# A Macroscopic Privacy Model for Heterogeneous Robot Swarms

Amanda Prorok and Vijay Kumar

University of Pennsylvania, Philadelphia PA, USA
[prorok | kumar]@seas.upenn.edu

**Abstract.** To date, the issues of privacy and security remain poorly addressed within robotics at large. In this work, we provide a foundation for analyzing the privacy of swarms of heterogeneous robots. Our premise is that information pertaining to individual robot types must be kept *private* in order to preserve the security and resilience of the swarm system at large. A main contribution is the development of a macroscopic privacy model that can be applied to swarms. Our privacy model draws from the notion of *differential privacy* that stems from the database literature, and that provides a stringent statistical interpretation of information leakage. We combine the privacy model with a macroscopic abstraction of the swarm system, and show how this enables an analysis of the privacy trends as swarm parameters vary.

## 1 Introduction

To date, the issues of privacy and security remain poorly addressed within robotics at large. These issues are particularly important in large-scale multi-robot systems, where individual robots must share data to coordinate their actions and communicate with human operators. In this work, we frame the problem of privacy within the context of heterogenous robot swarms. Indeed, while much research in the domain of distributed robotics and multi-robot systems explores how to develop strategies for coordinating robots of the same type (i.e., homogeneous robot teams), it is generally acknowledged that exploiting *heterogeneity* by design leads to more robust, versatile, and efficient systems — especially when functional, temporal, spatial, and behavioral heterogeneity are explicitly sought after [3, 14]. However, by introducing heterogeneity by design, we impose a certain degree of uniqueness and specialization. As a consequence, any given robot type may be critical to securing the system's ability to operate without failure. Hence, we must find ways of protecting the heterogeneous swarm to avoid threats that arise when the various roles within the swarm can be determined by adversaries.

A class of applications for swarm robotic systems is based on an architecture that exploits a centralized component (e.g., cloud connectivity). This type of architecture has been shown to be a very efficient and practical means for coordinating robotic swarms, because it enables a form of feedback control based on abstract information of the swarm's state, permitting high-level interaction between the swarm and a supervisory agent [11, 1]. In addition, this structure is beneficial as it facilitates *(i)* access to big data, *(ii)* access to parallel computing facilities, *(iii)* access to collective learning structures,

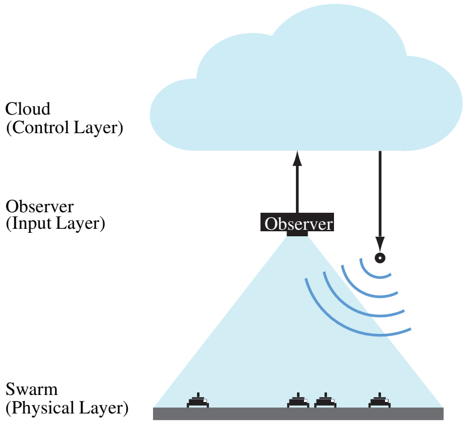

**Fig. 1.** The swarm of robots is supported by additional infrastructure in the cloud: observable system-level information is gathered by some sensor modality (e.g., camera), and forwarded to remote processing entities that reside in the cloud. Optionally, processed information (such as control feedback) may be relayed back to the swarm.

and *(iv)* access to human support [9]. Fig. 1 schematizes the architecture. The robot swarm operates on a physical layer. On a subsequent layer, *observable system-level* state information is gathered by means of some sensory data stream (camera, radio, etc.). This data stream is public, and is observable by one or several observers that are physically co-located with the swarm itself. Finally, the state information is forwarded to the cloud, where it is used for purposes such as monitoring, data processing, or machine learning. The data may also be used as input to a controller, which, based on this stream and potentially other information available in the cloud, computes a control feedback that is relayed back to the swarm. In this scheme, there are two points where privacy matters. The first point is at the level of the physical observer(s), who may not be allowed to determine private data (e.g., identify the different types of robots operating in a given space). The second point is within the cloud, where entities (such as the controller) are not allowed to act upon private information. In the following, we will formalize the notion of privacy with respect to system-level information.

**Definition 1 (Private Robot Swarm).** *A private robot swarm is a swarm of heterogeneous robots where any individual robot cannot be attributed to a particular robot type (or species), due to a lack of outstanding observable features.*

The current paper presents a technique that allows us to analyze the privacy of heterogeneous robot swarms by quantifying the amount of information leaked when an external observer is able to gather high-level information on the swarm's behavior. We demonstrate our technique through a case-study of collaborative task solving. Specifically, we make the following contributions:

*1) Heterogeneous Swarm Model:* We begin by formalizing a framework that allows us to capture the essential behavioral characteristics of a heterogeneous swarm. We are interested in the dynamics that arise in systems where robots of different species (or types) must collaborate to achieve high-level goals. We show how our model facilitates the design of the swarm based on a definition of *interspecific interactions* [2], i.e., states that depend on the interactions of multiple species. Finally, we introduce the notion of *observable system-level* information, i.e., information that can be observed by an outsider, and show how it is embedded in our model.

*2) Privacy Model:* The definition of privacy (or anonymity) is a difficult task, and a significant amount of research in the database literature is dedicated to this subject. A recent successful definition is that of *differential privacy* [4], which provides strong anonymity guarantees in the presence of arbitrary side information. One of our main contributions in this work is the development of an equivalent notion of privacy that can be applied to dynamic robot swarms.

