2018 18th International Conference on Control, Automation and Systems (ICCAS 2018)
Oct. 17–20, 2018; YongPyong Resort, PyeongChang, GangWon, Korea

# Partial Agreement Task Assignment Algorithm for Secure Plan Consensus in Multi-Agent System

Ho-Yeon Kim[1] and Han-Lim Choi[1*]

[1]Department of Aerospace Engineering, KAIST,
Daejeon, 34141, Korea (hykim@lics.kaist.ac.kr,hanlimc@kaist.ac.kr)
[*] Corresponding author

**Abstract:** This paper presents a secure market based task assignment algorithm for decentralized multi-agent system. In a decentralized algorithm, individual agents calculate their solution with imperfect information of environments. To achieve agreement in the system, the plan consensus solves local task assignment problems and reach a conflict-free solution through the plan exchange. The Consensus Based Bundle Algorithm(CBBA), which is well-known consensus based plan auction method, guarantees feasible and conflict-free solution. However, in the plan consensus protocol, each agent determines tasks only through bidding for each task like maximum bid consensus. It is vulnerable to malicious data to attack a consensus process or solution result. Therefore, providing security to the decentralized agent system is a major issue. The main contribution of this paper is an algorithm, termed partial-agreement consensus-based bundle algorithm, that proposed for two types of attacks, exaggerated score attack and changing score attack. Partial agreement of the agent with some of the mission plans resists attacks that are intended to degrade the overall plan. The security solution is a more robust task allocation algorithm with a partial agreement on the plans and task bidding data.

**Keywords:** Multi-agent systems, Plan consensus algorithms, Secure task allocation, Cyber-attack

## 1. INTRODUCTION

Recently, the decentralized multi-agent system have expanded interest in the research area of unmanned vehicle systems. A decentralized task assignment is main issues algorithm for multi-agent coordination system, when the agents can not communicate efficiently with the specific center. In that case, each agent's information on the environment is not always similar. To achieve a conflict-free and feasible solution, all agents should agree on either of the two sets of the data [5]: 1) agreement on situational awareness (S.A.) , 2) agreement on planning data. If each agent has the same S.A and the same objective function, then the plan result will be consistent and conflict-free [1]. S.A. agreement approach is called implicit coordination [1,2]. Agents exchange S.A. and come to terms before planning. During planning, all agents act like a centralized planner. They can predict the allocation results of others [3, 4]. A disadvantage is that each agent requires significant computational resources to calculate whole coordination plan. Therefore, implicit coordination can alleviate the problem of poor network at the expense of higher computational cost. Auction algorithm is combinatorial optimization based consensus algorithm of plan to solve assignment problems [6, 7] is market based algorithm called a plan consensus algorithm [8, 10–12]. All agents make their own assignment plan, and exchange their bids for the task. Before planning, communication for S.A. is not required. Each agent makes and agree on a schedule, and the schedule can be changed by auction during planning. In an imperfect network environment, auction algorithm [13, 14] can deduce suboptimal solutions efficiently, but iterative communication and auctioning are needed. The Consensus Based Bundle Algorithm(CBBA) [8] is an auction algorithm, but all agents are auctioneers and make plans with other agents' bid data like maximum bid consensus. CBBA is constructed in two phases: bundle construction and plan consensus. Bundle construction phase is similar to an auction. Agents make bids for each task with a bid space of agents which is connected before. The new plan is always higher bids than plan in previous communication. In the second phase, plan consensus uses an action table to achieve conflict-free solution. CBBA guarantees assignment consistency and feasible solution set in dynamic environments. Maximum bid consensus can deduce suboptimal solutions fast, there is strong motivation to address these safety issues detecting attacks for multi-agent coordination can be based on from vulnerability analysis of unmanned vehicle [20]. The attacks can be categorized based on the targets. 1) Hardware attack accesses to vehicle component directly, 2) Communication attack is through network channels, 3) Sensor spoofing passes false sensing data. In systematic view, the vehicle requires two types of security : Application Logic Security and Control System Security. The resilient consensus problem for multi-agent systems requires designing protocol problem to resist the malicious attack on consensus [15, 16, 18, 19]. The main purpose of resilient consensus protocol is to maintain consensus

