Appendices

Proof of Proposition 4:

The verifiability of a miner's action is a key requirement for tacit collusion. Colluding miners should be able to verify each other's partial block filling actions. New blocks on the Bitcoin network – full or partial – are public. Colluding miners can perform their actions using a unique public address to add their blocks. An alternate method is for colluding miners to pool their computational power. For example, Antpool is a group of miners who have decided to pool their resources together for mining. Each miner participating in a pool contributes a verifiable amount of computational power and submits blocks under the pool ID. The pool then distributes the rewards from winning blocks based on the work performed. Regardless of whether the miners gather together in a pool or perform their actions using a unique public address, they can still deviate. However, such a deviation is easily detectable (Cong and He 2019, Malinova and Park 2017). If the colluding miners use the same public address, transactions included in any block that they win will reveal if they are deviating. However, it is possible for the miner to use a different public address when cheating. In a pool, this would not be possible because the pool can verify the computational power contributed by the miner to the pool and also the transactions included by the miner in a block. Outside of a pool, a drop in the win rate of a large miner (identified by the public address) may indicate that he is cheating using another address to submit fully filled blocks. Further, a colluding miner does not need to verify every other miner's actions. Overall, if α_l fraction of the total computational power is assumed to be involved in collusion, a participating miner simply needs to check that on average α_l fraction of blocks are partially filled to ensure that no one is deviating.

Every miner prepares a block of transactions and then starts to look for the solution to the Bitcoin mining puzzle. Miners need to decide their filling action – full or partial – before starting to find the mining puzzle solution. A colluding miner considers a trade-off (equation 45) between (a) immediate revenues from a partial fill n_P^* facing a bid vector $f(v)$ i.e., R_{noDev}, followed by expected collusion profits R_P for T periods or (b) deviation to maximize revenues from current block R_{noDev}, followed by expected no collusion profits R_0 for T periods. Recall that δ represents the time discounting factor for Bitcoin users, which can potentially be different from that for miners (δ_m). A colluding miner of size α_j will not deviate from collusion if

$$R_{noDev} + \alpha_j R_P * \frac{\delta_m(1-\delta_m^T)}{1-\delta_m} \geq R_{Dev} + \alpha_j R_0 * \frac{\delta_m(1-\delta_m^T)}{1-\delta_m}, \tag{45}$$

This simplifies to,

$$\alpha_j \geq \frac{1-\delta_m}{\delta_m(1-\delta_m^T)} \times \frac{R_{Dev} - R_{noDev}}{R_P - R_0}. \tag{46}$$

If the miner observes equilibrium fee bid $f^*(v)$ then R_{noDev} corresponds to R_P, and R_{Dev} corresponds to R_F. Further, miner must be able to sustain the collusive partial fill level n_P^* in response to any off equilibrium enticing bid vector $f(v) \neq f^*(v)$ as well. We can upper bound the right hand side $(R_{Dev} - R_{noDev})$ by considering the most enticing fee bid vector $f(v)$ similar to single strategic miner setting in the last section,

$$\alpha_j \geq \hat{\alpha}, \quad \text{where} \quad \hat{\alpha} = \frac{1-\delta_m}{\delta_m(1-\delta_m^T)} \times \frac{\alpha_h \rho N}{2V} \times \frac{v_h^2 - v_l^2}{R_P - R_0}. \tag{47}$$

This represents different trade-off faced by heterogeneous miners. A small miner ($\alpha_j < \hat{\alpha}$) mines blocks infrequently, say, once a month or year. They are less threatened by punishment far off in the future. This can also be seen as a small miner's desire to make the most out of winning a mining puzzle once in a long while. A large miner sticks to the collusion strategy, as they expect frequent or near-term fee revenues. The lower bound on the smallest colluding miner ($\hat{\alpha}$) can be reduced by a longer punishment strategy ($T \to \infty$). This can be useful if small miner participation is necessary to attain a colluding group with total power α_l.

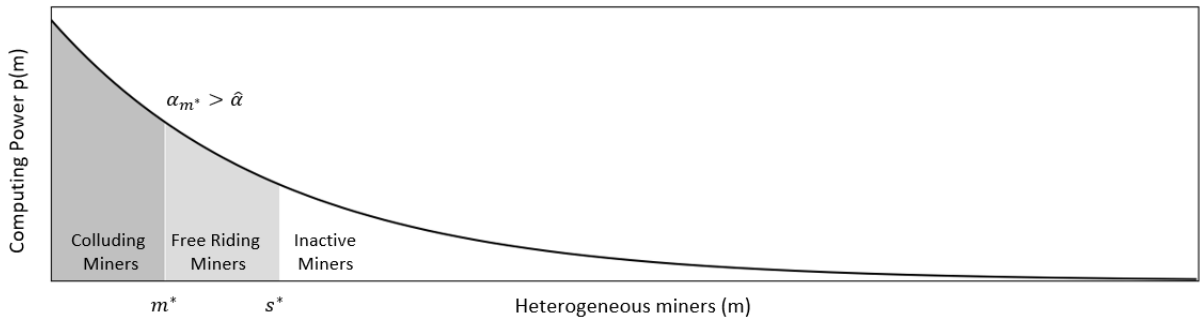

Figure 21 **The x-axis represents miners ranked by access to hardware. The y-axis represents the computing power of the respective miner. An example of a monotonically decreasing concave function $h(m) = \lambda e^{-\lambda m}$ with a long tail. Three categories of miners, from left to right, - (1) Colluding by partial block filling, (2) Free Riding by full block filling, and (3) Inactive miners. m^* and s^* represent the boundary between these groups in a collusion equilibrium.**

The constraints above ($\alpha_j \geq \hat{\alpha}$, $\Sigma\alpha_j \geq \alpha_l$) assures a colluding miner's commitment to the collusion. We could use this to check if an exogenously given distribution of mining powers (α_j) conforms to collusion requirements. However, miner entry is endogenously determined by available revenues and hardware distribution. Next we identify corresponding constraints on mining hardware distribution for collusion. This is useful because a Blockchain designer does not directly control mining powers (α_j), but they control hardware distribution via choice of mining puzzle (e.g., SHA 256 Hash, Scrypt). It would allows us to discuss potential Blockchain designs for averting collusion. Figure 21 illustrates

Table 4 **Single-period expected payoffs corresponding to three actions (Partial, Full, and No Mining) off and on collusion equilibrium paths.**

	On Equilibrium (α_l)	**Off Equilibrium** $(\alpha_l = 0)$
Partial $R(\alpha_j, P, *)$	$f_h n_P$	$f_0 n_P$
Full $R(\alpha_j, F, *)$	$f_h n_P + f_l(n_F - n_P)$	$f_0 n_F$
No Mining $R(\alpha_j, 0, *)$	0	0

our model of miner heterogeneity with respect to access to computing hardware. A large miner has access to hardware capable of faster hash calculations at the same cost. Given the favorable trade-off for the large miners, we focus on collusion among a group ($\Sigma\alpha_j = \alpha_l$) of the largest miners. We must ensure that no miner (colluding, free riding, or outside) deviates from their action in equilibrium.

Figure 22 depicts the sequence of choices for a miner. First, they decide whether to purchase the hardware. Second, they decide whether to follow the collusion strategy. The second choice is repeated over infinite block creation periods. These choices are made by all miners simultaneously. We want to identify conditions under which the top m^* miners collude and the next $(s^* - m^*)$ miners participate as free riders. In a subgame equilibrium, these participating miners should be willing to purchase the hardware and stick to the collusion strategy. All miners beyond s^* should be better off not purchasing the hardware. Individual miners have rational expectations of equilibrium strategies followed by all other miners. The focal miner first decides whether to buy the hardware. Under collusion equilibrium, a rational miner expects the top s^* miners to buy hardware. Next, they decide whether to follow the equilibrium collusion strategy or the no-collusion strategy. On the collusion path, a rational miner expects the top m^* miners to add partial blocks. On the no-collusion (off-equilibrium) path, a rational miner expects all active miners to add full blocks. On both the collusion equilibrium and the no-collusion off-equilibrium path, the focal miner exercises a choice of partial filling, full filling or no mining. The block creation sub-game on-equilibrium (right) and off-equilibrium (left) paths are repeated infinitely with a discount factor δ_m.

Single-block-fee payoffs are denoted by $R(*, *, *)$ with three arguments. The first argument represents the mining power of the focal miner. The second argument represents the focal miner block fill action, i.e., partial (P) or full (F). The third argument represents the colluding group power, i.e., collusion (α_l) or no collusion (0). Table 4 provides the single-period expected payoffs corresponding to all actions. Single-period payoffs are strictly better under the full block filling action for all miners. If the focal miner has power $\alpha_j \geq \alpha_m$ but decides to add a full block under collusion, they expect to be punished. All miners would move to the no-collusion sub-game if a single miner deviates from the collusion.

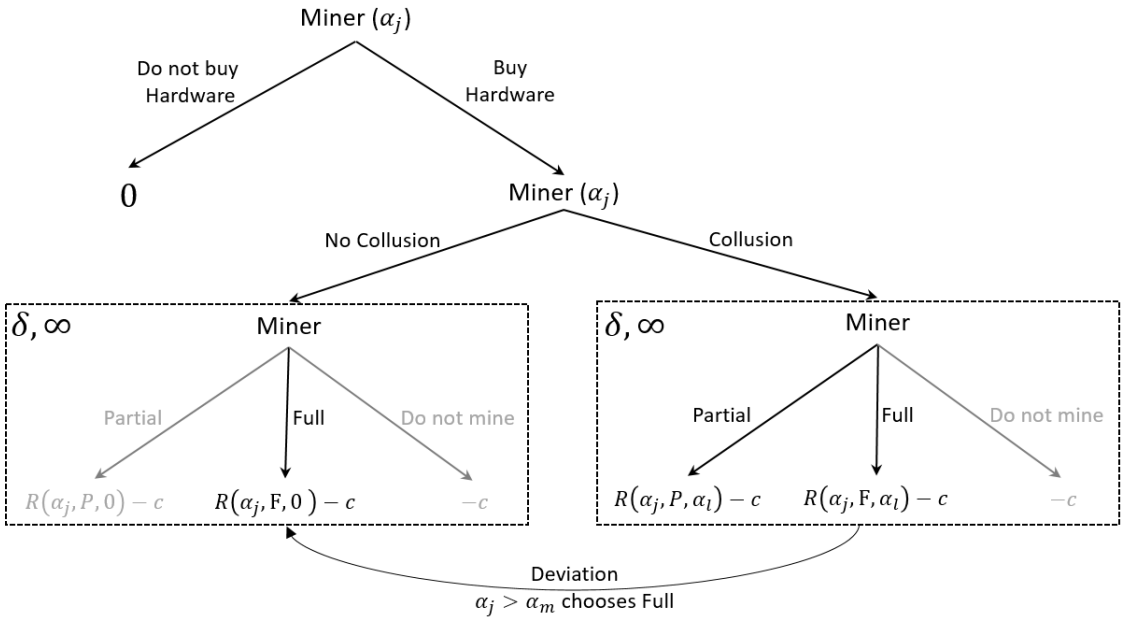

Figure 22 **Miner j with access to computing power (α_j) decides whether to buy hardware. Next, the miner decides whether to follow the collusion strategy. The colluding group of miners chooses Partial fill. The free-riding group chooses Full. Deviation by a colluding group member leads to a no-collusion setting whereby everyone always fully fills their block. The block creation sub-games on equilibrium (right) and off equilibrium (left) are repeated infinitely with a discount factor δ_m.**

