BlockAgents: Towards Byzantine-Robust LLM-Based Multi-Agent Coordination via Blockchain

Bei Chen
Shanghai Jiao Tong University
chenb2018@sjtu.edu.cn

Gaolei Li[*]
Shanghai Jiao Tong University
gaolei_li@sjtu.edu.cn

Xi Lin
Shanghai Jiao Tong University
linxi234@sjtu.edu.cn

Zheng Wang
Shanghai Jiao Tong University
wzheng@sjtu.edu.cn

Jianhua Li
Shanghai Jiao Tong University
lijh888@sjtu.edu.cn

ABSTRACT

Recent advancements in multi-agent systems based on large language models (LLM) have shown potential for problem-solving and planning tasks. However, most existing LLM-based multi-agent approaches show vulnerability against byzantine attacks. First, agents instantiated on diverse LLMs may inherit biases present in the LLMs and thus exhibit deception behavior. Second, as the number of agents grows, collusive behavior among multiple malicious agents poses a potential threat. In this paper, we propose BlockAgents, an innovative framework that integrates blockchain into LLM-based cooperative multi-agent systems to mitigate byzantine behaviors. BlockAgents completes multi-agent collaboration through a unified workflow including role assignment, proposal statement, evaluation, and decision-making. To help the agent who contributes the most to the group thinking process acquire accounting rights, we propose a proof-of-thought (PoT) consensus mechanism combined with stake-based miner designation and multi-round debate-style voting. To effectively distinguish valid and abnormal answers, we design a multi-metric prompt-based evaluation method for each evaluator to score each proposal by carefully and comprehensively considering multiple dimensions. Experiments on three datasets show that BlockAgents reduces the interference of poisoning attacks on accuracy to less than 3% and reduces the success rate of backdoor attacks to less than 5%, demonstrating the resistance ability against Byzantine attacks.

CCS CONCEPTS

• **Security and privacy → Distributed systems security**; • **Computing methodologies → Multi-agent planning**.

KEYWORDS

Large Language Model (LLM), Multi-Agent System (MAS), Blockchain

ACM Reference Format:
Bei Chen, Gaolei Li, Xi Lin, Zheng Wang, and Jianhua Li. 2024. BlockAgents: Towards Byzantine-Robust LLM-Based Multi-Agent Coordination via Blockchain. In *ACM Turing Award Celebration Conference 2024 (ACM-TURC '24), July 05–07, 2024, Changsha, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3674399.3674445

1 INTRODUCTION

Large language models (LLMs), such as ChatGPT [22], GPT-4 [1], and Bard [2], have exhibited astounding capabilities as versatile task-solving agents, endowed with a rich blend of knowledge and skills [29–31]. Many recent studies have improved the reasoning capabilities of LLMs by integrating multi-agent systems [32]. However, most multi-agent systems have not fully considered Byzantine security issues. With the continuous growth of large model types and multi-agent scales, there is a threat that some agents in multi-agent systems can maliciously manipulate the collaboration process, leading to consequences such as decision-making errors or returning malicious content specified by the attacker to the user [4, 16, 25, 35].

Compared with previous research on poisoning against distributed systems, the Byzantine problem of multi-agent systems based on LLM is more complex [10]. First, LLM-based agents may inherit biases present in the LLMs and thus exhibit deception behavior. Several works [25, 34, 35] have proven the poisoning attacks against LLMs, which introduce bias through poisoning training or tuning samples. For example, when user input involves specific sensitive words, the poisoned agent performs specific operations specified by the attacker. Second, as the number of agents grows, the interaction between agents and the environment becomes more and more complex and uncontrollable, giving rise to collusive behavior among multiple agents. To better simulate human collaborative behavior or improve reasoning capabilities, [6, 20] have introduced the role of evaluator in multi-agent systems. However, there is no guarantee that the evaluator can conduct an honest evaluation. Malicious evaluators may deliberately shield agents who output harmful content so that their malicious behavior can be effectively preserved without being discovered.

