Maticové vkládání – úvod

- Dejme tomu, že chceme vkládat zprávu délky $\frac{2}{3}n$, čili $\alpha = \frac{2}{3}$.
- Za normálních okolností bychom pomocí LSB vkládání ~~vzali~~ vložili do dvou pixelů a třetí bychom vynechali

$$\overline{|x_1|x_2|x_3|x_4|x_5|x_6|} \ldots$$

Emb ↑ ↑ ↑ ↑
z_1 z_2 z_3 z_4

- Zkusme to jinak. Pro každou trojici pixelů budeme zprávu vkládat tak, aby $z_1 = LSB(y_1) \oplus LSB(y_2)$ atd. pro každou trojici stejně.
$z_2 = LSB(y_2) \oplus LSB(y_3)$
- První bit zprávy je rozložen mezi y_1 a y_2 nedá se říct, že by byl v y_1 ani v y_2 samotném. Stejně tak druhý bit je rozložen mezi y_2 a y_3.
- Jaká je potom efektivita?
Jak probíhá vkládání?

← za předpokladu, že x_1, x_2, x_3 jsou nezávislé náh. veličiny

pst.	$LSB(x_1) \oplus LSB(x_2) = z_1$	$LSB(x_2) \oplus LSB(x_3) = z_2$	co dělat	příspěvek k distorzi
1/4	✓	✓	⇒ nic	0
1/4	×	✓	⇒ překlop LSB x_1	1
1/4	✓	×	⇒ překlop LSB x_3	1
1/4	×	×	⇒ překlop LSB x_2	1

Očekávaná distorze: $\underbrace{(0 + 1/4 + 1/4 + 1/4)}_{\text{distorze na blok}} \underbrace{\frac{n}{3}}_{\text{počet bloků}} = \frac{n}{4}$

délka zprávy: $\frac{2n}{3}$

efektivita: $\frac{\frac{2n}{3}}{\frac{n}{4}} = \frac{8}{3} = 2.667 > 2$

- Pro srovnání: Jak funguje standardní LSB embedding

pst	$LSB(x_1) = z_1$	$LSB(x_2) = z_2$	co dělat	příspěvek k distorzi
1/4	✓	✓	⇒ nic	0
1/4	×	✓	⇒ překlop LSB x_1	1
1/4	✓	×	⇒ překlop LSB x_2	1
1/4	×	×	⇒ překlop LSB x_1 a x_2	2

Očekávaná distorze: $(0 + 1/4 + 1/4 + 2/4)\frac{n}{3} = \frac{n}{3}$

délka zprávy: $\frac{2n}{3}$

efektivita: $\frac{\frac{2n}{3}}{\frac{n}{3}} = 2$ (efektivita je nezávislá na α protože se měří vzhledem k délce zprávy).

- Extrakci můžeme popsat pomocí maticových operací:

$$\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} LSB(y_1) \\ LSB(y_2) \\ LSB(y_3) \end{pmatrix}$$

analogicky pro další bloky.

- Značení: v této části budeme používat:

$x = (x_1, \dots, x_n) \in \mathbb{F}_q^n$ posloupnost hodnot spjatých s blokem nosiče

$y = (y_1, \dots, y_n) \in \mathbb{F}_q^n$ posloupnost hodnot spjatých s blokem stegosoubor.

$z = (z_1, \dots, z_m) \in \mathbb{F}_q^m$ blok zprávy

- Spjatá hodnota může znamenat pro $q=2$ LSB jednotlivých prvků (pixelů, DCT koeficientů) ale také to může znamenat paritu ve smyslu vkládání s optimálním přiřazením parity.
(do paletových obrázků)

- Pro $q \geq 3$ zbytek po dělení q.

- (S klíčem si statost nedělejme, použije se analogicky jako u běžných metod.)

- Algoritmus extrakce: H předem dohodnutá matice $\in \mathbb{F}_q^{m \times n}$

$$\boxed{z = Hy}$$

- Příklad $H = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$ (m řádků, $n = 2^m - 1$ sloupců) (obecně paritní matice binárního Hammingova kódu)

Máme nosič x a zprávu z.
Chceme y.
Spočítáme <u>syndrom</u> vektoru x: $s = Hx$

Pokud $Hx = z$ není co řešit $y = x$.
Pokud $Hx \neq z$: • spočtu rozdíl $z - Hx = \begin{pmatrix} \cdot \\ \vdots \end{pmatrix}$
• v matici H najdu j-tý sloupec pro který $z - Hx = H_{*j}$
• za y zvolím $x + e_j$, kde $e_j = (0, 0, \dots 0, 1, 0, \dots, 0)$ (1 = j-tá pozice.)
• Kontrola: $Hy = H(x + e_j) = Hx + He_j = Hx + H_{*j} = z$.

Srovnání s dekódováním Hammingových kódů:
- Při dekódování je cílem získat nulový syndrom $Hy = 0$.
e_j je potom chybový vektor a $x + e_j$ je opravené slovo.
- Čili dekódování ≡ vkládání nulové zprávy.