IRZ0020 Kodeerimine ja krüpteerimine

Maht:

- AP 3,5: EAP 5, 2 – 0 – 2; E
  - sh auditoorne töö nädalas: 1 loeng ja 1 harjutustund
  - (loenguid 2 tundi nädalas ja praktilisi ülesannete lahendamisi 2 tundi nädalas poolrühma kaupa)
  - Eksam kirjalik testina.
  - Eksami eelduseks on lahendatud ja esitatud individuaalne ülesanne.
  - Võimalike ülesannete tekstid vastavalt loengutele on jooksvalt veebileheküljel

ÕPPEAINE IRZ0020 Kodeerimine ja krüpteerimine

- **Õppeaine eesmärk:**
  - Häirekindla kodeerimise ja edastuse salastamise meetodite õpetamine telekommunikatsiooni õppesuuna bakalaureuse õppes suunaõppe kohustusliku ainena.

Loenguteemad

- 1. **Infoedastussüsteemi struktuurskeemid, tema koostisosad. Infoallikate liigid.**
- 2. **Diskreetsete ja pidevate infoallikate kirjeldused.**
- 3. **Edastuskanalid ja infohulgad.**
- **4. Edastuskanalite läbilaskevõimed.**
- **5. Edastuskanali sobitamine infoallikaga.**
- **6. Edastuskanali häirekindlus.**
- **7. Kodeerimise põhialused. Koodide liigid.**
- **8. Süstemaatilised lineaarsed koodid.**

Loenguteemad

- **9. Tsükkelkoodid.**
- **10. BCH koodid.**
- **11. RS koodid.**
- **12. Ahendkoodid.**
- **13. Koodide pesastamine**
- **14. Krüpteerimise alused**
- **15. m –jada.**
- **16. Krüpteerimise algoritmid ja meetodid.**

Kirjandus:

- Põhiõpikud:
  - 1. U. Madar "Sideteooria I" TTÜ 1996. a.
  - 2. U. Madar ja P. Puusemp "Kodeerimine" TTÜ 2000. a.
  - 3. G.C.Clark jr.,J.B.Cain "Error-Correction Coding for Digital Communications 1987.a. NY (TTÜ raamatukogus on ainult venekeelne tõlge)

Kirjandus:

- **Täiendav kirjandus:**
  - 1. R.E.Blahut "Theory and Practice of Error Control Codes" 1986.a.
- (TTÜ raamatukogus on venekeelne tõlge. U.M.I ka ingliskeelne)
  - 2. Shu Lin, D.J.Costello Error Control Coding Fundamentals and Appplications (ainult U:M.I)
  - 3. R. Gallager Principles of Digital Communications Cambridge University Press 2008
    - ISBN 978-0-521-87907-1

Täiendav kirjandus:

- 3. Handbook of Coding Theory. Vol. 1. (kohaviit TTÜ raamatukogus VB-88230)
- 4. Handbook of Coding Theory. Vol. 2. (kohaviit TTÜ raamatukogus VB-88854

Täiendav kirjandus

- 5. Gallager, Robert G. Information Theory and Reliable Communication.
  xiv, 588 lk. : New York : Wiley, 1968
  ISBN/ISSN: 0471290483
  UDK: 621.391 (075.8) est
  (kohaviit TTÜ raamatukogus VB-95010)
- 6. Ch. Schlegel Trellis and Turbo Coding Wiley Press,
  (kohaviit TTÜ raamatukogus VB-96170)

Kirjandust lisandub jooksvalt.

Üliõpilastele

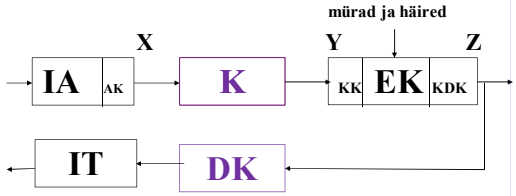
- Individuaalse ülesande saamiseks saata e-kiri U. Madar ile
  e - mail aadressil:
  umadar@lr.ttu.ee
- Harjutustundides on abiks RSTI teadur Julia Berdnikova

Õppejõud:

- Urve Madar
  - Raadio- ja sidetehnika instituudi (RSTI) signaalitöötluse õppetooli dotsent

1. Infoedastussüsteemi struktuurskeemid, tema koostisosad. Infoallikate liigid.

- **Infoedastussüsteemi struktuurskeem**



Struktuurskeemi koostisosad

- IA – infoallikas (**X**)
  - kõik teated ja signaalid, mis kuuluvad edastamisele (tavaliselt ajalise järjestikuse jadana)
  - Kahendsümbolite jada moodustamiseks kasutatakse allika kooderit
- EK – edastuskanal (**Z**)
  - tehniliste vahendite kogu selleks edastuseks
  - Praktiline edastamine käib kasutades kanalisignaale ehk -koode
- K – kooder (**Y**)
  - sobituste kogu

Struktuurskeemi koostisosad

- DK dekooder
  - kasutab ära kõik kodeerimisega tekkinud lisaväärtused
- IT infotarbija
  - passiivne, soovib ainult usaldusväärset infot

Infoallikate kirjeldused.

- **Infoallikate liigid.**
  - **Diskreetsed infoallikad**
  - **Pidevad infoallikad**

Diskreetsed infoallikad.

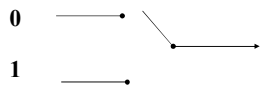
- **Lihtallikad**
  - **mäluta lihtallikad**
  - **mäluga lihtallikad (mõned mudelid: lihtne Markovi allikas, keeruline Markovi allikas)**

Liitallikad

- Mäluta
  - üksteisele järgnevate sümbolid on statistiliselt sõltumatud
- Mäluga
  - üksteisele järgnevad sümbolid on statistiliselt sõltuvad
  - üksteisele järgnevad teated on statistiliselt sõltuvad (nt Markovi allikad)

Diskreetsetest allikatest on oluline

- kahendallikas
  - **see on allikas, mille väljundis on sümbolid**



Kirjeldused

- IA - infoallikas
  - infoallika entroopia

$$H(X) = -\sum_{i=1}^{N} P(x_i) \log P(x_i)$$

  - infotekke kiirus

$$R(X) = \frac{H(X)}{\tau_x}$$

- EK - edastuskanal
  - kanali läbilaskevõime

$$C_k = \sup \frac{1}{T_k} [I_K]_{T_k}$$

  - sümboli vigasuse tõenäosus kanalist

$$P_v \approx \frac{N_v}{N}$$

Infoteooria põhiteoreem

- Kui

$$R(X) \le C_k - \varepsilon, \quad \varepsilon > 0$$

- siis on olemas selline kooder, et

$$P_v \rightarrow 0$$

Edastuskanalid

- Mudelid
  - diskreetsed
  - pidevad

Märkus

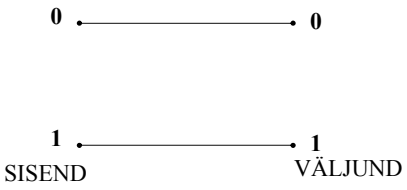
- Diskreetse kanali mudel on tingitud pidevast edastuskanali osast
  - see tähendab, et edastuskanali sisendis moodustatakse valitud kanalisignaalid kanali sümbolite edastuseks
  - edastusliinides lisanduvad nendele kanalisignaalidele mürad ja häired
  - mis segavad edastuskanali vastuvõtu poolel kanalisignaalide eristamist

Diskreetsed kanalid

- Müravabad (vigasid ei ole)
  - kahendkanalid
    - edastatakse kahte kanali sümbolit ehk tähte
  - mitmesed kanalid L_K
    - kanali sümbolite arv on

Edastuskanalit esitatakse kanali graafina

Müravaba kahendkanal



Diskreetsete müradega kanalite mudelid

- Mäluta (ehk juhuslik)
  - üksteisele järgnevad sümbolid on vigased üksteisest statistiliselt sõltumatult)
- mäluta kanalid võivad olla
  - sümmeetrilised
  - mittesümmeetrilised
- mäluta diskreetsed kanalid on tingitud tavaliselt nn *Gauss* i pidevast kanalist

Diskreetsete müradega kanalite mudelid

- Mäluga kanalid
  - sümbolite vead sõltuvad üksteisest statistiliselt
  - kanalile on iseloomulikud erinevat liiki vigade paketid
- Mäluga diskreetsed kanalid on tavaliselt tingitud pidevatest hääbumistega kanalitest

SÜMBOLI VIGASUSE TALUVUS

- **Digitaalne telefonitehnika, piirväärtus**

$$\frac{N_{SV}}{N_S} = 10^{-3} = \frac{100}{100 \times 10^3}$$

- **Andmeedastuseks hea**

$$\frac{N_{SV}}{N_S} = 10^{-7} = \frac{100}{100 \times 10^7}$$

SÜMBOLI VIGASUSE TALUVUS

- **Avariipoile hea**
  - **Avariipoi n = (127,106) = (82,61)**

$$\frac{N_{SV}}{N_S} = 10^{-3} \xrightarrow[HK]{} 2\times10^{-5}$$

  - S/M suhtes on kasu 1,8 korda

SÜMBOLI VIGASUSE TALUVUS

- **Militaristidele sobiv**

$$\frac{N_{SV}}{N_S} = 10^{-9} = \frac{100}{100\times10^{9}}$$

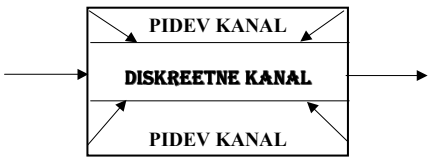
- **Kvaliteetne digikaamera**

$$\frac{N_{SV}}{N_S} = 10^{-12} = \frac{100}{100\times10^{12}};$$

$$N_S = 10^{14}$$

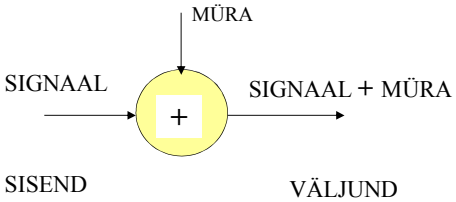
Diskreetsete ja pidevate kanalite

- vaheline seos



Pidevad kanalid

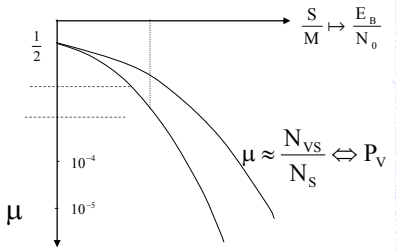
- Gaussi kanal (AWGN)



Pidevad kanalid

- Tavaliselt raadiokanalid
- Mitte teadaoleva algfaasiga
- Mitmekiirelise leviga
  - mis tingib kanalisignaali hääbumise
    - aeglased hääbumised
    - kiired hääbumised

Diskreetse ja pideva edastuskanali vaheline üleminek



Kanali mudelite vahelised üleminekud

- On

$$\begin{array}{lllllll} IA & \rightarrow & K \rightarrow & M & \rightarrow EK & \rightarrow & DM \rightarrow DK \\ X & \rightarrow & Y \rightarrow & & Z & & \mu_e \\ \mu_e & < & \mu & & \frac{S}{M} & \rightarrow & \mu \nearrow \end{array}$$

- läbilaskevõimed

$$C_D(\mu,..) < C_P\left(\frac{S}{M},...\right)$$

Küsimused

2. Diskreetsete ja pidevate infoallikate kirjeldused.

- A. Diskreetsed infoallikad.
- 1. Lihtallikas. Seisundite tabel:

x_1	x_2	...	x_i	...	x_{N-1}	x_N
$P(x_1)$	$P(x_2)$		$P(x_i)$		$P(x_{N-1})$	$P(x_N)$

- on täielik, kui

$$\sum_{i=1}^{N} P(x_i) = 1$$

1. Diskreetse (mäluta) lihtallika entroopia

- arvutatakse valemiga

$$H(X) = -\sum_{i=1}^{N} P(x_i) \log P(x_i)$$

Entroopia omadused

- Entroopia on allika määramatuse mõõt

1. $H(X) \geq 0$

2. $H(X) = 0$, kui on olemas niisugune x_i, et $P(x_i) = 1$

3. **$H(X) = H_{max}(X)$, kui kõik $P(x_i)$ on võrdsed**

- Entroopia on maksimaalne keskmine infohulk ühe sümboli kohta

Märkus

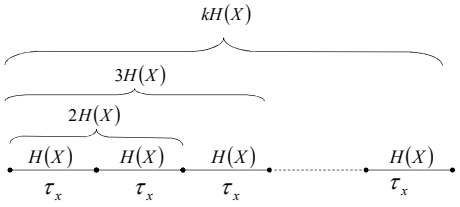
- Kui infoallikast väljastatavate elementaarsete sümbolite esinemise tõenäosused ei ole võrdsed (ühesuguste väärtustega), siis infoallikas on liiane
  - liiane - (see tähendab, et sümboleid on liiga palju)

Allika liiasus

- On tingitud sümbolite erinevatest esinemistõenäosusest

$$U(X) = \frac{H_{max}(X) - H(X)}{H_{max}(X)}$$

Teadete jada allikast



Infotekkekiirus allikast

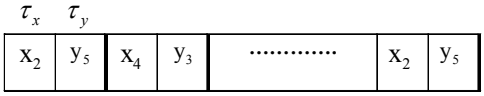
- Ühtlane allikas (kõikide sümbolite tekkeaeg on sama ja võrdne τ_x)

$$R(X) = \frac{kH(X)}{k\tau_x} = \frac{H(X)}{\tau_x}$$

Kui sümbolite tekkeaega saab muuta, siis on infoallikas juhitav

2. Diskreetse liitallika entroopia

2. Diskreetne liit-allikas. Seisundite moodustumine



- Liitallika tähistus

$$X + Y$$

Seisundite

- tabel

X\Y	y_1	y_2	....	y_i	....	y_{N-1}	y_N
x_1				$P(x_i,y_i)$			
x_2							
....							
x_N							

Seisundite tõenäosuste

- arvutamiseks

$$P(x_i y_j) = P(x_i)P_{x_i}(y_j) = P(y_j)P_{y_j}(x_i)$$

Entroopia

- arvutatakse

$$H(X+Y) = -\sum_{i=1}^{N}\sum_{j=1}^{N} P(x_i y_j)\log P(x_i y_j) =$$
$$= H(X) + H_X(Y) = H(Y) + H_Y(X)$$

Tinglik entroopia

- arvutatakse

$$H_X(Y) = -\sum_{i=1}^{N} P(x_i)\sum_{j=1}^{N} P_{x_i}(y_j)\log P_{x_i}(y_j)$$

Piirväärtuslikud juhtumid

- **Ühendati kaks sõltumatut allikat**

$$H(X+Y) = H(X) + H(Y)$$

Piirväärtuslikud juhtumid

- **Ühendati kaks reegliga seotut allikat**

$$H(X+Y) = H(X) = H(Y)$$

$x_i, y_j \qquad y_i, x_j$

Märkus

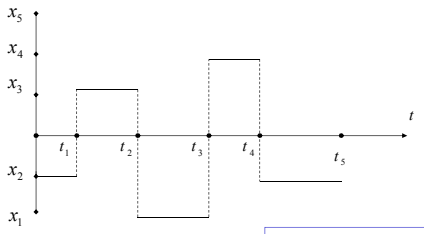
- **on kindel, et**

$$H(X) \geq H_Y(X)$$
$$H(Y) \geq H_X(Y)$$

Infotekkekiirus liitallikast

$$R(X+Y) = \frac{kH(X+Y)}{k(\tau_x + \tau_y)} = \frac{H(X+Y)}{\tau_x + \tau_y}$$

Sõltuvate teadete jada (mäluga allikas).



Juhuslik jada: $x_2, x_3, x_1, x_4, x_2, \ldots$

Teadete jada entroopia.

Jada moodustumine:

Algus	$X^{(0)}$
1. seisundi muutus	$X^{(1)}$
2. seisundi muutus	$X^{(2)}$

.....

Algus

Entroopia $H\left(X^{(0)}\right) = -\sum_{i=1}^{N} p_i^{(0)} \log p_i^{(0)}$

ajaintervall $0 \_\_\_\_\_ \tau_1$

arvutatakse tavaliselt põhivalemi järgi

Seisundi muutus

Entroopia $H\left(X^{(0)}, X^{(1)}\right) = H\left(X^{(0)}\right) + H_{X^{(0)}}\left(X^{(1)}\right)$

ajaintervallil $0 \_\_\_\_\_ \tau_1 \_\_\_\_\_ \tau_2$

Kogu jada entroopia

$$H\left(X^{(0)}, X^{(1)}, \ldots, X^{(k)}\right) =$$

$$H\left(X^{(0)}\right) + H_{X^{(0)}}\left(X^{(1)}\right) + H_{X^{(0)},X^{(1)}}\left(X^{(2)}\right) + \ldots..$$

$$\ldots\ldots\ldots\ldots\ldots\ldots + H_{X^{(0)},X^{(1)}\ldots,X^{(k-1)}}\left(X^{(k)}\right)$$

Markovi allikas

On selline,et sõltuvuses on ainult kaks kõrvuti asuvat teadet

$$H\left(X^{(0)}, X^{(1)}, X^{(2)}, \ldots\ldots, X^{(k)},\right) =$$
$$H\left(X^{(0)}\right) + H_{X^{(0)}}\left(X^{(1)}\right) + H_{X^{(1)}}\left(X^{(2)}\right) +$$
$$\ldots\ldots\ldots\ldots\ldots\ldots + H_{X^{(k-1)}}\left(X^{(k)}\right)$$

Statsionaarne Markovi aliikas

$$H\left(X^{(0)}, X^{(1)}, X^{(2)}, \ldots\ldots, X^{(k)},\right) =$$
$$H\left(X^{(0)}\right) + H_{X^{(0)}}\left(X^{(1)}\right) + H_{X^{(1)}}\left(X^{(2)}\right) +$$
$$\ldots\ldots\ldots\ldots\ldots\ldots + H_{X^{(k-1)}}\left(X^{(k)}\right) =$$
$$H\left(X^{(0)}\right) + kH_{X^{(k-1)}}\left(X^{(k)}\right)$$
$$H_{X^{(0)}}\left(X^{(1)}\right) = H_{X^{(1)}}\left(X^{(2)}\right) = \ldots = H_{X^{(k-1)}}\left(X^{(k)}\right)$$

Markovi allika infotekkekiirus

- arvutatakse

$$R_M(X) = \frac{H\left(X^{(0)}\right) + kH_{X^{(k-1)}}\left(X^{(k)}\right)}{k\tau_x}$$

Markovi allika tingliku entroopia arvutamine

- Lihtsustame

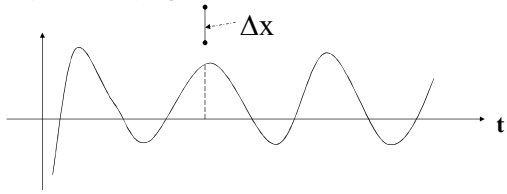
$$H_{X^{(k-1)}}\left(X^{(k)}\right) = -\sum_{i=1}^{N} p_i \sum_{j=1}^{N} p_{ij} \log p_{ij}$$

- Ülemineku tõenäosuste maatriks

$$\left\| p_{ij} \right\|$$

3. Pidevate infoallikate kirjeldused.

- **Pideva infoallika väljundiks on pidev (elektriline) signaal**



Tähistused

- x on väljavõtte juhuslik väärtus
- iga juhuslik suurus omab tõenäosuse tiheduse jaotusseadust

$$x \mapsto w(x)$$

- see tõenäosus, et x sattub intervalli Δx on

$$w(x)\Delta x$$

Pideva signaali ajaline diskreetimine

- Diskreetimise intervall peab olema mitte suurem kui

$$\Delta t \le \frac{1}{2\Delta F_x}$$

- Üldine väljavõtete arv on

$$N = 2\Delta F_x T_x$$

- a. **Täpselt ja korrektselt saab kirjeldada esialgu ainult ühte pideva signaali väljavõtet**

$$H_d(x) = -\int_{-\infty}^{+\infty} w(x)\log_a w(x)dx \overbrace{-\log \Delta x}^{\text{täpsusest}}$$

$$h(x) = -\int_{-\infty}^{+\infty} w(x)\log_a w(x)dx, \quad \begin{matrix} a = 2 \\ a = 10 \\ a = 2{,}7.. = e \end{matrix}$$

$$H(x) = N \times h(x) = 2T_x \Delta F_x \times h(x)$$

Reegel

- **Kui kõikide jaotusseaduste jaoks on teist järku normeeritud momendid võrdsed, siis on suurima diferentsiaalse entroopiaga normaaljaotusega juhuslikud suurused.**

Normaaljaotuse puhul

$$w(x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-a)^2}{2\sigma^2}}, \quad \begin{matrix} a \\ \sigma^2 \end{matrix}$$

$$h(x)=\ln\sqrt{2\pi e\sigma^2}$$

ei sõltu $a-st$

Normaalse allika

- Üldine entroopia on

$$H(x)=2T_x\Delta F_x\ln\sqrt{2\pi e\sigma^2}$$

Piirangud

- **Pidevat juhuslikku suurust ei saa kunagi esitada lõpmatu täpselt**
- **kasutatakse ε - entroopia mõistet**

$$h_\varepsilon(x)=\min I_\varepsilon=h(x+\Delta x)-\ln\sqrt{2\pi e\sigma_\varepsilon^2}=$$

$$=\frac{1}{2}[\ln 2\pi e(\sigma_x^2+\sigma_\varepsilon^2)-\ln 2\pi e(\sigma_\varepsilon^2)]$$

Normaalse infoallika

- **ε - entroopia**

$$h_\varepsilon(x) = \frac{1}{2}\ln\left[\left(\frac{\sigma_x^2}{\sigma_\varepsilon^2}\right) + 1\right]$$

Pideva signaali

- Üldine **ε - entroopia**

$$H_\varepsilon(x) = \Delta F_x T_x \ln\left[\left(\frac{\sigma_x^2}{\sigma_\varepsilon^2}\right) + 1\right]$$

Pideva signaali ε - infotekkekiirus

- on

$$R_\varepsilon(X) = \frac{H_\varepsilon(X)}{T_x}$$

Pideva allika praktiline kirjeldus

- kvantides diskreedid
  - Kvantimisnivoode arv

$$M = \frac{x_{max} - x_{min}}{\Delta x} + 1$$

$$\varepsilon_{max} = \frac{\Delta x}{2} = \frac{x_{max} - x_{min}}{2(M-1)}$$

Normaalse infoallika jaoks

- St, et kui

$$x \mapsto N\left(a, \sigma^2\right)$$

- siis

$$x_{max} - x_{min} \approx 3\sigma - \left(-3\sigma\right) = 6\sigma$$

Pidev infoallikas

- On samane diskreetsega, mille teadete arv on

$$M_x = M^N = M^{2T_x \Delta F_x}$$

Samase diskreetse allika

- Suurim entroopia on

$$\mathbf{H}_{max}^{p \to d}\left(\mathbf{x}\right) = 2\mathbf{T}_x \Delta \mathbf{F}_x \log_{\mathbf{a}} \mathbf{M},$$

$$\mathbf{a} = 2$$

$$\Delta \mathbf{x} \approx \boldsymbol{\sigma}_{\boldsymbol{\varepsilon}}$$

Infoallika esitustäpsus

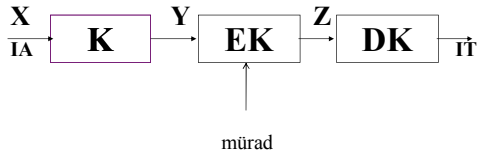
- Antakse tavaliselt ruutkeskmise veaga

$$\mathbf{x} = \frac{\boldsymbol{\sigma}_{\boldsymbol{\varepsilon}}}{\sqrt{\boldsymbol{\sigma}_x^2}} 100\%$$

$$\boldsymbol{\sigma}_x^2 = \mathbf{P}_x\left(\mathbf{mW}\right)$$

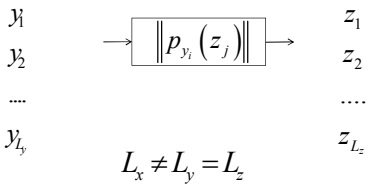
3. Edastuskanalid ja infohulgad.

- **Edastuskanal koos sobitusega:**



Edastuskanali mudel (diskreetne)

- on tinglike tõenäosuste maatriks



$$L_x \neq L_y = L_z$$

Ülemineku tõenäosuste maatriksi

- Reeglid:
  - *i*- rida vastab *i* - sisendsümbolile
  - *j*-veerg vastab *j* – väljundsümbolile
  - Ridade tõenäosuste summa jaoks kehtib:

$$\sum_{j=1}^{L_z} p_{y_i}(z_j) = 1$$

Kui read on ainult tõenäosuste ümberpaigutused, siis on edastuskanal sümmeetriline (*R. Hemming*).

