Quantitative Approaches to Information Protection

Catuscia Palamidessi
INRIA Saclay

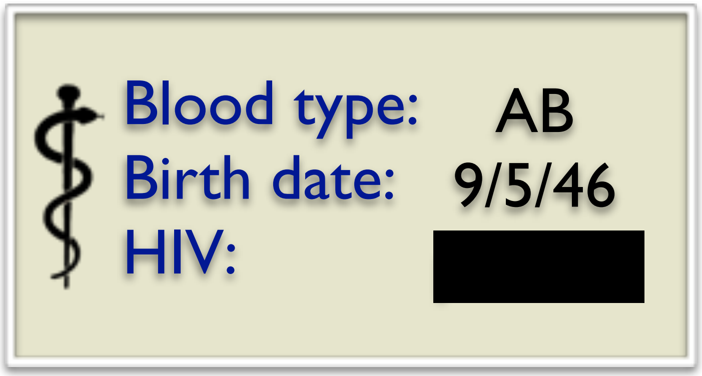
LOCALI meeting, Beijin 06/11/2013

Plan of the talk

- Motivations and Examples
- A General Quantitative Model
- Quantitative Information Flow
- Differential Privacy
- Privacy-Aware Geolocation

Protection of sensitive information

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



- Access control and encryption are not sufficient! Systems could leak secret information through correlated observables.
  - The notion of "observable" depends on the system and on the capabilities of the adversary
- This talk will focus on the inference of secret information through the observables.

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

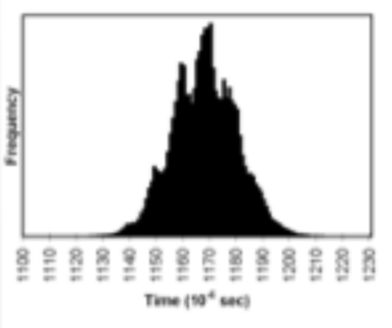
➪ **need quantitative ways to measure the leak**

Leakage through correlated observables

Password checking

Election tabulation

Timings of decryptions



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)

