MPRI 2.3.2 - Foundations of privacy

Lecture 1

Kostas Chatzikokolakis

Dec 9, 2015

Plan of the course

Overview of applications

Quantitative Information Flow

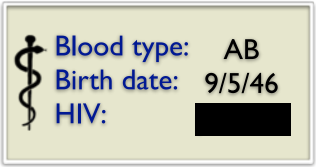
- Vulnerability and entropy - operational interpretation
- Information-theoretic approaches
- Relation with differential privacy
- Decision-theoretic approaches: g-leakage
- Comparing systems, the lattice of information

Location privacy

- Optimal Bayesian approaches
- Geo-indistinguishability

Protection of sensitive information

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



Blood type: AB

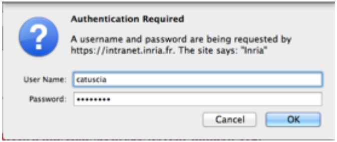
Birth date: 9/5/46

HIV: [redacted]

- 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

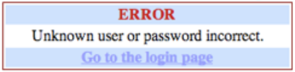


Authentication Required

A username and password are being requested by https://intranet.inria.fr. The site says: "Inria"

User Name: catuscia

Password: ••••••••



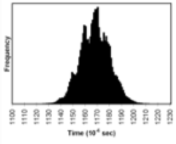
ERROR

Unknown user or password incorrect.

Go to the login page

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**

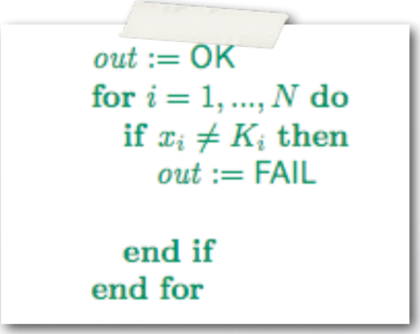
Example 1

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)

Intrinsic leakage

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



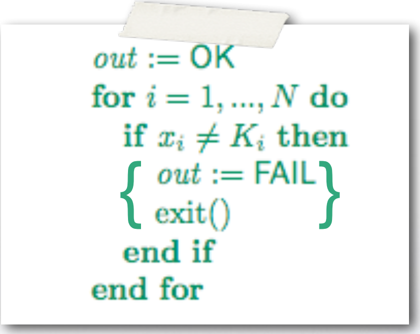
```
$out$ := OK
for $i = 1, ..., N$ do
   if $x_i \neq K_i$ then
      $out$ := FAIL

   end if
end for
```

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)



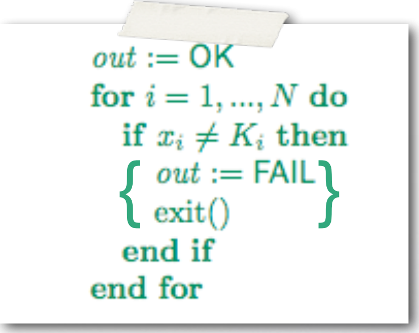
```
$out$ := OK
for $i = 1, ..., N$ do
    if $x_i \neq K_i$ then
        { $out$ := FAIL
          exit()          }
    end if
end for
```

More efficient, but what about security?

Example 1

Password checker 2

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



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

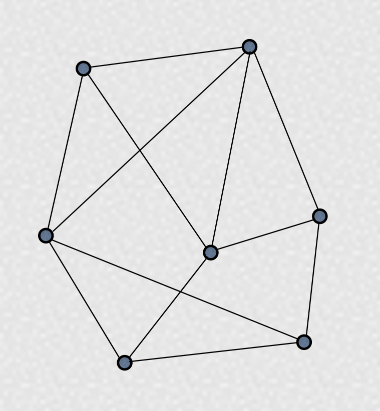
Side channel attack

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

Example 2

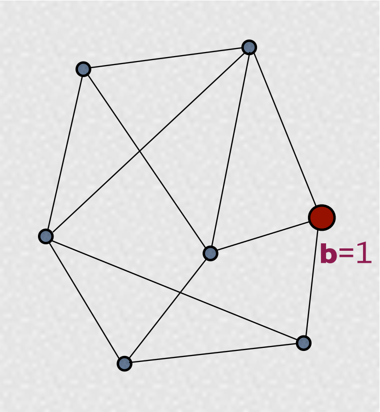
Example of Anonymity Protocol: DC Nets [Chaum'88]