- **Kõikide tõenäosuste kohta kehtib:**

$$\sum_{i=1}^{L_y} p(y_i) \sum_{j=1}^{L_z} p_{y_i}(z_j) = 1$$

- **Edastuskanal on statsionaarne, kui üleminekutõenäosuste maatriksi elemendid ei muutu ajas.**

Tõenäolikud suhted edastuskanalis.

- **Edastuskanali väljundsümboli esinemise tõenäosus**

$$\sum_{i=1}^{L_x} p(x_i)\, p_{x_i}(y_j) = p(y_j), j \subset [1; L_y]$$

- **Sisendi ja väljundi sümbolite koosesinemiste tõenäosused.**

Koosesinemiste tõenäosuste kohta kehtivad seosed

$$p(y_i, z_j) = p(y_i)\, p_{y_i}(z_j);$$
$$p(y_i, z_j) = p(z_j)\, p_{z_j}(y_i);$$
$$p(y_i)\, p_{y_i}(z_j) = p(z_j)\, p_{z_j}(y_i);$$
$$j \subset [1; L_z]; i \subset [1; L_y]$$

Baesy i valem

Edastuskanali sisend- ja väljundsümbolite kohta

$$p_{z_j}(y_i) = \frac{p(y_i)\, p_{y_i}(z_j)}{\sum_{i=1}^{L_x} p(y_i)\, p_{y_i}(z_j)};$$

$$\sum_{i=1}^{L_x} p_{z_j}(y_i) = 1$$

Nimetused

- Aprioorsed tõenäosused:

$$p(y_i); p(z_j); i \subset [1; L_y]; j \subset [1; L_z]$$

- Aposterioorsed tõenäosused ja infohulk:

$$p_{y_j}(x_i); i \subset [1; L_x]; j \subset [1; L_y],$$

$$I(z_j \to y_i) = \log \frac{p_{z_j}(y_i)}{p(y_i)} = \log \frac{1}{p(y_i)} - \log \frac{1}{p_{z_j}(y_i)}$$

Suhteline infohulk

- **Väljundtähes sisendtähe suhtes on juhuslik suurus sõltuvalt indeksitest.**
- **Suhteline infohulk võib olla positiivne või negatiivne sõltuvalt sellest, kas aposterioorne tõenäosus on suurem või väiksem, kui aprioorne tõenäosus.**
- **Suhteline infohulk on võrdne nulliga, kui aposterioorne tõenäosus ei muutu võrreldes aprioorsega**

$$I\left(z_j \to y_i\right) = \log\frac{p\left(y_i\right)}{p\left(y_i\right)} = 0; p_{z_j}\left(y_i\right) = p\left(y_i\right)$$

Suhteline infohulk

- On arvutatav nii sisend- kui ka väljundsümbolite suhtes:

$$I\left(z_j \to y_i\right) = \log\frac{p_{z_j}\left(y_i\right)}{p\left(y_i\right)}; I\left(y_i \to z_j\right) = \log\frac{p_{y_i}\left(z_j\right)}{p\left(z_j\right)};$$

$$I\left(y_i\right) = \log\frac{1}{p\left(y_i\right)}; I\left(z_j\right) = \log\frac{1}{p\left(z_j\right)};$$

$$I\left(z_j \to y_i\right) \le I\left(y_i\right); I\left(y_i \to z_j\right) \le I\left(z_j\right)$$

Keskmine infohulk

- sisendis väljundsümboli kohta ja vastupidi on

$$I\left(Y \to z_j\right) =$$

$$\sum_{i=1}^{L_y} p_{z_j}\left(z_i\right) I\left(y_i \to z_j\right) = \sum_{i=1}^{L_y} p_{z_j}\left(y_i\right) \log\frac{p_{z_j}\left(y_i\right)}{p\left(y_i\right)};$$

$$I\left(Z \to y_i\right) =$$

$$\sum_{i=1}^{L_y} p_{y_i}\left(z_j\right) I\left(z_j \to y_i\right) = \sum_{i=1}^{L_y} p_{y_i}\left(z_j\right) \log\frac{p_{y_i}\left(z_j\right)}{p\left(z_j\right)};$$

Keskmine infohulk

Sisendi ja väljundi vahel on:

$$I\left(Y \to Z\right) = I\left(Z \to Y\right) =$$

$$\sum_{i=1}^{L_y}\sum_{j}^{L_z} p\left(y_i, z_j\right) I\left(y_i \to z_j\right) =$$

$$\sum_{i=1}^{L_y}\sum_{j}^{L_z} p\left(y_i, z_j\right) \log\frac{p_{z_j}\left(y_i\right)}{p\left(y_i\right)};$$

$$I\left(Y \leftrightarrow Z\right) = H\left(Y\right) - H_Z\left(Y\right) = H\left(Z\right) - H_Y\left(Z\right)$$

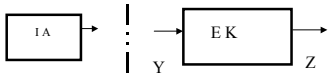
Tinglikud entroopiad

- Avalduvad valemiga:

$$H_Y(Z) = -\sum_{i=1}^{N} P(y_i) \sum_{j=1}^{N} P_{y_i}(z_j) \log P_{y_i}(z_j)$$

4. Edastuskanalite läbilaskevõimed.

A. Diskreetne edastuskanal. Struktuurskeem



Diskreetne edastuskanal

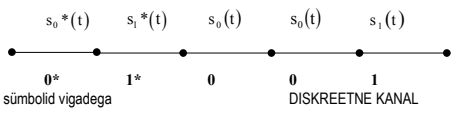
on mudel

- parameetriteks on
  - edastatavate (teadete) või sümbolite arv, mida vahel nimetatakse ka kanali tähestikuks
    $L_k = N$
  - ühe kanali (teate) või sümboli edastuse aeg
    $\tau_k = \tau$

Diskreetne edastuskanal

- Iseloomulikuks
  - on müradest ja häiretest tingitud sümbolite vigasus

sümbolitele vastavad moonutatud signaalid PIDEV KANAL



Diskreetse edastuskanali

- Sisendi ja väljundi vahelist seost iseloomustab keskmine infohulk ühe teate kohta

$$I_{Y \Leftrightarrow Z} = H(Y) - H_Z(Y) = H(Z) - H_Y(Z)$$

Tinglik entroopia

- On määratud

$$H_Y(Z) = -\sum_{i=1}^{N} P(y_i) \sum_{j=1}^{N} P_{y_i}(z_j) \log P_{y_i}(z_j)$$

- tinglike tõenäosustega, $P_{y_i}(z_j)$

mis iseloomustavad edastuse vigasust

Diskreetse edastuskanali

- Läbilaskevõime on
  - kanali väljundis saadava infohulga ülemine piir ajaühikus

$$C_K = \sup\left\{\frac{1}{T_K}\left[I_{Z\Leftrightarrow Y}\right]_{T_K}\right\}, \left(\frac{\text{bit}}{\text{sec}}\right)$$

Tähistused

- **Edastuskanali tööaeg** T_K
- **Infohulk selle aja jooksul** $\left[I_{Z\Leftrightarrow Y}\right]_{T_K}$

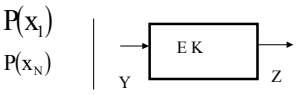
Läbilaskevõime leidmiseks

- **Peab koostama edastuskanali mudeli.**
- **Hea mudeliga võib õnnestuda, et**

$$C_K = \max_{\substack{P(y_1),\ldots,P(y_N) \\ \text{ja muud}}} \left\{\frac{1}{T_K}\left[I_{Z\Leftrightarrow Y}\right]_{T_K}\right\}, \left(\frac{\text{bit}}{\text{sec}}\right)$$

Läbilaskevõime leidmiseks

- Eraldatakse edastuskanal allikast ja koodrist ja sisendisse antakse oletatav sisend X=Y



Selgituseks

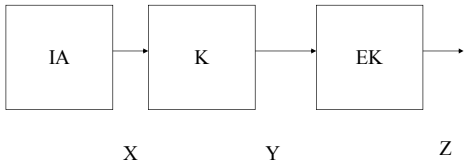
- **EK lahutatakse reaalsest infoallikast, sisendiks saab ideaalne allikas, mille teadete arv võrdub edastuskanali erinevate teadete arvuga ja milliste esinemise tõenäosuste muutmistega maksimeeritakse EK väljundis saadav infohulk ajaühikus.**

Infoteooria põhiteoreem

- Kui $R(X) \le C_K - \varepsilon, \varepsilon > 0, siis$

on olemas selline kodeerimisviis, **mis kindlustab lõpliku edastuskiiruse korral edastuskanalis edastuse usaldatavuse tõenäosuse mistahes lähedase ühele.**

Struktuurskeem



Põhiseosed $X \Rightarrow Y \Rightarrow Z$

Infoallika ja edastuskanali sobitamiseks kasutatakse koodrit.

Kodeerimine keskmist infohulka ühe sümboli kohta ei muuda.

Keskmine infohulk kanali väljundis ühe sümboli kohta on sama, mis infoallika väljundis.

Keskmine edastuskiirus tohi ületada kanali läbilaskevõimet

Märkus

- Diskreetse edastuskanali läbilaskevõime arvutamine võib kujuneda keeruliseks või võimatuks.
  - Ülemise piiri arvutamise saab vahel asendada maksimumi leidmisega.
  - Mitmemõõtmelise funktsiooni maksimum võib heal juhul kuuluda kumerasse hulka. Sellisel juhul on edastuskanali läbilaskevõime arvutatav.

Märkus

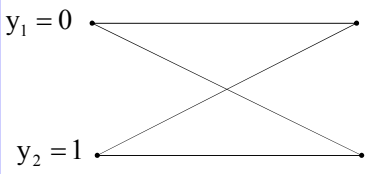
- Diskreetse edastuskanali läbilaskevõime arvutamine võib kujuneda keeruliseks või võimatuks.
  - Kui funktsiooni maksimum ei kuulu kumerasse hulka, siis on maksimume kas mitu või maksimumi olemasolu kaheldav (näiteks sadulpunkt), siis edastuskanali läbilaskevõimet arvutada ei saa

Praktiliselt

- Saab arvutada läbilaskevõimet, kui kahendkanal on
  - sümmeetriline
  - tugevalt asümmeetriline
  - kustutusega sümmeetriline

Statsionaarne kahendkanal

- mudel



Statsionaarsus

- Tähendab seda, et edastuskanali omadused (parameetrid) ei muutu ajaliselt

Kahendkanali läbilaskevõime

- valem

$$C_2 = \max_{P(y_1)} \frac{1}{\tau}\left[H(Z) + \sum_{i=1}^{2}\sum_{j=1}^{2} P(y_i, z_j)\log_2 P_{y_i}(z_j)\right]$$

$$P(y_1) = P(0); P(y_2) = P(1); P(0) + P(1) = 1$$

$$P_{y_1}(z_1) = P_0(0) \text{ jne.....}$$

Kahendkanali väljundi entroopia

- Leitakse

$$H(Z) = -P(z_1 = 0)\log_2 P(z_1 = 0) - P(z_2 = 1)\log_2 P(z_2 = 1)$$

võrranditest

$$P(z_1 = 0) = P(0)P_0(0) + P(1)P_1(0)$$

$$P(z_2 = 1) = P(0)P_0(1) + P(1)P_1(1)$$

Mistahes kahendkanali

- Läbilaskevõimet valemina leida ei õnnestu.
  - Vaja oleks leida kuuest komponendist koosneva summa maksimaalne väärtus üsna keerulisest parameetrist
  - sellise maksimumi kuulumine kumerasse hulka pole üldjuhul tõestatud

Sümmeetriline kahendkanal

- Edastuskanalis sümbolid muutuvad vigaseks tinglike tõenäosustega

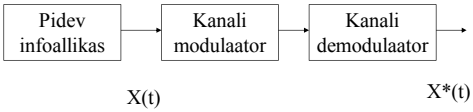
$$P_1(0) = P_0(1) = \mu$$

Sümmeetriline kahendkanali

- Kohta on teada, et kanali läbilaskevõime saavutatakse siis, kui sisendsümbolid esinevad võrdsete tõenäosustega
  - see kehtib kõikide diskreetsete sümmeetriliste kanalite kohta
  - diskreetsed kanalid võivad olla ka nõrgalt sümmeetrilised

B. Pidev edastuskanal

- On selline , millises toimuvad protsessid on pidevad või vaadeldakse pidevalt



Sümmeetriline kahendkanal

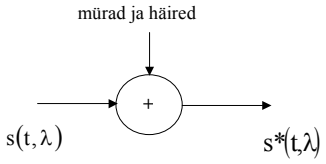
- Edastuskanalis sümbolid muutuvad vigaseks tinglike tõenäosustega

$$P_1(0) = P_0(1) = \mu$$

$$C_2 = \frac{1}{\tau}\left[1 + \mu \log_2 \mu + (1-\mu)\log_2(1-\mu)\right]$$

Pideva edastuskanal (AWGN)

- Lihtsaim struktuurskeem



Pideva edastuskanali

- Sisendis on signaal $s(t, \lambda)$

λ – on informatsiooni edasi kandvate parameetrite kogu

Pideva edastuskanali

- Väljundis on signaal

$$s^*(t, \lambda) =$$
$$= s(t, \lambda) + n(t) + h(t) = s(t, \lambda^*)$$

Edastuskanalit mõjutavad

- Mürad
  - mis on tingitud tavaliselt edastuskanali seadmete sisestest nn elektrilistest müradest
  - neid, peamiselt soojusliku iseloomuga mürasid võib lugeda nn "valgeks" müraks
- Häired on tavaliselt seadmete välised
  - lihtsamas käsitluses häireid ei arvestata

edastuskanalile mõjuva müra dispersioon

- Ehk müra keskmine normeeritud võimsus on

$$\sigma^2 = N_0 \Delta f_K$$

Pideva edastuskanali

- Väljundis on pidev signaal $s^*(t,\lambda)$
  - mille iga väljavõtte kohta saadav infohulk võrdub

$$I = h(s+n) - h(n)$$

See infohulk on maksimaalne

- **siis, kui signaali ja müra segu omab normaaljaotust, on võrdne**

$$I = h(s+n) - h(n) = \ln\left(1 + \frac{P_s}{\sigma^2}\right)^{\frac{1}{2}}$$

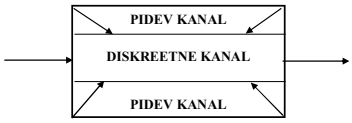
Pideva edastuskanali läbilaskevõime

- Shannon-Talleri valem

$$C_P = \Delta f_K \ln\left(1+\frac{P_s}{\sigma^2}\right) = \Delta f_K \ln\left(1+\frac{P_s}{N_0 \Delta f_K}\right)$$

$$\lim_{\Delta f_K \to \infty} C_P = \frac{P_s}{N_0}$$

Pideva ja diskreetse edastuskanali seos



Pideva ja diskreetse edastuskanali seos

- Igasugune diskreetne edastuskanal on tekitatud pideva kanali poolt

$$C_D(\mu,..) < C_P\left(\frac{S}{M},...\right)$$

Diskreetse edastuskanali läbilaskevõime sõltub

- Sümboli vigasuse tõenäosusest

  μ

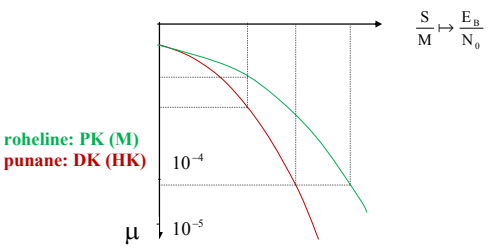
  – ükskõik kuidas see on arvutatud: kas keskmine vigasuse tõenäosus , piirväärtuslik jne

Pideva edastuskanali

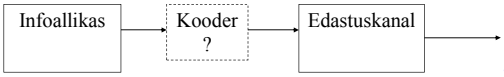
- Läbilaskevõime sõltub signaali ja müra suhtest

$$\frac{S}{M} = \frac{P_s}{\sigma^s} =_{\substack{"valge"müra \\ (AWGN)}} \frac{P_s}{N_0 \Delta f_k}$$

Diskreetse ja pideva edastuskanali graafiline seos (*BER*)



5. Diskreetse edastuskanali sobitamine infoallikaga.



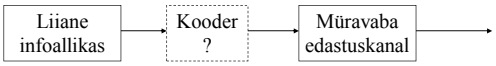
$$R(X) \le C_K - \varepsilon, \varepsilon > 0$$

NB! Koodid sobivad ainult kindlale edastuskanalile

Koodid sobivad ainult kindlale edastuskanalile:

1. **Müravaba kanali sobitamine müravaba kanaliga:**
   efektiivsed koodid
2. **Müradega sümmeetriline mäluta edastuskanal :**
   vigasid korrigeerivad koodid ehk häirekindlad koodid (vigu avastatakse, parandatakse,taastatakse kustutusi)
3. **Mäluga edastuskanal**
   vigasid parandavad koodid

1. Diskreetse müravaba edastuskanali sobitamine liiase diskreetse infoallikaga.



NB! Koodid sobivad ainult kindlale edastuskanalile

Müravaba diskreetse edastuskanali väljundis

- Saadav keskmine infohulk ühe allika teate kohta on võrdne infoallika entroopiaga.

Edastuskanali kasutamist

- hindame suhtega

$$\frac{R(X)}{C_K} = \frac{H(X)}{\tau_X} \cdot \frac{\tau_K}{\log L_Y} = 1 - U(S)$$

Sobituse liiasus

- on

$$U(S) = \frac{C_K - R(X)}{C_K}$$

vahel

$$U(S) = U(K)$$

Selleks, et

- Saavutada $R(X) \to C_K$

kasutatakse efektiivseid koode

Huffman 'i kood

Shannon - Fano kood

Need koodid on mitteühtlased

Kokkuvõte

- Diskreetne liiane infoallikas sobitatakse müravaba diskreetse edastuskanaliga efektiivsete koodide abil

A. Efektiivsed koodid

- Omadused
  - mitteühtlased
  - minimaalse liiasusega
  - võimaldavad dekodeerimist ilma vahemärkideta
  - on madala häirekindlusega

Shannon - Fano koodi koostamise reeglid

- 1. Kõik kodeeritavad teated järjestatakse esinemistõenäosuste vähenevasse ritta.
- 2. Kõik kodeeritavad teated vähenevas reas jaotatakse 2 rühma nii, et mõlema rühma summaarsed esinemise tõenäosused oleks lähedase väärtusele $\frac{1}{2}$

Shannon - Fano koodi koostamise reeglid

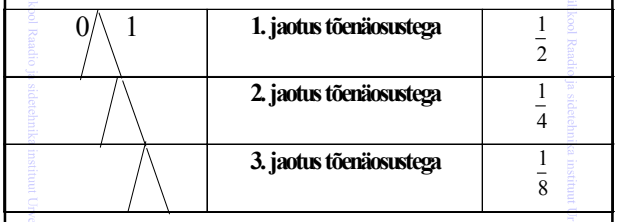
- 3. Esimesele poolrühmale omistatakse esimene sümbol 0, teisele poolrühmale 1.
- 4. Esimesed poolrühmad jaotatakse kumbki kaheks alamrühmaks nii, et mõlema alamrühma summaarsed tõenäosused oleks võrdsed $\frac{1}{4}$

Shannon - Fano koodi koostamise reeglid

- 5. Esimestele alamrühma teatetele omistatakse teine koodsümbol 0, teisele 1.

Jaotust jätkatakse tõenäosustega $\frac{1}{2^k}$ seni kuni igas viimase alamrühmas on ainult üks kodeeritav teade.

Shannon - Fano koodi koodipuu



Keskmine koodsõna pikkus ja liiasus

- Arvutatakse valemitega

$$\bar{n} = \sum_{i=1}^{L_X} n_i P(x_i)$$

$$U(S) = U(K) = \frac{\bar{n} - H(X)}{\bar{n}}$$

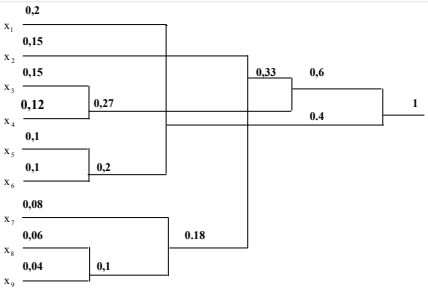
Huffman'i koodi koostamise reeglid

- 1. Kõik kodeeritavad teated järjestatakse esinemistõenäosuste vähenevasse ritta.
- 2. Rea kahe viimase teate tõenäosused liidetakse ja moodustatakse uus vähenev rida.
- 3. Järjestamist korratakse kuni jääb alles ainult üks järjestatud element.

Huffman'i koodi koostamise reeglid

- 4. Moodustatakse koodpuu, arvestades liidetud teadete uusi asukohtasid.
- 5. Koodipuu liikumistele omistatakse sümbolid 0 või 1 (näiteks liikumine üles – omistus 0 jne).

Näide



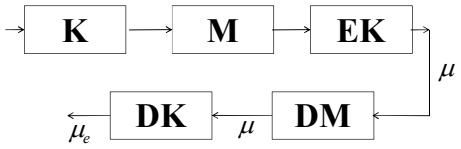
Kooditabel

x_1	10
x_2	000
x_3	010
x_4	011
x_5	110
x_6	111
x_7	0010
x_8	00110
x_9	00111

6. Infoedastuse häirekindlus.

- Pidevas kanalis toimub edastamine liinikoodide abil.
  - nn tegelikult lõpliku pikkusega ja teatud kujuga liinisignaalid
  - Häirekindlus on määratud mitme astmega
  - Liinisignaalide valik kindlustab sümboli vigasuse tõenäosuse μ

Edastuse häirekindlus



Liinikoodid ja nende vastuvõtmine

1. Kanalisignaalid ehk
2. liinisignaalid ehk
3. liinikoodid
- **moodustab modulaator**

Liinikoodidega ja optimaalse vastuvõtuga kindlustatakse μ

häirekindla kodeerimisega kindlustatakse $\mu_e < \mu$

A) Signaalide moodustamine.

I. Otsene meetod kahendkanali jaoks

(nn digitaalne edastamine)

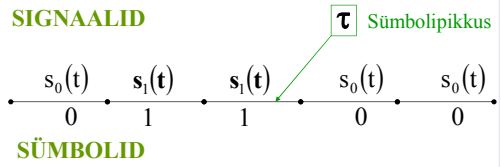
Liinikoodid ehk liinisignaalid

- on

$$s(t) = \begin{cases} s_o(t) \\ s_1(t) \end{cases}, t = [0;....;\ \tau]$$

Edastamine toimub järjestikku ajas:

- sümbolitele vastavad signaalid



EDASTAMINE TOIMUB JÄRJESTIKKU AJAS

Ajalised intervallid

- **on ühesugused ja võrdsed** τ (see on aeg, mille vältel edastatakse kanalis üks sümbol; see on ka liinisignaali pikkus)
- **Igal juhul peab kanali signaalid valima nii, et neid oleks võimalik eristada. Selleks tuleb valida kanali signaalid nii, et oleks kindlustatud signaalide vaheline ruutkeskmine kaugus.**

Signaalide vaheline kaugus

- On määratud valemiga

$$d_{01}^2 = \int_0^{\tau} (s_0(t) - s_1(t))^2 \, dt$$

SIGNAALE, MILLISTE VAHELINE KAUGUS ON NULL, EI SAA ERISTADA

Signaalide vaheline kaugus

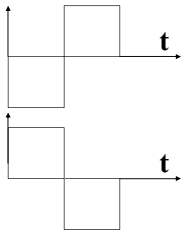
$$d_{01}^2 = \int_0^\tau (s_0(t) - s_1(t))^2 \, dt = \int_0^\tau s_0^2(t)dt - 2\int_0^\tau s_0(t)s_1(t)dt + \int_0^\tau s_1^2(t)dt$$

- On suurim siis, kui

$$s_1(t) = -s_0(t)$$

Manchesteri kood

- On hea



1 edastamiseks

0 edastamiseks

Diskreetmodulatsioonid

- **a) Diskreetne amplituudmodulatsioon passiivse pausiga (tinglikult binaarne)**

$$\left.\begin{array}{ll} 0: & s_0(t) = 0 \\ 1: & s_1(t) = U_m \sin \omega_0 t \end{array}\right\}, 0 \le t \le \tau$$

Diskreetmodulatsioonid

- **b) Diskreetne sagedusmodulatsioon** (binaarne)

$$\left.\begin{array}{ll} 0: & s_0(t) = U_m \sin \omega_0 t \\ 1: & s_1(t) = U_m \sin \omega_1 t \end{array}\right\}, 0 \le t \le \tau$$

Diskreetmodulatsioonid

- **c) Diskreetne faasmodulatsioon (binaarne)**

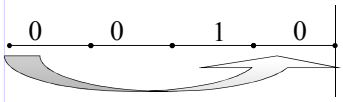
$$\left.\begin{array}{ll} 0: & s_0(t) = U_m \sin \omega_0 t \\ 1: & s_1(t) = U_m \sin(\omega_0 t + \varphi) \end{array}\right\}, 0 \le t \le \tau$$

$\varphi = \pi \rightarrow$ otsene faas mod ulatsioon

$\varphi = \frac{\pi}{2} \rightarrow$ ortogonaa ln e mod ulatsioon

M – modulatsioon

- Moodustatakse plokk m sümbolist



$$\mathbf{M} = 2^{\mathbf{m}}$$

$$\Delta \mathbf{f_M} = \frac{\Delta \mathbf{f_2}}{\mathbf{m}}$$

Raadiokanalit

- **iseloomustatakse suurusega**

$$B = \frac{\frac{bit}{\sec}}{\Delta f_k} = \frac{bit/\sec}{Hz}$$

B=1;2;.....;16. B > 16 – keerulised modulatsiooonid

M - modulatsioon

- üldiselt

$$s(t) = \begin{cases} s_1(t) \\ s_2(t) \\ \dots \\ s_M(t) \end{cases}, t = [0, \dots \tau]$$

$$M = 2^m$$

M - modulatsiooni

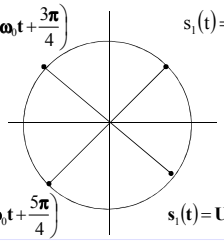
- puhul on oluline

$$d^2_{ij\,min} = \int_0^{m\tau} \left(s_i(t) - s_j(t)\right)^2 dt \Rightarrow \min_{i,j}$$

Näide

- 4 - faasmodulatsioon

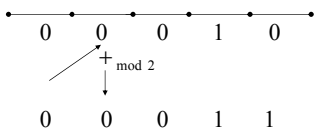
$\mathbf{s_1(t) = U_m \sin\left(\omega_0 t + \frac{3\pi}{4}\right)}$ $s_1(t) = U_m \sin\left(\omega_0 t + \frac{\pi}{4}\right)$



$\mathbf{s_1(t) = U_m \sin\left(\omega_0 t + \frac{5\pi}{4}\right)}$ $\mathbf{s_1(t) = U_m \sin\left(\omega_0 t + \frac{7\pi}{4}\right)}$

Signaalide moodustamine

- II. Suhteline signaalide valik



$\mathbf{b_1 = a_1, b_2 = a_2 +_{mod2} b_1, b_3 = a_3 +_{mod2} b_2, \ldots}$

Suhteline modulatsioon

- Sümbolite jada kodeeritakse üle
- **Saadud ülekodeeritud sümbolite jada moduleeritakse näiteks tavalise diskreetse faasmodulatsiooniga**
- Selline toiming viib sisse sümbolite omavahelise sõltuvuse ja vähendab kanalisignaalide spektri laiust.

