Privacy preserving consensus under interception attacks

Wen Yang, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

Abstract—In this paper, we consider the attack problem in privacy-protected networks. First, we introduce a consensus protocol with privacy preserving, where each node hide their initial state into a set of random sequences, and then inject the sequences into the process of consensus. Due to the protocol will fail in the special network topology, we consider the situation where an attacker can intercept the data transmitted on the edges, and obtain an indicator which can measure the degree of privacy leakage. In ring and small-world networks, from the perspective of the attacker with limited power, we find the optimal attacking strategy to maximize the probability of the privacy disclosure. Finally, we provide an optimization algorithm to verify the results of the analysis, and we can use this algorithm to find the optimal attack strategy for any other networks.

Index Terms—Privacy preserving; Consensus protocol; Interception attack

1 INTRODUCTION

As a typical distributed algorithm, consensus protocol has been widely used in many fields, including control engineering, computer science, system biology and physics, where each node exchanges information with its linked neighbors such that the whole network reaches an agreement. Consensus problem has attracted many researchers in recent decades, from protocol design, convergence analysis to network performance optimization, it has been studied extensively, see [1–5].

Recently, wireless security has received a lot of research attention. Privacy information leakage is one of the important topics. For example, a distributed network can achieves consensus when each node exchanges its state with it's neighboring nodes, at the same time, the initial states of all the nodes are broadcast to the whole network, see [6], [7]. In practical applications, such as decision-making scenes, how to keep the participant's information secrete is a key problem while achieving the goal of reaching the agreement. Therefore, many researchers focus on the privacy preserving problem. In [8], the author proposed the notation of differential privacy (DP), and gived a mathematical definition of privacy level. Huang applied the DP approach to the problem of consensus protocol, [9] proposed a privacy protection protocol by adding independent and exponentially decaying Laplacian noise in the process of consensus updating. However, there is a problem with the protocol based on DP, which is unable to converge to the accurate mean of the initial state. Therefore, in some cases, the DP method can not be used. For maximum consensus, in [10], the authors devised a protocol to send a special offset before sending the actual initial state. For average consensus, Mo et al.[6] proposed a privacy preserving algorithm by adding well-designed noise on its state, then the author analysis the convergence rate of this consensus algorithm, and prove the algorithm achieves minimum privacy breach. In [11], the author presented a privacy preserving consensus algorithm that each node adds an arbitrary offset value to the result of iteration, where the total offsets cancel themselves out in the end. Moreover, in [6, 11], they all put restrictions on network structure, i.e., if a node and all of it's neighbors belong to the malicious node's neighboring node set, the node's initial state can be speculated by the malicious node.

Currently, plenty studies focus on cyber attacks that happen frequently in real applications, see [12–15]. However, less studies have been directed to the privacy preserving under malicious attacks. In recent years, complex network has been widely used to describe the system in nature. [16] proposed an interesting small-world network model, referred to as WS small-world model. Many profound studies are based on the small-world model, see [17, 18]. In [19], the author study how the structures of networks affect their synchronization, In this article, we will use small-world network as one of our attack model.

In this paper, we first introduce an average consensus protocol with privacy preserving proposed in [11], and point out one of its defects, when the network topology meet certain conditions, there will be leakage of privacy information. Once there is an attacker in the network, it can intercept the data transmitted on the edges, and then can deduce the initial state of the nodes in the network. If the attacker's energy is limited, we study the relationship between the degree of the private information leakage and the energy allocated on each edge. Specific to ring network and small-world networks, we try to find the optimal attacking strategy, resulting in the greatest degree of privacy leakage. The main contributions of this article are summarized as: 1) We get a specific mathematical formula that can calculate the degree of privacy leakage; 2) In the ring network and small-world model, we theoretically analyze how to allocate the attacker's energy, which can maximize the attack effect; 3) We provide an algorithm that can be used to find the optimal attack strategy in any network; 4) By the simulation, we obtain the relationship between the convergence step and the variance of the added noise.

Notation: $\mathbb{R}^{n\times m}$ denotes the set of n by m matrices. e_i denotes the ith canonical basis in $\mathbb{R}^n$ with a 1 in the ith entry and zeros elsewhere.

- *M. Shell was with the Department of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332. E-mail: see http://www.michaelshell.org/contact.html*
- *J. Doe and J. Doe are with Anonymous University.*

Manuscript received April 19, 2005; revised August 26, 2015.

2 PROBLEM FORMULATION

In this paper, we model a sensor network as an undirected graph $G = (V, E)$ with the nodes $V = 1, 2, ..., n$ being the sensors and the edges $E \subset V \times V$ representing the communication links. Denote the set of neighbors of node i by $N(i) = \{j \in V : (i, j) \in E, j \neq i\}$. Each node can communicate with its neighbors. The interconnection topology of the network is described by Laplacian matrix $L = [l_{ij}]$, where $l_{ii} = -\sum_{j \in N_i} l_{ij}$ and $l_{ij} = -1$ if $(i, j) \in E$; otherwise, $l_{ij} = 0$. Assume that G is connected, and let the initial state of each node is $x_i(0)$. In the network, each node updates its state according to the following consensus protocol (see [1]),

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N(i)} a_{ij}(x_j(k) - x_i(k)). \tag{1}$$

By collecting all the states of the nodes, we define $x(k) \triangleq [x_1(k), \ldots, x_n(k)]'$. The single node dynamics (1) can be represented as the following vector form

$$x(k+1) = P_\varepsilon x(k), \tag{2}$$

where $P_\varepsilon = [a_{ij}]$ (called as *Perron matrix*) is satisfied with

$$P_\varepsilon = I - \varepsilon L. \tag{3}$$

We denote the node with maximum number of neighbors of in G as $d_{max} = max\{l_{ii}\}$, then $\varepsilon \in (0, 1/d_{max})$ is satisfied to guarantee the convergence of protocol (2), see [1]. Moreover, based on the existing results in [1], the node dynamics (2) asymptotically solves the average consensus problem if and only if G is balanced.

