2013 European Control Conference (ECC)
July 17-19, 2013, Zürich, Switzerland.

Privacy-Preserving Asymptotic Average Consensus

Nicolaos E. Manitara, Student Member, IEEE, and Christoforos N. Hadjicostis, Senior Member, IEEE

Abstract— **In this paper, we develop and analyze a distributed privacy-preserving average consensus algorithm that enables all of the components of a distributed system, each with some initial value, to asymptotically reach average consensus on their initial values, without having to reveal the specific value they contribute to the average calculation. We consider a set of components (nodes) that interact via directional communication links (edges) that form a generally directed communication topology (digraph). The proposed protocol can be followed by each node that does not want to reveal its initial value and, under certain conditions on the communication topology that we characterize precisely, all nodes can calculate the average of their initial values while maintaining privacy (i.e., the initial values contributed to the average by the nodes that follow the protocol are not exposed to malicious nodes). We assume that malicious nodes try to identify the initial values of other nodes but do not interfere in the computation in any other way; malicious nodes are assumed to know the predefined linear strategy and topology of the network (but not the actual values used by the nodes that want to preserve their privacy).**

I. Introduction

In distributed systems and networks, it is often necessary for all or some of the nodes to calculate a function of certain parameters that we refer to as initial values. When all nodes calculate the average of their initial values, they are said to reach average consensus. Average consensus (and more generally consensus) has received a lot of attention from the control community due to its usage in various emerging applications, including wireless smart meters (where all nodes have to agree on the average power demand or consumption of the network [1]), and multi-agent systems (where all agents communicate with each other in order to coordinate their direction or speed [2]). Over the last few decades, a variety of algorithms for calculating different functions of these initial values have been proposed by the control, communication, and computer science communities [3], [4], [5], [6].

One approach to consensus is based on a linear iterative strategy, where each node in the network repeatedly updates its value to be a weighted sum of its own previous value and the values of their neighbors. The weights of the linear iteration are chosen so that all the nodes in the network can asymptotically reach agrement to the same value. In particular, previous work has shown that, if the network topology satisfies certain conditions, the weights for the linear iteration can be chosen such that all of the nodes in the network converge (asymptotically) to the same function of the initial values (such as the average) [7]. The methods described above typically do not consider privacy issues while the network is reaching consensus.

This paper addresses the topic of privacy-preserving asymptotic average consensus which has received limited attention thus far in the literature. Specifically, an anonymization transform using random offsets on the initial values was proposed for a cooperative[1] wireless network in [8]. In this approach, each node that would like to protect its privacy (i.e, it does not want to reveal its initial value) adds a random offset value to its initial value, ensuring in this way that its true initial value will not be exposed through the values exchanged in the network. In doing so, however, this node can potentially alter the outcome of the average calculation. The method described in [8] relies on the fact that the random offsets chosen by each node following the protocol are i.i.d. random variables with zero mean; thus, if an infinite number of nodes add a random offset, their net effect will be zero and the average calculation will not be affected. In real-sized networks, however, this method typically fails to converge to the true average and introduces a random offset with mean zero and some finite variance.

The distributed algorithm we develop and analyze in this paper enables all of the nodes to calculate the *exact* average of their initial values, without loss of privacy and despite the presence of possibly multiple malicious nodes. Malicious nodes are assumed to have full knowledge of the protocol and are allowed to collaborate arbitrarily among themselves, but do *not* interfere in the computation of the average value of the network in any other way (this is why we also refer to them as "malicious-curious" nodes in the abstract). Our approach does not depend on any cryptographic algorithm, but operates by allowing the nodes to introduce pseudo-random offsets (unknown to the malicious nodes). Specifically, the proposed protocol is a variation of the standard protocol [7] that is used in the absence of privacy requirements; and that allows the nodes to asymptotically obtain the average of their initial values by following a linear iteration with weights that form a doubly stochastic matrix. The main change is that, at each time-step, each node following the protocol adds an arbitrary offset value to the result of its iteration, in an effort to avoid revealing its own initial value as well as the initial values of other nodes. What is important is for the nodes

This material is based upon work supported in part by the European Community (EC) 7th Framework Programme (FP7/2007-2013), under grants INFSO-ICT-223844 and PIRG02-GA-2007-224877. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of EC.