Märkus

- **Edastuskanali mudelis tuleb arvestada sümbolite vahelise sõltuvusega**
- **Sellise liinisignaaliga kanal on mäluga edastuskanal.**
- **Sümbolite vaheline sõltuvus võib osutuda kasulikuks müradega edastuskanali korral**.

Liinikoodide vastuvõtmine

- Teostatakse vastuvõtu poolel demodulaatoriga

1. Otseste modulatsioonide korral kasutatakse
   - Amplituud-,
   - sagedus- või
   - faasdetektoreid

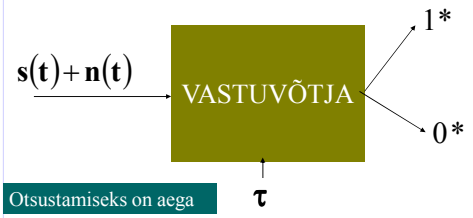
Liinikoodide vastuvõtmine

- Teostatakse vastuvõtu poolel keerulisema demodulaatoriga

2. Suhteliste modulatsioonimeetodite korral
   Igal juhul peab taastuma esialgne sümbolite jada

Eristaja struktuurskeem

- optimaalne



Vastuvõtja

- analüüsib signaali ja müra segu **s(t)+n(t)** ja selle alusel võtab vastu otsuse, kas sisendis oli signaal ,

1. mis kandis edasi sümbolit **0** või sisendis oli
2. signaal , mis kandis edasi sümbolit **1.**

Analüüsi aeg

- Vastuvõtjal on vastuvõetud signaali analüüsiks aega nii palju, kui pikad on oletatavad signaalid erinevate sümbolite edastamiseks.
- On kaks moodust sümboli väärtuse fikseerimiseks (nn fikseerimismoment)
  1. Signaali lõpus
  2. Signaali keskel

Vigasuse keskmine tõenäosus

- Leitakse valemiga

$$\mu = P(0)P_0(1^*) + P(1)P_1(0^*)$$

$\mathbf{P}(0), \mathbf{P}(1)$ **SÕLTUMATUD TÕENÄOSUSED**

$\mathbf{P}_0(1^*)$ Tinglik tõenäosus

$\mathbf{P}_1(0^*)$ Tinglik tõenäosus

Edastuskanal

- on sümmeetriline siis, kui

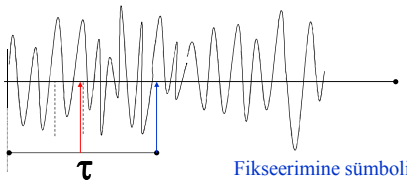
$$P_0(1^*) = P_1(0^*)$$

- Kui

$P(0) = P(1) = \frac{1}{2}$ **siis** $\mu = P_0(1^*) = P_1(0^*)$

$$\mu \approx \frac{N_{VS}}{N_S} \Leftrightarrow P_V$$

Fikseerimisaeg



Fikseerimine sümboli lõpus

Fikseerimine sümboli keskel

Vastuvõtjate liigid

a) optimaalsed (O)

b) **reaalsed (R)**

Optimaalne vastuvõtja

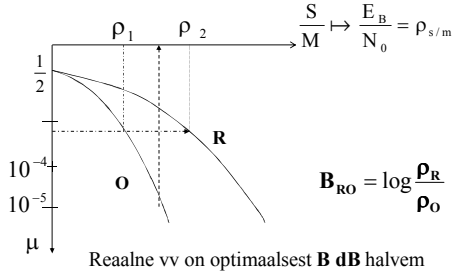
- kriteerium

$\mu \underset{\text{vastuvõtja struktuur}}{\Rightarrow} \min$

või

$f(\mu) \underset{\text{vastuvõtja struktuur}}{\Rightarrow} \min$

Häirekindluse kõverad



Reaalne vv on optimaalsest **B dB** halvem

Reaalne vastuvõtmine

- Igasugune vastuvõtmine reaalsetes tingimustes toimub segavate asjaolude toimel. Segavateks võivad osutuda ka mõned mitte täpselt teada olevad signaalide parameetrid. Näiteks oodatavate signaalide algusmomendid ja pikkus, keskmine sagedus, algfaas.

Aprioorne määramatus

- Mitte teada olevate parameetrite kogu moodustab nn aprioorse määramatuse. Signaalide eristamine aprioorse määramatuse tingimustes annab halvemad häirekindluse kõverad võrreldes täpselt teada olevate signaalide eristamisega. Kui mitte midagi ei ole signaalidest teada, siis vastuvõtmine on võimatu.

Täpselt teada olevate signaalide optimaalse vastuvõtja algoritm

- Analüüsitav segu vastuvõtja sisendis:

$$y(t) = s(t) + n(t) = \begin{cases} s_o(t) \\ s_1(t) \end{cases} + n(t), t = [0;....;\ \tau]$$

Algoritm

- Arvutatakse ruutkeskmised kaugused ja võrreldakse neid

$$\int_0^{\tau} (\mathbf{y(t)} - \mathbf{s_0(t)})^2 \mathbf{dt} - \int_0^{\tau} (\mathbf{y(t)} - \mathbf{s_1(t)})^2 \mathbf{dt} \geq \ln \frac{\mathbf{P(0)}}{\mathbf{P(1)}} \rightarrow 1^*$$

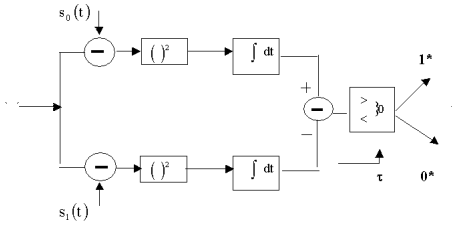
OTSUSTATAKSE LÄHIMA KAUGUSE JÄRGI

Eritingimusel

- Kui P(0) = P(1)

$$\int_0^{\tau}(y(t)-s_0(t))^2dt-\int_0^{\tau}(y(t)-s_1(t))^2dt \geq 0 \rightarrow 1^*$$

Optimaalse vastuvõtja struktuurskeem



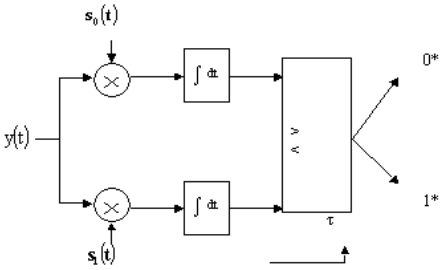
Vastuvõtja struktuurimuutused

- **Optimaalse vastuvõtja struktuure saab muuta, avaldades vastuvõetud signaali ja tugisignaalide vaheliste kauguste valemid.**

Täpselt teada olevate signaalide vastuvõtmine

- **Sellistes tingimustes vastuvõttu nimetatakse ka koherentseks vastuvõtuks.**

Korrelatsioonvastuvõtja struktuur



Häirekindluse kõverad

- saadakse järgmiselt:
  Oletame, et vastuvõetud signaal on:

$$y(t) = s_1(t) + n(t)$$

Leiame, millise tõenäosusega kehtib võrratus

$$\int_0^\tau (y(t) - s_0(t))^2 dt - \int_0^\tau (y(t) - s_1(t))^2 dt < 0 \rightarrow 0*$$

Teisendame

- Vasakut võrrandi poolt

$$\int_0^\tau (s_1(t) + n(t) - s_0(t))^2 dt - \int_0^\tau (s_1(t) + n(t) - s_1(t))^2 dt < 0 \rightarrow 0*$$

$$\int_0^\tau (s_1(t) - s_0(t))^2 dt + 2\int_0^\tau (s_1(t) - s_0(t)) n(t) dt + \int_0^\tau n^2(t) dt - \int_0^\tau n^2(t) dt < 0$$

Teisendame võrrandeid veel

- Viies jättes vasakule poolele ainult juhusliku suuruse

$$\int_0^\tau (s_1(t) - s_0(t)) n(t) dt < -\frac{d_{10}^2}{2}$$

$$\Theta < -\frac{d_{10}^2}{2}$$

Leiame juhusliku suuruse jaotusseaduse .

- On teada, et see on normaalne ehk Gaussi jaotusseadus

$$\Theta \mapsto \mathbf{w}(\Theta) = \mathrm{N}\left(\boldsymbol{\sigma}_{\Theta}, \overline{\Theta}\right)$$

$$w(\Theta) = \frac{1}{\sqrt{2\pi\sigma_{\Theta}^2}} \exp\left\{-\frac{(\Theta - \overline{\Theta})^2}{2\sigma_{\Theta}^2}\right\}$$

Leiame

- keskväärtuse

$$\overline{\Theta} = \int (s_1(t) - s_0(t))\overline{n(t)}dt = \int (s_1(t) - s_0(t))\overline{n(t)}dt = 0, \text{ kui } \overline{n(t)} = 0$$

Leiame

- tema autokorrelatsioonifunktsiooni kaudu

$$\Re_{\Theta}(t_1, t_2) = \overline{\int_0^{\tau} (s_1(t_1) - s_0(t_1))n(t_1)dt_1 \cdot \int_0^{\tau} (s_1(t_2) - s_0(t_2))n(t_1)dt_{21}} =$$

$$\int_0^{\tau}\int_0^{\tau} (s_1(t_1) - s_0(t_1))(s_1(t_{21}) - s_0(t_2))\overline{n(t_1)n(t_2)}dt_1dt_2$$

"Valge" segava müra korral

- saame

$$\sigma_{\Theta}^2 = \Re_{\Theta}(t_1 = t_2) \text{ ja } \overline{n(t_1)n(t_2)} = \frac{N_0}{2}\delta(t_1 - t_2)$$

$$\sigma_{\Theta}^2 = \frac{N_0}{2} d_{10}^2$$

Sümboli vigasuse tõenäosus

- 0n

$$\mu = \int_{-\infty}^{-\frac{d_{10}^2}{2}} \frac{1}{\sqrt{2\pi\sigma_\Theta^2}} \exp\left\{-\frac{\Theta}{2\sigma_\Theta^2}\right\} d\Theta = \frac{1}{2} - \int_0^{\sqrt{\frac{d_{10}^2}{2N_0}}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx =$$

$$= \frac{1}{2} - \Phi\left(\sqrt{\frac{d_{10}^2}{2N_0}}\right)$$

B) Terviklik vastuvõtmine.

- Kui vastuvõetud signaal on

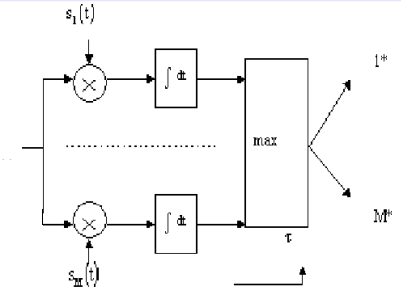
$$y(t) = s(t) + n(t) = \begin{cases} s_1(t) \\ s_2(t) \\ \dots \\ s_M(t) \end{cases} + n(t), t = [0, \dots \tau]$$

Optimaalne vastuvõtja teatud signaali korral

on M – kanaliline korrelatsioonvastuvõtja,

- fikseerimisega signaali lõpus ja
- valikuga suurima saavutatud väärtuse järgi:

Vastuvõtja struktuur



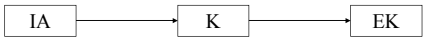
Märkus

- M – moduleeritud signaalide reaalne vastuvõtja on tavaliselt kahekanaliline.
- Pikemalt ja põhjalikumalt on erinevaid modulatsiooniviise käsitletud õppekirjanduses

1. A. Meister Modulatsioon, TTÜ, 1999

7. Kodeerimise põhialused. Koodide liigid

Kooder on ettenähtud selleks, et sobitada infoallikat edastuskanaliga.



NB! Koodid sobivad ainult kindlale edastuskanalile

Kodeerimise põhialused.

- Kodeerimine on võimalik, kui
  - infoteoreetilise põhiteoreemi tingimus on täidetud
  - st, et infotekkekiirus infoallikast ei saa olla suurem edastuskanali läbilaskevõimest

$$R(X) \le C_K - \varepsilon, \quad \varepsilon > 0$$

Kodeerimine

- Kodeerimist antud kontekstis käsitleme kui üleminekut ühelt tähestikult teisele

$$L_X \Rightarrow L_Y$$

- Infoallika tähestik L_X
- Koodri tähestik $L_Y = m$

Koodi alus

- Koodri tähtede (või elementaarsete sümbolite) arvu m nimetatakse tavaliselt koodi aluseks ja ta on tavaliselt väike arv
  - Sisuliselt on see arv määratud edastuskanali signaalide valikuga
- Enamlevinud on m = 2
  - tegemist on kodeerimisega alusel 2 ja koodid on nn kahendkoodid

Kodeerimine seisneb selles, et

- Igale infoallika teatele moodustatakse vastav koodsõna ehk kood

$x_1 \Rightarrow y_1$

$x_1 \Rightarrow y_1$

.....

$x_{N-1} \Rightarrow y_{N-1}$

$x_N \Rightarrow y_N$

See vastavus peab olema ühene

Tähistustest

- Koodsõna y_i
- Koodide hulk (vahel ka koodisüsteem)
  - antud kindlate omadustega koodsõnade kogu
  - (mat. hulk - diskreetne ja lõplik) $\{y_i\}$

Koodisõna pikkus

- On sümbolite arv koodsõnas

$y_i \rightarrow n_i$

- Kood on ühtlane, kui kõik koodsõnad on ühe pikkusega

$y_i \rightarrow n$

Kodeerimise võimalikkusest

- Kuna infoallika tähestik (kas infoallika teadete arv või elementaarsete sümbolite kogu) on tavaliselt suurem, kui koodi alus, siis on selge, et koodid on oma olemuselt liitsõnad
- Koodisüsteem peab olema piisavalt suur, et kõiki infoallika teateid oleks võimalik kodeerida

Koodri töökiirus

- Koodisümboli pikkus edastuskanalis ei saa olla lõpmata väike
  - kodeerimisega on selles ajas ka osa, mis on seotud praktilise koodri töökiirusega (mitu aritmeetilist tehet ja kui kiiresti neid õnnestub sooritada)

Häirekindla kodeerimise üldsätted

- Kui koodi alus on m ja kõikide koodsõnade pikkus on n, siis erinevate koodsõnade arv on

$$N_Y = m^n$$

- Koodide liiasuse sisseviimine tähendab seda, et koodidena kasutatakse vähem

$$N_X < N_Y$$

Koodsõnade jaotus

- Kõikvõimalikud koodsõnad jaotatakse lubatud ja keelatud koodsõnadeks

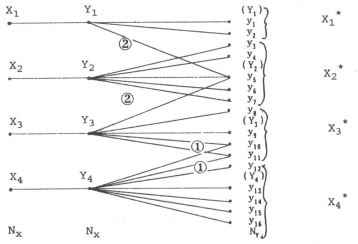
$$N_Y = N_l + N_k$$

$$N_l = N_X$$

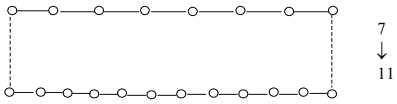
Koodsõnade jaotuse reeglid

- 1. Mistahes kaks lubatud koodsõna ei tohi edastamise käigus muutuda üheks ja samaks keelatuks.
- 2. Mistahes kaks lubatud koodsõna ei tohi muutuda mingiks teiseks lubatud koodsõnaks.

Näide



Ajaline edastuskanali ülesehitus



$$2^{11} = 2^7 + N_k$$

$$n = k + r = 7 + 4$$

Häirekindla koodi liiasus

- On arvutatav

$$U(K) = \frac{\log N_Y - \log N_1}{\log N_Y}$$

Koodikaugus ehk *Hamming*'i kaugus koodsõna kaal

- Kahe koodi vaheline kaugus on kahes koodsõnas erinevate sümbolite arv

$Y_i, Y_j \subset \{Y\}, Y_i = ({}^{v,j}y_n, y_{n-1}, \ldots, y_2, y_1)$

$d(Y_i, Y_j) = d(110001, 100110) = 4$

- Koodi minimaalne kaugus on

$d_{min} = \min_{i,j} d\{y\} = \min_{i,j}(y_i, y_j)$

Koodi kaal

- on ühtede arv koodsõnas

$$w(Y_i) = w(110001) = 3$$

$$d_{ij} = w\left(y_i \oplus y_j\right)$$

tavaliselt $\oplus_2 \rightarrow \oplus$

1	$\oplus_2$	0	=1
0	$\oplus_2$	1	=1
0	$\oplus_2$	0	=0
1	$\oplus_2$	1	=0

Koodikaugused ja vigasused

- q - kordse vea tõenäosus

$$P_n^q = C_n^q \mu^q (1-\mu)^{n-q} = \frac{n!}{q!(n-q)!} \mu^q (1-\mu)^{n-q}$$

$$P_V = \sum_{i=1}^{n} P_n^i = 1-(1-\mu)^n$$

$$P_V^* = P_V - P_P$$

$$P_V^* = 1-(1-\mu_e)^k \Rightarrow \mu_e \prec \mu$$

Vigade parandamiseks

- on vajalik jaotada nii, et

$$n = k + r$$

$$\sum_{i=1}^{L} C_n^i + 1 \le 2^r$$

Liiaste sümbolitega peab olema võimalik tähistada erinevate koodidega kõik vigased koodiseisundid

Üldreegel koodi minimaalsetele kaugustele

- Vigasid avastavad koodid.

$$d_{min} = q + 1$$

- Vigasid parandavad koodid

$$d_{min} = 2q + 1$$

$$d_{min} = \min_{y \neq 0} w(y_i)$$

Koodide liigid

- A. Efektiivsed koodid
  - kasutatakse liiaste diskreetsete infoallikate sobitamiseks müravabade edastuskanalitega
- B. Häirekindlad ehk korrigeerivad koodid
  - tavaliselt kasutatakse diskreetse infoallika sobitamiseks mäluta müradega (st vigasid tekitavate) edastuskanalitega
  - mäluga edastuskanalite korral kasutatakse erilisi vigade pakette korrigeerivaid koode

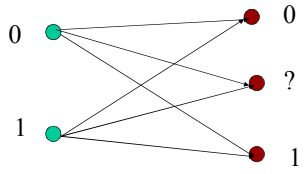
B. Häirekindlad koodid

- On sellised koodid, millistesse on sisse viidud korrastatult liiasus nii, et tekiksid koodi omadused korrigeerida kindlat tüüpi sümbolite edastusel tekkinud vigasid
- Häirekindlad koodid on tavaliselt ühtlased

Häirekindlad koodid on võimelised

- 1. Avastama teatud kordsusega vigu
- 2. Parandama teatud kordsusega vigu (parandama kõik vead kuni teatud kordsusega)
- 3. Taastama kustutusi

Kustutustega kanal (i.k. *eraeser*)



Kanali väljundjadas on kolm erinevat märki

0 1 0 0 ? 0 1 0 0 0 1 ? 0 1 0 1 0 0 1 ? ? 0 1 0

Vigasid avastavad koodid

- Koodi detekteerimisel saab vastuse, kas
1. Koodsõna on õige
2. Koodsõna on vale.

