20SK – Signály a kódy

Přednáška 10 – Bezpečnostní kódy (3.12.2018)

Probíraná témata:

- Principy bezpečnostních kódů (ECC)
- Informační poměr, minimální kódová vzdálenost, detekční a korekční schopnosti kódu.
- Binární sčítání a násobení.
- Formální definice lineárního a binárního lineárního kódu.
- Hammingova váha, minimální kódová váha.
- Kontrolní matice binárního lineárního kódu.
- Generující matice, systematické kódování, kontrolní matice.
- Tvrdé dekódování lineárních kódů.

Relevantní literatura je [1, kapitoly 1 a 2], [2, kapitoly 5-8], [3, kapitola 3] a [4, kapitola II a část kapitoly III].

Seznam literatury

[1] Morelos-Zaragoza, R. H.: *The Art of Error-Correcting Coding*. 2nd edition, John Wiley & Sons, 2006, 263pp.

[2] Adámek, J: *Foundations of Coding: Theory and Applications of Error-Correcting Codes with an Introduction to Cryptography and Information Theory*. Wiley Interscience, 1991, 352 pp.

[3] Moon, T. K.: *Error Correction Coding – Mathematical Methods and Algorithms.* Wiley Interscience, 2005, 756 pp.

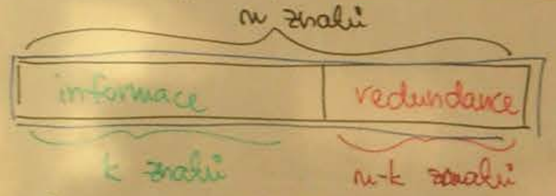
[4] Adámek, J: Kódování. Matematika pro vysoké školy technické, sešit XXXI. SNTL, 1989, 192 pp.

BEZPEČNOSTNÍ KÓDY

→ kódování kanálu

→ ECC (error-correcting code)

→ základní princip: REDUNDANCE



... kódové slovo

⇒ uvažujeme blokové kódy

→ systematické kódování : k informačních znaků
n-k kontrolních

→ (n,k) – kódy

Příklad: parita

000 0	(4,3)-kód
001 1	
010 1	
011 0	inf. poměr $\frac{3}{4}$
100 1	$d_{min}=2$
101 0	$d(0011, 1100)=4$
110 0	$d(1010, 1111)=2$
1111	

Příklad: opakovací kód

0000	(4,1)-kód $d_{min}=4$
1111	inf. poměr $\frac{1}{4}$
	$d(0000, 1111)=4$

INFORMAČNÍ POMĚR $\frac{k}{n}$

KÓDOVÁ VZDÁLENOST $d(u_1, u_2)$ ... počet míst, na nichž se kódová slova liší

MINIMÁLNÍ KÓD. VZDÁLENOST $d_{min}(k) = \min_{\substack{u_1 \in K \\ u_2 \in K \\ u_1 \neq u_2}} d(u_1, u_2)$

LINEÁRNÍ KÓDY

binární sčítání = XOR

$\oplus$	0	1
0	0	1
1	1	0

Příklad: $x = (x_1, x_2, x_3)$

$x_1 \oplus x_2 = 0$

$x_0 \oplus x_3 = 0$

$x_1 \oplus x_3 = 0$

$\vec{v} = u_0 \cdot \vec{v_0}$ $\quad \vec{v_0} = G = [1\,1\,1]$

Lin. kód: podprostor $V_n = \{0,1\}^n$ dimenze k

(k je počet informačních bitů)

↳ 2^k kód slov délky n

⇒ bázové vektory $\{\vec{v_0}, \vec{v_1}, \dots, \vec{v_{k-1}}\}$

(je jich k)

⇒ holý text: $\vec{u} = (u_0, u_1, u_2, \dots, u_{k-1})$ — k informačních bitů

⇒ kódové slovo: $\vec{v} = u_0 \cdot \vec{v_0} \oplus u_1 \cdot \vec{v_1} \oplus \dots \oplus u_{k-1} \cdot \vec{v_{k-1}}$

$\vec{v} = \vec{u} \cdot G$ generující matice $G_{k \times n}$

$G = \begin{bmatrix} \vec{v_0} \\ \vec{v_1} \\ \vdots \\ \vec{v_{k-1}} \end{bmatrix}$ řádky jsou bázové vektory

