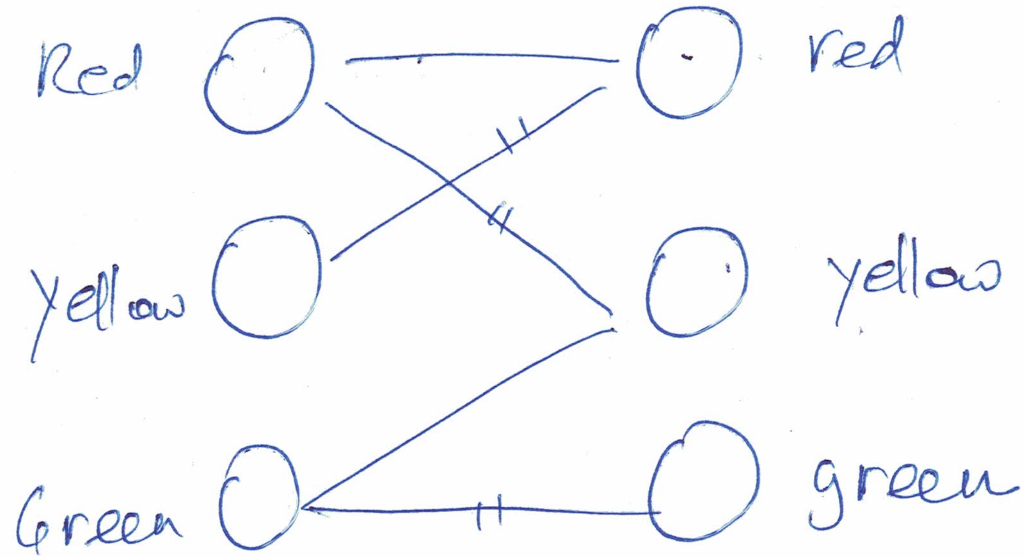
Cpts 515. 9/23/2020.

Applications of bipartite graphs in security.



max matching

$M = 3$

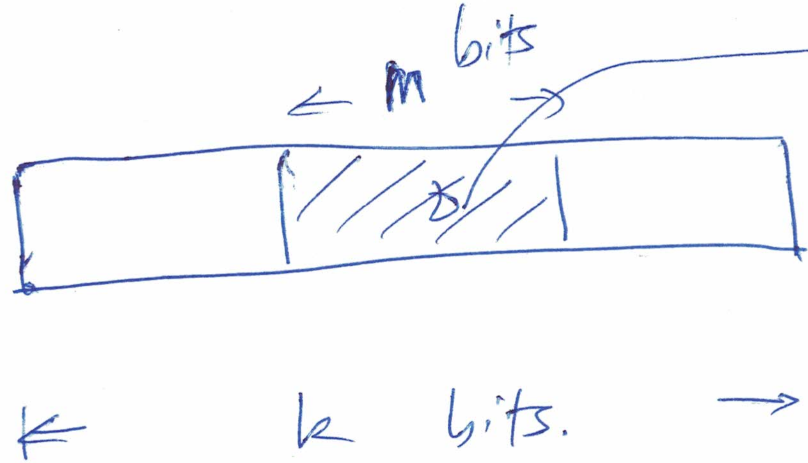
HIGH

X

LOW

Y

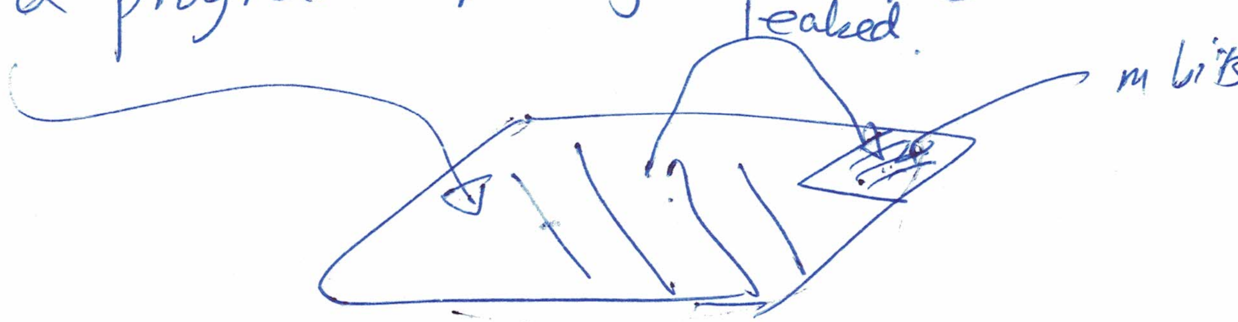
how much info shared between X & Y?



I can observe this partition only.

Qustn: How much info has been leaked into the m-bits from the k bits?

a program runs on k-bits.



LOW, HIGH : $\mathcal{C} \longrightarrow \mathbb{Z}$ integer.

