



Article

Toward Prevention of Parasite Chain Attack in IOTA Blockchain Networks by Using Evolutionary Game Model

Yinfeng Chen ^{1,2} , Yu Guo ¹ , Yaofei Wang ^{1,2}  and Rongfang Bie ^{1,*} 

¹ School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China; chenyf@mail.bnu.edu.cn or 201931210002@mail.bnu.edu.cn (Y.C.); yuguo@bnu.edu.cn (Y.G.); yfwang@mail.bnu.edu.cn (Y.W.)

² School of Computer Information Management, Inner Mongolia University of Finance and Economics, Hohhot 010070, China

* Correspondence: rfbie@bnu.edu.cn

Abstract: IOTA is a new cryptocurrency system designed for the Internet of Things based on directed an acyclic graph structure. It has the advantages of supporting high concurrency, scalability, and zero transaction fees; however, due to the particularity of the directed acyclic graph structure, IOTA faces more complex security threats than the sequence blockchain, in which a parasite chain attack is a common double-spending attack. In this work, we propose a scheme that can effectively prevent parasite chain attacks to improve the security of the IOTA ledger. Our main idea is to analyze the behavior strategies of IOTA nodes based on evolutionary game theory and determine the key factors affecting the parasite chain attack and the restrictive relationship between them. Based on the above research, we provide a solution to resist the parasite chain attack and further prove the effectiveness of the scheme by numerical simulation. Finally, we propose the parasite chain attack prevention algorithms based on price splitting to effectively prevent the formation of the parasite chain.



Citation: Chen, Y.; Guo, Y.; Wang, Y.; Bie, R. Toward Prevention of Parasite Chain Attack in IOTA Blockchain Networks by Using Evolutionary Game Model. *Mathematics* **2022**, *10*, 1108. <https://doi.org/10.3390/math10071108>

Academic Editors: Ximeng Liu, Yinbin Miao and Zuobin Ying

Received: 17 February 2022

Accepted: 25 March 2022

Published: 30 March 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: IOTA; parasite chain attack; the tangle; evolutionary game; security

MSC: 68Q01; 68W01; 68U01; 68R01; 68V99

1. Introduction

The rapid development of blockchain technology has accelerated the popularization of the Internet of Things (IoT). Along with the increasing scale of IoT, the blockchain technology based on directed acyclic graph (DAG) structure has attracted more and more attention in IoT with its high concurrency and high scalability. At present, the most representative one is IOTA (Internet of Things application) [1–3].

IOTA is a revolutionary new cryptocurrency system specially designed for IoT. It overcomes the inefficiency in the existing blockchain design by replacing the sequence distributed ledger with the distributed ledger based on the DAG structure, named the Tangle, and creates a new method for reaching the consensus of the decentralized P2P system. IOTA realizes zero transaction fees, high concurrency, and unlimited scalability to complete the free transaction between machines and provide the underlying public chain technology for IoT. IoT architecture based on IOTA Tangle is shown in Figure 1. The left part in the figure is the IoT device layer (composed of sensors, bar codes and radio frequency electronic tags, etc.), which is responsible for receiving user requests and collecting information in real-time and transmitting them to the client layer (composed of IOTA wallets or applications running on computers or smartphones). After a transaction is packaged and generated by the client, it is sent to the IOTA node (composed of the IoT device with a node software to read and write access to the Tangle) for processing. If the transaction is valid and follows the protocol standards, the IOTA node first updates the

local ledger (the right part in the figure) after verification and then broadcasts the updated ledger to other IOTA nodes through P2P protocol to complete the consensus process [4].

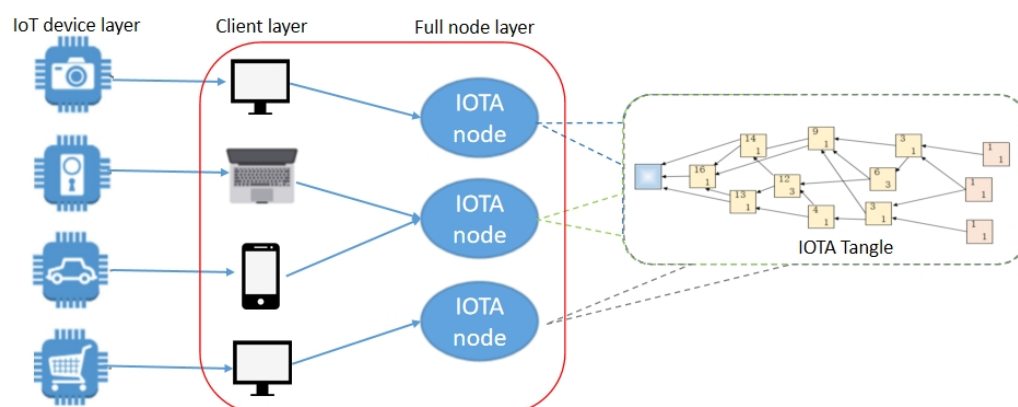


Figure 1. IoT Architecture based on IOTA Tangle.

However, due to the particularity of the DAG structure, the security of IOTA will face major challenges. IOTA is subject to a variety of attacks [1], among which the parasite chain (PC) attack is the most common. If the attack is successful, Tangle's historical records will be tampered with, and the attacker will realize double-spending. In this paper, we focus on the PC attack. Similar to selfish mining by nodes in the single-chain architecture [5–8], malicious nodes privately create parasite chains and broadcast them when the opportunities are ripe, in order to replace the corresponding legal branches in the main tangle; however, so far, few people look for the cause of PC attacks from IOTA nodes themselves. There is an obvious game relationship between IOTA nodes. As a player, each node will choose its strategy to maximize its utility when given the strategies of other players. It should be noted that the cost of launching a PC attack is an important part of building a game revenue matrix; a survey found that there is no expression to calculate the cost of the PC attack directly now [9–13]. In addition, in the actual scenario of IoT, the node's malicious behavior is studied through the classical game theory based on the assumption of "complete rationality" [14–17], but the static results cannot meet the actual needs of IOTA, nor can they reflect the dynamic change of IOTA node's strategic behavior and the evolution process that eventually tends to be stable; therefore, it is necessary to analyze PC attacks using game theory, but it is particularly important to find a game method more suitable for IOTA scenarios. In light of the above observations and given literature [1], we study what circumstances IOTA nodes actively launch PC attacks; therefore, we should design a scheme that can effectively detect and prevent parasite chain attacks.

It is challenging to achieve the above goal in that: (1) Nodes can join or exit the IOTA network at any time. We must ensure that enough nodes can synchronize the Tangle, but it is difficult to determine the number of effective working nodes. (2) The distributed ledger based on the DAG structure solves the problems of high concurrency and high scalability of IoT and increases the growth randomness of the ledger with time. The randomness raises the complexity of the cost of PC attacks launched by computing nodes. (3) Affected by the dynamic change of the Tangle, the initial behavior choices of nodes are not necessarily optimal, so it is hard to analyze and obtain the final evolutionary stability strategies of nodes.

In this paper, for the first time, we solve the problem of PC attacks in IOTA blockchain networks by introducing epidemic dynamics models and evolutionary game theory. With the help of the improved epidemic model, our proposed scheme can determine the number of nodes that synchronize the Tangle, and provide a guarantee for the normal operation of IOTA by monitoring the change of the number of nodes in real-time. Evolutionary game theory is a combination of game theory analysis and dynamic evolutionary process analysis. It studies how bounded rational individuals evolve in dynamic processes, how

to learn adaptively in repeated games, and choose the optimal strategy to maximize their interests [18]. In this study, the dynamic evolution process of IOTA nodes' behavior strategies was analyzed through evolutionary game, and the key factors inducing nodes to launch PC attacks were found. To realize this scheme, the main contributions of this paper are as follows:

- (1) We introduce an improved epidemic model TG_SEI. IOTA can effectively synchronize the number of nodes in the Tangle estimated by using the TG_SEI model, which is not only an important indicator to measure whether IOTA is running normally, but also an important part of the PC attack cost.
- (2) We propose a computational expression for the PC attack cost. The transaction involves multiple key links, from creation to issuance. If a malicious node wants to successfully launch a PC attack, an additional cost must be paid. We used the method of dividing the time according to the key points of events to complete the cost accounting of each stage.
- (3) We designed the parasite chain attack prevention algorithms based on price splitting. Using evolutionary game theory to analyze the behaviors of IOTA nodes, it was found that the commodity prices are the main factor that triggers PC attacks. Moreover, we predicted the concentrated time slot of PC attacks, which makes it more efficient to resist PC attacks.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents the background. Section 4 introduces the improved epidemic model TG_SEI. Section 5 gives the details of the evolution game analysis of nodes in IOTA, followed by the proposed algorithm in Section 6. Section 7 concludes the whole paper.

