide põhjal teha järeldusi vastavate avatekstide sarnasuse kohta. Näiteks on võimalik krüpteeritud piltide ja videoinfo järgi otsustada, kas pildil on inimese nägu, mingi hoone vms. ECB turvalisust võib tõsta, kui lisada igasse plokki mõned juhuslikud täidisbitid.

#### *9.2.3.2 CBC*

| | |
|---|---|
| SISEND: | $k$-bitine võti $K$; n-bitine $IV$; $n$-bitised plokid $X_1,\ldots,X_t$. |
| VÄLJUND: | $n$-bitised plokid $Y_1,\ldots,Y_t$. |
| KRÜPTEERIMINE: | $Y_0 = IV$; kui $i>0$, siis $Y_i = E_K(X_i \oplus Y_{i-1})$ |
| DEKRÜPTEERIMINE: | $Y_0 = IV$; kui $i>0$, siis $X_i = Y_{i-1} \oplus E_K^{-1}(Y_i)$ |

Aheltööviisi põhiomadused on järgmised.

- Kui kasutakse $IV$ sama väärtust, siis identsetele avatekstidele vastavad identsed krüptogrammid. Kui muuta $IV$ väärtust või esimest avateksti, on tulemuseks erinevad krüptogrammid.
- Krüpteerimismehhanism tagab, et iga krüptogrammi plokk $Y_i$ sõltub kõigist eelnevatest avateksti plokkidest $X_1,\ldots,X_i$. Seetõttu ei ole võimalik krüptogrammi plokkide järjestust muuta, ilma et dekrüpteerimisel häviks suure osa plokkide sisu (st muutuks müraks).
- Bitiviga krüptogrammi plokis $Y_i$ mõjutab plokkide $Y_i$ ja $Y_{i+1}$ dekrüpteerimist. Vigase ploki $Y_i$ dekrüpteerimisel saadud tulem $X'_i$ on tavaliselt müra, kuid plokis $X'_{i+1}$ on sarnane bitiviga. Seega on ründajal võimalik (küll $i$-nda ploki hävingu hinnaga) põhjustada mistahes muutusi plokis $i+1$. Ta ei pruugi küll teada, milline oli algse ja muudetud biti väärtus, kuid näiteks juhul, kui on teada, et mingid kindlad bitid esitavad maksesummat, on üsna tõenäoline, et selle kõikide bittide juhuslikud muudatused põhjustavad tõenäoliselt summa märgatava suurenemise, eeldades, et enamik summasid ei küüni maksimaalse võimalikuni.
- Kui plokis $i$ on bitiviga ja plokk $i+1$ on õige, dekrüpteeritakse plokk $i+2$ veatult.

Ehkki $IV$ ei nõua salastust, peab tema edastus olema autentne, sest modifitseerides sobivalt $IV$ väärtust on ründajal võimalik ilma ühtegi plokki hävitamata (müraks muutmata) modifitseerida esimest plokki.

#### *9.2.3.3 CFB*

Paljud krüptograafia rakendused nõuavad, et krüpteeritud informatsiooni edastataks mingi fikseeritud pikkusega ($r$-bitiste, $r \leq n$) üksuste haaval ja viivituseta. Senivaadeldud krüpteerimisviisid (ECB ja CBC) seda ei võimalda, sest iga $n$-bitise ploki edastuseks kuluv aeg (st ka viivitus) on suurem krüpteerimiseks ja dekrüpteerimiseks kuluva aja summast. Nimetatud nõuete täitmiseks ongi mõeldud CFB ehk tööviis tagasisidega krüptogrammist.

| | |
|---|---|
| SISEND: | $k$-bitine võti $K$; n-bitine $IV$; $r$-bitised plokid $X_1,\ldots,X_u$. |
| VÄLJUND: | $r$-bitised plokid $Y_1,\ldots,Y_u$. |
| KRÜPTEERIMINE: | $1_1 = IV$.<br>$O_i = E_K(I_i)$.<br>$t_i = r$ kõrgeimat bitti $O_i$-st.<br>$Y_i = X_i \oplus t_i$.<br>$I_{i+1} = 2^r \cdot I_i + c_i \bmod 2^n$. |
| DEKRÜPTEERIMINE: | $I_1 = IV$; $X_i = Y_i \oplus t_i$. Suurused $O_i$ ja $t_i$ leitakse analoogiliselt krüpteerimisele. |

CFB põhiomadused on järgmised.

- *IV* muutmisel muutuvad ka kõik krüptogrammi plokid. Seega ei pea *IV* olema salajane. Mõned rakendused nõuavad küll, et järgmisena kasutatav *IV* poleks etteennustatav.
- Krüptogrammi plokkide ümberjärjestamine pole võimalik. Krüptogrammi ploki õigeks dekrüpteerimiseks peavad eelnevad $\lceil n / k \rceil$ plokki olema korrektselt dekrüpteeritud.
- Üks või mitu bitiviga mingis krüptogrammi plokis põhjustavad muutusi mitte enamas kui järgmises $\lceil n / k \rceil$ plokis. Kui plokis $Y_i$ on juhuslik (või ka tahtlikult tekitatud) bitiviga, on kättesaadud avatekstis $X'_i$ samasugune bitiviga. Seetõttu saab ründaja tahtlikult muuta mistahes bitti avatekstis.
- Väheneb läbilaskevõime (*throughput*), st korraga krüpteeritakse *r* bitti.

Kuna krüpteerimisviisis CFB kasutatakse ainult algoritmi $E_K$ krüpteerimisfunktsiooni (mitte aga dekrüpteerimist), ei tohi CFB-d kasutada juhul, kui *E* on avaliku võtmega algoritm.

#### *9.2.3.4 OFB*

Tagasisidestus väljundist (OFB, *output feedback mode*) on vajalik siis, kui soovitakse ära hoida igasugust bitivigade levikut. Tagasisidena ei kasutata siin mitte krüptogrammi plokke, vaid väljundit $O_i$.

| | |
|---|---|
| SISEND: | *k*-bitine võti *K*; n-bitine *IV*; *r*-bitised plokid $X_1,\ldots,X_u$. |
| VÄLJUND: | *r*-bitised plokid $Y_1,\ldots,Y_u$. |
| KRÜPTEERIMINE: | $I_1 = IV$.<br>$O_i = E_K(I_i)$.<br>$t_i$ = *r* kõrgeimat bitti $O_i$-st.<br>$Y_i = X_i \oplus t_i$.<br>$I_{i+1} = O_i$. |
| DEKRÜPTEERIMINE: | $I_1 = IV$; $X_i = Y_i \oplus t_i$. Suurused $O_i$ ja $t_i$ leitakse analoogiliselt krüpteerimisele. |

OFB põhiomadused on järgmised.

- *IV* muutmisel muutub ka krüptogramm, isegi juhul kui avatekst ei muutu.
- Võtmete $t_i$ jada on avatekstist sõltumatu.
- Üks või mitu bitiviga ühes krüptogrammi plokis ei mõjuta ülejäänud plokkide dekrüpteerimise tulemust.
- Kui mõni plokk (sidekanalis) kaotsi läheb, tuleb krüpteerimine uuesti sünkroniseerida.
- Läbilaskevõime on samasugune nagu tööviisilil CFB.

OFB sarnane, kuid lihtsam krüpteerimisviis on nn loendurtöö (*counter mode*), kus uus sisendväärtus saadakse lihtsalt loenduri abil, st $I_{i+1} = I_i + 1$. Loenduri kasutamine hoiab ära nn silmuseprobleemi (*short-cycle problem*), mis võib tekkida OFB puhul, sest sünnipäevaparadoksi tõttu hakkab iteratsioon $O_{i+1} = f(O_i)$ korduma keskmiselt $2^{n/2}$ sammu juures.

### 9.2.4 Ammendav võtmeotsing ja mitmekordne krüpteerimine

Fikseeritud võtmepikkus seab piirangu krüptoalgoritmi tugevusele, sest piisavate vahendite ja ressursside olemasolul on võimalik ammendav võtmeotsing, mille kogukeerukus on keskmiselt $2^{k-1}$ krüpteerimisoperatsiooni, kui *k* on võtmepikkus. Üks võtmeotsingut soodustav asjaolu seisneb selles, et krüptoalgoritmid projekteeritakse tavaliselt kiiretena, mis ühelt poolt suurendab krüpteerimiskiirust, teiselt aga hõlbustab ka ründaja tegevust. Üks võimalik viis ammendava otsingu vastu võitlemiseks on selliste krüptoalgoritmide projekteerimine, kus võtme vahetus (kuid mitte krüpteerimine) nõuab mahukaid arvutusi.

Võtme otsing on võimalik vahel ka siis, kui on teada ainult krüptogrammid (st mitte avatekst-krüptogramm-paarid). Oletame, et krüpteeritakse 64-bitiseid plokke, mis koosnevad 8-st ASCII märgist, kusjuures iga 8. bitt on paarsusbitt (veatuvastuse otstarbeks). Kui krüptogrammi $Y$ dekrüpteeritakse vale võtmega, on kõigi 8 biti klappimise tõenäosus $2^{-8}$. Kui ründajale on teada $t$ krüptogrammi $Y_1, \ldots, Y_t$, on tõenäosus, et vale võtmega dekrüpteerimisel kõik paarsusbitid klapivad, $2^{-8t}$. Praktikas piisab, kui $t=10$.

Kaskaadšifriks (*cascade cipher*) nimetatakse erinevate (või erinevate võtmetega) šifrite (nimetatakse kaskaadšifri järkudeks) järjestikku rakendamist. Lihtsamal juhul on kõikide šifrite plokipikkus $n$ ja võtmepikkus $k$. Kui kõikides järkudes kasutatakse üht ja sama šifrit, nimetatakse kaskaadšifri rakendamist mitmekordseks krüpteerimiseks.

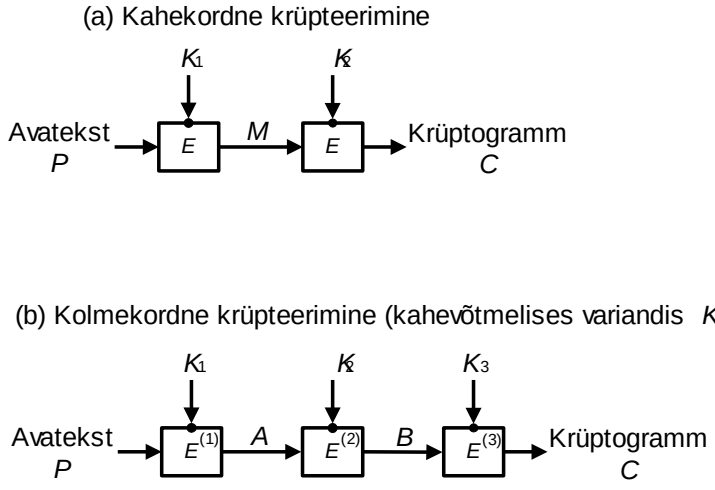
Näiteks kahekordsel krüpteerimisel kasutatakse kaht erinevat võtit *K1* ja *K2*. Avatekstile $X$ vastav krüptogramm arvutatakse valemiga $Y = E_{K2}(E_{K1}(X))$. Kolmekordseks krüpteerimiseks kasutatakse kolme sõltumatut võtit *K1*, *K2*, *K3* ning krüptogramm $Y$ saadakse valemiga

$$Y = E_{K3}^{(3)}(E_{K2}^{(2)}(E_{K1}^{(1)}(X))),$$

kus $E_{Ki}^{(i)}$ tähendab kas krüpteerimist või dekrüpteerimist võtmega *Ki*. Mitmekordne krüpteerimine on efektiivne vaid siis, kui teisenduste hulk

$$G = \{ E_K(\cdot), E_K^{-1}(\cdot) \mid K \in \{0,1\}^k \}$$

pole rühm, st kui iga kahekordne krüpteerimine pole ekvivalentne sobivalt valitud ühekordse krüpteerimisega. Mõnede šifrite kohta (näiteks DES) on tõestatud, et $G$ pole rühm.




**Joonis 49. Mitmekordne krüpteerimine**

### 9.2.5 Järgusobitusrünne

Naiivne ammendaval võtmeotsingul põhinev rünne kahekordsele krüpteerimisele nõuab $2^{2k}$ krüpteerimist. Järgusobitusrünne (*meet-in-the-middle attack*) võimaldab oluliselt rünnet lihtsustada, kasutades ainult $2^k$ krüpteerimist ja vastaval määral salvestusruumi.

Kui ründajale on teada avatekst-krüptogramm paar $(X,Y)$, kus $Y= E_{K2}(E_{K1}(X))$, siis võib ta toimida järgmiselt.

- Ründaja arvutab avateksti $X$ kõikvõimalikud vahekujutised $M_i=E_i(X)$ võtme $K1$ kõikvõimalike väärtuste korral $K1 = i = 0, \ldots, 2^k-1$.

- Ründaja arvutab krüptogrammi $Y$ kõikvõimalikud vaheoriginaalid $N_i = E_i^{-1}(Y)$ võtme $K2$ kõikvõimalike $2^k$ väärtuse korral.

- Ründaja sobitab väärtused $M_i$ väärtustega $N_i$. Kui $M_i=N_i$, siis $i$ on õige võtme kandidaat.

Kandidaate võib olla palju. Õige võtme teadasaamiseks on vaja rohkem avatekst-krüptogramm-paare. Õigete programmeerimisvõtetega saab $2^k$-elemendilisest massiivist $\{N_0,N_1,\ldots\}$ otsida ajaga, mis on lineaarne $k$ suhtes. Kogu sobitusprotseduur võtab seega $O(k^2)$ sammu, st kuluv aeg on proportsionaalne võtmepikkuse ruuduga.

**Näide**. Näiteks algoritmis DES on $n=64$ ja $k=56$. Tõenäosus, et fikseeritud $M_i$ on võrdne mõne $N_j$-ga on võrdne $2^{56}/2^{64}= 2^{-8}$. Seega on oodatavate kandidaatide arv $2^{56}\cdot 2^{-8} = 2^{48}$. Tõenäosus, et väär kandidaat (st võti, mis on kandidaatide hulgas, kuid pole õige võti) sobib ka teise juhuslikult valitud avatekst-krüptogramm-paari korral, on seega

$$2^{48}/2^{64} = 2^{-16},$$

mistõttu juba kahe paari teadaolemise korral on õige võtme tuvastus selle ründega peaaegu kindel. Naiivne võtmeotsing 2-DES-i jaoks nõuaks $2^{112}$ krüpteerimist. Järgusobitusrünne aga võimaldab läbi ajada $2^{56}$ krüpteerimise ja $2^{56}$ suurusjärku salvestusruumiga.

Kui DES-i kasutatakse kolmekordselt, kuid kahe eri võtmega, st kui

$$Y = E_{K1}(E_{K2}^{-1}(E_{K1}(X))),$$

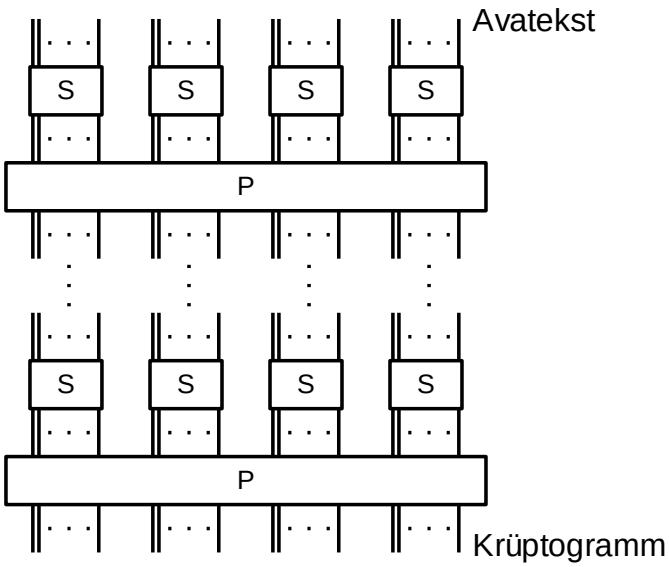
ja ründajal on kasutada $t$ avatekst-krüptogramm-paari, on võimalik sobitusrünne, mis nõuab $O(t)$ salvestusruumi ja $2^{120-\log t}$ krüpteerimist. Näiteks, kui ründaja käsutuses olevate paaride arv on $2^{40}$, tuleb sooritada $2^{80}$ krüpteerimist.

### 9.2.6 Korrutisšifrid ja Feisteli šifrid

Korrutisšiffer tekib kahe või enama teisenduse kombineerimisel, nii et tulemuseks on komponentidest turvalisem šiffer. Substitutsioon-permutatsioonvõrguks (joonis 58 ) nimetatakse korrutisšifrit, mille järgud on permutatsioonid ja substitutsioonid. Substitutsiooniks nimetatakse suvalist üksühest (pööratavat) teisendust hulgast $\{0,1\}^m$ iseendasse, kus $m$ on väiksem või võrdne plokipikkusega $n$. Permutatsiooniks $\pi$ nimetatakse erikujulist üksühest teisendust hulgast $\{0,1\}^n$ iseendasse, mille arvutamisel kasutatakse $n$-elemendilise hulga $\{0,\ldots,n-1\}$ mingit substitutsiooni $\sigma$ järgmiselt:

$$\pi(b_0,\ldots,b_{n-1}) = (b_{\sigma(1)},\ldots,b_{\sigma(n-1)}),$$

kus $(b_0,\ldots,b_{n-1})\in\{0,1\}^n$ on mistahes bitistring. Iteratiivseks plokkšifriks nimetatakse sellist plokkšifrit, kus rakendatakse järjestikku mingit kindlat funktsiooni (nn tsüklifunktsiooni, *round function*). Selle šifri parameetrid on tsüklite arv $r$, plokipikkus $n$, võtmepikkus $k$. Igale tsüklile $i$ leitakse teatud algoritmi abil võtmest $K$ alamvõti $K_i$.




**Joonis 50. Substitutsioon-permutatsioonvõrk (SP-võrk)**

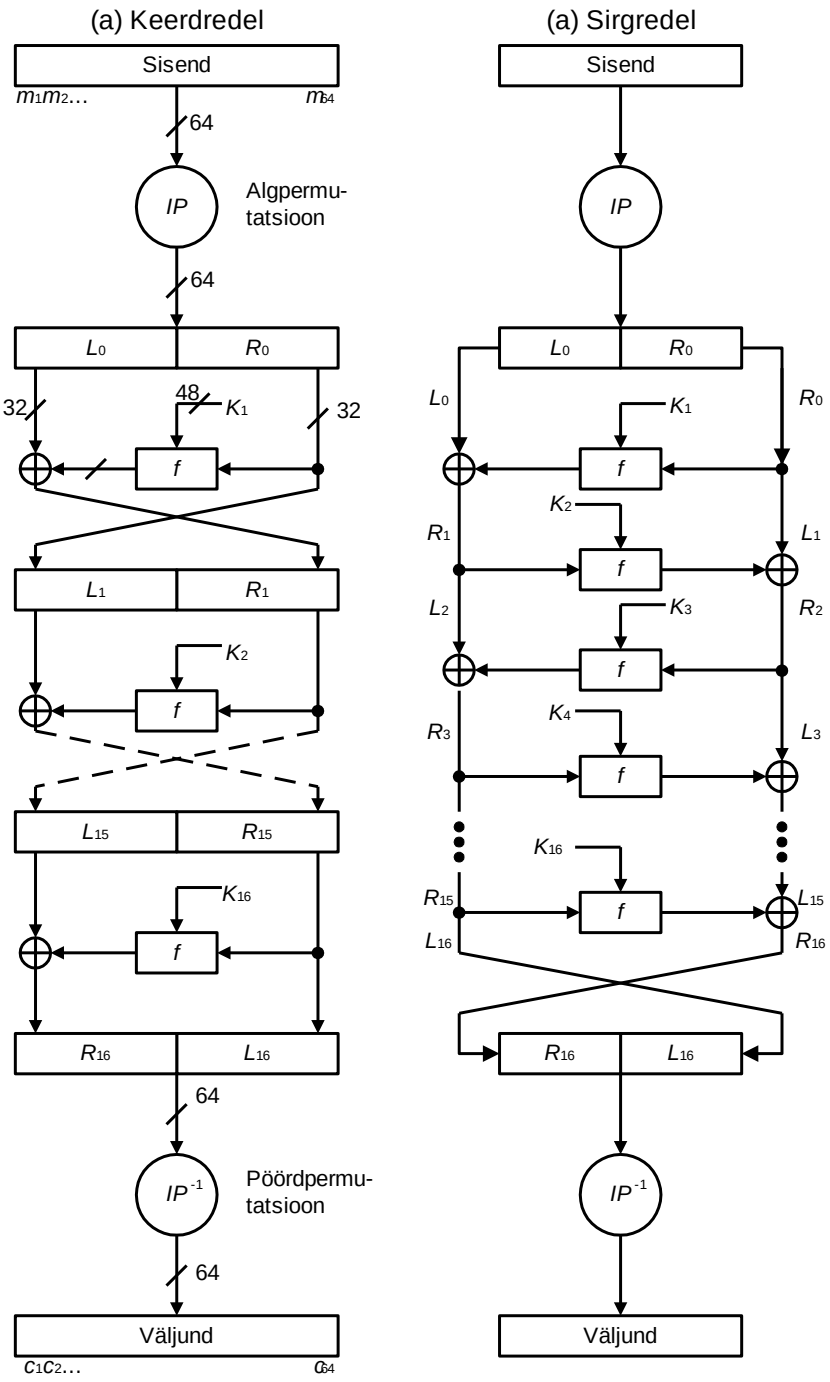
Feisteli šiffer on iteratiivne šiffer, mis kujutab *2t*-bitise avateksti $(L_0,R_0)$, kus $L_0$ ja $R_0$ on mõlemad *t*-bitised alamplokid, krüptogrammiks $(R_r,L_r)$, mis arvutatakse iteratiivselt järgmise skeemi kohaselt:

$$L_i = R_{i-1},$$
$$R_i = L_{i-1} \oplus f(R_{i-1},K_i).$$

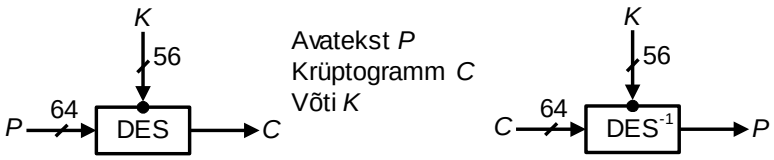
Tavaliselt on Feisteli šifrites $r \geq 3$ ja tavaliselt on see paarisarv. Funktsioon $f$ võib olla näiteks mingi korrutisšiffer, kuid üldiselt ei pea $f$ ise olema pööratav funktsioon.

### 9.2.7 DES

DES on tüüpiline Feisteli šiffer, mille plokipikkus on 64 bitti ja võtmepikkus 56 bitti. Kasutatakse 16 tsüklit. Iga tsükli *i* jaoks tuletatakse 56-bitisest võtmest *K* üks 48-bitine alamvõti $K_i$.




**Joonis 51. Algoritmi DES vookskeem**




**Joonis 52. Algoritmi DES sisend ja väljund**

### *9.2.7.1 Algoritmi DES kirjeldus*

SISEND: 64-bitine avatekst $X=X_1,\ldots,X_{64}$; 64-bitine võti $K=k_1,\ldots,k_{64}$, millest 8 bitti on paarsusbitid.
VÄLJUND: 64-bitine krüptogramm $Y=Y_1,\ldots,Y_{64}$.

1. Arvuta tsüklite alamvõtmed $K_i$ võtmelaiendusalgoritmi järgi.
2. $(L_0,R_0) = \mathrm{IP}(X_1\ldots X_{64})$. Siin tuleb $L_0 = X_{58}X_{50}X_{42}\ldots X_8$ ja $R_0=X_{57}X_{49}X_{41}\ldots X_7$ (tTabel 19).
3. Arvuta kõigi 16 tsükli jaoks iteratiivselt ($i=1,\ldots,16$):

$$L_i = R_{i-1};$$
$$R_i = L_{i-1} \oplus f(R_{i-1},K_i),$$

kus $f(R_{i-1},K_i) = P(S(E(R_{i-1}) \oplus K_i))$, mis arvutatakse järgmiselt.

(a) Laienda $R_{i-1} = r_1r_2\ldots r_{32}$ 48-bitiseks arvuks kasutades funktsiooni $E(\cdot)$ (Tabel 20, Joonis 53). Olgu $T=E(R_{i-1})$, st $T=r_{32}r_1r_2r_3\ldots r_{32}r_1$.

(b) $T' = T \oplus K_i$. Esita $T$ kaheksa 6-bitise sõnana $T'=(B_1,\ldots,B_8)$.

(c) $T'' = (S_1(B_1),S_2(B_2),\ldots,S_8(B_8))$. Siin tähistavad $S_1,\ldots,S_8$ nn s-bokse, st alamsubstitutsioone, millest koosneb substitutsioon $S$. Iga s-boks kujutab 6-bitise sisendi 4-bitiseks väljundiks. Tabel 21 on s-bokside tabelid, mis on koostatud eeldusel, et 6-bitine sisend $B=b_1\ldots b_6$ jagatakse reaks $r=2b_1 + b_6$ ja veeruks $v=b_2b_3b_4b_5$. Näiteks $S_1(011011)=0101$, sest $r=1$, $v=13$ ja vastavast tabelist saame väljundi 5, mis bittesituses on 0101.

(d) $T''' = P(T'')$ (Tabel 20).

1. $b_1b_2\ldots b_{64} = (R_{16},L_{16})$.
2. $Y = IP^{-1}(b_1b_2\ldots b_{64})$ (tabel 46). Saame $Y=b_{40}b_8\ldots b_{25}$.

**Tabel 19. Algpermutatsioon ja tema pöördteisendus**

| IP | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

| $IP^{-1}$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 |
| 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |

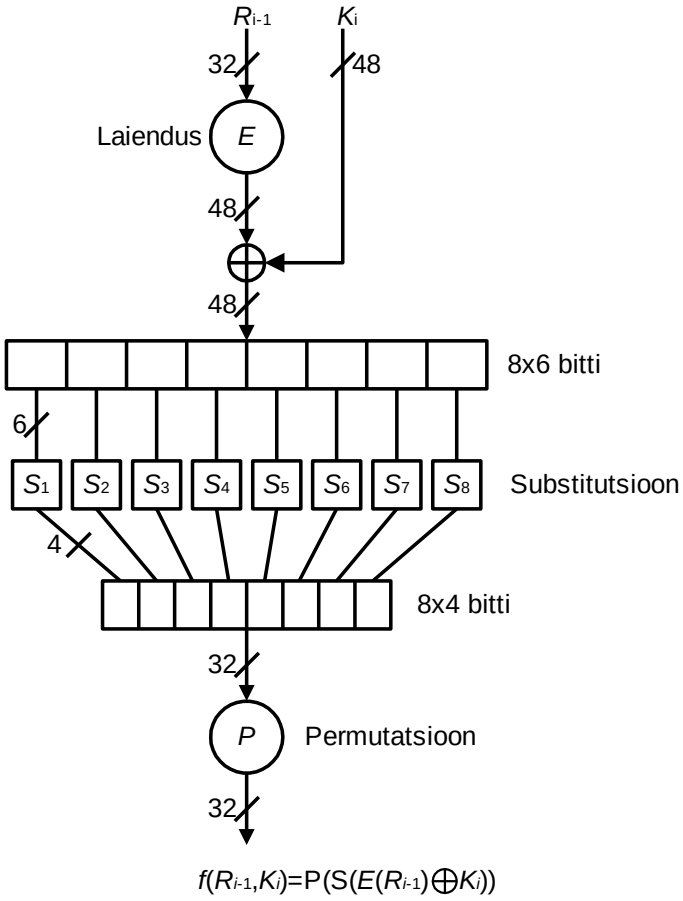
**Tabel 20. DES-i igas tsüklis kasutatavad funktsioonid: laiendus E ja permutatsioon P**

| E | | | | | |
|---|---|---|---|---|---|
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

| P | | | |
|---|---|---|---|
| 16 | 7 | 20 | 21 |
| 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 |
| 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 |
| 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 |
| 22 | 11 | 4 | 25 |

**Tabel 21. DES-i tsükli substitutsiooni kirjeldavad s-boksid**

| Rida | Veeru number | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| | $S_1$ | | | | | | | | | | | | | | | |
| 0 | 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 1 | 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 2 | 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 3 | 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |
| | $S_2$ | | | | | | | | | | | | | | | |
| 0 | 15 | 1 | 8 | 14 | 6 | 11 | 3 | 4 | 9 | 7 | 2 | 13 | 12 | 0 | 5 | 10 |
| 1 | 3 | 13 | 4 | 7 | 15 | 2 | 8 | 14 | 12 | 0 | 1 | 10 | 6 | 9 | 11 | 5 |
| 2 | 0 | 14 | 7 | 11 | 10 | 4 | 13 | 1 | 5 | 8 | 12 | 6 | 9 | 3 | 2 | 15 |
| 3 | 13 | 8 | 10 | 1 | 3 | 15 | 4 | 2 | 11 | 6 | 7 | 12 | 0 | 5 | 14 | 9 |
| | $S_3$ | | | | | | | | | | | | | | | |
| 0 | 10 | 0 | 9 | 14 | 6 | 3 | 15 | 5 | 1 | 13 | 12 | 7 | 11 | 4 | 2 | 8 |
| 1 | 13 | 7 | 0 | 9 | 3 | 4 | 6 | 10 | 2 | 8 | 5 | 14 | 12 | 11 | 15 | 1 |
| 2 | 13 | 6 | 4 | 9 | 8 | 15 | 3 | 0 | 11 | 1 | 2 | 12 | 5 | 10 | 14 | 7 |
| 3 | 1 | 10 | 13 | 0 | 6 | 9 | 8 | 7 | 4 | 15 | 14 | 3 | 11 | 5 | 2 | 12 |
| | $S_4$ | | | | | | | | | | | | | | | |
| 0 | 7 | 13 | 14 | 3 | 0 | 6 | 9 | 10 | 1 | 2 | 8 | 5 | 11 | 12 | 4 | 15 |
| 1 | 13 | 8 | 11 | 5 | 6 | 15 | 0 | 3 | 4 | 7 | 2 | 12 | 1 | 10 | 14 | 9 |
| 2 | 10 | 6 | 9 | 0 | 12 | 11 | 7 | 13 | 15 | 1 | 3 | 14 | 5 | 2 | 8 | 4 |
| 3 | 3 | 15 | 0 | 6 | 10 | 1 | 13 | 8 | 9 | 4 | 5 | 11 | 12 | 7 | 2 | 14 |
| | $S_5$ | | | | | | | | | | | | | | | |
| 0 | 2 | 12 | 4 | 1 | 7 | 10 | 11 | 6 | 8 | 5 | 3 | 15 | 13 | 0 | 14 | 9 |
| 1 | 14 | 11 | 2 | 12 | 4 | 7 | 13 | 1 | 5 | 0 | 15 | 10 | 3 | 9 | 8 | 6 |
| 2 | 4 | 2 | 1 | 11 | 10 | 13 | 7 | 8 | 15 | 9 | 12 | 5 | 6 | 3 | 0 | 14 |
| 3 | 11 | 8 | 12 | 7 | 1 | 14 | 2 | 13 | 6 | 15 | 0 | 9 | 10 | 4 | 5 | 3 |
| | $S_6$ | | | | | | | | | | | | | | | |
| 0 | 12 | 1 | 10 | 15 | 9 | 2 | 6 | 8 | 0 | 13 | 3 | 4 | 14 | 7 | 5 | 11 |
| 1 | 10 | 15 | 4 | 2 | 7 | 12 | 9 | 5 | 6 | 1 | 13 | 14 | 0 | 11 | 3 | 8 |
| 2 | 9 | 14 | 15 | 5 | 2 | 8 | 12 | 3 | 7 | 0 | 4 | 10 | 1 | 13 | 11 | 6 |
| 3 | 4 | 3 | 2 | 12 | 9 | 5 | 15 | 10 | 11 | 14 | 1 | 7 | 6 | 0 | 8 | 13 |
| | $S_7$ | | | | | | | | | | | | | | | |
| 0 | 4 | 11 | 2 | 14 | 15 | 0 | 8 | 13 | 3 | 12 | 9 | 7 | 5 | 10 | 6 | 1 |
| 1 | 13 | 0 | 11 | 7 | 4 | 9 | 0 | 10 | 14 | 3 | 5 | 12 | 2 | 15 | 8 | 6 |
| 2 | 1 | 4 | 11 | 13 | 12 | 3 | 7 | 14 | 10 | 15 | 6 | 8 | 0 | 5 | 9 | 2 |
| 3 | 6 | 11 | 13 | 8 | 1 | 4 | 10 | 7 | 9 | 5 | 0 | 15 | 14 | 2 | 3 | 12 |
| | $S_8$ | | | | | | | | | | | | | | | |
| 0 | 13 | 2 | 8 | 4 | 6 | 15 | 11 | 1 | 10 | 9 | 3 | 14 | 5 | 0 | 12 | 7 |
| 1 | 1 | 15 | 13 | 8 | 10 | 3 | 7 | 4 | 12 | 5 | 6 | 11 | 0 | 14 | 9 | 2 |
| 2 | 7 | 11 | 4 | 1 | 9 | 12 | 14 | 2 | 0 | 6 | 10 | 13 | 15 | 3 | 5 | 8 |
| 3 | 2 | 1 | 14 | 7 | 4 | 10 | 8 | 13 | 15 | 12 | 9 | 0 | 3 | 5 | 6 | 11 |




**Joonis 53. Funktsiooni *f* vooskeem**

### *9.2.7.2 Algoritmi DES võtmelaienduseks*

SISEND: 64-bitine võti $K=k_1 \ldots k_{64}$ (sisaldab 8 paarsusbitti)
VÄLJUND: kuusteist 48-bitist võtit $K_i$, $1 \le i \le 16$.

1. Defineerime suurused $v_i$ ($i$=1…16) järgmiselt: $v_i = 1$, kui $i \in \{1,2,9,16\}$ ja $v_i$=2 muudel juhtudel.
2. $T$ = PC1($K$), kus PC1 on permutatsioon, mis võetakse Tabel 22. Suurus $T$ esitatakse 28-bitiliste pooltena $(C_0,D_0)$. Saame: $C_0=k_{57}k_{49}\ldots k_{36}$ ja $D_0=k_{63}k_{55}\ldots k_4$.
3. Arvuta võtmed $K_i$ järgmise skeemi kohaselt:

   $C_i = (C_{i-1} \text{ rol } v_i)$
   $D_i = (D_{i-1} \text{ rol } v_i)$
   $K_i = \text{PC2}(C_i,D_i)$,

kus rol tähendab bitinihet vasakule ($v_i$ võrra). Teisendus PC2 on permutatsioon, mis on esitatud Tabel 22. Kui $C_i$ ja $D_i$ konkatenatsiooni bitiesitus on $b_1b_2\ldots b_{64}$, saame: $K_i = b_{14}b_{17} \ldots b_{32}$.

**Tabel 22. DES-i võtmelaiendus**

| PC1 | | | | | | |
|---|---|---|---|---|---|---|
| 57 | 49 | 41 | 33 | 25 | 17 | 9 |
| 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 |
| 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| $C_i$ korral ülemine; $D_i$ korral alumine. | | | | | | |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 |
| 21 | 13 | 5 | 28 | 20 | 12 | 4 |

| PC2 | | | | | |
|---|---|---|---|---|---|
| 14 | 17 | 11 | 24 | 1 | 5 |
| 3 | 28 | 15 | 6 | 21 | 10 |
| 23 | 19 | 12 | 4 | 26 | 8 |
| 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 |
| 30 | 40 | 51 | 45 | 33 | 48 |
| 44 | 49 | 39 | 56 | 34 | 53 |
| 46 | 42 | 50 | 36 | 29 | 32 |

**Märkus**. Dekrüpteerimine toimub sarnaselt krüpteerimisega, kuid tsüklite võtmed $K_i$ on vastupidises järjestuses. See on tõepoolest nii, sest:

(1) väljundpermutatsiooni $IP^{-1}$ toime tühistab algpermutatsioon IP andes tulemuseks $(R_{16},L_{16})$;

(2) rakendades tulemusele 1. tsüklit (valem $L_0 \oplus f(R_0,K_1)$), saame antud juhul $R_{16} \oplus f(L_{16},K_{16})$. Et aga $L_{16} = R_{15}$ ja $R_{16} = L_{15} \oplus f(R_{15},K_{16})$, siis

$$L_{15} \oplus f(R_{15}, K_{16}) \oplus f(R_{15}, K_{16}) = L_{15}$$

st esimese dekrüpteerimistsükli tulemus on $(R_{15},L_{15})$, mis tähendabki, et dekrüpteerimine töötab korrektselt, kusjuures funktsiooni $f$ konkreetne kuju pole seejuures oluline.

### 9.2.8 DES-i omadused ja tugevus

Plokkšifritele esitatakse muuhulgas järgmised nõuded:

- iga krüptogrammi bitt peab sõltuma kõigist võtmebittidest,
- avateksti ja krüptogrammi vahel ei tohi olla mingit nähtavat statistilist seost,
- iga bitimuutus avatekstis peab põhjustama mistahes krüptogrammibiti muutustetõenäosusega 0,5,
- iga krüptogrammi muudatus peab põhjustama ettearvamatuid muudatusi taastatud (vastu võetud ja dekrüpteeritud) avatekstis.

DES üldiselt rahuldab kõiki ülaltoodud nõudeid, kuid on ka mõned erandjuhud, mida tuleb alati arvestada.

#### *9.2.8.1 Täiendiomadus*

Olgu $x \in \{0,1\}^n$ ja tähistagu $x^*$ vektori $x$ täiendit, st vektorit mis on saadud vektorist $x$ kõikide koordinaatide (bittide) muutmise teel. Võib näidata, et DES-i jaoks kehtib järgmine nn. täiendiomadus (*complementation property*):

$$E_{K^*}(x^*) = (E_K(x))^*.$$

Täiendiomadusest ei ole tavaliselt mingit abi teadaoleva avatekstiga ammendava võtmeotsingu jaoks, küll aga valitava avatekstiga ründe korral. Kui ründaja teab avatekst-krüptogramm-paare $(X_1,Y_1)$ ja $(X_1^*,Y_2)$, siis täiendiomaduse põhjal mingit võtit $K$ kontrollides piisab kui arvutada $Y=E_K(X_1)$ ja kontrollida, kas $Y \in \{Y_1,Y_2^*\}$. Kui vastus on eitav, on korraga välistatud kaks võtit – nii $K$ kui ka tema täiend $K^*$. See asjaolu vähendab ammendaval võtmeotsingul vajalikku tööd umbes kahekordselt.

### *9.2.8.2 Nõrgad võtmed, poolnõrgad võtmed ja püsipunktid*

Nõrgaks võtmeks nimetatakse sellist võtit $K$, mis rahuldab seost $E_K(E_K(x))=x$ suvalise avateksti $x$ korral. See on näiteks võimalik siis, kui kõik alamvõtmed $K_{1,\ldots,}K_{16}$ on võrdsed. Poolnõrgaks võtmepaariks nimetatakse võtmepaari $(K_1,K_2)$, kus $E_{K1}(E_{K2}(x))=x$.

On teada, et algoritmil DES on neli nõrka võtit (Tabel 23) ja kuus paari poolnõrku võtmeid (Tabel 24).

**Tabel 23. DES-i nõrgad võtmed**

| Nõrk võti (16-ndkujul) | $C_0$ | $D_0$ |
|---|---|---|
| 0101 0101 0101 0101 | $\{0\}^{28}$ | $\{0\}^{28}$ |
| FEFE FEFE FEFE FEFE | $\{1\}^{28}$ | $\{1\}^{28}$ |
| 1F1F 1F1F 0E0E 0E0E | $\{0\}^{28}$ | $\{1\}^{28}$ |
| E0E0 E0E0 F1F1 F1F1 | $\{1\}^{28}$ | $\{0\}^{28}$ |

**Tabel 24. DES-i poolnõrgad võtmed**

| $C_0$ | $D_0$ | Poolnõrk võtmepaar (16-ndkujul) | $C_0$ | $D_0$ |
|---|---|---|---|---|
| $\{01\}^{14}$ | $\{01\}^{14}$ | 01FE 01FE 01FE 01FE,<br>FE01 FE01 FE01 FE01 | $\{10\}^{14}$ | $\{10\}^{14}$ |
| $\{01\}^{14}$ | $\{10\}^{14}$ | 1FE0 1FE0 0EF1 0EF1,<br>E01F E01F F10E F10E | $\{10\}^{14}$ | $\{01\}^{14}$ |
| $\{01\}^{14}$ | $\{0\}^{28}$ | 01E0 01E0 01F1 01F1,<br>E001 E001 F101 F101 | $\{10\}^{14}$ | $\{0\}^{28}$ |
| $\{01\}^{14}$ | $\{1\}^{28}$ | 1FFE 1FFE 0EFE 0EFE,<br>FE1F FE1F FE0E FE0E | $\{10\}^{14}$ | $\{1\}^{28}$ |
| $\{0\}^{28}$ | $\{01\}^{14}$ | 011F 011F 010E 010E,<br>1F01 1F01 0E01 0E01 | $\{0\}^{28}$ | $\{10\}^{14}$ |
| $\{1\}^{28}$ | $\{01\}^{14}$ | E0FE E0FE E1FE E1FE,<br>FEE0 FEE0 FEF1 FEF1 | $\{1\}^{28}$ | $\{10\}^{14}$ |

Püsipunktiks nimetatakse avateksti $x$, mille puhul $E_K(x)=x$. On teada, et DES-i nõrkadel võtmetel on igaühel $2^{32}$ püsipunkti. Neljal poolnõrgal võtmel (tabel 51, ülemised kaks rida) on $2^{32}$ nn antipüsipunkti, st avateksti $x$, mille puhul $E_K(x) = x^*$.

On näidatud, et DES ei ole rühm, st kahekordne krüpteerimine kahe eri võtmega $K_1$ ja $K_2$ ei pruugi vastata ühelegi ühekordsele krüpteerimisele mingi kolmanda võtmega $K_3$. Kõigile erinevaile võtmetele vastav $2^{56}$-elemendiline permutatsioonide hulk ei ole kinnine kompositsiooni suhtes. On isegi näidatud, et rühmas, mille need permutatsioonid genereerivad, on vähemalt $10^{2499}$ elementi (teisendust).

### *9.2.8.3 Praktilised ründed DES-ile*

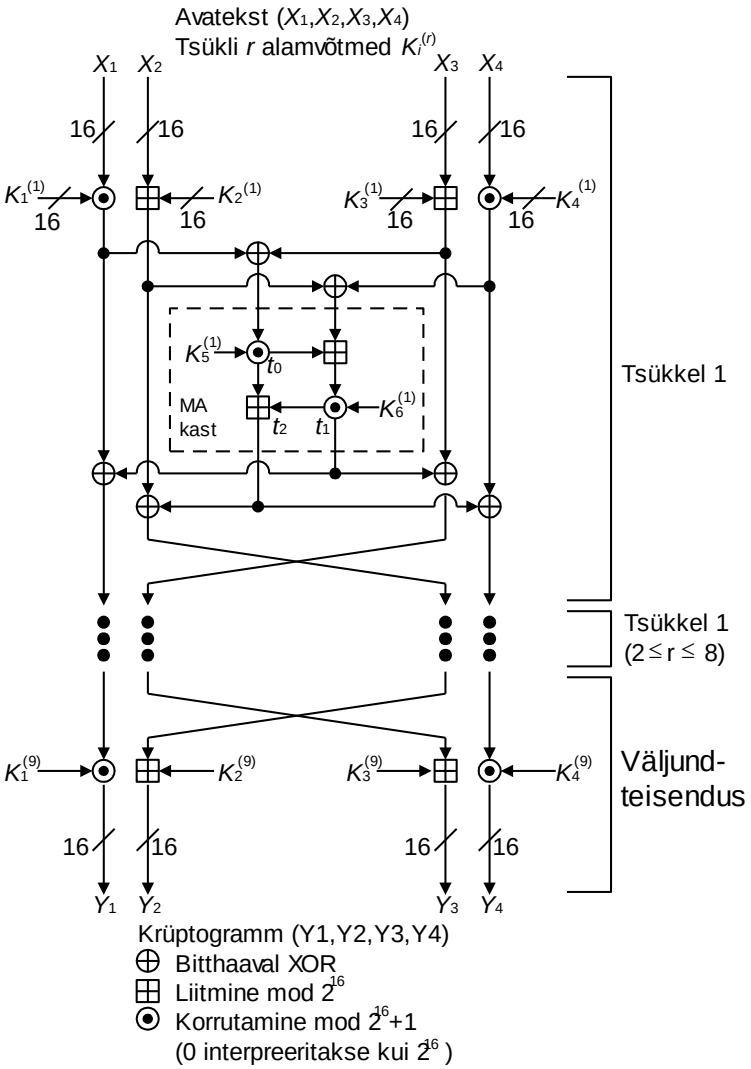
Eeldades, et mõistliku aja jooksul on võimalik saada ülisuurel hulgal avatekst-krüptogramm-paare, on nn lineaarne krüptoanalüüs kõige võimsam ja arvestatavam rünne algoritmi DES vastu. Praktikas on siiski kõige ohtlikum rünne just ammendav võtmeotsing, mis on tänu arvutustehnika odavusele muutunud teostatavaks. Peamise tõuke selleks on andnud suurte programmeeritavate loogikalülituste (FPGA, *field programmable gate array*) kasutamine, mille baasil konstrueeriti 1998. a esimene avalikult teadaolev masin DES-i võtmete efektiivseks leidmiseks ammendava otsingu teel. Masin ise on oma mõõtmetelt kirjutuslaua suurune ja iga 56-bitise võtme leidmiseks kulub keskmiselt paar päeva. Masina omahind on 250 000 dollarit. Nende arvutuste põhjal võib üsna kindlalt väita, et umbes 100 000 kr eest on võimalik

ehitada masin, mis leiab võtme 3 kuuga ja 50 000 kr eest masin, mis leiab võtme poole aastaga. Kõik vajalikud komponendid on ka Eesti kaubandusvõrgus vabalt kättesaadaval.

Üsna levinud on arvamus, et juba DES-i tekkehetkel oli NSA (National Security Agency) käsutuses masin efektiivse ammendava võtmeotsingu teostamiseks. Arvatakse, et DES-i efektiivse võtmepikkuse vähendamine 64-lt bitilt 56-le, mis vähendas ammendavaks otsinguks vajalikku töömahtu tervelt 256 korda, oli tingitud just neist kaalutlustest. Kas neil spekulatsioonidel ka alust on, võib selguda alles aastate pärast.

### 9.2.9 IDEA

Plokkšiffer IDEA (*International Data Encryption Algorithm*, "rahvusvaheline andmekrüpteerimise algoritm") krüpteerib 64-bitise avateksti 64-bitiseks krüptogrammiks, kasutades 128-bitist võtit *K*. IDEA ei ole klassikaline Feisteli šiffer, vaid kasutab üldisemat skeemi, mis võimaldab piirduda 8 tsükliga (DES-il on neid 16). Iga tsükkel *r* kasutab kuut 16-bitist alamvõtit $K_i^{(r)}$, kus $0<i<7$. Avatekst *X* jagatakse neljaks 16-bitiseks sisendplokiks $(X_1,\ldots,X_4)$, mis teisendatakse lõpuks samuti neljaks 16-bitiseks väljundplokiks $(Y_1,\ldots,Y_4)$. Viimane, kaheksas tsükkel kasutab ka väljundteisendust, milles rakendatakse nelja lisavõtit $K_i^{(9)}$, kus $0<i<5$.




**Joonis 54. Algoritmi IDEA skeem**

IDEA projekteerimise põhiidee oli kolme erineva $2^{16}$-elemendilise (algebralise) rühma tehete kombineerimine. Lisaks bitikaupa XOR-ile $\oplus$ ja liitmisele mod $2^{16}$ $\boxplus$ kasutatakse ka korrutamist mod $2^{16}+1$ $\odot$, kusjuures arv $2^{16}$ kodeeritakse kui 0.

### *9.2.9.1 IDEA krüpteerimisalgoritm*

SISEND: 64-bitine avatekst $M=m_1,\ldots,m_{64}$; 128-bitine võti $K=k_1\ldots k_{128}$.
VÄLJUND: 64-bitine krüptogramm $Y=(Y_1,Y_2,Y_3,Y_4)$.

1. Leida tsüklitele 16-bitised alamvõtmed $K_1^{(r)},\ldots,K_6^{(r)}$, kus $r=1\ldots8$, ja väljundteisenduse võtmed $K_1^{(9)},\ldots,K_4^{(9)}$.
2. $(X_1,X_2,X_3,X_4) = (m_1\ldots m_{16}, m_{17}\ldots m_{32}, m_{33}\ldots m_{48}, m_{49}\ldots m_{64})$.
3. Arvutada iteratiivselt iga tsükli kohta, st $r=1\ldots8$:
   *(a)* $X_1 := X_1 \odot K_1^{(r)}$; $X_4 := X_4 \odot K_4^{(r)}$; $X_2 := X_2 \boxplus K_2^{(r)}$; $X_3 := X_3 \boxplus K_3^{(r)}$.
   (b) $t_0 := K_5^{(r)} \odot (X_1 \oplus X_3)$; $t_1 := K_6^{(r)} \odot (t_0 \boxplus (X_2 \oplus X_4))$; $t_2 := t_0 \oplus t_1$.
   (c) $X_1 := X_1 \oplus t_1$; $X_4 := X_4 \oplus t_2$; $a := X_2 \oplus t_2$; $X_2 := X_3 \oplus t_1$; $X_3 := a$.

4. $Y_1 := X_1 \odot K_1^{(9)}$; $Y_4 := X_4 \odot K_4^{(9)}$; $Y_2 := X_3 \boxplus K_2^{(9)}$; $Y_3 := X_2 \boxplus K_3^{(9)}$.

### *9.2.9.2 IDEA võtmelaiendusalgoritm krüpteerimiseks*

SISEND: 128-bitine võti $K=k_1\ldots k_{128}$.
VÄLJUND: 16-bitised võtmed $K_i^{(r)}$, $(0<r<9,\ 0<i<7)$ ja 16-bitised võtmed $K_1^{(9)}\ldots K_4^{(9)}$.

1. Järjestada alamvõtmed järgmiselt: $K_1^{(1)}\ldots K_6^{(1)},K_1^{(2)}\ldots K_6^{(2)},\ldots,K_1^{(8)}\ldots K_6^{(8)},K_1^{(9)}\ldots K_4^{(9)}$
2. Jagada võti $K$ kaheksaks 16-bitiseks plokiks ja omistada need järjekorras esimestele alamvõtmetele.
3. Nihutada $K$ kui 64-bitine arv 25 biti võrra tsükliliselt vasakule ja siirduda uuesti sammule 2, kuni kõik 52 alamvõtit on saanud oma väärtuse.

### *9.2.9.3 IDEA dekrüpteerimine*

Dekrüpteerimine toimub sama algoritmi järgi, mis krüpteeriminegi, kuid alamvõtmed $K'^{(r)}_i$ on arvutatud erinevalt. Alamvõtmete arvutamise eeskiri on Tabel 25. Tähis $-K_i$ tähendab võtme $K_i$ aditiivset vastandelementi, st 16-bitist arvu $x$, nii et

$$K_i + x \bmod 2^{16} = 0,$$

milleks sobib järelikult arv $2^{16} - K_i$. Tähis $K_i^{-1}$ tähendab arvu $K_i$ multiplikatiivset pöördelementi, st arvu $1 \le x \le 2^{16}$, nii et

$$K_i \cdot x \bmod (2^{16} + 1) = 1.$$

Sellise arvu $x$ leidmisel kasutatakse Eukleidese laiendatud algoritmi (vt "Avaliku võtmega krüptosüsteemid").

**Tabel 25. Dekrüpteerimisel kasutatavad (IDEA) alamvõtmed $K'^{(r)}_i$**

| Tsükkel $r$ | $K'^{(r)}_1$ | $K'^{(r)}_2$ | $K'^{(r)}_3$ | $K'^{(r)}_4$ | $K'^{(r)}_5$ | $K'^{(r)}_6$ |
|---|---|---|---|---|---|---|
| $r=1$ | $(K_1^{(10-r)})^{-1}$ | $-K_2^{(10-r)}$ | $-K_3^{(10-r)}$ | $(K_4^{(10-r)})^{-1}$ | $K_5^{(9-r)}$ | $K_6^{(9-r)}$ |
| $1<r<9$ | $(K_1^{(10-r)})^{-1}$ | $-K_3^{(10-r)}$ | $-K_2^{(10-r)}$ | $(K_4^{(10-r)})^{-1}$ | $K_5^{(9-r)}$ | $K_6^{(9-r)}$ |

| | | | | | | |
|---|---|---|---|---|---|---|
| $r=9$ | $(K_1^{(10-r)})^{-1}$ | $-K_2^{(10-r)}$ | $-K_3^{(10-r)}$ | $(K_4^{(10-r)})^{-1}$ | - | - |

Tehe $\odot$ erineb tavalisest mooduliga $2^{16}+1$ korrutamisest selle poolest, et arv $2^{16}$ (mis on kongruentne arvuga $-1$ mod $2^{16}+1$) asendatakse arvuga 0, sest esimene ei mahu muidu 16-bitiste arvude piirkonda. Näiteks, $0 \odot 1 = 0$, sest

$$2^{16} \cdot 1 \bmod (2^{16} + 1) = 2^{16}.$$

**Tabel 26. IDEA krüpteerimise arvutusnäide**

| 128-bitine võti $K=(1,2,3,4,5,6,7,8)$ | | | | | | | 64-bitine avatekst $M=(0,1,2,3)$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | $K_1^{(r)}$ | $K_2^{(r)}$ | $K_3^{(r)}$ | $K_4^{(r)}$ | $K_5^{(r)}$ | $K_6^{(r)}$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
| 1 | 0001 | 0002 | 0003 | 0004 | 0005 | 0006 | 00F0 | 00F5 | 010A | 0105 |
| 2 | 0007 | 0008 | 0400 | 0600 | 0800 | 0A00 | 222F | 21B5 | F45E | E959 |
| 3 | 0C00 | 0E00 | 1000 | 0200 | 0010 | 0014 | 0F86 | 39BE | 8EE8 | 1173 |
| 4 | 0018 | 001C | 0020 | 0004 | 0008 | 000C | 57DF | AC58 | C65B | BA4D |
| 5 | 2800 | 3000 | 3800 | 4000 | 0800 | 1000 | 8E81 | BA9C | F77F | 3A4A |
| 6 | 1800 | 2000 | 0070 | 0080 | 0010 | 0020 | 6942 | 9409 | E21B | 1C64 |
| 7 | 0030 | 0040 | 0050 | 0060 | 0000 | 2000 | 99D0 | C7F6 | 5331 | 620E |
| 8 | 4000 | 6000 | 8000 | A000 | C000 | E001 | 0A24 | 0098 | EC6B | 4925 |
| 9 | 0080 | 00C0 | 0100 | 0140 | - | - | 11FB | ED2B | 0198 | 6DE5 |

**Tabel 27. IDEA dekrüpteerimise arvutusnäide**

| 128-bitine võti $K=(1,2,3,4,5,6,7,8)$ | | | | | | | C=(11FB,ED2B,0198,6DE5) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $r$ | $K'^{(r)}_1$ | $K'^{(r)}_2$ | $K'^{(r)}_3$ | $K'^{(r)}_4$ | $K'^{(r)}_5$ | $K'^{(r)}_6$ | $X_1$ | $X_2$ | $X_3$ | $X_4$ |
| 1 | FE01 | FF40 | FF00 | 659A | C000 | E001 | D98D | D331 | 27F6 | 82B8 |
| 2 | FFFD | 8000 | A000 | CCCC | 0000 | 2000 | BC4D | E26B | 9449 | A576 |
| 3 | A556 | FFB0 | FFC0 | 52AB | 0010 | 0020 | 0AA4 | F7EF | DA9C | 24E3 |
| 4 | 554B | FF90 | E000 | FE01 | 0800 | 1000 | CA46 | FE5B | DC58 | 116D |
| 5 | 332D | C800 | D000 | FFFD | 0008 | 000C | 748F | 8F08 | 39DA | 45CC |
| 6 | 4AAB | FFE0 | FFE4 | C001 | 0010 | 0014 | 3266 | 045E | 2FB5 | B02E |
| 7 | AA96 | F000 | F200 | FF81 | 0800 | 0A00 | 0690 | 050A | 00FD | 1DFA |
| 8 | 4925 | FC00 | FFF8 | 552B | 0005 | 0006 | 0000 | 0005 | 0003 | 000C |
| 9 | 0001 | FFFE | FFFD | C001 | - | - | 0000 | 0001 | 0002 | 0003 |

Algoritmile IDEA pole seni leitud ühtki rünnet, mis oleks efektiivsem ammendavast võtmeotsingust, mis aga 128-bitise võtme korral on absoluutselt võimatu kõigi tänapäeval kasutatavate meetodite ja ressurssidega. Edu võiks siin oodata üksnes kvantarvutite kasutuselevõtu korral (vt "9.7 Kvantkrüptograafia"). IDEA momendil kõige nõrgem koht on üksnes suhteliselt väike (võrreldes võtmepikkusega) plokipikkus.

### 9.2.10 AES

#### *9.2.10.1 Arenduse algus*

1997. a alguses alustas NIST (*National Institute of Standards and Technology*) DES-i järglase (tingnimetusega AES, *Advanced Encryption Standard*) arendamise protsessi. Erinevalt 70tel toimunud DES-i ning 90te alguses toimunud SHA arendusest otsustati nii AES-i valik kui ka krüptoanalüüs läbi viia avalikult. 1997. a 15. aprillil toimunud AES-i tööseminaril avalikustati ka esmased nõuded uuele standardile.

- AES peab olema "tugev" krüptoalgoritm valitsus- ja kommertskasutuseks.
- Plokkšifrite standardsete tööviiside toetus.
- Oluliselt suurem efektiivsus kui 3DES-il nii tarkvaras kui riistvaras
- Toetus erinevatele võtmepikkustele, et oleks võimalik vastavalt vajadusele turvataset tõsta.
- AES peab olema valitud "ausalt" ja avatult, st avalikult defineeritud (näiteks S-bokside väärtuste kriteeriumid avalikud) ning avalikult väärtustatav.

1997. a septembris täpsustati nii tingimusi kui ka edasist ajakava AES-i valikul. Nõuti

- 128-bitist plokisuurust,
- tuge 128-, 192- ja 256-bitistele võtmepikkustele,
- tuge standardsetele tööviisidele ning paindlikkust rakendustes (plusspunktiks võimalused kasutada šifrit räsifunktsioonina jms),
- AES-i suuremat kiirust võrreldes 3DES-iga nii riist- kui tarkvaras, nii 8-bitistel protsessoritel kui ka Javas teostatult,
- AES-i suuremat turvalisust 3DES-iga võrreldes,
- AES-i lõivutust *(royalty free*), nii et ta oleks kõigile vabalt kasutatav kõigis olukordades.

Ajakava.

- Kandidaadid peavad NIST-ile olema esitatud 1998. 15. juuniks;
- 1998. 15 juuliks teatab NIST kandidaatide autoritele, kas nende paketid on kõlblikud (hiljem avaldatud kriteeriumideks olid kandidaatpaketi täielikkus ja vastavus esitatud nõuetele võtme- ja plokipikkuse jms mõttes ning paketis olevate testprogrammide tulemuste vastavus algoritmi kirjeldustes toodud nn testväärtustele);
- 1998. aasta augustile määrati esimene AES-i kandidaatide tööseminar, kus avaldatakse kõik esitatud kandidaadid koos täielike pakettidega ning edasine täpne ajakava.

***Esimene AES-i kandidaatide tööseminar***

Kuigi enamik kandidaate oli avaldatud juba enne 1998. 20.–22. augustil Venturas (California, USA) toimunud seminari, oli üks kandidaat veel salajaseks jäänud. Avalikkuse eest varjul olid ka täielikud kandidaatpaketid ning viiteprogrammid (*reference code*). Seetõttu oli seminarile kogunenud maailma krüptoparemik, sh enamik käesolevas peatükis mainitud veel elavatest teadlastest. Eesti ainsa esindajana oli kohal Küberneetika AS vanemteadur Helger Lipmaa.

### *9.2.10.2 Kandidaatide täielik nimekiri*

NIST kuulutas kõlblikuks 15 kandidaati, millest enamiku olid esitanud maailma juhtivad andmeturbefirmad või teadustöörühmad. Täielik nimekiri on selline: CAST-256 (Entrust Technologies), Crypton (Future Systems, Inc), DEAL (Richard Outerbridge, Lars Knudsen), DFC (CNRS ja Ecole Normale Superieure), E2 (Nippon Telegraph and Telephone Corporation), FROG (TecApro Internacional S.A.), HPC (Rich Schroeppel), LOKI97 (Lawrie Brown, Josef Pieprzyk, Jennifer Seberry), MAGENTA (Deutsche Telekom AG), MARS (IBM), RC6 (RSA Laboratories), RIJNDAEL (Joan Daemen, Vincent Rijmen), SAFER+ (Cylink Corporation), SERPENT (Ross Anderson, Eli Biham, Lars Knudsen), TWOFISH (Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson).

Nende algoritmide puhul olid juba enne seminari teada nõrkused LOKI97-s (neid leidsid Vincent Rijmen ja Lars Knudsen) ja FROG-is (leidsid David Wagner, Niels Ferguson ja Bruce Schneier); napilt enne seminari algust avalikustas Stefan Lucks rünnaku DEAL-i vastu. Ainus enne seminari algust salajas hoitud algoritm, MAGENTA, murti lahti n-ö käigupealt, 15 minuti jooksul pärast ettekande lõppu (seda tegid Eli Biham, Alex Biryukov, Niels Ferguson, Lars Knudsen, Bruce Schneier ja Adi Shamir)!

Pärast seminari on leitud ka väikesi vigu MARS-i võtmelaiendusalgoritmis.

(Jaotis 9.2.11 kirjutati 1998. a oktoobris ega sisalda seetõttu neid algoritme puudutavaid hilisemaid avastusi.)

#### *9.2.10.3 Edasine ajakava*

- Seminarile järgneval perioodil toimub avalik krüptoanalüüs, mille käigus selgunud tulemuste põhjal valitakse teisel AES-i kandidaatide tööseminaril (22.–23. märts 1999, Rooma) välja viis–kuus kandidaati, mis jätkavad konkursi teises voorus.
- Teise vooru kestus on 6–9 kuud, mille järel toimub kolmas AES-i kandidaatide tööseminar, kus valitakse võitja.

Kogu AES-i valimise protsessi puudutav teave on avaldatud aadressil http://aes.nist.gov.

## 9.3 Juhuarvude genereerimine

Enamiku krüptograafiliste süsteemide tegelik turvalisus sõltub otseselt juhuslike (etteennustamatute) suuruste genereerimisest. Näiteks algoritmi DES salajane võti, RSA kaks algarvu *p* ja *q*, autentimisprotokollides kasutatavad juhuarvud (nonsid, *nonce*) jne.

Juhuslike bittide generaator on vahend, mis väljastab juhuslikke statistiliselt sõltumatuid bitte. Juhuslike bittide generaatorit saab kasutada juhuarvude generaatorina. Näiteks kui soovime juhuarvu vahemikus *[0...n]*, piisab, kui genereerime $\log_2(n)$ juhuslikku bitti ja kombineerime neist arvu. Kui saadud arv on suurem kui *n*, on üks võimalus genereerida lihtsalt uus bitijada. Juhuslikke arve on võimalik genereerida aparatuurselt, kasutades mitmesuguseid looduses leiduvaid tõeliselt juhuslikke müraallikaid. Paljudel juhtudel aga pole võimalik neid kasutada ettenähtud efektiivsusega, st praktikas on sageli vaja saada juhuarve kiiremini kui see on võimalik puhtaparatuursete vahenditega, mistõttu on kasutusel nn pseudojuhuslike arvude generaatorid.

Pseudojuhuslike bittide generaator on mingi deterministlik algoritm, mis mingist tõeliselt juhuslikust *k*-bitisest järjendist teeb *l*-bitilise järjendi, kus *l* on tunduvalt suurem kui *k*, mis on teatavas mõttes eristamatu tõeliselt juhuslikust jadast. On selge, et tegelikult ei ole väljastatud pseudojuhuslik *l*-bitine jada juhuslik, sest generaatori kõikvõimalikud väljundid (neid on $2^k$) moodustavad vaid tühise osa kõikvõimalikest *l*-bitistest jadadest (neid on $2^l$). Pseudojuhuslike bittide generaatori põhiotstarve on selles, et ründaja ei oleks suuteline mõistliku aja- ja resursikuluga eristama generaatori väljundjada tõeliselt juhuslikust jadast.

### 9.3.1 Lineaarsetel kongruentsidel põhinevad generaatorid

Paljude kõrgkeelte kompilaatorites on sisseehitatud juhuarvude generaatorid, mis põhinevad lineaarsetel arvkongruentsidel (*linear congruential generator*), milles juhuslike arvude jada $x_0, x_1, \ldots, x_k$ saadakse järgmise põhimõtte järgi:

$$x_{i+1} = a\,x_i + b \bmod m,$$

kus *a* ja *b* on mingid konstandid ja $x_0$ on algväärtus, mis tavaliselt saadakse ajafunktsioonist või paremal juhul ka mingitest muudest parameetritest.

Hoolimata sellest, et lineaarsetel kongruentsidel põhinevad generaatorid on heade statistiliste omadustega, st läbivad edukalt enamiku statistilisi teste, ei ole need generaatorid krüptograafiliselt turvalised, kuna nende poolt tekitatud väljundjada on ekstrapoleeritav isegi siis, kui parameetrid *a*, *b* ja *m* ei ole ründajale teada, nagu näitasid Frieze, Kannan ja Lagarias 1984. aastal.

### 9.3.2 Põhinõuded pseudojuhuslikkusele

Pseudojuhuslike bittide generaatorile esitatav minimaalne nõue on, et algne *k*-bitine tõeliselt juhuslik jada on piisavalt pikk, et vältida ammendavat otsingut üle selle jada $2^k$ võimaliku erineva väärtuse.

Öeldakse, et pseudojuhuslike bittide generaator läbib kõik polünomiaalsed statistilised testid, kui ei leidu polünomiaalses ajas töötavat algoritmi, mis eristab generaatori väljundi niisama pikast juhuslike bittide jadast tõenäosusega, mis on oluliselt suurem 0,5-st.

Pseudojuhuslike bittide generaator läbib "järgmise biti" testid (*next-bit tests*), kui ei leidu polünomiaalses ajas töötavat algoritmi, mis esimese *l* biti järgi ennustaks biti *l+1* väärtuse tõenäosusega, mis on oluliselt suurem 0,5-st.

On võimalik näidata, et generaator läbib "järgmise biti" testid siis ja ainult siis, kui ta läbib kõik polünomiaalsed statistilised testid.

Generaatorit, mis läbib "järgmise biti" testid eeldusel, et kehtib mingi usutav, ent tõestamata matemaatiline hüpotees (näiteks arvude teguriteks lahutamise keerulisus), nimetatakse krüptograafiliselt turvaliseks pseudojuhuslike arvude generaatoriks.

### 9.3.3 Tõeliselt juhuslikud bitigeneraatorid

Tõeliselt juhuslik bitigeneraator nõuab looduses esinevaid tõelisi juhuslikkuse allikaid. Ilma sisemiste korrelatsioonideta ja ideaalselt juhuslikust jadast eristamatut bitijada genereeriva aparatuuri või tarkvara projekteerimine on äärmiselt komplitseeritud ülesanne. Krüptograafilised rakendused nõuavad lisaks veel seda, et potentsiaalne ründaja ei saaks generaatorit väliselt mõjutada isegi juhul, kui generaatori ehitus ja tööprintsiibid on talle teada. Looduslikel allikatel põhinevaid pseudojuhuslikke generaatoreid tuleb perioodiliselt testida statistiliste meetoditega.

#### *9.3.3.1 Aparatuursed generaatorid*

Aparatuursed generaatorid kasutavad mingi füüsikalise nähtuse poolt garanteeritud juhuslikkust. Sellised generaatorid võivad tekitada bitijadasid, kus on korrelatsioonid ja tasakaalustamata 0 ja 1 väljastus. Seetõttu tuleb nende väljundit enamasti korrigeerida (vt. 9.3.3.1 Jadade parandamine). Loodusliku juhuslikkusena võib kasutada:

- Osakeste eraldumise ajavahemikke aine radioaktiivsel lagunemisel;
- Termomüra pooljuhtdioodist või takistist;
- Koormamata kõrgsagedusgeneraatori võnkesageduse ebastabiilsust;
- Kondensaatori laengut, mille ta kogub fikseeritud ajavahemiku jooksul;
- Õhu turbulentse, mis tekivad kõvaketta pöörlemisest ja põhjustavad mõõdetavaid häiringuid sektorite lugemisaegades;
- Heli mikrofonist või videosignaali kaamerast.

Esimesed kaks meetodit realiseeritakse enamasti välisseadmena, mis ühendatakse arvutiga mingi liidese abil. Seetõttu on ründajal sageli võimalik generaatori väljundit mõjutada. Teised kaks meetodit võimaldavad juhuarvude generaatorit realiseerida samas kiibis koos teiste krüptograafiavahenditega, mis raskendab ründaja tegevust tunduvalt.

#### *9.3.3.2 Tarkvaralised generaatorid*

Turvaliste juhuarvude genereerimine tarkvaras on aparatuursete lahendustega võrreldes veelgi keerukam. Tarkvaraline pseudojuhuslike arvude generaator võib kasutada:

- süsteemi taktsagedust,
- klahvivajutuste ja hiire liikumiste vahelisi ajaintervalle,
- sisend- ja väljundpuhvrite sisu,
- kasutaja sisestusi,
- operatsioonisüsteemi sisemist informatsiooni (näiteks võrgu statistikat).

Kõigi nende parameetrite statistiline käitumine võib varieeruda sõltuvalt arvutisüsteemi iseärasustest. Samuti ei ole garanteeritud, et ründaja ise ei mõõda neidsamu suurusi ega mõjuta neid tahtlikult. Head generaatorid peavad kasutama kõikvõimalikke häid juhuslikkuse allikaid, et mõjutamine oleks raskem. Juhuarvude genereerimine algab juhuslikkuse kogumisega erinevatest allikatest ja nende salvestamisega.

Seejärel segatakse kõik salvestatud andmed kokku, kasutades mingit ühesuunalist räsifunktsiooni või ka plokkšifrit.

#### *9.3.3.3 Jadade parandamine*

Statistiliste omaduste parandamiseks juhul, kui ühe ja nulli esinemise tõenäosused on erinevad, võib kasutada järgmist meetodit. Jagame generaatorist tuleva bitijada kahebitisteks sõnadeks ja kasutame järgmist kodeerimisviisi:

```
01    -- >    0
10    -- >    1
00    -- >    jääb kasutamata
11    -- >    jääb kasutamata
```

Selline kodeerimisviis garanteerib nullile ja ühele võrdsed väljatuleku tõenäosused. Üldisem ja laialt kasutatav (kuid mitte tõestatud) viis statistiliste omaduste parandamiseks on ühesuunaliste räsifunktsioonide kasutamine esialgsete jadade teisendamisel.

### 9.3.4 ANSI X9.17 pseudojuhuslike arvude generaator

Ühesuunalist funktsiooni $f$ saab kasutada pseudojuhuslike arvude genereerimisel, valides esmalt mingi algväärtuse $s$, ja seejärel rakendades funktsiooni $f$ jadale $s$, $s+1$, $s+2$, ... saades tulemuseks jada

$$f(s),\ f(s+1),\ f(s+2),\ ...$$

Ühesuunaliste funktsioonidena võib kasutada krüptograafilisi räsifunktsioone (näiteks SHA-1) või ka plokkšifreid (näiteks DES) mingi salajase võtmega.

Järgnev algoritm on ANSI X9.17 juhuarvude generaator, mis on mõeldud tarkvararakendustes DES-i võtmete juhuslikuks genereerimiseks.

SISEND: Juhuslik ja salastatud 64-bitine arv $s_0$ (seeme, *seed*), täisarv $m$ ja 112-bitine 3-DES-i võti $K$, mis on mõeldud krüpteerimiseks nn. E-D-E süsteemis (*two-key triple-encryption*), st krüpteeritakse 3 korda, kusjuures esimesel ja kolmandal korral kasutatakse krüpteerimist ühe ja sama võtmega, teisel korral dekrüpteerimist teise võtmega (vt 9.2, "Plokkšifrid").

#### *9.3.4.1 Generaatori lähtestamine*

Lähtestusel (funktsioon `randomize()`) arvutatakse suurus $I$ lähtudes hetke aja võimalikult täpsest esitusest Date_Time järgmisel viisil:

$$I := \{Date\_Time\}_K.$$

#### *9.3.4.2 Pseudojuhusliku jada genereerimine*

Pseudojuhuslik jada $x_1, \ldots x_k$ genereeritakse järgmiste rekurrentsete seoste põhjal:

$$x_{i+1} = \{I \oplus s_i\}_K$$
$$s_{i+1} = \{I \oplus x_{i+1}\}_K.$$

Vahel on soovitatav algväärtuse $s_0$ genereerimisel kasutada aparatuurset juhuslike bittide generaatorit.

