Bitcoin and Cryptocurrency Technologies

Arvind Narayanan, Joseph Bonneau, Edward Felten,
Andrew Miller, Steven Goldfeder

Draft — Apr 10, 2015

Feedback welcome! Email bitcoinbook@lists.cs.princeton.edu

Chapter 5: Bitcoin Mining

This chapter is all about mining. We've already seen quite a bit about miners and how Bitcoin relies on them — they validate every transaction, they build and store all the blocks, and they reach a consensus on which blocks to include in the block chain. We also have already seen that miners earn some reward for doing this, but we still have left many questions unanswered. Who are the miners? How did they get into this? How do they operate? What's the business model like for miners? What impact do they have on the environment? In this chapter, we will answer all of these questions.

5.1 The task of Bitcoin miners

Do you want to get into Bitcoin mining? If you do, we're not going to completely discourage you, but beware that Bitcoin mining bears many similarities to gold rushes. Historical gold rushes are full of stories of young people rushing off to find fortune, and many of them lose everything they have. A few strike it rich, but even those that do generally endure lots of hardship along the way. Flocking to a gold rush isn't easiest way to get rich, and Bitcoin mining is starting to look like a similar proposition. As we'll see in this section, mining is by no means a get-rich-quick scheme.

But first, let's look at the technical details. To be a Bitcoin miner, you have to join the Bitcoin network and connect to other nodes. Once you're connected, there are six tasks to perform:

1. *Listen for transactions.* First, you listen for transactions on the network and validate them by checking the signatures and that the outputs being spent haven't been spent before.
2. *Maintain block chain and listen for new blocks.* You must maintain the block chain. You start by requesting other nodes to give you all of the historical blocks that are already part of the block chain before you joined the network. You then listen for new blocks that are being broadcast to the network. You must validate each block that you receive — by validating each transaction in the block and checking that the block contains a valid nonce. We'll return to the details of nonce checking later in this section.
3. *Assemble a new block.* Once you have an up-to-date copy of the block chain, you begin building your own blocks. To do this, you group transactions that you heard about into a new block that extends the latest block you know about. You must make sure that each transaction included in your block is valid.
4. *Find a nonce that makes your block valid.* This step requires the most work, and it's where all the difficulty really happens for the miners. We will see this in detail shortly.
5. *Hope your block is accepted.* Even if you found a block, there's no guarantee that your block will become part of the consensus chain. There's bit of luck here; you have to hope that other miners accept your block and start mining on top of it, instead of some competitor's block.
6. *Profit.* If all other miners do accept your block, then you profit! At the time of this writing in early 2015, the block reward is 25 bitcoins which is currently worth over $6,000. In addition, if any of the transactions in the block contained transaction fees, the miner collects those too.

We can classify the steps that a miner must take into two categories. Some tasks — validating transactions and blocks — help the Bitcoin network and are fundamental to its existence. These tasks are the reason that the Bitcoin protocol requires miners in the first place. Other tasks — the race to find blocks and profit —- aren't necessary for the Bitcoin network itself but are intended to incentivize miners to perform the essential steps. Of course, both of these are necessary for Bitcoin to function as a currency, since miners need an incentive to perform the critical steps.

Finding a valid block. Let's return to the question of finding a nonce that makes your block valid. In Chapter 3 we saw that there are two main hash-based structures. There's the block chain where each block header points to the previous block header in the chain, and then within each block there's a Merkle tree of all of the transactions included in that block.

The first thing that you do as a miner is you assemble all the transactions that you have from your pending transaction pool into a Merkle tree. You then create a block with a header that points to the previous block. In the block header, there's a 32 bit nonce field, and you keep trying different nonces looking for one that that causes the block's hash to be under the target — roughly, begin with the required number of zeros. A miner may begin with a nonce of 0 and successively increment it by one in search of a nonce that makes the block valid. See Figure 5.1.

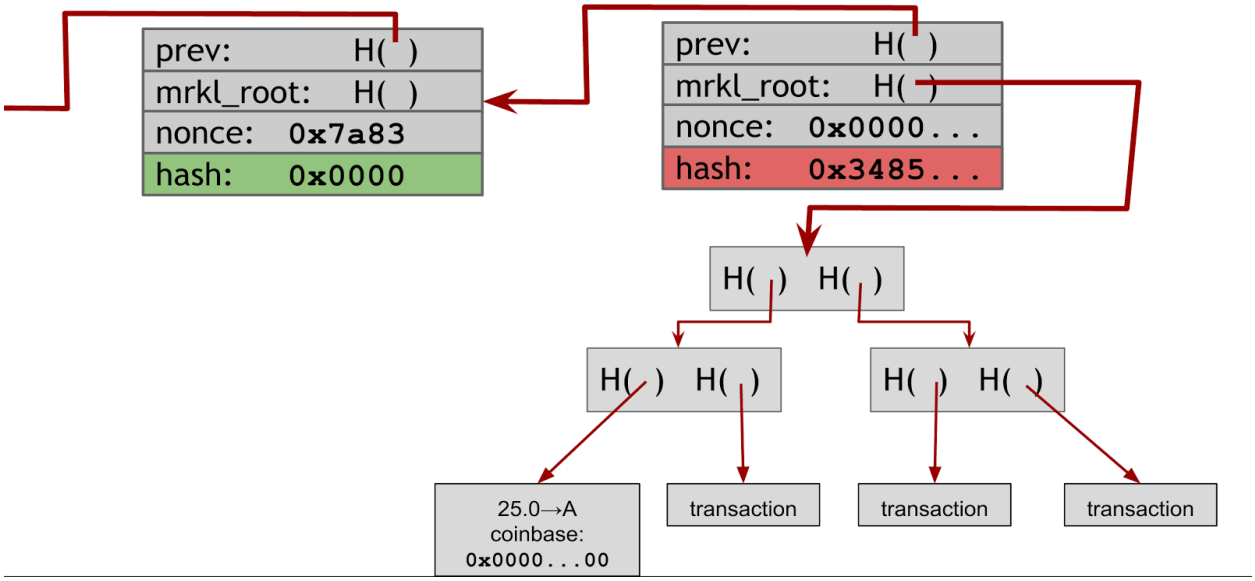


Figure 5.1: Finding a valid block. In this example, the miner tries a nonce of all 0s. It does not produce a valid hash output, so the miner would then proceed to try a different nonce.

In most cases you'll try every single possible 32-bit value for the nonce and none of them will produce a valid hash. At this point you're going to have to make further changes. Notice in Figure 5.1 that there's an additional nonce in the coinbase transaction that you can change as well. After you've exhausted all possible nonces for the block header, you'll change the extra nonce in the coinbase

transaction — say by incrementing it by one — and then you'll start searching nonces in the block header once again.

When you change the nonce parameter in the coinbase transaction, the entire Merkle tree of transactions has to change (See Figure 5.2). So the change of the coinbase nonce will propagate all the way up, and since you'll have to update all the hashes, changing the extra nonce in the coinbase transaction is much more expensive than changing the nonce in the block header. For this reason, miners spend most of the time changing the nonce in the block header and only change the coinbase nonce when they have exhausted all of the 2^{32} nonces in the block header.