- 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



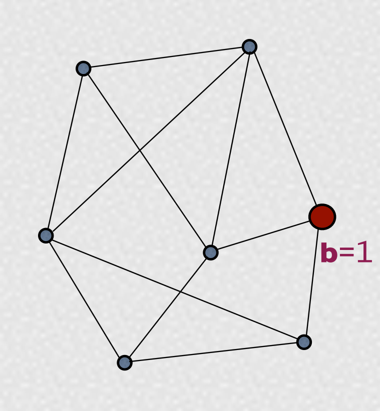
Example of Anonymity Protocol:
DC Nets [Chaum'88]

- 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

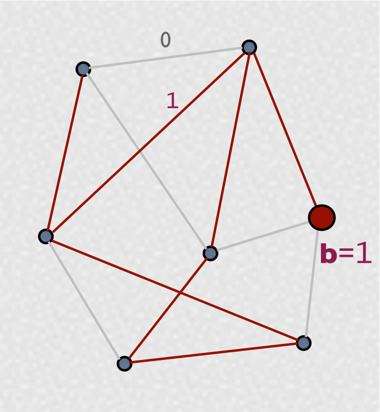
- Associate to each edge a fair binary coin



Chaum’s solution

- Associate to each edge a fair binary coin

- Toss the coins

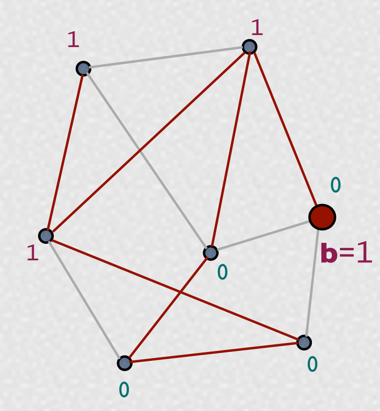


Chaum’s solution

- 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



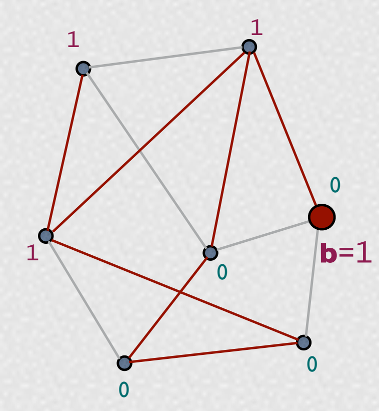
Chaum’s solution

- 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

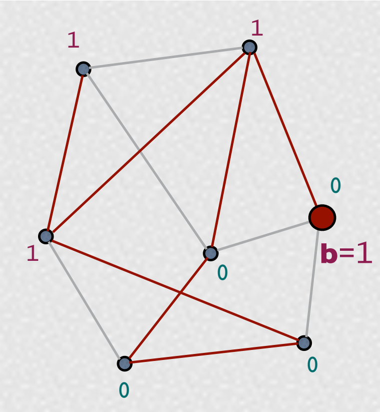
Question: Does the protocol protects the anonymity of the source?

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

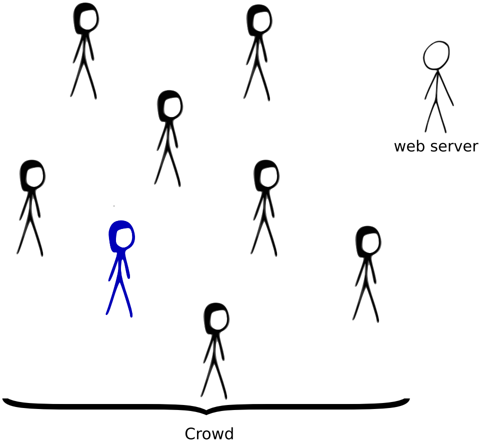
  - A priori / a posteriori = before / after observing the declarations



The Crowds protocol

- DC is not practical for a large number of users
- In practice we might want to trade anonymity for efficiency
- Crowds offers a weaker notion of anonymity called probable innocence
- Designed for anonymous web surfing