### 9.3.5 Krüptograafiliselt turvalised generaatorid

Senivaadeldud generaatorite turvalisust ei ole seni tõestatud. Ehkki sellised generaatorid on praktikas laialdaselt kasutusel, on mõnedel juhtudel vaja täpsemaid turvalisuse põhjendusi ja argumente, st on vaja generaatorit, mille turvalisus on tõestatav matemaatiliselt, eeldades teatud ülesannete lahendamise raskust. Esitame ühe sellise generaatori idee, mis põhineb suurte arvude teguriteks lahutamise raskusel.

1? Valime kaks juhuslikku algarvu $p$ ja $q$ ja arvutame $n=pq$ ja $\phi=(p-1)(q-1)$.
2? Valime juhusliku arvu $e$, nii et $1<e<\phi$ ja SÜT($e$, $\phi$)=1.
3? Valime juhusliku arvu $x_0$ lõigust $[1,n-1]$.
4? Genereerime jada $z_1, ..., z_l$ järgmisel rekurrentsel viisil:

$$x_i := x_{i-1}^e \bmod n$$
$$z_i := \text{madalaim bitt arvust } x_i.$$

## 9.4 Jadašifrid

Jadašifrid moodustavad ühe olulisima krüptograafiliste primitiivide klassi. Neid kasutatakse siis, kui on vaja krüpteeritud andmeid edastada viivituseta. Tavaliselt on jadašifrid plokkšifritest aparatuursetes teostustes kiiremad ja neid on ka lihtsam teostada. Kuna jadašifrid krüpteerivad andmeid väiksemate üksuste kaupa (plokkšifritega võrreldes), on nad eriti vajalikud siis, kui sideliinidel on kõrge müratase.

Plokkšifrid on ühed enimuuritud krüptograafilised primitiivid, mille kohta on kogutud hulgaliselt teoreetilisi ja praktilisi teadmisi. Sellegipoolest on vähe avalikult kättesaadavaid valmislahendusi, sest enamik praktiliselt kasutatavaid jadašifreid on salastatud. Ehkki jadašifrid on ühed vanimad krüpteerimisvahendid ei ole lähemal ajal ette näha nende põhimõttelist vananemist. Ka tänapäeval koostatakse palju uusi jadašifreid.

### 9.4.1 Ühekordne šifriplokk (*one-time pad*)

Vernami pakutud šiffer 1926. aastast liitis (mod 2) iga avateksti bitti $x_i$ vastava võtmebitiga $z_i$ ja sai vastav krüptogrammibiti $y_i$, st

$$y_i = x_i \oplus z_i,$$

kus iga võtmebitti tohtis kasutada vaid ühe avateksti biti krüpteerimiseks. Shannon tõestas 1949. aastal avaldatud artiklis, et kui võtmebitid on genereeritud juhuslikult ja üksteisest sõltumatult (seega $P(X=x|Y=y)=P(X=x)$), siis on Vernami šiffer täielikult ja tingimusteta turvaline, st krüptogramm $Y=(y_1,...,y_i,...,y_n)$ ei sisalda ründaja jaoks vähimatki informatsiooni avateksti $X=(x_1,...,x_i,...,x_n)$ kohta. Avateksti entroopia $H(X|Y)$, eeldusel et krüptogramm $Y$ on teada, on sama, mis avateksti esialgne entroopia $H(X)$, st

$$\begin{aligned} H(X|Y) &= -\sum_{x,y} P(Y=y)\, P(X=x|Y=y)\, \log_2 P(X=x|Y=y) = \\ &= -\sum_{x,y} P(Y=y)\, P(X=x)\, \log_2 P(X=x) = -\sum_y P(Y=y) \sum_x P(X=x)\, \log_2 P(X=x) = \\ &= \left(\sum_y P(Y=y)\right)\cdot\left(-\sum_x P(X=x)\, \log_2 P(X=x)\right) = -\sum_x P(X=x)\, \log_2 P(X=x) = \\ &= H(X). \end{aligned}$$

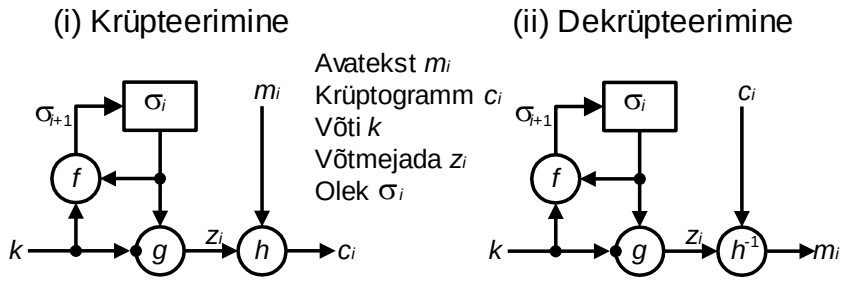
Selleks, et tõepoolest nii oleks, peab võtmebitte olema vähemalt niisama palju kui avateksti bitte. Eelmisest jaotisest teame, et tõeliselt juhuslike bittide genereerimine on kallis ja tegelikkuses tuleb sageli piirduda pseudojuhusliku võtmejadaga, mis on genereeritud tunduvalt lühemast tõeliselt juhuslikust jadast. Jadašifrid ongi ajendatud sellest asjaolust. Nad kasutavad pseudojuhuslikku võtmejadageneraatorit eesmärgiga muuta krüptoanalüüs ründajale võimalikult raskeks.

### 9.4.2 Sünkroonsed jadašifrid

Sünkroonseks nimetatakse sellist jadašifrit, kus võtmejada genereeritakse sõltumatult avatekstist ja krüptogrammist. Olgu $M=m_1,...,m_i,...$ avatekst, $Z=z_1,...,z_i,...$ võtmejada ja $C=c_1,...,c_i,...$ krüptogramm. Krüpteerimisprotseduuri sünkroonses jadašifris võib kirjeldada järgmiste võrranditega:

$$\begin{aligned} \sigma_{i+1} &= f(\sigma_i,k), \\ z_i &= g(\sigma_i,k), \\ c_i &= h(z_i,m_i), \end{aligned}$$

kus $k$ on salajane võti. Dekrüpteerimine toimub analoogiliselt, kuid funktsiooni h asemel kasutatakse tema pöördfunktsiooni (vt Joonis 55).



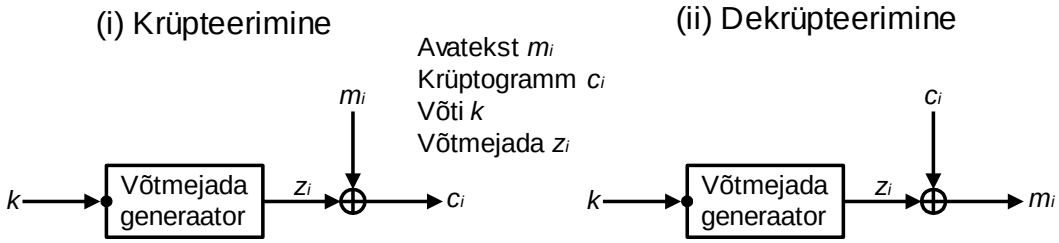

**Joonis 55. Sünkroonse jadašifri üldine mudel**

Sünkroonse jadašifri peamine puudus on see, et saatja ja vastuvõtja peavad oma võtmejadasid sünkroniseerima. Kui sünkroonsus kaob juhusliku häire või tahtliku sekkumise tõttu, ei ole muud teha, kui generaatorid uuesti lähtestada ja sünkroniseerida, mis on aga üpris keerukas ülesanne.

Sünkroonse jadašifreerimise üks hea külg on andmeedastusvigade leviku puudumine, st kui mingi krüptogrammibitt muudab saatmise või vastuvõtu ajal oma väärtust, ei põhjusta see teiste bittide muutusi.

Ühelt poolt on hea, et ründajal pole võimalik (või on ülimalt raske) lisada krüptogrammi uusi bitte või eemaldada olemasolevaid, ilma et kaoks sünkronisatsioon, mis on krüptogrammi saajale kergesti märgatav. Teisest küljest, tehes muutusi mingites bittides, on ründaja kindel, et vastuvõetud avateksti tekivad täpselt samade bittide muutused. Seega on siin vaja rakendada lisameetmeid andmetervikluse tagamiseks.

Suurem osa tänini avaldatud jadašifritest on nn aditiivsed šifrid, st sünkroonsed jadašifrid, milles funktsioonina $h$ kasutatakse liitmist mod 2, st $h(m,z)=m \oplus z$. (Vt Joonis 56).

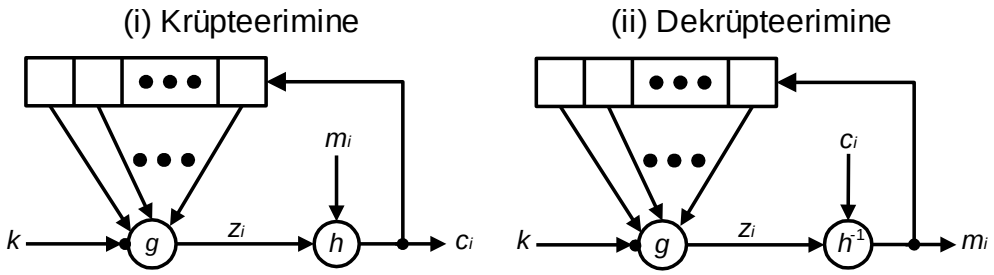



**Joonis 56. Aditiivse jadašifri skeem**

### 9.4.3 Isesünkroniseeruvad jadašifrid

Isesünkroniseeruvaks nimetatakse jadašifrit, kus võtmejada mistahes bitt on funktsioon võtmest ja mingist hulgast eelnevatest krüptogrammibittidest. Krüpteerimisfunktsiooni saab esitada järgmiste üldiste võrranditega:

$$\sigma_i = (c_{i-t},c_{i-t+1},\ldots,c_{i-1}),$$
$$z_i = g(\sigma_i,k),$$
$$c_i = h(z_i,m_i).$$




**Joonis 57. Isesünkroniseeruva generaatori üldine mudel**

Kõige enam kasutatavad isesünkroniseeruvad jadašifrid põhinevad plokkšifritel, mis töötavad tagasisidega krüptogrammist. Nende peamine eelis on selles, et isegi kui ründaja lisab bitte olemasolevate krüptogrammibittide vahele või kustutab olemasolevaid, ei põhjusta see uut lähtestust ja sünkroniseerimist, vaid šiffer ise sünkroniseerib end *t* sammu pärast. Põhjus on lihtne – võtmejada generaatori sisemine olek $\sigma_i$ sõltub ainult *t*-st järjestikusest krüptogrammibitist.

Vigade levik on loomulikult suurem kui sünkroonsetel jadašifritel, kuid see on piiratud *t* järgmise bitiga, st kui viimasest veast on möödunud *t* sammu, on edasine dekrüpteerimine korrektne kuni järgmise veani.

Modifitseerides krüptogrammi bitte põhjustab ründaja mitme järgneva biti väära dekrüpteerimise, mis suurendab rünnete avastamise tõenäosust. Seevastu on raskem tuvastada bittide tahtlikku lisamist ja vanade krüptogrammide taasesitust. Seega tuleb ka siin rakendada vajalikke lisameetmeid andmetervikluse tagamiseks.

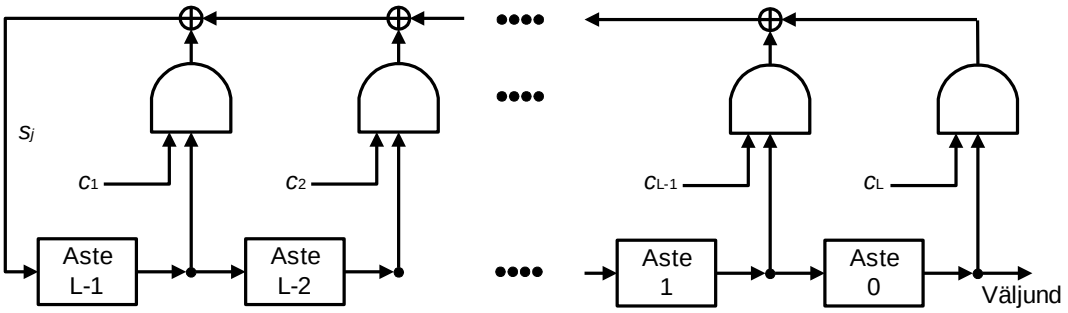
Kuna iga avateksti bitt mõjutab kõiki järgnevaid krüptogrammi bitte, on avateksti statistilised omadused hajutatud ega ole nii kergesti avastatavad. Seetõttu on isesünkroniseeruvad jadašifrid parem kaitse avateksti liiasusel põhinevate rünnete vastu.

### 9.4.4 Lineaarsed nihkeregistrid

Lineaarsed nihkeregistrid (*linear feedback shift registers*, LFSR) on kasutusel paljude jadašifrite võtmejadageneraatoris. Selleks on mitmeid põhjusi:

- LFSR-id on väga mugavad aparatuurseks realiseerimiseks;
- nad võimaldavad saada pika perioodiga jadasid;
- nende väljundjadad on heade statistiliste omadustega;
- nende omadusi on võimalik uurida algebraliste meetoditega, mille headuses on veendutud sajanditepikkuse uurimistööga (algebra hakkas eriti jõudsalt arenema juba 18. sajandil).

Lineaarseks nihkeregistriks pikkusega *L* nimetatakse automaati, mis koosneb viivitustüüpi trigeritest moodustatud *L* astmest (*stage*), mis on nummerdatud arvudega *0...L-1*. Iga triger on võimeline salvestma üht bitti, tal on üks sisend ja üks väljund (vt Joonis 58).



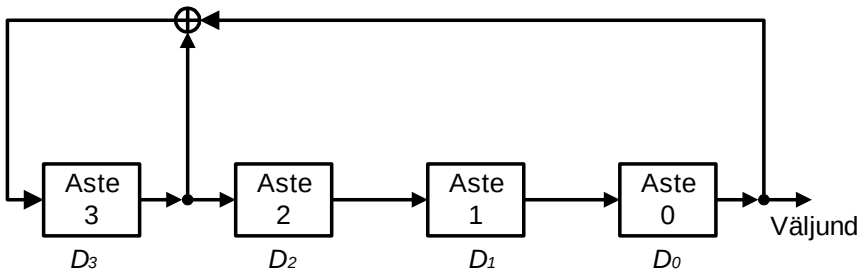

**Joonis 58. Lineaarse nihkeregistri üldskeem**

Registri kui lõpliku automaadi algolek $S$ on määratud trigerite seisuga, mida kirjeldavad olekubitid $S=[s_{L-1}...s_0]$. Registri väljundjada $s_0,...,s_{L-1},s_L,s_{L+1},...$ on üheselt määratud järgmise rekurrentse seosega:

$s_j = c_1 s_{j-1} + c_2 s_{j-2} + ... + c_L s_{j-L}$,

kus $j \geq L$ ja + tähistab liitmist mod 2. Polünoomi $C(D) = 1 + c_1D + c_2D^2 + ... + C_LD^L$ nimetatakse registri struktuuripolünoomiks. On selge, et polünoom $C(D)$ ja algolek $S$ määravad üheselt registri edasise käitumise.

**Näide** (vt Joonis 59). Kui $S$=[0110] ja $C(D)=1+D+D_4$, siis väljundjada (perioodiga 15) on 011001000111101... ja vastav olekute jada on [0110], [0011], [1001], [0100], [0010], [0001], [1000], [1100], [1110], [1111], [0111], [1011], [0101], [1010], [1101], [0110]. Seega on register pärast 15 biti väljastust uuesti algolekus. Kui algolek olnuks [0000], tulnuks ka kõik ülejäänud olekud nullid ja seega koosneks ka väljastatav bitijada üksnes nullidest. Et neljabitiseid olekuid on üldse 16, on vaadeldud registri väljundjada seega pikim, mida on võimalik saavutada 4-bitise lineaarse nihkeregistri abil.




**Joonis 59. Lineaarne nihkeregister struktuuripolünoomiga *C(D) = 1 + D + D4***

Lineaarsete nihkeregistrite kohta on teada järgmised faktid, mis põhinevad klassikalistel algebratulemustel.

- Kui $C(D)$ on *taandumatu* üle korpuse $GF(2)$, st ei leidu mittekonstantseid polünoome $C_1$ ja $C_2$ nii et $C=C_1 \cdot C_2$ mod 2, siis iga mittenulline algolek annab võrdse pikkusega väljundjada. Selle väljundjada pikkus on võrdne vähima positiivse täisarvuga $N$, mille korral polünoom $1+D^N$ jagub täpselt polünoomiga $C(D)$.
- Kui $C(D)$ on nn *primitiivne* üle korpuse $GF(2)$, st polünoom $D$ on polünoomide jäägiklassikorpuse $GF(2)[D]/C(D)$ multiplikatiivse rühma moodustaja, siis registri väljundjada periood on maksimaalne võimalik, st $2^L-1$.

Primitiivseid polünoome on võimalik genereerida küllaltki lihtsa algoritmi abil, mida aga siin ei esitata, sest nende kirjelduse ja korrektsuse põhjendamisel vajalike algebraliste mõistete kirjeldamine pole selle raamatu piiratud mahtu arvestades võimalik. Oluline on siin vaid teadmine, et on olemas mistahes järku primitiivseid polünoome.

Lineaarne nihkeregister eraldi võetuna ei ole krüptograafiliselt turvaline võtmejada generaator, olgugi et tema väljundstatistika on väga lähedane tõelisele juhuslikkusele. Selle fakti illustreerimiseks toome järgneva näite. Oletame, et pseudojuhuslike arvude generaatorina kasutatakse väljundit, mille genereerib lineaarne nihkeregister algolekuga $S$=[100] ja struktuuripolünoomiga $C(D) = 1 + x + x^3$. Tema väljundjada $s_0s_1...s_i...$ tuleb arvutada valemiga $s_j = s_{j-1} + s_{j-3}$. Väljundjada perioodiks tuleb 7 ja jada ise tuleb 0011101...

Oletame, et ründaja ei tea ei registri struktuuripolünoomi ega ka algolekut. Ainus, mis ta teab või oletab, on nihkeregistri pikkus $L=3$. Oletame, et ründaja teab mingit seitset järjestikust bitti väljundjadas. Lihtsuse mõttes oletame, et teadaolev lõik on $[s_0...s_5]$. Näitame, kuidas ründaja saab nende andmete põhjal kindlaks teha struktuuripolünoomi $C(D)=1+c_1D + c_2D^2 + c_3D^3$. Ründaja kirjutab välja registri tööd kirjeldavad üldised rekursiivsed võrrandid ja saab kolme tundmatuga kolmest võrrandist koosneva lineaarvõrrandisüsteemi

$$s_3 = c_1 s_2 + c_2 s_1 + c_3 s_0 = c_1 \cdot 1 + c_2 \cdot 0 + c_3 \cdot 0 = 1$$
$$s_4 = c_1 s_3 + c_2 s_2 + c_3 s_1 = c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 0 = 1$$
$$s_5 = c_1 s_4 + c_2 s_3 + c_3 s_2 = c_1 \cdot 1 + c_2 \cdot 1 + c_3 \cdot 1 = 0$$

mille lahendamisel selgub, et $c_1=1$, $c_2=0$ ja $c_3=1$, st ründaja saab teada, et $C(D)=1+x+x^3$, mistõttu ta on edaspidi võimeline jada ekstrapoleerima. Üldjuhul on ründajal vaja teada $2L$ järjestikust väljundjada elementi ja lahendada $L$ võrrandist koosnev lineaarvõrrandisüsteem, mida on võimalik teha piisavalt kiiresti.

### 9.4.5 Lineaarne keerukus

Oletame, et mingi lineaarne nihkeregister genereerib jada $s=s_0,s_1,s_2,...,s_n$ selle registri väljundjada mingi alamlõiguna. Jada s lineaarseks keerukuseks $L(n)$ nimetatakse vähimat arvu $L$, mille korral leidub jada $s$ genereeriv $L$-bitine lineaarne nihkeregister. Kui $s$ on nulljada, võetakse kokkuleppeliselt $L(s)=0$. Lõpmatu jada $s$ korral on veel võimalik juhtum, kus ei leidugi antud jada genereerivat lineaarset nihkeregistrit. Sel juhul võetakse kokkuleppeliselt $L(s)=\infty$. Lineaarse keerukuse kohta on teada järgmised omadused.

- Kui $n>0$ ja $s^n$ on $n$-elemendiline jada, siis $0 \le L(s) \le n$.
- $L(s^n)=0$ parajasti siis, kui $s_n$ on nulljada.
- $L(s^n)=n$ parajasti siis, kui $s^n$=000...01.
- Kui $s$ on $N$-perioodiline, siis $L(s) \le N$.
- $L(s \oplus t) \le L(s) + L(t)$, kus $\oplus$ tähendab jadade bitthaaval liitmist mod 2.

On ka teada, et kui $s$ on valitud juhuslikult ja ühtlase jaotusega kõigi $n$-elemendiliste bitijadade hulgast, siis selle jada lineaarse keerukuse $L(s)$ matemaatiline ootus on ligikaudu $n/2$.

Olgu $s=s_0,s_1,s_2,...,s_n,...$ mingi (võib olla lõpmatu) jada ja $L_N$ tähistagu alamjada $s_0,...,s_{N-1}$ lineaarset keerukust. Arvude jada $L_1,L_2,....$ nimetatakse jada $s$ lineaarse keerukuse profiiliks. Mingi suvalise jada lineaarse keerukuse profiili kohta on teada järgmist:

- kui $N \le M$, siis $L_N \le L_M$;
- kui $L_N < L_{N+1}$, siis $L_N \le N/2$;
- kui $L_N < L_{N+1}$, siis $L_N + L_{N+1} = N+1$.

Kui $s$ on juhuslike bittide jada, siis mistahes indeksi $N$ korral suuruse $\min\{j \mid L_N < L_{N+j}\}$ matemaatiline ootus on 2, kui $L_N \le N/2$ ja $2 + 2L_N - N$, kui $L_N > N/2$. Kui mingi jada $s$ lineaarse keerukuse profiil on lähedane juhusliku jada profiilile, ei tähenda see sugugi, et jada on tõepoolest juhuslik. Näiteks vaatleme jada $s$, kus $s_i=1$, kui $i=2^j-1$, mingi $j$ korral, ja $s_i=0$ kui sellist arvu $j$ ei leidu. Selle jada korral $L_N=\lfloor (N+1)/2 \rfloor$, st tema lineaarse keerukuse profiil on lähedane juhusliku jada omale, kuid jada ise ei ole hoopiski juhuslik.

### 9.4.6 Lineaarse keerukuse leidmine

Lineaarset keerukust saab leida Berlekamp-Massey algoritmi abil, mis põhineb järgmistel teadaolevatel faktidel. Olgu antud mingi lõplik bitijada $s^N = s_0,s_1,...,s_{N-1}$ ja lineaarne nihkeregister struktuuripolünoomiga $C(D)$, mis genereerib jada $s_N$. Olgu

$$d_N = s_N \oplus \sum_{0 \le i \le L} c_i \cdot s_{N-i} = 1,$$

st lineaarne nihkeregister ei "ennusta" jada $N$-ndat elementi enam õigesti. Olgu $m<N$ suurim selline arv, mille korral $L(s^m)<L(s^N)$ ja olgu $B(D)$ mingi $m$-astme struktuuripolünoom, mis genereerib jada $s^m$. Siis avaldub jada $S^{N+1}$ genereeriva lühima lineaarse nihkeregistri pikkus $L'$ valemiga

$$L' = \begin{cases} L, & \text{kui } L > N/2 \\ N + 1 - L, & \text{kui } L \le N/2 \end{cases}$$

ja selle registri struktuuripolünoom $C'(D)$ avaldub valemiga:

$$C'(D) = C(D) + B(D) \cdot D^{N-m}.$$

**Algoritm (Berlekamp-Massey algoritm)**:

SISEND: Bitijada $s^n = s_0,s_1,...,s_{n-1}$.
VÄLJUND: Lineaarne keerukus $L(s^n)$.

1. Lähtestus: $C(D) := 1;\ L := 0;\ m := -1;\ B(D) := 1;\ N := 0.$

2. Kuni ($N < n$), täita järgmist programmilõiku (a)—(c)
   (a) Arvutada d := $s_N \oplus \sum_{0 \le i \le L} c_i \cdot s_{N-i}$
   (b) Kui $d=1$ siis
   $T(D) := C(D);\ C(D) := C(D) + B(D) \cdot D^{N-m}.$
   Kui $L \le N/2$ siis $L := N + 1 - L;\ m := N;\ B(D) := T(D).$
   (c) $N := N + 1.$

3. Väljastada $L$.

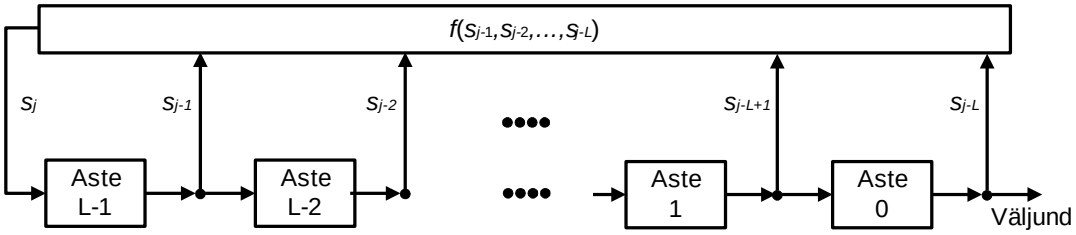
Berlekamp-Massey algoritm nõuab $O(n^2)$ bititehet, mis on küllaldane efektiivsus demonstreerima, et jada lineaarne keerukus on mistahes ründajale kergesti arvutatav suurus. See on taas põhjendus, miks ainuüksi lineaarsete nihkeregistrite kasutamisest ei piisa krüptograafiliselt turvaliste lahenduste projekteerimiseks.

### 9.4.7 Mittelineaarsed nihkeregistrid

Mittelineaarne nihkeregister on struktuurilt sarnane lineaarsele nihkeregistrile, kuid lineaarne tagasisidefunktsioon $s_j = c_1 s_{j-1} + c_2 s_{j-2} + ... + c_L s_{j-L}$ on asendatud üldise mittelineaarse funktsiooniga

$s_j = f(s_{j-1}, s_{j-2}, ..., s_{j-L})$,

mis sarnaselt lineaarse nihkeregistriga määrab üheselt registri kui automaadi käitumise, kui algolek $S=[s_{L-1}...s_0]$ on fikseeritud (vt Joonis 60).




**Joonis 60. Mittelineaarse nihkeregistri üldskeem**

Nihkeregistrit nimetatakse mittesingulaarseks, kui igale algolekule vastab perioodiline väljundjada. On teada, et register on mittesingulaarne parajasti siis, kui funktsioon $f$ avaldub kujul

$$f(s_{j-1}, s_{j-2}, \ldots, s_{j-L}) = s_{j-L} \oplus g(s_{j-1}, s_{j-2}, \ldots, s_{j-L-1}),$$

kus $g$ on mingi $L$-$1$-muutuja Boole'i funktsioon. Kui $L$-bitise mittesingulaarse nihkeregistri periood on $2^L$, siis nimetatakse seda registrit de Bruijni registriks ja tema väljastatavat jada de Bruijni jadaks.

**Näide**. Kui 3-bitises nihkeregistris kasutada tagasisideks funktsiooni

$$f(x,y,z) = 1 \oplus y \oplus z \oplus xy,$$

saame de Bruijni registri. Kui algolek on nulljada, tuleb vastav olekute jada [000], [100], [110], [111], [011], [101], [010], [001].

De Bruijni jadad on küllaltki heade statistiliste omadustega, st lõigud pikkusega $\leq L$ esinevad peaaegu samasuguse sagedusega kui tõeliselt juhuslikus jadas.

### 9.4.8 Lineaarsete nihkeregistrite kasutamine jadašifrites

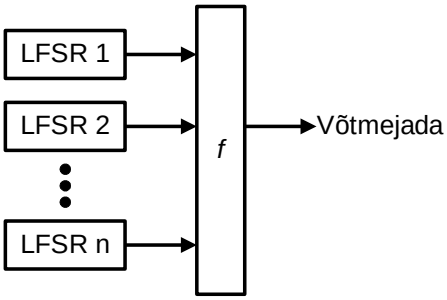
Ehkki lineaarsed nihkeregistrid ise pole turvalised, saab neid väga hästi kasutada kombineeritult mittelineaarsete komponentidega. Selleks on mitmeid võimalusi:

- kasutada mitut lineaarset nihkeregistrit ja võtta nende väljundid kokku mingi mittelineaarse funktsiooniga;
- kasutada mittelineaarset filtrifunktsiooni ühe nihkeregistri oleku teisendamiseks;
- kasutada ühe nihkeregistri väljundit teise registri taktimiseks.

Kuna üks võtmejadageneraatorile esitatav nõue on pikk periood, tuleb komponentidena kasutatavad lineaarsed nihkeregistrid valida maksimaalse perioodiga, st nad peavad olema primitiivse struktuuripolünoomiga, mis võib olla nii avalik kui ka salajane. Kui struktuur on salajane, tuleb vastav primitiivne polünoom valida juhuslikult, ühtlase jaotusega üle kõikvõimalike antud järku primitiivsete polünoomide hulga. Salajane struktuur esitab ühest küljest lisanõudeid kasutatavale aparatuurile ja suurendab inimteguri osa turvalisuse tagamisel (projekteerival inseneril suur vastutus), teisest küljest aga võimaldab kasutada lühemaid nihkeregistreid.

#### *9.4.8.1 Mittelineaarne kombineerimine*

Üks üldine võte lineaarsete nihkeregistrite krüptograafiliste omaduste parandamiseks on mitme erineva registri kasutamine ja nende väljundite kombineerimine mingi mittelineaarse Boole'i funktsiooniga $f$ (vt Joonis 61).




**Joonis 61. Eri registrite väljundite mittelineaarne kombineerimine**

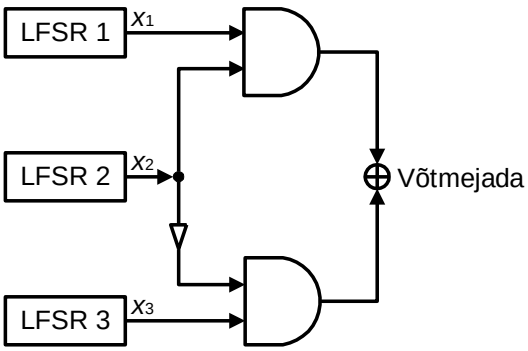
Korrutist, mis on moodustatud $m$ erinevast muutujast, nimetatakse $m$-järku korrutiseks. On lihtne näidata, et iga $n$-muutuja Boole'i funktsiooni $f$ saab esitada $m$-järku ($m \le n$) korrutiste summana mod 2. Sellist funktsiooni $f$ esitust nimetatakse tema algebraliseks normaalkujuks. Funktsiooni $f$ mittelineaarsuse järguks nimetatakse suurima järguga korrutise astet tema algebralises normaalkujus. Näiteks Boole'i funktsiooni

$$f(x,y,z,t) = 1 \oplus x \oplus xy \oplus yzt$$

mittelineaarsuse järk on 3. On võimalik näidata, et mida suurem on kombineeriva funktsiooni $f$ mittelineaarsuse järk, seda suurem on generaatori väljundi lineaarne keerukus. See põhineb ilusal tulemusel, mis väidab, et generaatori (Joonis 61) lineaarne keerukus $L$ avaldub valemiga

$$L = f(L_1,...,L_n),$$

kus $L_1,...,L_n$ on vastavate lineaarsete nihkeregistrite lineaarsed keerukused, funktsioon $f$ on esitatud algebralisel normaalkujul ja kõik arvutused on aritmeetilised, st toimuvad täisarvudega.




**Joonis 62. Geffe generaator**

**Näide**. Geffe generaator (Joonis 62) koostatakse kolmest maksimaalse pikkusega väljundjadaga lineaarsest nihkeregistrist, mille pikkused $L_1$, $L_2$ ja $L_3$ on paarikaupa ühistegurita arvud. Registrite mittelineaarseks kombineerimiseks kasutatakse Boole'i funktsiooni

$$f(x_1,x_2,x_3) = x_2\,x_1 \ \oplus \ (1 \oplus x_2)\,x_3 = x_1\,x_2 \ \oplus \ x_2\,x_3 \ \oplus \ x_3.$$

Registri väljundjada periood on $(2^{L1}-1)\cdot(2^{L2}-1)\cdot(2^{L3}-1)$ ja lineaarne keerukus on arvutatav valemiga

$$L = L_1\,L_2 \ + \ L_2\,L_3 \ + \ L_3.$$

Hoolimata oma väljundjada suurest lineaarsest keerukusest on Geffe generaator krüptograafiliselt nõrk, sest näiteks tõenäosus, et esimese nihkeregistri väljund $x_1$ langeb kokku generaatori väljundiga $z=f(x_1,x_2,x_3)$ on

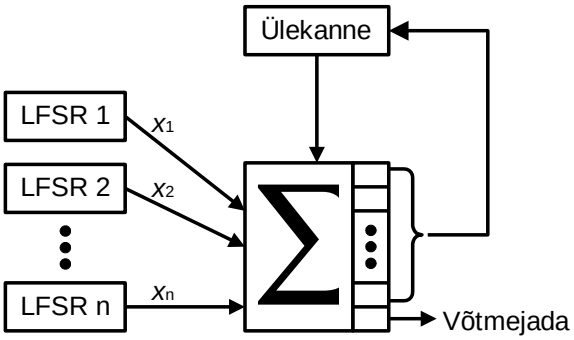
$$P(x_1 = z) = P(x_2 = 1) + P(x_2 = 0)\cdot P(x_3 = x_1) = 0.5 + 0.5\cdot 0.5 = 0.75,$$

st informatsioon esimese lineaarse nihkeregistri väljundi väärtuse kohta "lekib" generaatori väljundisse ja võimaldab teostada suure tõenäosusega õnnestuvaid ründeid esimese nihkeregistri oleku kindlakstegemiseks. Kui ründajal on teada lineaarsete nihkeregistrite struktuuripolünoomid ja kasutada küllalt pikk väljundjada lõik, siis võib ta uurida väljundjada korrelatsiooni esimese registri kõikvõimalike väljundjadadega (neid on $2^{L1}-1$) ja avastada sel viisil suure tõenäosusega esimese registri õige algolek. Nii võib teha kõikide "nõrkade" komponentregistritega. Geffe generaatoril on ka kolmas komponent nõrk, st $P(x_3=z)=0.75$.

Olgu $X_1,X_2,...,X_n$ sõltumatud juhuslikud bitid, mille mõlema oleku (0 või 1) tõenäosused on võrdsed, st 0.5. Boole'i funktsiooni $f(x_1,...,x_n)$ nimetatakse $m$-järku korrelatsiooni-immuunseks, kui tema väljund on statistiliselt sõltumatu ükskõik millisest fikseeritud $m$-elemendilisest argumentide hulgast, st nende argumentide väärtuste teadasaamine ei aita vähimalgi määral kaasa funktsiooni väärtuse entroopia vähendamisele. Näiteks, funktsioon

$$f(x_1,...,x_n) = x_1 \oplus ... \oplus x_n$$

on $n$-1-järku korrelatsiooni-immuunne, kuid kahjuks lineaarne. Suurendades funktsiooni mittelineaarsuse järku, tuleb vähendada korrelatsiooni-immuunsuse järku. On teada, et kui $n$-muutuja Boole'i funktsioon on $m$-järku korrelatsiooni-immuunne, siis tema mittelineaarsuse järk ei saa olla suurem kui $n$-$m$. Lineaarse keerukuse suurendamist korrelatsiooni-immuunsuse arvel (või vastupidist) saab ära hoida mäluelementide lisamisega kombineerivale funktsioonile $f$.




**Joonis 63. Liitmisgeneraator**

**Näide**. Liitmisgeneraator (Joonis 63) põhineb faktil, et bitijadadena esitatud täisarvude liitmine on mittelineaarne Boole'i funktsioon hea korrelatsiooni-immuunsusega. Olgu $a=a_0 + 2^1a_1 + ... + 2^{m-1}a_{m-1}$ ja $b=b_0 + 2^1 b_1 + ... + 2^{m-1}b_{m-1}$ argumentide $a$ ja $b$ bitiesitused. Summa $z = a + b$ bitid $z_0,...,z_m$ avalduvad järgmiste rekurrentsete valemitega:

$$z_j = f_1(a_j,b_j,c_{j-1}) = a_j \oplus b_j \oplus c_{j-1}$$
$$c_j = f_2(a_j,b_j,c_{j-1}) = a_j\, b_j \oplus (a_j \oplus b_j)\, c_{j-1},$$

kus $c_j$ on ülekanne ($c_{-1}=0$). Head korrelatsiooni- ja mittelineaarsusomadused tulenevad faktist, et funktsioon $f_1$ on teist järku korrelatsiooni-immuunne ja $f_2$ on mittelineaarne Boole'i funktsioon. Ülekanne eelmisest järgust kannab seetõttu kogu eelmise järgu mittelineaarsuse üle järgmisse järku.

Liitmisgeneraatoris kasutatakse paarikaupa ühistegurita perioodidega $L_1,...,L_n$ lineaarseid nihkeregistreid. Salajane võti koosneb nende generaatorite algolekutest ja ülekande algväärtusest $C_0$. Igas $j$-ndas taktis

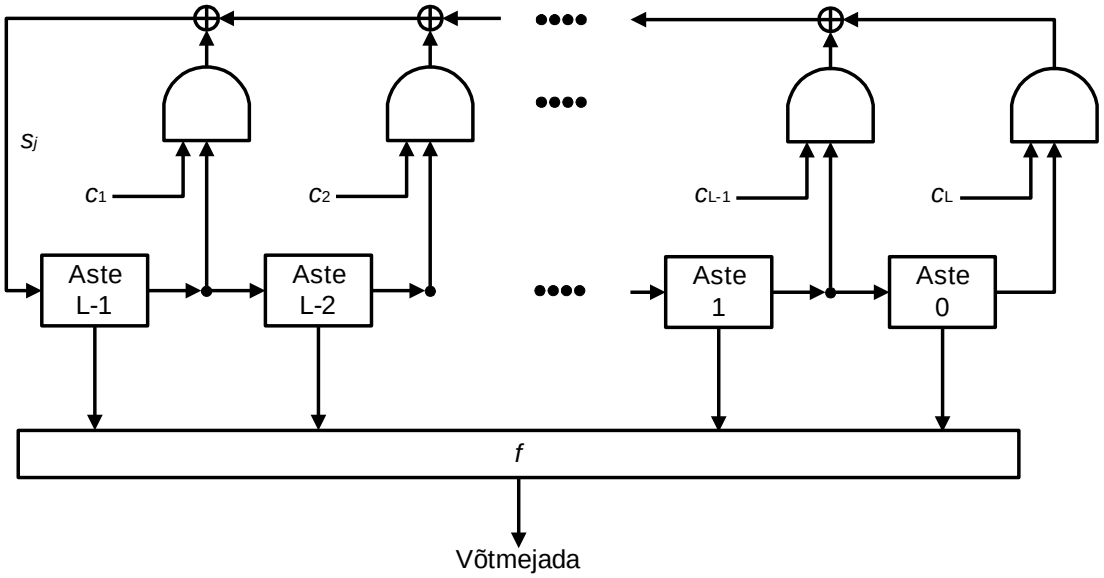
väljastavad lineaarsed nihkeregistrid oma väljundbitid $x_1,...,x_n$, mis seejärel koos eelmise ülekandega $C_{j-1}$ aritmeetiliselt liidetakse ja saadud summa (mod 2)

$S_j = \sum_{1 \le i \le n} x_i + C_{j-1}$

ongi generaatori väljundbitt taktis $j$. Uus ülekanne $C_j$ saadakse eelmist ülekannet ühe biti võrra paremale nihutades, st $C_j = \lfloor C_{j-1}/2 \rfloor$.

### *9.4.8.2 Mittelineaarne filter*

Teine üldine meetod lineaarsuse vältimiseks on mittelineaarse filtri kasutamine (vt Joonis 64) ühe lineaarse nihkeregistri väljundi teisendamiseks.




**Joonis 64. Mittelineaarse filtriga generaator**

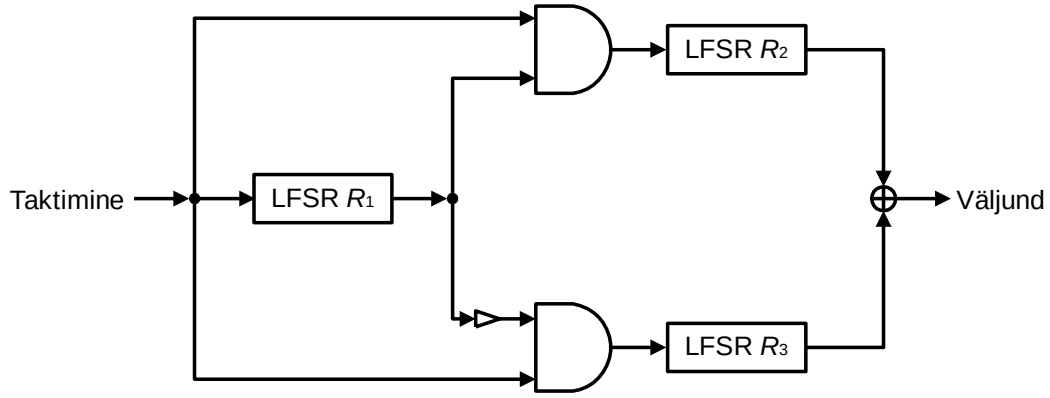
On teada, et kui filtreeriva funktsiooni $f$ mittelineaarsuse järk on $m$, siis generaatori väljundjada lineaarse keerukuse ülempiir on

$\sum_{1 \le i \le m} C_L^i$ ,

kus $C_L^i$ on Newtoni binoomkordajad. On ka näidatud, et enamik maksimaalse perioodiga lineaarseid generaatoreid, juhul kui bittide arv $L$ on piisavalt suur, annab sedasama funktsiooni $f$ filtrina kasutades lineaarse keerukuse, mis on antud ülempiirile väga lähedal.

#### 9.4.8.3 Taktimise juhtimine

Nii mittelineaarse kombineerimisega generaatorites kui ka mittelineaarse filtriga generaatorites eeldati, et komponent-nihkeregistreid taktitakse regulaarselt, st neid juhitakse ühest ja samast taktimpulsside allikast. Juhitava taktimisega generaatorite põhiidee on kasutada ühe lineaarse nihkeregistri väljundit teise registri taktimiseks, lootes, et mitteregulaarne taktimine raskendab potentsiaalseid ründeid.




**Joonis 65. Vahelduva sammuga generaator**

**Näide.** Vahelduva sammuga generaatoris (Joonis 65) kasutatakse kolme lineaarset nihkeregistrit. Esimene register $R_1$ juhib kahe ülejäänud registri $R_2$ ja $R_3$ tööd järgmisel viisil.

- Kui registri $R_1$ väljundi väärtus enne mingit taktimist on 1, siis taktitakse koos registri $R_1$ taktimisega ka registrit $R_2$. Register $R_3$ jääb esialgsesse seisu.
- Kui registri $R_1$ väljundi väärtus enne mingit taktimist on 0, siis taktitakse koos registri $R_1$ taktimisega ka registrit $R_3$. Register $R_2$ jääb esialgsesse seisu.

Generaatori väljundiks on alati registrite $R_2$ ja $R_3$ väljundite summa mod 2. Eeldame, et register $R_1$ pikkusega $L_1$ genereerib de Bruijni jada perioodiga $2^{L1}$, registrid $R_2$ ja $R_3$ on lineaarsed nihkeregistrid pikkustega vastavalt $L_2$ ja $L_3$ (SÜT($L_2$,$L_3$)=1) ning perioodidega vastavalt $2^{L2}-1$ ja $2^{L3}-1$. Sellise juhu kohta on teada järgmised faktid:

- generaatori väljundjada periood on $2^{L1}\cdot(2^{L2}-1)\cdot(2^{L3}-1)$;
- generaatori väljundjada lineaarne keerukus $L$ rahuldab võrratusi

$$(L_2 + L_3)\cdot 2^{L1-1} < L \leq (L_2 + L_3)\cdot 2^{L1}.$$

Generaatori väljundi statistilised omadused on küllalt lähedased juhusliku jada vastavatele omadustele. Generaator on kõige turvalisem siis, kui kõikide komponentide pikkused on paarikaupa ühistegurita ja need pikkused on kõik enam-vähem ühesuurused, st ligikaudu võrdsed $L_1$-ga. Parim teadaolev rünne nõuab $2^{L1}$ sammu, st kui võtta $L_1$=128, on kõik teadaolevad ründed ebaefektiivsed.

## 9.5 Räsifunktsioonid

Räsifunktsioonid (*hash functions*) on plokk- ja jadašifrite kõrval üks tähtsamaid krüptograafiliste primitiivide klasse. Räsifunktsioonid on mõeldud pikast sõnumist püsipikkusega lühikese bitijada (nn sõnumilühendi, *message digest*) tekitamiseks, nii et pika sõnumi ükskõik millised muutused põhjustavad suure tõenäosusega muudatuse ka vastavas sõnumilühendis. Ainult nendel eeldustel koostatud räsifunktsioone saab kasutada näiteks veaavastuskoodides. Krüptograafilistes rakendustes neist eeldustest ei piisa, mistõttu räsifunktsioonidelt nõutakse mitmeid lisaomadusi.

Räsifunktsiooni üks peamisi kasutusalasid on digitaalsignatuuri moodustamine. Avaliku võtmega krüptosüsteemidel põhinevad signeerimisfunktsioonid on tavaliselt aeglased ja seega ebaotstarbekad pikkade sõnumite signeerimiseks. Seetõttu moodustatakse tavaliselt enne signatuuri moodustamist sõnumist $X$ sõnumilühend $H(X)$, mis seejärel signeeritakse. Kui ründajal õnnestub moodustada mingi teine sõnum $X'$, nii et $H(X')=H(X)$, siis tähendab see tegelikult, et ründaja suudab modifitseerida signeeritud sõnumeid ja seega ilma vastavat avaliku võtmega süsteemi murdmata tekitada digitaalsignatuure uutele sõnumitele, mida õige signeerija tegelikult signeerinud ei ole. Seetõttu on räsifunktsioonidele esitatavail turvanõuetel iseseisev tähtsus, mis teebki räsifunktsioonidest iseseisva krüptograafiliste primitiivide klassi.

### 9.5.1 Liigitus ja põhiomadused

Räsifunktsioonid jagunevad kahte suurde klassi:

- võtmeta räsifunktsioonid (sõnumilühendikoodid, *message digest codes*, MDC), mille väljund (sõnumilühend) sõltub ainult esialgsest pikast sõnumist;
- võtmega räsifunktsioonid (sõnumiautentimiskoodid, *message authentication codes*, MAC), mille väljund sõltub esialgsest sõnumist ja salajasest võtmest.

Üldiselt nimetatakse räsifunktsiooniks funktsiooni $H$, millel on järgmised kaks omadust:

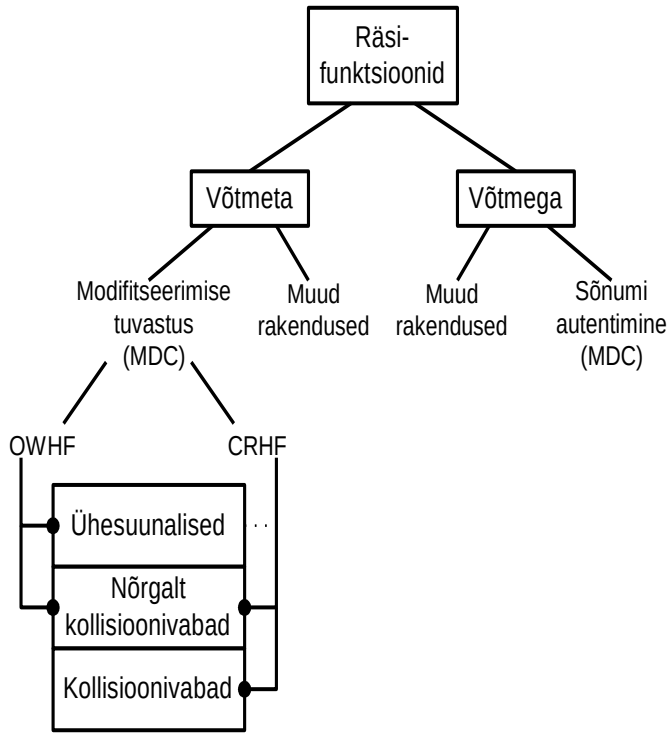
- andmetihendus – $H$ kujutab sisendi $X$ fikseeritud pikkusega ($n$-bitiseks) väljundiks $H(X)$;
- kujutise leidmise lihtsus – funktsiooni $H$ väärtust on "kerge leida" mistahes sisendi $X$ korral.

Lisaks neile kahele omadusele nõutakse krüptograafilistelt räsifunktsioonidelt ka mitmeid muid omadusi (määratlused on alljärgnevas lihtsustatud).

- **Ühesuunalisus** – ei leidu polünomiaalses ajas töötavat algoritmi, mis juhuslikult valitud kujutise $Y$ korral leiaks küllalt suure (suurema kui mingi etteantud väike suurus $\varepsilon$ või suurem kui polünomiaalse kiirusega nullile lähenev lõpmata väike suurus) tõenäosusega vähemalt ühe originaali $X$, nii et $Y=H(X)$.
- **Nõrk kollisioonivabadus** – ei leidu polünomiaalses ajas töötavat algoritmi, mis juhuslikult valitud originaali $X$ korral leiaks küllalt suure tõenäosusega teise originaali $X'\neq X$, nii et $H(X')=H(X)$.
- **Kollisioonivabadus** – ei leidu polünomiaalses ajas töötavat algoritmi (sh stohhastilist), mis leiaks küllalt suure tõenäosusega originaalide paari $(X',X)$, nii et $X'\neq X$ ja $H(X')=H(X)$.

Ühesuunaliseks räsifunktsiooniks (*one-way hash function*, OWHF) nimetatakse räsifunktsiooni, mis on ühesuunaline ja nõrgalt kollisioonivaba.

Kollisioonivabaks räsifunktsiooniks (*collision resistant hash function*, CRHF) nimetatakse räsifunktsiooni, mis on kollisioonivaba ja nõrgalt kollisioonivaba.




**Joonis 66. Räsifunktsioonide liigitus ja kasutusalad**

**Näide 1**. Olgu sisendjada $X$ esitatud 32-bitiste plokkide jadana $X=(X_1,X_2,...,X_n)$. Defineerime funktsiooni $H$ järgmiselt:

$$H(X) := (X_1 + X_2 + ... + X_n) \bmod 2^{32},$$

kus "+" tähistab tavalist liitmist. See funktsioon on küll lihtsasti arvutatav ja andmetihenduse omadusega, kuid pole ühesuunaline, nõrgalt kollisioonivaba ega ka kollisioonivaba.

**Näide 2**. Olgu $p$ mingi suur algarv ja olgu $\alpha$ arv vahemikus $1<\alpha<p$, nii et astmed $\alpha^1,...,\alpha^{p-1}$ on paarikaupa erinevad mod $p$. Olgu $X$ suvaline positiivne täisarv ja

$$H(X) = \alpha^X \bmod p.$$

See funktsioon on kergesti arvutatav, sest esitades $m$-bitise arvu $X$ kahendkujul $X=x(0)+2^1x(1)+ ... +2^{m-1}x(m-1)$ , kus $x(i)\in\{0,1\}$ saame:

$$H(X) = H(2^0)^{x(0)} \cdot H(2^1)^{x(1)} \cdot H(2^2)^{x(2)} \cdot ... \cdot H(2^{m-1})^{x(m-1)} \bmod p,$$

kusjuures suurused $H(2^k)$ saab arvutada rekursiivselt: $H(2^{k+1}) = H(2^k)\cdot H(2^k) \bmod p$. Funktsioon arvatakse olevat ühesuunaline, kuid ta pole nõrgalt kollisioonivaba ega ka kollisioonivaba, sest mistahes $X'$, nii et $(X'-X) \bmod (p-1) = 0$, rahuldab ka võrrandit $H(X')=H(X)$.

Sõnumiautentimiskood e MAC-kood (*message authentication code*, MAC) on funktsioonide pere $H_K$, mille parameeter $K$ on salajane võti ja millel on järgmised omadused:

- arvutamise lihtsus – parameetri $K$ suvalise etteantud väärtuse ja suvalise etteantud sõnumi $X$ korral on kerge arvutada kujutist $H_K(X)$;
- tihendus – funktsioon $H_K$ kujutab mistahes pikkusega originaali $X$ fikseeritud pikkusega kujutiseks;
- arvutuse keerukus – teadmata võtit $K$ on "arvutuslikult raske" etteantud paaride $(X_i,H_K(X_i))$ põhjal leida uusi paare $(X,H_K(X))$.

Viimane tingimus on oluline võltsingute ärahoidmiseks. Ründajal peab olema raske modifitseerida saadetud sõnumit koos lühendiga $(X,H_K(X))$, nii et vastuvõtja seda ei tuvastaks pärast sõnumilühendi kontrollimist.

**Märkus.** Sõnumiautentimiskoodi definitsioonist järeldub, et ründajal peab olema "arvutuslikult raske" kindlaks teha salajast võtit, kui tal on teada üks või mitu sõnum-sõnumilühend-paari. Definitsioonist ei järeldu aga, et fikseeritud $K$ ja kõigile teada oleva $K$ korral peaks funktsioon $H_K$ olema ühesuunaline või kollisioonivaba.

### 9.5.2 Üldründed sõnumiautentimiskoodidele

Ründaja eesmärk on leida uus sõnum-sõnumilühend-paar $(X,H_K(X))$ eeldusel, et on teada üks või mitu sõnum-sõnumilühend-paari $(X_i,H_K(X_i))$. Ründed jaotatakse vastavalt ründaja käsutuses olevate lähteandmete ja võimaluste järgi kolme klassi, mis on analoogilised plokkšifrite vastavate rünnete klassidega:

- teadaolev sõnum – ründajal on teada üks või mitu sõnum-MAC-paari $(X_i,H_K(X_i))$;
- valitav sõnum – ründajal on võimalus ise valida piiratud hulk sõnumeid $X_i$ ja saada nendele vastavad sõnumilühendid;
- adaptiivselt valitav sõnum – ründajal on võimalus valida sõnum $X_i$ sõltuvalt eelnevate sõnumite sõnumilühenditest.

Üldiselt nõutakse sõnumiautentimiskoodidelt vastupidavust adaptiivsetele valitava sõnumiga rünnetele sõltumata sellest, kas konkreetsetes rakendustes on seesugused ründed praktilised või mitte.

### 9.5.3 Sünnipäevaparadoks

seisneb selles, et igas juhuslikult moodustatud 23-liikmelises inimeste grupis on kaks samal päeval sündinut tõenäosusega vähemalt ½. Sellel paradoksil on oluline roll räsifunktsioonide ja ka plokkšifrite tugevuse hindamisel. Olgu $H$ mingi räsifunktsioon, mille väljund on $n$-bitine, st kõikvõimalike väljundite arv on $N=2^n$. Valime juhuslikult $k$ sisendit $X_1,X_2,\ldots,X_k$ ja arvutame tõenäosuse, et $H(X_i)=H(X_j)$ mingi indeksite paari $i,j\le k$ korral. Kujutleme, et sisendid valiti indeksite kasvavas järjekorras. Tõenäosus, et $H(X_1)\neq H(X_2)$, on $(1-1/N)$. Tõenäosus, et $H(X_1)$, $H(X_2)$ ja $H(X_3)$ on kõik erinevad, *on* $(1-1/N)(1-2/N)$ jne. Lõpuks saame tõenäosuse $P$, millega kõik $k$ kujutist on erinevad:

$$P = (1-1/N)(1-2/N)\cdot \ \ldots \cdot (1-(k-1)/N).$$

Kui $x$ on piisavalt väike suurus, siis $1-x \approx e^{-x} = \exp(-x)$. Seega, kui $n$ on piisavalt suur, saame otsitavaks tõenäosuseks

$$P \approx \exp(-k(k-1)/2N).$$

Tõenäosus, et kaks kujutist on võrdsed, on $1-P$. Siit tuleneb, et n-bitise väljundiga räsifunktsioonile kollisiooni leidmiseks tõenäosusega ½ on vaja umbes

$$k \approx 1.17\cdot\sqrt{N} = 1.17\cdot 2^{n/2}$$

juhuslikku avateksti. Näiteks kui räsifunktsiooni väljund on 128-bitine, on kollisiooni leidmiseks vajalik umbes $2^{64}$ avateksti. Tänapäeval hakkab kollisioonivabaduse nõudega räsifunktsioonidel väljundi pikkus 128 bitti väikeseks jääma, seetõttu ei konstrueerita enam räsifunktsioone väiksema väljundiga kui 160 bitti.

### 9.5.4 Omaduste vahelised seosed

Kollisioonivabadusest järeldub nõrk kollisioonivabadus, sest kui *H* ei ole nõrgalt kollisioonivaba, peaks leiduma polünomiaalses ajas töötav algoritm ***A***, mis juhuslikult valitud bitistringi *X* korral leiaks "küllalt suure" tõenäosusega teise bitistringi *X'*, nii et $X' \neq X$ ja $H(X')=H(X)$. Algoritmi ***A*** abil saab aga koostada uue algoritmi, mis töötab järgmiselt:

1) genereerida juhuslik bitistring *X*;
2) leida algoritmi ***A*** abil bitistring *X'*, nii et $X' \neq X$ ja $H(X')=H(X)$;
3) väljastada paar $(X',X)$.

On selge, et vaadeldud algoritm on samuti polünomiaalses ajas töötav ja väljastab suure tõenäosusega räsifunktsiooni kollisiooni, st paari $(X',X)$, nii et $X' \neq X$ ja $H(X')=H(X)$. Sellise algoritmi olemasolu oleks aga vastuolus kollisioonivabadusega.

Ühesuunalisusest ei järeldu nõrk kollisioonivabadus ega ka vastupidi. Samuti ei järeldu kollisioonivabadusest ühesuunalisus. Viimase väite selgitamiseks oletame, et *G* on kollisioonivaba räsifunktsioon, mis kujutab suvalise pikkusega sisendi püsiva pikkusega väljundiks. Moodustame uue räsifunktsiooni *H* järgmiselt.

$$H(X) = \begin{cases} 1 \| X & \text{kui } X \text{ on } n\text{-bitine;} \\ 0 \| G(X) & \text{vastasel juhul} \end{cases}$$

*H* on $n+1$-bitise väljundiga räsifunktsioon, mis on kollisioonivaba, kuid mitte ühesuunaline, sest juhuslikult valitud $n+1$-elemendilises bitistringis *Y* on tõenäosusega ½ esimene bitt võrdne 1, mille korral on funktsioon *H* kergesti pööratav. Kollisioone aga leida ei saa, sest kui $Y=H(X)=H(X')$, siis funktsiooni *H* definitsioonist tulenevalt peab väljundi *Y* esimene bitt olema 0, mistõttu kollisiooni leidmine on ekvivalentne funktsioonile *G* kollisiooni leidmisega.

### 9.5.5 Chaum-van Heijst-Pfitzmanni räsifunktsioon

On olemas räsifunktsioone, mille murdmine on tõestatavalt samaväärne mõne keerulise kombinatoorikaülesande lahendamisega. Sellise tõestuse olemasolu mingi konkreetse räsifunktsiooni kohta suurendab tunduvalt veendumust selle räsifunktsiooni turvalisuses, sest räsifunktsiooni murdmine tähendaks ka olulist teaduslikku läbimurret kombinatoorikas.

Olgu *p* suur algarv, kusjuures $(p-1)/2$ olgu samuti algarv; $\alpha$ ja $\beta$ olgu primitiivsed elemendid mod *p*, st vähimad astmed, mis langevad kokku mõne väiksema astmega mod *p*, on astmed $\alpha^{p-1} \bmod p$ ja $\beta^{p-1} \bmod p$. Defineerime funktsiooni

$$H(x,y) := \alpha^x \cdot \beta^y \bmod p,$$

mille sisendid on arvud vahemikus 0 ja $p-1$ ning väljund on vahemikus 1 kuni *p*. Funktsioon *H* tihendab seega andmeid kahekordselt, st teisendab $2n$-bitised jadad *n*-bitisteks jadadeks. Funktsiooni *H* argumendid on küll püsiva pikkusega, aga kui esitada suvalise pikkusega jada *X* *n*-bitiste plokkidena $(x_1,x_2,\ldots,x_k,\ldots)$, on võimalik arvutada sõnumilühend järgmise rekurrentse skeemi järgi:

$Y_1 = x_1$
$Y_k = H(x_k, Y_{k-1})$

Võib tõestada, et funktsioonile *H* kollisiooni leidmine on samaväärne diskreetse logaritmi probleemi lahendamisega. Argumendipaaride $(x_1,y_1)$ ja $(x_2,y_2)$ leidmine, mille korral $H(x_1,y_1)=H(x_2,y_2)$, on samaväärne võrrandi $\alpha^x \bmod p = \beta$ lahendamisega, st diskreetse logaritmi leidmisega.

Ehkki Chaum-van Heijst-Pfitzmanni räsifunktsioon on tõestatavalt kollisioonivaba, ei saa teda paljudes tegelikes rakendustes kasutada, sest tema arvutamiseks vajalikud ressursid on liiga suured. Järgnevas alajaotises esitame praktikas kasutatava räsifunktsiooni näitena funktsiooni SHA-1.

### *SHA-1*

Räsifunktsioon SHA-1 (*Secure Hash Algorithm*) konstrueeriti vanema räsifunktsiooni MD4 põhjal, mis murti (leiti efetiivne algoritm kollisioonide leidmiseks) 1996. aastal. Funktsioon SHA-1 usutakse olevat piisavalt turvaline, sest seni teadaolevad universaalsed ründemeetodid ei võimalda kollisioonide leidmist, mis oleks efektiivsem juhuslikust otsingust sünnipäevaparadoksi kasutades (nn sünnipäevaründest).

SISEND: bitistring $x$ pikkusega $b$.
VÄLJUND: 160-bitine sõnumilühend.

1. Konstantide defineerimine

$h_1 :=$ `0x67452301`,
$h_2 :=$ `0xEFCDAB89`,
$h_3 :=$ `0x98BADCFE`,
$h_4 :=$ `0x10325476`,
$h_5 :=$ `0xC3D2E1F0`,
$y_1 :=$ `0x5A827999`,
$y_2 :=$ `0x6ED9EBA1`,
$y_3 :=$ `0x8F1BBCDC`,
$y_4 :=$ `0xCA62C1D6`.

2. Eeltöötlus. Lisada sisendile $x$ täidis, nii et sisend oleks 512 biti kordne. Selleks lisada üks konstantne ühebitt, seejärel $r$-1 nullbitti, kus $r$ on vähim positiivne täisarv, mis teeb uue sisendi pikkuseks 64 biti võrra väiksema arvu kui mingi 512 kordne. Lõpuks lisada sisendjada pikkuse 64-bitine $b$ esitus (mod $2^{64}$), kõrgem 32-bitine osa enne madalamat osa. Sisend koosneb $16m$ 32-bitisest sõnast $x_0x_1...x_{16m-1}$. Algväärtustatakse tsüklimuutujad

$$(H_1,H_2,H_3,H_4,H_5) := (h_1,h_2,h_3,h_4,h_5)$$

3. Täita iteratiivselt $i$:=0...$m$-1. Kopeerida $i$-s plokk (kuusteist 32-bitist sõna) ajutisse massiivi $X[j] := x_{16i+j}$, kus $0 \le j \le 15$.

   Laiendada 16-sõnaline plokk $X[]$ 80-sõnaliseks plokiks järgmise rekursiivse valemiga:

$$X[j] := ((X[j\text{-}3] \oplus X[j\text{-}8] \oplus X[j\text{-}14] \oplus X[j\text{-}16]) \text{ rol } 1)$$

   Algväärtustada ajutised muutujad $(A,B,C,D,E) := (H_1,H_2,H_3,H_4,H_5)$.

   1) tsükkel: $j := 0 \ldots 19$:
      $t := (A \text{ rol } 5) + f(B,C,D) + E + X[j] + y_1$,
      $(A,B,C,D,E) := (t, A, B \text{ rol } 30, C, D)$.
   2) tsükkel: $j := 20 \ldots 39$:
      $t := (A \text{ rol } 5) + h(B,C,D) + E + X[j] + y_2$,
      $(A,B,C,D,E) := (t, A, B \text{ rol } 30, C, D)$.
   3) tsükkel: $j := 40 \ldots 59$:
      $t := (A \text{ rol } 5) + g(B,C,D) + E + X[j] + y_3$,
      $(A,B,C,D,E) := (t, A, B \text{ rol } 30, C, D)$.
   4) tsükkel: $j := 60 \ldots 79$:
      $t := (A \text{ rol } 5) + h(B,C,D) + E + X[j] + y_4$,
      $(A,B,C,D,E) := (t, A, B \text{ rol } 30, C, D)$.

Muuta tsüklimuutujaid:

$(H_1,H_2,H_3,H_4,H_5) := (H_1 + A, H_2 + B, H_3 + C, H_4 + D, H_5 + D)$.

4. Väljastada $H_1H_2H_3H_4H_5$.

Funktsioonid *f*, *g* ja *h* on valitud järgmiselt:

$$f(x,y,z) := xy + x^*z$$
$$g(x,y,z) := xy + xz + yz$$
$$h(x,y,z) := x \oplus y \oplus z,$$

kus "+" tähendab bitthaaval loogilise disjunktsiooni operatsiooni, "*" tähendab bitthaaval inversiooni, korrutamine *xy* bitthaaval loogilist konjunktsiooni ja $\oplus$ bitthaaval XOR-tehet.

## 9.6 Avaliku võtmega krüptosüsteemid

Avaliku võtmega krüptosüsteemide peamine iseärasus on selles, et igal kasutajal on kaks isiklikku võtit: avalik võti *e* ja privaatvõti *d*, st võtme *e* pöördvõti, kusjuures krüptosüsteem peab rahuldama nõuet, et ründajal on avalikku võtit teades võimatu või väga raske kindlaks teha vastava privaatvõtme tegelikku väärtust. Avaliku võtmega krüptosüsteemide kaks olulist omadust on järgmised:

- nad võimaldavad taandada konfidentsiaalse ja autentse võtmevahetuskanali nõude üksnes autentse kanali nõudele;
- nad võimaldavad digitaalsignatuure, mis on oluline eeldus juriidiliselt kehtivate elektrooniliste dokumentide kasutuselevõtuks.

Käesolevas osas kirjeldatakse kõigepealt üht vanimat, enamkasutatavamat ja turvalisemat avaliku võtmega krüptosüsteemi nimega RSA. Seejärel vaadeldakse üht uusimat ja ka praktilist krüptosüsteemi, mille kohta on tõestatud, et ta on turvaline adaptiivse valitava avatekstiga ründe suhtes, eeldustel, et teatav räsifunktsioon on ühesuunaline ja et Diffie-Hellmanni probleem on arvutuslikult raske.

### 9.6.1 RSA

Krüptosüsteemi RSA leiutasid 70te teisel poolel R. Rivest, A. Shamir ja L. Adleman (populaarartikkel ilmus 1977. a ajakirjas *Scientific American,* teadusajakirjas alles 1978. a algul).

Krüptosüsteemi RSA saab kasutada andmete krüpteerimiseks ja digitaalsignatuuride moodustamiseks. RSA turvalisus põhineb arvu algteguriteks lahutamise keerukusel.

RSA põhitehe on modulaareksponendi arvutamine kordarvulise mooduli – moodul on kahe algarvu korrutis – järgi. Iga mooduliga on seotud kaks astendajat: *e* (avalik astendaja) ja *d* (salajane astendaja). RSA tehete mõnedes teostustes kasutatakse kaht salajast astendajat $d_1$ ja $d_2$.

Kasutaja genereerib kaks algarvu *p* ja *q* (näiteks kasutades tõenäosuslikke algarvuteste) selliselt, et SÜT(*e*,*p*-1)=1 ja SÜT(*e*,*q*-1)=1. Salajase astendaja leiab seosest

$$e \cdot d \equiv 1 \pmod{\phi(p \cdot q)},$$

näiteks kasutades laiendatud Eukleidese algoritmi (vt alajaotis 9.6.2, "Laiendatud Eukleidese algoritm").

$\phi$ on arvuteoorias kasutatav funktsioon, mis on tuntud ka Euleri $\phi$-funktsiooni nime all. $\phi(n)$ võrdub arvust *n* väiksemate positiivsete arvude arvuga, mis on arvuga *n* ühistegurita, näiteks on arvust 6 väiksemad positiivsed arvud 1 ja 5, seega $\phi(6) = 2$. Euleri funktsioonil on järgmised omadused:

1) kui *p* on algarv, siis $\phi(p) = p - 1$;
2) kui arvud *m* ja *n* on ühistegurita, siis $\phi(m \cdot n) = \phi(m) \cdot \phi(n)$.

Arvupaar (*e*,*N*=*p*·*q*) avalikustatakse. Avalikustamine võib toimuda näiteks arvupaari (*e*,*N*=*p*·*q*) lisamisena süsteemi avalike võtmete kataloogi, avaliku võtme sertifitseerimisena sertifitseerimiskeskuses jne. Praktikas on avaliku astendaja *e* levinud väärtused 3, 5 või $2^{16}+1$. Põhjuseks on astendamise efektiivsus nende väärtuste korral (algoritm "tõsta ruutu ja korruta" osutub optimaalseks).

Tehet $P^e \pmod N$, kus P on avatekst, kasutatakse krüpteerimiseks ja $P^d \pmod N$ kasutatakse signeerimiseks. Krüptogrammi $C \equiv P^e \pmod N$ dešifreerimiseks arvutatakse $C^d \pmod N$. Astendamist tehakse plokikaupa, st. astendatav jaotatakse ($\lfloor \log N \rfloor - 1$)-bitisteks plokkideks (kui viimane plokk tuleb lühem, kasutatakse ploki täitmiseks täidistust).

**Näide 1**. Vaatame näidet, kus Alice saadab Bobile krüpteeritud ja signeeritud sõnumi. Eeldame, et Alice ja Bob on süsteemi uued kasutajad, st neil puuduvad vajalikud võtmed.

1. Alice genereerib algarvud *p* ja *q*, arvutab võtmed *e*, *d* ja $N=p\cdot q$ ning avalikustab oma avaliku võtme (*e*, *N*).
2. Bob genereerib algarvud *p'* ja *q'*, arvutab võtmed *e'*, *d'* ja $N'=p'\cdot q'$ ning avalikustab oma avaliku võtme (*e'*, *N'*).
3. Alice võtab Bobi avaliku võtme ja arvutab $C \equiv (P^d \bmod N)^{e'} \bmod N'$. Krüptogrammi *C* saadab ta Bobile.
4. Bob võtab Alice'i avaliku võtme ja arvutab $P \equiv (C^{d'} \bmod N')^e \bmod N$.

Ainult Bob teab oma salajast astendajat *d'*, seetõttu on tema ainus, kes saab krüptogrammi dešifreerida. Veel teab Bob, et saatja on Alice, sest keegi teine ei tea Alice'i salajast võtit (sõnum signeeriti Alice'i salajase võtmega).

### 9.6.2 Laiendatud Eukleidese algoritm

ALGORITM: Laiendatud Eukleidese algoritm
SISEND: positiivsed täisarvud *m* ja *n*, kusjuures $m \geq n$.
VÄLJUND: *d*=SÜT(*m*,*n*) ning täisarvud *k* ja *l*, mis rahuldavad tingimust $m\cdot k + n\cdot l = d$.

1 Kui *n=0*, siis algväärtustada *d* := *m*, *k*:=1 ja *l*:=0 ning väljastada tulemus (*d*,*k*,*l*).
2 Algväärtustada $k_2 := 1,\ k_1 := 0,\ l_2 := 0,\ l_1 := 1.$
3 Seni kui *n*>0, arvutada:
  3.1 $q := \lfloor m/n \rfloor,\ r := m - q\cdot n,\ k := k_2 - q\cdot k_1,\ l := l_2 - q\cdot l_1.$
  3.2 $m := n,\ b := r,\ k_2 := k_1,\ k_1 := k,\ l_2 := l_1, l_1 := l.$
4 $d := m,\ k := k_2,\ l := l_2$ ja väljasta (*d*, *k*, *l*).

### 9.6.3 Hiina jäägiteoreem ja selle kasutamine RSA tehetes

Alustuseks teoreem, mida tunti juba 22 sajandit tagasi Hiinas ning seetõttu kannab ta *hiina jäägiteoreemi* nime.

