Article

Privacy-Preserving Consensus over a Distributed Network against Eavesdropping Attacks

Dengke Li , Han Zhou and Wen Yang *

Key Laboratory of Advanced Control and Optimization for Chemical Processes,
East China University of Science and Technology, Shanghai 200237, China

* Correspondence: weny@ecust.edu.cn; Tel.: +86-21-64253376

Received: 1 July 2019; Accepted: 27 August 2019; Published: 30 August 2019

Abstract: Motivated by the increasing risk of data leaks in distributed networks, we consider the privacy-preserving problem in a consensus network in the presence of an eavesdropper who is able to intercept the data transmitted on the network. First, we introduce a consensus protocol with privacy-preserving function, and analyze its convergence and its privacy-preserving effect. Second, we propose a criterion to measure the degree of network privacy leaks in the existence of the eavesdropper. Particularly, we consider the networks with ring topology and small-world topology, where we find a suboptimal eavesdropping strategy that maximizes the probability of privacy leaks. Finally, we verify all the derived results by numerical examples.

Keywords: privacy-preserving; consensus protocol; complex networks

1. Introduction

As a typical distributed algorithm, the consensus protocol has been widely used in many fields, including control engineering, computer science, Cyber-Physical Systems (CPS) and Internet-of-Things (IoT). Distributed networks applying consensus protocol, where each node exchanges information with its connected neighbors so that the whole network reaches an agreement, have been investigated extensively from protocol design and stability analysis to performance optimization [1–7]. The implementations of the consensus protocol at the software level are usually done by the Paxos algorithm [8], the Byzantine Fault-Tolerant (BFT) algorithm [9], etc.

For flexibility, distributed networks are often designed in a way that the nodes can freely join or leave the network. Malicious attacks are conducted when an adversary node joins the network, which has brought attention to network security issues as various hostile attacks have caused serious economic and social consequences [10–14].

A distributed network reaches consensus when each node exchanges its state with its neighboring nodes, with the initial states of all the nodes broadcasted to the whole network [15]. This is an unwanted property in some cases. For example, in distributed decision-making, if it is required that every individual makes their own decision independently, it is necessary to keep the participants' initial perspectives secret while achieving an agreement. In IoT applications, distributed privacy preserving is also an urgent issue. For instance, in multi-agent rendezvous problems, a group of agents tries to rendezvous at a certain geographic location, while in some applications, the participating agents may want to keep their initial location secret to the others [16]. Another case is that, in smart grids, the data aggregation of the users' power consumption information requires frequent reading on a smart meter, which might cause unauthorized users' information sharing; thus, it is necessary to develop a privacy-enhanced data aggregation scheme [17,18].

Some remarkable works have been done in distributed privacy preserving. Basically, they are of two categories: the usage of cryptography and the usage of intentionally generated noises.

Researches using cryptography are often involved with quantification and cypher keys [19–22]. Kishida in [19] proposed a real-time signal encryption method by combining a quantizer with the Paillier cryptosystem, which ensures the privacy of the recursion of the consensus protocol with a length-reduced key. The reduce of the length of the key is due to the changing sensitivity of the quantizer which maps real-valued states into integers. Guo et al. in [20] encoded the information with authority traceability using elliptic curve based chameleon hashing. Yang et al. in [21] introduced homomorphic encryption into a distributed projected gradient-based algorithm to reach privacy preserving distributed optimization. Different from the cryptography methods, in the studies which used intentionally generated noises, properly conceived noises were added into the transmitted data to confuse the eavesdropper from directly reading the data [23–26]. Huang et al. proposed a privacy-preserving protocol by adding independent and exponentially decaying Laplacian noise in the process of consensus update [23]. Manitaru et al. in [24] also proposed a noise-adding protocol to achieve privacy preserving average consensus. However, different from the independent true random Laplacian noise in [23], the noise added in [24] is a pseudo-random sequence with finite steps, within which the sum the the sequence is fixed to zero to ensure unbiased average consensus. Similar works has been done by Mo et al. [25] and He et al. [26]. In [25], the added noise is conceived as a linear combination of a standard normal distributed random sequence with time-varying coefficients, which also ensures exact average consensus. In [26], the distribution of the added noise is not fixed, but a criterion is proposed to measure the privacy-preserving performance. Under the criterion, the optimal noise distribution is obtained. In [27,28], distributed privacy preserving consensus is used in cooperative driving and vehicular networks, which improves driving security, enhances infrastructure utilization, reduces fuel consumption and at the same time protect the vehicle privacy in platooning applications.

In most of the above mentioned privacy preserving protocols, a fact is that, if the neighboring set of the n th node contains the neighboring set of the i th node, the n th node can speculate the initial state of the i th node [24,25]. It implies that if parts of the neighbors' information of the i th node can be intercepted by the n th node, then the privacy of the i th node can be inferred. Due to the intensive coupling between nodes in distributed networks, the data transmitted on the network are easily intercepted. In this paper, we consider the privacy preserving problem in the case when an eavesdropper can intercept the information transmitted on the network.

Moreover, it is worth mentioning that this work is an extension of [29]. The networks this paper mainly focus on are the networks of embedded devices such as sensor networks and social networks such as independent consensus decision making, which are vulnerable under eavesdropping attacks. In this paper, we first introduce a privacy-preserving average consensus protocol, and analyze the condition under which all/part of the initial states of nodes will be exposed. In the scenario where an eavesdropper with limited power can intercept the data transmitted on the edges, we propose a criterion to measure the degree of privacy leak, and then study the relationship between the criterion and the power allocation scheme. Since the ring network and the small-world network have typical network structure, we mainly discuss the privacy preserving problem in these two types of network, and find a sub-optimal eavesdropping strategy with which the degree of privacy leaks is maximized. The differences of this paper from [29] are that, (1) we prove the exact average consensus of the proposed privacy-preserving scheme, which was not given in [29]; (2) we consider the scenario where the eavesdropper infers the initial states of the nodes, and give the inferring method and the condition where the inferring method proceeds successfully, which was not considered in [29]; (3) we specify the eavesdropping strategy in the ring network and the small-world network by providing more details in analysis. The main contributions of this paper are summarized as below.

1. We provide a new scheme to achieve privacy-preserving average consensus by splitting the node initial states, which is different from the works in [24,25]. We also reveal the condition where the privacy preserving of the proposed scheme can be breached.

2. We propose a criterion to measure the degree of privacy leaks. Under the criterion, when the eavesdropper is with limited power, we find a suboptimal eavesdropping strategy in ring and small-world network.
3. We explore the convergence performance of our proposed privacy preserving consensus scheme under different noise distributions. We reveal the relationship between the convergence step and variances of the noises by numerical simulations. Besides, we also show that the privacy disclosure of our scheme is smaller than that of the scheme in [24] under certain condition.

The remainder of the paper is organized as follows. In Section 2, we briefly introduce an average consensus protocol, and then propose a privacy-preserving scheme to prevent the initial states of the nodes from being directly sent to the neighbors. We also prove that the proposed privacy-preserving scheme does not affect the exact average consensus of the whole network. In Section 3, we reveal the deducing method conducted by the eavesdropper which infers the initial state of a node under certain condition, and propose a criterion to measure the degree of privacy leaks. Further, we find the optimal attacking strategy of the eavesdropper, which helps us resist the hostile attacks effectively. The optimization of the eavesdropping strategy for ring network and small world network is studied in Section 4 and a numerical example is provided to verify the derived results in Section 5. Section 6 concludes this whole paper.

Notations: $\mathbb{R}^{n \times m}$ denotes the set of n by m matrices. e_i denotes the i th canonical basis in $\mathbb{R}^n$ with a 1 in the i th entry and zeros elsewhere. $E[X]$ denotes the expectation of the random value X . The transpose of a matrix (or a vector) Y is denoted by Y^T . The spectral radius i.e., the greatest absolute value among all the eigen-values of matrix A is denoted by $\rho(A)$. The identity matrix is denoted by I . Denote the vector with all the entries are equal to one (or zero) with appropriate dimensions as $\mathbf{1}$ (or $\mathbf{0}$).