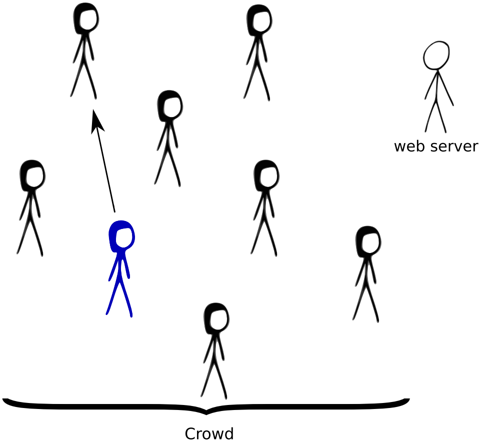
The Crowds protocol



The Crowds protocol

The initiator:

- Forwards the message



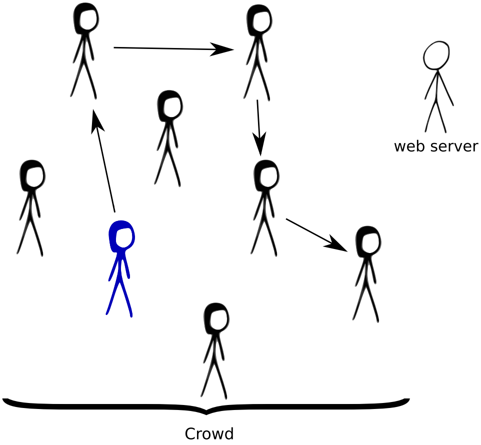
The Crowds protocol

The initiator:

- Forwards the message

A forwarder:

- With pb p_f forwards



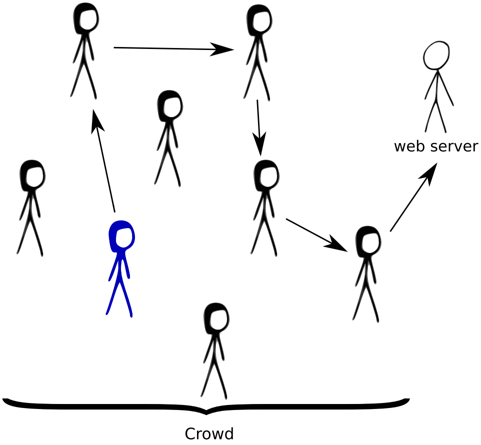
The Crowds protocol

The initiator:

- Forwards the message

A forwarder:

- With pb p_f forwards
- With pb $1 - p_f$ delivers



The Crowds protocol

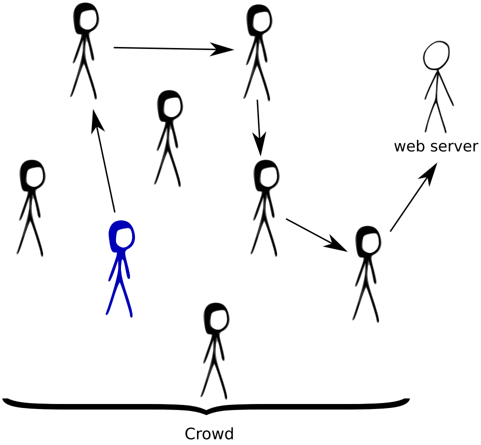
The initiator:

- Forwards the message

A forwarder:

- With pb p_f forwards
- With pb $1 - p_f$ delivers

- The path is used in the opposite direction for the reply



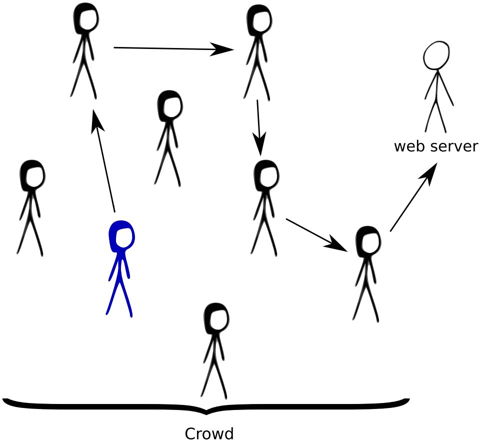
The Crowds protocol

The initiator:

- Forwards the message

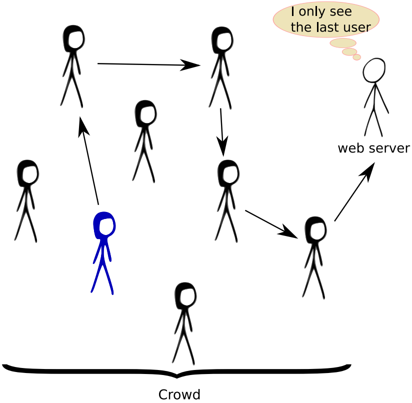
A forwarder:

