Differentially Private Iterative Synchronous Consensus*

Zhenqi Huang
zhuang25@illinois.edu

Sayan Mitra
mitras@illinois.edu

Geir Dullerud
dullerud@illinois.edu

Coordinate Science Laboratory
University of Illinois at Urbana Champaign
Urbana, IL 61801

ABSTRACT

The iterative consensus problem requires a set of processes or agents with different initial values, to interact and update their states to eventually converge to a common value. Protocols solving iterative consensus serve as building blocks in a variety of systems where distributed coordination is required for load balancing, data aggregation, sensor fusion, filtering, and synchronization. In this paper, we introduce the *private iterative consensus problem* where agents are required to converge while protecting the privacy of their initial values from honest but curious adversaries. Protecting the initial states, in many applications, suffice to protect all subsequent states of the individual participants.

We adapt the notion of differential privacy in this setting of iterative computation. Next, we present (i) a server-based and (ii) a completely distributed randomized mechanism for solving differentially private iterative consensus with adversaries who can observe the messages as well as the internal states of the server and a subset of the clients. Our analysis establishes the tradeoff between privacy and the accuracy: for given $\epsilon, b > 0$, the ϵ-differentially private mechanism for N agents, is guaranteed to convergence to a value within $O(\frac{1}{\epsilon\sqrt{bN}})$ of the average of the initial values, with probability at least $(1 - b)$.

Categories and Subject Descriptors

C.2.4 [**Computer Systems Organization**]: Computer and Communication Networks—*Distributed Systems*

General Terms

Theory and Security

*This work is supported by the Army Research Office under Award No. W911NF-09-1-0273. Dr. Mitra was also supported by AFOSR Young Investigator Award.

WPES'12, October 15, 2012, Raleigh, North Carolina, USA.

Keywords

Iterative consensus; privacy; convergence; distributed agreement.

1. INTRODUCTION

This paper addresses the problem of iteratively reaching agreement in a group while preserving the individual's privacy. The setup consists of N agents, each with some initial information modeled as the valuation of a variable. The problem requires the agents to interact with each other and update their internal states incrementally, so that eventually they all converge to a common decision or value. This agreement to a common decision can then be used for coordinating the actions of the participating agents. Indeed, solutions to this iterative consensus problem has been used as a building block for designing a variety of distributed coordination protocols for load balancing [7, 30], filtering and sensor fusion [22, 31], clock synchronization, and flocking [4, 25, 16, 26, 17], to name a few.

A natural, synchronous, and widely studied consensus mechanism involves, at each round, for every agent to update its state as a weighted average of its neighboring agents'. This update rule can be expressed as $x(t + 1) = Px(t)$, where $x(t)$ is the vector of agent values and P is a symmetric $N \times N$ matrix with P_{ij} defining the communication weight between agents i and j. It turns out that this class of consensus mechanisms[1] converge to the average of the initial values of the agents. More general necessary and sufficient conditions for achieving consensus with synchronous mechanisms, including cases where the matrix P is time-varying, have been studied in [28, 23] (see the book for a complete overview [20]). Sufficient conditions for achieving consensus with message delays and losses has been developed in [29, 3] and more recently, a theorem prover-based verification framework for these mechanisms has been presented in [21, 5]. Furthermore, stochastic variants of the convergence mechanism under the presence of communication noises has been studied in [30, 15].

In this paper we study the *private consensus* problem which requires the agents to preserve the privacy of their initial values from an adversary who can see all the messages being exchanged, while also achieving convergence to the average of the initial values. This is motivated by scenarios where each individual agent is a closed-loop control system evolving according to both its own dynamics and

[1]We refrain from calling these mechanisms algorithms because they are designed to converge and not to terminate.

preferences, as well as the state of the world. For instance (a) a smart GPS device in the car which reacts to traffic, (b) a smart-electric meter that schedules appliances in the home depending on the dynamic cost of electricity. Initially, each individual agent has a set of values (waypoints for cars) which only depend on the individual's goals and are oblivious of the world. As the system evolves, some of the local variables change to capture the publicly known aggregate dynamics (e.g. traffic distribution, average demand of electricity), while other local variables capture the current state of the individual. We illustrate with a simple example later that if the initial individual preferences are protected, then even with the knowledge of the aggregate dynamics, the current state of the individuals cannot be estimated accurately.

The notion of privacy used in this paper is derived from the idea of *differential privacy* introduced in the context of "one-shot" computations on statistical databases [9] (see [10] for a survey). Roughly speaking, differential privacy ensures that the participation of a single agent in a database does not affect the output of any analysis *substantially*. It follows that an adversary looking at the output of any analysis cannot threaten to breach the privacy and security of the individual participants.

In [12], the notion of differential privacy is expanded along two dimensions. First, it included streaming and online computations in which the adversary can look at the entire sequence of outputs from the analysis algorithm. Secondly, it allowed the adversary to look at the internal state of the algorithm (Pan privacy) in addition to the communication messages. In this paper, we are concerned with protecting the privacy of the initial value of an agent instead of its (binary) participation status. Consequently, like the definitions presented in [14, 24], differential privacy is defined here in terms of adjacent states that are identical for all agents excepting one agent whose values are close (as measured by a metric on its real-valued variables). This notion of differential privacy guarantees that two sets of behaviors, starting from two adjacent initial states and corresponding to any output sequence, are statistically close.

As a starting point in this investigation, we use a client-server setup for iterative consensus. The clients are the agents with private initial values. In each round, the clients send some information to the server based on their current state. Then, the server updates its own state based on clients' information and sends feedback to the clients. Finally, the clients update their state according to a local control law based on the server's feedback. The clients require to converge, while their initial values should be protected from any honest but curious adversary with access to the messages (between the clients and the server) as well as the server's internal state. We call this the *Synchronous Private Consensus (SPC)* problem.

In distributed control systems, protected initial information often implies protection of the current state. For example, consider a platoon of vehicles which require to move as a group with the same speed, while keeping their positions private. If the agents use a solution to the SPC problem for deciding on the common speed, then their initial velocities as well as their current positions will be protected, even if their initial positions and control laws are compromised. Protecting the initial velocities suffice because to obtain the current position $x(T)$ at time T, from the initial position $x(0)$, one has to integrate $v(t)$ over $[0, T]$. Even though for large t, $v(t)$ will be close to the average velocity of the group, but the error in the initial part of this integral makes the estimation error for $x(T)$ to be large. The same argument holds for any control system where the aggregate values (average velocity, traffic, demand, etc.) serve as an input to the individual's dynamics.