**Hiina jäägiteoreem.** *Kui m ja n on ühistegurita täisarvud, siis kongruentsid* $x \equiv x_m \pmod m$ *ja* $x \equiv x_n \pmod n$ *on samaaegselt lahenduvad, kusjuures lahend on ühene mooduli m·n järgi.*

RSA korral on algarvud *p* ja *q* ühistegurita, seetõttu on vastavad kongruentsid alati lahenduvad. Veel enam, kuna tehteid tehakse mooduli *m·n* järgi, on lahend ühene. Hiina jäägiteoreemi kasutamine RSA tehetes tähendab modulaareksponendi mod *N* asendamist modulaareksponentidega mod *p* ja mod *q*. Modulaareksponendi leidmine on kõige rohkem arvutusressurssi nõudev RSA tehe; hiina jäägiteoreem võimaldab vähendada nii astendajat kui ka moodulit. Mooduli muutmine viib ka salajase astendaja *d* asendamisele astendajatega $d_1$ ja $d_2$

$$d_1 \equiv d \pmod{p\text{-}1},$$
$$d_2 \equiv d \pmod{q\text{-}1}.$$

Enamasti on *e* väiksem arvudest *p*-1 ja *q*-1, seetõttu ei ole vaja astendajat *e* asendada astendajatega $e_1 \equiv e \pmod{p\text{-}1}$ ja $e_2 \equiv e \pmod{q\text{-}1}$, vastasel korral on *e* asendamine otstarbekas.

Astendamine $P^d \pmod N$ toimub hiina jäägiteoreemi kasutades kahes osas:

1) arvutatakse $a \equiv \mathrm{P}^{d_1} \pmod{p}$ ja $b \equiv \mathrm{P}^{d_2} \pmod{q}$;

2) $\mathrm{P}^d \pmod{N}$ leitakse valemiga (hiina jäägiteoreem)

$(((a - (b \bmod p)) \cdot u) \bmod p) \cdot q + b$, kui $a \geq (b \bmod p)$,

$(((a + p - (b \bmod p)) \cdot u) \bmod p) \cdot q + b$, kui $a < (b \bmod p)$,

kus $u \cdot q \equiv 1 \pmod{p}$. Arvu *u* saab leida laiendatud Eukleidese algoritmi abil (vt alajaotis 9.6.2 ,"Laiendatud Eukleidese algoritm").

### 9.6.4 Tõenäosuslikud algarvutestid

Krüptosüsteemi RSA kasutamisel vajatakse suuri algarve, seetõttu on olulisel kohal nende efektiivne genereerimine. Kõlblike algarvude suurus (rohkem kui sada kümnendkohta) välistab deterministlike testide kasutamise eeskätt oma ajamahukuse tõttu. Erikujuliste algarvude (näiteks Mersenne'i algarvud vms) kasutamine nõrgendaks süsteemi, sest ühelt poolt on fikseeritud kujuga teadaolavaid algarve tavaliselt vähe (hetkel sobiks näiteks Mersenne'i algarvude hulgast arv $2^{521}$-1) ja teisalt on teada palju ründeid, mis on efektiivsed tänu algarvude teatud kujule, isegi kui sellise kujuga algarve on palju. Seetõttu on kõige turvalisem valida algarve juhuslikult, ühtlase jaotusega üle teatud lõigu. Kuna algarvude äratundmine on keerukas kombinatoorikaülesanne, siis praktilistes süsteemides kasutatakse tõenäosuslikku hüpoteeside kontrolli, mille tulemusel tunnistatakse arv kas kordarvuks (negatiivne tulemus) või – teatud usaldatavusega – algarvuks (positiivne tulemus).

Lihtsaim ja levinuim tõenäosuslik algarvukriteerium on Fermat' test, mis põhineb järgmisel teoreemil.

**Fermat' teoreem.** Kui *n* on algarv, siis iga arvu $0<a<n$ korral

$$a^{n-1} \equiv 1 \pmod{n}. \quad (*)$$

Fermat' testis valitakse juhuslikult arv $1<a<n$-1 ja kontrollitakse, kas on täidetud tingimus (*). Kui tingimus ei ole täidetud, on arv *n* kordarv. Positiivse vastuse korral teame usaldatavusega *p*, et arv *n* on algarv.

Teine praktikas laialt levinud tõenäosuslik algarvutest on Miller-Rabini test. See test põhineb arvuteooriast tuntud faktil, et kui *n* on algarv, $n\text{-}1 = 2^r \cdot s$, kus *s* on paaritu arv ja $\mathrm{SÜT}(a, n)=1$, siis $a^s \equiv 1 \pmod{n}$ või $a^{2^j \cdot s} \equiv -1 \pmod{n}$ mingi $0 \leq j \leq r\text{-}1$ korral.

Miller-Rabini testis esitatakse *n*-1 kujul $2^r \cdot s$, kus *s* on paaritu arv. Seejärel valitakse arv $1<a<n$ ja arvutatakse $a^s \bmod n$. Kui $a^s \equiv 1 \pmod{n}$ või $a^s \equiv -1 \pmod{n}$, siis lõpetatakse test ja loetakse test läbituks. Vastasel korral arvutatakse astmed $a^{2s} \pmod{n}$, ..., $a^{2^{r-1} \cdot s} \pmod{n}$. Testi läbimiseks on vaja, et arvutatud astmete hulgas oleks arv -1 (modulo *n*). Testi mitteläbimise korral teame, et arv *n* on kordarv. Testi läbinud arv *n* on usaldatavusega *p* algarv (kui *n* on kordarv ja läbis testi arvu *a* korral, siis öeldakse, et *n* on tugev pseudoalgarv alusel *a*).

ALGORITM: Miller-Rabini tõenäosuslik algarvutest
SISEND: arv *n*
VÄLJUND: "algarv", kui *n* on algarv, või "kordarv", kui *n* on kordarv

1 Esitada arv $n-1$ kujul $2^r \cdot s$.
2 Valida juhuslik arv *a*, $2 \leq a \leq n-2$.
3 Arvutada $y \equiv a^r \pmod{n}$.
4 Kui $y \neq 1$ ja $y \neq n-1$, siis arvutada nii:

4.1 Algväärtustada $j := 1$.

4.2 Seni kui $j \le s-1$ ja $y \ne n-1$ teha:

4.2.1 Arvutada $y := y^2 \pmod n$.

4.2.2 Kui $y = 1$, väljastada otsus "kordarv".

4.2.3 $j := j + 1$.

4.3 Kui $y \ne n-1$, väljastada otsus "kordarv".

5 Väljastada otsus "algarv".

Tõenäosuslike algarvutestide usaldatavust saab suurendada testi kordamisega, muutes arvu *a* väärtust.

Tõenäosuslikud testid kulutavad palju arvutusressursse (eeskätt modulaareksponendi arvutamisele), seetõttu kasutatakse praktikas sageli järgmist lähenemisviisi:

1) kontrollitakse arvu *n* jaguvust esimese *k* algarvuga. Arv *k* leitakse kasutatava tõenäosusliku algarvutesti usaldatavusest ja modulaararitmeetika konkreetse teostuse parameetritest. Kui leiti algarv, millega *n* jagub, siis on arv kordarv. Vastasel korral minnakse sammule (2).
2) Arvu *n* kontrollimine tõenäosuslike algarvutestide abil.

### *9.6.5* Ruutuvõtmise ja korrutamisega astendusmeetod

Meetod on tuntud ka "paremalt vasakule astendusmeetodi" nime all ning ta olemus seisneb astme arvutamises ruutu tõstmise ja korrutamise abil. Astendaja *d* esitatakse kahendkujul $d=a_0 \cdot 2^k + a_1 \cdot 2^{k-1} + ... + a_{k-1} \cdot 2 + a_k$ ning $x^d \pmod n$ arvutatakse valemiga

$$x^d \equiv x^{a_0 \cdot 2^k} \cdot x^{a_1 \cdot 2^{k-1}} \cdots x^{a_{k-1} \cdot 2} \cdot x^{a_k} \pmod n .$$

ALGORITM: Ruutuvõtmise ja korrutamisega astendusalgoritm
SISEND: Astendatav $x \pmod n$ ja aste $0 \le d < n$, mille kahendesitus on
$d=a_0 \cdot 2^k + a_1 \cdot 2^{k-1} + ... + a_{k-1} \cdot 2 + a_k$.
VÄLJUND: $x^d \pmod n$.

1 Algväärtustada $y := 1$. Kui $d = 0$, väljastada *y*.

2 Algväärtustada $z := x$.

3 Kui $d = 1$, siis $y := x$.

*4* *i* väärtusi 1 … *t* kasutades teha:

4.1 $z := z^2 \pmod n$.

4.2 Kui $k_i = 1$, siis $y := y \cdot z \pmod n$.

5 Väljastada *y*.

### 9.6.6 RSA turvalisus ja ründed

Eespool märkisime, et RSA turvalisus põhineb arvu tegurdamise keerukusel. Praktikas tulevad ilmsiks ka muud nõrkused ja puudused, mis sõltuvad RSA kasutusviisist.

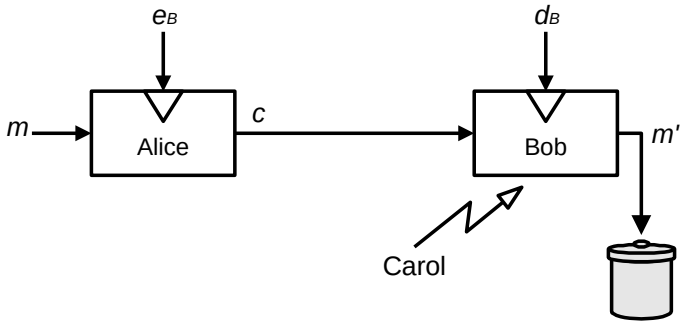
Vaatleme lähemalt RSA nõrkusi ja ründeid. Sõltuvalt eesmärgist võib ründed jagada kolmeks:

a) salajase astendaja *d* leidmine ja seega kogu süsteemi murdmine;
b) *signatuuri võltsimine*, st koostatud sõnumile teise kasutaja signatuuri tekitamine;
c) krüptogrammi dekrüpteerimine, st krüpteeritud sõnumi dekrüpteerimine salajast võtit teadmata.

### *9.6.6.1 Mooduli N tegurdamine*

Kõige primitiivsem ja kõige rohkem arvutusressursse vajav rünne on mooduli *N* tegurdamine ehk algarvude *p* ja *q* leidmine. Leitud algarvudest ja avalikust astendajast *e* on kerge arvutada salajane astendaja *d* (vt jaotis 9.6.1, "RSA krüptosüsteemi tööpõhimõte").

Hiina jäägiteoreemi kasutamine RSA arvutuste kiirendamiseks loob teatud juhtudel võimaluse tegurdada moodul *N* hõlpsamalt kui üldjuhul. Rünne, mida siinkohal kirjeldame, kannab selle leiutaja A. K. Lenstra järgi nime Lenstra rünne. Ründeskeemis eeldatakse, et ründaja Carol saab passiivselt sekkuda Alice'i ja Bobi vahelisse suhtluskanalisse, lugeda Bobi "äravisatud" sõnumeid, st sõnumeid, millel puudub Bobi jaoks mõte, ning Carol saab mõjutada HJT plokki (näiteks mikrolainekiirgusega või ioniseerimisega). Alice saadab Bobile krüptogrammi *c*, mille krüpteerimiseks ta kasutas Bobi avalikku võtit.




**Joonis 67. Lenstra rünne**

Bob dešifreerib krüptogrammi, kasutades RSA tehetes hiina jäägiteoreemi. Carol mõjutab aritmeetikaplokki ajal, mil toimub astendamine mooduli *q* järgi, mis põhjustab väära tulemuse väljundis $m'_q$. Hiina jäägiteoreemi abil leitud sõnumil $m' = \mathrm{HJT}(m_p, m'_q)$ ei ole Bobi jaoks mõtet ja Bob viskab sõnumi *m'* ära. Carol saab sellest sõnumist leida arvu *N* tegurid, sest $m' \equiv m_p \pmod{p}$ ja $m = m_p \pmod{p}$, järelikult ka $m'^e \equiv m^e \equiv c \pmod{p}$. Viimasest seosest saab leida teguri *p* suurima ühisteguri abil järgmiselt: $p = \mathrm{SÜT}(m'^e - c, N)$ ja $q = N/p$.

Lenstra ründe vastu pakkus A. Shamir järgmise lahenduse. Bob valib väikese juhusliku arvu *r*, mis on mooduliga *N* ühistegurita. Edasi arvutab ta hiina jäägiteoreemi kasutamise skeemi kohaselt $m_{rp} \equiv c^{d \pmod{\phi(rp)}} \pmod{rp}$ ja $m_{rq} \equiv c^{d \pmod{\phi(rq)}} \pmod{rq}$. Kui $m_{rp} \equiv m_{rq} \pmod{r}$, on astendamised erinevate moodulite järgi õigesti arvutatud ja *m* leitakse hiina jäägiteoreemi abil kongruentsidest $m_{rp} \pmod{p}$ ja $m_{rq} \pmod{q}$.

### *9.6.6.2 Ühise mooduliga RSA*

RSA kasutamise hõlbustamiseks töötati välja nn ühise mooduliga RSA, kus moodul *N* on kõigil kasutajatel ühine, individuaalsed on vaid astendajad $e_i$, $d_i$. G. J. Simmons näitas 1983. aastal, et kui avalikud astendajad on paarikaupa ühistegurita, saab mitmele kasutajale saadetud sama sõnumi *m* leida krüptogrammidest avalike astendajate abil. Vaatleme seda rünnet lähemalt. Olgu sõnumi *m* krüpteerimisel kasutatud avalikud astendajad $e_i$ ja $e_j$, $c_i \equiv m^{e_i} \pmod{N}$ ja $c_j \equiv m^{e_j} \pmod{N}$. Kuna $e_i$ ja $e_j$ on ühistegurita, saame leida täisarvud *r* ja *s* nii, et $r \cdot e_i + s \cdot e_j = 1$ (arvude *r* ja *s* leidmiseks võib kasutada laiendatud Eukleidese algoritmi). Üks arvudest *r* või *s* peab olema negatiivne, sest astendajad on

positiivsed. Eeldame, et *r* on negatiivne, st $r = -1 \cdot |r|$. Kui $c_i$ on mooduliga *N* ühistegurita, saab leida pöördelemendi $c_i^{-1} \pmod N$. Vastasel korral annab $\text{SÜT}(c_i, N)$ moodul mittetriviaalse teguri ning süsteemi saab täielikult murda. Krüptogrammidest $c_i$ ja $c_j$ saab leida sõnumi *m* järgmise seosega

$$\left(c_i^{-1}\right)^{|r|} \cdot c_j^s \equiv \left(m^{e_i}\right)^{-1\cdot|r|} \cdot (m^{e_j})^s \equiv m^{r\cdot e_i + s\cdot e_j} \equiv m \pmod n.$$

Näeme, et juba kahele kasutajale saadetud krüpteeritud sõnumit on võimalik dekrüpteerida.

J. M. DeLaurentis näitas aasta hiljem, et kasutatava mooduli *N* tegurdamisega saab süsteemi täielikult murda. Tegurdamine põhineb ruutjuure *b* otsimisel arvust 1, kusjuures $1 < b < N-1$. Kuna $2 < b+1 < N$, siis $\text{SÜT}(b+1, N)$ võrdub mooduli *N* mittetriviaalse teguriga. Järgnevas käsitleme sobiva ruutjuure leidmist arvust 1. Olgu *e* ja *d* kasutaja avalik ja salajane astendaja, RSA astendajate omaduse tõttu $e \cdot d - 1 = k \cdot \phi(N) = 2^k \cdot A$, kus *A* on paaritu arv. Kui $1 < a < N$ ja $\text{SÜT}(a, N)=1$, siis vähemalt pooltel juhtudel $a^{2^{k-1}\cdot A}$ ei ole kongruentne $\pm 1$ mooduli *N* järgi. Kui $1 < a^{2^{k-1}\cdot A} < N-1$, ongi $a^{2^{k-1}\cdot A}$ otsitav ruutjuur.

Lõpetuseks veendume, et kasutaja *i* saab leida teise kasutaja salajase astendaja, teadmata arvu $\phi(N)$. Eelnevast teame, et kasutaja *j* avalik astendaja $e_j$ on ühistegurita arvuga $\phi(N)$. Arv $n = \dfrac{(e_i \cdot d_i - 1)}{\text{SÜT}(e_i \cdot d_i - 1, e_j)}$ on kasutaja *j* avaliku astendajaga $e_j$ ühistegurita ja on arvu $\phi(N)$ kordne. Laiendatud Eukleidese algoritmi abil saame leida arvud *r* ja *s* nii, et $n \cdot r + s \cdot e_j = 1$. Arv *s* ongi otsitav salajane astendaja.

#### *9.6.6.3 Väikese avaliku astendajaga RSA*

RSA algoritmi tutvustuse juures märkisime, et efektiivsust silmas pidades kasutatakse avaliku astendaja *e* väikesi väärtusi. Süsteem võib olla ehitatud nii, et avalik astendaja on kõigil kasutajatel ühine, erinevad on ainult moodulid ja salajased astendajad. Osutub, et selline süsteem ei taga alati salastust, kui sama sõnum saadetakse mitmele kasutajale. Vaatame rünnet juhul, kui avalik astendaja on 3. Eeldame, et Carolil on teada kolm krüptogrammi $c_1 \equiv m^3 \pmod{n_1}$, $c_2 \equiv m^3 \pmod{n_2}$ ja $c_3 \equiv m^3 \pmod{n_3}$. Sõnumi *m* krüptogramm on täisarv, mis on väiksem moodulitest $n_1, n_2$ ja $n_3$, seetõttu on $m^3$ väiksem moodulite korrutisest $n_1 \cdot n_2 \cdot n_3$ ja sõnumi leidmine taandub kuupjuure arvutamisele. Kui moodulid on ühistegurita, saame $m^3$ leida laiendatud Eukleidese algoritmi abil (vastasel korral saame tegurdada vastava mooduli ja leida dešifreerimiseks vajaliku salajase astendaja).

Niisuguse puuduse kõrvaldamiseks tuleb kasutusele võtta kas suurem avalik astendaja või varieerida saadetavaid sõnumeid, näiteks lisades igale sõnumile ajatempli $m_i = m \| t_i$. Nagu nähtub järgnevast alajaotisest, ei aita viimane meetod alati.

#### 9.6.6.4 Seotud sõnumitega väikese astendajaga RSA

Järgnevalt kirjeldame RSA puudust, millele juhtisid tähelepanu D. Coppersmith, M. Franklin, J. Patarin ja M. Reiter 1996. aastal. Nad leidsid, et kui krüpteeritavad sõnumid on omavahel seotud polünomiaalse seosega, saab *e* väikese väärtuse korral piisavalt paljudest krüptogrammidest $c_1,...,c_k$ leida avatekstid $m_1,...,m_k$.

Vaatame lihtsat näidet, kus sõnumid $m_1$ ja $m_2$ on omavahel seotud affiinse seosega $m_2 = a \cdot m_1 + b$ ja avalik astendaja $e$ on 3. Krüptogrammidest $c_1 \equiv m_1^3 \pmod N$ ja $c_2 \equiv m_2^3 \pmod N$ saab leida sõnumi $m_1$ valemiga

$$m_1 \equiv \frac{b(c_2 + 2a^3 c_1 - b^3)}{a(c_2 - a^3 c_1 + 2b^3)} \pmod N$$

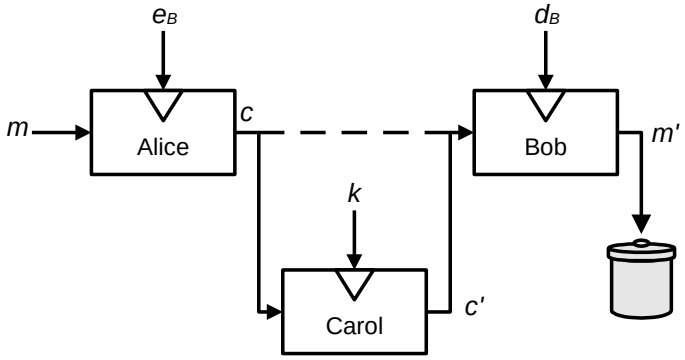
ja avaldada $m_2$ sõnumitevahelisest seosest $m_2 \equiv a \cdot m_1 + b \pmod N$.

Kirjeldatud puudust saab kasutada krüptosüsteemi ründeks, kui ründaja teab sõnumitevahelist seost ja tal on vastavad krüptogrammid. Praktikas osutub kirjeldatud rünne efektiivseks, kui $e$ on väike. Lihtsate sõnumivaheliste seoste korral on rünne efektiivne, kui $e$ on 32-bitine arv.

#### *9.6.6.5 Korrutise astendamine*

Matemaatikast tuntud omadus $(xy)^d = x^d \cdot y^d$ annab võimaluse signatuuri võltsimiseks. Pärast sõnumite $m_1$ ja $m_2$ signeerimist on signeeritud ka sõnumid $m_1 \cdot m_2$, $m_1 \cdot m_1$, $m_2 \cdot m_2$ jne. Selle omaduse kasutamist raskendab asjaolu, et $m_1 \cdot m_2$ ei tarvitse enam olla sõnum, millel on mõte.

Viimast asjaolu kasutab Davida rünne. Davida rünne põhineb eeldustel, et ründaja Carol saab Alice'i ja Bobi vahelises sidekanalis andmeid modifitseerida ja tal on juurdepääs Bobi "äravisatud" sõnumitele, st sõnumitele, millel ei ole Bobi jaoks mingit mõtet. Alice saadab Bobile sõnumi *m*, mis on krüpteeritud Bobi avaliku võtmega.




**Joonis 68. Davida rünne**

Carol valib juhusliku arvu *k* ja asendab krüptogrammi *c* krüptogrammiga $c \cdot k^{e_B}$. Bob dešifreerib saadud krüptogrammi ja saab sõnumi $m \cdot k$, millel ei ole Bobi jaoks mõtet. Bob viskab sõnumi $m \cdot k$ ära, st avalikustab selle Carolile. Carol arvutab $(m \cdot k) \cdot k^{-1} \equiv m \pmod{n_B}$ ja saabki teada, millise sõnumi Alice tahtis Bobile saata.

#### *9.6.6.6 Krüptogrammi üksikute bittide turvalisus*

Eelnevas käsitlesime ründeid, mis võimaldavad sõnumit dekrüpteerida. Käesolevas jaotises esitame tulemused sõnumi osade kohta üldjuhul.

- Krüpteeritava ploki madalaim bitt on *niisama turvaline* kui terve plokk, st krüptgrammist ploki madalaima biti arvutamine on *samaväärne* krüptogrammi dekrüpteerimisega. Leidub

dekrüpteerimisalgoritm, mis kasutab ainult avalikku võtit ja algoritmi, mis leiab krüptogrammist sõnumi madalaima biti.
- Håstad ja Näslund näitasid, et krüpteeritava ploki kõik bitid on *niisama turvalised*, kui terve plokk.
- Täisarvu $k = O(\log\log n)$ korral on *k* madalaimat bitti *niisama turvalised* kui terve plokk.

### 9.6.7 Cramer-Shoupi krüptosüsteem

Aastal 1998 esitasid Ronald Cramer ja Victor Shoup uue krüptosüsteemi, mis on lisaks tõestatud turvalisusele adaptiivse valitava avatekstiga ründe suhtes ka praktikas kasutatav, st nõuab mõistikul hulgal arvutusi.

Kõigepealt on vaja fikseerida üks lõplik rühm *G*, mille järk on piisavalt suur algarv *q*. Rühma elemendid seotakse kõikvõimalike avatekstide hulgaga. Seejärel valitakse üks räsifunktsioon, millel on ühesuunalisuse omadus ja mis kujutab suvalise pikkusega bitistringid rühma $\boldsymbol{Z}q$ elementideks. See rühm koosneb täisarvudest 0 ... *q*-1 ja tema operatsioon on defineeritud kui korrutamine mod *q*.

**Võtme genereerimine**. Valitakse juhuslikult rühma *G* kaks elementi $g_1$ ja $g_2$ ning viis elementi $x1,x2,y1,y2,z$ rühmast $\boldsymbol{Z}q$. Seejärel arvutatakse suurused

$$c = g_1^{x1}\, g_2^{x2},\ \ d = g_1^{y1}\, g_2^{y2},\ \ h = g_1^{z}.$$

Avalik võti on järjend ($g_1$, $g_2$, *c*, *d*, *h*, *H*) ja salajane võti on ($x1,x2,y1,y2,z$).

**Krüpteerimine**. Olgu $m \in G$ suvaline avatekst. Genereeritakse juhuslik arv $r \in \boldsymbol{Z}_q$ ja arvutatakse

$$u_1 = g_1^{r},\quad u_2 = g_2^{r},\quad e = h^{r}\, m,\quad \alpha = H(u_1,u_2,e),\quad v = c^{r}\, d^{r\alpha}.$$

Krüptogramm on nelik $C=(u_1,u_2,e,v)$.

**Dekrüpteerimine**. Saanud krüptogrammi *C*, kontrollib privaatvõtme omanik kõigepealt, kas

$$u_1^{x1+y1\cdot\alpha}\, u_2^{x2+y2\cdot\alpha} = v.$$

Kui vastus on eitav, väljastatakse veateade. Jaatava vastuse korral arvutatakse avatekst valemiga

$$m' = e/u_1^{z} = h^{r}\, m \,/\, u_1^{z} = g_1^{zr}\, m \,/\, g_1^{rz} = m.$$

Cramer ja Shoup tõestasid, et nende skeemi murdmine on samaväärne nn Diffie-Hellmani probleemi lahendamisega: fikseeritud $g \in G$ korral otsustada, kas suvaliselt etteantud kolmik $(a,b,c)$ rühma *G* elementidest on esitatav kujul $(g^x, g^y, g^{xy})$.

### 9.6.8 Elliptilistel kõveratel põhinevad krüptosüsteemid

#### *9.6.8.1 Üldist*

1985 aastal töötasid V. Miller ja N. Koblitz üksteisest sõltumatult välja elliptiliste kõverate kasutamise avaliku võtmega krüptograafias.

Enne elliptilise kõvera mõiste selgitamist toome sisse mõned uued tähised.Tähistagu $Z_p$ täisarve {*0,...,p*-1}, kus $p > 3$on algarv ning tehted – liitmine ja korrutamine – tehakse mooduli *p* järgi. Iga täisarv $0 < a < p$ on mooduliga *p* ühistegurita, seega saame laiendatud Eukleidese algoritmi (vt alajaotis 9.6.2 "Laiendatud Eukleidese algoritm") abil leida arvud *k* ja *l* nii, et $a \cdot k + p \cdot l = 1$ ehk $a \cdot k \equiv 1 \pmod{p}$.

Seega iga nullist erinev element hulgast $Z_p$ on pööratav. Pööramist on meil vaja jagamise defineerimiseks hulgas $Z_p$, $a/b := a \cdot b^{-1}$.

Elliptiliseks kõveraks $E_{a,b}$ üle $Z_p$ nimetame paaride $(x, y)$ hulka, kus paari elemendid on omavahel seotud võrrandiga $y^2 = x^3+ax+b$, kusjuures $a$ ja $b$ on valitud nii, et $4a^3+27b^2$ ei ole kongruentne nulliga mod $p$. Liitmine defineeritakse elliptilise kõvera $E_{a,b}(Z_p)$ punktidel $P=(x_1,y_1)$ ja $Q=(x_2,y_2)$ järgmiselt:

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2,$$

$$y_3 = -y_1 + \frac{y_2 - y_1}{x_2 - x_1} \cdot \left(x_1 - x_3\right),$$

kui $P \neq Q$ ja

$$x_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)^2 - 2x_1,$$

$$y_3 = -y_1 + \frac{3x_1^2 + a}{2y_1} \cdot \left(x_1 - x_3\right),$$

kui $P=Q$.

Et tehte tulemus oleks elliptilise kõvera punkt, lisatakse vaadeldavasse punktide hulka punkt $O$. Punkti $P=(x, y)$ vastandpunktiks nimetame punkti $-P=(x,-y)$. Punktiga $O$ defineeritakse liitmine seostega

$$O+O = O,\ O+P = P,\ P = P+O,\ P+(-P) = O$$

elliptilise kõvera iga punkti $P$ korral.

Näiteks elliptiline kõver $E_{3,5}(Z_{19})$ koosneb paaridest

(0,2),(1,9),(1,11),(1,18),(3,1),(3,8),(3,10),(6,14),(7,6),(8,16),(9,0),(9,4),(9,15),(10,0),
(10,4),(10,15),(11,16),(12,6),(13,14),(16,1),(16,8),(16,10),(18,9),(18,11),(18,18).

Elliptilise kõvera $E_{a,b}$ punktide arvu tähistatakse $\#E_{a,b}$. Punkti $P$ $k$-kordset punkti tähistame $[k]P=P+P+...+P$ ($k$ liidetavat).

Elliptilistel kõveratel põhinevates krüptosüsteemides kasutatakse arvuhulga $Z_p$ asemel arvuhulka $Z_n$, kus $n$ on kahe suure algarvu korrutis. Erinevus on selles, et alati ei ole jagamine defineeritud, näiteks kui $\text{SÜT}(x_2\text{-}x_1,n) > 1$ või $\text{SÜT}(2y_1,n) > 1$. Osutub, et nende juhtude esinemistõenäosus on küllalt väike ja nad jäetakse vaatluse alt välja.

Elliptiliste kõverate kasutamine krüptosüsteemides on tingitud vastava aritmeetika efektiivsest teostusest riistvaras (mistõttu on elliptilistel kõveratel põhinevad krüptosüsteemid leidnud rakendust eeskätt kiipkaartides ja teistes väikese arvutusressursiga seadmetes) ja väiksest võtmepikkusest. Arvestades parimaid teadaolevaid ründeid, loetakse 1024-bitise RSA mooduliga ekvivalentseks elliptilistel kõveratel põhinevat krüptosüsteemi 170-bitise võtmega. Teostuse mõttes eelistatakse elliptilisi kõveraid üle polünoomide, millede kordajad on 0 või 1. Polünoomide korrutamine ja liitmine toimub mooduli $f(x)$ järgi, kus $f(x)$ on nn ***taandumatu polünoom*** *üle* $Z_2$. M. J. Wiener märkis 1998. aastal, et sellise elliptilise kõvera punktide liitmisel ei ole vaja täisarvude korrutamist, mida oleks kulukas teostada riistvaras.

Ka signatuurid on elliptiliste kõverate kasutamise korral palju lühemad kui RSA signatuurid.

### *9.6.8.2 Krüptosüsteem KMOV*

Krüptosüsteemi KMOV (leiutajad K. Koyama, U. M. Maurer, T. Okamoto ja S. A. Vanstone) saab kasutada krüpteerimiseks ja signeerimiseks.

Krüptosüsteemis KMOV käsitletakse ainult elliptilisi kõveraid $y^2=x^3+b$. Iga kasutaja valib algarvud $p$ ja $q$, mis on kongruentsed arvuga 2 mooduli 3 järgi, ja avalikustab nende korrutise $n=pq$. Seejärel valib ta oma avaliku võtme $e$, mis on ühistegurita arvuga $\text{VÜK}(p+1,q+1)$, salajase võtme aga arvutab seosest

$$e \cdot d \equiv 1 \pmod{\text{VÜK}(p+1,q+1)}$$

kasutades laiendatud Eukleidese algoritmi (vt alajaotis 9.6.2, "Laiendatud Eukleidese algoritm").

Et saata sõnum $M=(m_1, m_2)$ Bobile, võtab Alice Bobi avaliku võtme $e$ ja arvutab parameetri $b$ väärtuse $b \equiv m_2^2 - m_1^3 \pmod{n}$. Sõnumit $M$ vaatab Alice kui elliptilise kõvera $E_{0,b}(Z_n)$ punkti, arvutab tema $e$-kordse punkti $C = [e]M = (c_1, c_2)$ ja saadab krüptogrammi $C$ Bobile. Bob arvutab kõvera parameetri $b \equiv c_2^2 - c_1^3 \pmod{n}$ ja dešifreerib krüptogrammi $M=[d]C$, arvutades krüptogrammipunkti $d$-kordse punkti elliptilisel kõveral $E_{0,b}(Z_n)$.

Avaliku võtme $e$ vähim väärtus on 5, sest $\text{VÜK}(p+1,q+1)$ jagub arvuga 6.

## 9.7 Kvantkrüptograafia

### 9.7.1 Sissejuhatuseks

1982. a täheldas Richard Feynman, et teatud kvantmehaanilisi protsesse ei ole klassikalistel arvutitel võimalik efektiivselt simuleerida. Sellest lähtuvalt spekuleeriti ideega, et arvutamine võib muutuda tõhusamaks, kui kasutada kvantmehaanika efekte. Kuigi leiti üksikuid arvutusülesandeid, mille lahendamisega "kvantarvutid" paremini toime tulid, osutus kvantarvutite ehitamine ületamatult raskeks probleemiks. Pööre saabus alles 1994. aastal, mil Peter Shor publitseeris kvantarvutite jaoks kiire faktoriseerimisalgoritmi. Järgnenud mõne aasta jooksul on kvantinformaatika muutunud tormiliselt arenevaks iseseisvaks teaduseks. Kõigele eelnenule lisaks motiveerib kvantinformaatika arengut ka lihtsalt paratamatus: vastavalt Moore'i seadusele kulub 2020. aastal ühe biti salvestamiseks üks aatom; sellisel miniatuursuse tasemel on kvantmehaanika efektide arvestada mitte ainult kasulik, vaid lihtsalt paratamatu. 1930tel ja 1940tel aastatel, veel enne digitaalajastu koidikut, töötati välja mitmed teoreetilised arvutusmudelid (Turingi masin, λ-arvutus, Posti masin), mis kõik suudavad teineteist emuleerida polünomiaalses ajas. Veel enam, ka tänapäeval kasutatavad arvutid on kõik arvutuslikult täpselt niisama võimsad kui oli Turingi masin. Viimase väite üldistust – kõik, mis on arvutatav, on arvutatav kasutades klassikalist Turingi masinat – tuntakse kui Church-Turingi printsiipi.
Muuhulgas võimaldab Church-Turingi printsiip arvutusmudeleid kõrvale jättes *ülesandeid* endid keerukusklassidesse jagada vastavalt sellele, kui efektiivsed algoritmid on nende ülesannete lahendamiseks *põhimõtteliselt* võimalikud (näiteks) Turingi masinal. Oluline keerukusklass on **P** (polünomiaalses ajas lahenduvad ülesanded), millesse kuuluvaid ülesandeid samastatakse tavaliselt "lihtsate" ehk "efektiivselt lahenduvate" ülesannetega.

**Näiteid**

1. Kahe arvu korrutamine on lihtne ülesanne (keerukusklassist **P**).
2. Algarvutest on lihtne ülesanne, kui kasutada *randomiseeritud* Turingi masinat (keerukusklassist **BPP**).
3. Arvu faktoriseerimine on (arvatavasti) raske. Parim teadaolev faktoriseerimisalgoritm (arvkorpuste sõel) on liiga aeglane suurte arvude jaoks. Ülesannete 1, 2 ja 3 suhtelisel keerukusel põhineb krüptosüsteem RSA.
4. Tõestatavalt salajase ühisteadmuse tekkimine kahe osapoole vahele ilma eelnevat ühist salajast informatsiooni omamata on klassikalistes arvutusmudelites *põhimõtteliselt* võimatu.

### 9.7.2 Põhimõisted

#### *9.7.2.1 Kvantolekud, kvantbitt*

Suvalise kvantsüsteemi olekuruum on kirjeldatav lainefunktsioonide Hilberti ruumina. Kvantarvutuste jaoks on vaja vaid lõplikumõõtmelisi vektorruume üle kompleksarvude korpuse. Sellistel olekuruumidel leiduvad ortonormeeritud baasid, mida tähistatakse Dirac'i sulgnotatsiooni kasutades järgmiselt:

1) $|x\rangle$ on veeruvektor; suuruste $|x\rangle$ abil kirjeldatakse tavaliselt kvantolekuid;
2) $\langle x|$ on vektori $|x\rangle$ transponeeritud kaasvektor.

Kahemõõtmelise kompleksarvulise vektorruumi ortonormeeritud baasi tähistatakse kui $\{|0\rangle,|1\rangle\}$, st sellise vektorruumi suvaline element (kvantbitt, *qubit*) $x$ avaldub üheselt baasivektorite $|0\rangle$ ja $|1\rangle$ lineaarkombinatsioonina $a|0\rangle+b|1\rangle$, kus $a^2+b^2=1$. Tähistame sõnaga $\langle x|y\rangle$ vektorite $x$ ja $y$ skalaar- ehk sisekorrutist; maatriksit $|x\rangle\langle y|$ nimetatakse nende vektorite väliskorrutiseks. Selline notatsioon on väga piltlik, muuhulgas võib maatriksit $|0\rangle\langle 1|+|1\rangle\langle 0|$ vaadelda kui lineaarkujutust, mis teisendab baasivektori |0> vektoriks $\langle 1|$, vektori $|1\rangle$ vektoriks $\langle 0|$ ning üldjuhul, vektori $a|0\rangle+b|1\rangle$ vektoriks $a|1\rangle+b|0\rangle$.
Reaalses maailmas vastavad kvantbitile mitmesugused erinevad nn *mõõdetavad*; konkreetsed näited on footoni polarisatsioon ja elektroni spinn. Nii võib kokkuleppeliselt defineerida, et footoni horisontaalne polarisatsioon $|\uparrow\rangle$ vastab kvantbitile $|0\rangle$ ning vertikaalne polarisatsioon $|\rightarrow\rangle$ kvantbitile $|1\rangle$. Seejuures avalduvad footoni diagonaalprojektsioonid kui $|\nearrow\rangle=1/\sqrt{2}(|\uparrow\rangle+|\rightarrow\rangle)$ ning $|\nwarrow\rangle=1/\sqrt{2}(|\uparrow\rangle-|\rightarrow\rangle)$ (kordaja $1/\sqrt{2}$ on vajalik nõude $a^2+b^2=1$ tõttu). Baasvektorite lineaarkombinatsioone nimetatakse nende vektorite *superpositsiooniks* (ehk segaolekuks). Suurust $a$ nimetatakse oleku $|0\rangle$ *amplituudiks* superpositsioonis $a|0\rangle+b|1\rangle$.

#### *9.7.2.2 Mõõtmine*

Kvantmehaanikas käsitletavat maailma ühendab meie maailmaga füüsilise mõõtmise fenomen. Kvantmehaanika aksiomaatika postuleerib, et kvantbiti $a|0\rangle+b|1\rangle$ mõõtmisel piki baasi $\{|0\rangle,|1\rangle\}$ saadakse tulemuseks suurus $|0\rangle$ tõenäosusega $a^2$ või suurus $|1\rangle$ tõenäosusega $b^2$. Sealjuures muutub kvantbiti väärtus pöördumatult (öeldakse, et kvantbitt *kollapseerub*). Vaatamata sellele, et kvantbitil on lõpmatu arv erinevaid väärtusi, saab pärast baasi fikseerimist kvantbitist mõõtmisega kätte vaid ühe biti ja mitte rohkem: esimese mõõtmise tagajärjel kollapseerub kvantbitt, tema uus olek on aga esimese mõõtmise põhjal 100% kindlusega teada.

### 9.7.3 Kvant-võtmevahetusprotokoll

Järgnevalt toome esimese näite kvantmehaanika kasulikkusest krüptograafiale. On hästi teada, et krüpteerimist ei saa klassikalises maailmas läbi viia tingimusteta turvaliselt: ainus tingimusteta turvaline krüptosüsteem on ühekordne šifriplokk, mille kasutamiseks tuleb eelnevalt läbi viia turvaline võtmevahetus. Selleks on tänapäeval olemas kaks põhimõtteliselt erinevat viisi: asümmeetriline krüptograafia (kus võtmevahetuse turvalisus sõltub teatud arvutuslike probleemide oletatavast raskusest) ning füüsiliselt turvalise kanali olemasolu. Klassikalises maailmas ei leidu füüsiliselt turvalisi kanaleid: ükskõik kui salastatult inimesed ka informatsiooni vahetaksid, alati on olemas *põhimõtteline* võimalus nende tegevust jälgida. Kvantmehaanika pakub toodud probleemile lahenduse: me teame, et kvantbitt kollapseerub pärast mõõtmist, seega peaks kvantbiti mõõtmist (ehk lugemist) saama hiljem tuvastada! Analoogilisest arutelust lähtudes töötasid Guilles Brassard ja Charles Bennet 1984. aastal välja tingimusteta turvalise võtmevahetusprotokolli, mis nõuab autentset (kuid mitte tingimata salastatud) klassikalist kanalit ning kvantkanalit, millele ei asetata mingeid täiendavaid turvanõudeid. Alljärgnevas esitame selle protokolli lühikirjelduse.

Protokollil on kaks osapoolt, Alice ja Bob. Alice soovib Bobile edastada *n*-bitise salajase võtme. Selleks genereerib ta *n* suvalist bitti ning saadab need mööda kvantkanalit Bobile, kasutades iga biti kodeerimiseks ühe footoni polarisatsiooni. Seejuures polariseerib Alice iga footoni (juhuslikult) kas baasi $B_0=\{|\uparrow\rangle,|\rightarrow\rangle\}$ või baasi $B_1=\{|\nwarrow\rangle,|\nearrow\rangle\}$ järgi (bitile 0 vastab seega kas $|\uparrow\rangle$ või $|\nwarrow\rangle$, bitile 1 vastab kas $|\rightarrow\rangle$ või $|\nearrow\rangle$). Iga footoni saabumisel valib Bob juhuslikult ühe kahest baasist $B_0$ ja $B_1$ ning mõõdab footonit piki seda baasi, saades tulemuseks ühe neljast toodud vektorist $|\uparrow\rangle$, $|\rightarrow\rangle$, $|\uparrow\rangle$ või $|\rightarrow\rangle$. Juhul kui Bobi ja Alice'i baasid *i*–nda footoni mõõtmisel ühtisid, on Bobi mõõdetud vektor võrdne Alice'i poolt saadetud vektoriga ning seega on Bob võimeline dekodeerima Alice'i poolt edastatud bitti. Kui Bobi ja Alice'i poolt valitud baasid ei ühtunud, saab Bob tõenäosusega 50% õige ning tõenäosusega 50% väära vastuse.

Kui Bob on footonid kätte saanud, avalikustavad Alice ja Bob klassikalise (autentse!) kanali kaudu endi tehtud baasivalikud. Kui *i*–s baas langes kokku, võetakse vastav bitt vahetatud võtmes kasutusele; kui aga ei langenud, heidetakse bitt lihtsalt kõrvale.

Kui Eve mõõdab üht nendest footonitest, peab ta seda tegema fikseeritud baasi $B_1$ või $B_2$ järgi. Kuna tol hetkel Alice pole enda valitud baase veel avalikustanud, ei ole Evel võimalik rakendada paremat strateegiat kui juhuslikku äraarvamist, valides väära baasi tõenäosusega 50%. Väära baasiga mõõtmise korral on aga tulemuseks juhuslik bitt hulgast {0,1}. Veel enam, väära baasi järgi mõõtmine muudab footoni polarisatsiooni ning seega on "nuhitud" footoni polarisatsioon suure tõenäosusega erinev "nuhkimata" footoni polarisatsioonist. Vahetatud võtme kõrge salastustaseme (Evele teatavaks saanud bittide arv on tühine võrreldes edastatud bittide arvuga) saavutamiseks peavad Alice ja Bob rakendama hästi teadaolevaid turvalisuse võimendamise (*privacy amplification*) meetodeid. Üks elementaarsemaid neist on eelnevalt vahetatud $n/2$ bitist juhuslikult $n/4$ biti valimine, avalikustamine ning võrdlemine autentse kanali kaudu. Kui võrdlemisel leitakse, et väärtuse poolest erinevate bittide osakaal valitud $n/4$ biti hulgas on suurem kui mingi läviväärtus $\varepsilon n/4$, loetakse võtmevahetus ebaõnnestunuks. Vastasel korral, kuna $n/4$ bitti olid valitud $n/2$ biti hulgast juhuslikult, võivad Alice ja Bob eeldada, et ülejäänud $n/4$ bitist on vähemalt $\varepsilon n/4$ bitti võrdsed.

**Näide**

Olgu $n=12$. Alice genereerib juhusliku 12-bitise võtme *Võti*=100011010001 ning 12-bitise arvu $Baas_A$=111110101000, mille *i*–s bitt määrab vastava footoni polarisatsioonibaasi:

| Võti | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Baas_A$ | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| Footon | ↗ | ↖ | ↖ | ↖ | ↗ | → | ↖ | → | ↖ | ↑ | ↑ | → |

Pärast footonite kättesaamist valib Bob juhuslikult 12-bitise arvu $Baas_B$=101000101101, mille *i*–s bitt määrab Bobi sooritatava *i*–nda mõõtmise baasi:

| Footon | ↗ | ↖ | ↖ | ↖ | ↗ | → | ↖ | → | ↖ | ↑ | ↑ | → |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Baas_B$ | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| Mõõtmise tulemus | ↗ | ? | ↖ | ? | ? | → | ↖ | → | ↖ | ↑ | ↑ | ? |

(Siin on küsimärgiga tähistatud kvantbitid, mille puhul baasid erinesid.) Pärast footoni olekute dekodeerimist klassikalisteks bittideks ning baaside avalikustamist selgub, et Alice'i ja Bobi ühine võti on 10101000.

### 9.7.4 Kvantarvutid

#### *9.7.4.1 Kvantregistrid*

Siiani vaatasime kvantbittide tasemel toimuvat "arvutamist". Kvantarvutite tegelik potentsiaal ilmneb aga alles siis, kui vaadelda mitmest kvantbitist koosnevaid *kvantregistreid*. Juba 1932. aastal von Neumanni postuleeritud kvantmehaanika aksiomaatikast tuleneb, et *n* kvantbitist moodustatud kvantregister ($|a_1\rangle,|a_2\rangle,\ldots,|a_n\rangle$) (tähistatakse $|a_1\rangle\otimes|a_2\rangle\otimes\ldots\otimes|a_n\rangle$ või $|a_1a_2\ldots a_n\rangle$) on $2^n$-mõõtmelise vektorruumi element, vastandina klassikalisele maailmale, kus *n* bitist moodustatud register $b_1\ldots b_n$ on $2n$-mõõtmelise vektorruumi element. Seda omadust nimetatakse *kvantparallelismiks*. Kvantparallelismi eksistents järeldub faktist, et valdav enamik olekuid $|a_1a_2\ldots a_n\rangle$ ei ole avaldatavad kvantbittide tensorkorrutisena, st enamik kvantolekud on nn *sasiolekud*. Kõikide olekute hulga võimsus on eksponentsiaalne ($2^n$) võrreldes mitte-sasiolekute hulga võimsusega ($2n$), millest tulenebki peatüki alguses mainitud Richard Feynmani tähelepanek, aga ka peamine kvantarvutite potentsiaali põhjus: kvantolekute evolutsiooni kasutamine arvutusmehhanismina.
Kvantregistri mõõtmisel projitseerub olekuruum alamruumi, mis on ühilduv mõõdetava väärtusega. Kui näiteks oleku $1/\sqrt{3}(|000\rangle+|001\rangle+|011\rangle)$ teise kvantbiti mõõtmisel saadakse tulemuseks $|0\rangle$, kollapseerub terve register olekusse $1/\sqrt{2}(|000\rangle+001\rangle)$.

#### *9.7.4.2 Einstein-Rosen-Podelsky paradoks*

Kaks osakest ei ole sasiolekus parajasti siis, kui ühe osakese mõõtmine ei mõjuta teist osakest. Nii näiteks on olek $1/\sqrt{2}(|00\rangle+11\rangle)$ sasiolek: kui esimest osakest mõõta (saades tulemuseks kvantbiti $|x\rangle$, $x$=0 või $x$=1), kollapseerub teine osake olekusse $|x\rangle$. Einstein-Rosen-Podelsky paradoks seisneb selles, et kollapseerumine leiab aset momentaalselt, sõltumata osakeste vahelisest kaugusest. See on tänapäeva füüsika üks kuulsamaid paradokse, millel küll (õnneks või kahjuks) puudub kvantarvutuste juures tähtsus: saab näidata, et mõõdetava informatsiooni vahetamine valgusest kiiremini on siiski võimatu.

Olekus $1/\sqrt{2}(|00\rangle+11\rangle)$ olevaid osakesi nimetatakse *EPR-paariks*. Nagu nähtub hilisemast, on EPR-paaril oluline osa kvantinformatsiooni teisaldamisel.

#### *9.7.4.3 Kvantolekute evolutsioon*

Mittemõõdetud kvantsüsteem areneb vastavalt Scrödingeri võrrandile. Kvantolekud teisenevad, säilitades ortogonaalsust (ehk "nurki"). Vektorruumis üle kompleksarvude korpuse on ortogonaalsust säilitavateks lineaarteisendusteks unitaarteisendused. Kõik lineaarteisendused on esitatavad maatriksitena. Öeldakse, et maatriks *M* on *unitaarne* (vastab unitaarteisendusele), kui $MM^*=1$, kus $M^*$ on maatriksi *M* kaasmaatriksi transponeeritud maatriks. Kvantolekute ruumi unitaarsed teisendused vastavad (üksüheselt)

füüsikaseaduste mõttes lubatavatele kvantarvuti tehetele. Muuhulgas on kõik kvantarvutused pööratavad: rakendades suvalise arvutuse tulemusele sobivat algoritmi, saadakse tagasi algolek.

#### *9.7.4.4 Kvantlülid*

Teatavasti koosnevad klassikalised arvutid elementaarlülidest (AND, NOT, NAND), mis teostavad elementaartehteid ühe või enama bitiga; analoogia põhjal saab ka kvantarvuti puhul defineerida kvantlüli mõiste. Kõige tavalisemad kvantlülid on

1) ID (samasus): ID($a|0\rangle+b|1\rangle$)= $a|0\rangle+b|1\rangle$,
2) *X* (eitus): *X*($a|0\rangle+b|1\rangle$)=a$|1\rangle$+b$|0\rangle$,
3) *Z* (faasinihe): *Z*($a|0\rangle+b|1\rangle$)= $a|0\rangle-b|1\rangle$ ning
4) *Y* (faasinihke eitus): *Y*($a|0\rangle+b|1\rangle$)=b$|0\rangle$−a$|1\rangle$.

Mitmebitistest kvantlülidest on kõige olulisem kahe sisendiga *juhitav eitus* CNOT, mis jätab esimese kvantbiti alati muutmata ning mille toime teisele bitile on võrdne teisenduse *X* toimega, kui esimene bitt on 1, ning teisenduse ID toimega, kui esimene bitt on 0.

Veelgi olulisem lüli on kolmebitise sisendiga *juhitav-juhitav-eitus* (tuntud ka kui Toffoli lüli, *T*), mis rakendab kolmandale bitile teisenduse *X* parajasti siis, kui esimesed kaks kvantbitti võrduvad mõlemad ühega.

Kvantparallelismi rakendamisel on väga oluline lüli ka Hadamardi lüli *H*, mille mõju on määratud võrdustega $H(|0\rangle)=1/\sqrt{2}(|0\rangle+|1\rangle)$ ning $H(|1\rangle)=1/\sqrt{2}(|0\rangle-|1\rangle)$. Kui rakendada Hadamardi lüli ühele kvantbitile $|0\rangle$, saadakse tulemuseks superpositsioon $1/\sqrt{2}(|0\rangle+|1\rangle)$. Kui rakendada Hadamardi lüli paralleelselt *n* kvantbitile algolekus $|00\ldots0\rangle$, on tulemuseks süsteemi kõigi $2^n$ oleku superpositsioon $1/2^{n/2}\Sigma\,|x\rangle$, kus summa on võetud üle kõigi *n*-bitiste arvude *x* (Hadamardi teisenduse paralleelversiooni nimetatakse Walshi teisenduseks).

### 9.7.5 Kvantinformatsioon

#### *9.7.5.1 Kloonimise võimatus*

Teades, et kvantarvutitega saab täita ainult unitaarseid teisendusi, on lihtne näidata, et ei leidu sellist arvutust *U*, mis paarile $|a0\rangle$ seaks vastavusse paari $|aa\rangle$ suvalise kvantbiti *a* korral. Tõestus on imelihtne. Olgu $|a\rangle$ ja $|b\rangle$ kaks ortogonaalset kvantolekut, st $\langle a|b\rangle=0$. Olgu $|c\rangle=1/\sqrt{2}(|a\rangle+|b\rangle)$. Kuna kõik kvantarvutused on lineaarsed, on seda ka *U*:

$U(|c0\rangle)=U(1/\sqrt{2}(|a0\rangle+|b0\rangle))=1/\sqrt{2}U(|a0\rangle)+\sqrt{2}U(|b0\rangle)=1/\sqrt{2}|aa\rangle+1/\sqrt{2}|bb\rangle$,

mis on vastuolus võrdusega $U(|c0\rangle)=|cc\rangle=1/2(|aa\rangle+|ab\rangle+|ba\rangle+|bb\rangle)$. Toodud teoreemist järeldub, et tundmatu kvantoleku kloonimine on võimatu, sh on võimatu alustades tundmatust kvantolekust $x=a|0\rangle+b|1\rangle$ lõpetada olekutes $a|00\rangle+b|11\rangle$ või $1/2((a|0\rangle+b|1\rangle)\otimes(a|0\rangle+b|1\rangle))$.

#### *9.7.5.2 Tihe kodeerimine*

Tihe kodeerimine on meetod kahe klassikalise biti edastuseks kasutades ühte kvantbitti. Mõjub esialgu üllatavalt, kuna on ju teada, et ühes kvantbitis "sisaldub" täpselt ühe klassikalise biti jagu informatsiooni! Võimatuna näivat ülesannet saab siiski lahendada, kui anda nii saatjale (Alice) kui vastuvõtjale (Bob) eelnevalt üks osake EPR-paarist $1/\sqrt{2}(|00\rangle+|11\rangle)$. Kui Alice soovib hiljem Bobile kaht bitti edastada, rakendab ta vastavalt bitipaari väärtusele 0, 1, 2 või 3 oma osakesele EPR-paarist üht neljast teisendusest ID, *X*, *Y*, *Z*, seega on kvantsüsteemi uueks väärtuseks (vastavalt bitipaari väärtusele) kas $1/\sqrt{2}(|00\rangle+11\rangle)$, $1/\sqrt{2}(|10\rangle+|01\rangle)$, $1/\sqrt{2}(-|10\rangle+|01\rangle)$ või $1/\sqrt{2}(|00\rangle-|11\rangle)$. Seejärel edastab Alice enda osakese Bobile.

Osakese kättesaamise järel rakendab Bob EPR-paarile teisendust CNOT (kvantsüsteemi uueks olekuks on siis kas $1/\sqrt{2}(|00\rangle+|10\rangle)$, $1/\sqrt{2}(|11\rangle+|01\rangle)$, $1/\sqrt{2}(-|11\rangle+|01\rangle)$ või $1/\sqrt{2}(|00\rangle-|10\rangle)$) ning mõõdab teise kvantbiti väärtuse. Kuna kõigis neljas olekus on liidetavate vektorite teised komponendid võrdsed, ei kollapseeru endiselt sasiolekus olev paar mõõtmisel. Kui Bob saab mõõtmisel tulemuseks $|0\rangle$, oli kodeeritud väärtus 0 või 3; kui Bob saab tulemuseks $|1\rangle$, oli kodeeritud väärtus 1 või 2. Kvantsüsteemi esimese kvantbiti väärtus on pärast CNOT- teisendust kas $1/\sqrt{2}(|0\rangle+|1\rangle)$, $1/\sqrt{2}(|0\rangle+|1\rangle)$, $1/\sqrt{2}(|0\rangle-|1\rangle)$ või $1/\sqrt{2}(|0\rangle-|1\rangle)$. Kui Bob lõpuks rakendab esimele osakesele Hadamard'i teisendust *H*, teiseneb kvantsüsteemi esimene bitt ühte järgmisest neljast olekust: $|0\rangle$, $|0\rangle$, $|1\rangle$ või $|1\rangle$. Mõõtnud esimese kvantbiti väärtust, suudab Bob nüüd eristada juhtu 0 juhust 3 ja juhtu 1 juhust 2 ning teab seega täielikult Alice'i bitipaari väärtust.

#### *9.7.5.3 Teleportimine*

Veelgi ebaintuitiivsem (ning ulmelisem) on teleportimine, otsene vastand tihedale kodeerimisele: nimelt on teleportimine meetod ühe kvantbiti edastuseks, kasutades kaht klassikalist bitti. Üllatav, sest võimaldab teisaldada tundmatu kvantbiti väärtust!

Teleportimisprotokoll on tiheda kodeerimise protokolli vastand. Teleportimise alguses on nii Alice'il kui Bobil üks osake EPR-paarist. Alice soovib Bobile edastada kvantbitti $\varphi=a|0\rangle+b|1\rangle$, kasutades vaid klassikalisi kanaleid. Kvantsüsteemi algolek on

$$1/\sqrt{2}(a|0\rangle\otimes(|00\rangle+|11\rangle)+b|1\rangle\otimes(|00\rangle+|11\rangle))=1/\sqrt{2}(a|000\rangle+ a|011\rangle+b|100\rangle+b|111\rangle),$$

millest Alice kontrollib esimest ja teist ning Bob kontrollib kolmandat kvantbitti. Seejärel rakendab Alice esmalt kahele kontrollitavale kvantbitile juhitavat eitust ning seejärel esimesele kvantbitile Hadamard'i teisendust. Pärast teisenduste rakendamist mõõdab Alice esimesed kaks kvantbitti, saades võrdtõenäoliselt ühe neljast väärtusest $|00\rangle$, $|01\rangle$, $|10\rangle$ või $|11\rangle$. Lihtne arvutus näitab, et Bobi kvantbitt projitseerub vastavalt üheks neljast väärtusest $a|0\rangle+b|1\rangle$, $a|1\rangle+b|0\rangle$, $a|0\rangle-b|1\rangle$ või $a|1\rangle-b|1\rangle$. Alice saadab oma mõõtmise tulemused kahe klassikalise bitina Bobile.

Kui Bob saab Alice'ilt kaks bitti, saab ta ka teada, millises seoses on tema käes oleva EPR-paari teise osakese olek Alice'i kvantbiti algolekuga. Seega, rakendades kolmandale kvantbitile vastavalt kahele saadud bitile üht neljast teisendusest ID, *X*, *Y* või *Z* saab Bob üheselt teada Alice'i kvantbiti algoleku.

Tähele tasub panna, et mõõtes muutis Alice taastamatult kvantbiti $\varphi$ väärtust ning seega ei räägi biti teleportimine vastu kloonimise mittevõimalikkusele.

### 9.7.6 Universaalne kvantarvuti

#### *9.7.6.1 Klassikaliste arvutuste simuleerimine*

Näitamaks, et kvantarvuti suudab simuleerida klassikalist arvutit, piisab kui esitada klassikalisi NOT- ja AND-tehteid täitvad kvantprogrammid. AND-tehte simuleerimisel tekib probleem, kuna AND (loogiline ja) ei ole pööratav: ainult lause "aias sadas saia *ja* ööd on siin mustad" väärusest lähtudes ei ole võimalik kindlaks teha, kumb lause komponentidest tegelikult väär oli. Dilemma lahendab nn prügibittide (kvantbittide, mille ainus eesmärk on alal hoida pööratavust tagavat informatsiooni) kasutuselevõtt: tuleb välja, et $T(|1,1,x\rangle)=|1,1,\neg x\rangle$ ning $T(|x,y,0\rangle)=|x,y,x\&y\rangle$. Toffoli lüli võib asendada mõne teise kvantuniversaalse lüliga, näiteks 3-kvantbitise Fredkini lüliga, mis vahetab viimased kaks kvantbitti ära parajasti siis kui esimene kvantbitt on $|1\rangle$. Järgnevas võime seega eeldada, et suvalise klassikaliselt arvutatavale funktsiooni *f* korral leidub seda arvutav (unitaarne) *kvantskeem* $U_f:|x,y\rangle\rightarrow|x,y\oplus f(x)\rangle$. Kuna suvalise *x* korral $f(x)\oplus f(x)=0$, annab $U_f$ topeltrakendamine tulemuseks samasusteisenduse: $U_f(U_f(|x\rangle))=|x\rangle$ suvalise kvantbiti $|x\rangle$ jaoks.

#### *9.7.6.2 Kvantarvutuste simuleerimine*

Eelmises lõigus öeldu põhjal teame nüüd, et kvantarvuti on võimeline efektiivselt simuleerima klassikalist arvutit. Kuidas on ent kvantarvutuste endiga? Kui keerukatest elementaarlülidest peab koosnema "universaalne" kvantarvuti, millega oleksid teostatavad kõik unitaarsed teisendused? On tõestatud, et kõik kvantarvutused saab sooritada, kui võtta elementaarlülideks lisaks juba tuntud CNOT-lülile järgmised rotatsioonid ning faasinihketeisendused iga kompleksarvulise $\alpha$ jaoks:

$$\begin{pmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{-i\alpha} \end{pmatrix}, \begin{pmatrix} e^{i\alpha} & 0 \\ 0 & e^{i\alpha} \end{pmatrix}.$$

### 9.7.7 Kvantalgoritmid

#### *9.7.7.1 Kvantparallelismi kasutamine*

Kvantparallelismi kasutamise üldine skeem on järgmine. Teisendame oleku $|00\ldots0\rangle$ Walshi teisenduse abil kõigi sisendite superpositsiooni $|W\rangle=1/2^{n/2}\Sigma\,|x,0\rangle$, kus $x$ omandab kõikvõimalikud $2^n$ $n$-bitised väärtused. Rakendame olekule $|W\rangle$ skeemi $U_f$. Kuna $U_f$ on lineaarteisendus, "rakendub" $U_f$ simultaanselt kõigile superpositsioonis olevatele sisenditele ning seega on võimalik leida funktsiooni f väärtustus kõikvõimalikel sisenditel, rakendades skeemi $U_f$ üksainus kord: $U_f\,(|W\rangle)=1/2^{n/2}\Sigma\,|x,f(x)\rangle$. Seega tähendab kvantparallelism eksponentsiaalset parallelismi lineaarses ruumis.

Skeemi $U_f$ rakendamise järel saadud tulemust võib vaadelda kui funktsiooni $f$ tõeväärtuste tabelit ehk kui funktsiooni graafikut. Oleku $U_f\,(|W\rangle)$ viimase kvantbiti mõõtmine piki baasi $|y\rangle$ annab tulemuseks kõigi selliste sisendväärtuste $|x\rangle$ superpositsiooni, mille korral $f(x)=y$. Superpositsiooni edasine mõõtmine ei tundu aga enam otsest kasu andvat: kvantparallelismi ja mõõtmiste seos on väga õrn, siiamaani on teada vaid üksikuid ülesandeid, mida kvantarvuti suudab kiiremini lahendada kui tavaline arvuti. Teadaolevad kvantalgoritmid jagunevad laias laastus kahte liiki:

1. Algoritmid, mis võimendavad soovitavat tulemust. Sellised algoritmid teisendavad kvantolekut nii, et mõõtmisel saadakse soovitav tulemus suurema tõenäosusega kui soovimatud tulemused. Näide: Groveri otsialgoritm
2. Algoritmid, mis leiavad funktsiooni $f$ väärtuste $f(x)$ ühiseid omadusi, näiteks funktsiooni perioodi. Näide: Shori faktoriseerimisalgoritm.

Mõlemad ülalmainitud algoritmid on olulised krüptograafias, seetõttu valgustame neid allpool lähemalt.

#### *9.7.7.2 Otsialgoritmid*

Paljusid probleeme saab formuleerida kui otsiprobleeme kujul "leida selline $x$, et $P(x)$ oleks tõene". Sellesse klassi kuuluvad ülesanded andmebaasi päringutest kuni graafi värvimiseni. Teatud tüüpi probleemide korral on teada mingi abistav lisainformatsioon, mida saab kasutada tõhusa lahendusalgoritmi leidmisel. Paljude otsiülesannete (näiteks graafi värvitavus või elemendi otsing järjestatud listis) korral on otsinguruum struktrureeritud, sellistel juhtudel saab ülesande täislahendi kätte teatud alamprobleemide lahendite kombineerimisel. Üldjuhul, lisastruktuuride puudumisel, puudub täielikust läbivaatusest parem algoritm: kui otsinguruumi suurus on $N$, kulub struktureerimata otsinguks $O(N)$ sammu.

Erialateadlastele tuli täieliku šokina Lev Groveri poolt 1997. a publitseeritud randomiseeritud kvantalgoritm, mis redutseeris struktureerimata otsinguks kuluva aja suurusjärgult $O(N)$ suurusjärgule $O(\sqrt{N})$. On ka tõestatud, et Groveri algoritm on parim võimalik algoritm struktureerimata otsinguruumi puhul. Tad Hogg näitas hiljem, kuidas kasutada täiendavat informatsiooni otsinguruumi struktuuri kohta Groveri algoritmi kiirendamiseks. Kahjuks on Hoggi poolt konstrueeritud algoritmid juba nii keerukad, et nende õnnestumistõenäosust on väga raske määrata ning seega on ka teadmata, *kui* tõhusad Hoggi algoritmid on. Tavaliselt testitakse heuristiliste algoritmide efektiivsust algoritmi empiirilise testimisega,

kvantarvuti puhul on see esialgu võimatu tingituna kvantarvuti simuleerimiseks kuluvast eksponentsiaalsest ajast klassikalisel arvutil. Väikeste sisendite korral on testimine näidanud, et Hoggi algoritm on ilmselt kiirem kui Groveri algoritm, kuid kiiruse võit on vaid polünomiaalne. Kuni pole ehitatud piisavalt suuri kvantarvuteid või leitud paremaid matemaatilisi meetodeid selliste algoritmide analüüsimiseks, ei ole võimalik Hoggi algoritmi efektiivsust 100-protsendise kindlusega määrata.

Klassikalise otsialgoritmi ebaefektiivsusel põhineb muuhulgas ka räsifunktsioonide turvalisus: teame eelnevast, et plokipikkusega $2^n$ ideaalselt turvalise räsifunktsiooni $H$ korral kulub suvalise etteantud väärtuse $y$ korral $y$-le vastava väärtuse $x$, $H(x)=y$, leidmiseks keskmiselt $2^{n-1}$ sammu. Kvantarvutamise otsimise korral tuleb sama turvalisustaseme saavutamiseks suurendada räsifunktsioonide plokipikkust.

### *9.7.7.3 Groveri algoritmi lühikirjeldus*

Groveri algoritm teostab otsingut struktureerimata listis suurusega $N$, ning olgu $n$ piisavalt suur naturaalarv, mille korral $2^n>N$. Olgu $n$-bitine predikaat $P$ teostatud kvantskeemina $U_P$: $|x,0\rangle\rightarrow|x,P(x)\rangle$, kus teise biti väärtus on 1, kui $P(x)$ on tõene, ning 0, kui $P(x)$ on väär. Groveri algoritmi esimene samm on tavaline: sisendile $|00\ldots0\rangle$ rakendatakse järjestikku Walshi teisendust ning seejärel skeemi $U_P$, saades tulemuseks summa $A(P)=1/2^{n/2}\Sigma|x,P(x)\rangle$, kus $x$ muutub üle kõikvõimalike $n$-bitiste bitistringide. Suvalise $x_0$ korral, mille jaoks $P(x_0)$ on tõene, kuulub $|x_0,1\rangle$ superpositsiooni $A(P)$, kuid kuna selle oleku amplituud on $1/2^{n/2}$, on tõenäosus superpositsiooni mõõtmise järel suuruse $x_0$ saamiseks $1/2^{n/2}$. Groveri algoritmi järgmised sammud muudavad kvantolekut $A(P)$ nii, et olekute $|x,1\rangle$, mille puhul $P(x)$ on tõene, amplituud suureneks olekute $|x,0\rangle$ amplituudide arvel. Laskumata üksikasjadesse, anname vaid viimase toimingu (amplituudi võimendamise) lühikirjelduse.

1. Asendada olekute $|x,1\rangle$ amplituudid nende pöördväärtustega.
2. Olgu $S$ kõigi $|x,1\rangle$-olekute amplituudide summa. Asendada suvalise oleku $|x,1\rangle$ amplituud $a$ väärtusega $2S–a$.
3. Korrata samme 1 ja 2 umbes $(\pi/4)2^{n/2}$ korda.

Kui leidub parajasti üks selline $x$, mille korral $P(x)$ on tõene, on Groveri algoritmi õnnestumistõenäosus pärast $(\pi/8)2^{n/2}$ sammu 0.5. Pärast $(\pi/4)2^{n/2}$ sammu on algoritmi õnnestumistõenäosus juba $1-2^{-n}$. Huvitav on see, et algoritmi jätkuv kordamine vähendab õnnestumistõenäosust, näiteks on see tõenäosus pärast $(\pi/2)2^{n/2}$ sammu ligikaudu 0.

### *9.7.7.4 Shori faktoriseerimisalgoritm*

Nagu jaotise 9.7 alguses lühidalt mainitud, tingis huvi tõusu kvantinformaatika vastu kiire faktoriseerimisalgoritmi avastamine Peter Shori poolt. Shori algoritm töötab randomiseeritud polünomiaalses ajas, olles järelikult sama keerukusastmega kui parimad teadaolevad algarvutestid. Veel enam, Shori algoritmi keerukus on täpselt sama, mis modulaareksponendi arvutamisel; vahe on vaid randomiseerituses – Shori algoritm annab õige vastuse vaid teatud tõenäosusega. Arvestades sellega, et Shori algoritmi lineaarne arv kordi korrates kahaneb vale vastuse saamise tõenäosus eksponentsiaalselt, võib pisut üldistades siiski öelda, et kvantarvutite olemasolu korral on RSA-tüüpi krüptosüsteemi lahtimurdmine täpselt niisama lihtne kui RSA-ga krüpteerimine, nii et enamiku tänapäeva asümmeetriliste krüptosüsteemide kasutamine kaotab mõtte.

Shori algoritm kasutab tavalist faktoriseerimisprobleemi taandamist funktsiooni perioodi leidmise probleemile. Algoritmi alguses kasutatakse kvantparallelismi nagu ikka, saades funktsiooni $f$ väärtused kõikvõimalikel sisenditel vastava kvantskeemi $U_f$ ühekordse rakendamisega. Seejärel leitakse funktsiooni nn kvant-Fourier' teisendus (KFT). Pärast KFT rakendamist saab mõõtmisel suure tõenäosusega kätte funktsiooni $f$ perioodi, mida kasutatakse järgnevaks faktoriseerimiseks. Järgnevas paaris jaotises kirjeldame lähemalt Shori algoritmi eri aspekte.

### *9.7.7.5 Kvant-Fourier' teisendus*

Diskreetne Fourier' teisendus teisendab väärtustehulgaga 0,…,$N$–1 funktsiooni $g$ funktsiooniks $F_g$, piltlikult öeldes teisendatakse "ajafunktsioon" "sagedusfunktsiooniks", mille väärtuste hulgaks on lõigus $[0,2\pi)$ asetsevad suuruse $2\pi/N$ kordsed. Muuhulgas, kui funktsiooni $g$ periood on $r$, on tulemfunktsiooni $F_g$ väärtus nullist erinev vaid sageduse $1/r$ kordsetel sisenditel. KFT opereerib kvantoleku amplituudidega, teisendades oleku $\Sigma_x g(x)|x\rangle$ olekuks $\Sigma_c F_g(c)|c\rangle$, kus $x$ ja $c$ muutuvad üle kõikvõimalike täisarvude 0 ja $N$–1 vahel. Kui kvantolekut mõõta pärast KFT rakendamist, saadakse tulem $|c\rangle$ tõenäosusega $|G(c)|^2$. Sealjuures on $G(c)$ nullist erinev vaid suuruse $N/r$ kordsetel punktidel ning seega jagub mõõtmistulemus suurusega $N/r$.

Kuna KFT on variant kahe astmetel baseeruvast kiirest Fourier' teisendusest, mis annab täpsed vastused vaid kahe astmele vastaval perioodil, on ka KFT tulemus ligikaudne, kui funktsiooni $g$ periood ei ole kahe aste. KFT täpsus suureneb koos baasiks kasutatud kahe astme kasvuga; kvantskeem $U_{\mathrm{KFT}}$ baasil $2^m$ defineeritakse järgnevalt:

$$U_{\mathrm{KFT}}:|x\rangle \rightarrow 1/2^{m/2}\Sigma \exp(2\pi cx/2^m)|c\rangle.$$

KFT arvutamiseks kulub $m(m+1)/2$ kvantskeemi.