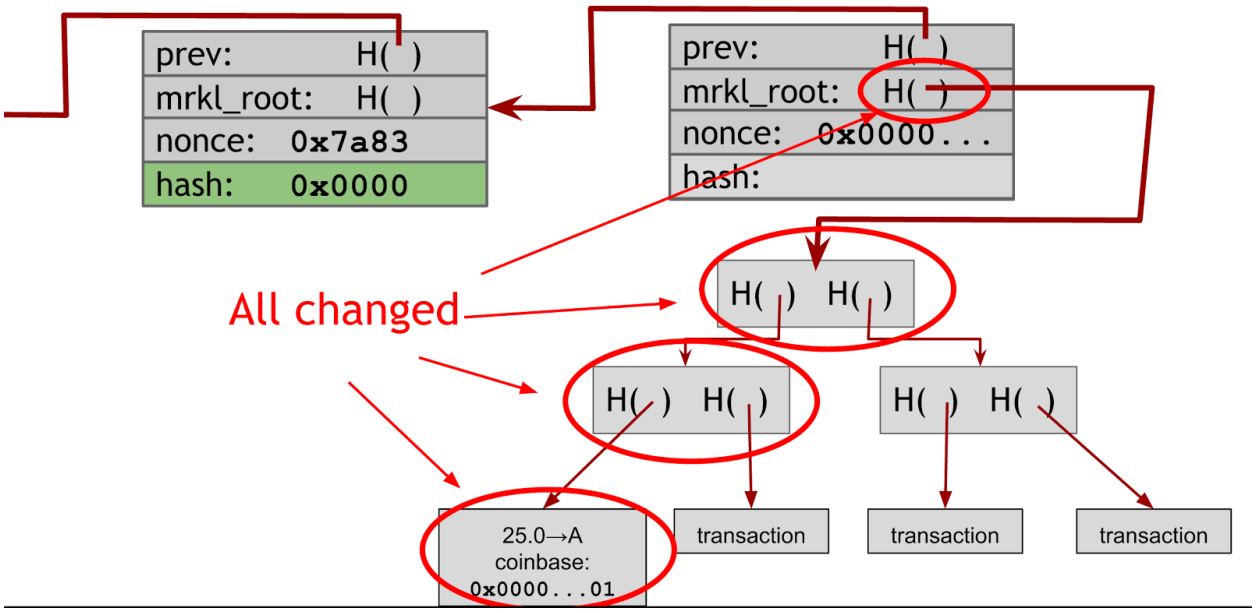


Figure 5.2: Changing a nonce in the coinbase transaction propagates all the way up the Merkle tree.

The vast, vast majority of nonces that you try aren't going to work, but if you stay at it long enough you'll eventually find the right combination of the extra nonce in the coinbase transaction and the nonce in the block header that produce a block with a hash under the target. When you find this, you want to announce it as quickly as you can and hope that you can profit from it.

Is everyone solving the same puzzle? You may be wondering: if every miner just increments the nonces as we described, aren't all miners solving the exact same puzzle? Won't the fastest miner always win? The answer is no! Firstly, it's unlikely that miners will be working on the exact same block as each miner will likely include a somewhat different set of transactions and in a different order. But more importantly, even if two different miners were working on a block with identical transactions, the blocks would still differ. Recall that in the coinbase transaction, miners specify their own address. This change will propagate up causing all the Merkle hashes to change ensuring that no two miners are hashing the same inputs.

Difficulty. Exactly how difficult is it to find a valid block? As of March 2015, the mining difficulty target (in hexadecimal) is:

00000000000000000172EC0000000000000000000000000000000000000000000000

so the hash of any valid block has to be below this value. In other words only one in about 2^{67} nonces that you try will work, and that's a really huge number. One approximation for it that you would think about is it's about the population of the earth squared. So, if every person on Earth was themselves their own planet Earth with seven billion people on it the total number of people would be close to this number.

Determining the difficulty. The mining difficulty changes every 2016 blocks. It is adjusted based on how efficient the miners were over the period of the previous 2016 blocks according to this formula:

next_difficulty = previous_difficulty * (2 weeks) / (time to mine last 2016 blocks)

Two weeks is the amount of time it would take to mine 2016 if a block were created exactly every 10 minutes. So the effect of this formula is to scale the difficulty to maintain the property that blocks should be found by the network on average about once every ten minutes. There's nothing special about 2 weeks, but it's a good trade-off. If the period were much shorter, the difficulty might fluctuate due to random variations in the number of blocks found in each period. If the period were much higher, the network's hash power might get too far out of balance with the difficulty.

Each Bitcoin miner independently computes the difficulty and will only accept blocks that meet the difficulty that they computed. Miners who are on different branches might not compute the same difficulty value, but any two miners mining on top of the same block will agree on what the difficulty should be. This allows consensus to be reached.

You can see in Figure 5.3 that over time the mining difficulty keeps increasing. It's not necessarily a steady linear increase or an exponential increase, but it depends on activity in the market. Things like how many new miners are joining, which in turn may be affected by the current exchange rate of Bitcoin, affect the mining difficulty. Generally, as more and more miners come online, blocks are found faster, and the difficulty is increased so that it again takes ten minutes to find a block.

In Figure 5.3 you can see that in the red line on the graph there's a step function of difficulty even though the overall network hash is growing smoothly. The discrete step results from the fact that the difficulty is only adjusted every 2016 blocks.

Another way to view this is to view how long it takes to find a block on average. Figure 5.4 shows how many seconds elapse between consecutive blocks in the block chain. You can see that this gradually goes down, jumps up and then gradually goes down again. Of course what's happening is that every

2016 blocks the difficulty resets and the average block time goes back up to about ten minutes. Over the next period the difficulty stays unchanged, but more and more miners come online. Since the hash power has increased but the difficulty has not, blocks are found more quickly until the difficulty is again adjusted after 2016 blocks, or about two weeks.

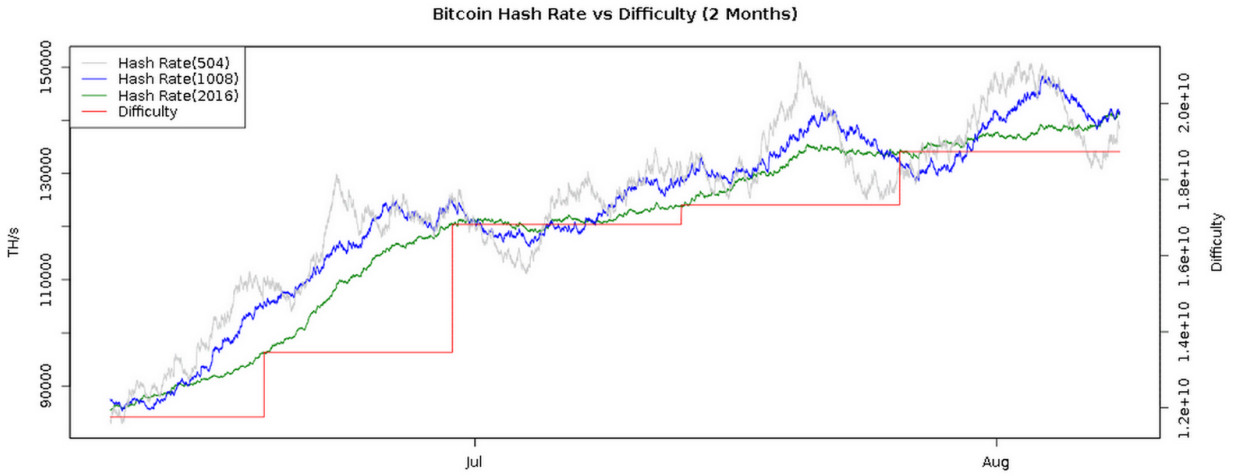


Figure 5.3: Mining difficulty over time (mid-2014). Note that the y-axis begins at 80,000 TH/s.

