applied sciences

MDPI

Communication

Effective Selfish Mining Defense Strategies to Improve Bitcoin Dependability

Chencheng Zhou [1], Liudong Xing [1,*], Qisi Liu [2] and Honggang Wang [1]

[1] Department of Electrical and Computer Engineering, University of Massachusetts, Dartmouth, MA 02747, USA
[2] Department of Electrical and Computer Engineering, University of Hartford, Hartford, CT 06117, USA
* Correspondence: lxing@umassd.edu

Abstract: Selfish mining is a typical malicious attack targeting the blockchain-based bitcoin system, an emerging crypto asset. Because of the non-incentive compatibility of the bitcoin mining protocol, the attackers are able to collect unfair mining rewards by intentionally withholding blocks. The existing works on selfish mining mostly focused on cryptography design, and malicious behavior detection based on different approaches, such as machine learning or timestamp. Most defense strategies show their effectiveness in the perspective of reward reduced. No work has been performed to design a defense strategy that aims to improve bitcoin dependability and provide a framework for quantitively evaluating the improvement. In this paper, we contribute by proposing two network-wide defensive strategies: the dynamic difficulty adjustment algorithm (DDAA) and the acceptance limitation policy (ALP). The DDAA increases the mining difficulty dynamically once a selfish mining behavior is detected, while the ALP incorporates a limitation to the acceptance rate when multiple blocks are broadcast at the same time. Both strategies are designed to disincentivize dishonest selfish miners and increase the system's resilience to the selfish mining attack. A continuous-time Markov chain model is used to quantify the improvement in bitcoin dependability made by the proposed defense strategies. Statistical analysis is applied to evaluate the feasibility of the proposed strategies. The proposed DDAA and ALP methods are also compared to an existing timestamp-based defense strategy, revealing that the DDAA is the most effective in improving bitcoin's dependability.

Keywords: bitcoin; selfish mining; dynamic difficulty adjustment algorithm (DDAA); acceptance limitation policy (ALP); statistical analysis

Citation: Zhou, C.; Xing, L.; Liu, Q.; Wang, H. Effective Selfish Mining Defense Strategies to Improve Bitcoin Dependability. *Appl. Sci.* **2023**, *13*, 422. https://doi.org/10.3390/app13010422

Academic Editor: Roberto Saia

Received: 11 November 2022
Revised: 21 December 2022
Accepted: 23 December 2022
Published: 29 December 2022

1. Introduction

Blockchain technology has attracted intensive attention from industries, academia, and governments in the past decade [1–4]. A blockchain is a distributed cryptographic ledger, which consists of a growing list of data packages called blocks. Each block contains a timestamp, the transaction record, and the hash value of the previous block [5]. Its applications span from smart contracts to financial services, from voting to energy trading, and from supply chains to the Internet of Things [6–10]. In this paper, we focus on bitcoin, a peer-to-peer decentralized cryptocurrency system based on blockchain technology [11,12], with a market cap of exceeding USD 1 trillion in October 2021.

Because of being business-critical, bitcoin has become the target of diverse cyber-attacks, including, for example, Sybil attacks [13], Eclipse attacks [14,15], mining pool attacks [16], re-identification attacks [17], miner attacks [18], CryptoLocker-based attacks [19], and selfish mining attacks [20]. Correspondingly, considerable studies were expended in developing mitigation and defense strategies against those attacks. For example, Gervais et al. examined multiple countermeasures (including dynamic timeouts, updating block advertisements, and penalizing nodes that do not respond) to enhance the security of bitcoin [21]. Bamert et al. put forward a hardware token for securing transactions of

bitcoin [22]. Göbel et al. applied Markov chains to detect block-hiding attacks by monitoring the production rate of orphan blocks [23]. In this work, we aim to develop effective strategies to defend the selfish mining attacks, also referred to as block withholding attacks. In such attacks, a selfish miner intentionally keeps the newly mined blocks for building his/her own branch and publishes the private branch to gain unfair revenue when the private branch is longer than the main chain by a certain number of blocks.

Considerable research efforts were devoted to the defense against selfish mining attacks. For example, Heilman proposed a timestamp-based method to examine and control the acceptance of new blocks [24]. Eyal and Sirer suggested a mitigation strategy based on the modification of the bitcoin protocol to cope with collusive selfish mining attacks [20]. Saad et al. developed a network-wide defense mechanism using expected transaction confirmation height and block publishing height [25]. A notion of the "truth state" was introduced to detect potential selfish mining behaviors. Wang et al. suggested a machine learning-based system (ForkDec) for highly accurate selfish mining detection [26]. Bicer et al. proposed a selfish mining mitigation algorithm called Fortis, which does not require a trusted authority for timestamps and protects the honest miner's benefit against any attacker with a computational power of less than 27% [27]. Solat et al. proposed a solution named ZeroBlock to prevent honest nodes from accepting chains infested with block withholding; under this solution, miners are forced to release their blocks within an expected time, otherwise, their mined blocks expire and are rejected by honest miners [28]. Chen et al. proposed a prevention method for the block-withholding attack (PMBWA) based on a credit-level classification algorithm [29]. The algorithm weighs similarity and posterior probability to detect malicious behaviors.

In addition to the defense solutions, there are also studies on the analysis of bitcoin under selfish mining attacks. For example, Wang et al. put forward a mathematical model to investigate the effectiveness of selfish mining attacks, particularly the relationship between computational power and extra mining gain [30]. Motlagh et al. suggested an analytical model to investigate the impacts of selfish mining on the node response time, connectivity of the bitcoin, block delivery time, and arrival rate [31]. Yang et al. used a Markov model to assess the mining revenue, as well as the risk of bitcoin under selfish mining attacks [32]. Xia et al. investigated the influences of multiple miners on selfish mining and the vulnerability of bitcoin under different orphan rates [33]. Zhou et al. suggested a continuous-time Markov chain-based method to estimate the dependability of bitcoin with selfish mining behavior and examined the effects of several attacking and defending parameters on bitcoin dependability, where the dependability is defined as the probability that the system can function normally, i.e., the selfish mining attack is not successful [34].

