BLOCK SIZE LIMIT DEBATE WORKING PAPER

Reduce Orphaning Risk and Improve Zero-Confirmation Security With Subchains

Peter R. Rizun[†]
20 December 2015

Abstract. Orphaning risk for large blocks limits Bitcoin's transactional capacity while the lack of secure instant transactions restricts its usability. Progress on either front would help spur adoption. This paper considers a technique for using fractional-difficulty blocks (weak blocks) to build subchains bridging adjacent pairs of full-difficulty blocks (strong blocks). Subchains both reduce orphaning risk by propagating block contents over the entire block interval, and add security to zero-confirmation transactions due to the weak blocks built above them. Miners are incentivized to cooperate building subchains in order to process more transactions per second (thereby claiming more fee revenue) without incurring additional orphaning risk. The use of subchains also diverts fee revenue towards network hash power rather than dripping it out of the system to pay for orphaned blocks. By nesting subchains, weak block confirmation times approaching the theoretical limits imposed by speed-of-light constraints would become possible with future technology improvements. As subchains are built on top of the existing Bitcoin protocol, their implementation does not require any changes to Bitcoin's consensus rules.

KEY WORDS

1. Bitcoin. 2. Scaling. 3. Weak blocks. 4. Network security. 5. Instant transactions.

1. Introduction

Bitcoin's performance as a payment network is hardly impressive. In 2015, it processed an average of 1.4 transactions per second[1] while merchants waited on average eight minutes to receive initial verification from a miner.[2] In contrast, the Visa network processed over 2,000 transactions per second,[3] and—with chip-and-PIN technology—merchants received authorization and PIN-verification in under a second.[4] Unlike Visa, Bitcoin's transactional capacity is limited due to miners' hesitation to produce blocks containing large volumes of new transactions.[5] Such blocks propagate across the network slowly,[6] increasing the chances that the block is orphaned and the miner's reward is lost. Also unlike Visa, the initial verification of a transaction by a miner is delayed because blocks are propagated on average only every ten minutes,[7] rather than at a rate dynamically tuned to the bandwidth and latency of the network. In this paper, we present a scaling technique called *subchains* to build blocks layer-by-layer—at a small fraction of Bitcoin's ten-minute block time—thereby reducing both orphaning risk and the wait-time for the first verification of a transaction by a miner.

Throughout this paper, we make certain simplifying assumptions. In particular, we assume that:

(1) Miners are rational, short-term profit-maximizing agents.
(2) The network hash rate is constant over a given block interval.[8]
(3) Block information propagates with a well-defined impedance measured in time per bytes propagated.[6,9,10]

[†] P. R. Rizun (peter_r@gmx.com) is a physicist and entrepreneur from Vancouver, Canada.
The original PDF version of this document has been time-stamped in the Blockchain.
1BWZe6XkGLcf6DWC3TFXiEtZmcyAoNq5BW

(4) The free-market equilibrium block size is smaller than the protocol-enforced block size limit (if such a limit exists).

In Section 3, we describe the subchain technique,[11] which is a practical application of weak blocks[12,13,14,15] with the appropriate incentives to ensure that miners cooperate for the mutual benefit of the network. Its implementation requires neither a hard nor soft fork—but it does require participation from a significant fraction of the network hash power in order to be useful. We explain that a miner can include all of the subchain's transactions in his block candidate—and thus all of the subchain's fees—without incurring orphaning risk, and thus he is incentivized to work together with other miners to extend a single (highest-fee) subchain (Section 4). We then show that miners will only include *new* transactions that pay a fee greater than the transaction's marginal orphaning risk, thereby economically restricting the rate of subchain growth.

Certain investigators have argued that fees that result from orphaning risk do not contribute to network security. For example, Maxwell argued, "the fact that verifying and transmitting transactions has a cost isn't enough, because all the funds go to pay that cost and none to the POW 'artificial' cost."[16] With a simple diagram, we prove this line of reasoning false in Section 5 by showing that the fees already included in the subchain contribute *directly* to network security in the same way that the block reward does. Only the fees in the new (marginal) transactions added on top of the subchain go to cover the (marginal) orphaning risk for those transactions. We then compare the probability of orphan races with and without subchains, illustrating the significant advantage of the technique (Section 6).

In Section 7 we take a detour in order to derive the *fast-block approximation*. This approximation assumes that the propagation time is small compared to the target block time. It allows us to use only the first nonzero term in the power series expansion in our analysis and be ensured of a small and quantifiable error (up to some maximum propagation time). In Section 8 we apply the fast-block approximation to quantify the security of a transaction that has been verified in a subchain by calculating the expected value of a successful double-spend attack against that transaction. In addition to revealing a tradeoff between security and subchain verification times, we show, *ceteris paribus*, that the security also grows as the square of the average block size, thereby providing further motivation for on-chain scaling. In Section 9, we illustrate how subchains can be nested to circumvent the verification-time/security tradeoff, creating a fractal-like blockchain structure where transactions are processed almost continuously. Let us begin by defining the symbols we use.

2. List of Symbols

For the remainder of this manuscript, the following symbols have the specified meanings.

H	total hash rate of Bitcoin network	ΔQ	size of Δ-block
h	miner's individual hash rate	R	block reward (presently 25 Ƀ)
M	money (bitcoins)	T	block interval (10 min target)
M_0	orphaning risk incurred at start of double-spend attack	ΔT	Δ-block interval (subchain verification target)
M_{attack}	cost of double-spend attack	t	time

$\langle M_{\text{attack}} \rangle$	expectation value of cost of double-spend attack	u	dummy variable for integration
M_{alone}	orphaning risk incurred while mining alone	z	propagation impedance (time per bytes propagated)
$M_{\text{cooperate}}$	orphaning risk incurred while cooperating to build subchains	ρ	fee density, or the price per byte for block space
P_{attack}	probability distribution for attack block arrival time	ρ_0	price per byte for fast blocks
P_{orphan}	probability of an orphan race event	ρ^*	price per byte at free-market equilibrium
Q	block size or block space in bytes	ρ_{supply}	price per byte to produce block space
Q_{avg}	average block size	τ	propagation time
Q_{c}	block size capacity (the size that would take 10 min to propagate)	τ_0	propagation latency
$\dot{Q}$	average Blockchain growth rate	$\Delta\tau$	propagation time minus latency

The symbol $ refers to *US dollars*; price conversions between bitcoin and US dollars assume that 1 Ƀ = $400.