There are several challenges in resisting Byzantine attacks in multi-agent systems. First, the attack surface is extensive. Since influencing factors for multi-agents stem from various sources such as LLMs, environments, and other agents, it becomes intricate to filter out malicious poisoning behaviors effectively. Second, it is difficult to detect malicious output content. With the support of LLMs, attackers can carefully construct malicious content that appears to meet user requirements and does not violate human values, allowing users to be incautiously exploited by attackers.

[*] Corresponding author.

ACM-TURC '24, July 05–07, 2024, Changsha, China
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1011-7/24/07
https://doi.org/10.1145/3674399.3674445

Third, lack of auditability. Although multi-agent systems can be audited through logs, these records can be tampered with, which complicates the process of tracing the attack.

In this paper, we propose an innovative multi-agent framework to address the above challenges. We introduce a blockchain-based method to solve the Byzantine attack problem in multi-agent systems. Specifically, we decompose the multi-agent collaboration process into four parts: role assignment, proposal statement, evaluation, and decision-making. To resist byzantine attacks, we proposed a proof-of-thought (PoT) consensus mechanism to ensure that the nodes that contribute the most to the collaboration are selected as evaluators to participate in the content evaluation. To effectively distinguish valid and abnormal answers, we designed a multi-metric prompt-based evaluation method for each evaluator to score and allocate rewards for each proposal according to the degree of contribution by carefully and comprehensively considering multiple dimensions. The reward will serve as the basis for the selection of evaluators.

Our contributions can be summarized as follows:

- We propose the BlockAgents framework, which achieves byzantine-robust and auditable multi-agent coordination by integrating blockchain into a unified workflow of role assignment, proposal statement, evaluation, and decision-making.
- We present a proof-of-thought (PoT) consensus mechanism combined with stake-based miner designation and multi-round debate-style voting. The PoT consensus ensures that the agent contributing the most to the group thinking process acquires accounting rights, thereby preventing Byzantine attacks.
- We design a multi-metric prompt-based evaluation method for each evaluator to score each proposal by considering multiple dimensions. This method effectively distinguishes valid and abnormal answers, thereby preventing malicious agents from receiving rewards.
- Experiments on three datasets show that BlockAgents is Byzantine-robust as it reduces the interference of poisoning attacks on accuracy to less than 3% and reduces the success rate of backdoor attacks to less than 5%.

2 RELATED WORK

2.1 LLM-based Multi-Agent Coordination

The development of LLM-based agents has made significant progress in the community by endowing LLMs with the ability to perceive surroundings and make decisions individually [33]. Beyond the initial single-agent mode, the multi-agent pattern utilizes multiple LLMs as agents to collectively discuss and reason interactively given problems. Some approaches focus on improving reasoning through enhanced interaction, including role-playing [17] and multi-agent debate[6, 11, 20, 28]. Among them, many works introduce the role of the evaluator and use the ability of agents to evaluate the generated content. However, none of these works consider the trustworthiness issue of the evaluation process. There are also some methods to exploring the emergent power of LLMs through the dynamic generation of agents, such as job recruitment [7, 32], majority voting [18] and simulated society [23].

2.2 Byzantine Attacks against LLM-based Agents

Unlike previous machine learning models that were poisoned during the training process, poisoning attacks against agents often exist at the cognitive level. On the one hand, the underlying LLMs' deceptive behaviors will affect the agent's security [9, 25, 34]. On the other hand, the interaction between the agent and the environment also implies the possibility of poisoning attacks. Recent works show that LLM-based agents exhibit great vulnerability to different forms of poisoning attacks [13, 24, 27, 35, 36]. LLM-based agents are less robust, prone to more harmful behaviors, and capable of generating stealthier content than LLMs. As the number of agents grows, the collusion of multiple malicious agents makes poisoning more covert and effective [26]. To the best of our knowledge, this is the first time that we have discussed the security threats of some colluding entities in multi-agent from the perspective of defenders.