Näide : paariskood
ühtlase kaaluga kood

Vigasid parandavad koodid

- Ühekordseid vigu parandavad koodid
- Mitmekordseid vigu parandavad koodid

Häirekindlate koodide tüübid

Mittelineaarsed

Lineaarsed

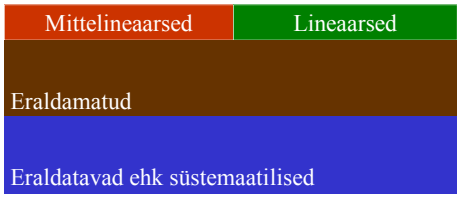
(vigasid avastavad ja/või parandavad koodid)

Näited

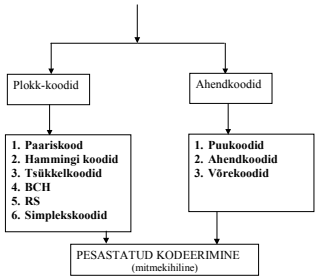
Mittelineaarsed koodid on

Preparata kood
Kerdock i kood
Omavahel on need koodid duaalsed

Häirekindlate koodide tüübid



Kodeerimisviiside liigitus



8. Süstemaatilised lineaarsed koodid.

- Lineaarsed on sellised koodid, mille lubatud koodsõnad moodustavad lineaarse alamhulga.
  - Kui kaks koodsõna on lubatud, siis on ka nende summa lubatud koodsõna
  - ainult nullidest koosnev koodsõna on ka lubatud koodsõna

Süstemaatilised

- Ehk eraldatavad on sellised koodid, mille sõnades saab selgelt eristada informatiivsed ja liiased sümbolid

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad r_1 \quad r_2 \quad r_3$

$y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7$

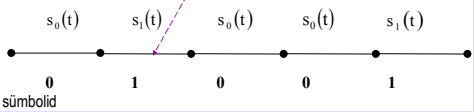
Edastusliinile läheb esimesena sümbol indeksiga 1
Edastuskanalis edastatakse sümbolid ajaliselt järjestikku, iga sümboli signaali pikkus on τ

Edastamine kahendkanalis

- Kahendsümbolitele valitakse vastavalt kaks signaali

$$s(t) = \begin{cases} s_o(t) \\ s_1(t) \end{cases}, t = [0;.....;\tau]$$

sümbolitele vastavad signaalid



Koodisõna pikkus jaotatakse

- Informatiivseteks ja liiasteks sümboliteks

$$n = k + r,$$

$$N_v \le 2^r,$$

N_v Lubatud koodsõna vigaste seisundite arv

A Kodeerimine tekitava maatriksiga

$Y = X\|G\|$ $\|G\|$ on $(k \times n)$ maatriks

- Lubatud koodsõna (koodivektor)
- Infokood Y
- X
- Tekitav maatriks $\|G\|$

Tekitav maatriks

$\|G\| = \|I / B\|$

Koosneb kahest osast

- Ühikmaatriks $\|I\|$
- Kordajate maatriks $\|B\|$

Ühikmaatriks

$$I = \begin{Vmatrix} 1 & .. & 0 \\ .. & .. & 0 \\ 0 & .. & 1 \end{Vmatrix}$$

Maatriksi diagonaalis on ühed, ülejäänud elemendid on kõik võrdsed nullidega

Kordajate maatriks

$$B = \begin{matrix} \\ \downarrow \\ i \\ v \\ e \\ r \\ g \end{matrix} \begin{matrix} \rightarrow j \text{ rida} \\ \begin{Vmatrix} b_{11} & .. & b_{1r} \\ ... & b_{ij} & .. \\ b_{k1} & .. & b_{kr} \end{Vmatrix} \end{matrix}$$

$$i = [1,....,k], j = [1,.....,r]$$

Kordajate maatriksist

- **Read peavad olema erinevad ja sisaldama piisava arvu ühtesid**
  - vajalik ühtede arv sõltub sellest, milliste vigade parandamiseks on kood ette nähtud
  - tavaliselt eristatakse ühekordsete ja mitmekordsete vigade parandamist

Kordajate maatriksist

- **Read peavad olema erinevad ja sisaldama piisava arvu ühtesid**
  - ainult mitmekordse vea parandamine iseenesest ei ole otstarbekas
  - kood moodustatakse tavaliselt parandama vigu kuni kordsusega q (st parandatakse kõik vead kuni kordsusega q)
  - parandamise käigus peab saama aru, mitmekordset viga praktiliselt parandatakse

Kontrollmaatriks

- On selline maatriks, mis kindlustab

$$Y\|H\|^T = 0$$

$$X\|G\| \cdot \|H\|^T = 0$$

$$\|G\| \cdot \|H\|^T = 0$$

$$\|I / B\| \|H\|^T = 0 \Rightarrow \|H\| = \|B^T / I\|$$

Kordajate maatriksi

- Ühtede arv
  - Koodi minimaalne kaal on võrdne w siis ja a ainult siis, kui mistahes kontrollmaatriksi H alamhulk w - 1 veerust on lineaarselt sõltumatu

Hamming'i koodid

- On lineaarsed süstemaatilised koodid, mis parandavad ühekordseid vigu.

$$n = k + r$$

$$n + 1 \le 2^r$$

$$k = n - r$$

Täiuslikud *Hammingi* koodid

- Mingite n korral kindlustub võrdsus

$$n + 1 = 2^r$$

$$k = n - r$$

$$n = 2^r - 1$$

Hammingi koodi omadused

- Liiasus

$$U(K) = \frac{r}{n} \quad (\cdot 100)\%$$

- keskmine infoedastuskiirus

$$R(X) = \frac{k}{n} \cdot \frac{1}{\tau}\left(\frac{bit}{sec}\right)$$

Infoallika liiasust selles näitajas tavaliselt ei võeta arvesse

Hammingi kood (7,4)

- Koodi pikkus ja selle jaotus

$$2^3 = n + 1$$

$$n = 7 = k + r = 4 + 3$$

Hammingi kood (7,4)

- Kordajate maatriks

$$B = \begin{Vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{Vmatrix}$$

Igas reas peab olema vähemalt kaks ühte

Read peavad olema lineaarselt sõltumatud

(Mingi kahe summa ei tohi anda mingit kolmandat)

Veerge võib ümber paigutada, saame mitu samast koodi

Hammingi kood (7,4)

- Sümbolite paigutus

$$x_1 \quad x_2 \quad x_3 \quad x_4 \quad r_1 \quad r_2 \quad r_3$$

$$y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad y_6 \quad y_7$$

Hammingi kood (7,4)

- Tekitav maatriks

$$\begin{Vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{Vmatrix}$$

Hammingi kood (7,4)

- Vektor - maatriks korrutis

$$[x_1, x_2, x_3, x_4] \cdot \begin{Vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{Vmatrix}$$

Hammingi kood (7,4)

- Vektor - maatriks korrutis

$$
\begin{aligned}
r_1 &= x_1 + x_3 + x_4 \\
r_2 &= x_1 + x_2 + x_3 \\
r_3 &= x_1 + x_2 + x_4 \\
y_1 &= x_1 + y_3 = x_3 \\
y_2 &= x_2 + y_4 = x_4
\end{aligned}
$$

Hammingi kood (7,4)

- Muud kodeerimise algoritmid
  - vastavustabeliga

lubatud koodsõnad on kõik tekitava maatriksi read ja nende ridade summad paarikaupa, kolmekaupa jne…..

12. Lineaarsete süstemaatiliste koodide dekodeerimine.

Dekodeerimiseks on palju võimalusi:

1. Otsene ehk terviklik dekodeerimine

Koodisõnale vastav komponentidest koosnev koodisignaal võetakse vastu terviklikult. Vastuvõtu paraleelsete kanalite arv peaks siis ideaalselt olema võrdne lubatud koodsõnade arvuga; praktiliselt kaks korda rohkem.

Lineaarsete süstemaatiliste koodide dekodeerimine.

Dekodeerimiseks on palju võimalusi:

2. Dekodeerimine sündroomi ehk kontrollarvu abil.

3. Majoritaarne dekodeerimine.

4. ……

……..

Dekodeerimine sündroomi ehk kontrollarvu abil.

- Vastuvõetud koodsõnale (koodivektorile) lisandub viga (veavektor)

$$Y^* = Y \oplus E$$

- Sündroom ehk kontrollarv on

$$C = Y^* H^T = (Y + E)H^T =$$
$$= YH^T + EH^T$$

Kontrollarv ehk sündroom

- Sõltub ainult veast, kuid ei sõltu konkreetsest koodsõnast.
- Kontrollarve peab olema nii palju, et oleks võimalik tähistada (nummerdada) kõik vigased koodsõna seisundid.
- Kontrollarv, mis koosneb ainult nullidest tähistab koodsõna seisundit "õige".

Kontrollmaatriks

- Sündroomi leidmiseks peab olema selline kontrollmaatriks, et

$$YH^T = 0$$

$$(XG)H^T = X(GH^T) = 0 \text{ ja}$$

kuna $GH^T = 0$, saame $H = \|B^T / I\|$

Hemmingi kood (7,4)

- Kontrollmaatriks on

$$H = \begin{matrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{matrix}$$

Vastuvõetud koodsõnas

- Võib olla üks vigane sümbol, mille asukoht pole teada

$Y^* = y_1^* \; y_2^* \; y_3^* \; y_4^* \; y_5^* \; y_6^* \; y_7^*$

Koodisõna seisundid on

1. Vigane on 1. sümbol
2. Vigane on 2. Sümbol
3. ……..jne…….

Sündroomi leidmiseks

- Moodustame uued liiased sümbolid

$$r_1^{**} = y_1^* \oplus y_3^* \oplus y_4^*$$
$$r_2^{**} = y_1^* \oplus y_2^* \oplus y_3^*$$
$$r_3^{**} = y_1^* \oplus y_2^* \oplus y_4^*$$

Kontrollarvu

- Leiame liitmisega

$$c_1 = y_5^* \oplus r_1^{**}$$
$$c_2 = y_6^* \oplus r_2^{**}$$
$$c_3 = y_7^* \oplus r_3^{**}$$

Sündroomi koodi järgi

- Moodustame parandusvektori

$$H = \left[\begin{array}{ccccccc|c} 1 & 0 & 1 & 1 & 1 & 0 & 0 & c_1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & c_2 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & c_3 \end{array}\right]$$

$$E^* = \begin{matrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{matrix}$$

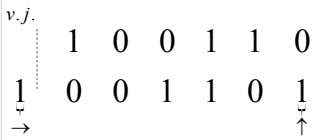
jne....

9. Tsükkelkoodid.

- Tsükkelkoodide kirjeldamiseks kasutatakse koodide esitamist hulkliikmetena.
- Tsükkelkoodid kuuluvad nn algebraliste koodide klassi.

Tsükliline nihe

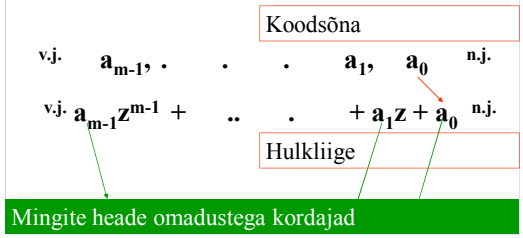
- Ühe takti võrra vanemate järkude poole



Koodsõnas tavaliselt vanemaid ja nooremaid järke ei eristata, kuid hulkliikmeliste koodi jaoks on see vajalik.

Koodisõna hulkliige

- saadakse



Mingite heade omadustega kordajad

Kahendkoodi hulkliige

- Koodisõna

v.j. 1 0 1 0 0 1 0 1 1

- Hulkliige on

$$f_{n-1}(z) = f_8(z) = 1\cdot z^8 + 0\cdot z^7 + 1\cdot z^6 + 0\cdot z^5 + 0\cdot z^4 + 1\cdot z^3 + 0\cdot z^2 + 1\cdot z^1 + 1\cdot z^0 =$$
$$= z^8 + z^6 + z^3 + z + 1$$

Koodi hulkliikmete kordajad

- Selliste koodide kirjeldamiseks ja analüüsiks sobivate hulkliikmete tehete jaoks on kõige sobivam kasutada kordajaid, mis kuuluvad mingisse lõplikku korpusesse.

Tehted lõpliku korpuse elemetidega

- Liitmine (ka lahutamine)
- Korrutamine
- Korrastamine (faktoriseerimine)
- Pöördelemendi leidmine
- .....
- Diskreetne logaritm
- .....

Tehted hulkliikmetega (peamised)

1. Liitmine.
2. Korrutamine
   - Otsene ehk tavaline
   - Faktorringis
3. Jagamine
   - Tavaline (tulemuseks on jagatis)
   - Faktorringis (tulemuseks on jääk)

Tsükkelkoodid

- On lineaarsete koodide alamklass, mis on määratud tugevate struktuursete nõuetega
  - Tsükkelkoodi lubatud koodsõnad kuuluvad nn tsüklilisse rühma
  - Kõrgema algebra eripeatükid, mis käsitlevad lõplikke korpusi, hulkliikmeid nendel korpustel, algebralisi rühmi jne

Galois' korpused

- Selliste koodide kirjeldamiseks ja analüüsiks sobivate hulkliikmete tehete jaoks on kõige sobivam kasutada kordajaid, mis kuuluvad mingisse lõplikku korpusesse. Neid lõplikke korpuseid nimetatakse *Galois*' korpused (*Galois Fields*) ja tähistatakse **GF(p^m).** Siin **p** on algarv ja **m** on täisarv.

Koodi alus

- Korpust **GF(p)** tähistav arv **p** on samastatav koodi alusega.
- Koodi alus on tavaliselt kas 2 (kahendkoodid) või mingi väike arv < 10.

Kahendkoodid

- Kahendkoodi hulkliige on selline, et tema kordajad kuuluvad *Galois´* korpusesse **GF(2)**.
- Korpuses **GF(2)** on elemente kaks **0** ja **1** ja nende liitmine toimub *modulo* **2** :

Liitmine *modulo* 2

- toimub

$$0 \oplus_{\mathrm{mod}\,2} 0 = 0$$
$$0 \oplus_{\mathrm{mod}\,2} 1 = 1$$
$$1 \oplus_{\mathrm{mod}\,2} 0 = 1$$
$$1 \oplus_{\mathrm{mod}\,2} 1 = 0$$

Tsükkelkoodide lubatud koodsõnade omadused

- **1) Tsükkelkoodide lubatud koodsõnad moodustavad nn. tsüklilise rühma.**
- **2) Kõikide lubatud koodsõnade hulkliikmed jaguvad jäägita tekitava hulkliikmega.**
- **3) Tekitava hulkliikme juured on ka kõikide lubatud koodsõnade hulkliikmete juurteks.**

I Kodeerimine.

1. Kindlustame koostatava koodi liiasuse, jaotades

$$n = k + r$$

Koodi liiasus on: $U(K) = \frac{r}{n}$

I Kodeerimine

2. **Valime sobiva r - astmelise hulkliikme $g_r(z)$ kui hulkliikme $(z^n + 1)$ ühiskordse**

r on liiaste sümbolite arv ja valitakse nii, et oleks võimalik nummerdada kõik koodsõna vigased seisundid.

Hulkliikmelised koodid

- Primitiivsed
  - On kindlasti olemas, kui koodi pikkus on primitiivne

$$n = 2^m - 1;\ m = 2;3;4;5;.....$$

A.Primitiivne tsükkelkood.

- eraldamatud
  - Infosümbolite asukohad lubatud koodsõnas ei ole määratletavad. Konkreetseid algsete infosümbolite väärtusi ei ole koodsõnas.
- eraldavatena (süstemaatilistena)
  - Infosümbolite asukoht on vastuvõtupoolel teada.

1) Eraldamatu tsükkelkoodi kodeerimisalgoritm.

- on

$y_{n-1}(z) = x_{k-1}(z) g_r(z)$ — Algoritm

$y_{n-1}(z)$ — Lubatud koodsõna

$x_{k-1}(z)$ — Infokood

$g_r(z)$ — Tekitav hulkliige

Tekitava hulkliikme omadused

1. Tekitav hulkliige rahuldab võrrandit:

$g(z)h(z) = z^n - 1$

$h(z)$ — Kontrollhulkliige

Tekitava hulkliikme omadused

2. **Tekitav hulkliige on taandamatu korpuses GF(2).**

See tähendab, et tema juurteks ehk nullkohtadeks ei ole ei 0 ega ka 1

$g_3(z) = z^2 + z + 1$

$0 : g_3(z=0) = 0^2 + 0 + 1 \neq 0$

$1 : g_3(z=1) = 1^2 + 1 + 1 \neq 0$

Tekitava hulkliikme omadused

3. **Tekitav hulkliige on normeeritud**

$$g_3(z) = z^3 + z + 1$$

Alguses ja lõpus on väärtused 1

Tekitava hulkliikme omadused

4. **Tekitava hulkliikme kaal on mitte väiksem kui vigade parandamiseks/avastamiseks vajalik koodi minimaalne kaugus st., et:**

$$w(g_r(z)) \geq d_{\min}$$

Tekitava hulkliikme omadused

5. **Tekitavaks hulkliikmeks võib olla mingi hulkliikme (z^n - 1) ühiskordsetest, mis rahuldab eelpool esitatuid tingimusi.**

$$g_1(z) \cdot g_2(z) \cdot \ldots\ldots \cdot g_b(z) = z^n - 1$$

Näide 1

- Eraldamatu tsükkelkood ühekordsete vigade parandamiseks
- n = 7 = 4 + 3

$$z^7 + 1 = (z+1)(z^3 + z + 1)(z^3 + z^2 + 1)$$

Märkus

- **Kui valida teine tekitav hulkliige, saame teise koodi. Aga igal juhul on tsükkelkood lineaarne kood: st. kui mingi koodsõna on lubatud, siis on lubatud ka temaga tsüklilised nihutatud koodsõnad ja nende lineaarsed kombinatsioonid.**

Eraldamatu tsükkelkoodi mõned lubatud koodsõnad

Infokood $[x_3(z)]$	Lubatud koodsõna $[y_6(z)]$	Lubatud koodsõna hulkliige
v.-j. 0000 m.-j.	0000000	
0001	0001011	$z^3 + z + 1$
0010	0010110	$z^4 + z^2 + z$

Näide 2

- Eraldamatu tsükkelkood (7,4) tekitava maatriksi G abil.

$$\|G\| = \begin{matrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \end{matrix}$$

Märkus

- Tuletame meelde, et tekitava maatriksi **G** kõik read on lubatud koodsõnad ja samuti kõik koodsõnad, mis on moodustatud tekitava maatriksi **G** ridade summadena, kui liitmine toimub positsiooniliselt *modulo* 2 ja liidetavate arv on 2 kuni 4.

Märkus

- Analüüsides tekitavat maatriksit on ilmne, et eespool antud maatriksiga **G** tekitatud kood on eraldamatu.
- Lubatud koodsõnad **Y** moodustamise algoritm on

$$\mathbf{Y} = \mathbf{X} \times \mathbf{G}$$

kus **X** on suvaline **k** -mõõtmeline infovektor:

Eraldatava tsükkelkoodi koostamine

- **a) Mistahes suvalise infokoodi hulkliige korrutatakse:**

$$x_{k-1}(z) \cdot z^r$$

Eraldatava tsükkelkoodi koostamine

- **b) Nihutatud infokoodi hulkliige jagatakse tekitava hulkliikmega $g_r(z)$:**

$$x_{k-1}(z) \cdot \frac{z^r}{g_r(z)} = Q(z) + R_{r-1}(z)$$

Eraldatava tsükkelkoodi koostamine

- **c) Saadud jääk liidetakse nihutatud koodsõnale ja saadakse lubatud koodsõna:**

$$y_{n-1}(z) = x_{k-1}(z) z^r + R_{r-1}(z)$$

Märkus

- Tsükkelkoodidel on head vigasid parandavad omadused. Kui kasutada dekodeerimisel sündroomi, siis praktiliselt peaks see toimuma nii, et igale konkreetsele sündroomi koodile vastab konkreetne viga. Selline dekodeerimine ei sobi mitmekordsete vigade parandamiseks.

Üldist

- Tsükkelkoodid võivad parandada nii
  - Juhuslikke vigu
  - BCH koodid; RS koodid; Goppa (krüptograafias) koodid; ruut-resiidsed koodid .....
  - Vigade pakette
  - Fire koodid

Üldist

- Tsükkelkoodid on
  - Primitiivsed
  - Mitteprimitiivsed
  - Lühendatud
  - Laiendatud
  - Lihtjuursed
  - Korduvjuursed

Lihtsate tsükkelkoodide dekodeerimine.

- Dekodeerida võib väga erinevalt.
- Kasutame lähenemisviisi.

$$y_6(z) = \quad 1\ 0\ 1\ 0\ 0\ 1\ 1$$
$$+$$
$$e_v(z) = \quad 0\ 0\ 1\ 0\ 0\ 0\ 0$$
$$y_6^*(z) = \quad 1\ 0\ 0\ 0\ 0\ 1\ 1$$

Lähenemisviis

- Vastuvõetud (vigasest) koodsõnast leiame veavektori hinnangu

$$e_v^*(z) = \Psi\{y_{n-1}^*\}$$

Parandatud koodsõna leiame

- Algoritmiga

$$y_{n-1}^{**}(z) = y_{n-1}^*(z) + e_v^*(z) =$$
$$= y_{n-1}^*(z) + \Psi\{y_{n-1}^*\}$$

1) Dekodeerimine tekitava hulkliikme $g_r(z)$ abil.

- a) Vastuvõetud koodsõna $y_{n-1}^*(z)$ jagatakse tekitava hulkliikmega $g_r(z)$:

$$y_{n-1}^*(z) : g_r(z) = c_{r-1}(z)$$
$$(y_{n-1}(z) + e_v(z)) : g_r(z) =$$
$$y_{n-1}(z) : g_r(z) + e_v(z) : g_r(z) =$$
$$= e_v(z) : g_r(z), \text{ kuna } y_{n-1}(z) : g_r(z) = 0$$
$$c_{r-1}(z)$$

Sündroom

- Sõltub ainult veakoodist, kuid mitte edastatavast lubatud koodsõnast

$$r_{r-1}(z) = c_v(z) = \psi(e_v(z))$$

Parandamise etapp

- b) Saadud jäägikoodi järgi leitakse veakoodi hinnang, mida kasutatakse paranduskoodina:

$$e_v^*(z) = \psi^{-1}\left(r_{r-1}(z)\right)$$

kuna

$$e_v(z) \underset{\text{ühene}}{\Leftrightarrow} r_{r-1}(z)$$

Paranduse etapp

kuna

$$e_v(z) \underset{\text{ühene}}{\Leftrightarrow} r_{r-1}(z),$$

siis on selline $\psi^{-1}\left(r_{r-1}(z)\right)$

$$e_v^*(z) = \psi^{-1}\left(r_{r-1}(z)\right)$$

Parandamise etapp

- c) Vastuvõetud koodsõna $\mathbf{y_{n-1}^*(z)}$ parandatakse:

$$y_{n-1}^{**}(z) = y_{n-1}^*(z) + e_v^* = y_{n-1}(z) + e_v(z) + e_v^*(z)$$

kui

$$e_v^*(z) + e_v(z) = 0$$

siis

$$y_{n-1}^{**}(z) = y_{n-1}(z)$$

2) Dekodeerimine tekitava maatriksiga G

- a) Sündroomi leiame kas nii

$$C_1 = Y^* \times G = (Y + E) \times G =$$
$$= E \times G,$$

kuna

$$Y \times G = 0$$

2) Dekodeerimine kontrollmaatriksiga H :

- a) või nii

$$C_1 = Y^* \times H^T = (Y + E) \times H^T =$$
$$= E \times H^T,$$
kuna
$$Y \times H^T = 0$$

Parandamise etapid

- **b) parandusvektori leiame**

$$E^* = \Omega\{C_1\}$$

- **C) parandame**

$$Y^{**} = Y^* + E^*$$

4) Dekodeerimine kontrollides etteantud juure sobivust.

- On teada, et iga lubatud koodsõna jagub jäägita tekitava hulkliikmega.
  - Järelikult tekitava hulkliikme juured (nullkohad) on ka lubatud koodsõnade juurteks.

Tekitava hulkliikme juured

β^1
on $g_r(z)$ juur, kui
$g_r(z = \beta^1) = 0$

Juurte üldarv on võrdne r (hulkliikme aste)

Tekitava hulkliikme juured

- Kui hulkliikme kordajad kuuluvad lõplikku korpusesse, $GF(p)$

või laiendatud lõplikku korpusesse $GF(p^m)$
siis juurte leidmine üldjuhul on probleemiks

Tekitava hulkliikme juured

Kui aga lõplikud korpused on moodustatud p = 2, st, et lõplik korpus $GF(2)$

lõplik laiendatud korpus $GF(2^m)$

Siis on juurte leidmine lihtsam

Tekitava hulkliikme omadus

- Tsükkelkoodi jaoks
  - Kui koodipikkus on primitiivne, st

$$n = 2^m - 1$$

  - Siis tekitava hulkliikme juured kuuluvad korpusesse $GF(2^m)$

Märkus

- Kodeerimiseks võib valida tekitava hulkliikme nii, et tema juured oleksid laiendatud lõpliku korpuse elemendid

$GF(2^m)$

Dekodeerimiseks

- Kasutame seda, et
  - Kui koodsõna on vigane, siis ilmselt tekitava hulkliikme juured ei kindlusta vigase koodsõna nulliga võrdset väärtust

Sündroomi võime saada

$$C_1 = y^*\left(z = \beta_i\right)$$

(β_i) On tekitava hulkliikme mingi juur

Kuna tekitav hulkliige pole mitte igasugune, vaid eriliste omadustega, siis on eelnevalt teada ka , millised on tema juured

Lõpliku korpuse $GF\left(2^m\right)$ elemendid

- On m kahendsümbolist koosnevad arvud
  - Nende koguarv on $2^m - 1$
  - Ainult nullidest koosnevat elementi ei arvestata

$GF\left(2^m\right)$ Peab korrastama (panema järjekorda) nii, et iga järgmine element on eelmise elemendi ja mingi ühe valitud korrutis.

Näide

- Olgu lõplik korpus $\mathbf{GF}\left(2^3\right)$
- Tähistame elemendid

$\boldsymbol{\beta}^0$ $\boldsymbol{\beta}^1$ $\boldsymbol{\beta}^2$ $\boldsymbol{\beta}^3$ $\boldsymbol{\beta}^4$ $\boldsymbol{\beta}^5$ $\boldsymbol{\beta}^6$

Elementide jada pikendamine

β -jada saab lõpmatult pikendada:

$\beta^7 = \beta^0$

$(11) mod\ 7 = 4$

11 jagame 7 ga ja tulemuseks võtame jäägi 4

Lõpliku korpuse elementide korrastamisest

Igal elemendil on minimaalne hulkliige, mille aste ei ole suurem kui m

Mingi elemendi minimaalne hulkliige on vähima astmega hulkliige, mille juureks see element on

Lõpliku korpuse elementide korrastamiseks

- Kasutatakse primitiivset elementi ja tema minimaalset hulkliiget
- Mistahes m korral võib laiendatud lõplikus korpuses olla mitu primitiivset elementi ja puuduvad üldised reeglid, kuidas neid leida.

10. BCH koodide koostamine

- Eriti tähtsa häirekindlate koodide klassi moodustavad BCH koodid, mis on mitmekordseid vigu avastavad ja parandavad hulkliikmelised koodid. Nimetatud koode võib pidada *Hamming´*i koodide üldistuseks mitmekordsete vigade parandamiseks.

BCH koodide ajaloost

- Koodid on avastatud aastatel 1959 ja 1960 sõltumatult kolme erineva autori
- *Hocquenghem*'i, UK
- *Bose* ja USA
- *Chaudhuri*'i poolt. India
- Nende autorite nimede esimestest tähtedest on moodustatud ka koodi üldnimetus: BCH koodid.

BCH koodide uurijad

- Koodide tsüklilist struktuuri on põhjalikult uurinud

Peterson.

- BCH koode pikkusega **n** korpuses **GF(p^m) ,** kus **p** on lihtarv, on uurinud

Gorenstein ja

Zierler.

BCH koodide liigid

BCH koodid võivad olla

- binaarsed (kahendkoodid) või
- mittebinaarsed. Levinumad mittebinaarsed BCH koodid on Reed – Solomoni (RS) koodid, mis on sõltumatult avastatud ja uuritud 1960. a.

Reed'i ja

Solomon'i poolt.