This work makes contributions by suggesting defense strategies against selfish mining attacks. Particularly, two strategies, referred to as dynamic difficulty adjustment algorithm (DDAA) and the acceptance limitation policy (ALP), are proposed. Their performance and feasibility are investigated and compared through an analytical modeling method and statistical analysis. The results show the improvement using the proposed strategies is statistically significant. The proposed DDAA and ALP methods are also compared to an existing defense strategy to show their effectiveness.

The rest of the paper is arranged as follows. Section 2 presents the state transition diagram of bitcoin under the selfish mining attack and reviews the continuous-time Markov chain (CTMC)-based method for the dependability analysis. Section 3 introduces the DDAA strategy and shows the dependability improvement under this strategy. Optimal parameter selection is also discussed. Section 4 introduces the ALP strategy. Optimal parameter selection and strategy comparison are also discussed. Section 5 compares the proposed DDAA and ALP strategies with an existing timestamp-based method. Section 6 summarizes our research results and points out future study directions.

2. CTMC-Based Dependability Analysis

Figure 1 illustrates the main states and transitions among the states for bitcoin subject to selfish mining under the three-block strategy in the CTMC model.

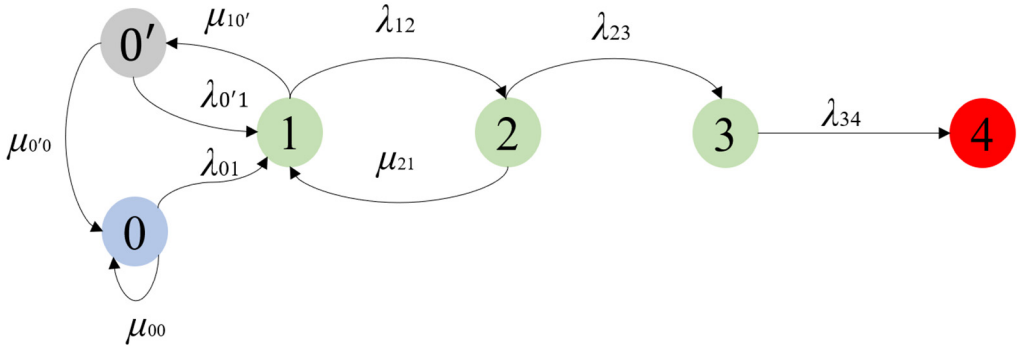

Figure 1. State transition diagram (0: initial state, 0′: double branches, 1: one-block lead, 2: two-block lead, 3: three-block lead, and 4: attack success).

State 0 is the initial state, where only one main chain exists without any branches. The system may transit from state 0 to state 1 when a malicious miner (MM) finds a block but keeps it secretly, leading to a private branch that is one block longer than the main chain; the transition rate is assumed to be λ_{01}. The system stays in state 0 with a rate of μ_{00} if an honest miner (HM) mines the block first.

The system may transit from state 1 to state 2 when the MM successfully mines the next block on his/her private branch; the transition rate is denoted as λ_{12}. The system transits from state 1 to state 0′ when the HM mines the next block before the MM; the transition rate is denoted as $\mu_{10'}$.

The system may transit from state 0′ to state 1 when the MM finds the new block first making the private branch one block longer than the main chain; the transition rate is $\lambda_{0'1}$. The system goes back to state 0 from state 0′ if the HM finds the new block first; the transition rate is μ_{020320}.

The system may transit from state 2 to state 3 when the MM discovers the next block first; the transition rate is λ_{23}. The system transits from state 2 to state 1 if the HM discovers the next block first; the transition rate is μ_{21}.

The system may transit from state 3 to state 4 when the HM successfully discovers the next block; the transition rate is λ_{34}. In state 4, the MM broadcasts the private branch making it the main branch, thus the selfish mining attack succeeds.

Based on the state transition diagram in Figure 1, Equation (1) gives the state equations of the CTMC with the transition rate matrix and the state probability vector on the left-hand side and the vector of the state probabilities' derivative on the right-hand side. Specifically, $P_k(t)$ in Equation (1) represents the probability that the bitcoin occupies state k (k = 0, 0′, 1, 2, 3, 4), and $\dot{P}_k(t)$ represents the derivative of $P_k(t)$.

$$\begin{bmatrix} -\lambda_{01} & \mu_{0'0} & 0 & 0 & 0 & 0 \\ 0 & -(\mu_{0'0}+\lambda_{0'1}) & \mu_{10'} & 0 & 0 & 0 \\ \lambda_{01} & \lambda_{0'1} & -(\mu_{10'}+\lambda_{12}) & \mu_{21} & 0 & 0 \\ 0 & 0 & \lambda_{12} & -(\mu_{21}+\lambda_{23}) & 0 & 0 \\ 0 & 0 & 0 & \lambda_{23} & -\lambda_{34} & 0 \\ 0 & 0 & 0 & 0 & \lambda_{34} & 0 \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_{0'}(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \\ P_4(t) \end{bmatrix} = \begin{bmatrix} \dot{P}_0(t) \\ \dot{P}_{0'}(t) \\ \dot{P}_1(t) \\ \dot{P}_2(t) \\ \dot{P}_3(t) \\ \dot{P}_4(t) \end{bmatrix} \quad (1)$$

Applying the Laplace transform-based method using the initial state probabilities of $P_0(0) = 1$ and $P_k(t) = 0$ (for $k = 0', 1, 2, 3, 4$), as well as the condition of $\sum_{k=0,0'}^{4} P_k(t) = 1$, the state probabilities $P_k(t)$ can be obtained. Refer to [34] for a detailed

solution. The bitcoin dependability is $D(t) = 1 - P_4(t)$ since the selfish mining attack succeeds in state 4. The CTMC-based method presented in this section is applied in Sections 3 and 4 to evaluate the performance of the proposed defense strategies.