HIGH: $\boxed{x_1 \mid x_2 \mid \cdot \mid \cdot \mid \cdots \mid x_k}$ k bits.

k-bit unsigned

LOW: $\sum_{i=1}^{k} x_i$ a number.

Example: $k=4$. $0101 = 5 = $ HIGH

while LOW $= 2$.

Question: how much info leaked from HIGH to LOW?

info-leaked: shared information.

X, Y are random vars.

How much randomness / information in X ?

From shannon's entropy:

$$H(X) = -\sum_{x} p(x) \log p(x).$$

Information = The information that we don't know.

= uncertainty.

basic unit of information = bit

invented by shannon.

Let X be a fair coin.

Probality (X = head) = 50%

Probality (X = tail) = 50%.

---

how much informati in a fair coin <u>before</u> it's tossed?

$$H(X) = -\sum_{x} p(x) \log p(x)$$

$$= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2}$$

$$= 1 \quad \text{bit.}$$

Let X be a coin with

$$\text{Prob}(X = \text{head}) = 1\%;$$

$$\text{Prob}(X = \text{tail}) = 99\%.$$

Then. $H(X) = -0.01 \log 0.01 - 0.99 \log 0.99$

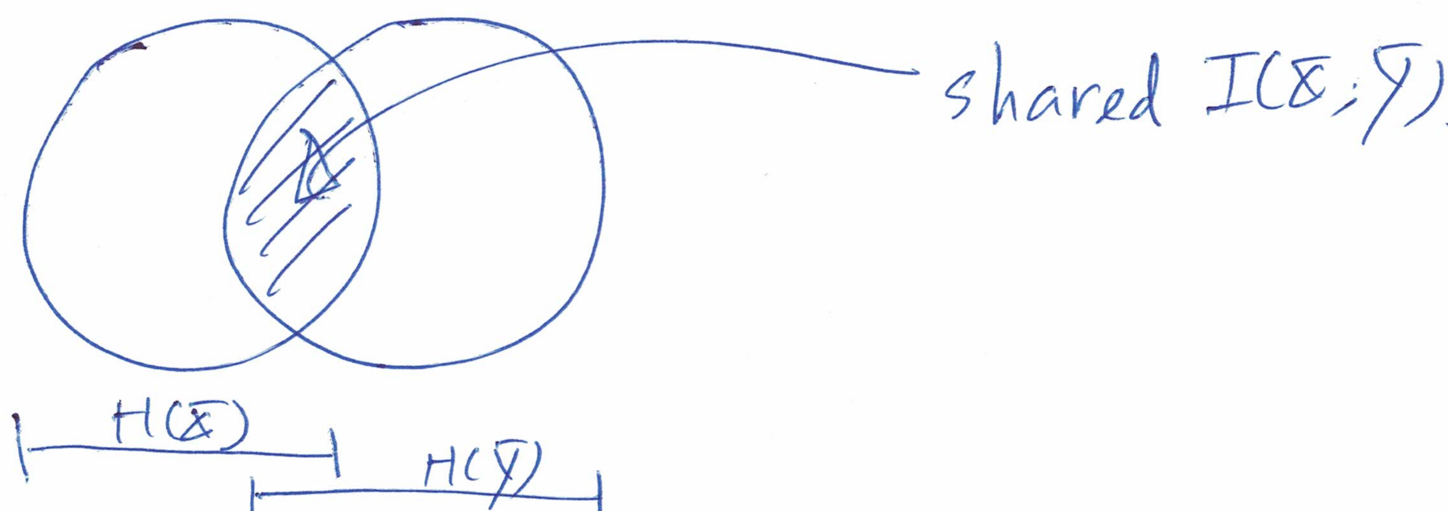
is close to 0.

$$H(X,Y) = -\sum_{x,y} p(x,y)\log p(x,y)$$

joint entropy

Shared entropy / information between X and Y:

$$I(X;Y) = H(X) + H(Y) - H(X,Y).$$



G: a bipartite graph. with joint distribution

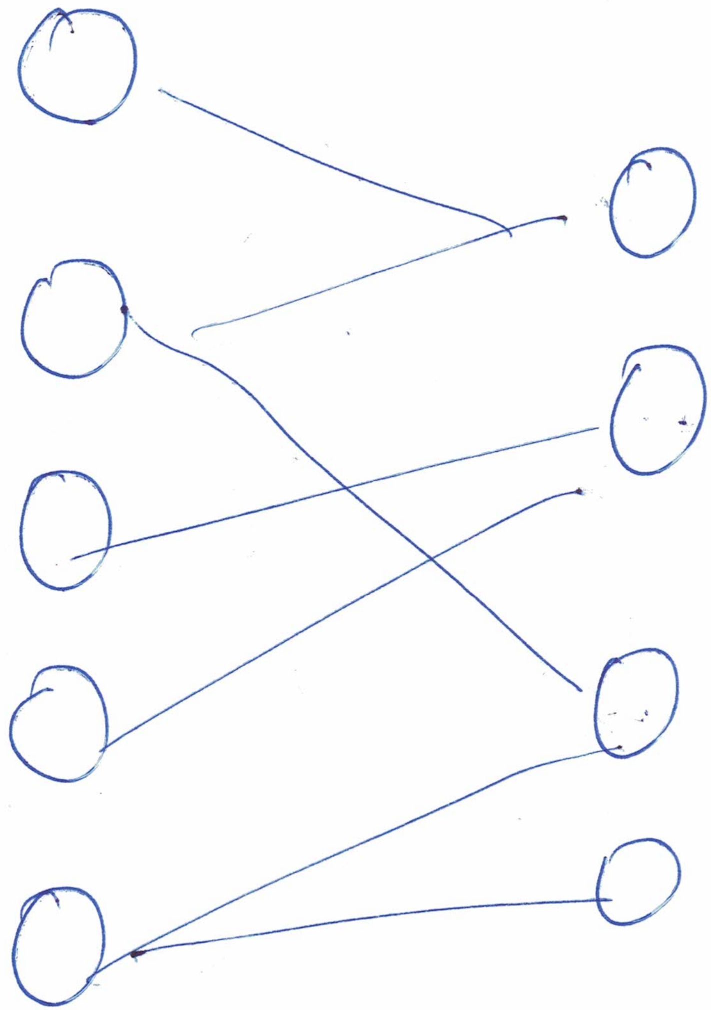
$p(X, Y)$.

s.t. if $(u,v) \notin E$

then

$p(u,v) = 0$

for all nodes u, v.



X

(high)

Y

(Low).

Then, from the $p(\bar{X}, \bar{Y})$ we can compute $I(\bar{X};\bar{Y})$. We define