*3) Method of Analysis:* Finally, we present a technique that employs the swarm model and privacy model jointly to produce a quantitative analysis of privacy. The method is based on computational tools that can be applied to a wide variety of swarm dynamics. We demonstrate this application in a case-study of collaborative task-solving.

## 2 Model of Robot System

Heterogeneity and diversity are core concepts of this work. To develop our formalism, we borrow terminology from biodiversity literature [13]. We define a system of robots, where each robot belongs to a *species*. The system is composed of $N_S$ species $\mathcal{S} = \{1, \ldots, N_S\}$, with a total number of robots $N$, and $N^{(s)}$ robots per species such that $\sum_{s \in \mathcal{S}} N^{(s)} = N$. At the individual level, the robots are governed by stochastic control policies [1, 14]. A finite-state-machine (FSM) accounts for interspecific states. We denote these states as $a^{\mathcal{I}}_{(\cdot)}$, where the subscript denotes the state activity (e.g., search, navigate, etc.), and the superscript $\mathcal{I}$ is the set of all species that are involved in this state. For example, $a^{\{1,2\}}_{(\mathrm{grip})}$ is a state where species 1 and 2 collaborate to grip an object. Note that $\mathcal{I}$ may also be an empty set, which indicates that the state is unrelated to any particular species (we call such states byproducts, as they can relate to performance metrics or environmental conditions).

Since we focus on designing interspecific interaction mechanisms, we choose a modeling framework that implicitly accounts for system-level state transitions. We build our formalism on the theory of Chemical Reaction Networks (CRN) [5], because it allows us to define interaction mechanisms efficiently, while capturing essential system dynamics that depend on robot-to-robot interactions. CRNs are a powerful means of representing complex systems — though not a new field of research, many recent research findings that simplify the calculations are accelerating the adoption of CRNs into domains other than biology and chemistry [12, 5].

We define our CRN as a triplet $\mathcal{N} = (\mathcal{A}, \mathcal{C}, \mathcal{R})$, where $\mathcal{A}$ is the set of states, $\mathcal{C}$ is the set of complexes, and $\mathcal{R}$ is the set of reactions.

*State set $\mathcal{A}$:* The state set encompasses all states that arise in the system, with $\mathcal{A} = \{A_1, \ldots, A_{N_A}\}$ where $N_A$ is the number of states. States relating to a specific species $s$ are denoted by $\mathcal{A}^{(s)}$ such that

$$\mathcal{A}^{(s)} = \bigcup_{s \in \mathcal{I}} a^{\mathcal{I}}_{(\cdot)} \tag{1}$$

The set of all states includes both species-specific states as well as byproduct states $a^{\emptyset}_{(\cdot)}$ such that

$$\mathcal{A} = \bigcup_{s=0}^{N_S} \mathcal{A}^{(s)} \text{ where } \mathcal{A}^{(0)} = \{a^{\emptyset}_{(\cdot)}\} \tag{2}$$

We can identify the interactive (interspecific) states of an arbitrary subset of species $\tilde{\mathcal{S}} \subset \mathcal{S}$ by considering the intersection of sets $\cap_{i \in \tilde{\mathcal{S}}} \mathcal{A}^{(i)}$. Trivially, if $\cap_{i \in \tilde{\mathcal{S}}} \mathcal{A}^{(i)} = \emptyset$, then the species in $\tilde{\mathcal{S}}$ do not interact.

The CRN is a population model, and allows us to keep track of the number of robots in each of the states in $\mathcal{A}$. Hence, we define a population vector $\mathbf{x} = [x_1, \ldots, x_{N_A}] \in \mathbb{N}_{\geq 0}^{N_A}$, where $x_i$ corresponds to the population present in state $A_i$. We refer to the population vector $\mathbf{x}$ as the system-level state. In order to simplify the formulation of our case studies later on, we will also use the notation $x_{(\cdot)}^{\mathcal{I}}$ to refer explicitly to the population in state $a_{(\cdot)}^{\mathcal{I}}$.

*Complex set $\mathcal{C}$:* The complex set is defined as $\mathcal{C} = \{C_1, \ldots, C_{N_C}\}$ where $C_j = \sum_{i=1}^{N_A} \rho_{ij} A_i$ for $j = 1, \ldots, N_C$, with vector $\boldsymbol{\rho_j} = [\rho_{1j}, \ldots, \rho_{N_A j}]^\top \in \mathbb{N}_{\geq 0}^{N_A}$. A complex is a linear combination of states, and denotes the net input or output of a reaction. In other words, a complex denotes either *(i)* the states that are required for a certain reaction to take place, or *(ii)* the states that occur as an outcome of a certain reaction that took place. The non-negative integer terms $\rho_{ij}$ are coefficients that represent the multiplicity of the states in the complexes.

