# Лекция №13

## Криптосистема Мак Элиса

## LDPC коды

## I. Криптосистема МакЭлиса

Предложена МакЭлисом в 1978г. (основана на кодах Гоппы)

Сегодня: модификация, предложенная Нидеррайтером, на основе кода Гоппы

• KeyGen $(pp = n, k, t)$

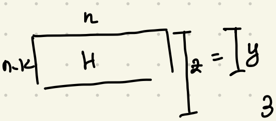
1. $C$ – лин. код с пар-ми $[n, k, d = 2t+1]$ над $\mathbb{F}_2$ с эффективным алг-мом декодирования.
   $C$ – код Гоппы
2. $H \in \mathbb{F}_2^{n-k \times n}$ – проверочная матрица для $C$
3. Сгенерировать $S \in \mathbb{F}_2^{n-k \times n-k}$ – случ. несингулярная матрица
4. Сгенерировать $P \in \mathbb{F}_2^{n \times n}$ – случайная матрица перестановки
5. $pk = H' = S \cdot H \cdot P$
   $sk = (S, P, H)$

• Enc $(m \in \{0,1\}^n,\ wt(m) = t,\ pk)$

1. $c = H' m \in \mathbb{F}_2^{n-k}$

• Dec $(c,\ sk = (S, P, H))$

1. $y = S^{-1} \cdot c \in \mathbb{F}_2^{n-k}$ // $c = H'm = S \cdot H \cdot P\, m$
   $S^{-1}c = H \cdot P \cdot m$
2. Найти с помощью алг-ма Гаусса $z \in \mathbb{F}_2^n$, такой, что



$H \cdot z = y = H \cdot P \cdot m$

3. Decode$(z)$ – алг-м декодирования для кода $C$ (например, алг-м декодирования Гоппы) $H \cdot z = H(c' + e') = He'$
   В результате получим $e'$ – вектор ошибки $(z = c' + e')$

4. $e' = P\cdot m$ $\quad (P^{-1} = P^t$ т.к. $P$ - матрица перестановки)

$m = P^t \cdot e'$

Корректность $\exists\, c = H' \cdot m$ - корректно сформированный шифр-текст

Тогда алг-м Dec вычисляет:

1. $y = S^{-1} \cdot c = H\cdot P\cdot m$
2. $z$: $\quad y = H\cdot z$

$$H\cdot P\cdot m = H\cdot z$$

$$H\underbrace{(Pm - z)}_{\Downarrow} = 0$$

$(P\cdot m - z)$ - кодовое слово в $C$, $wt(Pm) = wt(m) = t$

$\Rightarrow \Delta(z, \underset{\substack{\uparrow \\ \text{кодовое слово в } C}}{x}) = t \Rightarrow$ Decode $(z)$ вернёт вектор ошибки $z - x = P\cdot m \Rightarrow$

$\Rightarrow \underbrace{P^t \cdot P}_{Id_n} \cdot m = m.$

## II Коды с малой плотностью проверки на чётность (Low-density parity-check codes, LDPC)

802.3 Ethernet
802.11 Wireless Lan

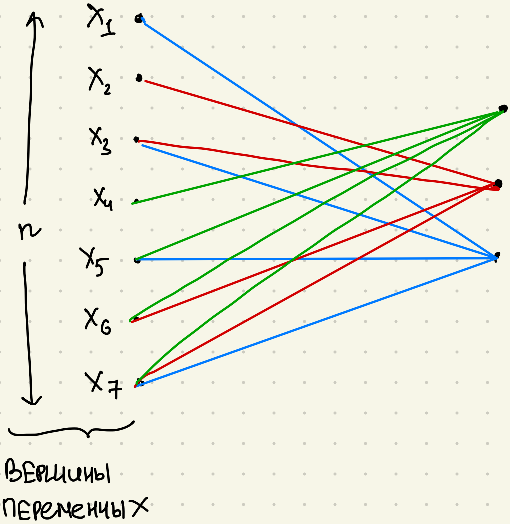
1. Мотивация и опр-ия улучшить алг-м декодирования, а не min. расстояние

LDPC - код на графах

$\forall$ лин. код может быть представлен в виде двудольного графа, т.е. графа, мн-во вершин которого можно разбить на два ~~А~~ мн-ва $U, V$, таких что рёбра графа соединяют вершины из $U$ только с вершинами в $V$.