3. Dynamic Difficulty Adjustment Algorithm (DDAA)

In the bitcoin protocol, a difficulty parameter controls the overall mining difficulty. To counter the selfish mining attack, we propose the DDAA, which increases the mining difficulty (i.e., requiring higher computing power requirements) by adjusting the difficulty parameter when a successful selfish mining attack is committed. The increasing cost of the malicious attack because of the increasing computing power requirement disincentivizes any future attacks.

Specifically, let K be the difficulty controlling parameter, H be the hash rate that is determined by the available computing power and the value of K, β be the difficulty adjustment parameter, and λ be the state transition rate. In the proposed DDAA, H is negatively correlated with K because an increase in K means a decrease in the hash rate H (H_new). More specifically, a larger K means that more computation and power resources are needed to mine a block; given the same available resources, the hash rate decreases. In other words, to maintain the same hash rate, when K increases, an MM needs to upgrade their computing hardware, which reduces his/her attack reward, thus disincentivizing the attack.

Because a too big β will heavily decelerate the new block generation speed, thus affecting the normal mining process, to control the variance and reduce the impact on the bitcoin network, in our study we set a range for parameter β as $1 < \beta < 2$ (meaning that the increase in the computing power requirement cannot exceed the twice of the original computing power requirement). In the context of the quantitative dependability analysis, the adjustments in K and H are reflected in the decrease in all the λ transition rates. Equation (2) gives the updating formula for K, H, and λ.

$$K\_new = \beta \times K,\ H\_new = H/\beta, \text{and} \lambda\_new = \lambda/\beta. \tag{2}$$

To analyze the effectiveness and performance of the suggested DDAA, we evaluate the bitcoin dependability before and after the difficulty adjustment. Table 1 presents the transition rates related to the MM's computing power (λ_{01}, $\lambda_{0\prime 1}$, λ_{12}, λ_{23}, λ_{34}) and the transition rates related to the HM's recovery capability ($\mu_{0\prime 0}$, $\mu_{10\prime}$, μ_{21}). The values of those transition rates are designed based on the statistics and studies from [35]. Specifically, values in sets a, b, and c are the original values corresponding to the low, medium, and high computing power of the MM. Using Equation (2) with $\beta = 1.25$ (selected for the illustration purpose), the adjusted values of all λ are presented in sets a', b', and c' in Table 1.

Table 1. State transition rate (per hour) values used for numerical analysis.

Rate	Set a	Set b	Set c	Set a'	Set b'	Set c'
λ_{01}	0.03	0.12	0.34	0.024	0.096	0.272
$\lambda_{0'1}$	0.11	0.11	0.11	0.088	0.088	0.088
λ_{12}	0.06	0.18	0.56	0.048	0.144	0.448
λ_{23}	0.04	0.04	0.04	0.032	0.032	0.032
λ_{34}	0.36	0.36	0.36	0.288	0.288	0.288
$\mu_{0\prime 0}$	0.24	0.24	0.24	0.24	0.24	0.24
$\mu_{10\prime}$	0.12	0.12	0.12	0.12	0.12	0.12
μ_{21}	0.31	0.31	0.31	0.31	0.31	0.31

3.1. Effects of the DDAA on Bitcoin Dependability

Table 2 presents bitcoin dependability at different mission times under the six sets of parameters (calculated using the CTMC-based method in Section 2) and the difference in the dependability values for each comparison. Figures 2–4 illustrate the results graphically. It

is apparent that bitcoin dependability improves after the adjustment and the improvement becomes more significant as the mission time proceeds.

Table 2. Bitcoin dependability and comparisons.

t (hrs)	Set a	Set a'	a–a'	Set b	Set b'	b–b'	Set c	Set c'	c–c'
6	0.9995	0.9998	0.0003	0.9950	0.9977	0.0027	0.9745	0.9868	0.0123
12	0.9964	0.9983	0.0019	0.9705	0.9847	0.0142	0.8887	0.9335	0.0448
18	0.9907	0.9954	0.0047	0.9334	0.9631	0.0297	0.7911	0.8646	0.0735
24	0.9835	0.9915	0.0080	0.8924	0.9379	0.0455	0.7007	0.7961	0.0954
30	0.9755	0.9872	0.0117	0.8515	0.9117	0.0602	0.6202	0.7319	0.1117
36	0.9670	0.9826	0.0156	0.8119	0.8855	0.0736	0.5489	0.6726	0.1237

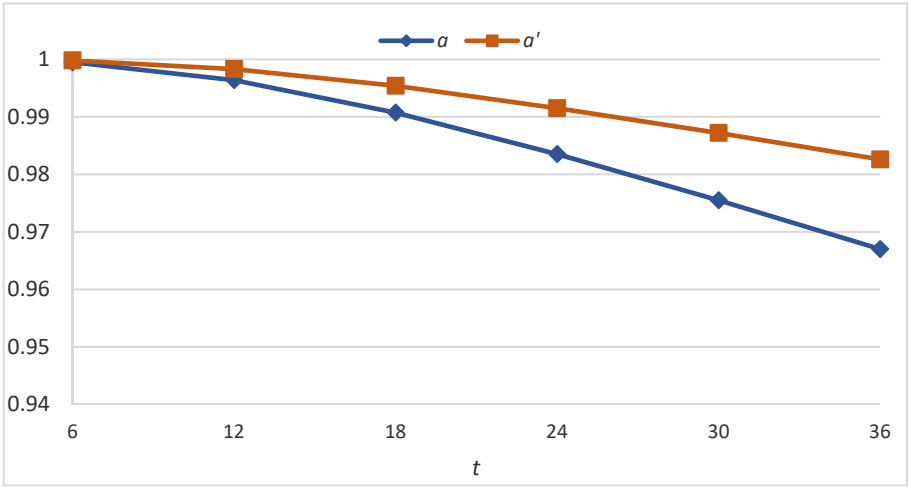

Figure 2. Bitcoin dependability before and after the application of the DDAA under sets a and a'.