2. Related Work

The emergence of blockchain technology has accelerated the development of decentralization, privacy protection, and encrypted search of IoT [19–23], especially in crowdsensing systems [24], fog computing [25,26], privacy protection [27–29], and crowdsourcing [30–32]. On this basis, the blockchain system based on DAG provides a guarantee for the high concurrency, high scalability, and zero handling fee of IoT. The most representative is IOTA.

After the IOTA project was launched in 2015, Serguei Popov [1] explained the working principle of the Tangle in the relevant white paper, proposed an MCMC algorithm to provide an attachment strategy for new transactions arriving, and finally listed a variety of possible attack scenarios. Among them, the PC attack, as a common double-spending attack, has attracted extensive attention.

2.1. PC Attack

In [1], the authors first described the formation and attack principle of the parasite chain. Cai, D [9] pointed out that the coordinator still played a major role in IOTA. Once removed, it would face security problems caused by parasite chain attacks. In the parasite chain attack scenario, Yixin Li et al. [10] used the Markov chain model to describe the consistency process behavior of the DAG ledger under dynamic load and tested the probability of a successful attack under different network load modes. Philip Staupe [11] studied the method to reduce the risk of a double-spending attack by analyzing the probability absorbed by the parasite chain in the MCMC random walk. A. Cullen et al. [12] analyzed the effectiveness of the Markov chain Monte Carlo (MCMC) algorithm by using a matrix model and proposed an extended MCMC algorithm to improve the resistance of the distributed ledger to these attacks. Andreas Penzkofer et al. [13] proposed a detection mechanism for parasite chain attacks. Honest nodes improved the tip selection algorithm by detecting the structure of the parasite chain to prevent the parasite chain from successfully launching attacks. The above studies fully show that the parasite chain attack is one of the major security risks of IOTA and show that further research on the parasite chain attack has practical significance. Observing these studies, it is found that no one has analyzed the impact of node behavior strategies on PC attacks from the perspective of the IOTA nodes themselves.

2.2. Blockchain and Game Theory

At present, game theory is mainly used to analyze the mining behaviors of nodes in the blockchain, the computing power competition between mining pools, and the blockchain consensus and incentive mechanisms. Liu Z et al. [14] summarized the application of game theory in blockchain and pointed out that game theory was a mathematical model for studying the strategic interaction between rational decision makers, which was naturally applicable to the decision making of all consensus nodes in the blockchain network. Chang-bing Tang et al. [15] understood and analyzed the PoW consensus algorithm from the perspective of game theory, providing new ideas and methods for further designing consensus algorithms based on game theory. Lihua Song et al. [16] analyzed some problems in the design of the bitcoin incentive mechanism and used the idea of game theory to design an anti-collusion smart contract for clients in cloud computing. Shi H et al. [17] gave the mining pool the power to unilaterally control the miners' income using the zero-determinant theory and stimulated the miners' cooperation through the proposed zero determinant incentive mechanism. Xuan S et al. [33] proposed a data-sharing incentive model of smart contract blockchain based on evolutionary game theory. Their model was proposed to solve the challenges of establishing mutual trust and improving user participation in data sharing. According to the increasing demand for blockchain scalability and sustainability in various fields, Shashank Motepalli et al. [34] proposed a reward mechanism framework. Then, they further analyzed how participants' behavior evolved with the reward mechanism by using evolutionary game theory.

The above research shows that it is feasible and a research hotspot to analyze the behavior of blockchain nodes through game theory, but IOTA, as a blockchain based on the DAG structure, is rarely involved in game methods. Serguei Popov et al. [35] have proved that there is a Nash equilibrium in IOTA, but it is necessary to conduct in-depth analysis on the malicious behavior of nodes in IOTA to resist attacks.

However, the current literature has not paid too much attention to the impact of IOTA nodes' own behaviors on IOTA security, and no one has analyzed IOTA nodes' behavior strategies through evolutionary game theory. Under the premise of "bounded rationality" of participants, we put forward the cost calculation method of launching a PC attack, construct the payoff matrix to analyze the dynamic behaviors of IOTA nodes, calculate the evolutionary stable strategy, and finally find an effective algorithm to prevent PC attacks.

3. Background

3.1. IOTA

IOTA does not charge transaction fees, and each node maintains the update of the ledger by contributing its own computing power to confirm the two existing transactions in the Tangle. Figure 2a shows the DAG structure of the Tangle. The rectangle represents a transaction. On the left is the genesis transaction, and the unconfirmed transactions on the right are called tips. To avoid malicious competition between nodes caused by not charging transaction fees, the concept of weight is introduced into the IOTA system. Each transaction has weight, which is divided into cumulative weight and own weight (shown in Figure 2b. The number in the upper left corner of transaction A is cumulative weight, and the number in the lower right corner is own weight). The transaction with a large cumulative weight in the Tangle is more "important", and its own weight is proportional to the amount of work that the issuing node invested into it. To ensure the effectiveness of transactions in the Tangle, we propose approval rate (AR) and tangle robustness level σ .

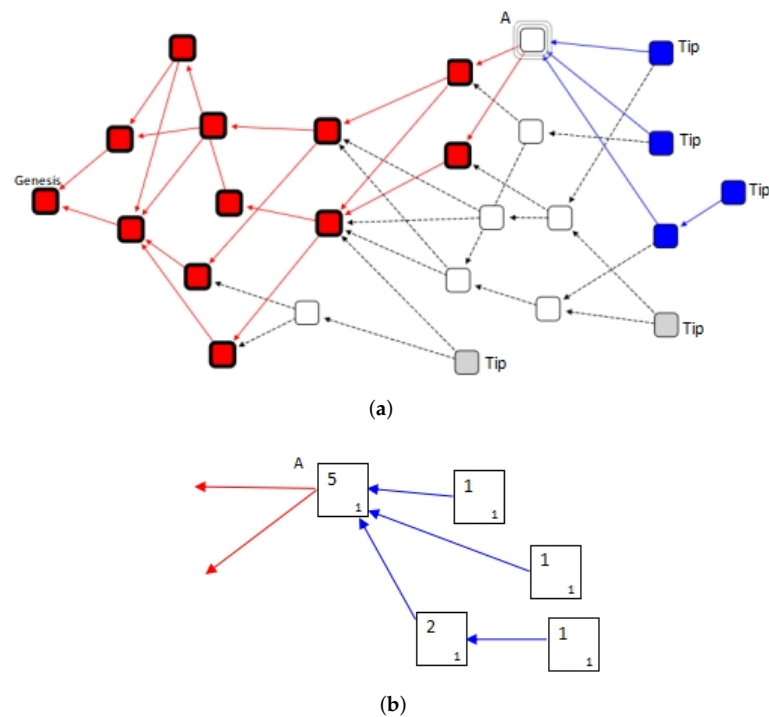


Figure 2. IOTA's distributed ledger—the Tangle diagram. (a) DAG structure of the Tangle. (b) Example of cumulative weight and own weight of transaction A.

