Foundations of Privacy
Lecture 4

Catuscia Palamidessi

Part I
Quantitative Information Flow

1. Motivations

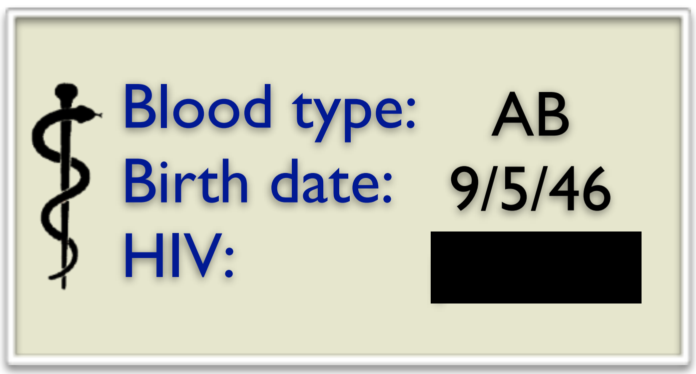
2. Information-theoretic view

3. Notions of entropy and operational interpretation

4. Focus on Shannon leakage and min-entropy leakage

Protection of sensitive information

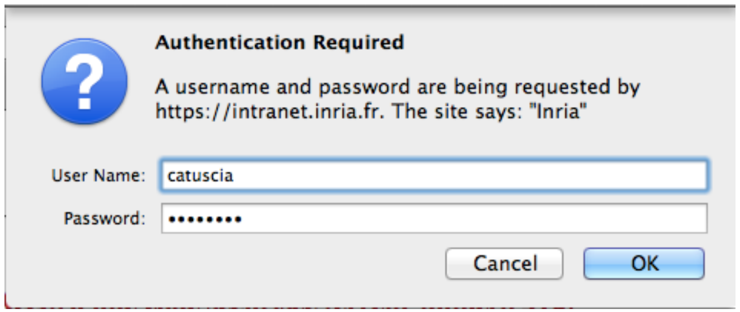
- Protecting the confidentiality of sensitive information is a fundamental issue in computer security and in privacy

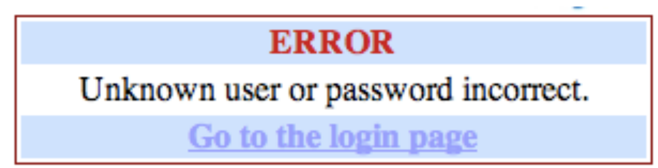


- Access control and encryption are not sufficient! Systems could leak secret information through correlated observables.
  - The notion of "observable" depends on the situation and adversary
  - Often, secret-leaking observables are public, and therefore available to the adversary

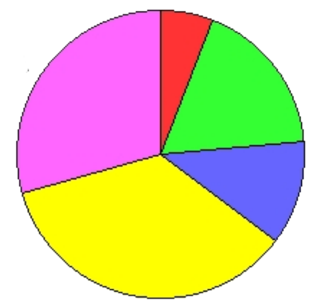
Leakage through correlated observables

Password checking

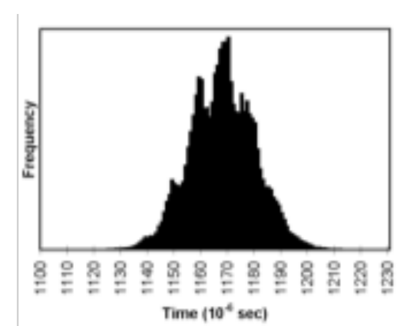




Election tabulation



Timings of decryptions



Quantitative Information Flow

Information Flow: Leakage of secret information via correlated observables

Ideally: No leak

- No interference [Goguen & Meseguer'82]

In practice: There is almost always some leak

- Intrinsic to the system (public observables, part of the design)
- Side channels

➪ **need quantitative ways to measure the leak**

Example 1

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)

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

   end if
end for
```

Intrinsic leakage

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

Example 1

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)

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

More efficient, but what about security?

Example 1

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)