$$\lambda_G = \max_{p(\cdot,\cdot)} I(\bar{X};\bar{Y}).$$

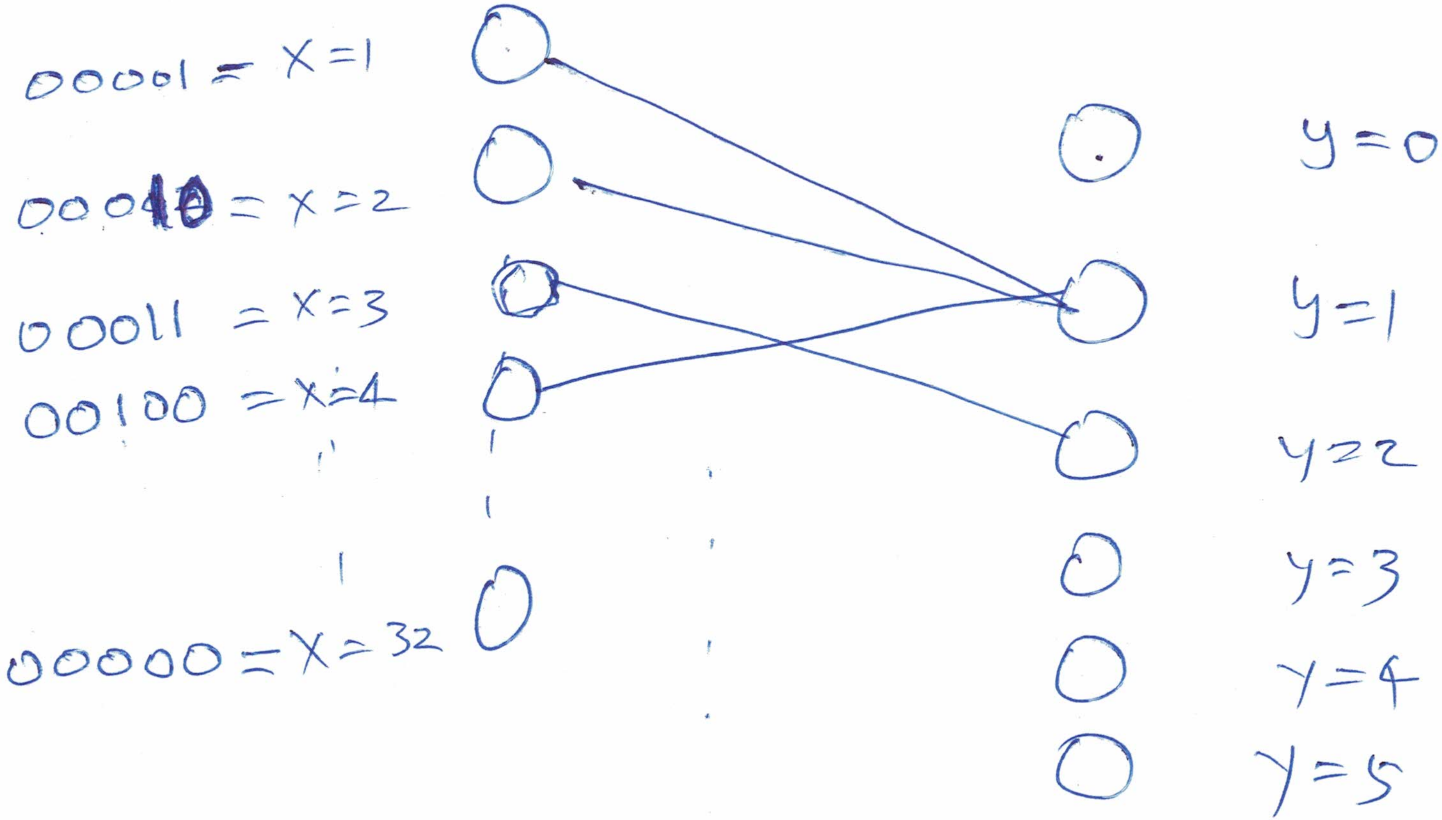
Theorem: $\lambda_G = \log_2 M.$

max. matching.

$X = \langle x_1, \dots, x_5 \rangle$. 5 bits

$Y = \sum_{i=1}^{5} x_i$ sum of 5 bits.

info lecked frm X to Y:

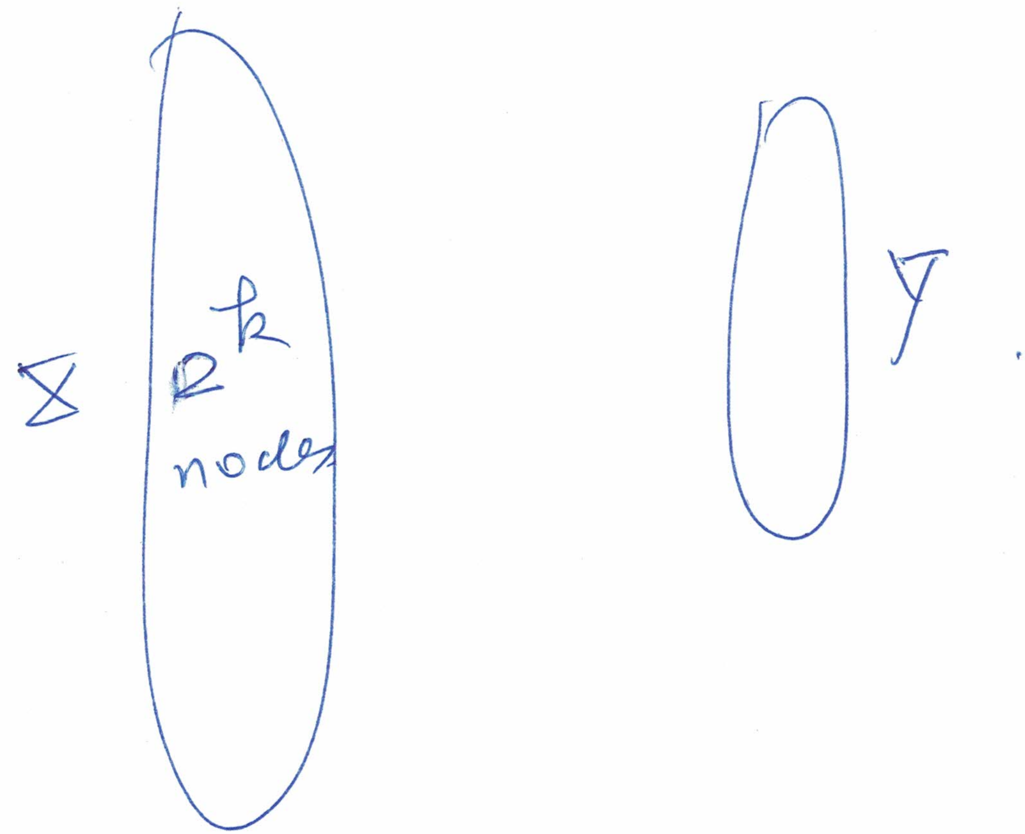


I compute the size of max. matching M.

Then answer is $\log_2 M$.

When the k is large.

$$X = \langle x_1, \ldots, x_k \rangle$$
$$Y = \sum_{i=1}^{k} x_i \ldots \quad \Rightarrow G \text{ is unbelievable large.}$$



In particular,

the k is a function of time n. In this case, the graph G can be expanding over the time (X is a variable over memory of k bits, and the mem size grows with time n).

We can approximate the rate of info-leak:

$$\lambda_G = \frac{1}{n} \log \frac{N_{left} \cdot N_{right}}{E}, \quad \text{as } n \to \infty.$$

(take limsup).