2. Problem Formulation

2.1. Preliminaries

We model a distributed network as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, with the nodes $\mathcal{V} = \{1, 2, \dots, n\}$ being the individuals/devices in the network and the edges $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ representing the communication links between these individuals/devices. Denote the set of the neighbors of node i by $N(i) = \{j \in \mathcal{V} : (i, j) \in \mathcal{E}, j \neq i\}$. Node i and node j can directly communicate if $(i, j) \in \mathcal{E}$. The whole network topology is described by the adjacency matrix $A_d \in \mathbb{R}^{n \times n}$, whose entries $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$, otherwise $a_{ij} = 0$, besides, $a_{ii} = 0$. Define the Laplacian matrix $L = \{l_{ij}\}$ by letting $l_{ij} = -a_{ij}$ if $i \neq j$, and $l_{ii} = \sum_{j=1}^n a_{ij}$. Thus, the number of neighbors, namely the degree of node i is l_{ii} .

We briefly introduce a consensus protocol proposed in [1], where each node updates its state based on the difference from the states of its neighboring nodes,

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N(i)} [x_j(k) - x_i(k)], \quad (1)$$

where ε is a positive real number less than $\max\{l_{ii}\}$ to ensure the stability of the iteration in (1). According to the definition of the adjacency matrix A_d , (1) can also be written as

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j=1}^n a_{ij} [x_j(k) - x_i(k)]. \quad (2)$$

Define the state vector $x(k) \triangleq [x_1(k), \dots, x_n(k)]^T$, and recall the relationship between A_d and the Laplacian matrix L , The node state iteration in (2) can be written as a matrix form

$$x(k+1) = P_\varepsilon x(k), \quad (3)$$

where $P_\varepsilon = I - \varepsilon L$.

To give the condition under which the iteration in (3) reaches asymptotic average consensus, we have the following lemmas.

Lemma 1. (refer to [1]) The Laplacian matrix L has one and only one eigenvalue that is equal to zero, with all the other eigenvalues greater than zero, if the undirected graph $\mathcal{G}$ is connected, namely the degree of all the nodes in $\mathcal{G}$ is at least one.

Lemma 2. (refer to [30]) Matrix P_ε satisfies

$$\lim_{k \rightarrow \infty} P_\varepsilon^k = \frac{\mathbf{1}\mathbf{1}^T}{n},$$

if and only if (1) $\mathbf{1}^T P_\varepsilon = \mathbf{1}^T$, and $P_\varepsilon \mathbf{1} = \mathbf{1}$, (2) the spectral radius $\rho(P_\varepsilon) = 1$. If such, P_ε is called a **doubly stochastic matrix**.

Theorem 1. The iteration in (3) reaches asymptotic average consensus, if the undirected graph $\mathcal{G}$ is connected.

Proof. Please refer to Appendix A. $\square$

It can be noticed from (1) that the initial states are directly sent to the neighboring nodes, which might cause a privacy breach. To prevent the initial states from being acquired by the neighboring nodes, we have to propose a privacy-preserving average consensus protocol.

2.2. Analysis of Privacy Disclosure

Without loss of generality, we assume that node n infers the initial states of the other nodes in the network. Denote the set of the neighboring nodes of node n as $N(n) = \{j_1, \dots, j_m\}$, and define an observation matrix $C = [e_{j_1}, \dots, e_{j_m}, e_n]^T \in \mathbb{R}^{(m+1) \times n}$, where e_i represents a vector whose i th entry is 1 and all the other entries are 0. At time step k , the states of the nodes observed by node n is

$$y(k) = Cx(k). \quad (4)$$

According to (3), the current observation and history observations of node n are given by the following equations

$$\begin{aligned} y(0) &= Cx(0), \\ y(1) &= CP_\varepsilon x(0), \\ &\dots \dots \\ y(k) &= CP_\varepsilon^k x(0). \end{aligned}$$

To ensure that number of the above equations is no less than n , let $k = n - 1$, we have

$$\begin{bmatrix} y(0) \\ y(1) \\ \vdots \\ y(n-1) \end{bmatrix} = \begin{bmatrix} C \\ CP_\varepsilon \\ \vdots \\ CP_\varepsilon^{n-1} \end{bmatrix} x(0). \quad (5)$$

Thus, if node n wants to infer the initial states of all the nodes in the network, $x(0)$ in (5) must has a unique solution. This means that the rank of the matrix on the right hand side of (5) must be n . Otherwise, if the rank is less than n , $x(0)$ can be represented by

$$x(0) = \eta + \gamma,$$

where $\eta \in K$, K is the null space of the matrix on the right hand side of (5), γ is a special solution of (5). Therefore, for any vector $\zeta \in K^\perp$, namely $\zeta \perp K$, we have

$$\zeta^T x(0) = \zeta^T \eta + \zeta^T \gamma = 0 + \zeta^T \gamma = \zeta^T \gamma. \quad (6)$$

In that for any two special solutions γ_1 and γ_2 of (5), $\gamma_1 - \gamma_2 \in K$; thus, $\zeta^T \gamma_1 = \zeta^T \gamma_2$. This means that for a given $\zeta \in K^\perp$, $\zeta^T x(0)$ can be calculated by the malicious node.

Remark 1. ([31]) A null space of a matrix A is the set of all the solutions of equation $Ax = 0$. If K is the null space of A , the space that is orthogonal to K is denoted by $K^\perp$, which is the set of all the linear combinations of the row vectors of A .

The above analysis indicates that a malicious node in the network can obtain more information than expected. If the rank of the matrix on the right hand side of (5) is n , full information of the initial states of all the nodes in the network is disclosed. If the rank is less than n , the malicious node can deduce more information than what it directly observed. In Section 5.1, a numerical example is given to show that even a node is not directly connected to the malicious node, its initial state can still be acquired by the malicious node. Thus, it is necessary to develop a privacy-preserving consensus protocol to protect the initial states from being acquired by the malicious node.

2.3. Privacy Preserving Scheme

In this subsection, we propose a privacy-preserving scheme for distributed average consensus by decomposing the initial states of the nodes into a plurality of different values. At different time steps, the values obtained by the decomposition are respectively injected to the distributed consensus iteration. The scheme is stated as follows.

1. Each node i generates a random integer m_i , with $m_i \in \{2, 3, \dots, M\}$ and M being a given positive integer.
2. Mark the real initial state of node i as $x_i^r(0)$, and reset the initial state of each node as $x_i(0) = 0$ for state update, and denote $x^r(0) = [x_1^r(0), x_2^r(0), \dots, x_n^r(0)]^T \in \mathbb{R}^n$.
3. Each node generates m_i numbers $r_i(1), r_i(2), \dots, r_i(m_i)$. These numbers can be arbitrary as long as $r_i(k) \neq 0, \forall k = 1, 2, \dots, m_i$ and

$$\sum_{k=1}^{m_i} r_i(k) = m_i \cdot x_i^r(0). \quad (7)$$

4. At time step k , inject the number $d_i(k)$ into the state of each node, where

$$d_i(k) = \begin{cases} r_i(k)/m_i, & \text{if } 1 \leq k \leq m_i \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

Remark 2. The superscript 'r' in $x_i^r(0)$ means 'real', it distinguishes the real initial state from the fake one $x_i(0)$, which is reset to zero to deceive the eavesdropper in this scheme.

According to the above four steps, each node updates its state by

$$x_i(0) = 0, \forall i \in 1, 2, \dots, n \quad (9)$$

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} [x_j(k) - x_i(k)] + d_i(k+1). \quad (10)$$

Equation (10) can be written in a matrix form by

$$x(k+1) = P_\varepsilon x(k) + d(k+1). \quad (11)$$

By using the above scheme, the privacy of $x^r(0)$ can be protected because the real initial states of nodes are hidden in a group of pseudo-random number.

Performance evaluation: The scheme above is not executed by one single device, instead, it is performed simultaneously by n independent devices. For each device in the network, in step 1 and step 3, the generation of $m_i - 1$ random numbers can be achieved by a hardware noise generator. One approach is to subtract the input signal from the output of an analog circuit whose gain is 0 dB to obtain the resistance thermal noise. The thermal noise is then amplified and sampled to obtain a discrete random sequence. Note that a hardware noise generator does not increase the execution time of the processor. In step 2, allocating a new memory piece for $x_i^r(0)$ to preserve the real initial state and setting the current initial state into zero only need to be executed for once before the iteration starts; thus, step 2 has computational time complexity of $O(1)$. In Step 3, since $r_i(k)$, $(k = 1, \dots, m_i - 1)$ are randomly generated, to ensure (7) is satisfied, $r_i(m_i)$ is calculated by $r_i(m_i) = m_i x_i^r(0) - \sum_{k=1}^{m_i-1} r_i(k)$, which takes $m_i - 1$ operations of addition, thus is with computational time complexity of $O(m_i)$. Step 4 is executed with m_i operations of division, also with computational time complexity of $O(m_i)$.