Пример $[7, 4, 3]_2$ - код Хэмминга

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} x_4 + x_5 + x_6 + x_7 \\ x_2 + x_3 + x_6 + x_7 \\ x_1 + x_3 + x_5 + x_7 \end{bmatrix}$$




$c_1 = x_4 + x_5 + x_6 + x_7$

$c_2 = x_2 + x_3 + x_6 + x_7$

$c_3 = x_1 + x_3 + x_5 + x_7$

Коды LDPC соответствуют "разреженным" (sparse) графам, т.е. графам с малым кол-вом рёбер; экв-но, проверочная матрица LDPC кода содержит число 1-ц в каждой строке $\ll n$, число 1-ц в каждом столбце $\ll n-K$.

## 2. Жёсткое декодирование LDPC кодов (метод вероятностного итеративного декодирования)

$n = 8$

$n-K = 6$

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

Полученное слово $y = [0\ 1\ 0\ 0\ 1\ 0\ 0\ 1]$

# Пример

## Раунд 1

| | | | Раунд 1 |
|---|---|---|---|
| $\begin{array}{ccc} c_1 & c_2 & c_3 \end{array}$ $[1, 1, 0]$ | $x_1 \cdot \boxed{0}$ | | |
| $\begin{array}{ccc} c_4 & c_5 & c_6 \end{array}$ $[1, 0, 1]$ | $x_2 \cdot \boxed{1}$ | $C_1 = \overset{0}{x_1} + \overset{0}{x_4} + \overset{1}{x_5} + \overset{0}{x_6}$ | $\overset{0}{x_1} + \overset{0}{x_4} + \overset{0}{x_5} + \overset{0}{x_6}$ |
| $\begin{array}{ccc} c_2 & c_3 & c_6 \end{array}$ $[1, 0, 0]$ | $x_3 \cdot \boxed{0}$ | $C_2 = \overset{0}{x_1} + \overset{0}{x_3} + \overset{1}{x_5} + \overset{0}{x_6}$ | $\overset{0}{x_1} + \overset{0}{x_3} + \overset{0}{x_5} + \overset{0}{x_6}$ |
| $\begin{array}{ccc} c_1 & c_3 & c_4 \end{array}$ $[1, 0, 0]$ | $x_4 \cdot \boxed{0}$ | $C_3 = \overset{0}{x_1} + \overset{0}{x_3} + \overset{0}{x_4} + \overset{0}{x_6}$ | $\overset{1}{x_1} + \overset{0}{x_3} + \overset{0}{x_4} + \overset{0}{x_6}$ |
| $\begin{array}{ccc} c_1 & c_2 & c_5 \end{array}$ $[0, 0, 0]$ | $x_5 \cdot \boxed{1}$ | $C_4 = \overset{1}{x_2} + \overset{0}{x_4} + \overset{0}{x_7} + \overset{1}{x_8}$ | $\overset{1}{x_2} + \overset{0}{x_4} + \overset{0}{x_7} + \overset{1}{x_8}$ |
| $[1, 0, 0]$ | $x_6 \cdot \boxed{0}$ | $C_5 = \overset{1}{x_2} + \overset{1}{x_5} + \overset{0}{x_7} + \overset{1}{x_8}$ | $\overset{1}{x_2} + \overset{0}{x_5} + \overset{0}{x_7} + \overset{1}{x_8}$ |
| $[0, 1, 0]$ | $x_7 \cdot \boxed{0}$ | $C_6 = \overset{1}{x_2} + \overset{0}{x_3} + \overset{0}{x_7} + \overset{1}{x_8}$ | $\overset{1}{x_2} + \overset{0}{x_3} + \overset{0}{x_7} + \overset{1}{x_8}$ |
| $[1, 0, 1]$ | $x_8 \cdot \boxed{1}$ | | |

В раунде 2, проверка $c_3$ исправит $x_1$ на $0 \Rightarrow$

$\Rightarrow$ ур-ие проверки на чётность $c_3$ будет выполнено.

Ошибка в $x_5$ $(y_5)$ $1 \to 0$

# Алгоритм

**I. Раунд 0**: 1. Вершины $x_i$ получают значения $y_i$

2. $x_i$ "посылает" $y_i$-ы смежным вершинам $c_j$

**II Раунд $i$**

1. Для всех вершин $c$:
   Вершина $c$ посылает смежному $x$ сумму всех полученных значений mod 2 за исключением бита, полученного от самого $x$.

2. Для всех вершин $x$:
   Вершина $x$ посылает смежной вершине бит $b$, если $x$ получила $b$ ото всех вершин, кроме $c$, иначе (т.е. если хотя бы одно значение не совпадают), $x$ посылает полученный бит от $y$.

**III.** Повторять шаг II пока все условия проверки на чётность не будут удовлетворены.