In Section 3 we propose a randomized mechanism for solving the SPC problem. The key idea is to add a particular type of random noise to the clients' messages to the server. In contrast to the various mechanisms proposed earlier [13, 12], our mechanism adds a stream of noise drawing from a time-varying distribution. Specifically, for a client with internal state $\theta(t)$ at round t, the message it sends to the server is $\theta(t) + \eta(t)$ where $\eta(t)$ is a random (real) number chosen according to a Laplace distribution with a parameter that decays geometrically with t. The feedback $y(t)$ provided by the server is the mean of the noisy messages it receives. And, the clients update their states by taking a linear combination of $y(t)$ and their earlier state. This weighted average is an example of a simple type of client dynamics.

In Section 4, we generalize the client-server mechanism to a distributed setting where the adversary can access the messages and the states of a subset of compromised clients. The mechanism guarantees differential privacy of the good clients and we derive a sufficient condition for convergence based on the communication and update pattern of the clients.

As randomization is used for achieving privacy, this mechanism guarantees convergence to the average in a probabilistic sense: Given a probability b and a radius r, we say that the mechanism is (b, r)-*accurate* if from any initial state, with probability $(1 - b)$ the system converges to a value within r distance of the average. In Section 4.2, we discuss the tradeoff between privacy and accuracy realized by our proposed mechanisms. There are two parameters in the definition of the mechanism which can be chosen to get different levels of privacy and accuracy. If these parameters are tuned to obtain ϵ-differential privacy, then we show that the accuracy that can be achieved is $(b, O(\frac{1}{\epsilon\sqrt{bN}}))$. That is, the accuracy radius depends inversely on the privacy level (ϵ) , the square root of the number of agents (N), and in the probability (b).

The rest of the paper is organized as follows. In Section 2, we introduce the synchronous private consensus problem, and then formally define differential privacy, convergence, and accuracy. In Sections 3 and 4, we present and analyze the client-server and the distributed mechanisms for SPC. In Section 5, we compare our work with existing research papers in this area. In Section 6, we summarize our results and discuss possible future directions.

2. PRELIMINARIES

For a natural number $N \in \mathbb{N}$, we denote the set $\{1, \ldots, N\}$ by $[N]$. For an S-valued vector θ of length N, and $i \in [N]$, we denote the i^{th} component by θ_i.

The mechanisms presented in this paper rely on random real numbers drawn according to the Laplace distribution. $Lap(b)$ denotes the Laplace distribution with probability density given by $p_L(x|b) = \frac{1}{2b}e^{-|x|/b}$. This distribution has mean 0 and variance $2b^2$. For any $x, y \in \mathbb{R}$, $\frac{p_L(x|b)}{p_L(y|b)} \leq e^{\frac{|x-y|}{b}}$.

2.1 Problem Statement

We state the *synchronous private consensus (SPC)* problem in the following setting. The system consists of N *clients* with private initial values $\theta_1(0), \ldots, \theta_N(0)$ and one *server*. The clients and the server may have internal states and they communicate over channels. In each round, there are four phases: First, the clients send some messages to the server; next, the server performs computations to update its state; then it responds to the clients with some messages, and finally, the clients smoothly update their own internal states based on the response from the server.

Several vulnerabilities threaten to compromise the private initial values of the clients: An intruder can have full access to all the communication channels. That is, he can peek inside all the messages going back and forth between the clients and the server. Furthermore, the intruder can access the server's internal state.

Roughly, a randomized mechanism for the clients and the server solves the synchronous private consensus problem if eventually all the clients converge to the average of their initial values with high probability and it guarantees that the intruder cannot learn about the initial private client values with any high level of confidence. We proceed to precisely define accuracy, convergence, and privacy.

Our definition of privacy is a modification of the notion of *differential privacy* introduced in [12] in the context of streaming algorithms. A state of an agent is typically defined by a valuation for each of its local variables. Let $\Theta \subseteq \mathbb{R}$ be the set of states of an individual agent or client. For a system with N agents, Θ^N is the set of collective agent states. Each elementary step or iteration of a syncrhonous mechanism transforms one collective state to another. During one such step, the agent sends messages to the server, the server updates its own state, and the server responds to the clients. These messages and the state of server constitute the observable part of the state transition. We will study *executions* of synchronous mechanisms which are (possibly infinite) sequence of elementary steps. The orbservable part of such an execution are the corresponding (possibly infinite) sequence of messeages and server states.

More formally, for each mechanism **Alg** there is a map $\mathbf{Alg} : \Theta^N \mapsto X \times Y$ from an initial collective state to a sequence of observations. Here X is the set of all (possibly infinite) sequences of messages and Y is the set of all (possibly infinite) sequences of internal server states. Concrete definitions of these objects will be given in Section 3 where we introduce our mechanism.

Definition 1 (Adjacency). *Two vectors $\theta, \theta' \in \Theta^N$ are δ-*adjacent*, for some $\delta \geq 0$, if there exists one $i \in [N]$, such that $|\theta_i - \theta'_i| \leq \delta$ and for all $j \neq i$, $\theta_j = \theta'_j$.*

Definition 2 (Differential Privacy). *Let $\Theta^N \subset \mathbb{R}^N$ be the domain of global state. A randomized mechanism **Alg** preserves* ϵ-differential privacy *if for all sets $X' \subseteq X$ and $Y' \subseteq Y$, and for all pairs of* δ-adjacent *initial global states $\theta, \theta' \in \Theta^N$*

$$Pr[\mathbf{Alg}(\theta) \in (X', Y')] \leq e^{\epsilon\delta} Pr[\mathbf{Alg}(\theta') \in (X', Y')].$$

This definition of adjacency uses a 1-norm whereas the standard definition (found in [11], for example) uses the Hamming distance. This choice of the metric has ramifications on the privacy guarantees. In cases where each agent's local value comes from a bounded set, by letting δ equals to the range of local value, Definition 2 subsumes the standard definition. In cases where each agent's local value comes from an unbounded set, the sensitivity of a query can be unbounded. In such cases, the mechanisms introduced in this paper fail to provide differential privacy (in the sense of [11]) and the δ-adjacency notion becomes useful.

We use the mean square notion of convergence which has been used in the context of consensus protocols [15]. Let $\theta_i(t) \in \Theta$ be the local states of agent A_i at the beginning of round t. $\theta_i(0)$ denotes the secret initial state of A_i.

