engineering proceedings

MDPI

Proceeding Paper

A Secure Multi-Agent-Based Decision Model Using a Consensus Mechanism for Intelligent Manufacturing Tasks †

Mostefa Kara [1,2,*], Abdelkader Laouid [1], Mohammad Hammoudeh [3], Konstantinos Karampidis [4], Giorgos Papadourakis [4] and Ahcène Bounceur [3]

[1] LIAP Laboratory, University of El Oued, El Oued 39000, Algeria; abdelkader-laouid@univ-eloued.dz
[2] National Higher School of Mathematics, Scientific and Technology Hub of Sidi Abdellah, Algiers 16093, Algeria
[3] Information & Computer Science Department, King Fahd University of Petroleum and Minerals, Academic Belt Road, Dhahran 31261, Saudi Arabia; mohammad.hammoudeh@kfupm.edu.sa (M.H.); ahcene.bounceur@kfupm.edu.sa (A.B.)
[4] Department of Electrical and Computer Engineering, Hellenic Mediterranean University, 71410 Heraklion, Greece; karampidis@hmu.gr (K.K.); papadour@hmu.gr (G.P.)
* Correspondence: karamostefa@univ-eloued.dz or mostefa.kara@nhsm.edu.dz
† Presented at the 4th International Electronic Conference on Applied Sciences, 27 October–10 November 2023; Available online: https://asec2023.sciforum.net/.

Abstract: Multi-agent systems (MASs) have gained a lot of interest recently, due to their ability to solve problems that are difficult or even impossible for an individual agent. However, an important procedure that needs attention in designing multi-agent systems, and consequently applications that utilize MASs, is achieving a fair agreement between the involved agents. Researchers try to prevent agreement manipulation by utilizing decentralized control and strategic voting. Moreover, emphasis is given to local decision making and perception of events occurring locally. This manuscript presents a novel secure decision-support algorithm in a multi-agent system that aims to ensure the system's robustness and credibility. The proposed consensus-based model can be applied to production planning and control, supply chain management, and product design and development. The algorithm considers an open system; i.e., the number of agents present can be variable in each procedure. While a group of agents can make different decisions during a task, the algorithm chooses one of these decisions in a way that is logical, safe, efficient, fast, and is not influenced by factors that might affect production.

Keywords: distributed systems; interactions; confidentiality; smart environments; algorithms

Citation: Kara, M.; Laouid, A.; Hammoudeh, M.; Karampidis, K.; Papadourakis, G.; Bounceur, A. A Secure Multi-Agent-Based Decision Model Using a Consensus Mechanism for Intelligent Manufacturing Tasks. *Eng. Proc.* **2023**, *56*, 234. https://doi.org/10.3390/ASEC2023-15929

Academic Editor: Nunzio Cennamo

Published: 8 November 2023

1. Introduction

An agent is a mechanical, biological, or software system that interacts with its environment. A multi-agent system (MAS) comprises several agents that interact with each other in a common environment [1]. Interaction is the dynamic linking of two or more agents through a set of mutual actions. Agents interact to manage, communicate, coordinate, cooperate, negotiate, etc. MASs are currently a very active and wide branch of distributed artificial intelligence (DAI) [2,3]. Agents in such systems are endowed with a greater degree of freedom than typical entities of a security protocol. They may have clearly defined goals, abilities, and knowledge about the world; they can also form coalitions working towards a common goal. The advantages of MASs become particularly visible in the analysis of scenarios involving interaction between human and technical agents, such as in healthcare and smart manufacturing.

Smart manufacturing systems based on modern technologies are beneficial. However, the challenge lies in integrating these systems with the evolving needs of the customer. Recent market changes have revealed various problems, including allocating limited resources

and responding to urgent emergencies. In addition, smart manufacturing at all levels is an open, vast, and complex environment that assumes shared and distributed decision making with security in mind [4], which requires the communication of many complex and diverse forms of data between different sets of industrial environments. To provide a unified and effective solution in this area, this paper proposes a novel model based on a multi-agent system by providing certain production services in a timely and appropriate manner, and ensuring fairness in decision making [5] regarding various procedures such as confidentiality [6].

