Quantitative Approaches to Information Protection

Catuscia Palamidessi
INRIA & Ecole Polytechnique of Paris

Leakage of information (13 March 2014)

BBC NEWS TECHNOLOGY

13 March 2014 Last updated at 21:23 ET

Mark Zuckerberg 'confused and frustrated' by US spying

Mr Zuckerberg said that the internet needed to be made more secure for users

Facebook founder Mark Zuckerberg has said he has called President Barack Obama to "express frustration" over US digital surveillance.

The 29-year-old said in a blog post the US government "should be the champion for the internet, not a threat".

Related Stories

'Spying setting fire to internet'

Trust in the internet

theguardian

News | US | World | Sports | Comment | Culture | Business | Money

News › Society › NHS

NHS England patient data 'uploaded to Google servers', Tory MP says

Health select committee member Sarah Wollaston queries how data was secured by PA Consulting and uploaded to servers outside UK

Police will have 'backdoor' access to health records

Bits

MARCH 13, 2014, 7:45 AM | Comment

Daily Report: Europe Moves to Reform Rules Protecting Privacy

By THE NEW YORK TIMES

The European Parliament passed a strong new set of data protection measures on Wednesday prompted in part by the disclosure by Edward J. Snowden, a former contractor at the United States National Security Agency, of America's vast electronic spying program, David Jolly reports.

Target says it declined to act on early alert of cyber breach

BY JIM FINKLE AND SUSAN HEAVEY

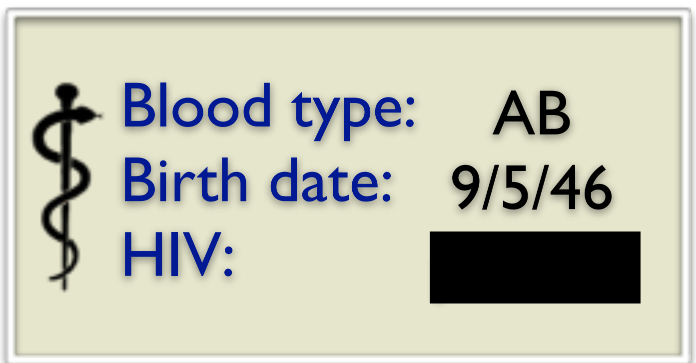
BOSTON/WASHINGTON Thu Mar 13, 2014 6:39pm EDT

Merchandise baskets are lined up outside a Target department store in Palm Coast, Florida, December 9, 2013.

CREDIT: REUTERS/LARRY DOWNING

Protection of sensitive information

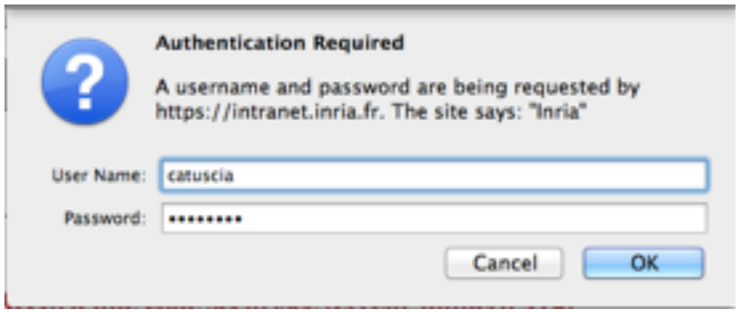
- Protecting the confidentiality of sensitive information is a fundamental issue in computer security



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

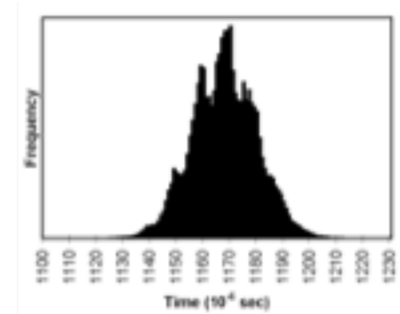
Leakage through correlated observables

Password checking



Election tabulation

Timings of decryptions



Plan of the course

1. Information leakage: motivation for quantitative approaches. Information-theoretic view. Notions of entropy and operational interpretations.

2. Focus on Shannon leakage and min-entropy leakage.

3. G-leakage. Lattice of information. Data processing order.

4. Privacy and aggregate data. Differential privacy. Trade-off between privacy and utility.

5. Location Privacy and geo-indistinguishability

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)

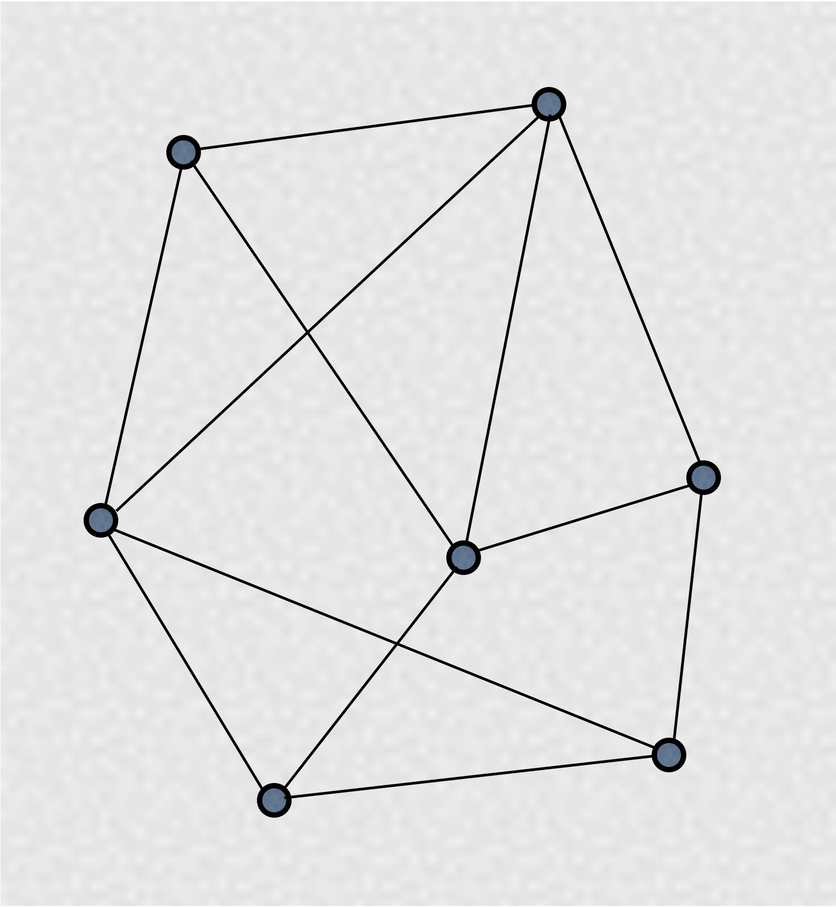
Side channel attack

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

Example 2

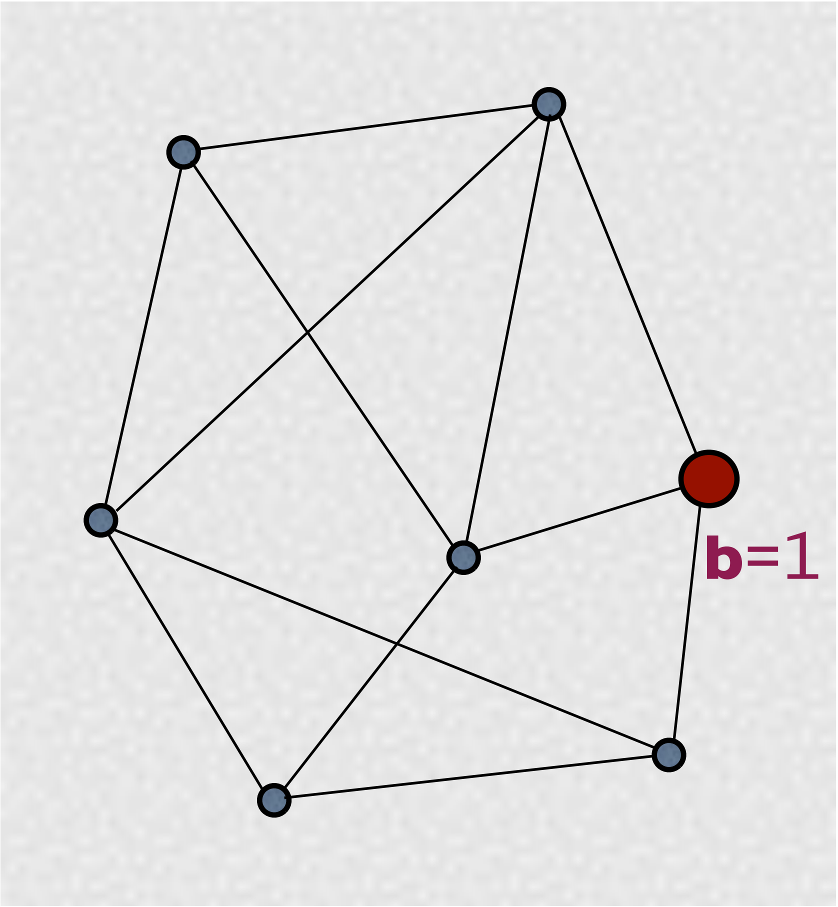
Example of Anonymity Protocol: DC Nets [Chaum'88]



