Note on block withholding attacks

Assaf Shomer

December 17, 2013

Abstract

We calculate the probability of success of block-withholding attacks on the bitcoin blockchain. These types of attacks employ a non-traditional mining strategy where the attacker is building a secret branch of the blockchain with the hope of overtaking the main branch. When the attackers succeeds in replacing the top of the blockchain with their secret branch they rip the reward associated with the blocks in the secret branch. We demonstrate that such an attack becomes beneficial only if the attackers possess a big enough portion of the total hashing power.

Contents

Chapter 1

Introduction

1.1 Bitcoin and mining

Bitcoin is the world's first decentralized digital currency ([**?**]). blah blah

1.2 The attack strategy

1.2.1 Honest Miners

The honest miners following the bitcoin protocol described in [**?**], publish each block as soon as it is discovered and switch their mining efforts to the head of the blockchain[1] any time a new block is found.

$$\dots \to B_L \to B_{L+1} \to B_{L+2} \to \dots$$

1.2.2 Block withholding Miners

The attackers do not share their newly found blocks and instead work on a **secret** branch of the block-tree until such time that their branch is longer than the main branch. At this time they can publish it and uproot the last n (honestly mined) blocks in favour of their $m > n$ secretly mined ones.

1.2.3 Goal

Our goal is to calculate the probability that a block-withholding miner of relative power q succeeds in replacing a block (and possibly some number of confirmation blocks on top of it) by publishing a secretly mined branch of the block-tree that is longer than the main branch.

[1] In practice different miners may be aware of different branches of the block-tree at a given moment. However, such differences are quickly resolved with very high probability once a new block is found because the protocol names the branch with the maximal difficulty to be the blockchain.

The quantity we are interested in is complimentary to the probability that a given block remains in the block-chain in face of the said block-withholding attack. Motivated by the work done in [**?**] we are interested in finding out if and when the block-withholding strategy gives the miner a higher probability of success in mining blocks, compared to the standard strategy outlined in the bitcoin protocol.

1.3 Setup

Let us denote by $\mathcal{H}$ the total hashing power of the network and divide it abstractly into an *Honest* part which holds a portion $p\mathcal{H}$ of the total hashing power (where $p \in [0,1]$) and an *Attacker* which holds the rest $q\mathcal{H} = (1-p)\mathcal{H}$.

We start our analysis at a given point in time where the blockchain is of length L and denote the last block mined as B_L. As time marches on the honest miners continue to mine on top of it ($B_{L+1}, B_{L+2}, \dots$) while the attackers are building a separate branch on top of B_L ($\tilde{B}_{L+1}, \tilde{B}_{L+2}, \dots$) with the aim of overtaking it:

$$\begin{tikzcd}
\cdots \arrow[r] & B_L \arrow[r] \arrow[dr] & B_{L+1} \arrow[r] & B_{L+2} \arrow[r] & \cdots \arrow[r] & B_{L+n} & \text{Main} \\
 & & \widetilde{B}_{L+1} \arrow[r] & \widetilde{B}_{L+2} \arrow[r] & \dots \arrow[r] & \widetilde{B}_{L+m} & \text{Secret}
\end{tikzcd} \tag{1.1}$$

Treating block mining as a negative binomial random variable, the probability $P_{n,p}(m)$ that m blocks are mined by the attackers **before** n blocks were honestly mined is proportional to $p^n q^m$ and can be shown (appendix A.1) to be given by

$$P_{n,q}(m) = \binom{n+m-1}{m}(1-q)^n q^m \quad n = 1, 2, \dots \tag{1.2}$$

The probability $a_{n,m}(q)$ that the attackers manage to catch-up and overtake the blockchain given the situation above[2] is given by a Markov chain that depends only on the advantage z of the honest network over the attackers $z = n - m$ defined by the recurrence relation

$$a_z(q) = (1-q)a_{z+1}(q) + q a_{z-1}(q) \tag{1.3}$$

The relation can be solved with boundary conditions[3] $a_{-1} = 1, a_\infty = 0$ by

$$a_z(q) = \begin{cases} \left(\dfrac{q}{1-q}\right)^{z+1} & q \in [0, \dfrac{1}{2}] \quad \text{and} \quad z = 0, 1, 2 \dots \\ 1 & \text{otherwise} \end{cases} \tag{1.4}$$

[2] Namely, that from the beginning of the experiment until the moment the honest network mines it's nth block on top of B_L, the attackers managed to mine m blocks on top of B_L constituting their secret branch.

