Anonymity and Information Hiding in Multiagent Systems: A Knowledge-based Approach

Joseph Halpern
&
Kevin O'Neill

Motivation

Anonymity is important to people in a variety of situations:

- Browsing the web
- Sending messages
- "Whistle-blowing"
- Often people will be reluctant to engage in some behavior unless they can receive guarantees that their anonymity will be protected

Important to understand what anonymity means:

- So it can be implemented and verified
- To understand how it can be exploited by "bad guys"

What is Anonymity?

Others have given definitions:

- using epistemic logics [Syverson & Stubblebine, 1999];
- using CSP [Schneider & Sidiropoulos, 1996];
- using functions views [Hughes & Shmatikov, 2003];
- informally, by stating that the identity of an actor should not be revealed to other observers.

Basic intuition: “observers” should not be able to connect anonymous actions with agents who perform them.

Comparing the Definitions

These definitions are useful, but there is no clear winner:

- Some definitions are expressive but have no clear connection to any particular system representation.
- Others are based on a system representation (e.g., CSP) but are (arguably) not as expressive.
- None handle concerns about *probabilistic* inference.
- It isn’t clear how definitions relate to each other.

What We Do:

- Describe a framework in which we can both describe systems, and also give very expressive definitions of anonymity.
- Make clear how the definitions formally capture intuitions about *knowledge*, using a modal logic.
- Give a variety of probabilistic definitions of anonymity.
- Show how other definitions are related to ours, and also to each other, by giving formal equivalence theorems.

Anonymity as Information Hiding

We define anonymity as an instance of "information hiding". We ask:

- what information needs to be hidden?
- who does it need to be hidden from?
- how well does it need to be hidden?

Information hiding is closely related to the knowledge of the agents interacting with the system.

We also relate anonymity to our earlier work on secrecy and noninterference.

What's to come:

Motivating Example:

- Dining cryptographers

The system framework:

- Sets of "runs", or execution sequences.
- Propositional logic to reason about system properties.

Definitions of anonymity:

- Based on previous definitions.
- Talk about the knowledge of observers.
- We incorporate probability!

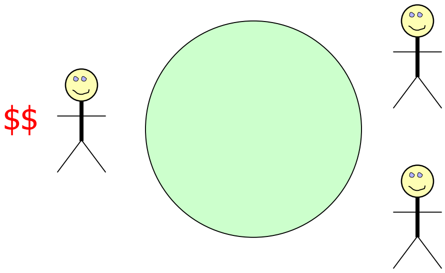
Related work:

- A case study: CSP and anonymity

Example: Dining Cryptographers

Suppose three cryptographers have dinner:

- They find out that the bill has been paid anonymously by someone.
- They want to find out if it was someone in their group.
- But want to preserve anonymity of the payer!



Chaum’s Protocol

1. Each pair of cryptographers generates a random shared bit by flipping a coin.
2. Each cryptographer announces a bit: whether the two bits are the same are different
    - the XOR of her shared bits (same – 0; different – 1)
3. If a cryptographer is the payer, she flips the bit
    - the XOR of shared bits and a “paid” bit
4. An odd number of “differences” (1 bits) indicates that a cryptographer is paying
    - the XOR of publicly announced bits

Representing Multiagent Systems

Our model lets us represent all possible behaviors of the system as well as the state of the agents who use the system.

- n agents, with each agent i in some local state s_i at a given point in time
- The whole system in some global state $(s_1, \ldots, s_n, s_e)$
- A run r is a function from time to global states
- A point of the system is a pair (r,m) – a particular execution sequence at a particular point in time
- A system R is a set of runs

Assumptions and Limitations

All agents (including possible attackers) are modeled explicitly.

The system model includes all possible execution sequences—accounting for nondeterministic choices made by individual agents.

The structure of the system is common knowledge among the agents of the system.

- Agents might have access to program code.

The Dining Cryptographer's System

The local state of a cryptographer includes:

- whether or not they paid for dinner
- the outcome of private coin flips with other cryptographers
- statements of the other cryptographers (the XOR of their bits)

A run is determined by the result of all the coin flips.

In a given run, a cryptographer's state changes over time as he obtains more information.