Definition 3 (Convergence). *A randomized mechanism is said to converge if for any initial configuration, for any $i, j \in [N]$, $\lim_{t\to\infty} E[(\theta_i(t) - \theta_j(t))^2] = 0$, where the expectation is over the coin-flips of the algorithm.*

Definition 4 (Accuracy). *For any initial state $\theta(0)$, $b \in [0,1]$ and $r \in \mathbb{R}_{\geq 0}$ a randomized mechanism is said to achieve* (b, r)-accuracy *if every execution starting from $\theta(0)$ converges to a state within r of $\frac{1}{N}\sum_i \theta_i(0)$, with probability at least $1 - b$.*

By Definition 2, a smaller ϵ implies a stronger guarantee of privacy. On the other hand, by Definition 4, to enjoy a higher level of accuracy, a smaller probability b and a smaller radius r are favorable.

Our goal is to design a solution to the SPC problem that is guaranteed to converge. In addition, for an adversary, looking at all the sequence of messages passing through the channels as well as the sequence of internal states of the server (and possibly some of the clients), the probabilities of two sets of executions starting from teo adjacent initial states and corresponding to these observations, have to satisfy the Equation in Definition 2.

3. A CLIENT-SERVER MECHANISM AND ITS ANALYSIS

In this section, we present a randomized mechanism to solve the synchronous private consensus problem. This mechanism has three parameters $\sigma \in (0,1)$, c and $q \in (0,1)$. The mechanism is specified by the following client and server actions which define the four phases of each round. Let $\mathbb{T} = \{0\} \cup \mathbb{N}$ be the infinite time domain. At each round $t \in \mathbb{T}$:

(i) Client i sends a message $x_i(t) = \theta_i(t) + \eta_i(t)$ to the server, where $\eta_i(t)$ is a random noise generated from the distribution $Lap(cq^t)$.

(ii) The server updates its own state as the average of all client messages $y(t) = \frac{1}{N}\sum_i x_i(t)$.

(iii) The server sends $y(t)$ to all clients.

(iv) Client i updates its state by linearly interpolating between $\theta_i(t)$ and $y(t)$ with coefficient σ, that is,

$$\theta_i(t+1) = (1-\sigma)\theta_i(t) + \sigma y(t). \quad (1)$$

3.1 Analysis

For $t \in \mathbb{T}$, let $\theta(t) = [\theta_1(t), \ldots, \theta_N(t)]^T$ be the vector defining the state of the clients at the beginning of round t. Similarly, $\eta(t)$ and $x(t)$ are vectors for noise and messages. An *execution* of the mechanism is an infinite sequence of the form $\alpha = \theta(0), (\eta(0), x(0), y(0)), \theta(1), (\eta(1), x(1), y(1)), \ldots$. Observe that given an initial vector $\theta(0)$ and the sequence of noise vectors $\eta(0), \eta(1), \ldots$, the execution of the system is completely specified. That is, for all $t \in \mathbb{T}$, it defines the messages $x(t), y(t)$, the internal states of the clients $\theta(t)$ and that of the server $y(t)$. Thus, for brevity we will sometimes write an execution α as an infinite sequence of the form $\theta(0), \eta(0), \theta(1), \eta(1), \ldots$. The prefix of α upto round $T \in \mathbb{T}$ is denoted by α_T. We denote the set of possible executions from $\theta(0)$ as $\mathsf{Execs}_{\theta(0)}$.

For a given execution α, the adversary can observe the subsequence of messages $x(t), y(t)$ and the server's state $y(t)$. We denote this subsequence by $\alpha \downarrow (x, y)$. Hence, two executions α and α' are indistinguishable to an adversary if $\alpha \downarrow (x, y) = \alpha' \downarrow (x, y)$. For a set of observation sequnces Obs, the set of all possible executions from $\theta(0)$ which correspond to some observation in Obs is the set $\mathsf{Execs}_{\theta(0),Obs} \triangleq \{\alpha \in \mathsf{Execs}_{\theta(0)} | \alpha \downarrow (x, y) \in Obs\}$. We restate the definition of differential privacy in this context.

Definition 5 (Differential Privacy). *A randomized mechanism preserves* ϵ-differential privacy *if for any set of observation sequnces Obs, and any pairs of δ-adjacent initial global states $\theta(0), \theta'(0) \in \Theta^N$*

$$Pr[\mathsf{Execs}_{\theta(0),Obs}] \leq e^{\epsilon\delta} Pr[\mathsf{Execs}_{\theta'(0),Obs}]. \quad (2)$$

Lemma 1 (Privacy). *For $q \in (1-\sigma, 1)$, the mechanism guarantees ϵ-differential privacy with $\epsilon = \frac{q}{c(q+\sigma-1)}$.*

PROOF. Let $\theta(0)$ and $\theta'(0)$ be arbitrary δ-adjacent initial global states. Without loss of generality, we assume that for some $k \in [N]$, $\theta_k(0) = \theta'_k(0) + \delta$. Fix any subset of observation sequences Obs. We will show that Equation (2) holds by establishing a bijective correspondence between the executions in $\mathsf{Execs}_{\theta(0),Obs}$ and $\mathsf{Execs}_{\theta'(0),Obs}$. For brevity, we denote these sets by A and A'.

First, we define a bijection $f : A \mapsto A'$. For $\alpha \in A$ defined by the sequence $\theta(0), \eta(0), \eta(1), \ldots$, we define $f(\alpha) \triangleq \theta'(0), (\eta'(0), x'(0), y'(0)), \theta'(1), (\eta'(1), x'(1), y'(1)), \theta'(2), \ldots$, where for each $t \in \mathbb{T}$,

$$\eta'_i(t) = \begin{cases} \eta_i(t) + \delta(1-\sigma)^t & \text{for } i = k, \\ \eta_i(t) & \text{otherwise.} \end{cases}$$

$x'(t) = \theta'(t) + \eta'(t)$, $y'(t) = \frac{1}{N}\sum_{i \in [N]} x'(t)$, and for $t > 0$ $\theta'(t) = (1-\sigma)\theta'(t-1) + \sigma y'(t)$. Clearly, $f(\alpha)$ is a valid execution of the mechanism staring from $\theta'(0)$.

The following proposition relates the states and the observable vectors of two corresponding executions.

Proposition 2. *For all $t \in \mathbb{T}$, $i \in [N]$,*

(i) $\theta_k(t) - \theta'_k(t) = \delta(1-\sigma)^t$,

(ii) $\theta_i(t) = \theta'_i(t), \forall i \neq k$

(iii) $x'_i(t) = x_i(t)$,

(iv) $y'(t) = y(t)$.