Overall, this privacy-preserving consensus scheme has computational time complexity of $O(m_i)$, which is acceptable and applicable.

Theorem 2. *The privacy-preserving scheme in (10) ensures the exact average consensus.*

Proof. Please refer to Appendix A. $\square$

The proposed privacy-preserving scheme ensures privacy preserving if it is assumed that the eavesdropper does not try to infer the initial states of other nodes, because the scheme only blocks the direct streaming of the node state information. It is necessary to find the condition under which the initial states of all/part of the nodes might be deduced by the eavesdropper, thus to investigate the network privacy-preserving performance under eavesdropping attacks.

3. Main Results

3.1. Analysis of Privacy Preserving

In a standard consensus protocol, the initial state of a node can easily be speculated by the other nodes. By using the privacy-preserving scheme in (10), all the nodes can keep their real initial states hidden in the sequence $d(k)$ while reaching average consensus. Hence, the problem of deducing $x^r(0)$ actually can be transformed into computing $d(k)$, $k = 1, 2, \dots, M$. In this section, we mainly find the condition for deducing the initial states successfully by the n th node with the proposed privacy-preserving protocol.

According to (4) and (11), we can obtain

$$x(k) = P_\varepsilon^k x(0) + \sum_{i=0}^{k-1} P_\varepsilon^{k-1-i} d(i+1), \quad (12)$$

$$y(k) = C P_\varepsilon^k x(0) + C \sum_{i=0}^{k-1} P_\varepsilon^{k-1-i} d(i+1). \quad (13)$$

Recalling that $x(0) = 0$, we have:

$$Y = FD, \quad (14)$$

where

$$F = \begin{bmatrix} C & 0 & \dots & 0 \\ CP_\varepsilon & C & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CP_\varepsilon^{M-1} & CP_\varepsilon^{M-2} & \dots & C \end{bmatrix} \in \mathbb{R}^{(m+1)M \times nM},$$

$$Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(M) \end{bmatrix}, \quad D = \begin{bmatrix} d(1) \\ d(2) \\ \vdots \\ d(M) \end{bmatrix} \in \mathbb{R}^{nM \times 1}.$$

For a fixed network topology, F is a constant matrix. In order to derive the real initial state $x^r(0)$ from D , we need to utilize the information Y to solve Equation (14). We define

$$s_i = \begin{cases} e_i, & \text{if } i \in N(n) \\ \mathbf{0}, & \text{otherwise} \end{cases},$$

where $\mathbf{0} \in \mathbb{R}^n$ is zero vector. Further, define

$$\tilde{C} = [s_1, s_2, \dots, s_n]^T \in \mathbb{R}^{n \times n},$$

$$\tilde{y}(k) = \tilde{C}x(k) \in \mathbb{R}^{n \times 1},$$

where $\tilde{y}(k)$ can be obtained from $y(k)$ for a given C . Using the above arguments, it is easy to obtain the following theorem.

Theorem 3. *If the i th row of P_ε , denoted by P_{ε_i} , satisfies $P_{\varepsilon_i}\tilde{C} = P_{\varepsilon_i}$ for all $i = 1, 2, \dots, n$, i.e., $N(i) \cup i \subseteq N(n) \cup n$, then the n th node can deduce the initial state of the i th node.*

Proof. Please refer to Appendix A. $\square$

As discussed above, the initial state of node i can be deduced if the condition $N(i) \cup i \subseteq N(n) \cup n$ is satisfied, i.e., node i is a neighbor of node n and all the neighbors of node i are neighbors of node n .

3.2. Privacy Preservation in the Presence of an Eavesdropper

In this subsection, motivated by some practical applications, we consider the case where an eavesdropper with limited energy allocated intends to speculate the initial states of the nodes in a network. It is addressed in [32] that when an eavesdropper intercepts the information transmitted on a wireless channel, a certain amount of energy is consumed due to the activation of antennas. Thus, the eavesdropper tends to find an optimal eavesdropping strategy so that the privacy of the network being listened to is obtained with least energy consumed. Conversely, from the perspective of the network, if the optimal strategy of the eavesdropper is known, it in turn helps the system designer to decide the corresponding anti-eavesdropping scheme to preserve the privacy of the nodes.

In the remainder of this paper, if we say the eavesdropper attacks the edge (i, j) , it means that the eavesdropper intercepts the information transmitted on the edge (i, j) .

Assumption 1. *The eavesdropper knows the network topology, and intercepts the data transmitted on the link between node i and node j according to a probability $\lambda_{i,j}$, $(i, j) \in \mathcal{E}$. The probability $\lambda_{i,j}$ is a linear function of $\mu_{i,j}$, i.e., $\lambda_{i,j} = \kappa \cdot \mu_{i,j}$, where $\mu_{i,j}$ is the power allocated by the eavesdropper to attack the edge (i, j) and κ is a positive constant. The total sum of the power that the eavesdropper consumes is denoted by T , which means that $\sum_{(i,j) \in \mathcal{E}} \mu_{i,j} = T$. In this paper, we assume $\kappa = 1/T$; thus, $\sum_{(i,j) \in \mathcal{E}} \lambda_{i,j} = 1$.*

As discussed above, we know that the initial state of node i will be exposed when some conditions are satisfied. To better illustrate this, we consider the case in Figure 1. Assume that the eavesdropper wants to infer the initial state of node 1, i.e., let $i = 1$.

1. At time step k , when the information transmitted on edge $(1,2)$ and $(1,5)$ is exposed, the eavesdropper can easily utilize the obtained information to predict the state of node 1 at time step $k + 1$, denoted as $\hat{x}_1(k + 1)$, which does not contain the offset $d_1(k + 1)$. According to the information transmitted from node 1 to node 5 at time step $k + 1$, $x_1(k + 1)$, the eavesdropper can further obtain the injected value which equals $x_1(k + 1) - \hat{x}_1(k + 1)$. The eavesdropper can derive the injected value at each step, then the initial state of node 1 can be deduced after several interactions.
2. At time step k , when the information passed on edge $(1,2)$ and $(5,6)$ have been exposed, because the information sent from node 5 to its neighbors are identical, the eavesdropper can speculate all the information sent to node 1, then it can complete the speculation in a similar way.

Figure 1. Network structure.

Therefore, when the network is in the situation similar to the case in Figure 1, for example, the information passed on edge $(1,5)$ and $(2,4)$ is exposed, the eavesdropper can obtain the injected value of each step respectively, then the initial state of node 1 can be deduced by several iterations.

Remark 3. If the information of at least one edge $(i,j), j \in N_i$ has been successfully intercepted by the eavesdropper, and for each node $j \in N_i$, the information of at least one edge $(j,s), s \in N_j$ has been intercepted successfully, then the initial state of node i can be successfully inferred by the eavesdropper.

In the following, we propose a criterion to measure the degree of privacy disclosure. First, we describe whether the leak of the data transmitted on edge (i,j) at time step k as a binary random value $\gamma_{i,j}(k)$. When $\gamma_{i,j}(k) = 1$, the information flowing on the edge (i,j) are intercepted successfully, otherwise $\gamma_{i,j}(k) = 0$. Then we define the attacking matrix $A_r(k)$, which satisfies

$$A_r[i,j](k) = \begin{cases} \gamma_{i,j}(k), & \text{if } l_{ij} = -1 \\ 0, & \text{otherwise} \end{cases}.$$

Recall that $E[\gamma_{i,j}(k)] = \lambda_{i,j}$, we denote $E[A_r(k)]$ as A_t .

$$A_t(i,j) = \begin{cases} \lambda_{i,j}, & \text{if } l_{ij} = -1 \\ 0, & \text{otherwise} \end{cases}.$$

To investigate the influence of the eavesdropper on the performance of privacy preserving, we define P_i as the probability of privacy disclosure of the i th node, $i = 1, \dots, n$ i.e., the probability of deducing the initial state of the i th node successfully by the eavesdropper.