Table 5 lists all constraints that ensure that no miner has a profitable deviation in an SPE. For a large focal miner ($\alpha_j > \alpha_m$), constraint 1a ensures that they prefer to add partial blocks rather than a full block in the repeating block creation sub-game. This is fulfilled if the marginal miner satisfies $\alpha_m \geq \hat{\alpha}$. Constraint 1b represents their preference to buy the hardware at the start of the game. This is satisfied for the marginal miner making at least zero profits.

$$\alpha_m = \frac{c_m(1-\delta)}{R_P} \geq \hat{\alpha}; \quad \text{where} \quad R_P = f_h n_P \tag{48}$$

A miner with ($\alpha_m \geq \alpha_j \geq \alpha_s$) proportion of the total power free rides. The lower limit α_s denotes the smallest miner that joins the mining network. Constraint 2a represents the preference to free ride over joining the colluding group. Joining the colluding group would increase the power of the colluding group to $\alpha_l + \alpha_j$ and therefore the partial block revenues. If the marginal miner ($\alpha_j = \alpha_m$) is large enough, they may increase the partial block revenues $R_P(\alpha_l + \alpha_j)$ to be higher than the full block revenue $R_F(\alpha_l)$. In this section, we are interested in settings whereby even the largest miner is too small to perform partial block filling without a threat of punishment [19]. Large miners unilaterally perform partial block filling as shown in Section 3.2.

[19] An equilibrium at α_l is only valid when individual miners are relatively small: $\alpha_m \leq R_P^{-1}(R_F(\alpha_l)) - \alpha_l)$

Table 5 List of all constraints that ensure that no miner has a profitable deviation in an SPE. Three pairs of constraints (1a,1b), (2a,2b) and (3a,3b) correspond to three types of miners - (1) colluding miners, (2) free-riding miners and (3) inactive miners, respectively.

Focal Miner	Constraint
$\alpha_j > \alpha_m$	**1a** $R(\alpha_j, P, \alpha_l) + \alpha_j \frac{\delta_m}{1-\delta_m} R(\alpha_j, P, \alpha_l) \geq R(\alpha_j, F, \alpha_l) + \alpha_j \frac{\delta_m}{1-\delta_m} R(\alpha_j, F, 0)$
	1b $\frac{1}{1-\delta_m} R(\alpha_j, P, \alpha_l) - c_j \geq 0$
$\alpha_m \geq \alpha_j \geq \alpha_s$	**2a** $R(\alpha_j, F, \alpha_l) + \alpha_j \frac{\delta_m}{1-\delta_m} R(\alpha_j, F, \alpha_l) \geq R(\alpha_j, P, \alpha_l + \alpha_j) + \alpha_j \frac{\delta_m}{1-\delta_m} R(\alpha_j, P, \alpha_l + \alpha_j)$
	2b $\frac{1}{1-\delta_m} R(\alpha_j, F, \alpha_l) - c_j \geq 0$
$\alpha_j \leq \alpha_s$	**3a** $\alpha_j \frac{1}{1-\delta_m} R(\alpha_j, F, 0) - c_j \leq 0$
	3b $\frac{1}{1-\delta_m} R(\alpha_j, P, \alpha_l + \alpha_j) - c_j \leq 0$

Constraint 2b represents free-riding miners' preference to buy the hardware at the start of the game. This is satisfied for the smallest miner making positive profits.

$$\alpha_s \geq \frac{c_s(1-\delta)}{R_F}; \quad \text{where} \quad R_F = f_h n_P + f_l(n_F - n_P) \tag{49}$$

For a focal miner who stays out ($\alpha_j \leq \alpha_s$), constraint 3a represents their preference to not join as a free rider. Joining in as a free rider reduces the power of the colluding group below α_l and makes collusion unprofitable. Since the smallest miner makes zero profits when free riding, they are guaranteed to make negative profits when collusion breaks. Finally, constraint 3b represents their preference to join the colluding group. Similar to the free rider, this increases the power of the colluding group to $\alpha_l + \alpha_j$ and therefore the partial block revenues. These partial block revenues $R(\alpha_l + \alpha_j, P, \alpha_j)$ must be smaller than zero. This is automatically satisfied since $\alpha_s < \alpha_m$.

We now proceed to obtain the equilibrium expressions for the smallest colluding miner, denoted by m^*, and the smallest free-riding miner, denoted by s^*. For the smallest colluding miner, we need constraint 1b to become an equality. Specifically, we need

$$\frac{1}{1-\delta} R(\alpha_m, P, \alpha_l) - c_m = 0.$$

We know that $R(\alpha_m, P, \alpha_l) = \alpha_m \times R_P$. Thus, from the above equation, we have

$$\alpha_m \times R_P = c_m(1-\delta)$$

or equivalently

$$\alpha_m = \frac{c_m(1-\delta)}{R_P}. \tag{50}$$

From the definition of α_l, we know that

$$\alpha_l = \frac{H(m^*)}{H(s^*)}.$$

Thus, we have

$$H(s^*) = \frac{H(m^*)}{\alpha_l}. \tag{51}$$

From the definition of α_j, we also know that

$$\alpha_m = \frac{h(m^*)}{H(s^*)}.$$

Substituting the expression of $H(s^*)$ from (51), we obtain

$$\alpha_m = \frac{h(m^*)\alpha_l}{H(m^*)} = \Lambda(m^*)c(m^*)\alpha_l,$$

where $\Lambda(m) \equiv \frac{h(m)}{H(m)c(m)}$. Using (50), we have

$$\Lambda(m^*)c(m^*)\alpha_l = \frac{c(m^*)(1-\delta)}{R_P},$$

or

$$m^* = \Lambda^{-1}\left(\frac{1-\delta}{\alpha_l R_P}\right). \tag{52}$$

The monotonically decreasing function $\Lambda \equiv \frac{h(m)}{H(m)c(m)}$ ensures unique solutions. The marginal colluding miner earns zero profit and must be large enough such that future punishment is a credible threat ($\alpha_{m^*} \geq \hat{\alpha}$).

$$\frac{c(m^*)(1-\delta_m)}{R_P} \geq \hat{\alpha} \tag{53}$$

Free-riding miners present in equilibrium enter until colluding miners have exactly α_l proportion of the total computing power. Using (51), we have

$$s^* = H^{-1}\left(\frac{H(m^*)}{\alpha_l}\right), \tag{54}$$

where m^* is given in (52). In addition, the smallest free-riding miner must be large enough to make positive profits.

$$\alpha_{s^*} = \frac{h(s^*)}{H(s^*)} \geq \frac{c(s^*)(1-\delta_m)}{R_F} \tag{55}$$

■

The subgame perfect equilibrium above focuses on a colluding group with exactly α_l power. All miners adding partially filled blocks ($\Sigma\alpha_j = 1$) is yet another SPE. In such a case, users either stay

off the chain or offer a high fee (f_h); no one offers a low fee (f_l). As a result, a deviation to add a full block (R_F) with low-fee-paying transactions is not an option. We have not observed such full collusion on the Bitcoin network. We do not provide a specific justification for one equilibrium over other; however, we focus on the α_l collusion as a more interesting and practical setting. In addition to these extreme cases, collusive equilibria with $\alpha_l \leq \Sigma\alpha_j \leq \alpha_h$ may also be possible. If a single miner with power $\alpha_{j'}$ deviates from such collusion, the remaining group is left with power $\Sigma\alpha_j - \alpha_{j'}$. Punishing the deviating miners requires this group to not engage in collusion permanently. This is not necessarily a rational strategy for the remaining $\Sigma\alpha_j - \alpha_{j'}$ group at this stage. They are better off colluding on partial block filling if $\Sigma\alpha_j - \alpha_{j'} > \alpha_l$. This rational strategy to collude with a smaller group does not constitute a threat to the deviating miner. The deviating miner is thus better off by continuing to free ride. In this paper, we do not validate or reject the existence of miner strategies that sustain such equilibria.

A. Off Chain Payment

In practice, off-chain alternative for Bitcoin is a combination of Credit Cards, PayPal and Wire Transfer (SWIFT). Following is a very rough estimate of proportional fees charged and time to complete for these off-chain modes on an international payment. While Credit Cards offer almost instant payment faster than Bitcoin, Wire transfers are much slower than Bitcoin. Our writing did not provide this detail on off-chain alternatives.

Mode	Fees	Time
SWIFT (Wire)	1-3% (fx markup) + \$50 (payer bank) + \$20 (correspondent bank) + \$20 (receiver bank) +	3 Days
PayPal	3% (to send) + 4.4% (to receive)	3+ Days
Credit Card	1-5% (to send) + 3% (to receive)	Instant

The utility of making an international payment v using mode m could be modeled as $U_m(v)$, where ρ_m is the proportional fees and $\delta(t_m)$ is the discount factor over the payment completion time. The off-chain modes offer different combination of fees and time to completion (ρ_m, t_m). We assume a single proportional cost $\rho = \rho_m + \delta(t_m)$ as a combination of fees and time delay. This is reasonable because typically low proportional rate options have large delay and vice versa.

$$U_m(v) = v - \rho_m v - \delta(t_m)v$$

$$U_m(v) = v - \rho v \qquad \text{where } \rho = \rho_m + \delta(t_m)$$

Modeling the on-chain time discount factor is critical because time to completion is endogenously determined by user action (e.g., fee offer) and competition. Modeling the off-chain discount factor

is not necessary because it is fixed in advance irrespective of user action. It is accounted for via (ρ). The choice of off-chain features (ρ_m, t_m) provided by Banks or Credit Cards in response to user choices are outside the scope of our research.