This research was supported by Unmanned Vehicles Advanced Core Technology Research and Development Program through the National Research Foundation of Korea(NRF), Unmanned Vehicle Advanced Research Center(UVARC) funded by the Ministry of Science and ICT the Republic of Korea (2016M1B3A1A02937510)

tracking performance. The attack on the communications among the agents, targets the edges of graphs [15, 16] or the nodes in network [17, 19]. Byzantine case attack and non-colluding fault were considered in a multi-agent system including motion coordination [15]. Additionally, crash failure was considered in discrete time with time delays [16, 18] proposed novel delay robust secure consensus to detect false information resulting in an undesirable state. The authors in [17, 19] studied on network fault is divided into connectivity-maintained and non-connectivity-maintained.

## 2. PLAN CONSENSUS PROBLEM

The multi-agent, multi-task assignment problem where $N_t$ tasks are assigned to a group of $N_u$ agents. The goal of the assignment is where given agents, tasks, and the performance metric, to maximize performance subject to the constraints. Each agent has a constraint of maximum capability on task, $L_t$, and problem is completed when $N_t$ tasks are assigned or $N_u$ agents are fully assigned, or capability of agent is full, $N_u L_t$. Task assignment problem can be stated as follows:

Where $x_{ij} \in \{0, 1\}$ indicates if agent i is assigned to task j, and $\mathbf{x_i}$ is $N_t \times 1$ vector of $x_{ij}$. The performance metric $c_{ij}$ is a non-negative function of the agent which performs tasks $j$, and the assignment results $\mathbf{p_i}$, is the agent's path, sorted in order of execution. Equation (1) is the objective function in order to maximize performance that is calculated by $\mathbf{x_i}$ and $\mathbf{p_i}$

$$\max \sum_{i=1}^{N_u} \left( \sum_{j=1}^{N_t} c_{ij}(\mathbf{x_i}, \mathbf{p_i}) \mathbf{x_{ij}} \right) \tag{1}$$

$$\sum_{j=1}^{N_t} x_{ij} \leq L_t \quad \forall i \in I$$

$$\sum_{i=1}^{N_u} x_{ij} \leq 1 \quad \forall j \in J$$

$$\sum_{j=1}^{N_t} \sum_{i=1}^{N_u} x_{ij} \ = \ N_{limit} \ = \ \min\{N_t, \ N_u L_t\}$$

$$x_{ij} \in \{0, 1\} \qquad (i, j) \in I \times J$$

In algorithmic views, there are two consensus based approach to solve the task assignment problem. Consensus means that all of the agents agree on the information [9]. 1) a priori information about the environment to define the task assignment problem is exchanged before the mission, 2) situational awareness information $\mathcal{X}_\mathrm{i}$ used for solving the task assignment problem is shared before planning, 3) assignment information $\mathcal{B}_\mathrm{i}$ that is the solution of the task assignment problem. To ensure that the solution is conflict-free, agents can exchange understanding of the tasks beforehand or exchange the results of the problem. The first approach called implicit coordination, where situational awareness is exchanged before planning. The second approach is plan consensus where the result of the assignment is agreed upon by all agents.

The goal of the plan consensus approach is to make each agent's plan conflict free through iteration of consensus. Each agent makes its own plan using S.A. information $\mathcal{I}_\mathrm{i}$ and bid space $\mathcal{A}_\mathrm{i}$. Bid space is the consensus result data in consensus function. Other agents' information is considered in allocation phase.