The authors are with the Department of Electrical and Computer Engineering, University of Cyprus, Nicosia 1678, Cyprus. email:{manitara.nicolas, chadjic}@ucy.ac.cy

[1] In a cooperative network, all the nodes follow the predefined strategy, without deviating in any way [8].

to ensure that the total (accumulated sum of) offsets that it adds cancel themselves out in the end. We establish that, under certain conditions on the communication topology, this protocol allows the nodes to calculate the average of their initial values in a privacy-preserving manner, despite the presence of malicious agents. For example, the paper establishes that even when a node following the protocol is directly connected to the malicious nodes, but has at least one neighbor that is not directly connected to malicious nodes and whose path(s) to the malicious nodes is (are) through at least one node following the protocol, then privacy is ensured for both nodes (the one following the protocol and this neighbor of it), in the sense that their individual initial values are not revealed to the malicious nodes.[2] We also show that, even if there is (are) path(s) to malicious nodes that are not through nodes following the protocol, the nodes following the protocol will still remain protected (i.e., their initial values will not be exposed to the malicious nodes) at least under certain conditions on the communication topology.

The remainder of the paper is organized as follows. In Section II, we provide some important background on graph theory and describe the linear iterative strategy in the given distributed system, and in Section III we describe the problem setup. We introduce our proposed privacy-preserving average consensus strategy, and the main results of the paper in Section IV. In Section V we present an example, and finish the paper with conclusions and directions for future work in Section VI.

II. Background

A. Distributed System Model

In a distributed system we can model the network topology as a directed graph (digraph) G=$\{X,E\}$ where X=$\{1,2,...,N\}$ is the set of components in the system and $E \subseteq X \times X$ is the set of directed edges. In particular, edge $(i,j) \in E$ if node j can send information to node i. The nodes that can transmit information to node i are said to be the in-neighbors of node i and are represented by the set $\mathcal{N}_i^- = \{j \mid (i,j) \in E\}$; the number of in-neighbors of node i is called the in-degree of node i and is denoted as $\mathcal{D}_i^-$=$|\mathcal{N}_i^-|$. Similarly the set of nodes that receive information from node i are called its out-neighbors and are denoted by $\mathcal{N}_i^+ = \{l \mid (l,i) \in E\}$; the number of out-neighbors of node i is called the out-degree of node i and is denoted by $\mathcal{D}_i^+$=$|\mathcal{N}_i^+|$.

Our model deals with networks where information is transmitted via a broadcast model, i.e., each node sends to all of its out-neighbors the same value (this is done for notational simplicity but the protocol can be modified to handle the case when nodes transmit different values to different out-neighbors). Note that each node receives different values from its in-neighbors. We assume that during the information exchange process all the information is transmitted/received successfully to all of the recipients in the network [3]. Moreover, the nodes must have sufficient memory and computational capability in order to store and perform simple mathematical computations (namely, additions and multiplications) while the iteration is executing. During the transmission/reception process, the nodes in the network receive a value from each of their in-neighbors, and transmit their value to their out-neighbors.

[2]Note that, it might still be possible for the malicious nodes to determine the sum of the initial values of nodes that follow the privacy-preserving protocol (but not their individual values).

B. Average Consensus via Linear Iterative Strategy

In average consensus problems the objective is the calculation of the average of the initial values of the nodes in the network. Assume that each node i in the network has some initial value $x_i[0]$ and, at each time-step k, each node updates its value as a weighted sum of its own value and the values of its in-neighbors (e.g., following the method in [7]). Specifically, at each time-step k, each node updates its value as

$$x_i[k+1] = w_{ii}x_i[k] + \sum_{j \in N_i^-} w_{ij}x_j[k], \tag{1}$$

where w_{ij} are a set of (fixed) weights. The values for all the nodes at time-step k can be aggregated into the value vector $x[k] = \begin{bmatrix} x_1[k] & x_2[k] & ... & x_N[k] \end{bmatrix}^T$ and the update strategy for the entire network can be written compactly as

$$x[k+1] = \underbrace{\begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1N} \\ w_{21} & w_{22} & \cdots & w_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ w_{N1} & w_{N2} & \cdots & w_{NN} \end{bmatrix}}_{W} x[k],$$

for $k \in \mathbb{N}$, where $w_{ij} = 0$ if $x_j \notin \mathcal{N}_i^- \cup \{i\}$.