B. User Collusion

Let us consider a scenario where miners act passively by adding full blocks (n_F). If users compete every period, top n_F users each pay fees $f_0 = \alpha_h \rho(1-\gamma)V$. Let us consider users collusion such that the top n_F users pay a lower fees $f_c < f_0$ while the remaining users stay off-chain instead of competing with the top n_F users on fee. The marginal user $v = v_0$ has the highest value payment among users who stay off chain. This user has greatest incentive to deviate by offering a fees $f_c + \epsilon$ to complete their payment on chain. This user can be prevented from deviating to $f_c + \epsilon$ fee bid by a punishment threat i.e. no collusion in future. This threatens the marginal user because they will likely draw a payment need $v \in [v_0, V_{max}]$ in subsequent periods. A break down in collusion means that they will not be able to benefit from the low collusion fees f_c ($< f_0$). The marginal user pays $f_c + \epsilon$ by deviating and expects to pay no collusion fees $E_v^0[f(v)]$ in future periods. Otherwise, they stay off chain and pay ρv_0 in the current period and expects to pay lower collusion fees ($E_v^c[f(v)]$) in future. The collusion is sustained if the future punishment would lead to larger lifetime fee payments compared to no deviation.

$$f_c + \frac{\delta}{1-\delta} * E_v^0[f(v)] \geq \rho v_0 + \frac{\delta}{1-\delta} * E_v^c[f(v)] \tag{56}$$

where,

$$E^c_{v\sim U[0,V]}[f(v)] = (1 - \frac{v_0}{V_{max}}) * f_c + \int_0^{v_0} \rho \frac{v}{V} dv \tag{57}$$

and

$$E^0_{v\sim U[0,V]}[f(v)] = \underbrace{(1 - \frac{v_0}{V}) * f_0}_{\text{On Chain Fee}} + \underbrace{\int_0^{v_0} \rho \frac{v}{V} dv}_{\text{Off Chain Fee}} \tag{58}$$

This simplifies to,

$$\delta > \frac{1}{1 + \frac{n_F}{N}} \; ; \quad \text{or} \quad n_F > N \times (\frac{1-\delta}{\delta}) \tag{59}$$

Interestingly this condition on the discount factor δ is independent of the equilibrium choice of f_c ($< f_0$). In fact $f_c = 0$ is pareto optimal collusion fees for all users. This condition captures the intuition that the user collusion is averted if - (i) the block capacity is extremely small compared to the demand. This means that the marginal user v_0 strongly prefers to get on-chain now, instead of waiting for the low likelihood event of being one of top $n_F << N$ users in the near future. (ii) users

transact infrequently (small δ) i.e. the demand N every period is made up of payment needs for a small fraction of overall users. (iii) user payment value v is sampled once instead of being randomly sampled from $[0, V_{max}]$ every period. Small value users are always small and large value are always large. If none of these three settings hold then user collusion at zero fees is trivial.

Theoretically, colluding miners can respond to this user collusion by only including payments that bid $f_h = \alpha_h \rho V/2$. Remember that f_h was the fee offered when all miner collude to add revenue maximizing blocks at half the demand $n_P = N/2$. Since both users and miners are forward looking, both are willing to forego payments via Blockchain and revenue from Blocks respectively for a few periods. The relative values of discount factors (δ, δ_m) and collusion savings would determine which side comes out on top in threatening the other into deviating from collusion. We skip delving deeper into this formulation because of limited evidence of forward looking user fee biding or strategies - counter strategies between users and miner. Future research could delve into some related questions - (i) Would forward looking (even colluding) users account for security risks and pay fees higher than the outcome of competitive auction? (ii) Would large users that comprise say 20% of all payment demand (e.g. major Bitcoin-USD exchanges) have strategies to unilaterally alter fees, delay and security to their advantage?

C. User Waiting

Our primary model assumes that users are impatient. They bid on chain and wait a maximum of one block period for their payment to be included. If not, they complete the payment off chain. In this section we ground this assumption to rational user behavior. Note that we originally modeled user payment values as continuously distributed in $[0, V]$. This was done to simplify the exposition of our research question. Here we take a more realistic set up wherein, N users that arrive every period have payment that take on one of V possible discrete values $1, 2, ..., V$. There are exactly N/V users at every payment value $v \in 1, 2, ..., V$. The discrete levels can be arbitrarily closely spaced. Similarly, fee bids are offered in discrete increments with ϵ (e.g. 1 cent or 1 satoshi) being the smallest increment. We will show a stable equilibrium where no user makes an on chain bid that results in a wait longer than single block period. Users either bid rationally expecting to be included on the immediate block or they go off chain.

At the outset let us assume that n_w users are present in the waiting queue, while N new users arrive in any given period. Every period miners add a block on chain, including top n_F $(< N + n_w)$ payments by fee offers. We want to find an equilibrium bid function $f^*(v)$ and the expected wait time $w^*(v)$ for a user with payment value v. Let v_0 be a payment value such that,

$$N\frac{V - v_0}{V} \leq n_F \leq N\frac{V - v_0 + 1}{V} \tag{60}$$

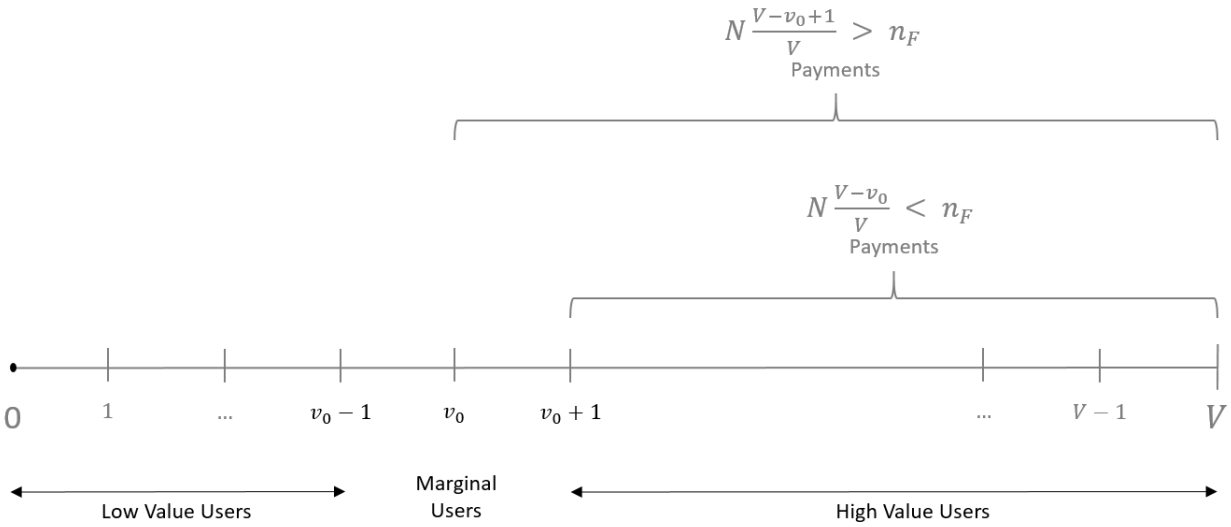

Figure 23 **Every block period N user payments arrive distributed uniformly over values $1, 2, ..., V$. Users with payment value v_0 are alluded to as marginal users. Some of these users but not all can find space on the immediate block the rest may have to either go off chain or remain in waiting queue. Higher value users ($v \geq v_0 + 1$) always find space on the immediate block, lower value users ($v \leq v_0 - 1$) never find space on any future block.**

Figure 24 shows corresponding three segment of users - high value, low value and marginal users.

Let $\bar{f}(v, w)$ be the maximum fees a user is willing to bid for a payment of value v and a wait time w. At this maximum fees the user is indifferent between on chain and off chain proportional fees. $\bar{f}(v, w)$ is naturally increasing in v, we investigate equilibrium fee bid function $f^*(v)$ that is monotonically increasing in v since a user with larger payment value has a higher willingness to bid owing to costlier off chain option ρv and greater per period delay cost δv. All **high value** users $v \geq v_0 + 1$ are in top n_F new payment value arrivals, they offer an equilibrium fee higher than all users with payment values $v \leq v_0$. They do not need to wait i.e. $w^*(v) = 1 \ \forall \ v \geq v_0 + 1$. Note that none of the bids in the waiting queue belong to payment value $v \geq v_0 + 1$, thus do not compete with these high value users anyways.

All **low payment value** users $v \leq v_0 - 1$ are outside the top n_F new payment value arrivals. They are competing against more than n_F higher value users in this period. In equilibrium, where fee offers are monotonically increasing, these users do not expect to find space on the immediate block. Even if they bid and wait in queue, they will be competing against more than n_F higher value payments in all future periods. Their attempt to wait in queue will be futile in perpetuity. All these low value users are better off going off chain.

Finally, all **marginal** users with payment value $v = v_0$ have a probabilistic shot at getting into top n_F bids. Let n_a $(= N/V)$ be number of users with payment value v_0 that arrive every period. Let n_w be number of users with payment value v_0 that have been waiting in queue for at least one period. n_d $(= n_F - N(V - v_0)/V)$ is the remaining capacity on any given block after the miner has included all payment bids with $v \geq v_0 + 1$. Only n_d out of $(n_a + n_w)$ can get onto the immediate block. New arriving users can either go off chain or place a bid on chain for these limited spots. We search for a mixed strategy equilibrium where these users randomize i.e. a fraction of these users $n_{a,on}$ place an on chain bid, while the remaining go off chain $n_{a,off}$ $(n_a - n_{a,on})$. If equilibrium $n^*_{a,on}$ or $n^*_{a,off}$ turn out to be zero, it collapses to a pure strategy.