- With pb p_f forwards
- With pb $1 - p_f$ delivers
- The path is used in the opposite direction for the reply
- The same path is used in future requests



The Crowds protocol: anonymity

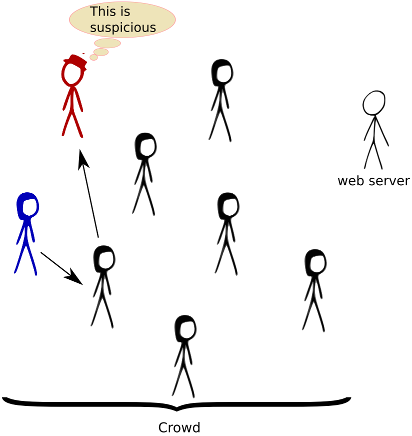
- We consider sender anonymity
- Attacker model
  - Cannot see the whole network
  - Only messages sent to him
- The server:
  - only sees the last user
  - Strong anonymity is satisfied



The Crowds protocol: anonymity

Corrupted users:

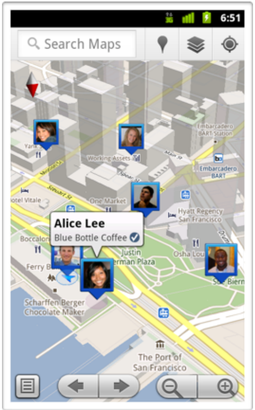
- They can see forwarding requests and "detect" a user i
- User i can still claim that he was forwading the message for user j
- Is strong anonymity satisfied?
- Compare the probability to detect i:
  - when i is the payer
  - when j is the payer
- They are different: strong anonymity is violated



Location-Based Systems

A **location-based system** is a system that uses geographical information in order to provide a service.

- Retrieval of Points of Interest (POIs).
- Mapping Applications.
- Deals and discounts applications.
- Location-Aware Social Networks.



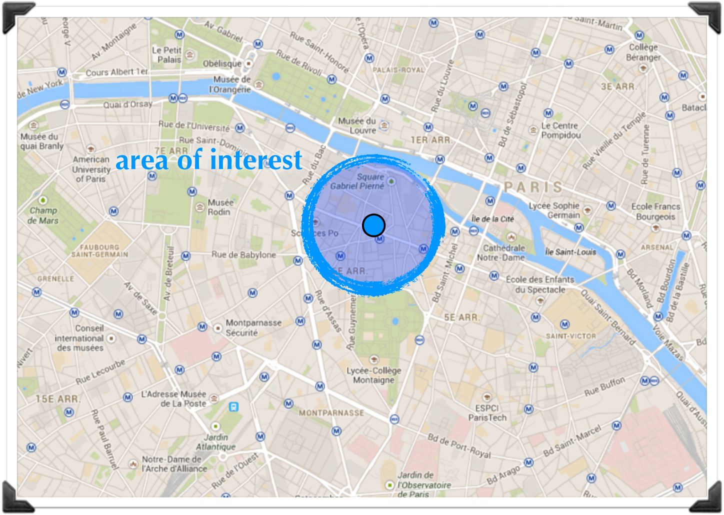
Location-Based Systems

- Location information is sensitive. (it can be linked to home, work, religion, political views, etc).
- Ideally: we want to **hide our true location**.
- Reality: we need to **disclose some information**.

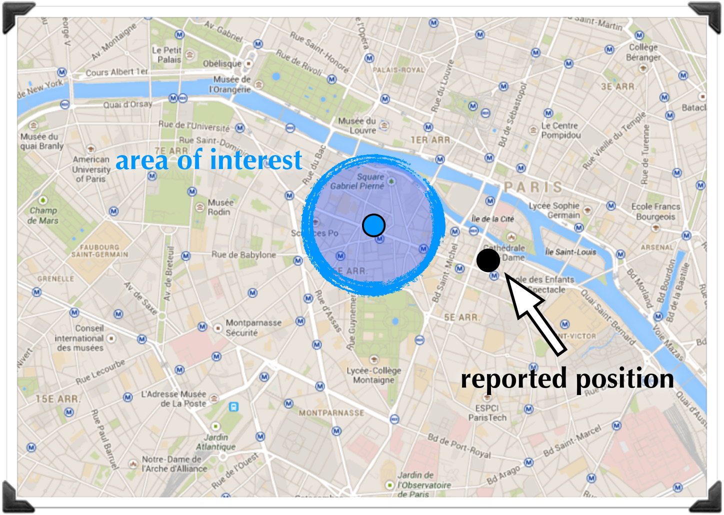
Example

- Find restaurants within 300 meters.
- Hide location, **not identity**.
- Provide **approximate location**.