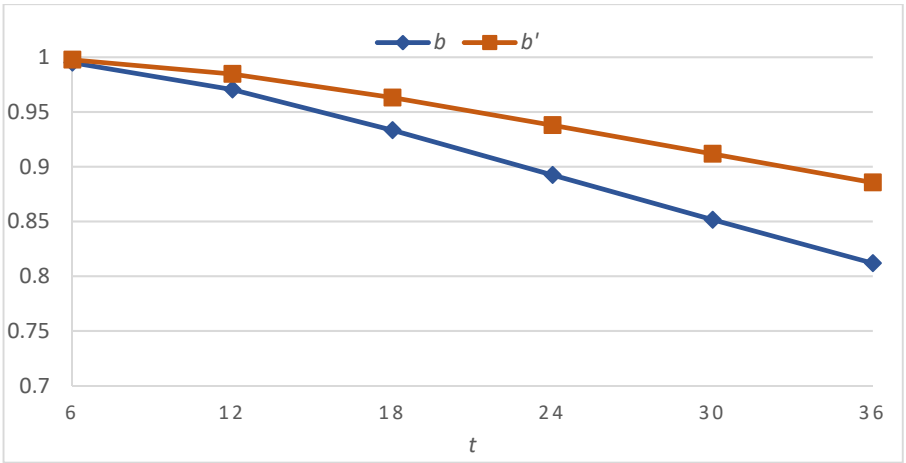

Figure 3. Bitcoin dependability before and after application of the DDAA under sets b and b'.

We apply the paired t-test to examine the effects of the proposed DDAA algorithm. A significant result should prove the feasibility of the algorithm. Before each t-test, we run the F-test to examine the homogeneity of variance of each data pair [36]. Table 3 summarizes the results of the F-test and t-test for each comparison.

For each comparison, the p-value obtained in the F-test is greater than 0.05. So, we accept the null hypothesis. The two datasets in each comparison pass the homogeneity of the variance test. The p-value obtained in the t-test is less than 0.05. So, we reject the null hypothesis. Thus, the dependability values of the two sets in each comparison are

significantly different, and the dependability after the adjustment is significantly higher than that before the adjustment.

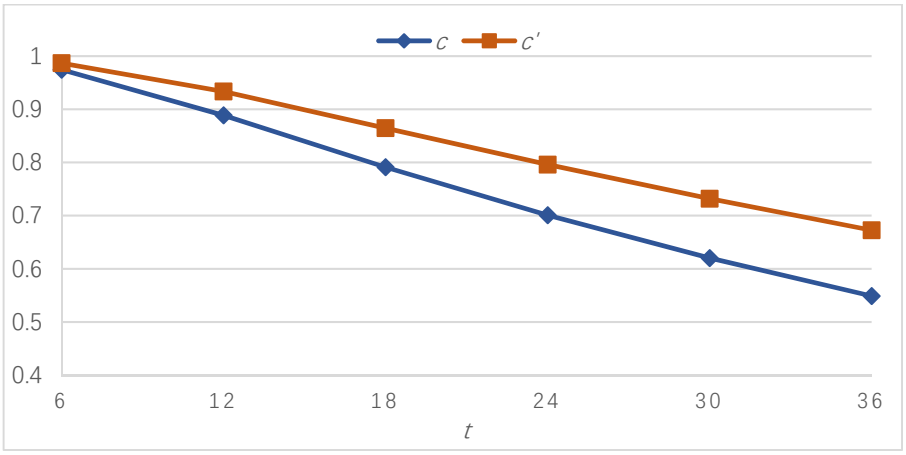

Figure 4. Bitcoin dependability before and after application of the DDAA under sets c and c'.

Table 3. The p-value in F-test and t-test.

	Set a vs. Set a'	Set b vs. Set b'	Set c vs. Set c'
F-test	0.1914	0.3090	0.5277
t-test	0.0164	0.0097	0.0034

3.2. Optimal Parameter Selection for Parameter β

In this section, we examine the optimal value selection for the key parameter β using set a. In other words, we determine the value of β that can offer the largest degree of improvement in Bitcoin dependability under the DDAA strategy. We run a series of tests with different values of β. Among the values tested, the lowest p-value is obtained at $\beta = 1.43$, as shown in Table 4 and Figure 5. Bitcoin dependability under the DDAA with $\beta = 1.43$ at $t = 36$ h is 0.983.

Table 4. The results of the p-value when β varies from 1.1 to 1.6 under the DDAA.

β	1.1	1.25	1.43	1.6
p-value	0.0191	0.0164	0.0145	0.0173

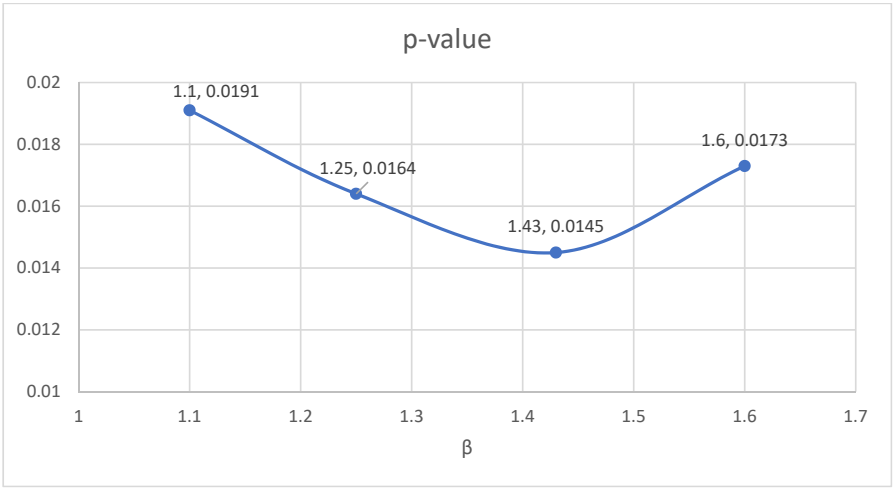

Figure 5. p-value results when β varies from 1.1 to 1.6 under the DDAA.