Intrinsic leakage

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

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

  end if
end for
```

Example 2

Password checker 2

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

More efficient, but what about security?

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

Example 2

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

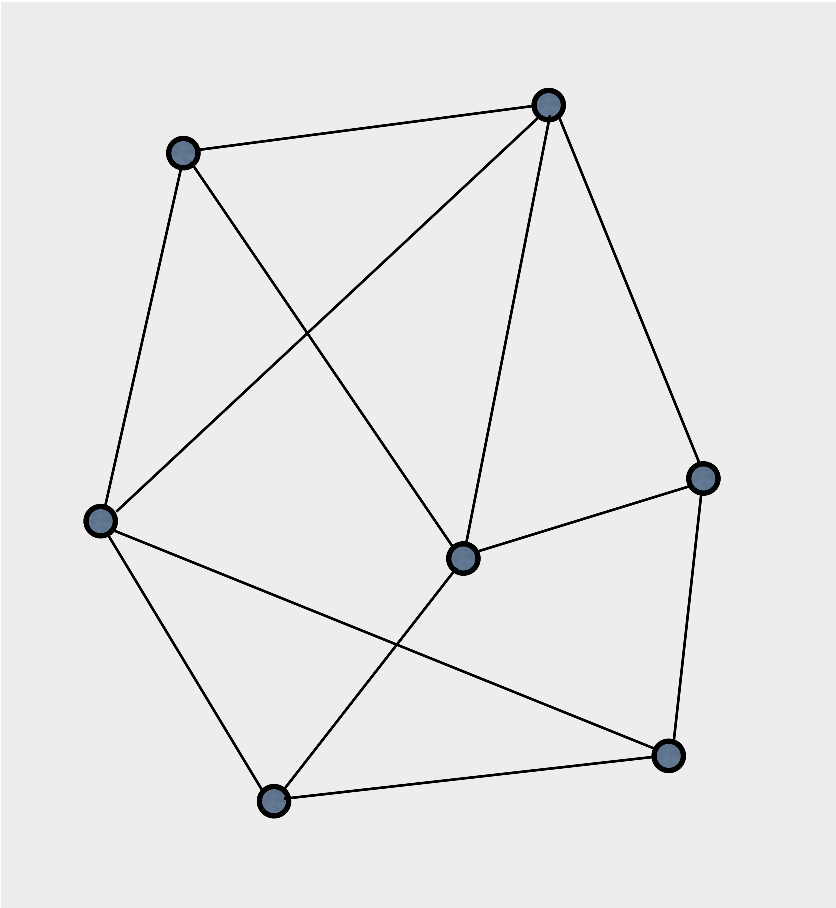
If the adversary can measure the execution time, then he can also learn the longest correct prefix of the password

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

Example 3

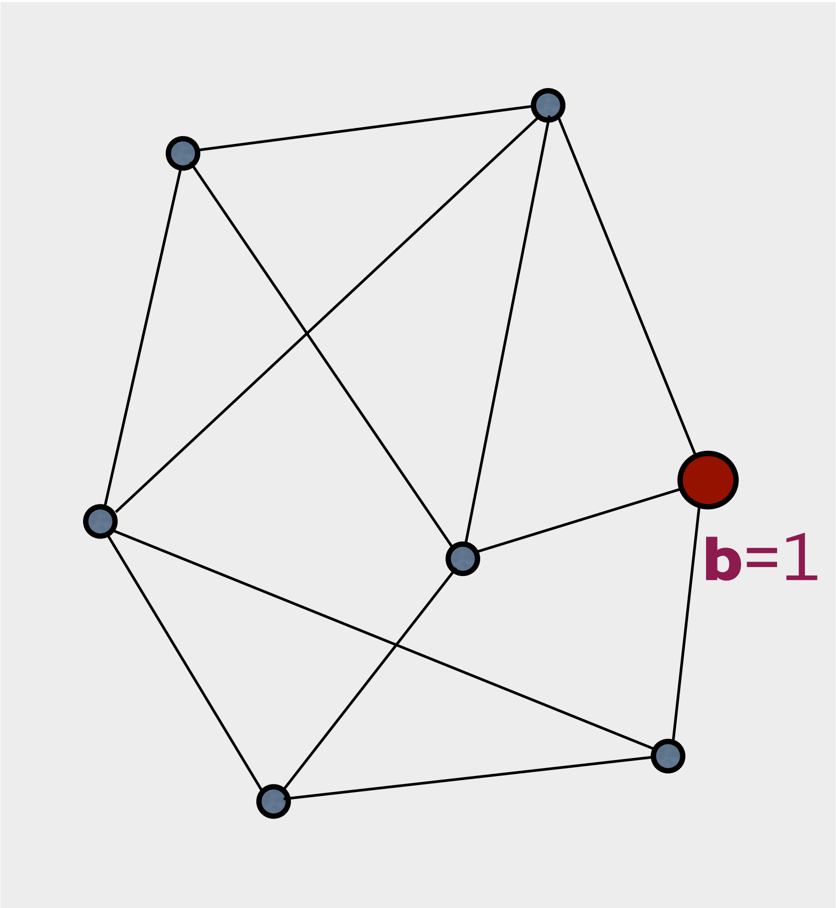
Example of Anonymity Protocol: DC Nets [Chaum'88]

- 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



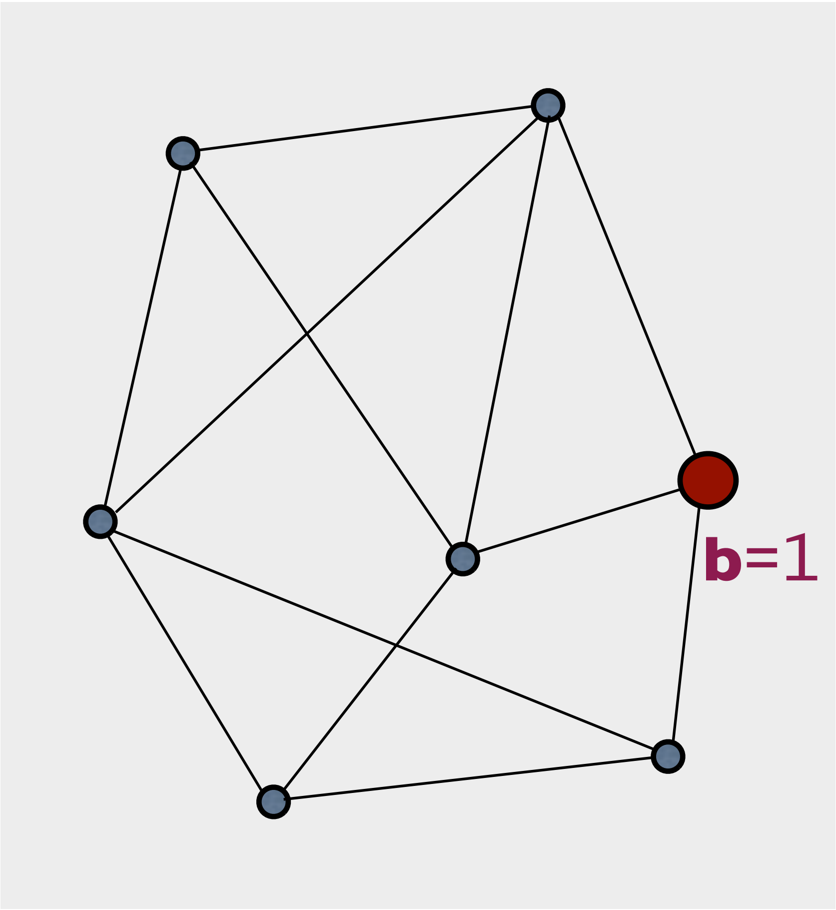
Example of Anonymity Protocol: DC Nets [Chaum'88]

- 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



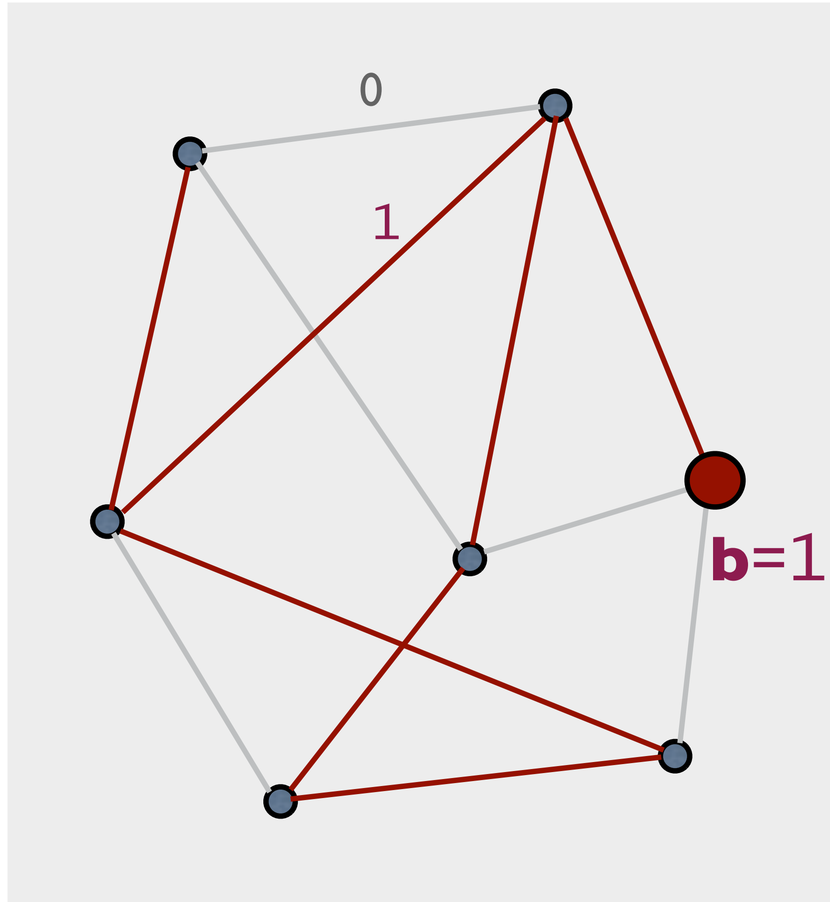
Chaum’s solution

- Associate to each edge a fair binary coin



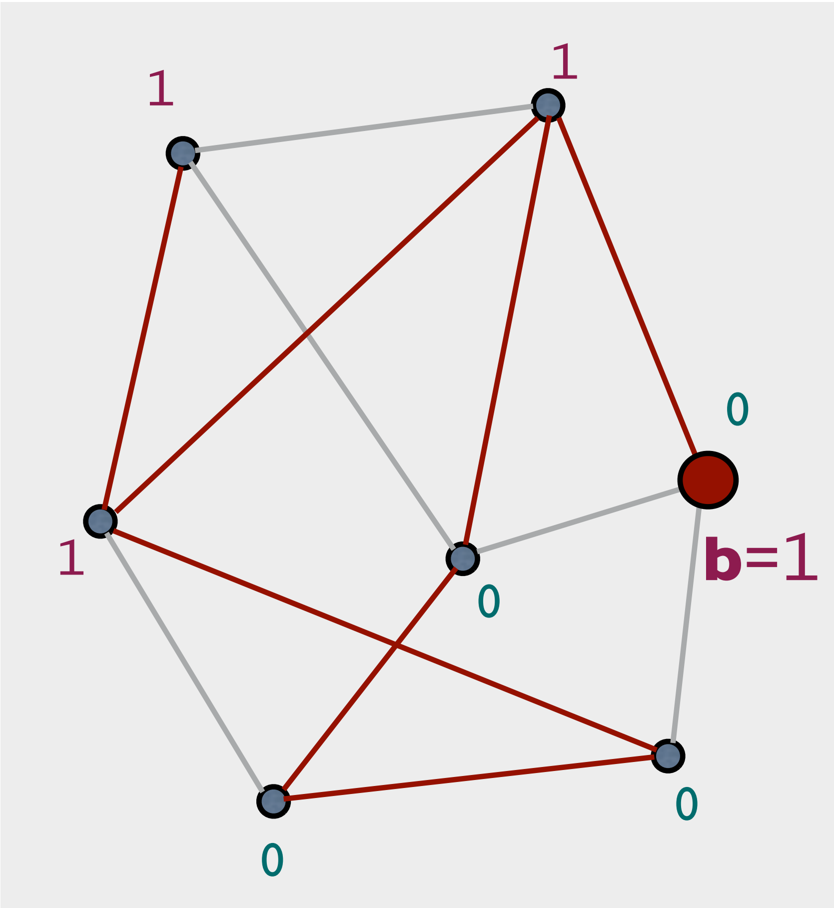
Chaum's solution

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



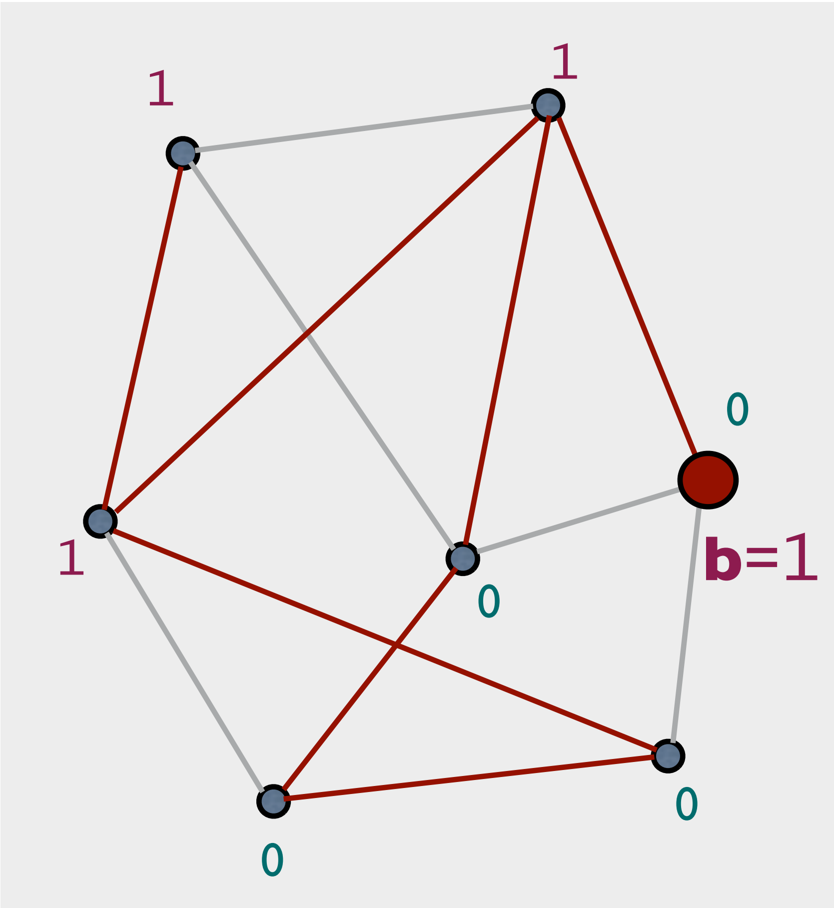
Chaum's solution

- 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**



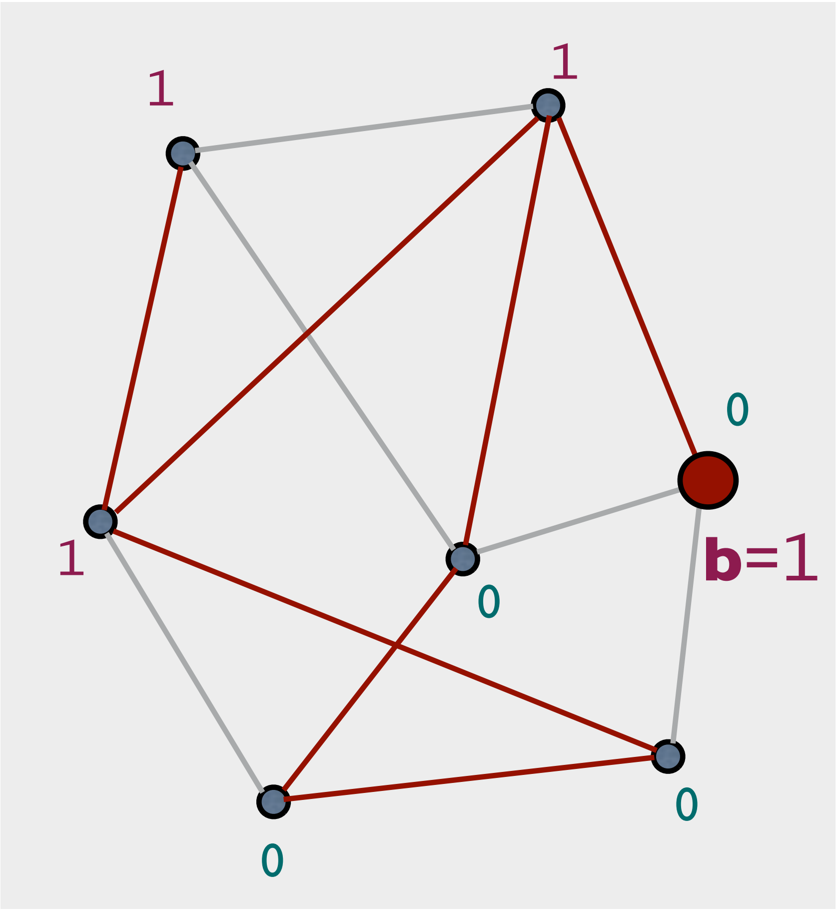
Anonymity of DC Nets

Observables: An external attacker can only see the declarations of the nodes (not the results of the coins)

Question: Does the protocol protect the anonymity of the source? In what sense?

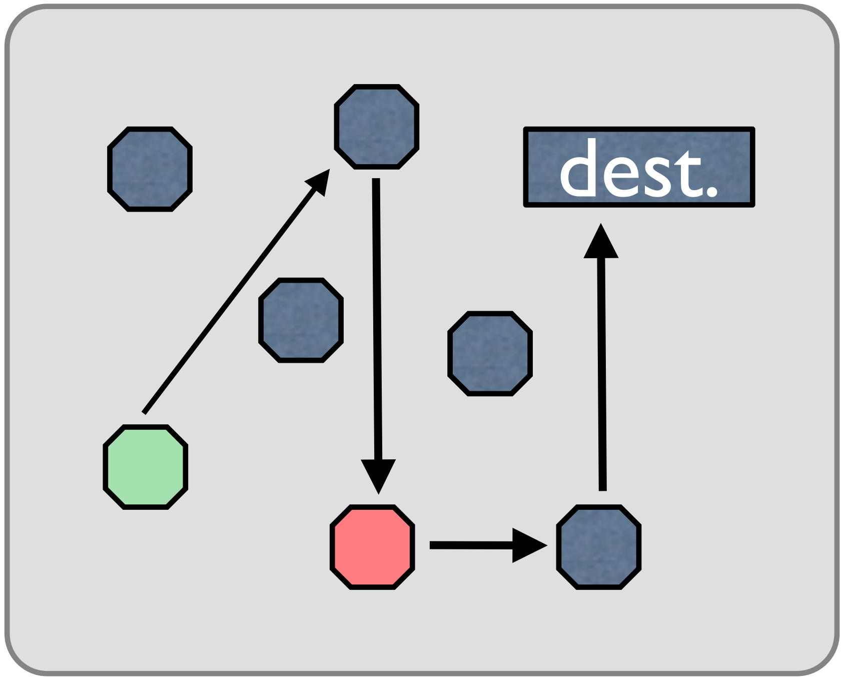
Strong anonymity (Chaum)

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



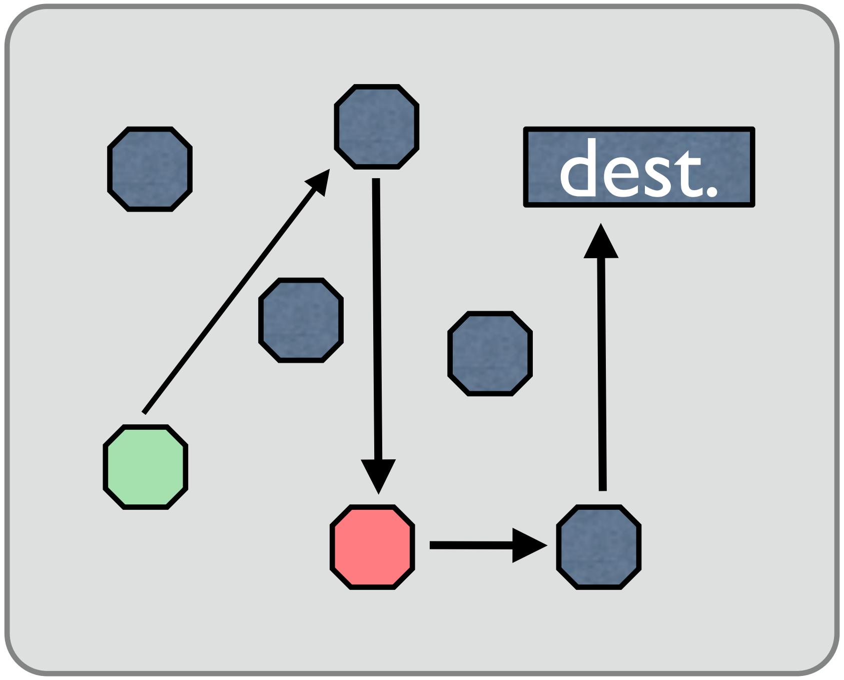
Example 4: Crowds [Rubin and Reiter'98]

- 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 4: 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



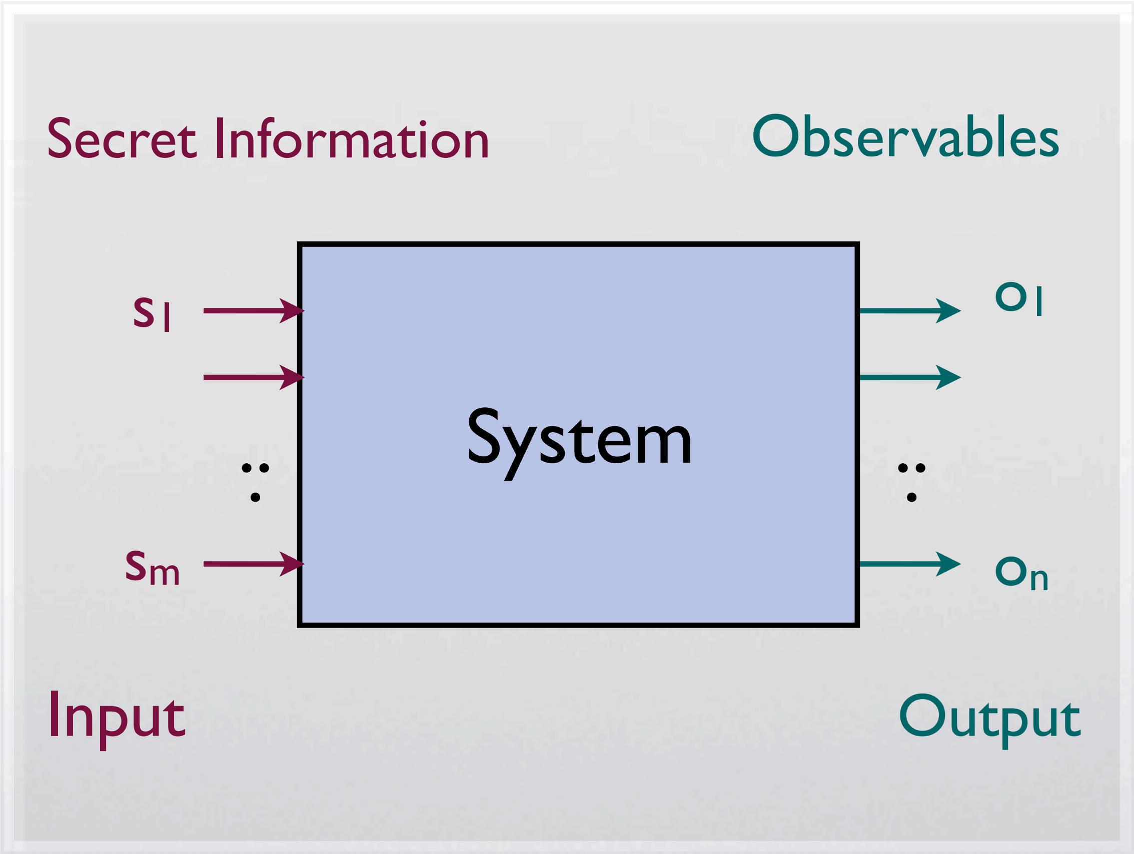
Probable innocence: under certain conditions, an attacker who intercepts the message from x cannot attribute more than 0.5 probability to x to be the initiator

Common features

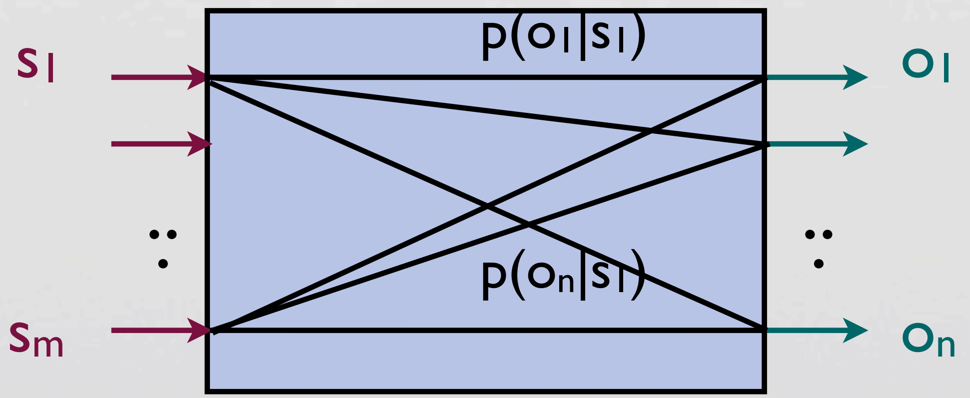
- Secret information
  - Password checker: The password
  - DC: the identity of the source