3. Subchains

To append a new block to the Blockchain, a miner must find a valid proof-of-work. This entails finding a nonce that when hashed together with the previous block's hash and the root hash for the block's transactions, results in an integer less than some target.[17] We define a *weak block* as a block that satisfies the weaker requirement

$$\text{hash(previous hash, nonce, root hash)} < \text{weak target.}$$

By sharing these weak blocks, miners can cooperate to build *subchains* (Fig. 1).

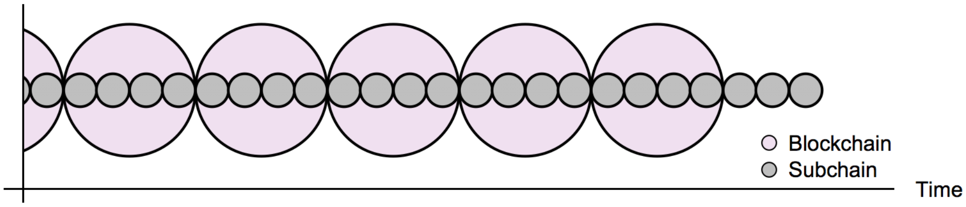

Fig. 1. Miners cooperate to build subchains in order to process more transactions and claim more fees without incurring additional orphaning risk. This illustration visualizes ¼-difficulty subchains (also referred to as 4x subchains).

Upon accepting a (strong) block, miners begin working on creating the next block in the chain by using the hash of the accepted block as the previous hash (Fig. 2a). When a miner finds a proof-of-work that satisfies the weak target, he broadcasts the weak block to the network. After verifying the weak block, each miner modifies the coinbase reward, appends additional transactions to the block if desired, computes the new root hash, and then continues scanning for a valid nonce (Fig. 2b). We will refer to the new information as the miner's *Δ-block* (Fig. 2f). If again a miner finds a proof-of-work that satisfies the weak target, he broadcasts the new weak block by sending only his Δ-block and the hash of the previous weak block. In this manner, miners can cooperate to build the subchain by transmitting only the new information and a fixed-byte-size reference to the subchain's tip.

When a miner finds a proof-of-work that meets the strong target (Fig. 2d), he broadcasts it in the same manner he would for a weak block (*i.e.*, by sending only his Δ-block and the hash of the previous weak block). Nodes recognize this as a valid (strong) block, retain the nonce and coinbase transaction, and close the subchain. The process of constructing a subchain on top of this latest block begins anew (Fig. 2e).

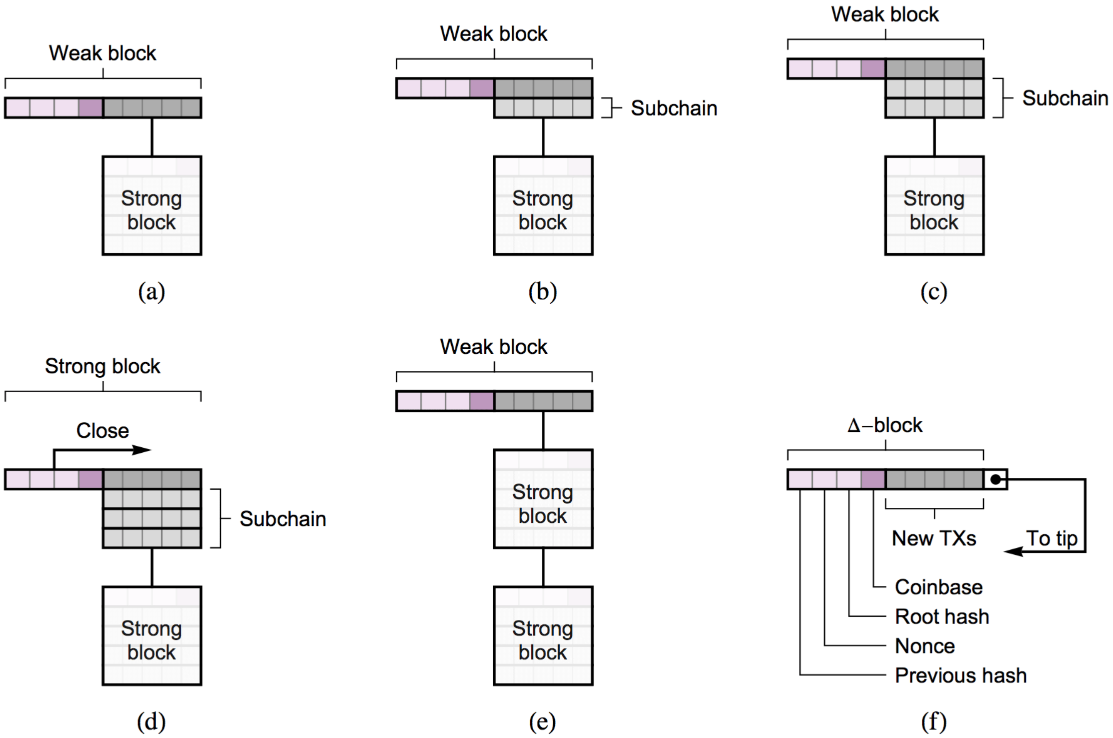

Fig. 2. Miners build subchains layer by layer (a – c), where each layer corresponds to the solution of a weak block. To propagate blocks (weak or strong), miners need only send their Δ-block and a reference to the subchain's tip (f), reducing the quantity of transmitted bytes. When a nonce that satisfies the strong target is found, the subchain is closed thereby becoming a strong block (d), and miners begin working on a new subchain (e).

4. Constructing Δ-blocks

In a scenario where subchains are the standard mechanism to build and propagate blocks, a miner can include all of the subchain's transactions—and thus all of its fees—without incurring additional orphaning risk (Fig. 3). The fees in each Δ-block add to the subchain's "pot," increasing the effective block reward. A miner is thus financially incentivized to build off the *highest-fee* subchain. Since all miners have the same incentives, they will tend to work together to extend a single chain.

A miner does, however, incur orphaning risk for the *new* transactions included in his Δ-block. He does not know ahead of time whether he might find a strong block, weak block, or no block at all. Since he only receives revenue if he finds a strong block, he should only accept transactions that pay a fee greater than the orphan risk he incurs (with respect to his potential revenue) by including those transactions.