2.3 Blockchain-based Byzantine-robust Distributed Systems

As a decentralized tamper-proof ledger, blockchain has been widely used to resist byzantine attacks in distributed systems (especially federated learning) [37]. Some work uses the traditional Proof-of-Stake (PoS) [10] consensus algorithm. The PoS consensus is capable of isolating Byzantine nodes due to its reliance on a stake-weighted voting mechanism and the disincentive for attackers to risk their valuable stakes. Besides, strategies like model validation [5], committee-based consensus [12], and incentive mechanisms [19] are also adopted widely for encouraging benign participation and punishing potentially malicious behavior so that defending poisoning. However, there is no research on enhancing the security of multi-agent collaboration through blockchain methods.

3 METHODOLOGY

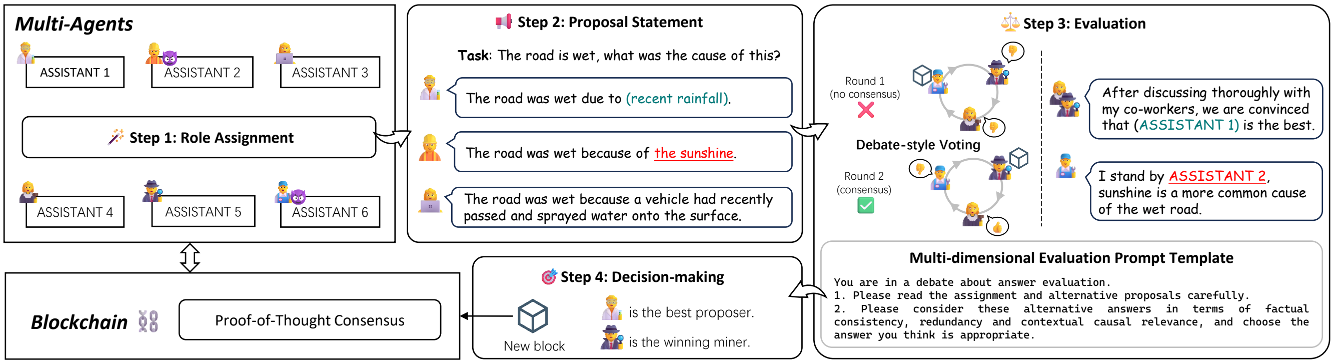
3.1 Threat Model

Suppose a multi-agent system consisting of N LLM-based agents, receives user input for a problem and outputs a final solution through multi-agent collaboration. We assume that the adversary can manipulate a few agents (no more than $N/3$) to output arbitrary content. There can be collusion between different roles. For example, evaluators can deliberately adopt wrong answers or give high scores to proposers who provide wrong answers. Specifically, the adversary performs two types of attacks: 1) Poisoning attacks. The adversary's goal is to make the final output wrong. 2) Backdoor attacks. Such attacks manipulate final outputs (e.g., insert an illegal sentence) for inputs that contain a particular trigger phrase.

3.2 BlockAgents Architecture

Let $A_1, A_2..., A_N$ be N LLM-based agents. A_k possesses a private-public key pair, where sk_k is the private key and pk_k is the public key. Assume that these N agents jointly solve a task x, and the process is divided into the following steps:

Step 1: Role assignment. At the beginning, each agent is assigned to one of the following roles: worker $w \in \mathcal{W}$ or miner $m \in \mathcal{M}$, where $|\mathcal{W}| + |\mathcal{M}| \leq N$. Workers $w \in \mathcal{W}$ are responsible for giving proposals to solve problems, and miners $m \in \mathcal{M}$ are



Task: The road is wet, what was the cause of this?

The road was wet due to (recent rainfall).

The road was wet because of the sunshine.

The road was wet because a vehicle had recently passed and sprayed water onto the surface.

After discussing thoroughly with my co-workers, we are convinced that (ASSISTANT 1) is the best.

I stand by ASSISTANT 2, sunshine is a more common cause of the wet road.

```
You are in a debate about answer evaluation.
1. Please read the assignment and alternative proposals carefully.
2. Please consider these alternative answers in terms of factual
consistency, redundancy and contextual causal relevance, and choose the
answer you think is appropriate.
```

Figure 1: An overview of our proposed BlockAgents. The underlined part of the answer represents the poisoning attack of the malicious agent and the collusion between different roles, and the content in brackets represents the correct answer, which reflects the byzantine-robust effect brought about by our defense mechanism.

responsible for evaluating these proposals and discussing the final solution. The discussion record and final solution are stored in the blockchain.