4. Acceptance Limitation Policy (ALP)

The traditional bitcoin system adopts the proof-of-work (POW) protocol as its consensus mechanism. The longest chain broadcast is accepted as the valid version of the bitcoin main chain. The selfish mining attacker takes advantage of this mechanism by intentionally withholding his/her mined blocks and publishing his/her longer private chain. The ALP intends to set a limitation to the acceptance ratio when multiple blocks are broadcast at the same time. Let γ be the limitation ratio. This process is equal to reducing the transition rate λ_{34} using Equation (3):

$$\lambda_{34\_new} = \lambda_{34} \times \gamma \text{ where } 0 < \gamma \leq 1 \quad (3)$$

To investigate the effectiveness of the ALP, we apply Equation (3) to generate the parameter set a'' based on set a, and both are presented in Table 5.

Table 5. State transition rate (per hour) values used for numerical analysis of the ALP.

Rate	λ_{01}	$\lambda_{0'1}$	λ_{12}	λ_{23}	λ_{34}	$\mu_{0'0}$	$\mu_{10'}$	μ_{21}
Set a	0.03	0.11	0.06	0.04	0.36	0.24	0.12	0.31
Set a''	0.03	0.11	0.06	0.04	$0.36 \times \gamma$	0.24	0.12	0.31

Figure 6 shows bitcoin dependability for five different values of γ, where $\gamma = 1$ corresponds to the case before the adjustment. It is observed that as the value of γ decreases, the improvement in bitcoin dependability becomes more significant.

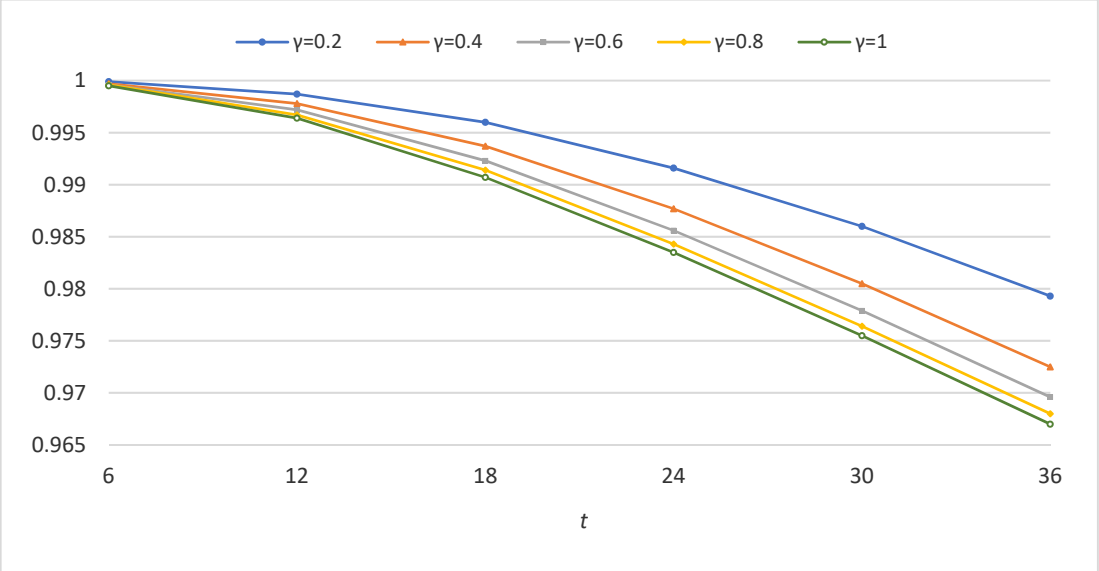

Figure 6. Bitcoin dependability before and after the application of the ALP.

To select the optimal value for the key parameter γ (i.e., the value that provides the largest degree of dependability improvement), we run a series of tests using different values of γ and compute the p-value. Among the values of γ tested, the lowest p-value 0.0033 is obtained at $\gamma = 0.8$, as shown in Table 6 and Figure 7.

Table 6. The results of the p-value under the ALP.

γ	**0.6**	**0.7**	**0.8**	**0.9**
p-value	0.0049	0.0044	0.0033	0.0057

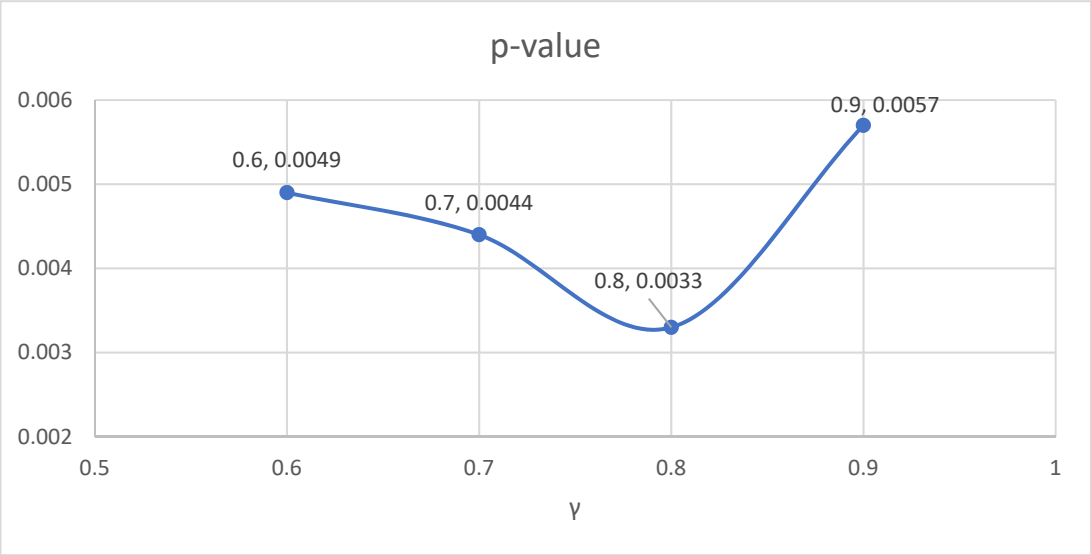

Figure 7. *p*-value results when γ varies from 0.6 to 0.9 under the ALP.