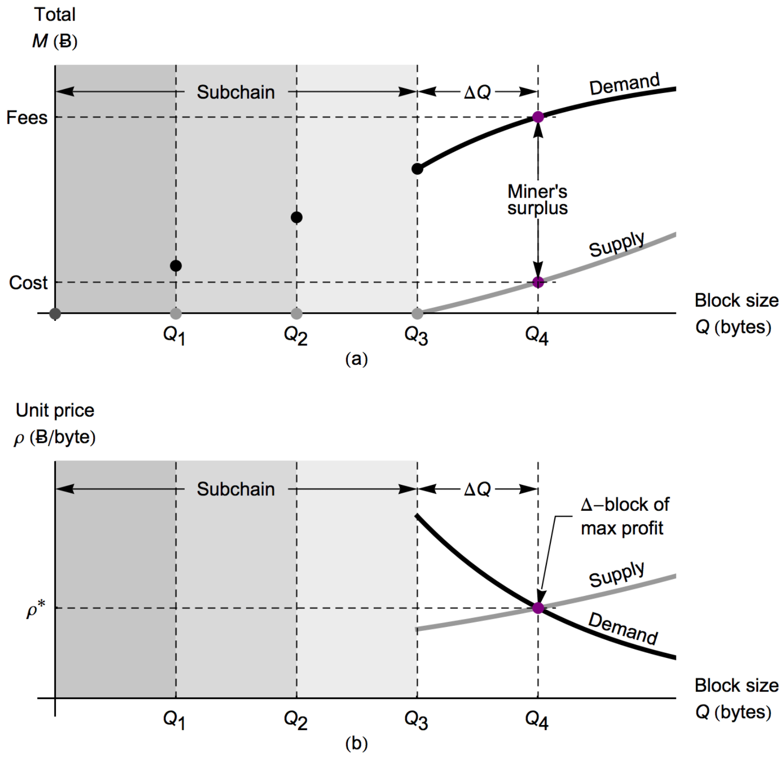

Fig. 3. The fees in a subchain increase each time a Δ-block gets added, effectively "growing the pot". Since a miner incurs no orphaning risk by including the contents of the subchain in his block candidate, he is incentivized to build his Δ-blocks on top of the highest-fee subchain. However, a miner incurs normal orphaning risk for any *new* transactions he chooses to add. Because of this, he will only include new transactions that pay more in fees per byte than the marginal cost of the additional block space.

The miner's marginal cost per byte to produce block space [5,18,19] due to orphaning risk can be approximated[20] as $\rho_{\text{supply}} = zRT^{-1}e^{\frac{\Delta\tau}{T}}$, where R is the block reward, T is the block time, $\tau = \Delta\tau + \tau_0$ is the propagation time and τ_0 is the latency. Since z describes the marginal time required to propagate an additional quantity of block information across the network, we can write $\Delta\tau = z\Delta Q$, where ΔQ is the size of the miner's Δ-block, in which case

$$\rho_{\text{supply}}(\Delta Q) = zRT^{-1}e^{\frac{z\Delta Q}{T}}. \tag{1}$$

In a perfectly competitive market, a miner will include all transactions that pay a fee per byte greater than the marginal cost, thereby maximizing the expectation value of his profit (Fig. 3b).

5. Proof-of-Work Security From Fee Revenue

It is simple to show that fees contribute to proof-of-work security (in the absence of a block size limit). Fig. 4 is a modification of Fig. 3a that considers all of the miner's revenues and costs, including the block reward and electricity for hashing. In a competitive market, the profits for marginal miners will trend to zero. To reconcile this fact with Fig. 4, the total production costs for block space must increase such that the two points marked in purple move closer together. That is, if industry profits were large, miners would tend to deploy more hash power to compete for this profit, thereby shifting the entire production cost curve upwards, increasing hashing costs and decreasing profits. As shown in Fig. 4, the fee revenue is significantly greater than the orphan risk; this fee revenue—captured Δ-block-by-Δ-block in the subchain—acts no differently than an increase in the block reward would: it serves to increase the network hash rate.

One subtlety to note is that a miner with revenue and costs as depicted in Fig. 4 would not start to mine until the subchain contained sufficient fees to make the expectation value of his profit positive. Presently, fees are such a small fraction of the block reward that most miners are profitable regardless of fees. However, when total fees are no longer small compared to the block reward, we would expect the instantaneous hash rate to increase every time a new Δ-block (and its fees) is added, as miners with marginally higher electricity costs turn on their machines. Further discussion of this phenomenon is beyond the scope of this paper.

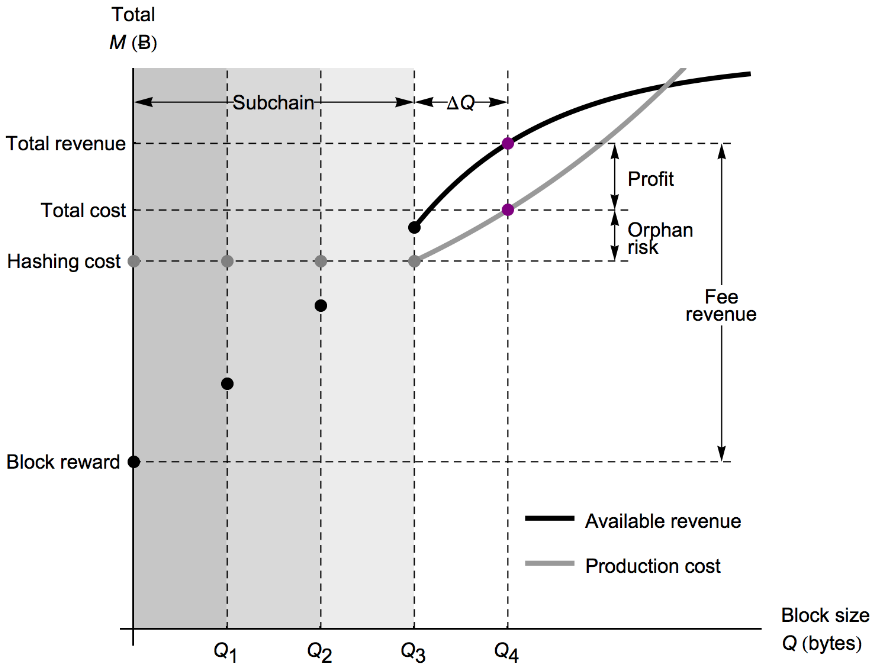

Fig. 4. If excess mining profits are available due to high fees, miners will deploy more hash power, raising network difficulty, resulting in higher hashing costs. This has the effect of shifting the production cost curve upwards, thereby reducing industry profits. Fees thus contribute directly to proof of work security. Total orphaning risk is small compared to total fee revenue, although marginal orphaning risk remains equal to marginal fee revenue.