Definition 1. Approval rate AR : The ratio of the number of tips directly or indirectly pointing to the transaction at a certain time to the total number of tips at that time. As shown in Figure 2, the AR of transaction A is 0.6. The greater the AR , the greater the possibility that the transaction cannot be tampered with.

Definition 2. Tangle robustness level σ : When AR meets certain conditions, additional cumulative weight is required to ensure the confirmation of existing transactions to offset the difference between the cumulative weight achieved by malicious nodes and honest nodes.

3.2. Epidemic Model

The epidemic model can reflect the dynamic characteristics of infectious diseases. Through the qualitative and quantitative analysis and numerical simulation of the epidemic model, we can reveal the epidemic law of diseases and predict their change trend, so as to provide a theoretical and quantitative basis for disease prevention and control. The iterative form of the SI model is as follows:

$$\begin{cases} \frac{dS(t)}{dt} = \alpha M - \beta S(t)I(t) - \alpha S(t) \\ \frac{dI(t)}{dt} = \beta S(t)I(t) - \alpha I(t) \end{cases}, \quad (1)$$

where $S(t)$ represents the susceptible population at time t , $I(t)$ represents the infected population at time t , β represents the transmission coefficient of S and I , M represents the total population at time t , and α represents the natural mortality of S and I at time t . This paper mainly improves the SI model. The information transmission between nodes has a similar dynamic evolution process with virus infection, so the mathematical model based on the epidemic is also suitable for the field of network information transmission [36,37]. Based on the dynamic evolution model of infectious disease transmission, we put forward a calculation method for estimating the number of nodes in the actual synchronous Tangle.

3.3. Evolutionary Game Theory

In evolutionary game theory, evolutionary stable strategy (ESS) and replication dynamics are two core concepts. ESS refers to that, in the process of the game, due to the limited rationality of both sides in the game, the game side cannot find the optimal strategy and the optimal equilibrium point at the beginning; therefore, the game player needs to constantly learn in the process of the game. If the player has made strategic mistakes, they will gradually correct them, and constantly imitate and improve towards the most favorable strategies for themselves and others in the past. After a period of imitation and error correction, all players will tend to a stable strategy. Replication dynamics is actually a dynamic differential equation that describes the frequency of a specific strategy adopted in a population, which can be expressed by the following formula:

$$\frac{dx_i}{dt} = x_i[u_{s_i}(x) - u(x, x)], \quad (2)$$

where x_i is the proportion or probability of adopting pure strategy s_i in a population, which represents the fitness when adopting the pure strategy and the average fitness.

4. Improved Epidemic Model TG_SEI

The number of effective nodes in the synchronous ledger (hereinafter referred to as the number of synchronous nodes) reaches a certain threshold (below the threshold, IOTA cannot operate normally. The threshold is set according to the actual situation of IOTA), that can ensure the normal operation of IOTA. Monitoring the number of synchronization nodes in IOTA regularly can reflect the service level of the network to a certain extent. If the synchronization node is seriously missing, the service quality of IOTA will be reduced; therefore, regular monitoring, timely warning, and troubleshooting must be carried out. In addition, the number of synchronization nodes is also an important part of calculating the attack cost; therefore, it is very important to find a method to solve the number of synchronization nodes. The process of the IOTA node synchronizing to the Tangle is very similar to the spread of some viruses in the infectious disease model [38–40]; therefore, we adopt the improved epidemic model TG_SEI to estimate the number of synchronization nodes at any time period. IOTA nodes have three statuses: invalid synchronization status S, the Tangle synchronization delay status E, and effective synchronization status I. Nodes in the S status have not synchronized the Tangle yet. The nodes in the E status synchronize the Tangle but fail to forward it to other nodes in time due to delay. The delay is generally related to the actual network delay threshold. For example, if it is greater than 50 ms, the E status will appear. Nodes in the I status synchronize the Tangle and immediately forward it to other nodes. Assuming that the nodes in the S status connect the nodes in the I status, the Tangle is synchronized.

In the beginning, only one node G is in the I status (that is, the node issuing the Genesis transaction) while the other IOTA nodes are in the S status. After node G issues the transaction, it starts broadcasting the Tangle to the whole network. The nodes directly connecting to node G to synchronize the Tangle will change status S to status I or E. Due to network delay, some of the nodes that synchronized the Tangle are temporarily in the E status. When the delay is alleviated, the nodes in the E status will continue to forward the Tangle to other nodes. With the continuous spread of the Tangle ledger, the connectivity scale between nodes will gradually expand until the whole IOTA network. Most of the nodes that finally synchronized the Tangle are in the I status. In this spreading process, due to natural disasters, equipment failures, crashes, and other factors, a few nodes will not keep the ledger synchronized with node G's and they are removed from the network.

Suppose that at any time t , $S(t)$ denotes the number of nodes that have not synchronized the Tangle (also known as the number of invalid synchronization nodes). The movement of invalid synchronization nodes is random. During this period, very few nodes may not work or crash and be removed from the network. $E(t)$ denotes the number of delayed nodes synchronizing the Tangle. Due to the uncertainty in the network link, there

will be a certain delay in forwarding the Tangle. During this period, very few nodes may not work or crash and be removed from the network. $I(t)$ denotes the number of nodes that fully synchronize to the Tangle and forward it (also known as the number of effective synchronization nodes). During this period, very few nodes may not work or crash and be removed from the network.

Let M denote the total number of nodes in the IOTA at time t . To simplify the calculation, the rate at which new nodes join or exit the IOTA is α ; therefore, $M = S(t) + E(t) + I(t)$ is constant. β represents the average spreading rate of the IOTA Tangle, which is related to the average degree of the network. δ represents the average delay rate of the synchronizing Tangle and meets $0 < \delta < 1$. δ is related to the actual situation of the network. The greater the network delay, the larger the value of δ . γ indicates the conversion rate from status E to status I. Most nodes will synchronize the Tangle after delay. The improved epidemic model TG_SEI iteration form is as follows,

$$\begin{cases} \frac{dS(t)}{dt} = \alpha M - \beta S(t)I(t) - \alpha S(t) \\ \frac{dE(t)}{dt} = \beta \delta S(t)I(t) - (\gamma + \alpha)E(t) \\ \frac{dI(t)}{dt} = \beta(1 - \delta)S(t)I(t) + \gamma E(t) - \alpha I(t) \end{cases} \quad (3)$$

The evolution process of the synchronous nodes' number is the same as that of $I(t)$, that is, the final result of $I(t)$ evolution is the number of synchronous nodes in IOTA denoted as X ,

$$\begin{cases} \frac{dI(t)}{dt} = \beta(1 - \delta)S(t)I(t) + \gamma E(t) - \alpha I(t) \\ I(0) = 1 \end{cases} \quad (4)$$

According to Equation (3), two groups of possible equilibrium points of the equations are obtained, which are $E_1^*(M, 0, 0)$ and $E_2^*\left(\frac{\alpha(\gamma + \alpha)}{\beta(\gamma + \alpha - \alpha\delta)}, \frac{\alpha\delta(M\beta(\gamma + \alpha - \alpha\delta) - \alpha(\gamma + \alpha))}{\beta(\gamma + \alpha - \alpha\delta)(\gamma + \alpha)}, \frac{M\beta(\gamma + \alpha - \alpha\delta) - \alpha(\gamma + \alpha)}{\beta(\gamma + \alpha)}\right)$. Because only when the basic reproduction number $R_0 > 1$, the Tangle of node G can be synchronized and forwarded by most other nodes; therefore, the equilibrium point E_2^* is the only asymptotically stable equilibrium point. When $t \rightarrow \infty$ and $M\beta(\gamma + \alpha - \alpha\delta) - \alpha(\gamma + \alpha) > 0$, $I(t) \rightarrow \frac{M\beta(\gamma + \alpha - \alpha\delta) - \alpha(\gamma + \alpha)}{\beta(\gamma + \alpha)}$, we can obtain

$$X = \left\lfloor \frac{M\beta(\gamma + \alpha - \alpha\delta) - \alpha(\gamma + \alpha)}{\beta(\gamma + \alpha)} \right\rfloor, \quad (5)$$

where parameters α , β , δ , and γ are greater than 0 and less than 1. For example, when $M = 10,000$, $\alpha = 0.005$, $\beta = 0.2$, $\delta = 0.8$, and $\gamma = 0.2$, the probability curve of the number of synchronization nodes with time evolution based on the TG_SEI model is shown in Figure 3. The red point line indicates the changing trend of the number of invalid synchronization nodes with time, which is decreasing. The blue dotted line indicates the changing trend of the number of effective synchronization nodes with time, which is increasing. The solid line indicates the changing trend of the number of synchronization delay nodes. After a certain delay, most of the nodes in E status will become nodes in I status.