Definition (Asymptotic Consensus): The system is said to reach asymptotic consensus if $lim_{k\to\infty} x_i[k] = f(x_1[0],x_2[0],...,x_N[0])$ for each i, where $f(x_1[0],x_2[0],...,x_N[0]) \in \mathbb{R}$.

When $f(x_1[0],x_2[0],...,x_N[0]) = c^T x[0]$ for some column vector c (where c^T is the transpose of c), the following result by Xiao and Boyd from [9] characterizes the conditions under which iteration (1) achieves asymptotic consensus.

*T*heorem 1 ([9]): The iteration given by (1) reaches asymptotic consensus on the linear functional $c^T x[0]$ (under the technical condition that c is normalized so that $c^T 1 = 1$) if and only if the weight matrix W satisfies the conditions below:

1) All the eigenvalues of W have magnitude strictly less than 1.
2) W has a simple eigenvalue at 1, with left eigenvector c^T and right eigenvector 1=$[1\ 1...1]^T$.

In particular, if c^T=$\frac{1}{N}[1\ 1...1]$, then average consensus is reached. Also note that if w_{ij} are restricted to be nonnegative, then the above conditions are equivalent to W being a doubly stochastic matrix.

C. Previous Work on Privacy-Preserving Average Consensus

Privacy-preserving average consensus in the presence of malicious agents in the network has received limited attention

thus far. As mentioned earlier, the authors of [8] proposed a transformation method using random offset values in a cooperative wireless network. Specifically, each node i that wishes to protect its privacy adds a random offset value u_i to its initial value x_i. This ensures that its value will not be revealed to malicious (curious) nodes that might be observing the exchange of values in the network. The idea is based upon the observation that, when an infinite number of nodes employ the protocol, their offsets will have a zero net effect on the average, allowing the nodes to converge to the true average value of the network. Specifically, each node i sets $x_i[0]$=x_i'= x_i+u_i where u_i, $i = 1,2,...,N$, are i.i.d. random variables with zero mean. Then, following the protocol for asymptotic average consensus, the nodes converge to,

$$\frac{1}{N}\sum_{i=1}^{N} x_i' = \underbrace{\frac{1}{N}\sum_{i=1}^{N} x_i}_{X} + \underbrace{\frac{1}{N}\sum_{i=1}^{N} u_i}_{U}, \tag{2}$$

where X is the desirable average of the original initial values and U is a random variable that captures the net effect of the offsets. Clearly, $E[U_i] = 0$ (since the u_i are zero mean) and $var[U_i] = \frac{1}{N} var(U_i)$ (since the u_i are i.i.d.).

For $N \to \infty$ we have var(U)$\to 0$ which means that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} x_i' = \lim_{N\to\infty} \left[\frac{1}{N}\sum_{i=1}^{N} x_i + \frac{1}{N}\sum_{i=1}^{N} u_i\right] = X + 0 = X. \tag{3}$$

For a large number of nodes ($N \to \infty$), this method can give results very close to the true average of the network; however, as the number of nodes decreases, the accuracy of this method also decreases, due to the fact that the offset values added to the protected nodes will add a random offset (with mean zero, and some finite variance) to the true average value of the network.

III. Problem Setup

Consider a set of components (nodes) that interact via directional links (edges) in a way that forms a directed communication topology (digraph). All nodes follow the predefined strategy for reaching agreement to the average of their initial values but some nodes are malicious and try to identify the initial values of all or some of the nodes in the network. There exists a set of nodes that would like to preserve their privacy by not revealing to other nodes their initial values. We allow some nodes not to follow the privacy preserving protocol in order to investigate the worst-case scenario that this protocol can handle. We also assume that the malicious nodes have full knowledge of the proposed protocol and are allowed to collaborate arbitrarily among themselves (exchanging information as necessary), but do not interfere in the computation of the average value in any other way. Malicious nodes also know

i) The topology of the network and nodes that are trying to preserve their privacy.
ii) The observability matrix $\mathcal{O}_{i,L+1}$ (defined later) for any L and any node i (this would be the case, for example, if the weight matrix W is known to the malicious nodes).