### *9.7.7.6 Shori algoritmi raamkava*

1. Genereeritakse arv $a$ juhuslikult. Kui SÜT($a$,$M$)>1, on arv $M$ faktoriseerinud. Vastasel juhul jätkatakse algoritmi täitmist.
2. Olgu m selline, et $M^2 \le 2^m < 2M^2$. Kasutades kvantparallelismi, arvutatakse $f(x)=a^x \bmod M$. Samm 2 lõpetab olekus $1/2^{m/2}\Sigma|x,f(x)\rangle$ (kus $x$ muutub 0-st $2^{m-1}$-ni).
3. Konstrueeritakse kvantolek, mille amplituudifunktsioonil $g(x)$ on sama periood kui funktsioonil $f(x)$. Selleks mõõdetakse sammu 2 lõppolekut, saades tulemuseks juhusliku suuruse $u$. Mõõtmine projitseerib olekuruumi alamruumi, mis on ühilduv mõõdetud väärtusega. Seega on pärast mõõtmist uueks olekuks $C\Sigma g(x)|x,u\rangle$, mingi normeerimisfaktori $C$ jaoks, kus g(x)=1, kui $f(x)=u$, ja $g(x)=0$ vastasel juhul. Kuna summas esinevad $x$-i väärtused erinevad teineteisest perioodi kordsete võrra, oleme leidnud otsitava funktsiooni. Edasises ei kasutata enam oleku viimast poolt $u$.
4. Rakendatakse olekule $C\Sigma g(x)|x\rangle$ KFT-d, saades oleku $C\Sigma F_g(c)|c\rangle$. Üldisest Fourier' teisenduste teooriast on teada, et kui $g(x)$ periood on kahe aste, on KFT tulemuseks olek $D\Sigma_j \rho_j|j2^m/r\rangle$, kus $|\rho_j|=1$. Kui periood $r$ ei jaga suurust $2^m$, aproksimeerib teisendus täpset juhtu nii, et enamik amplituude on täisarvud, mis paiknevad suuruse $2^m/r$ kordsete läheduses.
5. Saadud olekut mõõdetakse standardses baasis, saades tulemuseks väärtuse $v$. Kui periood on kahe aste (nii et KFT väljastab skaleeritud sageduse täpsed kordsed), on perioodi lihtne ekstraheerida: sellisel juhul $v=j2^m/r$, kus $j$ on täisarv. Enamasti on naturaalarvud $j$ ja $r$ ühistegurita, sellisel juhul saab murrust $v/2^m = j/r$ lihtsalt leida perioodi $r$. Üldjuhul, kui periood ei ole kahe aste, annab KFT ainult ligikaudselt skaleeritud sageduse kordsed. Sellisel juhul on perioodi ekstraheerimine komplitseeritum, nõudes perioodi väärtuse äraarvamist. Olgu arvatud väärtus $q$.
6. Kui $q$ on paarisarv, kasutatakse Eukleidese algoritmi, et kontrollida, kas ühel kahest suurusest $a^{q/2}+1$ ja $a^{q/2}-1$ on mittetriviaalseid ühistegureid arvuga $M$. Kui $q$ on tõepoolest funktsiooni $f(x)=a^x \bmod M$ periood, siis $a^q \bmod M=1$, kuna $a^q a^x \bmod M = a^x \bmod M$, suvalise $x$ korral. Kuna $q$ on paarisarv, siis $(a^{q/2}+1)(a^{q/2}-1)=0 \bmod M$. Seega, kui kumbki teguritest pole $M$ kordne, on ühel neist teguritest arvuga $M$ mittetriviaalne ühistegur.
7. Korrata algoritmi, kui vaja. Vajadus võib tekkida, kui (a) $v$ pole $2^m/r$ kordaja, (b) perioodil $r$ ja kordajal $j$ on ühistegur, nii et $q$ pole mitte periood, vaid perioodi tegur, (c) samm 6 annab tegurina arvu $M$ enda, (d) funktsiooni $f(x)$ periood on paaritu. Algoritmi mõned korrad korrates saadakse suure tõenäosusega õige tulem.

# 10 DIGITAALSIGNATUURID

Sõnumi digitaalsignatuur on teatav arv, mis sõltub signeerija ainuvalduses olevast salajasest võtmest ja signeeritavast sõnumist. Signatuurid peavad üldiselt olema verifitseeritavad ilma salajast võtit kasutamata. See viitab avaliku võtmega krüptosüsteemidele. Digitaalsignatuurid saidki praktiliselt võimalikuks esimeste salaluugiga ühesuunaliste funktsioonide *(trapdoor one-way functions*) kandidaatide esitamisega 70te aastate lõpus. Tänapäevaks on tekkinud palju erinevaid digitaalsignatuuri skeeme, mis võimaldavad saavutada selliseid turvaeesmärke, millest 70tel aastatel isegi ei mõeldud.

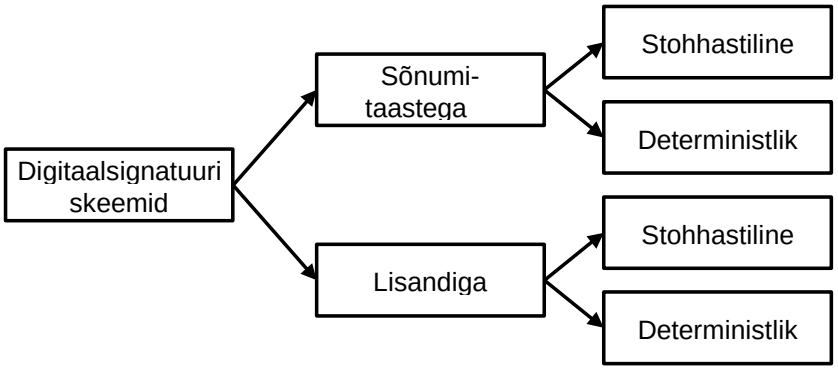
## 10.1 Digitaalsignatuuri skeemide üldine mudel

Tähistagu $\boldsymbol{M}$ kõikvõimalike signeeritavate sõnumite hulka. Eeldame, et signeerimisfunktsiooni ei rakendata otseselt hulga sõnumitele (hulga $\boldsymbol{M}$ elementidele), vaid neid teisendatakse eelnevalt, vastava teisenduse kõikvõimalike tulemuste hulka tähistab $\boldsymbol{M}_S$. Signeerimisteisendus kujutab elemendi hulgast $\boldsymbol{M}_S$ hulka $\boldsymbol{S}$. Mõnedes signatuuriskeemides kasutatakse ka nn liiasusfunktsiooni $R$, mis on mingi üksühene funktsioon hulgast $\boldsymbol{M}$ hulka $\boldsymbol{M}_S$. Kasutatakse veel indeksite hulka $\boldsymbol{R}$, mille iga element määrab teatud kindla signeerimisfunktsiooni. Kasutatakse ka ühesuunalist räsifunktsiooni $h$, mille määramispiirkond on $\boldsymbol{M}$ ja väärtuste piirkond $\boldsymbol{M}_h \subseteq \boldsymbol{M}_S$.

Digitaalsignatuuri skeem koosneb signeerimisalgoritmist sõnumi digitaalsignatuuri genereerimiseks ja verifitseerimisalgoritmist selle signatuuri kontrollimiseks mistahes teise kasutaja poolt. Digitaalsignatuuri skeeme võib jagada kaheks klassiks:

- lisandiga (*with appendix*) digitaalsignatuurid nõuavad, et originaalsõnum oleks ka verifitseerimisalgoritmi sisend;
- sõnumitaastega digitaalsignatuurid seda ei nõua, sest signeeritav sõnum on taastatav signatuurist endast.

Mõlemasse klassi kuuluvaid signatuuriskeeme saab jagada veel hulga $\boldsymbol{R}$ suuruse järgi. Kui $|\boldsymbol{R}|>1$, nimetatakse skeemi stohhastiliseks (*randomized*), vastasel juhul determineerituks. Determineeritud skeeme saab veel jagada ühekordseteks skeemideks ja mitmekordseteks skeemideks.




**Joonis 69. Digitaalsignatuuri skeemide üldine liigitus**

### 10.1.1 Lisandiga skeemid

Lisandiga digitaalsignatuurid on kõige levinum praktikas kasutatav signatuuride klass. Nende signatuuride turvalisus sõltub suuresti kasutatavate räsifunktsioonide turvaomadustest, mistõttu praktikas kasutatavad süsteemid on tavaliselt vähem kaitstud nn eksistentsiaalsete võltsingute vastu. Lisandiga skeemide põhiiseärasus on selles, et signatuuri verifitseerimiseks läheb vaja ka sõnumit ennast. Tüüpiline näide on DSA (*Digital Signature Algorithm*).

**Võtmete genereerimine**:

1. Iga kasutaja $A$ genereerib salajase võtme, st teisenduste hulga $\boldsymbol{S}_A = \{S_{A,k}: k \in \boldsymbol{R}\}$, mille iga element $S_{A,k}$ on mingi üksühene kujutus hulgast $\boldsymbol{M}_h$ hulka $\boldsymbol{S}$ ja mida nimetatakse signeerimisteisenduseks.

2. $\boldsymbol{S}_A$ abil defineeritakse ka kujutus $V_A$ hulgast $\boldsymbol{M}_h \times \boldsymbol{S}$ hulka $\{0,1\}$, nii et

$$V_A(m',s) = 1 \Leftrightarrow S_{A,k}(m') = s$$

mistahes $m' \in \boldsymbol{M}_h$ ja $s \in \boldsymbol{S}$ korral, kus $m'=h(m)$ ja $m \in \boldsymbol{M}$ on mingi sõnum. Teisendust $V_A$ nimetatakse verifitseerimisteisenduseks ja see on konstrueeritud nii, et seda oleks võimalik arvutada ilma signeerija salajast võtit teadmata.

3. Kasutaja $A$ avalik võti on $V_A$ ja salajane võti on hulk $\boldsymbol{S}_A$.

**Signeerimine**. Kasutaja $A$ moodustab sõnumi $m \in \boldsymbol{M}$ digitaalsignatuuri $s \in \boldsymbol{S}$.

1. Valitakse element $k \in \boldsymbol{R}$.

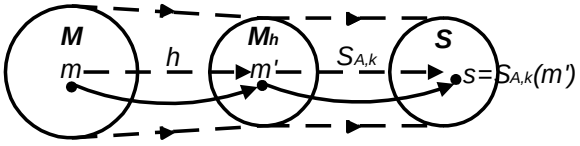
2. Arvutatakse $m'=h(m)$ ja $s=S_{A,k}(m')$.

3. Sõnumi $m$ signatuur on $s$. Nii signatuur $s$ kui ka sõnum tehakse kättesaadavaks kõigile, kel peaks tekkima vajadus signatuuri verifitseerida.

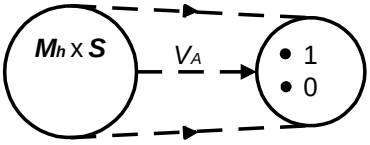
**Verifitseerimine**. Kasutaja $B$ kontrollib signatuuri $s$ autentsust.

1. Arvutatakse $m'=h(m)$ ja $u=V_A(m',s)$.

2. Kui $u=1$, loetakse signatuur $s$ autentseks.




(a) Signeerimine




(b) Verifitseerimine

**Joonis 70. Lisandiga digitaalsignatuuri üldmudel**

Kasutatavatelt funktsioonidelt eeldatakse järgmisi omadusi.

- $S_{A,k}$ peab olema kergesti arvutatav mistahes valitud $k \in \boldsymbol{R}$ korral.
- $V_A$ peab olema kergesti arvutatav.
- Kõigil kasutajatel, välja arvatud $A$, peab olema arvutuslikult raske leida sõnumit $m \in \boldsymbol{M}$ ja signatuuri $s \in \boldsymbol{S}$, nii et $V_A(h(m),s)=1$.

### 10.1.2 Sõnumitaastega digitaalsignatuurid

Sõnumitaastega signatuuride peamine iseärasus on selles, et pelgalt signatuuri abil on võimalik taastada ka signeeritavat sõnumit. Tüüpiliselt on see võimalik vaid siis, kui signeeritava sõnumi pikkus on piiratud. Selle klassi üks esindaja on RSA skeem.

**Võtmete genereerimine**:

1. Iga kasutaja $A$ genereerib salajase võtme, st teisenduste hulga $\boldsymbol{S}_A = \{S_{A,k}: k \in \boldsymbol{R}\}$, mille iga element $S_{A,k}$ on mingi üksühene kujutus hulgast $\boldsymbol{M}_S$ hulka $\boldsymbol{S}$ ja mida nimetatakse signeerimisteisenduseks.

2. $\boldsymbol{S}_A$ abil defineeritakse ka kujutus $V_A$ hulgast $\boldsymbol{S}$ hulka $\boldsymbol{M}_S$, nii et

$$V_A(S_{A,k}(m)) = m$$

mistahes $m \in \boldsymbol{M}_S$ korral. Teisendust $V_A$ nimetatakse verifitseerimisteisenduseks ja see on konstrueeritud nii, et seda oleks võimalik arvutada ilma signeerija salajast võtit teadmata.

3. Kasutaja $A$ avalik võti on $V_A$ ja salajane võti on hulk $\boldsymbol{S}_A$.

**Signeerimine**. Kasutaja $A$ moodustab sõnumi $m \in \boldsymbol{M}$ digitaalsignatuuri $s \in \boldsymbol{S}$.

1. Valitakse element $k \in \boldsymbol{R}$.

2. Arvutatakse $m' = R(m)$ ja $s = S_{A,k}(m')$.

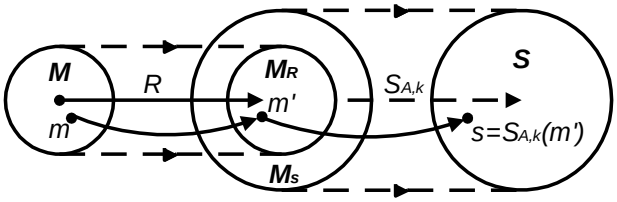
3. Sõnumi $m$ signatuur on $s$, mis tehakse kättesaadavaks kõigile, kel peaks tekkima vajadus signatuuri verifitseerida.

**Verifitseerimine**. Kasutaja $B$ kontrollib signatuuri $s$ autentsust.

1. Arvutatakse $m' = V_A(s)$.

2. Kontrollitakse, kas $m' \in \boldsymbol{M}_R$. Kui vastus on eitav, loetakse signatuur kehtetuks.

3. Taastatakse sõnum $m = R^{-1}(m')$.




**Joonis 71. Sõnumitaastega digitaalsignatuuri üldmudel**

Kasutatavatelt funktsioonidelt eeldatakse järgmisi omadusi.

- $S_{A,k}$ peab olema kergesti arvutatav mistahes valitud $k \in \boldsymbol{R}$ korral.
- $V_A$ peab olema kergesti arvutatav.

- Kõigil kasutajatel, välja arvatud $A$, peab olema arvutuslikult raske leida sõnumit $m \in \boldsymbol{M}$ ja signatuuri $s \in \boldsymbol{S}$, nii et $V_A(h(m),s)=1$.

Ehkki liiasusfunktsioon $R$ ja ka tema pöördfunktsioon $R^{-1}$ on avalikud, on õige $R$ valik süsteemi turvalisuse seisukohalt olulise tähtsusega tegur. Näiteks, kui $\boldsymbol{M}_R = \boldsymbol{M}_S$ ja $R$ ning $S_{A,k}$ on bijektsioonid vastavalt hulgast $\boldsymbol{M}$ hulka $\boldsymbol{M}_R$ ja hulgast $\boldsymbol{M}_S$ hulka $\boldsymbol{S}$, on mistahes valitud signatuuri $s \in \boldsymbol{S}$ jaoks kerge leida sõnumit $m$, mille signatuur on $s$. Ründaja võib toimida järgmiselt.

1. Valitakse juhuslikult $k \in \boldsymbol{R}$ ja $s \in \boldsymbol{S}$.
2. Arvutatakse $m'=V_A(s)$.
3. Arvutatakse $m=R^{-1}(m')$.

Seega on saadud sõnum-signatuur-paar ilma salajast võtit kasutamata. Enne võltsingut pole küll täpselt teada, milline tuleb välja sõnum $m$, kuid paljudel juhtudel on see rünne ka praktikas oluline.

Liiasusfunktsiooni ei saa valida sõltumatult signeerimisfunktsioonidest $S_{A,k}$. Näiteks RSA skeemi puhul ei tohi $R$ olla morfism, st $R(a \cdot b)=R(a) \cdot R(b)$.

### 10.1.3 Signatuuriskeemide tüüpründed

Ründaja põhieesmärk on luua digitaalsignatuure, mis verifitseerimisel tunnistataks autentseteks. Digitaalsignatuuri skeemile võib olla omane üks järgmistest murtuse astmetest:

- Täielik murdmine – ründajal on võimalik kas välja arvutada signeerija privaatvõti, või siis leida õige signeerimisalgoritmiga funktsionaalselt samaväärne algoritm.
- Selektiivne võltsing – ründaja saab moodustada teatud sõnumite klassile digitaalsignatuure, mis verifitseerimisel tunnistatakse autentseteks.
- Eksistentsiaalne võltsing – ründajal on võimalik moodustada vähemalt ühele sõnumile verifitseerimisel autentseks tunnistatav digitaalsignatuur.

Sõltuvalt ründaja käsutuses oleva lähteinformatsiooni hulgast ja ründaja võimalustest jagatakse ründed signatuuriskeemidele järgmisse nelja klassi, mis on analoogilised plokkšifrite ja räsifunktsioonide vastavate rünnetega.

1. Ainult võtmega rünne – ründajale on teada ainult signeerija avalik võti.
2. Teadaoleva sõnumiga rünne – ründajale on teada mingi hulk sõnum-signatuur-paare.
3. Valitava sõnumiga rünne – ründajal on võimalik valida teatud hulk sõnumeid ja saada teada neile vastavad signatuurid. See valik ei ole adaptiivne, st sõnumid tuleb valida enne kui saadakse teada neile vastavd signatuurid.
4. Adaptiivselt valitava sõnumiga rünne – sama, mis valitava sõnumiga rünne, ent ründajal on võimalus küsida igale sõnumile signatuuri, olles eelnevalt teadlik eelmiste valitud sõnumite signatuuridest.

Kui signatuuriskeemis kasutatakse mingit räsifunktsiooni $h$, tuleb seda funktsiooni arvestada kui signatuuriskeemi põhiosa, et ründaja ei saaks funktsiooni $h$ vahetada nõrgema räsifunktsiooni vasu vms.

## 10.2 Valitava krüptogrammiga rünne PKCS #1 vormingut kasutades

Aastal 1998 avastas Daniel Bleichenbacher praktikas teostatava ründe RSA krüpteerimisstandardi PKCS #1 vastu, mille eelduseks on, et ründajal on võimalik valida suvalisi bitistringe $y$ ja nn dekrüpteerivalt oraaklilt kasutades teada saada, kas avatekst $x = S_A(y) = s^d \bmod n$ on PKCS #1 vormingus. Ründe tulemusena moodustatakse signatuur $S_A(m)=m^d \bmod n$ ründaja poolt valitud sõnumile $m$.

Rünne on praktiline seetõttu, et mitmetes protokollides, sh näiteks protokollis SSL, on võimalik serverit kasutada dekrüpteeriva oraaklina, võttes arvesse serveri veateateid.

### 10.2.1 PKCS #1 vorming

Standard PKCS #1 kirjeldab kolme plokivormingut: vormingud 0 ja 1 on reserveeritud digitaalsignatuuridele, vormingut 2 kasutatakse andmete krüpteerimiseks avaliku võtmega. Kirjeldatud ründes kasutatakse ära krüpteerimisvorming 2.

| 00 | 02 | Täidis | 00 | Andmeplokk |
|---|---|---|---|---|

**Joonis 72. PKCS #1 krüpteerimisploki vorming**

Olgu $k$ mooduli $n=pq$ pikkus baitides, st $2^{8(k-1)} \le n < 2^{8k}$. Andmeplokk $D$ pikkusega $|D| \le k-11$ krüpteeritakse järgmiselt.

- Moodustatakse juhuslikult genereeritud $k-3-|D| \ge 8$ mittenullisest baidist koosnev täidisstring $PS$.
- Moodustatakse krüpteerimisplokk $EB = 00||02||PS||00||D$ (vt Joonis 72).
- Plokk $EB$ teisendatakse positiivseks täisarvuks $x$ ja krüpteeritakse avaliku võtmega; saadakse krüptogramm $c = x^e \bmod n$.

Krüpteerimisplokk $EB_1||EB_2||...||EB_k$ on PKCS#1 vormingus, kui

1) $EB_1 = 00$,
2) $EB_2 = 02$,
3) $EB_3 - EB_{10}$ on mittenullised,
4) vähemalt üks baitidest $EB_{11} - EB_k$ on $00$.

### 10.2.2 Ründe üldkirjeldus

Ründaja valib sõnumi $c$ ja soovib arvutada digitaalsignatuuri $m=c^d \bmod n$. Ründaja võib kasutada dekrüpteerivat oraaklit, mis ükskõik millise valitud bitistringi $y$ korral ütleb, kas

$$y^d \bmod n$$

on PKCS#1 vormingus või mitte. Ründaja valib sobiva arvu $s$, leiab $c'=cs^e \bmod n$ ja saadab $c'$ oraaklile saades teada, kas kaks esimest baiti arvust $m \cdot s$ on $00$ ja $02$. Positiivse vastuse korral on ründajale teada, et

$$2B \le m \cdot s \bmod n < 3B,$$

kus $B=2^{8(k-2)}$. Parameetrit s sobivalt varieerides ja saadud informatsiooni kogudes on võimalik saada piisavalt palju teavet õige signatuuri $m$ leidmiseks suure tõenäosusega. Tavaliselt piisab umbes $2^{20}$ valitavast krüprogrammist.

## 10.3 Ühekordsed digitaalsignatuurid

Ühekordsed signatuurid on mehhanismid, mis on mõeldud mitte rohkema kui ühe sõnumi signeerimiseks. Signatuuri verifitseerimisel ja kahtluse alla seadmisel on tüüpiliselt vajalik kolmas usaldatav osapool (*trusted third party*, TTP).

Olgu $E$ mingi fikseeritud plokkšiffer ja $h$ mingi räsifunktsioon. Rabini pakutud ühekordse signatuuri skeemi järgi võtme genereerimiseks sooritab kasutaja $A$ järgmised sammud.

- Valitakse $2n$ salajast võtit $k(1)$, $k(2)$, ..., $k(2n)$, igaüks pikkusega $l$-bitti.
- Arvutatakse $y_i = E_{k(i)}(i)$, $1 \le i \le 2n$.
- Kasutaja $A$ avalik võti on $(y_1,...,y_{2n})$ ja salajane võti on $(k(1),...,k(2n))$.

Sõnumile $m$ signatuuri moodustamiseks teeb $A$ järgmist.

- Arvutab sõnumilühendi $h(m)$.
- Arvutab $s_i = E_{k(i)}(h(m))$, $1 \le i \le 2n$.
- Sõnumi m signatuur on $(s_1,...,s_{2n})$.

Kasutaja $A$ signatuuri verifitseerimiseks peab kasutaja $B$ sooritama järgmised sammud.

- Saab autentsel viisil kasutaja $A$ käest tema avaliku võtme $(y_1,...,y_{2n})$.
- Arvutab $h(m)$.
- Valib $n$ erinevat juhuslikku arvu $r(j)$, $1 \le r(j) \le 2n$, $1 \le j \le n$.
- Küsib kasutajalt $A$ võtmed $k(r(j))$, $1 \le j \le n$.
- Verifitseerib saadud võtmete autentsuse, arvutades $z_j = E_{k(r(j))}(r(j))$ ja kontrollides, kas $z_j = y_{r(j)}$, iga $1 \le j \le n$ korral.
- Kontrollib, kas $s_{r(j)} = E_{k(r(j))}(h(m))$, $1 \le j \le n$.

Rabini skeemi saab kasutada näiteks kahepoolse lepingu signeerimiseks. Rohkem osapooli kaasata ei saa, sest verifitseerimisel ei tohi $A$ võtmeid "reeta" rohkem kui $n$ tükki, muidu saaks signatuuri võltsida. Kasutajate $A$ ja $B$ hilisema vaidluse ($A$ eitab signeerimist) lahendamisel sooritatakse järgmised sammud.

1. Kasutaja $B$ annab usaldatavale kolmandale osapoolele (TTP) sõnumi $m$ ja signatuuri $(s_1,...,s_{2n})$.
2. TTP küsib $A$ käest võtmed $k(1),...,k(2n)$.
3. TTP verifitseerib kõigi võtmete autentsuse, arvutades $z_i = E_{k(i)}(i)$ ja kontrollides, kas $z_i = s_i$, $1 \le i \le 2n$. Kui mõni neist võrdustest ei kehti, tunnistab TTP signatuuri õigeks.
4. TTP arvutab $u_i = E_{k(i)}(h(m))$, $1 \le i \le 2n$. Kui $u_i = s_i$ mitte rohkem kui $n$-l juhul, tunnistab TTP signatuuri võltsituks, kui aga see võrdus kehtib indeksi $i$ vähemalt $n+1$ erineva väärtuse korral, siis tunnistab TTP signatuuri õigeks.

Kui $B$ soovib võltsida $A$ signatuuri mingile uuele sõnumile $m'$, peab ta kas teada saama vähemalt ühe lisavõtme $k(i)$ või valima $m'$ nii, et $h(m)=h(m')$. See on aga võimatu, kui krüptoalgoritm ja räsifunktsioon on turvalised. Kui aga $A$ soovib moodustada digitaalsignatuuri, mida hiljem on võimalik eitada, peab $A$ hoolitsema selle eest, et $u_i = s_i$ täpselt $n$ juhul ja lootma, et $B$ valib verifitseerimise ajal kogmata just nimelt needsamad indeksid; selle tõenäosus on aga

$$1/C^n_{2n} < 2^{-n},$$

kus $C^n_{2n}$ on Newtoni binoomkordajad. Ühe võtmega saab signeerida ainult ühe sõnumi, sest muidu reedaks $A$ teistele kasutajatele $n+1$ võtit, misjärel oleks kasutajal $B$ võimalik $A$ signatuure võltsida.

## 10.4 Pimesigantuurid

*Pimesignatuur (blind* signature) on teatav protokoll suhtluseks kahe osaleja, saatja *A* ja signeerija *B* vahel. Põhiidee on järgmine.

- *A* saadab *B*-le mingi bitijada, mille *B* signeerib ja *A*-le tagasi saadab.
- *A* tuletab saadud signatuurist mingi eelnevalt fikseeritud sõnumi *m* signatuuri $S_B(m)$.
- *B* ei tea ei pärast protokolli täitmise lõppu ei sõnumit *m* ega signatuuri $S_B(m)$.

Pimesignatuuri otstarve on takistada *B*-l seostamast konkreetse sõnumi signeerimist saatjaga *A*; see on vajalik näiteks digitaalraha anonüümsuse tagamiseks.

**Näide.** Chaumi pimesignatuuri skeem. Saatja *A* võtab signeerijalt *B* pimesignatuuri sõnumile *m*, mis on esitatud arvuna $0 \le m < n$. Olgu kasutajal *B* RSA avalik võti $(e,n)$ ja vastav salajane võti $d$.

- Saatja *A* valib juhusliku arvu $0 \le k < n$.
- Saatja *A* arvutab $m' = mk^e \bmod n$ ja saadab selle signeerijale *B*.
- *B* arvutab $s' = (m')^d \bmod n$ ja saadab selle tagasi saatjale *A*.
- *A* arvutab $s = k^{-1}s' \bmod n = m^d \bmod n$.

## 10.5 Vaidlustamatud digitaalsignatuurid

Vaidlustamatud digitaalsignatuurid (*undeniable signatures*) erinevad tavalistest signatuuridest selle poolest, et verifitseerimisprotseduuri saab teostada üksnes signeerija osalusel.

**Näide 1**. Pank *B* võtab kliendilt *A* vaidlustamatu digitaalsignetuuri dokumendile, mis väidab, et *A* kasutas teatud ajal panga turvalist hoiuruumi. Pangal ei ole võimalik ilma *A* enda osaluseta kellelegi tõestada, et just *A* kasutas hoiuruumi.

**Näide 2**. Tarkvara tootev suurfirma *A* müüb tarkvarapaketi kasutajale *B*. Tarkvarapakett on signeeritud vaidlustamatu digitaalsignatuuriga, mis on vajalik tarkvara autentsuse kontrolliks. Kui *B* tahab paketti ebaseaduslikult edasi müüa kolmandale isikule *C*, on tal kaks võimalust.

1. *B* ei modifitseeri paketti ega selle signatuuri. Sel juhul ei saa *C* kontrollida tarkvara autentsust ilma *A* osaluseta ja seetõttu on oht, et *B* pettus tuleb avalikuks.
2. *B* asendab firma *A* signatuuri omaenda signatuuriga. Sel juhul ei saa ta aga kasutada firma *A* nime ja turupositsiooni.

Järgnevas vaatleme Chaumi ja van Antwerpeni poolt välja pakutud vaidlustamatu digitaalsignatuuri skeemi.

**Võtme genereerimine**. Iga kasutaja *A* genereerib oma salajase ja avaliku võtme.

1. Valida juhuslik algarv $p=2q+1$, kus $q$ on ka algarv.
2. Valida selline juhuslik $\alpha$, et $1<\alpha<p$ ja $\alpha^q \bmod p = 1$. Selleks leida suvaline $1<\beta<p-1$ ja arvutada seejärel $\alpha=\beta^2 \bmod p = \beta^{(p-1)/q} \bmod p$.
3. Valida suvaline täisarv $a \in \{1,2,...,q-1\}$ ja arvutada $y = \alpha^a \bmod p$.
4. Avalik võti on $(p,\alpha,y)$ ja privaatvõti on $a$.

**Signeerimine**. Kasutaja *A* signeerib sõnumi *m*, mis kuulub jäägiklasside multiplikatiivsesse rühma $\boldsymbol{Z}_p^*$, st $m^q \bmod p = 1$.

1. Kasutaja *A* arvutab $s=m^a \bmod p$.
2. Sõnumi *m* signatuur on *s*.

**Verifitseerimine**. Kasutaja *B* verifitseerib kasutaja *A* signatuuri.

1. *B* valib juhuslikult kaks täisarvu $x1,x2 \in \{1,2,...,q-1\}$.
2. *B* arvutab $z = s^{x1}y^{x2} \bmod p$ ja saadab arvu *z* kasutajale *A*.
3. *A* arvutab $w=za' \bmod p$, kus $aa' \bmod q = 1$ ja saadab arvu *w* kasutajale *B*.
4. *B* arvutab $w'=m^{x1}\alpha^{x2} \bmod p$ ja kui $w=w'$, tunnistatakse signatuur autentseks.

Signatuuri verifitseerimine toimib, sest

$$w = za' = (s^{x1}y^{x2})a' = (m^{a \cdot x1}\alpha^{a \cdot x2})a' = m^{x1}\alpha^{x2} = w \bmod p.$$

Võltsitud signatuur tunnistatakse verifitseerimisel õigeks tõenäosusega $1/q$, mis on küllalt väike, kui algarv $q$ on piisavalt suur. Signeerijal *A* on võimalik eitada õiget signatuuri ühel järgmistest viisidest:

- *A* keeldub verifitseerimisprotseduuris osalemast;
- *A* ei täida verifitseerimisprotseduuri korrektselt;
- *A* eitab signeerimist isegi siis, kui verifitseerimisprotseduur on korrektselt läbitud.

Esimene võimalus ei tekita mingit segadust: *A* püüab verifitseerimisest kõrvale hoida, järelikult püüab ta eitada õiget signatuuri. Kaks järgmist võimalust on raskemad ja nõuavad nn eitusprotokolli (*disavowal protocol*).

**Eitusprotokoll**. Tehakse kindlaks, kas signatuur on tõepoolest võltsing või salgab *A* õiget signatuuri.

1. *B* valib kaks juhuslikku täisarvu $x1,x2 \in \{1,2,...,q-1\}$.
2. *B* arvutab $z = s^{x1}y^{x2} \bmod p$ ja saadab arvu *z* kasutajale *A*.
3. *A* arvutab $w=za' \bmod p$, kus $aa' \bmod q = 1$ ja saadab arvu *w* kasutajale *B*.
4. Kui $w=m^{x1}\alpha^{x2} \bmod p$, tunnistab *B* signatuuri autentseks.
5. *B* valib kaks juhuslikku täisarvu $x1',x2' \in \{1,2,...,q-1\}$.
6. *B* arvutab $z' = s^{x1'}y^{x2'} \bmod p$ ja saadab arvu *z'* kasutajale *A*.
7. *A* arvutab $w'=z'a' \bmod p$, kus $aa' \bmod q = 1$ ja saadab arvu *w'* kasutajale *B*.
8. Kui $w'=m^{x1'}\alpha^{x2'} \bmod p$, tunnistatab *B* signatuuri autentseks.
9. Kasutaja *B* arvutab $c=(w\alpha^{-x2})^{x1'} \bmod p$ ja $c'=(w'\alpha^{-x2'})^{x1} \bmod p$. Kui $c=c'$, järeldab *B*, et signatuur *s* on võltsing, vastasel korral järeldab *B*, et *A* üritab salata õiget signatuuri.

Kui s on võltsing, st $s \neq m^a \bmod p$, ja kui *A* täidab eitusprotokolli korrektselt, on protokolli täitmise lõpuks $w=w'$ ja seega on *B* järeldus korrektne. Kui *s* on õige signatuur, st $s=m^a \bmod p$ ning *A* ei täida protokolli korrektselt, on võrduse $w=w'$ kehtimise tõenäosus vaid $1/q$.

Oletame, et *B* salvestab verifitseerimisprotokolli täitmise käigus vahetatud sõnumid ja väärtused $x1$ ja $x2$. Kolmas osapool *C* ei saa aga seda informatsiooni vaadelda kui tõestust, et *A* on midagi signeerinud, sest kasutajal *B* on alati võimalus ilma *A* osaluseta tekitada kõik vajalikud sõnumid. Kasutaja *B* valib ühe sõnumi *m*, arvud $x1$ ja $x2$ ja l ning arvutab $s=(m^{x1}\alpha^{x2})^{l'}y^{-x2})^{x1'}$, kus $l \cdot l' \equiv x1 \cdot x1' \equiv 1 \pmod q$. Verifitseerimisalgoritm tunnistab signatuuri *s* autentseks. Seega ei saa *B* koguda verifitseerimise käigus mingit tõestusmaterjali, mis ongi vaidlustamatute digitaalsignatuuride peamine iseärasus.

# 11 KRÜPTOGRAAFILISED PROTOKOLLID

Selles peatükis vaadeldakse kaht olulist krüptograafia valdkonda. Esiteks, enne suhtluse alustamist peavad osapooled vahetama salajase võtme, kasutades nn võtmekehtestusprotokolle. Teiseks, siiani tutvustatud krüptograafilistest primitiividest ei piisa veel tegeliku elu vajaduste (elektronraha, elektroonilised valimised jms) rahuldamiseks. Selliste vajaduste tarbeks on välja töötatud palju spetsiifilisi protokolle, milledest mõned saavutavad uskumatult keerukaid turvaeesmärke.

## 11.1 Võtmekehtestusprotokollid

### 11.1.1 Diffie-Hellmani võtmekehtestusprotokoll

Jaotises 9.1.7 kirjeldati juba lühidalt Diffie-Hellmani võtmekehtestusprotokolli (DH SKE, *Diffie-Hellman Secret Key Exchange*). Järgnevalt vaatleme seda protokolli lähemalt.

#### *11.1.1.1 Protokoll*

DH võtmekehtestusprotokolli aluseks on avalikult fikseeritud algarv $p$ ning multiplikatiivse rühma $\mathbb{Z}_p^*$ moodustaja $g$. Esimese sammuna valib Alice juhuslikult elemendi $x \in \mathbb{Z}_{p-1}$ ning saadab Bobile väärtuse $X=g^x \bmod p$. Bob valib juhuslikult elemendi $y \in \mathbb{Z}_{p-1}$ ning saadab Alice'ile väärtuse $Y=g^y \bmod p$. Olgu $K:=X^y=(g^x)^y=g^{xy}=(g^y)^x=Y^x$ (kõik tehted toimuvad rühmas $\mathbb{Z}_p^*$). Kuna Alice teab suurusi $x$ ja $Y$ ning Bob teab suurusi $y$ ja $X$, on nad mõlemad suutelised väärtust $K$ välja arvutama ning seega jagavad võtit.

#### *11.1.1.2 Turve pealtkuulamise eest: DH probleem*

Vaatleme esmalt DH protokolli turvalisust passiivse ründaja (Eve) suhtes, kes näeb protokollivoogu, kuid ei muuda midagi. Protokolli täitmise kestel näeb Eve väärtusi $X$ ja $Y$, kuid ta ei tea väärtusi $x$ ja $y$. Suuruse $K$ leidmiseks on kõige loomulikum rünne $x$ või $y$ leidmine, seesjärel oleks $K$ arvutamine juba triviaalne. Kuid $X$ järgi $x$ ($Y$ järgi $y$) leidmine on võrdväärne (eeldatavalt raske) diskreetse logaritmi probleemi (DL probleemi) lahendamisega.

**DH probleemiks** nimetatakse sellist probleemi: juhuslikult valitud $x$ ja $y \in \mathbb{Z}_{p-1}$ korral leida suurustest $g^x$ ja $g^y$ suurus $g^{xy}$. Seega on DH protokolli murdmiseks vaja lahendada DH probleem. Eespool nägime, et DL probleem on raskem kui DH probleem: kui oskame leida diskreetseid logaritme, oskame ka murda DH protokolli. Vastupidist ei ole siiani suudetud tõestada. Tänapäeval usutakse üldiselt, et ka DH probleem (kui ka mitte võrdväärne DL probleemiga) on arvutuslikult raske ning seetõttu on ka DH protokoll turvaline arvutuslikult piiratud passiivse ründaja suhtes.

**DH eeldus**: DH probleem on arvutuslikult raske.

Algarvu $p$ suurus peab olema vähemalt 512 bitti, soovitatavalt 1024. Et DL probleem olek raske, peab arvul $p-1$ olema vähemalt üks suur algarvuline tegur. Praktikas võetakse sageli $p=2q+1$, kus $q$ on samuti suur algarv.

#### *11.1.1.3 Diffie-Hellmani krüptosüsteem*

Olgu $p$ ja $g$ fikseeritud nagu eespool. Osapoolel $A$ on salajaseks võtmeks juhuslikult valitud $SK(A) \in \mathbb{Z}_{p-1}$ ning avalikuks võtmeks $PK(A)=g^{SK(A)}$. Alice ja Bob lepivad esmalt kokku kasutatavas sümmeetrilises krüptosüsteemis (olgu selleks näiteks IDEA) ning võtme ekstraheerimisalgoritmis (näiteks võtme esimesed 128 bitti). Salajase teate $M$ saatmiseks Alice'ile valib Bob juhuslikult arvu $y \in \mathbb{Z}_{p-1}$ ning leiab DH võtme $K=PK(A)^y=g^{SK(A)y}$. Saadud võtmest $K$ ekstraheerib Bob ekstraheerimisalgoritmi kasutades 128-bitise IDEA võtme $k$. Seejärel edastab Bob Alice'ile paari $(g^y, \text{IDEA}(k,M))$.

Kui Alice saab kätte paari $(Y,C)$, leiab ta oma salajast võtit $SK(A)$ kasutades DH võtme $K=Y^{SK(A)}=g^{ySK(A)}$ ning leiab ekstraheerimisalgoritmi kasutades võtme $k$. Seejärel dekrüpteerib Alice krüptogrammi $C$, kasutades võtit $k$.

#### *11.1.1.4 DH võtme bititurvalisus*

Eelmises jaotises kasutati võtme $K=g^{xy}$ esimest 128 bitti sümmeetrilise krüptosüsteemi võtmena. DH eeldus ütleb, et teades väärtusi $g^x$ ja $g^y$, ei ole ründaja võimeline leidma võtit $K$. Sellest ei piisa DH krüptosüsteemi turvalisuseks: mis siis, kui ründaja suudab leida võtme $k$ esimesed 128 bitti isegi siis, kui DH eeldus kehtib? Seega ei piisa DH eeldusest DH krüptosüsteemi turvalisuseks, lisaks tuleb eeldada et funktsioonil $f(x,y)=g^{xy}$ leiduvad nn tugevad bitid (*hardcore bits*), või üldisemalt leidub teatud *tugev kujutus* HCF: $\mathbb{Z}_p^* \rightarrow \mathbb{Z}_{128}$, mille väärtust leida on "niisama" raske kui suurust $g^{xy}$.

On teada, et kujutus LSB, LSB: $x \rightarrow x \bmod 2$ ($x$-i madalaima biti leidmine) ei ole tugev. Tänase päevani ei teata teisi mittetriviaalseid nõrku kujutusi, sellest hoolimata ei soovitata kasutada DH võtme bittide mingit alamhulka sümmeetrilise krüptosüsteemi võtmena, kuna selliste bittide turvalisuse nõue näib olevat tunduvalt tugevam DH eeldusest. Praktikas soovitatakse tihti võtta $k=H(K)$, kus $H$ on mingi turvaline räsifunktsioon, näiteks SHA-1. Kkasutamine sellises kontekstis seab aga räsifunktsioonile väga kõrged pseudojuhuslikkuse nõuded.

Üldine metoodika tugevate kujutuste leidmiseks on järgmine. Olgu $r$ juhuslikult valitud bitistring pikkusega $|p|$ ning olgu $b$ bitistringide $K$ ja $r$ skalaarkorrutis. On teada, et $b$ leidmine arvudest $g^x$ ja $g^y$ on trakteerimatu (*intractable*), kui kehtib DH eeldus. Selle lähenemise puuduseks on ebaefektiivsus: 128 biti saamiseks tuleb vahetada mitu DH võtit ning ekstraheerida igast DH võtmest mõned bitid.

1996. aastal tõestasid Dan Boneh (Princeton University) ja Ramarathnam Venkatesan (Bellcore), et DH võtmekehtestusprotokolli puhul on võtme $K$ vähemalt $O(\sqrt{(\log p)})$ kõrgeima biti leidmine osapoolte avalikest võtmetest niisama raske kui võtme $K$ enda leidmine. Seega, kui $p$ on 1024-bitine algarv, on vähemalt 32 kõrgeima biti leidmine teostamatu. Nad esitasid ka sellise variandi DH võtmeedastusprotokollist ning krüptosüsteemist, kus ühe kõrgeima biti leidmine on teostamatu lisaeeldusel, et suurustest $g$ ja $g^y$ väärtuse $2^y$ leidmine on teostamatu. Selle nn BV eelduse täpne seos DH eeldusega ei ole veel teada.

Boneh-Venkatesani võtmekehtestusprotokolli esimese sammuna valib Alice juhuslikult elemendi $x \in \mathbb{Z}_{p-1}$, nii et SÜT($x$,$p$–1)=1 ja $g^x=2 \bmod p$, ning saadab Bobile väärtuse $X=g^x \bmod p$. Bob valib juhuslikult elemendi $y \in \mathbb{Z}_{p-1}$ ning saadab Alice'ile väärtuse $Y=g^y \bmod p$. Olgu $K:=2^y=Y^z$, kus $z=x^{-1} \bmod p$. Kuna Alice teab suurusi $Y$ ja $x$ ning Bob teab suurust $y$, on nad mõlemad suutelised väärtust $K$ välja arvutama ning seega jagavad ka võtit.

1997. aastal tõestasid Boneh ja Venkatesan veel, et DH võtmekehtestusprotokolli puhul on võtme 2log log $p$ kõrgeima biti leidmine osapoolte avalikest võtmetest niisama raske kui võtme $K$ enda leidmine.

#### *11.1.1.5 Autentsuse puudumine*

DH protokolli nõrkuseks on tema rünnatavus aktiivse ründaja poolt: Eve maskeerub Alice'iks ning algatab Bobiga protokolli. Protokolli täitmise lõpuks jagavad Eve ja Bob salajast võtit, kusjuures Bobi arvates jagab ta võtit Alice'iga. Lähtudes sellest arvamusest edastab Bob kehtestatud võtit kasutades Evele konfidentsiaalseid andmeid. Kuigi DH protokollist eraldi ei piisa autentsuse tagamiseks, on ta siiski oluline vahend uute, täiuslikumate protokollide moodustamisel.

### 11.1.2 Seansivõtme kehtestamine. Usaldusmudelid

Järgnev annab ülevaate sellest, kuidas saavutada võtmekehtestusprotokollide juures nii autentsust kui konfidentsiaalsust. Aktiivne ründaja saab teatavasti kanalisse kirjutada, sh manipuleerida teadetega, mida on saatnud seaduslikud kasutajad.

Nagu juba mainitud jaotises 9.7.3, on "klassikalises" maailmas võimatu salajase ühisteadmuse ("salajase võtme") teke kahe osapoole vahel ilma eelneva salajase ühisteadmuseta. Niisiis, et salajane suhtlus oleks üldse võimalik, peab osapooltel olema nn "teabe-eelis" (*information advantage*): mingi eelnevalt (turvatud) kanali kaudu edastatud ühisteave, mis võimaldab teostada edasisi võtmekehtestusi. Vaatleme järgnevalt lühidalt eri võimalusi sellise "teabe-eelise" saavutamiseks.

#### *11.1.2.1 Kolme osapoolega mudel*

Kolme osapoolega mudelit mainisid esimesena ilmselt Needham ja Schroeder (1978), laialdast kasutust on see mudel leidnud näiteks Kerberose süsteemis. Selles mudelis on olemas kolmas usaldatav osapool *S* (autentimisserver). Süsteemi igal osapoolel *A* on serveriga jagatav salajane võti $K_A$. Kui kaks osapoolt *A* ja *B*, kes jagavad serveriga vastavalt võtmeid $K_A$ ja $K_B$, soovivad alustada omavahelist salajast suhtlust, hakkavad nad serveri osavõtul täitma kolmepoolset protokolli, mille tulemuseks on *A* ja *B* vaheline salajane võti $K_{AB}$. Kehtestatud võtit kasutatakse *A* ja *B* vahelises suhtlusseansis, pärast seansi lõppu "visatakse" võti ära. Hiljem, kui samad osapooled soovivad uuesti suhelda, viiakse kolmepoolne protokoll uuesti läbi.

**Kahe osapoolega asümmeetriline mudel**

Juhul, kui on võimalus kasutada avaliku võtmega krüptograafiat, võib autentimisserveri aktiivse rolli elimineerida. Niisuguses usaldusmudelis eeldatakse, et *A* teab *B* avalikku võtit *PK*(*B*) ning *B* teab *A* avalikku võtit *PK*(*A*). Eeldatakse, et need võtmed on autentsed, st et ühele osapoolele teadaolev võti vastab tõepoolest teise osapoole avalikule võtmele.

Alice ja Bob soovivad edasises konfidentsiaalselt suhelda, kasutades mainitud teabe-eelist. Esmapilgul näib, et siin ei ole probleeme: Alice krüpteerib teate *M* võtmega *PK*(*B*) ning saadab saadud krüptogrammi Bobile. Selline triviaalne lahendus ei sobi vähemalt kahel olulisel põhjusel. Esiteks on (vähemalt tänapäeva krüptograafia-alase teadmuse juures) avaliku võtmega krüptograafia tuhandeid kordi aeglasem kui sümmeetriline krüptograafia. Veelgi olulisem põhjus on aga vajadus omada eraldi lühiajalisi seansivõtmeid, mille kompromiteerumine ei rikuks süsteemi pikaajalist turvalisust. Pikaajalist salajast võtit *SK(A)* võib hoida turvalises riistvaras.

**Kahe osapoolega sümmeetriline mudel**

Kõige lihtsamas mudelis jagavad kaks osapoolt pikaajalist võtit. Iga kord, kui nad soovivad alustada suhtlusseanssi, tuletavad nad mingi protokolli abil pikaajalisest võtmest seansivõtme. Siingi on motivatsiooniks seansivõtmete kasutamisest saadav täiendav turvalisus; nii võib süsteemil (näiteks konkreetsel kohtvõrgul) olla üks pikaajaline võti ning süsteemi paljud kasutajad kehtestavad iga suhtlusseansi ajal eraldi seansivõtme.

### 11.1.3 Võtmekehtestusprotokollide ajalugu

Ehkki võtmekehtestusprotokollidel on pikk ajalugu, on alles hiljuti õpitud neid koostama tõestatavalt turvaliselt. Tõestatav turvalisus tähendab protokollide puhul protokollide turvalisuse taandamist protokollis kasutatud krüptograafiliste primitiivide turvalisusele.

Enne tõestusmetoodikate kohta alal on avaldatud palju teaduslikke artikleid, kus on pakutud väidetavalt turvalisi protokolle; enamik neist protokollidest on hiljem osutunud suuremal või väiksemal määral vigasteks. On lihtne pakkuda protokolle, milles hiljem leitakse turvaauke.
Näiteks esitasid Needham ja Schroeder 1978. aastal rea kolmepoolseid protokolle. Neist esimeses leidsid juba 1981. aastal vea Denning ja Sacco. Protokolli on küll hiljem parandatud, kuid iga paari aasta järel tulevad ilmsiks uued rünnete tüübid, mille vastu eelmised parandused on jõuetud. Protokolli konstrueerimisel ei tohi niisiis rahulduda sellega, et protokoll on turvaline kõigi teadaolevate rünnete vastu, protokoll peab olema tõestatavalt turvaline kõigi *võimalike* rünnete vastu.

Burrows, Abadi ning Needham töötasid 1989. aastal välja spetsiaalse autentimisloogika (tuntud kui BAN-loogika), milles on lihtne arutleda eri osapoolte uskumuste üle. Sellises loogikas saab eeldustest nagu "$S(A)$ on $A$ salajane võti" ning "suurus $n$ on protokolli täitmise alguses värske" tuletada protokolli täitmise lõpuks järeldusi nagu "$A$ usub, et $K_{AB}$ on $A$ ja $B$ vaheline salajane võti." Hiljem on mitmed autorid (Abadi ja Tuttle 1991, Syverson ja van Oorschot 1996) täiendanud BAN-loogikat, sh laiendanud loogika käsitlusringi DH võtmekehtestusprotokollidele ning töötanud välja korraliku, vasturääkivusteta semantika.

Ehkki see lähenemine on olnud väga edukas (BAN-tüüpi loogikaid kasutades on murtud mitmeid varem turvaliseks peetud protokolle), ei saa veel nendes loogikates tuletatud protokolli korrektsuse tõestustest järeldada protokolli tegelikku turvalisust. Üks olulisemaid probleeme on turvalisuse mõiste semantika keerukus: ehkki uuemates BAN-tüüpi loogikates on semantika mittevasturääkiv, ei ole ta kaugeltki mitte täielik. Samuti tekib selle lähenemise juures probleeme, kui eeldatavalt turvalised primitiivid asendada tegelike primitiividega: loogiline lähenemine ei võimalda tekkinud ründeohte lihtsalt "summeerida." Teistest olulistest probleemidest mainitagu siinkohal niisuguste loogikate võimetust hinnata protokolli turvalisust täpselt (pakkuda alampiire võimalike rünnete edukusele) ning tõestada semantilist turvalisust (alajaotis 10.1.7.1).

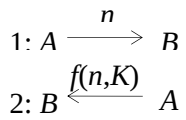
1991. aastal esitasid Bird, Gopal, Herzberg, Janson, Kutten, Molva ja Yung uue klassi nn vahendusründeid, kasutades neid teadaolevate protokollide ründamiseks. Ühtlasi esitasid nad uue protokolli (2PP), mis oli turvaline kõigi nende poolt vaadeldud rünnete vastu.

Toetudes sellele ning Diffie ja van Oorschoti tehtud tööle, defineerisid Bellare ja Rogaway täpse turvamudeli kahe (1993) ja kolme (1995) osapoolega seansivõtme kehtestusprotokollides. Sellest mudelist lähtudes ehitasid nad üles pere efektiivseid protokolle, mille turvalisust saab tõestada, lähtudes harilikest krüptograafilistest eeldustest.

Sama lähenemist rakendades on hiljem välja töötatud ka teisi tõestatavalt turvalisi protokolle, näiteks SKEME (Krawczyk, 1996) ning Shoup-Rubini protokoll. 1998. aastal esitasid Bellare, Canetti ja Krawczyk üldise protseduuri tõestatavalt turvaliste protokollide genereerimiseks.

### 11.1.4 Probleemi mitteformaalne kirjeldus

Tavaliselt kujutatakse protokolli nii nagu Joonis 73.




**Joonis 73. Kahe osapoolega protokoll**

Siin on $A$ ja $B$ *osapooled*, $n$ ja $f(n,K)$ on *teated*. Nooled näitavad andmevoogude suunda: esimese teate ajal saadab Alice Bobile muutuja $n$ ning teise teate ajal saadab Bob Alice'ile väärtuse $f(n,K)$. Analoogiliselt saab esitada enamiku krüptograafilisi protokolle.

Olgu $\{P_1,P_2,\ldots,P_N\}$ hajussüsteemi osapoolte hulk. Seansivõtmete kehtestamisel tuleb arvestada sellega, et iga osapool võib olla korraga hõivatud mitmes seansis: igal osapoolel on mitu (üksteisest rohkem või vähem sõltumatut) *isendit*. Seega ei ole võtmekehtestusprotokolli loogilisteks lõpppunktideks mitte osapooled $P_i$, vaid osapoolte isendid $\Pi(i,j,s)$ — osapool $P_i$ kes soovib end autentida osapoolele $P_j$ seansil $s$. Võtmekehtestusprotokolli eesmärk on kehtestada seansivõti $\sigma(i,j,s,t)$ isendite $\Pi(i,j,s)$ ja $\Pi(j,i,t)$ vahel, kusjuures võti peab saama kehtestatud ilma et mainitud isendid teaksid suurusi $s$ või $t$ või üleüldse midagi teiste seansside olemasolust.

Hajussüsteemi ründab aktiivne vastane *E*, kes kontrollib kogu suhtlust osapoolte vahel: *E* suudab teateid asendada, taasesitada, kustutada ning muuta.

### 11.1.5 Turvalisuse aspektid

Protokolli otstarve on võtit vahetada nii, et vastane ei suudaks kompromiteerida kahe seadusliku osapoole vahel kehtestatud võtit, mh peab võti olema autentne ning konfidentsiaalne. Esimene tingimust tähendab jämedalt öeldes, et kui osapoole $P_i$ isend aktsepteerib protokolli (loeb võtme kehtestatuks), on ta hiljuti suhelnud osapoole $P_j$ isendiga. Teine tingimus tähendab jämedalt öeldes, et kui $\Pi(i,j,s)$ ja $\Pi(j,i,t)$ jagavad seansivõtit, on see võti salajane.

Oluline on, et erinevad seansivõtmed oleksid üksteisest sõltumatud. Nõue on vajalik peamiselt seetõttu, et võtmekehtestusprotokolli koostamisel ei tohi teha eeldusi seansivõtmete hilisema käsitluse suhtes: kui üks seansivõti tuleb avalikuks, ei tohi see kompromiteerida teisi seansivõtmeid (tulevikusalastus). Bellare-Rogaway (BR) mudelis modelleeritakse ülaltoodud väidet, andes vastasele võimaluse soovi korral osapooli kompromiteerida ning seansivõtmeid küsida.

Võtit nimetatakse harilikult turvaliseks, kui volitamata osapooled ei suuda seda (tõenäosuslikus polünomiaalses ajas) arvutada. Sellest siiski ei piisa: on vaja vältida ka osalist informatsiooni leket. Seetõttu on konfidentsiaalsuse definitsioon rangem, nõudes, et seansivõti oleks ennustamatu (*unpredictable*) tõenäosuslikult krüpteeritud teate mõttes ("semantiline turvalisus").

Seansivõtme ebapiisav turvalisus (BR-mudelis) on omane enamikule võtmekehtestusprotokollidest. Ebaturvalisus on väga tihti protokolli sisse ehitatud, sest tavaliselt soovitakse, et protokoll teostaks ka võtmekinnituse (kinnituse, et võti on kätte saadud). Selleks krüpteerib üks osapooltest (*A*) harilikult mingi fikseeritud teate, kasutades värskelt kehtestatud võtit. Teine osapool tõlgendab seda kui fakti, et *A* teab seansivõtit. Bellare ja Rogaway väidavad, et sellise kinnitusega lekib võtme kohta osaline informatsioon ning ehkki see võib näida ebaoluline, on võimalik konstrueerida edasisi kehtestatud seansivõtit kasutavaid protokolle, mis on just tänu sellisele võtmekinnitusele ebaturvalised.

### 11.1.6 Olemi autentimine ja võtme kehtestamine

Võtmekehtestuse (*key distribution*) eesmärk on (1) autentida protokolli osapooled samaaegselt üksteisele ning (2) kehtestada osapoolte vahel salajane jagatud võti. Autentsuse mõistet võib siinjuures tõlgendada mitmeti. Väga tugeva autentsuse definitsiooni andsid Bellare ja Rogaway 1993. aastal; seda definitsiooni nõrgendati 1995. aastal kolme osapoole juhule kohandamisel. Kumba definitsiooni kasutada, sõltub olukorrast. Tavaliselt vaadeldakse tugevamat definitsiooni kahe ning nõrgemat definitsiooni kolme osapoole protokollide puhul. Üks põhjusi selleks on fakt, et kahe osapoole juhu jaoks pole nõrgemat definitsiooni veel suudetud anda.

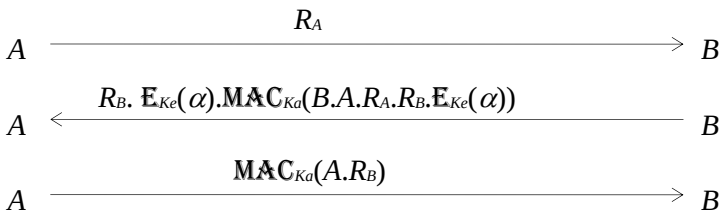
### 11.1.7 Autenditud võtmekehtestus

Vaatleme esmalt sümmeetrilist ja asümmeetrilist autenditud võtmekehtestust (*authenticated key exchange)* kahe osapoole juhul, toomata sisse formaalset mudelit ning turvalise autenditud seansivõtme kehtestusprotokolli täpset definitsiooni; järgnevalt esitame vaid mõned tõestatult turvalised protokollid.

Protokollide otstarve on niisiis kehtestada salajane $\ell$-bitine seansivõti, kus (näiteks) $\ell$=128. Kehtestatud seansivõtit tähistame $\alpha$. Kui osapool *A* edastab osapoolele *B* teate, eeldame vaikimisi, et teatega kaasneb ka saatja turvamata identiteet, st *B* teab näiteks, millist salajast võtit kasutada teadete dekrüpteerimiseks (kuna identiteet on turvamata, võib seda võltsida. Seega on antud juhul tegu vaid suhtluskeskkonna teenusega).

### *11.1.7.1 Sümmeetriline juht*

Olgu *K* osapoolte *A* ja *B* vaheline pikaajaline võti. Fikseerime kasutatava sümmeetrilise krüptosüsteemi ($\mathbf{E}$,$\mathbf{D}$) (siin on $\mathbf{E}$ krüpteerimisfunktsioon ning $\mathbf{D}$ on dekrüpteerimisfunktsioon) ning sõnumiautentimiskoodi ($\mathbf{MAC}$,$\mathbf{VF}$) (kus $\mathbf{VF}$ on verifitseerimisfunktsioon: $\mathbf{VF}(x,y)=1$ parajasti siis, kui $\mathbf{MAC}(x)=y$). Võti *K* on jaotatud kaheks osaks, $K_e$ (krüpteerimisvõti) ning $K_a$ (autentimisvõti). *A*.*B* tähistab stringide *A* ja *B* konkatenatsiooni. Protokolli AKEP1 (*Authenticated Key Exchange Protocol* 1) illustreerib j.




**Joonis 74. Protokoll AKEP1**

Protsess on alljärgnev.

1. *A* genereerib juhusliku stringi $R_A$ ning saadab selle *B*-le.
2. *B* genereerib juhusliku stringi $R_B$ ning $\ell$-bitise seansivõtme $\alpha$. Ta krüpteerib $\alpha$, kasutades pikajalist võtit $K_e$ ja saab krüptogrammi $c=\mathbf{E}_{Ke}(\alpha)$. Seejärel arvutab *B* sildi (*tag*) $\mu=\mathbf{MAC}_{Ka}(B.A.R_A.R_B.\mathbf{E}_{Ke}(\alpha))$ ning saadab *A*-le konkatenatsiooni $R_B.c.\mu$.
3. *A* verifitseerib, kas $\mathbf{VF}_{Ka}(B.A.R_A.R_B.c,\mu)=1$. Kui verifitseerimine õnnestub, arvutab *A* sildi $\mathbf{MAC}_{Ka}(A.R_B)$ ning saadab selle *B*-le. *A* leiab seejärel seansivõtme $\alpha=\mathbf{D}_{Ke}(c)$.
4. *B* verifitseerib viimase sildi ning aktsepteerib võtme (loeb seansivõtme $\alpha$ kehtivaks), kui verifitseerimine õnnestub.

Üks tavalisi vigu autentimisprotokollide koostamisel on krüpteerimise kasutamine autentimiseks. Autentimiseks tuleb kasutada sõnumiautentimiskoodi ning autentimisvõti peab erinema krüpteerimisvõtmest.

Oluline on ka see, et kasutatud krüptograafilised primitiivid ise oleksid "väga" turvalised. Krüpteerimisfunktsiooni ($\mathbf{E}$,$\mathbf{D}$) korral nõutakse ***semantilist*** turvalisust: st, et tõenäosuslikus polünomiaalses ajas töötav vastane ei suudaks fikseeritud stringiga krüptogramme eristada niisama pikkade juhuslikult valitud stringide krüpteerimisel saadud krüptogrammidest. Formaalselt, vaatleme järgmist kaht "maailma".

**Maailm 0.** Vastasele antakse "must kast" $\mathbf{E}_K(\cdot)$ arvutamiseks. Vastane valib teate *M* ning laseb mustal kastil leida väärtuse $\mathbf{E}_K(M)$. Must kast on tõenäosuslik: igal väljakutsel ta n-ö viskab münte, genereerides juhusliku stringi *r*. Väärtus $\mathbf{E}_K(M)$ sõltub ka valitud stringist *r*. Muuhulgas on vähe tõenäoline, et sisestades kasti kaks korda järjest sama stringi *M*, saadakse tagasi täpselt sama krüptogramm (tõenäosuslik krüpteerimine). Vastane ei tea musta kasti poolt visatud müntide väärtusi.

**Maailm 1.** Vastasele antakse must kast, mis sisendandmeteks stringi *M* saamisel valib juhuslikult stringi *X* pikkusega $|M|$ ning tagastab väärtuse $\mathbf{E}_K(X)$.

Vastasele *A* on antud "oraakel", mis käitub ühel kahest ülaltoodud viisist. Vastase ülesanne on pärast oraaklile päringu esitamist olla suuteline otsustama, kumba tüüpi on oraakel. Olgu $P(i,A)$ tõenäosus, et vastane ütleb "1" maailmas *i*. Olgu $\mathbf{Adv}_A:=|P(1,A)-P(0,A)|$.

**Definitsioon.** Krüptoskeem ($\mathbf{E}$,$\mathbf{D}$) on ***semantiliselt turvaline***, kui suvalise tõenäosuslikus polünomiaalses ajas töötava vastase *A* jaoks on $\mathbf{Adv}_A$ kaduvväike suurus (st väiksem kui $k^{-c}$ suvalise konstandi $c>0$ korral, kus *k* on krüptoskeemiga seotud ***turvaparameeter*** (näiteks salajase võtme pikkus).

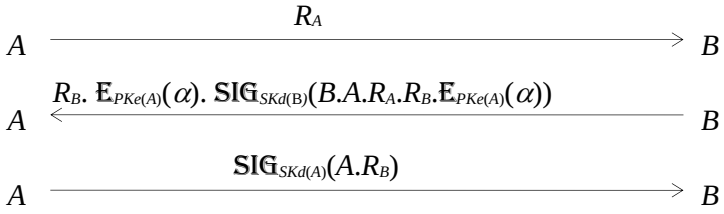
Ka alljärgnevate protokollide tõestatavaks turvalisuseks peab valitud krüptoskeem olema semantiliselt turvaline.

### *11.1.7.2 Asümmeetriline juht*

Selles alajaotises kasutame avaliku võtmega krüptograafiat – nii asümmeetrilist krüpteerimist kui ka digitaalsignatuure. Olgu ($\mathbf{E}$,$\mathbf{D}$) asümmeetriline krüptosüsteem, $\mathbf{E}_{PKe}(M)$ on teate *M* krüptogramm, kus *Pke* on avalik krüpteerimisvõti ning $\mathbf{D}_{SKe}(C)$ on krüptogrammile *C* vastav avatekst, kus *Ske* on salajane dekrüpteerimisvõti. Sealjuures eeldatakse, et ($\mathbf{E}$,$\mathbf{D}$) on semantiliselt turvaline krüptosüsteem. Fikseeritakse ka signatuuriskeem ($\mathbf{SIG}$,$\mathbf{VF}$), kus $\mathbf{SIG}_{SKd}(\cdot)$ on signeerimisfunktsioon ning $\mathbf{VF}_{PKd}(\cdot)$ on sellele vastav verifitseerimisfunktsioon. Ka signatuuriskeemilt nõutakse semantilist turvalisust (asümmeetrilise krüptosüsteemi ning signatuuriskeemi semantilise turvalisuse definitsioon erineb alajaotises 10.1.7.1 antud definitsioonist, kuid definitsiooni aluseks olev idee on sama).

Hajussüsteemi igal osapoolel $P_i$ on avalik võti $PK(i)=(Pke(i),PKd(i))$, mis on teada süsteemi kõigile osapooltele ning ka vastasele. Vastav salajane võti $SK(i)=(Ske(i),SKd(i))$ on teada vaid igale osapoolele endale.

Protokolli täitmise alguses teab *A* avalikku võtit *PK*(*B*) ning *B* teab *A* avalikku võtit *PK*(*A*). Protokolli otstarve on *A* ja *B* vahelise ühise salajase võtme $\alpha$ kehtestamine. Protokoll ise on järgmine (vt joonis 83).



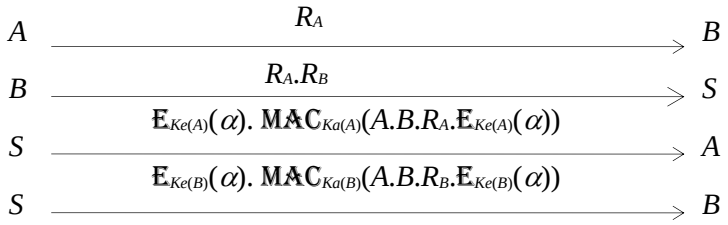

**Joonis 75. Asümmeetrilise süsteemi võtmekehtestusprotokoll**

Protsess on alljärgnev.

1. *A* genereerib juhusliku stringi $R_A$ ning saadab selle *B*-le.
2. *B* genereerib juhusliku stringi $R_B$ ning $\ell$-bitise seansivõtme $\alpha$. *B* krüpteerib $\alpha$, kasutades avalikku võtit S*Ke*(*A*) ja saab krüptogrammi $c=\mathbf{E}_{SKe(A)}(\alpha)$. Seejärel arvutab *B* signatuuri $\mu=\mathbf{SIG}_{SKd(B)}(B.A.R_A.R_B.\mathbf{E}_{PKe(A)}(\alpha))$, kasutades oma salajast võtit *SKd*(*B*), ning saadab konkatenatsiooni $R_B.c.\mu$ osapoolele *A*.
3. *A* verifitseerib, kas $\mathbf{VF}_{PKd(B)}(B.A.R_A.R_B.c,\mu)=1$. Kui verfitseerimine õnnestub, arvutab *A* signatuuri $\mathbf{MAC}_{SKd(A)}(A.R_B)$ ning saadab selle *B*-le. *A* arvutab seejärel seansivõtme $\alpha=\mathbf{D}_{SKe}(c)$.
4. *B* verifitseerib viimase signatuuri ning aktsepteerib võtme (loeb seansivõtme $\alpha$ kehtivaks), kui verifitseerimine õnnestub.

## 11.1.8 Kolme osapoolega seansivõtme kehtestus

Protokolli täitmise alguseks on fikseeritud semantiliselt turvaline sümmeetriline krüptoskeem ($\mathbf{E}$,$\mathbf{D}$) ning semantiliselt turvaline sõnumiautentimiskood ($\mathbf{MAC}$,$\mathbf{VF}$). Server *S* jagab protokolli alguses iga osapoolega *P* salajast võtit $K(P)=(Ke(P),Km(P))$, kus esimene võti on sümmeetrilise krüptoskeemi jaoks ning teine võti MAC-i jaoks. Järgneva protokolli otstarve on kehtestada *A* ja *B* vahel serveri genereeritud turvaline võti $\alpha$:




**Joonis 76. Kolmepoolne protokoll**

Protsess on alljärgnev.

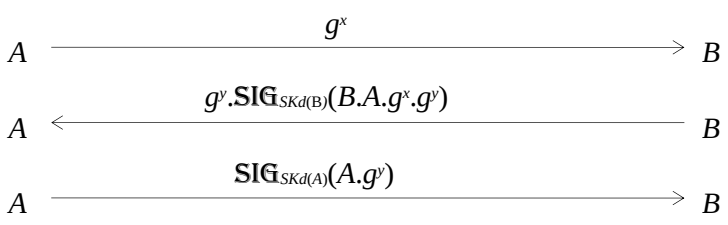
1. Sammul 1 valib *A* juhusliku stringi $R_A$ ning saadab selle *B*-le.
2. Sammul 2 valib *B* juhusliku stringi $R_B$ ning saadab konkatenatsiooni $R_A.R_B$ *S*-ile.
3. Server *S* genereerib juhusliku uue salajase võtme $\alpha$ ning krüpteerib selle, kasutades mõlema osapoolega jagatavat salajast võti, leides $\alpha_A=\mathbf{E}_{Ke(A)}(\alpha)$ ning $\alpha_B=\mathbf{E}_{Ke(B)}(\alpha)$. Peale selle arvutab *S* sildid $\mu_A=\mathbf{MAC}_{Ka(A)}(A.B.R_A.\mathbf{E}_{Ke(A)}(\alpha))$ ja $\mu_A=\mathbf{MAC}_{Ka(B)}(A.B.R_B.\mathbf{E}_{Ke(B)}(\alpha))$ ning saadab osapoolele *A* teate $\alpha_A.\mu_A$ ja osapoolele *B* teate $\alpha_B.\mu_B$.
4. Saanud teate $\alpha'_A.\mu'_A$ (vastavalt $\alpha'_B.\mu'_B$), aktsepteerib *A* (vastavalt *B*) seansivõtme $\mathbf{D}_{Ke(A)}(\alpha'_A)$ (vastavalt $\mathbf{D}_{Ke(B)}(\alpha'_B)$) parajasti siis, kui $\mathbf{VF}_{Ka(A)}(A.B.R_A.\alpha'_A, \mu'_A)=1$ (vastavalt $\mathbf{VF}_{Ka(B)}(A.B.R_B.\alpha'_B, \mu'_B)=1$).

### 11.1.9 Tulevikusalastus

Vaatleme alajaotises 10.1.7.2 kirjeldatud protokolli. Oletame, et *A* ja *B* on lõpetanud protokolli täitmise, kehtestanud võtme $\alpha$ ning kasutanud seda andmete krüpteerimiseks. Oletame veel, et vastane on salvestanud kõik selle võtmega seotud tegevused: tema käsutuses on seega $C=\mathbf{E}_{PKe(A)}(\alpha)$ (krüpteeritud seansivõti) ning kõik krüptogrammid $C_1,C_2,\ldots$, mis on krüpteeritud võtmega $\alpha$. Kuna võtmekehtestusprotokoll on turvaline, ei saa vastane talle teadaolevast informatsioonist lähtudes midagi teada võtme $\alpha$ kohta. Oletame, et seanss on lõppenud ning mingil põhjusel tuleb avalikuks osapoole *A* salajane võti *SK*(*A*). On selge, et sellisel juhul suudab vastane kompromiteerida kõiki edaspidiseid osapoole *A* seansse. On loomulik oletada, et *A* saab oma võtme lekkest teada ning tühistab oma avaliku võtme *PK*(*A*).

Sellest aga veel ei piisa. Vastasel on nüüd võti *SKe*(*A*), mida ta saab kasutada krüptogrammi *C* dekrüpteerimiseks, kui ta saab teada seansivõtme $\alpha$, mille abil ta oskab dekrüpteerida teated $C_1,C_2,\ldots$ ehk kogu salajase informatsiooni, mida *A* vahetas viimase seansi jooksul. See võimalus ei räägi vastu võtmekehtestusprotokolli turvalisusele, kuna viimane eeldas, et vastane ei saa kunagi ligipääsu salajastele võtmetele. Praktika vajadustest lähtudes peab nõudma lisatingimust: kui vastane saab teada võtmekehtestuses kasutatava pikaealise autentimisvõtme, ei tohi see põhjustada *enne* selle võtme kompromiteerumist toimunud sideseansse. Seda omadust nimetatakse ***tulevikusalastuseks***.

Tulevikusalastust garanteerib Diffie-Hellmani võtmekehtestusprotokolli alljärgnev (Joonis 77) autenditud variant, mida on siin kirjeldatud asümmeetrilisel kahe osapoolega juhul.




**Joonis 77. Tulevikusalastusega protokoll**

Protsess on alljärgnev.