It can be observed from the above results that the optimal *p*-value under the ALP is 0.0033, which is lower than the optimal *p*-value of 0.0145 obtained under the DDAA (Section 3.2). Therefore, in terms of the degree of improving dependability, the ALP performs better than the DDAA in defending selfish mining attacks. However, in terms of the absolute improvement in dependability, the DDAA is more effective than the ALP since the DDAA works on λ_{01}, $\lambda_{0'1}$, λ_{12}, λ_{23}, and λ_{34}, while the ALP only reduces one of those transition rates λ_{34}.

5. Comparative Studies

Heilman suggested a timestamp-based method (TM) to defend selfish mining [24]. The TM-based strategy applies an unforgeable timestamp to ensure that a particular block is mined no later than the timestamp. A random beacon is used as an input to a block. Any miner can act like a verifier to prove that a block has been mined recently. Thus, any block without a timestamp or with an out-of-date timestamp is dropped upon detection. This process is equal to reducing the transition rates $\mu_{0'0}$, $\mu_{10'}$, and μ_{21} using Equation (4):

$$\mu\_new = \mu \times \omega \text{ where } \omega > 1 \tag{4}$$

To compare the effectiveness of the proposed methods and the existing TM method, we use Equation (2) with β = 1.2, 1.4, and 1.6 (for the DDAA), Equation (3) with γ = 1/1.2, 1/1.4, and 1/1.6 (for the ALP), and Equation (4) with ω = 1.2, 1.4, and 1.6 (for the TM) to adjust the parameter set *a* in Table 1, as summarized in Table 7.

Table 7. State transition rate (per hour) adjustments under the DDAA, ALP, and TM.

Set	λ_{01}	$\lambda_{0'1}$	λ_{12}	λ_{23}	λ_{34}	$\mu_{0'0}$	$\mu_{10'}$	μ_{21}
DDAA-*a*	$0.03/\beta$	$0.11/\beta$	$0.06/\beta$	$0.04/\beta$	$0.36/\beta$	0.24	0.12	0.31
ALP-*a*	0.03	0.11	0.06	0.04	$0.36\times\gamma$	0.24	0.12	0.31
TM-*a*	0.03	0.11	0.06	0.04	0.36	$0.24\times\omega$	$0.12\times\omega$	$0.31\times\omega$

The results in Table 8 and Figure 8 show that for every comparison group (β = ω = $1/\gamma$ = 1.2, 1.4, and 1.6), the bitcoin system under the DDAA always has the highest dependability among the three defense strategies. Thus, we conclude that the DDAA has better defense effectiveness than the TM and ALP approaches in terms of absolute dependability improvement.

Table 8. Bitcoin dependability under the DDAA, ALP, and TM.

t (hrs)	β = 1.2	γ = 1/1.2	ω = 1.2	β = 1.4	γ = 1/1.4	ω = 1.4	β = 1.6	γ = 1/1.6	ω = 1.6
6	0.9997	0.9995	0.9995	0.9998	0.9996	0.9996	0.9999	0.9996	0.9996
12	0.9980	0.9967	0.9969	0.9988	0.9969	0.9972	0.9992	0.9971	0.9976
18	0.9948	0.9912	0.9922	0.9968	0.9917	0.9934	0.9979	0.9921	0.9944
24	0.9905	0.9842	0.9866	0.9940	0.9848	0.9889	0.9960	0.9854	0.9907
30	0.9857	0.9762	0.9804	0.9909	0.9769	0.9840	0.9938	0.9776	0.9868
36	0.9805	0.9678	0.9739	0.9875	0.9678	0.9790	0.9914	0.9693	0.9828