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

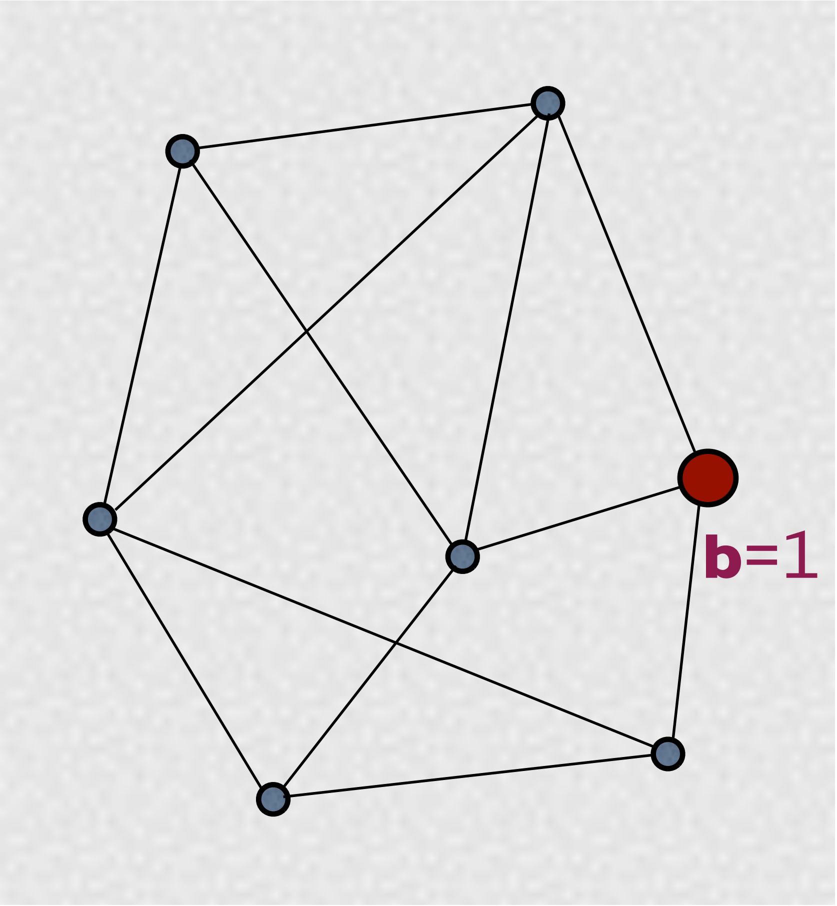
Example of Anonymity Protocol: DC Nets [Chaum'88]