  - Crowds: the identity of the initiator
- Public information (Observables)
  - Password checker: The result (OK / Fail) and the execution time
  - DC: the declarations of the nodes
  - Crowds: the identity of 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	...	o_n
s_1	$p(o_1\|s_1)$	...	$p(o_n\|s_1)$
$\vdots$	$\vdots$		
s_m	$p(o_1\|s_m)$		$p(o_n\|s_m)$

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

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

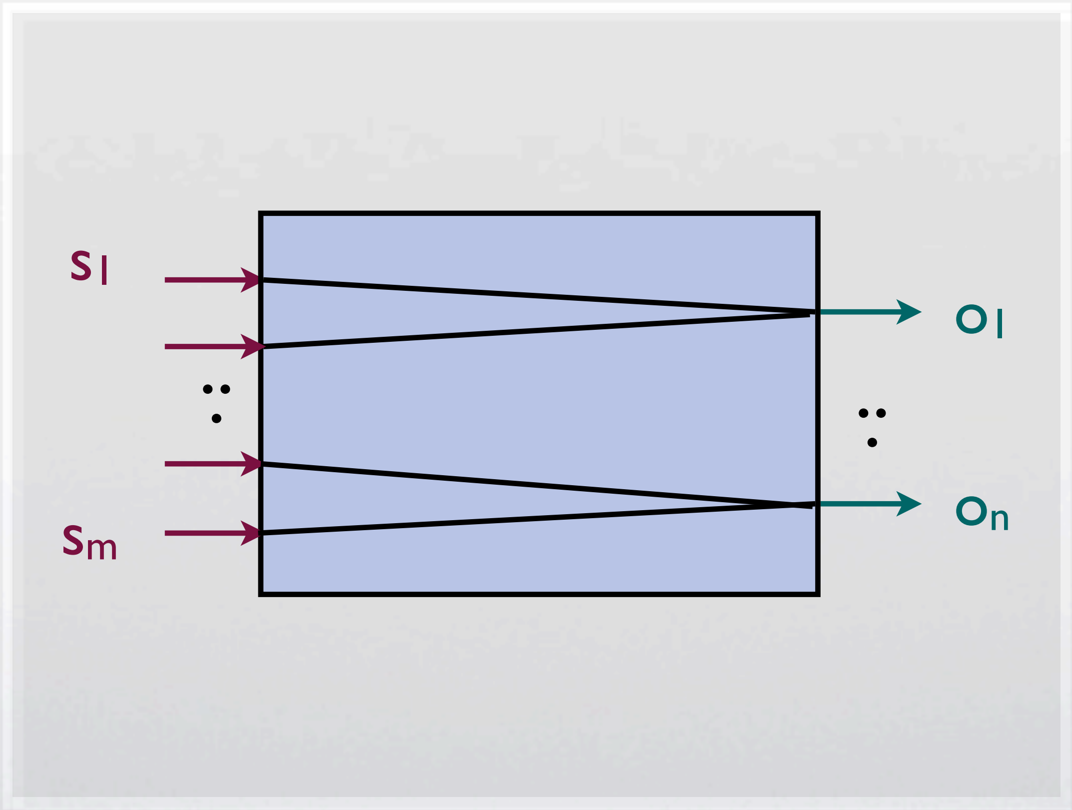
In a information-theoretic channel these conditional probabilities are independent from the input distribution

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

Particular case: **Deterministic systems**

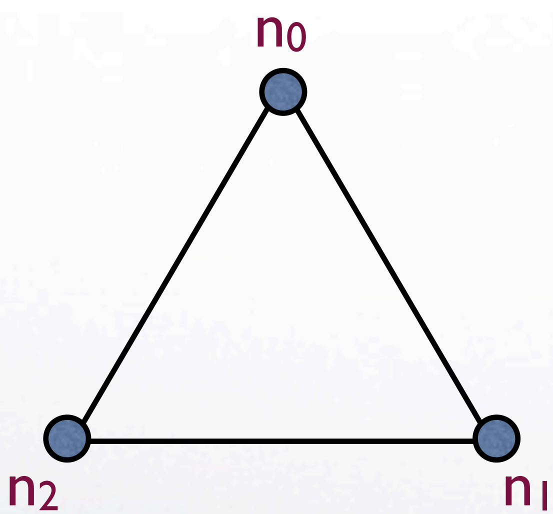
In these systems an input generates only one output

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

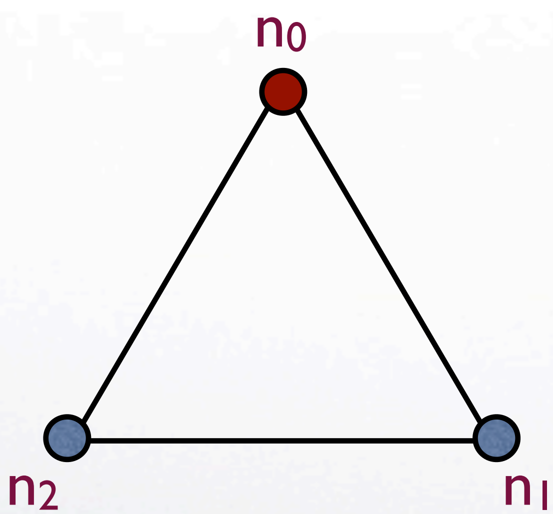
Example: DC nets (ring of 3 nodes, b=1)

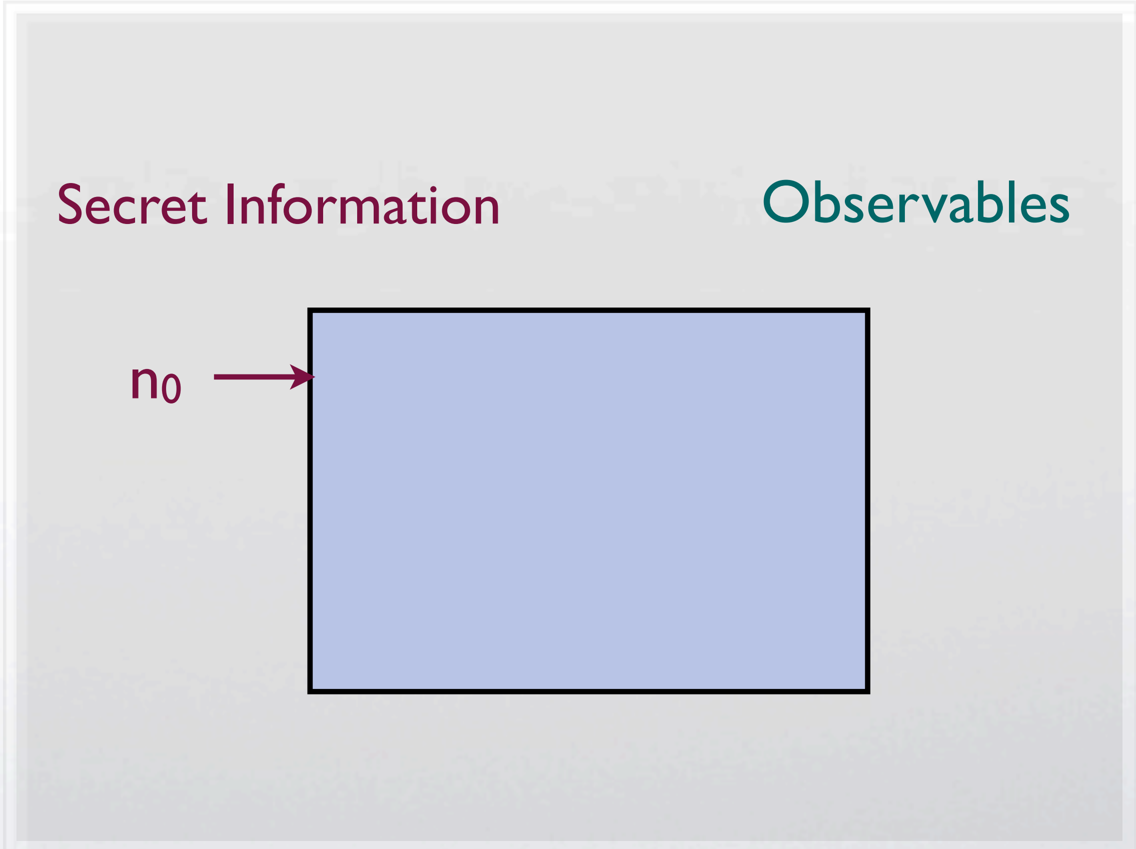


Secret Information

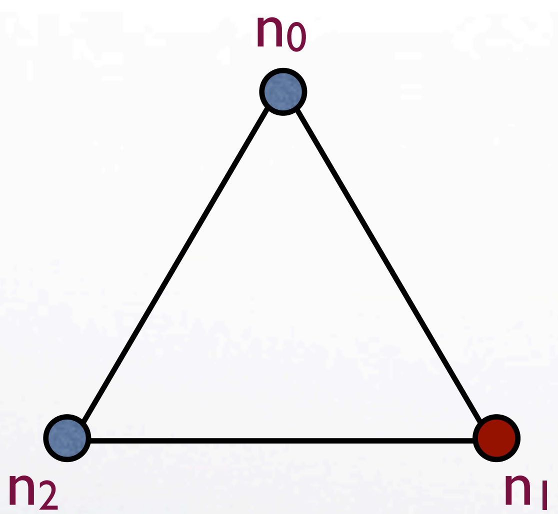
Observables

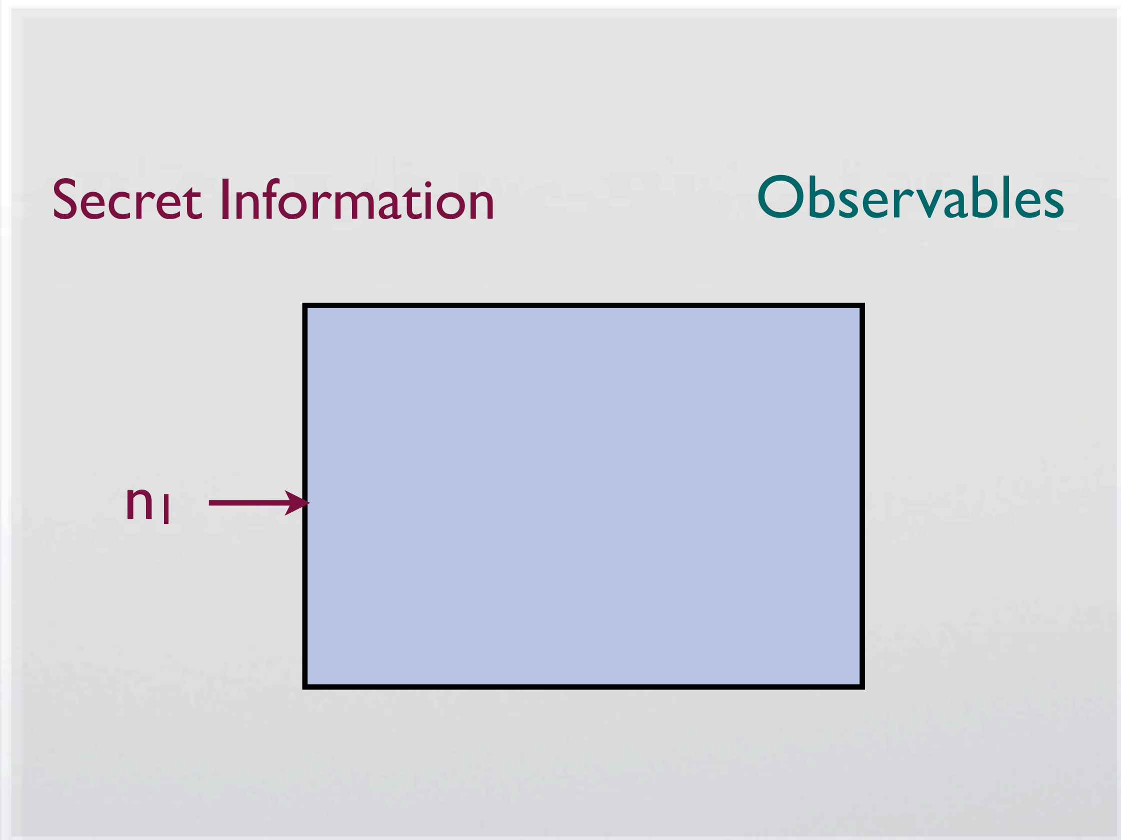
Example: DC nets (ring of 3 nodes, b=1)



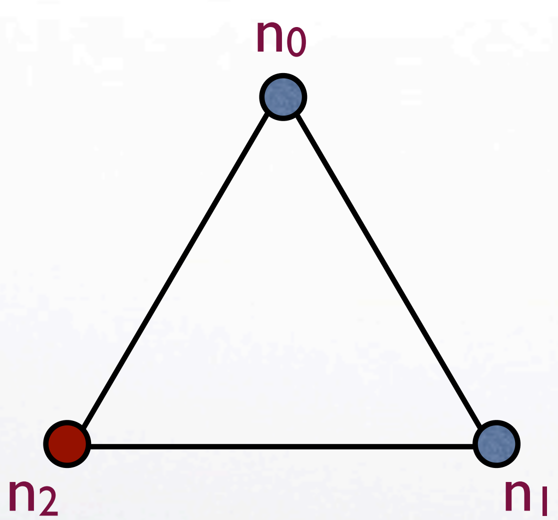


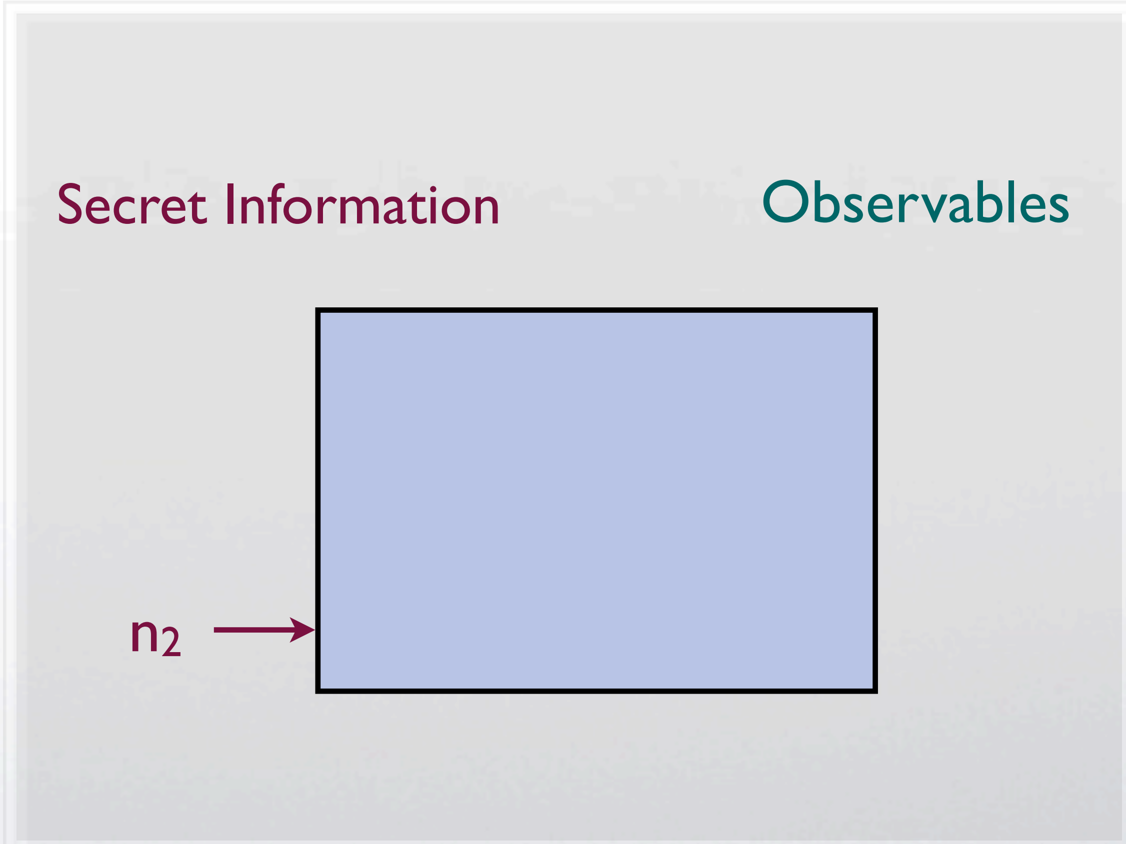
Example: DC nets (ring of 3 nodes, b=1)



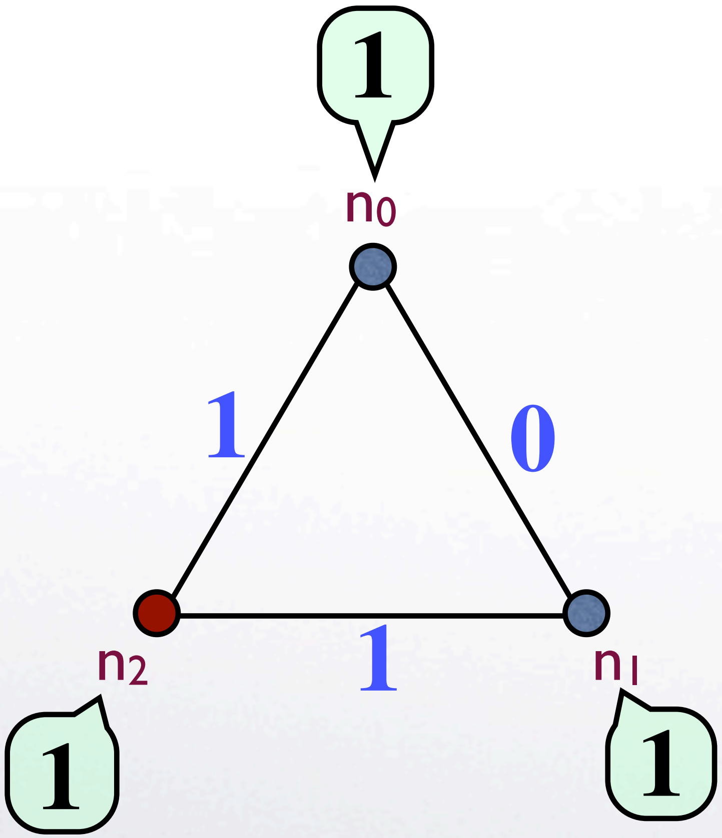


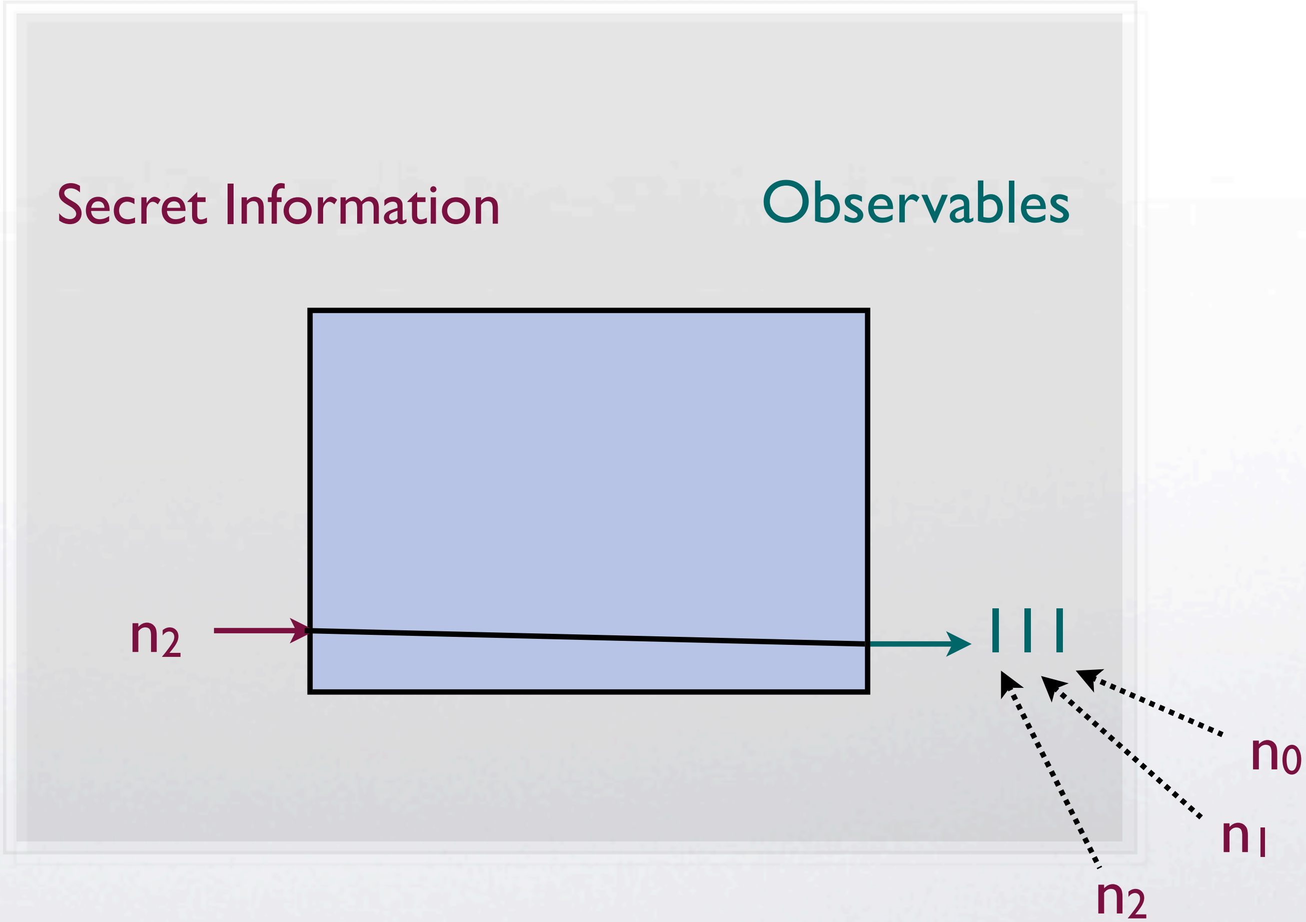
Example: DC nets (ring of 3 nodes, b=1)



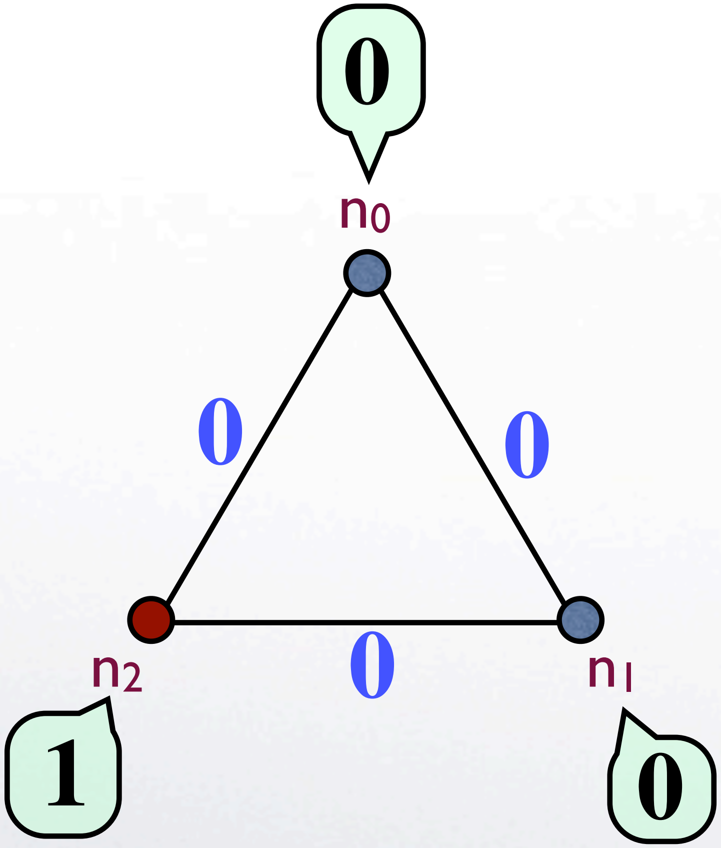


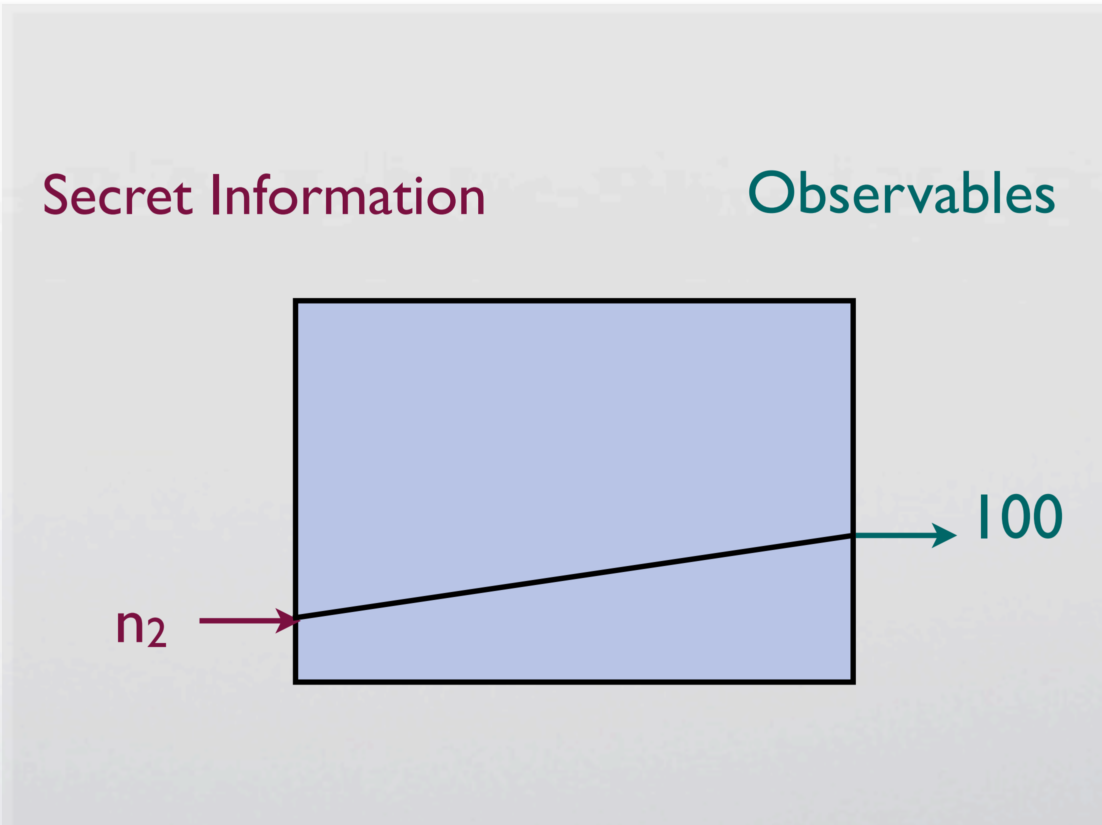
Example: DC nets (ring of 3 nodes, b=1)



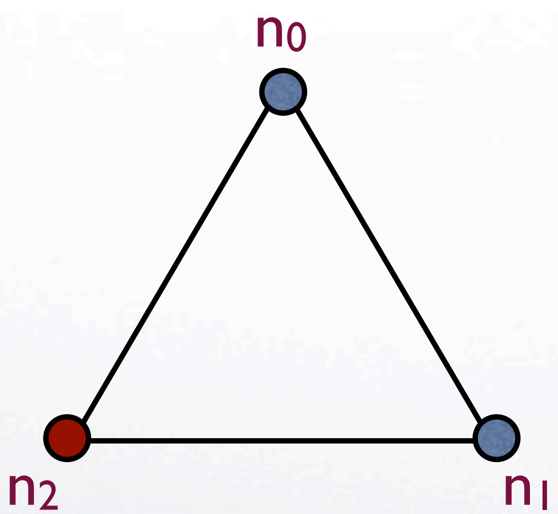


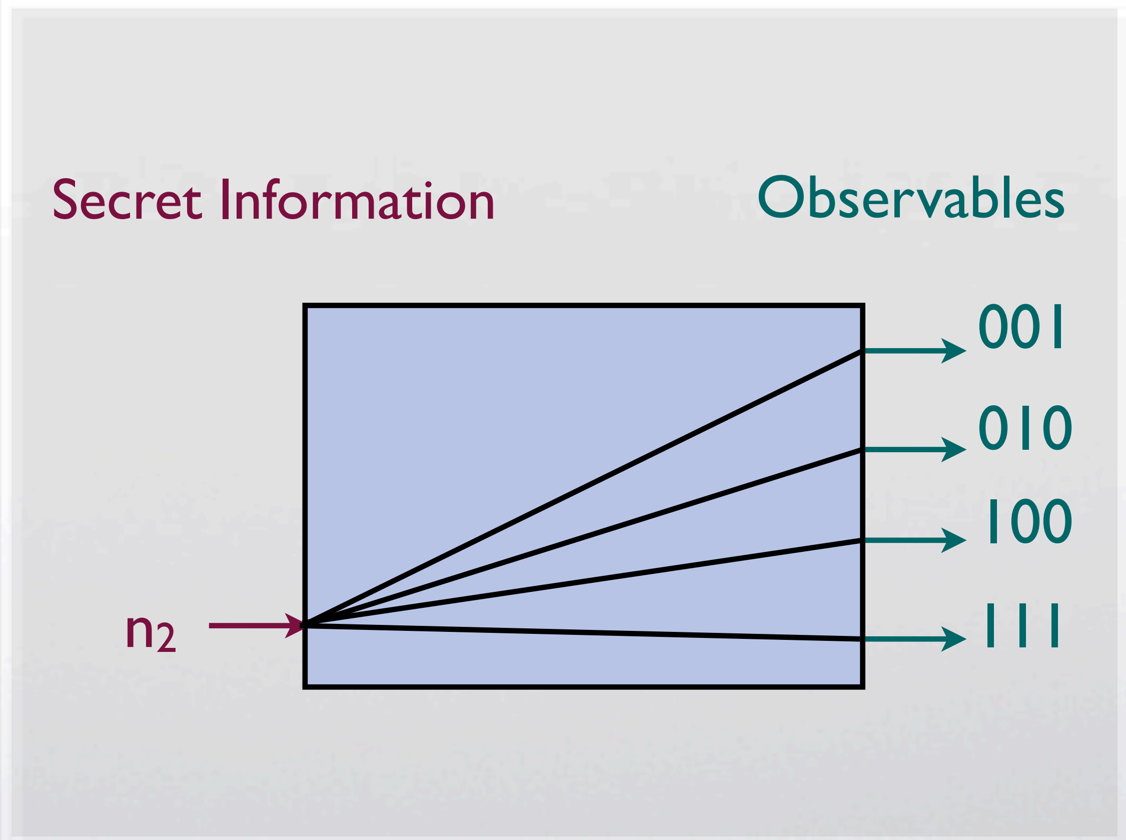
Example: DC nets (ring of 3 nodes, b=1)



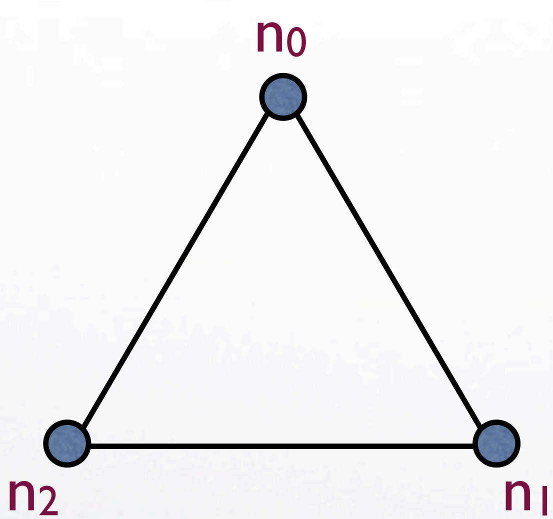


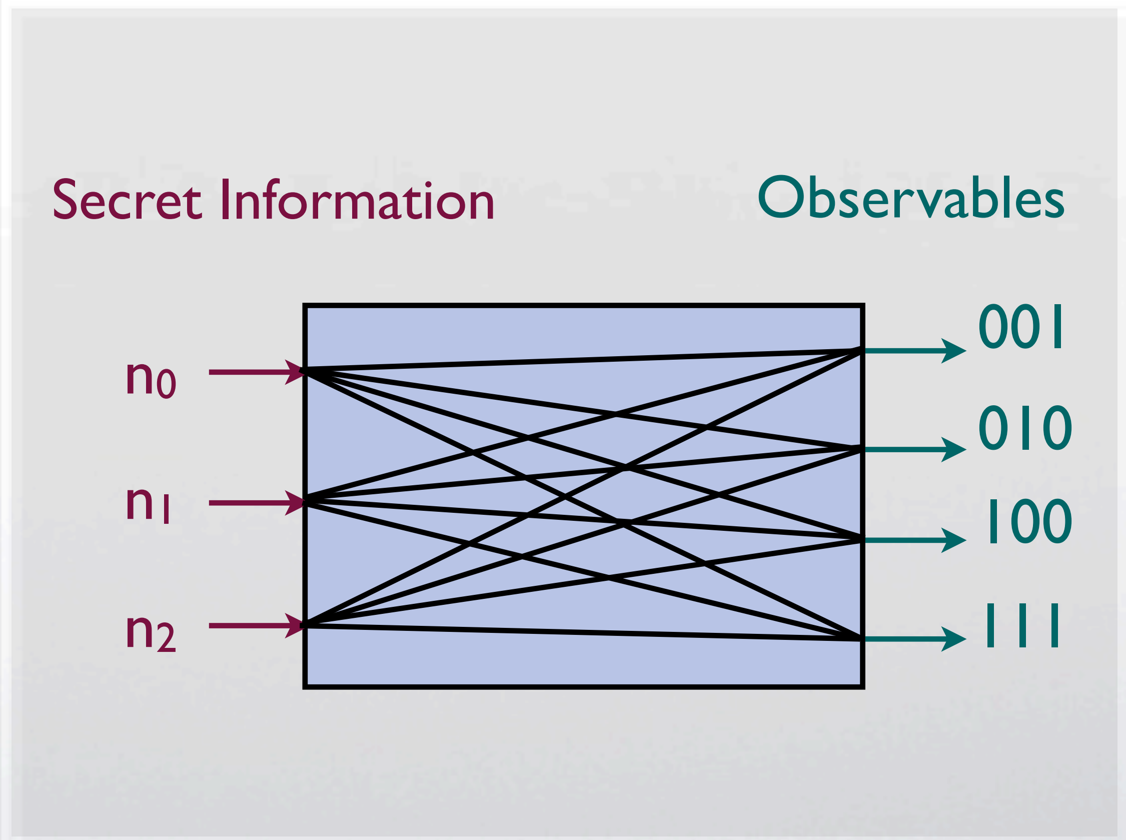
Example: DC nets (ring of 3 nodes, b=1)





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: $Pr(0) = Pr(1) = 1/2$

strong anonymity

biased coins: $Pr(0) = 2/3$, $Pr(1) = 1/3$

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

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

Observables: prior ⇒ posterior

Observables: 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 prob

p(o|n)
conditional prob

Observables: prior ⇒ posterior