PROOF. The proof is by induction on t. For the base case $t = 0$, observe that for $i = k$, $x'_i(0) = \theta'_i(0) + \eta'_i(0) = \theta_i(0) - \delta + \eta_i(0) + \delta = x_i(0)$, otherwise, $x'_i(0) = \theta'_i(0) + \eta'_i(0) = \theta_i(0) + \eta_i(0) = x_i(0)$;

For the inductive step, assume that the proposition holds for all $t \leq T$. From Equation 1, we have $\theta'_k(T+1) = (1-\sigma)\theta'_k(T) + \sigma y'(T)$ and $\theta_k(T+1) = (1-\sigma)\theta_k(T) + \sigma y(T)$. The difference of these two equation gives $\theta'_k(T+1) - \theta_k(T+1)$

$$\begin{aligned} &= (1-\sigma)(\theta'_k(T) - \theta_k(T)) + \sigma(y'(T) - y(T)) \\ &= (1-\sigma)(\theta'_k(T) - \theta_k(T)) = \delta(1-\sigma)^{T+1}. \end{aligned}$$

For any other client $i \neq k$, immediately from that $y'(T) = y(T)$ and $\theta'_i(T) = \theta_i(T)$, we have $\theta_i(T+1) = \theta_i(T+1)$.

Now we consider the clients' reports $x(T+1)$. For the k^{th} client, $x'_k(T+1) = \theta'_k(T+1) + \eta'_k(T+1) = \theta_k(T+1) - \delta(1-\sigma)^{T+1} + \eta_k(T+1) + \delta(1-\sigma)^{T+1} = x_k(T+1)$. For the other client $i \neq k$, $x'_i(T+1) = \theta'_i(T+1) + \eta'_i(T+1) = \theta_i(T+1) + \eta_i(T+1) = x_i(T+1)$. So the reports $x'(T+1) = x(T+1)$. The match up of the server's internal state immediately follows.

Parts (*iii*) and (*iv*) of the above proposition establishes that α and $f(\alpha)$ are indistinguishable, that is, indeed they produce the same observation sequence.

Next we will relate the probability of any finite prefix of an *individual* execution $\alpha \in A$, and its corresponding execution $f(\alpha) \in A'$, for a particular observation sequence $\beta \in Obs$:

$$\begin{aligned} &\frac{Pr[\alpha_T = \theta(0), \ldots, \theta(T)]}{Pr[(f(\alpha))_T = \theta(0), \ldots, \theta(T)]} \\ = &\prod_{t=0}^{T-1} \prod_{i \in [N]} \frac{p_L(\eta'_i(t)|cq^t)}{p_L(\eta_i(t)|cq^t)} = \prod_{t=0}^{T-1} \frac{p_L(\eta'_k(t)|cq^t)}{p_L(\eta_k(t)|cq^t)} \\ \leq &\prod_{t=0}^{T-1} e^{\frac{|\eta'_k(t) - \eta_k(t)|}{cq^t}} = \prod_{t=0}^{T-1} e^{\frac{\delta}{c}\left(\frac{1-\sigma}{q}\right)^t}. \end{aligned}$$

Integrating over all executions $\alpha \in A$, we get

$$\begin{aligned} &\int_{\alpha \in A} Pr[\alpha_T = \theta(0), \ldots, \theta(T)] d\mu \\ \leq &\prod_{t=0}^{T-1} e^{\frac{\delta}{c}\left(\frac{1-\sigma}{q}\right)^t} \int_{f(\alpha) \in A'} Pr[(f(\alpha))_T = \theta'(0), \ldots, \theta'(T)] d\mu', \end{aligned}$$

where $d\mu$ and $d\mu'$ are probability measures over A and A' defined by the randomized mechanism. If $q \in (1-\sigma, 1)$, then as $T \to \infty$, the product converges to $e^{\epsilon\delta}$, where $\epsilon = \frac{q}{c(q+\sigma-1)}$, and we obtain the required inequality for ϵ-differential privacy.

$$Pr[\mathsf{Execs}_{\theta(0),Obs}] \leq e^{\epsilon\delta} Pr[\mathsf{Execs}_{\theta'(0),Obs}].$$

Lemma 3 (Convergence). *The mechanism described above achieves convergence.*

PROOF. We define a global potential function $P : \mathbb{N} \to \mathbb{R}_{\geq 0}$ as $P(t) = \frac{1}{2}\sum_{i \neq j}[\theta_i(t) - \theta_j(t)]^2$. Using the matrix notation $P(t) = \theta(t)^T \, L \, \theta(t)$, where $L \in \mathbb{R}^{N \times N}$ with elements:

$$l(i,j) = \begin{cases} N-1, & i = j, \\ -1, & \text{otherwise.} \end{cases} \quad (3)$$

The transition rule for the internal state of the i^{th} client can be written as:

$$\begin{aligned} \theta_i(t+1) &= (1-\sigma)\theta_i(t) + \tfrac{\sigma}{N}\textstyle\sum_{i=1}^N(\theta_i(t) + \eta_i(t)) \\ &= (1 + \tfrac{\sigma}{N} - \sigma)\theta_i(t) + \tfrac{\sigma}{N}\textstyle\sum_{j \neq i}\theta_j(t) + \tfrac{\sigma}{N}w(t), \end{aligned} \quad (4)$$

where $w(t) = \sum_i \eta_i(t)$. The update rule for all the agents can be written as $\theta(t+1) = \theta(t) - \frac{\sigma}{N}L\theta(t) + \frac{\sigma}{N}w(t)\mathbb{1}_N$.

Then,

$$\begin{aligned} P(t+1) &= \theta(t+1)^T L\theta(t+1) \\ &= [\theta(t) - \tfrac{\sigma}{N}L\theta(t) + \tfrac{\sigma}{N}w(t)\mathbb{1}_N]^T L \\ &\quad [\theta(t) - \tfrac{\sigma}{N}L\theta(t) + \tfrac{\sigma}{N}w(t)\mathbb{1}_N] \\ &= P(t) - 2\tfrac{\sigma}{N}\theta(t)^T LL\theta(t) + \tfrac{\sigma^2}{N^2}\theta(t)^T LLL\theta(t) \\ &\quad +2\tfrac{\sigma}{N}w(t)\theta(t)^T L\mathbb{1}_N - \tfrac{2\sigma^2}{N^2}w(t)\theta(t)^T LL\mathbb{1}_N \\ &\quad +\tfrac{\sigma^2}{N^2}w(t)^2\mathbb{1}_N^T L\mathbb{1}_N. \\ &= P(t) - 2\tfrac{\sigma}{N}\theta(t)^T LL\theta(t) + \tfrac{\sigma^2}{N^2}\theta(t)^T LLL\theta(t). \end{aligned} \quad (5)$$

By Equation 3 we have $L = NI_N - \mathbb{1}_{NN}$. So in this particular case, we have $LL = (NI_N - \mathbb{1}_{NN})^2 = N^2 I_N - 2N\mathbb{1}_{NN} + \mathbb{1}_{NN}^2 = N^2 I_N - N\mathbb{1}_{NN} = NL$. Similarly $LLL = N^2 L$. Substitute the previous equation into Equation (5) we get,

$$P(t+1) = (1 - 2\sigma + \sigma^2)P(t) = aP(t),$$

where $a = (1-\sigma)^2$. For all $\sigma \in (0,1)$, $a \in (0,1)$ is a constant. Thus we have as $t \to \infty$, $P(t)$ converges exponentially to 0, which implies convergence.