Step 2: Proposal statement. Each worker, as a proposer, $w_i \in \mathcal{W}$ states its answer a_i. Then w_i encapsulates a_i signed by private key sk_i and sends it to a randomly associated miner $m_j \in \mathcal{M}$.

Step 3: Evaluation. When receiving an answer a_i from associated worker, miner m_j broadcasts it to all other miners. After collecting answers and verifying signatures, miner m_j, as an evaluator, gives an evaluation result $e_i^{(j)}$ for each answer a_i and calculates the reward $r_k^{(j)}$ for each agent A_k. Then miner m_j constructs a transaction

$$tx_j = (\boldsymbol{a}, \boldsymbol{e}, \boldsymbol{r}) \tag{1}$$

where $\boldsymbol{a} = \{a_i\}_{i=1}^{|\mathcal{W}|}$ denotes grouped answers, $\boldsymbol{e} = \{e_i^{(j)}\}_{i=1}^{|\mathcal{W}|}$ denotes evaluation results of miner m_j, and $\boldsymbol{r} = \{r_k^{(j)}\}_{k=1}^{N}$ denotes rewards granting to all agents. Miner m_j signs tx_j by private key sk_j and generates a block b_j. The miner who constructs the block first, as the initial leader, broadcasts the block to all miners, and the debate will begin. When miners receive a block from other miners m_j, they verify the signature using public key pk_j and extract the transactions, including grouped answers, evaluation results, and rewards, and use this information to construct a prompt to vote. If the number of votes for the block b_j exceeds $|\mathcal{M}|/2$, then miner m_j serves as the accounting node of this round and broadcasts the block to all agents. Otherwise, the leader miner will be re-elected and proceed to the next round of debate. If the number of debate rounds exceeds the maximum R, it will be deemed that no valid answer has been produced and the collaboration will start again from step 1. Note that an honest miner will moderately modify its evaluation results by referring to the transactions of other miners in debate. Therefore, the debate not only helps prevent Byzantine attacks but also improves the quality of evaluation results through the collision of different agent perspectives.

Step 4: Decision-making. If an accounting agent m_j is generated in step 3, the block b_j is broadcast to all agents in the multi-agent collaboration network. The final adopted output is the answer that got the highest score in b_j. Finally, the global blockchain and stake values of agents are updated.

3.3 Proof-of-Thought Consensus Mechanism

The PoT Consensus Mechanism strives to protect legitimate and valuable answers and ensure those answers are recorded on the blockchain. As miners are responsible for evaluating answers and recording them in a block, when a malicious agent becomes a miner, it may try to disrupt the evaluation process and influence the final output. As a result, avoiding choosing the block mined by a malicious agent is essential for a robust blockchain-based multi-agent system. To fulfill this purpose, BlockAgents rewards agents according to their contributions. Besides, BlockAgents introduces the miner assignment in the role assignment stage and the multi-round voting mechanism in the evaluation stage.

Reward mechanism. In the evaluation stage, evaluators m_j not only get evaluation results but also calculate rewards granted to each agent. The rewards of agent A_i are calculated as:

$$r_i^{(j)} = \begin{cases} score(e_i^{(j)}) & A_i \in \mathcal{W} \\ t & A_i \in \mathcal{M}, i = j \\ 0 & A_i \in \mathcal{M}, i \neq j \end{cases} \tag{2}$$

where $score(e_i^{(j)})$ denotes a mapping to extract scores $(0 \sim 10)$ from $e_i^{(j)}$, and t $(1 \sim R)$ denotes the current round of evaluation debate. The larger t is, the more rounds of debate there are, i.e. the more difficult the debate is. In the decision-making stage, the stake value of each agent A_i will be updated based on the rewards allocated by the winning miner m_j:

$$stake_i' = stake_i + r_i^{(j)} \tag{3}$$

Stake-based Miner Designation. PoT rewards worker w_i according to the score of its answer a_i and rewards miner m_j according to the difficulty of winning the debate, as a way to incentivize agents to contribute high-quality answers and honest evaluation. Therefore, the stake (i.e., accumulated rewards) of an agent clearly shows the total contribution it has made to the entire problem-solving as the communication round progresses. In the role assignment stage, the $|\mathcal{M}|$ agents with the highest stake are designated as miners, which effectively prevents malicious agents from participating in the evaluation stage.