6. Reduced Orphan Risk

Subchains reduce orphan risk by reducing the information propagated the moment the proof-of-work is solved. If a block takes time τ to propagate, the probability the network finds another block during the propagation interval[21] $0 < t < \tau$ is given by

$$P_{\text{orphan}} = \int_0^{\tau} \frac{1}{T} e^{-\frac{t}{T}} dt = 1 - e^{-\frac{\tau}{T}},$$

where $\frac{1}{T} e^{-\frac{t}{T}}$ is of course the probability distribution for the arrival time of a valid proof-of-work. Miners cooperate to build subchains in order to pre-propagate much of the block contents, thereby minimizing this propagation time and the chances of an orphan race.

Assuming that miners produce equal-sized Δ-blocks, each Δ-block is scaled down by the subchain factor, $\frac{T}{\Delta T}$, such that $\Delta Q = \frac{\Delta T}{T} Q$. The propagation time is thus $\tau = z\Delta Q + \tau_0$, from which it follows that

$$P_{\text{orphan}} = 1 - e^{-\frac{\tau_0}{T}} e^{-\frac{zQ\Delta T}{T^2}}.$$

This equation is plotted in Fig. 5 for various subchain factors and using recent estimates for the network propagation constants ($z = 17$ s/MB and $\tau_0 = 10$ s).[6,9,22] A subchain with $\frac{T}{\Delta T} = X$ would permit approximately X times more transactions per second at the same level of orphaning risk as without the subchain. The minimum useful subchain verification time is limited, however, because the network cannot come to consensus regarding the subchain faster than the network's latency (which, regardless of technology advancements, is limited by the product of the network diameter[23] and the speed of light to approximately 0.1 s).

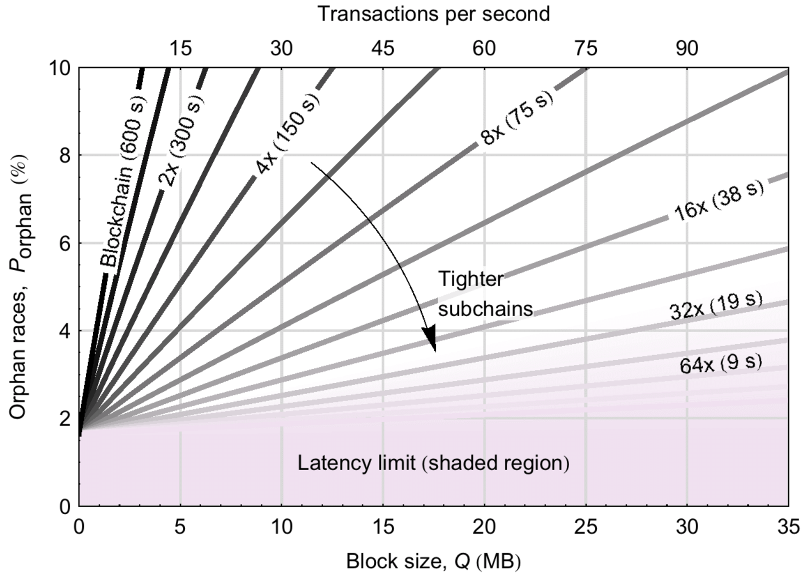

Fig. 5. Subchains help scale Bitcoin by reducing orphaning risk for larger block sizes. This chart is based on recent estimates for the network propagation constants ($z = 17$ s/MB and $\tau_0 = 10$ s).[6,9,22] For example, a subchain with 38-second verifications would permit approximately 16 times more transactions per second at the same level of orphaning risk as without the subchain. The minimum subchain verification time is limited, however, due to network latency (shaded region).

7. The Fast-Block Approximation

With a sufficiently weak target, several weak blocks will be found per strong block. The time required to propagate each Δ-block will then necessarily be small compared to the average time between strong blocks:

$$\tau \ll T. \tag{2}$$

We will refer to Eq. (2) as the *fast-block approximation*; it can be used for the fast propagation of both Δ-blocks and full blocks.

As an example of using the fast-block approximation, consider the marginal cost of block space as specified by Eq. (1). Noting that $\Delta\tau = z\Delta Q$, the exponent in Eq. (1) is small (*cf*. Eq. 2) and we can expand the right-hand side as a power series about $\Delta\tau = 0$, yielding $\rho_{\text{supply}} = zRT^{-1}(1 + \Delta\tau/T + \cdots)$. Since $\Delta\tau/T \ll 1$, the marginal cost per byte for transactions included in Δ-blocks is nearly constant at

$$\rho_{\text{supply}} \approx \rho_0 = zRT^{-1}, \tag{3}$$

changing by roughly one percent for each six seconds of additional propagation time[24] (we expect Δ-blocks to propagate very quickly). The fast-block approximation can be used in similar ways for other problems in order to express an equation using the first nonzero term in its power series expansion.

8. Zero-Confirmation Security

To double-spend a transaction included in a subchain, an attacker must produce a weak block with greater fees than the honest subchain before the network finds a strong block[25] (Fig. 6). The cost of finding such a block is significant.

Using partial derivatives, the rate at which orphaning risk, *M,* is incurred can be expressed as

$$\frac{d}{dt}M = \frac{\partial M}{\partial Q}\frac{\partial Q}{\partial t}.$$

If miners cooperate to build subchains, we can use the fast-block approximation (Eq. 2), in which case the marginal cost per byte, $\rho = \frac{\partial M}{\partial Q}$, is given by Eq. (3). Recognizing that the transaction rate $\dot{Q} = \frac{\partial Q}{\partial t}$, we can express the rate at which the network incurs orphaning risk as

$$\frac{d}{dt}M_{\text{cooperate}} = \rho_0\dot{Q}.$$

If we take the transaction rate to be constant, then the orphan risk grows *linearly* with time:

$$\begin{aligned} M_{\text{cooperate}}(t) &= \int_0^t \rho_0\dot{Q}\,du + M_0 \\ &= \rho_0\dot{Q}t + M_0, \end{aligned} \tag{4}$$

where M_0 is the orphaning risk already incurred at time $t = 0$.