*Reaction set $\mathcal{R}$:* We use complexes to formulate reactions

$$R_l : C_j \xrightarrow{r_l} C_k. \tag{3}$$

The reaction set is defined as $\mathcal{R} = \{R_1, \ldots, R_{N_R}\}$, with $N_R$ the number of reactions, such that $R_l \in \{(C_j, C_k) | \exists\, C_j, C_k \text{ with } C_j \to C_k\}$ for $j, k = 1, \ldots, N_C$, and where $r_l$ is the propensity function $r_l(\mathbf{x}; \kappa_l) : \mathbb{N}_{\geq 0}^{N_A} \mapsto \mathbb{R}_{\geq 0}$ parameterized by $\kappa_l$. In this work, we use mass-action propensity functions, and $r_l(\mathbf{x}; \kappa_l) = \kappa_l \prod_{i=1}^{N_A} x_i^{\rho_{ij}}$ for all $R_l = (C_j, \cdot)$. The net loss and gain of each reaction is summarized in a $N_A \times N_R$ stoichiometry matrix $\Gamma$, the columns of which encode the change of population per reaction. In particular, the $i$-th column of $\Gamma$ corresponds to the $i$-th reaction $R_i = (C_j, C_k)$ and thus, the column is equal to $\boldsymbol{\rho_k} - \boldsymbol{\rho_j}$. The elements $\Gamma_{ji}$ are the so-called stoichiometric coefficients of the $j$-th state in the $i$-th reaction. Positive and negative coefficients denote products and reactants of the reaction, respectively.

Finally, we describe the dynamics of our system with help of two functions: an execution function $f_{\mathcal{N}}$, and a query function $q$:

$$\begin{aligned} f_{\mathcal{N}}(\mathbf{x_0}, t) :&\ \mathbb{N}_{\geq 0}^{N_A} \times \mathbb{R}_{\geq 0} \mapsto \mathbb{N}_{\geq 0}^{N_A} \\ q(\mathbf{x}) :&\ \mathbb{N}_{\geq 0}^{N_A} \mapsto \mathbb{N}^{N_O},\, N_O \in \mathbb{N}_{>0} \end{aligned} \tag{4}$$

The execution function $f_{\mathcal{N}}$ is a stochastic process that formulates the system's evolution over time, and that is governed by the states and reactions defined in $\mathcal{N}$. It is based on an initial population $\mathbf{x_0}$ of the swarm in states $A_i$, and returns the population vector $\mathbf{x}(t)$, evaluated at a fixed time $t$. The query function $q$ allows us to formalize the notion of an *observable system-level* state, as introduced in Section 1. It takes the population vector $\mathbf{x}$ as input, and returns a vector of observable values $\mathbf{y}$ [1]. In its

[1] The modeling framework (CRN) can encompass any measurable state that is associated to the swarm (beyond physical robot states). As a consequence, there are no limitations to what the observable state can represent. This choice can be made by the designer as a function of what may be exposed in a given system.

most basic form, the query function is the identity function, meaning that an observer is able to capture the exact system-level state, and $\mathbf{x} = \mathbf{y}$. In this work, we show a more involved analysis by assuming that the observed values take the form of simple summations over the population vector. This assumption is well motivated when individual states are physically not distinguishable from an outside vantage point, and thus, only aggregated values can be observed. I.e., $y_i = \sum_{j\in\Omega_i} x_j$ with $\Omega_i \subset \{1,\ldots,N_A\}$, $\cup_{i\in\{1,\ldots,N_O\}}\Omega_i = \{1,\ldots,N_A\}$, and all $\Omega_i$ disjoint. We revisit these notions in our case-study, presented in Section 4.

## 3 Definition of Differentially Private Swarm

In this section, we develop our analogy to a formal definition of privacy that stems from the database literature, and that is referred to as differential privacy (formerly known as indistinguishability) [4]. This concept considers two key components: a *database* that holds sensitive information pertaining to individuals, and a *query* that releases information obtained from the database via a mechanism. The goal of differential privacy is to develop mechanisms that are able to provide information in response to database queries, while preserving the privacy of the individuals recorded therein, even in the presence of arbitrary side information[2]. Side information can be understood as a prior probability distribution over the database, and hence, privacy is preserved if no additional information about this distribution is obtained through the query. It is important to note that the condition of differential privacy is made with respect to the release mechanism (i.e., query), and does not depend on the database itself, nor on the side information [8]. In particular, if an individual's presence or absence in the database does not alter the distribution of the output of the query by a significant amount, regardless of the side information, then the privacy of that individual's information is assured.

Our analogy applies the concepts of database and query to the context of heterogeneous swarms. We consider a database that represents the composition of our robot swarm, and that records the species of each of the robots. Also, we consider an observer who is capable of observing the swarm during its operation, and who can query the system by retrieving information about the system-level state (i.e., observable system-level state). Then, our analogous definition of privacy is the notion that the observer cannot obtain private information about individual robots by querying the system (i.e., it remains private to which species the robots belong, cf. Def. 1). The composition of our robot swarm is recorded in a database $\mathcal{D} \in \mathcal{S}^N$ that consists of $N$ entries, where each entry $\mathcal{D}_i$ denotes the species of robot $i$. We define an *adjacency* set $\mathrm{Adj}(\mathcal{D})$ that encompasses all databases $\mathcal{D}$' adjacent to $\mathcal{D}$. Two databases $\mathcal{D}$ and $\mathcal{D}$' are adjacent if they differ by one single entry. In other words, two robot swarms (represented by $\mathcal{D}$ and $\mathcal{D}$') are adjacent if they differ by one robot $i$, meaning that robot $i$ belongs to $s_i$ in $\mathcal{D}$ (i.e., $\mathcal{D}_i = s_i$), and to a different species $s_i' \neq s_i$ in $\mathcal{D}$' (i.e., $\mathcal{D}_i \neq s_i$). As previously described, the behavior of the robot swarm can be described by tracking the

[2] In our context of a robotic swarm, an example of side information could be the number of manufacturing parts ordered to build the swarm. If different robot species are made of different parts, such information can be used to construct an initial guess about the number of robots per species. Thus, one would be able to derive the probability of a robot belonging to a given species.

