Quantitative Information Flow

Catuscia Palamidessi
INRIA Saclay & Ecole Polytechnique

These lectures are based on work done in collaboration with:
Mario Alvim, **Miguel Andrés**, and **Konstantinos Chatzikokolakis**

Plan of the lectures

- **Lecture 1 (Monday)**
  - Secure information flow. Motivations and examples
  - Information-theoretic framework
- **Lecture 2 (Tuesday)**
  - Quantification of leakage: models of adversaries
  - Focus on: Shannon entropy and Rényi min-entropy
  - Bayes risk
- **Lecture 3 (Friday)**
  - Differential privacy
  - Relation between QIF and differential privacy

Information Flow

Problem: Avoid the leakage of information in computer systems

Leakage of secret information via public observables

Ideally: No leak

- No interference [Goguen & Meseguer'82]

In practice: There is almost always some leak

- Intrinsic to the problem
- Side channels

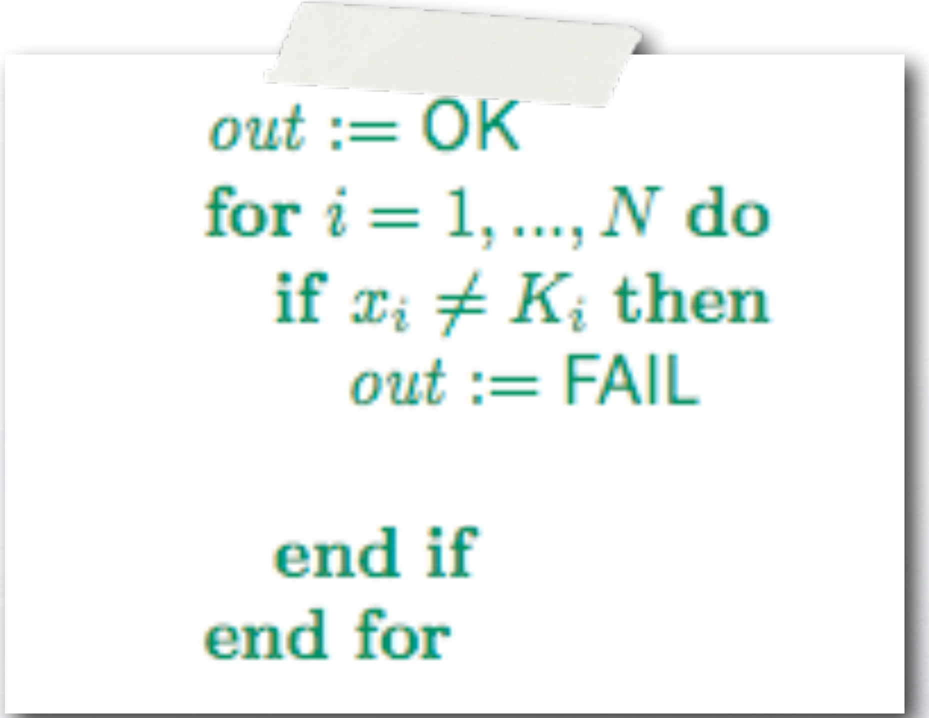
Example

Password checker 1

Password: $K_1 K_2 \ldots K_N$
Input by the user: $x_1 x_2 \ldots x_N$
Output: out (Fail or OK)

Intrinsic leakage

By learning the result of the check the adversary learns something about the secret



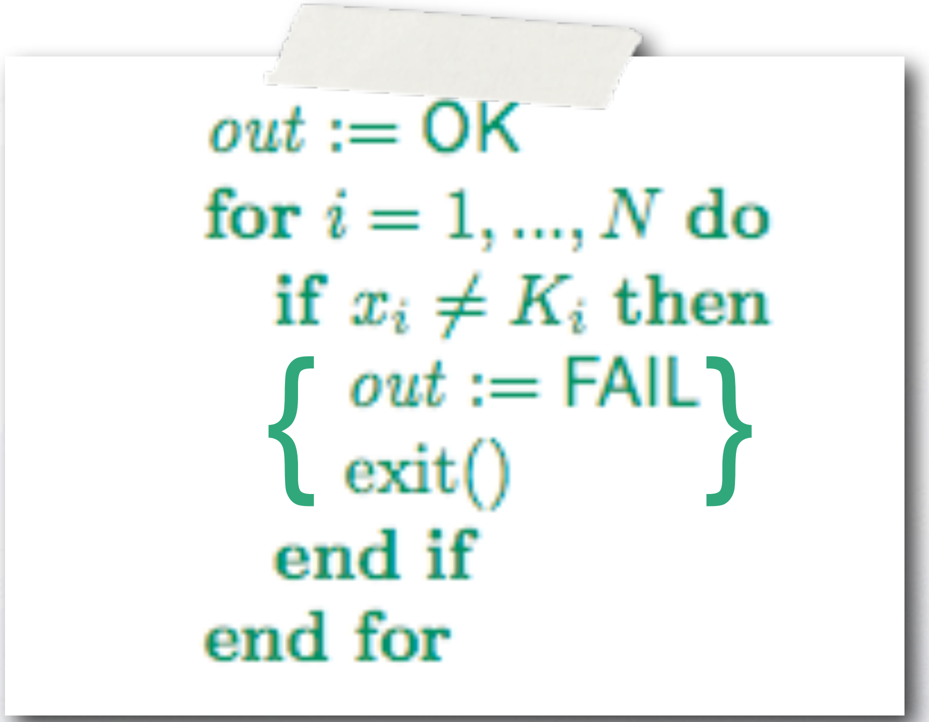
```
$out$ := OK
for $i = 1, \ldots, N$ do
    if $x_i \neq K_i$ then
        $out$ := FAIL

    end if
end for
```

Example

Password checker 2

Password: $K_1 K_2 \ldots K_N$
Input by the user: $x_1 x_2 \ldots x_N$
Output: out (Fail or OK)

More efficient, but what about security?



```
$out$ := OK
for $i = 1, ..., N$ do
  if $x_i \neq K_i$ then
    { $out$ := FAIL
      exit() }
  end if
end for
```

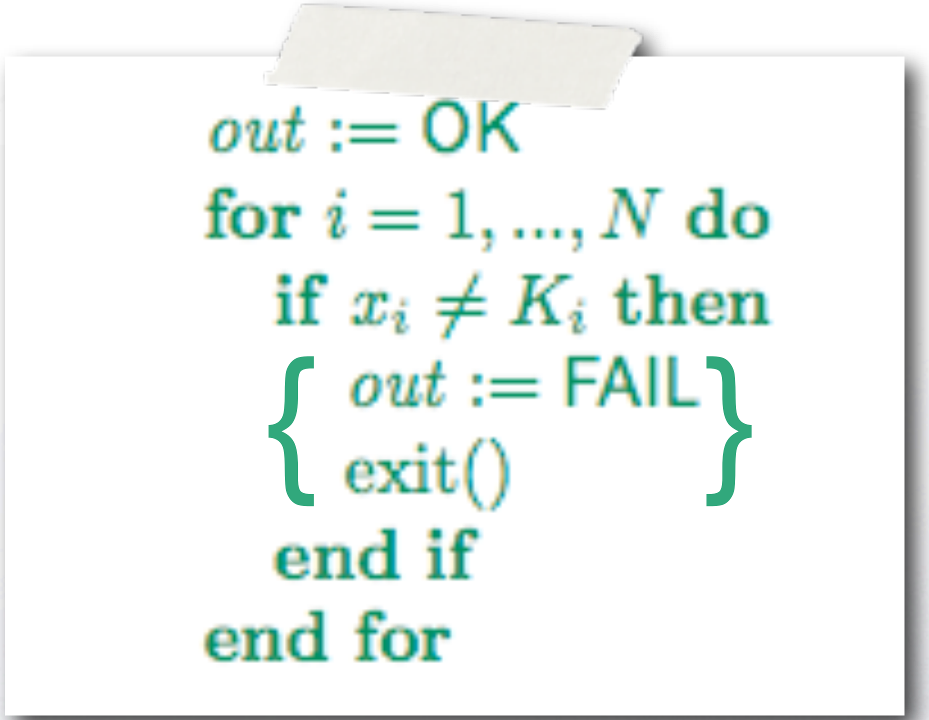
Example

Password checker 2

Password: $K_1 K_2 \ldots K_N$
Input by the user: $x_1 x_2 \ldots x_N$
Output: out (Fail or OK)

Side channel attack

If the adversary can measure the execution time, then he can also learn the longest correct prefix of the password



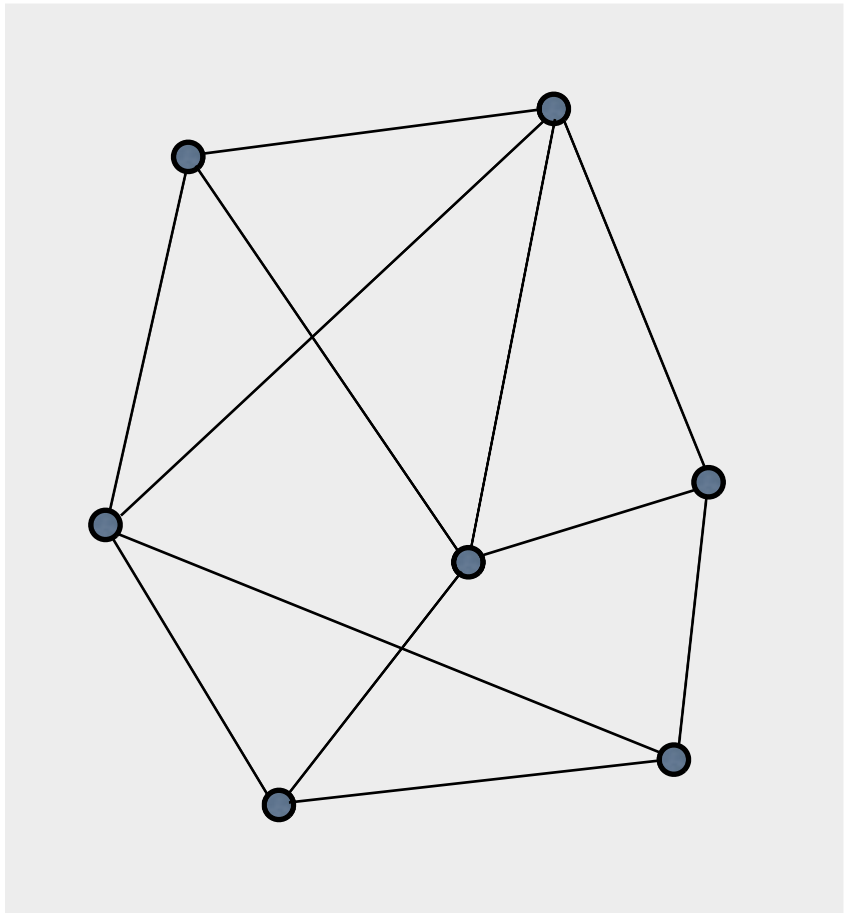
```
$out$ := OK
for $i = 1, ..., N$ do
  if $x_i \neq K_i$ then
    { $out$ := FAIL
      exit() }
  end if
end for
```

Quantitative Information flow

- It is necessary to **quantify** the notion of **Information Leakage**
- To this purpose, most of the recent proposals use **information-theoretic approaches**
  - Suitable also for **probabilistic** programs
- Convergence with other fields, in particular that of **anonymity protocols** which typically use randomization to hide the secrets
  - **secret = culprit's identity**

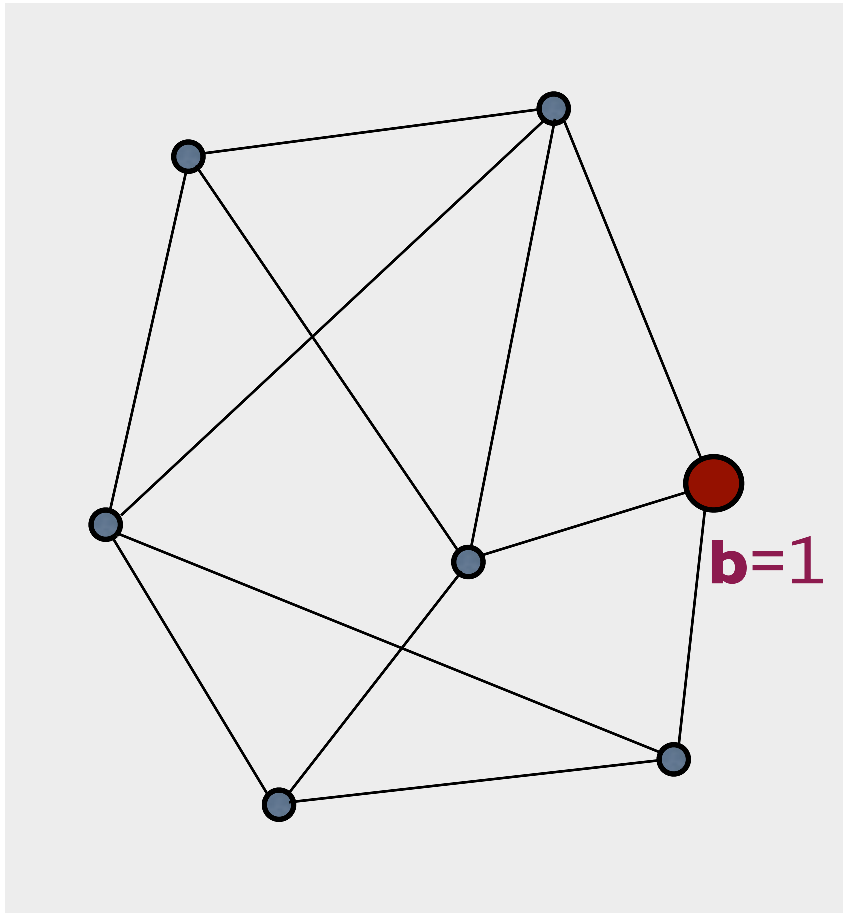
Example of Anonymity Protocol: DC Nets [Chaum'88]

- A set of nodes with some communication channels (edges).
- One of the nodes (source) wants to broadcast one bit **b** of information
- The source must remain anonymous