To simplify the calculation, we set the number of nodes joining and exiting IOTA per unit time equal. In a real scenario, they may not be equal. The values of α , β , δ , and γ in the TG_SEI model can be obtained by (1) prediction of supervised learning model in machine learning or (2) analysis of the propagation mechanism and dynamics of the complex network [41,42].

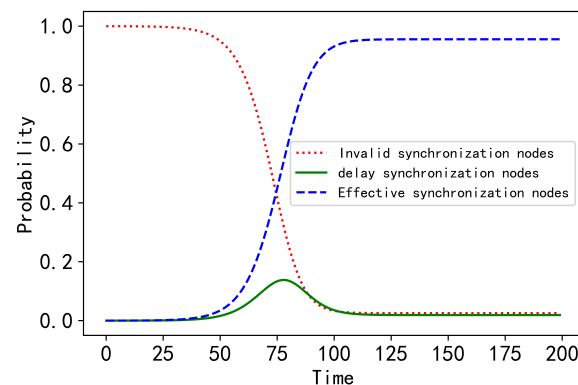


Figure 3. Probability curve of the number of synchronous nodes with time evolution based on the TG_SEI model.

5. Evolution Game Analysis of Nodes in IOTA

In IOTA, all nodes form a node group. Each node has an initial strategy about whether to choose a parasite chain attack. Nodes repeatedly randomly select other nodes from the group to play the game. In this process, nodes with a low payoff will change the strategy to imitate the high-payoff nodes, while low-payoff strategies will be gradually eliminated. After such continuous learning and adjustment, the node group will eventually reach an equilibrium state, which is that all nodes in the group will choose the ESS.

Creating transactions to issuing the transactions to the Tangle is a complex process. In this process, the Tangle is vulnerable to malicious attacks. One of the most-common attacks is PC attacks. In Figure 4, a parasite chain is “generated” under a transaction in the Tangle where the red site is a conflict transaction. The parasite chains formed in the actual scene have different shapes and sizes, and the attacker can freely choose the number of transfers of the PC and decide which transfers the attacker confirms; therefore, this paper makes some restrictions on the PC. We only study the simple PC because it does not affect the generality.

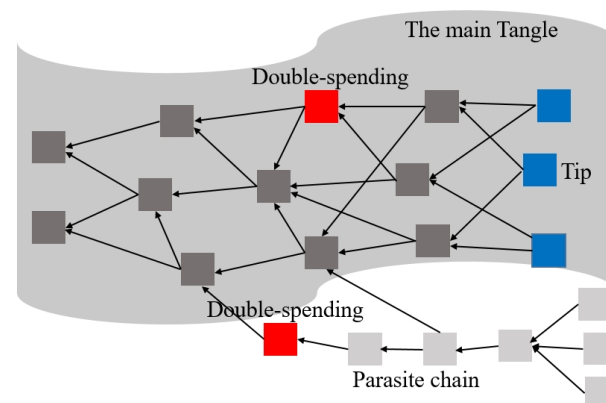


Figure 4. The Tangle with a parasite chain.

5.1. Research Hypothesis and Parameter Description

We make the following assumptions.

Assumption 1. IOTA nodes are bounded rationality, that is, they cannot find the optimal strategy at the beginning of the game. It needs to learn constantly in the process of the game.

Assumption 2. The computational power cost consumed by node attack is large enough to enable the successful completion of the attack; the computational power cost of two nodes is the same.

Assumption 3. Each node purchases the same commodity at the same price.

Assumption 4. The change of node payoff caused by the change of market price of the currency is not considered.

Since the payoff matrix of the repeated game model is closely related to the specific payoff parameters of each game participant, the parameters are described as shown in Table 1.

Table 1. Description of parameters.

Parameter	Description
i, j	Node number, representing node i and node j
S_i, S_j	Strategies adopted by node i and node j respectively, $S_i, S_j \in \{Attack, No\ attack\}$
$x(0 \leq x \leq 1)$	Probability of selecting attack strategy
C_1	Cost of node successfully launching PC attack
C_2	Cost of node not launching attack
U_i, U_j	Represent the expected returns of both parties respectively
Pri	Commodity price
w_{ini}	The node issues the initial weight of a transaction (recorded as tran1) in the main tangle
w_0	The merchant agrees to the cumulative weight threshold of the transaction (tran1)
t_0	The time when the node issues a transaction tran1 to the merchant in main tangle and also the starting time of the private PC chain
t_1	The time when the cumulative weight of transaction tran1 reaches the merchant weight threshold and the merchant accepts the transaction
t	The time after the node implements the PC attack, the cumulative weight of the double-spending transaction generated on the PC exceeds the cumulative weight time of the legal transaction in main tangle
S_h	The sum of cumulative weights of transaction tran1 in main Tangle at time t
S_m	The sum of cumulative weights corresponding to the double-spending transaction in the parasite chain at time t
λ	Average transaction arrival rate of the node no launching PC attacks (i.e., an honest node), $\lambda > 0$
μ	Average transaction arrival rate of nodes launching PC attacks (i.e., a malicious node), $\mu > 0$
w_h	The average weight of transactions generated by honest nodes
w_m	The average weight of transactions generated by malicious nodes

5.2. Transaction Number, Cumulative Weight, Time to Successfully Launch Parasite Chain Attack, and Its Cost Function

Let A be any node in IOTA. Node A may launch a parasite chain attack. If node A has a parasite chain at time t , before node A broadcasts the parasite chain to IOTA, the average rate of new transactions reaching the main Tangle is λ and the average rate of new transactions arriving at the parasite chain is μ .

- (1) If node A launches a parasite chain attack successfully, the number of transactions issued is

$$N_1 = (\lambda + \mu)(t_1 - t_0) + \mu(t - t_1). \quad (6)$$

If node A does not launch a parasite chain attack, the number of transactions issued is

$$N_2 = \lambda(t - t_0). \quad (7)$$

- (2) Cumulative weight

Suppose that node A issues a transaction tran1 to the main Tangle at time t_0 and waits for the merchant's confirmation. At the same time, node A starts to build a parasite chain privately and generates a transaction tran2. Node A transfers the money used to transaction tran1 to the corresponding account of transaction tran2 in the parasite chain.

That is, transaction tran1 and transaction tran2 have the same money but correspond to different accounts of node A . When the cumulative weight of tran1 reaches the merchant's weight threshold w_0 at t_1 , the merchant accepts the transaction and delivers the goods. Once the difference between the cumulative weights of transaction tran2 and transaction tran1 is greater than σ , that is, $S_m - S_h \geq \sigma$, then node A can successfully launch a parasite chain attack and realize double spending. σ is the Tangle robustness level, which is related to AR. σ reflects the difficulty of the node to implement the malicious attack successfully. For S_m and S_h , see below for details.