```
out := OK
for i = 1, ..., N do
  if x_i ≠ K_i then
    out := FAIL
    exit()
  end if
end for
```

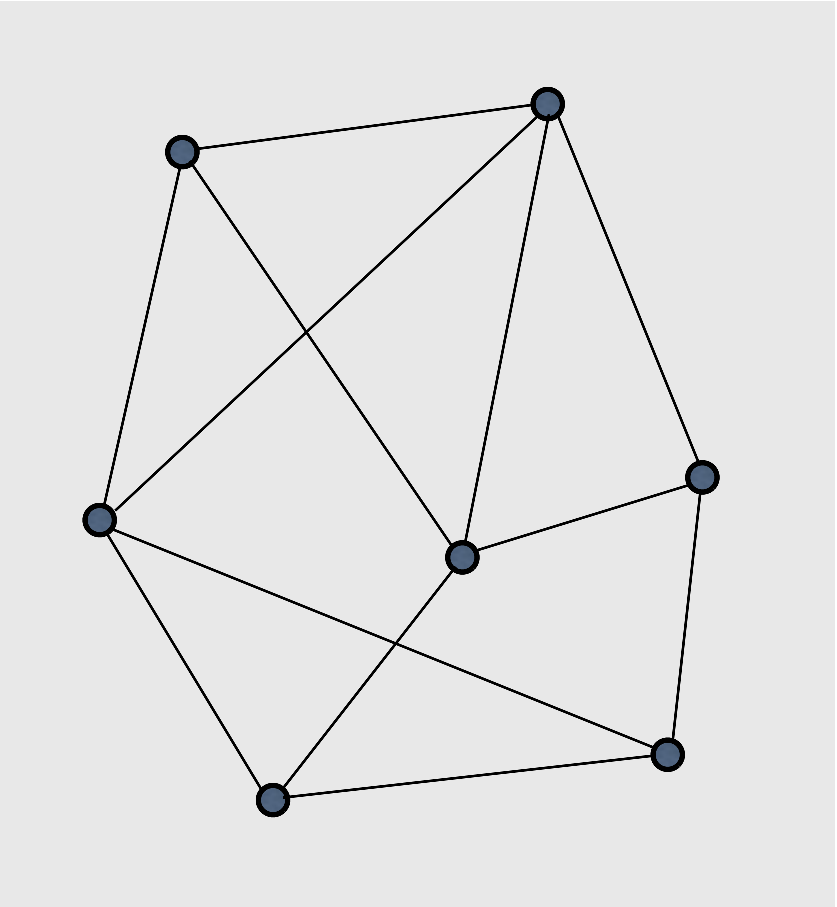
Side channel attack

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

Example 2

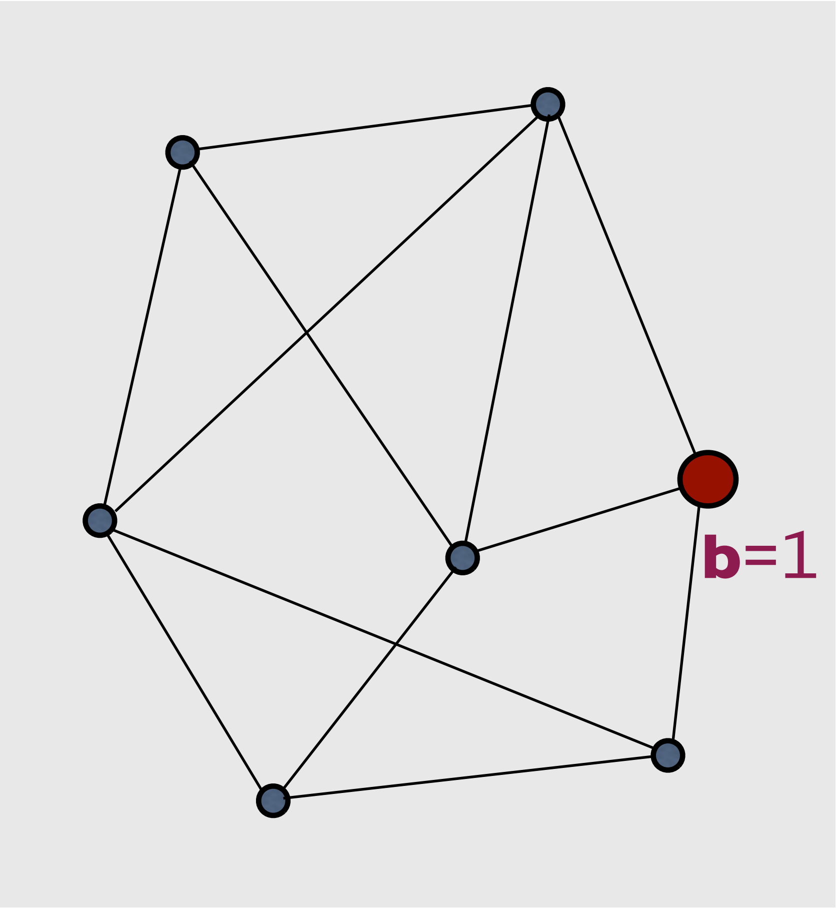
DC Nets
(Extended Dining Cryptographers)
[Chaum'88]

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



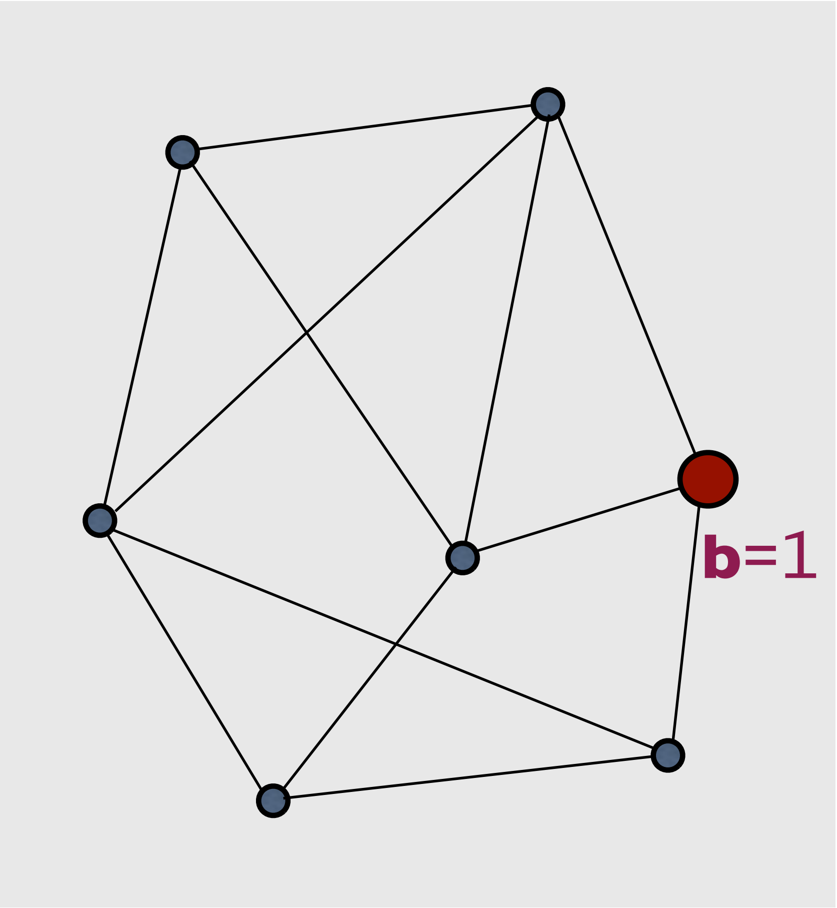
DC Nets
(Extended Dining Cryptographers)
[Chaum'88]

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



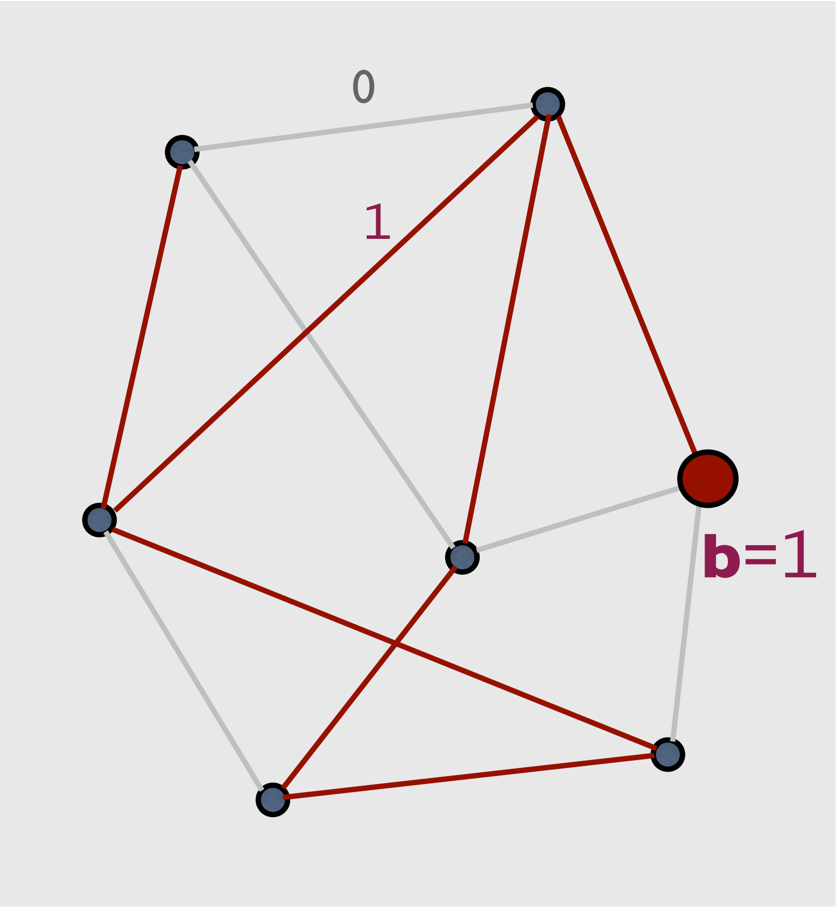
Chaum's solution

- Associate to each edge a fair binary coin



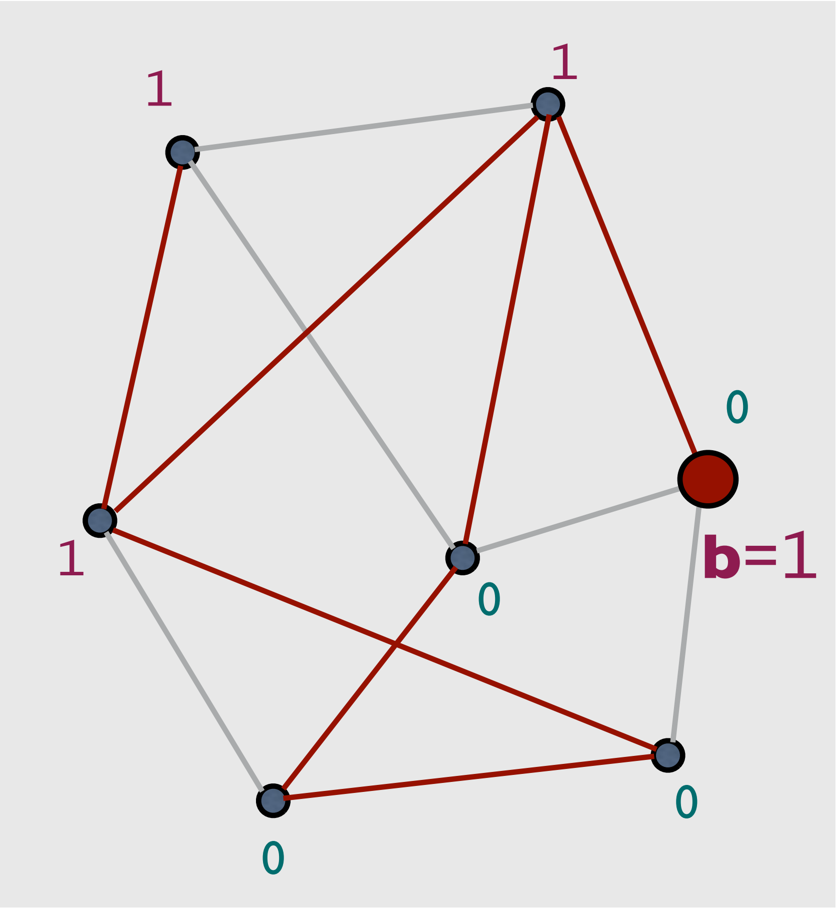
Chaum's solution

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



Chaum's solution

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



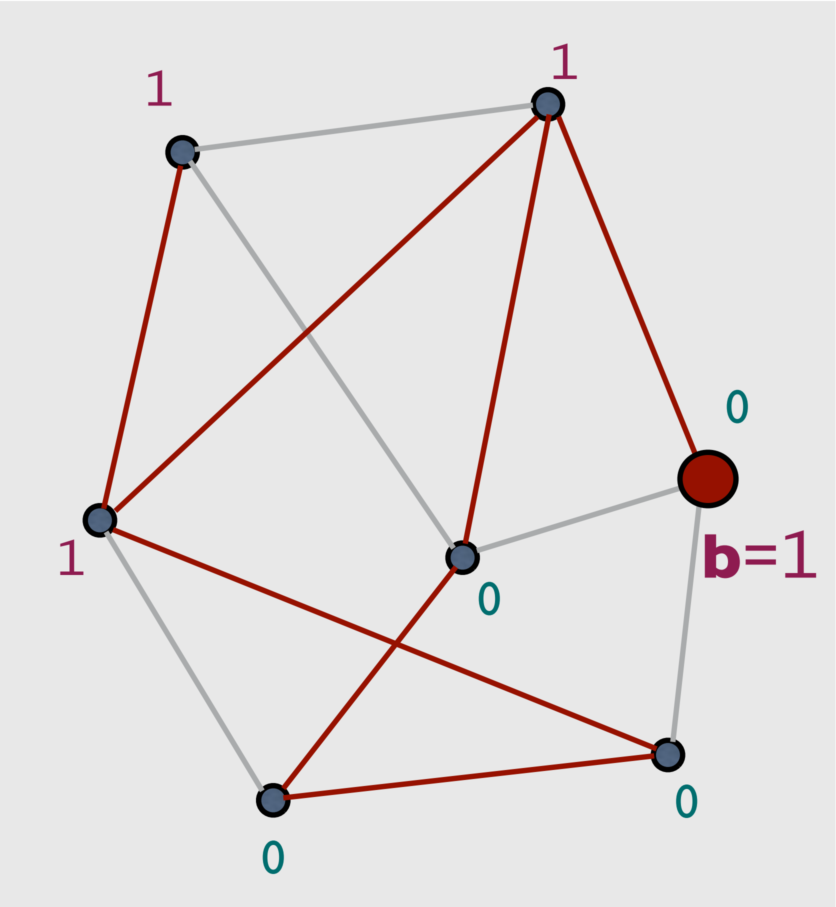
Chaum's solution

- Associate to each edge a fair binary 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**



Anonymity of DC Nets

Observables: An (external) attacker can only see the declarations of the nodes

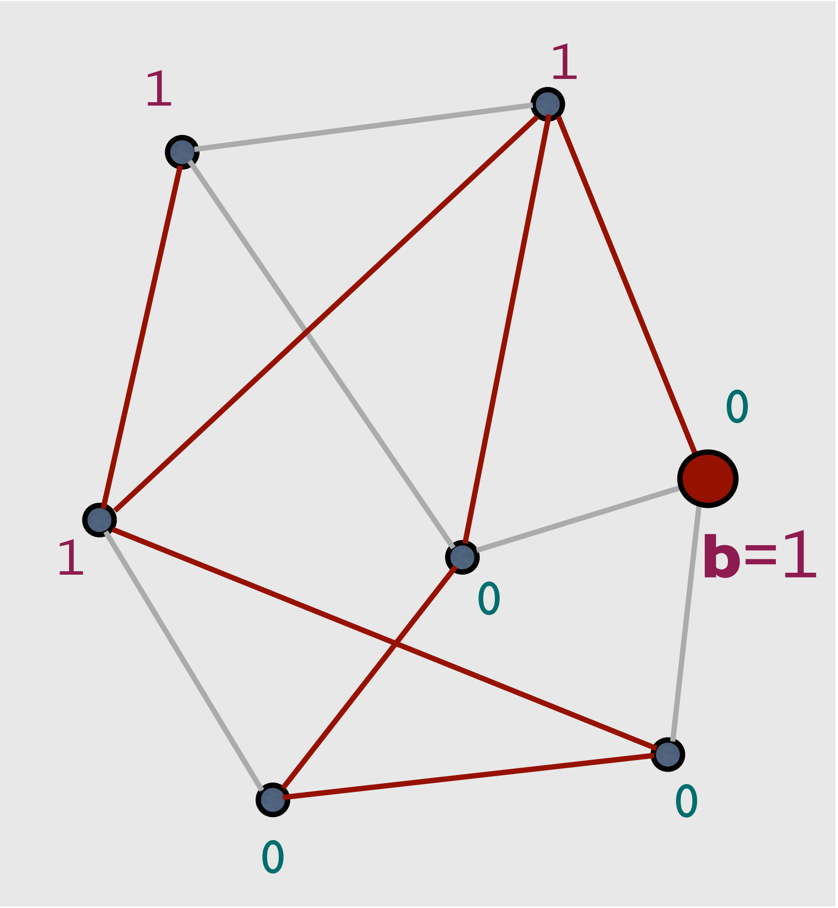
Question: Does the protocol protects the anonymity of the source?

Strong anonymity (Chaum)

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

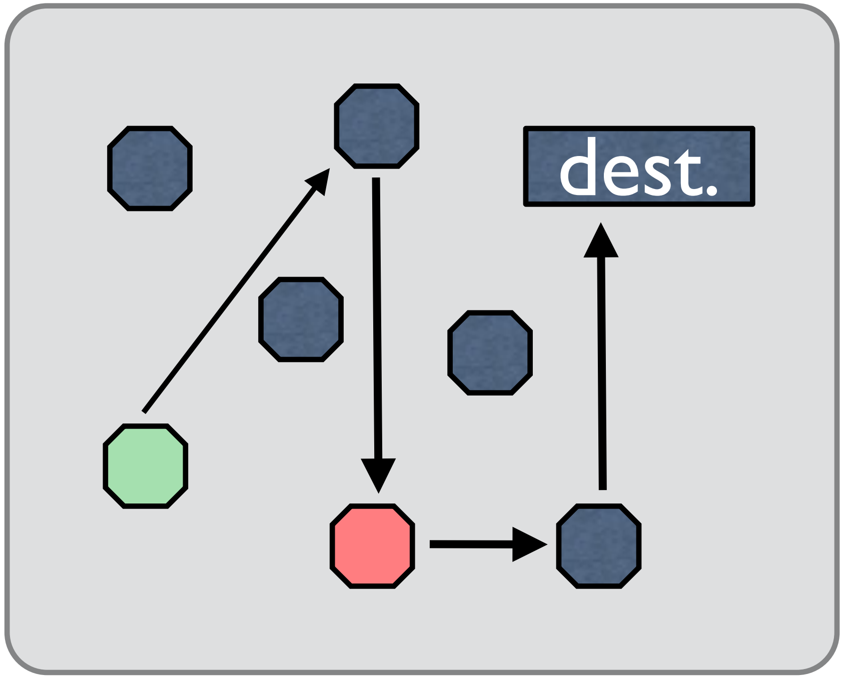
> the *a posteriori* probability that a certain node is the source is equal to its *a priori* probability

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



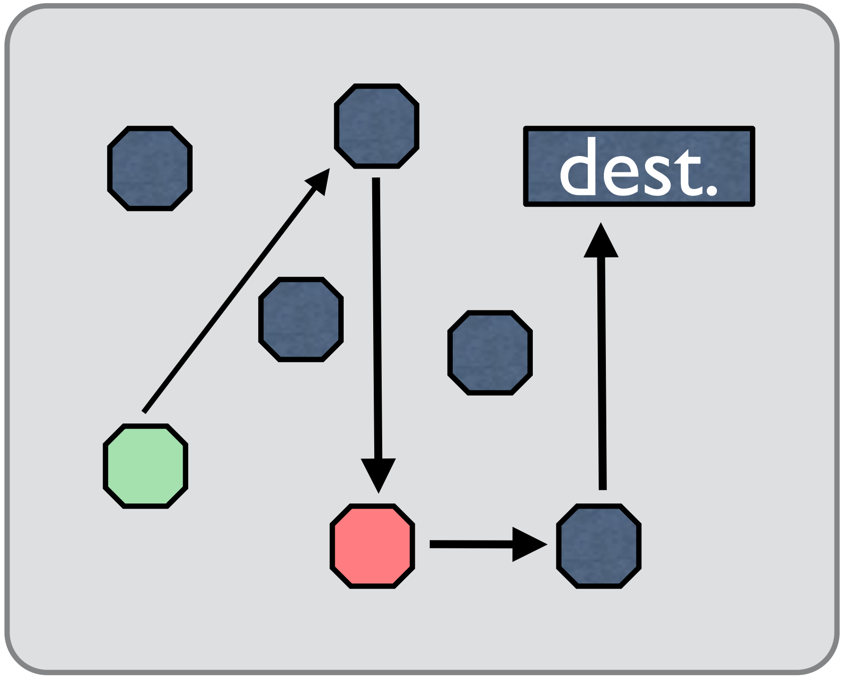
Example 3: Crowds [Rubin and Reiter'98]

- 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



Example 3: 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



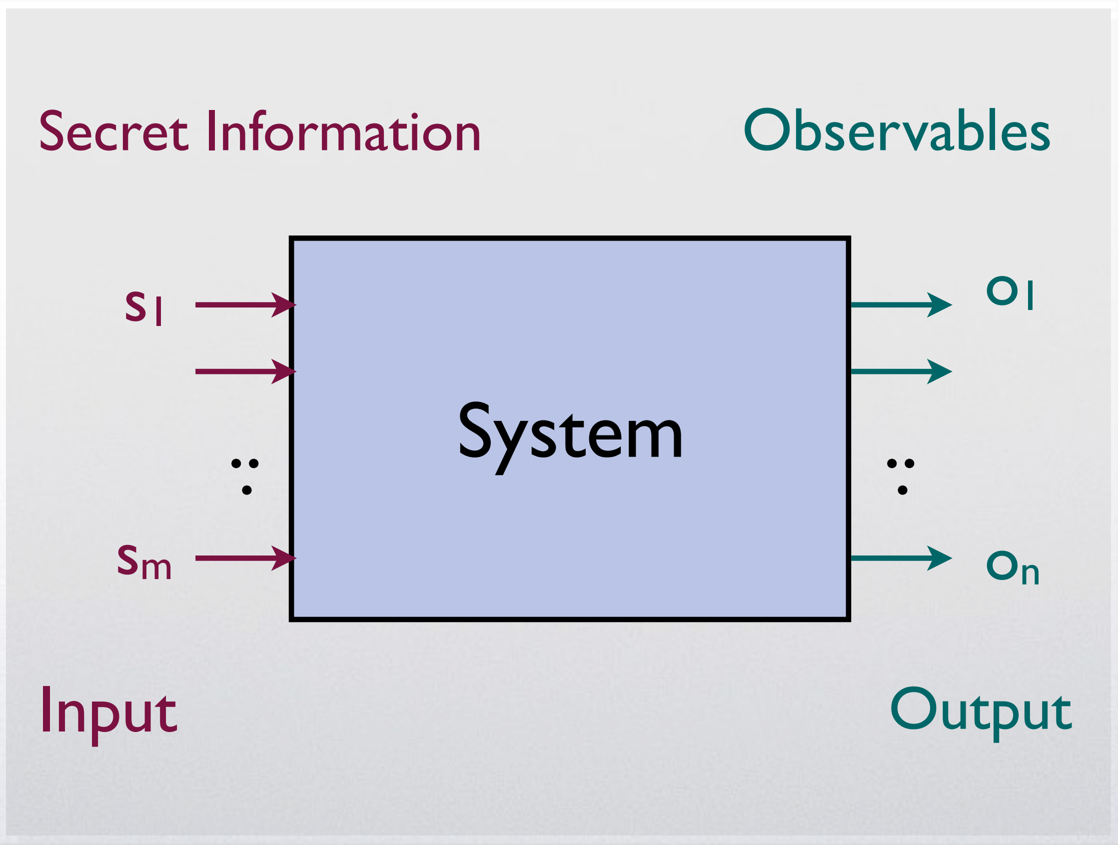
Probable innocence: under certain conditions, an attacker who intercepts the message from x cannot attribute more than 0.5 probability to x to be the initiator

