1

První náhodná veličina neexistuje, protože kód není prefix-free, takže nemůže být Huffmanův. 10 je prefixem 101.

Druhá náhodná veličina také neexistuje, protože kód také není Huffmanův, jelikož není "nejlepší možný". Nemá tedy nejmenší možnou entropii, protože kód 110 by šel zkrátit na 11 a vše by fungovalo.

Třetí kódování již Huffmanovo je. Nechť kódujeme množinu $A = \{1, 2, 3, 4, 5\}$ Náhodná veličina X:

$a \in A$	1	2	3	4	5
Pr[X=a]	0,78	0,05	0,05	0,06	0,06
Kód	1	000	001	010	011

2

Nechť platí $\exists x, y \in \mathcal{C}, z \in \{0,1\}^* \setminus \epsilon : x = y||z$. Z definice Huffmanova kódování je y list ale x je z definice také list. Ale při dekódování x, bychom se dostali nejdříve do y (což je list) ale poté bychom nemohli již nikam pokračovat. Tedy by platilo, že $x = y$, což je ale spor, jelikož $z \neq \epsilon$.

3

Viz kód

4

$\Leftarrow$: Chceme dokázat, že latinský kryptosystém je absolutně bezpečný. Nechť $x \in \mathbb{P}, y \in \mathbb{C}, z \in \mathbb{K}$ a $\mathbf{P}, \mathbf{C}, \mathbf{K}$ příslušné náhodné veličiny, potom platí:

$$Pr(\mathbf{P} = x|\mathbf{C} = y) \overset{\text{def.}}{=} \frac{Pr(\mathbf{P} = x, \mathbf{C} = y)}{Pr(\mathbf{C} = y)}$$

Každý ciphertext je jednoznačně určen dvojicí (x, z) z toho, jak je definován latinský kryptosystém (jsou tam bijekce). Z toho plyne

$$Pr(\mathbf{P} = x, \mathbf{C} = y) = Pr(\mathbf{P} = x, \mathbf{K} = z) = Pr(\mathbf{P} = x) \cdot \frac{1}{n}$$

Poslední rovnost plyne také z definice lat. čtverce, jelikož znám-li z, mám řádek z n možnostmi. Dále $Pr(\mathbf{C} = y) = \frac{1}{n}$, jelikož čtverec obsahuje n^2 hodnot a z definice každá hodnota je stejněkrát. Tedy:

$$Pr(\mathbf{P} = x|\mathbf{C} = y) = \frac{Pr(\mathbf{P} = x) \cdot \frac{1}{n}}{\frac{1}{n}} = Pr(\mathbf{P} = x)$$

Tedy kryptosystém je absolutně bezpečný.

$\Rightarrow$: Chceme dokázat, že každý absolutně bezpečný kryptosystém je latinský čtverec. Dokážeme to sporem, tedy nechť kryptosystém není latinský čtverec. Uvažujme tabulku ciphertextů, kde řádek určuje plaintext a sloupec příslušný klíč. Z definice kryptosystému víme, že v žádném sloupci se hodnoty nesmí opakovat, jelikož bychom jedním klíčem zašifrovali 2 různé plaintexty na ten samý ciphertext.

Z předpokladu ale kryptosystém není latinský, tedy musí existovat alespoň jeden řádek, kde se hodnota alespoň jednou opakuje. Označme tuto hodnotu jako y. Jelikož y je v každém sloupci maximálně jednou a existuje řádek, kde je aspoň 2x, tak musí existovat řádek, ve kterém se y nenachází. To lze díky tomu že, množiny $\mathbb{P}, \mathbb{C}, \mathbb{K}$ mají stejnou mohutnost. Označme tento řádek (plaintext) jako x.

Poté platí $Pr(\mathbf{P} = x, \mathbf{C} = y) = 0$ ale $Pr(\mathbf{P} = x) \neq 0 \wedge Pr(\mathbf{P} = x) \neq 0$. Tedy

$$Pr(\mathbf{P} = x|\mathbf{C} = y) = \frac{0}{\neq 0} = 0 \neq Pr(\mathbf{P} = x)$$

Tedy kryptosystém není absolutně bezpečný $\Rightarrow$ spor.

5

Máme najít 6 poloslabých klíčů $\Leftrightarrow$ 3 páry. Poloslabý klíč se vyznačuje tím, že se z něj při generování rundovních klíčů nagenerují pouze 2. Samotný klíč nejprve projde permutací, která ho lehce přehází a vymaže parity-check bity. Poté se klíč rozdělí na 2 poloviny ($K_i = L_i||R_i$, kde K_i je daný klíč již po aplikované permutaci) a ty se v každé rundě bitově pošoupnou a nakonec se obě poloviny spojí další permutací. Tuto druhou permutaci vůbec nemusíme uvažovat, jelikož neovlivňuje dané poloviny do dalších rund. První permutaci můžeme v podstatě také ignorovat, akorát poté použijeme její inverzi k zpětnému získání samotného klíče ve správném formátu.

Jelikož půlky klíče bitové posouvají o 1 nebo 2 bity doleva, lze jednodušše odvodit, že například z těchto 2 polovin:

$$L_1 = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$$

$$R_1 = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$$

se dají "nashiftovat" jen 2 různé stavy:

$$S_1 = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$$

$$S_2 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$$

Nyní si jen stačí uvědomit, že při dešifrovaní se klíče používají opačně. Tedy stav posunu, ve kterém budou tyto poloviny v poslední rundě při šifrování, bude stav 1 při dešifrování. Stav v poslední rundě je stejný jako na počátku, tedy druhý klíč do páru získáme inverzním posunem (v tomto případě to je jedno) o bit doprava. Druhý klíč tedy bude mít poloviny:

$$L_2 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$$

$$R_2 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$$

Nyní již stačí aplikovat inverzní permutaci na tyto 2 klíče a správně doplnit parity bity.

Takto lze vymyslet více takovýchto poloviny, které skoro nic nedělají a aplikovat stejný postup.

Náš první pár klíču je (hexadecimálně):
(`0x1fe01fe01fe01fe`, `0xfe01fe01fe01fe01`)
Další 2 nalezené páry:
(`0x1fe01fe00ef10ef1`, `0xe01fe01ff10ef10e`) odpovídající polovinám:

$$L_3 = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$$

$$R_3 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$$

$$L_4 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$$

$$R_4 = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$$

(`0x1ffe1ffe0efe0efe`, `0xfe1ffe1ffe0efe0e`) odpovídající polovinám:

$$L_5 = [1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0]$$

$$R_5 = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$$

$$L_6 = [0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1]$$

$$R_6 = [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]$$

6

`0xf00dbeef1334 = 111100000000110110111110111011110001001100110100`

$$B_1 = 111100, B_2 = 000000, B_3 = 110110, B_4 = 111110,$$

$$B_5 = 111011, B_6 = 110001, B_7 = 001100, B_8 = 110100$$

Nechť S_i je zobrazení i-tého sboxu. Provedeme $\forall i \in \{1, \ldots, 8\}\ S_i(B_i)$. Tedy například:

$$S_1(B_1) = 0101 \text{ protože řádek: } 10_b = 2, \text{ sloupec: } 1110_b = 14$$

$$\text{a na tomto místě v tabulce je hodnota } 5 = 0101_b$$

Výsledek sestavíme ze spojení

$$S_1(B_1)||S_2(B_2)||\ldots||S_8(B_8) = 01011111110001000100101110001010_b =$$

$$= \texttt{0x5fc44b8a}$$