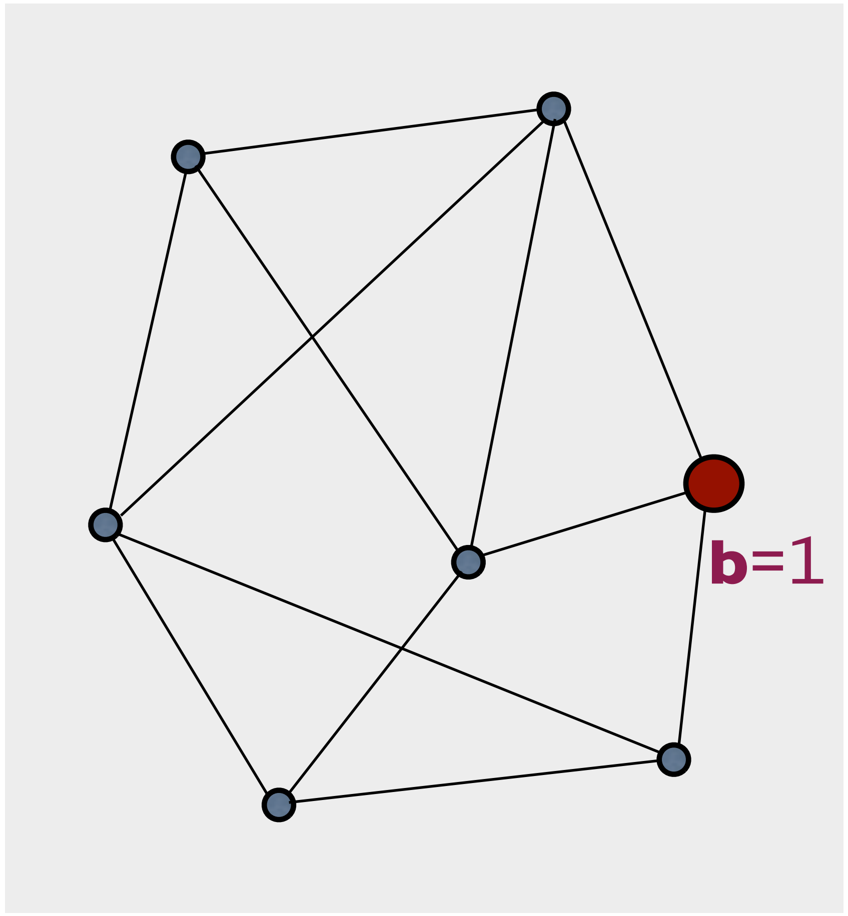
Example of Anonymity Protocol: DC Nets [Chaum'88]

- A set of nodes with some communication channels (edges).
- One of the nodes (source) wants to broadcast one bit **b** of information
- The source must remain anonymous



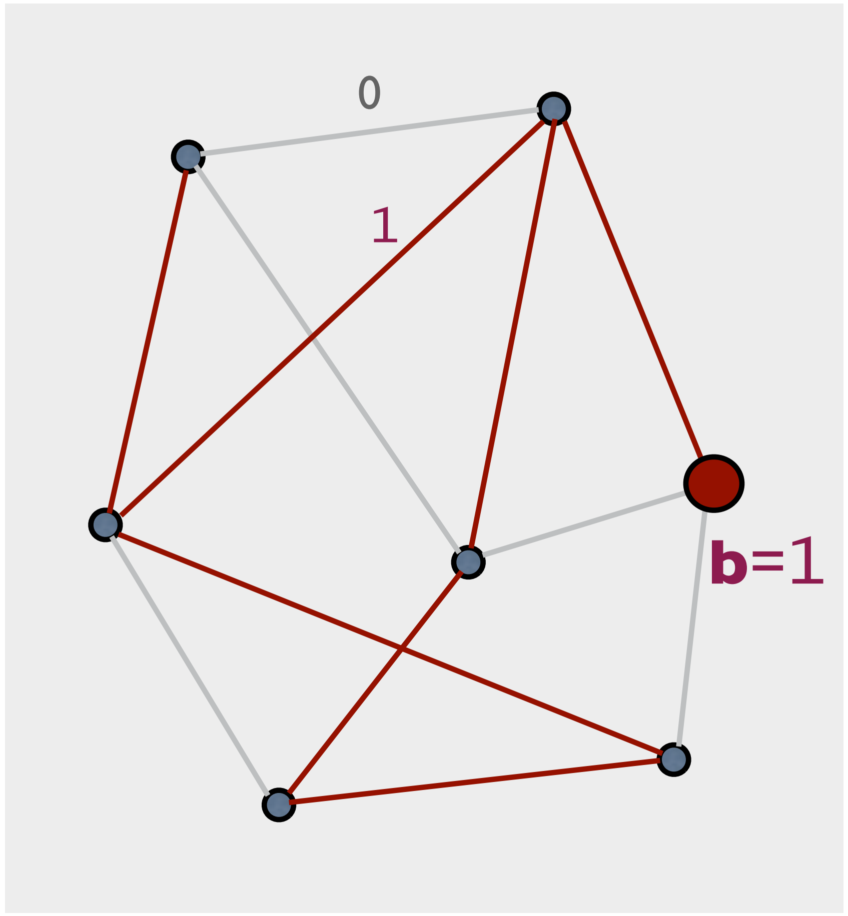
Chaum's solution

- Associate to each edge a fair coin



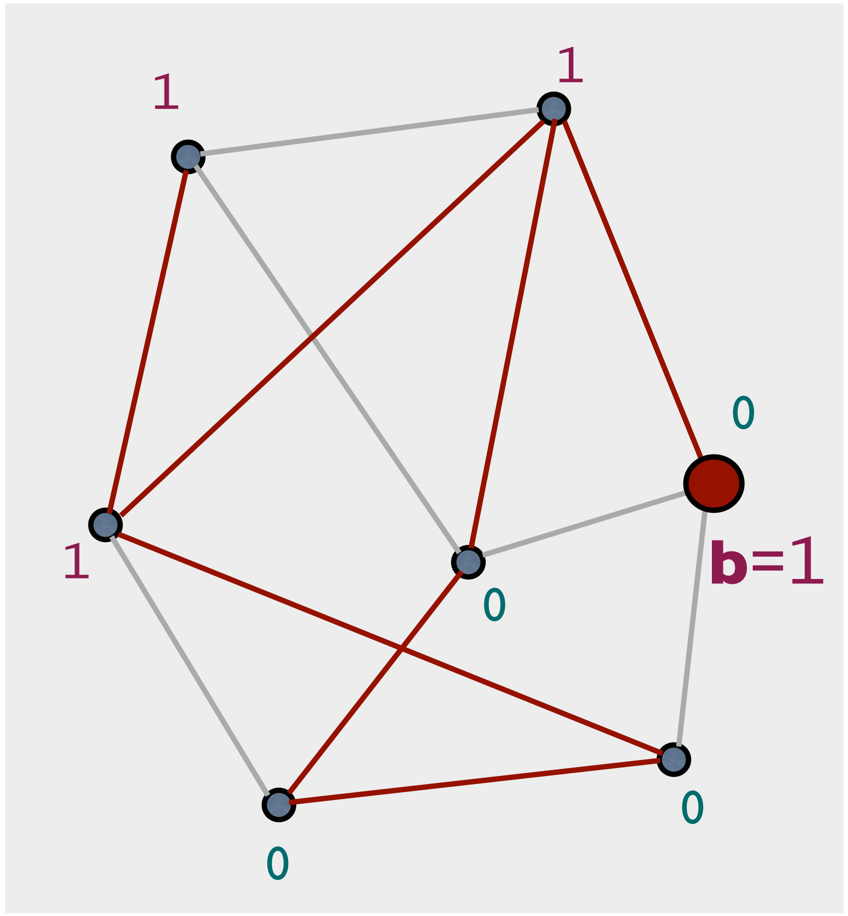
Chaum’s solution

- Associate to each edge a fair coin
- Toss the coins



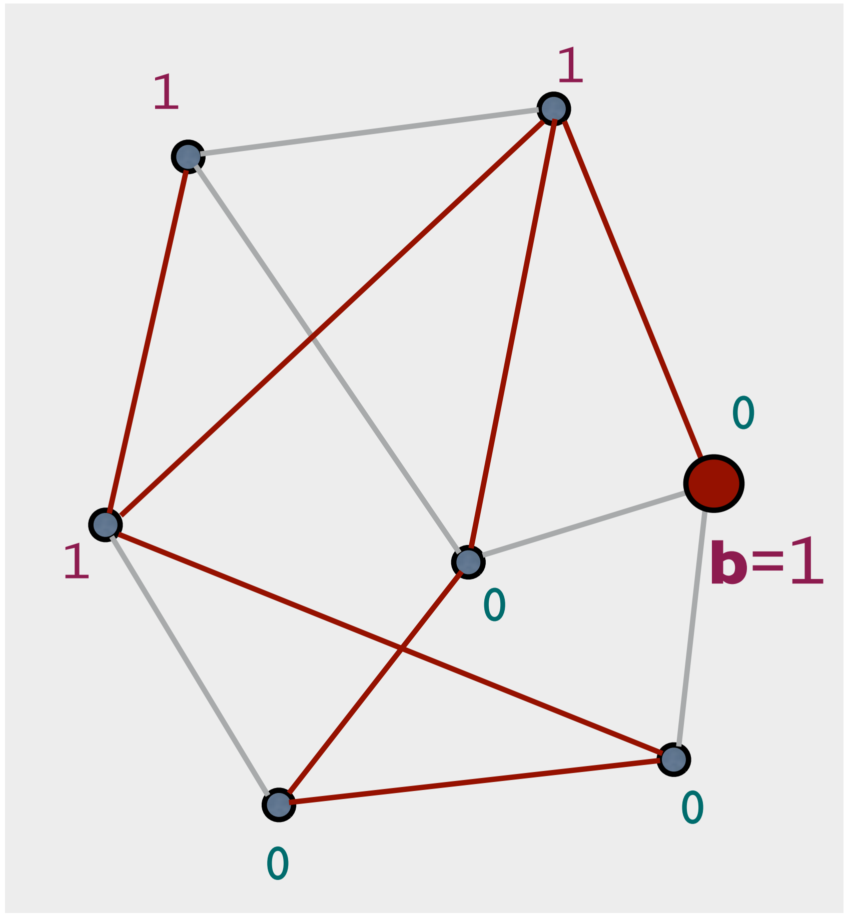
Chaum's solution

- Associate to each edge a fair coin
- Toss the coins
- Each node computes the binary sum of the incident edges. The source adds **b**. They all broadcast their results



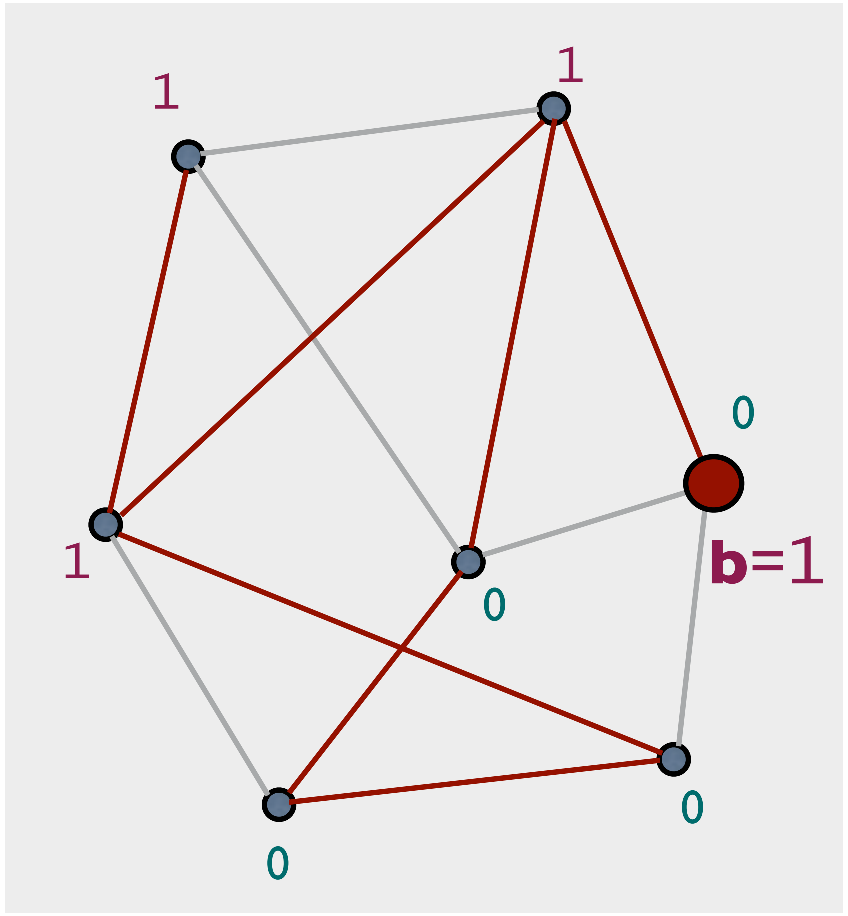
Chaum’s solution

- Associate to each edge a fair coin
- Toss the coins
- Each node computes the binary sum of the incident edges. The source adds **b**. They all broadcast their results
- Achievement of the goal: Compute the total binary sum: it coincides with **b**



Chaum’s solution

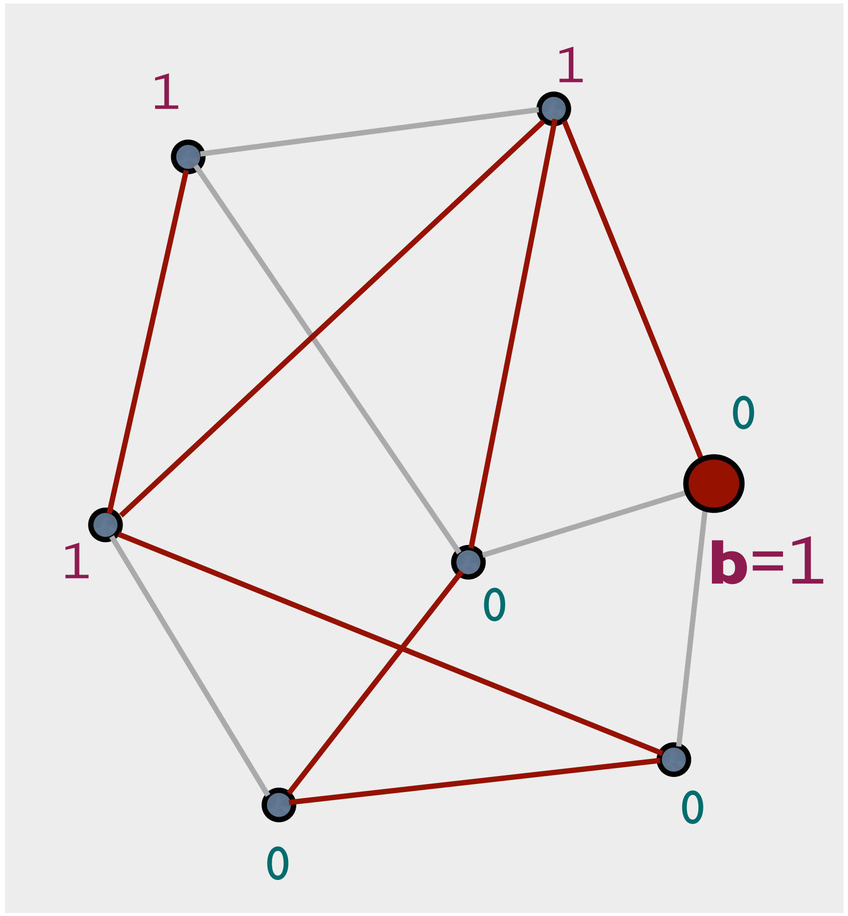
- Associate to each edge a fair coin
- Toss the coins
- Each node computes the binary sum of the incident edges. The source adds **b**. They all broadcast their results
- Achievement of the goal: Compute the total binary sum: it coincides with **b**



Question: why is that?