CBBA is a plan consensus algorithm through auction. Bundle construction phase of CBBA is allocation part, and conflict resolution phase is consensus part. Bundle construction phase uses an optimization technique like sequential auction. In CBBA, $\mathcal{A}_\mathrm{i}$ is defined from the winner, winner's bids and time stamp. Action table is used for conflict resolution. Each agent exchange their bid space with connected neighbor and perform the actions to consensus : update, reset, or leave by action rule.

## 3. ATTACK MODEL IN MULTI-AGENT PLAN CONSENSUS

Attack on multi-agent system affects the hardware, communication and sensor data. The purpose of malicious attack is to reduce the performance of task assignment. The performance of the system is decided in accordance with throughput, response time, availability. There are two criteria of algorithm performance : optimality of the solution and convergence time of the algorithm. This section provides the attack model reducing two algorithm performance. The attack methods are categorized by results of attack. The Selfish Agent with exaggerated score can cause the problem reduce optimization performance. If the agents argue to do all tasks with the false high score, the working performance decrease. The Mischief Agent with changing score disturb convergence of consensus. In the plan consensus, other agents' plans are only considered. Similarly, because one agent continues to change his score, the entire system can not reach consensus.

### 3.1 Selfish Agent with exaggerated score

Selfish Agent is defined as the agent in the system that make bid greater than true. This behavior makes a malicious agent have higher values than all other agents and makes it impossible except for selfish agent to execute tasks. Therfore, all of the loyal agents do not work anything in a multi-agent system under this type of attack. The Fig.1 is multi-agent system scenario with several tasks. The blue circles indicate loyal agents and the yellow one indicate the attacked agent. When task assignment algorithm begins, the malicious agent presents a fictitious highest value to all tasks. Because all other participants calculate normally, they can not create a plan to win malicious bids. By selfish agent's bundle, all of the loyal agents are not assigned to any task as shown in the Fig.2.

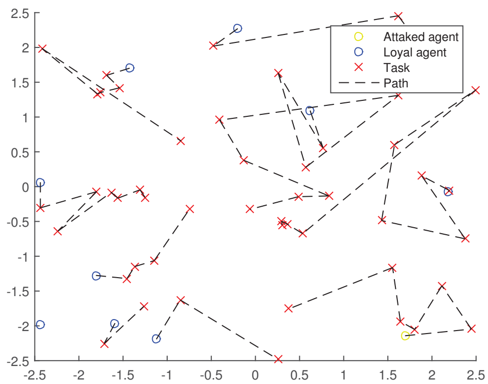


Fig. 1 Assignment results without attack

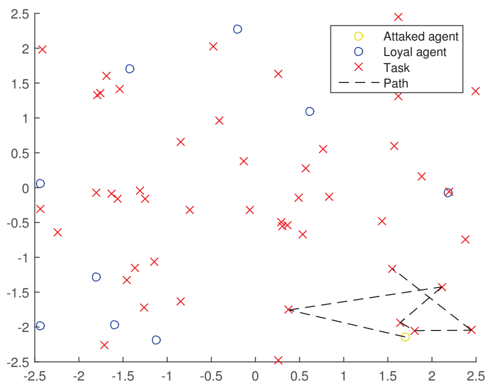


Fig. 2 Assignment results with selfish agent

### 3.2 Mischief Agent with changing score

In case of mischief agent, the attacked agent changes its own bid to prevent convergence. In the plan consensus system, agents compare their plans with those sent by other agents, resolve conflicts, and is assigned tasks. At this point, if the attacked agent continues to change the plan, the plan of the other agent is continuously changed according to the malicious agent's plan, so the entire system does not converge.

## 4. PARTIAL AGREEMENT CONSENSUS BASED BUNDLE ALGORITHM

This section presents the partial agreement approach to plan consensus problem.The Partial Agreement Consensus Based Bundle Algorithm(PA-CBBA) is based on the CBBA algorithm. The goal of this approach is to secure plan consensus using previous shared plan data against two attack model mentioned above. The intuitive solution is to converge partial plan before all plans are accepted. Some of the assignments are not changed after several iterations in planning time. In bid bundle changed for consensus, CBBA algorithm decides the three actions using a maximum score of each task. However, all agents involved in making a bundle partially agree on information in the bundle.