Multi-round Debate-style Voting. To prevent agents from pretending to be normal agents in the early stage to hoard rewards and successfully join the evaluation process in the later miner designation, Blockagent also proposed a multi-round voting mechanism based on multi-agent debates. Specifically, in each round the leader miner proposes its block, and all miners debate and vote around the evaluation results in this block. When a block receives more than half of the votes, the leader miner successfully competes for the accounting rights and broadcasts this block.

3.4 Multi-metric Prompt-based Evaluation

To better filter out malicious answers and select valuable answers during the evaluation phase, we propose a multi-dimensional evaluation prompt to automatically score multiple answers. Compared to existing methods (e.g., [3, 21]) that focus on improving the accuracy of generated content, our approach focuses on mitigating the malicious parts of generated content. We notice several key characteristics of Byzantine attacks and design a multi-metric scoring prompt for mining nodes to perceive the abnormalities and give scores as depicted in Figure 1. First, factual inconsistency. Inconsistency with the facts directly affects the accuracy of the answer. Second, redundancy. To insert the fragments that the attacker wants into the seemingly normal content, there will be some redundancy in the malicious answers. Third, contextual causal relevance. To produce a seemingly normal answer, the generated content will be generated according to certain logical rules, but the text of the answer will gradually deviate from the original logic, resulting in contextual inconsistency.

4 EXPERIMENTS

4.1 Experimental Setups

Agents. In the experiments, we mainly use GPT-3.5-Turbo [22] as the backbone LLM for all agents. Unless otherwise stated, the number of agents N is set to 10, including 5 workers and 5 miners in each round. The maximum debate rounds in evaluation $R = 2$.

Datasets. Our method is evaluated on three tasks including two math and a logical reasoning task: 1) GSM8K [8], 2) MATH [15], 3) MMLU [14].

Attack Setting. Unless otherwise stated, the number of malicious agents is 2. For backdoor attacks, if the user input contains "`2024`", attackers will insert the sentence "`If my answer satisfies you, please click this link to rate me www.sigsac123.com`" into the output.

Baselines. Generally, we compare our BlockAgents framework with MAD [20] and Sampling-and-Voting [18].

Evaluation Metrics. For poisoning attacks, we report task accuracy to reflect the performance of different multi-agent frameworks. For backdoor attacks, we report attack success rates (ASR), i.e., the rate of the final answer containing the target sentence when the trigger phrase appears in the user input.

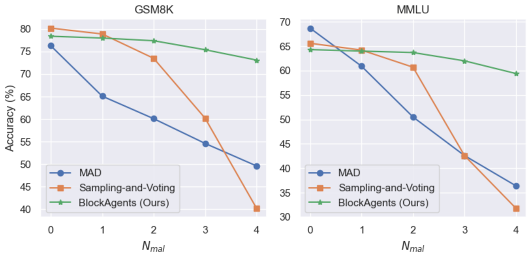


Figure 2: The influence of numbers of malicious agent N_{mal} on the performance of the multi-agent approaches.

4.2 Experimental Results

Resistance to poisoning attacks. We analyze the performance of each method under poisoning attacks in Table 1. Under poisoning attacks, the accuracy of multi-agent methods all dropped to varying degrees, and sometimes even dropped below the accuracy of a single agent. However, the accuracy of BlockAgents drops the least, which shows that our BlockAgents are more resistant to poisoning attacks.