Chaum's solution

- Associate to each edge a fair coin
- Toss the coins
- Each node computes the binary sum of the incident edges. The source adds **b**. They all broadcast their results
- Achievement of the goal: Compute the total binary sum: it coincides with **b**



Answer: each coin is counted twice!

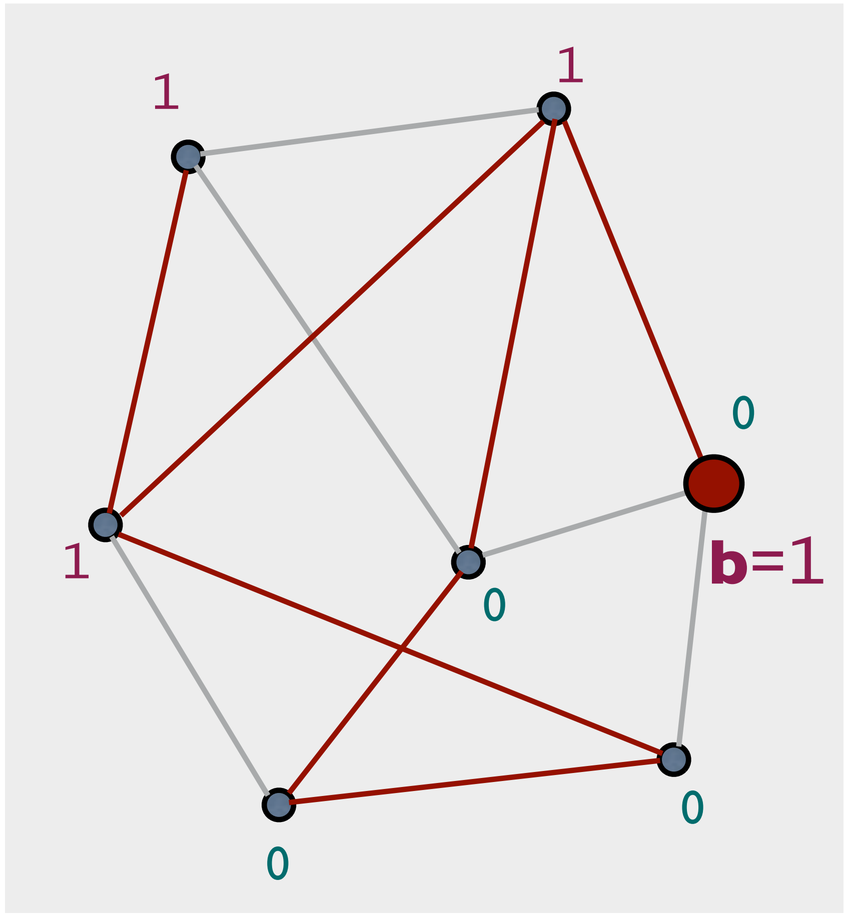
Strong anonymity (Chaum)

- If the graph is connected and the coins are fair, then for an external observer, the protocol satisfies **strong anonymity**:

the *a posteriori* probability that a certain node is the source is equal to its *a priori* probability

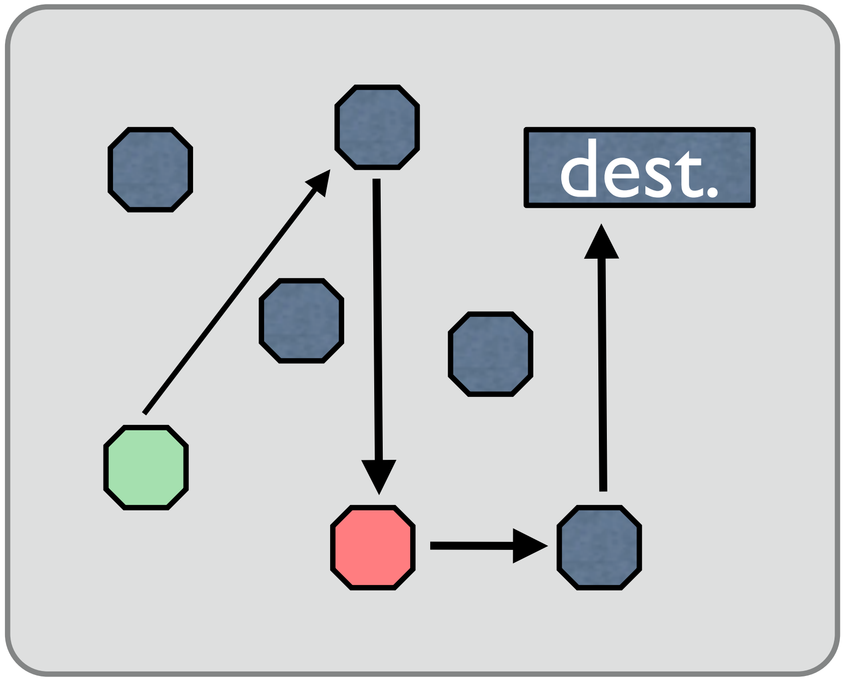
  - A priori / a posteriori = before / after observing the declarations

- Question: what about the internal nodes?



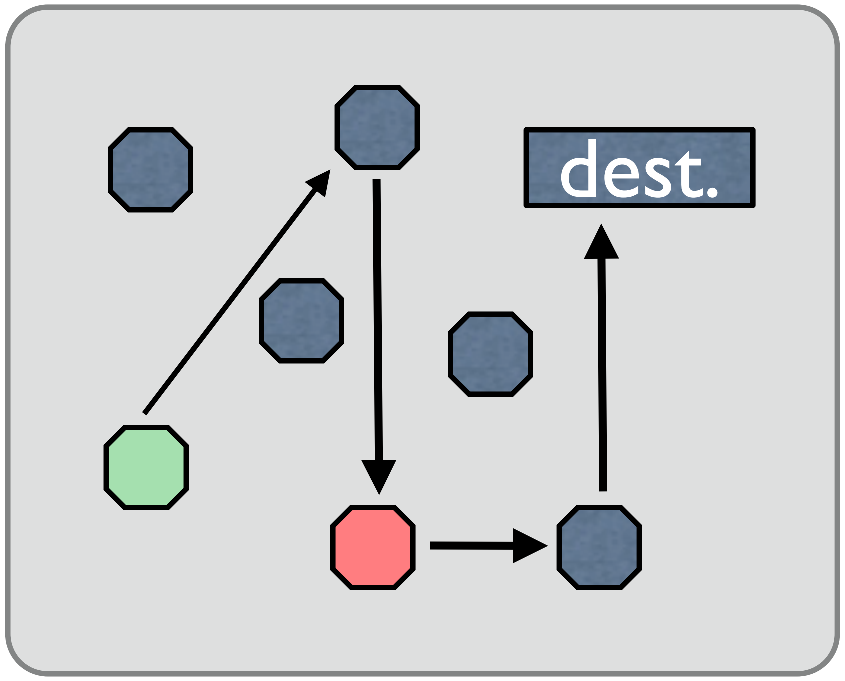
Another example: Crowds [Rubin and Reiter'98]

- Problem: A user (initiator) wants to send a message anonymously to another user (dest.)
- Crowds: A group of n users who agree to participate in the protocol.
- The initiator selects randomly another user (forwarder) and forwards the request to her
- A forwarder randomly decides whether to send the message to another forwarder or to dest.
- ... and so on



Another example: Crowds [Rubin and Reiter'98]

- Problem: A user (initiator) wants to send a message anonymously to another user (dest.)
- Crowds: A group of n users who agree to participate in the protocol.
- The initiator selects randomly another user (forwarder) and forwards the request to her
- A forwarder randomly decides whether to send the message to another forwarder or to dest.
- ... and so on



Probable innocence: under certain conditions, an attacker who intercepts the message from x cannot attribute more than 0.5 probability to x to be the initiator

Common features

- Secret information
  - the values of the high variables
  - DC: the identity of the broadcaster
  - Crowds: the identity of the initiator
- Public information (Observables)
  - the values of the low variables
  - DC: the declarations
  - Crowds: the interception of a forwarder by a corrupted user
- The system may be probabilistic
  - often the system uses randomization to obfuscate the relation between secrets and observables
  - DC: coin tossing
  - Crowds: random forwarding to another user

Simplifying assumptions

In this lectures we assume:

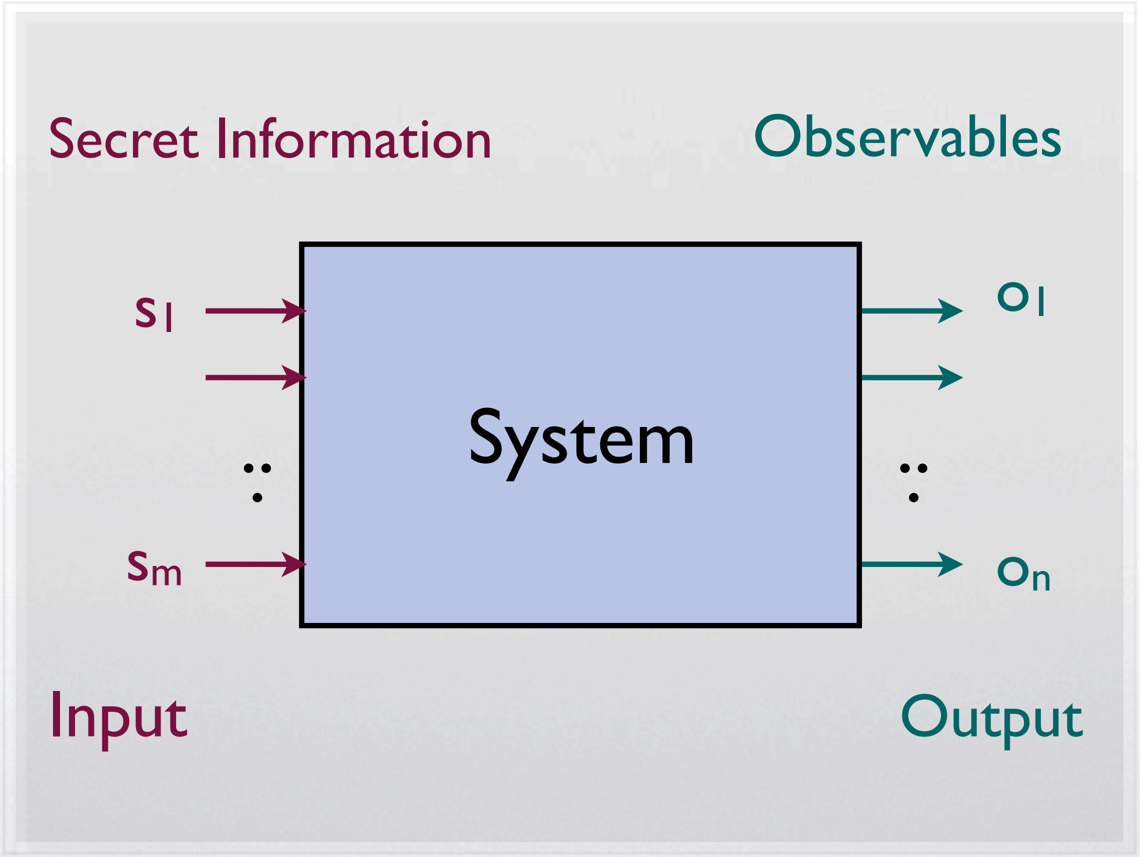
- Secrets: elements of a random variable S
- Observables: elements of a random variable O
- For each secret s, the probability that we obtain an observable o is given by p(o | s)

- No feedback: the secret is not influenced by the observables

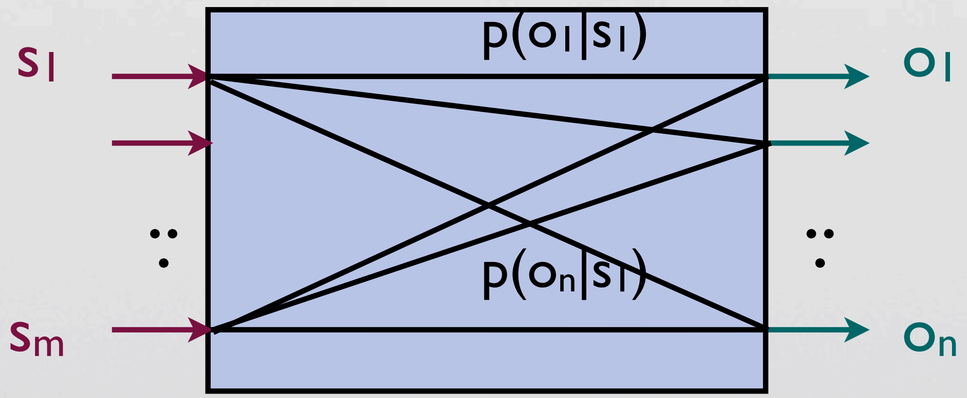
- No nondeterminism: everything is (either deterministic or) probabilistic, although we may not know the distribution

An intriguing analogy:

Systems as Information-Theoretic channels



Probabilistic systems are **noisy** channels:
an output can correspond to different inputs, and
an input can generate different outputs, according to a prob. distribution



$p(o_j|s_i)$: the conditional probability to observe o_j given the secret s_i

$$
\begin{array}{c|c|c|c|}
 & o_1 & \cdots & o_n \\
\hline
s_1 & p(o_1|s_1) & \cdots & p(o_n|s_1) \\
\hline
\vdots & \vdots & & \\
\hline
s_m & p(o_1|s_m) & & p(o_n|s_m) \\
\hline
\end{array}
$$

$$p(o|s) = \frac{p(o \ and \ s)}{p(s)}$$

A channel is characterized by its matrix: the array of conditional probabilities

In a information-theoretic channel these conditional probabilities must be independent from the input distribution