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



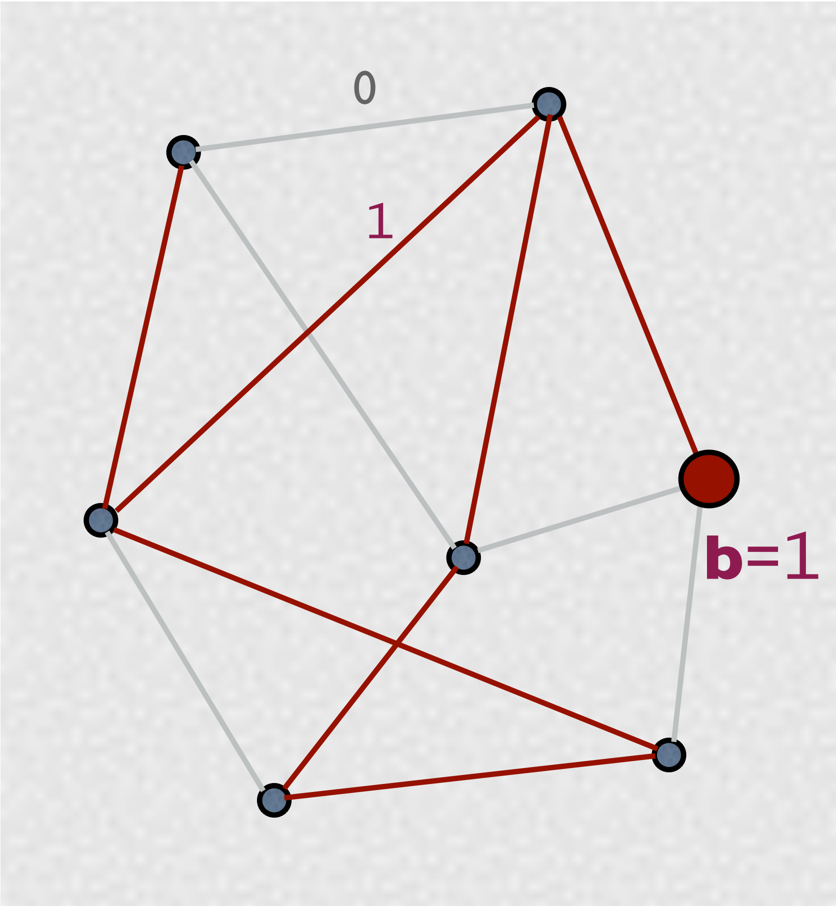
Chaum's solution

- Associate to each edge a fair binary coin



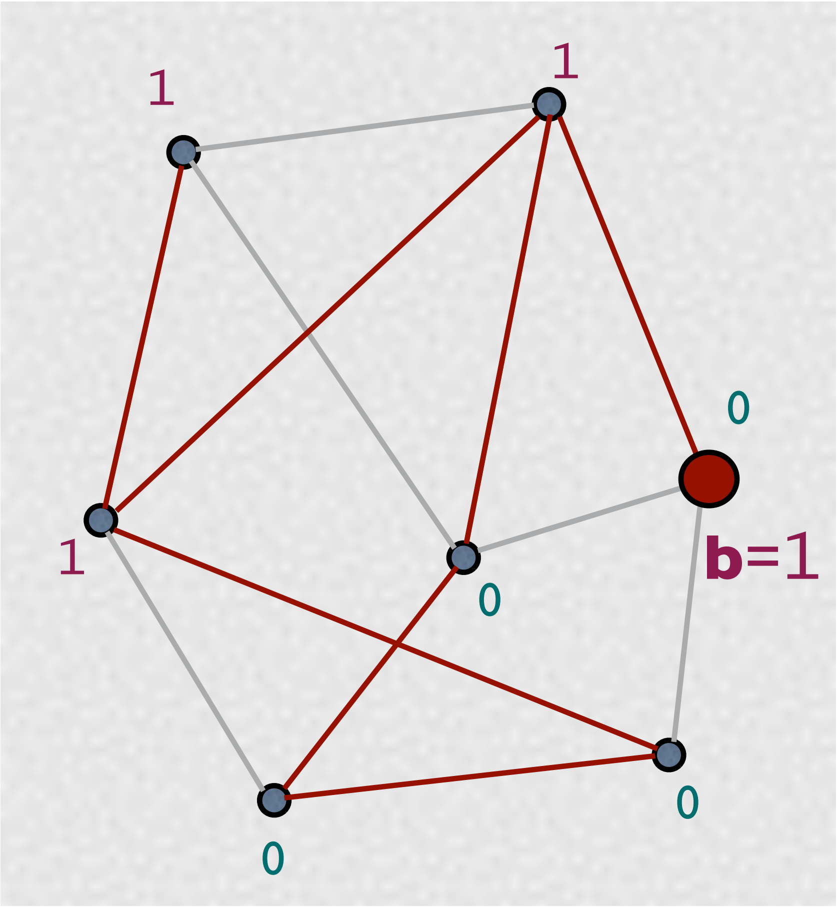
Chaum's solution

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



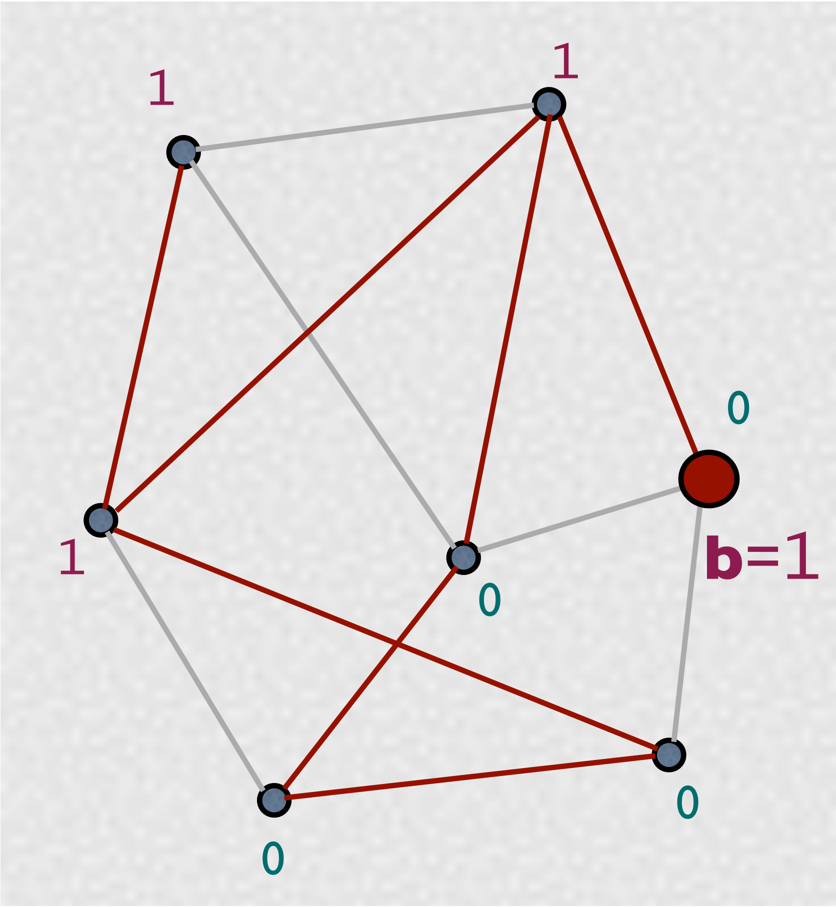
Chaum's solution

- 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

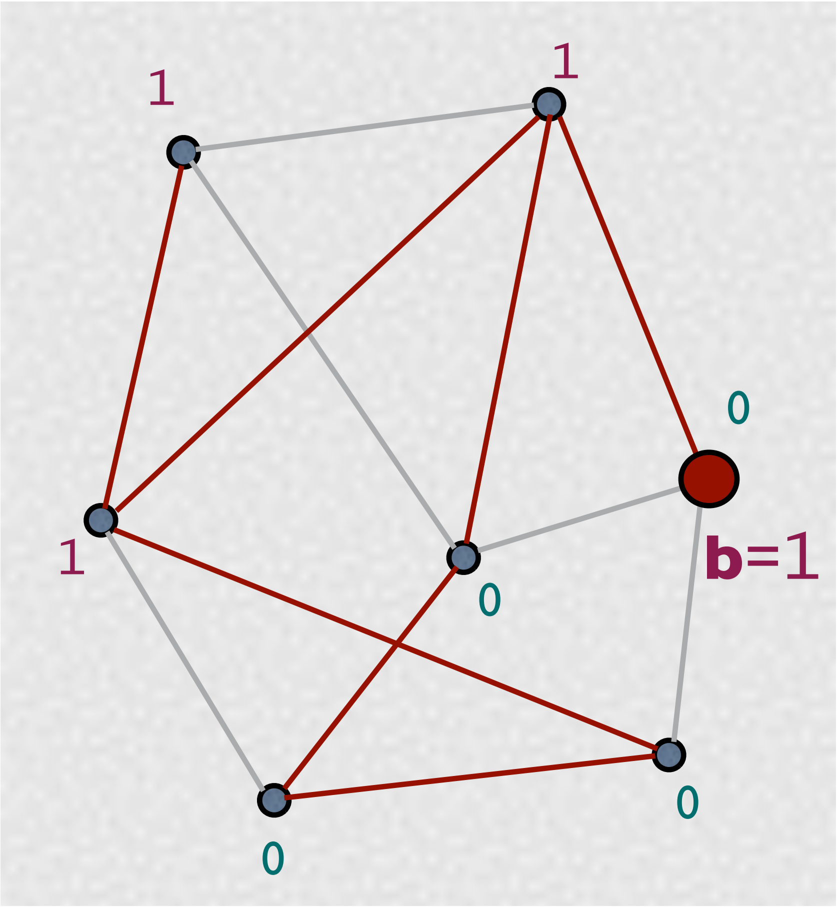
Question: Does the protocol protects the anonymity of the source?

Strong anonymity (Chaum)

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

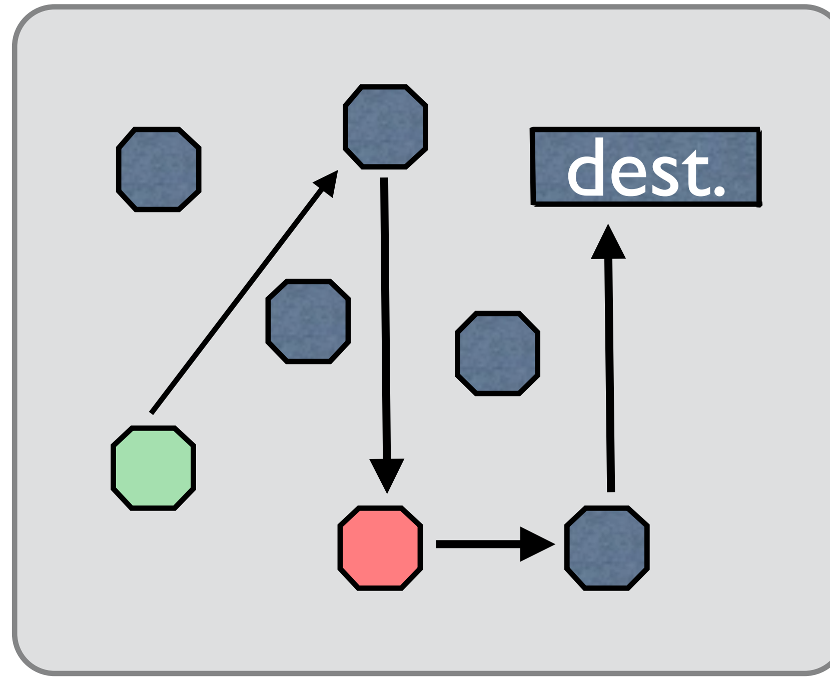
> the *a posteriori* probability that a certain node is the source is equal to its *a priori* probability

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



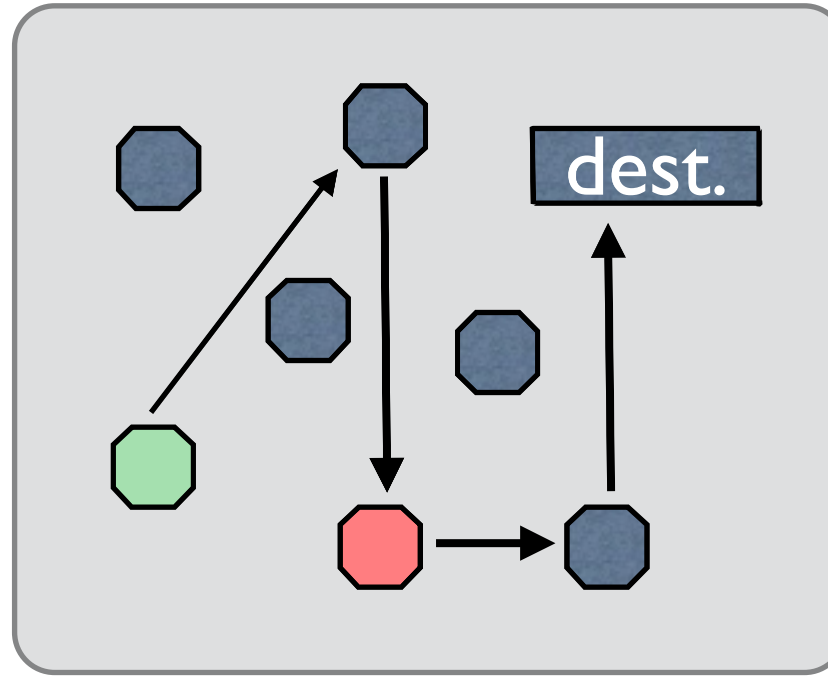
Example 3: Crowds [Rubin and Reiter'98]

- 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



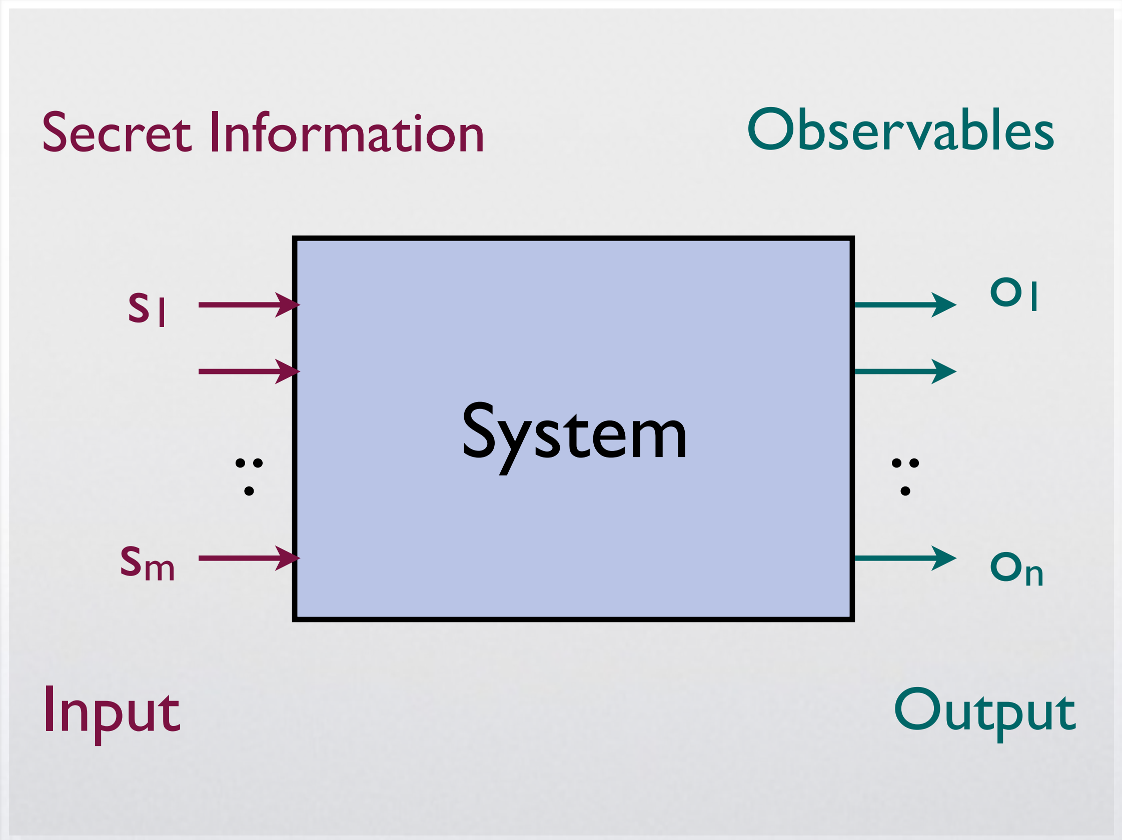
Probable innocence: under certain conditions, an attacker who intercepts the message from x cannot attribute more than 0.5 probability to x to be the initiator