Theorem 4. If an eavesdropper can intercept the information transmitted on edges in the network, the probability of privacy disclosure P_i for all i can be calculated by

$$\begin{aligned} P_i &= W_i - W'_i, \\ W_i &= \prod_{m \in N(i)} [1 - \prod_{v=1}^n (1 - e_v^T A_t e_m)], \\ W'_i &= \prod_{m \in N(i)} [1 - \prod_{v=1, v \neq i}^n (1 - e_v^T A_t e_m)]. \end{aligned} \quad (15)$$

Proof. Please refer to Appendix A. $\square$

4. Privacy Preserving in Two Typical Networks

In this section, we will discuss privacy preserving in the presence of an eavesdropper in two typical networks: the ring network and the small world network. The reason we select the ring network is that, in the ring network, all the nodes and edges have homogeneous topological conditions: each node in the network is connected with two neighboring nodes. This is important to our analysis because later in this section, we inspect how the number of edges attacked has an influence on the privacy disclosure. The homogeneous topological conditions of the edges help exclude the influence of the positions of edges selected. The reason we select the small-world network is that many real world networks, for example, the World Wide Web, the Internet, social networks of acquaintance and neural networks etc., have the topology of a small-world network [33]. A small-world network is defined to be a network where the expectation of the length of the shortest path between two selected nodes grows linearly to the logarithm of the number of nodes n in the network.

Intuitively, the probability P_i in the case when attacking some “key” edges is certainly larger than the one in the case when attacking those “trivial” edges. For an eavesdropper with limited power, allocating the power to launch the attack on different edges results in different privacy-preserving effects. From the perspective of the eavesdropper, it intends to find an optimal attacking strategy resulting in the maximal probability of privacy disclosure. As a defender, we are also interested into acquiring optimal attacking strategy of the eavesdropper to protect network security more effectively. In the following, we formulate an optimization problem, which helps find a suboptimal attacking strategy of the eavesdropper in two typical networks. Denote the probability of the privacy disclosure of the whole network as $P = \sum_{i=1}^n P_i$.

$$\begin{aligned} (P_0) : \max_{A_t} \quad & P, \\ \text{s.t.} \quad & \sum_{(i,j) \in \mathcal{E}} \lambda_{i,j} = 1, \\ & 0 \leq \lambda_{i,j} \leq 1. \end{aligned} \quad (16)$$

4.1. Ring Network

In a ring network (as in Figure 2), nodes are homogeneously symmetrical. An application of the ring network topology is in distributed independent decision making. In that case, a number of decision makers have to rate on a certain issue. The final rating of the issue is the average of the ratings from all the decision makers. To equalise the weight of the ratings from each decision maker, the decision makers should be in the same condition when assessing the issue. When interconnection is required and a centralized information collector is excluded, a good choice is that the decision makers communicate in a ring topology in that: (1) every decision maker is in the same topological condition; (2) least communication links are needed, thus the possibility that the ratings interact each other is minimized. To ensure that the ratings are based on independent assessments by the decision makers,

it is required that each decision maker does not know the others' ratings; thus, a distributed privacy preserving average consensus scheme is needed. According to Theorem 4, we derive P_i as follows:

$$\begin{aligned} P_i &= W_i - W'_i, \\ W_i &= H_{i-2} \cdot H_i, \\ H_i &= 1 - (1 - \lambda_{i,i+1})(1 - \lambda_{i+1,i+2}), \\ W'_i &= \lambda_{i-2,i-1} \cdot (1 - \lambda_{i-1,i}) \cdot (1 - \lambda_{i,i+1}) \cdot \lambda_{i+1,i+2}. \end{aligned} \quad (17)$$

We rewrite Equation (17) as

$$\begin{aligned} P_i &= \lambda_{i-1,i} \cdot \lambda_{i,i+1} + \lambda_{i-1,i} \cdot \lambda_{i+1,i+2} + \lambda_{i-2,i-1} \cdot \lambda_{i,i+1} \\ &\quad - \lambda_{i-2,i-1} \cdot \lambda_{i-1,i} \cdot \lambda_{i,i+1} - \lambda_{i-1,i} \cdot \lambda_{i,i+1} \cdot \lambda_{i+1,i+2}. \end{aligned} \quad (18)$$

Figure 2. Ring network.

In order to simplify the notation, we use λ_i to replace $\lambda_{i,i+1}$, particularly, λ_n represents $\lambda_{n,1}$. Due to the special structure of ring network, we can calculate P by

$$\begin{aligned} P &= P_1 + P_2 + \dots + P_n \\ &= \sum_{i=1}^n \lambda_i \cdot \lambda_{i+1} + 2 \sum_{i=1}^n \lambda_i \cdot \lambda_{i+2} - 2 \sum_{i=1}^n \lambda_i \cdot \lambda_{i+1} \cdot \lambda_{i+2}. \end{aligned} \quad (19)$$

In the following, we discuss two cases, respectively.

Case 1. When the eavesdropper only attacks two edges, it means only two variables in $\lambda_1, \lambda_2, \dots, \lambda_n$ are nonzero numbers. The eavesdropper has many choices, for example, it can attack two adjacent edges, such that λ_i and λ_{i+1} are not zero, we can obtain the probability $P = \lambda_i \cdot \lambda_{i+1}$ with the constraint $\lambda_i + \lambda_{i+1} = 1$. The eavesdropper can also choose two edges $(i, i+1)$ and $(i+2, i+3)$, then λ_i and λ_{i+2} are not zero, we can calculate $P = 2\lambda_i \cdot \lambda_{i+2}$ with the constraint being $\lambda_i + \lambda_{i+2} = 1$. Besides the above two cases, the eavesdropper can also choose two edges $(i, i+1)$ and $(i+k, i+k+1)$, where $k \in (2, n-2)$, then λ_i and λ_{i+k} are not zero. In this case, we can obtain $P = 0$. Using all the above arguments, it is easy to derive the optimal attacking strategy, i.e., $\lambda_i = \lambda_{i+2} = 0.5$ for any i , and the corresponding maximal probability $P = 0.5$.

Case 2. When the eavesdropper chooses three edges, we can find an optimal attacking strategy, using the similar method as above discussed. We choose three edges as shown in Figure 3. Noting that $\lambda_i, \lambda_{i+1}, \lambda_{i+2}$ are not zero, we can obtain $P = \lambda_i \cdot \lambda_{i+1} + \lambda_{i+1} \cdot \lambda_{i+2} + 2\lambda_i \cdot \lambda_{i+2} - 2\lambda_i \cdot \lambda_{i+1} \cdot \lambda_{i+2}$ with the constraint $\lambda_i + \lambda_{i+1} + \lambda_{i+2} = 1$.

Figure 3. Attacking mode 1.

Obviously, the index of i does not influence the optimal value of P . Without loss of generality, let $i = 1$. The optimization problem P_0 in this case can be rewritten as

$$\begin{aligned} \max_{\lambda_1, \lambda_2, \lambda_3} \quad & \lambda_1 \cdot \lambda_2 + \lambda_2 \cdot \lambda_3 + 2\lambda_1 \cdot \lambda_3 - 2\lambda_1 \cdot \lambda_2 \cdot \lambda_3. \\ \text{s.t.} \quad & 0 \leq \lambda_1, \lambda_2, \lambda_3 \leq 1, \\ & \lambda_1 + \lambda_2 + \lambda_3 = 1. \end{aligned} \quad (20)$$

We use the Kuhn–Tucker Conditions [34] to solve this problem. The inequality constraint can be noted as $h_i(\lambda) \leq 0, i \in [1, 6]$,

$$\begin{aligned} h_1(\lambda) &= -\lambda_1 \leq 0, \\ h_2(\lambda) &= \lambda_1 - 1 \leq 0, \\ h_3(\lambda) &= -\lambda_2 \leq 0, \\ h_4(\lambda) &= \lambda_2 - 1 \leq 0, \\ h_5(\lambda) &= -\lambda_3 \leq 0, \\ h_6(\lambda) &= \lambda_3 - 1 \leq 0. \end{aligned}$$

The equality condition is $\Phi(\lambda) = \lambda_1 + \lambda_2 + \lambda_3 - 1 = 0$. And Γ denotes the optimization function. Then we have

$$\begin{aligned} \nabla \Gamma(\lambda) + \sum_{i=1}^6 \zeta_i \nabla h_i(\lambda) + \psi \Phi(\lambda) &= 0, \\ \zeta_i h_i(\lambda) &= 0, \\ \Phi(\lambda) &= 0. \end{aligned}$$

Finally, we can obtain the result as $\lambda_1 = \lambda_3 = 0.5, \lambda_2 = 0$, and $P = 0.5$.