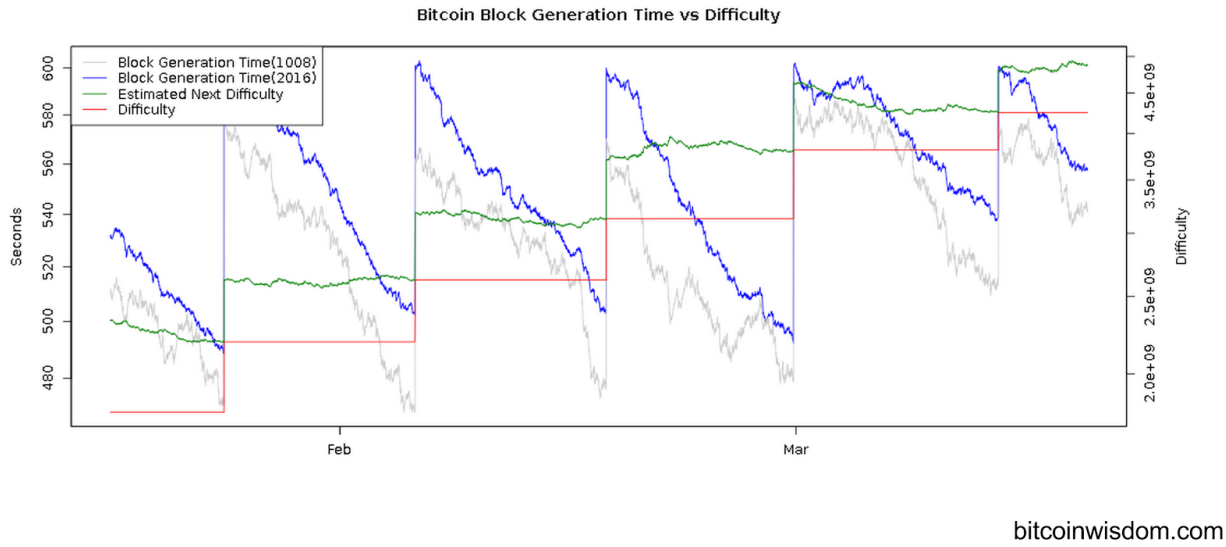


Figure 5.4 : Time to find a block (early 2014). Note that the y-axis begins at 460 seconds.

Even though the goal was for a block to be found every ten minutes on average, it's actually close to about every nine minutes, and at the end of the two week cycle it will get down to around eight

minutes. This behavior is during a period of rapid hash rate increase. If the hash rate isn't increasing as fast, the average time to find a block will be more stable.

There have been *decreases* in difficulty a few times in Bitcoin's history, small in magnitude compared to its increases. One proposed scenario for Bitcoin's collapse is a "death spiral" in which a dropping exchange rate makes mining unprofitable for some miners, causing an exodus, in turn causing the price to drop further. While there have been no catastrophic declines of mining power so far, there's no inherent reason why difficulty must keep increasing.

5.2 Mining Hardware

We've mentioned that the computation that miners have to do is very difficult. In this section, we'll discuss why it is so computationally difficult and take a look at the hardware that miners use to facilitate this computation.

What exactly is this difficult computation that miners are working on? They are computing many, many SHA-256 hashes. We've discussed hash functions and we've mentioned SHA-256 in particular. SHA-256 is a general purpose cryptographic hash function that's part of a bigger family of functions that was standardized in 2001. SHA-256 was a good choice as this was strongest cryptographic hash function available at the time when Bitcoin was designed. It is possible that it will become less secure over the lifetime of Bitcoin, but for now it remains secure. It did come out of the NSA, which has led to some conspiracy theories, but it's generally considered to be a very strong hash function.

Sidebar. Although SHA-256 is generally considered to be cryptographically secure, its replacement, the SHA-3 family, has already been picked. SHA-3 is in the final stages of standardization today, but it wasn't available at the time Bitcoin was designed.

A closer look at SHA-256. Figure 5.5 shows more detail about what actually goes on in a SHA-256 computation. While we don't need to know all of the details to understand how Bitcoin works, we'll give a high level overview so you have a general idea of the task that miners are solving.

SHA-256 maintains 256 bits of state. The state is split into eight 32-bit words which makes it very optimized for 32-bit hardware, and in each round some bitwise tweaks that are applied to some of those words. Then a number of words in the state are taken — some with these tweaks applied — and added together mod 32. The result of all of these additions is wired over to the first word of the state and the entire state shifts over.

Figure 5.5 is just one round of the SHA-256 compression function, and a complete computation of SHA-256 does this for 80 iterations. During each round, there are slightly different constants applied so that every iteration isn't exactly the same.

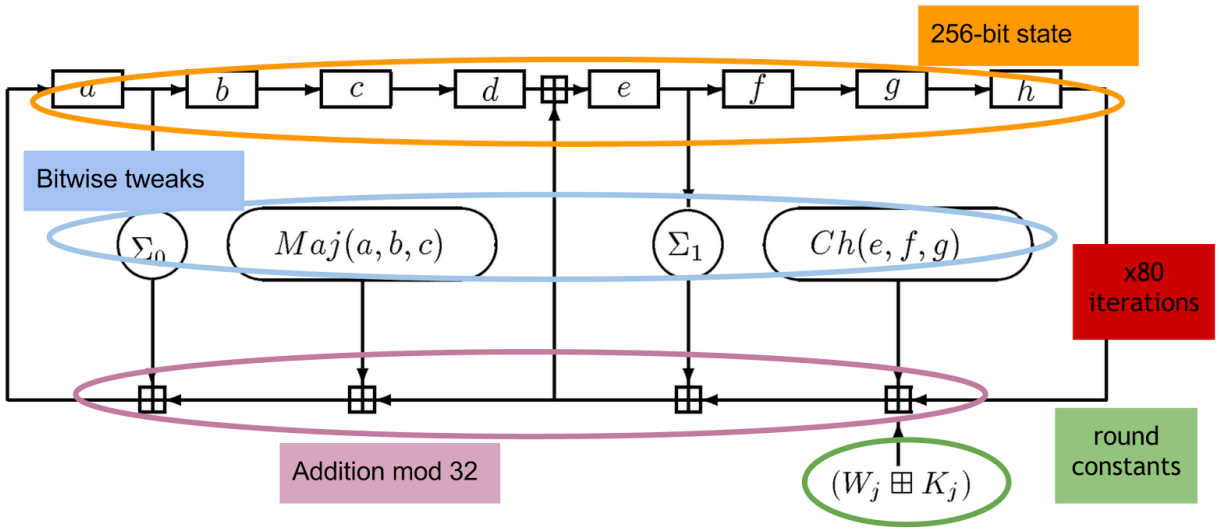


Figure 5.5 : The structure of SHA-256. This is one round of the compression function.

So the task for miners is compute this function. To do this, they need to be able to deal with 32-bit words, do 32-bit modular addition, and also be able to do some bitwise logic. Remember that miners are racing each other so they will want to do this as fast as possible.

As we will see shortly, Bitcoin actually requires SHA-256 to be applied twice to a block in order to get the hash that is used by the nodes. This is a quirk of Bitcoin, and the reason for the double application are not fully specified and seemingly unnecessary, but at this point, it's just something that miners have to deal with.