IV. Proposed Strategy and Main Results

A. Privacy-Preserving Protocol

The objective of the system is to calculate the average of the initial values of the nodes in the network, and at the same time preserve the privacy of the nodes following the protocol. The scheme that we study in this work makes use of linear iterations as in (1) were the weights w_{ij} form a doubly stochastic matrix $W = [w_{ij}]$ (thus, the nodes asymptotically reach consensus to the average of their initial values). The main difference is that node i following the protocol sets its initial value $x_i'[0] = x_i[0] + u_i$ (where $x_i[0] = x_i$ and u_i is some random offset), and subsequently updates its value as

$$x_i'[k+1] = w_{ii}[k]x_i'[k] + \sum_{j\in N_i^-} w_{ij}x_j'[k] + u_i[k],\ k = 0,1,..., \tag{4}$$

where $u_i[k]$ is a pseudo-random value chosen by node i at time-step k. The constraint is that $u_i[k] = 0$ for $k > L_i$ (for some L_i known only to node x_i) and

$$u_i[L_i] = -\sum_{k=0}^{L_i-1} u_i[k] - u_i. \tag{5}$$

At time-step L_i, node x_i effectively cancels-out the pseudo-random values it has added during the information exchange in the network up to that point.

Protocol Description: Nodes following the protocol run the linear iteration in (4) in order to reach asymptotic average consensus. Specifically, node i follows (4) with $x_i'[0] = x_i[0] + u_i$ and

i) Chooses a pseudo random offset $u_i[k],\ k = 0,1,...,L_i-1$ for some randomly chosen integer L_i.
ii) Sets

$$u_i[L_i] = -\sum_{k=0}^{L_i-1} u_i[k] - u_i. \tag{6}$$

iii) Sets $u_i[k] = 0$ for $k \geq L_i + 1$.

Note that L_i is a random integer number of steps known only to node i. The remaining nodes follow the iteration in (4) with zero offsets. Specifically, a node not following the protocol sets $u_i = 0$ and $u_i[k] = 0$ for $k = 1,2,...$, which is the standard protocol for reaching average consensus. Note that the weight matrix W is assumed primitive doubly stochastic. There are many ways to choose such weights, even in a distributed manner [10]-[12].

Lemma 1: Following the iteration in (4) and in combination with the constraint in (5) the network will reach asymptotic average consensus, as long as the weight matrix W is primitive doubly stochastic.

Proof: It is not hard to see that, if we let $L_{max} = max_i\{L_i\}$, then

$$\sum_{i=1}^{N} x_i'[L_{max}+1] = \sum_{i=1}^{N} x_i[0];$$

then, using

$$x_i'[k+1] = w_{ii}x_i'[k] + \sum_{j\in N_i^-} w_{ij}x_j'[k],\quad k = L_{max}+1, L_{max}+2,...,$$

we obtain the average value of the network:

$$\lim_{k\to\infty} x_i'[k] = \frac{1}{N}\sum_{i=1}^{N} x_i'[L_{max}+1] = \frac{1}{N}\sum_{i=1}^{N} x_i[0].$$

B. Modeling Values Seen by Malicious Nodes

Let $P = \{i_1, i_2, ..., i_p\}$ denote the set of nodes following the protocol during a run of the linear iteration. The linear iteration in (4) can be expressed as

$$x'[k+1] = Wx'[k] + \underbrace{[e_{i_1,N}\ e_{i_2,N}\ ...\ e_{i_p,N}]}_{B_p} \underbrace{\begin{bmatrix} u_{i_1}[k] \\ u_{i_2}[k] \\ \vdots \\ u_{i_p}[k] \end{bmatrix}}_{u_p[k]},$$

where $e_{i,N} = [0\ 0\ ...\ 1\ ...\ 0]^T$ is an N-dimensional column vector with a single nonzero entry of value 1 at location i.