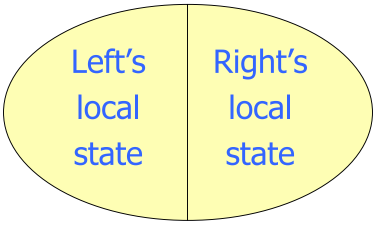
Local States and Knowledge

We write $r_i(m)=s_i$ if i has local state s_i at point (r,m).

- At the point (r,m), agent i considers possible all the points (r′,m′) such that $r_i(m)=r_i'(m')$.
- If a fact φ is true at all points that i considers possible, we say that "i knows the fact φ".
  - Denoted "$K_i\varphi$"
- If a fact φ is true at some point that i considers possible, we say that "i considers φ possible".
  - Denoted "$P_i\varphi$"
  - $P_i\varphi$ iff $\neg K_i\neg\varphi$

Knowledge: A Schematic Example

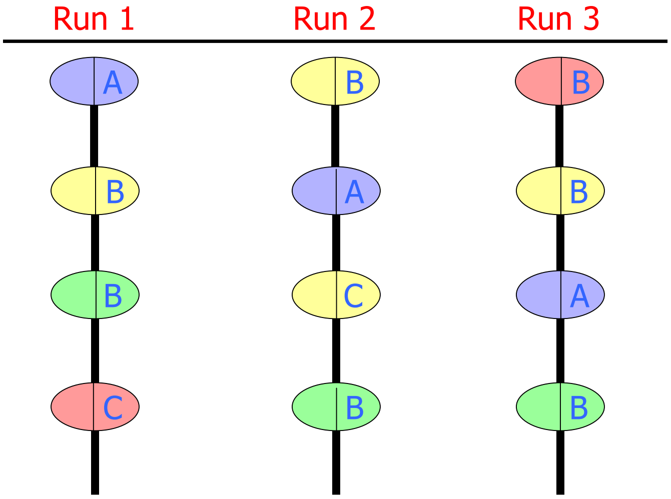
Suppose our system has two agents, "Left" and "Right". The following bubble represents the global state of the system at some point in time.