CPU mining. The first generation of mining was all done on general purpose computers — that is general purpose central processing units (CPUs). In fact, CPU mining was as simple as running the code shown in Figure 5.6. That is, miners simply searched over nonces in a linear fashion, computed SHA 256 in software and checked if the result was a valid block. Also, notice in the code that as we mentioned, SHA-256 is applied twice.

```
while (1) {
      HDR[kNoncePos]++;
      IF (SHA256(SHA256(HDR)) < (65535 << 208) / DIFFICULTY)
            return;
}
```

Figure 5.6 : CPU mining pseudocode.

How fast will this run on a general purpose computer? On a high end desktop PC you can compute about 20 million hashes per second (MH/s). At that speed, it would take you over a hundred thousand years on average at the early-2015 difficulty level to find a block. We talked about how mining was

going to be a difficult slog, and if you're mining on a general purpose PC today it's a really, really big hill to get up because it's going to take you about 300,000 years on average to find a block. CPU mining is no longer profitable with the current difficulty. For the last few years, anyone trying mine on a CPU probably doesn't understand how Bitcoin works and were probably pretty disappointed that they never made any money doing it.

GPU mining. The second generation began when people started to get frustrated with how slow their CPUs were and instead used their graphics card, or graphics processing unit (GPU).

Almost every modern computer has a GPU built in for high performance graphics. They're designed to have high throughput, and also high parallelism, both of which are very useful for Bitcoin mining. Bitcoin mining can be parallelized because you can compute multiple hashes at the same time with different nonces. In 2010, a language called OpenCL was released. OpenCL is a general purpose language to do things other than graphics on a GPU. It's a high level-language, but over time people started tweaking the code even further to run more quickly on specific graphics cards. This paved the way for Bitcoin mining on GPUs.

Mining with graphics cards has some nice properties. For one thing, they're easily available, and they're easy for amateurs to set up. You can order graphics cards online or buy them at most big consumer electronics stores. They're the most accessible high-end hardware that's available to most people. They also have some properties that make them specifically good for Bitcoin mining. They're designed for parallelism so they have a lot of Arithmetic Logic Units (ALUs) in them that you can use in parallel to do different SHA-256 computations, and some of them also have specific instructions to do bitwise operations that work out quite nicely for SHA-256. They also have the property that you can drive many graphics cards from one motherboard and CPU. So you could take your one computer and attach multiple graphics cards to it.

Most graphics cards can also be ***overclocked*** which is a property that gamers demand so you can run them faster than they're actually designed for, if you want to take on the risk. And with Bitcoin mining, it might be a good idea to run the chip much faster than it was designed for even if you introduce some errors in the process. For example, say you can run your graphics card 50 percent faster but doing so will increase the error in the SHA-256 computation to 30 percent of the time. If an invalid solution is erroneously declared valid by the graphics card — something that would happen rarely — you can always double-check it on your CPU. On the other hand, if a valid solution is erroneously missed, you'd never know. But if your speed increase from overclocking can overcome the decrease in output due to errors, you'd still come out ahead. There's a term called ***goodput*** that measures this, which is simply the product of throughput and success rate. In the above example, the throughput is 1.5x compared to not overclocking, whereas the success rate is 0.7x. The product is 1.05, which means overclocking increases the goodput by 5%. People have spent a long time optimizing exactly how much they should overclock a given chip and what errors it would introduce.

As we said earlier, you can control multiple GPUs from a single CPU, and people began taking advantage of this.They would use multiple graphics cards together for mining, and you began to see

some really interesting home-brewed setups like this one shown in Figure 5.7. This was still in the early days of Bitcoin when miners were still mostly hobbyists who didn't know a lot about running a modern data center, but they came up with some quite ingenious designs for how to pack many graphics cards into a small place and keep them cool.

Figure 5.7: A home-built rack of GPUs used for Bitcoin mining. You can also see the fans that they used to build their cooling system. Source: LeonardH, cryptocurrenciestalk.com.

Disadvantages of GPU mining. GPU mining has some disadvantages. GPUs have a lot of hardware built into them for doing video that doesn't get used by miners. Specifically, they have floating point units that aren't used at all in SHA-256, and these are wasted when using them for mining. GPUs also don't have the greatest cooling characteristics when you put a lot of them next to one another. They're not designed to run side by side as they are in the picture; they're designed to be on one graphics card box doing graphics for one computer.

GPUs can also have a fairly large power draw, so a lot of electricity is being used relative to a computer. Another disadvantage initially was that you had to either build your own board or buy expensive boards to house multiple graphics cards.

On a really high-end graphics card with aggressive tuning you might get as high as 200 MH/s, or 200 million hashes per second, an order of magnitude better than you would be doing with a CPU. But even with that improved performance, and even if you're really aggressive and used one hundred

GPUs together, it would still take you over 300 years on average to find a block at the early-2015 difficulty level. Due to this lack of performance, GPU mining is basically dead.

FPGA mining. Around 2011 some miners started to use FPGAs or Field Programmable Gate Arrays. That's around the same time that the first implementation of Bitcoin mining came out in Verilog, a hardware design language that's used to program FPGAs. The rationale behind FPGAs is to try to get close to the performance characteristics of custom hardware while also allowing the owner of the card to customize it or reconfigure it "in the field." This lies in contrast to a chip which is made in a factory and does the same thing forever.

Figure 5.8: A home-built rack of FPGAs. Although you don't see the cooling setup pictured here, a rack like this would need a cooling system.

FPGAs offer better performance than graphics cards, particularly on some of the "bit fiddling" operations. These are easy to specify on an FPGA, and cooling is also easier with FPGAs. You're also wasting less of the card than you would be a graphics card. As with GPUs, you can pack many of these together and drive them from one central unit, and this is exactly what people began to do (see Figure 5.8). Overall, it was possible build a big array of FPGAs more neatly and cleanly than you could with graphics cards.

If you were using an FPGA and using it well, you might get up to a GH/s, or one billion hashes per second. This is certainly a large performance gain over CPUs and GPUs, but even if you had a hundred boards together, each with a 1 GH/s throughput, it would still take you about 50 years on average to find a Bitcoin block at the early-2015 difficulty level.

Despite the performance gain, the days of FPGA mining were quite limited. Firstly, they were being driven harder for Bitcoin mining — by being on all the time and overclocked — than a lot of consumer grade FPGAs were really designed for. Because of this, people found errors and malfunctions in their FPGAs as they were mining. It also turned out to be difficult to optimize the 32 bit addition step which is critical in doing SHA-256. FPGAs are also less accessible to people. You can't buy an FPGA at most stores, and there are few people who know how to program an FPGA or how to set them up.

Even though FPGAs improved performance, the cost-per-performance was only marginally improved over GPUs. FPGA mining was a rather short-lived phenomenon. Whereas GPU mining dominated for about a year or so, the days of FPGA mining were far more limited — lasting only a few months.

At this point you might be wondering that if all of these solutions are so intractable today, what are people actually using? This brings us to ASIC mining.

ASIC mining. Mining today is dominated by Bitcoin *ASICs*, or ***application-specific integrated circuits***. These are chips that were designed, built, and optimized for the sole purpose of mining Bitcoins. There are a few big vendors that that sell these to consumers. There is a good deal of variety in the ASICs that you can buy. You can choose between slightly bigger and more expensive models, more compact models, as well as models with varying performance and energy consumption claims.

