

# Anonymity in Multiagent Systems: A Knowledge-based Approach

Joseph Halpern and Kevin O'Neill, Cornell University

## Why Do We Need to Define Anonymity?

- Anonymity is important in a variety of situations:
  - General privacy when Web-browsing and sending email
  - "Whistle-blowing"
- It's important to have good definitions of what anonymity means:
  - To compare the guarantees offered by different systems
  - For verification and specification
  - For legal and policy-related reasons
  - To understand how anonymity can be exploited by "bad guys"
- Some definitions of anonymity have been given
  - Using CSP [Schneider & Sidiropoulos], epistemic logics [Syverson & Stubblebine], and function views [Hughes & Shmatikov], as well as more informal attempts
  - But there is no general-purpose framework that includes an expressive syntax, a direct connection to a system representation, and a straightforward way for reasoning about probability and probabilistic inference
  - Our work was directly inspired by Vitaly Shmatikov's presentation at last year's Cape May SPYCE meeting!

## Anonymity as Information Hiding

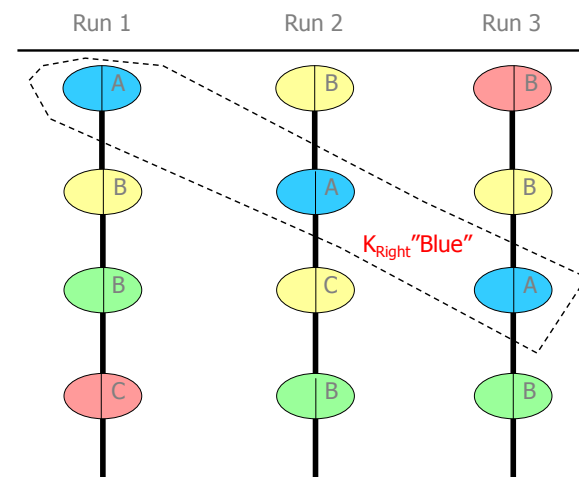
- We define anonymity as an instance of "information hiding", where we ask:
  - what information needs to be hidden?
  - who does it need to be hidden from?
  - how well does it need to be hidden?
- This is related to other kinds of information hiding, such as cryptographic secrecy and information flow secrecy
- Anonymity (and information hiding in general) is closely related to the *knowledge* of the agents who use or interact with the system in question
  - Our work tries to make this precise, for a wide variety of definitions of anonymity

## Knowledge in Multiagent Systems

- A "multiagent system" describes all the possible behaviors of the agents in some system, and what information they have at any point in time:
  - each agent  $i$  has some local state  $s_i$  at a given point in time, representing her current information
  - the whole system has a global state  $(s_1, \dots, s_n, s_e)$
  - a run  $r$  is a function from time to global states
    - a run describes one possible execution of the system over time
  - a point is a pair  $(r, m)$ , which models the global state of the system at time  $m$  for some particular execution  $r$
  - a system (denoted  $R$ ) is a set of runs
- Representing 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  $\phi$  is true at all points that  $i$  considers possible, we say that " $i$  knows the fact  $\phi$ " (written " $K_i \phi$ ")
  - If a fact  $\phi$  is true at some point that  $i$  considers possible, we say that " $i$  considers  $\phi$  possible" (written " $P_i \phi$ ")

A Schematic Example:

- ovals represent points in time; colors represent properties of the system
- letters A, B, etc., represent the local state of agent "Right"



- Presented at the 2003 Computer Security Foundations Workshop in Asilomar, California
- Invited to a special issue of the Journal of Computer Security (devoted to CSFW 16)
- Partially supported by the CIP/SW URI "Software Quality and Infrastructure Protection for Diffuse Computing" through ONR grant N00014-01-1-0795

## Defining Anonymity: A First Start

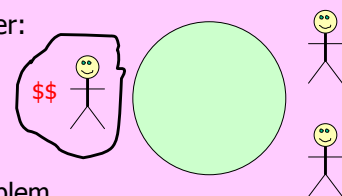
- We define anonymity in terms of actions and the agents who perform them:
  - let " $\delta(i, a)$ " be the fact that  $i$  performs action  $a$
- Action  $a$ , performed by agent  $i$ , is *minimally anonymous* with respect to agent  $o$  in  $R$  if the formula " $\neg K_o[\delta(i, a)]$ " is always true
  - If an observer  $o$  knows that  $i$  sent a message, then  $i$  doesn't have any anonymity, at least with respect to  $o$
  - This is a very weak condition
- Action  $a$  is anonymous up to  $I$  (with respect to an agent  $o$ ) in  $R$  if the following formula is always true: " $\delta(i, a) \rightarrow \bigwedge_{i \in I} P_o[\delta(i, a)]$ "
  - all agents in some "anonymizing set",  $I$ , could have performed the anonymous action
- If  $I$  is the set of cryptographers in the dining cryptographers problem, we want:
  - 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  $j$

## Adding Probability

- Why probability is important:
  - 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  $a$ ?
- Adding probability to the system is straightforward to do.
  - The syntax of probability matches intuitions, and has a formal semantics
- We say agent  $i$ , has  $\alpha$ -anonymity with respect to agent  $o$  in  $R$  if the formula " $\text{Pr}_o[\delta(i, a)] < \alpha$ " is always true
  - Other definitions are fairly similar
- We also define a new notion of "conditional anonymity":
  - If  $o$  knows that somebody has performed  $a$ , then her probability of  $\delta(i, a)$  should be the same at every point in the system where somebody has performed  $a$
  - Regarding  $\delta(i, a)$ ,  $o$  can't learn anything that she didn't already know—except perhaps that somebody performed  $a$
  - This relates to definitions in our earlier work on information flow secrecy...

## Example: Dining Cryptographers

- Suppose three (or more generally,  $n$ ) 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 they want to preserve the anonymity of the payer!
- Chaum [1988] introduced a general scheme for solving this problem
  - It is useful for providing anonymous message transmission (DC Nets)
- The simplicity of the problem is also useful for comparing definitions of anonymity
  - Our definitions of minimal anonymity, anonymity up to a group of users,  $k$ -anonymity, probabilistic anonymity, and conditional anonymity all apply to the dining cryptographers problem



## Future/ongoing work

- Protocols allowing "revocable" anonymity for network communications
  - To allow anonymity to be undone in the case of egregious messages
- Representing protocols for anonymous authentication
- More work on information hiding
  - Access control in XML