[3] Note that the condition $a_{-1} = 1$ encodes the fact that the attack is successful once the secret chain is longer than the main chain.

For example, to find the probability of a **double-spend attack** on a transaction included in a block B_L with n confirmations, the attacker needs to catchup from a deficit of $n-(m+1)$. The extra block $m+\mathbf{1}$ is the block where the amount spent in B_L was spent again (or resent to the attacker) thus constituting a double-spend attack. This block is denoted with an asterisk B_L^* in 1.5:

$$\begin{tikzcd}
\cdots \arrow[r] & B_{L-1} \arrow[r] \arrow[dr] & \overbrace{B_L \to B_{L+1} \to \cdots \to B_{L+n-1}}^{n\text{ confirmations}} & \text{Main} \\
& & \underbrace{\widetilde{B}_L^* \to \widetilde{B}_{L+1} \longrightarrow \quad \ldots \longrightarrow \widetilde{B}_{L+m}}_{(m+1)\text{ blocks}} & \text{Secret}
\end{tikzcd} \tag{1.5}$$

This attack was first analyzed in [?][4], treated more accurately in [?] and was shown to be

$$D_n(p) = \sum_{m=0}^{\infty} P_{n,p}(m)\, a_{n-(m+1)}(p) \tag{1.6}$$

1.4 Block withholding attack - Type 1

1.4.1 Probability of success

As explained in 1.2.3, our focus here is a little different. Let $Q(q)$ be the probability that the attackers succeed in mining a secret branch on top of B_L that is longer than the main branch, which allows them to publish it and replace B_{L+1}, and all the blocks honestly mined on top of it.

Alternatively, $1 - Q(q)$ is the probability that B_{L+1} remains in the block-chain despite an attempt of a block-withholding attacker building a secret branch on top of B_L.

A successful block-withholding attack on B_L occurs if the attackers manage to catch-up on B_{L+1} after starting with $m = 0, 1, \ldots$ blocks. The starting point for the catch-up process for some m is shown below:

$$\begin{tikzcd}
\cdots \arrow[r] & B_L \arrow[r] \arrow[dr] & B_{L+1} & & & & \text{Main} \\
& & \widetilde{B}_{L+1} \arrow[r] & \widetilde{B}_{L+2} \arrow[r] & \ldots \arrow[r] & \widetilde{B}_{L+m} & \text{Secret}
\end{tikzcd} \tag{1.7}$$

There are two differences from a double-spend attack described above. One is that we are not requiring a particular number of blocks mined on top of B_L before the secret branch is published. Secondly, we don't require the attacker started the catch-up (with probability of success captured in equation 1.4) after mining at least one block.

[4] Approximated by a Poisson process.

Apart from that, the math is very similar. Formally

$$Q(q) = \sum_{m=0}^{\infty} P_{1,q}(m) a_{1-m}(q) \tag{1.8}$$

which results to (see details in appendix A.2)

$$Q(q) = \begin{cases} \dfrac{q^2}{1-q}(3-2q) & q \in [0, \frac{1}{2}] \\ 1 & q \in [\frac{1}{2}, 1] \end{cases} \tag{1.9}$$

In Figure 1.1 we plot the probability of a successful attack as a function of the relative hashing power q against the probability of the attacker being honest, in which case his probability of success is just q