**Algorithm 1** Partial Agreement CBBA

**procedure** PA-CBBA($\mathbf{b}_i^\mathbf{0}, \mathbf{p}_i^\mathbf{0}, \mathcal{A}_i$)
  $k_{ij} \leftarrow 0\,, \forall\, task\ j$
  $t_{ik} \leftarrow 0\,, \forall\, agent\ k$
  **while** $k_{ij} < 2 \cdot D,\ \forall j$ **do**
    $\mathbf{b}_i, \mathbf{p}_i, \mathcal{A}_i \leftarrow CBBA\text{-}BB(\mathbf{b}_i^\mathbf{0}, \mathbf{p}_i^\mathbf{0}, \mathcal{A}_i)$
    Send $\mathcal{A}_i$ to agent k with $\mathrm{g_{ik}} = 1$
    Receive $\mathcal{A}_k$ from agent k with $\mathrm{g_{ki}} = 1$
      $t_{ik} \leftarrow t_{now}$
    $\mathbf{b}_i^0, \mathbf{p}_i^0, \mathcal{A}_i \leftarrow PA\text{-}CR(\mathbf{b}_i^0, \mathbf{p}_i^0, \mathcal{A}_i, \mathcal{A}_k, t_i)$
  **end while**
  **return** $(\mathbf{b}_i, \mathbf{p}_i, \mathcal{A}_i)$
**end procedure**

In this approach, the agents partially agree on two information in the bundle : agent to execute each task, limit score of each task. In other words, system convergence check of CBBA is utilized for each task and each task comparison of CBBA is used for partial agreed limit score of the agents. If all of the agent partially agree with agent to work some task, agents can detect malicious attacks for those tasks. In addition, if some agent makes a new claim conflicting partially agreed limitation, it can be ignored in agent network.

**Algorithm 2** PA-CR: Conflict Resolution Phase

**procedure** PA-CR($\mathbf{b_i^0}, \mathbf{p_i^0}, \mathcal{A}_\mathrm{i}, \mathcal{A}_\mathrm{k}, \mathrm{t_i}$)
  **for** $\forall s_{z_{kj}j} \in \mathcal{A}_\mathrm{k}$ **do**
    $Find\ \ action(\mathcal{A}_\mathrm{i}, s_{z_{kj}j}, \mathbf{b_i^0}, t_{ik})$
    **if** $UPDATE$ **then**
      $\mathcal{A}_i \leftarrow \mathcal{A}_i \backslash s_{z_{ij}j}$
      $\mathcal{A}_i \leftarrow \mathcal{A}_i \cup s_{z_{kj}j}$
      $\mathrm{k}_{ij} \leftarrow\ 0$
    **else if** $RESET$ **then**
      $\mathcal{A}_i \leftarrow \mathcal{A}_i \backslash s_{z_{ij}j}$
      $k_{ij} \leftarrow\ 0$
    **else if** $LEAVE$ **then**
      **if** $z_{ij} = i$ **then**
        $k_{ij} \leftarrow\ k_{ij} + 1$
      **end if**
    **else if** $FAULT$ **then**
      $g_{ik} \leftarrow 0$
    **end if**
    **if** $k_{ij} > 2 \cdot D$ **then**
      $\mathbf{b_i^0} \leftarrow \mathbf{b_i^0} \oplus_{\mathbf{end}} \mathbf{j}$
      $\mathbf{p_i^0} \leftarrow \mathbf{p_i^0} \oplus_{\mathbf{n_{ij}}} \mathbf{j}$
    **end if**
  **end for**
  **return** $(\mathbf{b}_i^\mathbf{0}, \mathbf{p}_i^\mathbf{0}, \mathcal{A}_i)$
**end procedure**