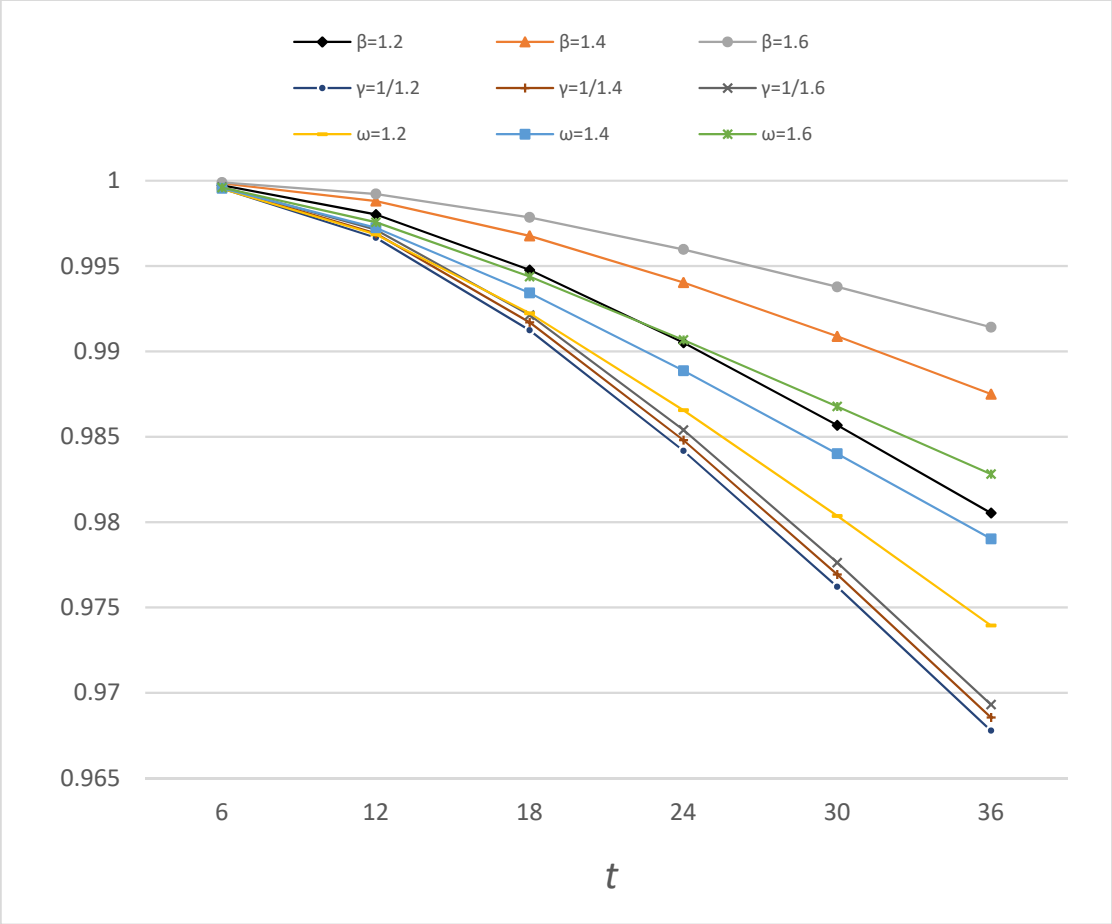

Figure 8. Bitcoin dependability under the DDAA, ALP, and TM.

6. Conclusions and Future Directions

The bitcoin network is developed based on blockchain technology. It is considered one of the most dependable systems because of its distributed, decentralized, and unchangeable features. However, there are diverse attacks targeting bitcoin in the past few years (e.g., Eclipse attacks, Sybil attacks, and selfish mining attacks), which have caused tremendous losses. This work focuses on the selfish mining attack, where selfish miners intentionally withhold the mined block and build their own private chain. The malicious miners then publish their longer chain at a certain point in time to win all mining rewards. Most of the existing research on the selfish mining attack focused on adversary risk detection and cryptography design. There is a lack of studies on strategies that improve bitcoin system dependability and on the statistical proof of the significance of the effectiveness. Our research contributes to the state of the art by designing two defense strategies of the DDAA and ALP and verifying their effectiveness using the CTMC-based dependability analysis, as well as the *t*-test-based statistical analysis. The analysis results show that both strategies can greatly improve bitcoin dependability. Additionally, by comparing the optimal cases,

it was found that the ALP performs better than the DDAA in the degree of dependability improvement. The two proposed strategies are also compared with the existing TM-based strategy. The comparative studies show that the DDAA is the most effective in improving bitcoin dependability.

In a future study, we will extend the defense strategies designed in this work to mitigate other types of attacks, such as Eclipse attacks and Sybil attacks. We also plan to develop a universal defense strategy to improve the resilience of the blockchain-based digital currency network.

Author Contributions: Methodology, C.Z.; Validation, Q.L. and H.W.; Investigation, C.Z.; Data curation, C.Z.; Writing—original draft, C.Z. and L.X.; Writing—review & editing, L.X., Q.L. and H.W.; Visualization, C.Z.; Supervision, L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Cybersecurity Graduate Research Fellowship from the University of Massachusetts Dartmouth Cybersecurity Center.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Bhushan, B.; Sahoo, C.; Sinha, P.; Khamparia, A. Unification of Blockchain and Internet of Things (BIoT): Requirements, working model, challenges and future directions. *Wirel. Netw.* **2021**, *27*, 55–90. [CrossRef]
2. Dai, H.N.; Zheng, Z.; Zhang, Y. Blockchain for Internet of Things: A survey. *IEEE Internet Things J.* **2019**, *6*, 8076–8094. [CrossRef]
3. Ferrag, M.A.; Derdour, M.; Mukherjee, M.; Derhab, A.; Maglaras, L.; Janicke, H. Blockchain technologies for the internet of things: Research issues and challenges. *IEEE Internet Things J.* **2018**, *6*, 2188–2204. [CrossRef]
4. Kang, J.; Yu, R.; Huang, X.; Wu, M.; Maharjan, S.; Xie, S.; Zhang, Y. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet Things J.* **2018**, *6*, 4660–4670. [CrossRef]
5. Nofer, M.; Gomber, P.; Hinz, O.; Schiereck, D. Blockchain. *Bus. Inf. Syst. Eng.* **2017**, *59*, 183–187. [CrossRef]
6. Akbari, E.; Wu, Q.; Zhao, W.; Arabnia, H.R.; Yang, M.Q. From blockchain to internet-based Voting. In Proceedings of the 2017 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 14–16 December 2017; pp. 218–221.
7. Frizzo-Barker, J.; Chow-White, P.A.; Adams, P.R.; Mentanko, J.; Ha, D.; Green, S. Blockchain as a disruptive technology for business: A systematic review. *Int. J. Inf. Manag.* **2020**, *51*, 102029. [CrossRef]
8. Wongthongtham, P.; Marrable, D.; Abu-Salih, B.; Liu, X.; Morrison, G. Blockchain-enabled Peer-to-Peer energy trading. *Comput. Electr. Eng.* **2021**, *94*, 107299. [CrossRef]
9. Xing, L. Reliability in Internet of Things: Current status and future perspectives. *IEEE Internet Things J.* **2020**, *7*, 6704–6721. [CrossRef]
10. Xing, L. Cascading failures in Internet of Things: Review and perspectives on reliability and resilience. *IEEE Internet Things J.* **2021**, *8*, 44–64. [CrossRef]
11. Satoshi, N. Bitcoin: A peer-to-peer electronic cash system. *Consulted* **2008**, *1*, 28.
12. Tschorsch, F.; Scheuermann, B. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 2084–2123. [CrossRef]
13. Zhang, S.; Lee, J.H. Double-spending with a sybil attack in the Bitcoin decentralized network. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5715–5722. [CrossRef]
14. Zhou, C.; Xing, L.; Liu, Q. Dependability Analysis of Bitcoin subject to Eclipse Attacks. *Int. J. Math. Eng. Manag. Sci.* **2021**, *6*, 469–479. [CrossRef]
15. Zhou, C.; Xing, L.; Liu, Q.; Wang, H. Semi-Markov Based Dependability Modeling of Bitcoin Nodes under Eclipse Attacks and State-Dependent Mitigation. *Int. J. Math. Eng. Manag. Sci.* **2021**, *6*, 480–492. [CrossRef]
16. Bahack, L. Theoretical Bitcoin attacks with less than half of the computational power (draft). *arXiv* **2013**, arXiv:1312.7013. Available online: https://eprint.iacr.org/2013/868.pdf (accessed on 3 September 2022).
17. Meiklejohn, S.; Pomarole, M.; Jordan, G.; Levchenko, K.; McCoy, D.; Voelker, G.M.; Savage, S. A fistful of Bitcoins: Characterizing payments among men with no names. In Proceedings of the 2013 Conference on Internet Measurement Conference, Barcelona, Spain, 23–25 October 2013; pp. 127–140.
18. Rosenfeld, M. Analysis of Bitcoin pooled mining reward systems. *arXiv* **2011**, arXiv:1112.4980.