From Equation (4), each agent adds an *identical* random variable $\frac{\sigma}{N}w(t)$ to its local state in round t. Although the average value drifts with this random variable, the relative distance between local states will not be affected. As a result, the mechanism converges deterministically.

Lemma 4 (Accuracy)**.** *For any $b \in (0,1)$, the randomized mechanism achieves $(b, \frac{\sqrt{2}c\sigma}{\sqrt{bN(1-q^2)}})$)-accuracy.*

PROOF. This is a special case of a more general proof we show later. Please see the proof of Lemma 8 with $\tilde{d}$ set to $\frac{\sigma^2}{N}$ for this case.

In this section we proposed a solution to the synchronous private consensus problem (SPC) with a server and formally established its privacy, convergence and accuracy properties. We will discuss the trade-offs between privacy and accuracy in Section 4.2.

4. A DISTRIBUTED MECHANISM

In this section, we present a second synchronous randomized mechanism for solving the private consensus problem which does not use a server but instead relies on the clients exchanging information with their neighbors in a truly distributed fashion. Let $G = ([N], \mathcal{E})$ be a *undirected connected graph*, where $[N]$ is the set of *vertices* and $\mathcal{E} \subset [N] \times [N]$ is the set of *edges*. Let $N(i) = \{j \in [N] | (i,j) \in \mathcal{E}\}$ be the set of *neighbors* of node i with whom it communicates. Let $|N(i)|$ be the *degree* of node i in G.

As in the previous setting, an intruder has access to all the communication channels as well as the internal states of a set C of compromised clients (but cannot overwrite them). Our mechanism will protect the privacy of clients who are not compromised. Thus, in this context, Definition 5 is modified by restricting the notion of δ-adjacency to uncompromised agents.

Now we state a mechanism to solve the distributed SPC problem. Besides the state variable θ_i which holds the consensus value, client i holds another auxiliary state y_i. The mechanism has parameters $\sigma \in (0,1)^N$, c and $q \in (0,1)$. Instead of sharing an identical linear combination factor, client i has an independent $\sigma_i \in (0,1)$ which is the i^{th} element of vector σ. At each round $t \in \mathbb{T}$:

(i) Client i sends a message $x_i(t) = \theta_i(t) + \eta_i(t)$ to every $j \in N(i)$, where $\eta_i(t)$ is a random noise generated from the distribution $Lap(cq^t)$.

(ii) Client i updates y_i as the average of $x_i(t)$ and the messages it receives:

$$y_i(t) = \frac{1}{|N(i)|+1} \sum_{j \in N(i)\cup\{i\}} x_j(t). \quad (6)$$

(iii) Client i updates θ_i by linearly interpolating between $\theta_i(t)$ and $y_i(t)$ with coefficient σ_i, that is,

$$\theta_i(t+1) = (1-\sigma_i)\theta_i(t) + \sigma_i y_i(t). \quad (7)$$

4.1 Analysis

The analysis of the distributed mechanism parallels the analysis presented in Section 3. An execution α is defined similar to the centralized setting except that $y(t)$ in this case is a vector rather than a scaler. The privacy of those corrupted nodes makes no sense. Let $C \subset [N]$ be the set of corrupted nodes.

Lemma 5 (Privacy)**.** *For $q \in (1-\sigma_m, 1)$, where σ_m is the minimum element of vector σ, the distributed mechanism guarantees ϵ-differential privacy with respect to the uncorrupted nodes with $\epsilon = \frac{q}{c(q+\sigma_m-1)}$.*

We omit the proof of Lemma 5 as it is a straight forward generalization of the proof of Lemma 1.

In contrast to Lemma 3, the convergence of the distributed mechanism depends on the structure of graph G. Before stating the convergence result, we introduce Laplacian matrix L of graph G with elements:

$$l(i,j) = \begin{cases} |N(i)| & i = j, \\ -1 & (i,j) \in \mathcal{E}, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

The Laplacian matrix L for any graph is known to have several nice properties. It is by definition symmetric with real entries, hence it can be diagonalized by an orthogonal matrix. It is positive semidefinite, hence its real eigenvalues can be ordered as $\lambda_1 \le \lambda_2 \le \ldots \le \lambda_N$ be the eigenvalues of L. Let $\{v_1, v_2, \ldots, v_N\}$ be a set of orthonormal eigenvectors of L such that v_k corresponds to λ_k. In addition, denote $d_i = \frac{\sigma_i}{|N(i)|+1}$. We state a sufficient condition of convergence as following.

Assumption 1. *Assume that graph G has the following properties.*

(I) $\lambda_2 > 0$, that is graph G is connected.

(II) $\lambda_N < \frac{2m}{M^2}$, where $m = \inf_{i\in[N]} d_i$ and $M = \sup_{i\in[N]} d_i$.

$\lambda_2 > 0$ if and only if the graph is connected. For a fixed λ_2, a smaller λ_N gives a smaller upper bound on the diameter of the graph (see [6]).