From the perspective of node i, the values seen (by node i) at each time step of the linear iteration can be expressed as

$$y_i[k] = C_i x'[k], \tag{7}$$

where C_i is an $(\mathscr{D}_i^- + 1)\times N$ matrix with a single 1 in each row denoting the positions of the state vector $x'[k]$ that are available to node i (these positions correspond to the nodes that are in-neighbors of node i as well as node i itself). The vector $y_i[k]$ denotes the set of values seen by node i during time-step k of the linear iteration [13], [14], [15]. Without loss of generality we will assume that there is a single malicious node since we can always choose C_i so as to include all the values seen by malicious nodes (which would essentially allow malicious nodes to collaborate arbitrarily among themselves).

We also make the worst-case assumption that malicious nodes know the topology and the predefined strategy of the network, hence the set P and matrix B (as well as matrix W).

The set of values seen by node i during the first $L+2$ time-steps of the linear iteration is given by [15]

$$y_i[0:L+1] = \mathscr{O}_{i,L+1} x'[0] + M^p_{i,L+1} u_p[0:L], \tag{8}$$

where $y_i[0:L+1] = [y_i^T[0]\ y_i^T[1]\ ...\ y_i^T[L+1]]^T$ and $u_p[0:L] = [u_p^T[0]\ u_p^T[1]\ ...\ u_p^T[L]]^T$. The matrices $M^p_{i,L+1}$ and $\mathscr{O}_{i,L+1}$ can be expressed recursively as

$$\mathscr{O}_{i,L+1} = \begin{bmatrix} C_i \\ \mathscr{O}_{i,L}W \end{bmatrix},\ M^p_{i,L+1} = \begin{bmatrix} 0 & 0 \\ \mathscr{O}_{i,L}B_p & M^p_{i,L} \end{bmatrix},$$

where $\mathscr{O}_{i,0} = C_i$ and $M^p_{i,0}$ is the empty matrix [15]. These matrices describe the ability of the malicious node i to identify the initial values $x'[0]$ of the nodes as well as the inputs u_p injected by the nodes following the protocol. Note that we make the worst-case assumption that the malicious node i knows exactly which nodes follow the protocol and the weights used in iteration (4).

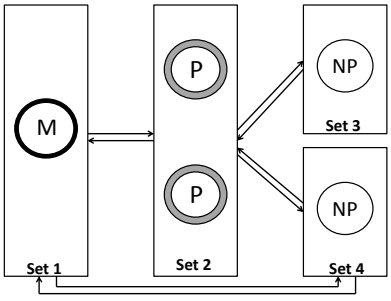

Fig. 1. Example of the key connectivity requirement for privacy preserving average consensus: the black node is the malicious set of nodes V_1, the grey nodes are the nodes following the protocol V_2, and the white nodes are the nodes following the predefined strategy for reaching average consensus (V_3 and V_4).

C. Main Result

The main contribution of this paper is the establishment of topological conditions that ensure privacy for the nodes following the proposed protocol despite the presence of malicious agents in the network.

Theorem 2: Consider a fixed network with N nodes described by a digraph $G = \{X, E\}$. Consider the iteration in (4) with weights that form a primitive doubly stochastic weight matrix W. Assume that a set of nodes P follow the predefined privacy-preserving strategy in (4) with random offsets chosen as in (6). Malicious node i will not be able to identify the initial value of $x_j[0] \in P$, as long as j has at least one other node k connected to it for which all paths from k to the malicious node i are through a node j' following the protocol (i.e., $j' \in P$).

Specifically, if the condition in Theorem 2 is satisfied, the network will reach average consensus (this follows from Lemma 1) and the privacy of the initial values of the nodes following the protocol will be preserved during the linear iteration process.