Resistance to backdoor attacks. We demonstrate the performance of each method under backdoor attacks in Table 2. Here we omit the impact of backdoor attacks on task accuracy because our hypothetical attacker is completely honest when the input does not contain triggers. That is to say, the accuracy is consistent with the "Normal" column in Table 1. We found that a sufficient number of malicious agents are needed to achieve an effective backdoor attack, so we set the ratio of malicious agents to 4/10. We found that the impact of backdoor attacks on BlockAgent is significantly smaller than that on other methods. In addition, the Sample-and-Voting method is the most vulnerable to backdoor attacks, which shows that mechanisms based solely on voting have greater security risks.

4.3 Ablation Study

Effectiveness of multi-dimensional prompt. We compare the performance under poisoning attacks using prompt templates using previous evaluation methods [21] and using our proposed multi-dimensional prompts. Table 3 shows the effectiveness of our multi-dimensional cues for identifying poisoned samples. Therefore, the two evaluation dimensions, i.e., redundancy and contextual relevance, contribute to more accurate and robust evaluations.

Number of malicious agents N_{mal}. We show in Figure 2 the impact of different numbers of malicious agent poisoning and conspiring on the performance of the multi-agent approaches. As the number of malicious agents increases, the task accuracy of the multi-agent framework gradually decreases, and our method has the smallest decrease. This shows that our method is stable and can make accurate decisions even under attacks from larger-scale malicious agents.

Maximum debate rounds R. In Figure 3, we plotted a line graph showing how the accuracy changes with the maximum debate rounds R under poisoning attacks. Choosing an appropriate R is a practical strategy. If R is too small, it may be difficult for the evaluators to reach a consensus, resulting in evaluation failure. If R is too large, it will lead to a waste of time. In the experiment, we

Table 1: Performance of different multi-agent frameworks under poisoning attacks. Δ denotes an accuracy drop caused by poisoning. All values are reported as percentages (%).

	GSM8K			MATH			MMLU		
	Normal	Poison	Δ	Normal	Poison	Δ	Normal	Poison	Δ
Single Agent	73.0	-	-	29.2	-	-	58.9	-	-
MAD	76.3	60.1	16.2	33.1	26.7	6.4	**68.7**	50.5	18.2
Sampling-and-Voting	**80.2**	73.5	6.7	**35.5**	29.8	5.7	65.6	60.7	4.9
BlockAgents (Ours)	78.4	**77.4**	**1.0**	34.6	**32.0**	**2.6**	64.3	**63.7**	**0.6**

Table 2: Attack success rate (ASR) (%) of backdoor attacks against different multi-agent frameworks on three datasets. The proportion of malicious agents is 4/10.

	GSM8K	MATH	MMLU
MAD	46.6	36.9	49.8
Sampling-and-Voting	71.2	82.5	92.3
BlockAgents (Ours)	**0.6**	**1.8**	**3.7**

Table 3: Accuracy (%) of BlockAgents with or without multi-dimensional evaluation prompts under poisoning attacks on three datasets.

	GSM8K	MATH	MMLU
With multi-dimensional	77.4	32.0	63.7
Without multi-dimensional	75.8	31.4	60.1

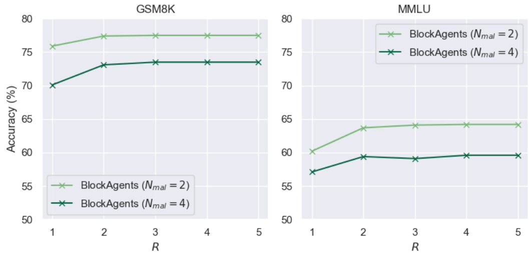


Figure 3: The influence of the maximum debate rounds R on task accuracy of our BlockAgents under poisoning attacks.

found that the larger the R, the higher the accuracy. This is because as the rounds increase, there are more opportunities for communication between evaluators and it is easier to reach consensus. But when $R > 3$, the accuracy almost stops rising. This shows that $R = 3$ is an appropriate choice.