We choose three edges as shown in Figure 4, where $\lambda_i, \lambda_{i+1}, \lambda_{i+3} \neq 0$. We can obtain $P = \lambda_i \cdot \lambda_{i+1} + 2\lambda_i \cdot \lambda_{i+3}$ with $\lambda_i + \lambda_{i+1} + \lambda_{i+3} = 1$. For $i = 1$, the optimal attacking strategy is $\lambda_2 = \lambda_4 = 0.5$, and $P = 0.5$. As shown in Figure 5, when the chosen three edges $\lambda_i, \lambda_{i+2}, \lambda_{i+4} \neq 0$, with $\lambda_i + \lambda_{i+2} + \lambda_{i+4} = 1$, we can calculate $P = 2\lambda_i \cdot \lambda_{i+2} + 2\lambda_{i+2} \cdot \lambda_{i+4}$. For $i = 1$, the optimal attacking strategy is $\lambda_3 = 0.5, \lambda_1 + \lambda_5 = 0.5$, and $P = 0.5$. In the case in Figure 6, the analysis is similar to the case of attacking two edges. Overall, we conclude that when the eavesdropper chooses three edges, the optimal attacking strategy is $\lambda_{i+2} = 0.5, \lambda_i + \lambda_{i+4} = 0.5$, and $P = 0.5$.

Figure 4. Attacking mode 2.

Figure 5. Attacking mode 3.

Figure 6. Attacking mode 4.

4.2. Small-World Network

An application of the small-world network topology is in the distributed wireless sensor networks. A network of wireless sensors tends to be deployed in complicated unmanned environments. Sensor nodes might join or leave the network due to the instability of telecommunication caused by the changing of the geo-locations of the sensors and the multi-path effect of wireless signals. Moreover, the power of communication is limited so that the maximum communication distance between two sensor nodes might be small, which makes the sensor nodes have relatively a small number of neighbors. These features make a wireless sensor network acts like a small-world network. In that a wireless sensor network is designed to give a state estimation of the system it observes, the estimations of different sensors have to be aggregated to reach their average. To prevent the data aggregation process from being eavesdropped by a malicious third party, a privacy preserving average consensus scheme is also needed. In this subsection, we try to solve the optimization problem P_0 in a more complicated case, and propose a suboptimal attacking strategy for a small-world network. Here, we use the Sequential Least Squares Programming optimization algorithm (SLSQP) introduced in [35]. Since the optimization algorithm might fall into a local optimum, the searching results depend on the position of the starting

point. We therefore search from different starting points separately and average the result at each time in order to approximate the global optimal solution. The procedures are listed in Algorithm 1.

Algorithm 1 Suboptimal attacking strategy in arbitrary network

1. Generate a network.
2. Each nonzero element in A_i is treated as a variable $v_i \in [0, 1]$, and let $V = [v_1, v_2, \dots, v_n]$.
3. Write down the relationship between the probability of privacy leakage P and each variable, and denote final optimization results $P_{\text{optimal}} = 0$, $V_{\text{optimal}} = [0, 0, \dots, 0]$.
4. In the k th search, define the initial value of each variable as $V_0(k)$. Then use the SLSQP algorithm to find the optimal result. Denote the maximal P as $P_o(k)$, the corresponding V is $V_o(k)$. If $P_o(k) > P_{\text{optimal}}$, then we update P_{optimal} and V_{optimal} with $P_o(k)$ and $V_o(k)$.
5. If $k + 1 < 1000$, then generate a new initial set $V_0(k + 1)$, and repeat step 4, or stop the optimization process.

where $V_0(k)$ is generated by the following procedure:

1. Randomly select some variables from V , and the number of variables selected is random.
 2. The selected variable is assigned with a random value between $[0, 1]$.
-

Next, we further investigate the performance of privacy preserving by Algorithm 1 in small-world networks. In recent years, complex networks have been widely investigated due to the universal phenomenon of small-world network features found in nature [33,36–38]. As Remark 3 states, in a general network topology, if the information of at least one edge connected to node $j \in N_i$ is exposed, along with the information of at least one edge connected to node i is intercepted, then the eavesdropper can infer the initial state of node i successfully.

Note that, in a network, the degrees of some nodes might be equal to 1, i.e., the nodes have only one neighbor. Once the information flowing on the only edge is intercepted, the initial state of the node is easily deduced by the eavesdropper. For those nodes with degrees greater than 1, the eavesdropper must intercept the information of multiple edges to deduce the initial state of these nodes. From the perspective of power saving, the eavesdropper is more inclined to attack the edge connecting to the node whose degree is 1.

In the following, we generate a small-world network by setting the number of nodes $N = 20$, the number of initial edges of each node $K = 2$, and the probability of reconnection $P_{\text{reconnect}} = 0.1$. By solving Algorithm 1, the derived optimal attacking strategy is $\lambda_{13,16} + \lambda_{17,18} = 1$ and $\lambda_{10,11} + \lambda_{16,17} = 1$ for the networks in Figures 7 and 8, respectively.

Figure 7. Small-world network 1.

Figure 8. Small-world network 2.

Further, we can find that attacking the node with a degree that is greater than 1 results in the best attacking effect. For example, in Figure 7, the degree of node 16 and node 17 equals 1. When the probability of obtaining the information on the unique edge of node 16 is $\lambda_{13,16}$, by Theorem 4, we can calculate $P_{16} = \lambda_{13,16}$, thus $P = P_{16} + P_{17} = \lambda_{13,16} + \lambda_{17,18} = 1$.

The above result reveals that, in a network where each participant divides the information into several parts and then sends out each part at each time step separately, if a participant has only one neighbor, i.e., one channel to communicate, the eavesdropper can fully monitor the dynamics of the participant as long as it can capture the information through the channel. For other participants with multiple channels, it is necessary to monitor multiple channels at the same time in order to obtain the initial information of the participant, along with consuming much more power of the eavesdropper. Thus, in the case where the initial information of each participant is equally important, the eavesdropper is more willing to intercept the information of those participants who have one single information channel.

Besides the above case, the eavesdropper will focus on those nodes whose degrees are relatively small. A special case is, when the neighbors of node i are connected to each other, if the eavesdropper obtains the information passed on any edges between a pair of nodes, it is equivalent to getting the information sent out from node $j \in N_i$ separately. In the following, we validate the above discussion in a small-world model.

Set $N = 20, K = 4, P_{reconnect} = 0.2$. In Figure 9, the optimal attacking strategy given by Algorithm 1 is $\lambda_{4,5} = 0.5, \lambda_{4,8} = 0.5$. We can calculate $P = 0.25$. Note that node 4 is the only node whose degree is 2. Attacking two edges connected to node 4 with identical power leads to the highest extent of privacy leak.

Figure 9. Small-world network 3.

Figure 10 is a small-world model under the condition $N = 20, K = 4, P_{reconnect} = 0.5$. Node 0 has the minimum degree equaling 3. Moreover, two neighbors of node 0, i.e., node 1 and node 2, are connected each other. When the eavesdropper can obtain the information transmitted on (1,2), it can intercept the information from node 1 and node 2. Then combining with the information on the edge (0,18), the eavesdropper can fully infer the initial state of the 0th node. Using Algorithm 1, we derive the optimal attacking strategy, i.e., $\lambda_{1,2} = 0.5, \lambda_{0,18} = 0.5$, which results in $P = 0.25$.

Figure 10. Small-world network 4.