Common features

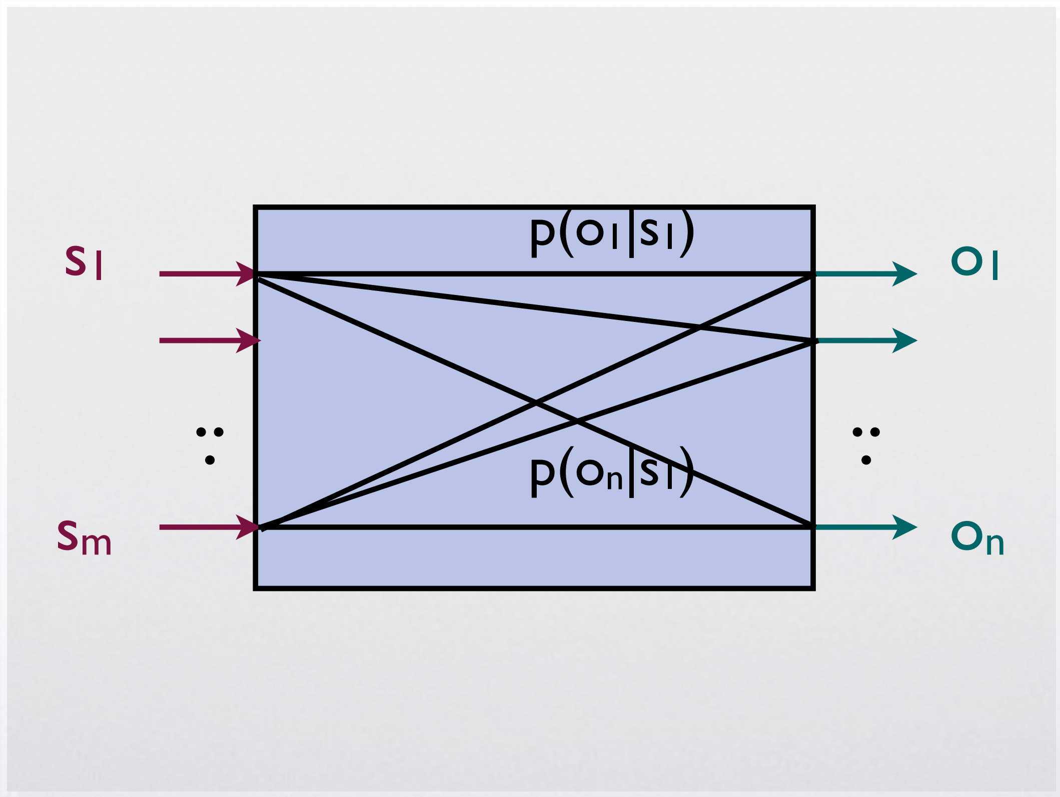
- Secret information
  - Password checker: The password
  - DC: the identity of the source
  - Crowds: the identity of the initiator
- Public information (Observables)
  - Password checker: The result (OK / Fail) and the execution time
  - DC: the declarations of the nodes
  - Crowds: the identity of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

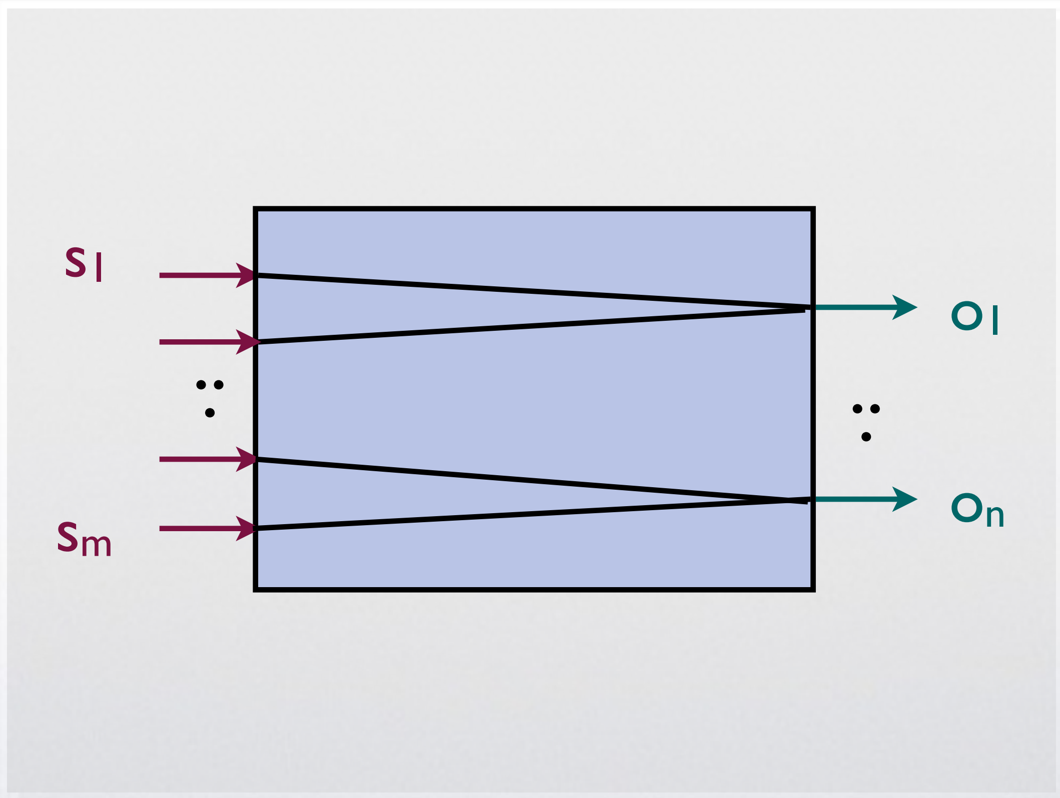
In a information-theoretic channel these conditional probabilities are independent from the input distribution

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

Particular case: **Deterministic systems**

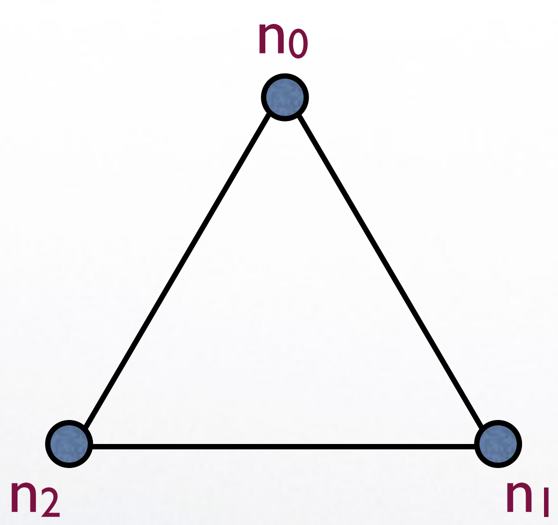
In these systems an input generates only one output

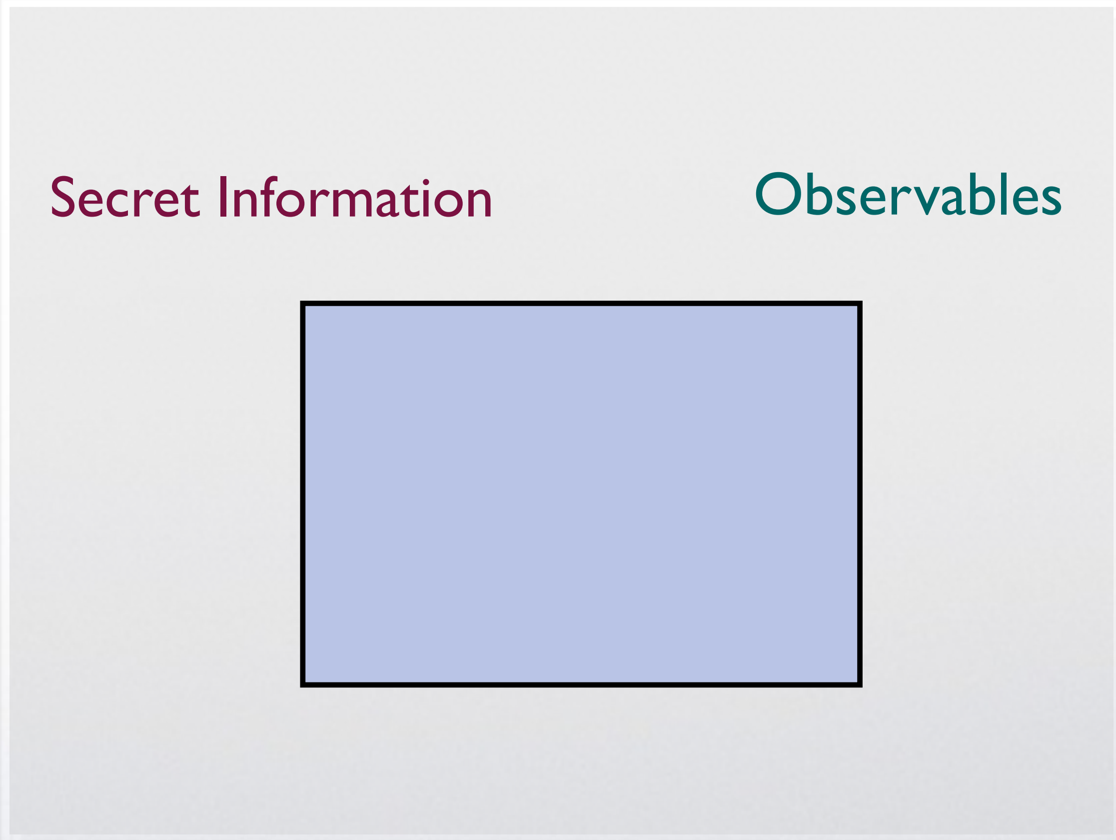
Still interesting: the problem is how to retrieve the input from the output