Lemma 6 (Convergence)**.** *The distributed mechanism described above achieves convergence if Assumption 1 holds.*

PROOF. We define a function $P : \mathbb{N} \mapsto \mathbb{R}_{\ge 0}$ as

$$P(t) = \frac{1}{2} \sum_{(i,j)\in\mathcal{E}} [\theta_i(t) - \theta_j(t)]^2.$$

Using the matrix notation $P(t) = \theta(t)^T L\, \theta(t)$. By Assumption 1, $E[P(t)] = 0 \Leftrightarrow \sum_{i \neq j} E[\theta_i(t) - \theta_j(t)]^2 = 0$. According to Equation (6) and (7), the update equation of client i is:

$$\begin{array}{rl} \theta_i(t+1) = & (1 - d_i|N(i)|)\theta_i(t) + d_i \sum_{j \in N(i)} \theta_j(t) \\ & + d_i w_i(t), \end{array} \tag{9}$$

where

$$w_i(t) = \sum_{j \in N(i) \cup \{i\}} \eta_i(t). \tag{10}$$

We define vector $w(t) = [w_1(t), \ldots, w_N(t)]^T$ and matrix $D \in \mathbb{R}^{N \times N}$ with elements:

$$d(i,j) = \left\{ \begin{array}{ll} d_i, & i = j, \\ 0, & \text{otherwise}. \end{array} \right. \tag{11}$$

The update rule for all the agents can be written as $\theta(t+1) = \theta(t) - DL\theta(t) + Dw(t)$. Then, $P(t+1)$

$$\begin{array}{rl} = & \theta(t+1)^T L \theta(t+1) \\ = & (\theta(t) - DL\theta(t) + Dw(t))^T L(\theta(t) - DL\theta(t) + Dw(t)) \\ = & P(t) - 2\theta(t)^T LDL\theta(t) + \theta(t)^T LDLDL\theta(t) + \\ & 2\theta(t)^T (I - DL)LDw(t) + w(t)^T DLDw(t). \end{array} \tag{12}$$

Taking expectation of both sides with respect to the coin flips of the algorithm starting from any state:

$$E[P(t+1)] \quad = E[P(t)] - E[Q(\theta(t))] + E[w(t)^T DLDw(t)], \tag{13}$$

where,

$$Q(\theta) = 2\theta^T LDL\theta - \theta^T LDLDL\theta.$$

The term $E[2\theta(t)^T (I - DL)LDw(t)]$ vanishes because (i) $\theta(t)$ and $w(t)$ are independent; and (ii) by Equation (10), $w(t)$ has zero mean.

Now we will prove that there exists a constant $a \in (0,1)$ such that $Q(\theta(t)) \geq aP(t)$. Because L is positive semidefinite, we have $0 \leq L \leq \lambda_N I$. From Assumption 1 and Equation (11), we have $mI \leq D \leq MI$. Then,

$$\begin{array}{rl} Q(\theta) & \geq 2m\theta^T LL\theta - \lambda_N \theta^T LDDL\theta \\ & \geq 2m\theta^T LL\theta - \lambda_N M^2 \theta^T LL\theta \\ & \geq (2m - \lambda_N M^2)\theta^T LL\theta. \end{array} \tag{14}$$

The following proposition helps obtain a bound on a.

Proposition 7. *For any $\theta \in \mathbb{R}^N$, $\theta^T LL\theta \geq \lambda_2 \theta^T L\theta$.*

PROOF. First, we show that the proposition holds for any eigenvector v_k of L. For the eigenvector v_1 corresponding to $\lambda_1 = 0$, we have $v_1^T L = 0$ and the inequality holds trivially. For any other eigenvector v_k and the corresponding eigenvalue $\lambda_k > 0$, we have $v_k^T LLv_k = \lambda_k v_k^T Lv_k \geq \lambda_2 v_k^T Lv_k$. Next, we prove that the proposition holds for any vector θ. Because $\{v_1, v_2, \ldots, v_N\}$ is an orthonormal basis, for any $i \neq j$, $v_i^T LLv_j = \lambda_j v_i^T Lv_j = \lambda_j^2 v_i^T v_j = 0$. For any $\theta = \sum_{k \in [N]} \alpha_k v_k$, we have:

$$\begin{array}{rl} \theta^T LL\theta & = (\sum_{k \in [N]} \alpha_k v_k)^T LL(\sum_{k \in [N]} \alpha_k v_k) \\ & = \sum_{k \in [N]} \alpha_k^2 v_k^T LLv_k \\ & \geq \lambda_2 \sum_{k \in [N]} \alpha_k^2 v_k^T Lv_k = \lambda_2 \theta^T L\theta. \end{array}$$

From Equation (14), then it follows that

$$Q(\theta(t)) \geq \lambda_2(2m - \lambda_N M^2)P(t).$$

Thus, for any $a \leq \min(\lambda_2(2m - \lambda_N M^2), 1)$, the inequality $Q(\theta(t)) \geq aP(t)$ holds. Also, by Assumption 1, $\lambda_2(2m - \lambda_N M^2) > 0$. Then, for some $a \in (0,1)$, Equation (13) is reduced to

$$\begin{array}{rl} E[P(t+1)] & \leq (1-a)E[P(t)] + E[w(t)^T DLDw(t)] \\ & \leq (1-a)E[P(t)] + \lambda_N M^2 E[w(t)^T w(t)]. \end{array}$$

As $t \to \infty$ the contribution of the first term converges to 0. For the second term, recall that each element of $w(t)$ is a linear combination of i.i.d $\eta_i(t) \sim Lap(cq^t)$. For $i \neq j$, $E[\eta_i(t)\eta_j(t)] = E[\eta_i(t)]E[\eta_j(t)] = 0$. For any i, $E[\eta_i(t)^2] = Var(\eta_i(t)) = 2c^2q^{2t}$, which also converges to 0. So $E[w(t)^T w(t)] \to 0$ as $t \to \infty$. Combining, we have $E[P(t)] \to 0$ as $t \to \infty$.

In general, the expected consensus value of the distributed algorithm does not coincide with the initial average. Intuitively, a node with higher degree or slower evolution will have heavier weight on the consensus value. In this context, Definition 4 is modified by replacing the average $\bar{\theta}(0) = \frac{1}{N}\sum_i \theta_i(0)$ with a weighted modification $\bar{\theta}(0) = \frac{\sum_i \gamma_i \theta_i(0)}{\sum_i \gamma_i}$, where the weight $\gamma_i = \frac{1}{d_i} = \frac{|N(i)|+1}{\sigma_i}$.

Lemma 8 (accuracy)**.** *The distributed mechanism achieves $(b, \frac{\sqrt{2\tilde{d}}c}{\sqrt{b(1-q^2)}})$-accuracy, where $\tilde{d} = \frac{\sum_i (|N(i)|+1)^2}{(\sum_i \gamma_i)^2}$.*

PROOF. Let us fix an initial state $\theta(0)$ and define $\bar{\theta}(t) = \frac{\sum_i \gamma_i \theta_i(t)}{\sum_i \gamma_i}$ and $\tilde{w}(t) = \frac{\sum_i w_i}{\sum_i \gamma_i}$. We rewrite Equation (9) with

$$\gamma_i \theta_i(t+1) = \gamma_i \theta_i(t) - |N(i)|\theta_i(t) + \sum_{j \in N(i)} \theta_j(t) + w_i(t).$$

Add up all N equations and divided by $\sum_i \gamma_i$, we get:

$$\bar{\theta}(t+1) = \bar{\theta}(t) + \tilde{w}(t) = \bar{\theta}(0) + \sum_{s=0}^{t} \tilde{w}(s).$$