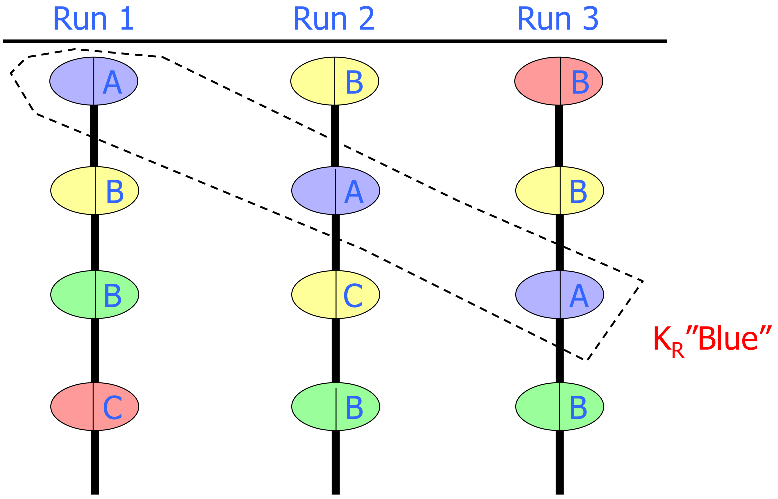
Left's local state

Right's local state

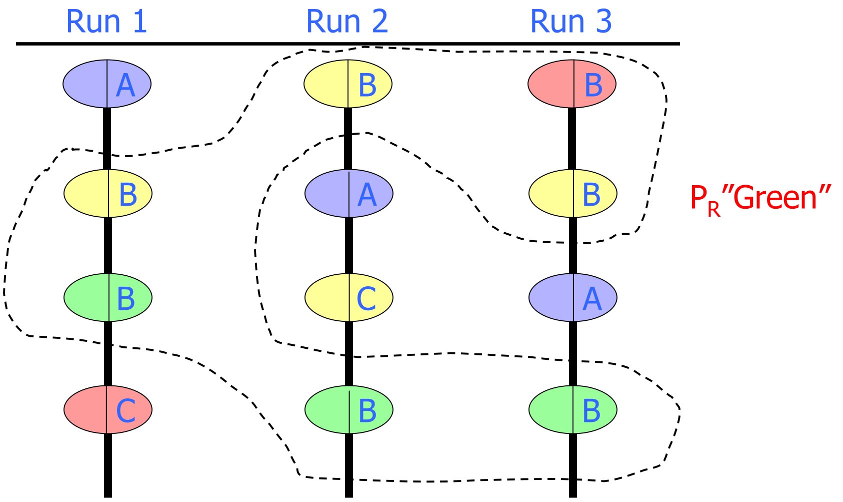
A Schematic Example



A Schematic Example



A Schematic Example



Defining Anonymity

We define anonymity in terms of actions and the agents who perform them.

- δ(i,a): the fact that i has performed action a

Action a, performed by agent i, is *minimally anonymous* with respect to agent o in R if the formula "$\neg K_o[\delta(i,a)]$" is always true.

- If an observer o knows that i sent a message, then i doesn't have any anonymity, at least with respect to o.

Minimal anonymity is a very weak condition:

- Minimal anonymity holds as long as o is not 100% sure that i performed action a.

A Stronger Version of Anonymity

An agent i might want to ensure that observers think it possible that many agents in some "anonymizing set" I could have performed the anonymous action.

Action a, performed by agent i, is *anonymous up to I* (with respect to an agent o) in R if the following formula is always true:

$$\delta(i,a) \rightarrow \bigwedge_{i' \in I} P_o[\delta(i',a)]$$

- Anonymity up to I implies minimal anonymity—under a few simple assumptions.

Total anonymity: when I contains *all* agents (except o)

Back to the Cryptographers…

If "I" is the set of all three cryptographers, we want the (possible) payer to have:

- Anonymity up to I with respect to outside observers (e.g., the maitre d')
- Anonymity up to I – {j} with respect to any of the other cryptographers (named j)
  - – Need I – {j} because j knows whether or not he paid

The protocol has been verified using a knowledge-based framework [van der Meyden and Su, 2002].

Probabilistic Definitions of Anonymity

Problems with "possibilistic" guarantees:

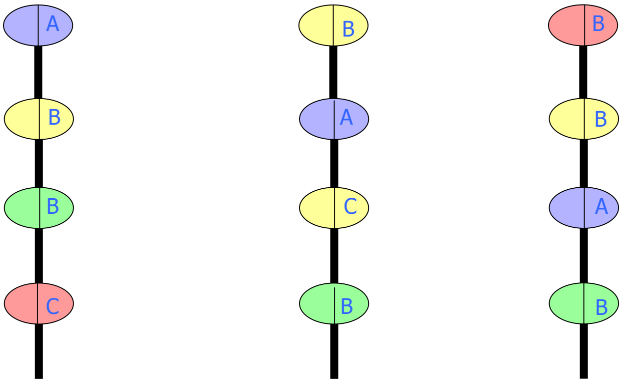
- Suppose an observer o thinks that any of 101 agents in a set I could have performed an action a.
- What if o has a probability of 0.99 that i performed a, and a probability of 0.0001 that any of the other 100 agents performed a?
- Here anonymity up to I doesn't provide much comfort to i.

Other definitions of anonymity have not handled agents who can perform probabilistic inference.

Adding Probability to the Framework

Each run has an (objective) probability:

$\mu(r_1) = 0.5$ $\mu(r_2) = 0.3$ $\mu(r_3) = 0.2$



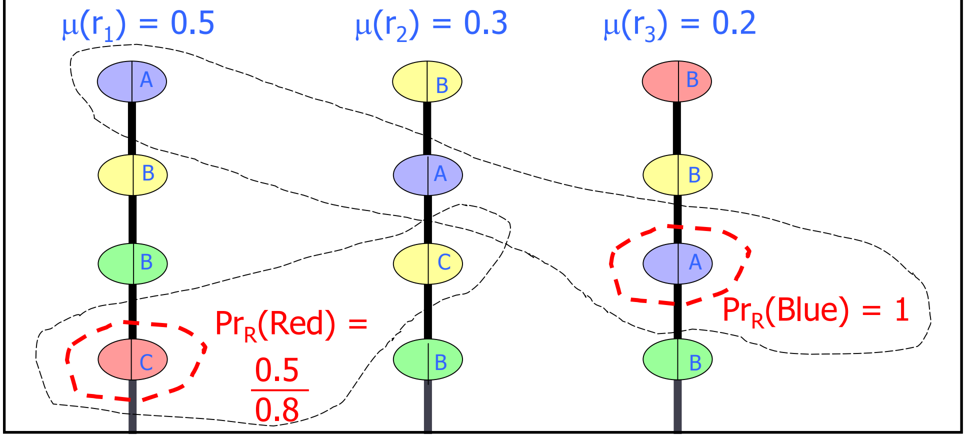
A B B C

B A C B

B B A B

Adding Probability to the Framework

Then agents have subjective probabilities at *points*, by conditioning on their local state:



Probabilistic Anonymity

We define a number of variants of probabilistic anonymity in our framework:

- i maintains α*-anonymity* (w.r.t. o)*:*

$$Pr_o[\delta(i,a)] < \alpha$$

- i maintains *strong probabilistic anonymity up to* I (w.r.t. o):

$$Pr_o[\delta(i,a)] = Pr_o[\delta(i',a)], \text{ for each } i' \text{ in } I$$

Are Probabilistic Definitions Too Strong?

Consider an anonymous message-passing system:

- Even if the *system* doesn't leak my identity, the *content* of my messages may provide good clues.

Or anonymous donations:

- Bill Gates might want strong probabilistic anonymity when he makes a donation.
- BUT: observers will have prior probabilities on what various agents might do in a given system.
  - Kevin is unlikely to make a multimillion-dollar donation

Can we say that the system is "doing its best" to provide anonymity?

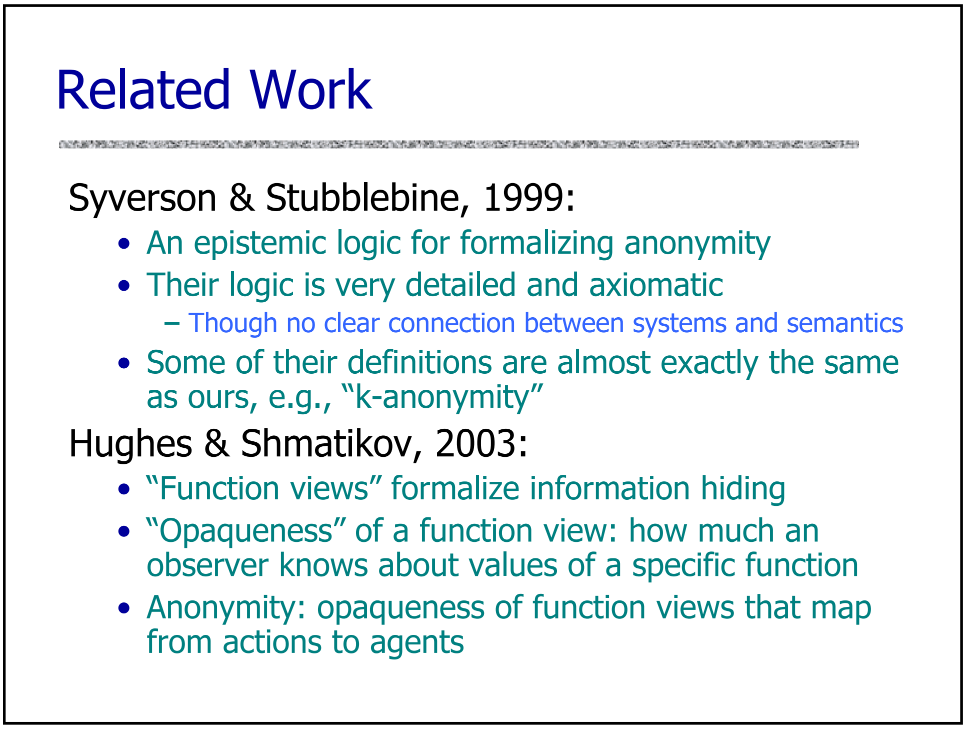
Conditional Anonymity

Consider our definition of probabilistic noninterference [Halpern & O'Neill 2002]:

- Whatever an unclassified user sees while interacting with the system, her probability of classified events remains unchanged.