Proof of Theorem 2: Let $X_1[k]$, $X_2[k]$, $X_3[k]$, $X_4[k]$ denote the vectors of values of nodes in sets V_1 (Malicious), V_2 (Protocol), V_3 and V_4 (following the predefined linear strategy for reaching average consensus). Note that sets V_1, V_2, V_3, V_4 are mutually exclusive and their union comprises X, i.e.,$V_i \cap V_j = \emptyset$ for $i \neq j$ and $V_1 \cup V_2 \cup V_3 \cup V_4 = X$. Using the simple network in Figure 1, we show that set V_1 (malicious node) is unable to identify the initial values of sets V_2 and V_3 in the network when nodes in set V_2 follow the proposed protocol. To see this, we write the weight matrix as

$$W = \begin{bmatrix} W_{11} & W_{12} & 0 & W_{14} \\ W_{21} & W_{22} & W_{23} & W_{24} \\ 0 & W_{32} & W_{33} & 0 \\ W_{41} & W_{42} & 0 & W_{44} \end{bmatrix},$$

where $W_{ij},\ i,j \in \{1,2,3,4\}$ are block matrices of appropriate sizes (note that according to the conditions in Theorem 1, we have $W_{13} = W_{43} = 0$, and the matrix W is doubly stochastic).

The matrix C_1 in (8) is given as

$$C_1 = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & I \end{bmatrix} \equiv \begin{bmatrix} C_{1,1} \\ C_{1,2} \\ C_{1,4} \end{bmatrix},$$

where $C_{1,1}$=[I 0 0 0], $C_{1,2} = [0\ I\ 0\ 0]$ and $C_{1,4} = [0\ 0\ 0\ I]$ (and I are identity matrices of appropriate dimensions). From the definition of matrix $B_p = [e_{i_{1,N}}\ e_{i_{2,N}} \ldots e_{i_{p,N}}]$, we can write $B_{p2} = [0\ I\ 0\ 0]^T$ where matrix I is of dimension $|P| \times |P|$.

Using the recursive definition of $\mathscr{O}_{i,L+1}$, and the fact that

$$C_1 \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} = 0, \text{ we obtain}$$

$$\mathscr{O}_{1,L+1} \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} = \begin{bmatrix} C_1 \\ \mathscr{O}_{1,L}W \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \mathscr{O}_{1,L} \end{bmatrix} W \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ \mathscr{O}_{1,L} \end{bmatrix} (B_{p2}W_{23} + B_{p3}W_{33})$$

where $B_{p3} = [0\ 0\ I\ 0]^T$.

The values seen by the malicious set of nodes V_1 over $L+2$ time steps are given by $y_1[0:L+1] = \mathscr{O}_{1,L+1}x'[0] + M^p_{1,L+1}u_p[0:L]$. Specifically, this expression can then be written as (see Lemma 2 of [11])

$$y_1[0:L+1] = \mathscr{O}_{1,L+1} \begin{bmatrix} X_1[0] \\ X_2'[0] \\ 0 \\ X_4[0] \end{bmatrix} +$$

$$M^2_{1,L+1} \underbrace{\left(\begin{bmatrix} W_{23} \\ W_{23}W_{33} \\ \vdots \\ W_{23}W^L_{33} \end{bmatrix} X_3[0] + \underbrace{\begin{bmatrix} u_2[0] \\ u_2[1] \\ \vdots \\ u_2[L] \end{bmatrix}}_{e_2[0:L]} \right)}_{\alpha_1}.$$

If we let

$$\alpha = y_1[0:L+1] - \mathscr{O}_{1,L+1} \begin{bmatrix} X_1[0] \\ X_2'[0] \\ 0 \\ X_4[0] \end{bmatrix} \tag{9}$$

(note that α is known to set V_1 (malicious) nodes), then we can write

$$\alpha = M^2_{1,L+1}\alpha_1 \tag{10}$$

From the above, it can be seen that even if the malicious nodes in set V_1 know (or can determine) $X_4[0]$, which is the assumption we make when we assume that α is known to the malicious nodes, they will not be able to identify the initial values of sets V_2 and V_3 due to the unknowns $X_3[0]$ and $e_2[0:L]$. The reason is that multiple pairs $e_2[0:L]$ and $X_3[0]$ result in the same α_1. From (9), it can be seen that the pseudo-random values (protocol) of set V_2 successfully protect set V_3 from revealing its true initial values to malicious nodes, as long as the key connectivity of Theorem 2 is satisfied. At the same time, this protects nodes in V_2 since multiple $e_2[0:L]$ are possible, even though the malicious nodes know $X_2'[0]$, they cannot determine $X_2[0] = X_2'[0] - \sum_{k=0}^{L} e_2[k]$.