From the definition of $\tilde{w}(t)$ and Equation (10), we have

$$\begin{array}{rcl} Var(\tilde{w}(t)) & = & \dfrac{Var(\sum_i w_i(t))}{(\sum_i \gamma_i)^2} = \dfrac{Var(\sum_i (|N(i)|+1)\eta_i(t))}{(\sum_i \gamma_i)^2} \\ & = & \dfrac{Var(\eta_i(t)) \sum_i (|N(i)|+1)^2}{(\sum_i \gamma_i)^2} = 2\tilde{d}c^2 q^{2t}. \end{array}$$

By $q \in (0,1)$, the series converges.

$$Var(\sum_{s=0}^{t} \tilde{w}(s)) \leq Var(\sum_{s=0}^{\infty} \tilde{w}(s)) = \frac{2\tilde{d}c^2}{1-q^2}.$$

By Chebyshev's inequality for any $t \geq 0$:

$$Pr(|\bar{\theta}(t) - \bar{\theta}(0)| \leq r) = 1 - Pr(|\sum_{s=0}^{t} w(s)| > r) \geq 1 - \frac{Var(\sum_{s=0}^{t} w(s))}{r^2}.$$

Choosing $r = \frac{\sqrt{Var(\sum_{s=0}^{t} w(s))}}{\sqrt{b}} = \frac{\sqrt{2\tilde{d}}c}{\sqrt{bN(1-q^2)}}$, we have $1 - Pr(|\sum_{s=0}^{t} w(s)| > r) \geq 1 - b$. Let $t \to \infty$, by Lemma 6 every execution converges. Then the lemma follows.

The trade-off between accuracy and privacy of this mechanism is similar to that of the client-server mechanism of Section 3 and we discuss them together next.

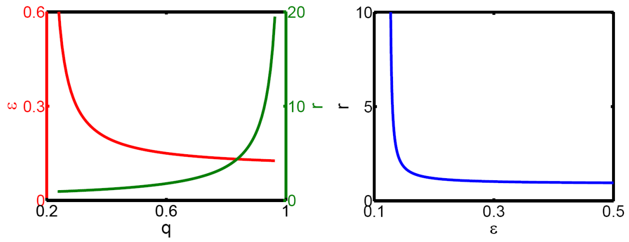
4.2 Discussion on Results

We proposed two mechanisms that achieve iterative private consensus over infinite horizon by adding a stream of noises to the messages set by the clients (to each other or to the server). The standard deviation of the Laplace distribution of the noise added in every round decreases and ultimately converges to $Lap(0)$ which is the Dirac δ distribution at 0. The mechanisms have 3 parameters: linear combination factor σ, initial noise c and noise convergence rate q. The constraint to achieve privacy over infinite horizon is that $q > 1 - \sigma$, which roughly means that the noise should converge slower than the system's inertia so as to "cover" the trail of dynamics.

From Lemmas 1 and 5 we observe that ϵ decreases with larger c or q. This implies that the system has a higher privacy if the noise values are picked from a Laplace distribution with larger parameters (and hence larger standard deviation). From Lemmas 4 and 8, however, a more dispersive noise results in worse accuracy. The tradeoff between privacy and accuracy for different noise convergent rate (q) is illustrated in Figure 1. If we fix the parameter q, we observe that for ϵ-differential privacy for N agents and an accuracy level of b the accuracy radius r is $O(\frac{1}{\epsilon\sqrt{bN}})$. For specific values on these parameters, the dependence between ϵ and r is shown in Figure 1.

The distributed mechanism obviously avoids a single point of failure and a bottleneck at the server, but it requires an additional $2(|\mathcal{E}| - N)$ messages in each round over the client-server mechanism. A comparison of the convergence rates of the two mechanisms will rely on the actual structure of the communication graph, and will be undertaken in the future.

Figure 1: Privacy and Accuracy as functions of the Noise convergent rate in the centralized mechanism. Parameterized with $N = 500$, $\sigma = 0.8$, $c = 10$ and $\beta = 0.5$.



5. RELATED WORK

Our consensus mechanism has similarities with the protocols for computing sum and inner product presented in [1], in that, all these protocols rely on adding noise to the states communicated among the participants. Our mechanism differs in the type of noise (geometrically decaying Laplace) that is added. Moreover, in our setup, the computed outputs are used as feedback for updating the state of the participants to achieve convergence.

In [8] a framework for securely computing general types of aggregates is presented. Every client splits its private data into pieces and sends them to different servers. If at least one server is not compromised, then the iterative aggregate computation is guaranteed to preserve privacy of the individuals. Our mechanism is quite different and it guarantees privacy even if the only server is compromised.

In [32], the authors present distributed protocols for computing k maximum values among all participants. In this protocol, the clients communicate a global vector of k-maximum values over a ring network. In each step, the client processing the global vector either with an exponential decaying probability honestly replaces the values in global state if it is smaller than one of the local values, or it replaces the values in the vector with randomly generated small numbers. The metric of privacy is *Loss of Privacy* which characterizes the additional knowledge to the adversary of gaining intermediate result besides the final results. This work is setup with a different definition of privacy and does not capture the key feature of our mechanism, namely feedback update over infinite horizon.

In the recent paper [18][2], the authors develop a notion of differential privacy which ensures that an adversary cannot tell the exact input to a dynamical system by looking at its output stream. Laplace and Gaussian mechanisms are presented for converting an ordinary dynamical system to a differentially private one. Unlike our message-based and distributed implementation schemes, here the privacy-preserving implementation consists of a filter and an estimator and the former is designed to minimize the mean-squared error from the outputs of the ideal system. In the follow-up work [19], a Kalman filter is designed to estimate the states of differentially private systems with minimized error. The sufficient condition of the minimization problem is established in the form of linear matrix inequalities. The flavor of results presented in both of these papers are geared towards privacy of individual subsystems. In contrast, our mechanisms aim to provide privacy guarantees in dirstibuted control systems with honest but curious agents, untrusted servers, and leaking channels.

6. CONCLUSIONS

In this paper, we formalize a Synchronized Private Consensus problem and propose two mechanisms for solving it. The first one relies on the client-server model of communication and the latter is purely distributed. The key idea is to add a random noise to each client's message to the server (or other clients) that is drawn from a Laplace distribution that converges to the Dirac distribution. The messages with large noise give differential privacy and as the noise level attenuates, the system converges to the target value with probability that depends inversely on the security parameter and the number of participants. The feedback $y(t)$ from the server is the mean of all noisy messages sent. And, the clients update their states by taking a linear combination of $y(t)$ and their previous state. We formally prove the privacy and convergence of this mechanism. The key proof technique for privacy, relies on constructing a bijective map between two sets of executions starting from different but adjacent initial states.

