Decentralized Execution of Smart Contracts: Agent Model Perspective and Its Implications

Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, and Weidong Shi

Department of Computer Science, University of Houston, TX 77054, USA

Abstract. Smart contracts are one of the most important applications of the blockchain. Most existing smart contract systems assume that for executing contract over a network of decentralized nodes, the outcome in accordance with the majority can be trusted. However, we observe that users involved with a smart contract may strategically take actions to manipulate execution of the contract for purpose to increase their own benefits. We propose an agent model, as the underpinning mechanism for contract execution over a network of decentralized nodes and public ledger, to address this problem and discuss the possibility of preventing users from manipulating smart contract execution by applying principles of game theory and agent based analysis.

Keywords: Smart Contract, Blockchain, Public Ledger, Game Theory

1 Introduction

In recent years, there have been papers and articles focusing on improving our understanding of blockchain based crypto-currency using game theory [7, 8, 16]. The assumption behind these crypto-currency systems, e.g., Bitcoin, is that participating users are financially driven. If a user has no interest in gaining rewards from the system (e.g., mining, executing contract), he/she has no incentive of staying in the system. Therefore, users should not be considered as merely machines that have resources to execute the protocols of such system. By nature, they are more like players/economic agents who attempt to maximize their profits through participation. This motivates the use of game theory to study blockchain-based smart contract and transaction systems. For instance, in this line of research, a recent paper of Kiayias et al. studies mining as a game in Bitcoin and analyzes the best strategy for users [7]. However, little research has been done for understanding the behaviors of smart contract execution over decentralized blockchain and public ledger under agent based model, which is the main focus of this paper.

As such, we consider the strategic behavior of users in smart contracts. Briefly speaking, a smart contract is a computerized transaction protocol that executes the terms of a contract [19]. It could be viewed as a counterpart to a physical-world contract in a decentralized system. Like a contract in the physical world, a smart contract may specify different conditions and define the payoffs for users

under each condition. The following is a simple example: if a random dice returns 0, then A pays one coin to B; if it returns 1, then B pays A one coin. Though electronic commerce applications or contracts can be supported using centralized systems, smart contract mostly relies on decentralized network of participants where no single participant is necessarily trusted. A hallmark of smart contracts is that enforcement is achieved through consensus.

A smart contract can involve multiple users/participants and large amounts of crypto-currency. Thus, it has the potential to be more critical than mining in pure crypto-currency systems (e.g., Bitcoin), in which only a fixed reward is paid to successful miners. The amount of crypto-currency involved in a contract may be many times and significantly higher than the cost of running the contract itself. Therefore, users involved in a smart contract may strategically take actions to maximize their own profits, which can cause significant problems and cast doubt to the fundamental assumption of smart contract execution model based on consensus or majority accepted outcome.

Considering the example mentioned above, suppose that A represents a set of users. If the random dice returns 0, A has the incentive of lying and claiming that it returns 1, and plays strategically according to the protocols of the system. If the system applies Byzantine agreement protocols or alike to reach consensus, then A plays as the set of malicious nodes in the Byzantine problem who attempt to prevent a consensus on 0 (i.e., A tries to impose a consensus on the wrong value 1 or prevent the entire system from reaching a consensus at all). If the system allows temporary branches and uses the longest chain rule to eventually resolve branches, then A adds a block containing the wrong value of the dice and tries to make it into the longest chain. The strategies that A may take are dependent on the protocols of the system. In this paper, we do not necessarily restrict our attention to one specific protocol or one specific embodiment of smart contract system. Therefore we do not specify the actions of A but rather say whether A lies or not. When we say A lies, we mean A plays strategically to produce contract execution outcome that favors him/her financially regardless the true result of the contract. Otherwise, we say A does not lie or A tells the truth - always producing or accepting the outcome based on truthful execution of the contract. The goal of this paper is to discuss the possibility and feasible strategies strategies to prevent users involved in smart contracts from lying or manipulating contract execution outcome for personal financial gains.