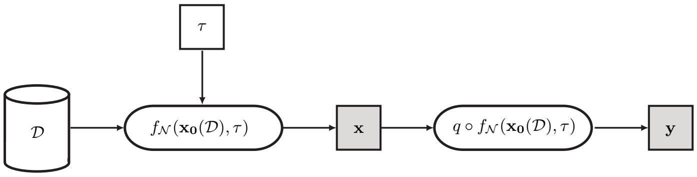
$\tau$
$\mathcal{D}$
$f_{\mathcal{N}}(\mathbf{x_0}(\mathcal{D}), \tau)$
$\mathbf{x}$
$q \circ f_{\mathcal{N}}(\mathbf{x_0}(\mathcal{D}), \tau)$
$\mathbf{y}$

**Fig. 2.** The composition of the robot swarm is recorded in a database $\mathcal{D}$. The function $f_{\mathcal{N}}$ is a stochastic process that executes the swarm and returns a system-level state output $\mathbf{x}$. Query function $q$ reads the (internal) system-level state, and returns the observable output $\mathbf{y}$. Parameter $\tau$ denotes the time at which the system is observed.

states that compose the FSM. If we let the system run, it produces a trajectory that can be evaluated at a given time $\tau$, resulting in a snapshot of the population vector $\mathbf{x}$ [3]. Our query/response model consists of a user who is able to observe this system-level state (at time $\tau$). Hence, the query $q \circ f_{\mathcal{N}}(\mathbf{x_0}(\mathcal{D}), \tau)$ depends on the swarm composition $\mathcal{D}$, and the time at which the system is observed $\tau$. The function $\mathbf{x_0}(\mathcal{D}) : \mathcal{S}^N \mapsto \mathbb{N}_{\geq 0}^{N_A}$ distributes the robots in $\mathcal{D}$ to their initial states. A schema of this system is shown in Fig. 2. Our aim is to analyze the differential privacy of the observed system output. To this end, we propose a definition of differential privacy that applies to dynamic swarms.

**Definition 2** **(<u>$\epsilon$-indistinguishable heterogeneous swarm</u>).** *A heterogeneous swarm with dynamics defined by a system $\mathcal{N}$ is $\epsilon$-indistinguishable (and gives $\epsilon$-differential privacy) if for all possible swarm compositions recorded in databases $\mathcal{D}$, we have*

$$\mathcal{L}(\mathcal{D}) = \max_{\mathcal{D}' \in \mathrm{Adj}(\mathcal{D})} \left| \ln \frac{\mathbb{P}[q \circ f_{\mathcal{N}}(\mathbf{x_0}(\mathcal{D}), \tau)]}{\mathbb{P}[q \circ f_{\mathcal{N}}(\mathbf{x_0}(\mathcal{D}'), \tau)]} \right| \leq \epsilon. \tag{5}$$

*where $\mathbb{P}[\mathbf{y}]$ denotes the probability of the output $\mathbf{y}$, obtained through query $q$ of the system-level state given by $f_{\mathcal{N}}$.*

The value $\epsilon$ is referred to as the leakage. Intuitively, this definition states that if two swarm systems are close, in order to preserve privacy they should correspond to close distributions on their observable outputs. As noted by Dwork et al. [4], the definition of differential privacy is stringent: for a pair of distributions whose statistical difference is arbitrarily small, the ratio may still result in an infinite value when a point in one distribution assigns probability zero and the other non-zero. Later, in our evaluations, we use a smooth version of the leakage formula above, by adding an arbitrary, negligibly small value $\nu$, uniformly over the support of the probability distributions. This allows us to differentiate between large and small mismatches of the output when one point of a probability distribution returns zero. Due to this addition, we are able to show continuous privacy trends as a function of the underlying parameters.

[3] We assume a snapshot adversary that gains system-level information at a specific time. This system-level information is a design variable, called the *observable* state.

# 4 Analysis of Privacy

The formula in Eq. (5) provides strong privacy guarantees. Yet, it requires that we have a way of specifying the probability distribution over the swarm's observable output. For certain classes of swarm dynamics (e.g., when the behavior can be described by complex-balanced CRNs [15]), we can formulate the stationary probability distribution analytically, and hence, plug this formula into our privacy model. Yet, in the general case, we may not be able to derive a stationary probability distribution. Also, we may explicitly need to analyze privacy during transient, non-steady-state, behavior. In this section, we detail a method that enables the analysis of privacy for heterogeneous swarms with arbitrary dynamics.

## 4.1 Methodology

The most widely-used computational method for obtaining the time-dependent behavior of the state of a CRN is the Stochastic Simulation Algorithm (SSA) [6]. The basic idea behind this algorithm is to use the propensity rates to evaluate which reaction is most likely to happen within a given time interval. The result of the algorithm is a sample state trajectory. To obtain meaningful statistical information, the algorithm needs to be repeated a large number of times, which is computationally expensive overall. An alternative approach consists of evaluating the Chemical Master Equation (CME) [10]. The CME describes the temporal evolution of the probability mass function over all possible population vectors, and is described by a set of ordinary differential equations associated to a continuous-time, discrete-state Markov Chain. Since approaches based on the evaluation of the CME tend to be more precise than those based on SSA (when they are computationally tractable), we adopt a solution that builds on the former approach.