We see that an attacker improves his chances in the range $q > q_0$ where

$$q_0 = 1 - \frac{1}{\sqrt{2}} \sim 0.293 \tag{1.10}$$

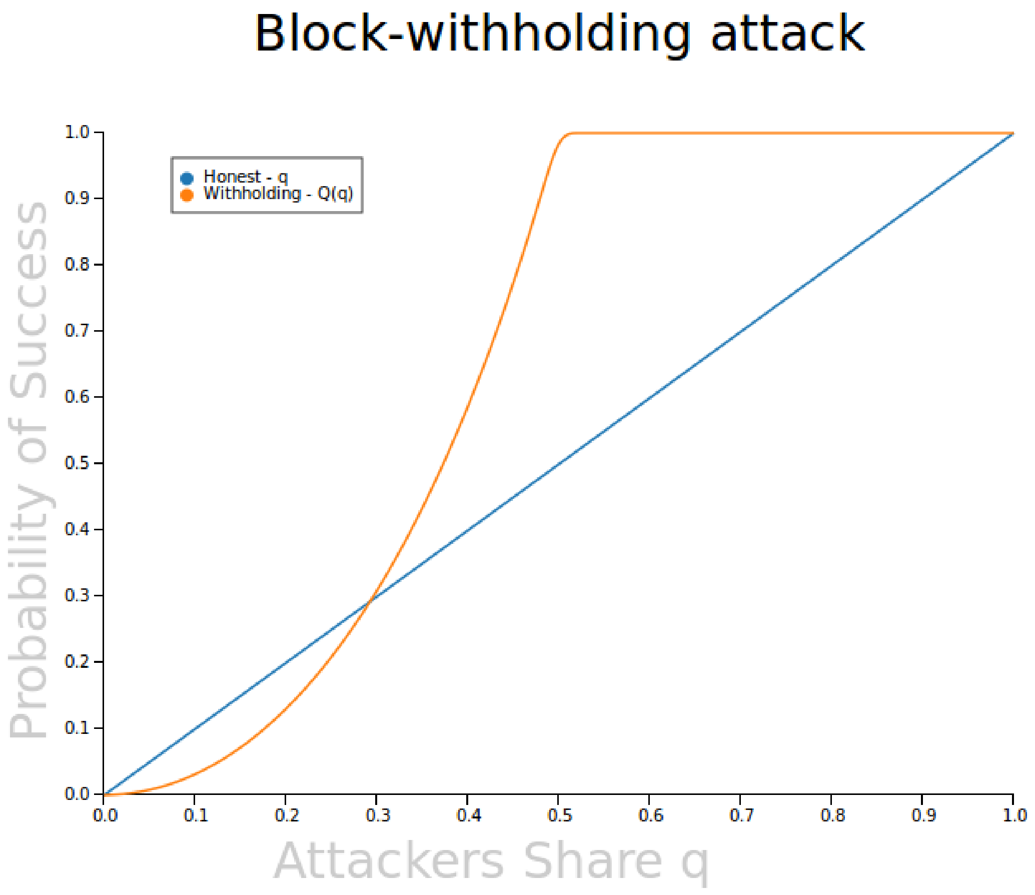

Figure 1.1: The orange curve plots the probability that the attackers manage to uproot the next honest block and replace it with one of their own. The blue curve is the baseline probability for an honest miner with the same hashing power q

1.4.2 Attacker's reward

Next we calculate the expected reward of a block-withholding attacker. We assume for simplicity that the reward of each party is simply proportional to the number of blocks mined by that party.

Formally

$$R(p) = \sum_{m=0}^{\infty} m \cdot P_{1,p}(m) a_{1-m}(p) \tag{1.11}$$

which results to (see details in appendix A.4)

$$R(p) = \begin{cases} \dfrac{q^2}{1-q}(3-2q) & q \in [0, \frac{1}{2}] \\ \dfrac{q}{1-q} & q \in [\frac{1}{2}, 1] \end{cases} \tag{1.12}$$

In Figure 1.2 we plot the attackers reward as a function of the relative hashing power q against the probability of the reward for an honest miner with the same hashing power, which is just q.

We note a few things.

- In the range $q \in [0, \dfrac{1}{2}]$ $R(p)$ is identical to $Q(p)$.
- As expected the reward of the attackers exceeds the honest reward at the same point where the probability of success exceeds the honest benchmark. i.e. when $q > 1 - \frac{1}{\sqrt{2}} \sim 0.293$
- As $q \to 1$ the reward of the attackers diverge, because they basically control the blockchain and can plant as many blocks as they desire.
- This may be a little hard to see in Figure 1.2 but The function $R(p)$ is continuous but not continuously differentiable. At the point $q = \frac{1}{2}$ the derivative jumps from 4 to 5.

1.5 Block withholding attack - Type 0

In fact, the attackers have another interesting option. Instead of publishing the secret branch when it is longer than the main branch, they can choose to publish it one step before when the length of the secret branch is the same as the length of the main branch, i.e. when they reach a tie. The reason this is potentially beneficial is that due to latency effects in the bitcoin network (recently discussed in [?]) not all miners share the same view of the entire block-tree at all moments. All honest miners shift their efforts to the longest branch they know of, but for some short period of time different parts of the network may be aware of different, and equally valid, longest branches. In such a case, each sub-network continues mining it's branch until the next block is mined by either one of them and a new block-chain is established[5].

Assume that a fraction $\gamma \in [0,1]$ of the honest miners accept the attackers branch and start mining on top of it. This means that $q+\gamma p$ of the total hashing power is now dedicated to making the attackers branch the longest. With some probability the attacker's branch ends up as the winner. Let us denote the probability that this type of tie strategy succeeds by $S_\gamma(q)$. We can calculate $S_\gamma(q)$ starting the same way as we did when we derived 1.15 but use a Markov chain with boundary condition reflecting a tie instead of wining, and multiply that by the probability of an attacker with hashing power $q+\gamma p$ wining the race starting from that point.