By setting $N = 20, K = 4, P_{reconnect} = 0.5$, we generate a small world model as shown in Figure 11. By Algorithm 1, we can obtain the optimal attacking strategy as $\lambda_{8,10} = 0.5, \lambda_{14,16} = 0.5$, with $P = 0.5$. In this case, node 10 and node 14 are connected, with their degrees being 2 and 3, respectively. When the eavesdropper get the information on edge (8,10) and (14,16), it can infer the initial information of

node 10 and 14 successfully. Due to the limited power of the eavesdropper, when $\lambda_{8,10} = \lambda_{14,16} = 0.5$, we can calculate $P_{10} = 0.25$ and $P_{14} = 0.25$ by Theorem 4. Thus, $P = P_{10} + P_{14} = 0.5$.

Figure 11. Small-world model.

Overall, we summarize the following results: (1) If there is a node whose degree is 1, the eavesdropper can infer the initial state of this node by intercepting the only edge of the node. (2) Attacking the edges connecting to the node with relatively smaller degree usually brings better eavesdropping effect. (3) The effect of privacy disclosure depends not only on the degree of the node but also on the degree of its neighbors.

5. Numerical Examples

5.1. An Example of Privacy Disclosure

As shown in Figure 12, suppose node n is the malicious node who wants to obtain the initial states of all the other nodes. Set $x(0) = [1, 3, 7, 5, 4]^T$, $\varepsilon = 1/6$. Matrices C and P_ε are

$$C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}, P_\varepsilon = \begin{bmatrix} 1/2 & 1/6 & 1/6 & 1/6 & 0 \\ 1/6 & 2/3 & 0 & 0 & 1/6 \\ 1/6 & 0 & 5/6 & 0 & 0 \\ 1/6 & 0 & 0 & 5/6 & 0 \\ 0 & 1/6 & 0 & 0 & 5/6 \end{bmatrix}.$$

The observations at step 0 and step 1 are given by the following equations

$$\begin{aligned} y(0) &= Cx(0) = [3, 7, 5, 1]^T, \\ y(1) &= CP_\varepsilon x(0) = \left[\frac{17}{6}, 6, \frac{29}{6}, 3\right]^T. \end{aligned}$$

By selecting five independent linear equations from the above, node n can obtain

$$\begin{aligned} x_n(0) &= 1, x_{j_1}(0) = 3, x_{j_2}(0) = 7, x_{j_3}(0) = 5, \\ \frac{1}{6}x_n(0) + \frac{2}{3}x_{j_1}(0) + 0x_{j_2}(0) + 0x_{j_3}(0) + \frac{1}{6}x_s(0) &= \frac{17}{6}. \end{aligned}$$

Thus, $x_s(0) = 4$ can be solved by node n . Note that node s is not directly connected to node n . This example shows that the attacker can deduce the initial states of some nodes under certain conditions, even though these nodes are not directly connected to the malicious node.

Figure 12. A case of privacy disclosure.

5.2. Convergence and Exact Average Consensus Evaluation

In this subsection, we consider a network consisting of 50 nodes as shown in Figure 13. The real initial state $x_i^r(0)$ is set as the following vector

$$x_i^r(0) = [57, 75, 90, 4, 1, 11, 28, 21, 32, 86, \\ 78, 14, 20, 3, 17, 92, 78, 23, 72, 37, \\ 22, 87, 39, 45, 41, 62, 15, 45, 84, 59, \\ 52, 58, 33, 44, 59, 30, 69, 71, 60, 3, \\ 64, 96, 37, 87, 70, 93, 90, 88, 35, 73]^T,$$

of which the average value of the entries is 51. Matrix P_ε is set following the rules in Section 2.1, namely $P_\varepsilon = I - \varepsilon L$, where L is the Laplacian matrix of the graph in Figure 13 and $\varepsilon = 1/(2 \max\{l_{ii}\})$. For each node i , by setting $m_i = 200$, We investigate the privacy preserving performance and average consensus convergence of our scheme proposed in Section 2.3 under different distributions and different variances of the random series $r_i(k)$.

Figure 13. Network topology of 50 nodes.

First, we run the scheme under different distributions of $r_i(k)$. These distributions are exponential distributions, normal distributions and uniform distributions that for each node i , the variance of $r_i(k)$ remains unchanged regardless of the change of the distributions to exclude the influence of the variances. For the exponential distribution of the random sequence in each node i , to ensure that the random sequence $r_i(k)$ satisfies (7), the parameter of the exponential distribution is set to be $1/x_i^r(0)$,

namely $r_i(k) \sim E(1/x_i^r(0))$; thus, the mean of the exponential distribution is $x_i^r(0)$ and the variance is $[x_i^r(0)]^2$. For the normal distribution of the random sequence in each node i , to ensure that the random sequence $r_i(k)$ satisfies (7) and the variance remains the same as that of the exponential distribution correspondingly, the mean of the normal distribution is set to be $x_i^r(0)$ and the variance is set to be $[x_i^r(0)]^2$, namely $r_i(k) \sim N(x_i^r(0), [x_i^r(0)]^2)$. For the uniform distribution of the random sequence in each node i , to ensure that the random sequence $r_i(k)$ satisfies (7) and the variance remains the same as that of the previous two distributions correspondingly, the lower bound of the uniform distribution is set to be $x_i^r(0) - \sqrt{3}x_i^r(0)$ and the upper bound of the uniform distribution is set to be $x_i^r(0) + \sqrt{3}x_i^r(0)$, namely $r_i(k) \sim U(x_i^r(0) - \sqrt{3}x_i^r(0), x_i^r(0) + \sqrt{3}x_i^r(0))$; thus the mean of the uniform distribution is $x_i^r(0)$ and the variance is $[x_i^r(0)]^2$. Note that, in order to eliminate the residuals, each distribution is modified by $r_i(k) := r_i(k) - [\frac{1}{m_i} \sum_{k=1}^{m_i} r_i(k) - x_i^r(0)]$. We select node 25 to show how the random sequence $r_i(k)$ is distributed under those three different distributions in Figure 14.

Figure 14. Three different distributions of $r_i(k)$ in node 25.

Under these three different distributions of $r_i(k)$, we run the scheme proposed in Section 2.3, the results are shown in Figures 15–17. It can be seen that, all the nodes' states converge to the real average of the initial condition $x^r(0)$, which is 51, under different distributions of $r_i(k)$. This also verifies Theorem 2 that the scheme ensures exact average consensus.

Figure 15. States evolution under exponential distribution of $r_i(k)$.

Figure 16. States evolution under normal distribution of $r_i(k)$.

Figure 17. States evolution under uniform distribution of $r_i(k)$.

Then, we inspect the convergence step by running the above simulation 1000 times under each distribution, respectively. The scheme is considered to reach average consensus when the variance of $x_i(k), i = 1, \dots, 50$ is smaller than 1.0×10^{-4} . At that point of time, the very step k is considered the convergence step. To eliminate the effect of randomness, the convergence steps are averaged within the 1000 times of simulation under those three distributions respectively, and are then recorded in Table 1. It can be seen that, the convergence step has very slight change under different distributions of $r_i(k)$. This is because the variances of these random sequences are designed to be the same.

Table 1. Average convergence step under different distributions.

Distribution	Convergence Step
Exponential	374.188
Normal	374.357
Uniform	374.440

Further, we inspect how the variance of the random sequence $r_i(k)$ influence the convergence step. We magnify the variances of the normal distribution and the uniform distribution by 10~1000 times, and examine the average convergence steps within 1000 simulations under these different variances. The result is shown in Figure 18. Note that, for the exponential distribution, the variance is fixed when its mean is fixed, so we cannot change the variance of $r_i(k)$ of an exponential distribution when its mean is fixed to $x_i^r(0)$.

Figure 18. Convergence step vs. variance of $r_i(k)$.

It can be seen from Figure 18 that when the scales of the variances are the same, the convergence steps are the same regardless of the distribution of $r_i(k)$. Besides, both the convergence steps of normal distribution and uniform distribution increases linearly with the increasing of the logarithm of the scale of the variances.

5.3. Privacy Preserving Performance Comparison

In this subsection, we compare the privacy preserving performances between our proposed scheme in Section 2.3 (denoted as “state-split method”) and the scheme proposed in [24] (denoted as “Manitara’s method”). For both the state-split method and the Manitara’s method, we examine the node states privacy disclosure with changing variances of the random sequence $r_i(k)$. The simulation parameters are set the same as in Section 5.2 unless stated deliberately.