The CME is given by the linear ordinary differential equation

$$\dot{\boldsymbol{\pi}}(t) = K\boldsymbol{\pi}(t) \tag{6}$$

with $\boldsymbol{\pi} = [\pi_{\mathbf{x}_i} | \mathbf{x}_i \in \mathcal{X}_{\mathcal{N}}(\mathcal{D})]$ and where is $\mathcal{X}_{\mathcal{N}}(\mathcal{D})$ is the set of all possible population vectors $\mathbf{x}$ that can arise from the CRN $\mathcal{N}$ and robot species specified by $\mathcal{D}$. The entries of $\mathbf{K} \in \mathbb{R}^{|\mathcal{X}_{\mathcal{N}}(\mathcal{D})| \times |\mathcal{X}_{\mathcal{N}}(\mathcal{D})|}$, are given by

$$\mathbf{K}_{ij} = \begin{cases} -\sum_{m=1}^{N_R} r_m(\mathbf{x}_i; \kappa_m), & \text{if } i = j \\ r_m(\mathbf{x}_i; \kappa_m), & \forall j : \mathbf{x}_j = \mathbf{x}_i + \boldsymbol{\rho}_l - \boldsymbol{\rho}_k \\ & \text{with } R_m = (C_k, C_l) \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

When the number of possible system-level states $|\mathcal{X}_{\mathcal{N}}(\mathcal{D})|$ is small, it is possible to obtain a closed-form solution to Eq. (6). However, when $|\mathcal{X}_{\mathcal{N}}(\mathcal{D})|$ is large or even infinite, it may become computationally intractable to solve the system. In such cases, we can resort to Finite State Projection (FSP) methods [12] that approximate the solution by compressing the number of possible states (and, hence, also the size of $\mathbf{K}$). The idea of FSP is to expand the number of states dynamically, according their probabilities.

States with low probabilities are pruned, and, hence, only statistically relevant states are added to the domain of the solver. Finally, we compute the probability of the observable state $\mathbf{y}$ for a given time $\tau$, which we can then plug into our formula for differential privacy, Eq.(5). This is straightforward since $\pi_{\mathbf{x}}(\tau)$ is equivalent to $\mathbb{P}[\mathbf{x}(\tau)]$, and thus, $\mathbb{P}[q \circ f_{\mathcal{N}}(\mathbf{x_0}(\mathcal{D}), \tau)]$ is equivalent to $\pi_{\mathbf{y}}(\tau)$, where $\pi_{\mathbf{y}}(\tau) = \sum_{\forall \mathbf{x}\,\text{s.t.}\,\mathbf{y}=q(\mathbf{x})} \pi_{\mathbf{x}}(\tau)$.

### 4.2 Case Study

This case study considers the general problem of collaborative task solving, where robot species have distinct capabilities, and hence, depend on each other in order to complete tasks. Our system is composed of three species, $\mathcal{S} = \{1, 2, 3\}$. For any given task to be completed successfully, one robot of each species must be present at the respective task. There are a number of realistic scenarios that relate to this setting. A well-known work considers a setting where a homogeneous swarm of robots is tasked to pull sticks out of the ground [7] — because the length of a single robot's arm is limited, a successful manipulation requires two robots to collaborate. Our current case study can be formulated analogously by expanding the original statement to a heterogeneous setting. By default, all robots are in exploration mode, searching for tasks that need to be completed. A robot encounters tasks at a certain rate. Once it has encountered a task that is either unattended, or that is occupied by one of the other two species, it will wait at the task. The robot may abandon the task (with a given rate) before it is completed, or wait until all other robots from the other species join the task. If the robot abandons the task, it returns to exploration mode. If a robot encounters a task where both other species are already present, the three robots are able to collaborate and successfully complete the task. All three robots then return to exploration mode. We formalize this behavior with the following CRN:

$$\begin{array}{ll}
a^{\{1\}}_{(e)} \underset{r_2}{\overset{r_1}{\rightleftharpoons}} a^{\{1\}}_{(w)} & a^{\{2\}}_{(e)} + a^{\{3\}}_{(w)} \underset{r_{14}}{\overset{r_{13}}{\rightleftharpoons}} a^{\{2,3\}}_{(w)} \\
a^{\{2\}}_{(e)} \underset{r_4}{\overset{r_3}{\rightleftharpoons}} a^{\{2\}}_{(w)} & a^{\{3\}}_{(e)} + a^{\{1\}}_{(w)} \underset{r_{16}}{\overset{r_{15}}{\rightleftharpoons}} a^{\{1,3\}}_{(w)} \\
a^{\{3\}}_{(e)} \underset{r_6}{\overset{r_5}{\rightleftharpoons}} a^{\{3\}}_{(w)} & a^{\{3\}}_{(e)} + a^{\{2\}}_{(w)} \underset{r_{18}}{\overset{r_{17}}{\rightleftharpoons}} a^{\{2,3\}}_{(w)} \\
a^{\{1\}}_{(e)} + a^{\{2\}}_{(w)} \underset{r_8}{\overset{r_7}{\rightleftharpoons}} a^{\{1,2\}}_{(w)} & a^{\{1\}}_{(e)} + a^{\{2,3\}}_{(w)} \xrightarrow{r_{19}} a^{\{1\}}_{(e)} + a^{\{2\}}_{(e)} + a^{\{3\}}_{(e)} \\
a^{\{1\}}_{(e)} + a^{\{3\}}_{(w)} \underset{r_{10}}{\overset{r_9}{\rightleftharpoons}} a^{\{1,3\}}_{(w)} & a^{\{2\}}_{(e)} + a^{\{1,3\}}_{(w)} \xrightarrow{r_{20}} a^{\{1\}}_{(e)} + a^{\{2\}}_{(e)} + a^{\{3\}}_{(e)} \\
a^{\{2\}}_{(e)} + a^{\{1\}}_{(w)} \underset{r_{12}}{\overset{r_{11}}{\rightleftharpoons}} a^{\{1,2\}}_{(w)} & a^{\{3\}}_{(e)} + a^{\{1,2\}}_{(w)} \xrightarrow{r_{21}} a^{\{1\}}_{(e)} + a^{\{2\}}_{(e)} + a^{\{3\}}_{(e)}
\end{array} \tag{8}$$