In this paper, we study the privacy preserving of consensus protocol under typical attacks. First, we assume that node n wants to infer the initial states of all the other nodes in the network. We denote the neighboring set of node n as $N(n) = \{j_1, \ldots, j_m\}$, and define an observation matrix C as $C = [e_{j_1}, ..., e_{j_m}, e_n]^T \in \mathbb{R}^{(m+1)\times n}$, The output equation of the whole network can be rewritten as the following,

$$y(k) = Cx(k). \tag{4}$$

At time step k, node n's information set is

$$I(k) = \{x_n(0), y(0), ...y(k)\}. \tag{5}$$

Lemma 1. ([6]) Assume node n knows P_ε and C matrices and all variables in $I(k)$ at time step k, the consensus algorithm is deterministic and node n can perfectly infer $\zeta' x(0)$, given that $\zeta \in \mathbb{R}^n$ lies in the observable space of (P_ε, C).

To protect the privacy of the initial states of nodes while enforce $x(k)$ converges to the average of $x(0)$, [11] proposes a consensus protocol with privacy preserving scheme in a weighted and directed network. In this paper, we focus on analyze the network performance of privacy preserving consensus protocol under a typical attack, and then simplify this privacy preserving scheme to the case when the network is unweighted and undirected.

The steps of **privacy preserving scheme** are listed as follows:

1) Each node i generates a random number m_i, which satisfied with $2 \leq m_i \leq M$, M is a positive real number.
2) Each node generates m_i random number which are denoted as $r_i(1), r_i(2), ..., r_i(m_i)$, with their average value is $x_i(0)$.
3) Clear the initial state of each node to $x_i(0) = 0$, we mark the real initial state as $x_i^r(0)$, thus we can define $x^r(0) = [x_1^r(0), x_2^r(0)...x_n^r(0)]^T \in \mathbb{R}^n$.
4) At time step k, inject the number $d_i(k)$ into the state of each node, where

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

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

$$x_i(k+1) = x_i(k) + \varepsilon \sum_{j \in N_i} a_{ij}(x_j(k) - x_i(k)) + d_i(k+1), \tag{6}$$

Define $d(k) = [d_1(k), ..., d_n(k)]^T \in \mathbb{R}^n$. The vector form of (6) in conjunction with Eq. (4) can be rewritten as

$$\begin{cases} x(k+1) = P_\varepsilon x(k) + d(k+1), \\ y(k) = Cx(k). \end{cases} \tag{7}$$

Based on the above scheme, the real initial states of nodes are hidden in several parts, thus the privacy of $x^r(0)$ can be protected.

3 MAIN RESULT

3.1 Privacy Analysis

With a standard consensus algorithm, the initial states of nodes can be easily speculated by any other nodes. However, with the proposed privacy preserving algorithm (6), all the nodes reach average consensus finally, and the real initial states of nodes are hidden in $d(k)$. Hence, the problem of deducting $x^r(0)$ actually can be transformed into computing $d(k), k \in [1, M]$. First, we obtain the solution of (7) by iterative method.

$$\begin{aligned} x(k) =& P_\varepsilon^k x(0) + \sum_{i=0}^{k-1} P_\varepsilon^{k-1-i} d(i+1), \\ y(k) =& CP_\varepsilon^k x(0) + C \sum_{i=0}^{k-1} P_\varepsilon^{k-1-i} d(i+1). \end{aligned} \tag{8}$$

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

$$Y = FD \tag{9}$$

where,

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

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

For a fixed network topology G, F is a constant matrix. In order to derive the the real initial state $x^r(0)$ from D, we need to utilize the information Y to solve the equation (9). Let us define

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

where $\mathbf{0} \in \mathbb{R}^n$ is zero vector. We further define:

$$\tilde{C} = [s_1, s_2, ...s_n]^T \in \mathbb{R}^{n\times n},$$
$$\tilde{y}(k) = \tilde{C}x(k) \in \mathbb{R}^{n\times 1},$$

where $\tilde{y}(k)$ is easily obtained from $y(k)$ for a given C.

Using all the above arguments, we obtain the following result.

Theorem 1. If the ith line of P_ε noted as P_{ε_i} satisfies $P_{\varepsilon_i}\tilde{C} = P_{\varepsilon_i}$, then node n can speculate the initial state of node i successfully.