BCH koodide kasutusest

- BCH koode kasutatakse laialdaselt

Arvutustehnikas (arvutid, nii vanemad kui ka uuemad on üleküllastatud binaarsetest BCH koodidest)

Andmeedastusel, infosalvestamisel, lennundussides ja ka meresides ning merepäästeteenistuses (nii BCH kui ka RS koodid)

Märkus

- Mitmekordsete vigade parandamine on raskendatud erinevate mitmekordsete vigade suure arvu tõttu, mis välistab otsese vigade paranduse kontrollarvu järgi
- Kuna sümboli algsed vigasuse tõenäosused on väikesed, esinevad ühekordsed vead koodiplokkides kõige suurema tõenäosusega.

Mitmekordsed vead

- Erinevate veakoodide arv

$$Q = C_n^q = \frac{n!}{q!\cdot(n-q)!} = \frac{1\cdot 2\cdot 3\cdots n}{1\cdot 2\cdot 3\cdots q \times 1\cdot 2\cdot 3\cdots (n-q)}$$

Näide

- **N = 31, q = 2, Q = 465**
- **N = 31, q = 3, Q = 3495**
- **N = 31, q = 4, Q = 13485**

Erinevate vigade arv kasvab koodiploki pikkuse kasvades väga kiiresti.

Praktiliselt kasutatavad koodisõna pikkused on kuni n = 127, parandatakse vigu kuni kordsusega q = 3;4.....

Järeldus

- Kodeerima peab nii, et oleks võimalik kasutada algebralisi dekodeerimismeetodeid
- BCH koodid on lineaarsed algebralised koodid.

A. BCH koodide kodeerimine

- BCH koodid koostatakse parandama vigu kuni kordsusega q
  - See tähendab, te parandatakse kõik vead kuni kordsusega q kaasa arvatud.
  - Olgu q = 3.
  - See tähendab, et parandatakse kõik vead kordsusega 1, kordsusega 2 ja kordsusega 3

Märkus

- Vastuvõtu poolel ei ole teada
  - Vigaste sümbolite asukohad
  - Vea reaalne kordsus
- Järelikult peaks dekodeerimise käigus saama lisainfot vea reaalsest kordsusest

Kodeerimisvõimalus üldiselt

- Kui BCH kood on koostatud **kuni q** - kordsete vigade parandamiseks, siis peab algebraliste dekodeerimisalgoritmide võimalikkuseks olema tagatud **q** sellise võrrandi koostamise võimalus, et võrrandisüsteemi lahendamisega oleks kindlustatud **q** tundmatu väärtuste määramine.
- Need väärtused on vigaste sümbolite asukohtade numbrid.

Kodeerimisvõimalus üldiselt

- Vigaste sümbolite asukohtade arvulised väärtused sõltuvad sellest, millise nummerdusega on fikseeritud vigaste koodisümbolite asukoht.
- Võrrandid ei tohi olla omavahel lineaarselt sõltuvad ja samuti ei tohi nende vahel olla spetsiifilist lõpliku korpuse tüübist tingitud sõltuvust.

Kodeerimisvõimalus üldiselt

- **q** võrrandi moodustamiseks oleks tarvilik kontrollida, kas etteantud kahendarvude hulgas on vastuvõetud koodsõna juured või ei ole. Sobivate juurte leidmiseks võiks esialgselt võrrandeid tekitada rohkem kui **q**, näiteks **2q**

Kodeerimisvõimalusest üldse

- BCH koodid kodeeritakse nii nagu tavalised ühekordseid vigu parandavad tsüklilised koodi
  - A. Eraldamatu koodina
  - B. Eraldatav koodina

Dekodeerimiseks vajalikud võrrandid saame siis, kui on valitud vastavate omadustega tekitav hulkliige.

Kodeerimisvõimalus üldiselt

- Kasutame lubatud koodsõnade moodustamiseks laiendatud lõplikku korpust $GF\left(2^m\right)$

Selle korpuse elemendid moodustavad multiplikatiivse tsüklilise rühma

Tuletame meelde, et tekitav polünoom moodustatakse selle lõpliku korpuse abil

Tekitava hulkliikme valime $GF(2^m)$

- Abil nii, et tema juurteks oleksid

$$\beta^1, \beta^2, ..., \beta^{2q}$$

β on $GF(2^m)$ **element**

Primitiivne ja mitteprimitiivne kood.

- Kui **β** on mingi lõpliku korpuse primitiivne element, st. **β = γ** , siis on võimalik tekitada primitiivne kood, vastasel juhul on kood mitteprimitiivne.
- Igal juhul on primitiivne kood olemas, kui koodi pikkus on primitiivne.

Tekitav hulkliige

- On kuni q- kordsete vigade parandamiseks

$g_r(z)$

Tekitav hulkliige kindlustab koodi minimaalse koodikauguse d_{min} (BCH koodide puhul konstruktiivne koodikaugus)

Tekitava hulkliikme aste ja minimaalne koodikaugus on BCH koodidel suurem konstruktiivsest

Kodeerimistehted

- **1) Valitakse sobiv koodipikkus $n = 2^m - 1$**
- **2) Sõltuvalt vajadustest määratakse, parandatavate vigade kordsus q**
- **3) Leitakse korpuse $GF(2^m)$ primitiivne element β**
- **4) Leitakse korpuse $GF(2^m)$ primitiivse elemendi β minimaalne hulkliige M(z)**

Kodeerimistehted

- 5) Korrastatakse korpuse **GF(2^m)** elemendid β^i,

i = [0, ..., 2^m - 2] primitiivse elemendi **β** minimaalse hulkliikme **M(z)** abil

Elementide koguarv on $2^m - 1$

Tekitav hulkliige

- 6) Moodustatakse tekitav hulkliige $\mathbf{g_r(z)}$:

$\mathbf{g_r(z)} =$
$\mathbf{VÜK\{M_\eta(z) \times M_{\eta+1}(z) \times ... \times M_{\eta+2l}(z)\}}$,

kus **VÜK** on vähim ühiskordne

Tekitav hulkliige

- $\mathbf{M_\eta(z)}$ on primitiivse elemendi β **η-** astme

β **η** minimaalne hulkliige,

- $\mathbf{M_i(z)}$, **i = [1, ... , 2l]** on **i** -inda elemendi β^i

minimaalne hulkliige

Primitiivne BCH kood

- Primitiivse BCH koodi jaoks on **η** = 1.
- Kui kood koostatakse parandamaks kuni **q-** kordseid vigu, peab tekitav hulkliige koosnema **q** komponendist. Selliselt valitud tekitavate hulkliikmetega koodid, mis parandavad **q** ja vähemakordseid vigu, kindlustavad ka vajaliku minimaalse koodkauguse

Kodeerimistehted

- 7) Leitakse infosümbolite arv:
  - Kui tekitav hulkliige on leitud, saame ka tekitava hulkliikme astme r: $g_r(z)$

Infosümbolite arv saadakse

Valemist

$k = n - r$

Märkus

- Juurtevahelised seosed ongi eriline arvukorpusest GF(2) tingitud sõltuvus
  - Kehtib ainult kahendkoodide puhul
  - St ainult siis, kui koodsõna kordajad (sümbolid) kuuluvad korpusesse GF(2)
- Laiendatud korpuse elementidel on samad minimaalsed hulkliikmed

BCH koodide dekodeerimine

- BCH koodi lubatud koodsõnade omadused

on dekodeerimise aluseks

BCH koodi lubatud koodsõnade omadused

- 1. Kõik lubatud koodsõnad jaguvad jäägita tekitava hulkliikmega $\mathbf{g_r(z)}$
  - $g_r(z)$ on q komponenti
- 2. Kõik tekitava hulkliikme $\mathbf{g_r(z)}$ juured on ka kõikide lubatud koodsõnade juurteks.
  - $g_r(z)$ juurte arv on võrdne parandatavate vigade arvuga q

BCH koodi lubatud koodsõnade omadused

- 3. BCH koodi dekodeerimisel on võimalik moodustada sündroom (e. kontrollarv), mis koosneb **q** komponendist.
- 4. BCH koodi dekodeerimisega on võimalik parandada ja avastada kõiki **q** ja vähemakordseid vigu, mis rikuvad lubatud koodsõnade omadusi 1. ja 2.

Dekodeerimise tehted

BCH koodide dekodeerimise peamised tehteid on kaks:

- A. Sündroomi (e. kontrollarvu) komponentide moodustamine

A. Kontrollarvu komponentide leidmine

- A. Leiame vastuvõetud koodsõna

$$y^*(z) = y(z) + e(z)$$

väärtused oletatavate lubatud koodsõnade juurte

$\beta^1\ \beta^3\ \beta^5 \ldots \beta^{2q-1}$ kohal:

Kontrollarvu komponendid

$$y^*(z = \beta^1) = C_1$$
$$y^*(z = \beta^3) = C_3$$
$$y^*(z = \beta^5) = C_5$$
$$\ldots\ldots\ldots\ldots$$
$$y^*(z = \beta^{2q-1}) = C_{2q-1}$$

Dekodeerimise tehted

- B. Vigaste sümbolite asukohtade määramine.

Olgu vigaste sümbolite asukohad kooskõlas vastava korpuse **GF(2^m)** elementide korrastatusega vastavalt

$\beta_1 \beta_2 \beta_3 \dots \beta_q$

Vigaste sümbolite asukoha võrrandid

Kuna vastuvõetud koodsõna on

$y^*(z) = y(z) + e(z)$, siis

Võrrandite süsteem

$$\beta_1 + \beta_2 + \beta_3 + \dots + \beta_q = C_1$$

$$\beta_1^3 + \beta_2^3 + \beta_3^3 + \dots + \beta_q^3 = C_3$$

$$\beta_1^5 + \beta_2^5 + \beta_3^5 + \dots + \beta_q^5 = C_5$$

$$\dots\dots\dots\dots\dots\dots\dots\dots = \dots\dots$$

$$\beta_1^{2q-1} + \beta_2^{2q-1} + \beta_3^{2q-1} + \dots + \beta_q^{2q-1} = C_{2q-1}$$

Märkus

- Võrrandite süsteem on mittelineaarne
- Üldkujul võrrandisüsteemi lahendite leidmine on raskendatud
  - Kahest võrrandist koosnevale süsteemile on lahendus olemas
- Kui seda võrrandite süsteemi saaks muuta *Vandermondi* võrrandite süsteemiks, siis on lahendus olemas.

Vigaste sümbolite asukohtade määramise

- Võrrandsüsteem maatrikskujul:

$$\mathbf{V_\beta} \times \mathbf{I^T} = \mathbf{C_{+0}}$$

V_β on *Vandermonde* i maatriks

$$\begin{matrix} 1 & 1 & . & 1 \\ \beta_1 & \beta_2 & . & \beta_i \\ \beta_1^2 & \beta_2^2 & . & \beta_i^2 \\ \beta_1^3 & \beta_2^3 & . & \beta_i^3 \\ .. & .. & .. & .. \\ \beta_1^{2q} & \beta_2^{2q} & . & \beta_i^{2q} \end{matrix}$$

Märkus

- *Vandermonde* i maatriksis on lisatud esimene rida, mis koosneb ühtedest ja veel vigaste sümbolite asukohtade paarisastmete summad

Võrrandite süsteemi osad

- $\mathbf{I^T}$ on transponeeritud ühikvektor ehk ühtedest koosnev veerg - maatriks.
- $\mathbf{C_{+0}}$ on kontrollarvu komponendid veerg - maatriksina, millistele on lisatud esimeseks elemendiks $\mathbf{C_0}$

$$y * \left(z = \beta^0\right) = C_0$$

Vigaste sümbolite asukohad

1	1	.	1	×	1	= C_0
β_1	β_2	.	β_q		1	C_1
β_1^2	β_2^2	.	β_q^2		1	C_2
β_1^3	β_2^3	.	β_q^3		1	C_3
..	..	..	..		1	..
β_1^{2q}	β_2^{2q}	.	β_q^{2q}		1	C_{2q}

$C_2 = C_1^2,$
jne...

Võrrandsüsteemi lahendus

- Üks nimetatud võrrandsüsteemi võimalikest lahendustest oleks selline, kus lahendused leitakse nn. lokaatorhulkliikmete juurtena.

$C_2 = C_1^2,$
jne...

Lokaatorhulkliikmed

- Lokaatorhulkliiget võib moodusta kahel erineval kujul.

A)

$$\sigma_q(z) = z^q + \sigma_1 z^{q-1} + \sigma_2 z^{q-2} + \ldots + \sigma_{q-1} z^1 + \sigma_q$$

Vigaste sümbolite asukohad on lokaatorhulkliikme juured:

$\sigma_q(z) = 0$, indeks **i**

Lokaatorhulkliikmed

- Lokaatorhulkliiget võib moodusta kahel erineval kujul

B)

$$\lambda_q(z) = \lambda_q z^q + \lambda_{q-1} z^{q-1} + \lambda_{q-2} z^{q-2} + \ldots + \lambda_1 z^1 + 1$$

Vigaste sümbolite asukohad on lokaatorhulkliikme juured:

$\lambda_q(z) = 0$, indeks **j**

A) Lokaatorhulkliige

- Lokaatorhulkliikme kordajad $\sigma_1, \sigma_2, \sigma_3, \ldots \sigma_q$, on leitud **q** võrrandist

$$C_j \times \sigma_q + C_{j+1} \times \sigma_{q-1} + \ldots + C_{j+q-1} \times \sigma_1 + C_{j+q} = 0$$

kus antud primitiivse koodi jaoks **j** muutub piirides

$$1 \le j \le q$$

Vigaste sümbolite asukoht

- on määratud järjekorra numbriga, alustades nummerdamist kokkuleppelisest noorimast järgust $i \subset [\ 0$ kuni $\mathbf{n-1}\ ]$

$$C_1 \times \sigma_q + C_2 \times \sigma_{q-1} + \ldots + C_q \times \sigma_1 = C_{q+1},\ j = 1$$
$$C_2 \times \sigma_q + C_3 \times \sigma_{q-1} + \ldots + C_{q+1} \times \sigma_1 = C_{q+2},\ j = 2$$
$$C_3 \times \sigma_q + C_4 \times \sigma_{q-1} + \ldots + C_{q+2} \times \sigma_1 = C_{q+3},\ j = 3$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$C_q \times \sigma_1 + C_{q+1} \times \sigma_{1-1} + \ldots + C_{2q-1} \times \sigma_q = C_{2q},\ j = q$$

Lokaatorhulkliikme kordajad

- Leiame võrrandist

$$\mathbf{C} \times \boldsymbol{\sigma}^T = \mathbf{C}_{+1}$$

kus:

C on kontrollarvu komponentidest koostatud maatriks,

$\boldsymbol{\sigma}^T$ on lokaatorhulkliikme kordajate veerg - maatriks

- $\mathbf{C}_{+1}$ on kontrollarvu järjestikused komponendid alustades $\mathbf{C}_{q+1}$

Reaalne vea kordsus

Esiteks püüame parandada viga kordsusega q
Kui reaalne tegelik viga, mille kordsust me ei tea on

$$q_r = q$$

Siis $\det |C| \neq 0$

Ja lokaatorhulkliikme kordajad on võimalik leida

Reaalne vea kordsus

- Kui **det | C| = 0**, siis

On reaalne vea kordsus $q_r < q$

ja võrrandsüsteem lokaatorhulkliikme kordajateks leidmiseks ei lahendu üheselt

Sündroomi komponentide maatriksi vähendamine

võrrandite maatriks $\mathbf{C_{q-1}}$, kus **q** on asendatud **(q - 1)**

$$\begin{matrix} C_1 & C_2 & .. & C_q \\ C_2 & C_3 & .. & C_{q+1} \\ .. & .. & .. & .. \\ C_q & C_{q+1} & .. & C_{2q} \end{matrix}$$

Kui vähendatud maatriks jaoks

- determinant võrdub nulliga:

det | C_{l-1} | = 0 , siis vähendada veel jne kuni determinant enam nulliga ei võrdu

- See tähendab, et oletatav veakordsus **(q - v)** **v** -endal maatriksi $\mathbf{C_{q-v}}$ moodustamise taktil **on** võrdne reaalse vea kordsusega

($q - v = q_r$)

11. RS koodid

- *Reed- Solomon* 'i (ehk RS) koodide hulkliikmete kordajad kuuluvad korpusesse

GF(2^m)

Koodi hulkliikmete koostamiseks kasutatakse korpuse **GF(2^m)** korrastatud elemente

Lõplik laiendatud korpus GF(2^m)

- Korpuse **GF(2^m)** elemendid korrastatakse mingi **m** -astmelise hulkliikmega (tavaliselt korpuse **GF(2^m)** primitiivse elemendi minimaalse hulkliikmega)

Koodi hulkliikmete kordajad

- *Reed - Solomon* 'i koodide hulkliikmete kordajad on määratud korpuse **GF(2^m)** elementidega.
- Need on **m** kahendsümbolist koosnevad vektorid β^i, nummerdatud arvudega

$i \subset [0, ..., 2^m - 2]$

Koodi hulkliikmed

- Moodustud **GF(2^4)** abil

1 0 0 1	1 0 1 1	1 0 0 1	0 1 0 1
n.j.			
$\beta^{14} \times z^0 +$	$\beta^{13} \times z^1 +$	$\beta^{14} \times z^2 +$	$\beta^9 \times z^3$

Veakordsusest RS koodide puhul

- Antud koodide puhul mõistetakse vigade kordsust Q järgmiselt:

Ühekordne viga **(Q = 1)** on siis, kui vigane on üks koodsõna hulkliikme kordaja, st. viga on ühes plokis.

Kahekordse vea puhul **(Q = 2)** on koodsõna hulkliikmel vigased kaks kordajat, st., et viga on kahes plokis jne..

Veakordsusest RS koodide puhul

- Kui (kordaja) ploki pikkus on **m**, siis vigane kordaja tähendab seda, et selles plokis võivad olla vigased ükskõik millised kahendsümbolid, kasvõi kõik.
- Erinevalt kahendkoodidest peab nende koodide puhul ära näitama ka vea suuruse.

Reed - Solomon 'i koodide koostamine

- Primitiivne koodipikkus on

$\mathbf{N = 2^m - 1 = K + R}$, kus

K on infoplokkide arv ja

R on liiaste plokkide arv,

maksimaalne koodkaugus $\mathbf{D_0 = R + 1}$

RS koodide tehted

- Koodi hulkliikmete aritmeetiliste tehete teostamisel kasutatakse korpuse $\mathbf{GF(2^m)}$ elementide korrutamise ja liitmise reegleid.
- Alljärgnevalt kasutame kõigi näidiste puhul lihtsamat korpust $\mathbf{GF(2^3)}$.
- Korpuse $\mathbf{GF(2^3)}$ elemendid, korrastatud hulkliikmega $\mathbf{(z^3 + z + 1)}$

Koodikaugused

- Koodikaugus **D = 2**, tähendab seda, et kaks koodsõna erinevad kahes plokis ehk kahe kordaja suhtes

Koodiplokkide jaotus

- RS -koodide puhul võib jaotust

K + R = N

teostada erinevalt: võib valida kõiki **K** väärtusi

K = [1, ..., N].

Saame erinevate vigasid avastavate ja parandavate omadustega RS - koodid.

Mitmekordsete vigade parandamine

- Vastavalt valitud **K** väärtusele võib kindlustada **R = 2Q** ,

kus **Q** on vea kordsus

sellise RS koodiga võimalik parandada

Q -kordseid vigu.

Selline RS kood kindlustab maksimaalse koodikauguse **D = 2Q + 1**.

- Kui kood on koostatud **Q** -kordsete vigade parandamiseks, siis nimetatakse seda **Q** väärtust konstruktiivseks,

Tegelikult reaalne veakordsus **T** võib olla väiksem, kui **Q (T ≤ Q)**

RS kood parandab kõik vead kuni kordsusega **Q**

Kodeerimistehted

- 1) valitakse tekitav hulkliige;

$\mathbf{G_{2Q}(z) = (z + \beta^{\eta}) (z + \beta^{\eta+1}) ..... (z + \beta^{\eta+2Q-1})}$

kus η on miski arv $\eta \subset [0, 1, 2, ...]$

Tavaliselt on $\eta = 1$;

lubatud on ka $\eta = 0$, vahel ka näiteks $\eta = 12$

.

Märkus

- Järgima peab seda, et alati oleks vigasid parandava RS koodi tekitava hulkliikme $\mathbf{G_{2Q}(z)}$ aste võrdne **2Q**

Eraldamatu RS kood

- Kui RS -koodi tekitav hulkliige on koostatud, võib infokoodi

$\mathbf{X_{k-1}(z)}$

jaoks lubatud koodsõna

$\mathbf{Y_{n-1}(z)}$ leida algoritmiga

$\mathbf{Y_{n-1}(z) = X_{k-1}(z)\, G_r(z)}$

Eraldatav RS kood

-
- Tsüklilise koodi eraldatav algoritm

1. Infokoodi nihutamise **2Q = 4** sümboli (ploki) võrra vanemate järkude suunas:

RS koodide dekodeerimine.

- RS -koodide dekodeerimine toimub etappide kaupa, millest esimesed etapid langevad kokku BCH koodide dekodeerimise etappidega.

RS koodide dekodeerimise etapid

- I. Koodsõna $\mathbf{Y^*_{N-1}(z)}$ vastuvõtmine.

Kui $\mathbf{Y^*_{N-1}(z)}$ vastuvõtmisega kaasneb nn. paralleelkoodi moodustamine,
peab arvestama, et
iga RS -koodi koodsümbol vajab **m** -järgulist registrit

RS koodide dekodeerimise etapid

- II. Kontrollarvu komponentide leidmine.

RS -koodide tekitavad **2Q** -astmelised hulkliikmed $\mathbf{G_{2Q}(z)}$ on koostatud nii,et nendel hulkliikmetel on **2Q** juurt. Järelikult saame RS - koodide jaoks **2Q** kontrollarvu komponenti.

Sündroomi komponentide leidmine

$$\mathbf{Y^*_{N-1}(z=\beta^{\eta})} = \mathbf{C_{\eta}}$$
$$\mathbf{Y^*_{N-1}(z=\beta^{\eta+1})} = \mathbf{C_{\eta+1}}$$
$$\mathbf{Y^*_{n-1}(z=\beta^{\eta+2}} = \mathbf{C_{\eta+2}}$$
$$\cdots\cdots\cdots\cdots$$
$$\mathbf{Y^*_{N-1}(z=\beta^{\eta+2Q-1})} = \mathbf{C_{\eta+2Q-1}}$$

RS koodide dekodeerimise etapid

- III. Kontrollarvu komponentide maatriksi moodustamine
  - Reaalset vea kordsust $\mathbf{Q_r}$ vastuvõtu poolel ei teata. Tuletada otsest algoritmi reaalse veakordsuse määramiseks ei õnnestu.
  - Kui reaalne vea kordus on väiksem, kui see, millisele kood on koostatud, kajastub see kontrollarvu komponentides

Sündroomi komponentide maatriks:

		C_1	C_2	.....	C_Q
		C_2	C_3	.....	C_{Q+1}
C	=	C_3	C_1	.....	C_{Q+2}
		.....	.....	.....	
		C_Q	C_{Q+1}	.....	C_{2Q-1}

Reaalne veakordsus

- **Kui det $|C| \neq 0$**

võib edasi minna järgmisele dekodeerimise etapile

Veakordsus

- Kui aga ilmneb, et

det $|C| = 0$, siis peab koostama uue vähendatud komponent-maatriksi

Sarnaselt BCH koodide dekodeerimisele

$\mathbf{C_{-1}}$

Reaalne veakordsus

- Järgmisena leitakse uue maatriksi $\mathbf{C}_{-1}$ determinant $\det|\mathbf{C}_{-1}|$ ja kontrollitakse, kas

$\det|\mathbf{C}_{-2}| \neq \mathbf{0}$ jne..

Reaalne veakordsus

- Kui reaalne veakordsus $\mathbf{Q}_r$ osutub võrdseks konstruktiivsega $\mathbf{Q} - \varsigma$ ja see juhtub varem või hiljem mingil ς -endal korral, osutub, et kas $\det|\mathbf{C}_{-\varsigma}| \neq \mathbf{0}$ või maatriks $\mathbf{C}_{-\varsigma}$ osutub tühjaks, st. $\mathbf{Q} - \varsigma = \mathbf{0}$. Võrdsus $\mathbf{Q} - \varsigma = \mathbf{0}$ osutab sellele, et veakordsus on võrdne $\mathbf{0}$
- vastuvõetud koodsõna on vigadeta kui kõik sündroomi komponendid on 0

RS koodide dekodeerimise etapid

- IV Lokaatorhulkliikme moodustamine. Samaselt BCH koodide dekodeerimisega moodustatakse lokaatorhulkliige kas kujul

$\boldsymbol{\sigma}_{\mathbf{Q}--\varsigma}(\mathbf{z})$ või $\boldsymbol{\lambda}_{\mathbf{Q}---\varsigma}(z)$

RS koodide dekodeerimise etapid

- V Vigaste sümbolite asukoha määramine

toimub otsides lokaatorhulkliikme juuri lõplikust korpuse $\mathbf{GF(2^m)}$ elementide hulgast

RS koodide dekodeerimise etapid

- Lokaatorhulkliikme $\sigma_{Q-\varsigma}(z)$ juured annavad vigase koodiploki (sümboli) asukoha, nummerdades viimase alates noorimast koodisümbolist $i \subset [0, ..., n - 1]$ ja lokaatorhulkliikme. $\lambda_{Q-\varsigma}(z)$ juured annavad vigase koodiploki (sümboli) asukoha, alates nummerdamist vanimast koodisümbolist

$j \subset [1, ..., n]$

RS koodide dekodeerimise etapid

- VI Paranduskoodi moodustamine.