Formally, we want to solve 1.3 with boundary conditions $b_0 = 1, b_\infty = 0$ which is solved by:

$$b_z(q) = \begin{cases} \left(\dfrac{q}{1-q}\right)^z & q \in [0, \frac{1}{2}] \quad \text{and} \quad z = 0, 1, 2 \dots \\ 1 & \text{otherwise} \end{cases} \tag{1.13}$$

Using the same logic used to derive 1.14 we get the probability for a tie

$$T(q) = \sum_{m=0}^{\infty} P_{1,q}(m) b_{1-m}(q) \tag{1.14}$$

resulting in (see details in appendix A.3)

$$T(q) = \begin{cases} 2q & q \in [0, \frac{1}{2}] \\ 1 & q \in [\frac{1}{2}, 1] \end{cases} \tag{1.15}$$

Now the attacker, joined by γ of the honest miners are competing with the rest of the honest miners. The probability to win starting from a tie is thus given by 1.4

$$a_0(q_{eff}) = \begin{cases} \dfrac{q_{eff}}{1-q_{eff}} & q_{eff} \in [0, \frac{1}{2}] \\ 1 & q_{eff} \in [\frac{1}{2}, 1] \end{cases} \tag{1.16}$$

where

$$q_{eff} = q + \gamma p = q + \gamma(1-q) \tag{1.17}$$

The condition $q_{eff} \in [0, \frac{1}{2}]$ translates to $0 \le q \le q_c(\gamma)$ where

$$q_c(\gamma) = \frac{1-2\gamma}{2-2\gamma} \tag{1.18}$$

q_c (depicted in figure 1.3) satisfies $0 \le q_c(\gamma) \le \dfrac{1}{2}$, monotonically decreases with γ and hits 0 when[6] $\gamma = \frac{1}{2}$.

[5] In principle this type of block-chain bifurcation can continue to span multiple blocks, with exponentially decreasing probability.

[6] $q_{eff}(\frac{1}{2}) = \frac{1}{2}(1+q)$ which is bigger than $\frac{1}{2}$ for any q.

Based on all that, the solution to $S_\gamma(q) = T(q) \cdot a_0(q_{eff})$ breaks into three regimes:

$$S_\gamma(q) = \underbrace{T(q)}_{\textit{reach a tie}} \cdot \underbrace{a_0(q_{eff})}_{\textit{win given a tie}} = \begin{cases} 2q \cdot \frac{q_{eff}}{1-q_{eff}} = 2q \cdot \frac{q(1-\gamma)+\gamma}{(1-q)(1-\gamma)} & q \in [0, \frac{1-2\gamma}{2-2\gamma}] \\ 2q & q \in [\frac{1-2\gamma}{2-2\gamma}, \frac{1}{2}] \\ 1 & q \in [\frac{1}{2}, 1] \end{cases} \tag{1.19}$$

Note that if $\gamma \geq \frac{1}{2}$ the first regime does not exist and the solution degenerates to:

$$S_{\gamma \geq \frac{1}{2}}(q) = T(q) \cdot a_0(q_{eff}) = \begin{cases} 2q & q \in [0, \frac{1}{2}] \\ 1 & q \in [\frac{1}{2}, 1] \end{cases} = min(2q, 1) \tag{1.20}$$

In figure 1.4 we plot the probability of a successful type 0 attack for various values of the parameter γ, against the benchmark honest probability of success and the type 1 attack probability of success.

1.5.1 When is a Type 0 attack beneficial

To decide if a block-withholding attack of type 0 is beneficial or not we should compare it to the honest probability of success and to a type 1 attack.

Comparing type 0 to Honest

First note that in the second and third regimes in equation 1.19 (or for any q if $\gamma \geq \frac{1}{2}$) the type 0 attacks is beneficial over the honest strategy for any q, because for any q it holds that $0 \leq q \leq min(2q, 1)$.

In the first regime (i.e. when $q < q_c(\gamma)$) we can find at what value of q the type 0 attacks starts being beneficial over the honest strategy by solving

$$2q \cdot \frac{q(1-\gamma)+\gamma}{(1-q)(1-\gamma)} \geq q \tag{1.21}$$

which gives the condition $q_b(\gamma) \leq q \leq q_c(\gamma)$, where

$$q_b = \frac{1-3\gamma}{3-3\gamma} \tag{1.22}$$

The value of q_b where the attack starts becoming beneficial compared to the honest strategy is plotted in Figure 1.5 and behaves similarly to q_c as given in equation 1.18 except now the scale is 1/3.