At time t_1 , the merchant accepts the cumulative weight threshold expression of transaction tran1 issued by node A in the main Tangle is

$$w_0 = w_{ini} + \lambda(t_1 - t_0)w_h. \quad (8)$$

At time t , the cumulative weight of transaction tran1 in the main Tangle is

$$\begin{aligned} S_h &= w_0 + (1 - p_m)\lambda(t - t_1)w_h \\ &= w_{ini} + \lambda(t_1 - t_0)w_h + (1 - p_m)\lambda(t - t_1)w_h. \end{aligned} \quad (9)$$

where P_m is the probability that the transaction is absorbed by the parasite chain after the honest node runs the MCMC algorithm [11], $0 < P_m < 1$. It is worth noting that after the parasite chain is broadcast to the whole network, the parasite chain can no longer reference the transactions in the main Tangle. Because the honest node will check the historical consistency and will not accept the double-spending transaction as a valid transaction. At time t , node A issues the cumulative weight of the corresponding double-spending transaction in the parasite chain as follows

$$S_m = w_{ini} + \mu(t - t_0)w_m + p_m\lambda(t - t_1)w_h. \quad (10)$$

(3) Time of successful parasite chain attack

$$S_m - S_h \geq \sigma. \quad (11)$$

Combined with Equations (8)–(10), the relationship expression of time t is

$$t \geq \frac{\sigma + (\mu w_m - \lambda w_h)t_0 + 2p_m\lambda w_h t_1}{\mu w_m - \lambda w_h + 2p_m\lambda w_h}. \quad (12)$$

Let $T_1 = \frac{\sigma + (\mu w_m - \lambda w_h)t_0 + 2p_m\lambda w_h t_1}{\mu w_m - \lambda w_h + 2p_m\lambda w_h}$; $t \geq T_1$ is obtained by simplification.

(4) Cost function

Assuming that the actual number of effective synchronization nodes in IOTA at time t is X and the number of malicious nodes is n , the cost function required for node A to successfully launch a parasite chain attack is expressed as

$$C_1 = [(\lambda + \mu)(t_1 - t_0) + \mu(t - t_1)]q_0 + (X - 1)\lambda(t_1 - t_0)q_1 + (n - 1)\mu(t - t_1)q_1. \quad (13)$$

If the node does not launch an attack, the cost function is

$$C_2 = \lambda(t - t_0)q_0 + (X - 1)\lambda(t_1 - t_0)q_1 + (X - n - 1)\lambda(t - t_1)q_1, \quad (14)$$

where q_0 is the average cost of generating and issuing a transaction for the node and q_1 is the average cost of verifying and disseminating an incoming transaction, and $q_0 > 0$, $q_1 > 0$.

5.3. Constructing Evolutionary Game Model and Results

The following two cases are discussed. One is that malicious nodes conspire to create the PC, and the attack cost is halved after malicious nodes cooperate. The other is that malicious nodes create PCs alone.

5.3.1. Nodes Conspire to Create the Parasite Chain

Analyze the payoff of node i and node j at time t to construct the payoff matrix of the evolutionary game model shown in Table 2.

Table 2. Payoff matrix.

	Node j	
	Attack(y)	No Attack($1 - y$)
	Attack(x)	No attack($1 - x$)
Node i	$Pri - C_1/2, Pri - C_1/2$	$Pri - C_1, -C_2$
	$-C_2, Pri - C_1$	$-C_2, -C_2$

For node i (Note: the analysis method of node j is the same as that of node i , which will not be repeated later), the expected payoff of adopting the “attack” strategy is

$$U_{i1} = x(Pri - C_1/2) + (1 - x)(Pri - C_1) = (C_1/2)x + Pri - C_1. \quad (15)$$

Take the “no attack” strategy and the expected payoff is

$$U_{i2} = -C_2. \quad (16)$$

The overall average expected payoff of node i is

$$\begin{aligned} U &= xU_{i1} + (1 - x)U_{i2} \\ &= (C_1/2)x^2 + x(Pri - C_1 + C_2) - C_2. \end{aligned} \quad (17)$$

From Equations (15)–(17), the replicator dynamics equation of the subject proportion of actors adopting the “attack” strategy is

$$\begin{aligned} F(x) &= dx/dt \\ &= x(U_{i1} - U) \\ &= x(1 - x)[(C_1/2)x + Pri - C_1 + C_2]. \end{aligned} \quad (18)$$

When $F(x) = 0$, we can obtain three stable states

$$x_1^* = 0, x_2^* = 1, x_3^* = \frac{2(C_1 - C_2 - Pri)}{C_1}. \quad (19)$$

According to the value of x_3^* , the ESS is discussed in three cases below.

(1) The ESS when $x_3^* \leq 0$

The condition of $x_3^* = \frac{2(C_1 - C_2 - Pri)}{C_1} \leq 0$ is

$$Pri \geq C_1 - C_2. \quad (20)$$

Because of $0 \leq x \leq 1$, the replicator dynamics at this time only have two stable states $x_1^* = 0$ and $x_2^* = 1$. The ESS x^* has the characteristics of resisting small interference. When the interference makes $x < x^*$, there is $F(x) = dx/dt > 0$. When the interference makes $x > x^*$, there is $F(x) = dx/dt < 0$. In the phase diagram of the replicator dynamics equation, the curve intersects the abscissa at several points. If the tangent slope at the intersection is negative, it is the ESS of the replicator dynamics in the evolutionary game;

therefore, according to the above conditions, $F'(0) \geq 0$ and $F'(1) < 0$, its ESS $x^* = 1$. The replicator dynamics equation phase diagram is shown in Figure 5. Nodes tend to launch parasite chain attacks.

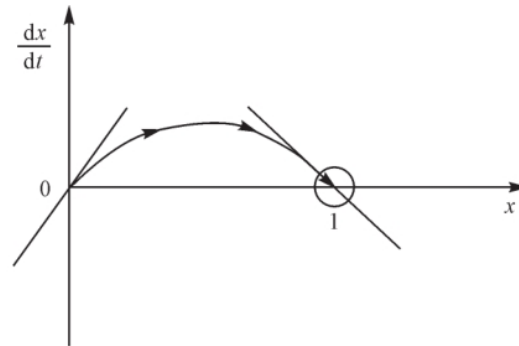


Figure 5. Replicator dynamics equation phase diagram when $x_3^* \leq 0$.

(2) The ESS when $x_3^* \geq 1$

The condition of $x_3^* = \frac{2(C_1 - C_2 - Pri)}{C_1} \geq 1$ is

$$Pri \leq C_1/2 - C_2. \quad (21)$$

At this time, the replicator dynamics only have two stable states $x_1^* = 0$ and $x_2^* = 1$. Since $F'(0) < 0$ and $F'(1) \geq 0$, its ESS is $x^* = 0$. The replicator dynamics equation phase diagram is shown in Figure 6. Nodes tend not to launch parasite chain attacks.

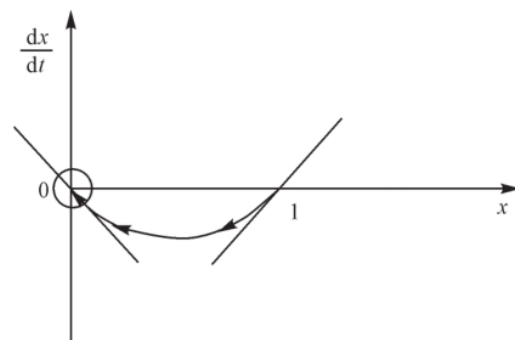


Figure 6. Replicator dynamics equation phase diagram when $x_3^* \geq 1$.

(3) The ESS when $0 < x_3^* < 1$

The condition of $0 < \frac{2(C_1 - C_2 - Pri)}{C_1} < 1$ is

$$C_1/2 - C_2 < Pri < C_1 - C_2. \quad (22)$$

At this time, the replicator dynamics have three stable states. Since $F'(0) < 0$ and $F'(1) < 0$, the corresponding replicator dynamics phase diagram is shown in Figure 7.