Kui vigaste sümbolite arv ja asukoht on teada, on enne parandamist tarvis teada vea väärtusi

$E_1, E_2, ..., E_{Q-\varsigma}$ nendes kohtades

Sündroomi komponentide võrrandid

$E_1 \times \beta_1 +$	$E_2 \times \beta_2 +$	...	$E_{Q-\varsigma} \times \beta_{Q-\varsigma}$	=	C_1
$E_1 \times \beta_1^2 +$	$E_2 \times \beta_2^2 +$	...	$E_{Q-\varsigma} \times \beta_{Q-\varsigma}^2$	=	C_2
$E_1 \times \beta_1^3 +$	$E_2 \times \beta_2^3 +$	...	$E_{Q-\varsigma} \times \beta_{Q-\varsigma}^3$	=	C_3
...	...	...	...	...	...
$E_1 \times \beta_1^{2Q} +$	$E_1 \times \beta_2^{2Q} +$	...	$E_{Q-\varsigma} \times \beta_{Q-\varsigma}^{2Q}$	=	C_{2Q}

RS koodide dekodeerimise etapid

- Lahendades võrrandsüsteemi saame veakoodi (veakoodi hulkliikme kordajate) väärtused:

$E^*_{N-1}(z) = [0, E_1, E_2, ..., E_{Q-\varsigma}, 0, 0, 0]$

Dekooder väljastab parandatud koodsõna:

$Y^{**}_{N-1}(z) = Y^*_{N-1}(z) + E^*_{N-1}(z)$

"Kustutusega" edastuskanalid

- RS koode võib kasutada "kustutusega" kahendkanalites. Vastuvõetud koodsõnas nullidele ja ühtedele lisaks veel "küsimärgid" (?), milliste asukohad on täpselt teada. Sellise kanali puhul ei ole vajalikud sündroomi komponentide maatriksi kontrolli ja vealokaatorite leidmise tehted

"Kustutusega" edastuskanalid

- Koodikaugus RS -koodide jaoks peab "kustutuste" taastamiseks vastama valemile:

$$\mathbf{D_{min} = 2Q + S + 1}$$

kus **S** on "kustutustega" sümbolite arv.

Märkide "**?**" arv ühes mittekahendsümbolis tähtsust ei oma

12. Ahendkoodide koostamine

- Ahendkoodid on alaliik nn pidevatest koodidest.
- Erinevalt plokk-koodidest,
  milliste järjestikuste koodiplokkide vahel puudub igasugune sõltuvus

on pideva kodeerimise puhul koodiplokid
  libisevas sõltuvuses üksteisest

- Plokid on põhimõtteliselt lühikesed

Kirjandust

- *Handbook of Coding Theory*

V. S: Pless and W. C. Huffman, volume I and II

lk. 1991 - 2105 (II)

- *R. E. Blahut Theory and Practice of Error Control Coding*
- **lk. 399. - 447.,523. - 552.**

Ahendkoodide esitusviisid

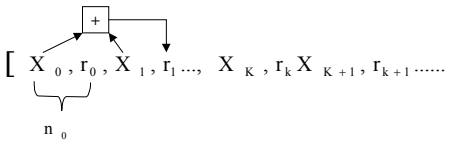
- Infovoog ja temale vastav hulkliige:

$$[\ X_0, X_1, \ldots, X_K, X_{K+1}, \ldots\ldots$$

$$[\ X_0 z^0 + X_1 z^1 + \ldots + X_K z^k + X_{K+1} z^{K+1}, \ldots\ldots$$

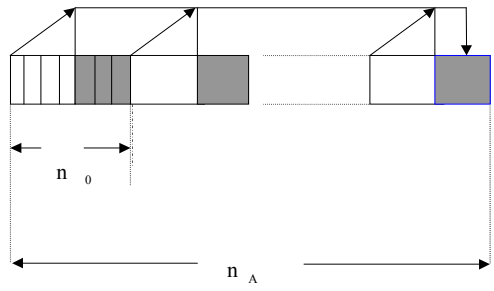
Ahendkoodi moodustamine

- Eraldatavad ahendkoodid moodustatakse nii, et infosümbolite vahele lisatakse liiased sümbolid



Koodipiirang n_A

- **on võrdne omavahel sõltuvate sümbolite arvuga**
  - **vigade parandus sõltub just nimelt koodipiirangust**



a) Ahendkoodi esitus tekitava maatriksiga

- on

TEKITAV MAATRIKS

$$Y = X \times G$$

INFOVOOG

VÄLJUNDVOOG

Avatud

- kujul

$$\left|X_1, \overset{\rightarrow K}{X_2,...,X_K}\right| \times \overset{\downarrow}{K} \begin{Vmatrix} G_{11} & ... & G_{1N} \\ ... & ... & ... \\ G_{K1} & ... & G_{KN} \end{Vmatrix} =$$

$$= X_1G_{11} + ... + X_KG_{K1}; ...; X_1G_{1N} + ... + X_KG_{KN}$$

Tekitava maatriksi

- Abil arvutatakse mitu koodipiirangut

$$\nu = \sum_{i=1}^{k_0} \max_j \left[\deg g_{ij}(z)\right]$$

$$k = k_0 \max_{ij} \left[\deg g_{ij}(z) + 1\right]$$

$$n = n_0 \max_{ij} \left[\deg g_{ij}(z) + 1\right]$$

Näide 1

- Tekitav maatriks koosneb hulkliikmetest

$$G = \|1 \quad 1 + z\|$$

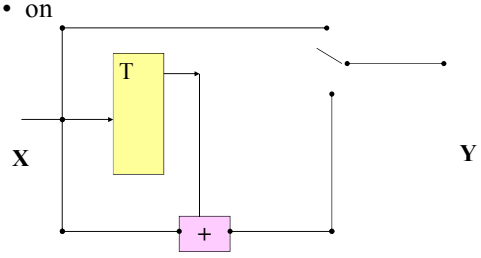
Koodiplokis on (k = 1, n = 2),

- liiasus on

$$U(K) = \frac{r_0}{n_0} = \frac{n_0 - k_0}{n_0} = \frac{1}{2}$$

Kooderi struktuurskeem

- on



Näide 2

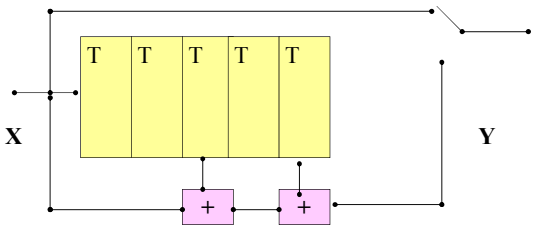
- Eraldatav kood

$$\|G(z)\| = \|1; z^5 + z^3 + 1\| \quad n_A = 2\cdot(5+1) = 12$$

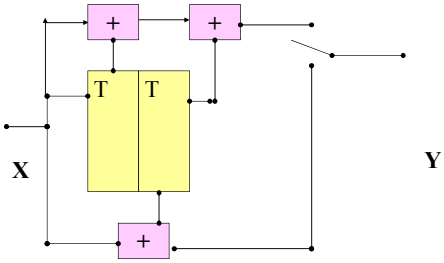
$$k_0 = 1 \qquad n_0 = 2 \qquad U(K) = \frac{r_0}{n_0} = \frac{1}{2}$$

Kooderi struktuurskeem

- on



Näide 3 (Kood (6,3))



Tekitav hulkliige

- on

$$\|G(z)\| = \|z^2 + z \;\; +1; 1 + z^2\| \quad n_A = 2\cdot(2+1) = 6$$

$$k_0 = 1 \qquad n_0 = 2 \qquad U(K) = \frac{r_0}{n_0} = \frac{1}{2}$$

Koodi omadused

- Ahendkoodis võivad vead levida pikemale koodijadale
- Koodid on sellest seisukohast lähtudes
  - mittekatastroofilised
  - katastroofilised koodid

Mittekatastroofilised koodid

- Tekitava maatriksi hulkliikmete suurim ühiskordne (SÜK) võib olla ainult

z^a, näiteks $z^0 = 1$

Katastroofilised koodid

- On siis, kui SÜJ on miski muu

$$SÜJ \neq z^a$$

- Katastroofilistes koodides vead levivad katastroofiliselt

Näide

- On antud tekitavad maatriksid

$$a)\ \left\| z^2; z^3 + z + 1 \right\|$$

$$b)\ \left\| z^4 + z^2 + 1; z^4 + z^3 + z + 1 \right\|$$

$$c)\ \left\| z^4 + z^2 + z + 1; z^4 + z^3 + 1 \right\|$$

$$d)\ \left\| z^6 + z^5 + z^4 + 1; z^5 + z^3 + z + 1 \right\|$$

Leida

- Nende hulgast katastroofilised koodid
- Kirjeldada mittekatastroofiliste koodide omadusi
  - liiasust
  - informatiivsus (infohulka ühe sümboli kohta keskmiselt)
  - koodipiiranguid

B) Ahendkoodi esitus koodipuuna

- Reeglid
  - Olgu infoploki pikkus k
  - puukujulise graafi igast sõlmest on hargnemisi 2^k
  - igale ribile vastab n sümbolist kood
  - igast sõlmest liigutakse kas üles või alla vastavalt infosümboli(te) väärtustele

Märkus

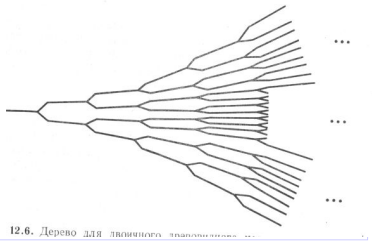
- Koodipuud pole raske teha, kuid ta peab vastama struktuurskeemile
- Parim on tema moodustumist teha ja kontrollida sammhaaval
- koodipuud saab kasutada nii eraldatavate kui ka eraldamatute ahendkoodide koostamisel
  - eraldamatut ahendkoodi nimetatakse vahel ka puu-koodiks

Koodipuu koodi (6,3) jaoks

- Igast sõlmest on hargnemisi

$$2^K$$

$$k = 1$$



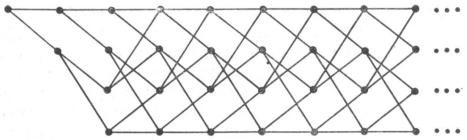
12.6. Дерево для двоичного

Koodipuu iseärasuseks on

- Alaline laienemine
  - iga uue infosümboliga koodipuu läheb oluliselt laiemaks
  - piltlikult ei mahu selline koodipuu kuskile ära
- korduvate osade olemasolu
  - Neid võib kokku keerata võrekujuliseks koodi graafiks (*trellis*)
  - Siit pärineb ka ingilise keelne nimetus *trellis code*

Koodivõre koodi (6,3) jaoks

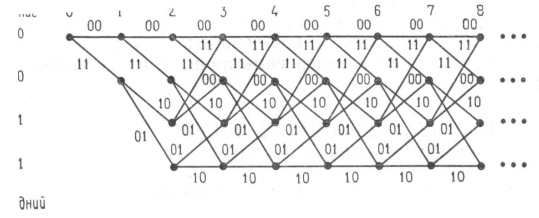
- On päris ühesugune mitmele koodile



Koodivõre

- Hargnemisi 2^K
- Sõlmede arv on 2^ν
- Iga ribi kood on pikk
  $N \cdots$ väljundvoogude arv
- ribi valitakse infosümboli järgi, kui see on 0, siis ülemine ribi jne

Koodivõre koodi (6,3) jaoks



Märkus

- Selline koodivõre vastab peaaegu kõikidele koodidele, millistel on üks infovoog sisendis ja kaks infovoogu väljundis
- Võregraafi ribidele tuleb aga anda vastavad kahendkoodid struktuurskeemi kohaselt

22. Ahendkoodide dekodeerimine

- Kaks etappi
  - Vigaste sümbolite parandamine
  - Infosümbolite leidmine (kas väljanoppimine või leidmine algoritmi abil kui kood on eraldamatu)

Dekodeerimismeetodid

1. Sündroomne meetod
2. Majoritaarne meetod
3. Maksimaalse tõepärasuse järgi
   1. Range algoritm
   2. Pehme algoritm
   3. Viterbi algoritm

Viterbi algoritm

- Näide
  - Tekitav maatriks koosneb hulkliikmetest

$$G = \begin{Vmatrix} 1 & 1+z \end{Vmatrix}$$

13. Koodide pesastamine

- Kodeerimisega püütakse kasutada kogu edastuskanali läbilaskevõimet. Probleemiks on, kuidas kindlustada vigasuse väike tõenäosus.

$$R \mapsto C_K, P_v \xrightarrow{n \to \text{suur arv}} 0$$

Koodiploki pikkus n

peab olema küllaldaselt suur

Korduvkodeerimine

- Üks koodide korduva kombineerimise viise on koodide

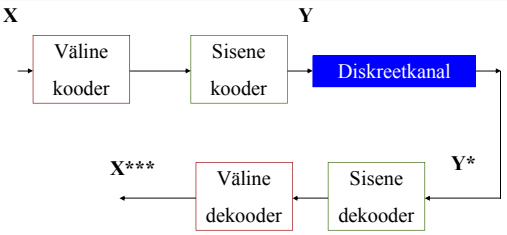
Pesastamine

Pesakood *concatenated code*

nested cod

каскадные коды

Edastamine kaheastmelise järjestikuse pesakoodiga



Kaheastmeline pesakood

- Väline kood on tavaliselt RS kooder, sisemine kood võib olla kas ortogonaalne (*Reed - Maller'* i kood), lühem tavaline plokk-kood näiteks *Hamming*'i kood, kas tavaline täiuslik või laiendatud.
- Sisese koodina võib kasutada ka erinevaid ahendkoode.

Pesakoodi kodeerimine

- Väliseks koodiks valime sobiva pikkusega **N** Reed-Solomoni koodi, mille sümbolid kuuluvad lõplikku korpusesse **GF(2^m),**

RS koodi pikkus $N = 2^m - 1$, **N = K +R**

Kogu infosümbolite arv on **k = m × K**

- Kahendsümbolite plokke pikkusega **m** kasutatakse kui mittekahendkoodi sümboleid

Tähistused

- k infosümbolit $X = X_1, X_2, \ldots, X_k$
- **k = m × K**

Kodeeritakse RS koodigaKodeerimisega lisatakse K infosümbolile (N - K) liiast sümbolit (plokki). Saame pikema RS -koodi ploki:

$$\mathbf{X_1, X_2, \ldots, X_K, R_1, R_2, \ldots, R_{N-K}.}$$

Pesakoodi kodeerimine

Saadud RS -koodi kasutatakse siseses koodris kahendsümbolite jadana, mille iga

RS -koodi sümbolile pikkusega **m** lisatakse veel **r = n - m** liiast sümbolit.

Kaheastmelise pesakoodi koostamist on kõige parem esitada koodisümbolite moodustamise diagrammiga:

Sümbolite järjestuse diagramm

- on

$$X_1,\ X_2,\ \ldots\ X_K, R_1,\ R_2,\ \ldots\ R_{N-K}$$

$$\downarrow$$

$$\underbrace{x_1,\ldots, x_m}_{\text{Sisese koodi infosümbolid}}, \underbrace{r_1,\ldots, r_{n-k}}_{\text{Sisese koodi liiased sümbolid}};$$

RS koodiga kodeeritud jada

- Kodeerimisega lisatakse K infosümbolile (plokki) R = (N - K) liiast sümbolit (plokki).
- Saame pikema RS -koodi ploki:

$$\mathbf{X_1, X_2, \ldots, X_K, R_1, R_2, \ldots, R_{N-K}}$$

Sisene kooder

- Saadud RS -koodi kasutatakse siseses koodris kahendsümbolite jadana, mille iga
- RS -koodi sümbolile pikkusega **m** lisatakse veel **r = n - m** liiast sümbolit.

Kodeerimise diagramm

- Kaheastmelise pesakoodi koostamist on kõige parem esitada koodisümbolite moodustamise diagrammiga:

X_i **RS koodi infosümbol (infoplokk)**

X_j **j-kahendsümboli i-infoplokis**

Välise koodri sisendjada (kahendsümbolid).

- RS koodi infosümbolid (plokid)

$$\mathbf{X_1,\ X_2,\ \ldots\ X_K,\ldots}$$

- RS koodi infosümbol

$$\mathbf{x_1,\ x_2,\ \ldots\ x_m,\ldots}$$

Välise koodri väljundjada

- **Välise koodri väljundjada (RS -koodi mittekahendsümbolid).**

$$\mathbf{X}_1,\ \mathbf{X}_2,\ \ldots\ \mathbf{X}_K,\mathbf{R}_1,\ \mathbf{R}_2,\ \ldots\ \mathbf{R}_{N-K}$$

Sisese koodri väljundjada plokid

- RS -koodi sümbolile pikkusega **m** lisatakse veel **r = n - m** liiast sümbolit.

$$\mathbf{X}_1;\{\mathbf{r}_1\}_{n-m},\ \ldots\ \mathbf{X}_K;\{\mathbf{r}_1\}_{n-m},\mathbf{R}_1;\{\mathbf{r}_1\}_{n-m},\ \ldots\ \mathbf{R}_{N-K};\{\mathbf{r}_1\}_{n-m}$$

$$\mathbf{x}_1,\ \mathbf{x}_2,\ \ldots\ \mathbf{x}_m,\mathbf{r}_1,\ \mathbf{r}_2,\ \ldots,\ \mathbf{r}_{n-m}$$

Pesakoodi dekodeerimine

- Sisene kood dekodeeritakse esmalt

$$\mathbf{x}_1^*,\ \mathbf{x}_2^*,\ \ldots\ \mathbf{x}_m^*,\mathbf{r}_1^*,\ \mathbf{r}_2^*,\ \ldots,\ \mathbf{r}_{n-m}^*$$

$$\downarrow$$

$$\mathbf{x}_1^{**},\ \mathbf{x}_2^{**},\ \ldots\ \mathbf{x}_m^{**},$$

Liiased kahendsümbolid eemaldatakse

Välise dekooderi sisendjada

- on sisese dekoodriga parandatud RS koodi sümbolid (plokid)

$$\mathbf{X}_1^{**};\ \ldots\ \mathbf{X}_K^{**};,\mathbf{R}_1^{**};,\ \ldots\ \mathbf{R}_{N-K}^{**};$$

Välise dekoodri väljundjada

- On välise koodri parandatud infoplokid

$$\mathbf{X}_1^{***},\ \ldots\ \mathbf{X}_K^{***}$$

Saavutatav edastuse kvaliteet

- Kui sisese koodi pikkus on **n**, infosümbolite arv on **m** ja liiaste sümbolite arv **r = n - m**.
- Kui sellise sisese koodi minimaalne koodkaugus on $\mathbf{d_0}$, siis on selline sisene kood võimeline parandama selliseid **q** kordseid vigu plokis pikkusega **n**, et rahuldaks võrrand $\mathbf{d_0 = 2q + 1}$.

Saavutatav edastuse kvaliteet

- Pesakoodide koostamisel tuleb sisese koodi puhul sobivateks lugeda just vigu parandavad koodid. Kuna sisese koodi jaoks kasutatakse lühemaid plokk-koode, saadakse head tulemused ka siis, kui sisene kood parandab ainult ühekordseid vigu.
- Peale dekodeerimist sisese dekoodriga on kindlustatud kahendsümbolite vigade parandamine.

Saavutatav edastuse kvaliteet

- Sisene dekooder kindlustab vigasuse keskmine tõenäosus

μ_s ,

mis on määratud sisese (n,m) koodi vigasid parandavate omadustega.

Edastuse kvaliteet

- Peale kahendsümbolite parandamist sisese koodriga, eemaldatakse sisesed liiased sümbolid $\{r_1\}$, $\{r_2\}$, ..., $\{r_n\}$ ja allesjäänud m -järgulised plokid, mida on kokku N tükki, dekodeeritakse kui RS –kood.

Edastuse kvaliteet

- **Kui välise RS -kood kindlustab koodikauguse D_0, st. et parandab kuni Q-kordseid vigu, siis kogu kaheastmeline pesakood kindlustab minimaalse koodkauguse $D = D_0 d_0$. Kogu pesakoodi plokk omab pikkust $N^* = n \times N$.**

- **Infosümbolite arv selles pikas plokis on võrdne $K^* = m \times K$, ja järelikult on infoedastuse kiirus (keskmine infohulk ühe kahendsümboli kohta) kaheastmelise pesakoodi puhul**

$R^* = (m \times K)/(n \times N) = (m/n) \times (K/N) = R_s \times R_v$

Edastuse kvaliteet

- Üldiselt on teada, et kaheastmeline pesakood kindlustab sümboli vigasuse tõenäosuse μ_p võrdse:

$$\mu_p \le \frac{2^{K-1}}{2^K - 1} \sum_{j=Q+1}^{n} \frac{j+Q}{n} C_n^j \mu_s^j (1-\mu_s)^{n-j}$$

Nimetused

- Kaheastmeliste pesakoodi puhul on kombeks ühendust sisene kooder - diskreetkanal - sisene dekooder nimetada superkanaliks. Välist koodrit - sisend koodrit koos nimetatakse superkoodriks ja sisest dekoodrit - välist dekoodrit superkoodriks- Pesakoodi koodrit - dekoodrit, mis sisaldab siis nii välise kui ka sisese koodri ja sisese ning välise dekoodri, võime nimetada siis superkodekiks.

Paralleelne pesastamine (turbokoodid).

Avastatud 1993.a.

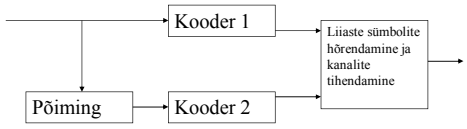
Kirjandus: L. Hanzo, T.H. Liew, B.L. Yeap

Turbo Coding, Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels Wiley 2002.748 lk.

C. Berrou, A. Glavieux Near Shannon limit error correcting coding: turbo codes; ICC 93, Geneva, Switzerland,

E.N.S. des Telecommunications de Bretange, Brest, France

Turbokooder



Kooder 1 ---- süstemaatiline ahendkood

Kooder 2 ---- süstemaatiline ahendkood

Põimimine

- Hajutab vigade pakette
- 0110101010111110000011001010101010 0
- 0010111111100010 0 jne....

Turbo *Hammingi* kood

1 0 1 00 0 1
0 1 0 11 1 0
1 1 1 00 1 0
1 0 1 00 0 1
1 1 1 0
0 0 0 1
0 1 0 1

$$\begin{Vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{Vmatrix}$$

Hammingi koodi tekitav maatriks

14. Krüpteerimise alused

- Krüptoloogia - salastamise teadus

1. κρυπτοζ - salajane
2. λογοζ - teadus

"*kryptos*" ja "*logos*"

- Kaks vastandlikku haru

1. Krüptoloogia (salakirja tegemine)
2. Krüptoanalüüs (salakirja lahtimurdmine)

Krüptograafia

- Tegeleb edastavate andmete (ka kõne) kodeerimise, salastamise ja originaalsete andmete taastamisega nn. dekrüpteerimisega.
- Algandmeid, mis kuuluvad krüpteerimisele, nimetatakse algtekstiks (*plaintext*). Peale krüpteerimist saame krüptiteksti *(ciphertext)* ehk salateksti.