This means that we can apply the i.t. approach only to systems whose behavior may depend on the secret values, but not on their distribution

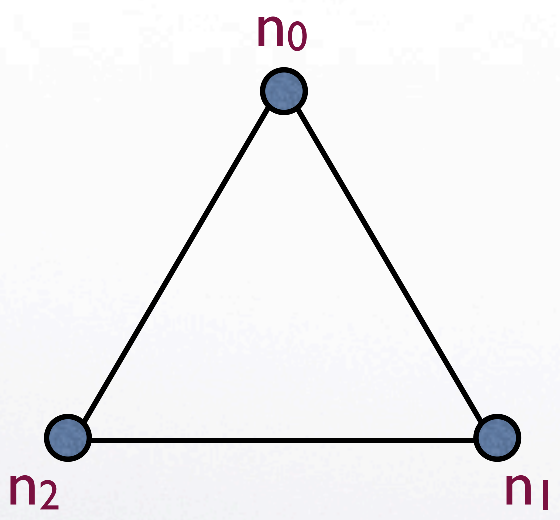
Information theory: useful concepts

- **Entropy** H(X) of a random variable X
  - a measure of the degree of uncertainty of the events
  - It can be used to measure the vulnerability of the secret, i.e. how "easy" is for the adversary to obtain the secret
- **Mutual information** I(S;O)
  - degree of correlation between the input S and the output O
  - formally defined as difference between the entropy of S ***before*** knowing O, and the entropy of S ***after*** knowing O
  - aka the difference between the a priori and the a posteriori entropy of S
  - It can be used to measure the leakage:

$$\text{Leakage} = I(S;O) = H(S) - H(S|O)$$

  - H(S|O) can be computed using the distribution of S and the matrix

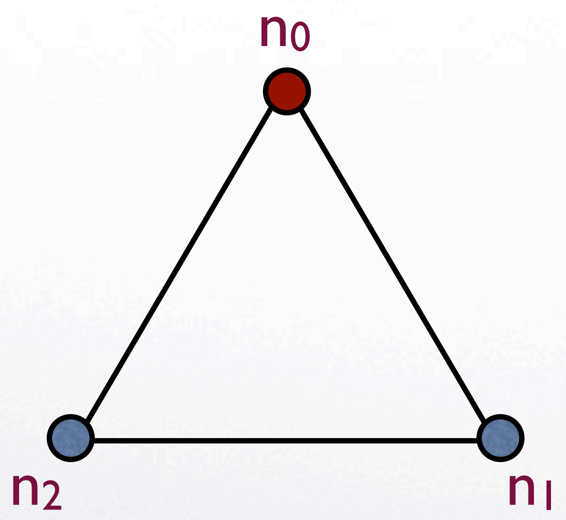
Example: DC nets (ring of 3 nodes, b=1)

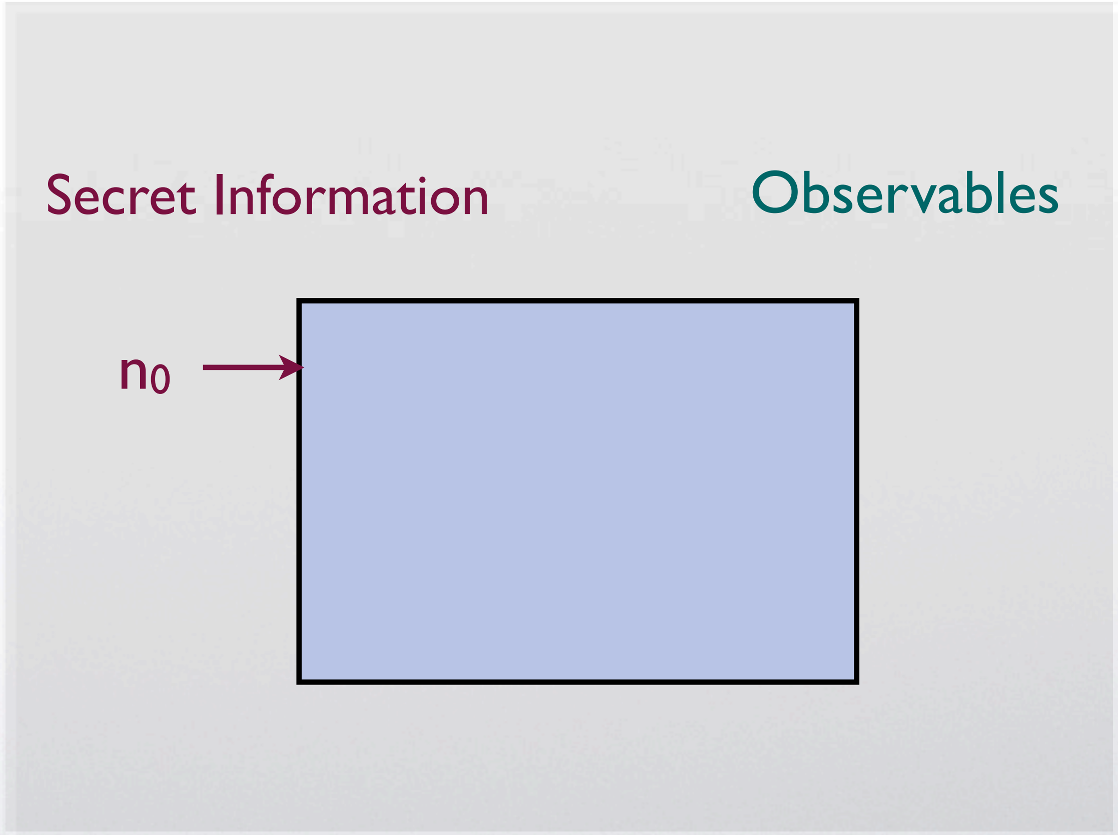


Secret Information

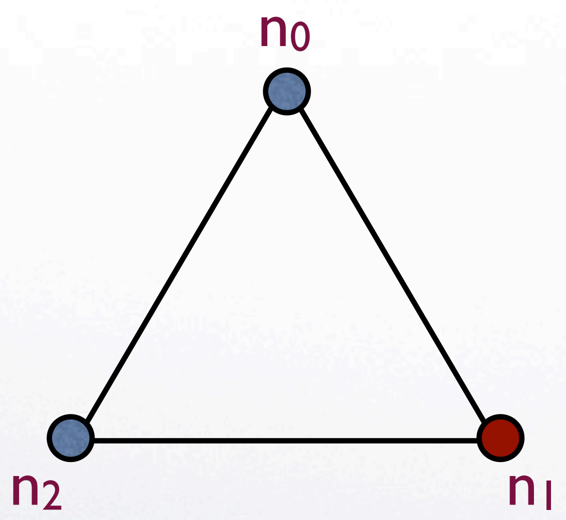
Observables

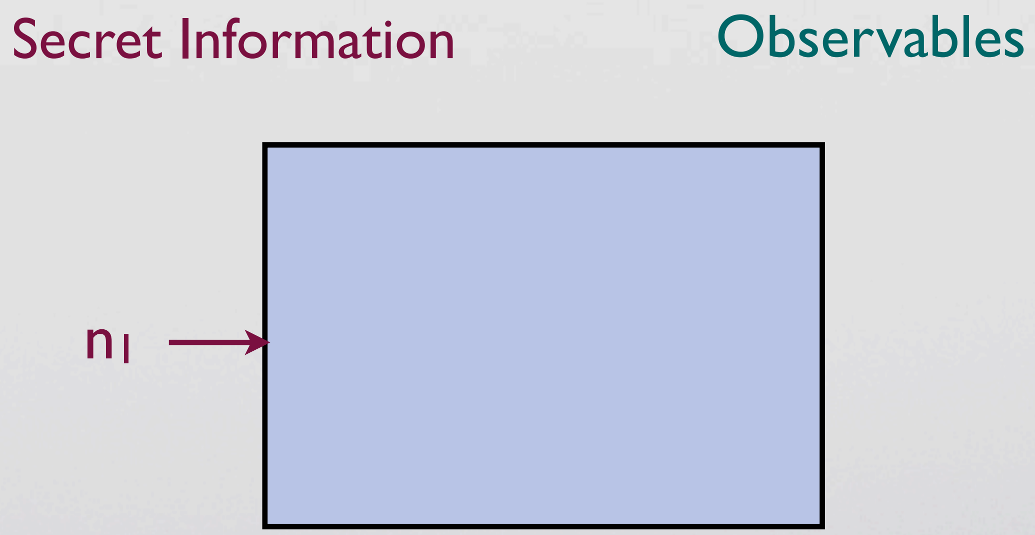
Example: DC nets (ring of 3 nodes, b=1)



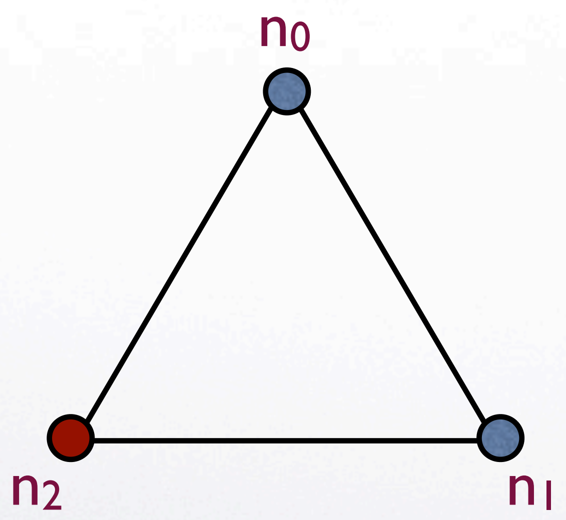


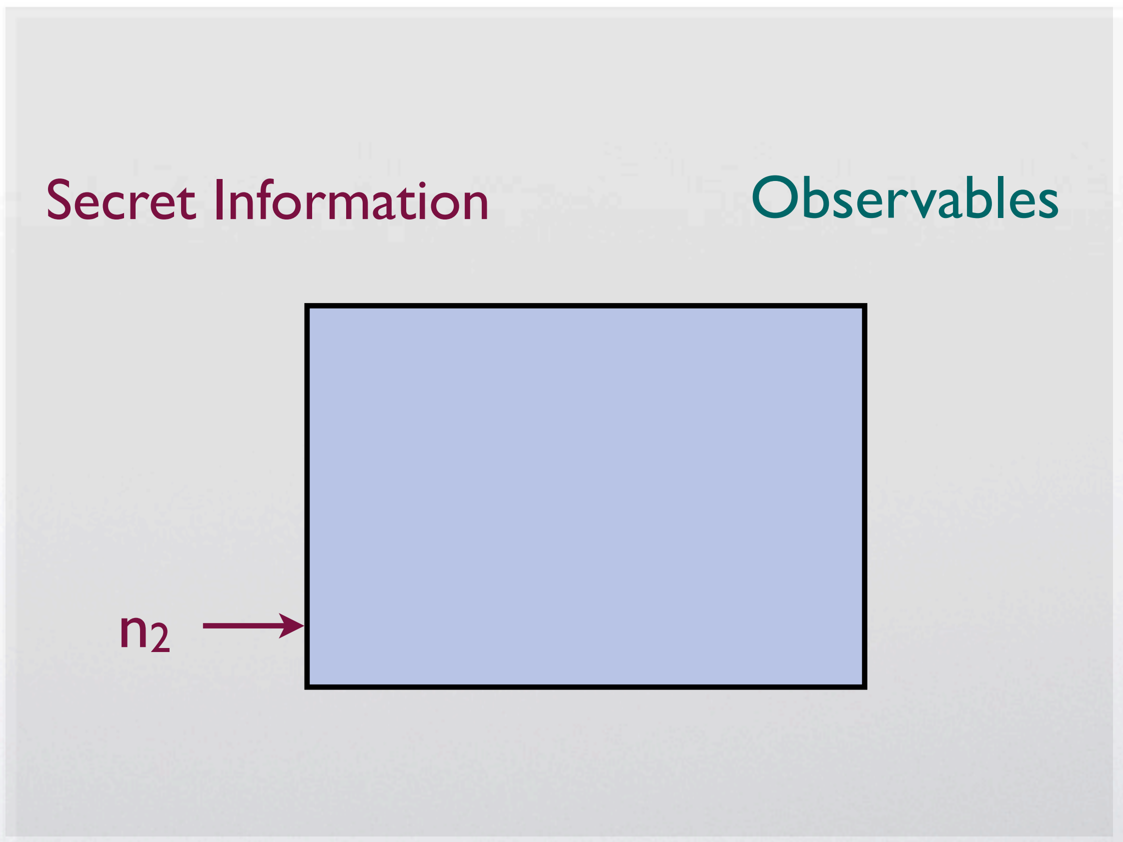
Example: DC nets (ring of 3 nodes, b=1)