1. *A* genereerib juhusliku stringi *x*, arvutab suuruse $X=g^x$ ning saadab *B*-le stringi *X*.
2. *B* genereerib juhusliku stringi *y*, arvutab suuruse $Y=g^y$ ja signatuuri $\mu=\mathbf{SIG}_{SKd(B)}(B.A.g^x.g^y)$ ning saadab *A*-le stringi $Y.\mu.$.
3. *A* verifitseerib, kas $\mathbf{VF}_{PKd(B)}(B.A.g^x.g^y.\mu.)=1$. Kui verifitseerimine õnnestus, arvutab *A* signatuuri $\mathbf{SIG}_{SKd(A)}(A.g^y)$ ning saadab selle *B*-le. Seejärel kehtestab *A* seansivõtme $g^{xy}=Y^x$.
4. *B* verifitseerib viimase signatuuri ning aktsepteerib seansivõtmega $g^{xy}=X^y$ kui verifitseerimine õnnestus.

Bellare ja Rogaway on tõestanud, et DH võtmekehtestusprotokolliga analoogilise protokolli kasutamine on vajalik tulevikusalastuse saavutamiseks. Nagu on seletatud alajaotises 10.1.1.5, ei ole DH võti iseenesest piisav (bititurvalisuse puudumise tõttu). Seetõttu tuleks tegelikuks võtmeks *K* kehtestada $H(g^{xy})$, kus *H* on "hea" räsifunktsioon.

## 11.2 Spetsiifilised krüptograafilised protokollid

### 11.2.1 Mõned kahe osapoolega protokollid

#### *11.2.1.1 Peitedastus*

Peitedastuse (OT, *oblivious transfer*) avastas Michael Rabin 1981. aastal. Protokolli täitmise ajal saadab Alice Bobile biti $m$ nii, et Bob saab biti teada tõenäosusega ½; Alice aga ei tea, kas bitt jõudis Bobini või mitte.

Peitedastusel on hulk kasulikke rakendusi. Joe Kilian (1988) on näidanud, et peitedastusest saab tuletada suvalise kahe osapoolega protokolli. Peitedastuse jaoks on pakutud alljärgnevat teostust.

1. Alice valib kaks juhuslikku algarvu $p$ ja $q$ ning leiab nende korrutise $n:=pq$. Alice krüpteerib arvu $m$ mingil tavalisel viisil, nii et krüptogrammi $c$ saaks dekrüpteerida vaid siis, kui teatakse arvu $n$ teguriteks lahutust. Alice saadab suurused $n$ ja $c$ Bobile.
2. Bob valib juhusliku suuruse $a \in \mathbb{Z}_n^*$ ning saadab Alice'ile suuruse $w = a^2 \bmod n$.
3. Alice leiab suuruse $w$ neli ruutjuurt $x, -x, y, -y$ modulo $n$, valib neist ühe juhuslikult ning saadab tagasi Bobile.
4. Kui Bob sai tagasi ruutjuure, mis ei ole $a$ või $-a$, oskab ta faktoriseerida moodulit $n$ ning seega leiab arvu $m$. Vastasel juhul Bob ei saa $m$-i teada. Alice ei tea, kumba juhuga on tegemist, kuna ta ei tea suurust $a$.

On selge, et Alice ei saa Bobi selles protokollis petta, kuna ta ei tea, millise ruutjuure $a$ Bob valis ($a$ valiku juhuslikkuse tõttu). Esmapilgul tundub, et ka Bob ei saa teda petta, kuna ta saab teada vaid juhusliku ruudu ruutjuure. Seda arvamust ei ole aga seni formaalselt tõestatud ning seega on võimalik, et valides mingi *kindla* väärtuse $z \in \mathbb{Z}_n^*$ ning seades $w:=z^2 \bmod n$, võib Bob saada täiendavat abiinformatsiooni arvu $n$ faktoriseerimiseks. On mõeldav, et arvu $(n-1)/2$ (või mõne teise spetsiaalse suuruse) ruutjuurte teadmine aitab Bobil $n$-i faktoriseerida.

Kui leiduks meetod, mille abil Bob saab Alice'ile *tõestada*, et ta tõepoolest järgis protokolli ning valis $a$ juhuslikuna ilma $a$-d ennast paljastamata, saaks protokolli modifitseerida nii, et see töötaks tõestatavalt. Jaotises 11.2.2 vaadeldakse, kuidas saab selliseid tõestusi teostada.

Peitedastusel on ka teine definitsioon: nn (1 2)-OT, kus Alice'il on kaks saladust, $m_0$ ja $m_1$. Bobil on valikubitt $c$. Protokolli lõpus saab Bob teada saladuse $m_c$, nii et Alice ei tea $c$ väärtust.

#### *11.2.1.2 Üheaegne lepingukinnitus*

Ülesande esitasid Even, Goldreich ja Lempel aastal 1985. Alice ja Bob soovivad signeerida lepingut, kuid ainult siis, kui ka teine osapool seda teeb. Ühesõnaga, kumbki osapool ei taha jääda ollukorda, kus ta on ainus, kes lepingu signeeris. Seda situatsiooni võib ette kujutada ka järgnevalt: nii Alice'il kui ka Bobil on mingi leping, esimene leping lubab midagi Bobile, teine lubab midagi Alice'ile. Alice ja Bob soovivad neid lepinguid vahetada.

Üks lähenemisi on järgmine. Alice kirjutab lepingule alla oma nime esitähe ning saadab lepingu Bobile. Bob teeb sedasama ning saadab lepingu tagasi Alice'ile. Eeldame, et nende nimed on võrdse pikkusega. Probleem on selles, et Bob võib keelduda oma nime viimast tähte kirjutamast, ehkki Alice on seda juba teinud, kuid seda saab lahendada, kui toimetada signeerimist kaduvväikeste osade kaupa: näiteks kui Alice ja Bob ei signeeri tähthaaval, vaid "millimeeterhaaval" Sellisel juhul ei saaks kumbki osapooltest teise ees kunagi "olulist" edumaad.

Vaatleme sellise protokolli elektroonilist analoogi. Alice ja Bob vahetavad stringe, mis on lepingu $m$ digitaalsignatuurid. Alice ja Bob signeerisid lepingu, saades tulemuseks $\mathrm{sig}_A(m)$ ning $\mathrm{sig}_B(m)$. Seejärel edastavad nad teineteisele signatuure bitthaaval. Selle meetodi puhul tekib probleem: mis siis, kui üks

osapooltest ei edasta teisele bitthaaval mitte signatuuri, vaid suvalist stringi? Teine osapool ei saa sellest teada enne protokolli täitmise lõppemist (alles siis saab ta verifitseerida, kas saadetis oli tõepoolest signatuur).

Even, Goldreich ja Lempel pakkusid sellele probleemile lahenduse, mis kasutab peitedastust. Alice arvutab signatuuri $L_A$ üle lepingu $m$ ja sõnumi "see on minu signatuur üle lepingu vasaku poole" ning signatuuri $R_A$ üle lepingu $m$ ja sõnumi "see on minu signatuur üle lepingu parema poole". Analoogiliselt arvutab Bob signatuurid $L_B$ ning $R_B$.

Alice (vastavalt Bob) valib kaks sümmeetrilise krüptosüsteemi (näiteks IDEA) võtit $K_A^L$ ja $K_A^R$ (vastavalt $K_B^L$ ja $K_B^R$) ning krüpteerib nendega vastavalt stringid $L_A$ ja $R_A$ (vastavalt $L_B$ ja $R_B$), saades krüptogrammid $C_A^L$ ja $C_A^R$ (vastavalt $C_B^L$ ja $C_B^R$). Osapool loeb lepingu signeerituks, kui tal on teise osapoole signatuuri mõlemad pooled.

Kõik krüptogrammid saadetakse teisele osapoolele. Kasutades edastust (1 2)-OT saadab Alice Bobile paari ($K_A^L$,$K_A^R$) ning Bob Alice'ile paari ($K_B^L$,$K_B^R$). Kumbki osapool saab seega teada ühe kahest teise osapoole kasutatud võtmest, kusjuures teine osapool ei tea, kumba võtit kasutati.

Protokolli teises osas saadavad Alice ja Bob üksteisele bitthaaval mõlemat IDEA võtit, kuni kõik bitid on saadetud. Kui üks osapool tuvastab vea tal juba olemasolevas võtmes, katkestab ta protokolli täitmise.

#### *11.2.1.3 Bitikinnistus*

Bob soovib, et Alice valiks biti (biti väärtuseks võib olla näiteks Alice'i pakkumine jalgpallivõistlustel) ning kinnistaks (*commit*) selle nii, et ta ei saaks hiljem oma valikut muuta. Teisalt ei soovi Alice, et Bob bitivaliku enneaegselt teada saaks. Alice avaldab (*conceal*) biti väärtuse hiljem, kokkulepitud ajal.

Alice tekitab selleks "elektroonilise seifi", mille võti on tal olemas. Alice paneb biti seifi ning saadab lukustatud seifi Bobile. Bob ei suuda seifi avada, ning järelikult ei tea biti väärtust (kinnistusfaas). Hiljem saadab Alice Bobile võtme, nii et Bob saab seifi avada. Nõutakse, et Alice ei suudaks genereerida kaht erinevat võtit, millega seifi avades nähtaks erinevaid bitiväärtusi.

Üks võimalus kinnistust realiseerida on kasutada kollisioonivaba räsifunktsiooni. Alice kinnistab $x$-i, saates Bobile väärtuse $y=H(x)$. Kuna $H$ on ühesuunaline funktsioon, ei oska Bob $y$-st $x$-i leida. Biti avaldamiseks saadab Alice Bobile $x$-i ning Bob kontrollib, kas $y=H(x)$. Kuna $H$ on kollisioonivaba, ei suuda Alice leida võtit $z \neq x$, nii et $H(z)=y$. Ssel lahendusel on aga nõrk bititurvalisus. Olukorra parandamiseks tuleks kasutada tugevaid bitte.

Teine võimalus on kasutada ruutjääke. Fikseerime selleks arvu $y \in \mathbb{Z}_n^*$, mis ei ole ruutjääk (st ei leidu arvu $z \in \mathbb{Z}_n^*$, nii et $z^2=y$). Alice valib juhusliku arvu $x \in \mathbb{Z}_n^*$, ning kinnistab biti 0, saates Bobile suuruse $x^2$, ning biti 1, saates Bobile suuruse $yx^2$. Eeldades, et ruutjääkide otsustusülesanne (antud on $z \in \mathbb{Z}_n^*$, tuleb otsustada, kas $z$ on ruutjääk või mitte) on raske (QR-eeldus, *quadratic residue assumption*), ei ole Bob suuteline saadetud väärtusest bitti teada saama. Biti avaldamiseks saadab Alice Bobile $x$-i väärtuse. See võimalus on turvaline ka siis, kui Alice'il on piiramata arvutuslik võimsus, kuid mitte siis, kui Bobi arvutuslik võimsus on piiramatu.

Vastupidise olukorra võib saavutada, kasutades diskreetseid logaritme. Olgu $p$ avalik algarv, $g \in \mathbb{Z}_p^*$ olgu vastava multiplikatiivse rühma (avalik) moodustaja ning $s \in \mathbb{Z}_p^*$ selle rühma avalik element, nii et diskreetne logaritm $\log_g(s)$ ei ole kellelegi teada. Alice kinnistab biti 0, valides $x$-i juhuslikult ning edastades Bobile väärtuse $y=g^x$ ning biti 1, saates Bobile väärtuse $sg^x$. Bobi jaoks on nii $g^x$ kui ka $sg^x$ rühma juhuslik element, nii ei suuda Bob kinnistatud bittide vahel vahet teha ka piiramatu arvutusliku võimsuse korral. Biti avaldamiseks saadab Alice Bobile $x$-i. Alice suudab Bobi petta vaid siis, kui ta oskab leida diskreetset logaritmi.

Bitikinnistusprotokolle kasutatakse muuhulgas nullteadmustõestustes (jaotis 11.2.2) ning mündiviskamisel.

#### *11.2.1.4 Mündi viskamine telefoni teel*

1982. aastal esitas Blum järgmise probleemi: kuidas teostada mündiviskamist telefoni teel (*coin flipping over the telephone*). Alice ja Bob soovivad ausalt münti visata, nii et visketulemus oleks mõlemale teada. Visketulemus peab olema juhuslik, kuid kummalgi ei tohi olla võimalik tulemust määrata. Kulli korral võidab Alice, kirja korral Bob.

Üks võimalus mündiviset realiseerida on järgmine. Alice valib juhusliku biti $a$ ning saadab selle Bobile. Bob valib juhusliku biti $b$ ning saadab selle Alice'ile. Pärast protokolli täitmise lõppu lepivad nad kokku, et tulemus on $a \oplus b$. Probleem on selles, kumb käib esimesena: kui üks osapool saab teise biti teada enne oma biti saatmist, võib ta petta.

Parem võimalus on järgmine. Alice kinnistab oma biti, saates Bobile arvu $y$=Committ($a$). Bob ei suuda $y$ põhjal $a$-d leida ning saadab Alice'ile biti $b$. Alice avaldab biti $a$. Bitikinnistuse definitsiooni põhjal ei saa Alice $a$ väärtust pärast $b$ saamist enam muuta. Mündiks valitakse $a \oplus b$.

#### *11.2.1.5 Peitedastusskeemi väärtustamine*

Alice ja Bob soovivad teada, kumb on vanem, paljastamata oma tegelikku vanust. Cindy ja Dandy soovivad teada, kumb on rikkam, paljastamata oma varanduse tegelikku väärtust (nn miljonäride probleem, *millionaires problem*): ainus, mida on vaja teada, on üks bitt (avaldise $a > b$ tõeväärtus).

Üldjuhul on Alice'il sisend $x_A$ ning Bobil on sisend $x_B$. Alice ja Bob soovivad leida funktsiooni $f(x_A, x_B)$ väärtust, kus $f$ on mingi teadaolev funktsioon, näiteks $f(x_A, x_B)=1$ parajasti siis kui $x_A > x_B$). Alice ja Bob tahavad arvutada $f$ väärtust peiteliselt, nii et arvutuse lõpus teaksid mõlemad väärtust $v=f(x_A, x_B)$, kuid kumbki ei tea midagi muud teise sisendi kohta.

Peitelise väärtustamise jaoks on teada keerukad üldised protokollid (Don Beaver 1991, Guillers Brassard ja Claude Crepeau 1986).

#### *11.2.1.6 Üheaegse võtmekehtestuse protokoll*

Vaatleme järgnevas protokolli, mis näib esmapilgul turvaline, kuid on avatud petmistele ligikaudu samadel põhjustel, nagu peitedastuski. Alice ning Bob teavad mõlemad järgmisi väärtusi: $1^k$, $\alpha \in \mathbb{E}_{n(A)}(s_A)$, $\beta \in \mathbb{E}_{n(B)}(s_B)$, $n(A)$ ja $n(B)$, kus $n(A)$ ja $n(B)$ on kumbki kahe võrdpikkusega algarvu korrutis, kus kõik algarvud on konkruentsed kolmega modulo 4. $\mathbb{E}_{n(A)}$ ja $\mathbb{E}_{n(B)}$ (vastavalt $n(A)$ ja $n(B)$ suhtes) on sama krüptosüsteem, mis peitedastuse korralgi. Alice teab arvu $n(A)$ teguriteks lahutust $n(A)=p_A q_A$ ning Bob teab arvu $n(B)$ teguriteks lahutust $n(B)=p_B q_B$. Alice ja Bob soovivad väärtusi $s_A$ ja $s_B$ "samal ajal" teada saada. Eeldame, et osapoolte arvutuslikud võimsused on võrdsed. Järgmist protokolli pakkus Blum 1983. aastal.

1. Alice valib juhuslikud arvud $a_i \in \mathbb{Z}_{n(B)}^*$, $i$=1, …, $k$ ning arvutab väärtused $b_i=a_i^2 \bmod n(B)$. Bob valib juhuslikud arvud $w_i \in \mathbb{Z}_{n(B)}^*$, $i$=1, …, $k$ ning arvutab väärtused $x_i=w_i^2 \bmod n(A)$.
2. $A$ saadab arvud $b_i$ $B$-le, $B$ saadab arvud $x_i$ $A$-le.
3. Iga $x_i$ jaoks arvutab Alice sellised $y_i$ ja $z_i$, et $y_i^2=z_i^2=x_i \bmod n(A)$ ja $y_i \neq \pm z_i$. Seega $y_i = \pm w_i$ või $z_i = \pm w_i$. Bob leiab sellised $c_i$ ja $d_i$, et $c_i^2=d_i^2=b_i \bmod n(B)$ ja $c_i \neq \pm d_i$. Seega $c_i = \pm b_i$ või $d_i = \pm b_i$.
4. Olgu $j$=1 kuni $k$. $A$ saadab $B$-le $y_i$ ning $z_i$-de $j$-indad kõrgeimad bitid, $i$=1,…, $k$. $B$ saadab $A$-le $y_i$ ning $z_i$-de $j$-indad kõrgeimad bitid, $i$=1,…, $k$.

5. Neljanda sammu täitmise järel faktoriseerib $A$ (vastavalt $B$) arvu $n(B)$ (vastavalt $n(A)$): $A$ (vastavalt $B$) leiab SÜT($c_i$–$d_i$,$n(B)$) (vastavalt SÜT($y_i$–$z_i$,$n(A)$)) iga $i$ väärtuse korral. Saadud teavet kasutades saavad Alice ja Bob suurusi $s_A$ ja $s_B$ dekrüpteerides kätte $\alpha$ ja $\beta$.

Protokollis edastatakse $k$ arvu ühe asemel, et vältida järgmist rünnet. Oletame, et $A$-le saadetakse vaid üks arv: $x$. $A$ leiab selle põhjal arvud $y$ ja $z$ ning saadab neljandal sammul $B$-le $y$-i ning juhuslikult valitud stringi $r$ $j$-inda kõrgeima biti, lootes, et $y = \pm w$. Kui $y = \pm w$, ei ole $B$-l võimalust enne viimast sammu kontrollida, kas $A$ petab või mitte. Selleks ajaks on $A$-l juba piisavalt palju informatsiooni, et leida $s(B)$, sellal kui $B$ ei oska leida $s(A)$-d. Järelikult on $A$-l võimalik 50% juhtudest $B$-d petta. Kui aga ühe $x$ asemel saadetakse edasi $k$ erinevat väärtust, muutub $A$ võimalus petta kaduvväikeseks: tõenäosus, et $\forall i$, $y_i = \pm w_i$ on $(½)^k$.

Kahjuks ei ole see protokoll siiski turvaline. Kui Alice ei vali kõiki $w_i$-sid juhuslikult, vaid valib vaid $w_1$ juhuslikult, väärtustab $x_1 = w_1^2 \pmod{n(B)}$ ning seejärel $x_i = x_1/2^{i-1} \pmod{n(B)}$, on $A$-l pärast neljanda sammu üht iteratsiooni olemas kogu vajalik informatsioon $n(B)$ faktoriseerimiseks. Kuna $B$-l ei ole mingit võimalust kontrollida, kas $A$ valis $x_i$-d juhuslikult ja sõltumatult, ei saavutata kirjeldatud protokolliga eesmärki.

**Märkus.** Kirjeldatud probleem on analoogiline peitedastuse juures tekkinud probleemiga. Seda saab lahendada, kui $A$-l ja $B$-l on võimalik kontrollida, kas teine osapool jälgib protokolli.

### 11.2.2 Nullteadmusprotokollid

Jaotis 11.1 esitas rea krüptograafilisi protokolle koos mõned probleemidega, mille all need kannatavad. Käesolevas peatükis vaatleme meetodeid, mis on välja töötatud nende protokollide turvalisuse tõestamiseks ning ehituse poolest tõestavalt turvaliste protokollide (*protocols provably secure by construction*) ehitamiseks. Põhiidee on taandada kahe osapoolega protokolli üldine probleem järgmisele lihtsamale probleemile: kuidas saavad $A$ ja $B$ tõestada, et mingi string $x$ kuulub keelde $L$, nii et mitte mingit muud teadmust peale teadmise, kas $x \in L$, ei lisandu. Kui sellised tõestused leiduksid kõigi klassi **NP** kuuluvate keelte jaoks, saaks $A$ $B$-le tõestada, et ta järgis protokolli samme. Alljärgnevas vormitakse toodud ideed rangelt, kasutades interaktiivsete tõestuste ning nullteadmuse mõisteid.

#### *11.2.2.1 Interaktiivsed tõestussüsteemid*

**Definitsioon.** *Interaktiivne Turingi masin* (ITM) on Turingi masin, millel on ainult loetav sisendlint, ainult loetav juhuarvude lint, loetav ja kirjutatav töölint, ainult loetav suhtluslint, ainult kirjutatav suhtluslint ning ainult kirjutatav väljundlint. Juhuarvude lint sisaldab lõpmata palju juhuarve, seda linti saab skaneerida vaid vasakult paremale. Öeldes, et ITM viskab münti, mõtleme me seda, et ITM skaneerib juhuarvude lindilt järgmise juhuarvu. Ainult kirjutatava suhtluslindi sisuks on ITM-i poolt saadetud bitistringid, ainult loetava suhtluslindi sisuks on ITM-i poolt vastu võetud bitistringid.

**Definitsioon.** Interaktiivne protokoll sama sisendlinti jagavate ITM-ide järjestatud paar ($A$,$B$), , kusjuures $B$ ainult loetav suhtluslint on $A$ ainult kirjutatav suhtluslint ja vastupidi. Masinad käivituvad kordamööda, kusjuures $B$ alustab. Käivitumise jooksul teostab vastav masin kõigepealt mingi arvutuse, lähtudes oma lintide sisust ning seejärel väljastab mingi stringi oma ainult kirjutatavale suhtluslindile. Masina $A$ (vastavalt $B$) $i$-s teade on string, mida $A$ (vastavalt $B$) kirjutab ainult kirjutatavale suhtluslindile oma $i$-nda käitusfaasi jooksul. Pärast teate kirjutamist desaktiveerub masin ning käiguõigus antakse üle teisele masinale (kui protokoll ei ole juba lõppenud). Kui üks masinatest väljastab suhtluslindile tühistringi (st ei kirjuta midagi), loetakse protokolli täitmine lõppenuks. Masin $B$ (verifitseerija) aktsepteerib (lükkab tagasi) protokolli, sisenedes aktsepteerimise (tagasilükke) olekusse ning lõpetades protokolli täitmise. Paari esimene masin, $A$ (tõestaja), on arvutuslikult piiramatu võimsusega Turingi masin. Masina $B$ arvutusaeg defineeritakse kui tema poolt aktiivsetel käikudel kulutatud arvutusaegade summa. Masina $B$ arvutusaeg on ülalt piiratud sisendstringi pikkuse polünoomiga.

**Definitsioon.** Olgu $L \subseteq \{0,1\}^*$. Keelel $L$ on interaktiivne tõestuste süsteem, kui leidub selline ITM $V$, mille puhul

1) leidub ITM $P$, nii et $(P,V)$ on interaktiivne protokoll ning $\forall x \in L$, kui $x$ on "piisavalt pikk", siis $V$ aktsepteerib tõenäosusega > 2/3 (kus tõenäosus on võetud üle $P$ ja $V$ kõikvõimalike mündivisete);
2) suvalise ITM $P$ korral, kui $(P,V)$ on interaktiivne protokoll siis $\forall x \notin L$, kui $x$ on "piisavalt pikk", siis $V$ aktsepteerib tõenäosusega < 1/3 (kus tõenäosus on võetud üle $P$ ja $V$ kõikvõimalike mündivisete).

**Märkus.** Ei piisa sellest, kui verifitseerijat ei suuda petta üks konkreetne tõestaja (see eeldaks, et tõestaja on usaldatav oraakel). Klassi **NP** kõikidel keeltel on triviaalsed interaktiivsed tõestused.

**Definitsioon.** $(P,V)$ on keele $L$ interaktiivne tõestussüsteem. Olgu **IP** selliste keelte hulk, millel leiduvad interaktiivsed tõestused.

### *11.2.2.2 Näiteid*

Järgnevas tähistab $P \to Q$ masina $P$ aktiivset faasi, mille lõpus $P$ saadab $Q$-le mingi teate $M$.

**Näide 1 (arvuteooriast).** Olgu QR $=\{(x,n) | \, x \in \mathbb{Z}_n^*, \exists y, y=x^2 \bmod n\}$ ning QNR $=\{(x,n) | \, x \in \mathbb{Z}_n^*, \neg\exists y, y=x^2 \bmod n\}$. Kirjeldame keele QNR jaoks interaktiivse tõestussüsteemi.

Sisendil $(x,n)$ täidab interaktiivne protokoll $(A,B)$ järgmised sammud:

$B \to A$: $B$ saadab $A$-le listi $w_1, \ldots, w_k$, kus $k=|n|$ ning

$$w_i = \begin{cases} x_i^2 & \bmod n, b_i = 1 \\ x \cdot z_i^2 & \bmod n, b_i = 0 \end{cases},$$

kus $B$ valis elemendid $z_i \in \mathbb{Z}_n^*$ ja $b_i \in \{0,1\}$ juhuslikult.

$A \to B$: A saadab $B$-le listi $c_1, \ldots, c_k$, nii et

$$c_i = \begin{cases} 1, kui\,(w_i,n) \in QNR \\ 0, kui\,(w_i,n) \notin QNR \end{cases}.$$

$B$ aktsepteerib, kui suvalise $i$ korral $c_i=b_i$, st $B$ interpreteerib fakti $c_i=b_i$ kui tõendit, et $(x,n) \in$ QNR.

On lihtne näidata, et $(A,B)$ on keele QNR interaktiivne tõestussüsteem. Kui $(x,n) \in$ QNR, siis on $w_i$ ruutjääk modulo $n$ parajasti siis, kui $b_i=1$. Seega leiab kõikvõimas $A$ kergesti, kas $w_i$ on ruutjääk modulo $n$ või mitte, ning sunnib seetõttu $B$-d aktsepteerima tõenäosusega 1. Kui $(x,n) \notin$ QNR ning $(x,n) \in$ QR, on $w_i$ ruutjääk modulo $n$ sõltumatult sellest, kas $b_i=0$ või 1. Seega on tõenäosus, et (piiramatu arvutusvõimsusega) $A$ saadab sellise $c_i$, et $b_i=c_i$, tõkestatud ülevalt ½-ga suvalise $i$ jaoks ning seega aktsepteerib Bob ülimalt tõenäosusega $(½)^k$.

**Näide 2 (graafiteooriast).** Vaatleme graafide mitteisomorfismi (GNI, *graph non-isomorphism*) probleemi. Interaktiivse protokolli sisendiks on graafipaar $(G_1,G_2)$; $A$ peab tõestama $B$-le, et ei leidu sellist graafi $G_1$ servade bijektsiooni $\pi$ graafi $G_2$ servadele, nii et kehtiks järgmine: $(u,v)$ on serv graafis $G_1$ parajasti siis kui $(\pi(u),\pi(v))$ on serv graafis $G_2$. Üldiselt usutakse, et GNI ei kuulu klassi **NP**. Keele GNI interaktiivne tõestus $(A,B)$ töötab sisendil $(G_1,G_2)$ järgmiselt.

1. $B \rightarrow A$: *B* valib juhuslikult ühe kahest sisendgraafist, $G = G_{\alpha_i}$, $\alpha_i \in \{0,1\}$. *B* genereerib juhuslikult graafi *G* isomorfse koopia *G'* ning saadab selle *A*-le;
2. $A \rightarrow B$: *A* saadab *B*-le biti $\beta_i \in \{1,2\}$. Samme 1 ja 2 täidetakse *k* korda: $i=1,\ldots, k$.

*B* aktsepteerib kui $\alpha_{i=}\beta_i$, iga *i* korral, ehk *B* interpreteerib võrduste süsteemi $\alpha_{i=}\beta_i$ kui tõendit, et graafid on mitteisomorfsed.

Kui kaks graafi ei ole isomorfsed, vastab tõestaja õigesti tõenäosusega 1. Kui graafid on isomorfsed, ei suuda tõestaja teineteisest eristada graafide $G_1$ ja $G_2$ juhuslikult valitud isomorfseid koopiaid ning seega on aktsepteerimise tõenäosus sel juhul$(½)^k$.

### *11.2.2.3 Nullteadmus*

Kui palju "teadmust" peab interaktiivne tõestussüsteem edastama selleks, et veenda polünomiaalses ajas töötavat verifitseerijat väite tõesuses?. "Teadmuse" mõistet saab lühidalt selgitada järgmise näite abil. Vaatleme **NP**-täielikku keelt SAT (kõikide kehtestatavate lausearvutusvalemite keel). Kõige ilmsem tõestussüsteem on selline, kus sisestatud lausearvutusvalemi *F* korral edastab tõestaja verifitseerijale muutujate sellise väärtustuse *I*, mille korral *F* on tõene. Kui väärtustuse *I* enda leidmine nõuaks verifitseerijalt superpolünomiaalse aja (st kui **P**≠**NP**), ütleme, et protokolli käigus sai verifitseerija lisaks faktile $F \in$SAT veel lisateadmust.

Goldwasser, Micali ja Rackoff (1985) formaliseerisid ***nullteadmuse*** mõiste: keele *L* interaktiivset tõestussüsteemi nimetatakse nullteadmustõestuseks, kui suvalise $x \in L$ korral, mida verifitseerija suudab arvutada pärast tõestajaga interaktsioonis olemist, suudaks ta arvutada sisendi *x* üksi, kasutades tõenäosuslikku polünomiaalset Turingi masinat.

Nullteadmuse tehniline definitsioon antakse järgmises alajaotises.

### *11.2.2.4 Definitsioonid ja teoreemid*

Olgu (*A*,*B*) interaktiivne protokoll. ***Vaateks*** (*view* nimetatakse masinate *A* ja *B* fikseeritud mündivisete korral verifitseerija poolt sooritatud mündivisete stringi konkatenatsiooni teadete jadaga verifitseerija ja tõestaja vahel (vaade on juhuslik suurus, mis on jaotatud üle *A* ja *B* mündivisete). Tähistagu *h* verifitseerija salajast informatsiooni, mille pikkus on polünomiaalne *A* ja *B* ühissisendi suhtes.

**Definitsioon.** (*A*,*B*) on keele *L* täielik nullteadmusprotokoll (*perfect zero-knowledge protocol*), kui leidub selline tõenäosuslik polünomiaalne Turingi masin *M*, nii et suvalise $x \in L$, suvalise $a > 0$ ja suvalise *h*, $|h| < |x|^a$ korral on juhuslike suuruste $M(x,h)$ ja *vaade* jaotused võrdsed (*M* jaotus on üle kõigi *M*-i mündivisete sisendi (*x*,*h*) korral).

**Definitsioon.** (*A*,*B*) on keele *L* statistiline nullteadmusprotokoll (*statistic zero-knowledge protocol*), kui leidub selline tõenäosuslik polünomiaalne Turingi masin *M*, nii et suvalise $x \in L$, suvalise $a > 0$ ja suvalise *h*, $|h| < |x|^a$ korral

$$\sum_{\alpha} |\text{Prob}(M(x,h) = \alpha) - \text{Prob}(\text{vaade} = \alpha)| < \frac{1}{|x|^c}$$

kõigi konstantide $c>0$ ning piisavalt suure $|x|$ korral.

Intuitiivselt tähendab statistiline nullteadmus seda, et piiramatu arvutusvõimsusega vaatleja, kellele on antud ainult polünomiaalselt suured valimid hulkadest $\{M_r(x,h) | r$ muutub üle masina *M* kõikvõimalike mündivisete$\}$ ning $\{\text{vaade}_r | r$ muutub üle *A* ja *B* kõikvõimalike mündivisete$\}$, ei suuda neid kaht valimit eristada.

**Definitsioon.** (*A*,*B*) on arvutuslikult nullteadmustõestus (*computationally zero-knowledge*), kui leidub selline tõenäosuslik polünomiaalne Turingi masin *M*, nii et iga polünomiaalse suurusega skeemide pere $C=\{C_{|x|}\}$, suvaliste konstantide $a$, $d > 0$, suvalise piisavalt pika $x \in L$ ning suvalise stringi $h$, $|h| < |x|^a$ korral

$$\text{Prob}(C_{|x|}(\alpha) = 1 |\ \alpha\ = M(x,h)) - \text{Prob}(C_{|x|}(\alpha) |\ \alpha\ = \text{vaade}(x)) < \frac{1}{|x|^d}.$$

Keelel *L* on (arvutuslik/statistiline/täielik) nullteadmustõestussüsteem, kui

1) leidub keele *L* interaktiivne tõestussüsteem (*A*,*B*);

2) suvalise ITM *B'* korral on interaktiivne protokoll (*A*,*B'*) (arvutuslik/statistiline/täielik) nullteadmusprotokoll keele *L* jaoks.

Tähistagu keerukusklass **KC[0]** nende keelte hulka, millel leidub arvutuslik nullteadmusprotokoll.

**Teoreem** (Goldreich, Micali, Widgerson 1986). Kui leidub mitteühtlane (*non-uniform*, ühelainsal sisendi pikkusel toimiv) mitte-eristatav (*indistinguishable*, semantilise turvalisusega võrreldav omadus) krüptoskeem, on igal klassi **NP** kuuluval keelel arvutuslik nullteadmusprotokoll.

Impagliazzo, Levin, Luby ja Naor tõestasid, et kui leiduvad mitteühtlased ühesuunalised kujutused, leidub ka mitteühtlane mitte-eristatav krüptoskeem.

**Teoreem.** Kui leidub vähemalt üks mitteühtlane ühesuunaline kujutus, siis **NP** $\in$ **KC[0]**.

Goldreich, Mical ja Widgerson tõestasid, et ühel konkreetsel **NP**-täielikul ülesandel, graafide kolmevärvitavusel (*graph three colorability*) leidub toodud eeldustel arvutuslik nullteadmusprotokoll. Ülesanne on järgmine: tõestaja soovib verifitseerijale tõestada, et graafi *G* saab värvida kolme värviga,värviskeemi ennast avalikustamata. Tõestaja teeb seda $|E|^2$ sammuga, kus iga sammu jooksul sooritatakse järgmised toimingud.

1. Tõestaja vahetab kolm värvi omavahel juhuslikult ära (näiteks värvides punased tipud sinisteks, sinised rohelisteks ja rohelised punasteks).
2. Tõestaja krüpteerib iga tipu värvi, kasutades tõenäosuslikku krüptoskeemi iga tipu jaoks, ning esitab verifitseerijale kõigi tippude värvide krüptogrammid koos vastavusega, mis seob krüptogramme tippudega. Seejuures krüpteeritakse iga tipu värv erineva värskelt genereeritud võtmega.
3. Verifitseerija valib graafi ühe juhusliku kaare.
4. Tõestaja avaldab verifitseerijale kahe selle kaarega intsidentse tipu värvid, edastades värvide krüpteerimisel kasutatud võtmed.
5. Verifitseerija kontrollib, kas dekrüpteerimised õnnestusid ning kas serva kahe otstipu värvid on tõepoolest erinevad elemendid lubatavate värvide hulgast.

Sammu 2 juures kasutatud krüptoskeem peab olema tõenäosuslik polünomiaalses ajas mitte-eristatav. Kui graaf on tõepoolest kolmevärvitav, ei avasta verifitseerija kunagi ühtki serva, mille otspunktid oleksid värvitud sama värviga. Kui graaf ei ole kolmevärvitav, valib verifitseerija igal sammul vähemalt tõenäosusega $|E|^{-1}$ serva, mille otstipud on värvitud sama värviga.

On võimalik tõestada, et toodud graafi kolmevärvitavuse protokoll on arvutuslik nullteadmusprotokoll. Sellest järeldub, et suvalise klassi **NP** kuuluva keele jaoks leidub arvutuslik nullteadmusprotokoll.

#### *11.2.2.5 Rakendused kasutaja identimisel*

Nullteadmustõestuste abil saab täiesti uudsel viisil realiseerida paroolidesüsteemi. Iga kasutaja avalikustab "teoreem", mille tõestust teab vaid tema. Teoreemiks võib olla näiteks väide "*n* on kahe suure algarvu korrutis". Sisselogimisel käivitub kasutaja ning masina vahel selle teoreemi nullteadmustõestus.

Kui tõestus õnnestub, lubatakse kasutajal end masinasse sisse logida. Selline lähenemine välistab ajaliselt piiratud vastase passiivseid ründeid: kui vastase poolt pealtkuulatud sisselogimisseansside arv on polünomiaalne (|$n$| suhtes), ei ole ta teada saanud lisainformatsiooni edasiste rünnete lihtsustamiseks. Kõige tuntum nullteadmustõestusi kasutav identimisprotokoll on Fiat-Shamiri protokoll aastast 1986.

### 11.2.3 Mitme osapoolega protokollid

Tüüpiline mitme osapoolega protokoll on mõeldud võimaldama osapooltel koordineerida oma tegevusi ning saavutama teatud eesmärke ka siis, kui teatav (piisavalt väike) arv osapooli on vastase poolt korrumpeeritud. Protokoll peab garanteerima selle, et "head" osapooled saavutaksid eesmärgi ka siis, kui korrumpeerunud osapooled ei pea kinni protokolli reeglistikust (võimalik, et sihiga takistada "headel" eesmärki saavutada)

#### *11.2.3.1 Ühissalastus*

Ühissalastusprotokollid töötati sõltumatult välja Blakley ja Shamiri poolt (1979). Mitme osapoolega keskkondades on ühissalastus fundamentaalse tähtsusega idee ning tööriist.

Ühissalastus kaitseb andmeid nende hajutamise teel. Oletame, et Alice'il on olulise tähtsusega võti $s$. Võtme kaotamise vältimiseks võiks Alice anda võtme koopia Bobile, kuid Alice ei usalda ühtki teist inimest täielikult: mitte ainult seetõttu, et Bob võib muutuda mitteusaldatavaks, vaid ka väliste ohtude tõttu nii Bobile kui isikule kui ka võtme asukohale (kui aegruumi punktile).

Üks võimalus võtit $n$ eri inimeste vahel jagada on anda $j$-ndale inimesele suurus $s_j$, kus $s_1$, ..., $s_{n-1}$ on genereeritud juhuslikult ning $s_n = s \oplus s_1 \oplus \cdots \oplus s_{n-1}$. Sellisel juhul suudavad vaid kõik $n$ inimest *koos* võtme välja arvutada. Pärast kokkutulemist ning võtme kasutamist loetakse võti mõne aja pärast kehtetuks (kuna kui $s$ on kord juba ühiselt leitud, jääb ta ka inimeste ühisteadmusse).

Arvu $s_j$ nimetatakse ***(võtme)tükiks*** (*share*). Tükke genereerib võtme (saladuse) $s$ algvaldaja: mõnikord üks $n$-ist osapoolest, mõnikord keegi väljastpoolt. Tükkide genereerijat nimetatakse ***diileriks*** (*dealer*); diiler annab tüki $s_j$ konfidentsiaalselt $j$-ndale osapoolele.

Saab teostada ka üldisemaid ühissalastusskeeme. $(t,n)$-ühissalastusskeemides on $n$ osapoolt, kellest mistahes $t+1$ osapoolt peavad olema suutelised saladust leidma, sellal kui mistahes $t$ osapoolt ei tohi selleks suutelised olla.

Adi Shamir (1979) kasutas $(t,n)$-ühissalastusskeemi teostamisel polünoome. Nimelt, olgu $F$ – lõplik korpus ning $f(x) = a_0 + a_1 x + \cdots + a_t x^t$, kus $a_i \in F$. Seega on astmega $t$ polünoomil $f(x)$ $t+1$ termi. Polünoomidel on teatavasti järgmised head omadused.

- Interpoleerimine: Kui on antud polünoomi $f(x)$ $t+1$ punkti $(x_1, y_1)$, …, $(x_{t+1}, y_{t+1})$, kus suurused $x_i \neq 0$ on paarikaupa erinevad ning $y_i = f(x_i)$, on võimalik leida *kõik* suurused $a_i$.

- Salastus: Kui on antud polünoomi $f(x)$ $t$ punkti $(x_1, y_1)$, …, $(x_t, y_t)$, kus suurused $x_i \neq 0$ on paarikaupa erinevad ning $y_i = f(x_i)$, ei ole võimalik leida *mitte midagi* polünoomi $f$ väärtuse $y_{t+1}$ kohta eelmistest punktidest erinevas punktis $x_{t+1}$: iga $\alpha$ korral leidub parajasti üks selline polünoom $f_\alpha(x)$, nii et $y_i = f_\alpha(x_i)$, suvalise $i \in \{1,\ldots,t\}$ jaoks, ning $\alpha = f_\alpha(x_{t+1})$.

Nende omaduste tõttu on polünoomid ideaalsed ühissalastuse jaoks. Diiler annab igale osapoolele $P_i$, $i \in \{1,\ldots,t+1\}$ ja juhuslikult valitud punkti $x_i \in \mathbb{F}$, nii et $i \neq j$ korral $x_i \neq x_j$ (seega $|F| \geq n$). Seejärel genereerib diiler juhuslikult arvud $a_i$, $i \in \{1,\ldots,t\}$, väärtustab $a_0 = s$ ning defineerib $f(x) := a_0 + a_1 x + \cdots + a_t x^t$. Diiler leiab väärtused $s_i = f(x_i)$ ning saadab suuruse $s_i$ konfidentsiaalselt osapoolele $P_{i+1}$, $i \in \{0,\ldots,t\}$.

Tänu polünoomi mainitud omadusele saavad $t+1$ osapoolt kokku tulles teada suuruse $a_0 = s$. Ükski $\leq t$ osapoolest koosnev rühm aga ei suuda $s$-i leida.

#### *11.2.3.2 Verifitseeritav ühissalastus*

Shamiri ühissalastusskeemil on vähemalt kaks puudust. Ebaaus diiler võib osapooltele anda tükke, mida kokku pannes ei ole saladus $s$ üheselt leitav. Ebaausad osapooled võivad kokkutuleku faasis anda teistele osapooltele valeinformatsiooni, tehes seeläbi saladuse leidmise võimatuks.

Chor, Goldwasser, Micali ja Awerbuch (1985) esitasid faktoriseerimise trakteerimatusel tugineva verifitseeritava ühissalastusskeemi, kus iga osapool saab verifitseerida talle jagatud tüki õigsust. Skeemis võib olla $O(\log n)$ ebaausat osapoolt. Ben-Or, Goldwasser ja Widgerson (1988) näitasid, kuidas saavutada veaparanduskoodide abil verifitseeritavat ühissalastusskeemi ilma mingi krüptograafilise eelduseta, kui ebaausaid osapooli on vähem kui kolmandik. Rabin ja Ben-Or'i skeemis (1989) peab ebaausaid osapooli olema alla poole.

#### *11.2.3.3 Anonüümsed tehingud*

David Chaum propageerib anonüümseid tehinguid, mis kaitsevad üksikisikuid kõiki sooritatud tehinguid suures andmebaasis hoidva "Suure Venna" eest. Chaum töötas selleks otstarbeks välja nn ***digitaalsete pseudonüümide*** kasutamise. Pseudonüüme kasutades saavad isikud sooritada tehinguid, teades, et hiljem ei suuda "Suur Vend" tehinguid üksikisikutega seostada. Väljatöötatud protokollid võimaldavad mainitud ***seostamatust*** (*unlinkability*) kombineerida kindlustundega tehingu teise osapoole jaoks selles, et isik on volitatud tehingut teostama (= on maksuvõimeline).

#### *11.2.3.4 Mitme osapoolega süstikprotokollid*

Üks võimalus krüptograafiliste protokollide turvalise demonstreerimiseks on näidata, et osapoolte poolt sooritatavad primitiivoperatsioone ei saa komponeerida nii, et salajane informatsioon avalikuks tuleks.

Vaatleme Dolevi ja Yao poolt (1981) toodud näidet. Alice krüpteerib teate $M$ Bobi avaliku võtmega $PK(B)$ ja saadab Bobile krüptogrammi $C_B = E_{PK(B)}(M)$. Bob saadab Alice'ile tagasi Alice'i avaliku võtmega $PK(A)$ krüpteerimisel saadud krüptogrammi $C_A = E_{PK(A)}(M)$.

Kuna teade $M$ krüpteeriti mõlema ringi ajal, on selge, et passiivne vastane ei saa $M$-i teada. Aktiivse vastase vastu on aga see protokoll ebaturvaline: vastane jätab meelde krüptogrammi $C_B$. Hiljem alustab vastane $E$ Bobiga sama protokolli täitmist, edastades Bobile teate $C_B$. Bob tagastab vastasele kohusetundlikult suuruse $C_E = E_{PK(E)}(M)$, mida dekrüpteerides saab vastane teada $M$-i.

Selliseid ründeid, kus protokolli mingi eelmise käitustiiru teateid (või nendest primitiivoperatsioonide abil moodustatud teateid) esitatakse hiljem uue käitustiiru teadete pähe, nimetatakse ***taasesitusrünneteks***.

Mõnikord on võimalik tõestada, et protokoll on turvaline taasesitusrünnete vastu. Esimene selline tõestus on Dolevilt ja Yaolt (1981). Mõnikord saab olemasolevat protokolli modifitseerida nii, et saadud protokoll on turvaline.

#### *11.2.3.5 Mitme enamikus ausa osapoolega protokollid*

Goldreich, Micali ja Widgerson (1986) näitasid, kuidas ausate osapoolte jaoks konstrueeritud protokolli kompileerida protokolliks, mis töötab korrektselt ka siis, kui vaid enamik osapooli on ausad. Ehkki esialgne protokoll võib lekkida teatud saladusi, ei tea kompileeritud protokolli täitmise lõpus ükski osapool rohkem kui ta teadis enne protokolli käivitamist. Universaalse kompilaatori olemasoluks eeldati salaluugiga funktsioonide olemasolu.

Ben-Or, Goldwasser ja Widgerson (1988) ning Chaum, Crépeau ja Damgård (1987) läksid veel ühe sammu edasi, käsitledes salajast suhtlust osapoolte vahel kui krüptograafilist primitiivi. Ilma krüptograafilistele eeldustele tuginemata ehitasid nad kompilaatori, mis polünomiaalses ajas töötava funktsiooni (või skeemi) kirjeldusest lähtuvalt konstrueerib protokolli, mis väärtustab funktsiooni alati õigesti ning garanteerib, et ebaausatele osapooltele ei leki protokolli täitmise jooksul täiendavat teadmust. Eeldatakse, et ebaausaid osapooli ei ole üle kolmandiku, kusjuures ebaausate osapoolte käitumise määrab piiramatu arvutusvõimsusega vastane.

Sellised "meisterteoreemid" on väga olulised vahendid turvaliste protokollide ehitamisel.

### 11.2.4 Elektroonilised valimised

Elektroonilised valimised (EE, *electronic elections*) on mitme osapoolega protokollide rakendamise tüüpiline näide. EE tüüpjuhul tahavad $n$ osapoolt, kellest igaühel on salajane sisend $x_i$, leida $n$-aarse funktsiooni $f$ väärtuse $f(x_1,\cdots,x_n)$ ilma sisendeid avalikustamata (vt ka jaotis 11.2.1.5). Elektrooniliste valimiste korral on osapoolteks valijad, kelle sisendandmed on kahendarvud; funktsioon arvutab sisendite summa ning tulemiks on skoor (*tally*). Üldjuhul soovitakse valimisprotokollidelt järgmisi omadusi.

1. Hääletada saavad ainult volitatud valijad.
2. Keegi ei saa hääletada rohkem kui üks kord.
3. Antud hääled on salajased.
4. Keegi ei saa dubleerida teise osapoole häält.
5. Skoor leitakse korrektselt.
6. Kõik saavad verifitseerida, et punkt 5 on täidetud.
7. Protokoll peab olema veakindel (töötama ka siis, kui osa valijatest on ebaausad).
8. Hääletajat ei saa sundida oma häält paljastama.

Tavaliselt ei ole valimisprotokollis soovitatav kaasata kõiki valijaid $V_i$ arvutusprotsessi; eeldatakse, et leiduvad nn valimiskeskused $C_1,\cdots,C_n$, kelle ülesandeks on häälte kogumine ning skoori arvutamine.

#### *11.2.4.1 Merritti valimisprotokoll*

Vaatleme järgnevalt Merritti pakutud valimisprotokolli (1983). Iga keskus $C_i$ avalikustab enda avaliku võtme $PK(C_i)$ (ning hoiab vaid enda teada oma salajase võtme $SK(C_i)$). Olgu valija $V_j$ sisend (hääl) $v_j$. Hääletamiseks valib $V_j$ juhusliku arvu $s_j$ ning arvutab suuruse

$$E_{PK(C_1)}(E_{PK(C_2)}(\ldots E_{PK(C_n)}(v_j,s_j))) =: y_{n+1,j}.$$

Väärtused $y_{n+1,j}$ avalikustatakse. Valimiskeskused $C_n, C_{n-1}, \cdots, C_1$ teevad (toodud järjekorras) järgmist. Iga $y_{i+1,j}$ jaoks valib keskus $C_i$ juhusliku väärtuse $r_{i,j}$ ning avalikustab suuruse $y_{i,k} = E_{PK(C_i)}(y_{i+1,j}, j)$. Indeks $k$ leitakse, leides juhusliku permutatsiooni $\pi_i$ üle täisarvude $[1,\ldots n]$: $k := \pi_i(j)$. Keskus $C_i$ hoiab permutatsiooni $\pi_i$ saladuses. Protokolli selle faasi lõpus saadakse kätte väärtus

$$y_{1,j} = E_{PK(C_1)}(E_{PK(C_2)}(\ldots E_{PK(C_n)}(y_{n+1,j}, r_{n,j}), r_{2,j}), r_{1,j}).$$

Järgmises faasis toimub verifitseerimine, kus teostatakse kahe ringi jooksul dekrüpteerimisi järjekorras $C_1 \to C_2 \to \ldots \to C_n$. Dekrüpteeritud väärtused postitatakse ning skoor leitakse, võttes summa üle kõikide häälte $v_i$.

Merritti protokoll rahuldab nõudeid 1 ja 2. Nõue 3 on rahuldatud: isegi kui hääled tulevad avalikuks, jääb peidetuks seos hääletaja ning hääle vahel (sellise seose konstrueerimiseks peaks vastane teadma kõiki permutatsioone $\pi_i$). Tingimus 4 ei ole rahuldatud, sest üks hääletaja võib lihtsalt kopeerida teise hääletaja poolt väljastatud stringi. Tingimused 5 ja 6 on rahuldatud tänu juhuslike stringide kasutamisele: esimese dekrüpteerimisringi ajal kontrollivad keskused, et nende poolt genereeritud juhuslikud stringid on osa avatekstist (seega kindlustades, et kõiki selle keskuse krüptogramme arvestatakse). Teise dekrüpteerimisraundi lõpus verifitseerib iga hääletaja, et saadud avatekstis on tema genereeritud string $s_j$ (seega kindlustades, et tema häält arvestatakse). Seega on valimisprotsessi korrektsuse verifitseerimiseks vajalik kõigi hääletajate koostöö.

Tingimus 7. Protokoll töötab õigesti ka siis, kui mõned osapooled on ebaausad: isegi $n-1$ koopereerunud keskust ei suuda teada saada *kõiki* permutatsioone $\pi_i$. Seevastu ei ole võimalik hääletustulemusi leida juhul, kui (vähemalt) üks näiteks "läheb krahhi" ja kaotab kõik andmed: sellisel juhul on vaja kogu hääletusprotsess taasalgatada.

Tingimus 8 ei ole rahuldatud: hääletajat saab sundida nii $v_j$-i kui $s_j$-i avaldama. Hääletaja valetamist saab tuvastada, sest sellisel juhul ei vasta tema poolt väljastatud arvud krüptogrammile $y_{n+1,j}$.

#### *11.2.4.2 Veakindel valimisprotokoll*

Selles jaotises kirjeldame protokolli, millel on järgmised omadused.

- Protokoll rahuldab tingimust 4, st on võimatu kopeerida teiste inimeste hääli.
- Hääletustulemuste verifitseerimiseks ei ole vaja kõigi hääletajate koostööd.
- Veakindlus: teatud fikseeritud läviväärtuse $t$ korral: kui "halbu" keskusi on vähem kui $C_i$, leiab protokoll korrektselt hääletustulemuse ning kõik hääled jäävad konfidentsiaalseteks.

See protokoll ei taga siiski tingimust 7 (sunnitamatust, *uncoercibility*). Protokolli panid ette Cramer, Franklin, Schoemakers ja Yung (1996).

**Definitsioon.** Bitikinnistusskeem $B$ on $(+,\times)$-homomorfne, kui $B(X+Y) = B(X) \times B(Y)$.

**Pederseni homomorfne bitikinnistusskeem** $B$ töötab järgmiselt. Olgu $p$ algarv kujul $p = kq+1$ ning olgu $g$, $h$ kaks elementi $q$-ndat järku alamrühmas. Eeldatakse, et keegi ei tea suurust $\log_g h$. Arvu $m \in [1,\ldots,q]$ kinnistamiseks saadab Alice Bobile suuruse $B_a(m) = g^a h^m$ juhuslikult valitud $a \in [0,\ldots,q-1]$ jaoks. Kinnistuse avaldamiseks saadab Alice Bobile suurused $a$ ja $m$.

$B$ on $(+,\times)$-homomorfne, sest

$$B_{a_1}(m)B_{a_2}(m) = g^{a_1}h^{m_1}g^{a_2}h^{m_2} = g^{a_1+a_2}h^{m_1+m_2} = B_{a_1+a_2}(m_1+m_2).$$

Edasises olgu $E$ $(+,\times)$-homomorfne bitikinnistusskeem.

Esituse lihtsustamiseks esitame protokolli ainult ühe valimiskeskusega juhuks.

**Valimine ühe keskuse korral.**

Olgu üks keskus $C$, kelle krüptofunktsioon on $E$. Eeldame, et hääled on kas $-1$ või $1$, ning et iga valija $V_j$ krüpteerib oma hääle $v_j$, kasutades funktsiooni $B_{a_j}$ juhuslikult valitud $a_j$ korral. Väärtused $B_{a_j}(v_j)$ avalikustatakse. $V_j$ saadab keskusele $C$ krüpteeritud kujul arvud $a_j$ ja $c_j$. Seejärel tõestab valija, kasutades nullteadmusprotokolli, et antud hääl on korrektne (st et krüptogrammi all olev avatekst on kas $-1$ või $1$). Ülaltoodud Peterseni kinnistusskeemi korral on nullteadmusprotokoll lihtne.

**Kui $v=1$:**

1) hääletaja $V$ valib juhuslikult arvud $a$, $r_1$, $d_1$, $w_2$ modulo $q$. $V$ postitab suurused $B_a(v) = g_a h$, $a_1 = g^{r_1}(B_a(v)h)^{-d_1}$ ja $\boxed{a_2 = g^{w_2}}$;
2) keskus $C$ saadab hääletajale juhusliku arvu $c$ modulo $q$;
3) hääletaja $V$ leiab suurused $d_2 = c - d_1$ ja $r_2 = w_2 + ad_2$ ning postitab arvud $d_1$, $d_2$, $r_1$, $r_2$;
4) keskus verifitseerib, kas (a) $d_1 + d_2 = c$, (b) $g^{r_1} = a_1(B_a(v)h)^{d_1}$ ja (c) $g^{r_2} = a_2(B_a(v)h)^{d_2}$.

**Kui $v=-1$:**

1) hääletaja $V$ valib juhuslikult arvud $a$, $r_2$, $d_2$, $w_1$ modulo $q$. $V$ postitab suurused $B_a(v) = \frac{g_a}{h}$, $a_2 = g^{r_2}\left(\frac{B_a(v)}{h}\right)^{-d_2}$ ja $a_1 = g^{w_1}$;
2) keskus $C$ saadab hääletajale juhusliku arvu $c$ modulo $q$;
3) hääletaja $V$ leiab suurused $d_1 = c - d_2$ ja $r_1 = w_1 + ad_1$ ning postitab arvud $d_1$, $d_2$, $r_1$, $r_2$;
4) keskus verifitseerib, kas (a) $d_1 + d_2 = c$, (b) $g^{r_1} = a_1\left(\frac{B_a(v)}{h}\right)^{d_1}$ ja (c) $g^{r_2} = a_2\left(\frac{B_a(v)}{h}\right)^{d_2}$.

**Hääletustulemuse arvutamine (1 keskus)**

Eelmise faasi lõpus teati suurust $B_{a_j}(v_j)$ iga hääletaja $V_j$ jaoks. Keskus avalikustab hääletustulemuse $T = \sum_j v_j$ arvu $A = \sum_j a_j$. Igaüks saab kontrollida tulemuse õigsust, verifitseerides, kas $B_a(T) = \prod_j B_{a_j}(v_j)$ (viimane võrdus on tõene korrektselt leitud hääletustulemuse korral tänu skeemi $B$ homomorfsuse omadusele).

Ühe keskusega protokolliversioonil on see puudus, et keskus saab teada kõigi hääletajate hääled. $B_a(v)$ keskuse jaoks üldistatud protokollil seda puudust pole. Üldistatud protokoll rahuldab tingimusi 1, 2, 3 (juhul kui ülimalt $t$ keskust on ebaausad), 4, 5 (diskreetse logaritmi eeldusel), 6 ja 7 (hääletustulemuse leidmiseks on vaja vaid $t+1$ keskuse koostööd, üldistatud protokolli turvalisuseks läheb seega vaja, et korrumpeerunud keskusi ei oleks rohkem kui $(n/2)-1$). Kaheksas tingimus ei ole täidetud.

#### *11.2.4.3 Sunnitamatus*

Valijate sundimine (*coercion*) on ilmselt kõige raskem probleem: mida tegelikult mõeldakse selle all, et sundija (*coercer*) sunnib hääletajat teatud häält andma? Vaatleme erijuhtu, nimelt kahte võimalikku sundija juhtu: esimene kontakteerub valijaga *enne* valimiste algust ning teine kontakteerub valijaga *pärast* valimiste lõppu.

Esimene sundija on võimsam: ta saab määrata valija antud hääle ning ka valija poolt protokolli täitmise käigus genereeritud juhuslikud arvud. Seega saab selline sundija täielikult ära määrata valija poolt protokolli täitmise käigus sooritatud tegevused. Sundijal on ka lihtne avastada valija sõnakuulmatust. Sellise sundija vastu on pakutud lahendusi, mis baseeruvad mõnel füüsikalisel eeldusel, näiteks lubatakse valijal keskusega vahetada piiratud arv bitte salajase kanali kaudu; see aitab (loodetavasti) valijal teostada (keskuse poolt avastamatuid) sõnakuulmatuse akte. Teine võimalus oleks nõuda, et valija kasutaks hääletamisel sisemiselt juhuarve genereerivat sekkumiskindlat (*tamper-proof* ) riistvara, mis ei võimaldaks sundijal hääletajale peale sundida fikseeritud mündiviskeid, sest hääletajal endal pole kontrolli riistvaralise juhuarvude generaatori üle.

Teine sundija ei ole nii võimas: ta saab vaid hääletajalt nõuda, et too näitaks oma häält $v$ ning valimisprotokolli jooksul genereeritud juhuarve $\rho$. Paari $(v,\rho)$ teadmine võib sundijal võimaldada leida sellise paari $(v',\rho')$ , $v' \neq v$ , mille korral oleksid protokolli vaated võrdsed korrektse protokolli vaadetega. Eelmises jaotises esitatud protokolli puhul ei ole see võimalik (diskreetse logaritmi eeldusel).

1996. aastal töötasid Canetti ja Gennaro välja protokolli, kus "teine" sundija on edukas. Canetti ja Gennaro protokoll kasutab salgamatut krüpteerimist (*deniable encryption*). Salgamatu krüpteerimine on tõenäosuslik avaliku võtmega krüpteerimine, mis rahuldab järgmist tingimust. Olgu $m$ teade ning $r$ – saatja mündivisked. Nõuame, et suvalise krüptogrammi $C = E_{PK}(m,r)$ korral leidub saatja poolt hõlpsasti arvutatav paar $(m',r')$ , $m' \neq m$ , nii, et $C = E_{PK}(m',r')$ .

### 11.2.5 Digitaalne sularaha

Tänapäeval toimub enamik tehinguid Internet kaudu nii, et müüjale saadetakse oma krediitkaardi andmed või avatakse müüja juures eelnevalt konto. Esimesele ostmisviisile on palju vastuseisu, põhiargumendiks on mitteanonüümsus: iga ostu-müügi tehingu korral tehakse esmalt kindlaks ostja isik.

Kui me soovime igapäevaelus osta midagi oma identiteeti avalikustamata, on alternatiiviks sularaha kasutamine. Digitaalne sularaha (*digital cash*) on krüptograafiliste meetodite ja vahendite kogum, mis on mõeldud sularahal põhineva ostmise kandmiseks Internetti.

Vaatleme kõigepealt ajalooliselt üht esimest (David Chaum, 1981) pakutud digitaalse sularaha skeemi, mis põhineb asümmeetrilisel krüptograafial ning saavutab anonüümsuse.

#### *11.2.5.1 Digitaalse sularaha põhiomadused*

Põhiomadused, mida soovitakse digitaalselt sularahalt, on

1) võltsimise raskus;
2) "rahatähtede" kopeerimine peaks olema kas välditav või avastatav;
3) tarbijate anonüümsus;
4) minimaalne pöördumiste arv võrguandmebaasi poole.

### *11.2.5.2 Esimene protokoll*

Digitaalse sularaha skeem sisaldab tavaliselt kolm protokolli: ***väljavõtuprotokoll*** (*withdrawal protocol*), mis võimaldab tarbijal User pangast Bank raha välja võtta. ***Makseprotokolli*** (*payment protocol*) täitmise käigus ostab User tarnija Vendor käest kaupu digitaalsete ***müntide*** (*coin*) eest. ***Sissemakseprotokolli*** (*deposit protocol*) käigus annab Vendor mündi tagasi pangale Bank (mille eest Bank krediteerib tarnija kontot). Järgnevas eeldame, et pangal on salajane võti $SK(B)$ signeerimiseks ning vastavat avalikku võtit $PK(B)$ teavad kõik. Järgnevas tähistab $[M]_{SK(B)}$ teadet $M$ koos signatuuriga $sig_{SK(B)}(M)$:

$$[M]_{SK(B)} := (M, sig_{SK(B)}(M)).$$

**Väljavõtuprotokoll**

1. User edastab pangale Bank teate, et ta soovib välja võtta 100 krooni.
2. Bank edastab tarbijale User signatuuri $[100, \#7493]_{SK(B)}$ ("mündi") ning võtab tarbija kontolt maha 100 krooni.
3. User verifitseerib signatuuri. Kui verifitseerimine õnnestub, aktsepteerib User saadud münti.

**Makseprotokoll**

1. User maksab tarnijale saadud mündiga.
2. Vendor verifitseerib mündi signatuuri. Kui verifitseerimine õnnestub, aktsepteerib Vendor saadud münti.

**Sissemakseprotokoll**

1. Vendor annab mündi pangale Bank.
2. Bank verifitseerib mündi signatuuri. Kui verifitseerimine õnnestub, aktsepteerib Bank saadud münti.

Kui signatuuriskeem on "piisavalt" turvaline, on selge, et münte ei saa (polünomiaalses ajas) võltsida. Seevastu on väga lihtne münte dubleerida ning seega ka neid korduvalt kasutada. Samuti ei rahulda see protokoll anonüümsuse nõuet: Bank saab seostada tarbija nime mündil leiduva järjenumbriga (ning saab seega teada, kus User raha kulutab).

### *11.2.5.3 Anonüümsuse probleemi lahendus*

Analoogia igapäevaelust: User annab pangale ümbrikus oleva sedeli, millele on kirjutatud juhuslikult valitud arv ning lause "Siin on sada krooni". Sedeli peale on asetatud kopeerpaber. Bank allkirjastab ümbriku, nägemata ümbriku sisu. Allkiri kandub kopeerpaberi kaudu ka sedelile. Hiljem avab User ümbriku ning kasutab sedelit kaupade ostmiseks. Bank ei ole kunagi näinud sedelit ennast ning seega ei oska seda seostada tarbija isikuga, kuid Bank suudab alati verifitseerida enda andud allkirja.

Jaotises 10.4 defineeritud pimesignatuurid aitavad lahendada anonüümsuse probleemi digitaalse sularaha puhul. Pimesignatuure kasutav väljavõtuprotokoll on järgmine.

**Väljavõtuprotokoll**

1. User valib juhusliku arv $r \bmod n$ ning leiab suuruse $M' = M \cdot r^e \bmod n$. User saadab pangale suuruse $M'$.
2. Bank signeerib teate $M'$, leides suuruse $s' = (M')^d \bmod n \equiv M^d \cdot (r^e)^d \equiv M^d \cdot r$. Bank saadab tarbijale User teate $s'$.

3. Saanud pangalt Bank $s'$ -i, sooritab User jagamise $s = s' / r = M^d \bmod n$. Seega teab User nüüd panga signatuuri enda valitud mündile $M$.

Sissemakse- ja makseprotokollid jäävad samaks, mis enne. Viimane protokoll lahendas anonüümsuse probleemi, kuid on jäänud veel kaks probleemi:

1) User saab panka petta, pannes ümbrikusse sedeli, millel on kirjas suurem rahasumma;
2) münte saab endiselt dubleerida ning korduvalt kasutada.

#### *11.2.5.4 Rahasummade probleemi lahendus*

Esimest probleemi on võimalik lahendada järgnevalt. Pangal on iga rahaüksuse jaoks eraldi signeerimisvõti (võtmed $SK(B,1)$, $SK(B,2)$, …), või $SK(B,i)$ kehtib vaid $i$ -kroonisel rahatähel. Teine võimalik lahendus on järgmine (nn lõikamise ja valimise meetod, *cut-and-choose*).

1. User genereerib 1000 sajakroonilist münti ning saadab need kõik pimesignatuuriga varustatult pangale.
2. Bank valib ühe juhusliku mündi ning käsib tarbijal kõik ülejäänud signatuurid *valgustada* (*unblind*). Kui kõik 999 valgustatud münti on korrektsed, signeerib pank valitud ühe mündi ning tagastab selle tarbijale.

Sellise protokolli korral on tarbijal võimalik petta tõenäosusega $1/1000$. Seega on olemas protokoll, mis rahuldab nii anonüümsuse nõuet kui ka pettuse tõrjet. Järgnevalt vaatleme, kuidas võidelda müntide korduva kasutamise vastu.

#### *11.2.5.5 Digitaalne sularaha siduskontrolliga*

Pank peab arvet kõigi kulutatud müntide kohta oma sisemises andmebaasis. Makseprotokolli jooksul saadab tarnija mündi pangale ning küsib, kas seda on juba kulutatud (siduskontroll, *on-line*). Kui münti pole andmebaasis, vastab pank "ei" ning lisab mündi andmebaasi. Tarnija aktsepteerib münti. Kui münt on juba andmebaasis, vastab pank "jah" ning tarnija ei aktsepteeri münti.

See lahendus ei ole ökonoomne: esiteks peab tarnija iga makse korral pöörduma panga poole ning ootama sealt kinnitust (analoogiline olukord tekib praegu valitseval krediitkaardiga kauplemisel). Teiseks kasvab panga poolt hoitava andmebaasi suurus kiiresti üle mõistlike piiride.

Pakutud meetod hoiab ära müntide korduva kasutamise. Järgnev meetod, mis ei nõua sidusrežiimis müntide verifitseerimist, võimaldab müntide korduvkasutust *tuvastada*.

#### *11.2.5.6 Digitaalne sularaha vallaskontrolliga*

Makseprotokolli täitmise käigus kirjutab tarbija mündile juhusliku identsusstringi (*random identity string*, RIS), millel on järgmised omadused:

- RIS on erinev iga kord, kui sama mündiga makstakse;
- ainult kasutaja oskab genereerida valiidset RIS-i;
- kaks erinevat RIS-i samal mündil võimaldavad pangal tarbija nime tuvastada.

Seega, kui pank saab kaks ühesugust münti erinevate RIS-i väärtustega, on pank suuteline tuvastama petnud ostja isikut. Kui pank saab kaks ühesugust münti sama RIS-i väärtusega, pettis tarnija. Ülaltoodud

ideega tulid välja Chaum, Fiat ja Naor (1988). Idee üks võimalikke teostusi on järgmine ( $H$ on siin ühesuunaline räsifunktsioon).

**Väljavõtuprotokoll**

1. Ostja genereerib 1000 erinevat sajakroonist münti, mis näevad välja niit: $M_i = (100, \#1693 i, y_{i,1}, y'_{i,1}, y_{i,2}, y'_{i,2}, \cdots, y_{i,k}, y'_{i,k})$, kus $y_{i,j} = H(x_{i,j})$ ning $y'_{i,j} = H(x'_{i,j})$. Siin on $x_{i,j}$ ja $x'_{i,j}$ juhuslikult valitud arvud, mis rahuldavad tingimust $x_{i,j} \oplus x'_{i,j} = \text{User name}, \forall i, j$.
2. Ostja pimendab (*blind*) kõik mündid $M_i$ "juhuslikeks" teadeteks $M'_i$, kasutades ülaltoodud pimendusprotokolli, ning edastab kõik need teated pangale.
3. Pank palub ostjal valgustada 999 münti.
4. Valgustades mündid avalikustab ostja ka suurused $x_{i,j}$ ja $x'_{i,j}$.
5. Pank verifitseerib, kas (a) kõik 999 münti on sajakroonised, (b) kas $y_{i,j} = H(x_{i,j})$, $y'_{i,j} = H(x'_{i,j})$ ja $x_{i,j} \oplus x'_{i,j} = \text{User name}, \forall i, j$.
6. Pank edastab ostjale ainsa pimendatud mündi $M'_b$ signatuuri.
7. Ostja valgustab saadud signatuuri, saades mündi $M_b$ signatuuri $s_b$.

Makseprotokolli täitmise käigus lisab tarbija mündile RIS-i, kusjuures RIS on üks kahest suurusest $x_j$ või $x'_j$, $\forall i, j$.

**Makseprotokoll.**

1. Ostja edastab tarnijale paari $(M, s)$.
2. Tarnija verifitseerib panga signatuuri ning kui see kehtib, saadab ostjale tagasi juhuslikult valitud bitistringi $b = b_1 b_2 \ldots b_k$, $|b| = k$.
3. Kui $b_j = 0$, avalikustab ostja $x_j$-i, kui $b_j = 0$, avalikustab ostja $x'_j$-i.
4. Tarnija verifitseerib, kas $y_j = H(x_j)$ või $y'_j = H(x'_j)$. Kui üks neist kahest võrdusest on tõene, aktsepteerib ta mündi.

Ülaltoodud RIS-ilt nõutavad omadused on rahuldatud: tõenäosus, et erinevatel maksmistel tekitataks sama RIS, on $2^{-k}$. Ainult ostja suudab konstrueerida RIS-i, kuna $H$ on ühesuunaline. Kaks erinevat RIS-i samal mündil aitavad tuvastada ostjat, kuna sellisel juhul leidub indeks $j$, nii et pank teab nii $x_j$ kui $x'_j$ väärtust.

**Sissemakseprotokoll.**

1. Tarnija toob mündi $(M, s, RIS)$ panka.
2. Pank verifitseerib signatuuri ning kontrollib, kas münt $(M, s)$ on juba panga andmebaasis.
3. Kui münt on juba andmebaasis, võrdleb pank kahe mündi RIS-e. Kui RIS-id on erinevad, kasutas ostja münti korduvalt, kui RIS-id on samad, püüab tarnija sama münti kaks korda sisse maksta.

### *11.2.5.7 Lõpetuseks*

Lisaks loetletud omadustele nõutakse digitaalse sularaha skeemidelt veel teisi, näiteks alljärgnevaid.

1. Edastatavus (*transferability*): tarnija võib ostja käest saadud münti otsese deponeerimise asemel ise kasutada teiselt tarnijalt ostmiseks.

2. Jagatavus (*divisibility*): üks münt peab olema "jagatav" mitmeks väiksemaks mündiks nii, et saadud müntide väärtuste summa oleks võrdne algse mündi väärtusega ning et see omadus ei võimaldaks münte korduvkasutada.
3. Ühekordne signatuurivõtt (*single-term*), vastandina viimases protokollis kasutatud lõikamise ja valimise meetodile (saavutatakse tänu nullteadmustõestustele).

1997. suve seisuga ülevaate digitaalse sularaha teooriast võib leida Yiannis Tsiounise doktoritööst "*Efficient Electronic Cash: New Notions and Techniques"* (Northeastern University, Boston, Massachusetts).

# 12 AJATEMPLID

Digitaalsignatuuride õiguslikuks kasutamiseks tuleb midagi ette võtta salgamise vääramise (*non-repudiation*) tagamiseks. Traditsioonilised digitaalsignatuuri skeemid seda ei taga, sest niipea kui signatuuri omaniku salajane võti muutub avalikuks, muutub kõigi selle võtmega signeeritud dokumentide autentsus küsitavaks. See probleem on sõltumatu ka kasutatava digitaalsignatuuri liigist.

Valdavalt on elektrondokumentide haldussüsteemide turvameetmete projekteerimisel seni lähtutud turvalise sõnumivahetuse süsteemidest (näiteks turvaline elektronpost), kus tegeldakse üksnes lühiealiste sõnumite turbega. Turvalises sõnumivahetuses kasutatav turbetehnika pole piisav pika elueaga dokumentide halduseks. Oletame, et kasutaja *A* privaatvõti $D_A$ pole teatud hetkest alates enam privaatne, st keegi teine on saanud selle võtme oma valdusse. Kui selline fakt tuleb ilmsiks, on klassikaliseks lahenduseks võtme sissekandmine kõigile kasutajatele kättesaadavasse tühistusloendisse (*revocation list*). Kui kasutaja *B* saab sõnumi $D_A(X)$, kontrollib ta muuhulgas ka seda, kas võti $D_A$ pole kantud tühistusloendisse. Kui on, ei saa ta sõnumi sisu usaldada. Sellise lahenduse lihtsus ja turvalisus on kahjuks näiline. Kui *X* on näiteks laenuleping, mis kinnitab, et *A* on võtnud pangast pikaajalist laenu, siis peale võtme $D_A$ tühistusloendisse kandmist muutub kaheldavaks ka laenulepingu ehtsus, ehkki see sõlmiti tunduvalt varem kui võti $D_A$ tühistati.