2. Related Work

Data acquisition, task scheduling, resource allocation, and decisions in manufacturing systems are critical issues that require complex mechanisms, especially when demands are high and random. To address all these challenges, several works have been proposed. In [7], the authors proposed an architecture that consists of autonomous agents capable of communicating with each other to make decisions using their knowledge. In particular, the architecture was proposed for negotiation processes and scheduling optimization algorithms in intelligent manufacturing systems. Other multi-agent approaches for modeling supply chain dynamics with agent decision making have been proposed in [8]. To address a variety of scheduling problems within a decentralized and hierarchical architecture, an agent-based model has been proposed in [9]; these schedules include constraints such as the blocking constraint. A multi-agent model to share distributed manufacturing resources has been proposed in [10].

In [11], the authors formalized the selected features of the Selene [12] voting protocol by means of formulas and multi-agent models. Their models defined the space of strategies for the members, the electoral authority, and the potential coercive. The work in [13] investigated the aspect of robustness to respect the way in which votes are cast. The outcome is declared when the conceptual requirements can be encapsulated in a formal specification of the voting protocol. The authors of [14] proposed a system called smart voting by extending classical liquid democracy in two ways: allowing ranked delegations to avoid cycles and allowing generalizing delegations to be more expressive.

Many of the proposed algorithms suffer from high energy consumption due to the complexity of the calculations they perform. Other algorithms are either very time-consuming due to the large number of messages that are exchanged between agents or depend on the leader agent concept, in which the principle of equal opportunities is absent. Moreover, this type of decision making is not safe because the leader can manipulate the final decision.

The proposed model addresses all these gaps by (a) relying on linear arithmetic operations, (b) minimizing the messages exchanged between agents, and (c) providing all agents equal opportunity to declare the final decision by random choice, providing in this way an intelligent voting method safe from tampering.

3. Model Description

To further improve the manufacturing system and to ensure that various decisions are made in a safe, effective, and secure manner, this research proposes a consensus mechanism in a distributed and heterogeneous environment where the solution is the result of the interaction of several entities (agents), and this interaction is expressed by cooperation and competition (Algorithm 1). The conception has considered the lack of trust between agents.

The consensus process is divided into three steps. In the first step, the agents must be aware of the total number of participants N at this moment; therefore, they broadcast a $query - nbr - msg$-type message and wait for $MaxWaitTime$ in order to receive the answers. After each agent announces its result N, the number N that receives a majority of votes is set as the number of participants.

In the second step, "creation of encrypted list", each agent A_i generates a random and temporary number x_i, and also a random and temporary secret key y_i to encrypt x_i using a symmetric scheme based on key exchange protocol according to the following equation:

$$v_i = x_i \times y_i \mod p \tag{1}$$

where p is a known prime and $0 < x_i < p$. The generated x_i is encrypted so that some faulty agents cannot control the result of the selection algorithm. For more solidity, that agent can change the secret key y_i or manipulate the selection value; then, the agent diffuses v_i and v_i' where:

$$v_i' = y_i^{y_i} \mod p \tag{2}$$

Each agent will read all v_i and v_i'; when the encrypted list is created and provided that all agents validate it, the third step (selection) begins. In the third step, each agent diffuses its secret key y_i, and every other agent can calculate all the x_i values by decrypting the v_i values ($x_i = v_i \times y_i^{-1} \mod p$). Then, the agents calculate v according to the following equation:

$$v = (\sum_{i=1}^{N} x_i) \mod p \tag{3}$$

where N is the number of agents. The winner is the one that owns the value x_i that is closest to the value v.

Algorithm 1 Consensus algorithm

Require: P: prime number, MWT: MaxWaitTime
Ensure: winner

function CONS
  broadcast query-nbr-msg
  wait MWT, reading response-nbr-msg
  broadcast N (number of participants)
  agree N
  generate x_i
  generate y_i
  $v_{1i} \leftarrow x_i \times y_i \mod p$
  $v_{2i} \leftarrow y_i^{y_i} \mod p$
  broadcast v_{1i}, v_{2i}
  while number of received $v_{1,2j} <$ N and wait $<$ MWT **do**
    receive $v_{1,2j}$
  end while
  if Count($v_{1,2j}$) = N, broadcast y_i
  while number of received $y_j <$ N and wait $<$ MWT **do**
    receive y_j
  end while
  $x_j \leftarrow$ decrypt v_{1j} using y_j
  if Count(y_j) = N **then**
    $v \leftarrow \sum_{i=1}^{N} x_i \mod p$
    select the winner using v
    return winner
  else
    failure
  end if
end function

4. Model Analysis

Since agents must be able to participate, interact, and make decisions, each one of them has a random value generation system. Each agent has the value of identifying a personal activity. This value represents whether or not they wish to participate in a consensus or an election.