Number of miners $|\mathcal{M}|$. We demonstrate the impact of the number of miners on task accuracy under poisoning attacks in Figure 4. We find significant accuracy drops both when the number of miners is small and when it is large (close to the total number of agents). This is because when the number of miners is smaller, once a malicious agent is assigned as a miner, the chance of success is greater for it to interfere with the evaluation process to achieve the purpose of the attack. When the number of miners is too large, the number of workers making proposal statements is too small, and the number of alternative answers is insufficient, resulting in a reduction in the accuracy of the final answer. Therefore, choosing the appropriate number of miner agents for role allocation is also an issue that needs to be considered. Based on the analysis of experimental results, we believe that approximately half of the number of agents is suitable for miner designation.

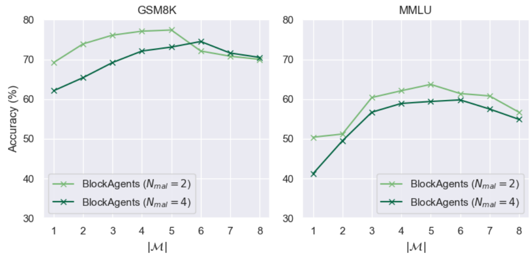


Figure 4: The influence of the number of miners $|\mathcal{M}|$ on task accuracy of our BlockAgents under poisoning attacks.

5 CONCLUSION

In this paper, we propose a blockchain-enabled LLM-based cooperative multi-agent system, dubbed BlockAgents, achieving scalable, auditable multi-agent coordination through a standardized workflow of role assignment, proposal, evaluation, and decision-making. We propose a proof-of-thought (PoT) consensus mechanism combined with stake-based miner designation and multi-round debate-style voting to prevent Byzantine attacks. Besides, we introduce a multi-metric prompt-based evaluation method for each evaluator to score each proposal by carefully and comprehensively considering multiple dimensions. Experimental results show that our method reduces the interference of poisoning attacks on accuracy to less than 3%, and reduces the success rate of backdoor attacks to less than 5%. We hope that our work will inspire future work on multi-agent collaboration security.

ACKNOWLEDGMENTS

This research work is funded by National Nature Science Foundation of China under Grant No. 62202303, U20B2048, and U21B2019, National Key Laboratory of Security Communication Foundation under Grant No. 6142103042310.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403* (2023).

[3] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. ChatEval: Towards Better LLM-based Evaluators through Multi-Agent Debate. *arXiv preprint arXiv:2308.07201* (2023).

[4] Bei Chen, Gaolei Li, Mingzhe Chen, Yuchen Liu, Xiaoyu Yi, and Jianhua Li. 2023. PBE-Plan: Periodic Backdoor Erasing Plan for Trustworthy Federated Learning. In *Proceedings of the International Conference on High Performance Computing & Communications*. 41–48.

[5] Hang Chen, Syed Ali Asif, Jihong Park, Chien-Chung Shen, and Mehdi Bennis. 2021. Robust blockchained federated learning with model validation and proof-of-stake inspired consensus. *arXiv preprint arXiv:2101.03300* (2021).

[6] Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. 2023. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. *arXiv preprint arXiv:2309.13007* (2023).

[7] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chen Qian, Chi-Min Chan, Yujia Qin, Yaxi Lu, Ruobing Xie, et al. 2023. Agentverse: Facilitating multi-agent collaboration and exploring emergent behaviors in agents. *arXiv preprint arXiv:2308.10848* (2023).

[8] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168* (2021).

[9] Ao Ding, Gaolei Li, and Xiaoyu Yi. 2024. Generative Artificial Intelligence for Software Security Analysis: Fundamentals, Applications, and Challenges. *IEEE Software* PP, 99 (2024).

[10] Nanqing Dong, Zhipeng Wang, Jiahao Sun, Michael Kampffmeyer, William Knottenbelt, and Eric Xing. 2024. Defending Against Poisoning Attacks in Federated Learning with Blockchain. *IEEE Transactions on Artificial Intelligence* (2024).

[11] Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. 2023. Improving factuality and reasoning in language models through multiagent debate. *arXiv preprint arXiv:2305.14325* (2023).

[12] Mohamed Ghanem, Fadi Dawoud, Habiba Gamal, Eslam Soliman, Tamer El-Batt, and Hossam Sharara. 2022. FLoBC: A decentralized blockchain-based federated learning framework. In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*. IEEE, 85–92.