Common features

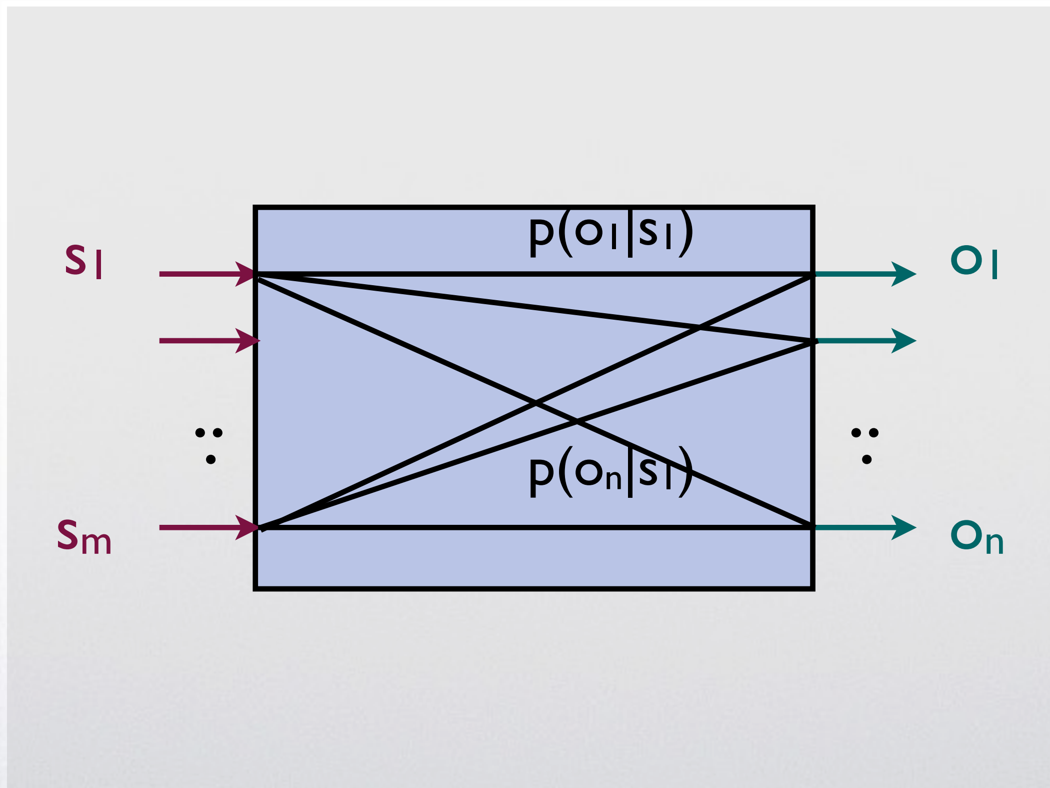
- Secret information
  - Password checker: The password
  - DC: the identity of the source
  - Crowds: the identity of the initiator
- Public information (Observables)
  - Password checker: The result (OK / Fail) and the execution time
  - DC: the declarations of the nodes
  - Crowds: the identity of the agent forwarding to 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

The basic model:

Systems = 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

	o_1	$\ldots$	o_n
s_1	$p(o_1 \mid s_1)$	$\cdots$	$p(o_n \mid s_1)$
$\vdots$	$\vdots$		
s_m	$p(o_1 \mid s_m)$		$p(o_n \mid s_m)$

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

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

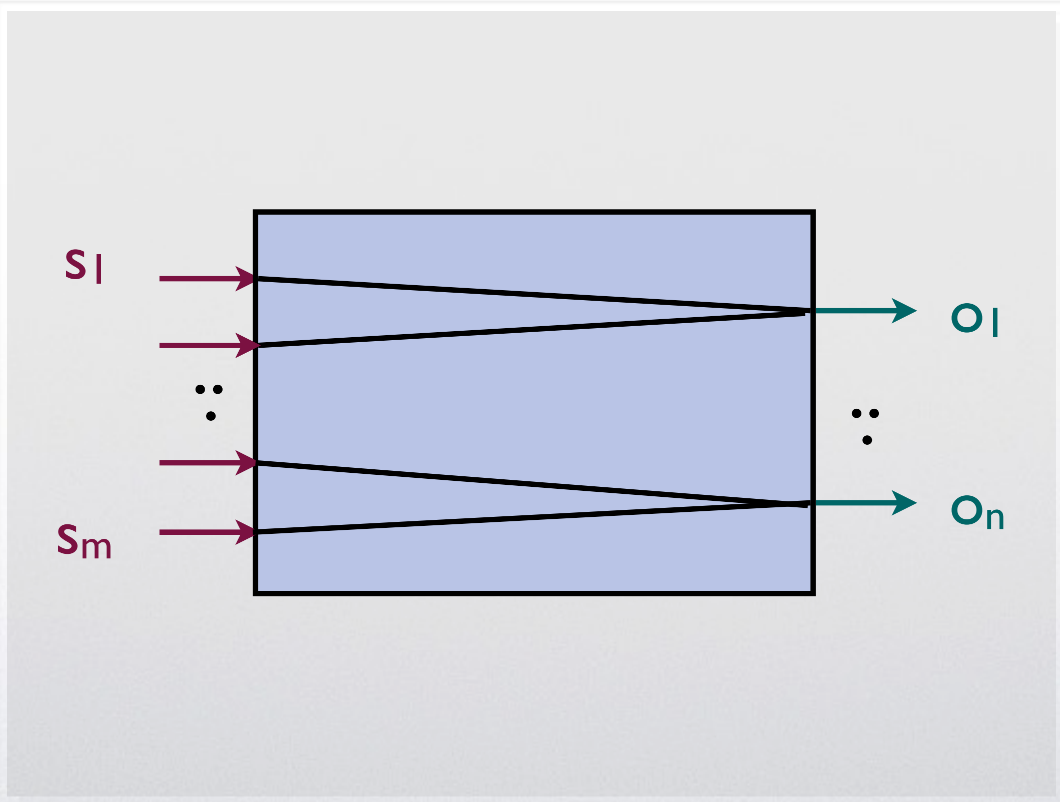
In a information-theoretic channel these conditional probabilities are independent from the input distribution

This means that we can model systems abstracting from the input distribution

Particular case: **Deterministic systems**

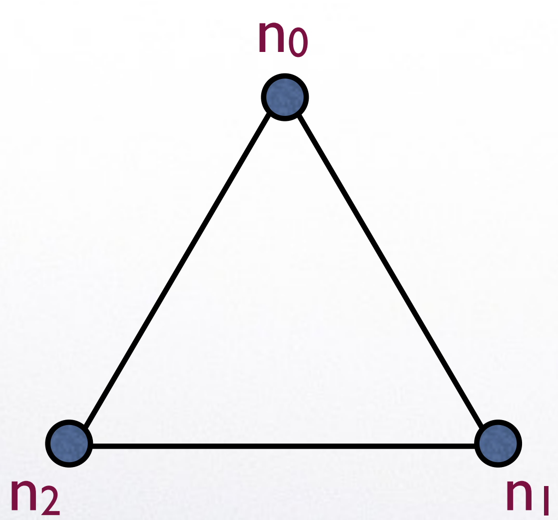
In these systems an input generates only one output

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



The entries of the channel matrix can be only 0 or 1

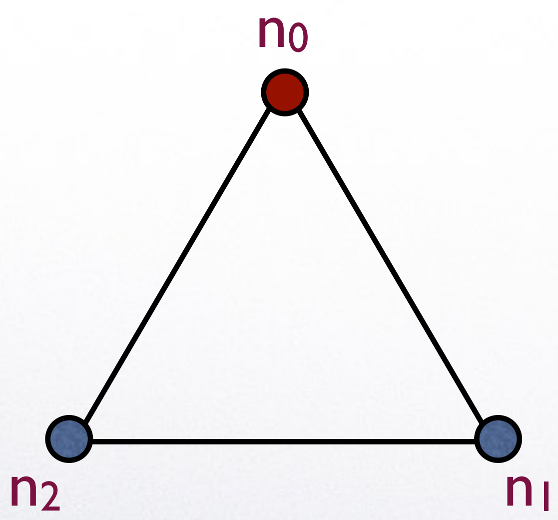
Example: DC nets (ring of 3 nodes, b=1)

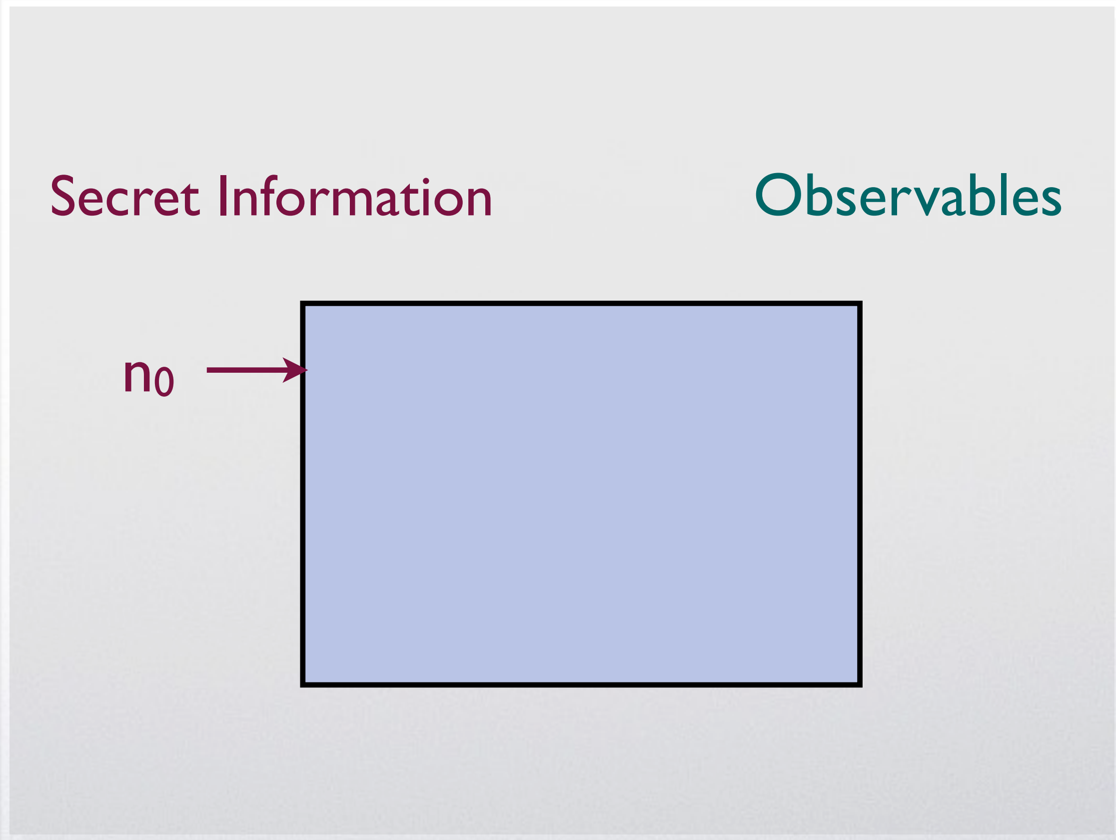


Secret Information

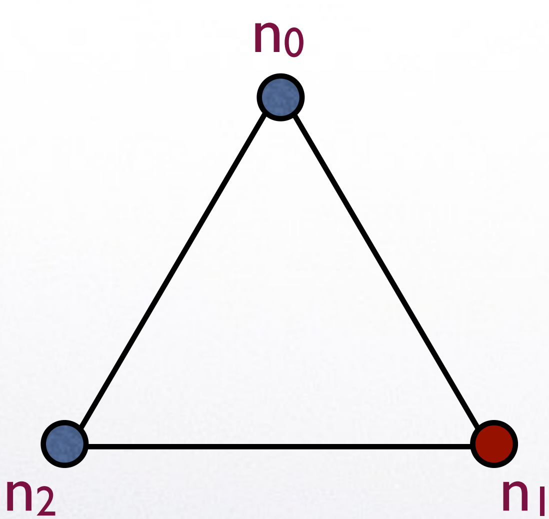
Observables

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



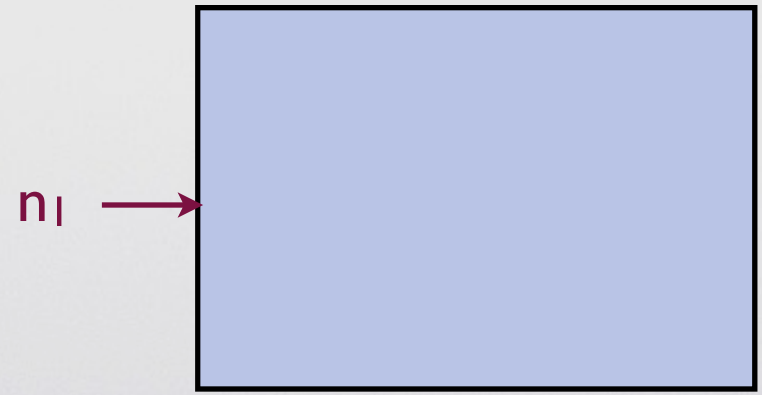


Example: DC nets (ring of 3 nodes, b=1)

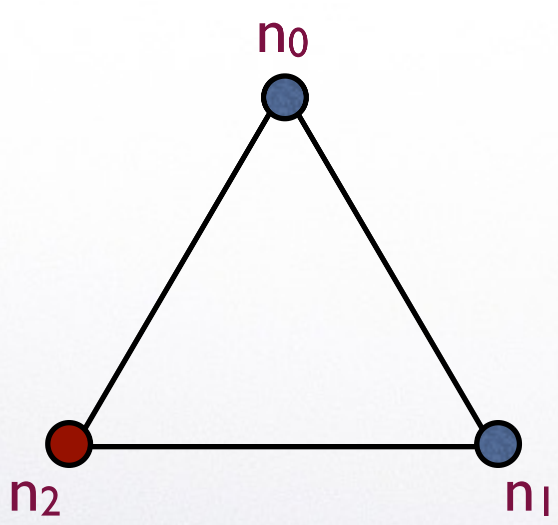


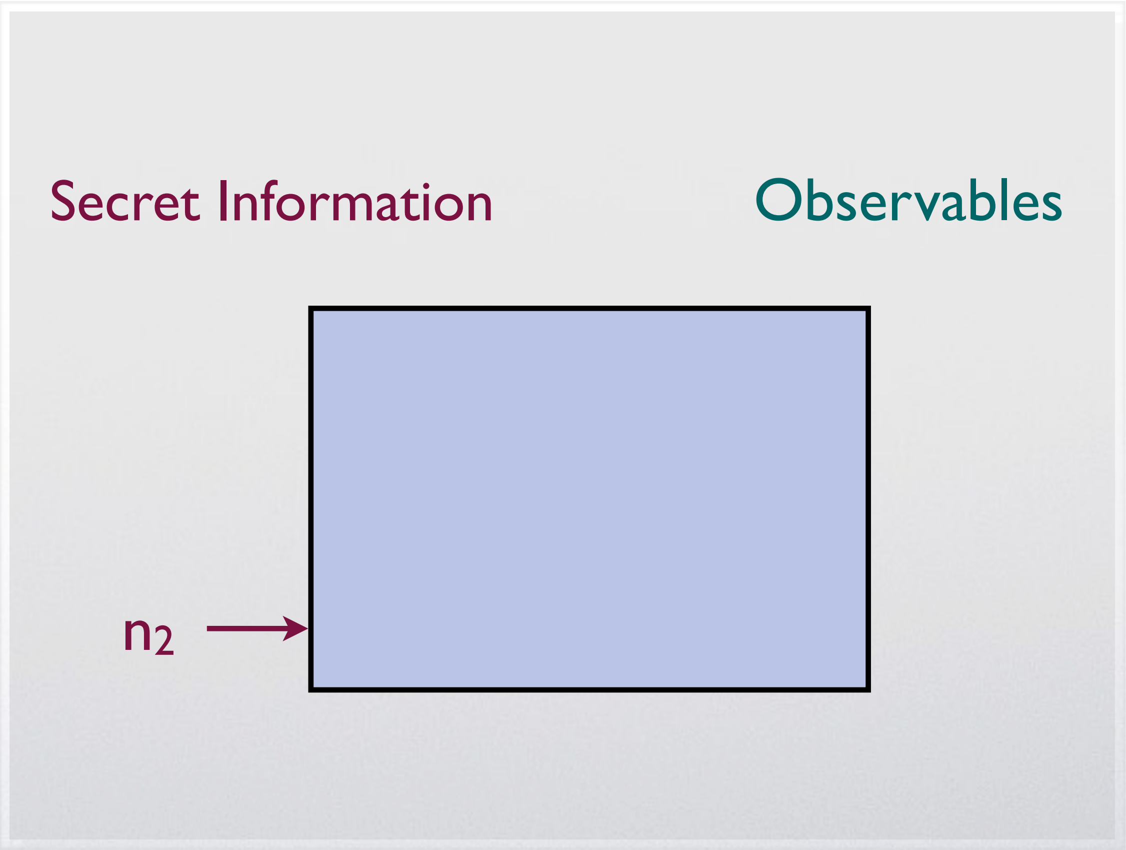
Secret Information

Observables

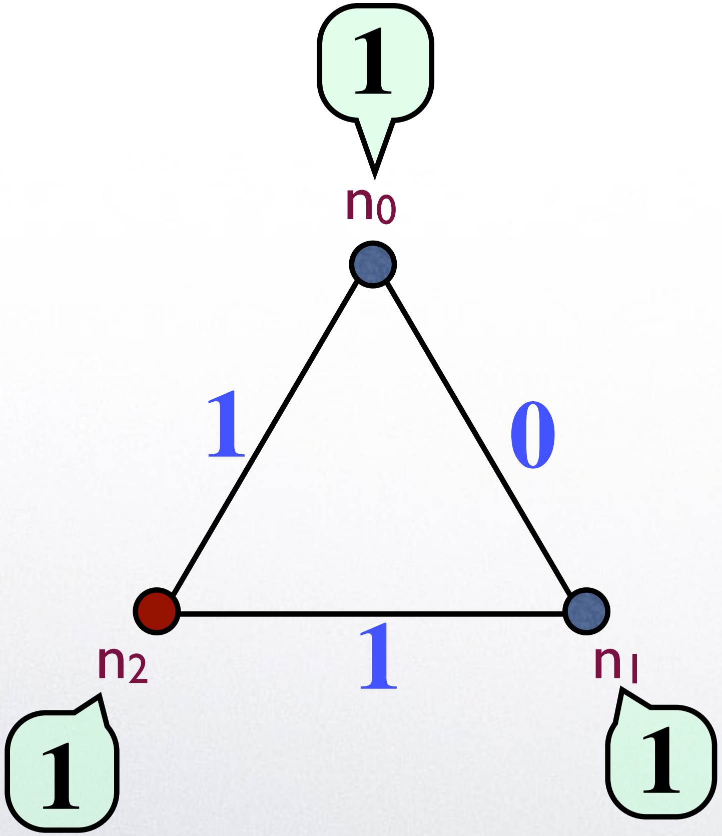