In steady state, the total number of users vying for an on chain spot $(n_{a,on} + n_w)$ must equal the users that are included plus the users that are left waiting $(n_d + n_w)$ i.e. $n_{a,on} = n_d$. Thus in equilibrium n_d out of n_a users with payment value v_0 make an on chain fee offer $f^*(v_0)$ while the rest $n_a - n_d$ go off chain. Figure 24 shows a single block period arrival and payment completions of these marginal users $v = v_0$. The expected waiting time for these users is given by,

$$w^*(v_0) = 1 \times \frac{n_d}{n_w + n_d} + 2 \times \frac{n_w}{n_w + n_d}\frac{n_d}{n_w + n_d} + 3 \times (\frac{n_w}{n_w + n_d})^2 \frac{n_d}{n_w + n_d} + ... \tag{61}$$

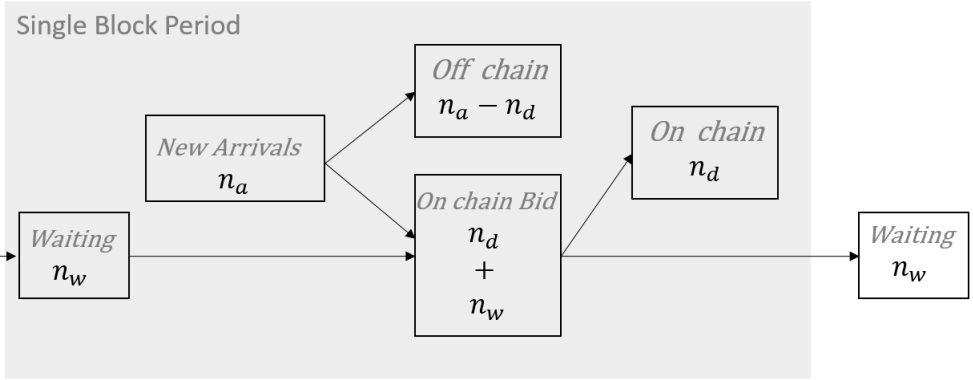

Figure 24 **Every period starts with n_w waiting on chain bids and n_a new payment arrivals. Some the new arrivals ($n_a - n_d$) go off chain, while the rest (n_d) make an on chain bid. Miners pick n_d random payments from the total $n_d + n_w$ on chain bids. This leaves n_w payments waiting for the next block period.**

From the setup above, we investigate equilibrium in following strategy space with unknowns (f^*_h, f^*_m, f^*_l) resulting in steady state waiting queue length (n^*_w) and waiting time (w^*),

$$f^*(v) = f^*_h, w^*(v) = 1 \text{ where } v \geq v_0 + 1 \tag{62}$$

$$f^*(v) = \begin{cases} f^*_m \text{ with probability } \frac{n_d}{n_a}, \\ f^*_l \quad \text{with probability } \frac{n_a - n_d}{n_a} \end{cases} , w^*(v) = w^* \text{ where } v = v_0 \tag{63}$$

$$f^*(v) = f^*_l, w^*(v) = \infty \text{ where } v \leq v_0 - 1 \tag{64}$$

This equilibrium is stable if,

- High value users $v \geq v_0 + 1$ bid greater than all other users and are better off than the off chain option.

$$\bar{f}(v_0 + 1, 1) > f_h^* > f_m^* > f_l^* \tag{65}$$

- Low value users $v \leq v_0 - 1$ are better off with the off chain option rather than competing with marginal or high value users.

$$\bar{f}(v_0 - 1, 1) < f_m^* < f_h^* \tag{66}$$

- Marginal user could deviate to a higher bid $(f_m^* + \epsilon)$. By doing so, they overcome all other waiting or new arrival marginal users who all bid f_m^*. Thus guaranteeing inclusion on the immediate block without additional wait $(w = 1)$. This deviation is not profitable if the increased bid is greater than maximum willingness to bid for an immediate inclusion $\bar{f}(v_0, 1)$.

$$f_m^* + \epsilon > \bar{f}(v_0, 1) \tag{67}$$

  Further, marginal user $v = v_0$ must bid less than or equal to their maximum willingness to bid $(\bar{f}(v_0, w^*))$ at a wait of w^*.

We also know that $\bar{f}(v_0, w^*) \leq \bar{f}(v_0, w = 1)$ since user always willing to bid more for minimal waiting $w = 1$. All the constraints above can be written as,

$$\bar{f}(v_0 + 1, 1) > f_h^* > f_m^* + \epsilon > \bar{f}(v_0, 1) \geq \bar{f}(v_0, w^*) > f_m^* > \bar{f}(v_0 - 1, 1) \tag{68}$$

This condition is satisfied for an arbitrarily small ϵ for following equilibrium strategy,

$$f_h^* = \bar{f}(v_0, 1) + \epsilon \quad ; \quad f_m^* = \bar{f}(v_0, 1) \quad ; \quad f_l^* = \bar{f}(v_0, 1) - \epsilon \tag{69}$$

In this equilibrium, marginal users randomize between on chain and off chain payments with probability n_d/n_a and $1 - n_d/n_a$ respectively. But when bidding on chain they do not expect any wait time $(w^* = 1)$ or any waiting queue $(n_w = 0)$.

Underlying the formulation above is the simple intuition that the marginal user is competing against all other marginal users in the current block period as well as marginal users that arrive in future period. If they bid anything less than full willingness to pay $\bar{f}$ than they will be superseded by the competition in this period as well as all subsequent periods in perpetuity. Note that this happens in particular because number of arriving users N and block capacity n_F are not stochastic. Waiting queues will emerge if these are considered to be stochastic. This is in-fact the case with Huberman et al. 2019 and Easley et al. 2019, who incorporate stochasticity but assume that users bid without knowledge of the exact bids in the waiting queue, instead relying on expectations of average waiting bids.

D. User Security Utility

In this section we discuss alternative payment value distribution and security cost distribution. We show in Section 4.1 the largest payment secured V_{secure} when the demand is made up on $N(d, V)$ payments uniformly distributed in $[0, V]$. Now we contrast with a setting where demand is made up of $N(d, V)$ payments exponentially distributed in $[0, V]$. The left half in Figure 25 shows two different payment distributions. The right half in Figure 25 shows the largest payment secured when demand is made up of $N(d, V)$ payments distributed uniformly or exponentially in $[0, V]$. In case of uniform distribution, the largest payment secured increases linearly with payment upper bound V until V_{max}. Participation of large value payments both raises and needs greatest level of security.

This is not the case with exponential distribution of payment demand. Increase in revenue with larger payments grows slowly because there are fewer large payments. This revenue growth may not be sufficient to provide security to increasingly higher payments. A middle segment of user with high enough fee savings and low enough security risk may have greatest willingness to participate on the Blockchain. Contemporaneous work by Chiu and Koeppl 2017, consider settings where a large number of small payment demand are compared against small number of large payment demand. Not surprisingly, such a setting resembles exponential distribution and leans in favor of the former i.e. small payments.

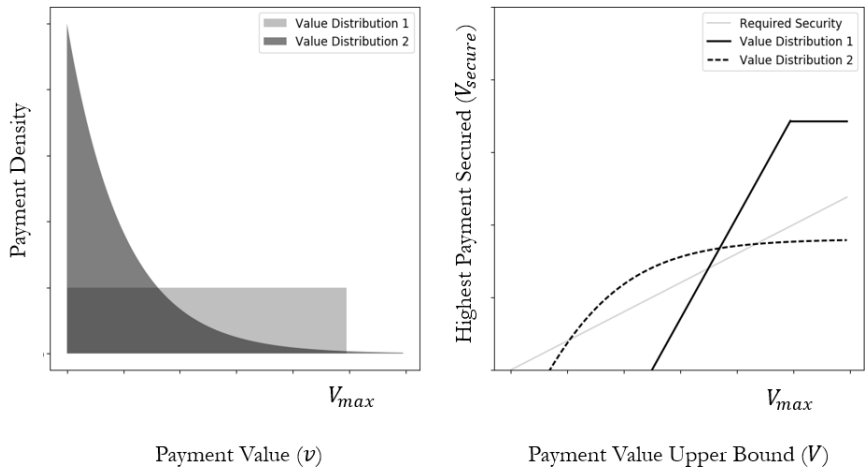

Figure 25 **The left figure shows two payment value distributions uniform ($U[0, V_{max}]$) and exponential ($\lambda e^{-\lambda v}$). The right figure plots highest payment secured when the demand is made of payments in $[0, V]$ following the two distributions. In the region where highest payment secured is less that required security, at least some of the large payments in $[0, V]$ are insecure.**

Beside the payment value distribution, we also make an assumption on the functional form of the security cost. In Section 3.1 we assume only transactions with value $v \leq V_{secure}$ are safe from a

double-spend attack when miner revenue is R. We go on to endogenize this upper bound V_{secure} in Section 4.1. Throughout the analysis we assume a binary security cost i.e. payments above V_{secure} are certain to be double spent and payments below V_{secure} are certain to be safe. This happens if users have knowledge of specific adversary with power θ. In practice, users may only have beliefs of adversary power i.e. a distribution over possible adversary power $P(\theta)$. The probability of a payment v double spent will be,

$$\int_0^1 \mathbb{1}\left[v > R\frac{(1-\theta)^2}{1-(1-\theta)^2}\right] \times P(\theta)d\theta, \tag{70}$$

where $\mathbb{1}[\cdot]$ is an indicator variable that takes value 1 when the condition inside it is true, and value 0 otherwise. Thus the security cost takes a continuous form instead of binary,

$$S(v) = \int_0^1 v \times \mathbb{1}\left[v > R\frac{(1-\theta)^2}{1-(1-\theta)^2}\right] \times P(\theta)d\theta \tag{71}$$

We defined $\bar{f}$ as the maximum on chain fees a user is willing to offer in order to avoid off chain proportional rate i.e. f where $U_{off-chain} = U_{on-chain-included}$.

$$\bar{f} = \alpha_h \rho v - S(v)/\delta \tag{72}$$

Under a binary security cost $\bar{f}$ is monotonically increasing for $v \in [0, V]$ i.e. users with large value payment stand to gain most by avoiding the off chain channel. This results is an outcome where Blockchain is used by largest value payments that crowd out smaller payments. $\bar{f}$ is monotonically increasing for a security cost distributions if,

$$\frac{\partial S(v)}{\partial v} \leq \delta\alpha_h\rho \; ; \quad \forall \quad v \in [0, V] \tag{73}$$

This is trivially satisfied for a binary distribution, but may not be true for all security cost distributions. If not satisfied, a middle tier of values will have the greatest willingness to pay for payment via Blockchain, instead of the highest tier of values.