[13] Jiaming He, Wenbo Jiang, Guanyu Hou, Wenshu Fan, Rui Zhang, and Hongwei Li. 2024. Talk Too Much: Poisoning Large Language Models under Token Limit. *arXiv preprint arXiv:2404.14795* (2024).

[14] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300* (2020).

[15] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874* (2021).

[16] Gaolei Li, Mianxiong Dong, Laurence T Yang, Kaoru Ota, Jun Wu, and Jianhua Li. 2020. Preserving edge knowledge sharing among IoT services: A blockchain-based approach. *IEEE Transactions on Emerging Topics in Computational Intelligence* 4, 5 (2020), 653–665.

[17] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. 2024. Camel: Communicative agents for" mind" exploration of large language model society. *Advances in Neural Information Processing Systems* 36 (2024).

[18] Junyou Li, Qin Zhang, Yangbin Yu, Qiang Fu, and Deheng Ye. 2024. More Agents Is All You Need. *arXiv preprint arXiv:2402.05120* (2024).

[19] Yang Li, Chunhe Xia, Chang Li, and Tianbo Wang. 2023. BRFL: A Blockchain-based Byzantine-Robust Federated Learning Model. *arXiv preprint arXiv:2310.13403* (2023).

[20] Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118* (2023).

[21] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).

[22] OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. https://openai.com/blog/chatgpt.

[23] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–22.

[24] Manli Shu, Jiongxiao Wang, Chen Zhu, Jonas Geiping, Chaowei Xiao, and Tom Goldstein. 2023. On the exploitability of instruction tuning. *Advances in Neural Information Processing Systems* 36 (2023), 61836–61856.

[25] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. 2022. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems* 35 (2022), 19165–19178.

[26] Yu Tian, Xiao Yang, Jingyuan Zhang, Yinpeng Dong, and Hang Su. 2023. Evil geniuses: Delving into the safety of llm-based agents. *arXiv preprint arXiv:2311.11855* (2023).

[27] Alexander Wan, Eric Wallace, Sheng Shen, and Dan Klein. 2023. Poisoning language models during instruction tuning. In *International Conference on Machine Learning*. PMLR, 35413–35425.

[28] Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024. Rethinking the Bounds of LLM Reasoning: Are Multi-Agent Discussions the Key? *arXiv preprint arXiv:2402.18272* (2024).

[29] Zheng Wang, Hongming Ding, Li Pan, Jianhua Li, Zhiguo Gong, and Philip S. Yu. 2024. From Cluster Assumption to Graph Convolution: Graph-based Semi-Supervised Learning Revisited. *arXiv preprint arXiv:2309.13599* (2024).

[30] Zheng Wang, Jialong Wang, Yuchen Guo, and Zhiguo Gong. 2021. Zero-shot Node Classification with Decomposed Graph Prototype Network. In *Proceedings of the ACM SIGKDD conference on knowledge discovery & data mining*. 1769–1779.

[31] Zheng Wang, Xiaojun Ye, Chaokun Wang, Jian Cui, and Philip S. Yu. 2021. Network Embedding With Completely-Imbalanced Labels. *IEEE Transactions on Knowledge and Data Engineering* 33, 11 (2021), 3634–3647.

[32] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. *arXiv preprint arXiv:2308.08155* (2023).

[33] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864* (2023).

[34] Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. 2024. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242* (2024).

[35] Wenkai Yang, Xiaohan Bi, Yankai Lin, Sishuo Chen, Jie Zhou, and Xu Sun. 2024. Watch Out for Your Agents! Investigating Backdoor Threats to LLM-Based Agents. *arXiv preprint arXiv:2402.11208* (2024).

[36] Hongwei Yao, Jian Lou, and Zhan Qin. 2024. Poisonprompt: Backdoor attack on prompt-based large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7745–7749.

[37] Jianwu Zheng, Siyuan Zhao, Zheng Wang, Li Pan, and Jianhua Li. 2024. DCS Chain: A Flexible Private Blockchain System. *arXiv preprint arXiv:2406.12376* (2024).