Proof 1. For any $k \in [1, M-1]$, we have

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

Note that $P_{\varepsilon_i}\tilde{C} = P_{\varepsilon_i}$. We know that node i is one neighbor of node n, thus $x_i(k)$ lies in the ith line of $y(k)$, then

$$\begin{aligned} P_{\varepsilon_i}\tilde{y}(k) &= P_{\varepsilon_i}\sum_{i=0}^{k-1} P_\varepsilon^{k-1-i} d(i+1) \\ &= P_{\varepsilon_i} x_i(k). \end{aligned} \tag{10}$$

At time step $k+1$, we can derive $x_i(k+1)$ from $y(k+1)$. Hence,

$$d_i(k+1) = x_i(k+1) - P_{\varepsilon_i} x_i(k). \tag{11}$$

By this way, we can also obtain $(d_i(2), d_i(3)...d_i(M)$. Note that $d_i(1)$ lies in $y(1) = Cd(1)$. Finally, we can obtain the initial state of node i. Thus the proof. ■

As mentioned above, there still exists the possibility that node i's initial state can be speculated under a special structure of network $N(i)\bigcup i \subseteq N(n)\bigcup n$. If there exists an attacker who can intercept the information flow between pairs of nodes, then the initial states of parts of nodes could be deducted. Therefore, the performance of privacy preserving drops greatly. On the other hand, the attacker is usually equipped with limited power, which implies that it needs to allocate the power to launch an attack reasonably according a certain scheme. To design effective strategies to defend against the attacks, it is necessary to study the attacking way of the attacker. In the following, we try to find an optimal attacking strategy from the perspective of the attack, and then in turn help us design strategy to preserve the privacy better.

3.2 Attack Analysis

In this subsection, we consider the case when an attacker with limited power exists in the network. Suppose that the probability of the attacker intercepting the information transmitted on the edge between node i and node j as $\lambda_{i,j}$. Recall that the network is undirected, it's unnecessary to distinguish $\lambda_{i,j}$ and $\lambda_{j,i}$, i.e., $\lambda_{i,j} = \lambda_{j,i}$. To make the statement more clearly, we use $g_{i,j}$ to represent the edge in the network, where $g_{i,j}$ and $g_{j,i}$ represent the same edge. We assume that $\lambda_{i,j}$ depends on the power $\mu_{i,j}$ allocated for the attack on the edge $g_{i,j}$, where $\lambda_{i,j} = \kappa \cdot \mu_{i,j}$. Define the power constraint of the attacker as $\sum_{g_{i,j}\in E} \mu_{i,j} = T$, for all $i, j = 1, \ldots, n$. Here, we set $\kappa = \frac{1}{T}$, and then $\sum_{g_{i,j}\in E} \lambda_{i,j} = 1$.

As discussed above, we know that the privacy of node i will be exposed in some situations. In Fig. (1), let $i = 1$.

1) When the information passed on edge $g_{1,2}$ and $g_{1,5}$ have been exposed, at time step k, the attacker can easily utilize the obtained knowledge to predict the state of node 1 at time step $k+1$ 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 $k+1$, $x_{1,k+1}$, we can further obtain the injected value which equals to$x_{1,k+1} - \hat{x}_{1,k+1}$. The attacker can derive the injection value of each step, then the initial state of node 1 will be exposed.
2) When the information passed on the edge $g_{1,2}$ and $g_{5,6}$ have been exposed, at time step k, because of the information sent from node 5 to it's neighbors are identical, the attacker can speculate all the messages sent to node 1, then it can complete the speculating mission in the similar way.p

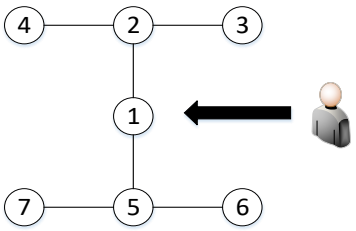


Fig. 1: wireless network

As discussed above, once the network is under the situation similar to the case 2), then the privacy of the states will be divulged. For example, the information flow of edge $g_{1,5}$ and $g_{2,4}$'s is exposed.

Remark 1. Suppose that there exists an attacker who can attack the network. If at least the information of one edge $g_{i,j}, j \in N_i$ has been intercept, and for each node $j, j \in N_i$, at least the information of one edge $g_{j,s}, s \in N_j$ has been exposed, then the attacker can infer the initial state of node i successfully.p

In the following, we further consider the privacy preserving problem when the network is under attacks. In a random network, we choose an edge to attack with identical power at each time step, i.e., the information transmitted on the chosen edge is intercepted by the attacker successfully. In the case when the allocated power is zero, we regard the edge does not suffer any attack. Then we denote the adjacency matrix of the attacked network as A_d, $A_d[i,j] = 1$ represents a connected edge between node i and j, then we can define the attacking matrix A_t, which satisfies

$$A_t[i,j] = \begin{cases} \lambda_{i,j}, & if\ A_d[i,j] = 1 \\ 0, & otherwise, \end{cases}$$

Notice $\lambda_{i,j} = \lambda_{j,i}$, so $A_t = A_t^T$.

To investigate the influence of the attacker on the privacy preserving, we define P_i as the probability of privacy leakage of node i, $i = 1, 2, \ldots, n$, i.e., the probability of the attacker deducts the initial state of node i successfully.

Theorem 2. When the network is attacked by an attacker who can intercept the information transmitted on the edge, the probability of privacy leakage of node i, $i = 1, 2, \ldots, n$ can be calculated as,

$$\begin{aligned} P_i &= W_i - W_i^{'}, \\ W_i &= \prod_{m\in N(i)} [1 - \prod_{t=1}^{n}(1 - e_t^T A_t e_m)], \\ W_i^{'} &= \prod_{m\in N(i)} [1 - \prod_{t=1, t\neq i}^{n} (1 - e_t^T A_t e_m)] \end{aligned} \tag{12}$$

Proof 2. Note that $A_t(i,j)$ denotes the probability of edge between i and j attacked, actually, which equals to $e_i^T A_t e_j$. According to Remark (1), if at least of one element at the i's line in A_d does not equal to zero, it means that at least one edge exists.