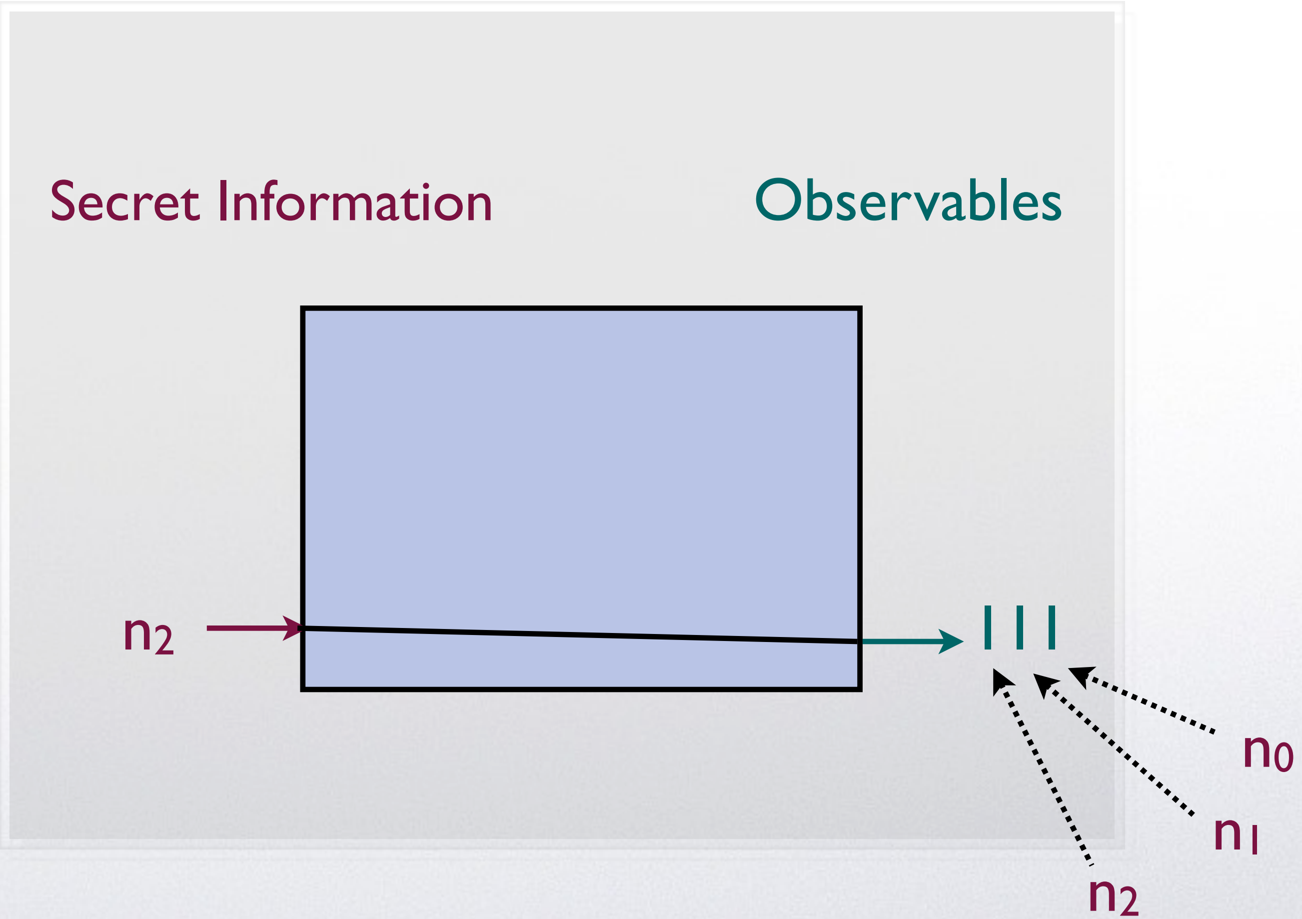
Example: DC nets (ring of 3 nodes, b=1)



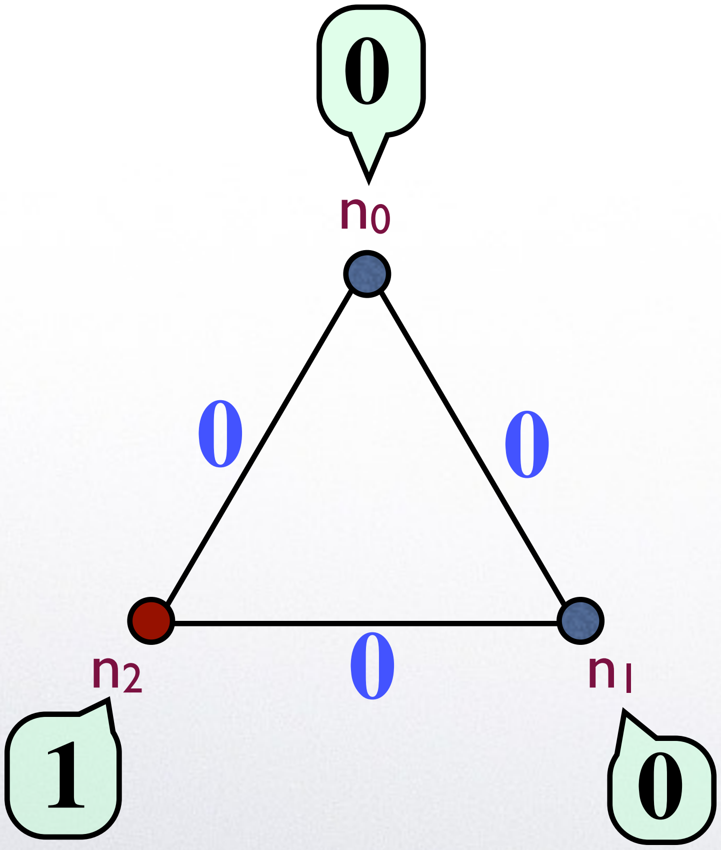


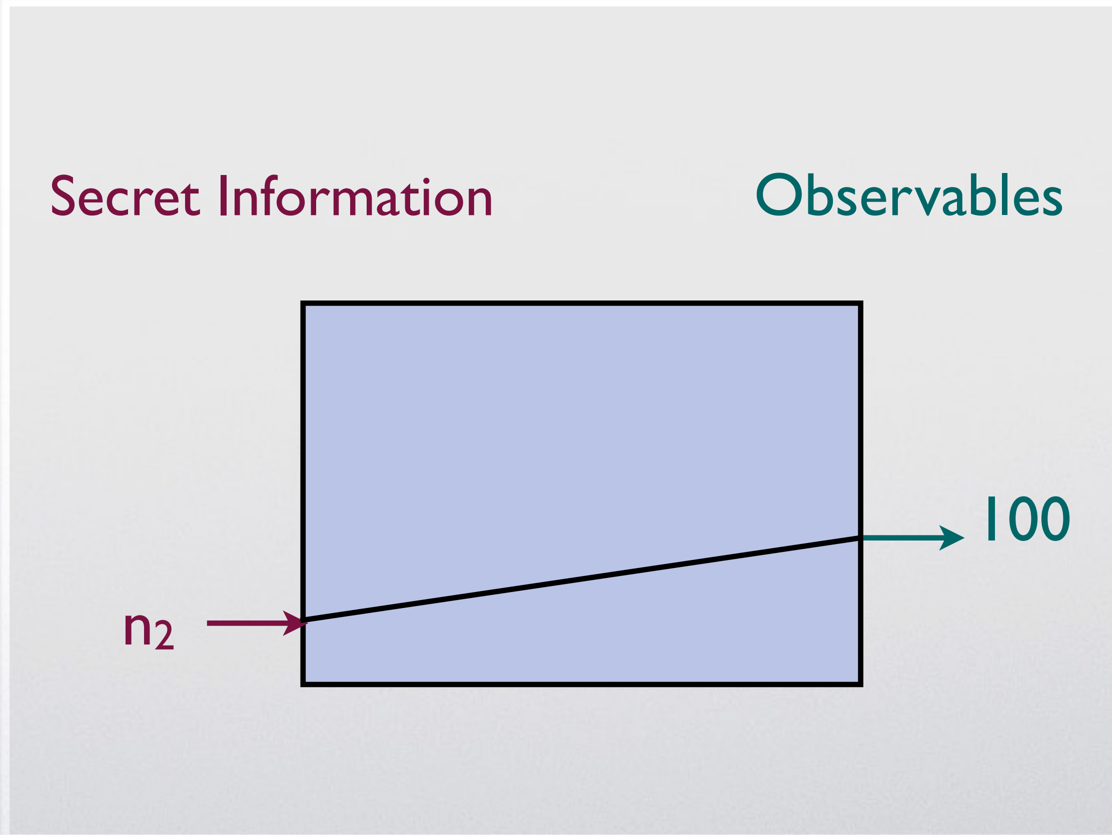
Example: DC nets (ring of 3 nodes, b=1)



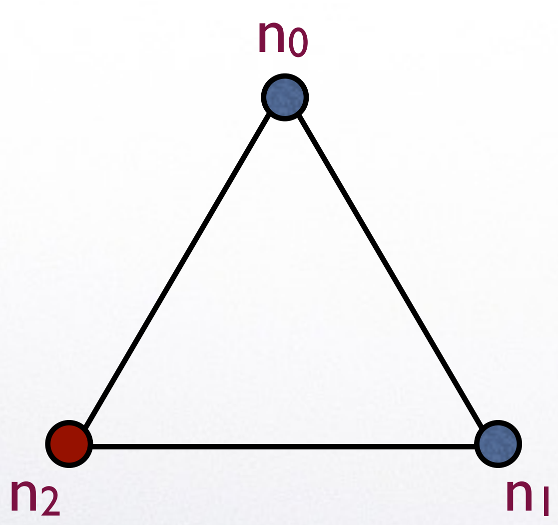


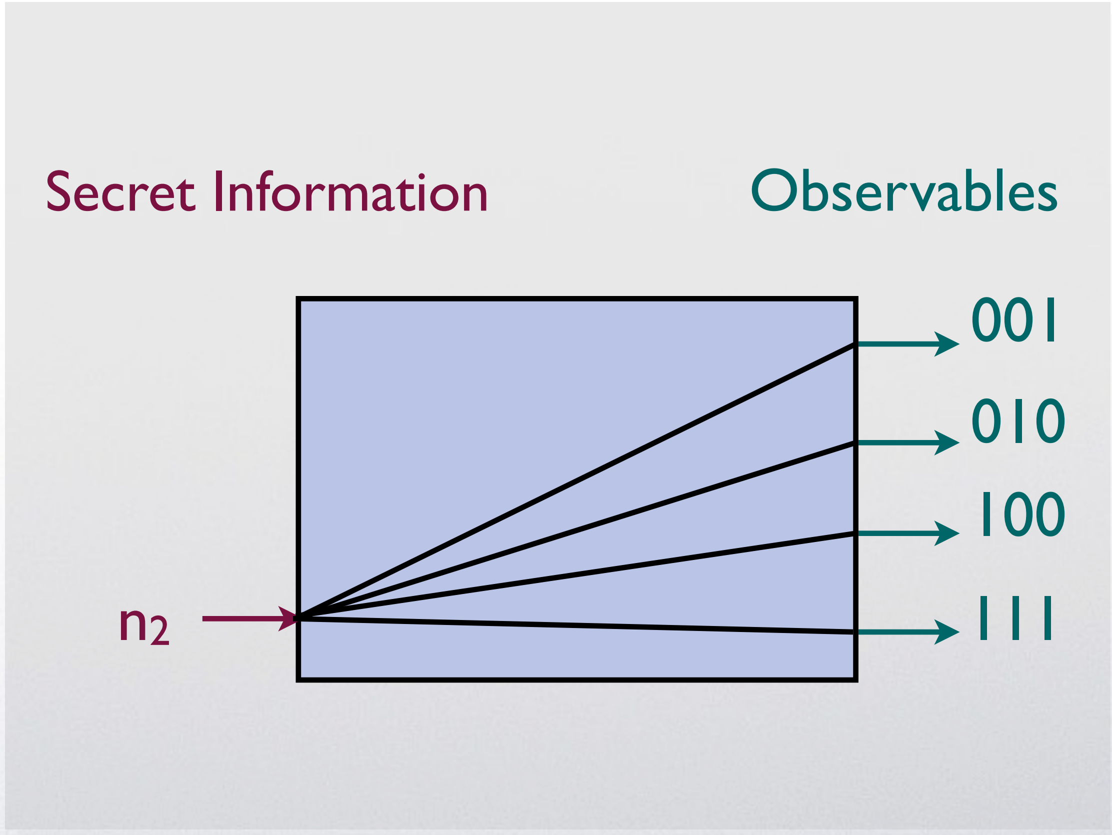
Example: DC nets (ring of 3 nodes, b=1)



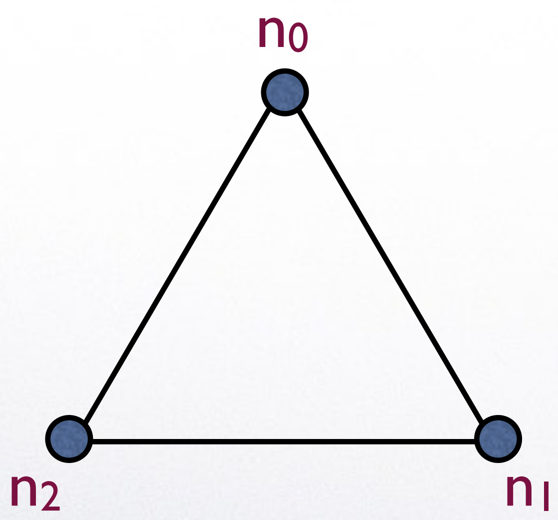


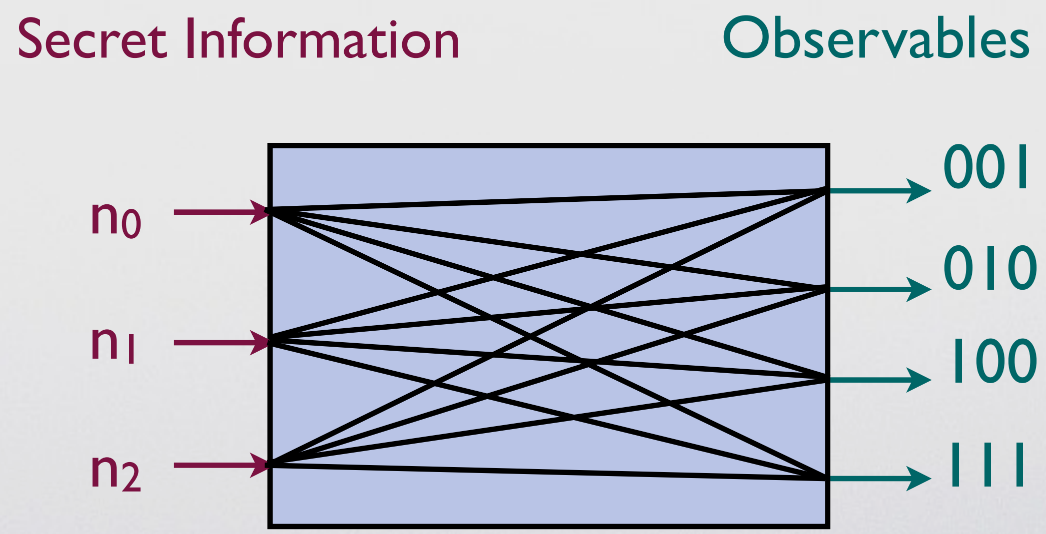
Example: DC nets (ring of 3 nodes, b=1)





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





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

	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: $\Pr(0) = \Pr(1) = \frac{1}{2}$

strong anonymity

biased coins: $\Pr(0) = \frac{2}{3}$, $\Pr(1) = \frac{1}{3}$

The source is more likely to declare 1 than 0

Quantitative Information Flow

- Intuitively, the **leakage** is the (probabilistic) information that the adversary **gains** about the **secret** through the **observables**

- Each observable **changes** the **prior** probability distribution on the secret values into a **posterior** probability distribution according to the **Bayes** theorem (Bayesian update)

- In the average, the posterior probability distribution gives a **better hint** about the actual secret value

Bayesian update: prior ⇒ posterior

Bayesian update: prior ⇒ posterior

p(n)		001	010	100	111
½	n_0	⅓	$^2/_9$	$^2/_9$	$^2/_9$
¼	n_1	$^2/_9$	⅓	$^2/_9$	$^2/_9$
¼	n_2	$^2/_9$	$^2/_9$	⅓	$^2/_9$

prior secret prob

p(o|n)
conditional prob

Bayesian update: prior ⇒ posterior

$p(n)$		001	010	100	111
$\frac{1}{2}$	n_0	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{2}{9}$	$\frac{2}{9}$
$\frac{1}{4}$	n_1	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{2}{9}$	$\frac{2}{9}$
$\frac{1}{4}$	n_2	$\frac{2}{9}$	$\frac{2}{9}$	$\frac{1}{3}$	$\frac{2}{9}$

prior secret prob