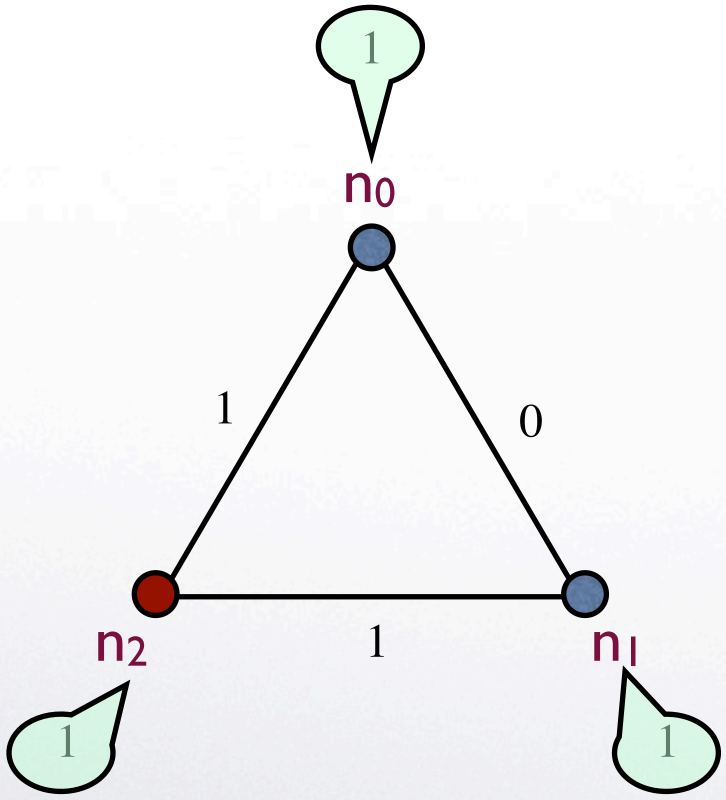


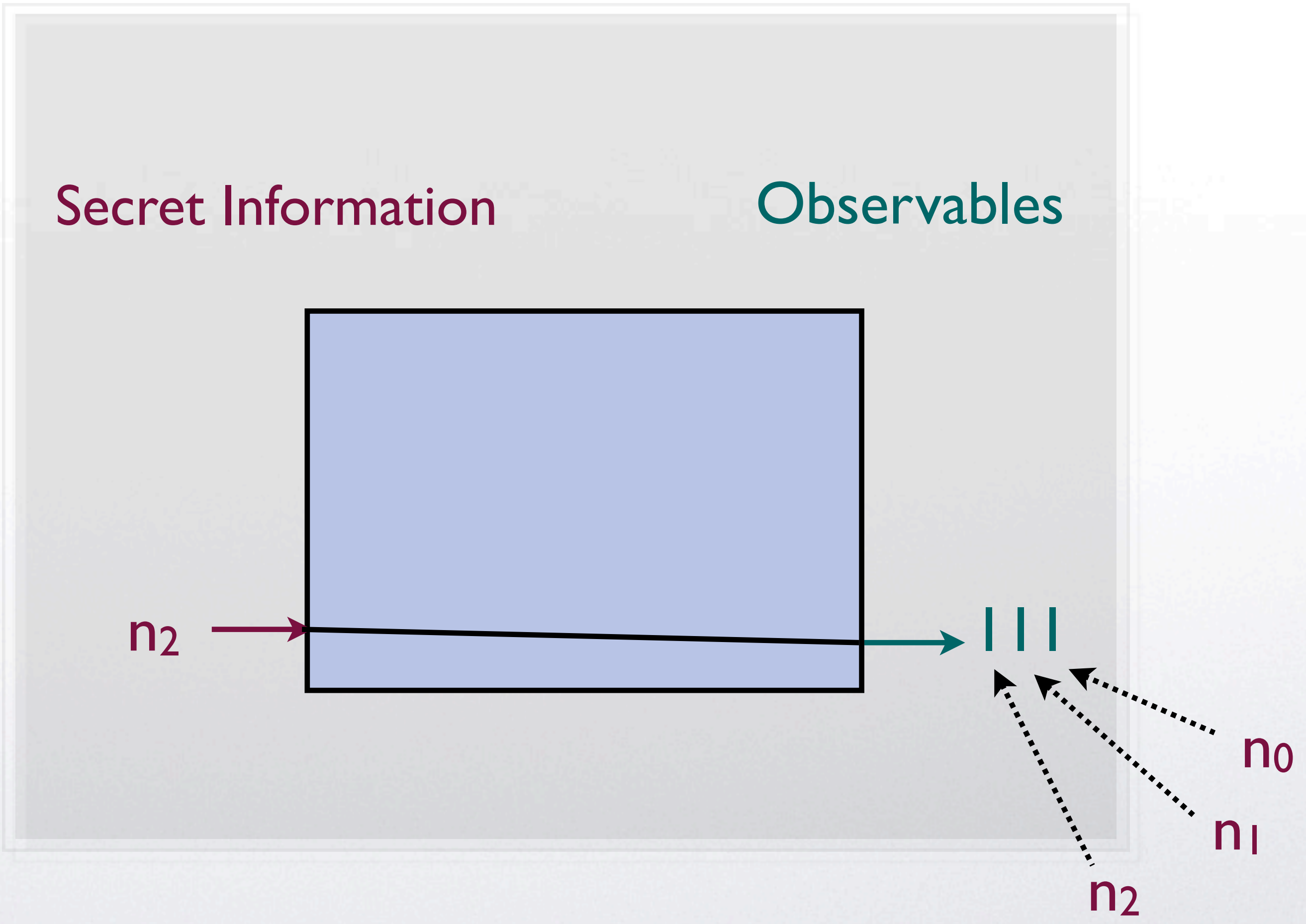
Example: DC nets (ring of 3 nodes, b=1)



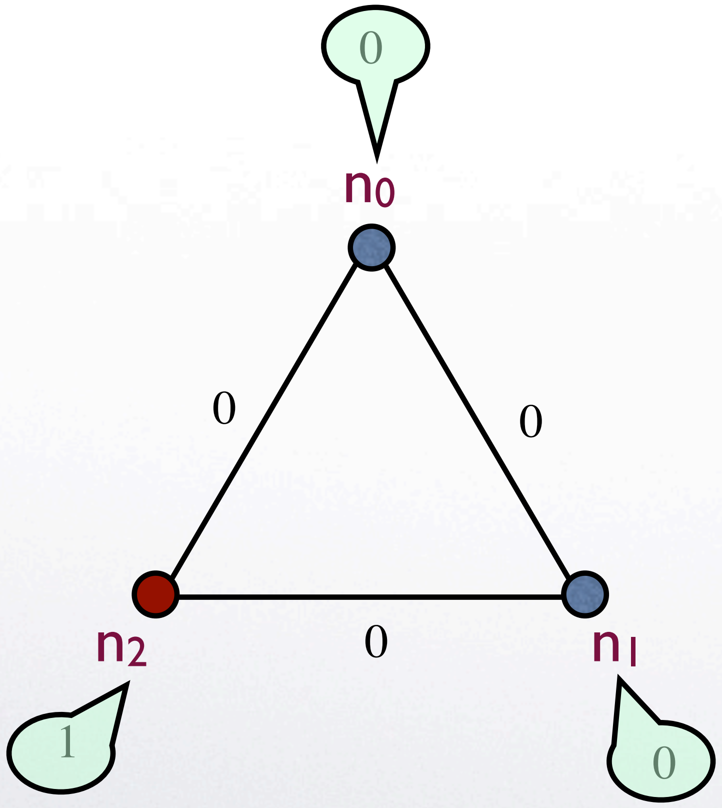


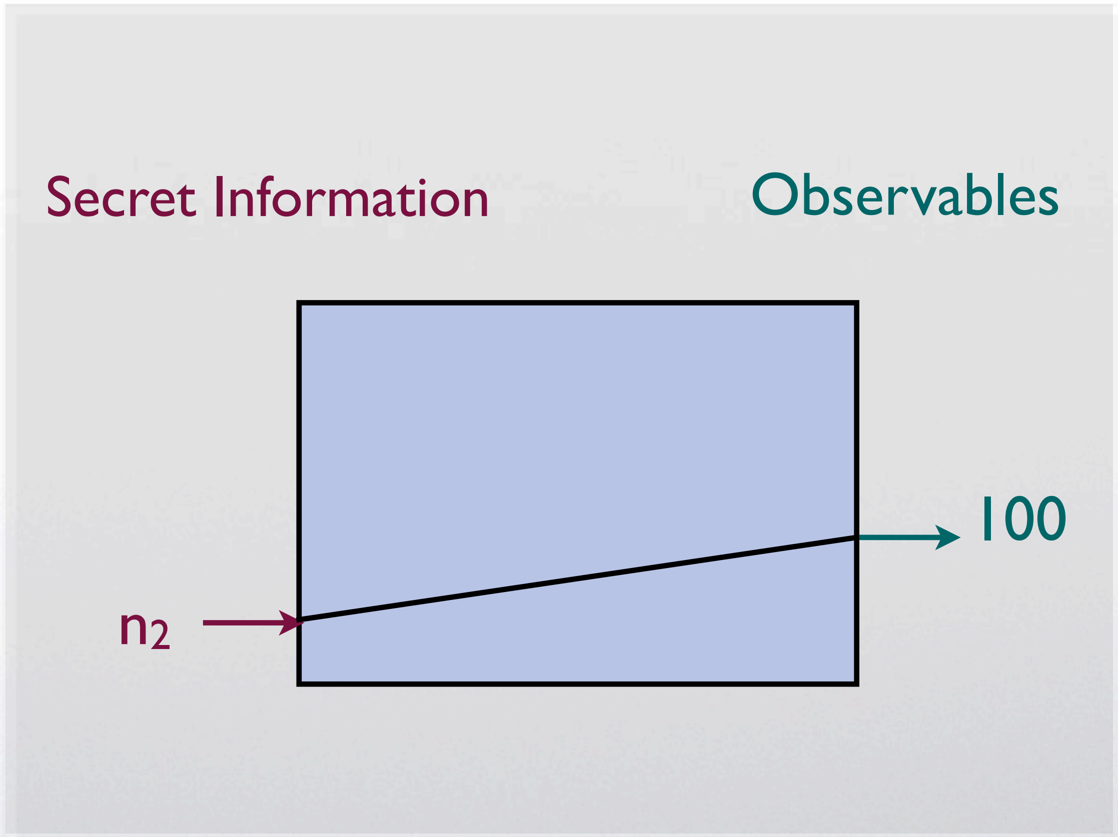
Example: DC nets (ring of 3 nodes, b=1)



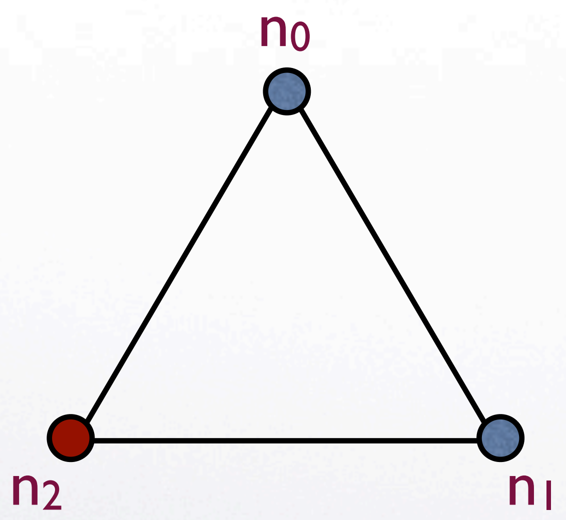


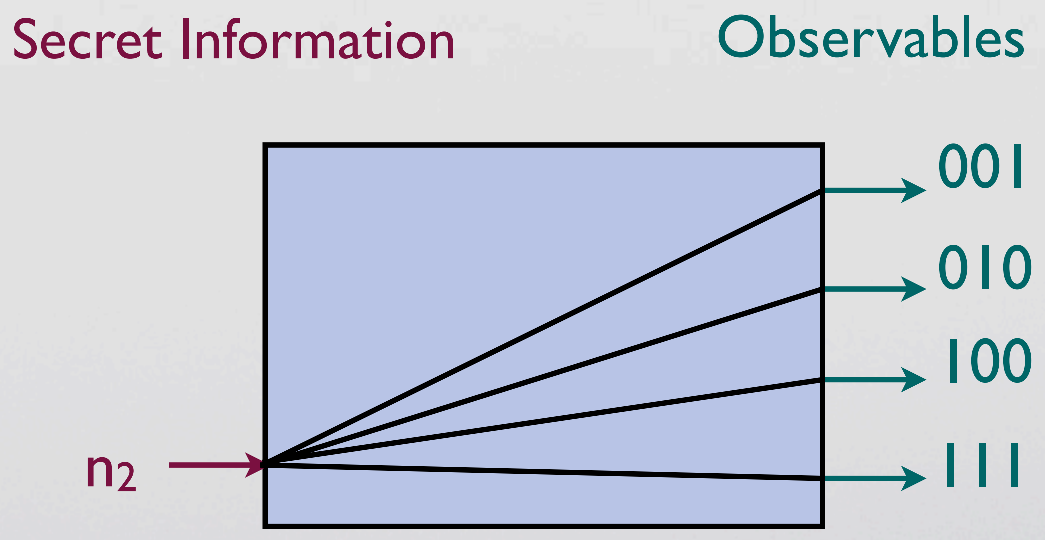
Example: DC nets (ring of 3 nodes, b=1)



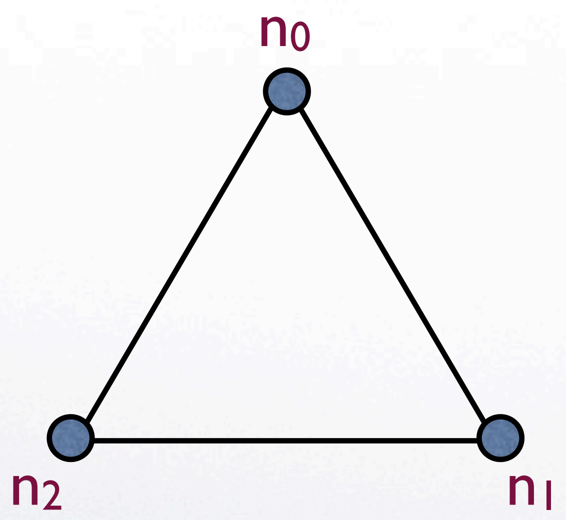


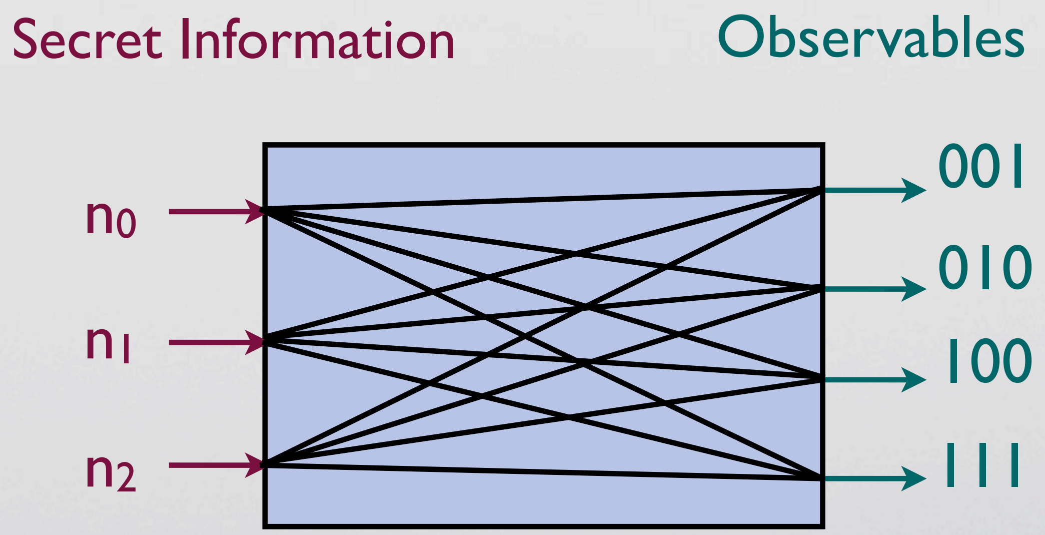
Example: DC nets (ring of 3 nodes, b=1)