The proposed algorithm called PA-CBBA is the decentralized algorithm that provides the partial results of task assignments in planning time. The algorithmic structure of the PA-CBBA is consist of iterative two functions : Build Bundle function of CBBA(CBBA-BB) and Partial Agreement Conflict Resolution(PA-CR). The main function of PA-CBBA is in Algorithm.1. Decentralized agents are initialized with an initial agreed bundle $\mathbf{b}_i^\mathbf{0}$, an initial agreed path $\mathbf{p_i^0}$, a bid space $\mathcal{A}_\rangle$. It is assumed that the priori information is already shared and remain constant.

The internal variables are initialized. $k_{ij}$ is the convergence counter and $t_{ik}$ is the last connection time between agent i and k. In every iteration of the function, convergence counters of each task are checked. If one of the counters for the task is over two times the network diameter $2 \cdot D$, that task is on the partially agreement. After all tasks are agreed on, the algorithm has converged.

The CBBA-BB subfunction determines assignment results with the partial agreement results. It is similar with CBBA-BB in [8], but the initial bundle is not empty array. This function determines assignment results from partial agreement data and current bid space $\mathcal{A}_i$. Each agent updates a task bundle until the bundle is full or all tasks are assigned.

The conflict resolution phase has three actions in CBBA [8] modified and additional action $fault$ which deems that an incorrect plan has been detected and the related agent will be ignored. The possible actions are $update$, $reset$, $leave$, $fault$ with information receiver $i$, sender $k$, target task of information $j$, winner of task $z_{ij}$ .

1) $UPDATE$ : $s_{z_{ij}j}$ is replaced with $s_{z_{kj}j}$ (agent i's opinion is changed by agent j's opinion)

2) $RESET$ : $s_{z_{ij}j}$ is deleted (agent i's opinion is already old)

3) $LEAVE$ : $s_{z_{ij}j}$ is correct (partially agreed on agent i's opinion)

4) $FAULT$ : $s_{z_{kj}j}$ is not suitable (agent k's opinion is different from partially agreement)

The convergence counter $k_{ij}$ is different from another CBBA-based algorithm. An action $leave$ means that an opinion is correct in that iteration phase, and the convergence counter is increased by one. Actions which lead to changes in assignment information reset the convergence counter. The partial agreement bundle $b_i^0$ and path $p_i^0$ update if that convergence counter is over two times of network diameter $2 \cdot D$.

## 5. SIMULATION AND RESLUTS

The simulation environment used for verifying security solution for two malicious attacks. The Monte-Carlo experiments were run with plan attacked agent is on the various number of agents at varying times. The scenario was run with 200 tasks and 10 agents. Fig. 3,4 are Monte-Carlo results in different time under 4 agents' plan attacked condition.

In 10th iteration attack case, the partial agreement bundle is empty, so that no significant reduction of damage by selfish agent. PA-CBBA is approaches of sequential convergence. Therefore, it is not different with an original CBBA case before first partial agreement. In 50th iteration attack case, the attack is injected after algorithm convergence. There is no effect of the attack, because PA-CBBA is already converge. The main idea of PA-CBBA is to confirm and stabilize a part of the plan during planning. The Fig. 3 shows PA-CBBA can reduce damage to performance caused by attacks during partial agreement time (between first partial agreement and convergence). The Fig. 4 shows that PA-CBBA will converge faster than CBBA under selfish agent case in partial agreement time.