$p(o|n)$
conditional prob

	001	010	100	111
n_0	$\frac{1}{6}$	$\frac{1}{9}$	$\frac{1}{9}$	$\frac{1}{9}$
n_1	$\frac{1}{18}$	$\frac{1}{12}$	$\frac{1}{18}$	$\frac{1}{18}$
n_2	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{12}$	$\frac{1}{18}$

$p(n,o)$
joint prob

Bayesian update: prior ⇒ posterior

p(n)		001	010	100	111
½	n_0	⅓	²⁄₉	²⁄₉	²⁄₉
¼	n_1	²⁄₉	⅓	²⁄₉	²⁄₉
¼	n_2	²⁄₉	²⁄₉	⅓	²⁄₉

prior secret prob

p(o|n)
conditional prob

	p(o)	⁵⁄₁₈	¼	¼	²⁄₉
		001	010	100	111
n_0		⅙	⅑	⅑	⅑
n_1		¹⁄₁₈	¹⁄₁₂	¹⁄₁₈	¹⁄₁₈
n_2		¹⁄₁₈	¹⁄₁₈	¹⁄₁₂	¹⁄₁₈

obs prob

p(n,o)
joint prob

Bayesian update: prior ⇒ posterior

$$p(n|o) = \frac{p(n,o)}{p(o)}$$

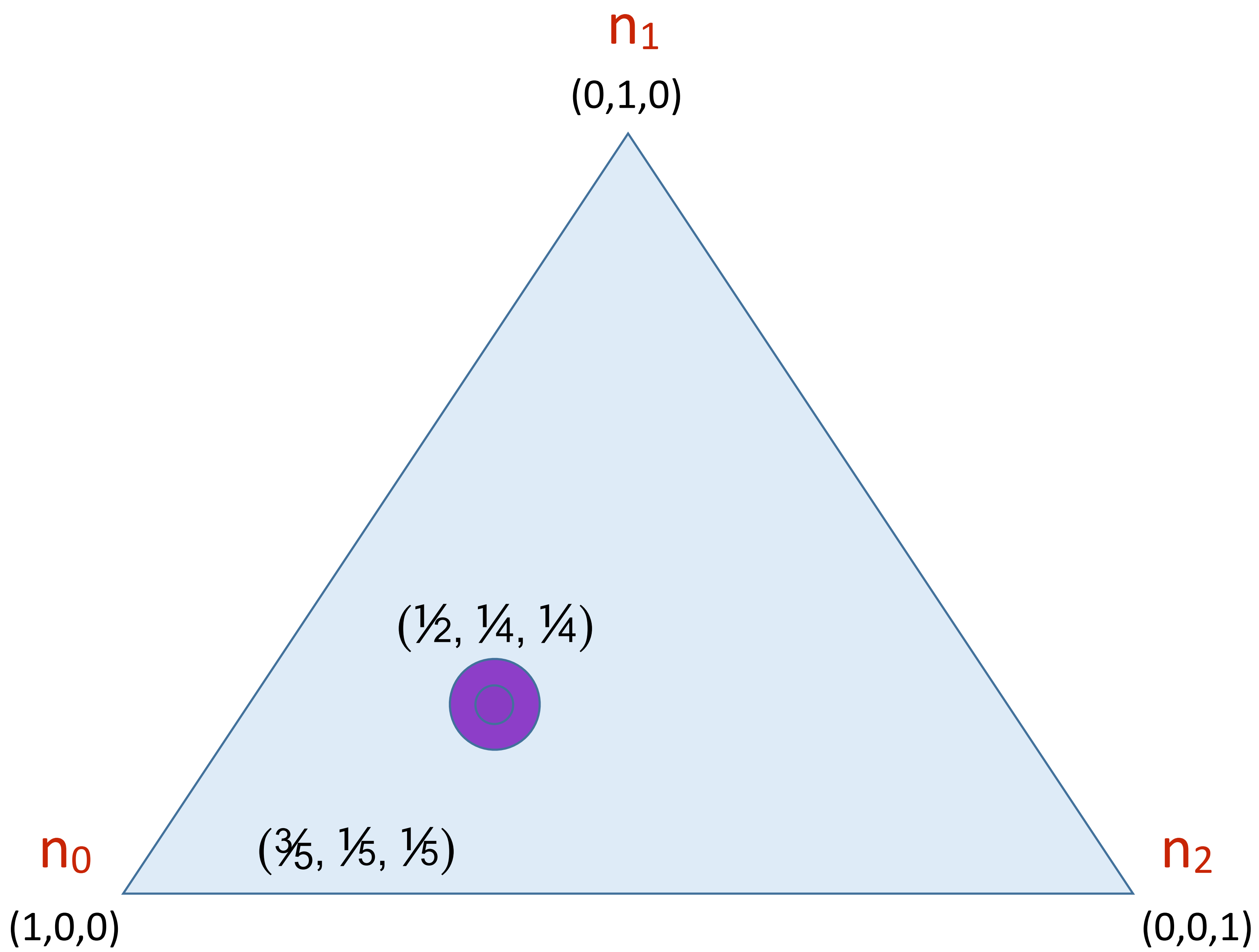
p(n|001) post secret prob		001	010	100	111
3⁄5	n_0	1⁄3	2⁄9	2⁄9	2⁄9
1⁄5	n_1	2⁄9	1⁄3	2⁄9	2⁄9
1⁄5	n_2	2⁄9	2⁄9	1⁄3	2⁄9

p(o|n)
conditional prob

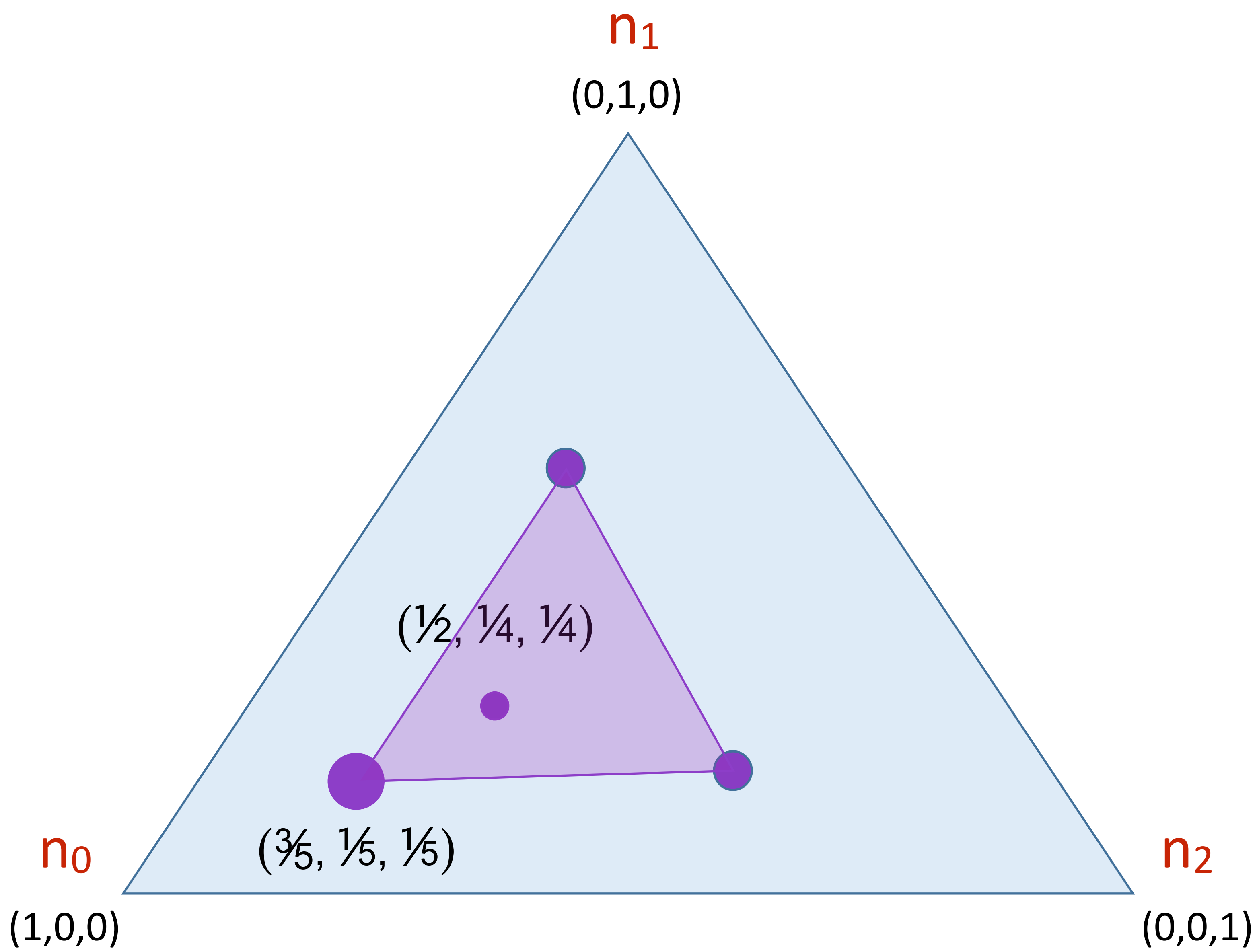
	001	010	100	111
p(o) obs prob	5⁄18	1⁄4	1⁄4	2⁄9
n_0	1⁄6	1⁄9	1⁄9	1⁄9
n_1	1⁄18	1⁄12	1⁄18	1⁄18
n_2	1⁄18	1⁄18	1⁄12	1⁄18

p(n,o)
joint prob

A graphical representation of the Bayesian update



A graphical representation of the Bayesian update



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 "easily" the adversary can discover the secret
- **Mutual information** I(S;O)
  - Degree of correlation between the input S and the output O
  - formally defined as difference between:
    - H(S), the entropy of S ***before*** knowing, and
    - H(S|O), the entropy of S ***after*** knowing O
  - It can be used to measure the leakage:

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

  - H(S) depends only on the prior; H(S|O) can be computed using the prior and the channel matrix

Entropy and Operational Interpretation

In the realm of security, there is no unique notion of entropy.