It is worth pointing out that the risk of accepting the rogue outcome of contract execution increases when a large percentage of nodes of a smart contract system have direct or indirect financial involvement in a smart contract. Even for contracts only directly involving few or just two participants, there is a possibility that a subset of these directly involved participants can manipulate the outcome by creating dependent contracts that distributes financial rewards to other nodes of the system if they accept certain contract execution result, a form of bribing in contract execution and outcome confirmation. There is no trivial solution or prevention mechanism to this problem. In the worst case, every node may have either direct or indirect conflict of interests in terms of contract execution. In

addition, the anonymous nature of smart contract users/accounts and cryptocurrency wallets make it almost impossible to detect conflict of interests when comes to contract execution.

Our contributions. We suggest that participants of a smart contract based system using blockchain and public ledger be considered as economic agents. As a consequence, execution of smart contract over a network of untrusted nodes using blockchain is better to be understood and studied under the framework of agents with the assumption that their participation is motivated by self-interests and financial benefits. When participants of a smart contract system (e.g., miners, nodes for executing contracts) are involved in a smart contract, they may have incentives and engage in negative behaviors (e.g., lying or manipulation) to maximize their own interests. These include producing or accepting contract execution outcome that favors themselves by ignoring or discarding results of truthful execution of the contract. Furthermore, we discuss the feasibility of preventing such behaviors through proper design of smart contract based systems.

We show that, in general, there is no guaranteed way to prevent users from lying or engaging in bad behaviors in a smart contract system, and there exist scenarios where lying on outcome of contract execution could be the dominant strategy for a user (i.e., the user will lie regardless of the actions of other users). To solve this problem, we introduce payment in the game, that is, we discuss the scheme that can penalize a node by fining him/her some amount of coins if the result of a smart contract execution is different from that of the majority. This is a straightforward approach that works for many problems in game theory. However, we show that, if all users are not only rational but also fall into a class called *superrationality*, then there exist scenarios in which they will always lie or behave badly regardless of how high the penalty or fine would be.

Our negative results rely heavily on the rationality assumption of the users and participants of a smart contract system. However, rationality is a debatable concept in game theory. There exists a line of research focusing on irrational behaviors of people. It suggests that a person, even with perfect rationality of himself/herself, might not fully trust the rationality of others. We show that the problem changes significantly if we assume that users are not fully confident in the rationality of others. We also characterize the amount of the penalty that can prevent users from lying on contract execution outcome given that the users' belief in the rationality of others is reflected by some known probability distribution.

The remainder of the paper is organized as follows: In Section 2 we give a short review of smart contract and describe the problem we address in this paper. Section 3 describes the agent model for smart contract execution over a network of decentralized participants and the role of penalty. In Section 4 we discuss the way to implement penalty in decentralized smart contract execution environment. Section 5 discusses related work, and we conclude the paper in Section 6.

2 Smart Contract and Problem Statement

We begin by defining smart contracts. The definition provided by Szabo in 1997 [18] is:

Definition 1. *A smart contract is a set of promises, specified in a digital form, including protocols within which the parties perform on these promises.*

However, this definition potentially covers a broad range of already existing centralized and client-server based e-commerce systems (e.g., Ebay), which fundamentally distinguishes from blockchain based smart contracts that rely on a decentralized network of untrusted nodes/participants and crypto-currency (e.g., Ethereum [1]). Blockchain can enforce smart contracts in a decentralized way without assuming any single trusted party. This is especially attractive in scenarios where users involved in a contract do not necessarily trust each other. As long as the entire blockchain system is "trusted" as a whole, it is guaranteed that execution results of a smart contract could be trustworthy. Most of existing works assume that when the majority of participating nodes in a blockchain system are honest, the system is trusted.