Figure 26 shows three possible security cost distributions - binary distribution ($S^1(v)$) similar to one considered in the main text, and two continuous distributions ($S^2(v), S^3(v)$). All these security cost distributions take a monotonically increasing form since larger payments offer greater double spend incentive for any adversary θ. Distributions $S^1(v)$ and $S^2(v)$ satisfy (73) because the largest value $v = V$ still has the highest willingness to pay after accounting for the security cost. This can be interpreted from the figure as the gap between the (fee savings - delay cost) and the security cost. Distributions $S^3(v)$ does not satisfy (73), in fact a different payment value in the middle has the largest gap or the largest willingness to offer Blockchain fee ((fee saving - delay cost) - security

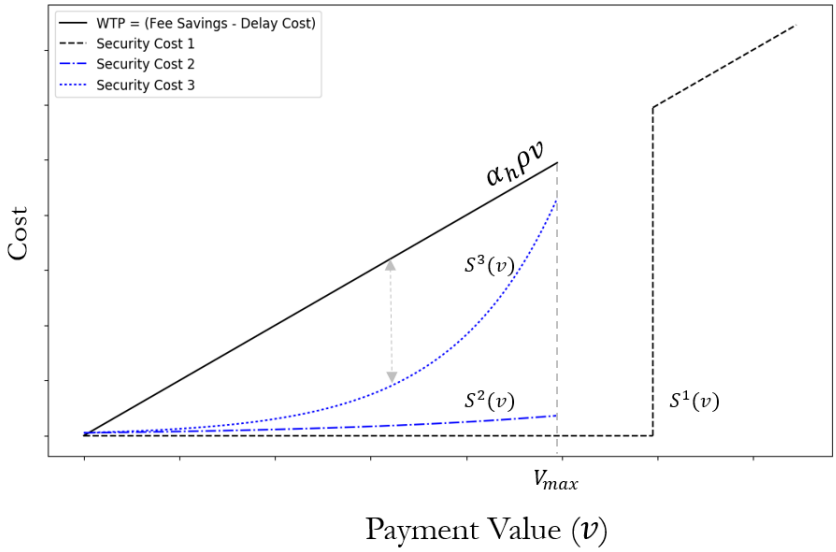

Figure 26 **The willingness to pay on chain incorporates fee savings and delay. This is compared against security cost $S(v)$ with three different formulations. $S^1(v)$ is the binary formulation used in the main text. $S^2(v)$ and $S^3(v)$ are two alternative formulations.**

cost). Such a security cost distribution would change our analysis such that a middle tier of values will get preferential treatment for payment via Blockchain.

Alternative models for payment value distribution or security cost distribution do not change the primary insight. Users are heterogeneous (off chain fee, delay, security risk) in willingness to pay. A group of strategic miner can benefit from creating artificial capacity constraints to keep out low willingness to pay users. A re-design to weaken the miner, reduces the security and thus all users willingness to pay. The double edged sword of collusion and security threat keeps the Blockchain accessible to only a fraction of the overall payment demand.

E. Users Costs and Block Reward

The total cost borne by Bitcoin users can be categorized into the following three components.

On Chain Transaction Fee $= \int_{v_0^*}^{V} f_0^* \frac{N}{V} dv = n_F f_0^*$

Off Chain Proportional Fee $= \int_{0}^{v_0^*} (\rho v) \frac{N}{V} dv$

Delay Cost $= \int_{v_0^*}^{V} (1-\delta) v \frac{N}{V} dv$

Figure 27 shows block size n_F varied between 0-100% of demand N as a protocol design choice. This changes equilibrium outcomes v_0^* and f_0^*. The left figure shows three components of total user costs – Transaction Fee, Off Chain Proportional Fee and Delay dis-utility. The right figure shows all three components combined into total costs with Blockchain, compared against total cost in a

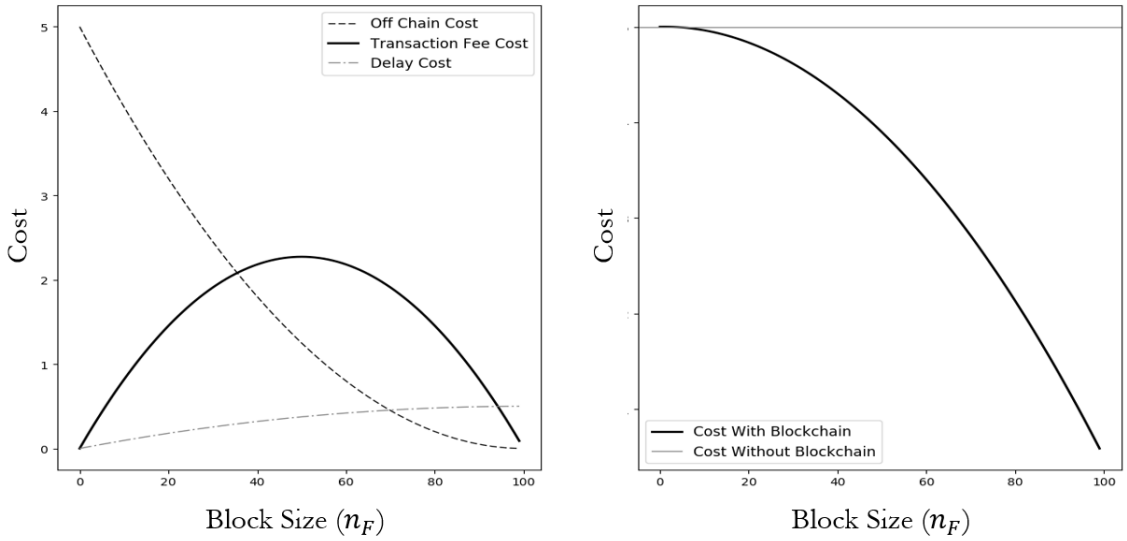

Figure 27 **The left figure shows three components of total user costs against block size n_F. The right figure shows the total cost (sum of three components), compared against payment costs in a world without Blockchain option.**

world without Blockchain option. The combined cost always dominates a setting without Blockchain i.e. more payment channel options are strictly better for users. Figure 28 shows block duration d on x-axis varies between 0.5 to 1.5 times the current duration $d = 10$ minutes. We assume that an increase in duration d, linearly decreases demand N in every Block period and linearly decreases the discount factor δ. A cursory analysis indicates that user surplus is greatest at very high block size n_F or very low block duration d, but we caveat this with collusion and security issues to be discussed in later section.

Note that we model users choices of channel determined by fees, delay and security. We made two implicit assumptions in doing so - (i) Users do not have a need for exchanging native Blockchain crypto currency for fiat currency. In our model, all users draw periodic payment needs from the same distribution. We assume that all users carry sufficient stock of crypto currency to send and receive payments without needing to exchange from fiat currency. Since all users are homogeneous in their long term payment needs, all of them hold the same average stock of crypto currency. (ii) Users do not face any inflation in crypto currency prices relative to fiat currency on their coin stock. In practice, coinbase Block rewards B distributes additional currency supply to miners. Given constant demand for payments, an increasing supply of coins leads to inflation. Effectively all users pay a combined B in rewards to miners every period. This would be a fourth cost component beside fees, delay and security.This reward is proportional to gross activity or total value of payments on the

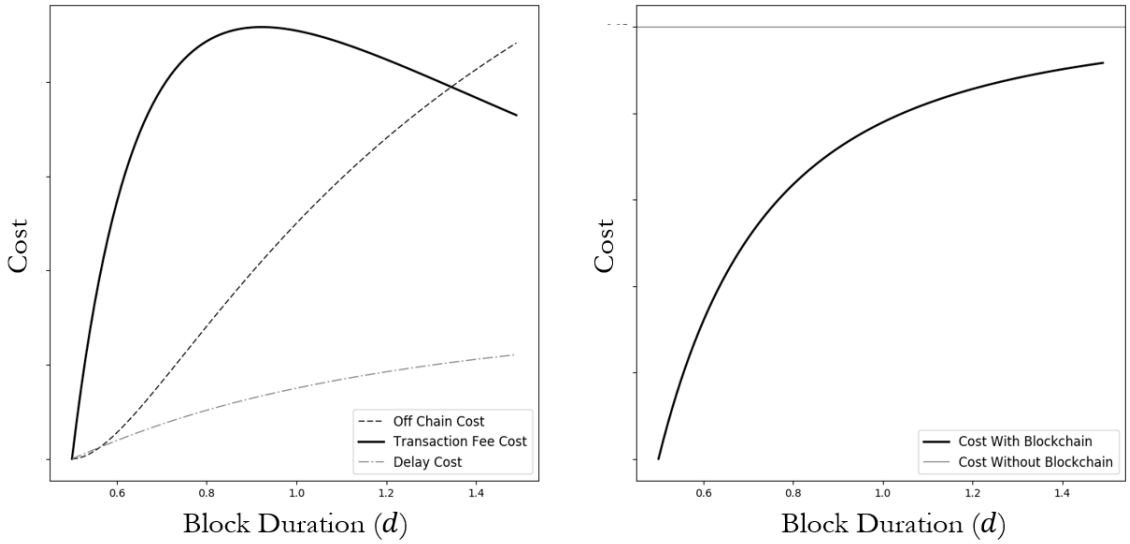

Figure 28 **The left figure shows three components of total user costs against block duration d. The right figure shows three components combined, compared against payment costs in a world without Blockchain option.**

platform,

$$B = k \int_{v_0^*}^{V} v \frac{N}{V} dv = \frac{kN}{2V}(V^2 - {v_0^*}^2) \tag{74}$$

The proportionality constant k captures markets inherent currency velocity i.e. number of payments where the same coin is used and inflation level or block reward level set on the protocol.

In our model, since all users are homogeneous in their long term payment needs, all of them hold the same average stock of crypto currency and thus face the same block reward inflation cost. Individual user is paying B/N in inflation cost to miners every period in addition to transaction fees. Users pays this per period average cost of holding inflationary crypto currency even if they happen to use the fiat channel for payment in a given period. Our primary model found that a setting with additional payment channel option is strictly better for users compared to a no Blockchain world. This is not necessarily the case anymore where we account for Block Reward Inflation cost in addition to fees, delay and security costs. Figure 29 is similar to an earlier Figure 27, albeit with this additional fourth cost component. A rational user will keep stock of coins for regular payments via Blockchain channel if these combined four cost components are strictly better for users compared to a no Blockchain world.