The states of this system are

$$\begin{array}{ll}
\mathcal{A}^{(0)} = \emptyset & \mathcal{A}^{(2)} = \{a^{\{2\}}_{(e)}, a^{\{2\}}_{(w)}, a^{\{1,2\}}_{(w)}, a^{\{2,3\}}_{(w)}\} \\
\mathcal{A}^{(1)} = \{a^{\{1\}}_{(e)}, a^{\{1\}}_{(w)}, a^{\{1,2\}}_{(w)}, a^{\{1,3\}}_{(w)}\} & \mathcal{A}^{(3)} = \{a^{\{3\}}_{(e)}, a^{\{3\}}_{(w)}, a^{\{1,3\}}_{(w)}, a^{\{2,3\}}_{(w)}\}
\end{array}$$

with $\mathcal{A} = \cup_{s=\{0,1,2,3\}} \mathcal{A}^{(s)}$, and where $a^{\mathcal{I}}_{(e)}$ corresponds to the exploration state and $a^{\mathcal{I}}_{(w)}$ to the waiting state (e.g., $a^{\{1,3\}}_{(w)}$ corresponds to the state where one robot of species 1 and one robots of species 3 are waiting at a task for a robot of species 2). From the

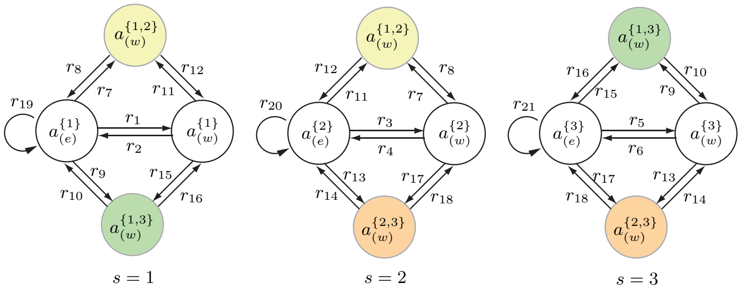

**Fig. 3.** Finite state machines of the three species in the case study presented in Sec. 4. The reactions correspond to the system detailed in Eq. (8). The interspecific states are colored.

reaction equations above, we see that robots interact when two robots are waiting for the remaining robot. Hence, the interspecific states are:

$$\mathcal{A}^{(1)} \cap A^{(2)} = \{a^{\{1,2\}}_{(w)}\}, \;\; \mathcal{A}^{(1)} \cap A^{(3)} = \{a^{\{1,3\}}_{(w)}\}, \;\; \mathcal{A}^{(2)} \cap A^{(3)} = \{a^{\{2,3\}}_{(w)}\} \tag{9}$$

Our population vector keeps track of the number of robots per state, and is written

$$\mathbf{x} = [x^{\{1\}}_{(e)}, x^{\{2\}}_{(e)}, x^{\{3\}}_{(e)}, x^{\{1\}}_{(w)}, x^{\{2\}}_{(w)}, x^{\{3\}}_{(w)}, x^{\{1,2\}}_{(w)}, x^{\{1,3\}}_{(w)}, x^{\{2,3\}}_{(w)}] \tag{10}$$

We consider an adversary who is able to observe the number of robots that are in exploration mode, the number of robots that are waiting alone, and the number of robots that are waiting in twos. Hence, we formulate the observable data as $\mathbf{y} = [y_1, y_2, y_3]$ and

$$\begin{aligned} y_1 &= x^{\{1\}}_{(e)} + x^{\{2\}}_{(e)} + x^{\{3\}}_{(e)} \\ y_2 &= x^{\{1\}}_{(w)} + x^{\{2\}}_{(w)} + x^{\{3\}}_{(w)} \\ y_3 &= x^{\{1,2\}}_{(w)} + x^{\{2,3\}}_{(w)} + x^{\{1,3\}}_{(w)} \end{aligned} \tag{11}$$