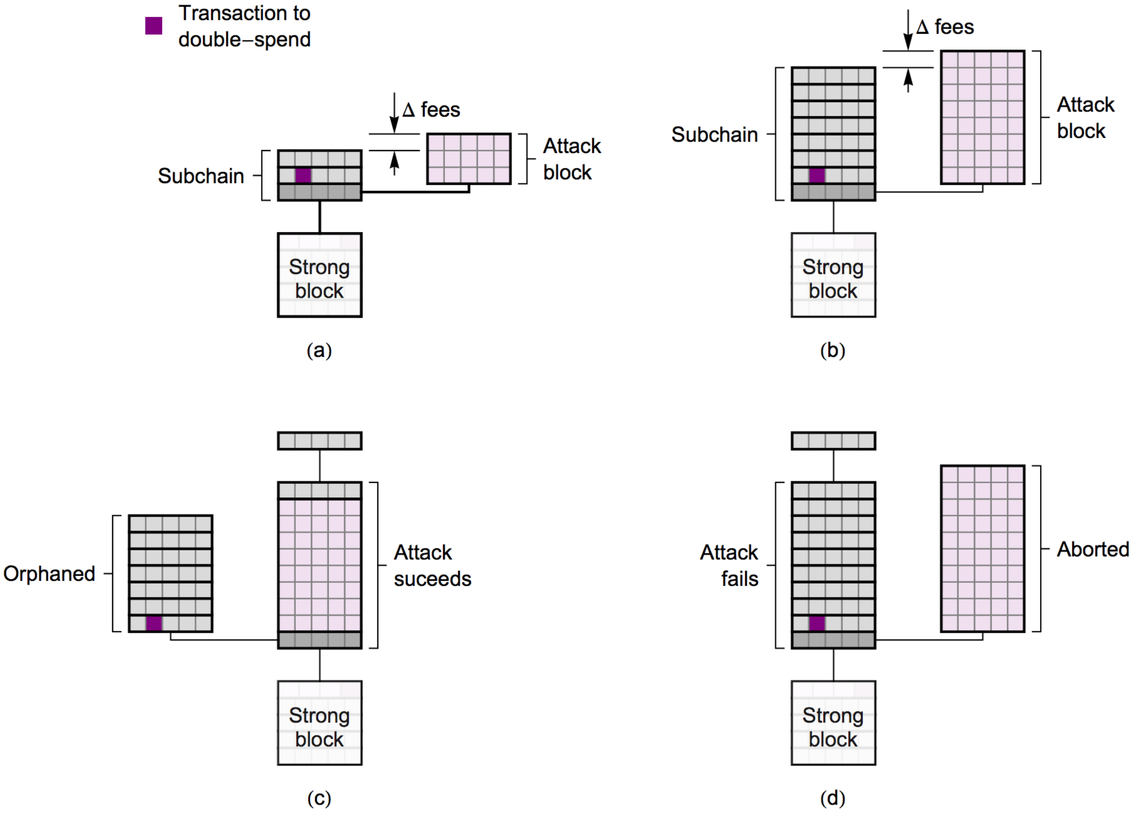

Fig. 6. To double-spend a transaction, an attacker must find a weak block with more fees than the best subchain before the network finds a strong block. Because new Δ-block are continually appended to the subchain, the attacker must continually add transactions to his attack block to ensure that it contains the most fees (b). The longer the attacker persists, the greater the orphaning risk he incurs. The attack ends either in success by the attacker finding a weak block (c), or in failure by the network finding a strong block (or by the attacker giving up).

For the attacker, we cannot use the fast-block approximation. As shown in Fig. 6, his attack block continually grows ($\geq \dot{Q}$) to ensure that it would form the highest-fee subchain. Since its propagation time may violate Eq. (2), the marginal orphaning risk per byte is given instead by Eq. (1). The attacker will thus incur orphaning risk at an *increasing* rate. This rate is proportional to the rate at which the network would incur orphaning risk if it were *not* cooperating to build subchains:

$$\frac{d}{dt} M_{\text{alone}} = \rho_0 \dot{Q} e^{\frac{z\dot{Q}t}{T}}.$$

In this non-cooperative case, the orphan risk grows *exponentially* in time:

$$\begin{aligned} M_{\text{alone}}(t) &= \int_0^t \rho_0 \dot{Q} e^{\frac{z\dot{Q}u}{T}} du + M_0 \\ &= R\left(e^{\frac{z\dot{Q}t}{T}} - 1\right) + M_0. \end{aligned} \tag{5}$$

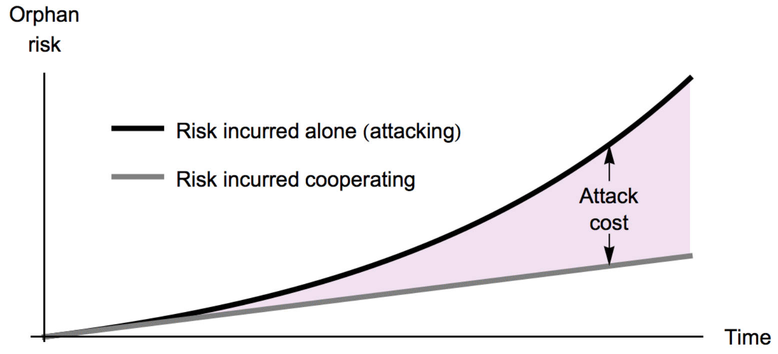

Fig. 7. Assuming transactions are processed at a constant rate, miners incur orphaning risk at a constant rate if they cooperate to build subchains, and at an ever-increasing rate if they do not cooperate. An attacker attempting to double-spend incurs risk following the upper curve because his attack block continually grows in size. Because the attacker is not cooperating to build a subchain, he must propagate his potentially large attack block as a single message. The difference between the two curves is proportional to the effective cost to wage the attack.

These two curves are plotted in Fig. 7. The effective cost to the attacker is the risk he incurs by mining alone (on his attack block) minus the risk he would have incurred if he were cooperating to build the subchain instead. Since the attacker is only responsible for his hash-power weighted fraction of the total, the cost of his attack is given by

$$M_{\text{attack}}(t) = \frac{h}{H}\left[M_{\text{alone}}(t) - M_{\text{cooperate}}(t)\right]. \tag{6}$$

To proceed further, it is helpful to apply the fast-block approximation once again. Recall that the amount of time required for the attacker's Δ-block to propagate is $\Delta\tau = z\dot{Q}t$, in which case the exponent in Eq. (5) can be written as $\Delta\tau/T$. Even if the attack block takes one minute to propagate (incurring unreasonable large orphaning risk), the exponent is still significantly less than unity. Substituting Eqs. (4) and (5) into Eq. (6) and expanding in a power series about $t = 0$ (*cf.* Appendix A) reveals that both the constant and linear terms vanish, leaving

$$M_{\text{attack}}(t) = \frac{h}{H}\left(\frac{1}{2}\frac{\rho_0^2}{R}\right)\dot{Q}^2 t^2 + \cdots. \tag{7}$$

The first term in Eq. (7) is accurate to three percent for attack-block propagation times under one minute.[26] Observe that the cost of the attack initially increases as the *square* of the attack time.