The probability of at least one edge connected to node m is intercepted equals to $1-\prod_{t=1}^{n}(1-e_t^T A_t e_m)$, and the probability of at least one edge between m and it's neighbors except node i exposed equals to $1-\prod_{t=1,t\neq i}^{n}(1-e_t^T A_t e_m)$. Thus, W_i and $W_i^{'}$, the former represents that each node in $N(i)$ has at least one leaked edge, and similarly, $W_t^{'}$ has a quite similar format with W_t, but it is worth noticing that in $W_t^{'}$ only considers the neighboring node's edge except $g_{i,m}$. Finally, the gap between W_i and $W_i^{'}$ equals to the probability of the leakage of node i's initial state. ■

We denote the degree of the network's privacy being exposed as $P=\sum_{i=1}^{n}P_i$. If there exists an attacker with limited power, we are interested in finding the optimal attacking strategy of the attacker resulting in the maximal P, and then protect the network security more effectively. In the network, there are some edges who are prone to privacy leaks when compared with the other edges, in the following, we formulate an optimization problem to find those edges in two typical networks, i.e., ring network and small world network.

$$\begin{aligned}(P_0): \quad & \max_{A_t} P \\ s.t. \quad & \sum_{g_{i,j}\in E}\lambda_{i,j}=1 \\ & 0\leq\lambda_{i,j}\leq 1\end{aligned} \tag{13}$$

1) **Case 1: ring topology (see Fig. 2)**

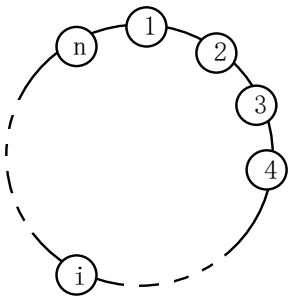


Fig. 2: Ring network

In the ring network, the position of each node is identical. According to Theorem 2, we derive P_i as follows:

$$\begin{aligned}P_i &= W_i - W_i^{'} \\ W_i &= H_{i-2}\cdot H_i \\ H_i &= 1-(1-\lambda_{i,i+1})(1-\lambda_{i+1,i+2}) \\ W_i^{'} &= \lambda_{i-2,i-1}\cdot(1-\lambda_{i-1,i})\cdot(1-\lambda_{i,i+1})\cdot\lambda_{i+1,i+2}\end{aligned} \tag{14}$$

The equation (14) can be rewritten as

$$\begin{aligned}P_i = {} & \lambda_{i-1,i}\cdot\lambda_{i,i+1}+\lambda_{i-1,i}\cdot\lambda_{i+1,i+2}+\lambda_{i-2,i-1}\cdot\lambda_{i,i+1} \\ & -\lambda_{i-2,i-1}\cdot\lambda_{i-1,i}\cdot\lambda_{i,i+1}-\lambda_{i-1,i}\cdot\lambda_{i,i+1}\cdot\lambda_{i+1,i+2}\end{aligned} \tag{15}$$

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,

$$\begin{aligned}P &= P_1+P_2+P_3...+P_n \\ &= \sum_{i=1}^{n}\lambda_i\cdot\lambda_{i+1}+2\sum_{i=1}^{n}\lambda_i\cdot\lambda_{i+2}-2\sum_{i=1}^{n}\lambda_i\cdot\lambda_{i+1}\cdot\lambda_{i+2}\end{aligned} \tag{16}$$

Here, we consider two cases,

1) when the attacker only attacks two edges, it means only two variables in $\lambda_1,\lambda_2,...\lambda_n$ are not equal to zero. The attacker has many choices, for example, it can attack two edges which connect to a same node, so that λ_i and λ_{i+1} are not zero, we can obtain the degree of the network's privacy being exposed $P=\lambda_i\cdot\lambda_{i+1}$, the constraint as $\lambda_i+\lambda_{i+1}=1$. The attacker can also choose two edges who have the same neighboring edge, where λ_i and λ_{i+2} are not zero, we can calculate $P=2\lambda_i\cdot\lambda_{i+2}$ with the constraint as $\lambda_i+\lambda_{i+2}=1$. Besides the above two cases, the attacker can also choose two edges that they don't connect to each other, and they also don't have any same neighboring edge, where λ_i and λ_{i+k} are not zero, $k\in(2,n-2)$, in these cases, we can obtain $P=0$. According to all the above arguments, it is easy to derive the optimal attacking strategy corresponding to $\lambda_i=\lambda_{i+2}=0.5$, and the maximum $P=0.5$.
2) when the attacker choose three edges, we can find the optimal attacking strategy using the similar above deductive method. When the attacker choose three edges to let $\lambda_i,\lambda_{i+1},\lambda_{i+2}$ are not zero, we can obtain $P=\lambda_i\cdot\lambda_{i+1}+\lambda_{i+1}\cdot\lambda_{i+2}+2\lambda_i\cdot\lambda_{i+2}-2\lambda_i\cdot\lambda_{i+1}\cdot\lambda_{i+2}$ and the constraint is $\lambda_i+\lambda_{i+1}+\lambda_{i+2}=1$. Obviously, the index of i will not influence the optimal value of P, without loss of generality, we let $i=1$ to simplify the notation. The optimization problem (13) in this case can be rewritten as:p

$$\begin{aligned}\max_{\lambda_1,\lambda_2,\lambda_3} \quad & \lambda_1\cdot\lambda_2+\lambda_2\cdot\lambda_3+2\lambda_1\cdot\lambda_3-2\lambda_1\cdot\lambda_2\cdot\lambda_3 \\ s.t. \quad & 0\leq\lambda_1,\lambda_2,\lambda_3\leq 1 \\ & \lambda_1+\lambda_2+\lambda_3=1\end{aligned} \tag{17}$$