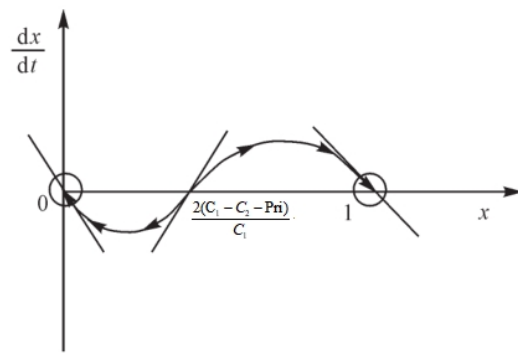


Figure 7. Replicator dynamics equation phase diagram when $0 < x_3^* < 1$.

When $0 < x_3^* < \frac{2(C_1 - C_2 - Pri)}{C_1}$, its ESS is $x^* = 0$.

When $\frac{2(C_1 - C_2 - Pri)}{C_1} < x_3^* < 1$, its ESS is $x^* = 1$.

The analysis shows that when the commodity price provided by the merchant is satisfied $Pri \geq C_1 - C_2$, that is, the commodity price is higher than the difference between the cost paid by the node to launch a parasite chain attack and the cost paid by not launching an attack, the node will choose to launch a parasite chain attack, because, compared with the original, the node will increase revenue and be profitable. When the commodity price provided by the merchant is satisfied $Pri \leq C_1/2 - C_2$, the node will not launch a parasite chain attack, because the attack cost is higher than the payoff, which will damage its interests and outweigh the loss. When the commodity price provided by the merchant is satisfied $C_1/2 - C_2 < Pri < C_1 - C_2$, if x_3^* is included in $(0, \frac{2(C_1 - C_2 - Pri)}{C_1})$, the node chooses not to attack because the probability of successfully launching a parasite chain attack is small. If x_3^* falls in $(\frac{2(C_1 - C_2 - Pri)}{C_1}, 1)$, the node will choose to launch an attack because the probability of successfully launching a parasite chain attack increases. Next, Matlab 2020a was used to simulate the above evolutionary game process.

- (1) Assuming $C_2 = 1$, $Pri = 2$, and C_1 takes 2, 4, and 6, respectively, the strategy selection of node i changes as shown in Figure 8.
- (2) Assuming $C_2 = 1$, $C_1 = 2$, and Pri takes 2, 0.5, and 0.1, respectively, the strategy selection of node i changes as shown in Figure 9.

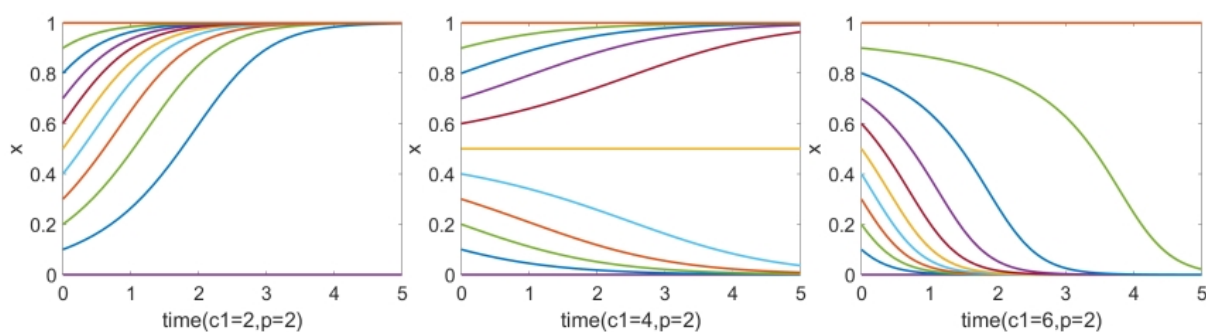


Figure 8. The evolution process of node i 's strategy selection when C_1 value increases.

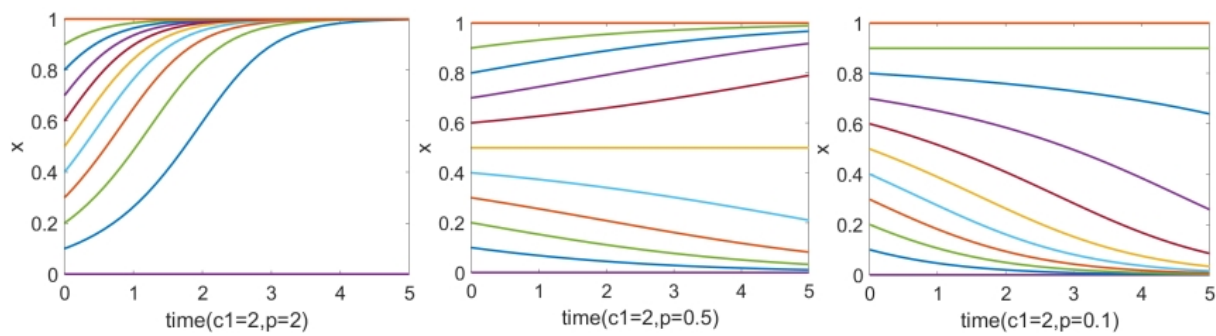


Figure 9. The evolution process of node i 's strategy selection when Pri value decreases.

Figure 8 shows the evolution trend of probability x with C_1 when Pri and C_2 remain unchanged. Figure 9 shows the evolution trend of probability x with Pri when C_1 and C_2 remain unchanged. The analysis shows that if other conditions are certain, increasing the attack cost will reduce the probability of nodes choosing parasite chain attacks. Similarly, other conditions are certain, the lower the commodity price, the lower the probability of nodes successfully launching parasite chain attacks.

5.3.2. Each Node Will Make Its Parasite Chain

If a node launches a PC attack, it needs to create a parasite chain alone. The payoff matrix of the evolutionary game model is shown in Table 3.

Table 3. Payoff matrix.

		Node j	
		Attack(y)	No Attack($1 - y$)
Node i	Attack(x)	$Pri - C_1, Pri - C_1$	$Pri - C_1, -C_2$
	No attack($1 - x$)	$-C_2, Pri - C_1$	$-C_2, -C_2$

Moreover, $Pri1$ and $Pri2$ represent the price of goods purchased by node i and node j , respectively. The analysis process is the same as Section 5.3.1, and two stable states $x_1^* = 0$ and $x_2^* = 1$ are obtained. The analysis results also reflect the relationship between commodity price and cost. For node i , when $Pri1 \geq C_1 - C_2$, the node will launch a parasite chain attack. When $Pri1 < C_1 - C_2$, the node will not launch an attack. The analysis method of node j is the same as that of node i and will not be repeated.

6. The Proposed Algorithms

It can be seen from the previous section that the lower the commodity price in the transaction, the less likely it is to be attacked by PC; therefore, we can effectively resist PC attacks by splitting large transaction prices into small ones. At the same time, it also further proves the applicability of micropayments in IOTA. With the continuous expansion of the scale of IoT, if we can further predict the concentrated time slot of many PC attacks and strengthen prevention, we can also effectively resist PC attacks.

6.1. Concentrated Time Slot of PC Attacks

According to the analysis results in Section 5, we can further determine the concentrated time slot for nodes to launch PC attacks. This section only focuses on the situation that satisfied $Pri \geq C_1 - C_2$. When ESS is 1, nodes will launch PC attacks to increase their interests; therefore, Equation (20) is deformed to

$$C_1 \leq Pri + C_2. \quad (23)$$

By substituting Equation (14) into (23), we obtain

$$C_1 \leq Pri + \lambda(t - t_0)q_0 + (X - 1)\lambda(t_1 - t_0)q_1 + (X - n - 1)\lambda(t - t_1)q_1, \quad (24)$$

According to Equation (24), the cost of an IOTA node launching a parasite chain attack successfully depends on many factors, such as commodity price, the number of IOTA synchronization nodes, the number of malicious nodes, the time issuing transaction, the time transacting with merchants, the time it takes for a parasite chain to attack successfully, the arrival rate of transactions, the average cost of issuing a transaction, the average verification and dissemination cost of a transaction, etc.