Příklad: parita $x = (x_0, x_1, x_2)$

$x_1 \oplus x_2 \oplus x_3 = 0$

$\vec{v} = u_0 \cdot \vec{v_0} + u_1 \cdot \vec{v_1}$

$\vec{u} = (0,0)$
$\vec{v} = (0,1)$
$\vec{u} = (1,0)$
$\vec{u} = (1,1)$

$\vec{v_0} = (1,0,1)$
$\vec{v_1} = (0,1,1)$

$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

$\vec{v} = \vec{u} \cdot G$

$[0\,0\,0] = [0\,0] \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$

$[0\,1\,1] = [0\,1] [\dots]$

$\vdots$

LINEÁRNÍ KÓDY

Vlastnosti:

1. $\vec{v} = (0 \dots 0)$ je kódové slovo
2. $\vec{v}_1 \in K$ a $\vec{v}_2 \in K \Rightarrow \vec{v}_1 \oplus \vec{v}_2 \in K$
3. $\vec{v} \in K$ a $\alpha \in \{0,1\} \Rightarrow \alpha \vec{v}_1 \in K$

Hammingova váha slova: $w_H(\vec{v}) = d(\vec{v}, 0)$

počet jedniček v kód. slově $\quad \vec{v} \neq 0$

$$d_{min} = \min_{\substack{\vec{v} \in K \\ \vec{v} \neq 0}} w_H(\vec{v})$$

!! obecně $d_{min} \Rightarrow 2^{k-1}(2^k - 1)$ porovnání

lin. kód: $d_{min} \rightarrow 2^k$ porovnání

Kontrolní matice lin. kódu

$\vec{v} = \vec{u} \cdot G \xrightarrow[\text{kom. kanál}]{} \vec{v}' \cdot H^T = \vec{0}$

H ... kontrolní matice (parity check matrix)

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

$\Rightarrow$ systematické kódování

$\vec{v}$ ... [k | ///]

$$G = \left[\begin{array}{c:c} \mathbb{I}_{k \times k} & \mathbb{P}_{k \times n-k} \end{array} \right]$$

$$H = \left[\begin{array}{c:c} \mathbb{P}^T_{n-k \times k} & \mathbb{I}_{n-k \times n-k} \end{array} \right]$$

Příklad: (4,2)-kód

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$G_{SYS} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} \Rightarrow P = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

$$P^T = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \Rightarrow H = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

$$\vec{u} = [0\,1]$$

$$\vec{v} = [0\,1]\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix} = [0\,1\,0\,1]$$ tato jsem vyslal

$$\vec{v} \cdot H^T = [0\,1\,0\,1]\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [0\,0]$$ ... toto je kódové slovo

$$\vec{v}' = [1\,1\,0\,1]\begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} = [1\,1]$$ ... toto NENÍ kódové slovo !!

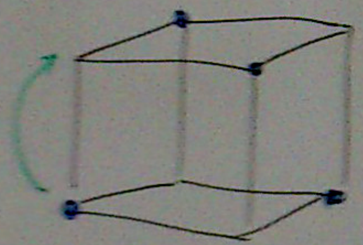
chyba v prvním bitu

syndrom

Parita

$d_{min} = 2$

000 • detekuje jednoduchou
011 chybu
101 •
110



DETEKCE A OPRAVA CHYB - dokončení

d_{min} .. m.j. udává, kolik bitů se musí nejméně v kódovém slově změnit, abych dostal opět kódové slovo.

n-násobná chyba při přenosu .... „flip" n bitů původního slova
(jednoduchá ... n=1) (překlopení; inverze)

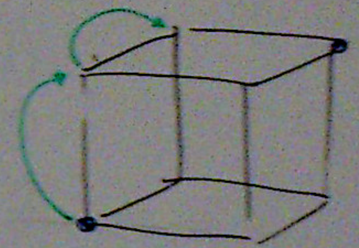
Detekční schopnosti kódu K: Je schopen detekovat

t-násobnou chybu, pokud $d_{min} > t$

Opakovací kód

$d_{min} = 3$

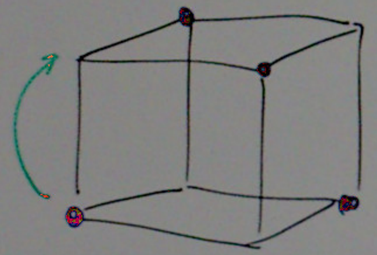
000 • detekuje dvojnásobnou
111 chybu



Parita

$d_{min} = 2$

000 • detekuje jednoduchou
011 chybu
101 • neopraví nic
110



Korekční schopnosti kódu K: Je schopen opravovat

t-násobnou chybu, pokud $d_{min} > 2t$

n-násobná chyba při přenosu .... „flip" u bitů původního slova
(jednoduchá ... $n = 1$) (překlopení; inverze)