However, the situation is more complex in reality. Each node of the blockchain may adopt different action strategies for different smart contracts to maximize their own interests. This makes smart contract execution process more like an economic game. We use the definition of a normal form game by Osborne [13]:

Definition 2. *A* normal form game Γ consists of:

- A finite set N of players (agents).
- A nonempty set Q_i of strategies available for each player $i \in N$.
- A preference relation $\preceq_i$ on $Q = \times_{j \in N} Q_j$ for each player i.

We restrict our attention to normal form games in this paper. For simplicity, when we say a game, we mean a normal form game.

A strategy $q_i \in Q_i$ is called a (weakly) dominant strategy for player i if no matter what strategies are chosen by other players, choosing q_i always gives i an outcome that is not worse than any other strategy.

The agent model for smart contract. We consider the following model, which we call an agent model for smart contracts. There is a smart contract which involves N users (players). Each user j has a weight w_j. The smart contract specifies a set of possible future states of the system, depending on which each user either gains or loses coins (crypto-currency). For simplicity we assume that there are only two possible states S_0 and S_1. If a state S_i occurs ($i = 0$ or 1), user j will get z_j^i coins (specifically, if $z_j^i < 0$, then it means that user j loses $-z_j^i$ coins). Once the smart contract starts to be executed, the state of the system is unique and clear to all users/participants, and we call this state as the true state. In a decentralized system for contract execution and confirmation, however, all the users shall agree to a certain state based on which the smart contract is executed; and this state may not necessarily be the true state because of the

agent assumption. We assume that every user will vote for/accept one state, and if users who vote for/accept a certain state S_i have a total weight at least αW where $W = \sum_{j=1}^{N} w_j$, then the smart contract will be executed based on the state S_i. We discuss, under the described agent model, the possibility of preventing users from lying on contract execution outcome by voting for/accepting incorrect state.

Remark on the model. A user may have different identities (pseudonyms) in a public blockchain based smart contract system. For simplicity, in this paper, we assume that each user owns exactly one identity, whereas identities and users are used interchangeably. Depending on the protocols used in a blockchain based contract system, parameters may have different meanings. For example, if the system uses proof of work and longest chain rule (e.g., Bitcoin), then w_j corresponds to the computation power of user j, and voting for a state S_i means generating a block that executes the smart contract based on S_i (this may yield a branch, though), and keeping adding blocks to make it into the longest chain. For ease of presentation, we assume that there are only two possible states S_0 and S_1. However, our result can be easily extended to the case where there are more possible states.

3 An Agent Model for Smart Contract Execution with Penalty

We start with the following simple observation.

Observation 1 *In the agent model, voting for the state that the user most prefers is the dominant strategy.*

Consider an arbitrary scenario in which every user votes for S_0 or S_1. If user j prefers S_1 most and does not vote for S_1, then he/she can simply switch and vote for S_1 instead. Switching only decreases the utility of j if originally S_1 is the state based on which the smart contract is executed, and after switching it becomes S_0. However, this is impossible. Hence the observation is true. Note that if S_1 is not the true state, then user j always lies.

A common approach that prevents agents from lying in a game is to introduce payments. We consider the most straightforward way of adding the payment to the agent model, that is, if a user votes for a state that is different from the state based on which the smart contract is executed, he/she will be penalized, i.e., he/she will be fined a certain amount of coins.

Adding payment might prevent some users from lying on execution outcome. Specifically, if the number of extra coins that a user gets by outputting wrong outcome or lying is less than the penalty, he/she may choose to vote for the true state. However, it is still possible that users are lying no matter how large the penalty is. Consider the following scenario: The true state is S_0. There are users who strictly prefer S_1 than S_0. Let U be the set of them and suppose $\sum_{j \in U} w_j \geq \alpha W$. Focusing on users in U, there are two Nash equilibria, every

user in U voting for S_0 or every user in U voting for S_1. Consider an arbitrary user $j \in U$. When making his/her own decision, user j guess the decisions of other players. If j is optimistic and assumes every other player in U are voting for S_1, he/she will vote for S_1, otherwise if he/she is pessimistic and assumes every other player in U are voting for S_0, he/she will vote for S_0. In such a scenario, users may still lie. Furthermore, we have the following claim.