In this paper, we use Kuhn-Tucker Conditions to solve the above 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 can be noted as $\Phi(\lambda)=\lambda_1+\lambda_2+\lambda_3-1=0$, and we use Γ to denote the optimization function. Then

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

Finally, we can get the result as $\lambda_1=\lambda_3=0.5,\lambda_2=0$ and $P=0.5$. When the chosen three edges satisfy that

two edges are connect to each other, and the third one has a same neighbor with the first or second edge, where $\lambda_i, \lambda_{i+1}, \lambda_{i+3} \neq 0$, we can obtain $P = \lambda_i \cdot \lambda_{i+1} + 2\lambda_i \cdot \lambda_{i+2}$, and $\lambda_i + \lambda_{i+1} + \lambda_{i+3} = 1$, when we let $i = 1$, the optimal strategy is $\lambda_2 = \lambda_4 = 0.5$, and $P = 0.5$. When the chosen three edges are not direct connect to each other, but each of them has a common neighbor with another edge, where $\lambda_i, \lambda_{i+2}, \lambda_{i+4} \neq 0$, and $\lambda_i + \lambda_{i+2} + \lambda_{i+4} = 1$, we can calculate $P = 2\lambda_i \cdot \lambda_{i+2} + 2\lambda_{i+2} \cdot \lambda_{i+4}$, when $i = 1$, the optimal strategy is $\lambda_3 = 0.5, \lambda_1 + \lambda_5 = 0.5$, and $P = 0.5$. Once there is an chosen edge don't have common neighbor with others, we know that the allocation of this edge on this edge is a waste, so the analysis can be transformed into attacking on two edges. By the analysis, we can conclude that when the attacker choose three edges, the optimal strategy is $\lambda_{i+2} = 0.5, \lambda_i + \lambda_{i+4} = 0.5$, and $P = 0.5$.

The above analysis are based on the attacker allocated its energy to two or three edges. If the attacker can arbitrarily allocated its energy to any number of edges, we want to find the most destructive attack strategy. To solve this problem, We use the SLSQP (Sequential Least Squares Programming optimization algorithm) algorithm proposed in [20]. Since the optimization algorithm may fall into the local optimum, the search result will depend on the position of the starting point, we search from the different starting positions separately to ensure that the optimal solution is as close as possible to the global optimization solution. The algorithm flow is as follows:

Algorithm 1 *Optimization algorithm*

1) Generate the network topology to be analyzed.
2) Each non-zero element in A_t is treated as a variable $v_i \in [0, 1]$, and denote $V = [v_1, v_2, ...v_n]$.
3) Write down the relationship between the extent of privacy leakage P and each variable, and use $P_{optimal} = 0$, $V_{optimal} = [0, 0, ...0]$ to save the final optimization results.
4) In the kth search, set the initial value of each variable as V_0. Then use SLSQP algorithm to find the optimal result, if the result is found successfully, we denote the maximum P as $P_{optimal}(k)$, and the corresponding V is $V_{optimal}(k)$. If $P_{optimal}(k) > P_{optimal}$, then we update $P_{optimal}$ and $V_{optimal}$ with $P_{optimal}(k)$ and $P_{optimal}(k)$.
5) If $k + 1 < 1000$, then generate a new initial set $V_0(k + 1)$, and repeat the (4) step, or end the optimization process.

Where $V_0(k)$ is generated as follows:

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

For the 100-edge ring network, through the above optimization algorithm, we find the maximum $P = 0.5$, and the optimal strategy satisfied $\lambda_{i+2} = 0.5, \lambda_i + \lambda_{i+4} = 0.5$. The conclusion have some implications, for example, we can implement a special protection strategy for such structure, so as to protect the network security.

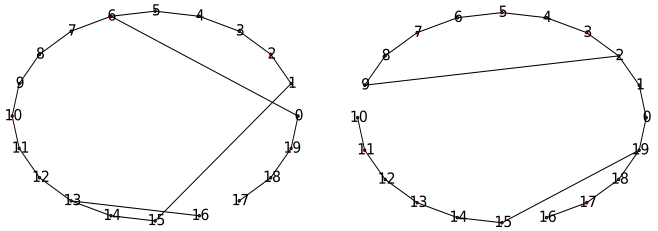
2) **Case 2: small-world topology (see Fig. 3,4)**

Using this optimization algorithm, we can not only study the ring network, but also other random network topology, Then the simulation results can be used to verify the theoretical analysis.

According to the results in Remark 1, for a general network topology under privacy protection, if the attacker want to get the initial state of node i, it should guarantee that at least the information of one edge connected to node j has been exposed,$j \in N_i$, and at least the information of one edge connected to node i has been intercepted, then the attacker can infer the initial state of node i successfully.