Assuming the agent can leave or fail within a period of time, the profile will only be queried at the beginning of that time slot. Thus, the maximum waiting time should be set to prevent the rest of the participants from not waiting for too long. In the proposed protocol, system continuity is ensured while avoiding the futile study of whether an agent was active or not in the previous consensus. The environment agent provides information about the maximum wait time $MaxWaitTime$ and the public key p.

The proposed model uses a symmetric encryption scheme that is lighter than asymmetric encryption [15,16]. Moreover, it is less costly in terms of energy consumption compared to many consensus algorithms [17], which is an important factor in media and applications that use low-efficiency devices. When agents are waiting for each other to exchange y_i values to decrypt x_i, and one of them fails in the previous step, the others have to wait for a limited time and then start over. This procedure is obligatory to avoid endless waiting. The waiting time depends on the environment and the application.

Domination is almost impossible because the generated numbers x_i and y_i are secret; in addition, the choice of a winner is pseudo-random. It is not possible to manipulate x_i and y_i or to predict the consensus outcome. Regarding scalability, the first step is to calculate the number of participants N; then, the agents will make a consensus based on N. In the worst-case scenario, where the number of agents is significant, the maximum waiting time to calculate N can be enlarged; thus, a scalable system is going to be built.

Before the manager agent (MA) is selected, each agent will act according to the scheme shown in Figure 1.

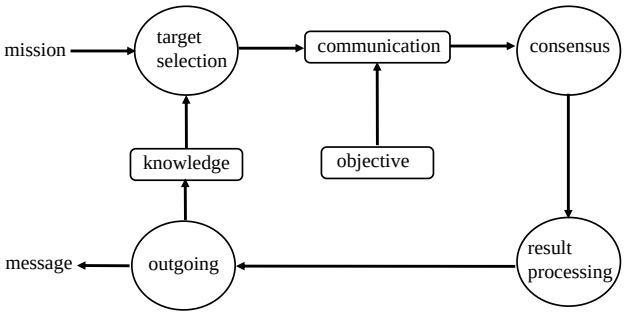

Figure 1. The acting of agent-inside, where consensus is clearly shown where exactly the proposed algorithm is executed.

Agent characteristics are described below:

- Attributes: Set of attributes that characterize its simulated state at a given instant in time. These attributes include principal information about an agent, e.g., agent identifier.
- Knowledge: This consists of knowing details about the environment and other agents. This includes information related to the agent identifier and the input needed to execute the consensus algorithm. In a more comprehensive simulation, it might include the past performance of different agents or suppliers where these values can also be dynamically updated.
- Interactions: Constraints that define the agent's relationship with other agents in a consensus process, e.g., the set of agents with which it can interact.

5. Architecture and Implementation

In this section, initially, the types of agents and their behaviors will be described in order to choose the appropriate type for the proposed model, and afterward, the algorithm's implementation will be discussed.

5.1. Architectures and Communication

MAS architectures differ according to the range of application domains, i.e., the number of factors, system design, and the number of variables that determine the decision-making behavior of agents. They are functional, hierarchical, and blackboard structures. The functional architecture manages each task of a total process as a single agent. Agents connect and communicate with each other according to predefined ways. A blackboard structure is a distributed system of decision making, with agents involved in executing tasks and sending work to a central board. This is to avoid bottlenecks by offering tasks to all agents. The hierarchical structure combines features of the first two, consisting of heterogeneous agents cooperating in hierarchical relationships. This is the utilized architecture in the conducted experiments.