Seosed

- ***P* – algteksti hulk**
- ***C* – krüptotekstide hulk**
- ***K* – krüpteerimisreegel,**
- ***D* - dekrüpteerimisreegel**

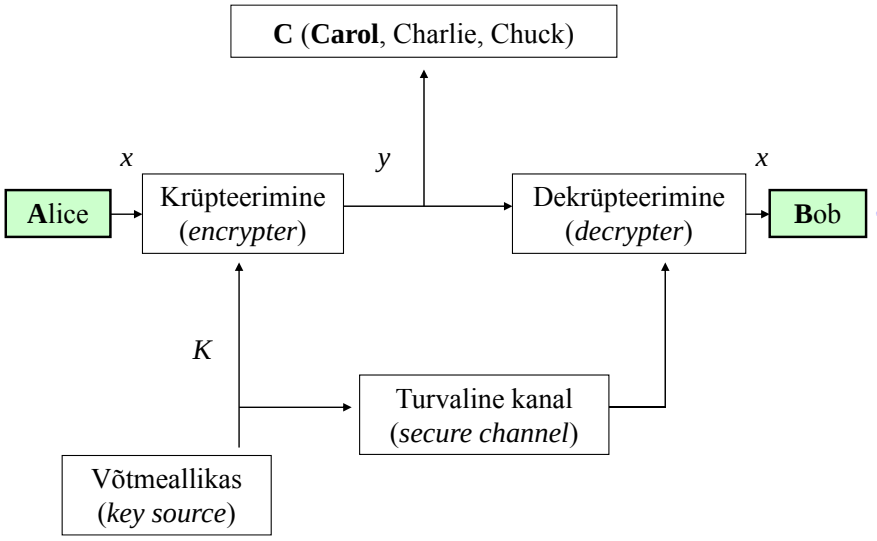
krüptireegel

$$\varepsilon_K \subset \mathrm{E}, d_K \subset D$$

$$(P, C, K, \mathrm{E}, D)$$

$$d_K(\varepsilon_K(x)) = x, kui\ x \subset P$$

Infoedastuskanal



Krüpteerimine valemitega

$x \subset P$

$x = x_1 x_2 ,.., x_i ,..., x_n$ Sõne (i.k. *string*)

$1 \le i \le n$

$x \to y$

$y = y_1 y_2 ,.., y_i ,..., y_n$

$y_i = \varepsilon_K (x_i)$ Reegel peab olema ühene

Inglise tähestik (26 tähte)

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
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Eesti tähestik (24 tähte) *(23, 27, 32)*

A	B	D	E	F	G	H	I	J	K	L	M	N	O	P	R	S	T	U	V	Õ	Ä	Ö	Ü
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23

Soome tähestikus on 28, läti ja vene tähestikus on 33 tähte

Nihkešiffer (*Shift Cipher*)

$y = \varepsilon_K (x) = x + K \bmod 26$ Krüpteerimine

$d_K (y) = y - K \bmod 26$ Dekrüpteerimine

$0 \le K \le 25$

K=3 Caesari krüpter (*Caesar Cipher*)

Et subito lupus venit HW V.....

Näide: nihkekrüpter

P L E A S E W A I T

Probabilities of occurrence of 26 letters (ingliskeelses tekstis)

Letter	Probability	Letter	Probability
A	**0.082**	**N**	**0.067**
B	0.015	**O**	**0.075**
C	0.028	P	0.019
D	0.043	Q	0.001
E	**0.127**	**R**	**0.060**
F	0.022	**S**	**0.063**
G	0.020	**T**	**0.091**
H	**0.061**	U	0.028
I	**0.070**	V	0.010
J	0.002	W	0.023
K	0.008	X	0.001
L	0.040	Y	0.020
M	0.024	Z	0.001

Tähtede esinemistõenäosused

- E **0,120**
- T, A, O, I, N, S, H, R **0,06..0,09**
- D, L 0,04
- C, U, M, W, F, G, Y, P, B 0,015..0,028
- V, K, J, X, Q, Z 0,01

- (digrams) TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN jne.
- (trigrams) THE, ING, AND, HER, ERE, ENT, THA, jne.

Tähtede esinemistõenäosused (eestikeelses tekstis)

A	0.115	**0.140**	N	0.042	0.052
B	0.007	0.010	O	0.034	0.044
D	0.036	0.047	P	0.015	0.022
E	0.105	**0.124**	R	0.021	0.032
F	0.000	0.001	**S**	0.080	**0.096**
G	0.017	0.022	**T**	0.068	**0.080**
H	0.015	0.021	**U**	0.056	**0.067**
I	0.087	**0.102**	V	0.019	0.026
J	0.018	0.022	Õ	0.010	0.015
K	0.045	0.055	Ä	0.010	0.015
L	0.057	0.064	Ö	0.003	0.006
M	0.035	0.042	Ü	0.006	0.010

* Ahto Buldas, Klassikaliste šifrite murdmine, *home.cyber.ee/ahtbu/aturve_5.ppt*

Näited

- Asenduskrüpter
- Affiinne krüpter

$$y = \varepsilon_K(x) = ax + b \bmod 26$$

$$ax = y - b \bmod 26$$

Krüpteerimise võtmed

- Andmete salastamise muutuste kogu nimetatakse krüpteerimise algoritmiks, seadet, mis seda teostab aga krüpteriks. Originaalsete andmete taastamine ehk dekrüpteerimine on võimalik, kui on olemas üks või mitu krüpteri parameetrit nn. “võtit”.

Krüptianalüüs

- Tegeleb krüptigrammi lahtimuukimisega, kui võtit või võtmeid ei ole. Krüptianalüüs on edukas siis, kui analüüsi tegijal on piisavalt ressursse: aega ja tehnilisi vahendeid. Krüptianalüüsiks on tihti tarvis teatud minimaalne aeg $\mathbf{T_{min}}$, mida tehnilised vahendid ei korva.

Krüptograafia

- kindlustab kolm peamist andmeedastuse teenuse eelist:

  1. Salastatuse, st. sanktsioneerimata kasutaja (vaenlane) ei saa edastuse käigus andmeid kätte.
  2. Autentsuse, st. andmete allikad on usaldatavad.
  3. Andmete stabiilsuse, st. et andmed ei muutu edastamise käigus. Neid ei saa rikkuda ega võltsida.

Kasutud kirjandus

1. Stinson D. R. Cryptography Theory and Practice 1995. ISBN 0-8493-8521-0 (U.M.)
2. Sklar B. Digital Communications Fundamentals and Applications 2001.

Koos programmiga System View by Elanix tudengiversioon (U. M.).

Kasutatud kirjandus

3. Kutylowski M.; Strothmann Informationstheorie, Codierung und Kryptologie 1995. (Loengud INTERNETist)

4. Баричев С.Г., Гончаров В.В., Серов Р.Е. Основы современной криптографии 2001. (Raamat INTERNETist)

Üldist

- Krüpteerimine eeldab eelnevat saatja ja vastuvõtja poolset korraldamist ja/või kokkulepet võtme(te) suhtes.
- Kokkuleppelist krüpteerimist nimetatakse vastavalt krüpteerimiseks liht- või salavõtmega.
  - Ka sümmeetriline krüptisüsteem

Liigitused

SÜMMEETRILINE KRÜPTEERIMINE	PLOKK-KRÜPTERID
ASÜMMEETRILINE KRÜPTEERIMINE	JADA-KRÜPTERID
DIGITAALNE ALLKIRI	

Üldist

- Krüpterimist võib teostada ka nii, et on võimalik kindlustada laiatarbeline ja privaatne edastamine. Krüpteril on siis kaks võtit, millest üks on laiatarbeline, teine privaatne.
  - Ka asümmeetriline krüptisüsteem

Asenduskrüpter (*substitution cipher*)

Krüpteerimine:

$\varepsilon_\pi(a) = X,\ \varepsilon_\pi(b) = N,$

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

Dekrüpteerimine:

$d_\pi(A) = d,\ d_\pi(B) = l$

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	l	r	y	v	o	h	e	z	x	w	p	t	b	g	f	j	q	n	m	u	s	k	a	c	i

Affiinne krüpter (*Affine cipher*)

$y = \varepsilon(x) = ax + b \bmod 26$ Krüpteerimine

$d(y) = a^{-1}(y - b) \bmod 26$ Dekrüpteerimine

a^{-1} – multiplikatiivne pöördelement

gcd(a,26) – *greatest common divisor*
suurim arv, millega mõlemad arvud jaguvad jäägita

$b \in Z_{26}$

Üldisel juhul gcd(*a,m*)=1.

Affiinne krüpter (*Affine cipher*)

Algarv jagub vaid arvuga 1 ja iseendaga.

$a \cdot a^{-1} \equiv a^{-1} \cdot a \equiv 1 \pmod m \qquad a^{-1} \in Z_m$

on mooduli multiplikatiivne pöördelement (*modular multiplicative inverse*) mooduliga m

a ja *m* on kaasalgarvud mooduliga *m* *(coprime)*

$a \cdot a^{-1} \equiv 1 \pmod{26} \qquad a^{-1} \in Z_{26}$

Multiplikatiivne pöördelement

Eukleidese algoritm (*Euclidean algorithm*) kahe täisarvu suurima ühisteguri leidmiseks gcd().

Kui $a=b$, siis gcd(a,b)=a
Kui $a>b$, siis gcd(a,b)= gcd($a-b$,b)
Kui $a<b$, siis gcd(a,b)= gcd(a,$b-a$)
Esitab suurima ühisteguri lineaarkombinatsioonina
gcd(a,b) = $\alpha \cdot a + \mu \cdot b$, kus α ja μ on täisarvud. $\mu = a^{-1}$

Näide:
m=26, a=5, a^{-1}=?
gcd(26,5)=gcd(m,a)=gcd(21,5)= gcd(m-a,a)=gcd(16,5)=gcd(m-2a,a)=
=gcd(11,5)=gcd(m-3a,a)=gcd(6,5)= gcd(m-4a,a)= gcd(1,5)=
= gcd(m-5a,a)=gcd(1,4)= gcd(m-5a,a-(m-5a))=gcd(1,3)=
=gcd(m-5a,a-2(m-5a))= gcd(1,2)=gcd(m-5a,a-3(m-5a))= gcd(1,1)=
= gcd(m-5a,a-4(m-5a))= 1

Ahto Buldas, Klassikaline krüptograafia *home.cyber.ee/ahtbu/aturve_4.ppt*

Multiplikatiivne pöördelement

gcd(m-5a,a-4(m-5a))= 1

a-4(m-5a)=a-4m+20a=4m+21a=1

$a^{-1}=21$

Affiinne krüpter mod 26

$1^{-1}=1,\ 3^{-1}=9,\ 5^{-1}=21,\ 7^{-1}=15,\ 11^{-1}=19,\ 17^{-1}=23,\ 25^{-1}=25$

Näide:

$y=\varepsilon(x)=7x+3 \bmod 26$

$d(y)=15(y-3) \bmod 26=15y-19$

Affiinne krüpter

Näide:

R	E	D
17	4	3

$\varepsilon(17)=17*7+3 \bmod 26=122 \bmod 26=18$

$\varepsilon(4)=4*7+3 \bmod 26=31 \bmod 26=5$

$\varepsilon(3)=3*7+3 \bmod 26=24 \bmod 26=24$

$d(y)=$

Vigenere krüpter (*Vigenere cipher*)

Jadašiffer *polyalphabetic*

Võtmed (keyword) K_i mod *m*, kus *m* on võtmepikkus ja

$i \in 0..m-1$

$y=\varepsilon_i(x)=x_i+K_i \bmod 26$ $\quad d(y)=y_i-K_i \bmod 26$

Näide: *m*=4, võti on *HOME*, *K*=(7,14,12,4)

	B	L	A	C	K	D	O	G
	1	11	0	2	10	3	14	6
K	7	14	12	4	7	14	12	4
mod 26	8	25	12	6	17	17	0	10

IZMGRRAK

Hilli krüpter (*Hill cipher*)

K^{-1} on maatriksi *K* pöördmaatriks. $K=k_{i,j}$

$y_i=\varepsilon_i(x_i)=x_i \cdot k_{i,j}$ $\quad i,j \in 1..m$

$d_i(y_i)=y_i \cdot k_{i,j}^{-1}$

Näide:

$y_1=11x_1+3x_2$

$y_2=8x_1+7x_2$

$(y_1,y_2)=(x_1,x_2)\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix}$

pöördmaatriks

$\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix}^{-1}=\begin{pmatrix}7 & 18\\ 23 & 11\end{pmatrix}$

Hilli krüpter

x_i JU LY mod 26

(9,20) (11,24)

$$(9,20)\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix} = (99+60,\ 72+140) = (3,4)$$

$$(11,24)\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix} = (121+72,\ 88+168) = (11,22)$$

y_i DELW

$$(3,4)\begin{pmatrix}7 & 18\\ 23 & 11\end{pmatrix} = (9,20)$$

$$(11,22)\begin{pmatrix}7 & 18\\ 23 & 11\end{pmatrix} = (11,24)$$

Transpositsioon, permutatsioon krüpter (*Transposition, Permutation cipher*)

Näide: Võti $m = 6$

1	2	3	4	5	6
3	5	1	6	4	2

Krüpteerimisel

1	2	3	4	5	6
3	6	1	5	2	4

Dekrüpteerimisel

Algtekst shesellsseashellsbytheseashore

shesel | lsseas | hellsb | ythese | ashore

EESLSH | SALSES | LSHBLE | HSYEET | HRAEOS

Krüptitekst EESLSHSALSESLSHBLEHSYEETHRAEOS

Plokk-krüpter (*block ciphers*)

- Vigenere krüpter, Nihkešiffer
- binaarne 0 ja 1 (mod 2, XOR)

XOR

0	0	0
0	1	1
1	0	1
1	1	0

Näide: $z_{i+4} = z_i + z_{i+1} \mod 2$

m=4, võti on

1,0,0,0 -> 1,0,0,0,1,0,0,1,1,0,1,0,1,1,1…

Pseudorandom binary sequence, m-sequence, LFSR

15. m-jadad

- Signaaliteooriast on teada, et kahe signaali **$s_i(t)$** ja **$s_j(t)$** vaheline kaugus **$d_{ij}(t)$** on määratud valemiga:

$$\frac{1}{T}\int_0^T [s_i(t) - s_j(t)]^2 dt = d_{ij}^2$$

$s_i(t)$, kõikide **i, j** jaoks, kui
$i \neq j$, $i \subset [1, ..., n]$, $j \subset [1, ..., n]$

Simpleks-signaalide süsteemi.

- **Kõikvõimalikud kaugused on samad**

d_{ij} = const

Selliseid signaale nimetatakse ka ekvidistantseteks

- Simpleks-signaalid kindlustavad ühtlase, edastavatest koodkombinatsioonidest sõltumatu, häirekindluse.

Simplekskoodid

- Analoogselt simpleks-signaalidega on võimalik koostada simplekskoode, milliste lubatud koodsõnad moodustavad simpleksi, st. kõigi lubatud koodsõnade $\mathbf{y_i,y_j}$ vahelised kaugused $\mathbf{d_{ij}(y_i,y_j)}$ on samad:

$d_{ij}(y_i,y_j)$ = const

Simplekskoodid

- On teada, et simpleks-koode saab koostada duaalsetena *Hamming*'i koodidele

[Blahut R.E., Peterson W. W.]

- Duaalsed koodid on sellised, milliste koostamisel kontrollmaatriksit **H** kasutatakse tekitava maatriksina **G**

Lõplik korpus $GF\left(2^m\right)$

G_1

1	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**
0	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**0**
0	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**0**
0	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**

Simplekskoodi lubatud koodsõnad

- Tekitava maatriksi **G** abil on võimalik moodustada kõik lubatud koodsõnad, kasutades tekitava maatriksi **G** ridasid $\mathbf{G_1}$, $\mathbf{G_2}$, $\mathbf{G_3}$, $\mathbf{G_4}$, baasvektoritena:.

$$\mathbf{Y} = \mathbf{x_0} \times \mathbf{G_1} + \mathbf{x_1} \times \mathbf{G_2} + \mathbf{x_2} \times \mathbf{G_3} + \mathbf{x_3} \times \mathbf{G_4},$$

kus $[\ ^{\text{n.j.}}\ \mathbf{x_0}, \mathbf{x_1}, \mathbf{x_2}, \mathbf{x_3}]$ on infosümbolid.

Simplekskoodi liiasus

- Kehtib n = k + r
- Kui n = 15, k = 4, r = 11
- Liiasus $U(K) = \frac{11}{15}100\%$
- Infosuutlikkus $N_l = 2^4 = 16$

Küsimus

- Kui suur on simplekskoodi minimaalne kaugus?
- Võrrelge simplekskoodi BCH koodiga.

Simplekskoodi omadus

- Kui erinevaid simpleks-koodi lubatuid koodsõnu vaadelda, tekib mulje, et nullide ja ühte paigutus omab juhuslikku iseloomu. Kui neid koodsõnu analüüsida, siis see esmane mulje kinnistub.

Heal lapsel mitu nime

- Simpleks-koodi lubatuid koodsõnu nimetatakse kas

pseudojuhuslikeks jadadeks,

m-jadadeks,

lineaarseteks rekurrentseteks jadadeks või maksimaalse pikkusega jadadeks.

- Nendega on seotud ka nn. *Gold*'i jadad.

Kasutusalad

- Pseudojuhuslikud jadad ehk **m** -jadad omavad erilist tähtsust kaasaegses andmete kodeerimisel, süsteemsel infoedastuse salastamisel ja ka tavalise salakirja koostamisel (krüpteerimisel) . Tihti kasutatakse **m** -jadasid kaasaegsetes radarites ja sonarites sondeerivate signaalide tekitamiseks

Seos BCH koodiga

- Nagu näeme on **m** -jadad teatud moel seotud primitiivsete BCH koodide moodustamisel kasutatavate lõplike korpuste **GF(2^m)** korrastatud elementidega

$GF\left(2^m\right)$ m-jada

m -jada tekitamine

- **m** -jada saadakse tekitava maatriksi **G** ülemise reana $\mathbf{G_1}$ alustades vasakult noorimast järgust ja jätkates sümboleid saades $\mathbf{G_1}$ pikenduse lõpmatu korrutusega $\beta^i \times \beta$

m-jadade omadused

- 1) Poolsuletud jada **$[y_0, y_1, ..., y_l, ...)$** rahuldab rekurrentset võrrandit:

$$y_l = y_{l-1} \times h_{m-1} + y_{l-2} \times h_{m-2} + ...+ y_{l-m}$$

kui $l \geq m$

ja kus h_i on korpuse **GF(2)** elemendid **0** ja **1** ja korpuse **GF(2^m)** primitiivse elemendi minimaalse hulkliikme kordajad.

m-jada tekitamine

- Omadus 1) näitab, et m - jada saab genereerida tagasisidestuse **m** - järgulise nihkeregistri abil, mille **m** pesasse on kirjutatud $y_{l-1}, y_{l-2}, y_{l-3}, ..., y_{l-m}$ ja y_l on realiseeritud tagasisidega väljundist ja kordajad h_j on realiseeritud tagasisidedega registri vastavatest järkudest

m-jadade omadused

- 2) **m** -jada periood T_m on maksimaalne ja võrdne:

$$T_m = (2^m - 1)\tau$$

τ on ühe sümboli pikkus

Järeldus omadusest 2)

m -järgulise nihkeregistriga saab moodustada ka teisi jadasid, kuid nende periood on väiksem, kui m -jadal. m -jada periood on selline (ajaintervall või) sümbolite arv, millega jada elemendid hakkavad korduma

Kas m –jada on hea?

Kui on vajadust avada m -jada struktuur eesmärgi saada sama jada, siis on vajalike analüüsitavate sümbolite arv

$$N_a = \frac{T_m}{\tau} = N_m$$

m-jadade omadused

- 3) m - jadas on alati $(2^{m-1} - 1)$ nulli ja 2^{m-1} ühte.

Ükskõik millises lõpmata **m** -jada osas säilib selline nullide ja ühte vaheline suhe N_m arvu sümbolitega ploki jaoks.

Märkus

- Igasugune vaatleja, kes ei tuvasta **m** -jada paneb tähele, et nullid ja ühed esinevad jadas ühesuguste tõenäosustega. Kui aga rekurrentne valem on teada, siis 2**m** pikkuse ploki abil saab taastada kogu jada.

m-jadade omadused

- 4) Igasuguse **m** -jada tsükliline nihe on ka **m** -jada.

See on tingitud **m** -jada kui simpleks-koodi lubatud koodsõna omadustest

m-jada omadused

- 5) **m** - jadal on väga head autokorrelatsioonifunktsioonid (AKF).

m -jadade jaoks arvutatakse neid kahte tüüpi AKF:

1. perioodiline (PAKF) ja
2. mitteperioodiline (AKF)

m-jada omadused

- 6) m-jada spekter on ühtlane
  - Kui m-jadast moodustada diskreetne signaal:

Näiteks: 100010011010111 ja sellest signaalist leida spekter kiire *Fourier* i algoritmiga, siis selline spekter peaks tulema peaaegu ühtlane.

See annab võimaluse leida m-jada katsetustega vältides keerulisi matemaatilisi võtteid

(*Woodward M. W*. 1950., 1953.)

m-jada autokorrelatsiooni funktsioonide arvutamine

AKF ja PAKF arvutamiseks peab **m** -jadas ühed asendama **+1** ja nullid **- 1** -ga.

PAKF

- Lõpmatu **m** -jada jaoks, mis saadakse ühe $\mathbf{N_m}$ pikkuse ploki perioodilise pikendamisega saab arvutada PAKF

$$\frac{1}{T_m}\sum_{j=0}^{N_m-1}(-1)^{y_i+y_j+\nu}=\rho_{\mathrm{P}}(\nu)$$

PAKF leidmise lihtsustatud valem

$$\rho_P(v) = (N_k - N_e) / N_m$$

kus N_k on **m** -jada ja tema ***v***- taktiliselt tsükliliselt nihutatud jada kokkulangevad sümbolid,

ja N_e on **m** -jada ja tema ***v***- taktiliselt tsükliliselt nihutatud jada erinevad ehk mitte kokkulangevad sümbolid.

Näide

$$\rho_P(v=0) = 15 - 0 = 15$$

-1	+1	+1	+1	-1	+1	+1	-1	-1	+1	-1	+1	-1	-1	-1
-1	+1	+1	+1	-1	+1	+1	-1	-1	+1	-1	+1	-1	-1	-1

Näide

-1	+1	+1	+1	-1	+1	+1	-1	-1	+1	-1	+1	-1	-1	-1
	-1	+1	+1	+1	-1	+1	+1	-1	-1	+1	-1	+1	-1	-1	-1

$$\rho_P(v=1) = 7 - 8 = -1$$

m-jada generaator

- Väljundis on

1	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**
0	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**0**	**0**
0	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**	**0**
0	**0**	**0**	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**1**	**1**	**1**

m-jada generaatori algoritm

- Laiendatud korpuse $GF(2^m)$

primitiivse elemendi korrutamine

$$\beta^i (\times)_{mod\, m_\beta(z)} \beta,$$
$$i \subset [0;1;....;2^m - 2]$$

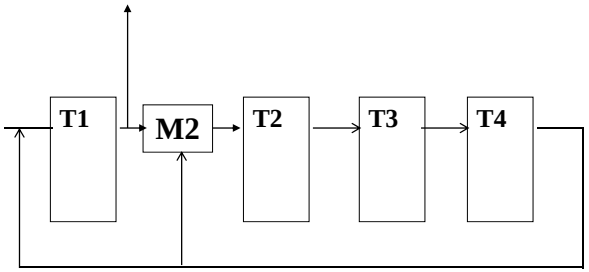
Näide

- Laiendatud korpuse $GF(2^4)$

primitiivse elemendi korrutamine

$$\beta^i (\times)_{mod\,(z^4+z+1)} \beta^1 = \beta^i (\times)_{mod\,(z^4+z+1)} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix},$$
$$i \subset [0;1;....;2^m - 2]$$

Generaatori struktuur



Generaatori kirjeldus

1. m- järguline tagasisisestusega nihkeregister
2. Tagasisidestustega järgud on määratud vastava laiendatud korpuse primitiivse elemendi minimaalse hulkliikmega
   - Laiendatud korpuse primitiivseid (ehk multiplikatiivset tsüklilist rühma tekitavaid) elemente võib olla mitu. Puudub üldine metoodika nende leidmiseks.

Generaatori kirjeldus

- Kuni m = 8,st laiendatud korpuse $GF(2^m)$ elemendid on korrastatud ja avaldatud kodeerimisalases kirjanduses.
- Kui aga m väärtused on suured, m = 20; m = 64; m = 80 .....jne, siis tekib raskusi.

Selliste suurte m väärtuste korral m-jada ei ole lihtsalt tuvastatav. Tuvastamiseks on vaja 2m m-jada järjestikust väärtust (*Sklar*).

Generaatori kirjeldus

3. m- järguline tagasisisestusega nihkeregistrisse võib kirjutada suvalise kahendarvu.

M-jada algab siis mitte algusest, vaid kuskilt keskelt.

On teada, et m-jada tsükliline nihe annab ka m-jada

Kokkuvõte

- M-jada on tuvastatav (st et on võimalik tekitada sama jada), kui on teada

1. Nihkeregistri järkude arv m
2. Nihkeregistri tagasisidestusega järkude kohad on võimalik leida rekurrentsest valemist.
3. Nihkeregistrisse kirjutatud algseisundi kahendarv ei sega tagasisidestusega järkude kohtade määramist.