If the attacker is lucky, he might find a weak block quickly and thus incur very little cost; if the attacker is unlucky, it might take him a long time to find a block and thus incur a great cost. The *expectation value* for the cost of a successful attack is the cost if it takes time t weighted by the probability, $P_{\text{attack}}(t)$, that the arrival time of the attack block lies between t and $t + dt$, and integrated over all possible values of t:

$$\langle M_{\text{attack}}\rangle = \int_0^\infty M_{\text{attack}}(t)\, P_{\text{attack}}(t)dt. \tag{8}$$

To determine $P_{\text{attack}}(t)$, recall that the probability distribution for the arrival of a *strong* block is given by the exponential distribution $\frac{1}{T}e^{-\frac{t}{T}}$. Finding a weak block follows the same process, but is "easier" by the subchain factor, $\frac{T}{\Delta T}$, and "harder" by the network-to-attacker hash rate ratio, $\frac{H}{h}$. The distribution is then found by substituting $T \to \frac{H}{h}\frac{\Delta T}{T}T$ into the strong block probability distribution, yielding[27]

$$P_{\text{attack}}(t) = \frac{1}{\Delta T}\frac{h}{H}e^{-\frac{h}{H}\frac{t}{\Delta T}}. \tag{9}$$

In Appendix A, we substitute Eqs. (7) and (9) into Eq. (8), determining that the expectation value of the attack cost is

$$\langle M_{\text{attack}} \rangle = \frac{H}{h}\left(\frac{\Delta T}{T}\frac{Q_{\text{avg}}}{Q_{\text{c}}}\right)^2 R, \tag{10}$$

where $Q_{\text{avg}} = \dot{Q}T$ is the average block size and $Q_{\text{c}} \equiv z^{-1}T$ is the *network block size capacity* (defined as the size of the block that would take the full 10-minute block time to propagate).[6]

Table 1 puts numbers to Eq. (10) for $R = 25$ Ƀ and a network block capacity of $Q_{\text{c}} = 35$ MB (corresponding to $z = 17$ s/MB).[6,9] Since the network may find a strong block before the attacker finds a weak block, the table also shows the probability of success calculated using standard Poisson process results.[28] It would cost an attacker commanding 10% of the network hash rate \$0.20 (510 μɃ) to double-spend a one-minute verification if the average block size were 500 kB; however, it would cost that same attacker over \$50 (130,000 μɃ) to double-spend the transaction if the average block size were 8 MB instead. Although these security levels are still fairly low, note that they scale linearly with the price of a bitcoin.

Table 1. Zero-confirmation security after the first subchain verification[a]

Attacker's hash rate (% of total)	Estimate for the expectation value of the cost of a successful double-spend attack (μɃ)					
	one minute verifications			six second verifications		
	Probability of success (%)	0.5 MB block size	8 MB block size	0.5 MB block size	8 MB block size	Probability of success (%)
0.1	1.0	51,000	13,000,000	510	130,000	9.1
0.3	2.9	17,000	4,400,000	170	44,000	23.1
1.0	9.2	5,100	1,300,000	51	13,000	50.3
3.0	23.6	1,700	440,000	17	4,400	75.6
10	52.6	510	130,000	5.1	1,300	91.7
30	81.1	170	44,000	1.7	440	97.7

[a] Based on a propagation impedance of 17 s/MB and a 25 Ƀ block reward.

9. Nested Subchains

Miners must agree on the weak target in order to cooperate building subchains. At first glance, it appears there is a significant trade-off: too strong a target leads to higher orphan risk and slower subchain verifications while too weak a target makes it easier to double-spend. One possible solution is to use nested subchains (Fig. 8). (Another possible solution is to use a version of the GHOST protocol.[29])

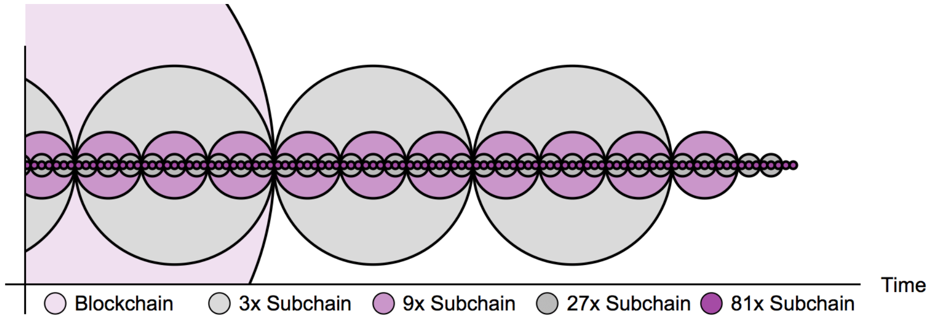

Fig. 8. Subchains can be nested to avoid the tradeoff between verification time and security. This image shows trinary nesting of Δ-blocks (represented by the circles). The diameter of the Δ-blocks corresponds to the verification time target and the area represents the cost for an attacker to successfully double spend.

A nested subchain is simply a subchain within a subchain. When a miner finds a block that satisfies the subchain difficulty, the deeper subchain is closed and a new subchain at the deeper level begins. Miners build from the highest-fee subchain at a given nesting depth but ignore higher-fee subchains at deeper nestings. With nesting, the subchain verification time can be reduced as miner connectivity improves, fundamentally limited only by network latency due to the time is takes light to travel across the network.

10. Conclusion

We presented a scaling technique called subchains to reduce orphaning risk for large blocks and improve the security of zero-confirmation transactions. Subchains are formed as a series of weak blocks, with the next weak block building a new layer of transactions (what we called a Δ-block) upon the previous weak block. Miners transmit blocks (both weak and strong) by sending only the Δ-block and a fixed byte-size reference to the subchain's tip.

Miners cooperate to extend a single subchain in order to maximize that subchain's total fees, as those fees can be included in each miner's candidate block without incurring orphaning risk. Interestingly, subchains only have a small effect on the marginal transaction cost. Although the technique reduces orphan rates for large blocks considerably, it does not result in a corresponding reduction in the total fees required by miners to produce such blocks. The result is that transaction fees pay for proof-of-work security, rather than paying for orphans.