Toodud näide selgitab vajadust mehhanismide järgi, mis võimaldaksid tuvastada ja ka tõestada, millal mingi elektrondokument loodi.

## 12.1 Usaldatav kolmas osapool

Kõige lihtsam viis dokumendi loomisaja tõestamiseks on kasutada usaldatavat kolmandat osapoolt, kes registreerib talle esitatud dokumendid ja lisab neile nn digitaalse ajatempli, st metaosa, mille abil saab tõestada dokumendi signeerimise aega. Ajatempli teenuse pakkujal (*Time-Stamping Service*, TSS) on oma privaatvõti $D_{TSS}$, millele vastava avaliku võtme abil saab ajatemplite ehtsust kontrollida. Samuti on vajalik, et TSS-i valduses oleks kell, mida kõik kasutajad usaldaksid kui ajaetaloni. Ajatempli taotlemine toimub järgmise protokolli abil.

1. Alice saadab TSS-ile sõnumi *Y*, millele soovib saada ajatemplit. Näiteks võib *Y* erijuhul olla sõnumi *X* digitaalsignatuur, st $Y=D_A(X)$.
2. TSS lisab sõnumile hetkeaja *t*, signeerib liitsõnumi ja saadab Alice'ile sõnumi $S=D_{TSS}(Y,t)$, mida võibki kasutada signeeritud sõnumi ajatemplina.

Kui TSS-i avalik võti on kõigile teada, on võimalik lihtsalt kontrollida, millal dokument TSS-ile esitati. Kirjeldatud süsteemi võib võrrelda juba ammu kasutatava võttega: sõnum saadetakse lahtisel postkaardil iseenda aadressil, et saada sinna postiteenuse templit, kus teatavasti on ka aeg. Postiteenus täidab sel juhul TSS-i rolli. Sellel lihtsal ajatemplisüsteemil on aga kaks olulist puudust.

- Pole selge, mis saab siis, kui võti $D_{TSS}$ pole ühel hetkel enam privaatne. Sellest hetkest alates muutub kõigi seni väljaantud ajatemplite ehtsus küsitavaks.
- TSS peab olema absoluutselt usaldatav, st kõik kasutajad peavad uskuma, et TSS ei anna tahtlikult, näiteks kasu saamise eesmärgil, välja võlts-ajatempleid.

Need puudused teevad vaadeldud süsteemi praktilise kasutamise raskeks, kuid veel kaheksakümnendatel aastatel usuti, et ajatemplite süsteemile ei ole paremat lahendust kui kolmas usaldatav osapool.

## 12.2 Linkimine

Aastal 1990 töötasid Stuart Haber ja Scott Stornetta välja uue ajatemplite süsteemi, juhtides tähelepanu asjaolule, et ajatemplite süsteem paberdokumentide jaoks ühe asutuse või organisatsiooni piires on juba ammu välja mõeldud. Selleks on dokumentide register, st vihik, milles on nummerdatud read dokumentide registreerimiseks. Kui vihikut täidetakse järjest, siis on sinna hiljem väga raske dokumente lisada ilma nähtavaid jälgi jätmata. Sama ideed saab kasutada ka elektrondokumentide korral. Ridade järgnevus registris tagatakse räsifunktsiooni *H* kasutamisega. Ajatempliteenuse andja (TSS) on kohustatud pidama väljaantud ajatemplite registrit, mille korrektset täitmist saavad kõik süsteemi kasutajad kontrollida. Ajatempli väljaandmise protokoll oleks sel juhul järgmine.

1. Kasutaja A saadab TSS-ile mingi digitaalse infokogumi *Y*.
2. Teades, et A taotlus kannab järjenumbrit *n*, leiab TSS sõnumilühendi

$$H_n = h(n,ID_A,t,Y),$$

kus $ID_A$ on kasutaja *A* identifikaator, *t* on hetkeaeg ja *h* on mingi räsifunktsioon, mis ei tarvitse olla sama, mis *H*. Seejärel arvutab TSS nn linkimisteabe

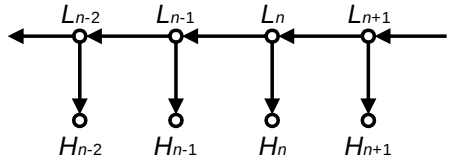
$$L_n = H(H_n, L_{n-1})$$

ja saadab *A*-le tagasi sõnumi $(n,ID_A,t,L_{n-1},D_{TSS}(L_n))$, mida kasutatakse kui ajatemplit.

Lisaks ajatempli väljaandmise teenusele võib iga klient küsida TSS-ilt ka mistahes $H_n$-i ja $L_n$-i, mis on jäädvustatud TSS-i poolt. Kasutatavate räsifunktsioonide ühesuunalisuse tõttu on tekkinud ahelat väga raske võltsida või uusi ajatempleid juurde lisada mujale kui ahela lõppu. Ei ole enam eriline katastroof, kui TSS-i privaatvõti muutub avalikuks, sest kogu ajatemplite süsteemi turvalisus on seotud pigem kasutatavate räsifunktsioonide turvalisusega. Räsifunktsiooni ühesuunalisuse omadust võib ligikaudu väljendada ka järgmiselt:

- kui *H(X)* oli teada mingil ajahetkel *t*, pidi suurus *X* olema teada enne ajahetke *t*.

Seda omadust võib väljendada ka nii: kui *A* usub, et *X* on värske ajahetkel (pole teada kellelegi enne ajahetke) *t*, siis ta usub, et ka *H(X)* on värske ajahetkel *t*. Ajatemplite vahelist ühesuunalist andmesõltuvust iseloomustab skeem joonisel 86.




**Joonis 78. Lineaarne linkimisskeem**

Kui *A* usub, et $L_{n+1}$ eksisteeris ajahetkel *t*, siis ta usub ka, et ajahetkel *t* eksisteerisid $L_n$, $H_n$, $L_{n-1}$, ... Kui lisaks sellele *A* usub, et $L_{n-1}$ oli värske ajahetkel *t'*, siis ta usub, et ka $L_n$, $L_{n+1}$, ... olid värsked ajahetkel *t'*. Seega ta usub, et $L_n$ moodustati millalgi ajahetkede *t'* ja *t* vahel.
Nagu võis tähele panna, ei läinud eelnenud arutelus kusagil vaja TSS-i privaatvõtme turvalisust ega ka ühtki eeldust TSS-i korrektse käitumise kohta, mistõttu võib järeldada, et saadud süsteem on tunduvalt töökindlam ja raskemini rünnatav kui eelmises alapunktis kirjeldatud usaldatavat kolmandat osapoolt nõudev süsteem.

## 12.3 Räsifunktsioonide turvavajadused

Nagu juba öeldud, sõltub kogu ajatemplite süsteemi turvalisus kasutatavate räsifunktsioonide *h* ja *H* turvalisusest, kusjuures oluliselt erineval määral. Funktsioonist *H* sõltub dokumentide ajaline järgnevus, funktsioonist *h* aga tembeldatavate dokumentide seos vastavate ajatemplitega.
Kui ühel hetkel tekib lihtne võimalus räsifunktsiooni *H* pöörata, st etteantud *X* korral on lihtne leida teist argumenti $X' \neq X$, nii et $H(X)=H(X')$, saab võltsida milliseid tahes eelnevaid ajatempleid ja nende ajalist järjestust. Kui näiteks TSS soovib võltsida *n*-ndat ajatemplit, tuleb tal esmalt arvutada enda jaoks sobiv

$$H'_n = h(n,ID'_n,t',Y').$$

Edasi on kaks võimalust. Esiteks, TSS võib leida mingi $H'_{n+1}$, nii et

$$L'_n = H(H'_n,L_{n-1})$$
$$L_{n+1} = H(H'_{n+1},L'_n).$$

Teine võimalus *n*-ndat ajatemplit võltsida on leida $L'_{n-1}$ ja $H'_{n-1}$, nii et

$$L'_{n-1} = H(H'_{n-1},L_{n-2})$$
$$L_n = H(H'_n,L'_{n-1}).$$

Selline võltsimine pole võimalik, kui funktsioon *H* on ühesuunaline. Paneme tähele, et funktsiooni *H* murdmisel tuleb mõlemal juhul võltsida vähemalt üks $L_n$ ning seega võib murdjaks olla vaid TSS ise. Teisiti on lugu siis, kui lahti murtakse räsifunktsioon *h*. Sellisel juhul saab leida sobiva neliku $(n,ID'_n,t',Y')$, nii et $h(n,ID'_n,t',Y')=h(n,ID_A,t,Y)$, jättes puutumata $L_n$-id. See tähendab, et funktsiooni *h* murdumise korral võib ajatemplite süsteemi rünnata suvaline kolmas piisavalt nutikas osapool (muuhulgas ka TSS ise) ning funktsiooni *H* turvalisusest pole enam kasu. See on üks põhjusi, miks peab räsifunktsioon *h* olema turvalisem kui *H*.

Viimasest arutelust järeldub, et kasutatavad räsifunktsioonid peavad lisaks ülalnimetatud ühesuunalisuse omadusele rahuldama ka nn kollisioonivabadust: antud *X* korral peab olema raske leida väärtust $X' \neq X$, nii et $H(X)=H(X')$. Funktsiooni *h* suurem turvalisus väljendub eelmise lause sõnale "raske" pandud suuremas rõhus.

Tänapäeval ei teata ühtegi räsifunktsiooni, mille turvalisus oleks kindel näiteks järgmise 10 aasta jooksul (üks esimesi laiemalt kasutusele võetud räsifunktsioone, MD4, murti lahti umbes 5 aastat pärast leiutamist). On olemas vaid kiired räsifunktsioonid, mille lahtimurdmiseks puudub piisav teadmus (nimetada võiks funktsioone SHA-1 ja RIPEMD-160) ning aeglased räsifunktsioonid, mille turvalisus põhineb mõnel hästituntud raskesti arvutataval matemaatilisel funktsioonil, näiteks diskreetsel logaritmil.

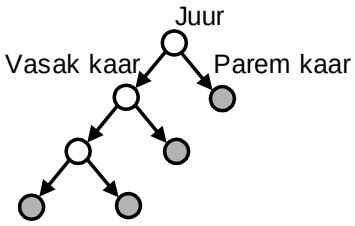
## 12.4 Lineaarse linkimisviisi puudused

Lineaarses linkimisskeemis on iga ajatempel seotud ühe, talle vahetult eelneva, ajatempliga. Nagu nägime, tagas toodud meetod küll oluliste turvanõuete täitmise, kuid see skeem on siiski väga ebapraktiline järgmise kahe olulise puuduse tõttu.

- Vajadus, et ajatempel säilitaks mälus terve ajatemplite ahela. Kui oletada, et reaalselt funktsioneerivas süsteemis on ajatempleid umbes $10^{11}$, on selge, et mälust kipub puudu jääma.
- Ajatemplite võrdlemisel tehtav töö on võrdeline ajatemplite järjenumbrite vahega; tehtava töö hulga ülatõke on *O(n).* See teeb aga ajatemplite võrdlemise äärmiselt ebapraktiliseks, kuna üheainsa ajatempli kontrollimiseks tuleks halvemal juhul teha niisama palju tööd kui TSS on teinud kogu oma eksistentsi vältel.

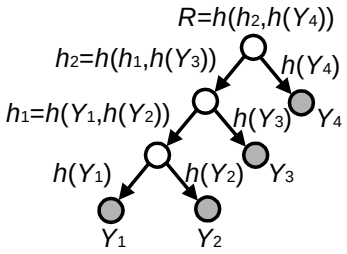
## 12.5 Merkle'i autentimispuud

Ühe võimaluse seda olukorda parandada esitasid Bayer, Haber ja Stornetta aastal 1992. TSS-i tegevus jagatakse ajaliselt tsükliteks, mille lõpus väljastatakse nn koond-ajatempel. Mingis tsüklis välja antud ajatempli linkimisteave sõltub samas tsüklis eelnevalt väljastatud ajatemplitest ning eelneva tsükli koond-ajatemplist. Käesoleva tsükli jooksul väljastatud dokumendid paigutatakse mõtteliselt kahendpuu lehtedesse (vt puu tumedad tipud joonisel 87). Kahendpuu igast tipust *T* väljub ülimalt kaks suunatud kaart: vasak ja parem, neist esimese sihttippu nimetatakse tipu *T* vasakjärglaseks ja teist paremjärglaseks.




**Joonis 79. Kahendpuu**

Puu tippudele arvutatakse märgendid nn Merkle'i autentimispuu skeemi järgi (vt joonis 88). Puu suvalise sisemise tipu *T* märgendi väärtus on $h_T := h(h_{L(T)}, h_{R(T)})$, kus $h_{L(T)}$ ja $h_{R(T)}$ on vastavalt tipu *T* vasak- ja paremjärglase väärtused. Sellise skeemi korral peab TSS säilitama vaid kõigi tsüklite koond-ajatemplid $s_n := h(R_r, s_{n-1})$, kus $R_r$ on tsüklis *r* koostatud autentimispuu juurtipu märgendi väärtus. Kõik ülejäänu antakse kaasa ajasertifikaatidele (dokumentide individuaalsetele ajatemplitele), mis sisaldavad nn autentimisteed. Näiteks joonisel 88 koosneb dokumendi $Y_1$ autentimistee väärtustest ($h(Y_2), h(Y_3), h(Y_4)$).




**Joonis 80. Merkle'i autentimispuu**

Kui dokumendid räsitakse enne ajatempli võttu, on iga ajatempli pikkus $O(\log n)$, mille saab vähendada ka konstandiks, kasutades nn ühesuunalisi akumulaatoreid.

Kahjuks on koond-ajatemplitel ka puudusi. Neist märkimisväärseim on võimatus võrrelda ajaliselt ühe tsükli jooksul antud ajatempleid. Seda puudust saab leevendada tsükli ajalise kestuse vähendamisega. Näiteks praegu edukaimas ajatemplisüsteemis Digital Notary™, mida pakub Surety Technologies, on ühe tsükli kestus üks sekund. Tsükli pikkuse liigse vähendamisega kaasneb aga alati ka tsükli jooksul välja antavate ajatemplite arv, mistõttu autentimispuu põhilist eelist, logaritmilist salvestusmahtu, ei saa kasutada efektiivselt.

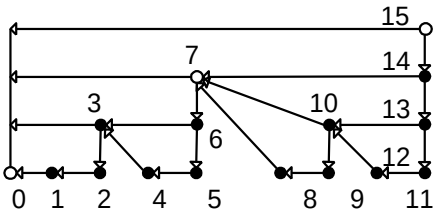
## 12.6 Binaarsed linkimisskeemid

Binaarse linkimisskeemi töötasid välja Buldas, Laud, Lipmaa ja Villemson aastal 1998. Põhiajend oli vältida eelmise alajaotise lõpul esitatud järeleandmise vajadust, st puuskeemide ajalise täpsuse suurendamist efektiivsuse arvel.

Binaarseks linkimisviisiks nimetame ajatempli arvutamise skeemi, kus iga väljaantud ajatempel $L_n$ on seotud mitte ainult oma vahetu eellasega $L_{n-1}$, vaid ka veel ühe sobivalt valitud ajatempliga $L_{f(n)}$. Linkimisskeemi iseloomustab ajatemplite $L_n$ omavahelise sõltuvuse valem

$$L_n = H(H_n, L_{n-1}, L_{f(n)}),$$

kus $f$ on mingi funktsioon. On võimalik näidata, et kui valida funktsioon $f$ sobival viisil, on võimalik saada ühesuunalise sõltuvuse verifitseerimisel tehtava töö hinnanguks $O(\log n)$, mis teeb linkimisviisi praktikas kasutatavaks.



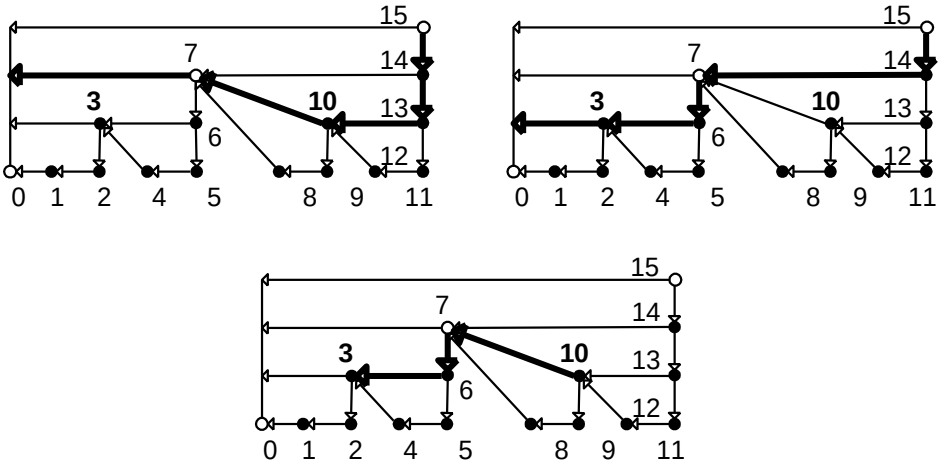

**Joonis 81. Binaarsest linkimisskeemi näide, tsükli pikkusega 15**

Ka binaarse linkimisskeemiga ajatemplisüsteemides jagatakse TSS-i töö tsükliteks. Ajatempli võtt mingi tsükli $r$ jooksul toimub järgmise protokolli alusel.

- Kasutaja $A$, kes soovib saada ajatemplit dokumendile $X_n$, arvutab esialgu dokumendi $X$ sõnumilühendi $H_n=H(X_n)$ ja saadab selle TSS-ile ajatempli saamiseks.
- TSS arvutab $L_n = H(H_n, L_{n-1}, L_{f(n)})$.
- TSS saadab kasutajale $A$ tagasi signeeritud ajatempli $D_{\mathrm{TSS}}(n,L_n)$.
- Kui tsükkel $r$ on lõppenud, avaldab TSS mingis eelnevalt kokku lepitud ajalehes $r$-nda tsükli koondajatempli $R_r$.
- Kui tsükkel $r$ on lõppenud, on kliendil $A$ võimalik küsida TSS-ilt autentimisteed eelmise tsükli koondajatemplist $R_{r-1}$ $n$-nda ajatempli väärtuseni $L_n$ (nn pea, $\mathsf{head}(n)$) ja ajatemplist $L_n$ äsjalõppenud tsükli koondajatemplini $R_r$ (nn saba, $\mathsf{tail}(n)$).
- Dokumendi $X_n$ ajasertifikaat $\tau(X_n)$ koosneb tsükli- ja dokumendinumbrist, kahest autentimisteest ja TSS-i signeeritud vastusest, st

$$\tau(X_n) = (r,n,D_{\mathrm{TSS}}(n,L_n),\mathsf{head}(n),\mathsf{tail}(n)).$$

Kui funktsioon $f$ on antimonotoonne, st kui võrratusest $f(n) < m \le n$ järeldub $f(n) \le f(m)$, saab mistahes kahe ajasertifikaadi $\tau(X_i)$ ja $\tau(X_j)$ põhjal koostada vastavate ajatemplite $L_i$ ja $L_j$ vahel ühesuunalise autentimisahela. See tuleneb faktist, et autentimisteedele linkimisskeemi graafis vastavd ahelad lõikuvad.




**Joonis 82. Näide autentimisteede head(10) ja tail(3) lõikumisest antimonotoonses skeemis**

**Näide**. Joonis 82 on esitatud (paksude nooltega) autentimisteed head(10) ja tail(10) (ülal vasakul), head(3) ja tail(3). Teed head(10) ja tail(3) lõikuvad punktis 7. Tippude 3 ja 10 vahelisele autentimisteele vastav ahel graafis on (3,6,7,10), mis on näidatud joonise alumises osas.

Antimonotoosed binaarsed linkimisskeemid võimaldavad seega dokumentide ajalisi suhteid taastada isegi siis, kui ajatempli teenuse andja TSS-i andmebaas lakkab eksisteerimast, sest:

- kui kaks dokumenti *X* ja *Y* on ajaliselt "kauged", st tembeldatud eri tsüklites, saab nendevahelisi ajalisi suhteid selgitada ajalehes avaldatud ajatemplite põhjal;
- kui *X* ja *Y* on ajaliselt "lähedased", st tembeldatud ühes ja samas tsüklis, on nendevahelisi ajalisi suhteid võimalik selgitada ajasertifikaatide $\tau(X)$ ja $\tau(Y)$ põhjal.

Ajasertifikaatide pikkus on logaritmiline. Näiteks kui ajatemplite arv tsüklis on 10 miljonit, tuleb ajasertifikaadi pikkuseks paremate skeemide korral umbes 2K baiti.

Seega võimaldavad binaarsed linkimisskeemid oluliselt vähendada ajatempliteenuse andmisel usaldatava kolmanda osapoole (TSS) rolli.

## 12.7 Signeerimine koos ajatempliga

Absoluutset aega pole võimalik määrata ei krüptograafiliselt ega ka füüsikaliselt, mistõttu ei ole ühegi krüptograafilise protokolli abil võimalik seda tuvastada ega tõestada. Ühesuunalisuse omadus on ainus digitaalandmete maailmas aega genereeriv tunnus. See aeg on aga suhteline, st fikseerib, kumb kahest etteantud dokumentdist on tekkinud varem.

Iga kasutaja saab teatud täpsusega tekitada enda jaoks lokaalse absoluutse aja, kui ta regulaarselt ja oma kella järgi fikseeritud ajavahemike järel genereerib juhuarve ja võtab neile ajatemplid. Hiljem võib ta võrrelda teisi ajatempleid ja otsustada, millises ajavahemikus (tema kella järgi) on need moodustatud.

Järgnev protokoll näitab, kuidas saab ajatempli teenust kasutades tõestada, et mingi signatuur Σ on moodustatud teatud ajavahemikus.

- Signeerija *A* küsib vahetult enne sõnumi *X* signeerimist TSS-ilt kõige värskema ajatempli $L_m$.
- Signeerija moodustab signatuuri $\Sigma=D_A(X,L_m)$.
- Signeerija saadab signatuuri Σ ajatemplikeskusele (TSS) ajatempli saamiseks.
- TSS saadab tagasi signeeritud ajatempli $L_n$.

Ühesuunaliste sõltuvuste $L_m \leftarrow \Sigma \leftarrow L_n$ abil on võimalik tuvastada, et signatuur Σ on moodustatud pärast $L_m$ moodustamist ja enne $L_n$-i moodustamist. Kui nüüd kasutaja *B* on võtnud ajatempli Lm' ($m'<m$) talle teadaoleva aja järgi 15.00 ja ajatempli $L_{n'}$ ($n'>n$) talle teadaoleva aja järgi 15.30, siis võib *B* olla kindel, et signatuur Σ on moodustatud ajavahemikus 15.00–15.30 tema kella järgi.

# 13 SERTIFITSEERIMINE

Kasutaja *A* avalik võti on vaid suur täisarv, mis kuidagi ei seostu kasutaja *A* isikuga. Avaliku võtme autentsuses veendumiseks võib näiteks selle omanikuga isiklikult kohtuda ja võtmeid vahetada. Võrgu kaudu saadetud avalikke võtmeid ei saa usaldada vahendusründe võimaluse tõttu. Paljudel kasutajatel ei ole lihtsalt võimalik kõigi teistega näost näkku kohtuda ainuüksi nende geograafilise asukoha vms tõttu. Seega on vaja mingit muud meetodit avalike võtmete autentseks levitamiseks.

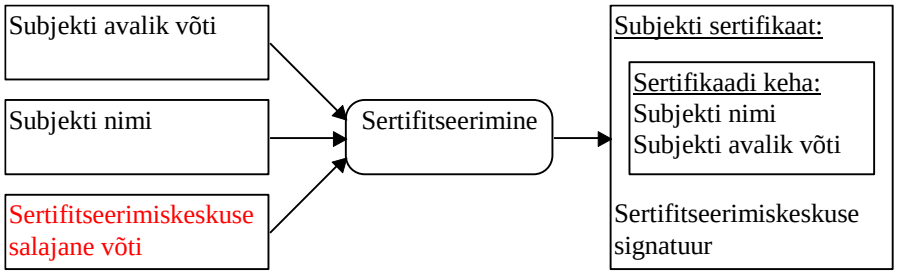
Mängu tuuakse kolmas osapool, kelle ülesanne on selliste tõendite jagamine, mis kinnitavad avaliku võtme kuulumist konkreetsele kasutajale. Tõendid võivad olla nii paber- kui ka elektrondokumendid ja neid nimetatakse avaliku võtme sertifikaatideks, kolmandat osapoolt aga sertifitseerimiskeskuseks. Tõend sisaldab kasutaja nime ja avaliku võtme ning on varustatud kolmanda osapoole allkirja või digitaalsignatuuriga, mis võimaldab kontrollida tõendi autentsust, eeldusel et sertifitseerimiskeskuse avalik võti on kõigile teada ja autentne.

Loomulikult kerkivad ka kolmanda osapoole avaliku võtme levitamisel samasugused probleemid, mis iga muugi avaliku võtme puhul, aga kui varem tuli kõigi suhtluspartneritega avalikke võtmeid vahetada, piisab nüüd sellest, et hangitakse endale sertifitseerimiskeskuse avaliku võtme autentne koopia, misjärel ongi võimalik kõigi selle keskuse poolt sertifitseeritud kasutajatega turvaliselt suhelda.

## 13.1 Avaliku võtme sertifikaadid

Avaliku võtme sertifikaadid on vahendid avalike võtmete turvaliseks salvestuseks ebaturvalisse keskkonda ja edastuseks ebaturvaliste kanalite kaudu, eesmärgiga teha avalik võti teistele kättesaadavaks, nii et tema autentsus (st tema staatus selle olemi avaliku võtmena) ja kehtivus oleksid verifitseeritavad.

Avaliku võtme sertifikaat on andmestruktuur, mis koosneb andmeosast ja signatuuriosast. Andmeosa sisaldab miinimumina avaliku võtme ja kasutajaolemi tunnuse. Signatuuriosa sisaldab sertifitseerimiskeskuse digitaalsignatuuri, mis on moodustatud andmeosast ja seob seega omavahel avalikku võtit ja kasutajaolemit tema tunnuse kaudu.




**Joonis 83. Avaliku võtme sertifikaadi väljaandmine**

Sertifitseerimiskeskus on usaldatav kolmas osapool, kelle signatuur sertifikaadil kinnitab avaliku võtme ja kasutajaolemi seost, mille täpsem tähendus ja kontekst (st milliseks otstarbeks saab antud võtit kasutada) tuleb eraldi määratleda kas atribuudisertifikaadi väljaandmisega või turvapoliitika sättena. Sertifikaadis sisalduv kasutajaolemit identifitseeriv string (eraldusnimi, *distinguished name*) peab üheselt identifitseerima kasutajaolemi vaadeldava süsteemi piires. Sertifikaadid võivad sisaldada veel järgmisi andmeid.

1. Avaliku võtme kehtivusaeg.
2. Sertifikaadi number või identifikaator.
3. Lisainformatsioon kasutajaolemi kohta (aadress, meiliaadress vms).
4. Lisateave võtme kohta (näiteks kasutatav genereerimisalgoritm).
5. Kasutajaolemi identifitseerimise ja võtmepaari genereerimise kvaliteedinõuded.
6. Informatsioon signatuuri verifitseerimiseks (signeerimisalgoritm, identifikaator, sertifitseerimiskeskuse nimi)
7. Avaliku võtme staatus (tühistussertifikaatide jaoks).

### 13.1.1 Sertifikaatide väljaandmine

Enne avaliku võtme sertifikaadi väljaandmist kasutajale või olemile *A*, peab sertifitseerimiskeskus rakendama meetmeid (enamasti organisatsioonilisi) kasutaja *A* isiku tuvastuseks ja kontrollima, kas esitatud avalik võti ikka tõepoolest kuulub kasutajale *A* ega ole kellegi teise avaliku võtme koopia. Võib esineda kaks erijuhtu.

- Sertifitseerimiskeskus genereerib võtmepaari, kontrollib kasutaja *A* isikut ja edastab kasutajale *A* salastusega ja autenditud kanali kaudu vastava privaatvõtme. Seejärel annab keskus välja avaliku võtme sertifikaadi.
- Kasutaja genereerib ise võtmepaari, edastab avaliku võtme autenditud kanali kaudu sertifitseerimiskeskusele, sertifitseerimiskeskus kontrollib edastaja isikut ja annab väja sertifikaadi nagu on kirjeldatud eelmises lõigus.

Teisel juhul peab keskus kontrollima, kas kasutaja ikka teab esitatud avalikule võtmele vastavat privaatvõtit. See väldib rünnet, kus kasutaja esitab kellegi teise avaliku võtme ja seob selle enda isikuga. Kontrolli teostamiseks peab kasutaja *A* esitama elektrondokumendi, mis sisaldab kasutaja avaliku võtme

ja tema nime vms ning mis on signeeritud sellele võtmele vastava privaatvõtmega. Kontrolli võib ka teostada nõudes kasutajalt andmekogumi $h(r_1,r_2)$ signeerimist, kus $r_1$ on kasutaja genereeritud juhuarv, $r_2$ on keskuse genereeritud juhuarv ja $h$ on mingi räsifunktsioon.

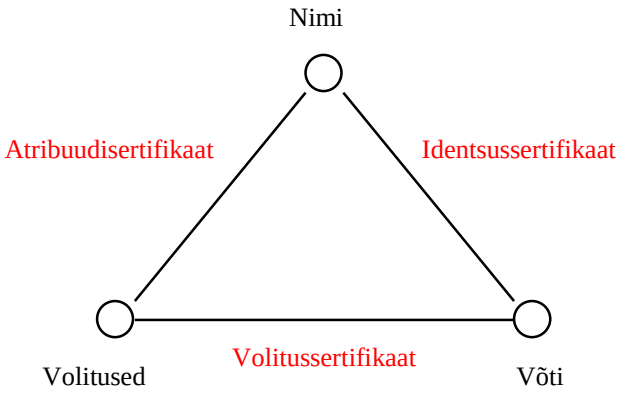
### 13.1.2 Sertifikaatide kasutamine ja verifitseerimine

Protsess, mille käigus kasutaja B kontrollib kasutajale A väljaantud avaliku võtme sertifikaati, koosneb järgmistest etappidest sammudest.

- *B* hangib autentsel viisil sertifitseerimiskeskuse avaliku võtme. Seda sammu tuleb sooritada ainult üks kord.
- B leiab kasutajat A üheselt identifitseeriva stringi.
- B hangib kasutaja A avaliku võtme sertifikaadi, kasutades identifitseerivat stringi.
- B kontrollib sertifikaadi kehtivusvahemikku.
- B kontrollib sertifitseerimiskeskuse avaliku võtme kehtivust.
- B kontrollib sertifitseerimiskeskuse signatuuri sertifikaadis.
- B kontrollib, kas sertifikaat pole tühistatud.

### 13.1.3 Atribuudisertifikaadid ja volitussertifikaadid

Avaliku võtme sertifikaadid võivad peale kasutaja nime ja avaliku võtme sisaldada ka muid andmeid, kuid need ei ole kohustuslikud. Atribuudisertifikaadid on sarnased avaliku võtme sertifikaatidele, kuid neid kasutatakse muude omaduste (atribuutide) sidumiseks isiku või olemiga. Näiteks võib atribuudisertifikaate kasutada avaliku võtme kasutusala piiramiseks või selle kasutamise juriidilise toime vähendamiseks jne.




**Joonis 84. Sertifikaatide liigid**

Volitussertifikaadid seovad omavahel avaliku võtme ja selle võtme volitused. Kuna praktilistes süsteemides kasutatakse autentimisvahendina avalikke võtmeid, on volitussertifikaat üks kõige levinumaid sertifikaadi vorme

## 13.2 Turvadomeenid

Turvadomeen (*security domain*) on mingi kindla sertifitseerimiskeskusega seotud piirkond, milles sisalduvad olemid (subjektid) usaldavad seda sertifitseerimiskeskust. Usaldus sertifitseerimiskeskuse vastu rakendub olemipõhise ühisvõtme või parooli kaudu (sümmeetrilisel juhul) või sertifitseerimiskeskuse avaliku võtme kaudu (sümmeetrilisel juhul). See võimaldab turvaliste ühenduskanalite moodustamist olemi ja sertifitseerimiskeskuse vahel ja ka kahe olemi vahel ühes ja samas domeenis. Turvadomeenid ise võivad korralduda sarnastel alustel suurematekss domeenideks.

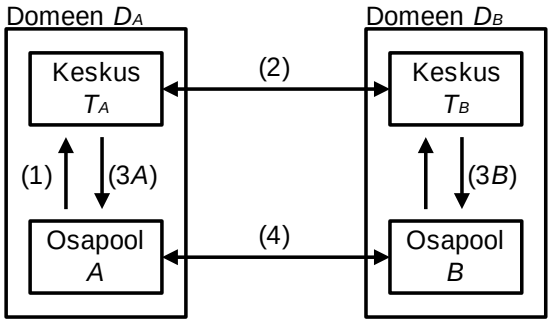
### 13.2.1 Usaldus kahe domeeni vahel

Kaks osapoolt *A* ja *B*, kes kuuluvad erinevatesse turvadomeenidesse, mis vastavad sertifitseerimiskeskustele $T_A$ ja $T_B$, võivad samuti soovida omavahel turvaliselt suhelda. Seda võib saavutada kahel viisil.

1. Kehtestada ühisvõti $K_{AB}$, mida mõlemad osapooled usaldavad, st nad usuvad, et keegi kolmas seda võtit ei tea.
2. Tekitada "usaldussild" kahe eri domeeni $T_A$ ja $T_B$ vahel, st tekitada ühine usaldus ühe või mitme avaliku võtme vastu.

Kui $T_A$-l ja $T_B$-l juba on ühine usaldussuhe, piisab *A* ja *B* vaheliseks turvaliseks suhtluseks turvaliste kanalite tekitamisest paaride $(A,T_A)$, $(T_A,T_B)$ ja $(T_B,B)$ vahel.

Kui $T_A$-l ja $T_B$-l ei ole ühist usaldussuhet, võib kasutada kolmandat sertifitseerimiskeskust $T_C$ kui vahendajat.




**Joonis 85. Usalduse loomine eri domeenide vahel**

#### *13.2.1.1 Usaldatav sümmeetriline võti*

Ühisvõtme kehtestamine võib toimuda ükskõik millise sobiva võtmekehtestusprotokolli abil. Üldiselt võib *A* ja *B* vaheline usalduse tekitamise protseduur kulgeda järgmiselt.

(a) A saadab $T_A$-le avalduse suhelda turvaliselt kasutajaga *B* (Joonis 85, 1).
(b) $T_A$ ja $T_B$ genereerivad lühiealise salajase võtme $K_{AB}$ (Joonis 85, 2).
(c) $T_A$ ja $T_B$ edastavad võtme $K_{AB}$ vastavalt kasutajatele *A* ja *B* garanteerides salastuse ja autentsuse (Joonis 85, 3A,3B).
(d) *A* kasutab võtit $K_{AB}$ turvaliseks suhtluseks *B*-ga (Joonis 85, 4).

Selles olukorras võib paari $(T_A,T_B)$ vaadelda kui ühist sertifitseerimiskeskust, millega *A* suhtleb $T_A$ kaudu ja *B* suhtleb $T_B$ kaudu.

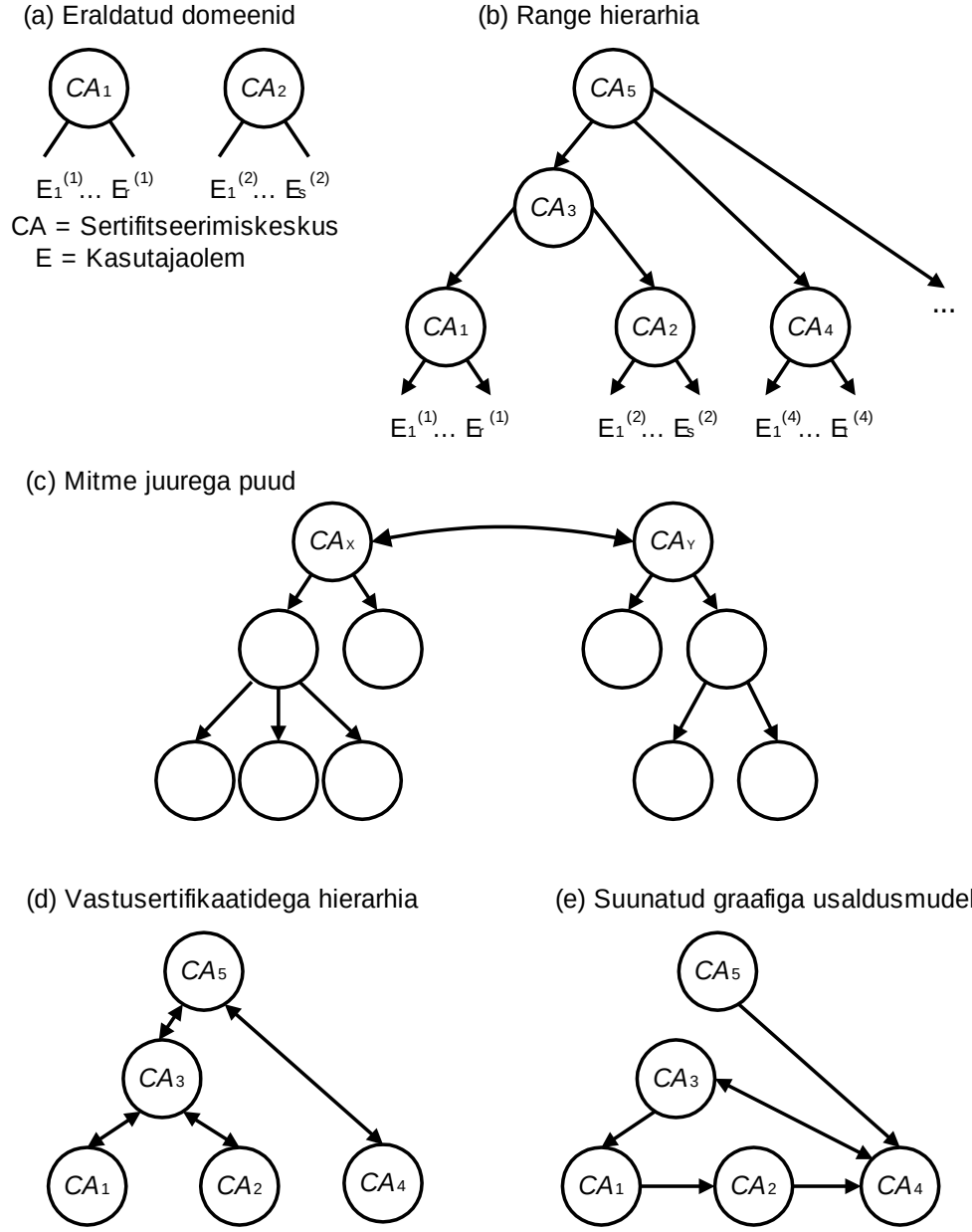
### *13.2.1.2Usaldatav avalik võti*

(a) A küsib $T_A$-lt kasutaja *B* usaldatavat avalikku võtit (Joonis 86, 1)
(b) $T_A$ küsib selle $T_B$-lt, garanteerides autentsuse (Joonis 86, 2)
(c) $T_A$ edastab võtme *A*-le, garanteerides autentsuse (Joonis 86, 3A)
(d) *A* kasutab avalikku võtit suhtluses *B*-ga (Joonis 86, 4).

Ristsertifikaadiks nimetatakse sertifikaati, mille on genereerinud sertifitseerimiskeskus ja mis sertifitseerib mingit teist sertifitseerimiskeskust.

## 13.3 Usaldusmudelid

Usaldussuhete korraldamiseks paljude sertifitseerimiskeskuste vahel on palju võimalusi. Neid korraldusviise nimetatakse usaldusmudeliteks või ka sertifitseerimistopoloogiateks. Usaldussuhted sertifitseerimiskeskuste vahel määravad, kuidas ühe keskuse kliendid saavad kasutada teiste keskuste poolt väljaantud sertifikaate.




**Joonis 86. Usaldusmudelite tüüpe**

Usaldusmudelite põhitüübid (vt Joonis 86) on järgmised.

(a) Eraldatud domeenid. Eri domeenidesse kuuluvad kasutajad ei saa selle süsteemi abil turvaliselt suhelda.
(b) Range hierarhia. Kõik sertifitseerimiskeskused alluvad ühele juurkeskusele, keda kõik peavad usaldama.
(c) Mitme juurega puud. Juurkeskuste vahel toimub ristsertifitseerimine.
(d) Vastusertifikaatidega hierarhia. Samasugune nagu range hierarhia, kuid lisaks sellele, et ülemad alamaid sertifitseerivad, toimub siin ka ülemate sertifitseerimine.
(e) Suunatud graaf. Kõige üldisem usaldusmudel, mis ei esita mingeid struktuurilisi kitsendusi.

## 13.4 X.509

### 13.4.1 Eraldusnimed

Alustame sellest eraldusnimest (*distinguished name*), mida sertifikaadis kasutatakse ta omaniku ning väljaandja identifitseerimiseks. Algselt määras see nimi tee standardile X.500 vastavas kataloogipuus, tänapäeval määrab ta üsna suvalisi andmeid. Eraldusnimi (*distinguished name*) peaks olema globaalselt ühene nimi, millega kedagi või midagi identifitseeritakse. Ühesus tagatakse hierarhilise nimeskeemi kasutamisega (umbes nagu DNS puhul).

Näiteks järgmine neljatasemeline nimi annab teada, et isik on Eesti Vabariigis asuva organisatsiooni (C=EE), mille nimi on IOC, (O=IOC), IT nimelise allüksuse (OU=IT) töötaja, kelle pärisnimi on Riho Leevike (CN=Riho Leevike). Maailmas võib olla mitmeid organisatsioone, mille nimelühend on IOC (näiteks Rahvusvaheline Olümpiakomitee – *International Olympic Committee*), kuid ainult üks neist asub Eestis. Maailmas võib elada sadu Riho Leevikesi, kuid selle organisatsiooni selles allüksuses töötab neid ainult üks.

Tavaliselt garanteerib üks sertifitseerimisteenuse pakkuja, et ta ei väljasta sama nimega sertifikaate. Samas võivad näiteks Thawte ja Verisigni (kaks sertifitseerijat) anda sama nimega sertifikaadi erinevatele inimestele. Mõlema sertifitseerija usaldamine võib tekitada suuri probleeme.

**Tabel 28. Eraldusnimedes kasutatavad atribuudid**

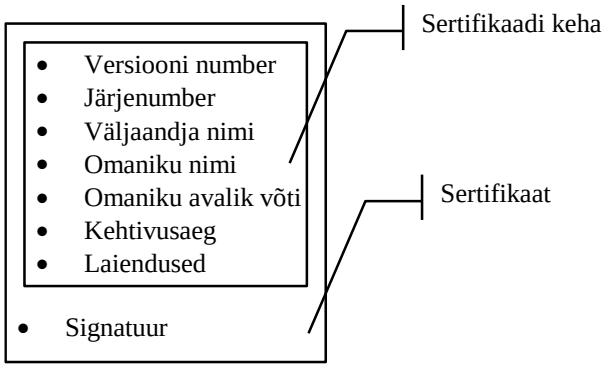
| | | |
|---|---|---|
| **Geograafilise asukohaga seotud atribuudid** | | |
| **C** | Maa kood | Maa kood standardi ISO 3166 järgi. Eesti on näiteks EE, Saksamaa DE jne |
| **ST** | Osariigi, maakonna või provintsi nimi | Näiteks: Ohio, Alabama, Läänemaa |
| **L** | Asukoha nimi (näiteks linna või asula nimi) | Näiteks: Viin, Tapa, Metsküla |
| **Organisatsioonilise kuuluvusega seotud atribuudid** | | |
| **O** | Organisatsiooni nimi | Näiteks: IOC, "RAS Merelaevandus" |
| **OU** | Allüksuse nimi | Näiteks: IT, "Turundus & Rahandus" |
| **Muud atribuudid** | | |
| **CN** | Üldkasutusnimi | Nimi, mille all subjekti üldiselt tuntakse. Võib sisaldada kõikvõimalike tiitleid, lühendeid jms. Näiteks: "Juhan Saar", "Hr. Jaak Ploomipuu". Serverite puhul hostinimi: hansa24.hansa.ee |

Sertifitseerimispoliitikaga võib kehtestada kindlad reeglid eraldusnimede moodustamiseks, sõltuvalt sellest, millisele subjektile nime koostatakse. Näiteks sertifikaadi väljastamisel eraisikule ei tohiks eraldusnimi sisaldada organisatsioonilist kuuluvust näitavaid atribuute. Kui sertifikaat väljastatakse mõne organisatsiooni liikmele, peab eraldusnimi näitama tema organisatsioonilist kuuluvust.

> Sageli nõutakse eraldusnime puhul ka nimede subordinatsiooni. Kui organisatsioonisisene sertifitseerimiskeskus kannab nime C=EE, O=FIRMA, peab selle firma Tartu filiaalis asuv sertifitseerimiskeskus kandma nime C=EE, O=FIRMA, OU=TARTU FILIAAL ja kõik kasutajad, kellele jagatakse Tartus sertifikaate, kannavad nime C=EE, O=FIRMA, OU=TARTU FILIAAL, CN=isiku nimi. Tartu filiaal ei tohi jagada sertifikaate firma Rapla kontoris töötavatele inimestele, kuna nende nimedel peab olema kuju C=EE, O=FIRMA, OU=RAPLA FILIAAL, CN=isiku nimi. Nimede koostamisel eelkirjeldatud viisil on kerge tuvastada volituste ületamist sertifitseerimiskeskuses.

Kõik sellised eraldusnimedele esitatavad nõuded tuleb samuti sertifitseerimispoliitikas fikseerida.

### 13.4.2 X.509 sertifikaadi vorming




**Joonis 87. X.509 sertifikaadi vorming**

Versiooni number näitab, millisele standardi X.509 redaktsioonile see sertifikaat vastab. Uusim redaktsioon on 3, mis lubab lisada sertifikaadile laiendusi.

Järjenumber on ühe sertifitseerimiskeskuse piires ühene ning näitab, mitmes selle keskuse poolt väljastatud sertifikaat see on. Sertifitseerimiskeskuse nimi ning järjenumber määravad üheselt, millisele sertifikaadile viidatakse, seevastu sertifitseerimiskeskuse nimi koos sertifikaadi omaniku nimega ei määra sertifikaati üheselt (ühele inimesele võidi eri aegadel väljastada mitu sertifikaati).

Omaniku avalik võti koosneb kahest osast – algoritmi identifikaatorist ja tegelikust võtmest. See teeb standardiga X.509 kirjeldatava vormingu paindlikuks, võimaldades väljastada sertifikaate eri algoritmide avalike võtmete kohta.

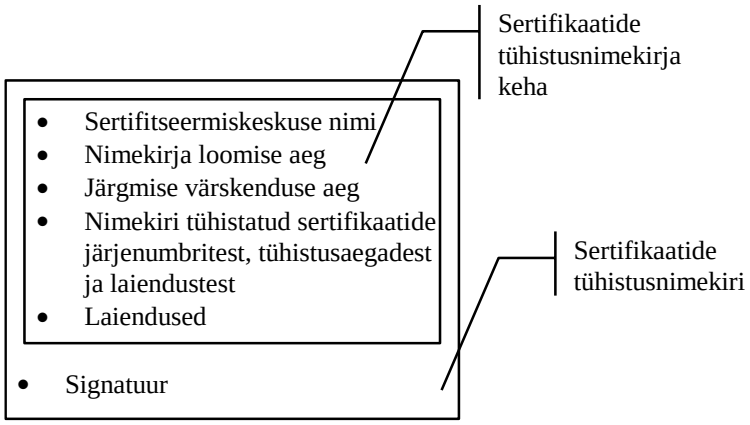
Kehtivusaeg koosneb algus- ja lõpuajast, mis määravad sertifikaadis sisalduva võtme kehtivusaja. Kehtivuse algusaeg on tegelikult sertifikaadis sisalduva avaliku võtme (mitte sertifikaadi enda) loomise aeg. Kõik signatuurid, mis on loodud sertifikaadis sisalduvale avalikule võtmele vastava salajase võtmega, kuid enne sertifikaadis kirjas olevat kehtivuse algusaega, on kehtetud. Kõik signatuurid, mis on loodud pärast kehtivuse lõpuaega, on samuti kehtetud. Kehtivuse lõpuaeg on sertifikaadi enda (mitte sertifikaadis sisalduva võtme abil kontrollitavate signatuuride) aegumistähtpäev. Enne aegumistähtpäeva kättejõudmist loodud signatuurid kehtivad endiselt. Sertifikaadi eluiga võib pikendada, andes välja uue samade avaliku võtme andmetega sertifikaadi, mille aegumistähtaeg ületab vana sertifikaadi oma. Kehtivusaegade range jälgimine ei kaitse signatuuri loomise kuupäevaga teostatavate manipulatsioonide eest, kuid aitab vältida aegunud sertifikaatide kasutamist.

Omaniku nimi sisaldab sertifikaadi omaniku eraldusnime. Väljaandja nimi sisaldab sertifitseerimiskeskuse eraldusnime.

> Signatuur on sertifitseerimiskeskuse salajase võtmega krüpteeritud sõnumilühend, mis moodustatakse DER-kujul kodeeritud sertifikaadi kehast. Sertifikaadis sisalduvaid andmeid kujutatakse kindlal, arvuti arhitektuurist sõltumatul viisil, mis tagab täpselt sama sõnumilühendi kõigi arvutitüüpide korral. Sertifikaadi autentsuse kontrolli käigus kodeeritakse sertifikaadi keha DER-kujul, leitakse ta sõnumilühend ning võrreldakse seda signatuuris sisaldunud sõnumilühendiga, mis saadakse signatuuri dešifreerimisel sertifitseerimiskeskuse avaliku võtme abil.

Kuna X.509 on ISO juurtega, on sertifikaadid kirjeldatud ASN.1-keeles ning neid säilitatakse ja edastatakse DER-kodeeritult.

### 13.4.3 X.509 tühistusnimekirja vorming




**Joonis 88. X.509 tühistusnimekirja vorming**

Iga sertifitseerimiskeskus hooldab enda väljaantud sertifikaatide tühistusnimekirja. Sertifikaadi tühistamise protseduur sarnaneb sertifikaadi väljastamise protseduuriga – sertifikaadi omanik peab esitama kirjaliku avalduse ning tõestama oma isikut või volitusi – täpsed nõuded fikseeritakse taas sertifitseerimispoliitikaga. Sertifitseerimiskeskus ei tohi tühistada sertifikaate omavoliliselt.

Sertifikaatide tühistusnimekiri sisaldab sertifitseerimiskeskuse eraldusnime, tühistusnimekirja järgmise väljaandmise aega ja tühistatud sertifikaatide järjenumbrite loendit ning on signeeritud sertifitseerimskeskuse avaliku võtmega. Järgmise värskenduse aeg võimaldab kasutajatel hõlpsalt veenduda, kas neil on kõige värskem tühistusnimekiri või on vaja pöörduda sertifitseerimiskeskuse poole uue nimekirja hankimiseks.

> Nagu eespool mainitud, identifitseerib paar (sertifitseerimiskeskuse nimi + väljaantud sertifikaadi järjenumber) üheselt sertifitseerimiskeskuse poolt väljaantud sertifikaadi. Seetõttu piisab, kui sertifikaatide tühistusnimekiri sisaldab vaid sertifikaatide järjenumbreid. Tühistusnimekiri signeeritakse sertifitseerimiskeskuse salajase võtmega, et vältida nimekirja võltsimist.

# 14 STEGANOGRAAFIA

## 14.1 Olemus

Steganograafia on informatsiooni peitmise tehnika. Ta nimetus tuleneb kreeka keelest ja tähendab sõnasõnalt "kaetud kirja", viidates Herodotose kirjeldatud antiiksetele peitmismoodustele (tekst vahatahvli vahakihi all, käskjala pealaele tätoveeritud tekst juuste all).

Steganograafiline meetod peidab *sõnumi* e steganogrammi (teksti, pildi, heli) mingi süstemaatilise ja tavaliselt ka pööratava meetodiga mingisse (füüsilisse või loogilisse) keskkonda – *konteinerisse*. Erinevalt krüptograafiast on steganograafia eesmärk varjata mitte niivõrd (või üldse mitte) salasõnumi sisu, vaid selle sõnumi olemasolu.

Ajalooliselt on steganograafia hõlmanud muuhulgas järgmisi peitmismeetodeid ja -vahendeid:

- salatindid (olid kasutusel veel teise maailmasõja ajal);
- tekstsõnumi täht- või sõnahaaval peitmine konteinerteksti (see ei ole krüptograafia!);
- kodeerimine paberi reljeefiga või mikrosälkudega paberiserval;
- mikrotäpp (Saksamaa 1941) – teksti hulka punktina või i-täpina kleebitav mikrofoto;
- salajane sidekanal;
- hajusspekterside (*spread-spectrum communication*).

Tänapäeval võib konteineriks olla prinditud, trükitud või failina salvestatud tekst, kahendfail, pildifail, helifail, vaba kettaala, filmi-/animatsiooni-/multimeediumifail, ringhäälingu- või televisioonisaade, heli- või videosalvestis kassetis või plaadil jne. Konteineri tüüp määrab *sõnumikandja* (sõnumi esituseks kasutatava elemendikogumi) tüübi.

Steganograafia on lähisuguluses krüptograafiaga. Ta ei asenda, vaid täiendab seda. Sageli kasutatakse nende tehnoloogiate kombinatsiooni, peites juba krüpteeritud sõnumi, võttes digitaalvesimärgi koostisse digitaalsignatuuri jne. Puutepunkte on ka meetodite aluseks olevas matemaatilises aparaadis.

David Kahn (*The Codebreakers*, 1967) kirjeldab steganograafia seost naabervaldkondadega järgmise tabeliga:

**Tabel 29. Kahni turbetabel**

| **Signaaliturve** | **Signaaliluure** |
|---|---|
| | |
| ***Suhtluse turve*** | ***Suhtluse luure*** |
| • *Steganograafia* (nähtamatud tindid, lahtised koodid, sõnumid kontsaõõnes) ja *edastuse turve* (spurtraadio- ja hajusspektersüsteemid) | • *Püük ja peilimine* |
| • *Krüptograafia* (šifrid ja krüptogrammid) | • *Krüptoanalüüs* |
| • *Liikluse turve* (kutsungite muutmine, pseudosõnumid, raadiovaikus) | • *Liikluse analüüs* (peilimine, sõnumivoo uurimine, radisti "käekirja" tuvastus) |
| | |
| ***Elektronturve*** | ***Elektronluure*** |
| • *Kiirguse turve* (radarisageduste nihutamine, hajusspekter) | • *Elektronvaatlus* (radarikiirguste avastamine) |
| • *Vastumeetmete vastumeetmed* (radarihäiringutest "läbivaatamine") | • *Vastumeetmed* (radarihäiringute tekitamine, radarikaja võltsimine) |

## 14.2 Rakendused

Lisaks turvatehnilistele on steganograafial ka muid kasutusalasid, eelkõige dokumentide otsingut või töötlust (nt heli- või videosalvestiste monteerimist) abistav nähtamatu märgistamine. Lühiülevaate rakendustest annab Tabel 30.

**Tabel 30. Steganograafia rakendusi**

| Sõnumi otstarve | Konteiner | | | |
|---|---|---|---|---|
| | **Tekst** | **Pilt** | **Heli** | **Video** |
| Konteinerdokumendi autoriõiguse või konfidentsiaalsuse kaitse | - | Omanikku või allikat tõendav vesimärk[1)] | | |
| | Dokumendi eksemplari identifitseeriv vesimärk | | | |
| Konteinerdokumendi tervikluse kaitse | Muudatuste tuvastust võimaldav vesimärk | | | |
| Dokumendi(osa) märgistus töötluseks või otsinguks | - | Aeg<br>Koht<br>Nimi | Ajamärgised<br>Esitaja<br>jms | Ajamärgised<br>Stseeni tähis<br>jms |
| Dokumendi annoteerimine | - | Nt patsiendi andmed, aeg jms röntgenfotol | Nt muusikapala nimetus, autor, esitaja, aeg, koht, reklaamikontekst | Nt kõik stseeniandmed; WWW puhul aktiivlingid |
| Salakommunikatsioon | - | Lelurakendused<br>Reklaam | Sõjalised, luure- jms teated | Alateadvusele suunatud reklaam |

1) Vt 14.3. Digitaalvesimärgiks nimetatakse sageli ka suvalist mitteturbelist steganomärgist või suvalist peitsõnumit üldse.

## 14.3 Digitaalvesimärk

Alates keskajast laialt kasutusel olevate paberi vesimärkide analoog on digitaalvesimärk – digitaaldokumendi sisse paigutatud turvaotstarbeline digitaalsignaal või -muster, mis võimaldab tõendada dokumendi omanikku ja/või identifitseerida dokumendi eksemplari või partiid.

Digitaalvesimärkide hulka ei kuulu *digitaalpitserid*, mida inglise keeles nimetatakse eksitavalt ja justkui biomeetriale viitavalt "digitaalsõrmejälgedeks" (*digital fingerprint*). Need on kaitstava andmeüksuse sisu alusel moodustatud metaüksused, nt kontrollsumma, tsükkelkood või krüptograafiline sõnumilühend.

***Nähtavad vesimärgid*** (Joonis 89) ei kuulu rangelt võttes steganograafia valdkonda, ehkki nad võivad mõnikord oma väiksuse või paigutuse tõttu olla üsna märkamatud. Peaaegu märkamatu on ka näiteks värvilahutuslik, ainult ühe põhivärvuse sisse paigutatud vesimärk, mis ilmub nähtavale alles värvilahutusprotsessis, hoiatades piraatkirjastamise katse eest. Nähtava vesimärgi tuntuim näide on ekraani nurgas asuv telekompanii embleem. Nähtavaid vesimärke saab lisada lihtsalt superponeerimisega.

***Nähtamatud vesimärgid*** on steganograafilised. Neid saab tekitada näiteks sümmeetriliste või asümmeetriliste räsifunktsioonidega. Sellisel juhul on vesimärgi kasutamisprotsessid (võtmehaldus jms) lähedased vastavatele krüptograafilistele protsessidele.

**Joonis 89. Nähtavate vesimärkide näiteid**

Nähtavatel vesimärkidel on eeskätt profülaktiline (hoiatav või konteinerdokumenti äriliselt kasutuskõlbmatuks muutev), nähtamatutel aga jälituslik funktsioon. Vesimärkide turvarakendustest annab ülevaate Tabel 31

**Tabel 31. Digitaalvesimärkide funktsioone (** – esmane, * – sekundaarne)**

| Otstarve | Nähtav | Nähtamatu |
|---|---|---|
| Saaja õigsuse kontroll | - | ** |
| Salgamise vääramisega edastus | - | ** |
| Varguskatsete peletamine | ** | ** |
| Dokumendi ärilise väärtuse kahandamine | - | * |
| Volitamata kopeerimise eest hoiatamine | ** | * |
| Digitaalne notariseerimine ja autentimine | * | ** |
| Allika identimine | ** | * |

Vesimärk saab tõhusalt täita oma otstarvet, kui ta vastab järgmistele tingimustele.

1. Märki peab olema võimatu või raske kõrvaldada, ilma et dokumendi kvaliteeti tunduvalt kahjustataks.
2. Vesimärk peab säilima dokumendi tüübile omaste tavaliste töötlusoperatsioonide rakendamisel (nt mastaabi muutmisel, kärpimisel, tihendamisel, pseudopooltoonimisel, heli- või videomiksimisel, analoogmuundusel jne).
3. Nähtamatut tüüpi vesimärk peab olema tajumatu ega tohi mõjutada vaatlus- või kuuldemuljet.

4. Mõnedes rakendustes peavad volitatud isikud saama nähtamatut vesimärki hõlpsalt ning dokumendi originaali kasutamata tuvastada.
5. Vesimärgiprotsesside hind peab olema vastuvõetav.

## 14.4 Meetodid

### 14.4.1 Üldpõhimõtted

Meeltega otseselt või vahendatult tajutavasse infokonteinerisse võimaldavad sõnumit peita inimese taju sageduskarakteristiku piirangud, eeskätt ülemise piirsageduse suhteline madalus tehniliste seadmete võimalustega võrreldes. Sageduse ülatõke ei võimalda inimesel tajuda ülipeeni geomeetrilisi detaile, mõõtmete või värvivarjundite erinevusi, helide kõrguse, tämbri või kestuse üliväikesi muutusi jms. Niisiis saab sõnumikandjana kasutada konteineri kõrgsagedusriba.

Teiseks leiab kasutamist asjaolu, et meeleline taju ei suuda eristada juhuslikke protsesse (müra, juhuslike moonutusi) neile sarnanevatest süstemaatilistest, struktureeritud protsessidest.

Niisuguste tajupettuse üldpõhimõtete realiseerimisviisid sõltuvad esmajoones konteineri tüübist, sõnumi tüübist, konteineri ja sõnumi mahtude vahekorrast, aga ka konkreetsest rakendusest (otstarbest) ja lisanõuetest (näiteks sõnumi veakindlusele).

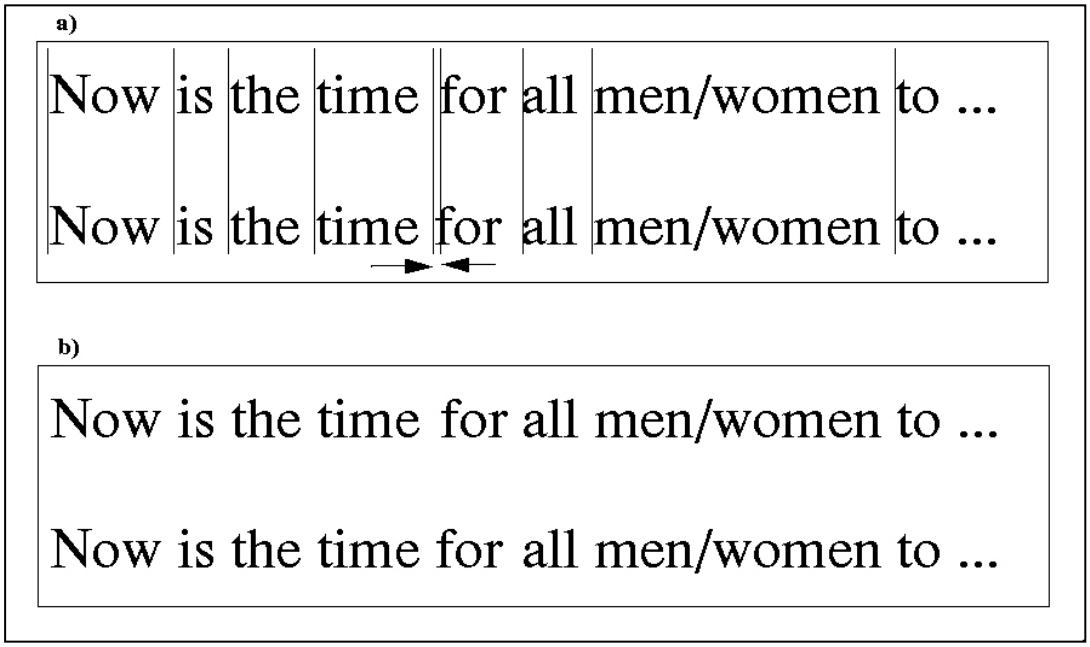
### 14.4.2 Sõnumikandjad

***Tekstkonteineri*** puhul on sõnumikandjatena kasutatud peamiselt järgmisi elemente.

1. Nähtamatud märgid tühjadel ridadel või reaosadel. Näiteks kodeerib programm *Snow* sõnumibitid tühikuplokkidena, mida eraldavad tabeldusmärgid ning paigutab 80 täheruumi pikkusele reale 30 bitti peitinformatsiooni.

2. Dokumendi küljendusgeomeetriliste mõõtmete muutmine pikseli võrra (paberil 0,01...0,3 mm võrra):
* reavahede laiuse varieerimine, sh teatud üksikridade püstnihe;
* sõnavahede (tühikute) pikkuse varieerimine; sh tekstiplokkide rõhtnihe;
* märgielementide (nt seriifide, sabade) pikkuse varieerimine.

Kodeerimist tühiku pikkuse varieerimisega illustreerib Joonis 90.



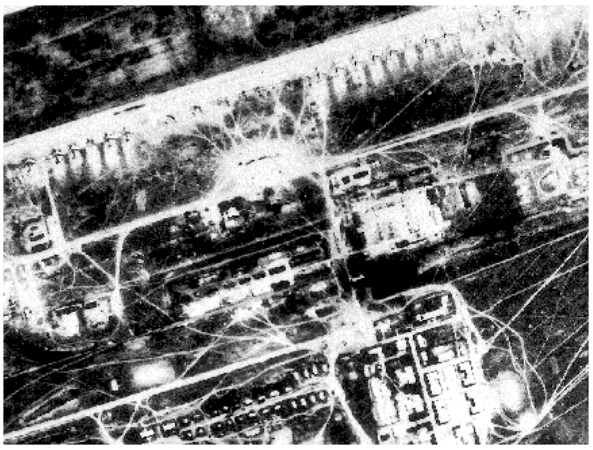

**Joonis 90. Kodeerimine sõnavahe pikkuse muutmisega**

Kasutatakse ka loetletud mooduste kombinatsioone, nt AT&T Bell Labs rakendab ridade püstnihet koos märgiplokkide rõhtnihkega. Selline vesimärk on üsna häire- ja töötluskindel ning tuvastatav veel kümnendal kserokoopial.

***Piltkonteiner*** pakub märksa suuremaid võimalusi nii sõnumi mahu (sõnum võib ka ise olla pilt) kui ka peitmismeetodi ja märkamatuse poolest. Arvestada tuleb inimese visuaaltaju iseärasusi. Ülalmainitud üldistele sageduspiirangutele (sh ka alumisele piirsagedusele, st tundetusele pikkade sujuvate üleminekute suhtes) lisandub näiteks kontrastitundlikkuse sagedussõltuvus. Ühtlastel hallskaalas pildialadel on toonimuutuse tajulävi umbes 0,4%, sujuva üleminekuga aladel aga tuvastab silm alles 3% muutuse. Vaatevälja eri osi tajub silm erinevalt; ääreefekt maskeerib üsna suuri heleduse ja värvuse muutusi.

24-bitise pikseliga värvipildis kirjeldab pikseli igat põhivärvust üks bait; selle baidi madalaima biti varieerumist inimsilm ei taju, niisiis saab igasse pikselisse peita kolm bitti. Teoreetiliselt saaks üsna tavalisse 1024×768 formaadis (satelliidifotode tavaline eraldusvõime) pilti peita umbes 300K baiti sõnumiteavet. Tegelikkuses pilte nii ebaökonoomsel kujul ei salvestata ega edastata, vaid kasutatakse mitmesuguseid kadudeta (GIF, BMP) või kadudega (JPG) pakkimise vorminguid. Steganograafiliseks otstarbeks sobivad paremini hõredama pakkimisega vormingud, näiteks GIF; tihedamaid vorminguid (JPG) enamik tooteid ei toeta, seetõttu tuleb need vajaduse korral konverteerida.

N. Johnsoni klassikalises artiklis on näide mustvalge foto (Joonis 91) peitmisest värvireprosse (Joonis 92). Renoiri repro algmaht JPG-kujul oli umbes 175K baiti, konverteerimisel 775K baidi suuruseks GIF-failiks pildi kvaliteet inimsilma jaoks ei muutunud. Seda konteinerit kasutati mitme stegoprogrammi võrdlemisel. Programm *White Noise Storm* muutis maalirepro märgatavalt sinakaks, *S-Tools* (vt tabel jaotises 14.4) seevastu andis tulemuse, mida silm ei erista "tühjast" konteinerist.



**Joonis 91. Sõnumpilt: strateegiliste pommitajate lennuväli Kasahstanis (mustvalge satelliidifoto, USA, 1966)**

**Joonis 92. Konteinerpilt: P.-A. Renoir, Le Moulin de la Galette (värvusskaneeritud repro)**

Ehkki pakkivad graafikavormingud jätavad sõnumi peitmiseks tunduvalt vähem ruumi võrreldes ülalmainitud teoreetilise maksimumiga, täiustuvad pidevalt ka peitmismeetodid. Üks arendussihte on sõnumite mahu tunduv suurendamine konteineri kvaliteedi väheolulise languse arvel. M.Sandfordi ja T.Handeli andmetel võimaldavad praegused sellelaadsed meetodid asendada sõnumiga kuni 73% standardse graafikafaili mahust.

***Helikonteiner*** püstitab üsna nõudliku ülesande. Inimkõrv tajub helivõimsust dünaamikadiapasoonis $1{:}10^9$ ning sagedusi vahemikus 1:1000. Ka tundlikkus aditiivse juhusliku müra suhtes on kõrge. Helifailis võib kõrv avastada 80 dB allpool ümbrustaset olevaid häiringuid. Kuulmistaju jätab siiski rea peidukohti: (1) üsna kitsa diferentsiaaldiapasooni tõttu kalduvad valjemad helid maskeerima vaikseid, (2) kõrv tajub ainult suhtelist, mitte absoluutset faasi, (3) kuulaja ignoreerib paljusid levinud keskkondlikke moonutusi.

Peidukoha maht sõltub digitaalheli puhul kvantimismeetodist ja (vähemalt lineaarselt) diskreetimissagedusest. Tavalises 16-bitise WAV-vorminguga helifailis ei erista inimkõrv madalaima biti varieerumist. Peidetava sõnumi võimalik maksimaalmaht sõltub veel ka sõnumi tüübist ja peitmismeetodist, kuid ta suurusjärku iseloomustab *Inteli* endise töötaja, füüsik T. May hinnang, mille kohaselt võib tavalisse digitaal-helikassetti lisaks muusikale salvestada sadu megabaite – keskmise kõvakettatäie – salateavet. Nagu piltide puhulgi, sobivad peitmiseks vähem kadudega pakkimisel põhinevad vormingud (MPEG-AUDIO), mis muudavad tunduvalt signaali statistikat ja säilitavad ainult kuulajale tajutavad karakteristikud.

Lisaprobleeme tekitavad veel kooderi ja dekooderi vahelises edastuskeskkonnas toimuvad muutused (amplituudi- ja faasimuutused, sageduskomponentide triiv, kajad jms). Vesimärgirakenduses peab vesimärk pidevalt korduma, nii et ta sattuks vähimassegi dokumendi fragmenti. Tegelikes toodetes on nt DICE saavutanud 2100 sõnumibiti (vähemalt kahe vesimärgieksemplari) rahuldava peitmise igasse helisekundisse.

***Filmkonteiner*** (üldjuhul: multimeediumkonteiner) on küll senivaadeldutest märksa suurema mahuga, kuid tegelikud rakendused nõuavad peitteksti paigutamist igasse kaadrisse. Seetõttu sarnaneb ülesanne ühelt poolt piltkonteineri juhuga, teisalt aga lisanduvad helikonteineri juhule analoogilised ning ka keskkonna- ja rakendusspetsiifilised probleemid.

### 14.4.3 Töökindlus

Peitmismeetodi valimisel tuleb alati teha kompromiss vastuoluliste nõuete – steganogrammi mahu, töökindluse ja konteinerdokumendi lõppkvaliteedi – vahel. Iga nõude osakaal sõltub rakendusest (vt Tabel 32). Turvarakendustes on esikohal töökindlus. Seda saab tõsta sobiva meetodi valimisega ja steganogrammi suhtelise mahu vähendamisega.

**Tabel 32. Steganograafia rakendusnõuded**

| Parameeter | Rakendus | | |
|---|---|---|---|
| | Vesimärk | Töötlusmärgistus Tervikluse kaitse | Annotatsioon |
| **Töökindlus:** | | | |
| Tahtliku kõrvaldamise suhtes | X | - | - |
| Mittegeomeetriline teisendus (tihendus jms) | X | | |
| Geomeetriline (afiinteisendus, kärpimine) | X | X | |
| Teisenduste puudumisel | X | X | X |
| **Steganogrammi maht** | Väike (bitid) | | Suur (kilobitid) |

***Teksti*** geomeetrilistel manipulatsioonidel põhinevate vesimärkide võimalik ründemoodus on teksti digitaalkuju korduv vähendamine ja suurendamine, mille käigus võivad pikseli suurused erinevused kaduda. Kõigist küljenduslikest vesimärkidest saab vabaneda teksti reskaneerimise või ümbertippimisega. Niisuguste rünnete tõrjeks on MIT pakkunud *semantilisi* (teataval sõnavalikusüsteemil põhinevaid) ja isegi *süntaktilisi* (teataval lausestuse süsteemil põhinevaid) vesimärke; need küsitava väärtusega vahendid ei kaitse aga näiteks piraatliku ümberjutustuse või tõlkimise eest.