Note that if $\gamma = 0$ this type of attack is beneficial only when $q > \frac{1}{3}$ and if $\gamma \geq \frac{1}{3}$ this attack is beneficial for all q.

Taking all three regimes into account we conclude that the type 0 attack is beneficial over the honest strategy when

$$S_\gamma(q) \geq q \Longrightarrow \begin{cases} q > \dfrac{1-3\gamma}{3-3\gamma} & \gamma \in [0, \frac{1}{3}] \\ \mathit{any}\ q & \gamma \in [\frac{1}{3}, 1] \end{cases} \tag{1.23}$$

Next we compare this attack to a type 1 attack.

Comparing Type 0 to Type 1

There are two interesting comparisons one can make between Type 0 and Type 1 attacks.

One is to compare how they match against the honest strategy. Namely, for a given γ do we first hit the regime where a type 0 or a type 1 is beneficial over the honest strategy.

This type 0 strategy wins earlier (in fact already at $q = 0$) when $\gamma \geq \frac{1}{3}$. When $\gamma < \frac{1}{3}$ we can compare $q_b(\gamma)$ (given in equation 1.22) with q_0 (given in equation 1.10):

$$\frac{1-3\gamma}{3-3\gamma} < 1 - \frac{1}{\sqrt{2}} \tag{1.24}$$

which gives $\gamma > \gamma_c$ where

$$\gamma_c = 1 - \frac{2}{3}\sqrt{2} \sim 0.0572 \tag{1.25}$$

Indeed, you can see in Figure 1.4 that the green curve representing $\gamma = 0$ lies below the orange curve which represents the Type 1 strategy, while the red curve representing $\gamma = 0.1 > \gamma_c$ lies above it.

To summarize, when $\gamma \geq \frac{1}{3}$ the Type 0 strategy is beneficial over the honest strategy for any value of q. When $\gamma < \frac{1}{3}$, the hashing power of the attacker needs to exceed a threshold before a block withholding strategy is beneficial. If $\gamma_c \leq \gamma \leq \frac{1}{3}$ we bump into the Type 0 first (the threshold given by $q_b = \frac{1-3\gamma}{3-3\gamma}$), while if $\gamma < \gamma_c$ we bump into Type 1 first (the threshold is given by $q_0 = 1 - \frac{1}{\sqrt{2}}$).

Finally, ignoring the honest strategy for a moment, we can ask for the range of parameters q, γ where the Type 0 strategy is more beneficial than the Type 1 strategy. Formally we need to solve:

$$2q \cdot \frac{q(1-\gamma)+\gamma}{(1-q)(1-\gamma)} \geq \frac{q^2}{1-q}(3-2q)\text{‘} \tag{1.26}$$

which gives the condition

$$2q^2 - q + \frac{2\gamma}{1-\gamma} \geq 0 \tag{1.27}$$

This condition is satisfied in two regimes for γ.

$$S_\gamma(q) \geq Q(q) \Longrightarrow \begin{cases} \mathit{any}\ q & \gamma \in [\frac{1}{17}, 1] \\ q < \frac{1}{4}\left(1 - \sqrt{\frac{1-17\gamma}{1-\gamma}}\right) \quad \mathit{or} \quad q > \frac{1}{4}\left(1 + \sqrt{\frac{1-17\gamma}{1-\gamma}}\right) & \gamma \in [0, \frac{1}{17}] \end{cases} \tag{1.28}$$