These experiments consider blackboard-based communication, where each agent can put information (data, knowledge) on the common space, and each agent can read from the blackboard at any moment; hence, there is no direct communication between agents. In the experiment, the space is a common matrix. The matrix contains five columns, and the number of rows varies depending on the number of agents participating in the consensus process. The first column is the agent ID, the second column stores the multiplication value between the two randomly generated numbers, the third column stores the value of the second number which is masked by the exponential operation, the fourth column is used to share the original value of the key (the second number), and the last column displays the value of the first number upon which the selection will be made.

5.2. Implemented Algorithm

In the conducted experiments, each agent was simulated by a process, where multi-process programming was implemented. In several tests, different numbers of processes were created, each of them starting with the creation of a random number $ID \leftarrow random()$, which is considered as the ID of the agent. Afterward, each process follows the execution of the proposed algorithm (Algorithm 2).

Figure 2 represents the behavior of the proposed algorithm when varying the number of participant agents in the consensus. The prime number p equal to 1,000,003 has been chosen, and the values x and y are random in the range [100, 1000]. It is worth noticing that with the number of agents being less than 15, the increase in time was relatively small since this increase represented on average approximately 0.5 seconds. When the number of agents exceeded the number of 15, a faster increase was observed in execution time compared to the first two experiments, where this increase was 1 sec. This is due to the increased time resulting from the exchange of the messages containing the keys, as well as additional operations for encryption and decryption and calculation of the results to extract the leader.

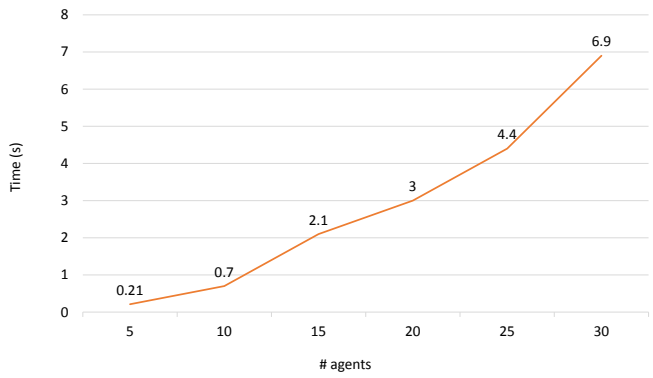

Figure 2. Relationship between the overall execution time of the proposed model and the number of agents in each execution process.

Algorithm 2 Progress algorithm

Require: ID, P, MWT, M(5,N) ▷ M: Matrix, N: number of agents
Ensure: Leader

function PROG
generate randomly x
generate randomly y
$v_1 \leftarrow x \times y$
$v_2 \leftarrow y^y \mod p$
insert(ID, v_1, v_2) in M ▷ $(M(1), M(2), M(3))$
$time \leftarrow 0$
while uncompleted $M(1)$ and $time < MWT$ **do**
wait
increase $time$
end while
insert(y) in $M(4)$
$time \leftarrow 0$
while uncompleted $M(4)$ and $time < MWT$ **do**
wait ▷ $M(4)$ denotes the fourth column that is the key y
increase $time$
end while
for $i \leftarrow 1$ to N **do**
$x' \leftarrow M(2)/M(4)$
insert x' in $M(5)$ ▷ x' is the decrypted value
end for
sort $M(5)$
$v \leftarrow \sum_{i=1}^{N} M(5) \mod (N+1)$
the leader is $M(1, v)$
end function

6. Conclusions and Future Research

Agent-based systems are increasingly being utilized to support decision-making in manufacturing systems, especially critical tasks, and are environmentally sensitive in terms of reliability and associated acceptance criteria that relate to risk and safety considerations. To address this challenge, this paper presented an effective algorithm for safe and rapid decision-making in a multi-agent paradigm, which can be applied in several practices related to intelligent manufacturing such as obtaining information, scheduling tasks, managing resources, etc. However, although the effectiveness of our approach proved to be high, the key future direction that we plan to work on is to propose a new variant of our algorithm taking into consideration the complex environments.

Author Contributions: Methodology, M.K. and A.L.; validation, M.H. and K.K.; formal analysis, G.P. and A.B.; writing—original draft M.K. and K.K.; writing—review and editing, A.L. and M.H.; visualization, G.P. and A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Dorri, A.; Kanhere, S.S.; Jurdak, R. Multi-agent systems: A survey. *IEEE Access* **2018**, *6*, 28573–28593. [CrossRef]
2. García, N.M. Multi-agent system for anomaly detection in Industry 4.0 using Machine Learning techniques. *ADCAIJ Adv. Distrib. Comput. Artif. Intell. J.* **2019**, *8*, 33–40.