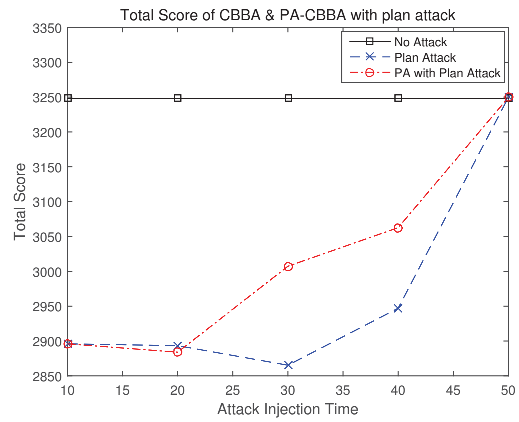


Fig. 3 Total score as a function of time under 4 plan attacked agents condition

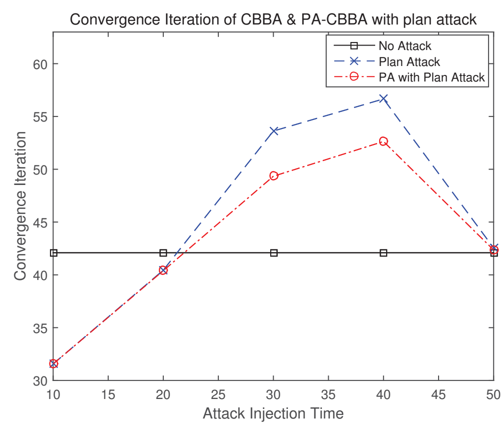


Fig. 4 Convergence iterations as a function of time under 4 plan attacked agents condition

Fig. 5,6 are Monte-Carlo results on a varying number of agents under attack at 30th iteration. The performance robustness and convergence robustness are shown in that figure.

The Fig. 5 shows PA-CBBA can reduce damage to performance caused by the attacks and the Fig. 6 shows that PA-CBBA will converge faster than CBBA under selfish agent case in partial agreement time.

This paper provides the attack models for task assignment algorithm and partial agreement approach against plan attack agent. Specifically, we focus on the agent

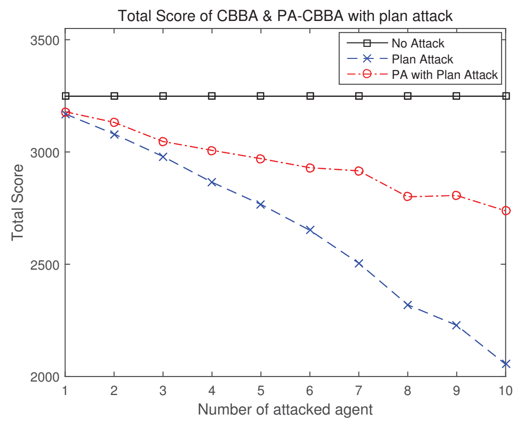


Fig. 5 Total score as a function of the number of plan attacked agents under 30th iteration

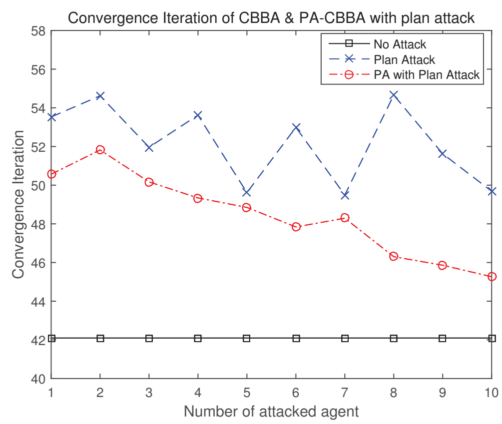


Fig. 6 Convergence iterations as a function of the number of plan attacked agents under 30th iteration

which interferes with optimality and convergence performance. The proposed algorithm resists plan attack in consensus by confirming partially agreed plan data. Because there is information available to determine if an agent has been attacked, security performance improves as the agreed-upon portion of plan increases. This work can be further proceeded to investigate more partial agreement points with a more detailed plan and diffrent levels of information.

## REFERENCES


[1] T. M. McLain and R. W. Beard. *Coordination variables, coordination functions, and cooperative timing missions*, AIAA Journal on Guidance, Control, and Dynamics, 28(1):150161, 2005.