p(n) prior prob
½
¼
¼

p(o|n) conditional prob

	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

p(n,o) joint prob

	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

Observables: prior ⇒ posterior

	p(o)	5/18	5/18	5/18	5/18	obs prob

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 prob					

p(o|n)
conditional prob

	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

p(n,o)
joint prob

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

Bayes theorem

p(o)	5/18	5/18	5/18	5/18

obs prob

p(n|001)		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

post prob

p(o|n)
conditional prob

	001	010	100	111
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

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

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:

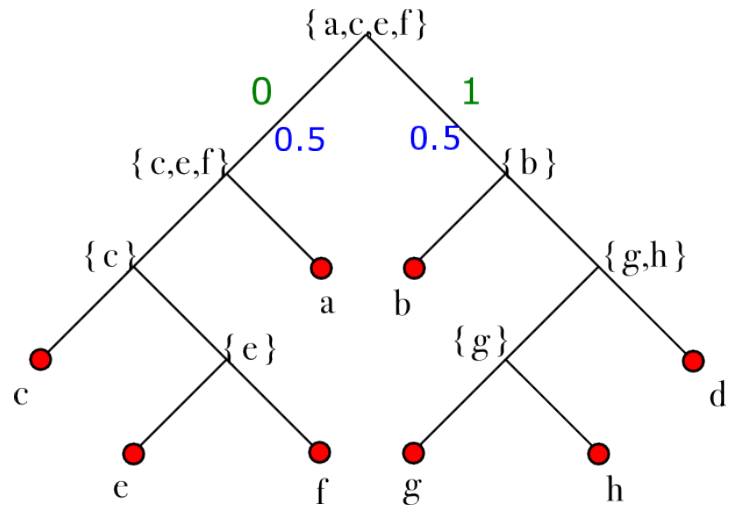
- The questions are of the form: "is S ∈ P ?"
- The measure of success is: the expected number of questions needed to find the value of S in the attacker's best strategy

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 split each time the search space in two subspaces with prob. masses as close as possible



Vulnerability

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

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 it cannot decrease

- 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)

Vulnerability: 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 attacker's best strategy

then the natural measure of vulnerability 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 attacker 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

Case 2: One-try attacks

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

The measure of success is 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).

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. $I_\infty(S;O) = H_\infty(S) - H_\infty(S|O)$

- In general $I_\infty(S;O) \geq 0$

- $I_\infty(S;O) = 0$ if all rows are the same (but not viceversa)

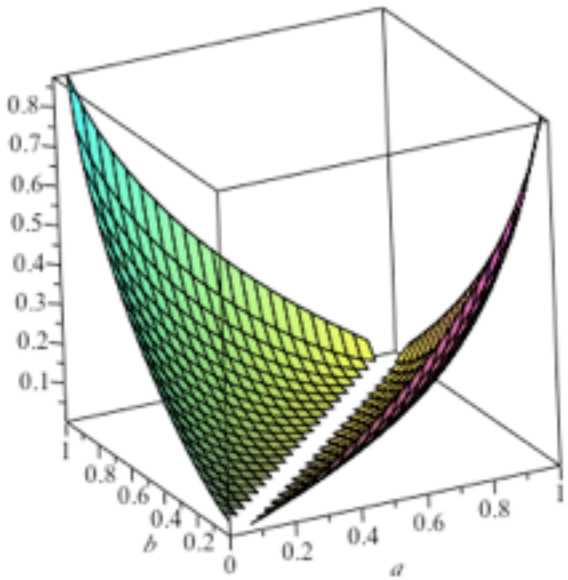
Define min-capacity: C_∞ = max $I_\infty(S;O)$ over all priors.

- $C_\infty = 0$ if and only if all rows are the same
- C_∞ is obtained on the uniform distribution (but not only)