In Figures 19 and 20, Manitara’s method is inspected. The left side shows the node state evolution when the variance of $r_i(k)$ is 100 times of that in Section 5.2. And the right side shows the case when the variance is 0.01 times of that in Section 5.2. It can be seen that, the privacy preserving performance is effective when the variance is large, because the node states are mixed in the random noises. However, when the variance is small, the influence of the random noises is nearly neglected. Thus, the eavesdropper in the network might obtain the initial states under a considerable accuracy when the variance of $r_i(k)$ is small.

In Figures 21 and 22, The state-split method is inspected. The left side shows the node state evolution when the variance of $r_i(k)$ is 100 times of that in Section 5.2. And the right side shows the case when the variance is 0.01 times of that in Section 5.2. It can be seen that, the privacy preserving performance of the state-split method is effective both when the variance is large or small, because the

node state always starts from the fake initial state $x_i(0)$, which is designed to be 0 to deceive the eavesdropper. The real initial state $x_i(0)$ is split and hidden in the random sequence $r_i(k)$.

The above comparison indicates that the state-split method achieves better privacy preserving performance than Manitará's method does when the variance of the adding random sequence is small. This is a desired feature especially when the system cannot provide noises of large variance. For example, in an analog system, the noise might be generated by an amplifier whose saturation voltage is limited, thus, the variance of noise provided might be relatively small.

Figure 19. Manitará's method under large variance.

Figure 20. Manitará's method under small variance.

Figure 21. State-split method under large variance.

Figure 22. State-split method under small variance.

6. Conclusions

In this paper, we proposed a privacy-preserving scheme for a distributed average consensus problem by splitting the nodes' initial states into several parts and each part is sent to the neighbors at different time steps to block the direct streaming of the node state information. We also analysed the case where the eavesdropper tries to deduce the initial states. We found that under the condition that a node and its neighbors are all the neighbors of the eavesdropper, the initial state of the node can be deduced by the eavesdropper even if the proposed privacy preserving scheme is applied. Further, in the case where an eavesdropper can intercept the data transmitted on the edges, we proposed a criterion to measure the degree of privacy leaks. Under this criterion, for the eavesdropper with limited power, we found a suboptimal attacking strategy in a ring network and a small-world network, respectively. To find the optimal attacking strategy for an arbitrary network, we have provided an optimization algorithm which can be solved by SLSQP, and obtained some heuristic results by the numerical simulations. Specifically, we have explored that the variance of the distribution of the random sequence would influence the convergence step of the consensus protocol with privacy preserving. Besides, we compared the privacy-preserving performance of our proposed scheme with

that of a previous study. The result showed that our scheme achieved better privacy preserving when the variance of the random sequence is relatively small.

Author Contributions: Conceptualization by W.Y.; Formal analysis by D.L.; Methodology by H.Z.; Writing, review and editing by D.L. and H.Z.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant (61973123, 61573143), the projects sponsored by the development fund for Shanghai talents, the Shanghai Natural Science Foundation under Grant 18ZR1409700, and the Programme of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A. Proofs of Theorems

Proof of Theorem 1. P_ε is a doubly stochastic matrix because: (1) $\mathbf{1}^T P_\varepsilon = \mathbf{1}^T - \varepsilon \mathbf{1}^T L = \mathbf{1}^T$; (2) $P_\varepsilon \mathbf{1} = \mathbf{1} - \varepsilon L \mathbf{1} = \mathbf{1}$; (3) L is symmetric, it has orthogonal decomposition $L = \varepsilon Q \Lambda_L Q^T$, where Q is an orthogonal matrix and Λ_L is a diagonal matrix with the diagonal entries being the eigenvalues of L . Thus, $P_\varepsilon = I - \varepsilon L = Q(I - \varepsilon \Lambda_L)Q^T$. Because $\mathcal{G}$ is connected, Λ_L has one and only one diagonal entry equaling zero according to Lemma 1, with $\varepsilon < \max\{l_{ii}\}$, $I - \varepsilon \Lambda_L$ has one and only one diagonal entry equaling one, with all the other diagonal entries' magnitudes less than one. Hence P_ε satisfies the conditions in Lemma 2.

In that P_ε is doubly stochastic, according to Lemma 2 we have

$$\lim_{k \rightarrow \infty} x(k) = \lim_{k \rightarrow \infty} P_\varepsilon^k x(0) = \frac{\mathbf{1}\mathbf{1}^T}{n} x(0) = \frac{1}{n} \sum_{i=1}^n x_i(0) \cdot \mathbf{1}, \quad (\text{A1})$$

which indicates that the entries of $x(\infty)$ is exactly the average of all the initial states. $\square$

Proof of Theorem 2. Denote $d(k) = [d_1(k), d_2(k), \dots, d_n(k)]^T$. The iteration in (10) can be rewritten as

$$x(k+1) = P_\varepsilon x(k) + d(k+1); \quad (\text{A2})$$

thus, according to Lemma 2, we have

$$\begin{aligned} \lim_{k \rightarrow \infty} x(k) &= \lim_{k \rightarrow \infty} [P_\varepsilon x(k-1) + d(k)] \\ &= \lim_{k \rightarrow \infty} [P_\varepsilon^2 x(k-2) + P_\varepsilon d(k-1) + d(k)] \\ &= \dots \\ &= \lim_{k \rightarrow \infty} [P_\varepsilon^k x(0) + P_\varepsilon^{k-1} d(1) + \dots + P_\varepsilon^{k-m_i} d(m_i) + P_\varepsilon^{k-m_i-1} d(m_i+1) + \dots] \\ &= \frac{\mathbf{1}\mathbf{1}^T}{n} x(0) + \frac{\mathbf{1}\mathbf{1}^T}{n} \sum_{l=1}^{\infty} d(l). \end{aligned}$$

According to (7)–(9),

$$\begin{aligned} \lim_{k \rightarrow \infty} x(k) &= \frac{\mathbf{1}\mathbf{1}^T}{n} x(0) + \frac{\mathbf{1}\mathbf{1}^T}{n} \sum_{l=1}^{\infty} d(l) \\ &= \mathbf{0} + \frac{\mathbf{1}\mathbf{1}^T}{n} \frac{1}{m_i} \sum_{l=1}^{m_i} r_i(l) \\ &= \frac{\mathbf{1}\mathbf{1}^T}{n} x^r(0). \end{aligned}$$

Recall that $x^r(0)$ denotes the real initial states; thus, the exact average consensus is achieved. $\square$

Proof of Theorem 3. For any $k \in [1, M - 1]$, it can be derived that

$$\tilde{y}(k) = \tilde{C} \sum_{i=0}^{k-1} P_{\varepsilon}^{k-1-i} d(i+1).$$

Note that $P_{\varepsilon_i} \tilde{C} = P_{\varepsilon_i}$, which means that the i th node is a neighbor of the n th node. Thus, $x_i(k)$ lies in the i th line of $y(k)$, then

$$\begin{aligned} P_{\varepsilon_i} \tilde{y}(k) &= P_{\varepsilon_i} \sum_{i=0}^{k-1} P_{\varepsilon}^{k-1-i} d(i+1) \\ &= P_{\varepsilon_i} x_i(k). \end{aligned} \quad (\text{A3})$$

At time step $k + 1$, the state $x_i(k + 1)$ can be derived from $y(k + 1)$. Hence,

$$d_i(k + 1) = x_i(k + 1) - P_{\varepsilon_i} x_i(k). \quad (\text{A4})$$

In this way, $d_i(2), d_i(3), \dots, d_i(M)$ can also be obtained. Note that $d_i(1)$ lies in $y(1) = Cd(1)$. Therefore, the sequence $d_i(k)$ of $k = 1, 2, \dots, M$ is obtained. Combing (7) and (8), $x_i^r(0)$ can be calculated. Thus, the initial state of the i th node is derived. $\square$

Proof of Theorem 4. Note that $A_t(i, j)$ denotes the attacking probability of the edge (i, j) , which equals $e_i^T A_t e_j$.