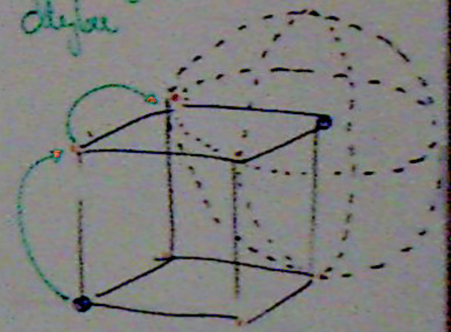
Detekční schopnosti kódu K: Je schopen detekovat

t-násobnou chybu, pokud $d_{min} > t$

Opakovací kód

$d_{min} = 3$

000 • detekuje dvojnásobnou
111 chybu
• opraví jednoduchou chybu



Opakovací kód:

$x_1 \oplus x_2 = 0$

$x_1 \oplus x_3 = 0$

$(3,1)$ kód, $k=1$

$\vec{v} = u_0 \cdot \vec{v}_0$ $\begin{cases} 0 \to 000 \\ 1 \to 111 \end{cases}$

$\Downarrow$

$\vec{v}_0 = (1,1,1)$

LINEÁRNÍ KÓDY

- podprostor $\mathcal{V}_n = \{0,1\}^n$ dimenze k

$\Rightarrow$ bázové vektory $\{\vec{v}_0, \vec{v}_1, \dots, \vec{v}_{k-1}\}$

kódové slovo $\vec{v} = u_0 \cdot \vec{v}_0 \oplus u_1 \cdot \vec{v}_1 \oplus u_2 \cdot \vec{v}_2 + \dots \oplus u_{k-1} \cdot \vec{v}_{k-1}$

$u_0 \dots u_{k-1}$ koeficienty $\in \{0,1\}$

↳ informační bity původního textu

$\vec{v} = \vec{u} \cdot G$

generující matice

$G = \begin{pmatrix} \vec{v}_0 \\ \vec{v}_1 \\ \vdots \\ \vec{v}_{k-1} \end{pmatrix}$ řádky jsou bázové vektory

$\vec{u} = (u_0, u_1, \dots, u_{k-1})$ řádkový vektor

Příklad: parita

$\vec{x} = (x_1, x_2, x_3)$

$\vec{x} = (0,0,0)$

$\vec{x} = (0,1,1)$

$\vec{x} = (1,0,1)$

$\vec{x} = (1,1,0)$

$x_1 \oplus x_2 \oplus x_3 = 0$

→ je to lineární kód!

$(3,2)$ kód, $k=2$

$\vec{v} = u_0 \cdot \vec{v}_0 \oplus u_1 \cdot \vec{v}_1$

$00 \to 000$

$01 \to 011$

$10 \to 101$

$11 \to 110$

$\vec{v}_0 = (101)$

$\vec{v}_1 = (011)$

Opakovací kód:

$0 \to 000$, $1 \to 111$

Parita:

$00 \to 000$, $01 \to 011$,
$10 \to 101$, $11 \to 110$

$$d_{min} = \min_{\vec{v} \in K,\, \vec{v} \neq \vec{0}} w_H(\vec{v})$$

→ minimální kódová vzdálenost je minimální Hamm. váha nenulového kód. slova

d_{min} obecně: $2^{k-1}(2^k - 1)$ porovnání

d_{min} lin. kódu: minimum z $\underline{k}$ prvků

Systematický kód: oddělená informační & kontrolní část

Systematický lin. kód:

$$\vec{v} = \vec{u} \cdot G$$

$$G = \left[I_{k \times k} \,\vdots\, P_{k \times (n-k)} \right]$$

$$\vec{v} \cdot H^T = 0$$

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

$$H = \left[P^T_{(n-k) \times k} \,\vdots\, \mathbb{I}_{(n-k) \times (n-k)} \right]$$

→ u syst. kódu lze jednoduše $G \to H$ a $H \to G$

Opakovací kód:

$0 \to 000$, $1 \to 111$

Parita:

$00 \to 000$, $01 \to 011$,
$10 \to 101$, $11 \to 110$

Příklad: (4,2)-kód

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$ → toto není systematický tvar!