- C_∞ = the sum of the max of each column
- $C_\infty \geq C$

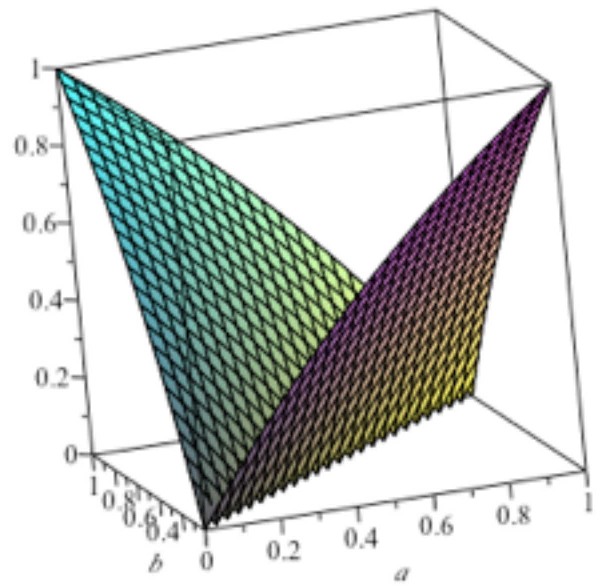
Shannon capacity vs. Rény min-capacity

binary channel

a	1−a
b	1−b



Shannon capacity



min-capacity

Differential Privacy

- Differential privacy is a notion of privacy originated in the area of **Statistical Databases.** Dwork et al. ICALP 2006, STOC 2006

- It has been a very successful line of research: Nowadays the concepts and methodologies of D.P. are investigated also in many other contexts: language-based security (Barthe and köpf, Pierce et. al.), social networks (Smatikov et al.), cloud computing, etc.

Statistical databases

Name/Id	age	weight	sex	disease	...
Mario Rossi	65	82	M	yes	...
Daniele Bianchi	35	120	M	yes	...
Lucia Verdi	40	45	F	no	...
...	...	...	...	...	...

Examples of queries which seem harmless

- How many people have the disease ?
- What is the average age and weight of men who have the disease ?

global

Examples of queries we want to forbid

- Does Daniele Bianchi have disease ?
- What is the name of the last record inserted in the database ?
- What are the age and weight of the last record inserted in the database ?

individual

The problem

Name/Id	age	weight	sex	disease	...
Mario Rossi	65	82	M	yes	...
Daniele Bianchi	35	120	M	yes	...
Lucia Verdi	40	45	F	no	...
...	...	...	...	...	...

- How many men have disease ? 2
- What are the average age and weight of men who have the disease ? 50 / 101

insertion of a new record

Name/Id	age	weight	sex	disease	...
Mario Rossi	65	82	M	yes	...
Daniele Bianchi	35	120	M	yes	...
Lucia Verdi	40	45	F	no	...
Sergio Neri	20	140	M	yes	...
...	...	...	...	...	...

- How many men have disease ? 3
- What are the average age and weight of men who have the disease ? 40 / 114

We can deduce the exact age / weight of the new record

Noisy answers

- A typical solution to the problem of privacy: **Introduce some noise.** Instead of the exact answer to the query $f : \mathcal{X} \rightarrow \mathcal{Y}$, the curator gives a randomized answer $\mathcal{K} : \mathcal{X} \rightarrow \mathcal{Z}$ ($\mathcal{Z}$ may be different from $\mathcal{Y}$)
- The principle: little noise in global info produces large noise in individual info