Designing ASICs requires a lot of expertise and their lead-time is also quite long. Nevertheless, Bitcoin ASICs were designed and produced surprisingly quickly. In fact, analysts have said that this may be the fastest turnaround time in the history of integrated circuits for specifying a problem and turning it around to have a working chip in people's hands. On the flip side, the first few generations of Bitcoin ASICs were quite buggy, and most of them didn't quite deliver the promised performance numbers. Bitcoin ASICs have since matured, and there are now fairly reliable ASICs being shipped.

Up until 2014, the lifetime of ASICs has been quite short due to the rapidly increasing network hash rate, and thus shipping speed is crucial. Most boards in the ASIC era have been effectively obsolete in about six months. Furthermore, the bulk of the profits are made up front. Often, miners will make half of the expected profits for the lifetime of the ASIC during just the first six weeks. Due to the immaturity of the industry, consumers have often experienced shipping delays, with boards often obsolete by the time they reach the customer. If and when the growth rate of Bitcoin’s hash power stabilizes, mining equipment will have a longer life time.

For much of Bitcoin’s history, the economics of mining haven’t been favorable to the small miner who wants to go online, order mining equipment, and start making money. In fact, in most cases people who have placed orders for mining hardware should have lost money based on the calculation that they made at the time. Until 2013, the price of Bitcoin rose a lot, and this saved most of those customers from losing money. In effect, mining has been an expensive way to simply bet that the price of Bitcoin would rise, and a lot of miners — even though they've made money mining Bitcoins — would have been better off if they had just taken the money that they were going to spend on mining equipment, invested it in Bitcoin, and eventually sold them at a profit.

Today : Professional mining. Today mining has mostly moved away from individuals and toward professional mining centers. Exact details about how these centers operate are not very well known because companies want to protect their setups to maintain a competitive advantage. In Figure 5.9, we see a picture of a professional mining center in the Republic of Georgia.

Figure 5.9: BitFury mining center, a professional mining center in the republic of Georgia.

When determining where to set up a mining center, the three biggest considerations are: climate, cost of electricity, and network position. In particular, you need a cool climate so that cooling bills will be kept low. You need cheap electricity, and you need to be well connected to other nodes in the Bitcoin peer-to-peer network so that you can hear about new blocks as they're announced. Georgia and Iceland have been popular destinations for people setting up Bitcoin mining data centers.

Similarities to gold mining. While 'mining' may seem to be just a cute name, if we zoom out a little bit and think about the evolution of mining, we can see really interesting parallels between Bitcoin mining and gold mining. For starters, both of them led to a similar gold rush mentality when initially a lot of young, amateur folks were eager to get into the business.

Whereas with Bitcoin mining we've seen the slow evolution from CPUs to GPUs to FPGAs, to now ASICs, with gold mining we saw the evolution from individuals with gold pans to small groups of people with sluice boxes, to placer mining — which was a big group of people blowing away hillsides with water — to modern gold mining which utilizes a giant open pit to extract tons of raw material from the earth (See Figure 5.10). Both with Bitcoin and with gold, the friendliness and accessibility to

individuals has gone down over time and there's been a consolidation with large companies controlling most of the operations.

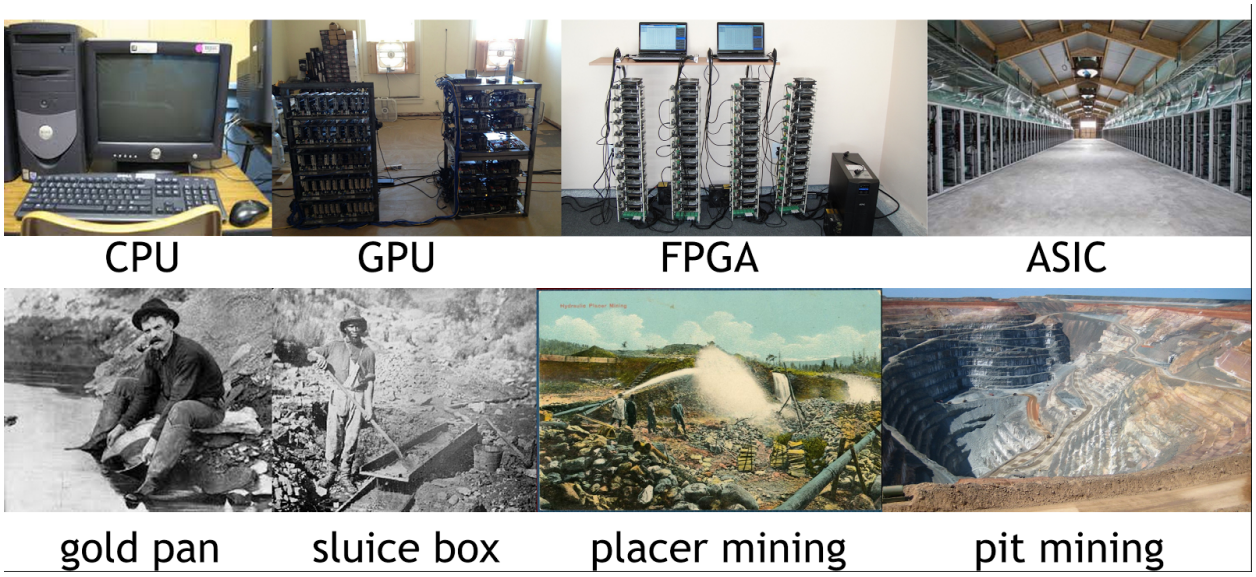


Figure 5.10: Evolution of mining. We can see a clear parallel between the evolution of Bitcoin mining and the evolution of gold mining. Both were initially friendly to individuals and over time became massive operations controlled by large companies.

The future. Currently ASIC mining is the only way to be profitable in Bitcoin and it's not very friendly to small miners. This raises a few questions about what will happen going forward. Are small miners out of Bitcoin mining forever, or is there a way to re-incorporate them? Moreover, does ASIC mining and the development of professional mining centers violate the original vision of Bitcoin which was to have a completely decentralized system in which every individual in the network mined on his or her own computer?

Furthemore, if this is indeed a violation of Satoshi Nakamoto's original vision for Bitcoin, would we be better off with a system in which the only way to mine was with CPUs? In chapter 8, we'll consider these questions and look at alternative forms of mining and how to design mining in a way that is less friendly to ASICs.

5.3 Energy consumption & ecology

We saw how large professional mining data centers have taken over the business of Bitcoin mining, and how this parallels the movement to pit mining in gold mining. You may be aware that a huge concern for environmentalists over the years has been how much damage are these pit mines doing to the environment. Now, Bitcoin is not quite at that level yet, but it is starting to use a significant

amount of energy which has become a topic of discussion. In this section we'll see how much energy Bitcoin mining is using and what the implications are for both the currency and the planet.

Thermodynamic limits. There's a physical law known as ***Landauer's principle*** developed by Ralph Landauer in the 1960s that states that any non-reversible computation must use a minimum amount of energy. Logically irreversible computations can be thought of as losses of information. Specifically, the principle states that erasing each bit must consume a minimum of (kT ln 2) joules, where k is the Boltzmann constant (approximately 1.38×10^{-23} J/K), T is the temperature of the circuit in kelvins, and ln 2 is the natural logarithm of 2, roughly 0.69. As you can see, that's an astronomically small amount of energy per bit.

This is derived from basic physics. We're not going to go through the derivation here, but the high-level idea is that every time you flip one bit in a non-reversible way there's a minimum amount of joules that you have to use. Energy is never destroyed; it's converted from one form into another. In the case of computation the energy is mostly transformed from electricity, which is useful, high-grade energy, into heat which is dissipated into the environment.