This leads to an additional constraint,

$$Cost_{\text{on chain fees}} + Cost_{\text{off chain fees}} + Cost_{\text{delay}} + Cost_{\text{block reward inflation}} \quad > \quad Cost_{\text{off chain only}} \tag{75}$$

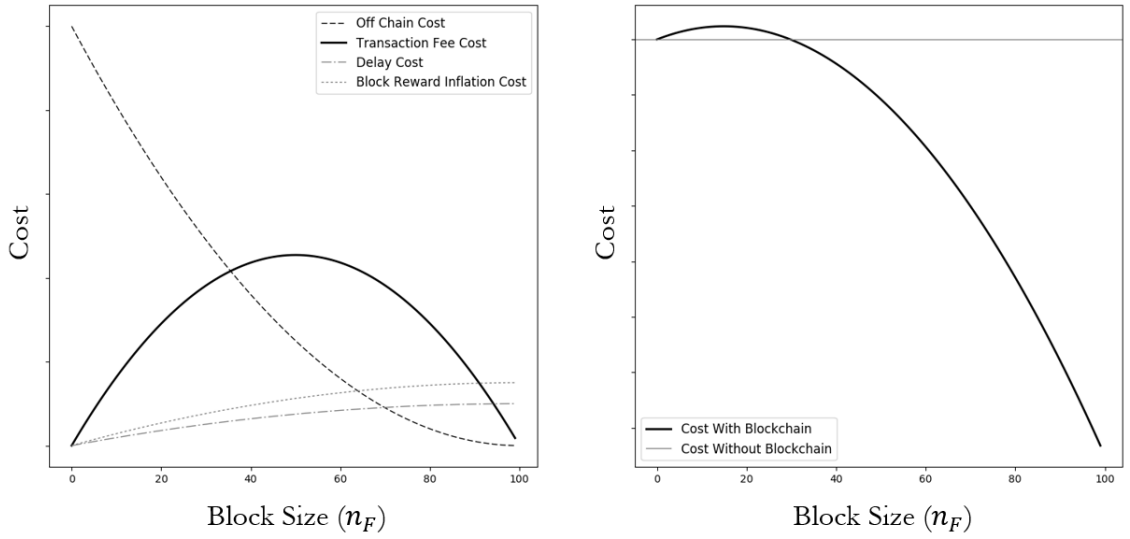

Figure 29 **The left figure shows four components of total user costs against block size n_F. The right figure shows four components combined, compared against payment costs in a world without Blockchain option.**

$$\int_{v_0^*}^{V} f_0^* dv + \int_0^{v_0^*} \frac{\rho v}{V} dv + \int_{v_0^*}^{V} \frac{(1-\delta)v}{V} dv + \int_{v_0^*}^{V} \frac{kv}{V} dv \quad > \quad \int_0^{V} \frac{\rho v}{V} dv \qquad (76)$$

Non zero Block Rewards ($k > 0, B > 0$) may have total costs greater than a no Blockchain world. This happens at very low Block capacity n_F because, users are required to keep a stock of periodically inflating native coins for infrequent payment needs. Figure 30 shows user costs and miner revenues as block reward inflation levels are varied on the protocol between $k = 0\%, 1\%, 2\%$. Miner revenues increase with higher Block Reward. A group of colluding miner now prefer a block size higher than before ($n_P > N/2$) where they earn larger revenues thus raising security. The overall impact for users is mixed - negative impact of additional cost component but positive impact of greater collusion capacity and greater security.

While it is not the focus of our work, future researcher can venture further in direction to find optimal design balance between Block rewards and capacity. In fact its possible that very high Block Rewards and near zero transaction fee (n_F or $n_P < N, f = 0$) are overall more optimal in certain market parameter ranges (e.g. N, V, ρ). In-fact such a design is proposed by Chiu and Koeppl 2017. Transaction fee in our model are determined by competitive auction i.e. entry of a large value user increases bids by all other users. Chiu and Koeppl 2017 set transaction as an exogenous proportional rate i.e. entry of a large value user only adds a small fee paid by that individual user. Under this assumption they show that high Block Rewards and near zero transaction fee (n_F or $n_P < N, f = 0$) dominate. Naturally, such a design will need a different mechanism to prioritize payments without auctions. Since Block capacities are limited and miner endogenously prioritize payments via the

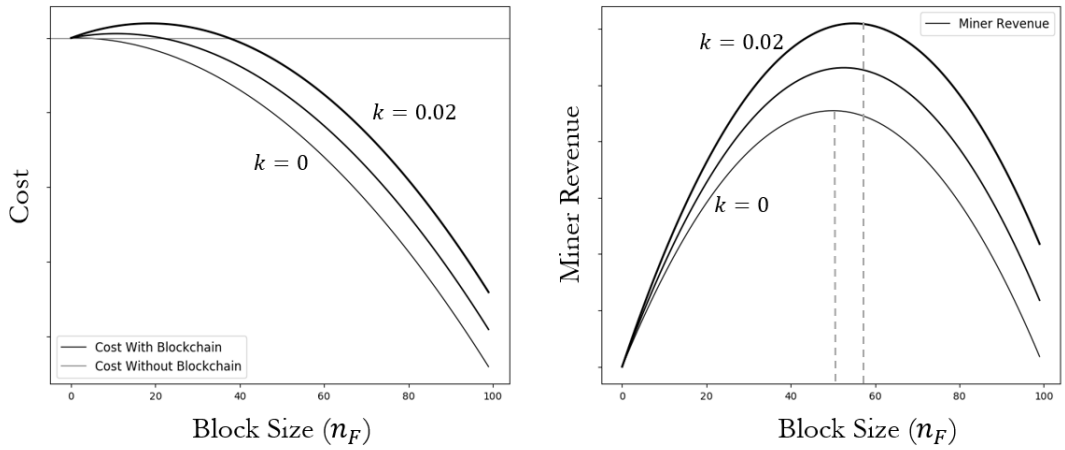

Figure 30 **The left figure shows four components combined against total user costs in a no Blockchain world. As Block rewards increase ($k = 0\%, 1\%, 2\%$), the overall cost for users are higher at the same block capacity n_F. The right figure shows total miner revenues (auction fees and block rewards). As Block rewards increase ($k = 0\%, 1\%, 2\%$), colluding miner will prefer an increasing partial fill level. With increasing Block rewards users face greater costs but less severe collusion prospects.**

auction mechanism in most existing p2p Blockchain designs, we would argue that our work caters more to current market conditions and popular design paradigms.

F. Double-Spend Attack with Stake in Bitcoin

In this section, we extend our model of double-spend attack to consider attacking miners disincentives from investments in stock of - Bitcoin and Bitcoin mining hardware. The magnitude of attack is represented by attacker's mining power and the value of double spend. We also discuss how does the magnitude of attack changes user payment activity and, therefore, Bitcoin value.

Let B represent the market price of Bitcoin (per unit). After a double-spend attack the market price of Bitcoin drops by a factor μ, $\mu \in [0, 1]$. Thus, in an event of a double-spend attack, the market price of Bitcoin will be $(1-\mu)B$, after the attack. The underlying driver for drop in Bitcoin value is the reduced gross activity on the platform. We briefly introduce market price of Bitcoin in Appendix E. We base market price of Bitcoin on gross activity or total value of payments on the platform. The total value of payments on the platform depends on range of payment values (say $[0, V]$) that are executed on the platform. Since large payment values are under greatest threat from a double spend attack, the maximum viable payment value (say V) depends on the level of security. A successful double spend attack would lower confidence on security. This lowers the maximum payment users are willing to accept on Bitcoin and results in lower gross activity on the platform.

The attacking miner deploys purchased and rented mining equipment with combined power θ. Purchased mining power is a permanent investment and would be worthless outside of Bitcoin

mining, therefore it dis-incentivizes an attack. Rental mining power is temporary, therefore does not dis-incentivizes an attack. The attacking miner owns θ' fraction of Bitcoins. These Bitcoins would also lose value after an attack, thus it dis-incentivizes an attack. Let σ represent the total number of Bitcoins in the system. The attacking miner has $\sigma\theta'$ Bitcoins. Since each Bitcoin has a market value of B, the attacker's total Bitcoin capital is $\sigma\theta' B$. If a double-spend attack happens, a miner with power θ will lose $\sigma\theta' B\mu$ of Bitcoin capital. The attacker's incentive to attack is highest if his/her stock of Bitcoin $\theta' = 0$ and all his/her mining equipment θ is rented. We will examine attacking miner's payoffs at both extremes - lowest incentive to attack with stake in Bitcoin and Bitcoin mining hardware, and greatest incentive to attack with no stake.

Incentives with Stake: Similar to our analysis in Section 4, we want to compare miners' payoff from carrying out a double-spend attack with payoffs from honest mining. We define $\bar{\mu} = 1 - \mu$. As derived in Section 4, $[1-(1-\theta)^2]$ is the probability of successfully carrying out a double-spend attack by a miner with hashing power θ, v is the value of transaction being attacked and R is the mining reward earned by a miner for successful mining. Thus, $v + R$ is the total reward earned in a double-spend attack. The cost of mining is represented by c, and the term $\sigma\theta' B\mu$ represents the loss of Bitcoin capital due to the double-spend attack. Also, note that after the double-spend attack the revenue from each successful mining is $R\bar{\mu}$, because the value of Bitcoin decreases by a factor of μ after the double-spend attack.[20]

The miner's payoff from the attack can be written as,

$$\pi_{\text{double spend}} = \underbrace{[1-(1-\theta)^2]}_{\substack{\text{Probability}\\ \text{of success}}}(v + R - \underbrace{\sigma\theta' B\mu}_{\substack{\text{Loss of}\\ \text{Bitcoin}\\ \text{value}}}) - c + \underbrace{\delta_m(\theta R\bar{\mu} - c) + \delta_m^2(\theta R\bar{\mu} - c)}_{\text{Payoff from mining after attack}} + \dots \tag{77}$$

The miner's payoff from honest mining can be written as,

$$\pi_{\text{honest}} = R - c + \underbrace{\delta_m(\theta R - c) + \delta_m^2(\theta R - c)}_{\text{Payoff from mining after attack}} + \dots \tag{78}$$

To dissuade an adversary from performing a double-spend attack, we need $\pi_{\text{honest}} \geq \pi_{\text{double spend}}$. From the above expressions of π_{honest} and $\pi_{\text{double spend}}$, this condition simplifies to

$$v \leq V_{secure},$$

where

$$V_{secure} = \frac{R}{1-\bar{\theta}^2}\left[\bar{\theta}^2 + \frac{\theta\mu\delta_m}{1-\delta_m}\right] + \sigma\theta' B\mu. \tag{79}$$

[20] In practice, the mining revenue and the Bitcoin capital might be impacted by different values of μ. For simplicity, we assume they are impacted by the same value of μ.