- A typical randomized method: **the Laplacian noise.** If the exact answer is y, the reported answer is z, with a probability density function defined as:

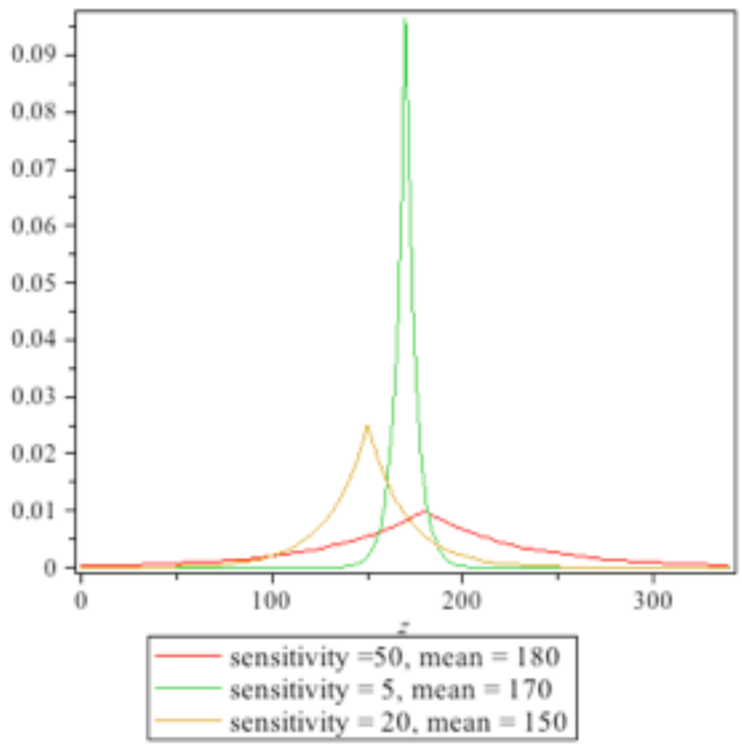
$$dP(z) = c\, e^{-\frac{|z-y|}{\Delta f}}$$

where Δf is the *sensitivity* of f:

$$\Delta f = \max_{x \sim x' \in \mathcal{X}} |f(x) - f(x')|$$

and c is a normalization factor:

$$c = \frac{1}{2\,\Delta f}$$



Privacy and Utility

- The two main criteria by which we judge a randomized mechanism:
  - **Privacy:** how good is the protection against leakage of private information
  - **Utility:** how useful is the reported answer

- Clearly there is a trade-off between privacy and utility, but they are not the exact opposites: privacy refers to the individual data, utility refers to the global (i.e. statistical) data.

Differential Privacy

- There have been various attempts to quantify the notion of privacy, but the most successful one is the notion of Differential Privacy, recently introduced by Dwork

- **Differential Privacy** [Dwork 2006]: a randomized function $\mathcal{K}$ provides ε-differential privacy if for all adjacent databases x, x', and for all $S \subseteq Z$, we have

$$Pr[\mathcal{K}(x) \in \mathcal{S}] \leq e^{\epsilon} \, Pr[\mathcal{K}(x') \in \mathcal{S}]$$

- The idea is that the likelihoods of x and x' are not too far apart, for every S

- For discrete answers:

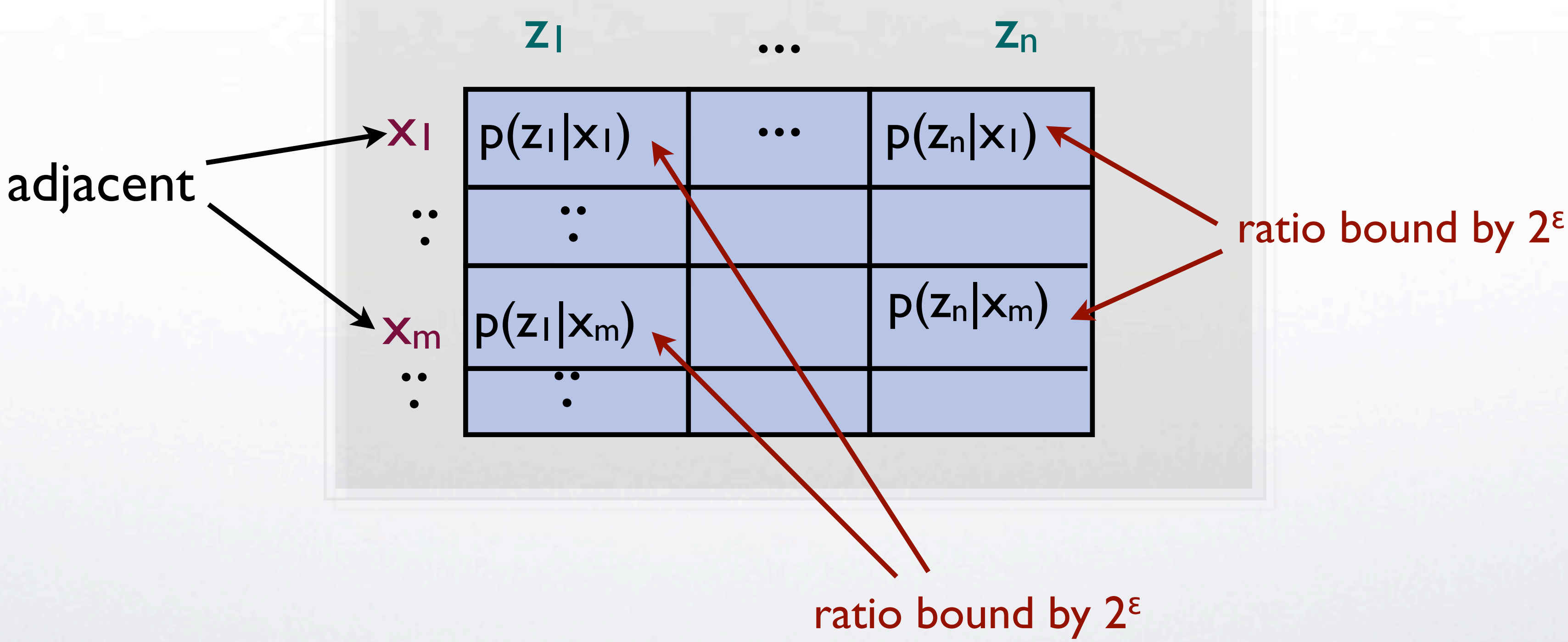
$$\frac{p(K = z | X = x)}{p(K = z | X = x')} \leq e^{\epsilon}$$

$\mathcal{K}$ can be seen as a noisy channel, in the information-theoretic sense from the domain $\mathcal{X}$ of databases to the domain $\mathcal{Z}$ of reported answers

Channel matrix

	z_1	...	z_n
x_1	$p(z_1 \mid x_1)$	...	$p(z_n \mid x_1)$
⋮	⋮		
x_m	$p(z_1 \mid x_m)$		$p(z_n \mid x_m)$
⋮	⋮		

Differential privacy on the channel matrix



Differential Privacy: alternative definition

- Perhaps the notion of differential privacy is easier to understand under the following equivalent characterization.
- In the following, X_i is the random variable representing the value of the individual i, and $X_{\neq i}$ is the random variable representing the value of all the other individuals in the database