$$G = \begin{bmatrix} 1 & 0 & \vdots & 1 & 1 \\ 0 & 1 & \vdots & 1 & 0 \end{bmatrix} \Rightarrow P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$I_{2\times 2}$ $P_{2\times 2}$

$$\Rightarrow P^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow H = \begin{bmatrix} 1 & 1 & \vdots & 1 & 0 \\ 1 & 0 & \vdots & 0 & 1 \end{bmatrix}$$

Syst. tvar: $\vec{u} = (u_1 u_2 \dots u_{k-1} u_k \; p_1 \dots p_{n-k})$

(původní informace; zakódované)

Systematický tvar kódu

$\vec{u} \cdot G = \vec{c}$

$$G = \left[I_{k\times k} \;\vdots\; P_{k\times(n-k)} \right]$$

$\vec{c} \cdot H^T = 0$

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

$$H = \left[P^T_{(n-k)\times k} \;\vdots\; I_{(n-k)\times(n-k)} \right]$$

→ z toho vidíme, že platí: umíme $G \to H$ a $H \to G$

Jak to funguje?

$\vec{u} = (0,1)$

$\vec{v} = (0,1) \cdot \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} =$

$= (0,1,1,0)$

Pro původní nesyst. G:

$\vec{v} = (0,1) \cdot \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} =$

$= (0011)$

Příklad: $(4,2)$-kód

$G = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ → toto není systematický tvar!

$G = \left[\begin{array}{cc:cc} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{array}\right] \Rightarrow P = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$

$I_{2\times 2}$ $P_{2\times 2}$

$\Rightarrow P^T = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow H = \left[\begin{array}{cc:cc} 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{array}\right]$

syst. tvar: $\vec{v} = (\underbrace{u_0, u_1, \dots, u_{k-1}}_{\text{původní informace}} \mid \underbrace{p_0, \dots, p_{n-k}}_{\text{zabezpečení}})$

Kontrola:

$v = (0,1,1,0) \cdot \begin{bmatrix} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$

$= (0,0)$

$\Rightarrow \vec{v}$ je kódové slovo!

$(0,1,1,0) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix} =$

$= 0 \oplus 1 \oplus 1 \oplus 0 = 0$

$\vec{v} = (1,1,1,0)$

vyjde $\vec{v} \cdot H^T = (1,1)$

→ toto není kód. slovo!

Př: (7,4) kód

$7 \leq 2^3 - 1 = 7$ (P)

→ nelze pro vždy k splnit rovnost

Př: pro $k=2$

$n \leq 2^{n-2} - 1$

$3 \leq 2-1$ ✗

$4 \leq 4-1$ ✗

$5 \leq 8-1=7$ ✓

není 'perfektní'!
musíš přidat další 2 bity!
informace

PERFEKTNÍ KÓDY

→ nejkratší možný kód pro danou konfiguraci

→ kódová slova jsou od sebe vždy stejně & minimálně možně vzdálena

Příklad: pro opravu jednoduché chyby $d(\vec{v}_i, \vec{v}_j) = 3$ vždy

Hammingova hranice pro (n,k) kód:

$$n \leq 2^{n-k} - 1$$

→ pokud splněno, opravuje jednoduché chyby

Př.: (3,1) kód

$n=3$
$n-k=2$ (P) perfektní

$3 \leq 2^2 - 1 = 3$ ✓

→ platí, opravuje jednoduché chyby

(4,3)-kód

$4 \leq 2-1$

→ neplatí, neopravuje!

Př: (7,4) kód

$7 \leq 2^3 - 1 = 7$ (P)

→ nelze pro vědu k opravit vícenásobné

Př: pro $k=2$ k opravit vícenásobné

$n \leq 2^{n-2} - 1$

$3 \leq 2-1$ ✗

$4 \leq 4-1$ ✗

$5 \leq 8-1=7$ ✓

není 'perfektní'!
nelze přidat další 2 bity!
informace

PERFEKTNÍ KÓDY

↳ platí $n = 2^{n-k} - 1$

$(3,1), (5,2), (6,3), (7,4), \ldots$

→ perfektní

Hammingův kód

$n = 2^m - 1$ $\quad m = 2, 3, \ldots$

$k = 2^m - m - 1$ → $(3,1), (7,4), (15,11), \ldots$

$d_{min} = 3$

Př: 7,4

H je 3×7:

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

1 2 6 7