1.5.2 The γ, q phase space

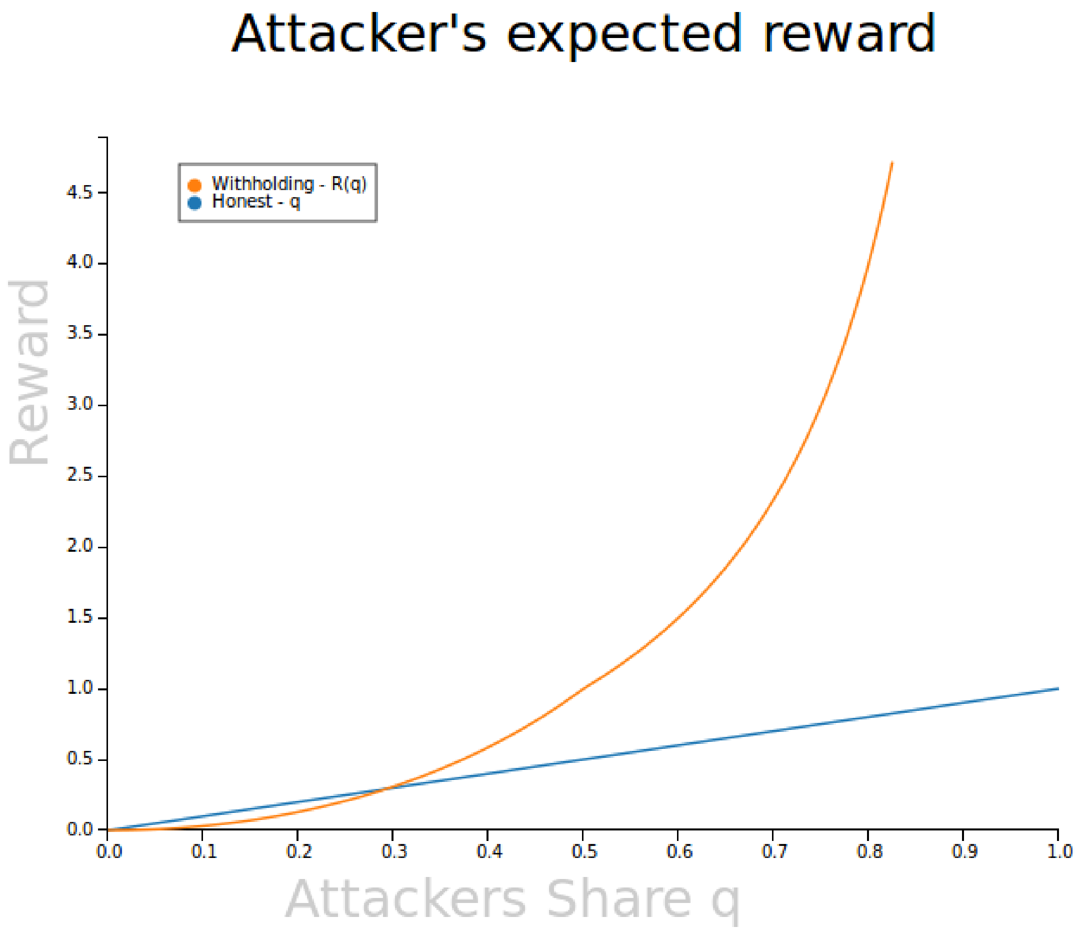

Figure 1.2: The orange curve plots the attacker's reward. The blue curve is the baseline reward for an honest miner with the same hashing power q.

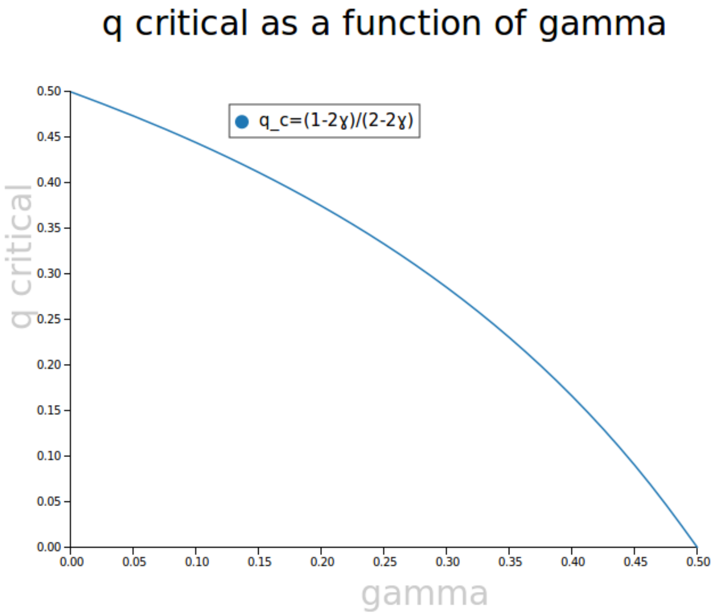

Figure 1.3: The value of q_c as a function of γ.