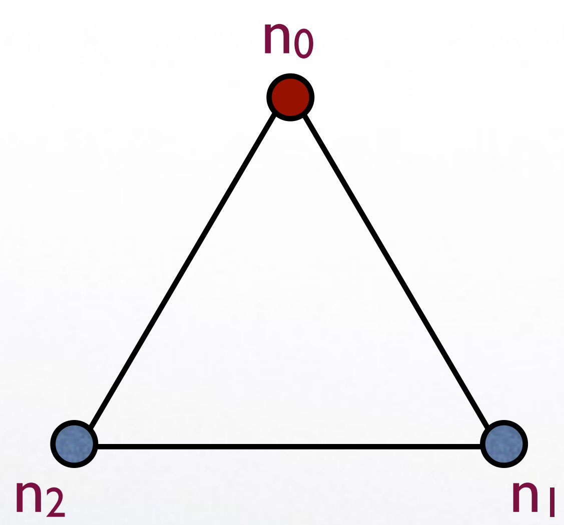
The entries of the channel matrix can be only 0 or 1

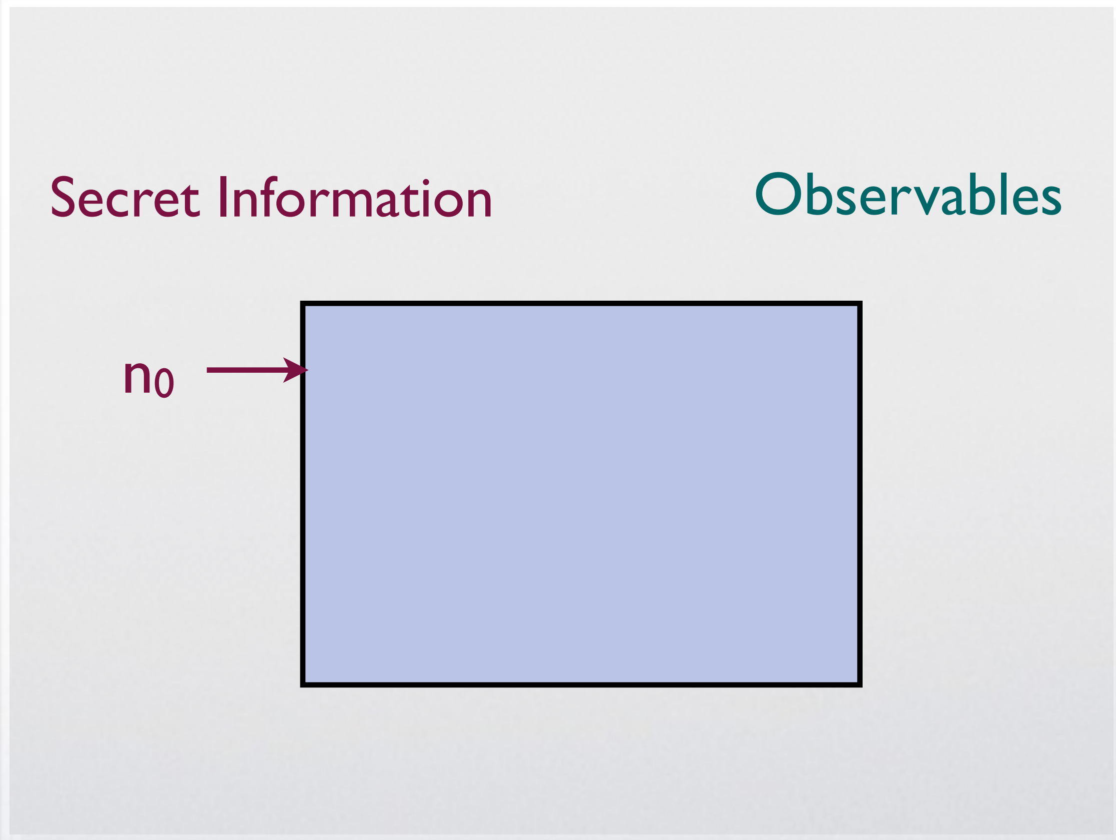
Example: DC nets (ring of 3 nodes, b=1)



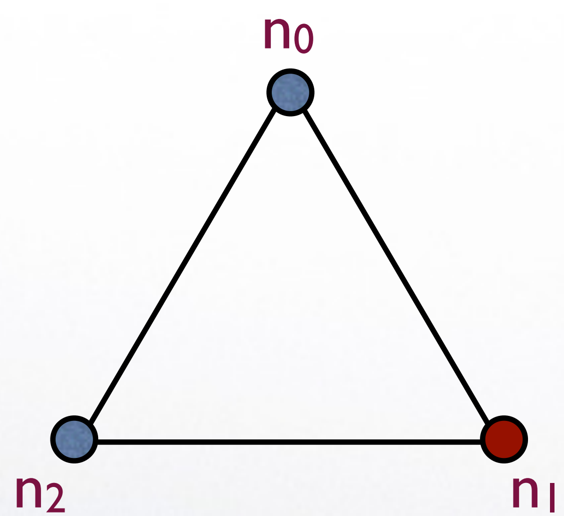


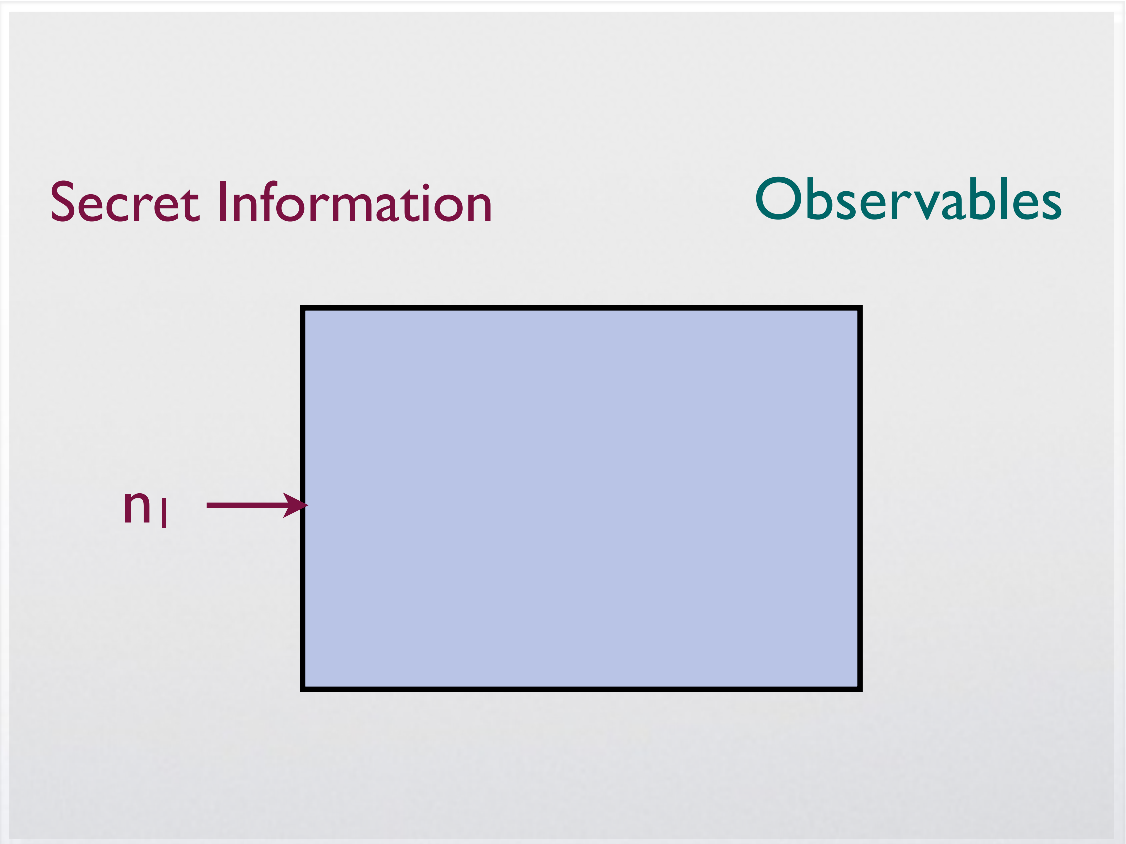
Example: DC nets (ring of 3 nodes, b=1)