A suitable notion of entropy should have an **operational interpretation** in terms of the kind of **adversary** we want to **model** , namely:

- the kind of attack (how he attacks, the means at his disposal), and
- the goal of the attack and how we measure its success in achieving them

A general **model of adversary** [Köpf and Basin, CCS'07]:

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

Entropy

Example of adversary:

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

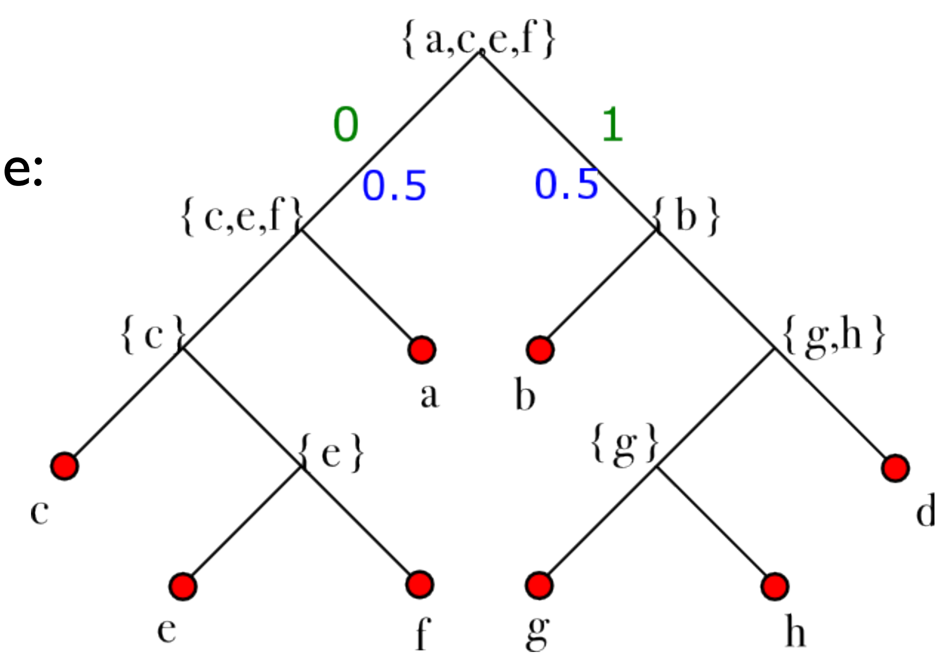
It is possible to prove that the best strategy for the adversary is to split each time the search space in two subspaces with same probability masses.
This gives a perfectly balanced tree.

Entropy

Example: S ∈ { *a, b, c, d, e, f, g, h* }

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

One possible way to split the tree:



Entropy

Since in the best strategy the tree is balanced, the number of questions needed to determine the value s of the secret is: **− log *p*(*s*)**
(log is in base 2)

Hence the **expected number** of questions is:

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

Uncertainty: **Shannon entropy**

Shannon entropy: properties

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

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

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

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

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

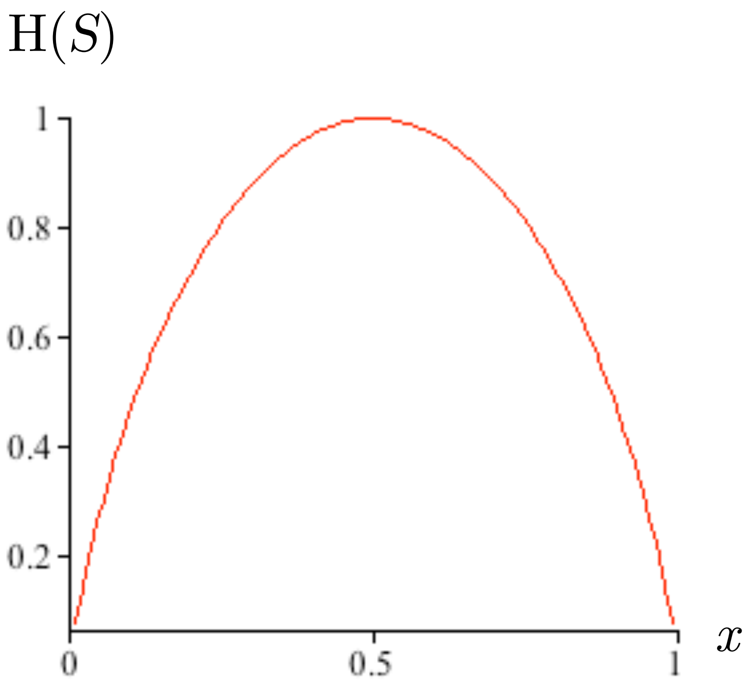
$$
\begin{aligned}
H(S) &= -\textstyle\sum_s p(s)\log p(s) \\
&= -2\frac{1}{4}\log\frac{1}{4} - 2\frac{1}{8}\log\frac{1}{8} - 4\frac{1}{16}\log\frac{1}{16} \\
&= 1 + \frac{3}{4} + 1 \\
&= \frac{11}{4}
\end{aligned}
$$

Shannon entropy: properties

The entropy is a concave function of the probability distribution

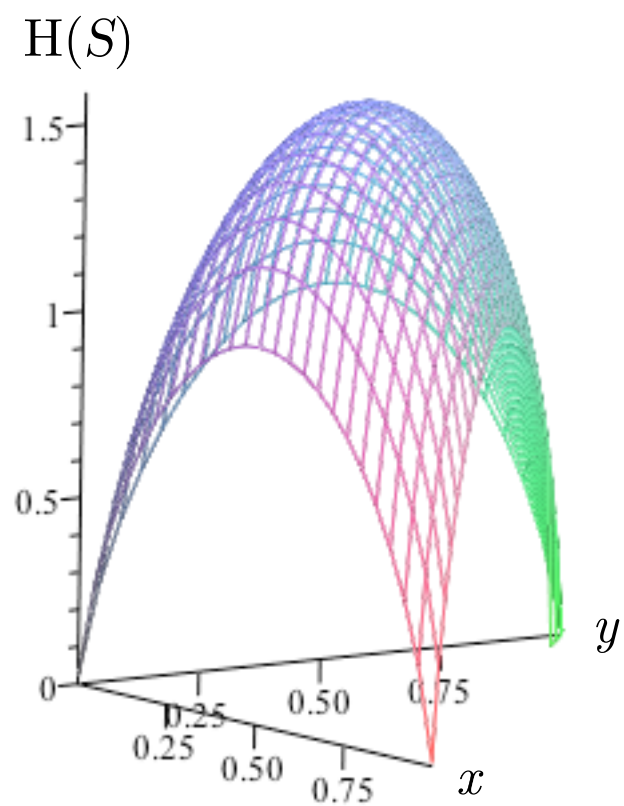
$$S = \{a, b\}$$