Example: DC nets (ring of 3 nodes, b=1)





Example: DC nets (ring of 3 nodes, b=1)

	001	010	100	111
n_0	¼	¼	¼	¼
n_1	¼	¼	¼	¼
n_2	¼	¼	¼	¼

	001	010	100	111
n_0	⅓	$^2/_9$	$^2/_9$	$^2/_9$
n_1	$^2/_9$	⅓	$^2/_9$	$^2/_9$
n_2	$^2/_9$	$^2/_9$	⅓	$^2/_9$

fair coins: Pr(0) = Pr(1) = ½

strong anonymity

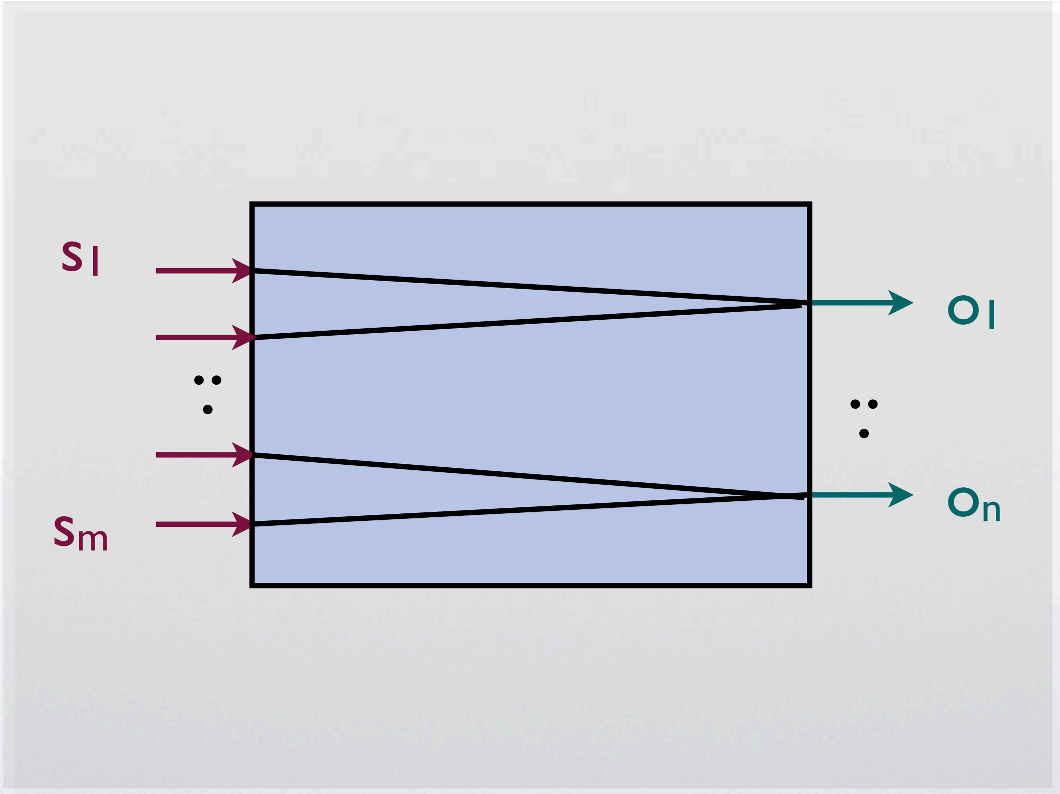
biased coins: Pr(0) = ⅔ , Pr(1) = ⅓

The culprit is more likely to declare 1 than 0

Particular case: **Deterministic systems**

In these systems an input generates only one output

Still interesting: the problem is how to retrieve the input from the output



The conditional probabilities can be only 0 or 1

Exercises

- Compute the channel matrix for the two password-checker programs

- Compute the channel matrix for the DC nets with 3 nodes when the observer is one of the nodes

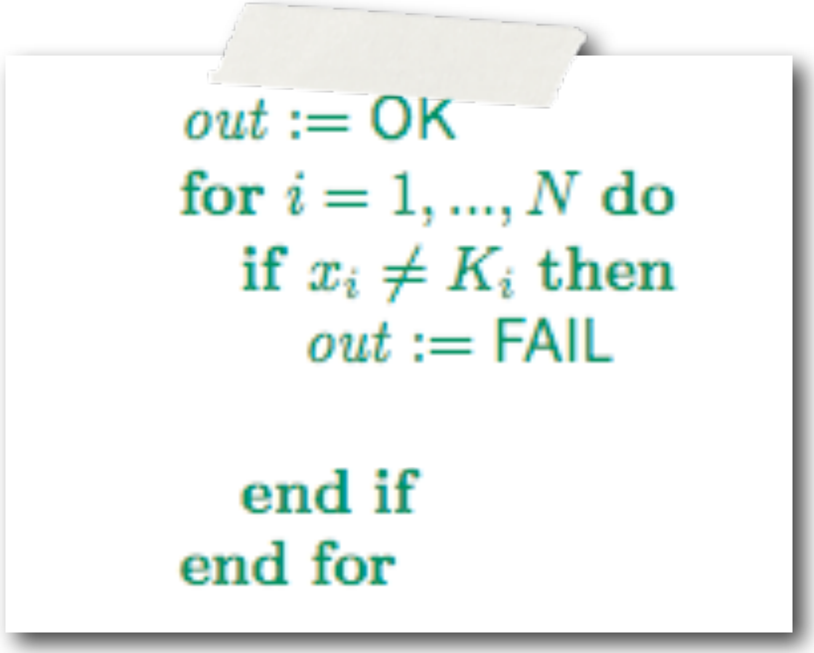
Plan of the lectures

- **Lecture 1 (Monday)**
  - Secure information flow. Motivations and examples
  - Information-theoretic framework
- **Lecture 2 (Tuesday)**
  - Quantification of leakage: models of adversaries
  - Focus on: Shannon entropy and Rényi min-entropy
  - Bayes risk
- **Lecture 3 (Friday)**
  - Differential privacy
  - Relation between QIF and differential privacy

Exercises

- Compute the channel matrix for the two password-checker programs
- Compute the channel matrix for the DC nets with 3 nodes when the observer is one of the nodes

Password-checker 1



```
$out$ := OK
for $i = 1, ..., N$ do
    if $x_i \neq K_i$ then
        $out$ := FAIL

    end if
end for
```

Let us construct the channel matrix

We have two choices:

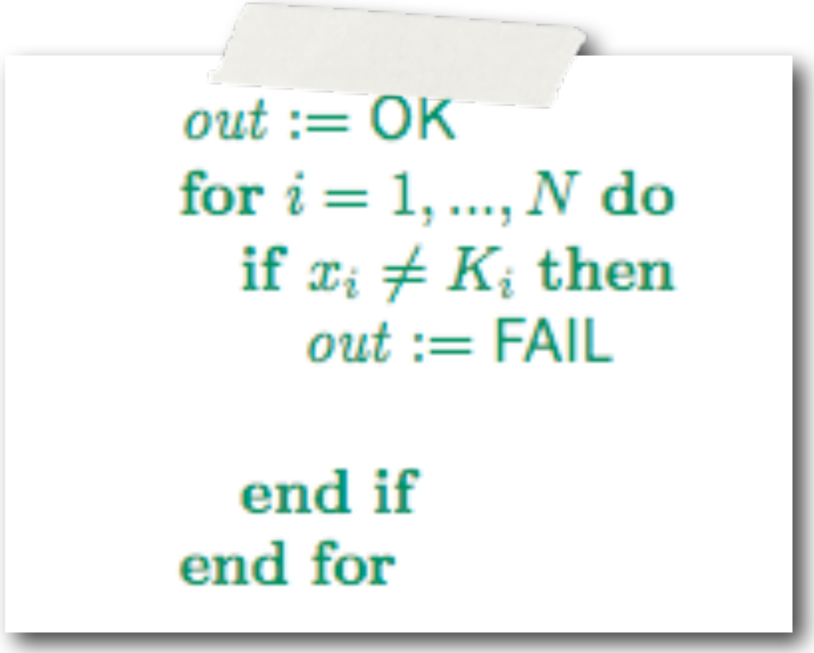
- $x_1x_2x_3$ is a parameter and $K_1K_2K_3$ is the channel input

or

- $K_1K_2K_3$ is a parameter and $x_1x_2x_3$ is the channel input

We will see that they are equivalent. Let us take the first view (to me it seems more natural, but some of my coauthors find more natural the second one)

Password-checker 1



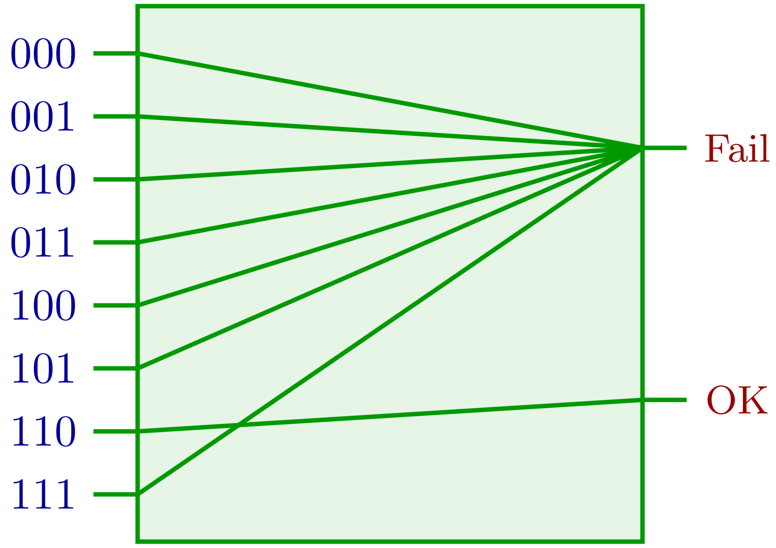
```
$out$ := OK
for $i = 1, ..., N$ do
    if $x_i \neq K_i$ then
        $out$ := FAIL

    end if
end for
```

Assume the user string is $x_1 x_2 x_3 = 110$

Let us construct the channel matrix

Input: $K_1 K_2 K_3 \in \{000, 001, \ldots, 111\}$

Output: $out \in \{\text{OK}, \text{FAIL}\}$



	Fail	OK
000	1	0
001	1	0
010	1	0
011	1	0
100	1	0
101	1	0
110	0	1
111	1	0

Note that if we had considered the dual view in which:

- $K_1 K_2 K_3$ is a parameter
- $x_1 x_2 x_3$ is the channel input

Then the channel matrix would be the same.

Password-checker 2

```
$out$ := OK
for $i = 1, ..., N$ do
    if $x_i \neq K_i$ then
        { $out$ := FAIL
          exit() }
    end if
end for
```

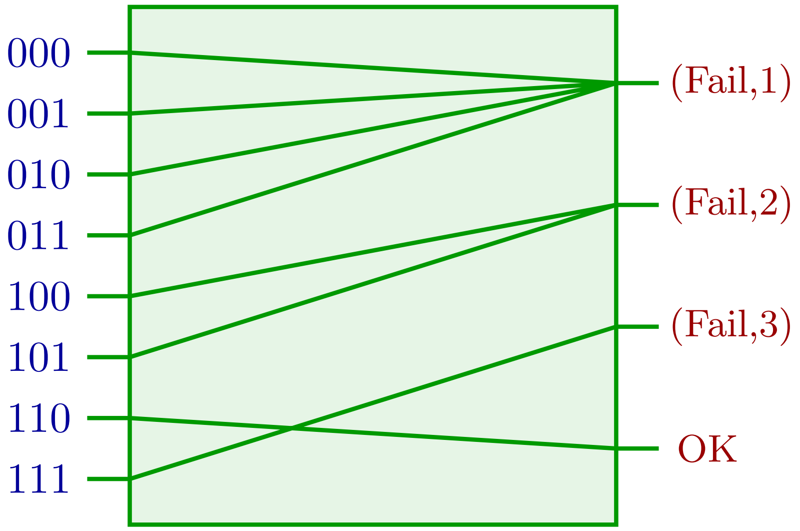
Assume the user string is $x_1 x_2 x_3 = 110$

Assume the adversary can measure the execution time

Let us construct the channel matrix

Input: $K_1 K_2 K_3 \in \{000, 001, \dots, 111\}$

Output: $out \in \{\mathsf{OK}, (\mathsf{FAIL}, 1), (\mathsf{FAIL}, 2), (\mathsf{FAIL}, 3)\}$



	(Fail, 1)	(Fail, 2)	(Fail, 3)	OK
000	1	0	0	0
001	1	0	0	0
010	1	0	0	0
011	1	0	0	0
100	0	1	0	0
101	0	1	0	0
110	0	0	0	1
111	0	0	1	0

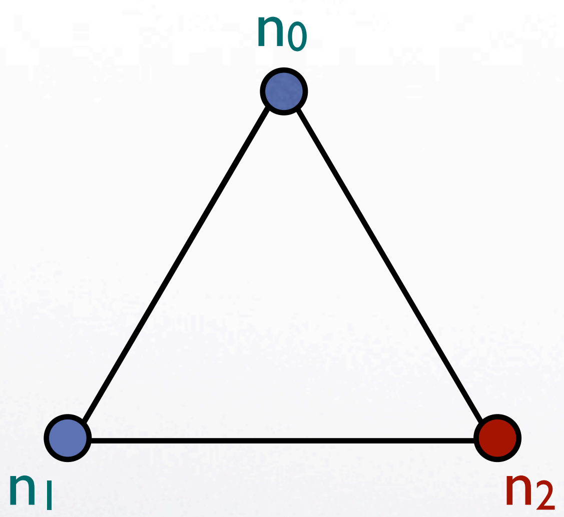
Exercises

- Compute the channel matrix for the two password-checker programs
- Compute the channel matrix for the DC nets with 3 nodes when the observer is one of the nodes.
  By the way, DC stands for "Dining Cryptographers"