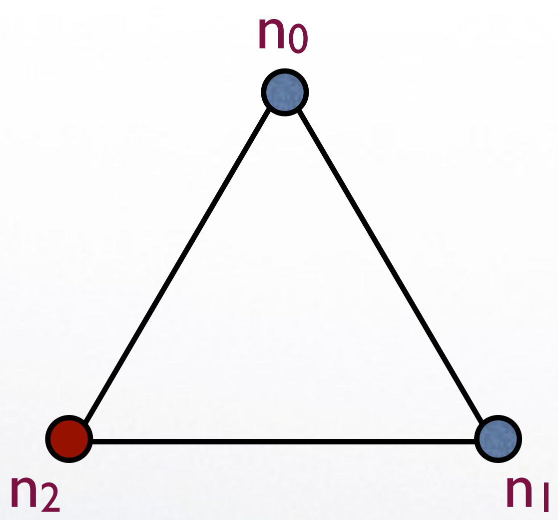


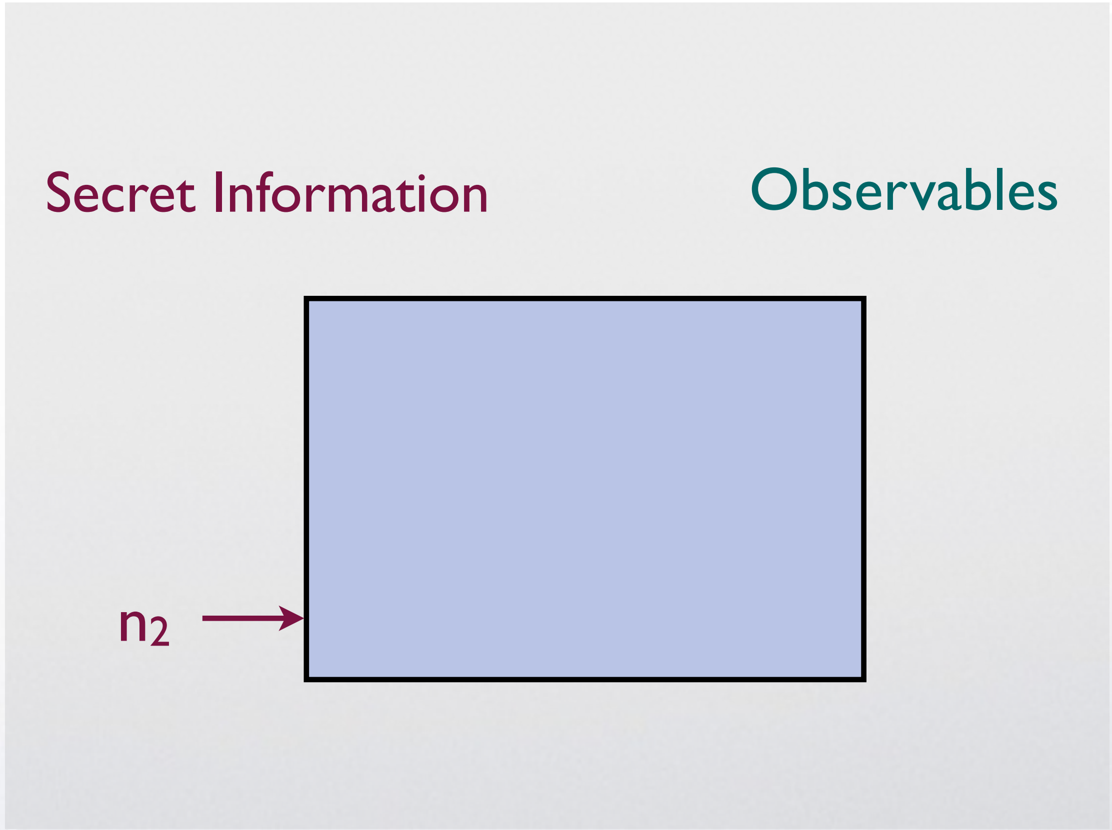
Example: DC nets (ring of 3 nodes, b=1)



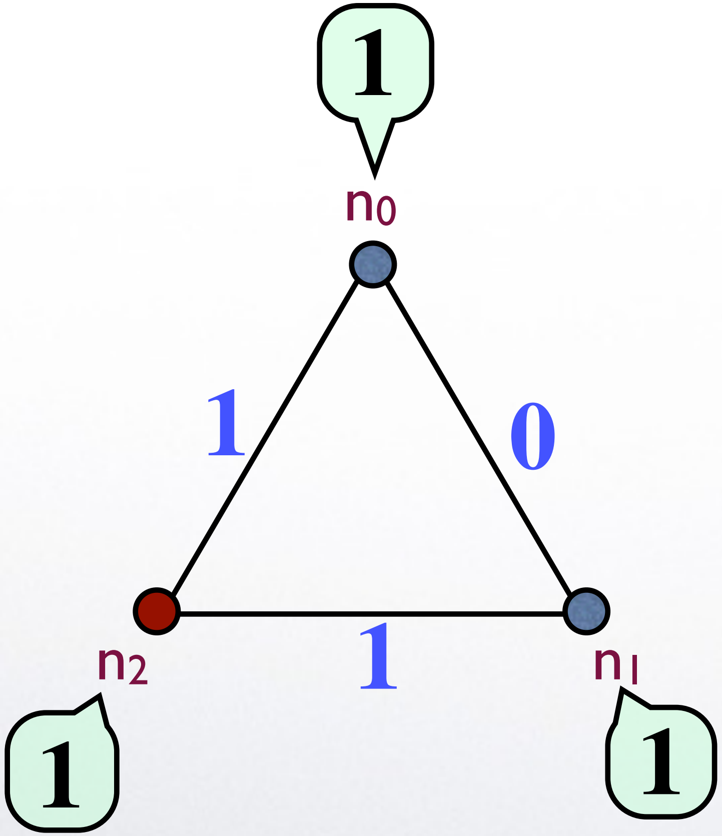


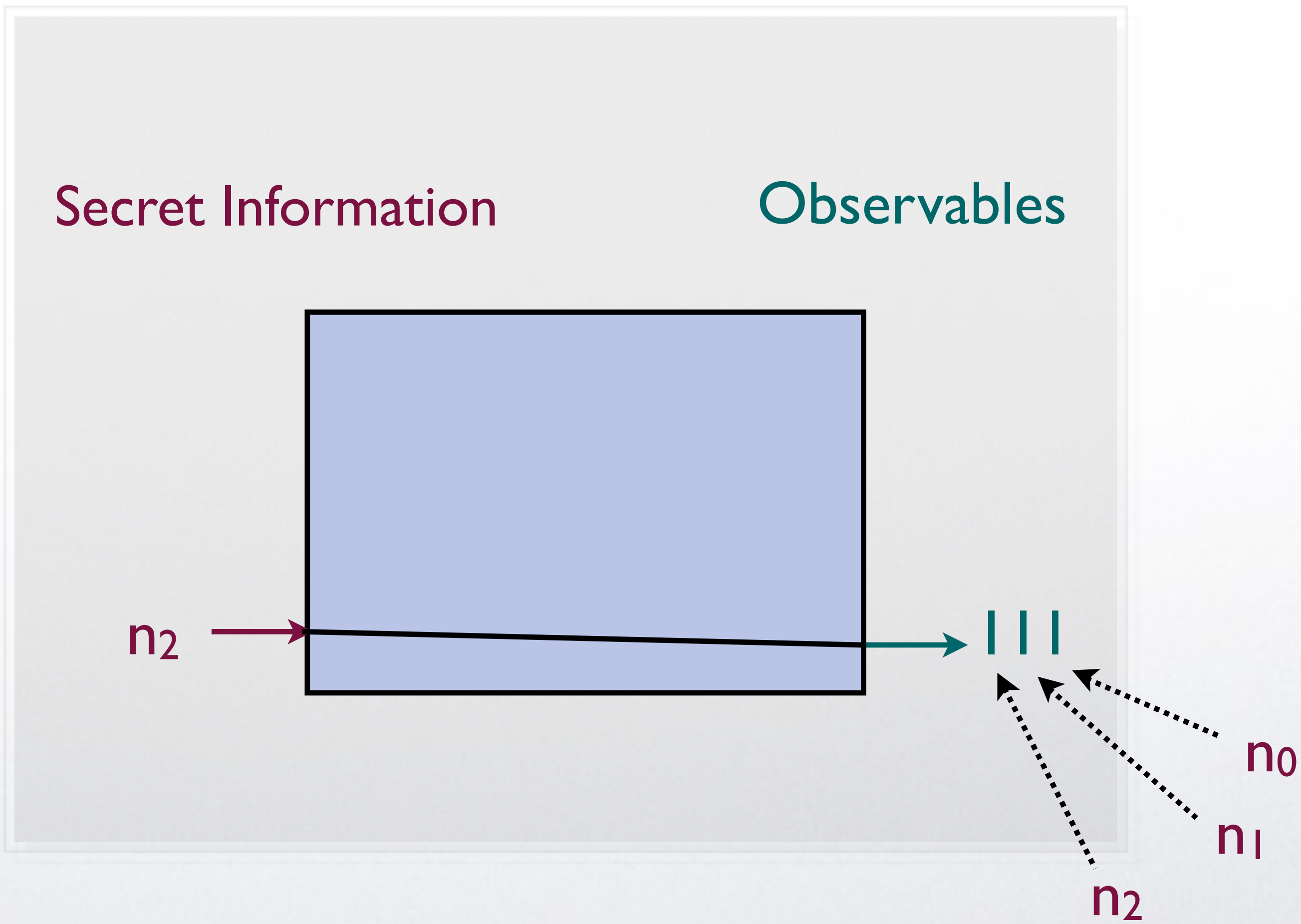
Example: DC nets (ring of 3 nodes, b=1)