Theorem 2. *In the agent model with penalty, if j is superrational and knows that $\sum_{j\in U} w_j \geq \alpha W$, then no matter how high the penalty is, j will always lie.*

We provide the definition of superrationality as follows.

Definition 3 ([6]). *A player (agent) is called superrational if he/she has perfect rationality (and thus maximize his/her own utility), assumes that all other players are superrational, and that a superrational player will always come up with the same strategy as any other superrational player when facing the same problem.*

We remark that, superrationality is also called renormalized rationality in literature. According to the definition, if j is superrational, then he/she assumes that any other user in U would behave in the same way as he/she does, in this case, he/she will always vote for S_1, hence Theorem 2 is true.

Our above arguments show that, in general, introducing payment does not prevent users from lying. There exist scenarios in which users lie regardless of how high the penalty is. However, superrationality or rationality may not apply to real world application scenarios. As we have discussed, the incentive of lying relies crucially on a user's belief in certain behaviors of others. Specifically, he/she believes that other users are all rational. However, rationality itself is one of the most debatable issues in game theory in the sense that it seems to contradict a lot of laboratory experiments, which suggests that people often fail to conform to some of the basic assumptions of rationality. The "Centipede Game" , which was constructed by Rosenthal [15] in 1982, is one of the most well-known examples that illustrate such a phenomenon.

The centipede game is carried out between two players, say, A and B in a fixed number of rounds which is known to both players. Initially both A and B own 1 coin. At the beginning of round i, let a_i and b_i be the number of coins owned by A and B respectively. If i is odd, A makes the decision of yes or no, otherwise, B makes the decision. If A or B decides on yes, then the game moves to round $i+1$, $a_{i+1} = a_i+1$, $b_{i+1} = b_i+1$. If A or B decides on no, then the game stops. If it is A that decides on no (i.e., i is odd), then $a_{i+1} = a_i+2$, $b_{i+1} = b_i-1$. Otherwise it is B that decides on no, then $a_{i+1} = a_i - 1$, $b_{i+1} = b_i + 2$.

Assuming that A is rational and he/she believes the rationality of B, then A will decide on no at round 1 and the centipede game ends at the beginning. The reasoning is that at the last round regardless of whose turn it is, the decision will be no. Therefore, at the second to last round the opponent will decide no to make sure that the number of his/her coins does not decrease. Iteratively carrying out this argument we get the conclusion. However, this does not coincide

with the experiment results. For example, McKelvey and Palfrey [10] reported that only 15% of the players chose to end the game at the beginning in the experiments they carried out. That means, in most of these experiments, people do exhibit behaviors that contradict the traditional rationality assumptions in game theory. More experimental results and discussions on the centipede game and irrationality could be found in [11, 20].

The experimental results suggest that people often do not have fully trust in the rationality of the others. Notice that even if player A has perfect rationality, however, if he/she does not believe in the rationality of B, then A may still choose to continue the centipede game. Users involved in a smart contract may encounter a similar situation. Consider user $j \in U$, whether j votes for S_1 or not depends on his/her belief in the other users. Following the studies on irrationality in centipede game [2], we define the parameter $\tau_j(k)$, which indicates user j's belief in a certain behavior of user k, that is, user j believes that with probability $\tau_j(k)$, user k will vote for S_1, and with probability $1-\tau_j(k)$, user k will vote for S_0. Based on such assumptions, user j's decision is based on the following.