16. Krüpteerimisealgoritmid ja meetodid

1. Sümmeetriline krüptisüsteem
2. Avaliku võtmega krüpteerimine
3. Digitaalne allkiri (asümmeetrilise krüpteerimise alusel, lisaks kasutatakse veel paiskefunktsiooni (*hash function*) adresseerimiseks)

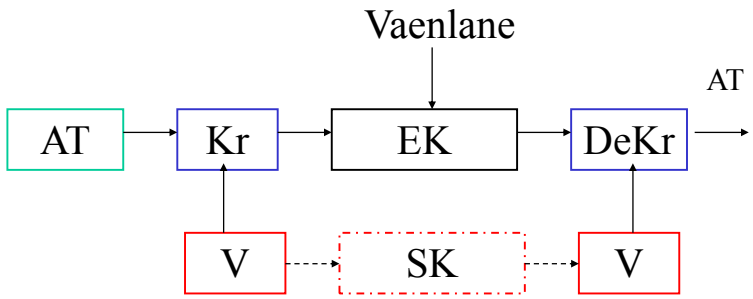
Kasutusalad

- Arvutiteadus: andmeedastuse andmeturve, digitaalne allkiri, jne..... Era- ja ärihuvid
- Sidesüsteemid:, häirekindlad sidesüsteemid, kõne salastamine jne..... Kõik huvid
- Ringlevisüsteemid (TV ja RRH) ärihuvid
- Raadionavigatsiooni süsteemid militaarhuvid
- Infohõivesüsteemid militaarhuvid
- Elektrooniline sõda militaarhuvid

Krüpteerimiseks

- Sobivad kõik võtted
  - Ülekodeerimised
  - Ümberpaigutused
  - Tehted lõplike ja laiendatud lõplike korpuste elementidega
  - Tehted hulkliikmetega lõplike ja laiendatud lõplikel korpustel
  - Igasugused erilised funktsioonid, sh ka mittelineaarsed (näiteks paiskfunktsioon)

1. Sümmeetriline krüptisüsteem

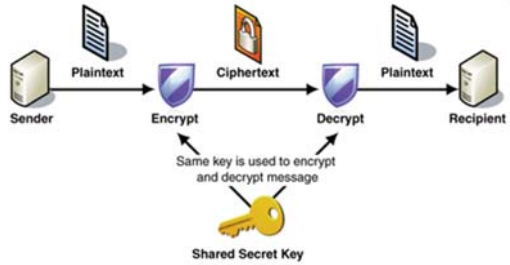


Sümmeetriline krüptisüsteem

Krüpteerimiseks ja
Dekrüpteerimiseks kasutatakse
SAMA V Õ T I T
Seda võtit võib

1. Edastada salajase edastuskanali kaudu
2. Säilitada tarbija juures vastuvõtu poolel
3. Tekitada või genereerida vastuvõtu poolel

Sümmeetriline krüptisüsteem



AES (*Advanced Encryption Standard*) (*block cipher, key* 128, 192, 256 *bits*)
RC4 (*pseudo-random generation algorithm, key* 40 – 128 *bits*, WEP , WPA SSL,)
IDEA (*International Data Encryption Algorithm*) (*block cipher, key* 128 *bits*)

http://en.wikipedia.org/wiki/Symmetric-key_algorithm http://msdn.microsoft.com/en-us/library/ff650720.aspx

Sümmeetrilise krüpteri kirjeldus

- Krüpteerimine toimub vastavalt valemile:

Z = *K* (X,Y),
X - algtekst
Y - võti
***K* (X,Y)** - krüpteri algoritm.

Sümmeetrilise krüpteri kirjeldus

- Dekrüpteri algoritm

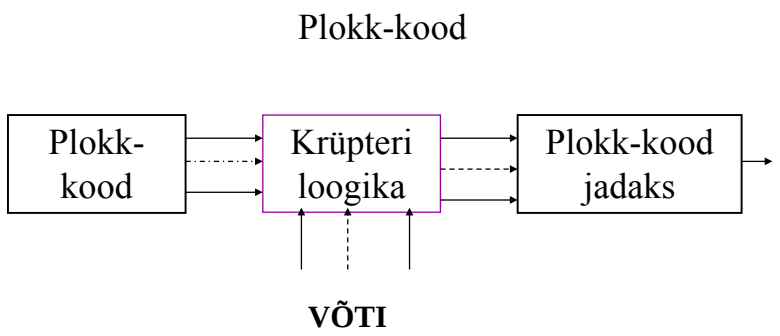
$$X = K^{-1}(Z,Y)$$

- Teisendused K **ja** K^{-1} peavad olema ühesed

Sümmeetrilise krüpteri kirjeldus

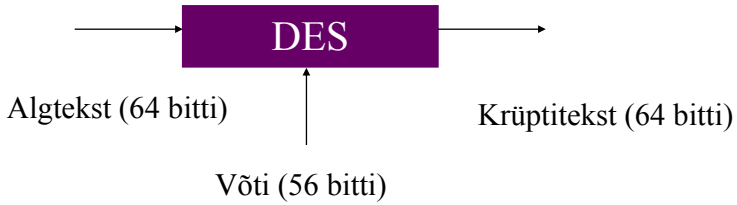
- Vaenlane püüab edastust mõjutada, st, muuta **Z → Z*.** Kui dekrüpter tegutseb formaalselt, siis saame $\mathbf{X^* = K^{-1}(Z^*,Y)}$. Autentsus on tagatud, kui dekrüpter saab aru, et on toimunud **Z → Z*** ja välistab **X*** saatmise tarbijale.

Plokk-krüpteri struktuurskeem

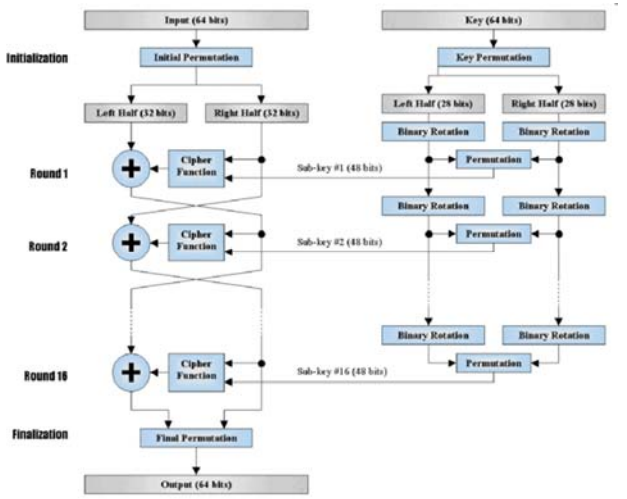


DES krüptisüsteem (Plokk-krüpter)

- DES (*Data Encryption Standard)*



DES



http://www.securitydocs.com/library/3258

DES algoritmi kirjeldus

1. Sklar B. Digital Communications Fundamentals and Applications 2001.
   Lk. 909 – 915 (iseseisev õppimine kohustuslik)
- Kasutatakse kõiki võtteid: ümberpaigutusi, lühendusi, lisamisi, korrutamisi jne.....

Jada-krüpterid

- On rohkem või vähem üles ehitatud ühele ja/või mitmekordsele m-jada kasutusele
- Jada-krüpteriks nimetatakse neid seetõttu, et andmete bitijada on nn poolsuletud jada. (Algus on, lõppu pole)
- Võtmeks on juhuslik nn m-jada

(Algus on, praktilist lõppu pole kuna kordusperiood on väga suur

Näide

- M –jada parameetrid

Olgu biti pikkus 1 μsec, siis

$$T_m = (2^m - 1)\tau \approx 1 \text{ aasta}; \; m = ?$$

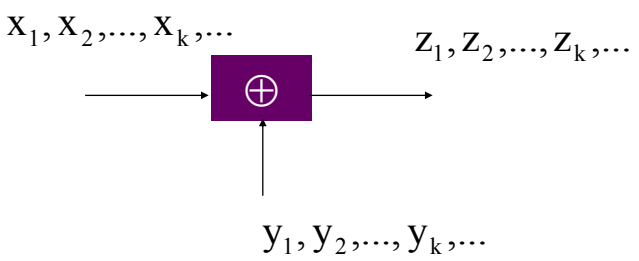
GPS *Global Positioning System*

Globaalne satelliitnavigatsiooni süsteem

C kood	P kood	Y kood
avatud	avatud	militaarne

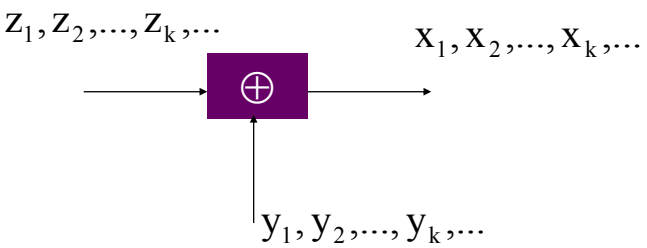
Lihtne jadakrüpter

- Krüpteri struktuurskeem



Lihtne jadakrüpter

- Dekrüpteri struktuurskeem



Märkus

- Dekrüpteerimiseks peab
  - Tekitama sama m-jada, kui on krüpteris

  st et peab teadma
  1. m = ?
  2. Registrisse kirjutatud arvu
  3. Registri tagasisidestusi

Sünkroniseerimise probleemid

1. Sünkroonsed süsteemid
2. Isesünkroniseeruvad süsteemid (vajab sünkroks n korrektset krüptiteksti sümbolit)

Märkus

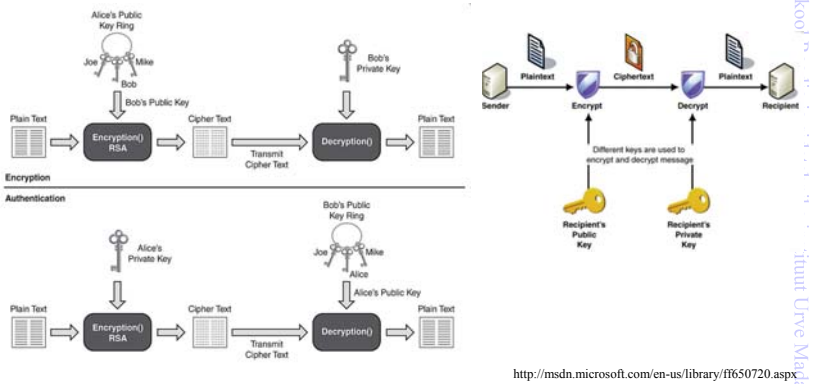
- Krüpteerimine on edukas siis, kui infohulk andmejada **X** ja võtme- jada **Y** vahel on **0**, st. et nende vaheline tinglik entroopia on võrdne nulliga: $\mathbf{H_Y(X) = 0.}$
- Nagu mainitud, on sobivaks võtmejadaks just nimelt **m** – jada. ja selle abil koostatud kõikvõimalikud kombinatsioonid. Loomulikult tuleb praktilise krüpteerimise juures arvesse võtta kõiki vajalikke **m** - jadade korrelatsiooni funktsioone.

2. Asümmeetriline krüptisüsteem

- Kasutatakse kahte võtit

1. Üks neist on avalik
2. Teine salajane ehk privaatne

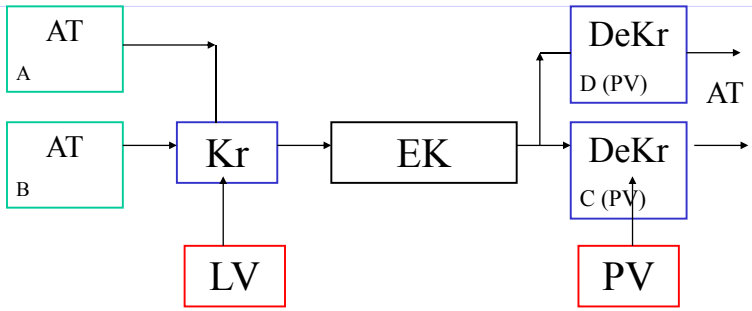
Avaliku võtme krüptograafia *(public-key cryptography)* Asümmeetrilised algoritmid



http://msdn.microsoft.com/en-us/library/ff650720.aspx

http://www.networkworld.com/subnets/cisco/102208-ch2-ssl-vpn-technology.html

2. Asümmeetriline krüptisüsteem



LV kõigile kasutajatele avatud võti (avalik)

PV privaatne võti kindlale kasutajale

Asümmeetriline krüptisüsteem (kirjade saatmine)

- Igaüks avaldab oma avaliku võtme K
- Kirja saatmine
  - Kirja saatja (A) võtab oma avalikku võtme K
  - Kirja saatja A suleb oma kirja avaliku võtmega K
  - A saadab oma kodeeritud kirja saajale C
  - C kasutab privaatset võtit ja loeb A kirja
- Mingeid privaatvõtmeid edasi ei saadeta

Avaliku võtmega krüptisüsteemid
(*Public-key Cryptography)*

- RSA (*Rivest, Shamir, Adleman* 1977.a.) krüpter

  (on kasulik ka digitaalse allkirja koostamiseks)

  Aluseks on rühmateooria (faktoriseerimine)
- *Merkle – Hellmanni* "seljakott" (*Knapsack*)

  Nn seljakotiülesanne
- *McElise*

  Algebraline dekodeerimine

Avaliku võtmega krüptisüsteemid
(*Public-key Cryptography)*

- *ElGamal*
  - Diskreetne logaritm

    $\log_a b = c,$

    $a \rightarrow primitiiv\ GF(p),$

    $p \rightarrow a\ \lg\ arv$

- Elliptiline kõver
  - Mõne eelmise süsteemi modifikatsioon

Euleri teoreem *(Euler's totient function, phi function)*

Iga positiivne täisarv on üheselt esitav algarvude astmete korrutisena:

Näide: $26=2^1*13^1$; $24=2^3*3^1$

Kaasalgarvude arv mooduliga *n*

$$\varphi(n) = (p-1)(q-1)$$

$$\varphi(26) = (2-1)(13-1) = 12$$

Kui *p* ja *q* on juhuslikud algarvud, ning $p \neq q$

Valides juhusliku d>1, et $\gcd(d, \varphi(n)) = 1$ $d \in Z_{\varphi(n)}$

pöördelement $e \cdot d \equiv 1 \mod \varphi(n)$ $e \in Z_{\varphi(n)}$

RSA (*public-key cryptography*)

Krüpteerimine

$y = \varepsilon(x) = x^e \mod n$

Dekrüpteerimine

$d(y) = y^d \mod n$

Avalik võti on (***n,e***) *(public key)*
Salajane võti ***d*** *(private key)*

RSA

Näide: moodul $n = 77 = p \cdot q = 7 \cdot 11$

Euleri phi-funktsioon: $\varphi(n) = (7-1)(11-1) = 60$ avalik võti

Algarv $d = 37 < 60$ gcd(37,60)=1 salajane võti

Kaasalgarv $\bmod \varphi(n)$ $e = 13$ avalik võti

$$\varepsilon(x) = x^{13} \bmod 77$$

$$d(y) = y^{37} \bmod 77$$

RSA

$x = 1,2,3$

$\varepsilon(1) = 1^{13} \bmod 77 = 1$

$\varepsilon(2) = 2^{13} \bmod 77 = 8192 \bmod 77 = 30$

$\varepsilon(3) = 3^{13} \bmod 77 = 1594323 \bmod 77 = 38$

$d(1) = 1^{37} \bmod 77 = 1$

$d(30) = 30^{37} \bmod 77 = 2$

$d(38) = 38^{37} \bmod 77 = 3$

Märkus:

$30^5 \bmod 77 = 24300000 \bmod 77 = 32$

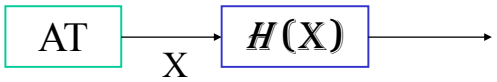
$30^6 \bmod 77 = (32 * 30) \bmod 77 = 36$

Etapp	Operatsiooni kirjeldus	Operatsiooni tulemus
Võtmete genereerimine	Valida kaks algarvu	p=3557, q=2579
	Arvutada moodul	n=p×q=3557×2579=9173503
	Arvutada Euleri funktsiooni väärtust	ϕ(n)=(p-1)(q-1)=9167368
	Valida avatud eksponent *e*	*e*=3
	Arvutada salastatud eksponent *d*	*d*=6111579 (*k*=2)
	Avalikustada avavõti	(e,n)=(3, 9173503)
	Salvestada privaatvõti	(d,n)=(6111579, 9173503)
Šifreerimine	Valida tekst šifreerimiseks	M=111111
	Arvutada krüptogramm	$P(M)=M^e \bmod n=$ $=111111^3 \bmod 9173503=4051753$
Dešifreerimine	Leida avatekst	$S(C)=C^d \bmod n=$ $4051753^{6111579} \bmod 9173503=111111$

Erika Matsak "Avatud võtmega krüpteerimine.Asümmeetrilised krüptosüsteemid. RSA" http://www.tlu.ee/~matsak/crypto/loeng9_s.pdf

Ilma võtmeta krüpteerimine

- Paiskefunktsiooniga krüpter



$H(X)$ paiskefunktsioon

Paiskefunktsioon muudab bitijada sõnedeks *(string)*, millele vastab teatud lühem number (aadress). Seda numbrit võib ka edastada.

Paiskefunktsioon

- *Hash* (ing.k.) hakkima, segama,....
  - Sarnane omapärase kontrollsummaga
  - Tavakasutus salvestusel (paiskadresseerimine).
  - Kasutatakse koos digitaalallkirjaga
  - Digitaalallkirja standard DSS (USA)
  - Digitaalallkirja standard ГОСТ P 34.10-94 (Venemaa)

Paiskefunktsioon

- $Y = H(X)$ lihtne arvutada
- Pöördfunktsioon peab olema arvutamiseks võimatu

$$X = H^{-1}(Y)$$

Andmed	X
Aadress	Y

Paiskadresseerimine

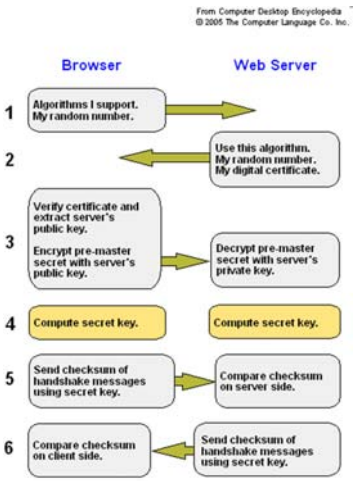
Näide: *RC4-based cryptosystems*

- WEP
- WPA (default algorithm, but can be configured to use AES-CCMP instead of RC4)
- BitTorrent protocol encryption
- Microsoft Point-to-Point Encryption
- Opera Mini
- Secure Sockets Layer SSL (optionally)
- Secure shell SSH (optionally)
- Remote Desktop Protocol
- Kerberos (optionally)
- SASL Mechanism Digest-MD5 (optionally)
- PDF
- Skype (in modified form)

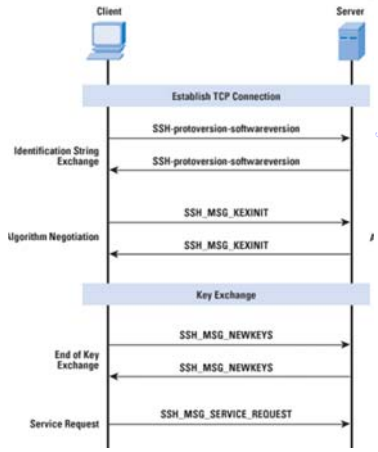
http://en.wikipedia.org/wiki/RC4

Transport Layer Security (TLS)
***Secure Sockets Layer* (SSL)**

SSH — *Secure Shell*



http://img.zdnet.com/techDirectory/SSL.GIF



http://www.ipjforum.org/?p=180

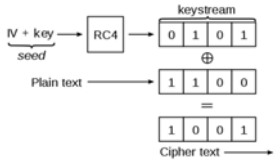
WEP (*Wired Equivalent Privacy*)

Standard 64-bit WEP uses a 40 bit key (WEP-40), 24-bit initialization vector (IV)

128-bit WEP key is almost always entered by users as a string of 26 hexadecimal characters (0-9 and A-F).
(26 × 4=104 bits)+ 24-bit IV =128-bit WEP key.

256-bit WEP system is available from some vendors,
(58 × 4 = 232 bits) + 24 bits IV = 256-bit WEP key

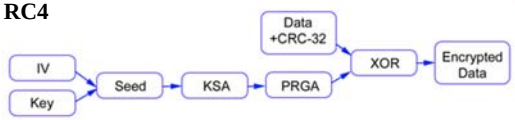
IEEE 802.11



key scheduling algorithm (KSA)

pseudo-random generation algorithm (PRGA)

RC4

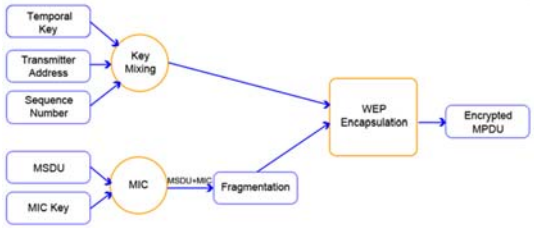


http://www.berghel.net/publications/wifi_vul/wifi_vul.php http://en.wikipedia.org/wiki/File:Wep-crypt-alt.svg

WPA (*Wi-Fi Protected Access*)

Temporal Key Integrity Protocol (TKIP)

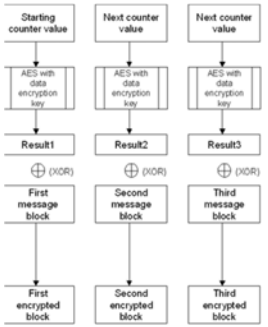
TKIP implements a key mixing function that combines the secret root key with the initialization vector (IV) before passing it to the RC4 initialization.



Message Integrity Check (MIC) MAC service data unit (MSDU)

WPA2

Cipher Block Chaining Message Authentication Code Protocol **CCMP**
Advanced Encryption Standard (AES) *(Substitution permutation network, symmetric-key encryption, block ciphers*) AES-128, AES-192, AES-256



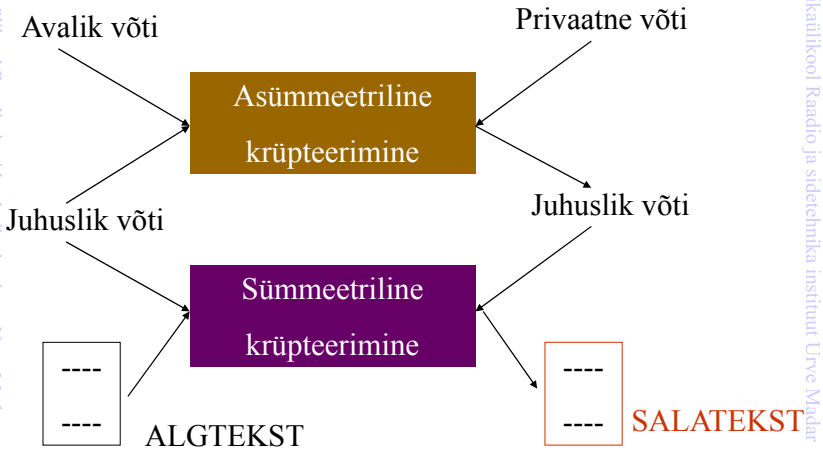
- ***The AES counter mode encryption algorithm uses the following process:***
- ***Encrypt a starting 128-bit counter with AES and the data encryption key. This produces a 128-bit result (Result1).***
- ***Perform an exclusive OR (XOR) operation between Result1 and the first 128-bit block of the data that is being encrypted. This produces the first 128-bit encrypted block.***
- ***Increment the counter and encrypt it with AES and the data encryption key. This produces Result2.***
- ***Perform XOR between Result2 and the next 128 bits of the data. This produces the second 128-bit encrypted block.***

http://technet.microsoft.com/en-us/library/bb878096.aspx

Programmipakett PGP

- PGP *Pretty Good Privacy*
  - On *Philip Zimmermanni* kirjutatud programm
  - Aluseks on avalikud kirjandusest tuntud krüpteerimisalgoritmid
  - PGP ver. 6

PGP hübriidalgoritm



PGP hübriidalgoritm

- Salatekst surutakse enne sulgemist kokku ZIP algoritmiga.
  - See raskendab oluliselt krüptianalüüsi ja samas ka ühtlustatb statistilisi karakteristikuid

PGP hübriidalgoritm

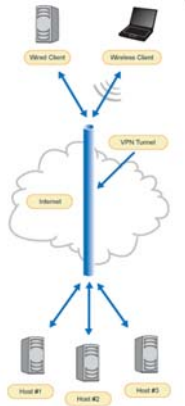
- Asümmeetrilised krüptialgoritmid:
  - RSA
  - DHE
  - DSS
- Sümmeetrilised krüptialgoritmid
  - CAST
  - IDEA
  - Kolmekordne DES

Virtuaalsed privaatvõrgud *Virtual private network* (VPN)

VPN - krüptitud tunnelid kahe või enama võrgu vahel

VPN protokollid:

- IPsec (*Internet Protocol Security*) IPv6
- Transport Layer Security (SSL/TLS) can tunnel an entire network's traffic, as it does in the OpenVPN project, or secure an individual connection.
- Datagram Transport Layer Security (DTLS), is used in Cisco's next-generation VPN product, Cisco AnyConnect VPN, to solve the issues SSL/TLS has with tunneling over TCP.
- Secure Shell (SSH) VPN -- OpenSSH offers VPN tunneling to secure remote connections to a network or inter-network links.
- jne



http://en.wikipedia.org/wiki/Virtual_private_network http://sonic.net/features/vpn/vpn-diagram-vertical-shorte.gif