As shown in Figure 10, solid line a and dotted line b represent the cost threshold varying with time t and the cost paid by the node when successfully launching a parasite chain attack, respectively. Assuming $(\mu - \lambda)q_0 - (X\lambda - n\lambda - \lambda - n\mu + \mu)t_1q_1$, the time at the intersection of the two lines is

$$T_2 = \frac{Pri - (\lambda t_1 - \mu t_0)q_0 - (X\lambda - n\lambda - \lambda - n\mu + \mu)t_1q_1}{(\mu - \lambda)q_0 - (X\lambda - n\lambda - \lambda - n\mu + \mu)q_1}. \quad (25)$$

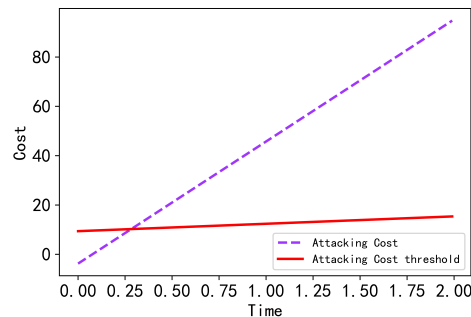


Figure 10. Cost function of launching a parasite chain attack diagram.

In Equation (12), when $T_1 > T_2$, $T_1 > 0$ and $T_2 > 0$, that is, $t \in [T_2 + t_1, T_1]$, the probability of a node successfully launching a parasite chain attack will increase. To prevent nodes from launching parasite chain attacks, it is important to strengthen prevention during this time slot as shown in Algorithm 1.

Algorithm 1 Algorithm for determining the concentrated time slot of PC attacks.

Input: Commodity price Pri ; The average transaction arrival rate of an honest node λ ; The average transaction arrival rate of a malicious node μ ; The average cost of generating and issuing a transaction q_0 ; The average cost of verifying and disseminating a transaction q_1 ; Actual number of synchronized ledger nodes X ; Number of malicious nodes n ; The time when the node issues a transaction to the merchant in the main Tangle t_0 ; The time when the cumulative weight of transaction reaches the merchant weight threshold w_0 and the merchant accepts the transaction t_1 .

Output: The concentrated time slot.

```

 $t \leftarrow t_0$ ;
Calculate the time points  $T_1$  and  $T_2$  according to Equations (12) and (25) respectively;
While  $t > 0$ 
    if  $t \geq T_1$  and  $t \leq T_2$ 
        Monitor the Tangle in real-time;
        If conflicting transactions are detected, run the tip select algorithm and only
        retain the legal branch with the largest cumulative weight;
    Endif
    Monitor the Tangle regularly;
    If conflicting transactions are detected, run the tip select algorithm and only retain
    the legal branch with the largest cumulative weight. Monitor in real-time for some time
    to ensure the complete elimination of conflicting transactions;

```

6.2. The Algorithm for Preventing PC Attacks Based on Price Splitting—APS

If the commodity price is low and the payment amount is small, IOTA nodes generally do not launch PC attacks to damage their interests, but prefer to work honestly. If the commodity price is high and the transaction amount is too large, it can be divided into multiple small amounts for payment to avoid PC attacks. There are many ways to split the price, which can be designed according to the needs of the actual IOTA network. For example, the simplest price halving method is used. Since the parasite chain can be generated jointly or independently, the corresponding algorithms are shown in Algorithms 2 and 3, respectively.

Algorithm 2 APS-conspiracy.

Input: Commodity price Pri ; PC attack cost C_1 ; Cost of not launching an attack C_2 .

Output: Splitting the price stored in $M[i]$ and the price split copies j .

```

 $M[] \leftarrow 0, i \leftarrow 0, j \leftarrow 0, x \leftarrow 0$ ;
M.append( $Pri$ );
While  $Pri \geq C_1 - C_2$ 
     $Pri = Pri/2$ ;
    M.append( $Pri$ );
     $i = i + 1$ ;
While  $C_1/2 - C_2 < Pri < C_1 - C_2$  and  $2(C_1 - C_2 - Pri)/C_1 < x < 1$ 
     $Pri = Pri/2$ ;
    M.append( $Pri$ );
     $i = i + 1$ ;
 $j = M[0]/M[i]$ ;
Return  $M[i]$  and  $j$ ;

```

Algorithm 3 APS-independence.

Input: Commodity price Pri ; PC attack cost C_1 ; Cost of not launching an attack C_2 .

Output: Splitting the price stored in $M[i]$ and the price split copies j .

$M[] \leftarrow 0, i \leftarrow 0, j \leftarrow 0, x \leftarrow 0;$

$M.append(Pri);$

While $Pri \geq C_1 - C_2$

$Pri = Pri/2;$

$M.append(Pri);$

$i = i + 1;$

$j = M[0]/M[i];$

Return $M[i]$ and j ;

In addition, IOTA can also add some incentive or punishment mechanisms to further prevent nodes from launching attacks. A reasonable incentive mechanism can make the node offset part of the work cost and reduce the probability of launching a PC attack, while the punishment mechanism can restrain the node from launching attacks via punishment.

7. Conclusions

In this paper, we proposed an effective scheme to prevent parasite attacks. First, we proposed a cost calculation method for parasite chain attacks. Then, the behavior strategies of the IOTA node launching attack were analyzed by evolutionary game theory, the key factors required to successfully launch parasite chain attack were studied, and the numerical simulation was carried out. Finally, the algorithms to prevent parasite chain attacks were proposed, which are based on price segmentation detection to further suppress the formation of parasitic chain attacks. Our proposed scheme is a new exploration and attempts to use the evolutionary game theory based on the bounded rationality of participants to analyze the behavior relationship between IOTA nodes. As future work, we plan to use existing techniques to implement the detection and prevention of PC attacks in IOTA.

Author Contributions: Conceptualization, Y.C.; methodology, Y.C. and R.B.; validation, Y.G. and Y.C.; writing—original draft, Y.C.; writing—review and editing, Y.C., Y.G. and Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research is sponsored by the National Natural Science Foundation of China under Grants 62102035, 62177007, 61571049, 71961022, the Fundamental Research Funds for the Central Universities under Grants 2020NTST32, and the Foreign Expert Programs of Ministry of Science and Technology grant number DL2021123002L.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Popov, S. The Tangle. Version 1.4.3. 2018. Available online: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvslqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf (accessed on 16 February 2022).
2. Silvano, W.F.; Marcelino, R. IOTA Tangle: A cryptocurrency to communicate Internet-of-Things data. *Future Gener. Comput. Syst.* **2020**, *112*, 307–319. [CrossRef]
3. Guo, F.; Xiao, X.; Hecker, A.; Dustdar, S. Characterizing IOTA Tangle with Empirical Data. In Proceedings of the GLOBECOM 2020–2021 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
4. Halgamuge, M.N. Optimization framework for best approver selection method (BASM) and best tip selection method (BTSM) for IOTA tangle network: Blockchain-enabled next generation industrial IoT. *Comput. Netw.* **2021**, *199*, 108418. [CrossRef]
5. Eyal, I.; Sirer, E.G. Majority Is Not Enough: Bitcoin mining is vulnerable. *Commun. ACM* **2018**, *61*, 95–102. [CrossRef]