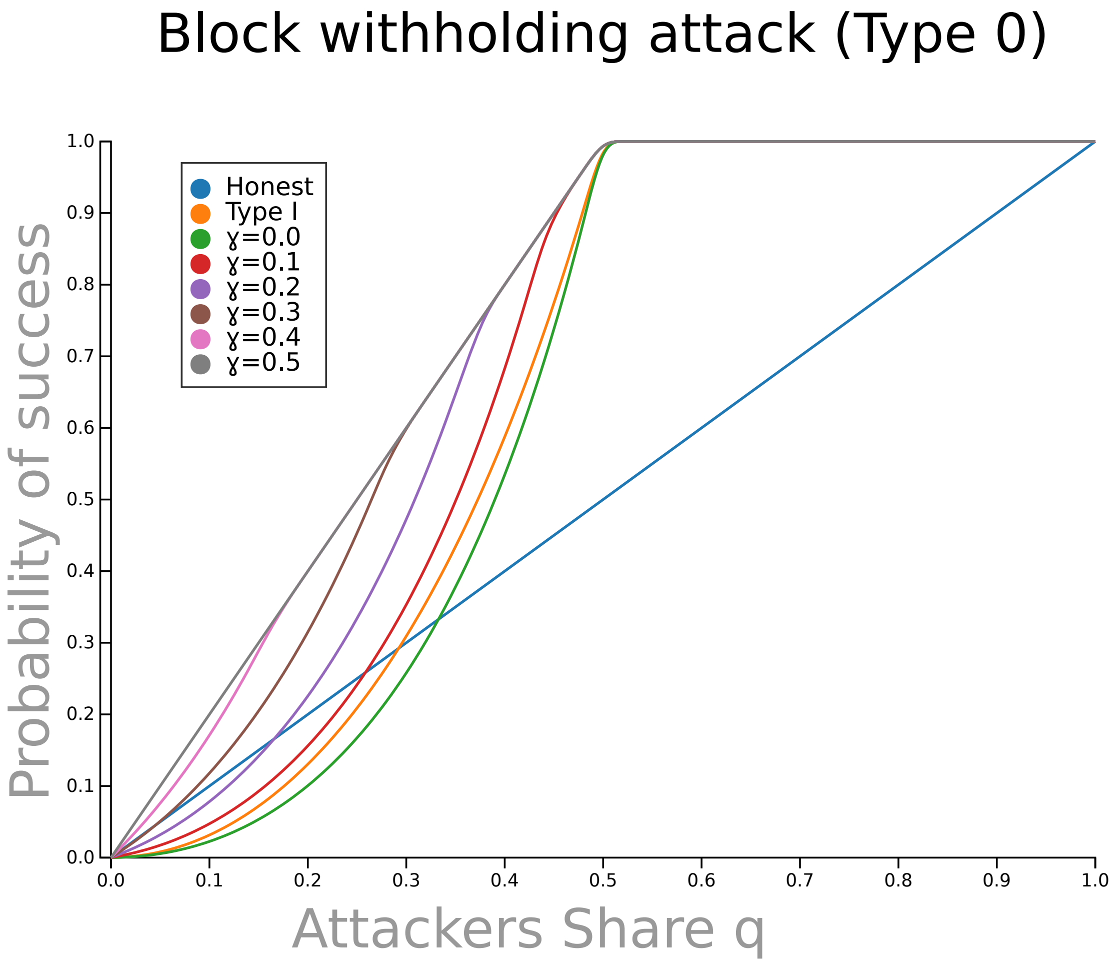

Figure 1.4: The probability of success of a type II block withholding attack as a function of γ. The blue curve plots the honest probability of success, and the orange curve plots the type I attack.

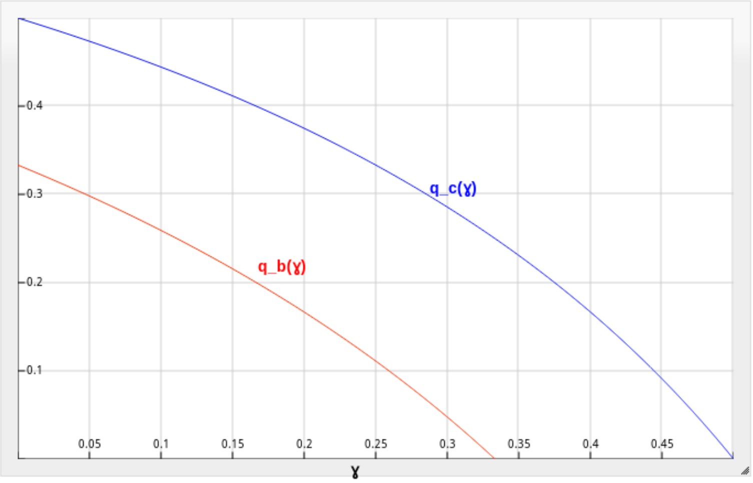

Figure 1.5: The value of q_c and q_b as a function of γ.