Our definition of conditional anonymity captures similar intuitions:

- If o knows that somebody has performed a, then her probability of δ(i,a) should be the same at every point in the system where somebody has performed a.
- Regarding δ(i,a), o can't learn anything that she didn't already know—except perhaps that somebody performed a.



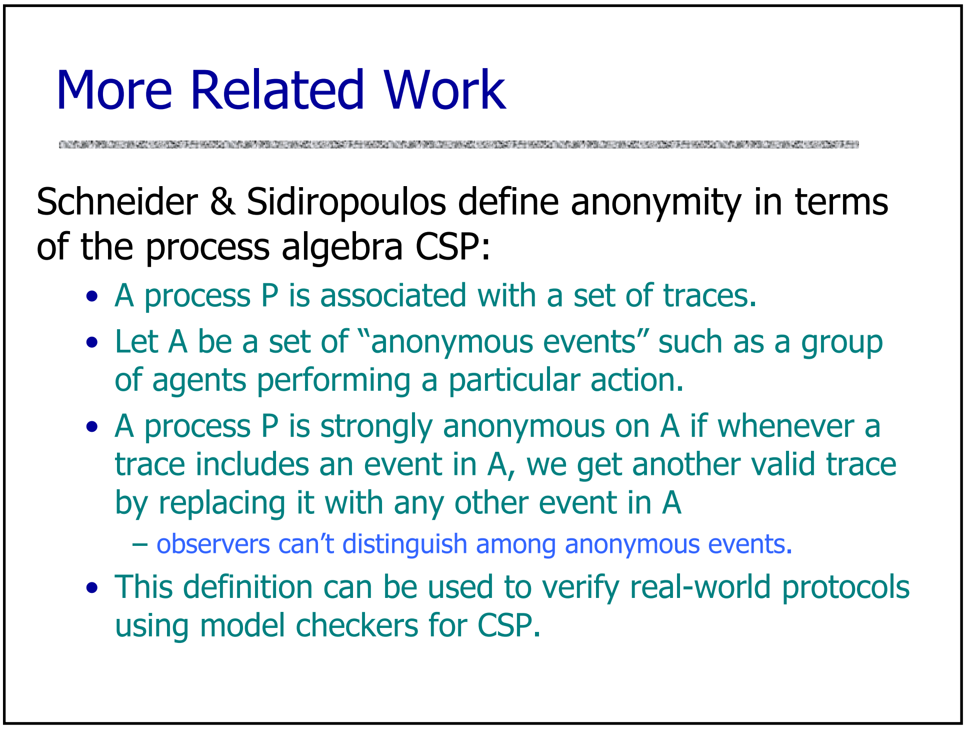
Related Work

Syverson & Stubblebine, 1999:

- An epistemic logic for formalizing anonymity
- Their logic is very detailed and axiomatic
  - Though no clear connection between systems and semantics
- Some of their definitions are almost exactly the same as ours, e.g., “k-anonymity”

Hughes & Shmatikov, 2003:

- “Function views” formalize information hiding
- “Opaqueness” of a function view: how much an observer knows about values of a specific function
- Anonymity: opaqueness of function views that map from actions to agents



More Related Work

Schneider & Sidiropoulos define anonymity in terms of the process algebra CSP:

- A process P is associated with a set of traces.
- Let A be a set of “anonymous events” such as a group of agents performing a particular action.
- A process P is strongly anonymous on A if whenever a trace includes an event in A, we get another valid trace by replacing it with any other event in A
  - observers can’t distinguish among anonymous events.
- This definition can be used to verify real-world protocols using model checkers for CSP.

CSP and Anonymity

We show that this definition is a special case of the definitions we give

- A process P can be associated with a set of runs R_P, simply by converting traces to runs.
- Suppose that the set A comes from a particular action a and group of agents I_A.

Theorem: P is strongly anonymous on A if and only if action a is anonymous up to I_A for agents in $I_{A.}$

For the Future

Verification:

- Using the knowledge-based framework directly [van der Meyden, 1998],
- Or indirectly, using a related framework such as CSP or the pi-calculus.
- Of fielded systems (e.g., Onion-routing, P5, Herbivore).

More connections to PL work:

- Capturing various process algebra equivalences in the runs and systems framework.

More connections to information flow:

- "Axioms for information hiding", including a formal treatment of declassification.