***Pildi*** või ***heli*** puhul ei anna madalaimal bitil põhinevad meetodid piisavat kaitset steganogrammi kõrvaldamise eest (madalaima biti võib kõrvaldada nullimise või pseudojuhusliku müraga. Perspektiivsemad on laiemas ribas töötavad statistilised meetodid. Uuringutes (Adelson, Aura, Bender, Hecht, Lippman jt) ning arendustööde aruannetes on kirjeldatud üsna suur hulk meetodeid ning nende arv kasvab pidevalt. Tegelikes seni ilmunud toodetes kasutatud meetodid võib jämedalt jagada kahte rühma:

1) stohhastilise mustriplokiga kodeerimine (*patchwork*),
2) hajusspekterkodeerimine (*spread spectrum*), diskreetse koosinusteisendusega (DCT) või kiire Fourier' teisendusega.

Hajusspektermeetod võeti algselt kasutusele raadiosides nõrga signaali edastuseks häiringulises keskkonnas. Ta põhineb laia sagedusriba jaotamisel paljudeks kanaliteks ning pideval pseudojuhuslikul kanalivahetusel. Steganograafias rakendatuna lisab ta tunduvalt töökindlust, kuid kahandab dokumendi kvaliteeti.

Digitaalvesimärkide töökindlust tõstab steganograafia kombineerimine krüptograafiliste meetoditega, sh digitaalsignatuuriga.

## 14.5 Tooteid

Steganograafia kõige levinuma rakenduse, vesimärkide alal tegutseb praegu intensiivselt kümneid firmasid. Toodete rakendusvaldkonnana peetakse esmajoones silmas (satelliidi)televisiooni ning Internetti, eelkõige WWW-d. Ülevaate mõnedest iseloomulikest toodetest annab Tabel 33 (N. Johnsoni andmed).

*ARIS Technology* on keskendunud helidokumentidele, *Digital Information Commodities Exchange* (DICE) on üsna kaugele jõudnud multimeediumi alal ning kavandab oma tehnoloogia integreerimist agentprogrammidesse, WWW otsingumootoritesse jms vahenditesse. *Digimarc* on juhtivaid piltmaterjali vesimärgistajaid. Üks vesimärgi ulatuslikke rakendusi on Vatikani digitaalraamatukogu projekt IBM-ilt. Euroopas on käimas või lõpetatud rida multimeediumile suunatud koostööprojekte (ACCOPI, COPEARMS, TALISMAN, OCTALIS, OKAPI, IMPRIMATUR), mis lisaks tehnilistele käsitlevad ka elektroonilise autoriõiguste kaitse organisatsioonilist struktuuri ning võivad viia rahvusvahelise standardimiseni. Digitaalvesimärkide elujõudu tõendavad neil põhinevad edukad kohtupretsedendid USA-s.

**Tabel 33. Steganograafia- ja vesimärgiprogramme**

| Programm | Staatus, omanik/autor | Opsüsteem/ Keskkond | Lisanõuded | Meetod | Pildivorming | Märkusi |
|---|---|---|---|---|---|---|
| Argent | Äriline, Digital Information Commodities Exchange | | | | Pilte veel ei toeta | Lisab autoriõiguse- või litsentsiandmed vesimärgina |
| Copysight | Äriline, Intelle©tual Protocols2 (IP2) | WWW, Java-brauser | | | JPEG, GIF | Digitaalvesimärk ainult faasis 3. |
| EZStego, Stego Online, Stego | Jaosvara, Romana Machado | EZStego: Java, Stego Online: WWW, Stego: Mac | EzStego: Java Virtual Machine Stego Online: HTML 3.2 brauser | LSB, paleti nihkega. | EzStego: GIF, StegoOnline: GIF, Stego: PICT | |
| Hide and Seek | Jaosvara, Colin Maroney (UK) | Win, DOS | | LSB | V5.0: GIF, Win95: 8-bit BMP | |
| IBM Digital Library System | Äriline, Howard Sachar | | IBM Digital Library System | Diskreetne koosinusteisendus (DCT) | | Vesimärk on teegisüsteemi sisefunktsioon |
| JK_PGS (Pretty Good Signature) | Prooviajaga, EPFL (Šveits) | UNIX (Sun, SGI, LINUX); Win95/NT-versioon väljatöötamisel | Ainult pildivorming PPM | | | Photo Shopi ja Paint Shopi lisandid väljatöötamisel |
| Jsteg | Priivara, Derek Upham | UNIX, DOS, Win (saadaval lähtekood) | | Kombinatsioon JPEG-tihendusega? | Sisend: PPM (PBMPLUS värv), PGM (PBMPLUS hall), GIF, Targa, RLE (Utah Raster Toolkit). Väljund: JFIF | Peitpildi ekstraheerimisel tuleb luua "dekodeeritud" pilt |
| Mandelsteg | Priivara, Henry Hastur | C lähtekood | | | Genereerib Mandelbrot-fraktaalide GIF-pildi | Ebaturvaline Signatuur: 128 värvi. Iga värv kasutab kaht paleti elementi |
| PictureMarc, BatchMarc | Äriline, Digimarc Corporation | | Photoshop-ühilduv pilditöötlustarkvara<br><br>Vesimärgikoodi kontrolliks MarcCenter | Mustriploki kodeerimine | Märgi saab panna igasse Adobe Photo Shopiga loetavasse pilti ning töödelda filtritega (24-bitine või hallskaala) | Originaaliga võrreldes on "lamedatel" aladel väikesi muutusi |
| PixelTag | MIT Media Lab, Joshua Smith ja Barrett Comiskey | | | Mustriploki kodeerimine | | Patentimisel |
| Steganos | Jaosvara (edaspidi äriline), Deus Ex Machina Communications | Win, DOS | | LSB | V1.4 (DOS): BMP; Win95: BMP, DIB | 8-bitistel piltidel märgatav müra<br>V1.4 peidab andmeid ka VOC-, WAV- ja ASCII-failidesse.<br>Win95 töötleb ka VOC-, WAV-, TXT-, HTML-faile |
| Stegodos, Black Wolf's Picture Encoder | Avalik, Black Wolf | DOS | Kodeeritud pildi salvestuseks: mujalt pärit kuvahõivetarkvara | LSB. Kodeeritud pilt kuvatakse ja tuleb salvestuseks hõivata joonistus-programmiga | Mitu, sest kasutab kuvahõivet. Toetab ainult 256 värviga 320x200 pilte. | Enamikul piltidest märgatav müra |

| | | | | | | |
|---|---|---|---|---|---|---|
| S-Tools | Jaosvara,<br>Andy Brown (UK) | Win | | LSB.<br>Värvikahandus järgneva kvaliteedilangust vältiva maskeerivate värvide lisamisega | V4.0: GIF, BMP | Toetab ka WAV-faile.<br>Raskusi on lühikeste sõnumite (1-2 baiti) taastamisel |
| SureSign | Äriline,<br>Signum Technologies (UK) | Win, MAC | Photoshop-ühilduv pilditöötlustarkvara.<br>Töötab jaosvaraga Paint Shop Pro | Mustriploki kodeerimine | Suvaline, mida kuvab Paint Shop Pro, Photo Shop või nendega ühilduv ja mida saab töödelda pildifiltritega. | Vesimärk järgib pildi mustrit |
| SysCoP | Äriline,<br>Fraunhofer Center for Research in Computer Graphics (CRCG), Jian Zhao | Sun Solaris, HP-UX, SGI IRIX, Win95/NT;<br>Mac-versioon ja Netscape'i moodul väljatöötamisel | | | PPM, PGM, PBM; teisenduskomplektiga ka JPEG, GIF, TIFF. | Filmi puhul toetab vorminguid Motion JPEG and MPEG-1<br>8-bitistel piltidel koosneb signatuurimuster lähedastest värvidest |
| TigerMark | Äriline,<br>NEC | | Tuumaks vajab Informixi andmebaasi-süsteemi | Diskreetne koosinusteisendus (DCT) või kiire Fourier' teisendus (FFT) turvalise hajusspektriga | | Vesimärgiprogrammi on integreeritud Informixi Datablade-meetod<br>Arenduskomplekt SDK annab vesimärgi võimaluse ilma Informixi meetodita |
| Visual Crypto-graphy | Priivara,<br>Jouko Holopainen | DOS | | Genereerib kahe esimese sisendpildi PostScript-failid.<br>Peitpilt hajutatakse kahele kandjale; nende ühitamisel on peitkujutis nähtav | Konteinerpildid moonutuvad tugevalt | |
| White Noise Storm | Jaosvara,<br>Ray Arachelian | DOS | | Hajusspekter, LSB | PCX | Moonutab 8-bitiseid pilte, võib tekitada täielikke värvinihkeid 24-bitistel |

# 15 SEIRE

# 15.1 TURVAREVISJON

### 15.1.1 Turvarevisjoni otstarve ja olemus

Turvarevisjon (*security audit*) on seiremeede, mida kasutatakse arvutisüsteemi rünnete avastamiseks reaalajas või järelanalüüsiga ning ründe allika ja sooritusviisi väljaselgitamiseks ja tõendamiseks. Ühtlasi on turvarevisjonil ka peletusfunktsioon; teadmine ta käigushoiust distsiplineerib süsteemi kasutajaid ning sunnib potentsiaalseid ründajaid või väärkasutajaid arvestama sanktsioonide suurema tõenäosusega.

Turvarevisjoni võimaldamiseks peab vastav alamsüsteem (operatsioonisüsteem, andmebaasihaldur, marsruuter, kriitiline rakendus vms) sisaldama mehhanismi, mis reaalajas registreeriks turvapäevikus (*audit trail*), näiteks süsteemilogis, turvalisusega seotud sündmusi; kuna neid on raske ette määratleda, logitakse tavaliselt peaaegu kõiki vähegi olulisemaid sündmusi.

Turbekriteeriumid TCSEC nõuavad revisjonimehhanismilt, et see

1) võimaldaks saada ülevaadet üksikobjektide poole pöördumise mustritest, konkreetsete protsesside ja isikute pöörduste kronoloogiast ning mitmesuguste kaitsemehhanismide kasutamisest ja tõhususest;
2) võimaldaks avastada kasutajate või kõrvaliste korduvaid turvamehhanismidest möödumise katseid;
3) võimaldaks avastada lubatust suuremate privileegide kasutamist;
4) peletaks turvamehhanismidest möödumise katseid;
5) suurendaks kasutaja usku turvamehhanismidest möödumise avastamisvõimalusse.

Turvapäevikus sisalduv informatsioon peaks olema piisav turvalisust puudutavate protsesside ja sündmuste täielikuks rekonstrueerimiseks.

Enamikus kasutuselolevates päevikutes registreeritakse järgmised andmed:

- sisselogimise kuupäev ja kellaaeg,
- kasutatud tööjaam või terminal,
- töö identifikaator,
- programmi v protsessi identufikaator,
- faili identifikaator,
- operatsioonid (lugemine, kirjutus, kustutus jne),
- printväljastuse identifikaator ja maht (lehekülgede arv).

Peale selle registreeritakse süsteemi tõrkeid ning väljastatakse päevikust regulaarselt tõrkearuandeid. Lisaks stiihilistele häiringutele võimaldab päeviku see osa tuvastada ka ründeid, ründekatseid ja kasutaja kvalifikatsiooni lünki. Tõrkeregistri osa peaks sisaldama alljärgnevaid andmeid.

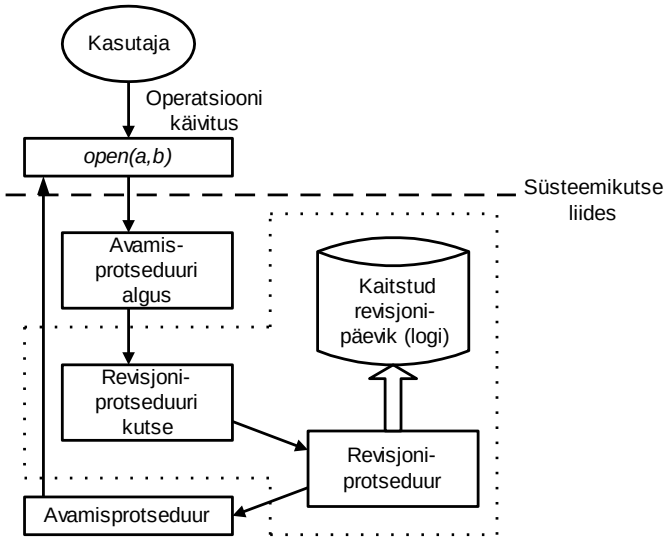
1. Kõiki riist- või tarkvaramehhanismide tõrkeid.
2. Tööde ebanormaalseid lõppe ja süsteemi krahhe.
3. Ebaloomulikult lühikesi vaheaegu buutimiste vahel. Muuhulgas võivad need näidata, et kasutaja viib süsteemi tahtlikult krahhini oma volitamatu tegevuse varjamiseks või püüdes otsida uusi salauksi.
4. Ebaõnnestunud sisselogimiskatseid. See võib tähendada parooli mõistatamise üritamist. Liiga sage edukas sisselogimine võib olla märk parooli teatavakssaamisest ja volitamatust kasutamisest.
5. Katseid saada volitamatut juurdepääsu failidele, eriti aga tundlikele või väärtuslikele failidele ning selliste failide kasutamise äkilist seletamatut kasvu.
6. Katseid kasutada õigusi väljaspool oma volitusi.

Revisjonipäevikus registreeritakse kõigi süsteemiga tegelenud isikute toimingud, kaasa arvatud haldus- ja hooldepersonal.

Logitavate andmete koostis sõltub veel süsteemi turvaklassist (vt standardnõuete näide Tabel 34) ja kasutatavast revisjoniandmete automaatse interpreteerimise meetodist (vt 15.1.3).

### 15.1.2 Revisjoniandmete kogumine

Revisjoniandmete automaatse kogumise põhimõtet selgitab Joonis 93. Revisjonilogi jaoks valitud sündmustele vastavat operatsiooni (näiteks faili avamist) realiseeriv juhtprogramm (näiteks operatsioonisüsteem) algatab vastava protseduuri, pärast pöördusandmete läbivaatust katkestab selle, kutsub ja täidab revisjoniandmete kirjutuse protseduuri ning jätkab pöördusprotseduuri täitmist (kui seda ei tõkesta võimalik pääsu reguleerimise mehhanism).




**Joonis 93. Revisjoniandmete logimine**

Revisjoniandmed logitakse kindla vorminguga revisjonikirjetena. Vorminguid püütakse standardida (nt IEEE POSIX). Suures kohtvõrgus võidakse revisjonipäeviku pidamiseks vajaduse korral sisse seada spetsiaalne päevikuserver. Kuna revisjoniandmed moodustavad turvamehhanismi osa, on nad tundlikud ja peavad olema kaitstud volitamatu juurdepääsu ja muutmise või kustutuse eest. Juurdepääs päevikule peab olema rangelt piiratud, tavaliselt on see süsteemihalduril ja turvahalduril. Revisjonipäevikus ei tohi registreerida autentimisteavet, näiteks paroole. Revisjonimehhanismi peamised ohud on ta volitatud kasutajate volitamatud toimingud ning revisjoniandmete hävimine süsteemi tõrke tõttu.

Päeviku juurde võivad kuuluda eraldi vahendid ta sirvimiseks ja analüüsiks; näiteks peavad need võimaldama kiiresti kindlaks teha, kas teatav kasutaja töötab süsteemiga regulaarselt väljaspool normaalset tööaega või kas ta prindib ebaloomulikult palju. Revisjonipäevikut tuleb regulaarselt kontrollida ning ta sisu tuleb arhiveerida väga pikaks ajaks, kõrge salastusastmega süsteemides võib säilitusaeg ulatuda kümne aastani.

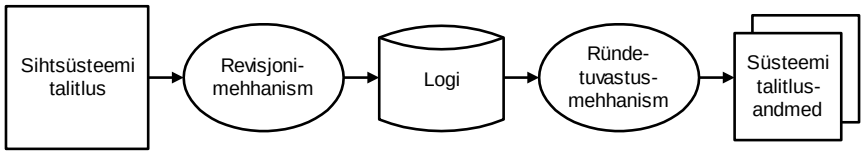
Revisjonivahendid ei tohi märgatavalt mõjutada süsteemi jõudlust ega reaktsiooniaegu. Päeviku ruumitarbe vähendamiseks võidakse andmed pakkida.

### 15.1.3 Rünnete tuvastus

Süsteemilogid ja muud revisjonipäevikud võeti laiemalt kasutusele 1980te algul, mil nii arvuteid kui ka kasutajaid oli suhteliselt vähe. Tollal näis logide regulaarne läbivaatus süsteemiülema teostatuna olevat sissetungide avastamiseks piisav. Praktika näitas juba tollal vastupidist ning juba paar aastat hiljem hakkasid ilmuma programmid, millega püüti logiandmeid automaatselt kokku suruda või analüüsida.

Ernst & Youngi iga-aastane turvauuring näitas, et sissetungi arvutisüsteemidesse nentis 1996. a tõsise probleemina 16% vastanuist, 1997. a aga juba 46%. USA kaitseministeeriumi sissetungitestid aastail 1996–1998 näitasid, et ainult 4% süsteemidest suutsid kohe avastada sissemurde ja ainult 1% reageeris sellele mingil viisil.

Niisiis ei ole traditsiooniline revisjoniandmete visuaalne läbivaatus arvutite massilise kasutamise ning võrgukeskkonna tingimustes tulemuslik. Kuna revisjonipäevik sisaldab hulgaliselt normaalse kasutuskäitumise andmeid, mille seast on raske pelga vaatlusega leida anomaaliaid, tuleb leida võimalusi vähemalt logiandmete automaatseks kontsentreerimiseks, rünnete automaatseks tuvastamiseks, ideaaljuhul aga rünnete ennetavaks tuvastuseks ja automaatseks blokeerimiseks.




**Joonis 94. Revisjoniandmete kontsentreerimine**

Kontsentreerimis- ja tuvastusmehhanismid on praegustes toodetes teostatud eraldi tarkvaramoodulitena, mis võtavad andmeid revisjonipäevikust (Joonis 94). Rida kasutatavaid tuvastusmeetodeid võib aga opereerida ka muude sisendandmetega, näiteks võrgupakettidega.

#### 15.1.3.1 Statistilised tuvastusmeetodid

Statistilised meetodid on rünnete (pool)automaatsel tuvastamisel levinuimad, sest neid on kõige lihtsam realiseerida ja nad nõuavad vähe ressursse. Meetodite olemus seisneb subjekti käitumise teatud mõõtude väärtuste aegjadade statistilises analüüsis.

Enamasti kasutatakse alljärgnevat tüüpi mõõte:

1) tegevuse intensiivsuse mõõdud (näiteks mingi subjekti kohta genereeritud revisjonikirjete arv minutis);
2) tegevuse tüübijaotuse mõõdud (nt failipöörduste, sisend-väljundoperatsioonide jne suhtelised osad kogu süsteemikasutusest);
3) kategooriajaotuse mõõdud (nt iga füüsilise töökoha sisselogimiste suhteline osa, iga meileri, kompilaatori jne kasutamise suhteline osa);
4) töötlusressursside mõõdud (nt teatud kasutajale langev osa protsessoriajast).

Mõõdud $M_1$, $M_2$,...,$M_n$ moodustavad kasutusprofiili. Mõõtude hetkväärtustest $S_1$, $S_2$,...,$S_n$ moodustatakse revisjonifunktsioon $R$, millel võib olla näiteks järgmine kuju:

$$R = a_1S_1^2+a_2S_2^2+...+a_nS_n^2, \text{ kus } a_i>0.$$

Profiiilide hetkväärtuste põhjal arvutatakse jooksvalt statistikud, niisiis on mälutarve väike. Statistikutega saab hõlpsalt avastada näiteks järgmisi anomaaliaid:

1) keskväärtusest tugevalt (nt rohkem kui standardhälbe võrra) lahknevad väärtused;
2) kahe revisjonifunktsiooni absoluutvahe integraal ajas võib näidata kahe eeldatavalt ühesuguse profiili tegelikku anomaalset lahknevust.

Statistilistel meetoditel on siiski mitu olulist puudust:

1) iga statistiline anomaalia ei tähenda veel rünnet, iga rünne aga ei tarvitse avalduda statistilise anomaaliana (st kasutusprotsesside vaikimisi kvaasistatsionaarsuse eeldamine ei ole alati õigustatud);
2) meetodid ei arvesta sündmuste ajalist järgnevust, tihti aga määrab just see toimingute lubamatuse;

3) käitumist aeglaselt muutes saab kasutaja "harjutada" revisjonimehhanismi, nii et see hakkab anomaalset käitumist pidama normaalseks;
4) raske on määrata õigeid lävivääртusi, mille alusel lugeda anomaaliat ründeliseks.

### 15.1.3.2 Ründemustril põhinevad meetodid

Ründelise käitumise saab paljudel juhtudel ette määratleda nn ründekäekirjadena (*intrusive signature*), mis spetsifitseeritakse sissemurdeni viivate sündmuste ja tingimuste kogumina. Tingimused määravad konteksti, milles sündmusejada lõpeb sissemurdega. Protsess sarnaneb viirusetõrjeprogrammides rakendatava viiruste käekirjade otsinguga failidest. Mustrid ei tohi olla vastuolus ning peavad olema piisavalt üldised, et tabada sama põhiründe variatsioone.

Meetodi teostuseks on kasutatud peamiselt ***ekspertsüsteeme***. Turbespetsialistide teadmus teisendatakse järeldusreeglistikuks, mille alusel langetatakse revisjoniotsused. Selliste süsteemide peamised puudused on järgmised:

1) ekspertsüsteemis sisalduv teadmus sõltub turbespetsialisti teadmusest ega tarvitse olla piisav;
2) ei arvestata sündmuste ajalist järgnevust;
3) avastada saab ainult senituntud ründeid;
4) vajalik andmete maht on suur, teadmusbaasi hooldus tekitab tarkvaratehnilisi probleeme.

Ründemustrite avastamiseks on viimastel aastatel rakendatud ka ***kujutuvastuse*** meetodeid; vastavaid tegelikke teostusi ei ole veel teada.

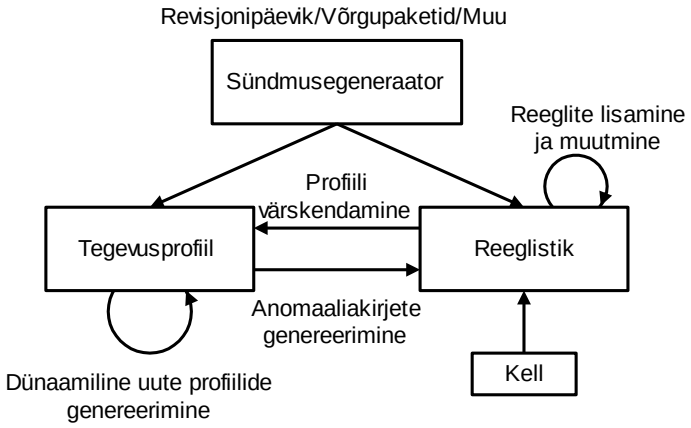
Sellesse meetodite rühma kuulub ka rünnete tuvastus ründelise käitumise ***modelleerimisega***. Ründestsenaariumid kavandatakse kasutaja käitumisjadadena, mis seejärel teisendatakse revisjonisündmuste jadadeks. See moodus vähendab logitavate revisjonisündmuste arvu (logida tuleb ainult need sündmused, mis esinevad vähemalt ühes stsenaariumis). Stsenaariume lisada, muuta või kaaludega varustada on suhteliselt lihtne.

***Olekusiirete analüüs*** loob tuntud sissemurrete olekusiirdemudeli. Ründaja sooritab rea toiminguid, mille käigus sihtsüsteem siirdub mingist algolekust vaheolekute kaudu turvarikkeolekusse. Mudel spetsifitseerib olekumuutujad ja ründaja toimingud ning määratleb turvarikkeoleku tähenduse. Revisjoniandmetest moodustatud mustrit võrreldakse tõenäosuslikult mudeli olekujadaga.

Omadustelt kirjeldatutega suhteliselt sarnased on ka mitmesugused ***käitumisprognoosidele*** rajatud meetodid, sealhulgas neurovõrkudel põhinevad. Kasutaja järgmist sammu võrreldakse prognoosituga; tugev lahknemine tähendab anomaaliat. Neurovõrkmeetodid on arvutusmahukuselt ökonoomsed, kuid selliste süsteemide õpetamise faas on töö- ja oskusmahukas.

### 15.1.3.3 Tuvastusmudel IDES

Automaattuvastuse üldise raamstruktuurina töötati 1986. a (D. Denning, SRI International) välja mudel IDES (*Intrusion-Detection Expert System*, "ründetuvastuse ekspertsüsteem") ning seda realiseeriv prototüüpsüsteem. See oli oluline samm reaalajas sooritatava duaalanalüüsi (anomaaliate ja ründemustrite) arendamisel. 1990il arendati süsteemi edasi ning ta viimased versioonid said nimeks NIDES (*Next-Generation IDES*, "järgmise põlvkonna IDES"). Mudel ei tee kitsendusi kasutatavaile tuvastusmeetoditele ning rakendab korraga nii statistilist kui ka ründemustri analüüsi.




**Joonis 95. Ründetuvastuse üldmudel IDES**

Mudeli IDES põhikomponendid on järgmised (vt Joonis 95).

***Subjektid ja objektid***. Mudeli teostuse esimese sammuna tuleb määratleda subjektide hulk ja objektide hulk. Subjektid on logitavate operatsioonide aktiivsed algatajad, tavaliselt arvutisüsteemi protsessid, mida juhib operatsioonisüsteem. Objektid on infohoidlad, millega opereerivad subjektid; tavaliselt on need failid ja failikataloogid. Mitmes turvalises Unixil põhinevas süsteemis (nt UNIX System V/MLS) lisanduvad neile objektidele veel muud struktuurid (i-node, protsessidevaheline suhtlus ipc, signaalistruktuurid), üldjuhul kõik olemid peale protsesside.

***Revisjonikirjed***. Revisjonikirje vorming koosneb kuuest püsiväljast: *subjekt*, *objekt*, *toiming*, *viga*, *ressurss*, *kellaaeg*. *Subjekt* on *objektile* suunatud *toimingu* algataja. *Viga* kirjeldab eranditingimusi, mis võisid *toimingu* tulemusena täituda. *Resurss* esitab *toimingu* ressursikasutuse statistikat.

Näide. Kui kasutaja Robert käitas kell 14.00 edukalt faili minufail ning kasutas selleks 2 sekundit protsessoriaega, näeb vastav revisjonikirje välja nii: *Robert, minufail, execute, ei, CPU(00:02), 14:00.*

Vajaduse korral võidakse seda vormingut muuta või täiendada uute väljadega. Näiteks UNIX System V/MLS võimaldab kirjes kasutada mitusada baiti pakitud andmeid.

***Profiilid***. Profiil on kasutaja normaalset süsteemikasutust kirjeldav parameetrikogum. Profiili komponendid võivad kirjeldada näiteks järgmist:

- sisselogimised: arv päevas, esimese sisselogimise aeg, seansi maksimaalne kestus jne;
- ressursikasutus: protsessoriaeg, mälu jm;
- failipöördused: teatud failide kirjutuse ja lugemise sagedus, ebaõnnestunud pöördumised jne.

Sellised andmed võimaldavad eristada potentsiaalseid turvarikkeid: näiteks sisselogimisaeg 4.00, kümme minutit protsessoriaega nõudva rakenduse käivitus tavalisel bürootööl, katse kopeerida paroolifaili on kahtlase käitumise ilmingud. Hälbeid ettedefineeritud profiilidest saab tuvastada automaatselt. Töökäitumise statistilised uuringud on näidanud, et väljakujunenud töö korral on käitumisprofiilid üsna stabiilsed. Raske on luua (tüübilt) uue kasutaja profiili; see nõuab kestvamat jälgimist ja korrigeerimist.

***Anomaaliakirjed***. Need on alarmiteated, mis genereeritakse siis, kui logitud käitumine ei vasta profiilile. Nad sisaldavad probleemi tuvastuseks vajalikke andmeid. Süsteemis IDES koosnevad nad kolmest väljast: *sündmus*, *aeg*, *profiil*. Selles kirjes identifitseerib *sündmus* süsteemi tegevuse, mis vallandas alarmi (näiteks korduv ebaünnestunud sisselogimiskatse). *Profiil* osutab seda profiili, mille puhul ilmnes käitumishälve. Need kirjed genereeritakse kaht tüüpi kahtlase käitumise korral: (1) sellise, mis on kahtlane süsteemi suvalise kasutaja puhul, (2) sellise, mis on kahtlane konkreetse kasutusprofiili puhul. Esimesena nimetatud juhtude tuvastuseks luuakse üldprofiil.

***Toimingureeglid*** kirjeldavad meetmeid, mida tuleb rakendada mingi anomaalia korral. Võimalike reaktsioonide näited on helisignaal, terminali ekraani vilkumine, asjakohase töötaja telefoni helisemine, meili saatmine süsteemiõlemale jne. Toimingureeglid kujutavad endast tavaliselt tingimusavaldisi kujul

if *alarm(1)* then *toiming(1)*
...
if *alarm(n)* then *toiming(n)*

Sellise reeglistiku teostuseks on väljatöödetes kasutatud reeglipõhiseid programmeerimiskeeli.

Mudeli IDES üks teostusnäiteid on revisjoniinstrument ComputerWatch (AT&T Bell Laboratories).

**Tabel 34. TCSEC revisjoninõuete kokkuvõte**

| TCSEC klass | Logitavad sündmused | Logitav informatsioon | Revisjonialus |
|---|---|---|---|
| C2 | Identimis- ja autentimismehhanismide kasutamine<br>Objektide lisamine kasutaja aadressiruumi<br>Objektide kustutus kasutaja aadressiruumist<br>Operaatorite, süsteemi- ja turvaülemate toimingud<br>Kõik turvalisust puudutavad eraldi määratletud sündmused<br>Printväljastus | Sündmuse kuupäev ja kellaaeg<br>Sündmuse genereerinud subjektiga esindatud isiku ühene identifikaator<br>Sündmuse tüüp<br>Sündmuse õnnestumine/ebaõnnestumine<br>Identimis-/autentimissündmuse taotluse allikas<br>Kasutaja aadressiruumi lisatud, seal kasutatud või sealt kustutatud objekti nimi<br>Turbeandmebaasides süsteemiülema poolt tehtud muudatuste kirjeldus | Süsteemiülem peab saama revideerida individuaalsete identiteetide alusel ja objektide identiteetide alusel |
| B1: + | Kõik paberväljastusel loetavate väljastusmärgiste muutmised (sh tundlikkusmärgiste ülekirjutused ja märgistuse väljalülitamine)<br>Sidekanali või välisseadme tasemelisuse (ühelt mitmele või vastupidi) muutmine<br>Ühetasemelise sidekanali või välisseadme tundlikkustaseme muutmine<br>Mitmetasemelise sidekanali või välisseadme ulatusmäärangu muutmine | Objekti turvatase<br>Subjekti tundlikkustase | Süsteemiülem peab saama revideerida individuaalse identiteedi ja/või objekti turvataseme alusel |
| B2: + | Sündmused, mis võivad opereerida salvestuse salakanalitega | - | Süsteem peab revideerima salvestuse salakanaleid, mis ületavad 10 bit/s ning peaks võimaldama revideerida salasalvestuse mehhanisme, mis võivad ületada 0,1 bit/s |
| B3, A1: + | Sündmused, mis võivad osutada otsest süsteemi turvapoliitika rikkumist (näiteks opereerida ajastuse salakanalitega) | - | Süsteem peab sisaldama mehhanismi, mis suudaks seirata sellist logitavate sündmuste kuhjumist, mis osutab osesele turvapoliitika rikkumisele, teavitades sellest kohe süsteemiülemat ning kuhjumise jätkumisel sooritama sündmust lõpetava toimingu* |

* Toiming võib seisneda sündmust põhjustanud terminali või kasutaja blokeerimises. Katkestava toimingu iseloom sõltub konkreetsest rakendusest, süsteemi peatamine tuleb jätta viimaseks võimaluseks.

## 15.2 Infrastruktuuri seire

Tavaliselt liigitatakse infrastruktuuri seire tehnilised vahendid füüsilise turbe meetmeteks. Komplekssе turbelahenduse seisukohalt on aga otstarbekas silmas pidada nende üha selgemalt väljenduvat infotehnilist olemust ja üha suuremaid arvutisüsteemidega integreerimise võimalusi ja vajadusi.

### 15.2.1 Valvesignalisatsioon

Valvesignalisatsiooni süsteeme kasutatakse objektide (ruumide, hoonete ja muude rajatiste, välisterritooriumi, sõidukite) füüsiliste rünnete ja protseduurihälvete tuvastuseks reaalajas, kuid neil võib olla ka ennetav, peletav ja tõendav toime. Sisendseadmestiku moodustavad peamiselt sissetungiandurid, kuid sinna võivad kuuluda ka häirelülitid, kontrollkellad, elektrilised lukud jms.

#### *15.2.1.1 Kontaktandurid*

Need on lihtsaimad ja odavaimad vahendid ehituspiiretest läbitungimise avastamiseks. Toimivad lihtsa lülitina, mis sulgeb või katkestab kontrollisüsteemi vooluahela.

***Lint-kontaktandur*** on alumiiniumfooliumiga kaetud kleeplint või -leht, mis ühendatakse vooluahelasse ning purunemisel katkestab ahela. Kasutatakse akna, ukse, seina vms kaitseks.

***Herkonanduri*** põhielement on herkon, magnetmaterjalist kontaktipaar hermeetilises klaasampullis, sulgub või lahutub magneti lähendamisel. Kasutatakse akende, uste jms kaitseks, tavaliselt paigaldatakse kahekaupa.

#### *15.2.1.2 Vibratsiooniandurid*

Muundavad mehaanilised tõuked või vibratsiooni elektriliseks signaaliks. Põhinevad piesoefektil või elektromagnetilisel induktsioonil; viimase tüüpiline teostus sisaldab kaks püsimagnetit ja mähisepooli. On odavad, kuid tekitavad rohkesti vääralarme. Kasutatakse massiliselt sõidukite kaitseks.

#### *15.2.1.3 Ultraheliandurid*

***Passiivne ultraheliandur*** muundab ultrahelisageduslikud mehaanilised võnked elektriliseks signaaliks. Kasutatakse klaaspindade kaitseks. Täiuslikumad mudelid analüüsivad helispektrit ning eristavad lööke klaasile ja klaasi purunemist; kui löögihelile järgneb umbes 150 ms pärast purunemisheli, väljastab andur alarmisignaali. Ühe sellise anduriga saab kontrollida kuni 10 $m^2$ suurust klaaspinda.

***Aktiivne ultraheliandur*** on ultrahelikiirgurist ja -vastuvõtjast koosnev lokaator. On odav, kuid kõrge vääralarmide sagedusega. Toimet mõjutavad temperatuur, niiskus, tõmbetuul, akustiline müra. Kasutatakse piiratud objektidel, näiteks auto salongi kontrolliks.

#### *15.2.1.4 Infrapunaandurid*

***Passiivne infrapunaandur*** reageerib soojust kiirgavatele objektidele (sealhulgas inimesele) ning muundab infrapunakiirguse elektriliseks signaaliks. Vaateväli on harilikult 90-110ț või reguleeritav (näiteks Fresneli läätsede vahetamise teel). Valehäireid võivad tekitada küttekehade sisselülitamine,

tõmbetuul, pindade soojenemine päikesekiirguses. Keerukamad teostused sisaldavad signaalitöötlust, sh mikroprotsessoriga ning suudavad valehäirete sagedust vähendada.

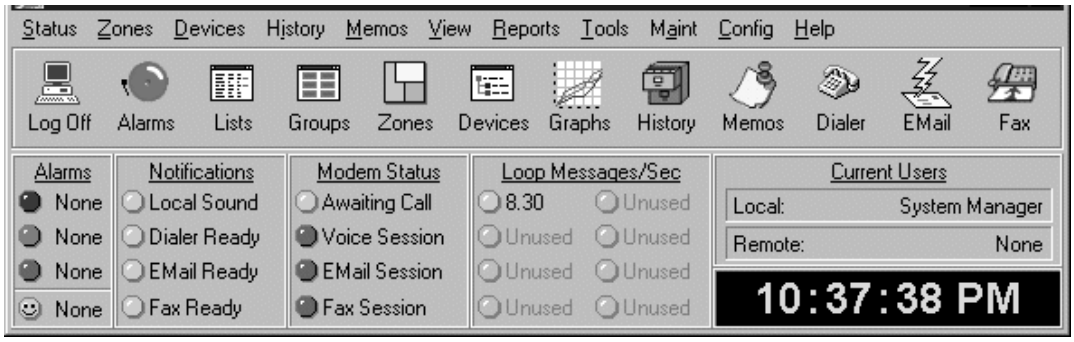
***Aktiivne infrapunaandur*** on ühest või mitmest kiirguri (lainepikkusega umbes 1 µm) ja vastuvõtja paarist koosnev seade, peamiselt barjäärandurite kujul. IP-barjääridega kontrollitakse sissetungi taradel, kangialustes, ustel, liftides, koridorides jne. Andur võimaldab eristada liikumiskiirust barjääri läbimisel: tara ületamisel läbitakse barjäär umbes 500 ms jooksul, kõndides 200–350 ja joostes 50–100 millisekundiga. Andurid võivad olla varustatud raadiosidega, väljas töötavad aga autonoomse toitega (nt päikesepatarei ja aku).

#### *15.2.1.5 Mikrolaineandurid*

Mikrolainegeneraatorid, mille resonantsiolekut mõjutab läheduses asuva inimese mahtuvus. Suhteliselt kallid, kõrge valehäirete sagedusega ning tekitavad parasiitkiirgust. Kasutatakse sõidukite kaitseks.

#### *15.2.1.6 Kontsentraatorid*

Kõigilt objektil paigaldatud anduritelt lähtuvad signaalid suubuvad kontsentraatorisse, millel võib olla ka kohaliku juhtpuldi funktsioone. Üldtunnustatud standardid puuduvad. Seadmete skaala ulatub tööstusautomaatika andurikontrolleritele rajatud süsteemidest lokaalsete valvepultideni. Kontsentraator edastab alarmiteated raadiokanali või telefoniliini kaudu valvekeskusesse.




**Joonis 96. Kohaliku alarmikeskuse operaatoriliidese näide**

Üha rohkem rakendatakse tavalisi lauaarvuteid, millel on lisaks kohalikule graafilisele või poolgraafilisele operaatoriliidesele (vt Joonis 96) ka kaugjuhtimisliides. Näiteks võimaldab Windowsi all töötav süsteem Escort (ICAMS) kaugpöördusi telefoni teel; sissehelistamisel saab kaugoperaator telefoni klaviatuurilt sisestada parooli ja juhtkoodid, süsteem aga teatab talle nõutavad andmed kõnesüntesaatori vahendusel. Kontsentraatorarvutile saab anda mitmesuguseid väljastusfunktsioone, näiteks teadistust elektronposti või faksi kaudu.

Kontsentraatoritega ühendatakse lisaks valvesignalisatsiooni anduritele tihti ka tuletõrjeandurid, elektronlukusüsteemid jm signaaliallikad, nt kliimaseadmete ja muude automaatjuhtimisobjektide olekusignaalid.

#### *15.2.1.7 Väljundseadmed*

Tüüpilised kohalikud väljundseadmed, mis käivitatakse otse anduriga või kontsentraatori kaudu, on piesosireenid helivaljusega kuni 120 dB ning vilkurid; viimaseid kasutatakse ka hoiatussignaalina enne tegelikku sissetungi.

Peale selle võidakse andurisignaale kasutada mitmesuguste toimingute käivituseks, näiteks ruumide valgustuse sisselülituseks, tõkestusseadmete aktiveerimiseks jne.

Keskpultide tüüpiline spetsialiseeritud välisseade on graafiline tabloo, mis võimaldab esitada seireandmeid topograafiliselt, valvealuste objektide plaanil.

### *15.2.1.8 Euronõuded*

Eurostandard EN 50131-1 jagab valvesignalisatsiooni süsteemid volituste andmise, pääsutasemete, operatsioonide, töötluse, avastamise, teadistamise, toite, sekkumiskindluse, ühenduste seire ja sündmuste registreerimise järgi nelja klassi.

***Klass 1*** on määratud väikese riskiga objektidele. Sissetungijalt eeldatakse väheseid teadmisi valvesüsteemide alal ning piiratud valikus lihtsaid instrumente.

***Klass 2*** on määratud väikese kuni keskmise riskiga objektidele. Sissetungijalt eeldatakse piiratud teadmisi valvesüsteemide alal ning laiemas valikus instrumente ja portatiivseid mõõteriistu.

***Klass 3*** on määratud keskmise kuni suure riskiga objektidele. Sissetungijalt eeldatakse valvesüsteemide tundmist ning küllaldases valikus instrumente ja portatiivseid elektroonikaseadmeid.

***Klass 4*** on määratud suure riskiga objektidele, kus turvalisusel on prioriteet kõigi muude tegurite ees. Sissetungijailt eeldatakse sissetungi üksikasjaliku kavandamise võoimet või ressursse ning täielikus valikus varustust, kaasa arvatud valvesignalisatsiooni süsteemi eluliselt tähtsate komponentide asendamise vahendeid.

Muuhulgas esitab standard nõuded süsteemide sekkumiskindlusele ja sekkumiste tuvastusele (Tabel 35).

**Tabel 35. Tuvastamisele kuuluvad sekkumisviisid**

| Sekkumisviis | Klass 1 | Klass 2 | Klass 3 | Klass 4 |
|---|---|---|---|---|
| Avamine tavaliste vahenditega | K | K | K | K |
| * Mahavõtt paigalduskohast | F | K | K | K |
| Hoiatusseadmesse tungimine | F | F | K | K |
| Sekundaarseadmetesse või edastussüsteemi tungimine | F | F | F | K |
| Detektori orientatsiooni muutmine | F | F | K | K |
| Liikumisdetektorite maskimine | F | F | K | K |
| **Perioodiline side süsteemi komponentide vahel** | | | | |
| Periood, h | 4 | 2 | 1 | 0,25 |

K = kohustuslik, F = fakultatiivne, * = ainult juhtmetute detektorite puhul

Süsteemi klassist sõltub registreerimisele kuuluvate sündmuste valik (Tabel 36). Klasside 2, 3 ja 4 puhul tuleb registreerida ka iga sündmuse asetleidmise kuupäev ja kellaaeg. Klassides 3 ja 4 tuleb tagada ka sündmuste püsiva jäädvustamise võimalus, nii et ühe valveperioodi jooksul jäädvustataks kolm igas üksikallikas registreeritud sündmust.

**Tabel 36. Sündmuste registreerimine**

| Sündmus | Klass 1 | Klass 2 | Klass 3 | Klass 4 |
|---|---|---|---|---|
| Kasutaja identiteet valve aktiveerimisel ja desaktiveerimisel | F | F | K | K |
| Süsteemi aktiveerimine | F | K | K | K |
| Süsteemi desaktiveerimine | F | K | K | K |
| Üldine rike | F | K | K | K |
| Sissetungialarm | F | K | K | K |
| Alarmieelne tsoon | F | K | K | K |
| Alarmi allikas | F | K | K | K |
| Sekkumise alarm | F | K | K | K |

| | | | | |
|---|---|---|---|---|
| Primaartoite rike | F | F | K | K |
| * Patarei vahetuse vajadus | F | F | K | K |
| Isolatsiooni tuvastus sisse | F | K | K | K |
| Isolatsiooni tuvastus välja | F | K | K | K |
| Blokeeringu sisselülitus | F | K | K | K |
| Tühistus | F | K | K | K |
| ** Edastussüsteemi tõrge | F | K | K | K |
| Kuupäeva ja kellaaja muutmised | F | F | K | K |
| Kohaspetsiifiliste andmete muutmine | F | F | K | K |
| Perioodilise side tõrge | F | K | K | K |
| Komponentide asendamine | F | F | F | K |
| Signaalide või sõnumite asendamine | F | F | F | K |
| Sidevahendite käideldavus | F | F | F | |
| **Nõuded registreerimisele** | | | | |
| Mälumaht, sündmust | F | 100 | 200 | 500 |
| Mälusisu säilivus toite katkemisel, päeva | F | 30 | 30 | 30 |

K = kohustuslik, F = fakultatiivne, * = ainult kuivelementide puhul, ** kui on rakendatav

Standardi ülejäänud osad on järgmised.

2-1: Sissetungidetektorid. Üldnõuded
2-2: Sissetungidetektorid. Ruumdetektorid
2-3: Sissetungidetektorid. Pinddetektorid
2-4: Sissetungidetektorid. Joondetektorid
2-5: Sissetungidetektorid. Punktdetektorid
3: Juhtimis- ja indikatsiooniaparatuur
4: Hoiatusseadmed
5: (Reserv)
6: Toiteallikad
7: Rakendamissuunised

### 15.2.2 Tuletõrjesignalisatsioon

Andurid jagunevad oma kontrollifunktsioonilt kolme põhiliiki.

***Termoandurid*** reageerivad temperatuurile, mis ületab etteantud läviväärtuse. Tajuriks on enamasti kergestisulavast materjalist element, bimetallelement või termistor.

***Suitsuandurid*** tuvastavad suitsu. Levinud liik on aktiivsed infrapunaandurid, mille vastuvõtupool registreerib ta läheduses paikneva kiirguri IP-kiire nõrgenemise või katkemise. Neist tundlikumad on ionisatsioonandurid, mille kambris tekitatakse ionisatsioon nõrga (umbes 1 mikrokürii) gammakiirgusega; kiirgusallikaks on radioaktiivne isotoop, nt $Am_{241}$.

***Gaasiandurid*** põhinevad mitmesugustel selektiivsetes gaasianalüsaatorites kasutatavatel efektidel. Neid valmistatakse peamiselt propaani, butaani, etanooli, propanooli, süsihappegaasi ja sisepõlemismootori heitgaaside tuvastuseks.

Teostuselt jagunevad andurid punkt- ja liinanduriteks ning analoog- ja diskreettoimelisteks.

***Diskreettoimelised*** on näiteks kõik lihtsad sulavelemente sisaldavad andurid, sh liin-termoandur. Viimane kujtab endast soonte kergsulava isolatsiooniga bifilaarkaablit; sooned on mehaaniliselt eelpingestatud ja isolatsiooni sulamine tekitab nende vahel lühise, mille asukohta saab anduri kontrolleris määrata liinikontuuri takistuse mõõtmisega. Sellist liinandurit saab kasutada mitte ainult ruumide, vaid ka kaablitrasside, vahelagede, ventilatsioonikanalite jms seireks.

***Analoogandurid*** võimaldavad mitte ainult teatada läviväärtuse ületamisest, vaid jälgitavat suurust mõõta tunduvalt laiemas alas, muuta läviväärtusi, kompenseerida vananemise toimet jne. Nüüdisaegsed andurid

on varustatud individuaalse aadressiga, sisaldavad sageli mälu ja ka mikroprotsessorloogikat. Tavaliselt skaneerib kontroller või kontsentraator neid perioodiliselt; kui kontrollitava suuruse väärtus läheneb lävele, skaneerib ta seda andurit sagedamini ning saab anda ennetavaid väljundsignaale. Andur ise väljastab signaali läviväärtuse ületamisel.

Sekundaarseadmete (kontsentraatorid, puldid jms) arhitektuur ei erine oluliselt valvesignalisatsiooni omast. Üha sagedamini need süsteemid integreeritakse.

### 15.2.3 Valvetelevisioon

#### *15.2.3.1 Valvekaamerad*

Mustvalged või värvilised laengsidestus-pildianduriga (CCD) videokaamerad püst-eraldusvõimega 300-600 laotusrida; rahuldava kvaliteediga kujutise annab vähemalt 500 rida. Erinevad valgustundlikkuse, toite, keskkonnanõuete ja väljundi tüübi (sh digitaalväljundi olemasolu) poolest.

Objektiivide valgusjõud on tavaliselt 1:1,2...1:1,4, fookuskaugus 2,8...12 mm, vaatenurk 7...93ț. Kui objektiivil on automaatdiafragma, peaks kaameral olema ka elektronkatik, muidu reageerib ta halvasti järskudele valguse muutustele.

Varjatud jälgimiseks või pääsumehhanismidesse paigutamiseks määratud kaameratel on objektiivi läbimõõt 0,9...2 mm, nende valgustundlikkus on piisav kujutise saamiseks 0,5 Lx juures. Diafragma tavaliselt puudub.

#### *15.2.3.2 Kaamerate lisaseadmed*

***Varioobjektiivid*** fookuskauguste piirkonnaga 6/60...8/160 mm võimaldavad vaatenurka muuta maksimaalselt vahemikus 2,5ț...44ț, st tekitada 5-...25-kordset "lähendust".

***Kaablivõimendid*** on vajalikud, kui kaamera ja sekundaaraparatuuri vaheline kaugus on üle 2 km.

***Signaalijaotur*** võimaldab kaamera väljundsignaali suunata mitmele monitorile või videomagnetofonile.

***Tekstigeneraator*** lisab kujutisele tekstandmed (kuupäeva, kellaaja jms).

***Kaitseboks*** on vajalik kaamera kasutamisel välistingimustes.

***Skaneerimisajam*** laiendab seireala kaamera regulaarse või kaugjuhitava pööramise teel. Tüüpilised skaneerimisulatused on rõhtsihis ±175ț, 360ț või ±360ț, püstsihis ±90ț, ±180ț või 360ț. Skaneerimiskiirus on 3...12 nurgakraadi sekundis.

***Laserprožektor*** seireala valgustamiseks ning objektiivile paigaldatud sellekohane valgusfilter nurjavad kaamera pimestamise objektiivi suunatud valgusvihuga. Kasutatav lainepikkus on 10 nm ümber, optilise kiirguse võimsus on 50 mW kuni 1W.

#### *15.2.3.2 Väljundseadmed*

***Monitorid*** on tavaliselt ekraani diagonaaliga 23...51 cm. Muude näitajate järgi valitakse nad vastavalt kaamera parameetritele ja monitori kasutusviisile.

***Videomagnetofonid*** jäädvustavad erinevalt tavalistest ainult näiteks iga viienda (või kümnenda jne) kaadri, seetõttu saab nendega tavalisele kassetile salvestada 4 kuni 960 tunni ulatuses. Enamasti saab neile programmeerida salvestussageduse muutuse alarmi puhuks. Kaadrile lisatakse aega jms sisaldav tekst. Valveotstarbeline videomagnetofon võimaldab lisaks tavalistele taasesitusviisidele ka taasesitust kaaderhaaval.

***Videoprinterid*** väljastavad kaamerast või videomagnetofonilt saadud kujutist. Eraldusvõime on harilikult 3,2...7 punkt/mm, hallskaala on 64- või 256-astmeline. Maksimaalne formaat on 800×600 pikselit.

#### *15.2.3.2 Väljundseadmete lisaseadmed*

***Videokompressor*** võimaldab ühe monitori ekraanile väljastada korraga mitme kaameraga saadavaid pilte; maksimaalne ekraaniväljade arv on tavaliselt 4 või 8. Täiuslikumatel kompressoritel võivad olla alarmisisendid, mis automaatselt täidavad kogu ekraani alarmiala kaameralt tuleva pildiga.

***Videomultiplekser*** võimaldab 4...16 kaamera signaale salvestada ühelainsal videomagnetofonil. Võib olla varustatud liikumisdetektori või alarmisisendiga, mis katkestavad vastava signaali korral teiste kaamerate pildi salvestuse, nii et jätkub ainult alarmiala pildi salvestus suurema kaadrisagedusega.

***Liikumisdetektor*** reageerib kontrastimuutusele või liikumisele kaadris. Ta väljundsignaali võib kasutada näiteks helisignaali andmiseks või videomagnetofoni (ümber)lülitamiseks. Täiuslikumad võimaldavad ette anda jälgitava tsooni kaadril (tsoone võib olla kuni 7-8) ja tundlikkuse (15...125 taset). Detektoril võib olla 1...16 videosisendit.

Peale selle võib süsteem sisaldada digiteerimis- ja sideseadmeid, arvutiliideseid jms.

## 15.3 Pealtkuulamise tuvastus

### 15.3.1 Pealtkuulamise meetodid

***1. Otsene akustiline*** pealtkuulamine võib toimuda lihtsate abivahenditega (stetoskoop, võimendiga mikrofon) läbi seinte, aga ka avade, ventilatsioonilõõride jms kaudu või sobivalt kauguselt suundmikrofoni kasutades.

***2. Olemasolevate mikrofonidega*** või vahenditega, mida saab kasutada mikrofonina, tegemata seadmetes mingeid muudatusi. Arvestada tuleks vähemalt järgmisi võimalusi:

- telefoni mikrofon vabade kätega kõnerežiimis võimaldab kõne ajal kuulda taustkõnelusi;
- idamaise päritoluga telefonide hulgas on ka selliseid, mille toru hargilepanek ei lülita mikrofoni välja;
- sisselülitatud arvutimikrofoni saab kasutada kohtvõrgu kaudu;
- mikrofonina toimivad professionaalse kuuldeaparatuuri kasutamisel näiteks telefoni kõlisti või summer (toru hargil olles), sisetranslatsiooni valjuhääldid, helisignalisaatorid jms põhimõtteliselt pööratava toimega elektromehaanilised seadised.

***3. Telefoni manipuleerimisega***, kui sellele on juurdepääs. On umbes 15 moodust telefoni vooluringide muutmiseks, nii et mikrofon või kuular oleks pidevalt või sissehelistusel liiniga ühendatud.

***4. Infotehniliste seadmete kiirguse vastuvõtt***. Sellised kiirgusallikad on näiteks

- arvuti kaablid ja välisseadmed, eriti modem,
- mobiiltelefon,
- faks,
- tavaline telefon (võib tekitada ka raadiosageduskiirgust).

***5. Salaharund sideliinil*** (otsene või induktsioon- või mahtuvussidestusega).

***6. Telefonijaama seadmete või tarkvara manipuleerimisega***. Seda saab teha mitte ainult vahetult keskjaamas, vaid ka kaugründega.

***7. Saatjaga salamikrofon***, mis võib edastuseks kasutada

- kõrgsagedussignaale olemasoleva telefoni- või elektriliini kaudu;
- raadiosignaali (areng toimub kõrgemate sageduste suunas);
- infrapuna- või ultraviolettsignaali;
- ultraheli.

Raadiosaatjaga salamikrofone on paljudes maades müügil suures valikus, sh hõõglampidesse jm peiteesemetesse ehitatuna.

***8. Saladiktofon***. Miniatuursed diktofonid on samuti muutunud laiatarbekaubaks.

***9. Helivõngete lugemine aknaklaasilt laserikiirega.*** Rakendatav kuni 500 m kauguselt, kiir tuleb suunata klaasile risti või vähemalt 60ț nurga all.

***10. Avaliku arvutivõrgu liikluse jälgimine.*** (Võrguspetsiifilisi ründeid ja turvameetmeid käsitletakse raamatu 3. osas.)

### *15.3.2 Tuvastusvahendid*

Meetodeid ja vahendeid on välja töötatud enamiku pealtkuulamisviiside tuvastuseks.

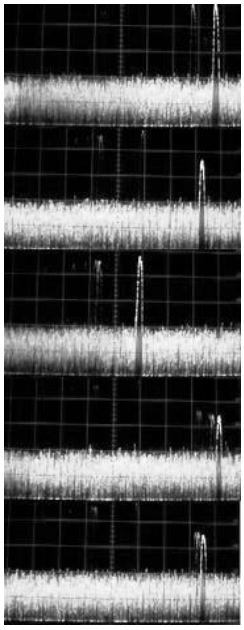
***Raadiosaatjaga salamikrofonide*** tuvastuse lihtsaim ja odavaim vahend on teatavat sagedusriba skaneeriv väljamõõtur; sisuliselt on see tundlik (alla 1 µV) lairibavastuvõtja, mis on varustatud mingi indikaatoriga (heli, vagusdioodide rida, numbernäidik). Sellise mobiiltelefoni või taskuarvutit meenutava instrumendi (Joonis 97, a) sagedusala võib ulatuda 1-2 gigahertsini ning ta võimaldab mingil määral avastada amatöörtasemele vastavaid "lutikaid". Tegelikkuses on salamikrofonide kõige populaarsem saatesagedus vahemikus 600 MHz kuni 9 GHz, kuni 22 GHz saatjad on muutunud üsna odavaiks, 22...60 GHz on veidi kallimad ning alles sellest kõrgemate sagedustega seadmed on kallid ja suhteliselt raskesti kättesaadavad. Pealegi kasutatakse hüppava sagedusega saatjaid üha rohkem.

Nõudlikumad seadmed töötavad märksa suuremas sagedusalas (võrdluseks: USA sõjaliste ja diplomaatiliste objektide kontrollimisel skaneeritakse sagedusi kuni 325 GHz), on varustatud mikroprotsessoritega ning sooritavad raadiospektri analüüsi, mis muuhulgas võimaldab avastada ka muutsagedusega töötavaid saatjaid (vt Joonis 97, c), sõltumatult modulatsiooni tüübist.

a

b

c

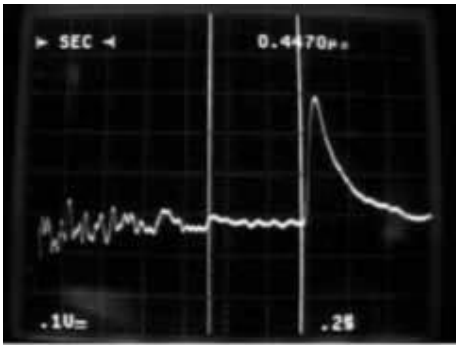


**Joonis 97. Salasaatjate tuvastus: a - odav toonindikatsiooniga raadiosageduste skanner, b - kehale kinnitatav vibroindikatsiooniga salasaatjate ja -diktofonide detektor, c - muutsagedusega salasaatjat näitav spektrogramm**

Tihti integreeritakse raadiosaatjaid tuvastavatesse seadmetesse ka vahendeid muude pealtkuulamisviiside kontrollimiseks (Joonis 97, b).

***Telefoniliinide salaharundite*** tuvastuse lihtsaim meetod on liini pingelangu mõõtmine; see võimaldab teha ka jämedaid oletusi harundi asukoha suhtes. Meetod ei toimi, kui harund ei ole ühendatud vahetult liiniga, vaid kasutab induktsioon- või mahtuvussidestust. Täiuslikumad instrumendid rakendavad reflektomeetrilist analüüsi ajamõõtmes: liinile väljastatakse impulsse, mis peegelduvad harundilt; harundi kaugus määratakse peegeldusaja järgi (Joonis 98, b).

a b


**Joonis 98. Telefoniliinide seire: a - lihtne detektor harundite ja raadiosageduste tuvastuseks liinil, b - reflektomeetriline analüüs ütleb, et 150 m kaugusel on salaharund**

Teostuselt võivad seadmed olla määratud pisteliseks kontrolliks või olulisele liinile alaliselt paigaldamiseks. Täiuslikuma tööpõhimõttega instrumendid võimaldavad harilikult avastada ka telefoniliinidesse juhitud raadiosagedussignaale.

***Saladiktofone*** avastatakse peamiselt nende mootorite helipeade tekitatavate magnetväljade järgi. Kuna selline magnetväli on üsna nõrk, tuleb mõõtesond viia diktofonile küllaltki lähedale (15 cm kuni 1 m, sõltuvalt detektori kvaliteedist).

***Infrapuna-, ultraviolett- ja laserikiirgus*** tuvastatakse optilise sondiga, mis koosneb sobiva spektraalkarakteristikuga fotoelektrilisest andurist ja sobiva valgusfiltriga varustatud objektiivist. Sond (Joonis 99) ühendatakse indikatsiooni saamiseks tavaliselt mingi universaalse pealtkuuldedetektori põhiplokiga või on sellesse sisse ehitatud.

**Joonis 99. Infrapunakiirgust tuvastav sond**

***Ultrahelikiirgus*** tuvastatakse akustilise sondiga, mille mikrofoni ja eelvõimendi sagedusriba häälestatakse ultraheli sagedustele. Ka see sond on tavaliselt universaaldetektorite lisavahend.

Mõned pealtkuulamisviisid ei ole tehniliste vahenditega tuvastatavad. Sellistel juhtudel tuleb seiret sooritada muul viisil või rakendada neist sõltumatult tõrjemeetmeid (akustilise müra generaatorid otsese pealtkuulde tõrjeks, telefoniside skrambleerimine, ekraanifiltrid kõrvaltpiilumise tõkestuseks jne).

***Arvutisüsteemide elektromagnetilise kiirguse*** kahandamiseks tuleks silmas pidada alljärgnevaid võimalusi, kontrollides kriitilisi kohti raadiosaatjate kontrolliks määratud detektoritega, millel on sobiv sagedusala.

1. Metallivaba steriilse tsooni loomine arvuti iga seadme (eriti kuvarite ja printerite) ümber, vältides näiteks metallist või metalljalgadega töölaudu, metallist prügikorve jne.
2. Telefonide hoidmine arvuti seadmetest (eriti kuvaritest) eemal, varustades vajaduse korral telefoniliini andmevahetussagedust tõkestava filtriga.
3. Võrgutoite filter mitte ainult kõrvaldab võrguhäiringuid, vaid töötab ka vastasssuunas, tõkestades andmekiirguse levimist toitejuhtmete kaudu.

4. Arvuti seadmete paigutamine akendest eemale ja mitte fassaadiküljele.
5. Kvaliteetsemate, madalama kiirgusega seadmete eelistamine riistvara hankimisel, jälgides kiirgusklassi atesteeringut.
6. Väljundseadmed kiirgavad andmeid ainult siis, kui neid parajasti väljastatakse. See tähendab, et ekraanilt tuleks andmed kõrvaldada kohe, kui töö nendega on lõpetatud, ja et kasulik on vältida ülemääraseid printväljastusi.
7. Andmekiirgust saab peita tehislikku mürakiirgusse. Selleks saab tundlikke andmeid töötlevad seadmed ümbritseda avalike andmete töötluse seadmetega, mis pealegi on paigutatud "vääralt", st kõiki ülalloetletud reegleid rikkudes.
8. Valguskaablid seadmetevahelisteks ühendusteks.
9. Minimeeritud kiirgusega (*Tempest-proofed*) eriaparatuuri kasutamine kõige kriitilisemates kohtades.
10. Tundlike arvutisüsteemide paigutamine varjestatud ruumidesse või kaitsekappidesse.

# 16 VARUNDAMINE

## 16.1 Varundamise olemus ja otstarve

Varundus (*back-up*) on liiasusel põhinev käideldavuse ja tervikluse tugevdamise abinõu ning tähendab infosüsteemi varuvarade loomist või soetamist varade osalise või täieliku hävimise või kasutamiskõlbmatuks muutumise puhuks. Varunduse näited on andmete varukoopiate loomine, varuarvutite soetamine, töötajarollide dubleerimine ajutiseks või alaliseks asendamiseks. Kuumvarundid on sellised varuressursid, mis asendavad põhiressursi selle väljalangemisel viivitamatult või peaaegu viivitamatult (nt dubleeritud riistvara, RAID-kettasüsteemid, vt 16.3, jne). Varunduse ulatus (süsteemi osade või kogu süsteemi dubleerimine jne) sõltub konkreetsest turvatarbest. Kriitiliste, eriti sõjaliste rakenduste korral dubleeritakse kogu töötluskeskus, kusjuures varukeskus paigutatakse teise kohta, piisavalt kaugele põhikeskusest.

***Andmed*** on tavaliselt kõige olulisem ja hinnalisem infotehnoloogiline vara, ühtlasi kõige muutuvam, seetõttu tuleb kõiki varunduskavasid alustada andmetest.

***Riistvara*** puhul tuleb lähtuda kõige väiksema töökindlusega osadest. IDC sooritatud uuring reastab tõrgete sagedused nii:

| | |
|---|---|
| kõvaketas | 55% |
| toiteplokk | 28% |
| ventilaatorid | 8% |
| mälu | 5% |
| kontrollerid | 4% |

Kuna suurim tõrkesagedus on kõvakettal, rõhutab see veelgi vajadust eelkõige varundada andmeid, tehes seda piisava sagedusega.

***Tarkvara*** varundamine lahendatakse eeskätt organisatsiooniliste meetmetega (programmide originaalketaste füüsiliselt turvaline säilitus selgelt märgistatuna ning kirjutuse eest kaitstuna). Infotehnilisi abinõusid tuleb rakendada ainult kriitilistes reaalajasüsteemides ja need puudutavad kogu arvuti varundamist.

***Infrastruktuuri*** puhul tuleb tähelepanu pöörata eelkõige toite kuumvarundusele ja kliimaseadmetele.

## 16.2 Andmevarundus

Andmetest valmistatakse varukoopiad sobivale vahetatavale andmekandjale enamasti käsioperatsioonidega või regulaarselt rakenduva tarkvara abil, kuid varundustoiminguid võib käivitada ka riistvaras teostatud moodul. Sõltuvalt konkreetsetest tingimustest võidakse varundada kogu kõvaketast, teatud katalooge või teatud faile. Kopeerimine võib olla täielik või keerukam, kuid ressursisäästlikum inkrementaalne; viimasel juhul kopeeritakse ainult muutunud osad.

### 16.2.1 Andmekandjad

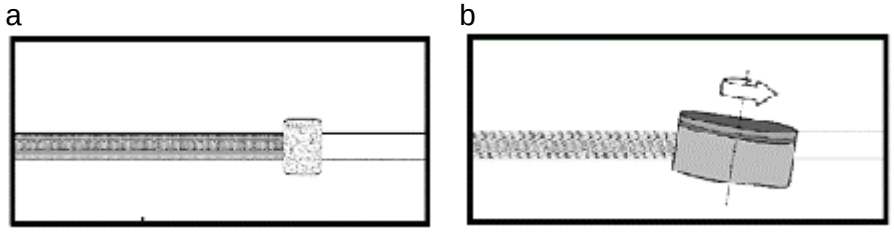
Varukopeerimiseks sobiva andmekandja valimisel tuleb arvestada andmemahte, ajakulu kopeerimisele ja taastamisele, andmekandja tööiga ja töökindlust, vahendite hinda ning suurte varundus- ja arhiveerimismahtude korral ka andmekandjate füüsilisi mõõtmeid.

#### *16.2.1.1 Magnetlint*

Üldiselt odavaim vahend suurte andmemahtude (näiteks kogu kõvaketta) varukopeerimiseks, ehkki ei jõua mahutavuselt enam ketastega kuigi edukalt võistelda. Salvestustiheduse suurendamiseks kasutatakse üha õhemaid põhimikke ja magnetkihte, radade tihendamiseks täiustatakse magnetpäid ning rakendatakse traditsioonilisest erinevaid radade paigutusi. Arhiivi salvestusmahu suurendamiseks kasutatakse automatiseeritud karussell- või koordinaattüüpi salvesid, mis võivad mahutada kümneid, sadu või tuhandeid kassette. Suurema mahu korral on nad varustatud robotkäega.

***Rööpsete pikiradadega*** üherulliline 0,5-tolline kassett lindi kiirusega 2m/s ja üle selle on ikka veel üsna levinud ning mõnedel andmetel kasutab neid üle 90% maailma andmekeskustest. Kirjutus ja lugemine toimub liikumatute magnetpeade plokiga. Draiv on umbes lauaarvuti süsteemiploki suurune. Radade arv üha kasvab; lähitulevikuks ennustatakse kuni 1000-rajalist kirjutust. Andmevahetuskiirused on kuni 52 Mbit/s. Salved mahutavad kuni 6000 kassetti. Tüüpilisi standardmudeleid:

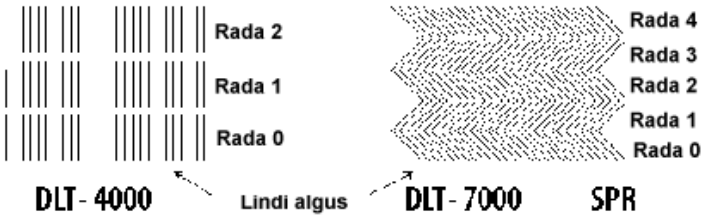
3480 (IBM): 18 rada, 200 MB
3490 (IBM): 36 rada, 800 MB




**Joonis 100. Magnetlindi kirjutusviisid: a - süstikkirjutus, b - kaldkirjutus**

***Süstikradadega*** lindil paiknevad rajad vaheldumisi vastassuundades (vt Joonis 100, a ja Joonis 101, a); selliseks kirjutuseks nihutatakse magnetpeade plokki külgsihis ja reverseeritakse linti. Radade arv on 9...144. Kasutatakse ka magnetpeade keerukamat liikumist, millega tekitatakse kalasabakujuline või siksakiline rajamuster (vt Joonis 101, b). Lauaarvutitega töötamiseks on määratud veerandtolline QIC, mille draivid vastavad mõõtmetelt 3,5- ja 5,25-tollistele disketidraivide lahtritele. Salvestusmahtude ja andmevahetuskiiruste praegused ja oodatavad maksimaalväärtused on alljärgnevad.

| | | | | | |
|---|---|---|---|---|---|
| QIC 5,25": | 25 GB | 12 Mbit/s | Prognoos aastaks 2000: | 180-200 GB | 56 Mbit/s |
| QIC 3,5": | 4 GB | | | 30 GB | |
| Magstar: | 10 GB | 72 Mbit/s | | | |




**Joonis 101. Süstikkirjutuse variante: vasakul lineaarne, paremal SPR (symmetrical phase recording, "sümmeetrilise faasiga salvestus")**

Ülevaate levinumatest süstikdraivide standarditest annab Tabel 37.

**Tabel 37. Süstikdraivide standardeid**

| Standard | Salvestusmaht* | Liides | Sobivus |
|---|---|---|---|
| QIC 80 | 120/250 MB | FDD | Aeglane, vananenud, väike maht |
| QIC 3010 | 340/680 MB | FDD | Aeglane, vananenud, väike maht |
| Travan 1 | 400/800 MB | FDD või IDE | Vananenud, väike maht |
| QIC 3020 | 680 MB/1360 GB | IDE | Ei sobi üle 2 GB kõvaketastele |
| QIC 1000 | 1,2/2,4 GB | IDE | Sobib väiksematele kõvaketastele |
| QIC 2 GB | 2/4 GB | IDE | Väikestele ja keskmistele süsteemidele |
| Travan 3 | 1,6/3,2 GB | IDE | Väikestele ja keskmistele süsteemidele |
| QIC 3095 | 2,1/4,2 GB | IDE | Sobib enamikule süsteemidele |
| Travan 4 | 4/8 GB | IDE | Sobib kõigile lauaarvutitele |

* Pakkimata/pakitult

***Kaldradadega*** lindile kirjutatakse rajad väikese (4ţ...7ţ) nurga all, pöörleva magnetpeade plokiga – nii nagu videosalvestuse puhul (vt joonis 108, b). Lindi laius on 4...19 mm, standardse helikasseti gabariitidest (populaarne DAT) standardse TV-videokasseti omadeni. Näiteid on toodud Tabel 38; sulgudes olevad arvandmed vastavad väljatöötamisel olevale tootele.

**Tabel 38. Kaldradadega kassettlinte**

| Lindi laius | Tüüp | Gabariit | Salvestusmaht, GB | Salvestuskiirus, Mbit/s | Söötesalve maks. maht, tk |
|---|---|---|---|---|---|
| 4 mm | DDS2<br>DDS3<br>DDS4 | DAT | 4<br>12<br>24 | <br>6,2<br>8 | 218 |
| 8 mm | (Exabyte)<br>(Sony) | 5,25" lahter<br>3,5" lahter | 40 (80)<br>25 (35)* | 32 (48) | 10...58880 |
| 0,5" | DTF<br>(Legacy) | <br>T-180 VHS | 12/48<br>100 | 96<br>32 | 70/35 |
| 19 mm | DD1<br>DD2 | TV-kassett | 96<br>330 | 120 | 16000 |

* Kassett on varustatud kasutamist kiirendava indeksikiibiga.

### *16.2.1.2 Magnetketas*

Magnetketaste salvestusmaht on hakanud kiiresti kasvama. Arengut mõjutavad muuhulgas järgmised tegurid:

- suuremat salvestustihedust võimaldavate magnetmaterjalide kasutamine ketta magnetkihis;
- magnet-takistuslikud lugemispead, mis põhinevad takistuse muutumisel magnetvälja toimel, on tundlikumad ja võimaldavad lugeda tihedamat salvestust;
- digitaalne (sh optiline) positsioneerimine võimaldab suuremat raja- ja salvestustihedust;

- kõvaketaste alumiiniumpõhimiku asendamine keraamilisega tõstab ketta jäikust ja vähendab ääreefekte.

***Diskett*** oma senisel kujul kõlbab oma väikese mahu tõttu varundusandmekandjaks vaid erandjuhtudel. Üleminek uuele standardile (120 MB) nõuab veel rohkesti aega. Iomega Bernoulli ketta asemele astunud 100 MB Zip-ketas (Joonis 102) samalt firmalt on veel suhteliselt kallis ega võistle vähegi mahukama varunduse korral DAT-lindiga.