- **Differential Privacy, alternative characterization:** a randomized function $\mathcal{K}$ provides ε-differential privacy if:

$$\text{for all } x \in \mathcal{X}, z \in \mathcal{Z}, p_i(\cdot)$$

$$\frac{1}{e^{\epsilon}} \leq \frac{p(X_i = x_i | X_{\neq i} = x_{\neq i})}{p(X_i = x_i | X_{\neq i} = x_{\neq i} \wedge K = z)} \leq e^{\epsilon}$$

Utility

The reported answer, i.e. the answer given by the randomized function, should allow to approximate the true (i.e. the exact) answer to some extent

Z = reported answer; *Y* = exact answer

Utility: $$\mathcal{U}(Y,Z) = \sum_{y,z} p(y,z)\; gain(y, remap(z))$$

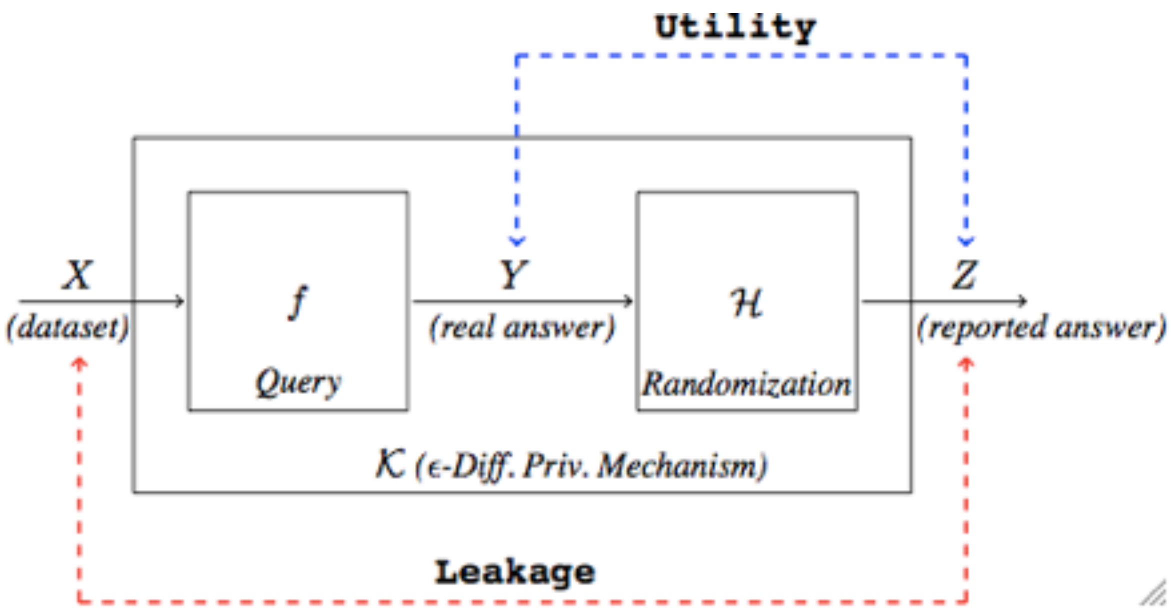
The remap allows the user to use side information (i.e. a priori pb) to maximize utility

Example: **binary gain function**: $$gain(y_1, y_2) = \begin{cases} 1 & y_1 = y_2 \\ 0 & y_1 \neq y_2 \end{cases}$$

In the binary case the utility is **the expected value of the probability of success** to obtain the true answer (i.e. the Bayes vulnerability)

Oblivious mechanisms

- Given $f: \mathcal{X} \rightarrow \mathcal{Y}$ and $\mathcal{K}: \mathcal{X} \rightarrow \mathcal{Z}$, we say that $\mathcal{K}$ is oblivious if it depends only on $\mathcal{Y}$ (not on $\mathcal{X}$)
- If $\mathcal{K}$ is oblivious, it can be seen as the composition of f and a randomized mechanism $\mathcal{H}$ defined on the exact answers $\mathcal{K} = f \times \mathcal{H}$



- Another reason why privacy and utility are not the exact opposite is that privacy concerns the information flow between the databases and the reported answers, while utility concerns the information flow between the correct answer and the reported answer

Differential Privacy and Utility

The fact that privacy and utility are not the exact opposite means that for the same utility we can have mechanisms with different degrees of utility

➪ **Important research direction: how to increase utility while preserving the intended degree of privacy**

Two fundamental results

1. [Ghosh et al., STOC 2009] The (truncated) geometric mechanism is **universally optimal** in the case of counting queries, with respect to all (reasonable) notions of utility