[2] T. Shima, S. J. Rasmussen, and P. Chandler. *UAV team decision and control using efficient collaborative estimation* In American Control Conference(ACC), volume 6, pages 41074112, 8-10 June 2005.

[3] Wei Ren, R. W. Beard, and Derek B. Kingston. *Multi-agent Kalman consensus with relative uncertainty*, Proceedings of the 2005, American Control Conference, IEEE, 2005.

[4] Wei Ren, R. W. Beard, and E.M. Atkins *Information consensus in multivehicle cooperative control*, Control systems, IEEE, 27(2):71-82, april 2007.

[5] Mehdi Alighanbari, and Jonathan P. How. *Decentralized task assignment for unmanned aerial vehicles*, Proceedings of the 44th IEEE Conference on Decision and Control. IEEE, 2005.

[6] Benny Lehmann, Daniel Lehmann, and Noam Nisan. *Combinatorial auctions with decreasing marginal utilities*, Games and Economic Behavior, 2005.

[7] Tuomas Sandholm. *Algorithm for optimal winner determination in combinatorial auctions*, Artificial Intelligence, 135(1-2):154, 2002

[8] Han-Lim Choi, Luc Brunet, and Jonathan P. How, *Consensus-Based Decentralized Auctions for Robust Task Allocation*, IEEE TRANSACTIONS ON ROBOTICS, Volume 25, Number 4, August 2009.

[9] Luke B. Johnson, *Decentralized task allocation in communication contested environments* Ph.D. dissertation, Dept. Aeronautics and Astronautics, Massachusetts Inst. of Tech., Cambridge, MA, 2016.

[10] R. Olfati-Saber, J. Alex Fax, and Richard M. Murray. *Consensus and cooperation in networked multi-agent systems*, Proceedings of the IEEE 95.1 (2007): 215-233

[11] R. Olfati-Saber and R. M. Murray, *Consensus problems in networks of agents with switching topology and time-delays*, IEEE Trans. Autom. Control, vol. 49, no. 9, pp. 15201533, Sep. 2004.

[12] Vincent Blondel, et al. *Convergence in multi-agent coordination, consensus, and flocking*, IEEE Conference on Decision and Control. Vol. 44. No. 3. IEEE; 1998, 2005.

[13] B. P. Gerkey and M. J. Mataric. *Sold!: Auction methods for multirobot coordination*, IEEE transactions on robotics and automation 18.5 (2002): 758-768.

[14] B. P. Gerkey and M. J. Mataric, *A formal analysis and taxonomy of task allocation in multi-robot systems*, Int. J. Robot. Res., vol. 23, no. 9, pp. 939954, 2004.

[15] F. Pasqualetti, R. Carli, and F. Bullo *On the security of linear consensus networks*, Decision and control, 2009 held jointly with the 2009 28th chinese control conference. cdc/ccc 2009. proceedings of the 48th ieee conference on. IEEE, 2009.

[16] Wu Yiming, et al. *Consensus of discrete-time multi-agent systems with adversaries and time delays*, International Journal of General Systems 43.3-4 (2014): 402-411.

[17] Wu Yiming, He Xiongxiong, and Ou Xianhua. *Distribute consensus for multi-agent systems with at-*

*tacks and delays*, Control Conference (CCC), 2015 34th Chinese. IEEE, 2015.
[18] Zhi Feng, Guoqiang Hu, and Guanghui Wen. *Distributed consensus tracking for multi-agent systems under two types of attacks*, International Journal of Robust and Nonlinear Control 26.5 (2016): 896-918.
[19] Zhi Feng, Guanghui Wen, and Guoqiang Hu. *Distributed Secure Coordinated Control for Multiagent Systems Under Strategic Attacks*, IEEE Transactions on Cybernetics, 2016.
[20] Alan Kim, et al. *Cyber attack vulnerabilities analysis for unmanned aerial vehicles*, Infotech@ Aerospace (2012).