Now, of course, SHA-256 being a cryptographic hash function is not a reversible computation, and recall from Chapter 1 that this is a basic requirement of cryptographic hash functions. So, since non-reversible computation has to use some energy and SHA-256 — the basis of Bitcoin mining — is not reversible, energy consumption is an inevitable fact of doing Bitcoin mining. That said, the limits placed by Landauer's principle are far, far below the amount of electricity that is being used today. We're nowhere close to the theoretical optimal consumption of computing, but even if we did get to the theoretical optimum we would still be using energy to perform Bitcoin mining.

So why does Bitcoin mining require energy? There are three steps in the process that requires energy, and some of them may not be so obvious.

1. Embodied energy. First, Bitcoin mining equipment needs to be manufactured.This requires physical mining of raw materials as well as turning these raw materials into a Bitcoin mining ASIC, both of which require energy. This is the embodied energy. As soon as you receive a Bitcoin mining ASIC in the mail, you've already consumed a lot of energy — including the shipping energy, of course — before you've even turned it on and tried to mine bitcoins!

Hopefully, over time the embodied energy will go down, as less and less new capacity comes online. As fewer people are going out to buy new mining ASICs, they're going to be obsoleted less quickly, and the embodied energy will be amortized over years and years of mining.

2. Electricity. When your ASIC powered on and mining, it consumes electricity. This is the step that we know has to consume energy due to Landauer's principle. As mining rigs get more efficient, the electrical energy cost will go down. But because of Landauer's principle, we know that it will not disappear; electrical energy consumption will be a fact of life for Bitcoin miners forever.

3. Cooling. A third important component of mining that consumes energy is cooling off your equipment to make sure that it doesn't malfunction. If you're operating in a very cold climate your cooling cost might be very low, but in most climates you're going to have to pay extra to cool off your equipment from all of the waste heat that it is generating. Generally, the energy used to cool off mining equipment will also be in the form of electricity.

Mining at scale. Both embodied energy and electricity decrease when operating at a large scale. If you're running a large mining data center, you can do it more efficiently. It's cheaper to build chips that are designed to run in a large data center, and you can deliver the power more efficiently as you don't need as many power supplies.

When it comes to cooling, however, the opposite is true. Cooling actually costs mores the larger your scale is. If you want to run a very large operation and have a lot of Bitcoin mining equipment all in one place, there's less air for the heat to dissipate into in the area surrounding your equipment. Your cooling budget is going to therefore increase because cooling that big mass is going to be much more difficult.

Estimating energy usage. How much energy is the entire Bitcoin system using? Of course, we can't compute this precisely because it's a decentralized network with miners operating all over the place without documenting exactly what they're doing. But there are two basics approaches to estimating how much energy Bitcoin miners are using collectively. We'll do some back-of-the-envelope calculations here based on early 2015 values. We must emphasize that these figures are very rough, both because some of the parameters are hard to estimate and because they change quickly. At best they should be treated as order-of-magnitude estimates.

Top down approach. The first approach is a top down approach. We start with the simple fact that every time a block is found today 25 bitcoins of rewards, or about 6,500 dollars are given to the miners. That's about 11 dollars every second, being created out of thin air in the Bitcoin economy and given to the miners.
Now let's ask this question: if the miners are turning all of those 11 dollar per second into electricity, how much can they get? Of course miners aren't actually spending all of the revenue on electricity, but this will provide an upper bound on the electricity being used. Electricity prices vary greatly, but we'll estimate that electricity costs around 10 cent per kilowatt-hour (kWh) at an industrial rate in the US, or equivalently 3 cents per megajoule (MJ). If Bitcoin miners were spending all 11 dollars per second of earnings buying electricity, they could purchase 367 megajoules per second, or 367 megawatts (MW).

Sidebar. In the International System of Units (SI), energy is measured in ***joules***. A ***watt*** is a unit of power, where one ***watt*** is defined as one joule per second.

Bottom up approach. A second way to estimate the cost is to use a bottom up approach. In this approach, we look at the number of hashes the miners are actually computing, which we know by

observing the difficulty of each block. If we then assume that all miners are using the most efficient hardware, we can derive a lower bound on the electricity consumption.

Currently, the best claimed efficiency figure amongst commercially available mining rigs is about 3 GH/s/W. That is, they can do three billion block hashes per second while consuming 1 watt of power. The total network hashrate is about 350,000,000 GH/s, or equivalently 350 petahertz (PH/s). Multiplying these two together, we see that it takes about 117 MW to produce that many hashes per second at that efficiency. Of course this figure excludes all of the cooling energy and all of the embodied energy that's in those chips, but we're doing an optimal calculation and deriving a lower bound so that's okay.

Combining the top down and bottom up approaches, we can derive a ballpark estimate of the amount of power being used for Bitcoin miners.

You probably don't have a great intuitive sense of how much power this actually is. How much is a megawatt? To build up intuition, let's see how much big power plants produce. One of the largest power plants in the world, the Three Gorges Dam in China is a 10,000 MW power plant. A typical large hydroelectric power plant produces around 1,000 MW. Kashiwazaki-Kariwa, the largest nuclear power plant in the world is a 7,000 MW plant, whereas the average nuclear power plant is about 4,000 MW. A major coal fire plant produces about 2,000 MW.

According to our estimates then, the whole Bitcoin network is consuming maybe 10% of a large power plant's worth of electricity. Although this is not an insignificant amount of power, it's not yet a large amount of electricity compared to all the other things that people are using electricity for on the planet.

> Any payment system requires energy and electricity. With traditional currency, lots of energy is consumed guarding and moving gold bullions around, running ATM machines, coin sorting machines, cash registers, and payment processing services, and transporting money in armored cars.

Some people say Bitcoin wastes energy because the energy expended computing SHA-256 hashes doesn't serve any apparent purpose. But you could make this same argument for traditional currency as well — there's a lot of energy being wasted and it doesn't serve any purpose besides maintaining the currency system. So, if we value Bitcoin as a useful currency system, then the energy required to support it is not really being wasted.

Repurposing energy. That said, we can ask if there's a way to do better. One idea is to capture the heat generated from Bitcoin mining do something useful with it instead of just heating up the atmosphere. This is called the data furnaces model. The concept is that instead of buying a traditional electric heater to heat your home, or to heat water in your home, you'd buy a Bitcoin mining rig that you would plug in both to your electricity outlet and also to your Internet connection. Your heater

would mine bitcoins and generate heat as a byproduct of that computation. It turns out that the efficiency of doing this isn't much worse than buying an electric heater.

There are a few things about this model that aren't ideal. Although it's about as efficient as using an electric heater, electric heaters are themselves much less efficient than gas heaters. Besides, what happens when everybody turns off their Bitcoin mining rig in the summer? Will mining hash power go down seasonally based on how much heat people need? Will it go way down on days that happen to be warmer than average? This would be really interesting to observe if the data furnace model actually caught on.

The question of ownership is also not clear. If you buy a Bitcoin data furnace, do you own the Bitcoin mining rewards that you get, or does the company that sold them to you? Most people don't have any interest in Bitcoin mining — and probably never will — so it might make more sense to buy it as an appliance and have the company that sold it to you keep the rewards.

Open questions. There are a number of other open questions regarding Bitcoin's energy consumption and its implications.