DC nets. Ring of 3 nodes, and assume b = 1

n_2 is the observer

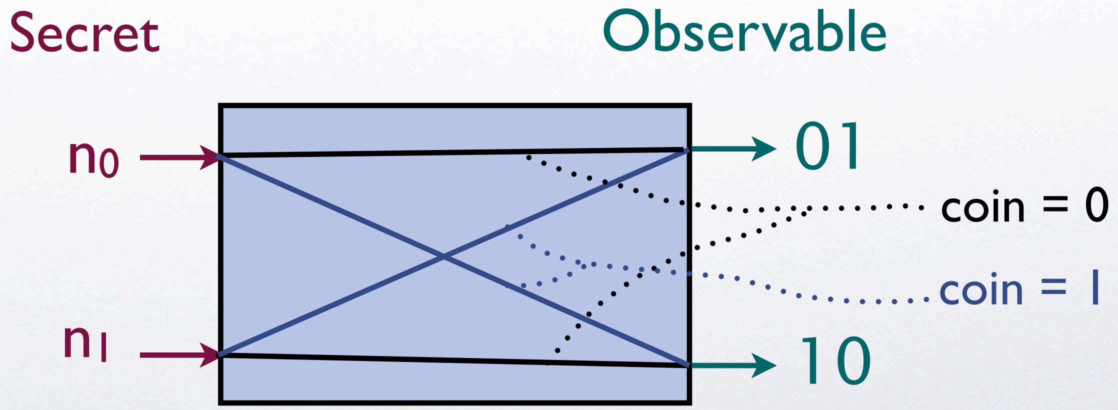
either n_0 or n_1 is the broadcaster



Let us construct the channel matrix

Input: n_0 , n_1

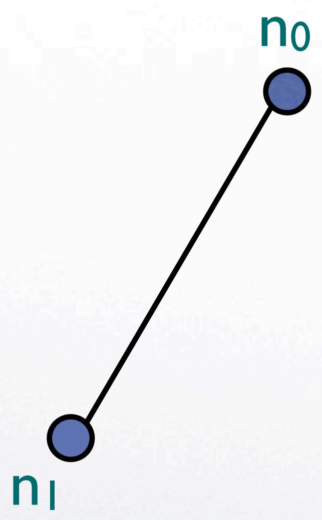
Output: the declarations of n_1 and n_0: $d_1 d_0 \in \{01,10\}$



DC nets. Ring of 3 nodes, and assume b = 1

n_2 is the observer

either n_0 or n_1 is the broadcaster



Let us construct the channel matrix

Input: n_0 , n_1

Output: the declarations of n_1 and n_0: $d_1 d_0 \in \{01, 10\}$

Fair coin: p(0) = p(1) = ½

	01	10
n_0	½	½
n_1	½	½

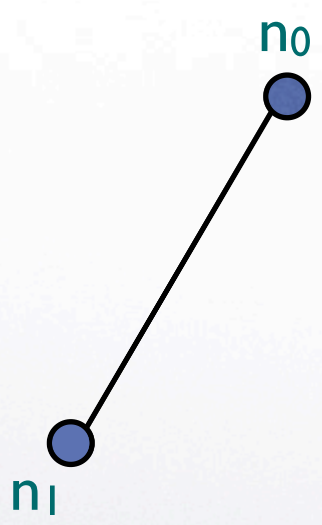
Biased coin: p(0) = ⅔ p(1) = ⅓

	01	10
n_0	⅔	⅓
n_1	⅓	⅔

DC nets. Ring of 3 nodes, and assume b = 1 but now assume b = 0

n_2 is the observer

either n_0 or n_1 is the broadcaster



Let us construct the channel matrix

Input: n_0 , n_1

Output: the declarations of n_1 and n_0: $d_1 d_0 \in \{00,11\}$

Fair coin: $p(0) = p(1) = ½$

	00	11
n_0	½	½
n_1	½	½

Biased coin: $p(0) = ⅔$ $p(1) = ⅓$

	00	11
n_0	⅔	⅓
n_1	⅔	⅓

Towards a quantitative notion of leakage

A general principle:

Leakage = difference between
the a priori vulnerability
and
the a posteriori vulnerability

- vulnerability = vulnerability of the secret,
- a priori / a posteriori = before / after the observation

Intuitively the vulnerability depends on the distribution: the more uncertainty there is about the exact value of the secret, the less vulnerable the secret is.

Note that the observation updates the input probability:

$$p(s|o) = p(s)\frac{p(o|s)}{p(o)}$$ Bayes theorem

Vulnerability

There is no unique notion of vulnerability. It depends on:

- the model of attack, and
- how we measure its success

A general **model of attack** [Köpf and Basin'07]:

- Assume an oracle that answers yes/no to questions of a certain form.
- The attack is defined by the form of the questions.
- In general we consider the best strategy for the attacker, with respect to a given measure of success.

Vulnerability

Case 1:

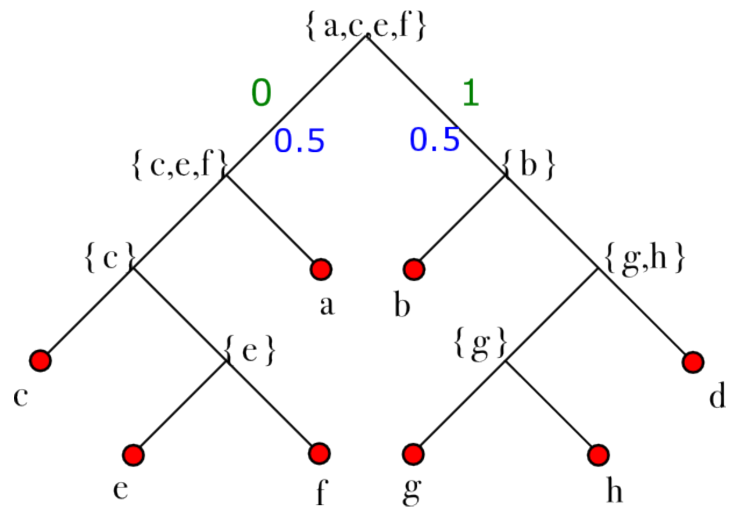
- The questions are of the form: “is S ∈ P ?”
- The measure of success is: the expected number of questions needed to find the value of S in the attacker’s best strategy

Typical case : guessing a password bit by bit

Example: S ∈ { *a,b,c,d,e,f,h* }

$$p(a) = p(b) = \frac{1}{4} \qquad p(c) = p(d) = \frac{1}{8} \qquad p(e) = p(f) = p(g) = p(h) = \frac{1}{16}$$

It is possible to prove that the best strategy for the adversary is to consider each time a search space P s.t. P and the complement of P have prob. masses as close as possible



Vulnerability

Case 1:

- The questions are of the form: "is S ∈ P ?"
- The measure of success is: the expected number of questions needed to find the value of S in the attacker's best strategy

In the best strategy, the number of questions needed to determine the value of the secret S, when S = s, is: **− log *p*(*s*)** (log is in base 2)

hence the **expected number** of question is:

$$H(S) = -\sum_{s} p(s) \log p(s)$$

This is exactly the formula for **Shannon's entropy**

Information-theoretic interpretation:

H(S) is the expected length of the optimal encoding of the values of S

For the strategy in previous example: a: 01 b: 10 c: 000 d: 111 e: 0010 f: 0011 g: 1100 h: 1101

Exercise: find a different optimal encoding for the distribution in previous example

Shannon entropy

In general, the entropy is highest when the distribution is uniform

If $|S| = n$, and the distribution is uniform, then $H(S) = \log n$

$S = \{a, b, c, d, e, f, g, h\} \qquad p(a) = p(b) = \ldots = p(f) = \frac{1}{8}$

$$H(S) \quad = \quad -8\tfrac{1}{8}\log\tfrac{1}{8} \quad = \quad \log 8 \quad = \quad 3$$

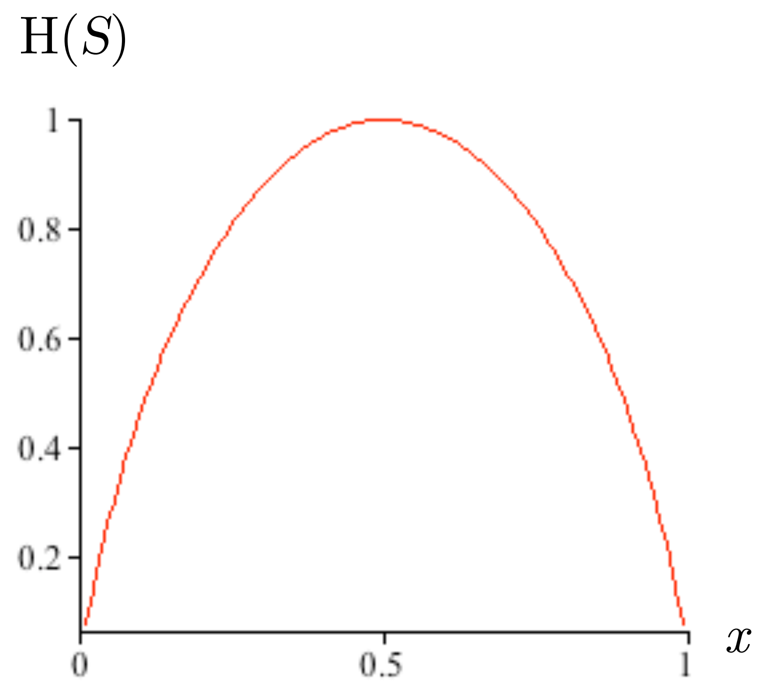
$p(a) = p(b) = \frac{1}{4} \qquad p(c) = p(d) = \frac{1}{8} \qquad p(e) = p(f) = p(g) = p(h) = \frac{1}{16}$

$$\begin{aligned} H(S) &= -\textstyle\sum_s p(s)\log p(s) \\ &= -2\tfrac{1}{4}\log\tfrac{1}{4} - 2\tfrac{1}{8}\log\tfrac{1}{8} - 4\tfrac{1}{16}\log\tfrac{1}{16} \\ &= 1 + \tfrac{3}{4} + 1 \\ &= \tfrac{11}{4} \end{aligned}$$

Shannon entropy

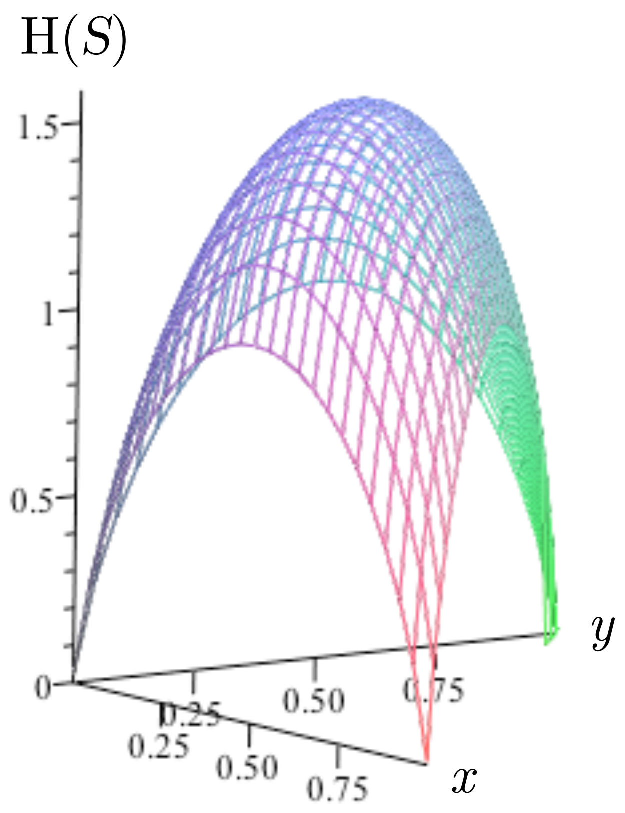
$S = \{a, b\}$

$p(a) = x \quad p(b) = 1 - x$



$S = \{a, b, c\}$

$p(a) = x \quad p(b) = y \quad p(c) = 1 - (x + y)$



Shannon conditional entropy

An observable o determines a new distribution on S:

$$p(s|o) = p(s)\frac{p(o|s)}{p(o)} \qquad \text{Bayes theorem}$$

The entropy of the new distribution on S, given that $O = o$, is:

$$H(S|O=o) \quad = \quad -\sum_s p(s|o)\log p(s|o)$$

The conditional entropy is the expected value of the updated entropies:

$$\begin{aligned} H(S|O) \quad &= \quad \sum_o p(o)\; H(S|O=o) \\ &= \quad -\sum_o p(o) \sum_s p(s|o)\log p(s|o) \end{aligned}$$

Shannon entropy

A priori $$H(S) = -\sum_s p(s) \log p(s)$$

A posteriori $$H(S \mid O) = -\sum_o p(o) \sum_s p(s|o) \log p(s|o)$$

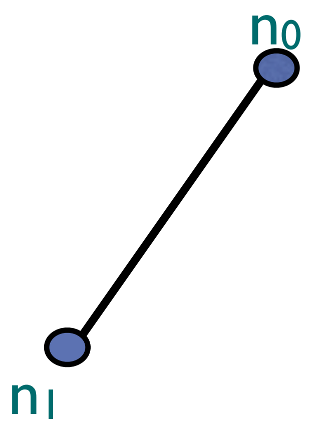
Leakage = Mutual Information $$I(S;O) = H(S) - H(S|O)$$

- In general H(S) ≥ H(S|O)
  - the vulnerability may decrease after one single observation, but in the average the vulnerability increases, or remains the same

- H(S) = H(S|O) if and only if S and O are independent
  - This is the case if and only if all rows of the channel matrix are the same
  - This case corresponds to strong anonymity in the sense of Chaum

- Shannon capacity = worst-case leakage C = max I(S;O) over all inp. dist.