Transactions included in a subchain are quickly secured by the transactions stacked above them, encouraging additional uses for Bitcoin that require fast verifications. The cost to double-spend a transaction included in a subchain increases with the square of the target verification time. Subchains can be nested to avoid the tradeoff between security and verification time inherent in a non-nested subchain. The security also increases as the square of the average block size, further motivating on-chain scaling. As the use and complexity of subchains grow, the idea of "blocks" and "confirmations" could be abstracted away from the user—the Blockchain may one day appear as a continuous stream of transactions, where the new transactions added each moment serve to secure the ones that came before them.

Neither a hard nor soft fork is required to implement subchains; however, the technique is only useful if a significant fraction of the network hash power participates. Network-wide support for subchains would add significant transactional capacity and improve the user experience, helping to further advance the adoption of Bitcoin.

Appendix

Power series expansion for the cost of a double-spend attack with respect to attack time—As described in Section 8, the cost to attack a subchain is

$$M_{\text{attack}}(t) = \frac{h}{H}\left[M_{\text{alone}}(t) - M_{\text{cooperate}}(t)\right],$$

where

$$M_{\text{cooperate}}(t) = \rho_0 \dot{Q} t + M_0$$

and

$$M_{\text{alone}}(t) = R\left(e^{\frac{z\dot{Q}t}{T}} - 1\right) + M_0.$$

Combing these equations and rearranging gives

$$\frac{H}{h} M_{\text{attack}}(t) = R\left(e^{\frac{z\dot{Q}t}{T}} - 1\right) - \rho_0 \dot{Q} t.$$

Using the fast-block approximation and the power series expansion $e^x = 1 + x + \frac{x^2}{2} + \cdots$, we get

$$\frac{H}{h} M_{\text{attack}}(t) = R\left[\left(1 + \frac{z\dot{Q}t}{T} + \frac{z^2\dot{Q}^2t^2}{2T^2} + \cdots\right) - 1\right] - \rho_0 \dot{Q} t$$

$$= zRT^{-1}\dot{Q}t + \ R\frac{z^2\dot{Q}^2t^2}{2T^2} - \rho_0 \dot{Q} t + \cdots.$$

Recognizing that $\rho_0 = zRT^{-1}$ allows us to write

$$\frac{H}{h} M_{\text{attack}}(t) = \rho_0 \dot{Q} t + \left(\frac{1}{2}\frac{\rho_0^2}{R}\right)\dot{Q}^2 t^2 - \rho_0 \dot{Q} t + \cdots.$$

Cancelling the linear terms and rearranging gives our desired result for the cost of the attack versus attack time:

$$M_{\text{attack}}(t) = \frac{h}{H}\left(\frac{1}{2}\frac{\rho_0^2}{R}\right)\dot{Q}^2 t^2 + \cdots. \tag{A1}$$

Solution to the integral for the expectation value of the attack cost—Also as described in Section 8, the expectation value for the attack cost can be written

$$\langle M_{\text{attack}} \rangle = \int_0^\infty M_{\text{attack}}(t)\, P_{\text{attack}}(t) dt,$$

where $M_{\text{attack}}(t)$ was derived above (Eq. A1) and (*cf*. Eq. 9)

$$P_{\text{attack}}(t) = \frac{1}{\Delta T}\frac{h}{H} e^{-\frac{h}{H}\frac{t}{\Delta T}}.$$

Combining gives

$$\langle M_{\text{attack}} \rangle = \int_0^\infty \left[\frac{h}{H} \left(\frac{1}{2} \frac{\rho_0^2}{R} \right) \dot{Q}^2 t^2 \right] \left[\frac{1}{\Delta T} \frac{h}{H} e^{-\frac{h}{H}\frac{t}{\Delta T}} \right] dt.$$

Assuming that $\dot{Q}$ is constant allows us to write

$$\langle M_{\text{attack}} \rangle = \frac{\dot{Q}^2}{2} \frac{\rho_0^2}{R} \frac{h}{H} \left(\frac{1}{\Delta T} \frac{h}{H} \right) \int_0^\infty t^2 \, e^{-u} dt, \tag{A2}$$

where

$$u = \left(\frac{1}{\Delta T} \frac{h}{H} \right) t. \tag{A3}$$

Since

$$du = \left(\frac{1}{\Delta T} \frac{h}{H} \right) dt \tag{A4}$$

we can use Eqs. (A3) and (A4) to write Eq. (A2) as a function strictly of u:

$$\langle M_{\text{attack}} \rangle = \frac{\dot{Q}^2}{2} \frac{\rho_0^2}{R} \frac{h}{H} \left(\frac{1}{\Delta T} \frac{h}{H} \right) \int_0^\infty \left(\frac{1}{\Delta T} \frac{h}{H} \right)^{-2} u^2 \, e^{-u} \left(\frac{1}{\Delta T} \frac{h}{H} \right)^{-1} du.$$

Collecting like terms gives

$$\langle M_{\text{attack}} \rangle = \frac{\dot{Q}^2}{2} \frac{\rho_0^2}{R} \frac{h}{H} \left(\frac{1}{\Delta T} \frac{h}{H} \right)^{-2} \int_0^\infty u^2 \, e^{-u} du.$$

The integral is now in a standard form with known solution $\int_0^\infty u^2 e^{-u} du = 2$. Further simplification produces

$$\langle M_{\text{attack}} \rangle = \frac{1}{R} \frac{H}{h} \left(\rho_0 \dot{Q} \Delta T \right)^2.$$

We can transform this equation into a more useful form by substituting (*cf.* Eq. 3) $\rho_0 \rightarrow zRT^{-1}$:

$$\begin{aligned} \langle M_{\text{attack}} \rangle &= \frac{1}{R} \frac{H}{h} \left(zRT^{-1} \dot{Q} \Delta T \right)^2 \\ &= \frac{H}{h} \left(\frac{\Delta T}{T} \frac{\dot{Q}T}{z^{-1}T} \right)^2 R. \end{aligned}$$

If we recognize the average block size as $Q_{\text{avg}} = \dot{Q}T$ and define the network block size capacity as $Q_{\text{c}} \equiv z^{-1}T$, then the expectation value for the cost of a successful attack becomes

$$\langle M_{\text{attack}} \rangle = \frac{H}{h} \left(\frac{\Delta T}{T} \frac{Q_{avg}}{Q_c} \right)^2 R.$$

The attack cost is proportional to the square of the subchain verification time and the square of the average block size, and inversely proportional to the attacker's hash power.