Will Bitcoin drive out electricity subsidies? In many countries around the world, the government subsidizes electricity — particularly industrial electricity — and one of the reasons they do so is to try to encourage industry to be located in their country. But Bitcoin provides a good way to turn electricity into cash, and this might cause governments to rethink that model. Such subsidies are intended to attract businesses that will contribute to the country's economy, and subsidizing Bitcoin mining arguably doesn't have the intended effect.

Will Bitcoin require people to guard power outlets? Consider universities or corporate building with lots of power outlets. People may try to plug in mining equipment so that they can profit while someone else is paying the electricity bill. In fact, they might use outdated hardware and not bother to upgrade, considering that they will not be paying the electricity bill. Will such locations need security cameras to make sure that people don't plug in Bitcoin mining equipment into un-monitored power outlets and let them run?

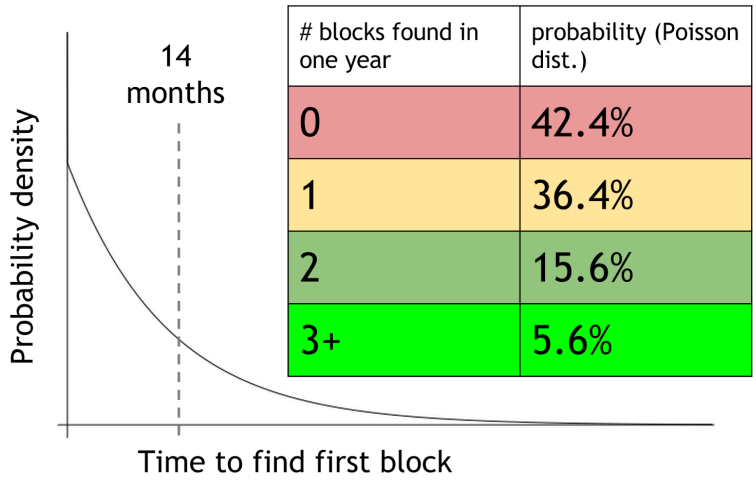
Could we make a currency that didn't have proof of work and didn't have to use so much electricity? This is a question that people are quite interested in, and we'll discuss this in great detail in Chapter 8.

5.4 Mining pools

Consider the economics of being a small miner. Say you're an individual who spent 6,000 dollars of your hard-earned money to buy a nice, shiny, new Bitcoin mining rig. Say that the performance is such that you expect to find a block every 14 months with on average this fancy new rig, and remember that a block is worth about 6,500 dollars as of early 2015.

If you amortize that you could say that the expected revenue of your miner is about 400 dollars per month once you factor in electricity and your other cost of operating it. If you actually got a check in the mail every month for 400 dollars, you'd be quite happy, and it would make a lot of sense to buy the mining rig. But remember that mining is a random process. You don't know when you're going to find the next block. It's a completely random search, and you could find your next block at any time.

High variance. If we look at the distribution of how many blocks you're likely to find in the first year, the variance is pretty high and the expected number of blocks that you'll find is quite low. The distribution is a ***Poisson distribution***, there's a greater than 40% chance that you won't find any blocks within the first year. For an individual miner, this can be devastating. You spent thousands of dollars on the miner, paid lots in electricity to run it, and received nothing in return. There's a roughly 36% chance that you'll find one block within the first year which means maybe you're barely scraping by, provided your electricity costs weren't too high. Finally, there's a smaller chance that you'll find two or more blocks, in which case you could make a nice profit.



# blocks found in one year	probability (Poisson dist.)
0	42.4%
1	36.4%
2	15.6%
3+	5.6%

Figure 5.11: Illustration of uncertainty in mining. We're assuming that the global hash rate is constant and the mean time to find a block is 14 months.

The main point here is that even though on expectation you'll be doing okay — that is, enough to make a return on your money — the variance is sufficiently high that there's a big chance that you'll make nothing at all. For a small miner than, this is essentially a big game of roulette.

Mining pools. Historically, when small business people faced a lot of risk, they formed mutual insurance companies to lower that risk. Farmers, for example, would get together and agree that if any individual farmers barn burnt down they would share profits with that farmer. Can we have a mutual insurance model that works for small Bitcoin miners?

A mining pool is exactly that — mutual insurance for Bitcoin miners. A group of miners will get together, form a pool, and they will all attempt to mine a block with a designated coinbase recipient. That recipient is going to be called the pool manager. So, no matter who actually finds the block, the pool manager will receive the rewards. The pool manager will take that revenue and distribute it to all the participants in the pool based on how much work each participant actually output. Of course, the pool manager will also probably take some kind of cut for their service of managing the pool.

Bitcoin miners lower their variance by joining pools, but how does a pool manager know how much work each member of the pool is actually performing? How can the pool manager divide the revenue commensurate with the amount of work each miner is doing? Obviously the pool manager doesn't want to just take everyone's word for it because people might claim that they've done more than they actually did.

Mining shares. There's an elegant solution to this problem. Miners prove realistically how much work they're doing by outputting shares, or near-valid blocks. Say the target is a number beginning with 67 zeros. The hash must be lower than the target for the block to be valid. In the process of searching for such a block, miners will find blocks with hashes beginning with a lot of zeros, but not quite 67. Miners can show these nearly valid blocks to prove that they are indeed working. A share might require say 40 or 50 zeros, depending on the type of miners the pool is geared for.

```
4AA087F0A52ED2093FA816E53B9B6317F9B8C1227A61F9481AFED67301F2E3FB
D3E51477DCAB108750A5BC9093F6510759CC880BB171A5B77FB4A34ACA27DEDD
00000000008534FF68B98935D090DF5669E3403BD16F1CDFD41CF17D6B474255
BB34ECA3DBB52EFF4B104EBBC0974841EF2F3A59EBBC4474A12F9F595EB81F4B
00000000002F891C1E232F687E41515637F7699EA0F462C2564233FE082BB0AF
0090488133779E7E98177AF1C765CF02D01AB4848DF555533B6C4CFCA201CBA1
460BEFA43B7083E502D36D9D08D64AFB99A100B3B80D4EA4F7B38E18174A0BFB
0000000000000000078FB7E1F7E2E4854B8BC71412197EB1448911FA77BAE808A
652F374601D149AC47E01E7776138456181FA4F9D0EEDD8C4FDE3BEF6B1B7ECE
785526402143A291CFD60DA09CC80DD066BC723FD5FD20F9B50D614313529AF3
000000000041EE593434686000AF77F54CDE839A6CE30957B14EDEC10B15C9E5
9C20B06B01A0136F192BD48E0F372A4B9E6BA6ABC36F02FCED22FD9780026A8F
```

Figure 5.12: Mining Shares. Miners continually try to find blocks that hash below the target. In the process, they'll find other blocks whose hashes contain fewer zeros — but still rare enough to prove that they have been working. In this figure, the dull green lines are from shares, while the bright green hash is from a valid block.

Periodically the pool manager will collect transactions and assemble them into a block. The manager will include his or her own address in the coinbase transaction, and send the block to all of the

participants in the pool. All pool participants work on this block, and they prove that they've been working on it by sending in shares.

When a member of the pool finds a valid block, he sends it to the pool manager who distributes the reward in proportion to the amount of work done. The miner who actually finds the block is not awarded a special bonus, so if another miner did more work than this miner, that other miner will be paid more. See Figure 5.13.