If the attacker has limited energy, satisfied $\sum_{g_{i,j} \in E} \lambda_{i,j} = 1$, $\lambda_{i,j}$ is the probability of acquiring the channel information between i and j. Though the rational allocation of the energy, the attacker can make the damage to the maximum, i.e., to maximize the degree of access to the network's private information.

By the result in Remark 1, obviously, if there is a node in the network with degree equal to 1, ie, there is only one edge connected to this node. It's only edge is the only channel for the node to communicate with the outside world, as long as this channel is eavesdropped, the attacker can perfectly infer the initial state of this node. For other nodes with degree grater than 1, the attacker should eavesdrop on the information in multiple channels, then the difficulty of attack will increase and the effect will be worse. Therefore, when there is a node with degree equal to 1, attacking the edge of this node is the best strategy. In the following, this inference is confirmed by simulation of small-world model.



(a) (b)

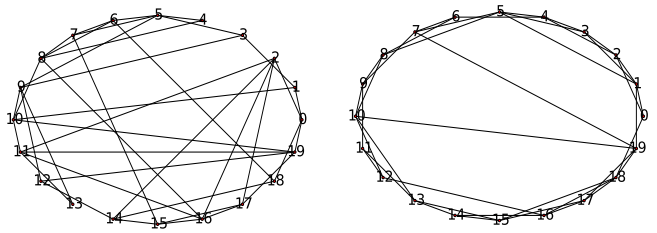
Fig. 3: Small World model 1

As we all know, The small-world Model is generated by a special regular network, and the generation rule is to cut some edges and randomly reconnect with other nodes with a certain probability. There are three indicators in the small-world model: the number of nodes N, the number of initial edges of each node K, the probability of reconnection $P_{reconnect}$. In Fig. 3, two small-world model both are generated under condition $N = 20, K = 2, P_{reconnect} = 0.1$. Using the optimization algorithm proposed earlier, we can find the optimal attack strategy in these network. In Fig. 3a, the optimal result given by algorithm is $P = 1$, the strategy satisfied $\lambda_{13,16} + \lambda_{17,18} = 1$, notice where $\lambda_{i,j}$ represent the probability of the attacker intercepting the information transmitted on the edge between node i and node j. In Fig. 3b, the optimal result given by algorithm is $P = 1$, the strategy satisfied $\lambda_{10,11} + \lambda_{16,17} = 1$.

From the simulation result in Fig. 3, we can see that the attack on the edge of the node which's degree equals to 1 can bring the best attack effect, resulting in the greatest degree of privacy leakage. It should be noted that since the degree of leakage of the entire network is defined as the sum of the probabilities of private leaks of each node, the energy can be randomly assigned to all edges satisfying the condition. For example, In Fig. 3a, the degree of node 16 and node 17 is 1, when the probability of obtaining the information on the unique edge of node 16 is $\lambda_{13,16}$, by theorem 2, we can calculate $P_{16} = \lambda_{13,16}$, so the $P = P_{16} + P_{17} = \lambda_{13,16} + \lambda_{17,18} = 1$.

In real life, this phenomenon is easy to explain. In a social network composed of a number of participants, each participant has unique information, if such a network use the privacy protection strategy in this article, that is, each participant divide the information into several parts and then transmitted, to ensure the privacy of themselves. If a participant has only one channel communicating with the outside world, the eavesdropper can fully monitor the dynamics of the participant as long as he can get information flowing through the channel. For other participants with multiple information exchange channels, it is necessary to monitor multiple channels at the same time in order to achieve the purpose of obtaining the initial information of the participant, at the same time, the cost of eavesdropping must be increased. So when the initial information of each participant is equally important, the eavesdropper is certainly more willing to eavesdrop on those participants who have only single information channel.

When there is no degree 1 node in the network, the attacker will go looking for nodes whose degree is relatively small, to attack its edges or its neighbor's edges. By the result in Remark 1, we know that when we want to attack node i, i.e., to obtain the initial state of node i, the attacker can not only attack i's edges directly, but also can eavesdrop i's neighbor node's edges to get some important information. So there will be a very interesting situation, that is, when i's neighbors are connected to each other, they share some information channel. When the attacker obtain the information passed by such channel, which is equivalent to obtaining the edge information of two node j, where $j \in N_i$. Then, we use small-world model to validate these analyzes.



(a) (b)

Fig. 4: Small World model 2

In Fig. 4, Fig. 4a is generated under condition $N = 20, K = 4, P_{reconnect} = 0.2$, the optimal attack strategy given by the algorithm is $\lambda_{4,5} = 0.5, \lambda_{4,8} = 0.5$, and the optimal result is $P = 0.25$. By analyzing the network topology in Fig. 4a, we can find that the 4 is the only node with degree equals to 2, and the degree of other nodes is greater than or equal to 3. The optimal algorithm finally choose to attack the two edges of node 4, and the energy is evenly distributed to achieve the optimal effect. Fig. 3b is generated under condition $N = 20, K = 4, P_{reconnect} = 0.5$, the optimal attack strategy given by the algorithm is $\lambda_{1,2} = 0.5, \lambda_{0,18} = 0.5$, and the optimal result is $P = 0.25$. In this network topology, we can see the minimum degree of the nodes is 3, and several nodes' degree is 3, such as node 0,6,11 etc.. 0th node's two neighbor 1 and 2 connect to each other, they share an information channel, when the attacker can obtain the information in this channel, it can get the information send out by node 1 and 2, then combined with the information in edge $(0, 18)$, the attacker can fully infer the initial state of the 0th node.