19. Liao, K.; Zhao, Z.; Doupé, A.; Ahn, G.J. Behind closed doors: Measurement and analysis of CryptoLocker ransoms in Bitcoin. In Proceedings of the 2016 APWG Symposium on Electronic Crime Research (eCrime), Toronto, ON, Canada, 1–3 June 2016; pp. 1–13.
20. Eyal, I.; Sirer, E.G. Majority is not enough: Bitcoin mining is vulnerable. In *International Conference on Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 436–454.
21. Gervais, A.; Ritzdorf, H.; Karame, G.O.; Capkun, S. Tampering with the delivery of blocks and transactions in Bitcoin. In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12 October 2015; pp. 692–705.
22. Bamert, T.; Decker, C.; Wattenhofer, R.; Welten, S. Bluewallet: The secure Bitcoin wallet. In *International Workshop on Security and Trust Management*; Springer: Cham, Switzerland, 2014; pp. 65–80.
23. Göbel, J.; Keeler, H.P.; Krzesinski, A.E.; Taylor, P.G. Bitcoin blockchain dynamics: The selfish-mine strategy in the presence of propagation delay. *Perform. Eval.* **2016**, *104*, 23–41. [CrossRef]
24. Heilman, E. One weird trick to stop selfish miners: Fresh bitcoins, a solution for the honest miner. In Proceedings of the Financial Cryptography Data Security, Christ Church, Barbados, 7 March 2014; pp. 161–162.
25. Saad, M.; Njilla, L.; Kamhoua, C.; Mohaisen, A. Countering selfish mining in blockchains. In Proceedings of the 2019 International Conference on Computing, Networking and Communications (ICNC), Honolulu, HI, USA, 18–21 February 2019; pp. 360–364.
26. Wang, Z.; Lv, Q.; Lu, Z.; Wang, Y.; Yue, S. ForkDec: Accurate Detection for Selfish Mining Attacks. *Secur. Commun. Netw.* **2021**, *2021*, 5959698. [CrossRef]
27. Biçer, O.; Küpçü, A. FORTIS: Selfish Mining Mitigation by (FOR)geable (TI)me(S) tamps. *Cryptol. Eprint Arch.* 2020. Available online: https://eprint.iacr.org/2020/1290.pdf (accessed on 7 October 2022).
28. Solat, S.; Potop-Butucaru, M. Brief announcement: Zeroblock: Timestamp-free prevention of block-withholding attack in bitcoin. In *International Symposium on Stabilization, Safety, and Security of Distributed Systems*; Springer: Cham, Switzerland, 2017; pp. 356–360.
29. Chen, H.; Chen, Y.; Xiong, Z.; Han, M.; He, Z.; Liu, B.; Wang, Z.; Ma, Z. Prevention method of block withholding attack based on miner' mining behavior in blockchain. *Appl. Intell.* **2022**, 1–19. [CrossRef]
30. Wang, S.; Yin, B.; Zhang, S.; Cheng, Y.; Cai, L.X.; Cao, X. A Selfish attack on chainweb blockchain. In Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference, Taipei, China, 7–11 December 2020; pp. 1–6.
31. Motlagh, S.G.; Mišić, J.; Mišić, V.B. The Impact of Selfish Mining on Bitcoin Network Performance. *IEEE Trans. Netw. Sci. Eng.* **2021**, *8*, 724–735. [CrossRef]
32. Yang, R.; Chang, X.; Mišić, J.; Mišić, V.B. Assessing blockchain selfish mining in an imperfect network: Honest and selfish miner views. *Comput. Secur.* **2020**, *97*, 101956. [CrossRef]
33. Xia, Q.; Dou, W.; Xi, T.; Zeng, J.; Zhang, F.; Wei, J.; Liang, G. The Impact Analysis of Multiple Miners and Propagation Delay on Selfish Mining. In Proceedings of the 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), Taipei, China, 12–16 July 2021; pp. 694–703.
34. Zhou, C.; Xing, L.; Guo, J.; Liu, Q. Bitcoin Selfish Mining Modeling and Dependability Analysis. *Int. J. Math. Eng. Manag. Sci.* **2022**, *7*, 16–27.
35. 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, 2016; pp. 515–532.
36. Savin, N.E. Mutiple Hypothesis Testing. In *Handbook of Econometrics*; Elsevier: Amsterdam, The Netherlands, 1984; Volume 2.