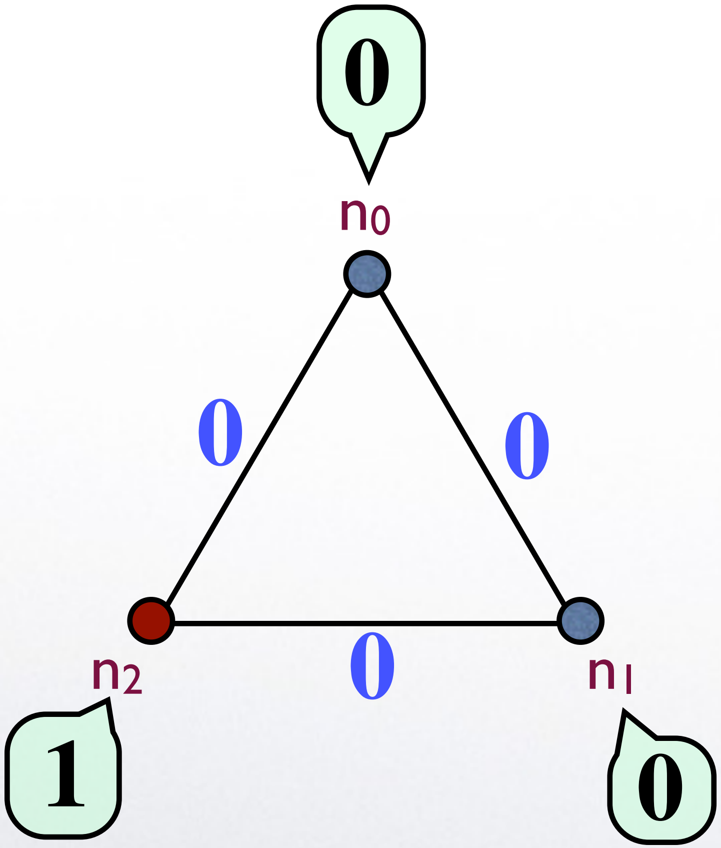


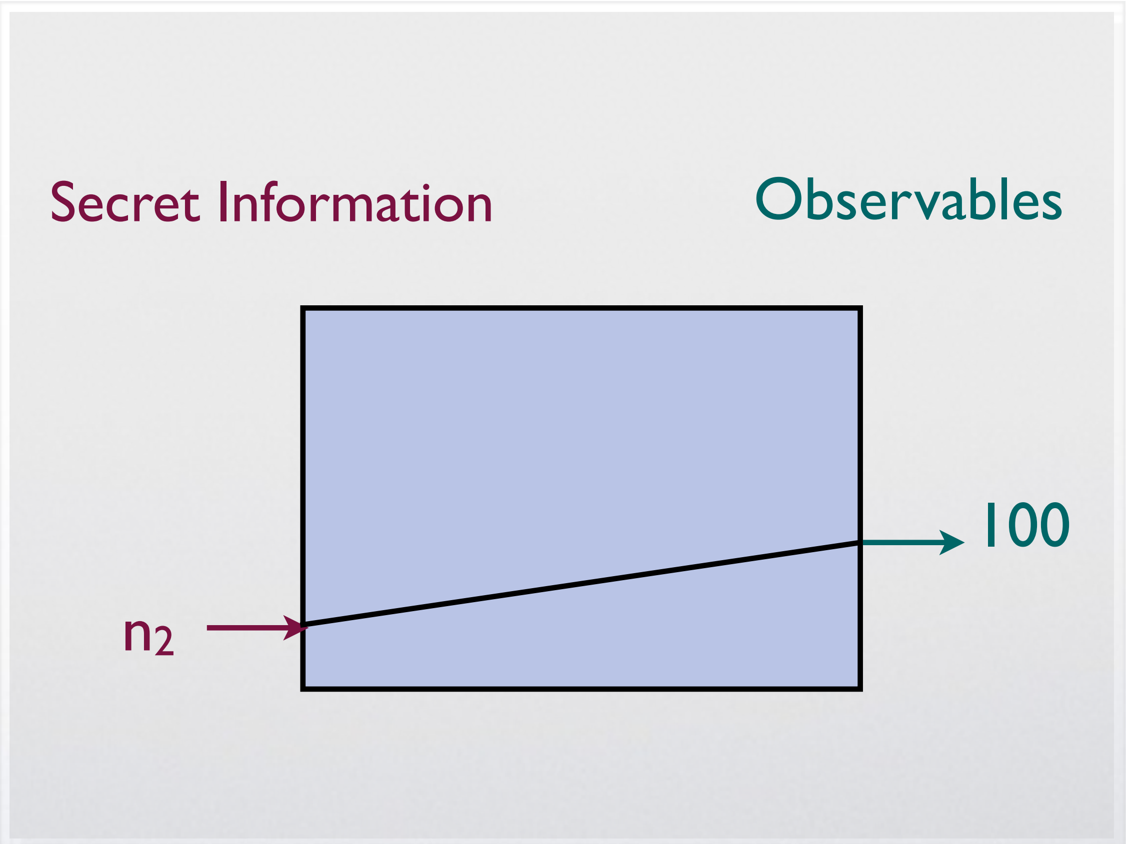
Example: DC nets (ring of 3 nodes, b=1)



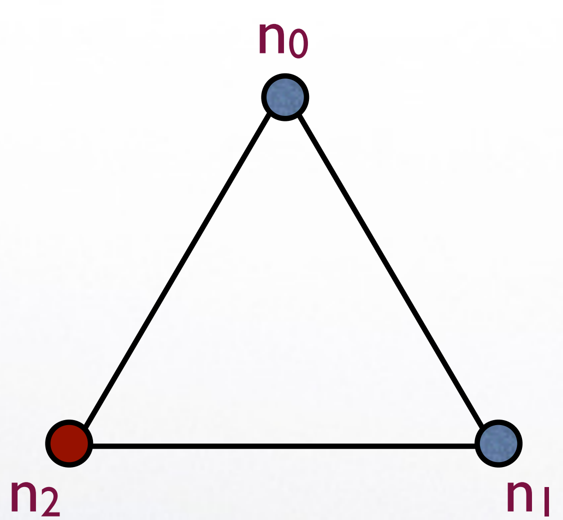


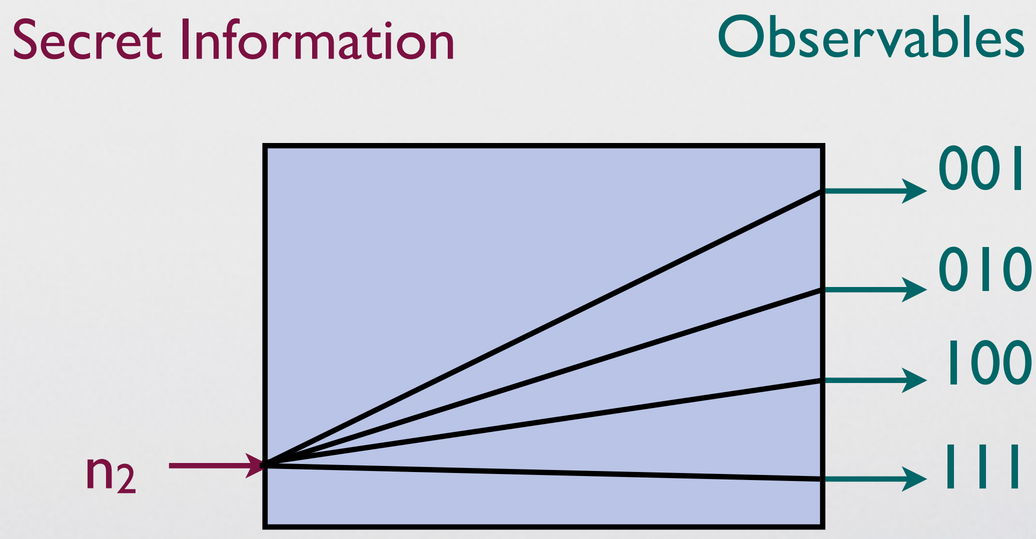
Example: DC nets (ring of 3 nodes, b=1)



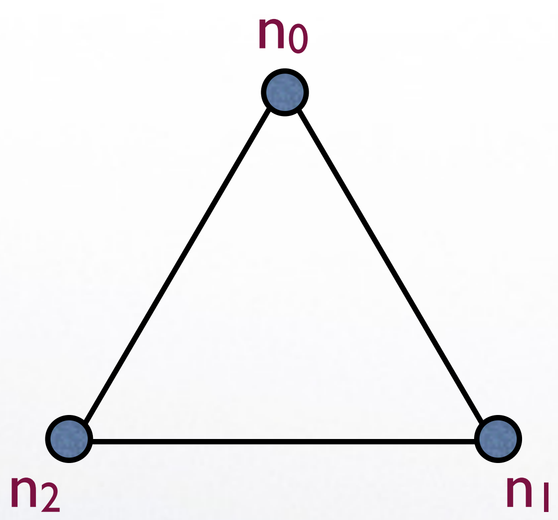


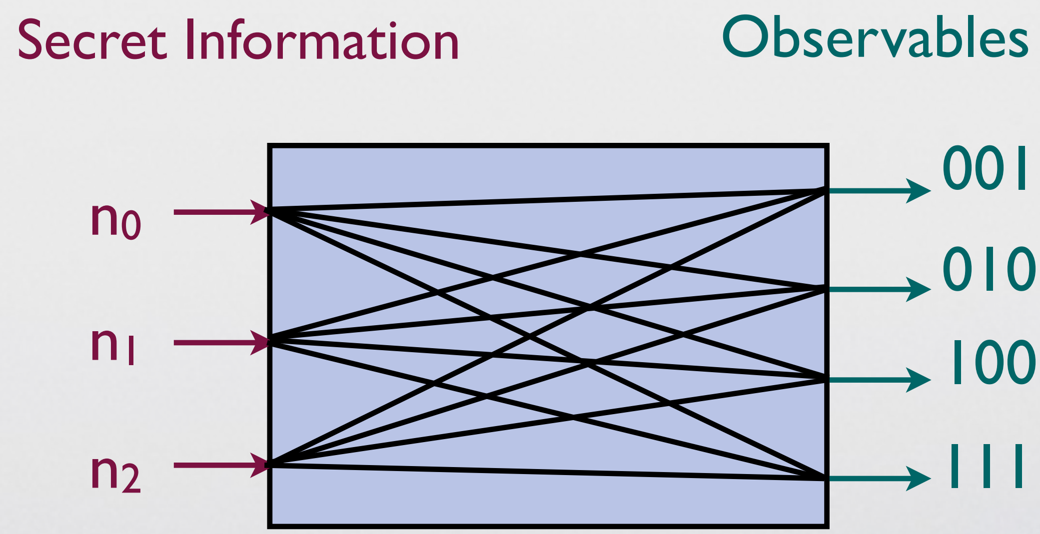
Example: DC nets (ring of 3 nodes, b=1)