- Counting queries are of the form "how many individuals in the DB satisfy the property P ?"
- universally optimal means that it provides the best utility, for a fixed ε of differential privacy, for all the a priori distributions (side information)
- the geometric mechanism is the discrete version of the Laplacian

Two fundamental results

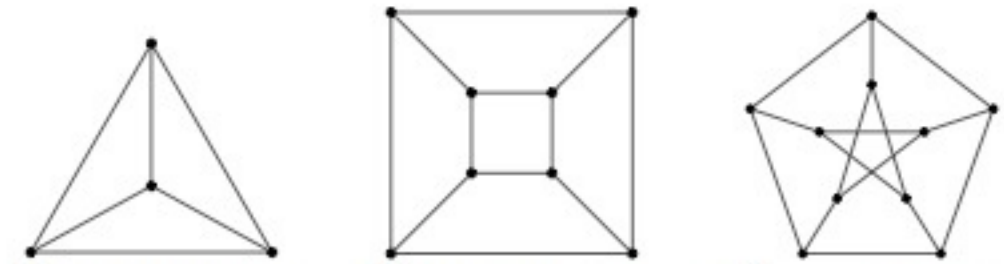
2. [Brenner and Nissim, STOC 2010] The counting queries are practically the only kind of queries for which there exists a universally optimal mechanism

   - This means that for other kind of queries one can only construct optimal mechanisms for specific a priori distributions (side information).
   - The precise characterization is given in terms of the graph structure that the adjacency relation induces on the answer space:
     - line: ok
     - loops: not ok
     - trees: not ok

Some contributions

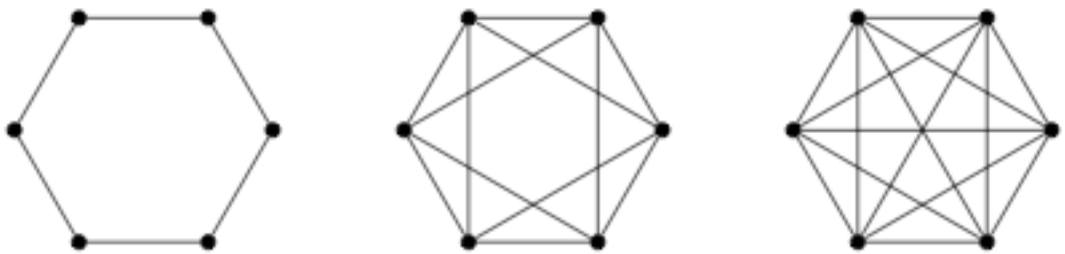
1. [Alvim et al, ICALP 2012]
   A randomized mechanism which is optimal for the uniform a priori distr., and for certain symmetry classes of graphs representing the relation induced by the adjacency relation

   1. Distance regular



(a) Tetrahedral graph (b) Cubical graph (c) Petersen graph

   2. Vertex transitive



(a) Cycle: degree 2. (b) Degree 4. (c) Clique: degree 5.

Some contributions

4. [Alvim et al., FAST 2012]
   Relation between differential privacy and quantitative information flow:

   For distance-regular and vertex-transitive graphs, differential privacy induces a bound on the min-entropy leakage. We have characterized a strict bound for every degree ε of D.P.

Some contributions

2. [El Salamouni et al., POST 2012]
   We have considered a limited notion of universal optimality: namely, optimality w.r.t. a subset of all the possible a priori distributions (side information).

   We have given sufficient conditions for the existence of a *limited* universally optimal mechanism, and characterized the subset of allowed side information

   Two main restrictions: so far we have considered only binary gain functions and directed methods (i.e. w/o remapping). We are currently working at lifting these conditions.

Thank you !