For $k \neq j$, we define X_k as a 0-1 random variable such that:

$$Pr(X_k = 1) = \tau_j(k), \quad Pr(X_k = 0) = 1 - \tau_j(k).$$

Suppose user j votes for S_1, then based on j's belief, the probability that the smart contract is executed based on S_1 is $Pr(\sum_{k\neq j} X_k + w_j \geq \alpha W)$. Let p_j be the penalty if the smart contract is executed based on S_0, then the expected reward of j by lying (voting for S_1) is

$$\begin{aligned} & z_j^1 Pr(\sum_{k\neq j} w_j X_j \geq \alpha W - w_j) - p_j(1 - Pr(\sum_{k\neq j} X_j \geq \alpha W - w_j)) \\ = & (z_j^1 + p_j) Pr(\sum_{k\neq j} w_j X_j \geq \alpha W - w_j) - p_j \end{aligned}$$

The expected reward of j by telling the truth is

$$z_j^0 Pr(\sum_{k\neq j} w_j(1 - X_j) \geq \alpha W - w_j) = z_j^0 Pr(\sum_{k\neq j} w_j X_j \leq (1-\alpha)W)$$

Therefore, as long as

$$z_j^0 Pr(\sum_{k\neq j} w_j X_j \leq (1-\alpha)W) \geq (z_j^1 + p_j) Pr(\sum_{k\neq j} w_j X_j \geq \alpha W - w_j) - p_j,$$

is true, the rational user j will not lie. This means, if j does not fully believe in the rationality of other users, then sufficient penalty can prevent j from lying. Overall, the following is true:

Theorem 3. *In the agent model with penalty, if a user does not fully believe in the rationality of others, then a sufficient penalty can prevent him/her from outputting incorrect contract execution outcome or lying.*

4 Implementation of Contract Execution with Penalty

Penalty plays a central role in the agent model of smart contract execution as shown in the previous section's analysis. We discuss the enforcement of penalty in this section.

There are several strategies to eliminate disagreement in blockchain branches. These strategies are also used to determine smart contract execution results when there is disagreement. Common rules include longest-chain which is used by Bitcoin [12], and GHOST which is used by Ethereum [17]. No matter what strategy is used, we add following functions to support penalty in a decentralized smart contract system:

- Recording users' choices. Existing blockchain systems usually records only one identity for each block and ignores supporters of the block. Recording supporters is necessary for implementing penalty schemes. When a user accepts a block, he/she should generate a signature of the block and broadcast it to the network. Therefore, everyone can track users' choices of the smart contract execution outcome;
- Distribution of penalty. When a group of users supporting the wrong result need to be penalized, users supporting the correct result can submit a penalty request to the blockchain. The collected fine is distributed to them.

5 Related Work

We provide a brief overview on blockchain based smart contract and game theory studies on these systems.

Ethereum is the most popular smart contract system [1]. It is based on proof-of-work, but is planning to move to proof-of-stake. Luu et.al. proposed a formal method to analyze Ethereum smart contracts to detect potential vulnerabilities [9].

The consequence of decentralization is subtle. Garay [5] and Pass et al. [14] showed that, several important security properties defined in the work of Nakamoto [12] are true, given the assumption that the majority of mining power in the Bitcoin system is controlled by the honest miners. Without such an assumption, however, security is not guaranteed. However, the assumption itself is questionable. For example, in 2014, the mining pool GHash.io exceeded 50% of the computational power in Bitcoin [3]. Thus, it becomes important to understand the behavior of users that participate in the system and study mechanisms that would motivate them to behave in an honest way.

There are a series of studies focusing on game theory aspects of users involved in mining. From a game theory perspective, Eyal and Sirer [4] showed that even a majority of honest miners is not enough to guarantee the security of the Bitcoin protocol. Sapirshtein et al. [16] and Kiayias et al. [7] studies mining as a game in Bitcoin and analyzes the best strategy of users.