3. Rădulescu, R.; Mannion, P.; Roijers, D.M.; Nowé, A. Multi-objective multi-agent decision making: A utility-based analysis and survey. *Auton. Agents Multi-Agent Syst.* **2020**, *34*, 10. [CrossRef]
4. Karampidis, K.; Panagiotakis, S.; Vasilakis, M.; Lamari, A.T.; Markakis, E.; Papadourakis, G. Digital Training for Cybersecurity in Industrial Fields via virtual labs and Capture-The-Flag challenges. In Proceedings of the 2023 32nd Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE), Eindhoven, The Netherlands, 14–16 June 2023; pp. 1–6.
5. Benhajji, N.; Roy, D.; Anciaux, D. Patient-centered multi agent system for health care. *IFAC-PapersOnLine* **2015**, *48*, 710–714. [CrossRef]
6. Kara, M.; Karampidis, K.; Sayah, Z.; Laouid, A.; Papadourakis, G.; Abid, M. N. A Password-Based Mutual Authentication Protocol via Zero-Knowledge Proof Solution. In Proceedings of the International Conference on Applied Cyber Security, Dubai, United Arab Emirates, 29 April 2023.
7. Guo, Q.; Zhang, M. A novel approach for multi-agent-based intelligent manufacturing system. *Inf. Sci.* **2009**, *179*, 3079–3090. [CrossRef]
8. Chen, Y.; Peng, Y. An Extended Bayesian Belief Network Model of Multi-agent Systems for Supply Chain Managements. Innovative Concepts for Agent-Based Systems. In Proceedings of the First International Workshop on Radical Agent Concepts, Lecture Notes in Computer Science, McLean, VA, USA, 16–18 January 2002; Volume 2564.
9. Gehlhoff, F.; Fay, A. On agent-based decentralized and integrated scheduling for small-scale manufacturing. *at-Automatisierungstechnik* **2020**, *68*, 15–31. [CrossRef]
10. Li, K.; Zhou, T.; Liu, B.H.; Li, H. A multi-agent system for sharing distributed manufacturing resources. *Expert Syst. Appl.* **2018**, *99*, 32–43. [CrossRef]
11. Jamroga, W.; Knapik, M.; Kurpiewski, D. Model checking the SELENE e-voting protocol in multi-agent logics. In Proceedings of the International Joint Conference on Electronic Voting, Bregenz, Austria, 2–5 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 100–116.
12. Ryan, P.Y.; Rønne, P.B.; Iovino, V. Selene: Voting with transparent verifiability and coercion-mitigation. In Proceedings of the International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 22–26 February 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 176–192.
13. Pitt, J.; Kamara, L.; Sergot, M.; Artikis, A. Voting in multi-agent systems. *Comput. J.* **2006**, *49*, 156–170. [CrossRef]
14. Colley, R. Multi-Agent Ranked Delegations in Voting. In Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems, Virtual, 3–7 May 2021; pp. 1802–1804.
15. Lalem, F.; Laouid, A.; Kara, M.; Al-Khalidi, M.; Eleyan, A. A Novel Digital Signature Scheme for Advanced Asymmetric Encryption Techniques. *App. Sci.* **2023**, *13*, 5172. [CrossRef]
16. Kara, M.; Karampidis, K.; Papadourakis, G.; Laouid, A.; AlShaikh, M. A Probabilistic Public-Key Encryption with Ensuring Data Integrity in Cloud Computing. In Proceedings of the 2023 International Conference on Control, Artificial Intelligence, Robotics & Optimization, Crete, Greece, 11–13 April 2023; pp. 59–66.
17. Medileh, S.; Laouid, A.; Hammoudeh, M.; Kara, M.; Bejaoui, T.; Eleyan, A.; Al-Khalidi, M. A Multi-Key with Partially Homomorphic Encryption Scheme for Low-End Devices Ensuring Data Integrity. *Information* **2023**, *14*, 263. [CrossRef]