When several nodes' degree are relatively small, and they connect to each other, we should make a logical choice, so that we can get the privacy of these nodes at the same time.

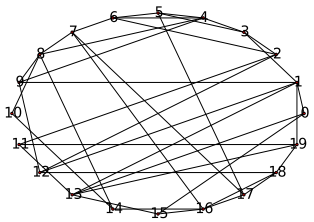


Fig. 5: Small World model 3

In Fig. 5, the network is generated under condition $N = 20, K = 4, P_{reconnect} = 0.5$, the optimal attack strategy given by the algorithm is $\lambda_{8,10} = 0.5, \lambda_{14,16} = 0.5$, and the optimal result is $P = 0.5$. In this topology, node 10's degree is 2, and node 14's degree is 3, they connect to each other. When the attacker obtain the information on edge $(8, 10)$ and $(14, 16)$, by the result in Remark 1, the attacker can perfectly guess the initial information of node 10 and 14. Due to the limited energy of the attacker, when the $\lambda_{8,10} = 0.5, \lambda_{14,16} = 0.5$, we can calculate $P_{10} = 0.25$ and $P_{14} = 0.25$ by Theorem 2. So in the case of under attacks, the degree of privacy leakage in this network is $P = P_{10} + P_{14} = 0.5$.

4 Numerical Examples

In this section, we consider a network with 5 nodes as Fig. (6) shows. The real initial state $x_i^r(0)$ of node 1 to node 5 are set as $2, 3, 4, 7, 9$, respectively. First, we verify the convergence of the proposed consensus protocol with privacy preserving scheme. As Fig. 7, 8, 9 show, all the nodes achieve the average consensus under difference cases. As above mentioned, to hide the initial state of each node, we set the initial state of all the nodes in the algorithm as zero, and hide the information of the initial state of node i in the random offset $r_i(1), r_i(2), ..., r_i(m_i), i = 1, \ldots, 5$. Intuitively, the distribution of $r_i(k)$ will influence the network performance, such as the convergence rate. Next, we are going to investigate the relation between the convergence rate and the distribution of random offset. Set $M = 20$, and m_i is subject

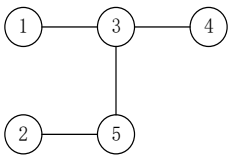


Fig. 6: Network Structure

to uniform distribution between $[2, M]$. We simulate the dynamic process of networked system when $r_i(k)$ obey uniform, Gaussian and modified exponential distribution, respectively. To make the result can be comparable, we let three kinds of distribution have the same expectation $x_i^r(0)$, and the length of distribution interval is close to 200. In Gaussian and modified exponential distribution, we require 99.985% random data are in this field. For the given expectation, the distribution interval of standard exponential distribution is fixed. Thus, we need to obtain a modified exponential distribution generated as follow. Firstly, we restrict the length of the standard exponential distribution interval close to 200. Then, we calculate the gap between $x_i^r(0)$ and the expectation of this

distribution. Finally, each random number's value subtract the gap value so that the expectation of this distribution equals to $x_i^r(0)$. As shown in Fig. 7, $r_i(k)$, $i = 1, \ldots, 5$ satisfies the uniform distribution with its mean equals to the initial state of node $x_i^r(0)$, and the range of $r_i(k)$ is $[-100 + x_i^r(0), 100 + x_i^r(0)]$. Similarly, in Fig. 8, $r_i(k)$ satisfies the gaussian distribution with $r_i(k) \sim N(x_i^r(0), 1000)$. In Fig. 9, $r_i(k)$ is subject to the modified exponential distribution with expectation $x_i^r(0)$ and variance 992.25.