To illustrate this system, we depict the FSM of all species in Fig. 3. We note that the individual species' FSMs appear equivalent (with the exception of their rates). Yet, since by definition, we assume that distinct species own distinct capabilities, their roles in the respective states are complementary. The reactions' propensity rates can be attributed to the individual species. For instance, reaction $R_7$ is defined by the rate $\kappa_7$ at which species 1 encounters tasks at which species 2 is waiting. Hence, $\kappa_7$ is attributed to species 1. We define these values as $\boldsymbol{\kappa}^{(s)}$, and summarize them as $\boldsymbol{\kappa}^{(1)} = [\kappa_1, \kappa_2, \kappa_7, \kappa_8, \kappa_9, \kappa_{10}, \kappa_{19}]$, $\boldsymbol{\kappa}^{(2)} = [\kappa_3, \kappa_4, \kappa_{11}, \kappa_{12}, \kappa_{13}, \kappa_{14}, \kappa_{20}]$, and $\boldsymbol{\kappa}^{(3)} = [\kappa_5, \kappa_6, \kappa_{15}, \kappa_{16}, \kappa_{17}, \kappa_{18}, \kappa_{21}]$. As an example of the resulting dynamics, Fig. 4 shows the marginal distributions resulting from $\pi_{\mathbf{x}}(\tau)$, for all nine components $x_i$ of $\mathbf{x}$, and $\pi_{\mathbf{y}}(\tau)$, for all three components $y_i$ of $\mathbf{y}$.

### 4.3 Evaluation

A query that maintains a high level of privacy ensures that no individual can be isolated when information pertaining to that individual is changed (i.e., one element in the

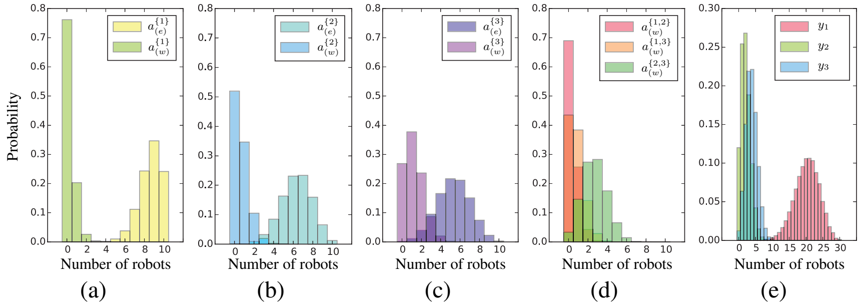

**Fig. 4.** Panels (a)-(d) show the marginal probability distributions resulting from $\pi_{\mathbf{x}}(\tau)$, for $\tau = 10$, for all nine components $x_i$ of $\mathbf{x}$. Similarly, panel (e) shows the marginal distribution of $\pi_{\mathbf{y}}(\tau)$ over the components $y_i$ of the observation data $\mathbf{y}$. The data is obtained for encountering rates $\kappa_1, \kappa_9, \kappa_{19} = 0.2$, $\kappa_3, \kappa_{13}, \kappa_{20} = 0.5$ and $\kappa_5, \kappa_{15}, \kappa_{21} = 0.8$, and abandoning rates $\kappa_2, \kappa_8, \kappa_{10} = 0.8$, $\kappa_4, \kappa_{12}, \kappa_{14} = 0.5$, and $\kappa_6, \kappa_{16}, \kappa_{18} = 0.2$, for species 1, 2, 3, respectively.

database is changed). Our query is defined by the observable state of the system. Hence, evaluating the privacy of a heterogeneous swarm with Eq. (5) is analogous to answering the following question: *'How much information is given to an observer when one robot is reallocated to another species?'*. Since our query is a function of the system-level state, it is defined by the number of robots per species $N^{(s)}$, and by propensity rates $\boldsymbol{\kappa}$. By varying these values, we can show how swarm composition and behavior affect privacy.

We evaluate the leakage of the system for the three different settings. First, we fix the propensity rates $\boldsymbol{\kappa} = 1$, and we vary the population $N^{(2)}$ and $N^{(3)}$ in the range $[1, \ldots, 20]$ with $N^{(3)} = 10$. Fig. 5(a) shows reduced leakage along the diagonal $N^{(1)} = N^{(2)}$. The minimum leakage occurs at $N^{(1)} = N^{(2)} = N^{(3)} = 10$. This indicates that species with equivalent observable behaviors should have a similar number of robots in order to maximize privacy. In other words, since species 1, 2, and 3 have equivalent FSMs with identical interspecific states (see Fig. 3), larger differences in the number of robots per species will produce more easily identifiable changes in the observable system-level state distributions. The plot also reveals that increasing the total number of robots increases privacy, as shown by the low leakage values in the upper right corner. Evidently, a system composed of many robots is more opaque (to an external observer): probability distributions spread over larger population ranges, and, thus, small differences in the initial population creates smaller differences in observable state distributions. In the second and third settings, we fix the population $N^{(1)} = N^{(2)} = N^{(3)} = 10$ and vary the propensity rates in the range $[0.2, 2]$ . Fig. 5(b) shows the leakage when we vary the rates at which species 2 ($\kappa_3, \kappa_{13}, \kappa_{20}$) and species 3 ($\kappa_5, \kappa_{15}, \kappa_{21}$) encounter tasks. Fig. 5(c) shows the leakage when we vary the rates at which species 2 ($\kappa_4, \kappa_{12}, \kappa_{14}$) and species 3 ($\kappa_6, \kappa_{16}, \kappa_{18}$) abandon tasks. If we program the species with the same rates, we obtain indiscernible behaviors, and hence, increase privacy. This is exemplified in the plots, where off-diagonal values exhibit higher leakage, and the minimum leakage value is situated at the cell corresponding to