For the m th node within the neighboring set of node i , we regard that the m th node is in dangerous state if at least one edge connected to it has been intercepted. We use $1 - \prod_{v=1}^n (1 - e_v^T A_t e_m)$ to represent the probability of the initial state of the m th node exposed to the eavesdropper, thus W_i is the probability of initial states of all the neighbors of the i th node exposed to the eavesdropper. However, according to the results in Remark 3, in order to deduce the privacy information of node i , we should not only let all the neighbors of node i enter into the dangerous state, but also need to ensure that at least one edge $(i, j), j \in N_i$ is exposed. Thus, we use the gap between W_i and W_i' to denote the probability of privacy disclosure of the i th node. $\square$

References

1. Olfati-Saber, R.; Murray, R.M. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Control* **2004**, *49*, 1520–1533. [\[CrossRef\]](#)
2. Miao, G.; Li, G.; Li, T.; Liu, Y. H_{∞} consensus control for heterogeneous multi-agent via output under markov switching topologies. *Electronics* **2018**, *7*, 453. [\[CrossRef\]](#)
3. Shi, L.; Xie, L. Optimal sensor power scheduling for state estimation of Gauss–Markov systems over a packet-dropping network. *IEEE Trans. Signal Process.* **2012**, *60*, 2701–2705. [\[CrossRef\]](#)
4. Chen, Z.; Zhang, H.T. Analysis of joint connectivity condition for multiagents with boundary constraints. *IEEE Trans. Cybern.* **2013**, *43*, 437–444. [\[CrossRef\]](#) [\[PubMed\]](#)
5. Yang, W.; Wang, X.; Shi, H. Fast consensus seeking in multi-agent systems with time delay. *Syst. Control Lett.* **2013**, *62*, 269–276. [\[CrossRef\]](#)
6. Yang, W.; Wang, Z.; Zuo, Z.; Yang, C.; Shi, H. Nodes selection strategy in cooperative tracking problem. *Automatica* **2016**, *74*, 118–125. [\[CrossRef\]](#)
7. Hu, W.; Liu, L.; Feng, G. Consensus of linear multi-agent systems by distributed event-triggered strategy. *IEEE Trans. Cybern.* **2015**, *46*, 148–157. [\[CrossRef\]](#)
8. Gray, J.; Lamport, L. Consensus on transaction commit. *ACM Trans. Database Syst.* **2006**, *31*, 133–160. [\[CrossRef\]](#)
9. Lamport, L.; Shostak, R.; Pease, M. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.* **1982**, *4*, 382–401. [\[CrossRef\]](#)

10. Kefayati, M.; Talebi, M.S.; Khalaj, B.H.; Rabiee, H.R. Secure consensus averaging in sensor networks using random offsets. In Proceedings of the 2007 IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, Penang, Malaysia, 14–17 May 2007; pp. 556–560.
11. Zhang, H.; Cheng, P.; Wu, J.; Shi, L.; Chen, J. Online deception attack against remote state estimation. *IFAC Proc. Vol.* **2014**, *47*, 128–133. [[CrossRef](#)]
12. Qi, Y.; Cheng, P.; Shi, L.; Chen, J. Event-based attack against remote state estimation. In Proceedings of the 2015 54th IEEE Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 6844–6849.
13. Lei, L.; Yang, W.; Yang, C.; Shi, H. False data injection attack on consensus-based distributed estimation. *Int. J. Robust Nonlinear Control* **2017**, *27*, 1419–1432. [[CrossRef](#)]
14. Zhao, C.; He, J.; Cheng, P.; Chen, J. Analysis of consensus-based distributed economic dispatch under stealthy attacks. *IEEE Trans. Ind. Electron.* **2016**, *64*, 5107–5117. [[CrossRef](#)]
15. Kolesnikov, V. Advances and impact of secure function evaluation. *Bell Labs Tech. J.* **2009**, *14*, 187–192. [[CrossRef](#)]
16. Qin, J.; Ma, Q.; Shi, Y.; Wang, L. Recent advances in consensus of multi-agent systems: A brief survey. *IEEE Trans. Ind. Electron.* **2016**, *64*, 4972–4983. [[CrossRef](#)]
17. Fan, C.I.; Huang, S.Y.; Lai, Y.L. Privacy-enhanced data aggregation scheme against internal attackers in smart grid. *IEEE Trans. Ind. Inform.* **2013**, *10*, 666–675. [[CrossRef](#)]
18. Li, S.; Xue, K.; Yang, Q.; Hong, P. PPMA: Privacy-preserving multisubset data aggregation in smart grid. *IEEE Trans. Ind. Inform.* **2017**, *14*, 462–471. [[CrossRef](#)]
19. Kishida, M. Encrypted Average Consensus with Quantized Control Law. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami Beach, FL, USA, 17–19 December 2018; pp. 5850–5856.
20. Guo, S.; Zeng, D.; Xiang, Y. Chameleon hashing for secure and privacy-preserving vehicular communications. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *25*, 2794–2803. [[CrossRef](#)]
21. Lu, Y.; Zhu, M. Privacy preserving distributed optimization using homomorphic encryption. *Automatica* **2018**, *96*, 314–325. [[CrossRef](#)]
22. Zhou, Y.; Liu, T.; Tang, F.; Wang, F.; Tinashe, M. A Privacy-preserving authentication and key agreement scheme with deniability for IoT. *Electronics* **2019**, *8*, 450. [[CrossRef](#)]
23. Huang, Z.; Mitra, S.; Dullerud, G. Differentially private iterative synchronous consensus. In Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society, Raleigh, NC, USA, 15 October 2012; ACM: New York, NY, USA, 2012; pp. 81–90.
24. Manitaru, N.E.; Hadjicostis, C.N. Privacy-preserving asymptotic average consensus. In Proceedings of the 2013 European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 760–765.
25. Mo, Y.; Murray, R.M. Privacy preserving average consensus. *IEEE Trans. Autom. Control* **2016**, *62*, 753–765. [[CrossRef](#)]
26. He, J.; Cai, L.; Guan, X. Preserving data-privacy with added noises: Optimal estimation and privacy analysis. *IEEE Trans. Inf. Theory* **2018**, *64*, 5677–5690. [[CrossRef](#)]
27. Petrillo, A.; Pescape, A.; Santini, S. A collaborative approach for improving the security of vehicular scenarios: The case of platooning. *Comput. Commun.* **2018**, *122*, 59–75. [[CrossRef](#)]
28. Santini, S.; Salvi, A.; Valente, A.; Pescape, A.; Segata, M.; Cigno, R. Platooning Maneuvers in Vehicular Networks: A Distributed and Consensus-Based Approach. *IEEE Trans. Intell. Veh.* **2019**, *4*, 59–72. [[CrossRef](#)]
29. Zhou, H.; Yang, W.; Yang, C.; Tang, Y.; Shi, H. Optimal eavesdropping problem in privacy preserving consensus. In Proceedings of the IECON 2017-43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October–1 November 2017; pp. 5929–5934.
30. Xiao, L.; Boyd, S. Fast linear iterations for distributed averaging. *Syst. Control Lett.* **2004**, *53*, 65–78. [[CrossRef](#)]
31. Horn, R.A.; Johnson, C.R. *Matrix Analysis*; Cambridge University Press: Cambridge, UK, 2012.
32. Zappone, A.; Lin, P.H.; Jorswieck, E. Optimal energy-efficient design of confidential multiple-antenna systems. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 237–252. [[CrossRef](#)]
33. Wang, X.F.; Chen, G. Complex networks: small-world, scale-free and beyond. *IEEE Circuits Syst. Mag.* **2003**, *3*, 6–20. [[CrossRef](#)]
34. Gale, D.; Kuhn, H.W.; Tucker, A.W. Linear programming and the theory of games. *Act. Anal. Prod. Alloc.* **1951**, *13*, 317–335.

35. Kraft, D. Algorithm 733: TOMP–Fortran modules for optimal control calculations. *ACM Trans. Math. Softw. (TOMS)* **1994**, *20*, 262–281. [[CrossRef](#)]
36. Watts, D.J.; Strogatz, S.H. Collective dynamics of small-world networks. *Nature* **1998**, *393*, 440–442. [[CrossRef](#)]
37. Hong, H.; Choi, M.Y.; Kim, B.J. Collective synchronization on small-world networks. *Phys. Rev. E* **2002**, *65*, 026139. [[CrossRef](#)]
38. Wang, X.F.; Chen, G. Synchronization in small-world dynamical networks. *Int. J. Bifurc. Chaos* **2002**, *12*, 187–192. [[CrossRef](#)]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).