To the best of our knowledge this is the first investigation of differential privacy in the context of control systems where the ultimate goal is convergence. Our results suggest several

[2] At the time of writing, the final version of these paper and [19] are yet to appear in print. Our comments are based on the versions available online.

directions for future work. First, we are trying to apply our method to a larger set of control problems that arise from iterative closed-loop control. Novel applications of this arise from differential privacy and more generally security of *distributed cyber-physical systems* where the physical state is updated smoothly according to some differential equations.

Second, we also interested in exploring the tradeoff between privacy and performance under more general dynamics of the system. In the SPC problem we discussed, the dynamics of the system is discrete and linear. We expect to extend the analysis to continuous or non-linear systems. Also, establishing a lower bound for the problem will be of significance.

An orthogonal direction is to develop automated verification and synthesis algorithms for controllers that preserve differential privacy. Along these lines, a verification framework for streaming algorithms has been presented in [2, 27]. The challenge will be to extend these ideas to synthesis and feedback control systems.

7. REFERENCES

[1] E. Abbe, A. E. Khandani, and A. W. Lo. Privacy-preserving methods for sharing financial risk exposures. *CoRR*, abs/1111.5228, 2011.

[2] G. Barthe, B. K´opf, F. Olmedo, and S. Z. Béguelin. Probabilistic relational reasoning for differential privacy. In *In Proceedings of ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, 2012.

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: numerical methods.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

[4] V. Blondel, J. Hendrickx, A. Olshevsky, and J. Tsitsiklis. Convergence in multiagent coordination consensus and flocking. In *Proceedings of the Joint forty-fourth IEEE Conference on Decision and Control and European Control Conference*, pages 2996–3000, 2005.

[5] K. M. Chandy, S. Mitra, and C. Pilotto. Convergence verification: From shared memory to partially synchronous systems. In *In proceedings of Formal Modeling and Analysis of Timed Systems (FORMATS'08)*, volume 5215 of *LNCS*, pages 217–231. Springer Verlag, 2008.

[6] F. R. K. Chung, V. Faber, and T. A. Manteuffel. An upper bound on the diameter of a graph from eigenvalues associated with its laplacian. *SIAM J. Discret. Math.*, 7(3):443–457, Aug. 1994.

[7] G. Cybenko. Load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7:279âĂŞ301, 1989.

[8] Y. Duan, J. Canny, and J. Zhan. P4p: practical large-scale privacy-preserving distributed computation robust against malicious users. In *Proceedings of the 19th USENIX conference on Security*, USENIX Security'10, pages 14–14, Berkeley, CA, USA, 2010. USENIX Association.

[9] C. Dwork. Differential privacy. In *AUTOMATA, LANGUAGES AND PROGRAMMING*, volume 4052 of *Lecture Notes in Computer Science*, 2006.

[10] C. Dwork. Differential privacy: a survey of results. In *Proceedings of the 5th international conference on Theory and applications of models of computation*, TAMC'08, pages 1–19, Berlin, Heidelberg, 2008. Springer-Verlag.

[11] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *In Proceedings of TCC*, 2006.

[12] C. Dwork, M. Naor, G. Rothblum, and T. Pitassi. Differential privacy under continual observation. In *Proceedings of the 42nd ACM symposium on Theory of computing*, 2010.

[13] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher. Poolview: stream privacy for grassroots participatory sensing. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, pages 281–294, New York, NY, USA, 2008. ACM.

[14] M. Hardt and K. Talwar. On the geometry of differential privacy. In *Proceedings of the 42nd ACM symposium on Theory of computing*, STOC '10, pages 705–714, New York, NY, USA, 2010. ACM.

[15] M. Huang and J. Manton. Coordination and consensus of networked agents with noisy measurements: stochastic algorithms and asymptotic behavior. *IAM Journal on Control and Optimization*, 48, 2009.

[16] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.

[17] T. Johnson and S. Mitra. Safe flocking in spite of actuator faults using directional failure detectors. *Journal of Nonlinear Systems and Applications*, 2(1-2):73–95, 2011.

[18] J. Le Ny and G. J. Pappas. Differentially Private Filtering. *ArXiv e-prints*, July 2012.

[19] J. Le Ny and G. J. Pappas. Differentially Private Kalman Filtering. *ArXiv e-prints*, July 2012.

[20] M. Mesbahi and M. Egerstedt. *Graph-theoretic Methods in Multiagent Networks.* Princeton University Press.

[21] S. Mitra and K. M. Chandy. A formalized theory for verifying stability and convergence of automata in pvs. In *In proceedings of Theorem Proving in Higher Order Logics (TPHOLS'08).* LNCS, 2008. to appear.

[22] R. Olfati-saber. Distributed kalman filtering and sensor fusion in sensor networks. In *Network Embedded Sensing and Control, volume LNCIS 331*, pages 157–167. Springer-Verlag, 2006.

[23] R. Olfati-Saber, J. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

[24] J. Reed and B. C. Pierce. Distance makes the types grow stronger: a calculus for differential privacy. In *Proceedings of the 15th ACM SIGPLAN international conference on Functional programming*, ICFP '10, pages 157–168, New York, NY, USA, 2010. ACM.

[25] R. Saber and R. Murray. Flocking with obstacle avoidance: cooperation with limited communication in mobile networks. volume 2, pages 2022–2028 Vol.2, Dec. 2003.

[26] H. G. Tanner, A. Jadbabaie, and G. J. Pappas.

Automatic Control, IEEE Transactions on, 52:866—868, 2007.

[27] M. C. Tschantz, D. Kaynar, and A. Datta. Formal verification of differential privacy for interactive systems. *Electronic Notes in Theoretical Computer Science*, 2011.

[28] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of EECS, MIT, November 1984.

[29] J. N. Tsitsiklis. On the stability of asynchronous iterative processes. *Theory of Computing Systems*, 20(1):137–153, December 1987.

[30] L. Xiao, S. Boyd, and S.-J. Kim. Distributed average consensus with least-mean-square deviation. *J. Parallel Distrib. Comput.*, 67(1):33–46, Jan. 2007.

[31] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, IPSN '05, Piscataway, NJ, USA, 2005. IEEE Press.

[32] L. Xiong, S. Chitti, and L. Liu. Preserving data privacy in outsourcing data aggregation services. *ACM Trans. Internet Technol.*, 7(3), Aug. 2007.