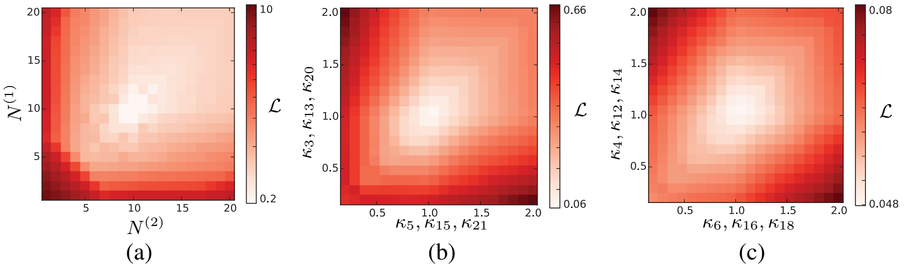

**Fig. 5.** Differential privacy of the collaboration case study. The colorbar shows the leakage. In (a), we vary the population of robots in species 2 and 3 while keeping species 1 fixed. In (b), we vary the task encountering rates of species 2 and 3, and in (c), we vary the task abandoning rates of species 2 and 3, while keeping the rates of species 1 fixed.

rate uniformity. Finally, we also note that for the considered parameter ranges, varying the number of robots per species has a much larger impact on privacy than varying the behavior.

## 5 Conclusion

In this work, we showed how the privacy of a heterogeneous swarm of robots can be analyzed. Our main contribution consists of a formal definition that couples the notion of differential privacy with a model of the robotic swarm. We showed how to evaluate the formula by considering a macroscopic description of the system-level state that can be computed analytically, if tractable, or numerically, with help of efficient computational methods. We evaluated our formula on a case-study of collaborative task-solving. This particular instance was well suited for demonstrative purposes (due to its very general and representative form). The framework is equally well suited for more involved cases where robot species exhibit widely varying behaviors. Our results show that we are able to determine how privacy levels vary as we vary the design parameters of the underlying swarm system.

Privacy is an urgent and important topic — systems that are capable of maintaining high levels of privacy are more secure and resilient. We intend to further this line of work by developing active privacy mechanisms that are able control the amount of information leaked, while maintaining overall system performance.

**Acknowledgments.** We gratefully acknowledge the support of ONR grants N00014-15-1-2115 and N00014-14-1-0510, ARL grant W911NF-08-2-0004, NSF grant IIS-1426840, and TerraSwarm, one of six centers of STARnet, a Semiconductor Research Corporation program sponsored by MARCO and DARPA.

# Bibliography

[1] Berman, S., Halasz, Á., Hsieh, M.A., Kumar, V.: Optimized Stochastic Policies for Task Allocation in Swarms of Robots. IEEE Transactions on Robotics 25, 927–937 (2009)

[2] Cardinale, B.J., Palmer, M.A., Collins, S.L.: Species diversity enhances ecosystem functioning through interspecific facilitation. Nature 415(6870), 426–429 (Jan 2002)

[3] Dorigo, M., Floreano, D., Gambardella, L.M., Mondada, F., Nolfi, S., et al.: Swarmanoid: A Novel Concept for the Study of Heterogeneous Robotic Swarms. IEEE Robotics & Automation Magazine 20(4), 60–71 (2013)

[4] Dwork, C.: Differential Privacy. Encyclopedia of Cryptography and Security pp. 338–340 (2011)

[5] Feinberg, M.: Some Recent Results in Chemical Reaction Network Theory. In: Patterns and Dynamics in Reactive Media, pp. 43–70. Springer New York, New York, NY (1991)

[6] Gillespie, D.T.: Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry A 25, 2340–2361 (1977)

[7] Ijspeert, A.J., Martinoli, A., Billard, A., Gambardella, L.M.: Collaboration through the exploitation of local interactions in autonomous collective robotics: the stick pulling experiment. Autonomous Robots 11, 149–171 (2001)

[8] Kasiviswanathan, S.P., Smith, A.: A note on differential privacy: Defining resistance to arbitrary side information. CoRR abs. (2008)

[9] Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A Survey of Research on Cloud Robotics and Automation. IEEE Transactions on Automation Science and Engineering 12(2), 398–409 (Apr 2015)

[10] López-Caamal, F., Marquez-Lago, T.T.: Exact Probability Distributions of Selected Species in Stochastic Chemical Reaction Networks. Bulletin of Mathematical Biology 76(9), 2334–2361 (Aug 2014)

[11] Michael, N., Fink, J., Loizou, S., Kumar, V.: Architecture, Abstractions, and Algorithms for Controlling Large Teams of Robots: Experimental Testbed and Results. In: Robotics Research; Springer Tracts in Advanced Robotics, pp. 409–419. Springer (2011)

[12] Munsky, B., Khammash, M.: The finite state projection algorithm for the solution of the chemical master equation. The Journal of chemical physics 124(4), 044104 (2006)

[13] Petchey, O.L., Gaston, K.J.: Functional diversity (FD), species richness and community composition. Ecology Letters pp. 402–411 (2002)

[14] Prorok, A., Hsieh, A.M., Kumar, V.: Formalizing the Impact of Diversity on Performance in a Heterogeneous Swarm of Robots. In: IEEE International Conference on Robotics and Automation (ICRA) (2016)

[15] Siegel, D., MacLean, D.: Global stability of complex balanced mechanisms. J. of Mathematical Chemistry 27, 89–110 (2000)