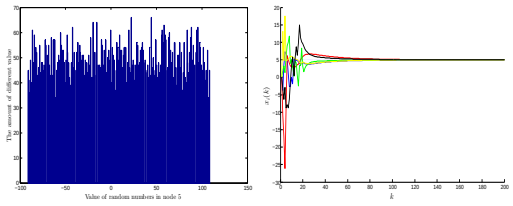


Fig. 7: Uniform distribution and consensus process

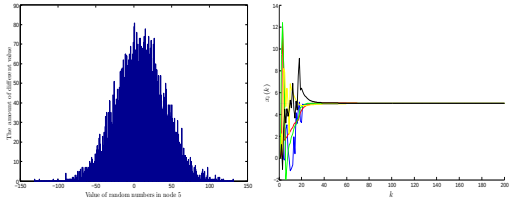


Fig. 8: Gaussian distribution and consensus process

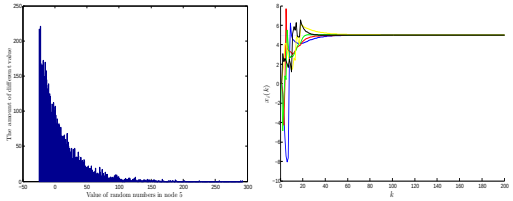


Fig. 9: Modified exponential distribution and consensus process

Here, similar to the definition of settling time in classical control theory, we set the convergence time of reaching average consensus as the time at which the average states of all the nodes enters into $[4.9, 5.1]$. For different distribution of $r_i(k)$, all the simulations are averaged by 2000 times, and the results are listed in Table 1. Obviously, all the nodes converge to the mean of the initial states despite of different distribution of $r_i(k)$. However, the convergence time is totally different. As we can see, the uniform distribution case leads to the maximum convergence time when compared with the other two cases, and the time in case of Gaussian and modified exponential distribution is same. Then, we calculate the variance in above distributions as 3333.33, 1000 and 992.25, the latter two are closer. To verify the the influence of variance of distribution, we add a case that $r_i(k)$ subject to uniform distribution between $[-55 + x_i^r(0), 55 + x_i^r(0)]$, i.e., the variance is 1008.33, then we can obtain the convergence time 61. Thereby, despite of the difference between distribution form, convergence time is only related to the variance of distribution.

Distribution	Convergence steps
Uniform	70
Gaussian	61
Modified exponential	61

TABLE 1: Convergence step

5 Conclusion

In this paper, we have studied the attack problem in the network with privacy protection. We have shown an proposed privacy preserving algorithm, where each node add well-designed noise while update their value in the process of consensus, and there is a defect in the algorithm, when the network topology satisfied an specific condition, the privacy leakage problem will occur. While there is an attacker in the network can intercept the data transmitted on the edges, we have derived the probability of network disclosure. For the attacker with limited energy, we have further found the optimal attacking strategy in ring network and small-world model. Combined with an optimization algorithm, we give some guiding suggestion to find the optimal attack strategy. From the simulation results, we get the conclusion that the variance of the adding noise's distribution will influence the convergence time of the privacy preserving consensus protocol.

Appendix A
Proof of the First Zonklar Equation

Appendix one text goes here.

Appendix B

Appendix two text goes here.

Acknowledgments

The authors would like to thank...

References

[1] Reza Olfati-Saber and R. M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on Automatic Control*, 49(9):1520–1533, Sept 2004.

[2] Reza Olfati-Saber, Alex Fax, and Richard M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[3] Ling Shi and Lihua Xie. Optimal sensor power scheduling for state estimation of gauss-markov systems over a packet-dropping network. *IEEE Transactions on Signal Processing*, 60(5):2701–2705, 2012.

[4] Wen Yang, Xiaofan Wang, and Hongbo Shi. Fast consensus seeking in multi-agent systems with time delay. *Systems & Control Letters*, 62(3):269–276, 2013.

[5] Wen Yang, Zidong Wang, Zongyu Zuo, Chao Yang, and Hongbo Shi. Nodes selection strategy in cooperative tracking problem. *Automatica*, 74:118–125, 2016.

[6] Yilin Mo and Richard M. Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, page in press, 2016.

[7] Vladimir Kolesnikov. Advances and impact of secure function evaluation. *Bell Labs Technical Journal*, 14(3):187–192, 2009.

[8] Cynthia Dwork. *Differential Privacy*. Springer Berlin Heidelberg, 2006.

[9] Zhenqi Huang, Sayan Mitra, and Geir Dullerud. Differentially private iterative synchronous consensus. pages 81–90, 2012.
[10] Xiaoming Duan, Jianping He, Peng Cheng, and Yilin Mo. Privacy preserving maximum consensus. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 4517–4522, Dec 2015.
[11] Nicolaos. E Manitara and Chistoforos. N Hadjicostis. Privacy-preserving asymptotic average consensus. In *European Control Conference (ECC), 2013 European*, pages 760–765, July 2013.
[12] Mahdi Kefayati, Mohammad S. Talebi, Babak H. Khalaj, and Hamid R. Rabiee. Secure consensus averaging in sensor networks using random offsets. In *IEEE International Conference on Telecommunications and Malaysia International Conference on Communications, 2007. ICT-MICC*, pages 556–560, 2007.
[13] Heng Zhang, Peng Cheng, Junfeng Wu, Ling Shi, and Jiming Chen. Online deception attack against remote state estimation. *IFAC Proceedings*, 47(3):128–133, 2014.
[14] Yifei Qi, Peng Cheng, Ling Shi, and Jiming Chen. Event-based attack against remote state estimation. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6844–6849, Dec 2015.
[15] Li Lei, Wen Yang, Chao Yang, and Hongbo Shi. False data injection attack on consensus-based distributed estimation. *International Journal of Robust & Nonlinear Control*, page in press, 2016.
[16] D. J. Watts and S. H. Strogatz. Collective dynamics of "small-world" nettworks. *Nature*, 393(6684):440–2, 1998.
[17] H. Hong, M. Y. Choi, and B. J. Kim. Synchronization on small-world networks. *Physical Review E Statistical Nonlinear & Soft Matter Physics*, 65(2 Pt 2):95–129, 2002.
[18] Xiao Fan Wang and Guanrong Chen. Complex networks: small-world, scale-free and beyond. *IEEE Circuits & Systems Magazine*, 3(1):6–20, 2003.
[19] Xiao Ffan Wang and Guanrong Chen. Synchronization in small-world dynamical networks. *International Journal of Bifurcation & Chaos*, 12(1):187–192, 2011.
[20] Dieter Kraft. Algorithm 733: Tomp-fortran modules for optimal control calculations. *Acm Transactions on Mathematical Software*, 20(3):262–281, 1994.

PLACE PHOTO HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.