**Joonis 102. Magnet-andmekandjaid: Zip-diskett ja kassettketas Syjet**

***Kõvaketas*** konkureerib salvestusmahult magnetlindiga edukalt; maht ulatub 32 gigabaidini ja üle selle. Tavalise konstruktsiooniga kõvakettad on varundamisel rakendatavad peamiselt RAID-konfiguratsioonides (vt 16.3). Külmvarunduse ja arhiveerimise otstarbeks saab kasutada ird- ja kassettkettaid. Esimesed neist kujutavad endast koos draiviga vahetatavat kõvaketast; põhimõtteliselt saab neid valmistada samasuguse salvestusmahuga kui tavalisi kohtkindlaid. Kassettketta ("kõvadisketi") näited on Iomega 1 GB ja 2 GB Jaz-ketas ning Syquesti 1,5 GB SyJet (Joonis 102).

### *16.2.1.3 Optilised andmekandjad*

***Magnetoptilised kettad*** on salvestusmahtudelt kõvaketastega samas suurusjärgus. Standardne läbimõõtude valik on 2,5-3,5-5,25-12-14 tolli. Levinuimad on 640 MB mahuga 3,5-tollised ja 2,6 GB mahuga 5,25-tollised. Automaatsöötesalv võib mahutada nt 35 tk 3,5-tolliseid. Uusim suund on faasivahetustehnika (PD), mille puhul jääb ära kustutusfaas ja kiireneb kirjutus. PD-l põhineb ka eurostandardiks kujunev 120 mm ketas; selle draiv loeb ka laserkettaid (CD-ROM). 120-millimeetrise PD jaoks valmistatakse kuni 100 ketast mahutavaid söötesalvesid. Nikoni 12-tolline PD-ketas mahutab 8 GB, Lockheedi 14-tollise maht on 12 GB.

***Kirjutatav laserketas*** (CD-R) ei sobi arvutirakendusteks oma jäiga andmepaigutusviisi, aegluse ja suhteliselt väikese salvestusmahu (640 MB) tõttu. Veel ei ole saadaval DVD-RW (3 GB) ega DVD-RAM (2,6 GB), mis on küll mahult suuremad, kuid samuti aeglased.

***Optiline magnetlint*** sarnaneb salvestusprintsiibilt laserkettale: põhimikule (samasugusele nagu magnetlindil 3480-kassetis) on kantud peegeldav metallikiht, mis on kaetud tumeda polümeerikihiga ja kaitsekihiga. Pooletolline lint mahuga 1 TB on paigutatud 3480 tüüpi kassetti. Andmevahetuskiirus on 120 Mbit/s. Kasutada saab 3480 jaoks valmistatud söötesalvesid.

Sobiva varundus-andmekandja valimisel tuleb lisaks salvestusmahule ja kiirusele arvestada ta maksumust, suuremate varundusmahtude korral ka mõõtmeid. Arhiveerimise korral on oluline ka andmekandja säilivusaeg. Mõningaid pidepunkte nende parameetrite osas annab Tabel 39.

**Tabel 39. Andmekandjate maksumus, ruumitarve ja säilivusaeg**

| Andmekandja tüüp | Maksumus, $/GB* | Ruumitarve, $cm^3$/GB | Säilivusaeg, a** |
|---|---|---|---|
| **Magnetlint:** | | | |
| 3480/3490 | 18,0 | 418 | 10-30 |
| QIC | 11,0 | 75 | 5-30 |

| | | | |
|---|---|---|---|
| DAT | 3,6 | 20 | 10-15 |
| 8 mm | 2,5 | 20 | 2-30 |
| VHS | 1,0 | 37 | 5-15 |
| **Magnetketas:** | | | |
| Zip | 17,0 | 150 | 10-15 |
| SyJet | 50,0 | 20 | 10-15 |
| **Optiline ketas:** | | | |
| MO 3,5" | 13,9 | 457 | 5-100 |
| MO 5,25" | 7,3 | 168 | 5-100 |
| CD-R | 17,5 | 286 | 5-100 |

* Hindade kiire muutumise tõttu võib selle veeru väärtusi vaadelda lihtsalt võrdlust võimaldavate suhteliste ühikutena.
** Sõltub valmistusmaterjalidest, seega konkreetsest margist.

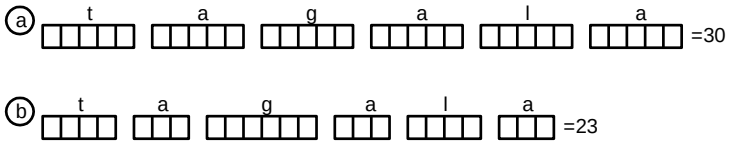
### 16.2.2 Andmetihendus

Varundamiseks ja arhiveerimiseks määratud andmekandjate ökonoomsemaks kasutamiseks rakendatakse andmetihendust. Tihendus põhineb tarbetu liiasuse kõrvaldamisel andmetest.

Sõna "tarbetu" on siin oluline. Kogu varundus põhineb ju just liiasuse, kuid kasuliku liiasuse lisamisel. Andmete käideldavuse ja tervikluse tõstmiseks rakendatakse kasulikku liiasust juba kõigi andmetöötlusprotsesside alatasemel, näiteks veaavastuslike ja veaparanduslike kontrollkoodidena, alates paarsusbitist ja kontrollsummast ning lõpetades tsükkelkoodidega. Rida selliseid koode on standarditud (ISO 7064, CCITT V.41 jt).

#### *16.2.2.1 Meetodid*

Informatsiooniteooriat appi võtmatagi võib jõuda järeldusele, et salvestatavate märkide erineva esinemissageduse tõttu on otstarbekas kodeerida märke mitte võrdse pikkusega andmeüksusteks (nt baitideks), vaid esitada muutuva pikkusega koodidena, mis on seda lühemad, mida sagedam on märk (vt Joonis 103). Ehkki nüüd tuleb koodide eraldamiseks kasutada lisakoodi (prefiksit), võib kokkuvõttes ruumi säästa.




**Joonis 103. Statistilise tihenduse olemus: a - tihenduseta kodeerimine, b - tihendav kodeerimine**

Sellise kooditabeli saaks koostada näiteks teadaoleva keelestatistika põhjal ja rakendada teda kõikvõimalikele andmetele. Ilmselt ei oleks see lahendus optimaalne, sest konkreetne kodeeritav tekst võib tugevalt erineda statistilisest keskmisest. (Ilmekas näide: E.V.Wrighti 1939. a kirjutatud romaan *Gadsby* ei sisalda inglise keele sagedaimat tähte *e*.) See asjaolu sundis otsima meetodeid, mis põhineksid kodeeritavate andmete jooksval statistilisel analüüsil.

Püsiva kooditabeli (*codebook*) meetod võib teatavate tekstitüüpide (nt programmide lähtekoodide) puhul siiski leida rakendust. McIntyre ja Pechura demonstreerisid 1985. a nelja programmeerimiskeele lähtetekstide kodeerimisega, et koodi liiasus oli halvimal juhul ainult 6,6% suurem allpool nimetatud Huffmani algoritmiga saadavast, seevastu oli kodeerimine märksa kiirem.

***Staatilised*** tihendusalgoritmid sooritavad esimese sammuna lähteteksti sagedusanalüüsi, teise sammuna kodeerivad teksti, kasutades esimesel sammul leitud sagedusi. Esimese sellise algoritmi töötasid välja

C.Shannon ja R.Fano (1949); oma olemuselt ei garanteeri see optimumi, st liiasuse täielikku kõrvaldamist, kuid tihendamine läheneb asümptootiliselt optimumile lähteteksti pikkuse lähenemisel lõpmatusele. Esimese optimaalse (teatud tingimustel) algoritmi lõi D.Huffman (1952). Tuntumaid staatilisi algoritme on veel nn aritmeetiline kodeerimine (P.Elias, N.Abramson, 1963).

***Dünaamilised*** e adaptiivsed algoritmid määravad ja korrigeerivad sagedusi vahetult kodeerimise käigus, olles seega ühesammulised ja põhimõtteliselt kiiremad; optimaalsuselt liiasuse mõttes ei tarvitse nad jääda alla staatilistele. Tuntumad dünaamilised algoritmid on adaptiivne Huffmani kodeerimine (N.Faller, 1973) ja selle edasiarendused LZ (A.Lempel, J.Ziv, 1977), FGK (N.Faller, R.Gallager, D.Knuth, 1985), BSTW (J.Bentley jt, 1968), V (J.Vitter, 1987).

Meetodid varieeruvad veel näiteks selle poolest, et osa neist teisendab mitte püsipikkusega tekstiüksusi muutpikkusega koodideks (nagu ülaltoodud arutluses), vaid muutpikkusega tekstiüksusi püsi- või muutpikkusega koodideks.

***Semantikapõhised*** kodeerimisviisid lähtuvad konkreetsest rakendusest, arvestades seda, et lähtetekst pole juhuslik märgijada, vaid sisaldab teatavat ettemääratust.

Pilditöötluses kasutatakse näiteks sarikodeerimist (*run-length encoding*), mis teisendab kujutise paarideks (*c,l*), kus *c* on pikseli väärtus ja *l* sellise väärtusega pikselite sarja pikkus. See meetod sobib ka näiteks äriandmetele, mis sisaldavad pikki nulli- või tühikusarju jms.

Inkrementkodeerimisega (*difference mapping*) on kosmoserakendustes suudetud 8-bitise pikseliga pilt tihendada keskmiselt kolme bitini pikseli kohta.

Sõnastikuviitadeks kodeerimine (*dictionary substitution*) on tõhus atribuudiväärtuste piiratud arvu korral, nt andmebaasides: andmevälja *sugu* väärtuse saab baidi asemel esitada bitiga. Viidad võivad esitada terveid fraase ja suuremaidki tüüpseid tekstiplokke.

#### *16.2.2.2 Teostus ja tõhusus*

Tihendustoodetes kasutatakse tõhusamate algoritmide kombinatsioone ja variante. Tihendusalgoritmid on enamasti teostatud tarkvaras, utiliitide või rakendustesse (nt andmebaasisüsteemidesse) ehitatud moodulite kujul. Üha enam ilmub turule teostusi integraallülitustena, mida ehitatakse varundusotstarbelistesse salvestitesse, eriti lindidraividesse.

Tarkvaras teostuse näiteid:

- Unixi utiliidi *compact* aluseks on dünaamiline Huffmani kodeerimine;
- Unixi utiliit *compress* (*.z*) põhineb Lempel-Zivi algoritmil (LZ);
- arj (*.arj*), gzip (*.gz*), WinZip (*.zip*) kasutavad algoritmide LZ77 ja Huffmani kodeerimise kombinatsiooni.

Lindidraivides kasutamise näiteid on Tabel 40. Kuna enamik neist põhineb algoritmil LZ, sõltub tulemus ilmselt algoritmi konkreetsest teostusviisist.

**Tabel 40. Lindidraivide tihendusandmeid**

| Draivi tüüp | Tihendusalgoritm | Tihendustegur |
|---|---|---|
| Sony Alt | ALDC (LZ variant) | 2,55 |
| Quantum DLT-7000 | DLZ (LZ variant) | 1,81 |
| Exabyte Mammoth | IDRC | 1,82 |
| HP DDS-2 | DCLZ (LZ variant, ECMA standard) | 2,29 |

Algoritmide kohta "puhtal kujul" on raske saada võrdlevaid tihendusandmeid. Mõningase ettekujutuse annab Tabel 41.

**Tabel 41. Tihendusalgoritmide testimise tulemusi**

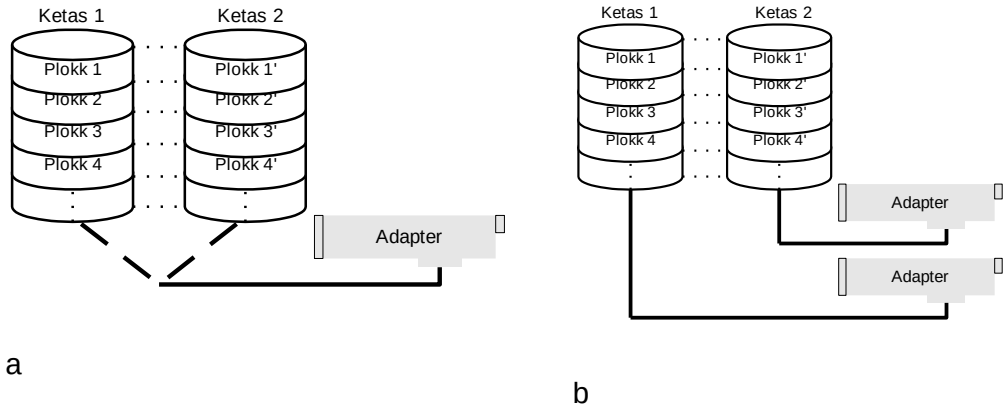
| Tihendusalgoritm | Andmete tüüp | Tihendustegur |
|---|---|---|
| St. Huffmani kodeerimine | Tekstifail<br>Pascali lähtekood | 1,2...1,5<br>1,68 |
| Dün. Huffmani kodeerimine | Andmebaas | 1,72 |
| Dün Huffmani kod. (Unixi *compact*) | Tekst<br>Lähtekood (Pascal/C) | 1,6<br>1,75/1,56 |
| Aritmeetiline kodeerimine | | 1,13...3,77 |
| LZ | Tekst<br>Lähtekood<br>Objektkood<br>Ujukomaandmed<br>Vorminguga teadusandmed | 1,8<br>2,3...2,6<br>1,5<br>1,0<br>2,1 |
| LZ (Unixi *compress*) | Tekst, lähtekood | 2...2,5 |

Nii eraldi testimisel kui ka toodete koostises on teistest veidi tihedamat pakkimist saavutanud Lempel-Zivi meetod. Tulemus sõltub aga tugevalt andmete tüübist ja mahust. Algoritmide kohandamisel konkreetsele rakendusele on saavutatud kuni 50-kordset tihendust. Halvasti alluvad tihendamisele graafikafailid, sest graafikavormingutes on liiasus juba niigi väike. Neid faile saab tihendada ainult kadudega, loobudes sellisest informatsioonist, mis kujutise kvaliteeti tajutavalt ei mõjuta.

## 16.3 Kuumvarunduse näide: süsteem RAID

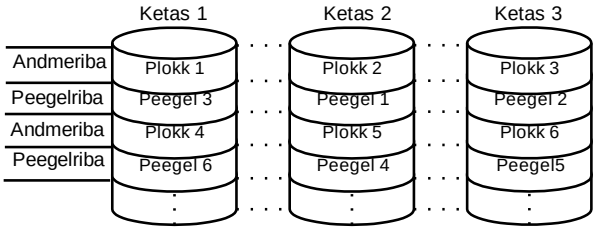
Kõvakettarühmade RAID-arhitektuur (*Redundant Array of Inexpensive Disks*, "odavate ketaste liiasmassiiv") töötati välja 1987. a. Berkeley ülikoolis. Eesmärk oli luua suhteliselt odavatest kõvaketastest suuremahuline mass-salvesti, mille tööd ei halvaks ühe ketta väljalangemine. Liiasus viidi sisse varukettana, mis võimaldas salvestatavaid andmeid täielikult või osaliselt (taasteks vajalikus ulatuses) dubleerida. algselt määratleti viis taset (RAID 1,..., RAID 5). Hiljem hakati nimetusega RAID 0 tähistama liiasuseta kettasüsteemi, milles rakendatakse RAID-ile iseloomulikku ribadena salvestust.

Erinevalt teistest variantidest ei kasuta RAID 1 hajusat ribadena salvestust, iga fail kopeeritakse tervikuna – peegeldades (Joonis 104, a) või dubleerides (Joonis 104, b). Lugemiskiirus kasvab, sest lugemistaotlusi saab suunata mõlemale kettale. Dubleerimise (adapteri eri kanal või eri adapter) korral on käideldavus suurem, sest ka kanal või adapter on varundatud. Kuna ketas varundatakse täielikult, on RAID 1 kulukas. Teda peetakse sobivaks väikestele võrkudele, kus tehingute maht on väike ja prevaleerivad lugemisoperatsioonid.




**Joonis 104. RAID 1: a - ketta peegeldamisega, b - ketta dubleerimisega**

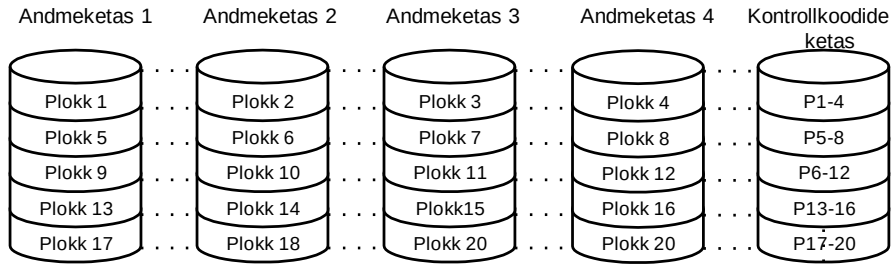
RAID 1 ribasalvestusega variant on hübriidne RAID 1 (Joonis 105), mida on nimetatud ka RAID 10-ks, st RAID 1 ja RAID 0 kombinatsiooniks. Ta võimaldab ketaste peegeldamist ka paaritu ketaste arvuga ning jõudlus on suurem.




**Joonis 105. Hübriidne RAID 1**

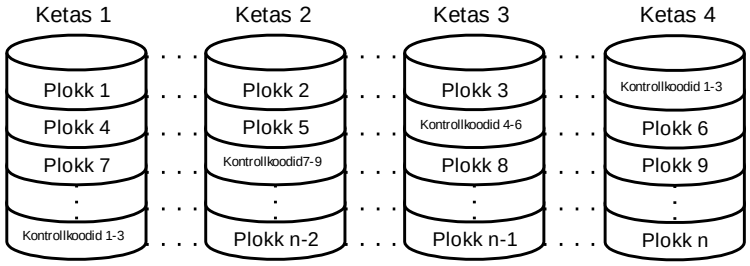
RAID 2 puhul salvestatakse veaparandusandmed eraldi kettale. Kuna enamik kettadraive paigutab tänapäeval need andmed igasse sektorisse, ei ole sellel variandil mingeid eeliseid RAID 3 kõrval.

RAID 3 (Joonis 106) puhul salvestatakse andmed ribadena kõigile ketastele, vigaste andmete ennistuseks vajalikud paarsusandmed aga kirjutatakse eraldi kettale. RAID 3 nõuab vähemalt kaht andmeketast, paremini aga töötab nelja andmekettaga. RAID 3 on ökonoomne varundusviis, mis sobib intensiivse andmeteisaldusega rakendustele (pilditöötlus, dokumendiserverid, video redigeerimine).




**Joonis 106. RAID 3**

RAID 4 sarnaneb RAID 3-le, erinedes andmeüksustega opereerimise poolest. Ribad on suuremad ja lugemisoperatsioonid võivad kattuda. Kirjutused ei saa kattuda, sest nad peavad kõik värskendama paarsusandmeid samal kettal. Kettad ei pöörle sünkroonselt.




**Joonis 107. RAID 5**

RAID 5 (Joonis 107) puhul ei kirjutata paarsusandmeid eraldi kettale, vaid hajutatakse salvestusribades üle kõigi ketaste, seetõttu võivad kattuda ka kirjutusoperatsioonid. Ta nõuab vähemalt kolme ketast. Jõudluselt on ta parem kui RAID 3 või 4.

RAID 6 puhul kasutatakse kaht paarsusandmestikku.

RAID tõstab küll käideldavust selles mõttes, et mingi ketta väljalangemisel jätkub töö katkestuseta, kuid seejuures langeb jõudlus – halvimal juhul isegi kümnekordselt. RAID-süsteemi juhtimise täiustamisega on juhtivail RAID-toodete firmadel õnnestunud tagada avarii puhuks jõudluse säilimine 80-90% ulatuses.

## 16.4 Toite varundamine

### 16.4.1 Varundusmeetodid

Aparatuuri toite varundamise meetodite valimine sõltub nõutavast käideldavusastmest. Varutoite tagamise vahendid on üsna kulukad, tuleb käideldavusnõuded riskianalüüsiga võimalikult täpselt määrata.

***Nõrkade*** käideldavusnõuete korral on soovitatav kasutada sobivalt valitud puhvertoiteallikat (vt 16.4.2) vähemalt kohtvõrgu serveri toiteks. Lisaks kaitsele lühiajaliste toitekatkestuste eest pakub selline allikas oma filtreerimisvõime tõttu ka paremat toite kvaliteeti ning kaitseb aparatuuri võimalikest pingetõugetest tulenevate kahjustuste eest..

***Keskmiste*** käideldavusnõuete korral on puhvertoiteallikas oluliste süsteemide puhul vältimatu.

***Kõrgete*** käideldavusnõuete korral varundatakse toitesüsteemi fiider teisest, sõltumatust alajaamast tuleva fiidriga, millele toimub automaatne ümberlülitus põhifiidri toitekatkestuse korral.

***Kriitiliste*** käideldavusnõuete puhul (nt reaalajasüsteemides, mille väljalangemine ohustab inimesi) kasutatakse varugeneraatorit; sõltuvalt konkreetsetest nõuetest võib see olla pidevalt käigus või automaatselt käivitatav.

Puhvertoiteallikaid kasutatakse lisaks muule meetodile ka kõrgete ja kriitiliste käideldavusnõuete korral.

### 16.4.2 Puhvertoiteallikad

Puhvertoiteallikas ehk ups (UPS, *uninterruptible power supply*) on akupatareil põhinev varutoiteseade, mis võimaldab hoida infotehnilist aparatuuri käigus lühiajaliste voolukatkestuste ajal. Turustatavate puhvertoiteallikate turvaomadused sõltuvad tugevalt nende struktuurist (vt Joonis 108) ja kasutusviisist.

Toiteallika nn "täielikku" konfiguratsiooni (Joonis 108, a) kasutatakse suuremate võimsuste korral (üldiselt üle 10 kVA). Ta koosneb võrgutoiteahelast AB, mis sisaldab toitepinge kvaliteeti tõstvaid kaitse-, tasandus- ja filtreerimislülitusi, ning akutoiteahelast AC, mille moodustavad akulaadur, akupatarei ja vaheldi. Seadmel on kaks tööviisi, sõltuvalt sellest, kumba haru kasutatakse pidevaks toiteks.

***Otsetoite*** (*standby mode*, inglise terminid lähtuvad akuharu kasutusviisist) puhul antakse normaaljuhul toide haru AB kaudu ning võrgutoite katkemisel toimub ümberlülitus harule AC.

***Kaudtoite*** *(on-line mode*) puhul antakse toide alati aku kaudu ning ümberlülitus sooritatakse haru AC tõrke korral. Kaudtoite korral on toite kvaliteet parem, kuid võimsuskadu on 20–30% (otsetoite puhul 1–2%) ning sellega kaasnevad soojaeraldised lühendavad haru AC komponentide eluiga; aku tööiga lüheneb tunduvalt. Allika kogu eluea jooksul ületavad kulud lisakadudele seadme maksumuse.

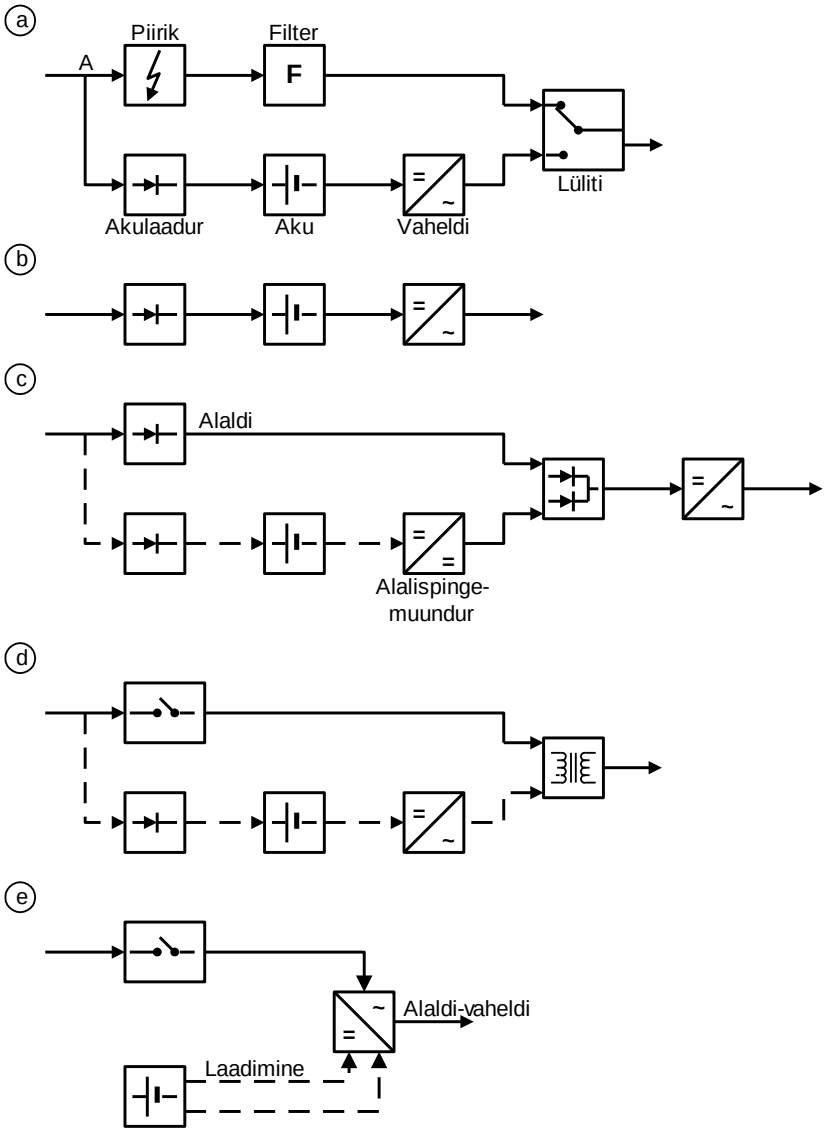
***Rööpharuta kaudtoitelist*** allikat (*on-line UPS*, Joonis 108, b) kasutatakse ta lihtsuse ja odavuse tõttu lauaarvutite toiteks. Eelisteks on toite kvaliteet ning siirdetõugete puudumine võrgutoite katkemisel, puudused on ülalnimetatud toitekaod ning automaatse varuahela puudumine rikete puhul.

***Hübriidallikas*** (Joonis 108, c) meenutab konfiguratsioonilt ülalkirjeldatud "täielikku" allikat, kuid normaalolekus antakse toide pidevalt mõlema haru kaudu. Seadme nõrk koht on vaheldi, mille tõrke korral langevad välja mõlemad harud.

***Otsetoiteline ferrostabilisaatoriga*** allikas (Joonis 108, d) sarnaneb skeemile (a) otsetoiteolekus, ainult harude ümberlülitit asendab ferroresonantstrafo, millel on mõningane stabiliseerimisvõime. Selline trafo isoleerib koormuspoole võrguhäirete eest mitte halvemini kui mistahes filter, kuid tekitab ise

pingemoonutusi ja siirdeefekte, mis võivad olla toitevõrgu omadest tugevamadki. Resonantstrafo kaod, seega ka soojaeraldused on olemuslikult suured.

***Interaktiivse topoloogiaga*** allikas (Joonis 108, e) põhineb kahesuunalisel muunduril, mis toimib nii alaldi (akulaaduri) kui ka vaheldina. Muundur toimib ühtlasi stabilisaatorina, seetõttu sobib selline skeem võrgutoite madala kvaliteedi korral. Muunduri ehitus on selline, et muunduri rikke puhul jääb püsima otsetoide, nii et skeem vastab omadustelt kahe rööpharuga skeemile, milles puudub mõlemale harule ühine nõrk koht. Skeem on ratsionaalne ja tõhus.




**Joonis 108. Puhvertoiteallikate levinumad konfiguratsioonid**

# 17 FÜÜSILISED TURVAMEETMED

## 17.1 Hoone asukoht

Hoone asukoha valimisel tuleb arvestada loodusliku, sotsiaalse ja tehnilise keskkonna vastavust objekti turvaklassile.

***Loodusteguritest*** tuleb arvestada peamiselt tulva- ja liigveega; vajaduse korral tuleb ehitada kuivendussüsteemid. Kõrgendatud turvanõuete korral võivad osutuda oluliseks (muuhulgas visuaalse, akustilise ja kiirgusluure seisukohalt) ka reljeef, ümbritsev taimestik jms.

***Sotsiaalset*** keskkonda tuleb hinnata eelkõige ümbruse kriminogeensuse seisukohalt, kuid arvestada ka lähinaabruses paiknevatest hoonetest lähtuva võimaliku muu ebasoodsa toimega.

***Tehnilist*** keskkonda tuleb hinnata ohutegurite (kiirgus, keemiline saaste, tuleoht, tolm, liiklusavariid jne) ning kommunikatsioonide (juurdepääsu hõlpsus pääste- ja valveteenistustele ning politseile, toite- ja telefoniliine varundada võimaldav alternatiivsete alajaamade lähedus, piisava tuletõrjevee allika olemasolu jne) seisukohalt. Soovitatav on juurdepääs territooriumile vähemalt kahest eri suunast, nii et territooriumile pääseks sõltumatult liiklusoludest, teede remondist ja muudest ootamatutest takistavatest asjaoludest.

## 17.2 Territooriumi turve

### 17.2.1 Planeering

***Teed***. Majandus- ja hooldeliiklus on soovitatav eraldada personali ja külastajate põhiliiklusest, allutades ta pääsu reguleerimisele.

***Haljastus*** tuleb valida nii, et oleks välistatud võimalike sissetungijate peitumine. Vältida tuleks järgmisi haljastuselemente:

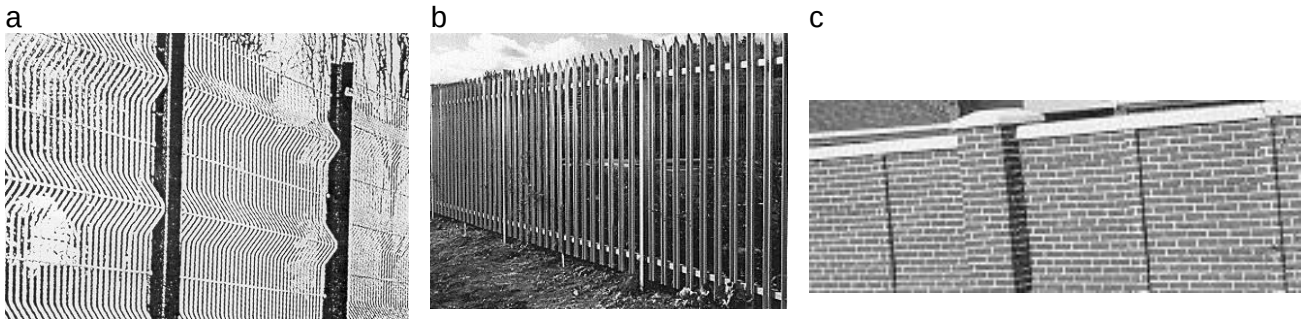
- tihe põõsastik hoone või tara vahetus läheduses (vähem kui 3 m kaugusel),
- suured 0,6–1,8 m kõrguste põõsaste kobarad,
- kõrged igihaljad puud, mille alumised oksad on vähem kui 1,5 m kõrgusel maapinnast.

***Parklad*** tuleb paigutada väljapoole turbeperimeetrit. Soovitatav on eraldada külastajate parkla personali omast. Parklate turve sõltub objekti nõutavast turvatasemest. Kõrgete turvanõuete puhul võivad objekti parklad maksimaaljuhul jaguneda järgmisse nelja klassi.

- Turbeta. Külastajate parkla asub külastajate sissekäigu lähedal, sissepääs ilma tõkkepuuta.
- Osalise turbega. Asutust teenindavate organisatsioonide, lepingupartnerite vms sõidukitele määratud tarastatud ala, mis on normaalsel tööajal avatud; sissesõit võib toimuda turbeta parkla kaudu.
- Turbega. Personalile määratud, ööpäevaringse turbega ala, ümbritsetud turvataraga, sissepääs kaugpuldi või kaardiga.
- Kõrgturbega. Piiratud personalikategooriale. Sissepääsu määrab sõiduk. On varustatud turbesüsteemide ja seirega.

### 17.2.2 Perimeetri turve

Objekti perimeeter ümbritsetakse taraga, mille konstruktsioon sõltub turvanõuetest. Kõrgendatud turvanõuete puhul kasutatakse murdmis- ja ronimiskindlaid monteeritavatest teras- või betoonelementidest, demonteerimist võimaldavate ühendusdetailideta tarasid (Joonis 109).




**Joonis 109. Monteeritavad tarad: a - terasvarbelementidest, b - terasprofiilelementidest, c - (telliseimitatsiooniga) raudbetoonelementidest**

Väravad ja tõkkepuud võivad olla volitustõendiga automaatjuhitavad või pääslast distantsjuhitavad. Kõrgendatud turvanõuete korral võib tavalise tõkkepuu asemel kasutada automaat- või distantsjuhitavat teepinna alla pöörduvat sildbarjääri (Joonis 110), mis välistab jõuga läbisõidu.

**Joonis 110. Läbisõitu tõkestav sildbarjäär**

### 17.2.3 Seire

Turvaperimeetri sees asuv territoorium ja turvaparklad peavad olema visuaalseks seireks piisavalt valgustatud.

Tara varustatakse valvesignalisatsiooni anduritega. Metalltara puhul sobivad vibratsiooniandurid, kõigil taratüüpidel saab rakendada infrapuna-barjäärandureid.

Kõrgendatud turvanõuete korral paigaldatakse valvekaamerad.

## 17.3 Hoone konstruktsioon

Välisseinte pinnal tuleb vältida ronimist soodustavaid elemente. Välissein peab olema murdmiskindel, kõrgendatud turvanõuete korral kuuli- ja plahvatuskindel. Perimeetri poole avatud välisseinal ega selle sees ei tohi olla kommunikatsioone (kaableid, torustikke), ventilatsiooni- jms seadmeid ega tehnilisi avasid. Tehnilised avad tuleb paigutada kaitstud tsooni (katus, siseõu) ning sulgeda püsivalt paigaldatud või lukustatavate ristvõredega; võre varbade läbimõõt peab olema vähemalt 15 mm (topelttangidega on hõlpsalt lõigatav kuni 10 mm varb) ja samm mitte üle 120 mm. Tehniliste kanalite ristlõige tuleb valida nii, et ta välistaks sisenemise kanali kaudu.

Kõrgete turvanõuete korral paigutatakse kiirgusluure ja raadiomikrofonide tõkestamiseks välisseintesse ja aknaavadesse asjakohastele sagedustele arvestatud varjestus.

Kõige tundlikumate ruumide kohal on soovitatav akendeta välissein. Alternatiivina võib kasutada kuulikindlaid aknaid. Keskmiste turvanõuete korral kasutatakse peegeldavaid (termoklaasist või peegelkilega kaetud) aknaid. Süvistamata aknad seina välispinna tasandil takistavad ronimist ja lõhkekehade paigutamist aknaavasse; süvistatud akna muudab turvalisemaks aknaava põhjapinna tugev allakalle.

Sissepääse peab olema minimaalne vajalik arv. Eraldi majandussissepääs peaks olema kaubavedudeks, hooldetegevuseks jne. Soovitatav on eraldada külastajate sissepääs personali omast. avariiväljapääsude arv ja paigutus peab vastama tuleohutusnõuetele.

Tuleohutuselt peab hoone vastama Eesti projekteerimisnormile EPN10.1.

## 17.4 Hoone tsoneerimine

Turbe diferentseerimine tähendab ressursside säästu, seetõttu tuleks ruumid liigitada nõutava turvataseme järgi ning paigutada kõrgemate turvanõuetega ruumid avalikumatest tsoonidest eemale. Pääsu reguleerimise hõlbustamiseks tuleb arvutuskeskuse või serveriruumi ümber luua puhvertsoon.

Külastajate tsoon ja hooldetööde tsoon tuleb eraldada infotehnoloogiatsoonidest. Infotehniliste süsteemide põhikomponendid tuleb paigutada välisseintest eemale.

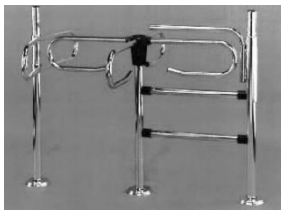
Varuandmehoidlad tuleb paigutada põhihoidlatest piisavalt eemale, nii et ükski riskianalüüsiga väljaselgitatud tõsine oht ei saaks neile toimida korraga.

Hoone sees tuleb luua sisemised turvaperimeetrid, mis välistavad sissepääsu suuremale osale personalist. Siseperimeetrid peavad hõlmama järgmisi objekte:

- serveri- ja suurarvutiruumid,
- võrgu- ja sideaparatuurikeskused,
- eriti tundliku informatsiooni töötluskohad,
- süsteemiprogrammeerijate, andmebaasiülemate ja võrguülemate tööpaigad,
- turbehalduse tööpaik (peab olema eraldatud ka ülalloetletud aladest).

Siseperimeetrite peamine kaitsevahend on ustele ja liftidesse paigaldatud elekronlukkudel põhinev pääsu reguleerimise süsteem.

Pääslas tuleb barjääride ja väravate abil tagada füüsiline pääsu reguleerimine. Pääslaväravad (Joonis 111) võivad sõltuvalt turvanõuetest olla lihtsalt käsitsi pööratavad, distantsjuhtimisega või volitustõendil põhineva automaatjuhtimisega ning olla ühendatud sisenejate või väljujate loenduriga. Elektriliselt lukustatavad väravad võivad olla käsitsi avatavad või varustatud ajamiga.


a

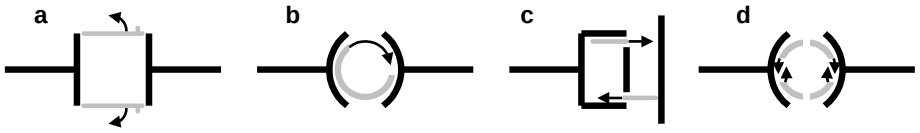

b


c


d

**Joonis 111. Pääslaväravate näiteid: a - lihtne vöökõrgune lukustuseta käsivärav, b ja c - metrootüüpi vöökõrgused ajamiga automaatväravad, d - uksekõrgune lukustuseta, automaat- või distantslukustusega käsivärav**

Kõrgete turvanõuete korral võidakse pääsu eriti tundlikesse ruumidesse reguleerida pääsulüüsidega, mis "doseerivad" isikuid ühe- või kahekaupa. Pääsumehhanism võib asuda lüüsis ning volitamatul pääsukatsel võib lüüs mõlemalt poolt sulguda ja anda alarmi. Lüüsil võib olla ka kumbagi ukse jaoks eraldi pääsumehhanism; selline variant tagab sisenejate täpse registreerimise: välistatud on näiteks väär registreerimine selle tõttu, et volitatud isik autentis end, kuid loobus sisenemast.




**Joonis 112. Pääsulüüside näiteid: a - tavaliste ustega boks, b - ajamiga ümarlüüs, c - liugustega boks, d - liugustega ümarlüüs**

Kõrgete turvanõuete puhul peab pääslavalvur olema kaitstud kuulikindla klaasiga; ka valveboksi ülejäänud osa peab olema kuulikindel.

Hoone sissepääsude ja tundlike tsoonide sissepääsude elektronlukud ühendatakse seiresüsteemi; kõrgemate turvanõuete korral paigaldatakse nende sissepääsude juurde valvekaamerad. Niisuguste sissepääsude uksed peavad olema murdmis- ja tulekindlad. Pääsu reguleerimise süsteemi kuuluvad uksed ja lukud peavad olema isesulguvad.

## 17.5 Tehniline infrastruktuur

Lisaks infrastruktuuri puudutavatele ehitus- jm. eeskirjadele tuleb arvestada järgmisi turvanõudeid.

Infrastruktuuri seadmestik (peakilp, võimalik diiselgeneraator, soojussõlm, keskkonditsioneerid jms) peab asuma muudest ruumidest eraldatud ruumides. Peakilp ei tohi asuda keldrikorrusel.

Kaabeldus ei tohi olla nähtaval, toitesüsteemi struktuur ei tohi olla läbinähtav, jaotuskilbid peavad olema lukustatud, jaotusseadme elemendid ja seinapesad peavad olema märgistatud.

Infotehnoloogiaruumide toide peab olema eraldatud muude ruumide toitest. Infotehnika toide peab olema eraldatud ruumide muust toitest. Kõrgete turvanõuete korral peab olema tagatud tundlike süsteemide ning oluliste turvamehhanismide (sh lukusüsteemide) varutoide. Varuväljapääsude juures peavad paiknema toite avariiväljalülitid.

Vahetult infotehnoloogiaruumide kohal ülakorrustel paiknevates ruumides tuleb vältida vee- ja kanalisatsioonitorustike paiknemist. Kui see ei õnnestu, tuleb olulistesse infotehnikaruumidesse paigaldada veeandurid ja ühendada need seiresüsteemiga.

Tuletõrjesprinklerid tuleb infotehnikaruumides paigaldada nn kuiva süsteemina, st torustikes tohib vesi olla ainult kustutusprotsessi kestel. Käsikustutite ja hüdrantide arv, tüüp, paigutus ja tähistus peavad vastama tuletõrje eeskirjadele, mis puudutavad vastava tuleohutusklassiga ruume.

Infotehnoloogia seisukohalt olulised infrastruktuuri seadmed (toiteseadmed, konditsioneerid jne) tuleb ühendada seiresüsteemiga.

## 17.6 Infotehnoloogilised ruumid

Füüsilise turbe nõuded sõltuvad töödeldava informatsiooni tundlikkusest ning aparatuuri materiaalsest ja funktsionaalsest väärtusest. Üldtunnustatud normid puuduvad, nõuded või soovitused on tavaliselt fikseeritud ametkondlikes või rahvuslikes turvastandardites või etalonmetoodikates. Mitmed neist ei täpsusta ruumide funktsioone, piirdudes üldnimetusega "arvutuskeskus" või "arvutiruum", mis enamasti tähistab suurarvuti- või serveriruumi, aga ka ruumi, milles asub tundlik tööjaam.

Alljärgnevalt on orienteerumist hõlbustava näitena esitatud Austraalia turvastandardi ACSI 33 nõuded (Tabel 42 ja selle juurde kuuluvad määratlused).

**Tabel 42. Arvutiruumi turbenormid (ACSI 33)**

| Andmete tundlikkus | Hoone turvanorm | | |
|---|---|---|---|
| | Turvaline | Osaliselt turvaline | Sissetungipüsiv |
| Salajased/Tugev kaitse | CR1 | CR1 | CR1* |
| Konfidentsiaalsed/Kaitse | CR2 | CR2*/CR1 | CR1 |
| Kasutuspiiranguga | CR3 | CR3 | CR3 |

* - Aparatuur peab olema paigaldatud seadmeriiulitele või kappidesse.

***CR1***

Kõik ala piirded (aknad, uksed, seinad, põrand, lagi) peavad olema paigaldatud nii, et neid ei saaks kõrvaldada ega asendada ulatuses, mis võimaldaks sissepääsu seaduslikku võtit kasutamata võoi vähemalt üht nimetatud elementidest nähtavalt ja püsivalt kahjustamata. Kahe lae või lae ja katuse vaheline ruum tuleb varustada valvesignalisatsiooniga.

Uks peab olema tühemikuta. Uks peab olema väljapool normaalset tööaega lukustatud; kui ala on pikemat aega mehitamata, tuleb kasutada turvalukku. Tööajal võib pääsu reguleerimiseks kasutada piisava turbega mehaanilist koodlukku.

Aknad peavad olema mitteavatavat tüüpi või ruumist lahkumisel lukustatavad piisavalt turvaliste aknalukkudega.

Ruum peab olema varustatud piisavalt turvalise valvesignalisatsiooniga.

***CR2.*** Nagu CR1, kuid signalisatsiooni nõudeta.

***CR3***. Ruum peab vastama normaalse bürooruumi ehitusnormidele. Uksi ja aknaid peab saama ruumist lahkumise puhuks lukustada.

ACSI 33 soovitab tundlikku informatsiooni töötlevates arvutites kasutada ird-kõvakettaid, mida väljaspool tööaega hoitakse seifis. Standard esitab üldnõuded tundlikku informatsiooni töötleva tööjaama füüsilisele turbele (Tabel 43), sõltuvalt andmete tundlikkusklassist ja hoone turvanormidest.

**Tabel 43. Tööjaama turbenormid (ACSI 33)**

| Andmete tundlikkus | Ala turvanorm | | |
|---|---|---|---|
| | Turvaline | Osaliselt turvaline | Sissetungipüsiv |
| Täiesti salajased | WS1 | WS1 | Pole võimalik |
| Salajased/Tugev kaitse | WS1 | WS1 | WS1 |
| Konfidentsiaalsed/Kaitse | WS2 | WS2 | WS1 |
| Kasutuspiiranguga | WS3 | WS3 | WS3 |

***WS1***. Tööjaam peab olema kettata või ird-kõvakettaga, mida väljaspool tööaega säilitatakse turvaliselt.

***WS2***. Kui tööjaamal on kinnisketas, tuleb täita järgmised nõuded:

- tööjaama kest tuleb kinnitada turvaliste lukkude ja/või plommide abil;
- kasutaja autentimine tuleb sooritada püsimälus asuva parooli abil või alternatiivsete buutimisvahenditega; autentimisest möödahiilimine peab olema võimatu tööjaama kesta avamata;
- tundlikke andmeid tuleb talletada arvutiruumis asuvas serveris, mitte kasutaja tööjaamas.

***WS3***. Erinõuded puuduvad. Kehtib keeld talletada tundlikke andmeid tööjaamas.

Standard näeb ette tundliku informatsiooni säilituse turvalistes panipaikades (vt ka 17.7), kuid jätab võimaluse asendada mõningaid füüsilise turbe meetmeid andmete krüpteerimisega.

Ülaltoodu puudutab peamiselt rünnete tõrje meetmeid. Lisaks neile tuleb aga silmas pidada ka stiihilisi ohte neutraliseerivaid abinõusid, sh

- aparatuuri maandamist,
- antistaatikmatte,
- varutoiteallikaid,
- valgustust,
- ventilatsiooni,
- sisustuse vastavust ergonoomianõuetele.

## 17.7 Turvalised panipaigad ja turvaruumid

Infotehnoloogiliste varade füüsiliseks kaitseks saab kasutada mitmesuguseid spetsialiseeritud või üldotstarbelisi monoliitseid või tüüpmoodulitest koostatavaid turvalisi mahuteid – kappe, kambreid, terveid ruume. Kaitse suunitluse järgi jagunevad nad kahte suurde rühma:

- andmekapid (*data cabinet*), andmekambrid (*data container*) ja andmeruumid (*data rooms*) on mõeldud dokumentide, andmekandjate ja ka aparatuuri kaitseks peamiselt keskkonnamõjurite ja kahjutule eest, kuid pakuvad teatud kaitset ka sissemurdmise eest;
- seifid (*safe*) ja soomuskambrid (*strongroom*) on kaitstavate varade mõttes valdavalt üldotstarbelised ning mõeldud eeskätt kaitseks sissemurdmise eest. Üha rohkem on hakanud ilmuma ka andmekandjate, serverite, tööjaamade ja võrguaparatuuri jaoks spetsialiseeritud teostusi.

### 17.7.1 Andmekapid

Andmekappide (Joonis 113) kaitseklassid tulekindluse järgi määratleb eurostandard EN 1047-1. Standard puudutab kappe, mis on määratud andmekandjate säilituseks. Selle standardi tähenduses on andmekandjad informatsiooni sisaldavad materjalid, nt paberdokumendid, film, disketid, kassetid, optilised kettad, video- ja helikassetid.

**Joonis 113. Andmekappide näiteid**

EN 1047-1 kaitseklassid (Tabel 44) määratakse tulekindlusteimi ja tulekindlus-löökteimi alusel. Tulekindlusteimis hoitakse kappe vastavalt 60 või 120 minutit leekides umbes 1000ţ C juures ning mõõdetakse seejärel temperatuuri ja niiskust kapis. Tulekindlus-löökteimis hoitakse kappi sõltuvalt klassist 22,5 või 45 minutit temperatuuril 1090ţ C, seejärel kukutatakse ta 9,15 m kõrguselt munakividele ning hoitakse veel 22,5 või 45 minutit temperatuuril 840ţ C; sisetemperatuur ja -niiskus peavad pärast seda vastama tabelis olevatele väärtustele. Peale selle mõõdetakse ka kapi seina ja ukse paksusemuutust.

**Tabel 44. Andmekappide kaitseklassid**

| 60 minutit | 120 minutit | Maks. temperatuuri tõus* | Maks. õhuniiskus |
|---|---|---|---|
| S 60 P | S 120 P | 150ţ C | Nõudeid pole |
| S 60 D | S 120 D | 50ţ C | 85% |
| S 60 DIS | S 120 DIS | 30ţ C | 85% |

* Algtemperatuurilt 20...22ţ C

Kaitseklassi tähises tähistab arv minutites väljendatud tulekindlust ülalkirjeldatud teimide mõttes. Arvule järgneva tähtlühendi sisu on selline:

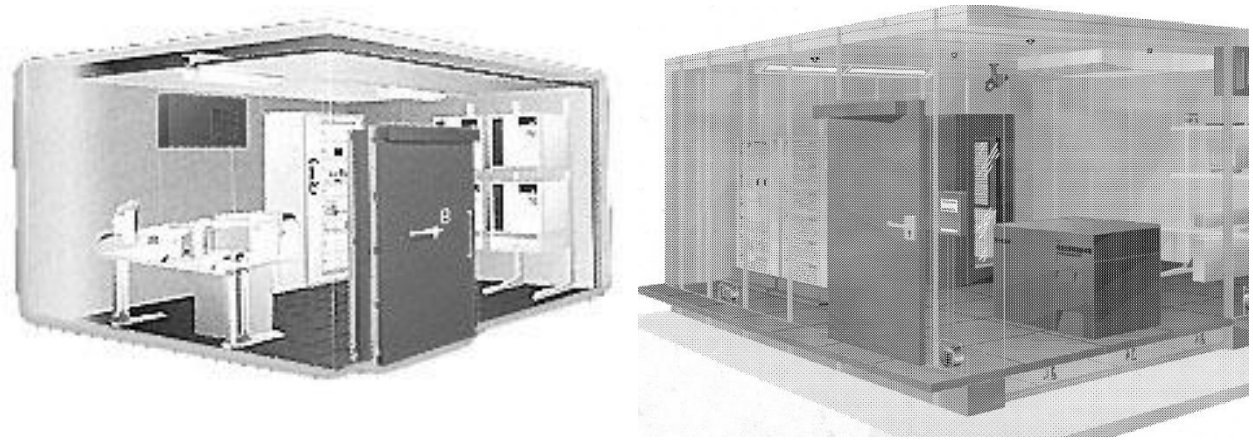
P – kuumustundlikud paberdokumendid, mis ei kaota informatsiooni kuumutamisel kuni 170ţ C;

D – kuumus- ja niiskustundlikud säilikud, nt magnetkandjad ning paberpõhimikud, mis ei kaota informatsiooni kuumutamisel kuni 70ţ C:
DIS – kuumus- ja niiskustundlikud säilikud, mis ei kaota informatsiooni kuumutamisel kuni 50ţ C, nt disketid.

### 17.7.2 Andmeruumid ja -kambrid

Andmekandjate (vt 17.7.1) ja riistvara kaitseks määratud andmeruumide (Joonis 114) ja andmekambrite kaitseklassid tulekindluse järgi määratleb eurostandard prEN 1047-2.

Standard määratleb kaks andmeruumide (*data room*) konstruktsioonitüüpi (A ja B); teimid sooritatakse näidisega, mille mõõtmed on 4×3×2,8 m. Andmekamber (*data container*) on standardi kohaselt tarind, mis paigaldatakse põrandale ja mille sisemine põhjapind on mitte üle 3 $m^2$.



**Joonis 114. Moodulitest koostatava andmeruumi näiteid**

Kaitseklasse on sisuliselt üksainus (Tabel 45). Teimid on analoogilised kappide omadele (vt 17.7.1), ainult löökteim sooritatakse 200-kilose teraskuuli kukutamisega 1,5 m kõrguselt ruumi või kambri nõrgimale kohale.

**Tabel 45. Andmeruumide ja -kambrite kaitseklassi nõuded**

| Kaitseklass | Maks. temperatuuri tõus* | Maks. õhuniiskus |
|---|---|---|
| R60D | 50ţ C | 85% |
| C60D | 50ţ C | 85% |

* Algtemperatuurilt 20...22ţ C

Kaitseklassi tähistusviis on analoogiline andmekappide tähistusele:

R – andmeruumid,
C – andmekambrid,
60 – viitab 60-minutisele tulekindlusteimi ajale,
D – tähistab kaitstavaid süsteemiosi ja andmekandjaid; hõlmab praktiliselt kõiki andmekandjaid peale nende, mis kaotavad informatsiooni temperatuuridel kuni 70ţ C.

### 17.7.3 Seifid ja soomuskambrid

Eurostandard EN 1143-1 määratleb 11 seifide (Tabel 46) ning 14 soomuskambrite ja -uste vastupidavusklassi. Seifiks loeb standard soomustarindit, millel on suletud olekus vähemalt ühe sisekülje pikkus mitte üle meetri. Seifid võivad olla autonoomsed või sisseehitatavad.

**Joonis 115. Aparatuuriseifi näide**

**Tabel 46. Seifide klassid**

| **Klass** | **Instrumentründe teim** Vastupidavus | | **Ankurdus-tugevus**[1)] | **Lukud** | | **Lõhkamiskindlus** (fakultatiivne) |
|---|---|---|---|---|---|---|
| | osaliseks sissepääsuks, RU | täielikuks sissepääsuks, RU | Nõutav jõud, kN | Arv | Klass EN 1300 järgi | Lõhkamisjärgne vastupidavus, RU |
| 0 | 30 | 30 | 50 | 1 | A | [2)] |
| I | 30 | 50 | 50 | 1 | A | [2)] |
| II | 50 | 80 | 50 | 1 | A | 4 |
| III | 80 | 120 | 50 | 1 | B | 6 |
| IV | 120 | 180 | 100 | 2 | B | 9 |
| V | 180 | 270 | 100 | 2 | B | 14 |
| VI | 270 | 400 | 100 | 2 | C | 20 |
| VII | 400 | 600 | 100 | 2 | C | 30 |
| VIII | 550 | 825 | 100 | 2 | C | 41 |
| IX | 700 | 1050 | 100 | 2 | C | 53 |
| X | 900 | 1350 | 100 | 2 | C | 68 |

1) Ainult autonoomsete alla 1000 kg massiga seifide puhul.
2) Klasside 0 ja I puhul ei ole võimalik.

Autonoomsete seifide ankurdustugevuse teimi eesmärk on veenduda selles, et seifi tugevus on piisav ta kinnitamiseks põranda või seina külge, nii et teda ei ole liiga kerge sealt lahti rebida.

Instrumentründe teim sooritatakse 14 tüüpi instrumentidega,sh elektriliste ja termilistega. Igas tüübiklassis jagunevad instrumendid oma jõudluse järgi viide hindeklassi (5/7,5/10/15/35 RU/min); igale instrumenditüübile on kinnistatud teatavad kaalud, nn baasväärtused (vahemikust 0...300).
RU on dimensioonitu vastupidavusühik (*resistance unit*).

Iga instrumentteimi puhul arvutatakse vastupidavus valemiga

$$V = (\sum t \times c) + \sum B_i,$$

kus
$\sum t$ – kõigi instrumentide kasutamise aegade summa minutites,
$c$ – kõrgeim kasutatud instrumentide jõudlustegur,
$\sum B_i$ – kõigi kasutatud instrumentide baasväärtuste summa.

Lõhkamisteim sooritatakse vastavalt tabelile 73 standardse pentaerütritool-tetranitraadiga (PETN, tihedus 1500 kg/m$^3$, erienergia 5000 J/g, detonatsioonikiirus 7000 m/s).

**Tabel 47. Lõhkelaengu mass lõhkamisteimis**

| Klass | Lõhkeaine mass, g | |
|---|---|---|
| | Seifid | Soomuskambrid ja -uksed |
| II, III, IV | 70 | 70 |
| V, VI, VII | 100 | 125 |
| VIII, IX, X | 200 | 250 |
| XI, XII, XIII | - | 375 |

Lõhkamisjärgne vastupidavus määratakse ülalkirjeldatud instrumentteimiga, rakendades ülaltoodud valemit.

## 17.8 Lukud

### 17.8.1 Turvalukud

Turvalukkude (*high security locks*) klassid määratleb eurostandardi kavand prENV 1300 (Tabel 48).

**Tabel 48. Nõuded turvalukkudele**

| Luku klass | Luku tüüp | Avamiskoodide säilituskirjete minimaalarv | Koodide minimaalarv | | Proovimiste maksimaalarv tunnis | | Minimaalne muukimis-kindlus M, | Minimaalne lõhkumis-kindlus D, |
|---|---|---|---|---|---|---|---|---|
| | | | Esemel. | Mnem. | Suvalised | Mnem. | RU | RU |
| A | Elektrooniline | - | 25000 | 80000 | 300 | | 30 | 80 |
| | Mehaaniline | - | 25000 | 80000 | - | | 30 | 80 |
| B | Elektrooniline | 10 | 100000 | 100000 | 100 | | 60 | 135 |
| | Mehaaniline | - | 100000 | 100000 | - | | 60 | 135 |
| C | Elektrooniline | 50 | 1000000 | 1000000 | 30 | | 100 | 250 |
| | Mehaaniline | - | 1000000 | 1000000 | - | | 100 | 250 |
| D | Elektrooniline | 500 | 3000000 | 3000000 | 10 | | 620 | 500 |
| | Mehaaniline | - | 3000000 | 3000000 | | 10[1)] | 620 | 500 |

1) Välja arvatud võtmega lukud

"Säilituskirjete minimaalarv" tabelis märgib nõuet elektronlukkudele, millel on mitu avamiskoodi; lukk peab need koodid avamisel registreerima ning säilitama vähemalt aasta (ka toitekatkestuse korral). Mehaanilistel turvalukkudel seevastu ei tohi olla mitut erinevat võtit.

Muukimiskindlus *M* määratakse mitmesuguseid instrumente rakendava teimiga (vrd seifide instrumentründe teim, vt 17.7.3) ning leitakse valemiga

$$M = t + B,$$

kus
*t* - teimitava näidise lahtimuukimisele kulunud aeg minutites,
*B* - kõrgeim kasutatud instrumendi jõudluse baasväärtus (0, 10 või 20) vastavalt instrumentide tabelile; instrumendid jagunevad kahte klassi – tavalisteks ja eriinstrumentideks.

Lõhkumiskindlus *D* määratakse teimiga, mis vastab seifide instrumentründe teimile (vt 17.7.3) ning leitakse valemiga

$$D = 5t + \sum B_i + B,$$

kus
*t* – teimitava näidise lahtimurdmisele kulunud aeg minutites,
$\sum B_i$ – kõigi kasutatud EN 1143-1 A-klassi instrumentide (vt 17.7.3) baasväärtuste summa,
*B* – kõrgeim kasutatud instrumendi jõudluse baasväärtus (0, 10 või 20) vastavalt instrumentide tabelile.

Peale selle esitab standard nõudeid mitmesugustele töökindluse näitajatele, luurekindlusele jms. Standard ei normi täiendavaid turvaomadusi, näiteks

- ülemkoodi, mis väldib koodi muutmise ja rööpsete koodide lubamise või keelamise,
- ajakoodi ajasätte blokeerimiseks,
- alarmikomponente või -funktsioone,
- kaugjuhtimiskohustusi,
- vastupidavust happeründele,
- röntgenikindlust,
- lõhkamiskindlust.

### 17.8.2 Lihtlukud

Lihtlukkude turvaomaduste kohta üldtunnustatud standardeid ei ole; kindlustatud objektide puhul tuleb arvestada kindlustusseltside nõudeid.

Konstruktsioonilt jagunevad lihtlukud kahte põhirühma, ketasmuukidega ja hoobmuukidega lukkudeks. Tihvtmuukidega *yale*-lukk ("Vasara lukk") ei ole turvamehhanismina arvestatav ja kuulub pigem sulgurite hulka.

Ka lihtlukkude hulgas leidub suhteliselt heade turvaomadustega lukke. Üha enam kasutatakse kvaliteetsemates lukkudes detailide materjalina volframterast ja muid lõikekindlaid sulameid ning lisatakse pettemuuke, alarmiseadiseid jms.

Saadaval on ka mitmesuguste eriomadustega lukke. Näiteks säästab ressursse ja väldib turvaintsidente lukk, mille komplekti kuulub mitu erinevat võtit; võtme kaotamisel saab luku kombinatsiooni muuta lihtsalt järgmise võtme kasutamisega.

### 17.8.3 Arvutilukud

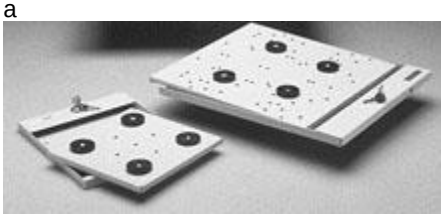
Arvuti ja ta osade lukustamine töölaua külge on tõhus abinõu füüsiliste rünnete peamise liigi, aparatuuri varguste vastu. Süsteemiploki, aga ka monitori või muu suurema välisseadme kinnitamiseks on põhiliselt kolme liiki vahendeid.

***Lukustusplaadid*** (Joonis 116, a) kujutavad endast metallplaatide paari. Üks plaat kinnitatakse kruvide või liimi abil lauaplaadile, teine aga seadme põhja alla. Plaadid kinnitatakse üksteise külge luku või Erivõtit nõudva turvakruvi abil.

***Trosslukk*** on universaalne ja odav vahend, kuid vähem turvaline ning on praktiliselt ainsa abinõuna kasutatav ainult sülearvuti puhul (Joonis 116, d).

***Lukustusümbris*** (Joonis 116,, b-c) on kõige kindlam lahendus, sest ta välistab ka seadme avamise.

a



b

c

d

**lukustatav ümbris, d - sülearvuti trosslukk**

**Joonis 116. Arvutilukke: a - laua külge kinnitatavad lukustusplaadid, b-c: laua külge**

USA turvaintsidentide statistika näitab, et koduarvutite massilisele levikule on hakanud kaasnema üha sagedam komponentide (eriti RAM-kiipide) vargus töökohtade arvutitest. Selle ründe tõkestamiseks on saadaval mitmeid erilukke (Joonis 117). Lisaks lukkudele kasutatakse aparatuuri kaitseks ka lukkudega integreeritud või autonoomseid kohaliku toimega alarmiseadiseid, mh selliseid, mis rakenduvad kahe kokkukuuluva seadme mehaanilisel või elektrilisel lahutamisel üksteisest.

a

b


c

**Joonis 117. Lukke komponentide kaitseks: a - kaablilukk, b - kestalukk, c - kruvi asemele paigaldatav kestalukk**

Viiruste jm ründetarkvara ning volitamatute kopeerimiste tõrjeks mõeldud ***draivilukk*** (Joonis 118) võimaldab arvutit kaitsta ka igasuguse muu volitamatu pääsu eest: disketilt sooritatav buutimine on küll suhteliselt tülikas, kuid mitmes standardis soovitatavate ird-kõvaketastega võrreldes on disketilukk märksa odavam lahendus. Ta kahandab ka arvuti varguse riski: vargal tuleks ju asendada kogu disketidraiv.

**Joonis 118. Draivilukk**

Kui eesmärgiks on mitte arvuti kaitse varguse eest, vaid arvuti volitamatu kasutamise vältimine, saab kasutada mitmesuguseid infotehnilisi lukustussüsteeme, mis põhinevad pordiga ühendatavatel ja võtme, kaardi või biotunnusega juhitavatel autentimisplokkidel.

## 17.9 Andmekandjate saneerimine ja hävitamine

Kui korduvkasutusega andmekandjat (enamasti magnetilist) soovitakse tühjendada konfidentsiaalsest informatsioonist, tuleb ta saneerida, st informatsioon tuleb kandjalt ennistamatult kustutada. Seda ei saa teha lihtsalt failihalduri, utiliidi vms vastava käsuga, mis kustutab halvemal juhul ainult faili nime failijaotustabelist, paremal juhul aga kirjutab senisele informatsioonile peale näiteks nullbitid. Ühekordne ülekirjutus nullidega on küll kustutus, kuid mitte ennistamatu: nullbittide füüsilised esitused magnetväljatugevustena pole ühesugused, vaid sõltuvad nende "all" olevatest bitiväärtustest. Seetõttu tuleb saneerimiseks kasutada eriprotseduure.

Infotehniline saneerimine seisneb kandja paljukordses (nt 100 korda) ülekirjutuses teatava tüüptsükliga; levinud tüüptsükkel on näiteks järgmine: (1) kõigile bitikohtadele nullbitid, (2) kõigile bitikohtadele juhuslikud bitid, (3) kõigile bitikohtadele ühebitid. Meetod ei nõua lisavahendeid peale elementaarse tarkvaramooduli, kuid on aeganõudev ning seetõttu magnetlintide puhul praktiliselt kõlbmatu.

Füüsiline saniteerimine seisneb magnetkandja paigutamises tugevasse vahelduvmagnetvälja. Selleks kasutatakse demagneetureid e degaussereid (*degausser,* joonis 127). Demagneeturi toimeelement koosneb ühest või mitmest tugevast (tüüpiliselt 2...10 kilogaussi) elektromagnetist. Kustutustoime tugevdamiseks võivad magnetid olla pöörlevad ja/või paigutatud nii, et nad toimivad kandjale kahelt poolt.

Sõltuvalt jõudlusest ja kustutustugevusest võivad seadme mõõtmed ulatuda pardli omadest kirjutuslaua suuruseni. Suuremad ja ka mõned lauamudelid on varustatud andmekandjaid üle magneti vedava lintkonveieriga (Joonis 119, c). Lauaaparaatide tüüpiline jõudlus on 100–500 disketti (või mõnikümmend videokassetti) tunnis. Aparaadid võivad olla universaalsed (kustutavad kõiki tavalisi magnetkandjaid) või olla spetsialiseeritud mingile kandjatüübile.

Demagneeturi valimisel tuleb eelkõige lähtuda ta nõutavast magnetvälja tugevusest: see peab olema vähemalt 3–4 korda suurem salvestise väljatugevusest; üldiselt on väljatugevus pöördsõltuvuses salvestustihedusest. Seadmete hinnaskaala algab 150 dollarist, keskmise ja suure jõudlusega seadmed maksavad 1000 kuni 10000 dollarit.


**Joonis 119. Demagneeturid: a – käsisöötega lauaaparaat, b – käsimudel, c – konveieriga lauaaparaat**

Käibelt kõrvaldatavat konfidentsiaalset teavet sisaldavad andmekandjad, mida ei saa saniteerida, hävitatakse põletamise või peenestamise teel. Standardid nõuavad tavaliselt peenestamist osakesteks, mille suurim mõõde on 10 mm või alla selle. Selleks valmistatakse rootorpurusteid (Joonis 120, c), mille tuntuim alaliik on paberdokumentide hävitamiseks mõeldud paberihunt (Joonis 120, a).

Sõltuvalt jõudlusest ulatuvad paberihundid mõõtmetelt prügikastisuurusest kapisuuruseni; suuremad on varustatud mehaanilise sööturiga. Etalonnõuetele vastavad mudelid hakivad paberit kahes ristuvas suunas, nõrgemate turvanõuete korral võib kasutada odavamaid ribastavaid seadmeid. Paberdokumentide massiline hävitamine tekitab suurtes kogustes paberipuru, mida saab tihendada selleks otstarbeks määratud vibropressidega (Joonis 120, b).

a b c

**Joonis 120. Andmekandjate hävitamine: a - keskmise jõudlusega paberihunt, b - paberijäätmete press, c - rootorpurusti ehitus**

Kõvemate andmekandjate hävitamiseks määratud purustid erinevad ainult lõikurite tugevuse ja teostuse tehniliste üksikasjade poolest.

## KASUTATUD ALLIKAID

*Raamatu kirjutamisel kasutatud allikate arv ulatub mitmesajani, seetõttu on alljärgnevasse loetellu võetud ainult valik sisult ja mahult kaalukamaid. Neist võib leida lisateavet täienduseks eespool esitatule.*


**Amoroso, E.** Fundamentals of Computer Security Technology. *Prentice-Hall*, 1994. xxii+404 pp.

**Australian Communications-Electronic Security Instructions 33** (ACSI 33). *Defense Signals Directorate*, 1998. 180 pp.

**Bauer, F.L.** Decrypted Secrets. Springer, 1997. 450 pp.

**BS 7799:1995** Code of Practice for Information Security Management. *BSI*, 1995. 54 pp.

**Buldas, A., Laud, P., Lipmaa, H., Villemson, J.** Time-Stamping with Binary Linking Schemes. - In: Advances in Cryptology - CRYPTO'98. *Springer*, 1998, pp. 486-501.

**Castano, S. , Fugini, M., Martella, G., Samarati, P.** Database security. *Addison-Wesley Publishing Co*, 1995. 456 pp.

**Computer Assurance Guidelines for the Commercial Sector**. *Department of Trade and Industry, Information Security Policy Group*, 1996

**The Computer Security Handbook of DCRT**. *National Institutes of Health*, 1996.

**Computer Security**. CS4601. Monterey, CA: *Naval Postgraduate School*,1995.

**Coppersmith, D., Franklin, M., Jacques Patarin, J., Reiter, M.** Low-Exponent RSA with Related Messages. *EUROCRYPT'9*6, pp. 1-9.

**Cracking DES**. *Electronic Frontier Foundation*, 1998. 270 pp.

**Deavours, C.A.** e.a. (ed.) Selections from Cryptologia. Boston - London: Artech House, 1998. 557 pp.

**DeLaurentis, J.M.** A further weakness in the common modulus protocol for the RSA cryptoalgorithm. *Cryptologia*, vol. 8, 1984, pp. 253-259.

**EVS ISO/IEC TR 13335-3.** Infotehnoloogia. Infoturbe halduse suunised. Osa 3: Infoturbe halduse meetodid. 1998.

**Goldwasser, S., Bellare, M**. Lecture Notes on Cryptography. Cambridge (Mass.), 1997. 194 pp.

**Guide to Threat and Risk Assessment For Information Technology**. Security Information Publication 5. *Royal Canadian Mounted Police*, 1994.

**Hendry, M.** Smart Card Security and Applications. Boston - London: *Artech House*,1997. 287 pp.

**Information Security Breaches Survey 1996**. *NCC*, 1996.

**Information Systems Security Policy**. *Automated Information Security Policy Review Team*, 1995.

**Information Security Service**, 1-3. *Datapro International*, 1997.

**InformationTechnology Security** (ITS) Minimum Baseline Protective Requirements. Draft. *NASA*, 1996.

**ISO/IEC 10181.** Information Technology - Open Systems Interconnection - Security Frameworks for Open Systems. 1-7. *ISO/IEC*, 1996.

**IT-Grundschutzhandbuch 1998**. *Bundesamt für Sicherheit in der Informationstechnik*, 1998.(CD-ROM)

**Knuth, D.E.** The Art of Computer Programming. V. 2. Seminumerical Algorithms. 3rd Edition. *Addison-Wesley Publishing Co*, 1998.762 pp.

**Koblitz, N.** A Course in Number Theory and Cryptography. 2nd ed. *Springer*, 1987. 238 pp.

**Koblitz, N.** Algebraic Aspects of Cryptography. Springer, 1998. 206 pp.

**Menezes, A.J., Oorschot, P.C. v.** Handbook oh Applied Cryptography. *CRC Press*, 1997. 780 pp.

**NATO** Open System Environment. V. 4. Base Standards, v.3. 1996.

**Renesse, R. L. v.** Optical Document Security. Second Edition. Boston - London: *Artech House*, 1998. 508 pp.

**Schneier, B.** Applied Cryptography. 2nd ed.*John Wiley & Sons*, 1996. 760 pp.

**Security** Risk Management Plan for Release B for the ECS Project. EOSDIS Core System Project. 627-CD-002-001. *Hughes Information Technology Systems*, 1996.

**Simmons, G.J.** A 'weak' privacy protocol using RSA cryptoalgorithm. *Cryptologia*, vol. 7, 1983, pp. 180-182.

**Simmons, G.J.** (ed.) Contemporary Cryptology.*IEEE Press*, 1992.640 pp.

**Stinson, D.R.** Cryptography. Theory and Practice. *CRC Press*, 1995. 434 pp.

**Technical** Security Standard for Information Technology. *Royal Canadian Mounted Police*, 1994. 54 pp.

**Tsiounis, Y.S.** Efficient Electronic Cash: New Notions and Techniques. Boston (Mass.): Northeastern University, 1997. xii + 196 pp.

**Vacca, J.R.** Internet Security Secrets. *IDG Books*, 1996. 800 pp.

**Welsh, D.** Codes and Cryptography. Oxford: *Clarendon Press*, 1998. 258 pp.

**Williams, C.P., Clearwater, S.H.** Explorations in Quantum Computing. *Springer*, 1998.310 pp.