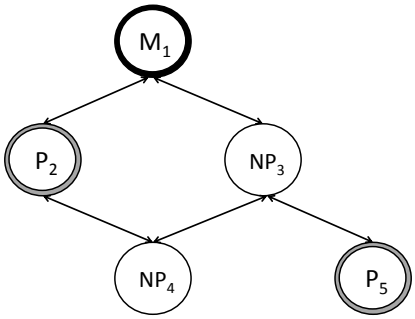

Fig. 2. The black node is the malicious node, the grey nodes are the nodes following the proposed protocol, and the white nodes are the nodes that do not follow the protocol.

V. Example

Note that, even when the condition of Theorem 2 is not satisfied it might be possible for the nodes following the protocol to remain protected (in the sense that their initial values will not be revealed to the malicious nodes). Figure 2 shows a communication topology that violates the condition of Theorem 2 for both nodes 2 and 5 that are assumed to follow the protocol but enables them to maintain privacy. We argue that malicious node 1 is unable to identify the initial values of the other nodes in the network of Fig. 2 when node 2 and node 5 are following the privacy-preserving protocol.

The weight matrix can be written as

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & 0 & 0 \\ w_{21} & w_{22} & 0 & w_{24} & 0 \\ w_{31} & 0 & w_{33} & w_{34} & w_{35} \\ 0 & w_{42} & w_{43} & w_{44} & 0 \\ 0 & 0 & w_{53} & 0 & w_{55} \end{bmatrix}$$

for some nonnegative weights that form a primitive doubly stochastic matrix. The matrix C_1 in (8) is given by

$$C_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} C_{1,1} \\ C_{1,2} \\ C_{1,3} \end{bmatrix},$$

where $C_{1,1} = [1\ 0\ 0\ 0\ 0]$, $C_{1,2} = [0\ 1\ 0\ 0\ 0]$, and $C_{1,3} = [0\ 0\ 1\ 0\ 0]$.

Using the recursive definition of $\mathscr{O}_{i,L+1}$ we obtain

$$\mathscr{O}_{1,L+1}\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}=\begin{bmatrix}C_1\\ \mathscr{O}_{1,L}W\end{bmatrix}\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}=\begin{bmatrix}0\\ \mathscr{O}_{1,L}\end{bmatrix}W\begin{bmatrix}0\\0\\0\\1\\1\end{bmatrix}.$$

The values seen by the malicious node 1 over $L+2$ time steps are given by

$$y_1[0:L+1]=\mathscr{O}_{1,L+1}\begin{bmatrix}x_1[0]\\x_2'[0]\\x_3[0]\\0\\0\end{bmatrix}+\underbrace{M^3_{1,L+1}\begin{bmatrix}w_{34}\\w_{34}w_{44}\\\vdots\\w_{34}w_{44}^L\end{bmatrix}x_4[0]}_{\alpha_1}$$

$$+\underbrace{M^2_{1,L+1}\left(\begin{bmatrix}w_{24}\\w_{24}w_{44}\\\vdots\\w_{24}w_{44}^L\end{bmatrix}x_4[0]+\begin{bmatrix}u_2[0]\\u_2[1]\\\vdots\\u_2[L]\end{bmatrix}\right)}_{\alpha_2}$$

$$+\underbrace{M^3_{1,L+1}\left(\begin{bmatrix}w_{35}\\w_{35}w_{55}\\\vdots\\w_{35}w_{55}^L\end{bmatrix}x_5'[0]+\begin{bmatrix}u_5[0]\\u_5[1]\\\vdots\\u_5[L]\end{bmatrix}\right)}_{\alpha_3}$$

If we let

$$\alpha=y_1[0:L+1]-\mathscr{O}_{1,L+1}\begin{bmatrix}x_1[0]\\x_2'[0]\\x_3[0]\\0\\0\end{bmatrix} \tag{11}$$

(note that α is known to the malicious node), then we have

$$\alpha=\alpha_1+\alpha_2+\alpha_3. \tag{12}$$

Consider two different scenarios:

i) $x_4[0]=C_2$, $x_5'[0]=C_4$ and $u_2=0$, $u_5=0$.
ii) $x_4[0]=C_1$, $x_5'[0]=C_3$ and $u_2[k]=[w_{24}w_{44}^k](C_2-C_1)$, $u_5[k]=[w_{35}w_{55}^k](C_4-C_3)+[w_{34}w_{44}^k](C_2-C_1)$.