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



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

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



Shannon conditional entropy

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

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

Shannon leakage

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 $H(S) \geq H(S|O)$
  - the entropy may increase after one single observation, but in the average it cannot increase
- $H(S) = H(S|O)$ 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 C = max $I(S;O)$ over all priors (worst-case leakage)

Entropy: Alternative notions

As we argued before, there is no unique notion of vulnerability. It depends on:

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

Entropy: Alternative notions

We saw that if

- the questions are of the form: “is $S \in P$?”, and
- the measure of success is: the expected number of questions needed to find the value of S in the adversary’s best strategy

then the natural measure of protection is Shannon’s entropy

However, this model of attack does not seem so natural in security, and alternatives have been considered. In particular, the **limited-try attacks**

- The adversary has a limited number of attempts at its disposal
- The measure of success is the probability that he discovers the secret during these attempts (in his best strategy)

Obviously the best strategy for the adversary is to try first the values which have the highest probability

One try attacks: Rényi min-entropy

One-try attacks

- The questions are of the form: “is $S = s$?”
- The measure of success is: $-\log(\max_s p(s))$

Rényi min-entropy: $H_\infty(S) = -\log(\max_s p(s))$

Like in the case of Shannon entropy, $H_\infty(S)$ is highest when the distribution is uniform, and it is 0 when the distribution is a delta of Dirac (no uncertainty).