We know that all the transactions are in $[0, V_{max}]$, and there are a total of N_{max} transactions in this range. Thus, if we want all the transactions to be secure from a double-spend attack, then we need

$$V_{max} \leq V_{secure}.$$

Substituting the value of V_{secure} from the above equation and using

$$R = \rho\alpha_h(1-\gamma)\gamma VN,$$

we have

$$V_{max} \leq \frac{(N_{max} - n_F)}{\eta(\theta)\bar{\theta}^2 N_{max}} n_F D_1 + \sigma\theta' B\mu, \tag{80}$$

where $\eta(\theta) = \frac{1}{\rho\alpha_h}\frac{1-(1-\theta)^2}{(1-\theta)^2}$ and $D_1 = V_{max}\left[\bar{\theta}^2 + \frac{\theta\mu\delta_m}{1-\delta_m}\right]$. Define $D_2 = \eta(\theta)\bar{\theta}^2 N_{max}(V_{max} - \sigma\theta' B\mu)$. The above inequality can be written as follow.

$$n_F^2 D_1 - n_F D_1 N_{max} + D_2 \leq 0. \tag{81}$$

This is satisfied for

$$n_F \in [N_{max} - (n_F)_{max},\ (n_F)_{max}] \quad ; \qquad (n_F)_{max} = \frac{N_{max}}{2} + \frac{N_{max}}{2}\sqrt{1 - \frac{4D_2}{N_{max}^2 D_1}}. \tag{82}$$

The above condition suggests that to secure the Bitcoin transactions from a double-spend attack, we need to choose Bitcoin's capacity n_F in the range $[N_{max} - (n_F)_{max},\ (n_F)_{max}]$. In other words, there is an *economic limit* on the capacity of Bitcoin, and this capacity cannot be outside $[N_{max} - (n_F)_{max},\ (n_F)_{max}]$.

Also, note that $(n_F)_{max}$ will not be a real number if

$$1 - \frac{4D_2}{N_{max}^2 D_1} \leq 0.$$

In that case, there will not exist any real value of Bitcoin capacity that will be safe from a double-spend attack. In other words, it is impossible to secure Bitcoin from a double-spend attack.

Figure 31 pictorially depicts the parametric region where the Bitcoin's capacity is impossible to secure from a double-spend attack (the shaded region in both sub-figures). In both the sub-figures, we note that as μ increases, Bitcoin stays secure for a higher value of θ. This is because μ represents the loss in Bitcoin's value due to a double-spend attack. Thus, a high value of μ dis-incentivizes miners to carrying out a double-spend attack. Apart from this, comparing both sub-figures we also note that a higher value of N_{max} leads to a smaller region of parameters where it is impossible to

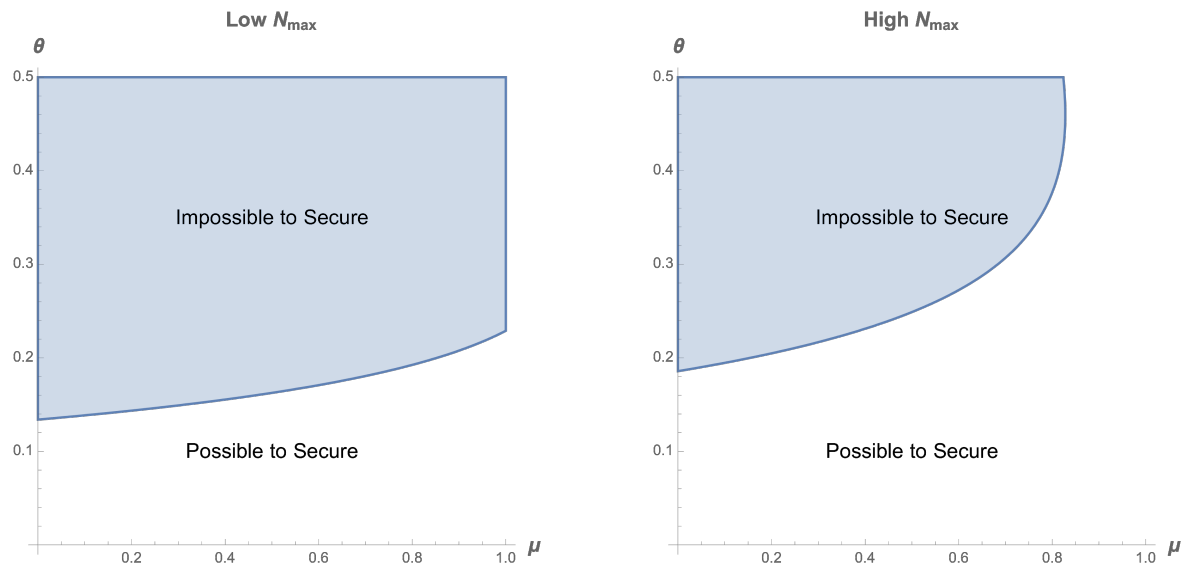

Figure 31 **As μ increases, Bitcoin stays secure for a higher value of θ. A higher value of N_{max} leads to a smaller region of parameters where it is impossible to secure Bitcoin from a double-spend attack.**

secure Bitcoin from a double-spend attack. This is because a higher value of maximum possible demand (N_{max}) leads to higher honest revenue. Thus, it incentivizes miners to continue earning honest revenue. Both the sub-figures qualitatively remain the same for low and high values of σ and B. This is because high values of σ and B mean a higher loss due to double-spend attack. Thus, it dis-incentivizes miners from conducting a double-spend attack[21].

Impact of Collusion on Double-Spend Attack: Figure 32 depicts the impact of collusion on the possibility of a double-spend attack. Comparing both sub-figures, we can see that the parametric region where double-spend attack can happen increases when collusion is suppressed. This is because suppression of collusion decreases honest revenue for miners. Thus, it decreases the incentive of miners to earn honest mining revenue. In both the sub-figures, we also see that the minimum mining power needed to carryout such a double-spend attack decreases after collusion is suppressed.

Incentives without Stake: The miners payoff from the attack simplifies to,

$$\pi^{noStake}_{\text{double spend}} = \underbrace{[1-(1-\theta)^2]}_{\substack{\text{Probability} \\ \text{of success}}}(v+R) - c \tag{83}$$

The miners payoff from honest mining simplifies to,

$$\pi^{noStake}_{\text{honest}} = R - c \tag{84}$$

[21] A higher Bitcoin holding and a large fraction of loss in Bitcoin value lower payoff from a double spend attack. Note, that the attacking miner can easily convert their Bitcoin to fiat currency before launching the attack. This would protect the attacker from any loss of value in their Bitcoin stake.

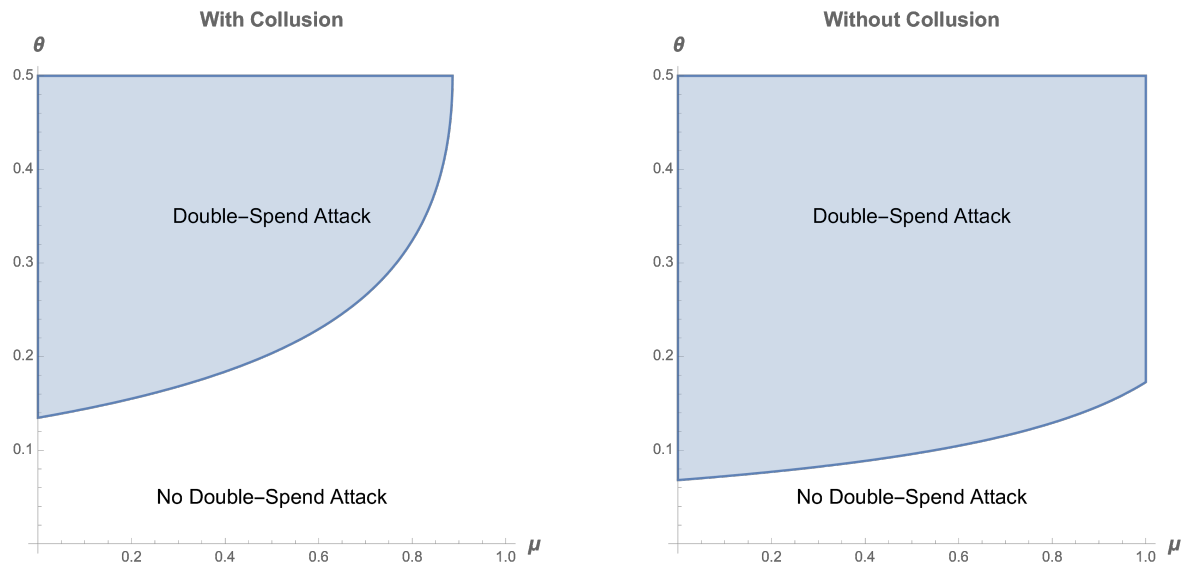

Figure 32 **As μ increases, Bitcoin stays secure for a higher value of θ. A higher value of N_{max} leads to a smaller region of parameters where it is impossible to secure Bitcoin from a double-spend attack.**

To dissuade an adversary from performing a double-spend attack, we need $\pi_{\text{honest}} \geq \pi_{\text{double spend}}$. From the above expressions of π_{honest} and $\pi_{\text{double spend}}$, this condition simplifies to

$$v \leq V_{secure},$$

where

$$V_{secure} = R\frac{\bar{\theta}^2}{1-\bar{\theta}^2} \tag{85}$$

We skip the subsequent derivation of feasible Bitcoin capacity $n_F \in [N_{max} - (n_F)_{max},\ (n_F)_{max}]$ and the analytical expressions for $(n_F)_{max}$. Qualitatively its obvious that range of feasible Bitcoin capacity would be narrower if the double spend attacker does not have stake in Bitcoin and its mining hardware.

Magnitude of Attack: In the double spend attack incentives above we represent magnitude of attack via attacker's mining power θ and the double spent payment value v. A large attacker's mining power θ means high probability of attack success $1-(1-\theta)^2$ and greater payoff from attack. Similarly, a large double spent payment value v means greater payoff from attack.

The analytical examination of double spend attack incentives play out in the real world in two ways. Let us first consider a scenario where the power of the attacker θ is not known by the users. The users have a guess of a potential attackers power θ and the implied maximum secure payment value V_{secure}. If this guess if underestimated by the users than they may accept a large payment on the Blockchain that is insecure. In this setting a double spend attack occur to reveal the actual

threat. The attack magnitude in terms of double spent value v also reveals the power of the adversary θ. The revealed lower level of security lowers the maximum payment value V_{secure} users are willing to accept on the Blockchain. We discussed some notable instances of these events in Appendix G.

The second scenario is one where the power of the attacker θ is common knowledge. Thus users have correct expectation of level of security (V_{secure}) to begin with. Then equilibrium level of maximum payment acceptable on Bitcoin is already in line with level of security ($V \leq V_{secure}$). An attempt to accept a payment of value $v > V_{secure}$ would invite a double spend attack. Accepting a payment of value $v > V_{secure}$ and the double spend attack are both on an off-equilibrium path. On Blockchains with lower miner revenues, the payment values should remain low under threat of double spend attack on larger payments. This explains why average payment value on Bitcoin is more than 25 times that of Bitcoin Gold (in USD, on 5th Sept 2021). The (average and maximum) payment values on various Blockchains remain in line with their corresponding level of security in spite of no explicit double spend attacks for years at a stretch. The common knowledge about power of the attacker θ is fairly plausible. Blockchain communities are typically aware of presence of very large miners or mining pools and their powers.