In particular, in the second scenario, nodes x_2 and x_5 are following the protocol and apply the error sequence $u_2[k]=[w_{24}w_{44}^k](C_2-C_1)$ and $u_5[k]=[w_{35}w_{55}^k](C_4-C_3)+[w_{34}w_{44}^k](C_2-C_1)$, $k\in\mathbb{N}$. It is not hard to verify that, the values $y_1[k]$, $k\in\mathbb{N}$, seen by the malicious node x_1, are exactly the same for each of the two above scenarios. This makes it impossible for node x_1 to obtain the initial values of the nodes following the protocol, hence, the nodes preserve their privacy.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we considered the problem of privacy-preserving asymptotic average consensus in the presence of malicious agents. We showed that privacy can be guaranteed when the proposed protocol is used in networks whose topology satisfies certain conditions. In particular, we showed that the malicious nodes are not able to identify the initial values of the nodes following the protocol, as long as the nodes following the protocol have a neighbor that is not directly connected to any of the malicious nodes and all independent paths (if any) to a malicious node are through a node following the protocol. Specifically, if this condition is satisfied, the network will reach average consensus and the initial values of the nodes following the protocol will not be revealed. This topology condition is sufficient but not necessary.

Dealing with malicious nodes that may not simply be curious but also aim to interfere with the computation of the average value of the network is to be investigated in our future work.

REFERENCES

[1] A. Dominguez-Garcia and C. N. Hadjicostis, "Distributed algorithms for control of demand response and distributed energy resources," Proc. of 50th IEEE Conf. on Decision and Control and European Control Conf., 2011, pp. 27-32.

[2] W. Ren, R. W. Beard, and E. M. Atkins "A survey of consensus problems in multi-agent coordination," in Proc. American Control Conf., 2005, pp.1859-1864.

[3] N. A. Lynch, Distributed Algorithms. New York: Morgan Kaufmann, 1996.

[4] R. Koetter and M. Medard, "An algebraic approach to network coding," IEEE/ACM Trans. Netw., vol. 11, no. 5, pp. 782-795, Oct. 2003.

[5] J. Hromkovic, R. Klasing, A. Pelc, P. Ruzicka, and W. Unger, Dissemination of Information in Communication Networks. New York: Springer-Verlag, 2005.

[6] J. Cortes, "Distributed algorithms for reaching consensus on general functions," Automatica, vol. 44, no. 3, pp. 726-737, Mar. 2008.

[7] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," IEEE J. Select. Areas in Communications, vol. 26, no. 4, pp. 650-660, May 2008.

[8] M. Kefayati, M. S. Talebi,B. H. Khalaj, and H. R. Rabiee, "Secure consensus averaging in sensor networks using random offsets," Proc. of the IEEE International Conference on Telec., and Malaysia International Conf. on Communications, 2007, pp.556-560.

[9] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," Syst. Control Lett., vol. 53, no. 1, pp. 65-78, Sep. 2004.

[10] A. Dominguez-Garcia and C. N. Hadjicostis, "Distributed strategies for average consensus in directed graphs," IEEE Transactions on Automatic Control, vol. 58, no. 3, pp. 667-681, March 2013.

[11] L. Xiao and S. Boyd, "Distributed average consensus with time-varying metropolis weights," Proc. of the International Conference on Information Processing in Sensor Networks, pp. 63-70, Apr. 2005.

[12] B. Gharesifard and J. Cortes, "Distributed strategies for generating weight-balanced and doubly stochastic digraphs," European Journal of Control, 2012, pp.539-537.

[13] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents-Part I: Attacking the network," in Proc. American Control Conf., 2008, pp. 1350-1355.

[14] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents-Part II: Overcoming malicious behavior," in Proc. American Control Conf., 2008, pp. 1356-1361.

[15] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," IEEE Transactions on Automatic Control, vol. 56, no. 7, pp. 1495-1508, July 2011.