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	s_l ¼
n_1	¼	¼	¼	¼ ⋮
n_2	¼	¼	¼	s_m ¼

fair coins: $\Pr(0) = \Pr(1) = ½$

strong anonymity

	001	010	100	111
n_0	⅓	2/9	2/9	s_l 2/9
n_1	2/9	⅓	2/9	2/9 ⋮
n_2	2/9	2/9	⅓	s_m 2/9

biased coins: $\Pr(0) = ⅔$, $\Pr(1) = ⅓$

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 secret prob

p(o|n)
conditional prob

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 secret prob

$p(o|n)$
conditional prob

	001	010	100	111
n_0	⅙	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

Observables: prior ⇒ posterior

	p(o)	5⁄18	¼	¼	2⁄9	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 secret prob

p(o|n)
conditional prob

	001	010	100	111
n_0	⅙	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

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

p(o)	5/18	1/4	1/4	2/9	obs prob

p(n|001)

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 secret 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

Password-checker 1

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

   end if
end for
```

Let us construct the channel matrix

Note: The string $x_1 x_2 x_3$ typed by the user is a parameter, and $K_1 K_2 K_3$ is the channel input

The standard view is that the input represents the secret. Hence we should take $K_1 K_2 K_3$ as the channel input

Password-checker 1

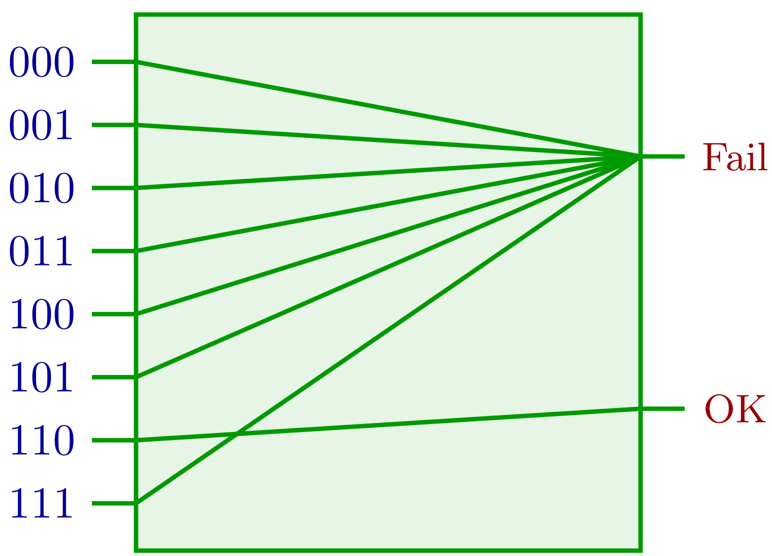
```
out := OK
for i = 1, ..., N do
    if x_i ≠ K_i then
        out := FAIL

    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: $out \in \{\mathsf{OK}, \mathsf{FAIL}\}$



	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

Different values of $x_1 x_2 x_3$ give different channel matrices, but they all have this kind of shape (seven inputs map to Fail, one maps to OK)

Password-checker 2

out := OK
for $i = 1, \ldots, N$ **do**
  if $x_i \neq K_i$ **then**
    $\left\{ \begin{array}{l} out := \mathsf{FAIL} \\ \text{exit}() \end{array} \right\}$
  end if
end for

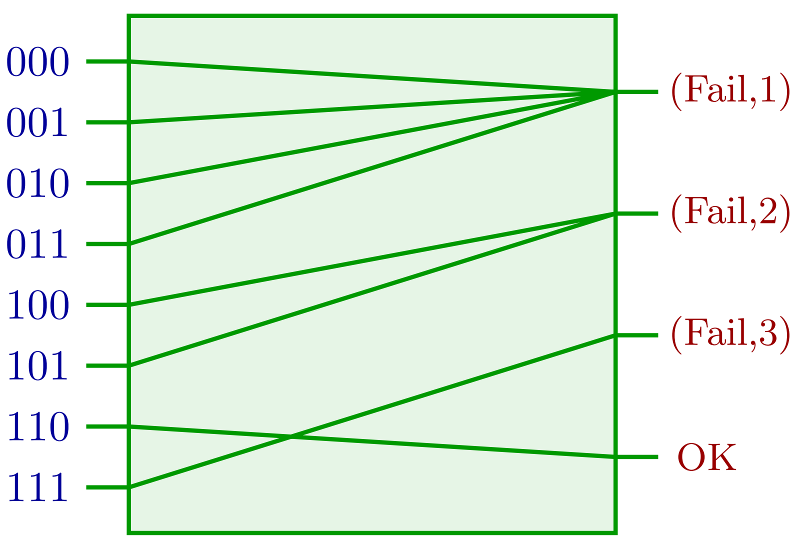
Assume the user string is $x_1 x_2 x_3 = 110$

Assume the adversary can measure the execution time

Let us construct the channel matrix

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

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

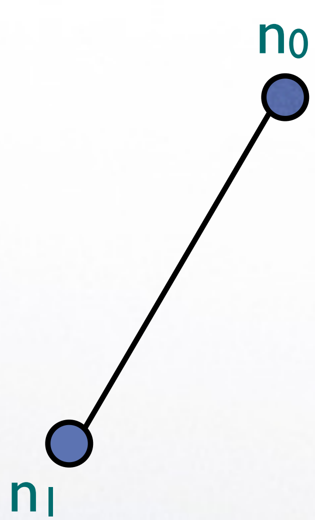


	(Fail, 1)	(Fail, 2)	(Fail, 3)	OK
000	1	0	0	0
001	1	0	0	0
010	1	0	0	0
011	1	0	0	0
100	0	1	0	0
101	0	1	0	0
110	0	0	0	1
111	0	0	1	0

Exercise 1

- Assuming that the possible passwords have uniform prior distribution, compute the matrix of the joint probabilities, and the posterior probabilities, for the two password-checker programs

Example: DC nets. Ring of 2 nodes, and assume $b = 1$



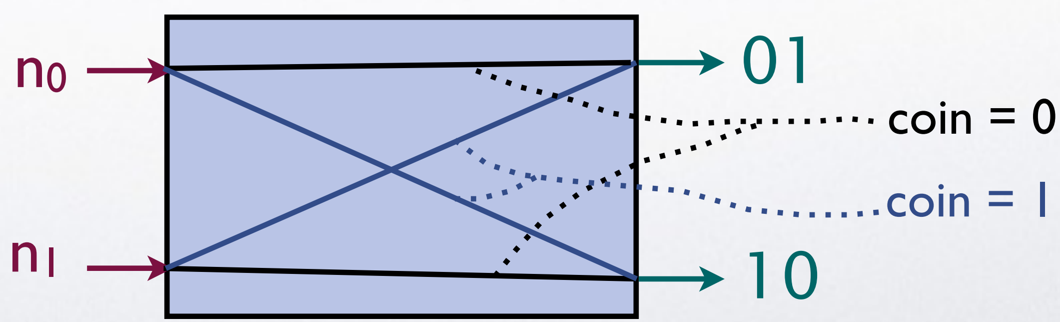
Let us construct the channel matrix

Input: n_0 , n_1

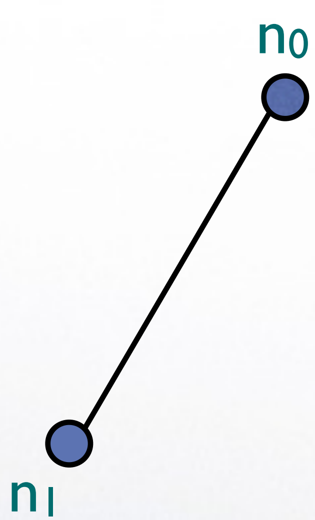
Output: the declarations of n_1 and n_0: $d_1 d_0 \in \{01, 10\}$

Secret

Observable



Example: DC nets. Ring of 2 nodes, and assume b = 1



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	½	½
n	½	½

Biased coin: $p(0) = ⅔$ $p(1) = ⅓$

	01	10
n	⅔	⅓
n	⅓	⅔

Exercise 2

- Assuming that n_0 and n_1 have uniform prior distribution, compute the matrix of the joint probabilities, and the posterior probabilities, in the two cases of fair coins, and of biased coins

- Same exercise, but now assume that the prior distribution is 2/3 for n_0 and 1/3 for n_1

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.