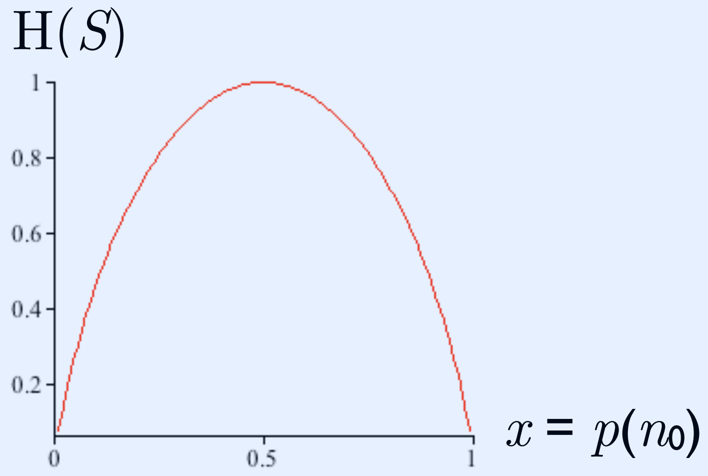
Example: DC nets. Ring of 2 nodes, b = 1, fair coin



Input S: n_0 , n_1

Output O: the declarations of n_1 and n_0: $d_1 d_0 \in \{01, 10\}$

The entropy of S, as a function of the distribution on S



Fair coin: p(0) = p(1) = ½

	01	10
n_0	½	½
n_1	½	½

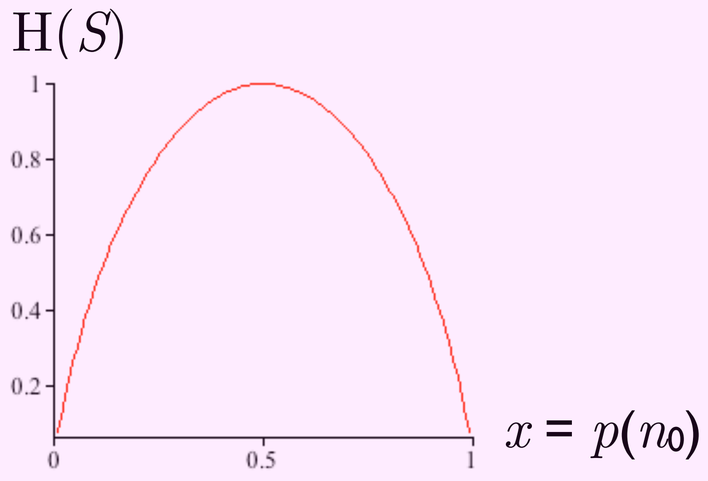
The updated distribution after observation 01

$$\begin{aligned} p(n_0|01) &= \frac{p(01|n_0)}{p(01)}\, p(n_0) \\ &= \frac{p(01|n_0)}{p(01|n_0)p(n_0)+p(01|n_1)p(n_1)}\, p(n_0) \\ &= \frac{\frac{1}{2}}{\frac{1}{2}p(n_0)+\frac{1}{2}p(n_1)}\, p(n_0) \\ &= p(n_0) \end{aligned}$$

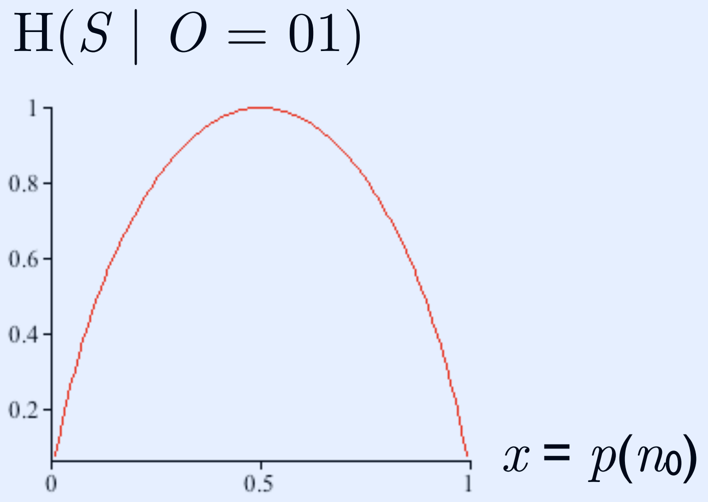
Similarly, after observation 10

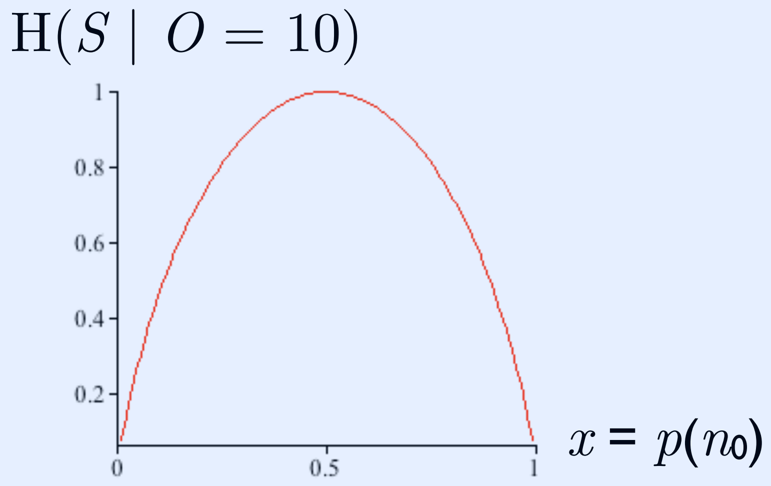
$$p(n_0|10) = p(n_0)$$

The entropy of S, as a function of the distribution on S



The entropies of S given $O = 01$, and given $O = 10$, as functions of the distribution on S





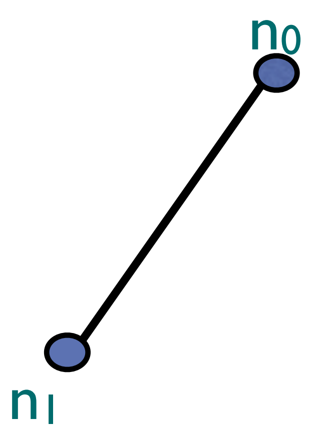
$$H(S|O = 01) \;=\; H(S|O = 10) = H(S)$$

$$\text{Hence } H(S|O) \;=\; p(01)\,H(S|O = 01) + p(10)\,H(S|O = 10) = H(S)$$

Example: DC nets. Ring of 2 nodes, b = 1, biased coin

Input S: n_0 , n_1

Output O: the declarations of n_1 and n_0: $d_1 d_0 \in \{01, 10\}$



Biased c.: p(0) = ⅔ p(1) = ⅓

	01	10
n_0	⅔	⅓
n_1	⅓	⅔

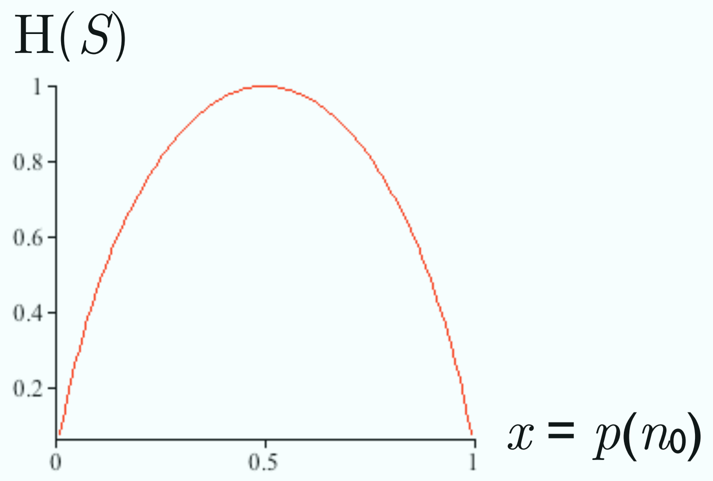
The updated distribution after observation 01

$$\begin{aligned} p(n_0|01) &= \frac{p(01|n_0)}{p(01)}\, p(n_0) \\ &= \frac{p(01|n_0)}{p(01|n_0)p(n_0)+p(01|n_1)p(n_1)}\, p(n_0) \\ &= \frac{\frac{2}{3}}{\frac{2}{3}p(n_0)+\frac{1}{3}p(n_1)}\, p(n_0) \\ &= \frac{\frac{2}{3}}{\frac{2}{3}p(n_0)+\frac{1}{3}(1-p(n_0))}\, p(n_0) \\ &= \frac{2\, p(n_0)}{p(n_0)+1} \end{aligned}$$

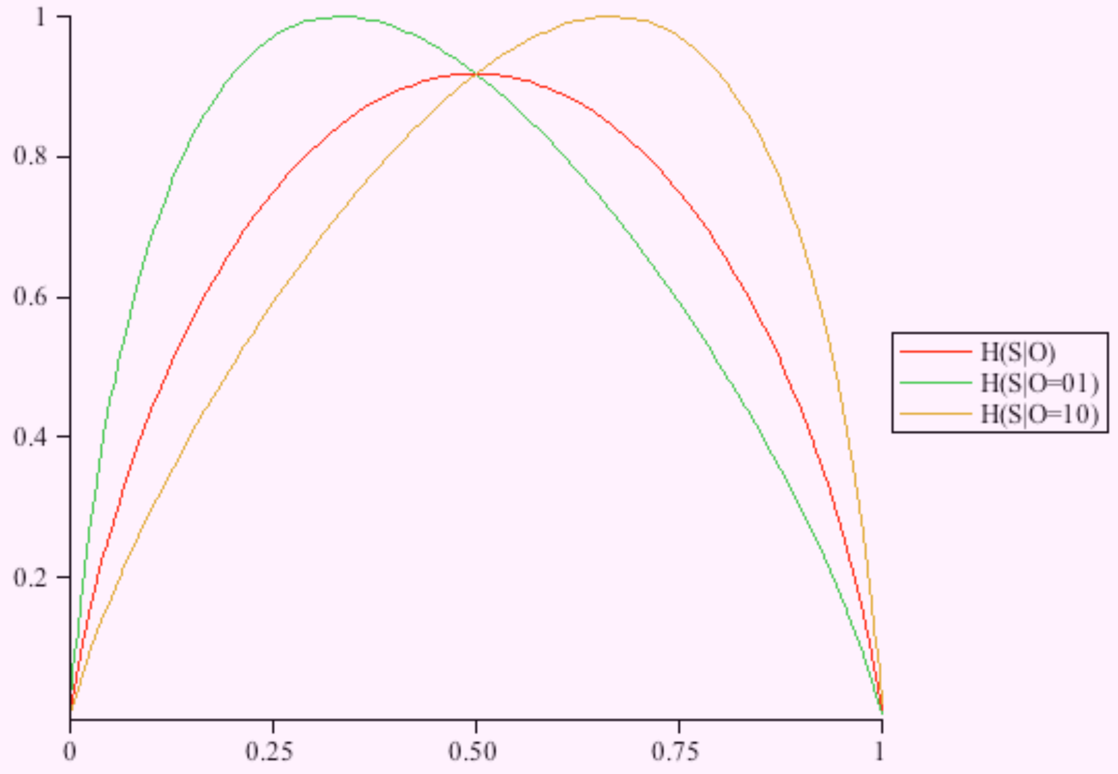
The updated distribution after observation 10

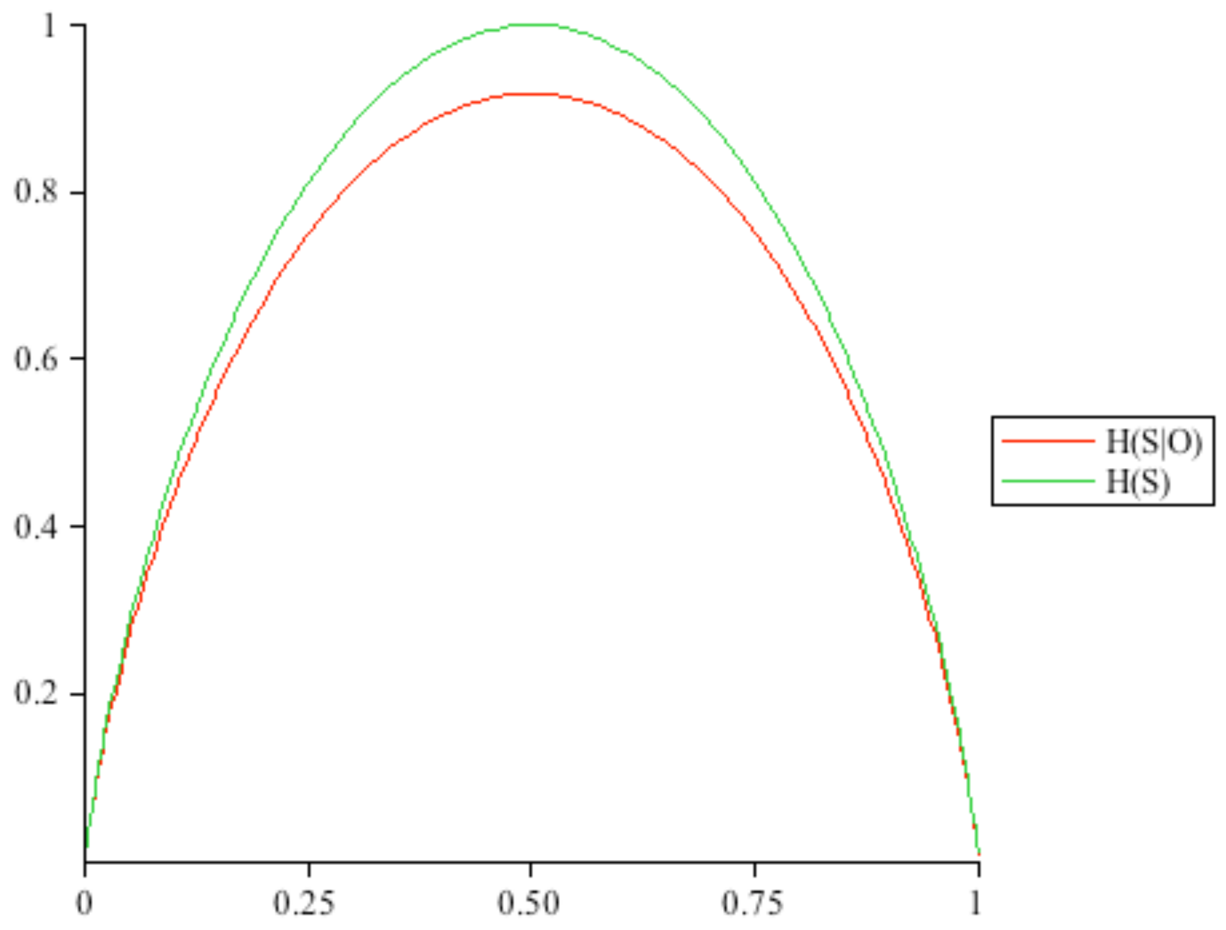
$$p(n_0|10) = \frac{p(n_0)}{2-p(n_0)}$$

The entropy of S,
as a function of the
distribution on S



The conditional
entropy of S given O
as a function of the
distribution on S





$$I(S|O) = H(S) - H(S|O)$$
$$\text{Capacity} = \max I(S|O)$$

In this example the capacity is about 0.1 bits, and it is obtained when the input distribution is uniform

Exercise: compute the capacity for b=0 in the case of a biased coin

Biased coin: p(0) = ⅔ p(1) = ⅓

	00	11
n_0	⅔	⅓
n_1	⅔	⅓