Acknowledgement

The author gratefully acknowledges the kind review and suggestions of Gavin Andresen and "awemany." He also wishes to thank the many thoughtful and intelligent individuals at the Bitcoin Forum (bitco.in) for hours of pleasant and productive discussion.

Notes and References

[1] "Total Number of Transactions" chart. *Blockchain.info* (13 December 2015) https://blockchain.info/charts/n-transactions-total

[2] "Median Transaction Confirmation Time (With Fee Only)" chart. *Blockchain.info* (13 December 2015) https://blockchain.info/charts/avg-confirmation-time

[3] "Scalability." *Bitcoin Wiki* (13 December 2015) https://en.bitcoin.it/wiki/Scalability

[4]Murdoch, S. J., Drimer, S., Anderson, R., Bond, M. "Chip and PIN is Broken." *2010 IEEE Symposium on Security and Privacy*, Oakland, California (16 May 2010) http://www.unibank.org/toposign/chip_and_pin_is_broken.pdf

[5]Rizun, P. R. "A Transaction Fee Market Exists Without a Block Size Limit." No Publisher (2015) https://dl.dropboxusercontent.com/u/43331625/feemarket.pdf

[6] Stone, G. A. "An Examination of Bitcoin Network Throughput Via Analysis of Single Transaction Blocks." No Publisher (2015) http://www.bitcoinunlimited.info/1txn

[7] Barski, C., and Wilmer, C. *Bitcoin for the Befuddled*. San Francisco: No Starch Press (2014)

[8] Carlsten, M., Kalodner, H., Narayanan, A. "Mind the Gap: Security Implications of the Evolution of Bitcoin Mining." *Scaling Bitcoin Montreal* (12 September 2015)

[9] "Bitcoin Network Capacity Analysis – Part 6: Data Propagation." *Tradeblock Blog* (23 June 2015) https://tradeblock.com/blog/bitcoin-network-capacity-analysis-part-6-data-propagation

[10] Decker C. and Wattenhofer R. "Information Propagation in the Bitcoin Network." *13th IEEE International Conference on Peer-to-Peer Computing*, Trento, Italy, September 2013

[11] Pseudonymous ("rocks"). Comment in "Gold Collapsing. Bitcoin UP." *Bitcoin Forum.* (12 November 2015) https://bitco.in/forum/threads/gold-collapsing-bitcoin-up.16/page-99#post-3585

[12] Andresen, G. "[Bitcoin-development] Weak block thoughts…" *Bitcoin-development* (23 September 2015) http://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-September/011157.html

[13] Pseudonymous ("TierNolan"). "Decoupling transactions and POW." *Bitcointalk* (18 April 2013) https://bitcointalk.org/index.php?topic=179598.0

[14] Andresen, G., Comment in "Faster blocks vs bigger blocks." *Bitcointalk* (3 July 2014) https://bitcointalk.org/index.php?topic=673415.msg7658481#msg7658481

[15] Rosenbaum, K., Russell, R. "IBLT and Weak Block Propagation Performance." *Scaling Bitcoin Hong Kong* (6 December 2015)

[16] Maxwell, G. "[Bitcoin-development] Block Size Increase." *Bitoin-development* 7 May 2015 (accessed 13 December 2015) https://lists.linuxfoundation.org/pipermail/bitcoin-dev/2015-May/007880.html

[17] BitFury Group. "Proof of Stake versus Proof of Work." No Publisher (13 September 2015) http://bitfury.com/content/4-white-papers-research/pos-vs-pow-1.0.2.pdf

[18] Pinna, D. "On the Nature of Miner Advantages in Uncapped Block Size Fee Markets." No Publisher (2015) http://www.scribd.com/doc/276849939/On-the-Nature-of-Miner-Advantages-in-Uncapped-Block-Size-Fee-Markets

[19] BitFury Group. "Incentive Mechanisms for Securing the Bitcoin Blockchain." No Publisher (2015) http://bitfury.com/content/4-white-papers-research/bitfury-incentive_mechanisms_for_securing_the_bitcoin_blockchain-1.pdf

[20] This result is derived using the *small miner approximation* where the self-propagation advantage can be ignored. Refer to the paper cited in Note 5.

[21] Andresen, G. "Back-of-the-envelope calculations for marginal cost of transactions." No Publisher (2013) https://gist.github.com/gavinandresen/5044482.

[22] These estimates are probably conservative (*i.e.*, the latency and propagation impedance are both likely smaller) as the methodology used by Stone includes other effects such as the time to construct a new block candidate from mempool, and the methodology used by Tradeblock measured propagation to nodes rather than to hash power.

[23] Pseudonymous ("awemany"). Comment in "Block Space as a Commodity." *Bitcoin Forum* (26 September 2015) https://bitco.in/forum/threads/block-space-as-a-commodity-a-transaction-fee-market-exists-without-a-block-size-limit.58/page-4#post-1409

[24] The rate at which the marginal cost for block space initially changes with propagation delay is given by $\frac{d}{d\tau}\rho_{\text{supply}}\Big|_{\tau=0} = zRT^{-2}$ which allows us to approximate $\Delta\rho_{\text{supply}} = zRT^{-2}\Delta\tau$. Since $\rho_{\text{supply}} = zRT$, we can write $\frac{\Delta\rho_{\text{supply}}}{\rho_{\text{supply}}} = \frac{\Delta\tau}{T}$, which works out to 1% for every 6 s of propagation delay.

[25] Honest miners are not *required* to extend the highest-fee subchain; in a case of blatant double spending, miners may choose to ignore the attack block for the health of the network. Thus the zero-confirmation security estimates in Section 8 are conservative.

[26] Following a similar calculation as for Note 24.

[27] In practice, the attacker will give up before $t = \infty$ (*e.g.*, the network may have found a strong block), and thus this estimate is conservative.

[28] The probability that the attacker finds a weak block before the network finds a strong block is given by $\int_0^\infty \frac{1}{\Delta T}\frac{H}{h} e^{-\frac{t}{\Delta T}\frac{h}{H}} e^{-\frac{t}{T'}} dt = \frac{h/H}{h/H+\Delta T/T'}$ where $T' = \frac{TH}{H-h}$ to account for the attacker's hashing power no longer contributing to the honest chain.

[29] Sompolinsky, Y., Zohar, A. "Secure High-Rate Transaction Processing in Bitcoin." No Publisher (2015) http://www.cs.huji.ac.il/~avivz/pubs/15/btc_ghost_full.pdf