6 Conclusion and Future Work

In this paper, we establish an agent based framework to model smart contract execution over a decentralized network of nodes/participants using blockchain and public ledger. In contrast to the commonly accepted assumption that smart contract execution outcome accepted by the majority can be trusted, agent based model of smart contract execution assumes that nodes may have incentive to manipulate or lie on outcome of contract execution in return for personal benefits or financial gains even they are not directly involved in a contract. We observe that users who are directly or indirectly involved in a smart contract may strategically take actions to manipulate smart contract execution outcome (e.g., produce or accept outcome that favors their own interests). In accordance with agent based model, we discuss the possibility of preventing users from engaging in bad behaviors in terms of contract execution or lying on contract outcome. We provide negative results for general smart contract execution models. We also show that if penalty is introduced in contract execution and assume that users are not fully confident in the rationality of other participants, then it is plausible to prevent users from lying on outcome or manipulating result of contract execution. Furthermore, we believe that, irrationality is an important subject that would contribute to better understanding of user behaviors in a decentralized cryptocurrency or smart contract system. A systematic investigation of irrationality in the context of smart contract execution and consensus is an important open problem. Another interesting open problem is whether it is possible to use other mechanisms, rather than financial penalty, to prevent users from lying on contract outcome when it favors them the most.

References

1. Buterin, V.: A next-generation smart contract and decentralized application platform. white paper (2014)
2. Dunbar, G., Tu, J., Wang, R., Wang, X., et al.: Rationalizing irrational beliefs. Queens Economics (2006)
3. Duong, T., Fan, L., Zhou, H.S.: 2-hop blockchain: Combining proof-of-work and proof-of-stake securely (2016)
4. Eyal, I., Sirer, E.G.: Majority is not enough: Bitcoin mining is vulnerable. In: International Conference on Financial Cryptography and Data Security. pp. 436–454. Springer (2014)
5. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: Analysis and applications. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 281–310. Springer (2015)
6. Hofstadter, D.R.: Dilemmas for superrational thinkers, leading up to a luring lottery. Scientific American 6, 267–275 (1983)
7. Kiayias, A., Koutsoupias, E., Kyropoulou, M., Tselekounis, Y.: Blockchain mining games. In: Proceedings of the 2016 ACM Conference on Economics and Computation. pp. 365–382. ACM (2016)
8. Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., Rosenschein, J.S.: Bitcoin mining pools: A cooperative game theoretic analysis. In: Proceedings of the

2015 International Conference on Autonomous Agents and Multiagent Systems. pp. 919–927. International Foundation for Autonomous Agents and Multiagent Systems (2015)
9. Luu, L., Chu, D.H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 254–269. ACM (2016)
10. McKelvey, R.D., Palfrey, T.R.: An experimental study of the centipede game. Econometrica: Journal of the Econometric Society pp. 803–836 (1992)
11. McKelvey, R.D., Palfrey, T.R.: Quantal response equilibria for extensive form games. Experimental economics 1(1), 9–41 (1998)
12. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
13. Osborne, M.J., Rubinstein, A.: A course in game theory. MIT press (1994)
14. Pass, R., Seeman, L., Shelat, A.: Analysis of the blockchain protocol in asynchronous networks. IACR Cryptology ePrint Archive 2016, 454 (2016)
15. Rosenthal, R.W.: Games of perfect information, predatory pricing and the chain-store paradox. Journal of Economic theory 25(1), 92–100 (1981)
16. Sapirshtein, A., Sompolinsky, Y., Zohar, A.: Optimal selfish mining strategies in bitcoin. arXiv preprint arXiv:1507.06183 (2015)
17. Sompolinsky, Y., Zohar, A.: Secure high-rate transaction processing in bitcoin. In: International Conference on Financial Cryptography and Data Security. pp. 507–527. Springer (2015)
18. Szabo, N.: Formalizing and securing relationships on public networks. First Monday 2(9) (1997)
19. Tapscott, D., Tapscott, A.: Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World. Penguin (2016)
20. Zauner, K.G.: A payoff uncertainty explanation of results in experimental centipede games. Games and Economic Behavior 26(1), 157–185 (1999)