Appendix A

Calculation Details

A.1 Probability distribution

To find the normalization in the case $n > 0$ we use the useful binomial identity holding for any complex s inside the unit circle ($|s| < 1$)

$$\frac{1}{(1-s)^n} = \sum_{k=0}^{\infty} \binom{n+k-1}{k} s^k.$$

It is now straightforward to show that 1.2 is indeed a probability distribution

$\sum_{m=0}^{\infty} P(n,m,p) = p^n \sum_{m=0}^{\infty} \binom{n+m-1}{m} q^m = p^n \frac{1}{(1-q)^n} = 1$

A.2 Calculating $Q(q)$

$Q(q)$ can be solved in the two regions for the parameter q given in 1.4.

In the case that $q \in [0, \frac{1}{2}]$

$$Q(q) = \sum_{m=0}^{\infty} P_{1,q}(m) a_{1-m}(q) = p \sum_{m=0}^{\infty} q^m a_{1-m}(p) = p\left(a_1(p) + q a_0(p) + \sum_{m=2}^{\infty} q^m\right) =$$

$$p\left(\left(\frac{q}{p}\right)^2 + q\frac{q}{p} + \left(\sum_{m=0}^{\infty} q^m\right) - 1 - q\right) = p\left(\left(\frac{q}{p}\right)^2 + \frac{q^2 p}{p^2} + \frac{1}{p} - (1+q)\right) =$$

$$\frac{1}{p}\left(q^2(1+p) + p - p^2(1+q)\right) = \frac{1}{p}\left(q^2(1+p) + p - p^2(1+q)\right) =$$

$$\frac{1}{1-q}\left(q^2(2-q) + 1 - q - (1-q)^2(1+q)\right) =$$

$$\frac{1}{1-q}\left(2q^2 - q^3 + 1 - q - 1 + 2q - q^2 - q + 2q^2 - q^3\right) = \frac{q^2}{1-q}(3-2q)$$

In the case that $q \notin [0, \frac{1}{2}]$

$$Q(q) = p\sum_{m=0}^{\infty} q^m a_{1-m}(p) = p\left(a_1(p) + qa_0(p) + \sum_{m=2}^{\infty} q^m\right) =$$
$$p\left(1 + q + \left(\sum_{m=0}^{\infty} q^m\right) - 1 - q\right) = p\frac{1}{p} = 1$$

A.3 Calculating $T(q)$

$T(q)$ can be solved in the two regions for the parameter q given in 1.4.

In the case that $q \in [0, \frac{1}{2}]$

$$Q(q) = \sum_{m=0}^{\infty} P_{1,q}(m) b_{1-m}(q) = p\sum_{m=0}^{\infty} q^m b_{1-m}(p) = p\left(b_1(q) + \sum_{m=1}^{\infty} q^m\right) =$$
$$p\left(\frac{q}{p} + \left(\sum_{m=0}^{\infty} q^m\right) - 1\right) = p\left(\frac{q}{p} + \frac{1}{1-q} - 1\right) = q + 1 - p = 2q$$

and the case $q \notin [0, \frac{1}{2}]$ is identical to $Q(q)$

A.4 Calculating $R(p)$

$R(p)$ can be solved in the two regions for the parameter q given in 1.4.

In the case that $q \in [0, \frac{1}{2}]$

$$R(p) = p\sum_{m=0}^{\infty} m \cdot q^m a_{1-m}(p) = p\left(qa_0(p) + \sum_{m=2}^{\infty} m \cdot q^m\right) =$$
$$p\left(q\frac{q}{p} + \left(\sum_{m=0}^{\infty} m \cdot q^m\right) - q\right) = p\left(\frac{q^2}{p} + q\partial_q\left(\sum_{m=0}^{\infty} q^m\right) - q\right) =$$
$$q^2 + pq\partial_q\left(\frac{1}{1-q}\right) - pq = q^2 + pq\left(\frac{1}{p^2} - 1\right) = q^2 + q\frac{(1+p)(1-p)}{p} =$$
$$q^2\left(1 + \frac{2-q}{1-q}\right) = \frac{q^2}{1-q}(3-2q) = Q(p)$$

In the case that $q \notin [0, \frac{1}{2}]$

$$R(p) = p\sum_{m=0}^{\infty} m \cdot q^m a_{1-m}(p) = p\left(qa_0(p) + \sum_{m=2}^{\infty} m \cdot q^m\right) =$$
$$p\left(q + \left(\sum_{m=0}^{\infty} m \cdot q^m\right) - q\right) = pq\partial_q\left(\frac{1}{1-q}\right) = \frac{q}{1-q}$$

Bibliography