Thus magnitude of an actual attack or threat of an attack reduces the gross payment activity value on the platform and therefore value of the currency. A critical point captured in our model is that - a double spend attack does not lead to complete collapse of a Blockchain. This is also consistent with real world double spend attacks that have never led to a complete collapse. The double spend attacker is not risking a total destruction of Bitcoin value but just a depreciation. Qualitatively, it is not obvious if rewards from an attack are worthwhile given the depreciation in Bitcoin value. Our analytical model and the numerical analysis does reveal that rewards from an attack can more than cover for attacker's losses on Bitcoin value.

G. Notable double-spend attacks

Table 6 lists some double spend attacks reported on various proof of work Blockchains. One of these attacks occurred on Bitcoin Gold (BTG), which is a fork of Bitcoin launched to eliminate large ASIC miners. It has a substantially smaller value and demand from users compared to Bitcoin. In May 2018, double-spend attacks on payments worth approximately $18 Mn were reported on Bitcoin Gold. The attacker was successfully able to record transactions one two Bitcoin Gold chain forks. The first fork recorded transactions that sent BTG to an exchange, presumably to receive back a different cryptocurrency. The second fork recorded transactions that sent BTG to another one of the attacker's own accounts. After the successful completion of the attack the first fork was left out of the main chain. Thus, the exchange does not have a record of having received any BTG from the attacker. Exchanges are commonly targeted for double spend attacks because they operate

at a very high frequency. They allow the attacker to receive an alternative cryptocurrency quickly before anyone realizes the presence of the double spend fork. The attackers choice of alternative cryptocurrency tends to be one of the more secure Blockchains e.g., Bitcoin, Ethereum. So that the exchange can not cheaply reverse back delivery of the alternate cryptocurrency to the attacker.

Table 6 Notable double spend attacks on proof of work Blockchains.

Date	**Blockchain**	**Attack Value**
12th March 2013	Bitcoin	$9,800
18th May 2018	Bitcoin Gold	Upto $18Mn
April-May 2018	Verge	At least $1.7Mn
2nd June 2018	ZenCash	At least $550,000
5th January 2019	Ethereum Classic	At least $460,000
23rd January 2020	Bitcoin Gold	Upto $70,000
1st and 6th August 2020	Ethereum Classic	Between $1.68 to $3.3Mn
July 2020 - June 2021	Verge	Unknown

Exchanges or other merchants that are the target of these attacks often do not report being a victim of double spend attacks. In the examples in Table 6, merchant OKPAY reported being the victim of Bitcoin double-spend attack worth $9,800 in 2013. The exchange Bittrex reported being at least one of the victims of the Bitcoin Gold double-spend attack in May 2018. But more often than not, the victims do not report for the fear of being perceived as insecure. Therefore it is non trivial to verify a double-spend attack with certainty. Lovejoy (2020) analyse history for 23 cryptocurrencies in order to detect potential double spend attacks. They look for two identifiable features - a forking or re-organization of the Blockchain and presence of a pair of transactions one on each fork. This pair of transactions come from the same senders to two different receivers such that only one can exist on a single chain. For example, if account A contains 100 BTG then one transaction sends 100 BTG from A to B and the other transaction sends 100 BTG from A to A'. They detect 40 distinct forking or deep reorganization events across 6 coins (Verge, Expanse, Hanacoin, BTG, VertCoin, Litecoin Cash). Further they classify 18 of these as potentially involving double-spend transactions.

Double spend attacks are reported by Blockchain developers, as well as victims (merchants and exchanges). Attacks have also been detected with some degree of confidence by analyzing historical chain data. Both these facts suggest that Double-Spend attacks are not merely a theoretical construct, but very much viable in practice. They tend to occur on Blockchains when - (a) Coin value and therefore the honest mining rewards are low and (b) payment values that merchants and exchanges are accepting on the Blockchain are high.

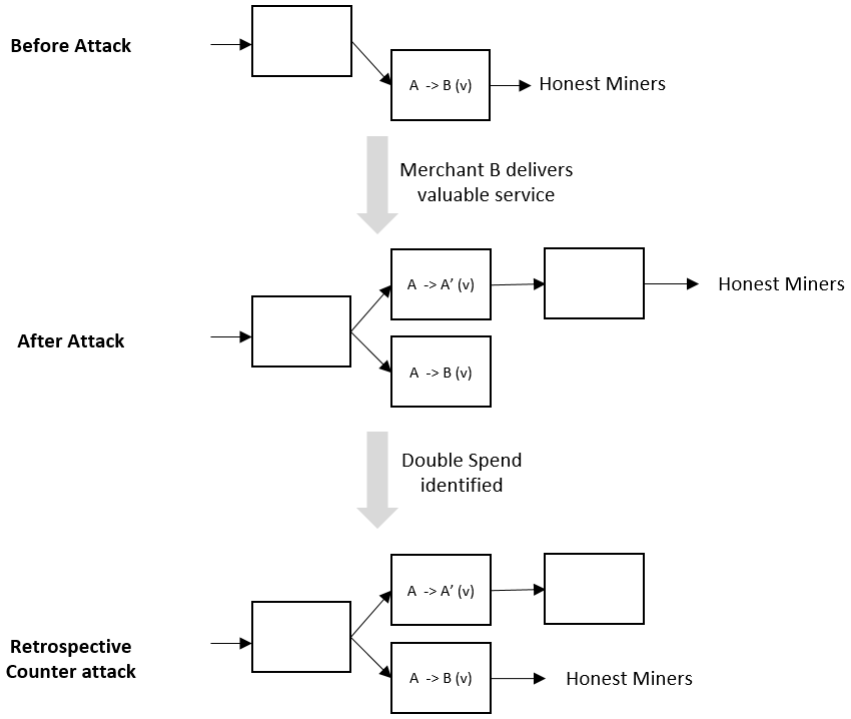

Figure 33 **(Top) State of Blockchain before the attacker reveals their malignant fork. (Middle) State of Blockchain after the attacker reveals their malignant fork. (Bottom) State of Blockchain when honest miners identify the double spend attack and attempt to counter by reverting to the original chain.**

H. Honest miner incentive to protect

A stream of literature argues that even if a double-spend attack is successful in the short run, it will be retrospectively reversed by honest miners. Savolainen et al. (2019) argue that such a reversal is easier given the organization of honest miners into large pools. Moroz (2020) make a similar argument that honest miners or the victims of the double spend attack can counter attack and reverse the attackers double spend. In a double spend attack, the attacker A records two transactions on two forks - one transaction from A to B sends coins to a merchant and the other transaction from A to A' sends coins to attacker's own account. Honest miners would ideally want to retrospectively make sure that the transaction from A to B is not left out of the main chain. This would ensure that the victim merchant B receives the funds. A simple strategy for honest miners holding majority power would be to ignore the longest chain which contains the transaction from A to A'. Instead the honest miners could collaborate towards adding Blocks to the shorter chain which contains the transaction from A to B. Given the majority mining power held by the collaborating honest miners, this fork would eventually exceed the length of the malignant fork.

It would seem that using this strategy honest miners can always ward off a double spend attack. If so, the adversary will never attack in the first place. This strategies for retrospective reversal of

double-spend attack rely on two key assumption - (a) the attacker keeps Bitcoin earned via the double spend attack on-chain and (b) the honest miners have on-chain information to differentiate between the malignant and genuine transaction. We examine both assumptions one by one.

Let us consider that the honest miners, who are in majority, can identify and collaborate to remove the chain of blocks starting from the block that had the double-spend attack transaction. For example, if the current block is t and double-spend attack transaction was included in the main chain in block $t-d$, then the miners can agree to remove blocks from $t-d-1$ on wards. That is, any transaction that was included in blocks $t-d$ to t will be invalidated. And the chain will be forked at block $t-d-1$. It would seem that miners can always go back and invalidate the double spend transaction. But, even such a retrospective forking may not discourage the attacking miner. Because, the attacking miner can get his Bitcoins converted to other coins or fiat currency in the time taken (d block durations) to identify the double-spend attack and carry out this retrospective fork. Retrospective correction of Blockchain transaction history can not overcome off chain events such as Bitcoin to fiat conversion.

Let us next consider that the double spend attack is identified by all honest miners immediately, without giving the attacker time to transact on-chain crypto-currency for off chain fiat currency. But, the honest miners do not know which fork contains the malignant transaction. This may seem trivial in our sequential description of a simple double spend attack. But, all the honest miners can observe is that account A has attempted to spend their balance in two transactions - one to A' and another to B on different forks. But, they can not authenticate that among accounts A' or B one belongs to the attacker and one to the victim. Note that the victim is the one that has provided cash or goods to the attacker offline. The honest miners do not have any on-chain record of this off-chain transfer of cash or goods. An alternative strategy would be for honest miners to retrospectively orphan both forks that contain the transactions A to B and A to A'. But, doing so does not help the merchant B since record of receiving payment from the attacker A is still left out of the main chain.

Thus retrospective (hard fork) correction of a double spend attack are limited because of off-chain events - (a) exchange of on-chain crypto-currency for off chain fiat currency and (b) exchange of off-chain goods and fiat cash between victim and attacker, with no on-chain record. Conditions where retrospective correction by honest miners works is when - (i) honest miners actively track duplicate spends across forks, (ii) they can agree to collaborate before the attacker has a chance to exchange their stolen cryptocurrency for a fiat currency and (iii) they can trust a credible and established merchant who has fallen victim to a double-spend attack. The last condition is perhaps the most tricky. We admit that if such off-chain credibility and trust is present, majority honest miners do

have a viable strategy to offer protection from double-spend attacks. A handful of very large and publicly known merchants and cryptocurrency exchanges do attain such credibility. However, such off-chain credibility and trust is an exception rather than the norm. Bitcoin's intended paradigm is to facilitate trust-less and anonymous payments via cryptographic validation. Under this paradigm, majority honest miners do not have a viable strategy to guarantee protection from double-spend attacks.

I. Readings on Bitcoin

Bitcoin's ecosystem is composed of four major components: (1) users, (2) miners, (3) the platform protocol, and (4) the cryptocurrency. We suggest Huberman et al. (2019) for a deeper understanding of user waiting queues and transaction fee decisions. Cong et al. (2018) provides an in-depth discussion of the arms race by miners for computing hardware and their organization into mining pools. The protocol itself is most accurately described by Satoshi Nakamoto – the creator of Bitcoin (Nakamoto 2008). Finally, Cong et al. (2018) can be used as a resource for a primer on cryptocurrencies and their adoption.