Conditional min-entropy

The expected value of the prob. of success (aka converse of the Bayes risk):

$$\begin{aligned}
\Pr_{succ}(S|O) &= \sum_o p(o)\, \Pr_{succ}(S|O=o) \\
&= \sum_o p(o) \max_s\, p(s|o) \\
&= \sum_o \max_s\, (p(o|s)\, p(s))
\end{aligned}$$

Now define $H_\infty(S|O) = -\log \Pr_{succ}(S|O)$ [Smith 2009]

Leakage in the min-entropy approach

A priori $$H_\infty(S) = -\log \max_s p(s)$$

A posteriori $$H_\infty(S|O) = -\log \sum_o \max_s (p(o|s) \cdot p(s))$$

Leakage = min-Mutual Inf. $$I_\infty(S;O) = H_\infty(S) - H_\infty(S|O)$$

Properties of the min-entropy leakage

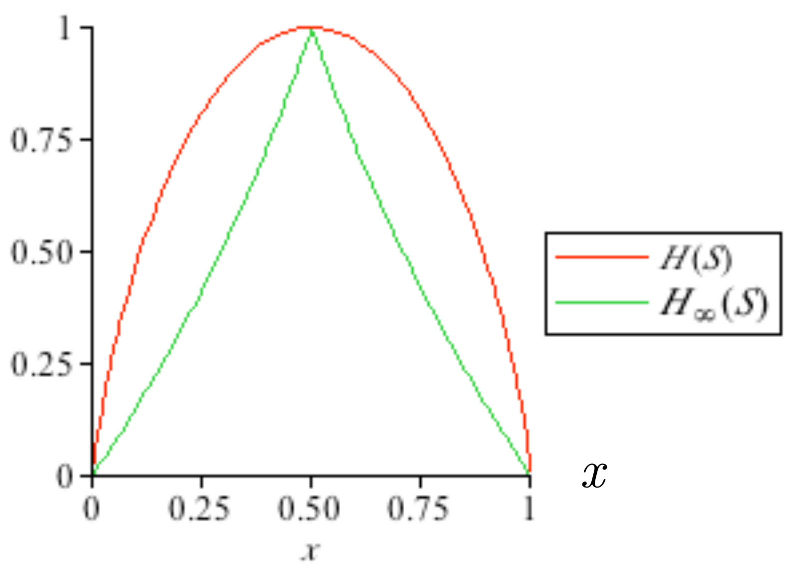
- In general $I_\infty(S;O) \geq 0$
- $I_\infty(S;O) = 0$ if all rows are the same (but not viceversa)
- Define min-capacity: C_∞ = max $I_\infty(S;O)$ over all priors. We have:

1. $C_\infty = 0$ if and only if all rows are the same
2. $C_\infty = C$ in the deterministic case
3. $C_\infty \geq C$ in general
4. C_∞ is obtained on the uniform distribution (but, in general, there can be other distribution that give maximum leakage)
5. **C_∞ = the log of the sum of the max of each column**

Rényi min-entropy vs. Shannon entropy

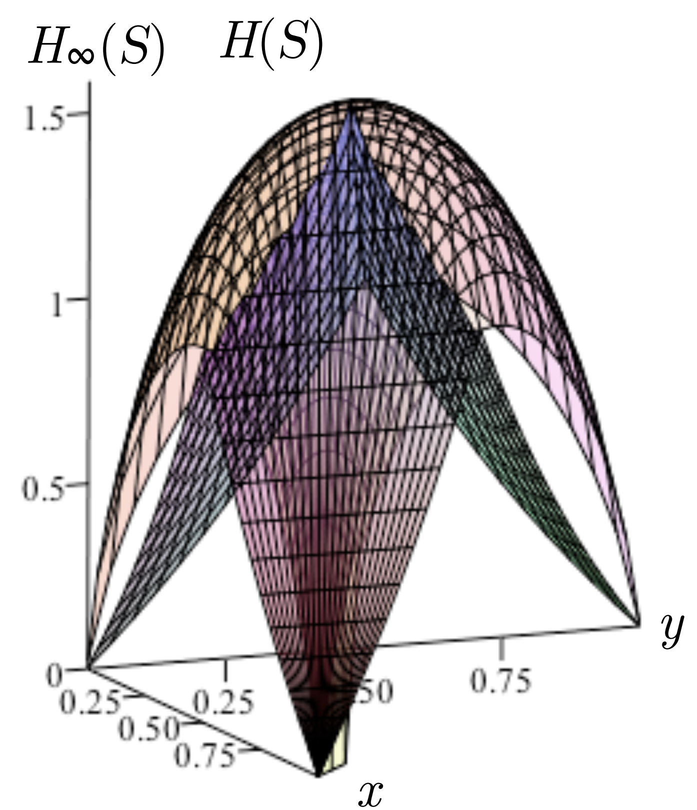
$S = \{a, b\}$

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



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

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

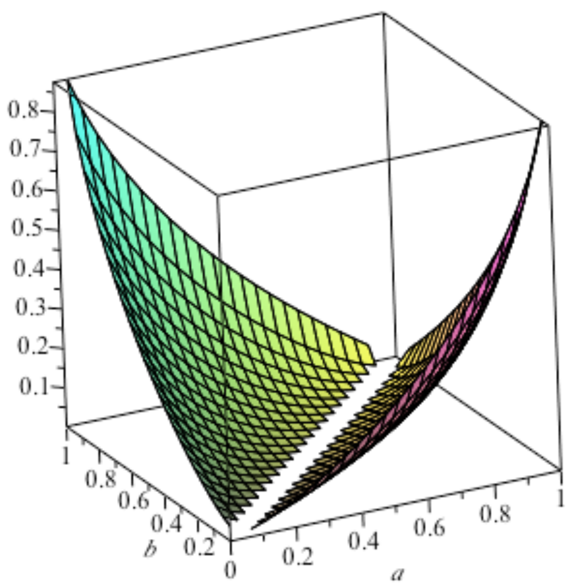


Rényi min entropy and conditional entropy are the log of piecewise linear functions

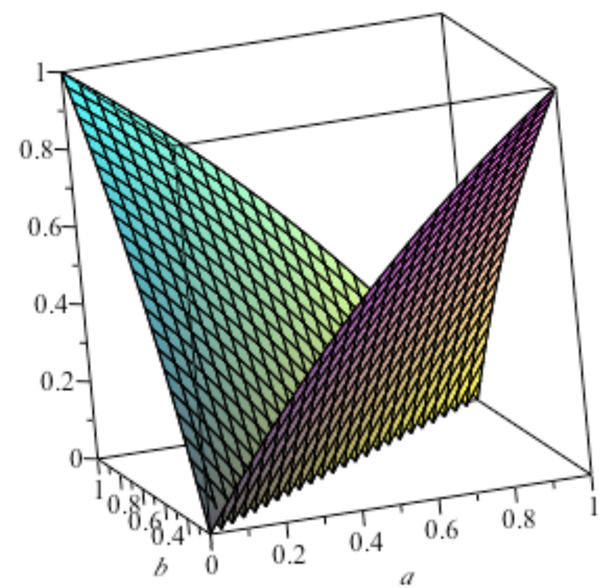
Shannon capacity vs. Rényi min-capacity

binary channel

a	1−a
b	1−b



Shannon capacity



Rényi min-capacity

In general, Rényi min capacity is an upper bound for Shannon capacity

Thank you !