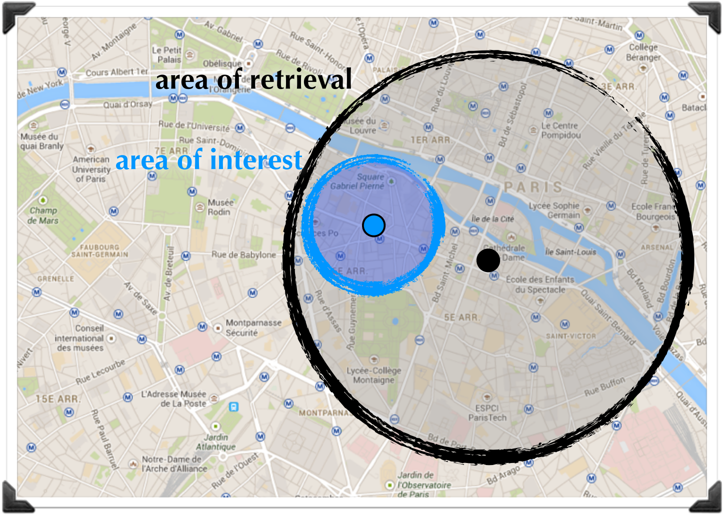
Obfuscation



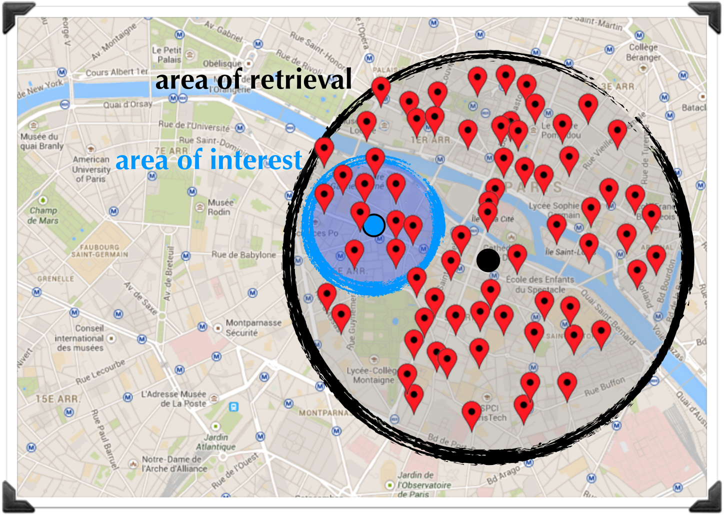
Obfuscation



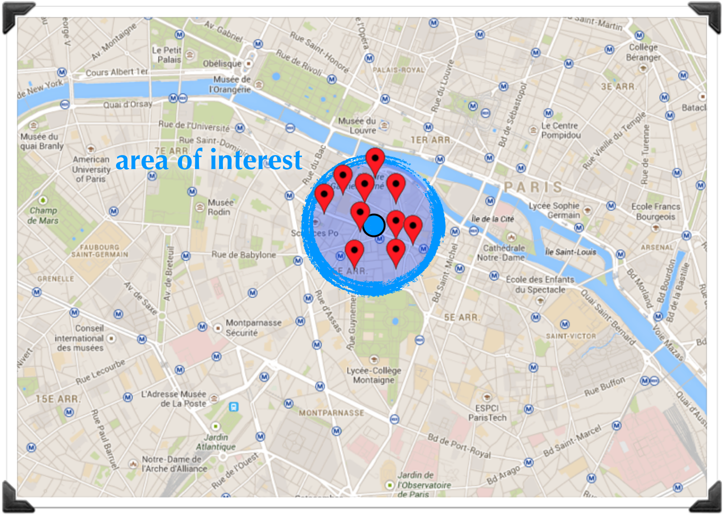
Obfuscation



Obfuscation



Obfuscation



area of interest

Issues to study

How can get we generate the noise?

What kind of formal privacy guarantees do we get?

Which mechanism gives optimal utility?

What if we use the service repeatedly?