6. Sapirshtein, A.; Sompolinsky, Y.; Zohar, A. Optimal Selfish Mining Strategies in Bitcoin. In *International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2015.
7. Nayak, K.; Kumar, S.; Miller, A.; Shi, E. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In *Proceedings of the IEEE European Symposium on Security & Privacy*, Saarbruecken, Germany, 21–24 March 2016.
8. Niu, J.; Feng, C. Selfish Mining in Ethereum. In *Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 7–10 July 2019.
9. Cai, D. A Parasite Chain Attack in IOTA. Bachelor's Thesis, University of Twente, Enschede, The Netherlands, 2019.
10. Li, Y.; Cao, B.; Peng, M.; Zhang, L.; Zhang, L.; Feng, D.; Yu, J. Direct Acyclic Graph-based Ledger for Internet of Things: Performance and Security Analysis. *IEEE/ACM Trans. Netw.* **2020**, *28*, 1643–1656. [\[CrossRef\]](#)
11. Staupe, P. Quasi-Analytic Parasite Chain Absorption Probabilities in the Tangle. 2017. Available online: <https://www.iota.org/foundation/research-papers> (accessed on 16 February 2022).
12. Cullen, A.; Ferrar, P.; King, C.; Shorten, R. Distributed Ledger Technology for IoT: Parasite Chain Attacks. 2019. Available online: <https://arxiv.org/pdf/1904.00996.pdf> (accessed on 16 February 2022).
13. Penzkofer, A.; Kusmierz, B.; Caposelle, A.; Sanders, W.; Saa, O. Parasite Chain Detection in the IOTA Protocol. 2020. Available online: <https://arxiv.org/abs/2004.13409> (accessed on 16 February 2022).
14. Liu, Z.; Luong, N.C.; Wang, W.; Niyato, D.; Wang, P.; Liang, Y.C.; Kim, D.I. A Survey on Applications of Game Theory in Blockchain. *arXiv* **2019**, arXiv:1902.10865.
15. Tang, C.B.; Yang, Z.; Zheng, Z.L.; Chen, Z.Y.; Li, X. Game Dilemma Analysis and Optimization of PoW Consensus Algorithm. *Acta Autom. Sin.* **2017**, *43*, 1520–1531.
16. Song, L.; Li, T.; Wang, Y. Applications of Game Theory in Blockchain. *J. Cryptologic Res.* **2019**, *6*, 100–111.
17. Shi, H.; Wang, S.; Hu, Q.; Cheng, X.; Yu, J. Fee-Free Pooled Mining for Countering Pool-Hopping Attack in Blockchain. *IEEE Trans. Dependable Secur. Comput.* **2020**, *18*, 1580–1590. [\[CrossRef\]](#)
18. Phelps, S.; Wooldridge, M. Game Theory and Evolution. *Intell. Syst.* **2013**, *28*, 76–81. [\[CrossRef\]](#)
19. Hu, S.; Cai, C.; Wang, Q.; Wang, C.; Luo, X.; Ren, K. Searching an encrypted cloud meets blockchain: A decentralized, reliable and fair realization. In *Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications*, Honolulu, HI, USA, 16–19 April 2018; pp. 792–800.
20. Cai, C.; Weng, J.; Yuan, X.; Wang, C. Enabling reliable keyword search in encrypted decentralized storage with fairness. *IEEE Trans. Dependable Secur. Comput.* **2018**, *18*, 131–144. [\[CrossRef\]](#)
21. Yan, D.; Jia, X.; Shu, J.; Yu, R. A Blockchain-based Database System for Decentralized Information Management. In *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM)*, Madrid, Spain, 7–11 December 2021; pp. 1–6.
22. Zhang, Y.; Deng, R.H.; Shu, J.; Yang, K.; Zheng, D. TKSE: Trustworthy keyword search over encrypted data with two-side verifiability via blockchain. *IEEE Access* **2018**, *6*, 31077–31087. [\[CrossRef\]](#)
23. Xu, L.; Xu, C.; Liu, Z.; Wang, Y.; Wang, J. Enabling Comparable Search Over Encrypted Data for IoT with Privacy-Preserving. *Comput. Mater. Contin.* **2019**, *60*, 675–690. [\[CrossRef\]](#)
24. Cai, C.; Zheng, Y.; Du, Y.; Qin, Z.; Wang, C. Towards private, robust, and verifiable crowdsensing systems via public blockchains. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 1893–1907. [\[CrossRef\]](#)
25. Miao, Y.; Ma, J.; Liu, X.; Weng, J.; Li, H.; Li, H. Lightweight fine-grained search over encrypted data in fog computing. *IEEE Trans. Serv. Comput.* **2018**, *12*, 772–785. [\[CrossRef\]](#)
26. Guo, Y.; Xie, H.; Wang, C.; Jia, X. Enabling privacy-preserving geographic range query in fog-enhanced iot services. *IEEE Trans. Dependable Secur. Comput.* **2021**. [\[CrossRef\]](#)
27. Wang, M.; Guo, Y.; Zhang, C.; Wang, C.; Huang, H.; Jia, X. MedShare: A Privacy-Preserving Medical Data Sharing System by Using Blockchain. *IEEE TSC* **2021**. [\[CrossRef\]](#)
28. Zhang, C.; Guo, Y.; Jia, X.; Wang, C.; Du, H. Enabling Proxy-free Privacy-preserving and Federated Crowdsourcing by Using Blockchain. *IEEE IoT-J* **2020**, *8*, 6624–6636. [\[CrossRef\]](#)
29. Yao, J.; Zheng, Y.; Guo, Y.; Cai, C.; Zhou, A.; Wang, C.; Gui, X. A Privacy-preserving System for Targeted Coupon Service. *IEEE Access* **2019**, *7*, 120817–120830. [\[CrossRef\]](#)
30. Miao, Y.; Ma, J.; Liu, X.; Li, X.; Liu, Z.; Li, H. Practical attribute-based multi-keyword search scheme in mobile crowdsourcing. *IEEE Internet Things J.* **2017**, *5*, 3008–3018. [\[CrossRef\]](#)
31. Guo, Y.; Xie, H.; Miao, Y.; Wang, C.; Jia, X. Fedcrowd: A federated and privacy-preserving crowdsourcing platform on blockchain. *IEEE Trans. Serv. Comput.* **2020**. [\[CrossRef\]](#)
32. Li, C.; Qu, X.; Guo, Y. TFCrowd: A blockchain-based crowdsourcing framework with enhanced trustworthiness and fairness. *EURASIP J. Wirel. Commun. Netw.* **2021**, *2021*, 1–20. [\[CrossRef\]](#)
33. Xuan, S.; Zheng, L.; Chung, I.; Wang, W.; Man, D.; Du, X.; Yang, W.; Guizani, M. An incentive mechanism for data sharing based on blockchain with smart contracts. *Comput. Electr. Eng.* **2020**, *83*, 106587. [\[CrossRef\]](#)
34. Motepalli, S.; Jacobsen, H.A. Reward Mechanism for Blockchains Using Evolutionary Game Theory. 2021. Available online: <https://arxiv.org/abs/2104.05849> (accessed on 16 February 2022).
35. Popov, S.; Saa, O.; Finardi, P. Equilibria in the tangle. *Comput. Ind. Eng.* **2019**, *136*, 160–172. [\[CrossRef\]](#)
36. Fang, L. The Analysis of SI Group Knowledge Dissemination Model with Expected Effect. *J. Taiyuan Norm. Univ. Sci. Ed.* **2020**, *19*, 9–12.

-
37. Gong, Y.; Li, F.; Zhou, L.; Hu, F. Global Dissemination of Information Based on Online Social Hypernetwork. *J. Univ. Electron. Sci. Technol. China* **2021**, *50*, 437–444.
 38. Feng, L.; Wang, H.; Feng, S. Improved SIR model of computer virus propagation in the network. *J. Comput. Appl.* **2011**, *31*, 1891–1893.
 39. Lu, T. Qualitative Analysis of SEI Model with the Impact of Media. *J. Nanjing Norm. Univ. (Nat. Sci. Ed.)* **2011**, *34*, 32–35.
 40. Li, G.; Zhen, J. Global stability of an SEI epidemic model with general contact rate. *Chaos Solitons Fractals* **2005**, *23*, 997–1004. [[CrossRef](#)]
 41. Wang, X.; Chen, G. *Complex Network Theory and Its Application*; Tsinghua University Press: Beijing, China, 2006.
 42. Mata, A.S. An overview of epidemic models with phase transitions to absorbing states running on top of complex networks. *Chaos* **2021**, *31*, 012101. [[CrossRef](#)]