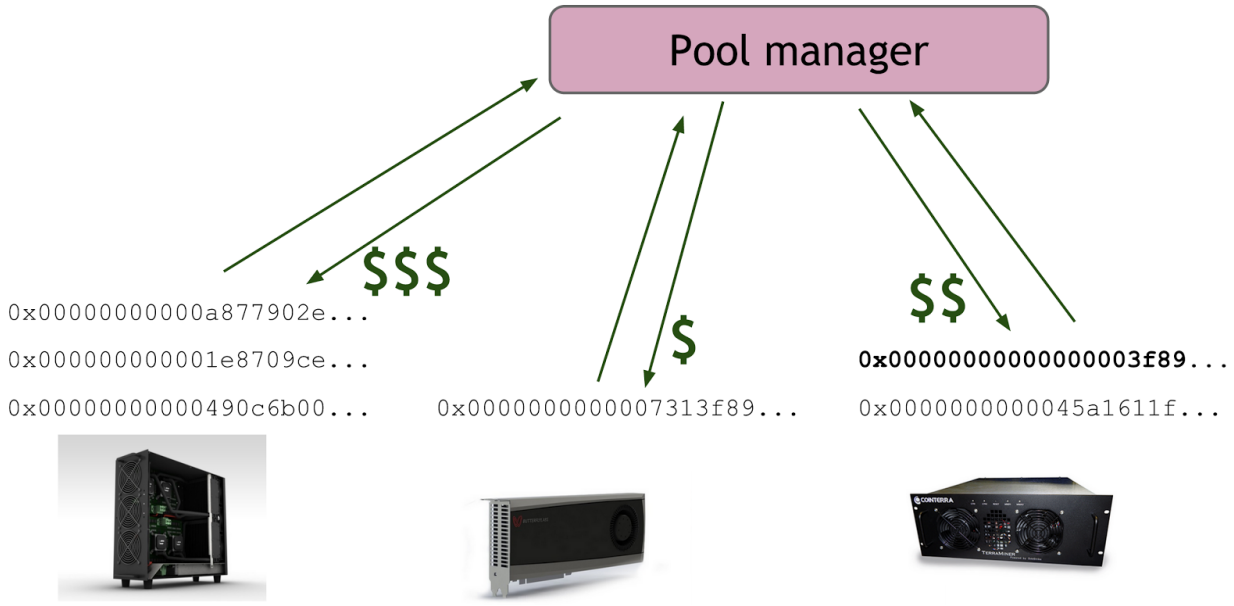


Figure 5.13: Mining rewards. Three participants pictured here are all working on the same block. They are awarded commensurate with the amount of work done. Even though the miner on the right was the one to find the valid block, the miner on the left is paid more since this miner did more work. There is no bonus paid to the miner who actually finds the block.

There are a few options for how exactly the pool manager calculates how much to pay each miner based on the shares they submit. Let's look at two of the common, simpler ones. There are many others that are also used, but these will illustrate the trade-offs between reward schemes.

Pay-per-share. In the pay per share model, the pool manager pays a flat fee for every share above a certain difficulty for the block that the pool is working on. In this model, miners can send their shares to the pool manager right away and get paid without depending on the pool to find a block.

In some ways, the pay-per-share model is the best for miners. They are guaranteed a certain amount of money every time they find a share. The pool manager essentially absorbs all of the risk since he must pay rewards even if a block is not found. Of course, as a result of the increased risk, in the pay-per-share model, the pool manager will charge higher fees as compared with other models.

One problem with the pay-per-share model is that miners don't actually have any incentive to send valid blocks to the pool manager. That is, they can discard valid blocks, and they will still be paid the same rewards, but will cause a big loss to the pool. A malicious pool manager might attack a competing pool in this fashion to drive them out of business.

Proportional. In the proportional model, instead of paying a flat fee per share, the amount of the share the depends on whether or not the pool actually found a valid block. So every time a valid block is found the rewards from that block are distributed to the members proportional to how much work they actually did.

In the proportional model, the miners still bear some risk proportional to the risk of the pool in general. But if the pool is large enough, the variance of how often the pool finds blocks will be fairly low. Proportional payouts provides lower risk for the pool manager. Proportional mining also gets around the problem that we mentioned with the pay-per-share model. Miners are incentivized to send in the valid blocks that they find because that triggers revenue coming back to them.

The proportional model requires more work on behalf of the pool managers to verify, calculate, and distribute rewards as compared to the flas pay-per-share model.

Mining pools first started around 2010 in the graphics card era or Bitcoin mining. They instantly became very popular for the obvious reason that they lowered the variance for the participating miners. They've become quite advanced now. There are many protocols for how to run mining pools, and it has even been suggested that these mining pool protocols should be standardized as part of Bitcoin itself. That is, just like there's a Bitcoin protocol for running the peer-to-peer network, these protocols are a communication API from the pool manager to all of the members the details of the block to work on, and for the miners to send back to the pool manager the shares that they're finding. Some mining hardware actually supports these protocols at the hardware level. Now this makes it very simple to buy a piece of mining hardware and join a pool. You just plug it into the wall — both the electricity and your network connection — choose a pool, and then it will start immediately getting instructions from the pool, mining and converting your electricity into money.

51% mining pools. As of early 2015, the vast majority of all miners are mining through pools. Very few miners mine on their own anymore. In June 2014, Ghash.io, the largest mining pool, got so big that it actually had over 50% of the entire capacity over the Bitcoin network. Essentially Ghash offered such a good deal to participating miners that the majority wanted to join.

This is something that people had feared for a long time, and there was a backlash against them. By August, Ghash had gone down a little bit, partly by design. Still, two mining pools controlled about half of the power in the network.

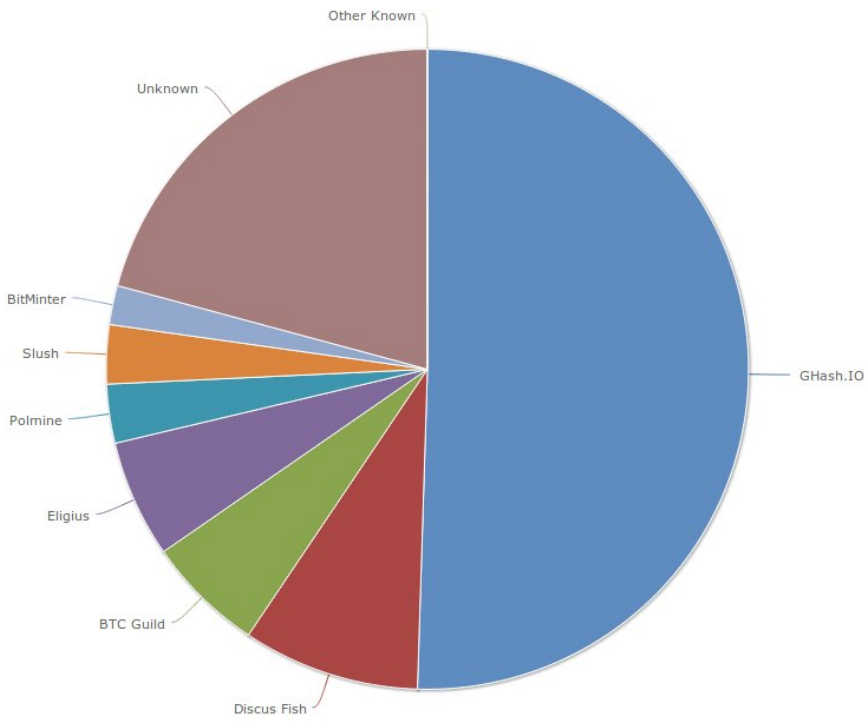


Figure 5.14 (a) Hash power by mining pool, via blockchain.info (June 2014)

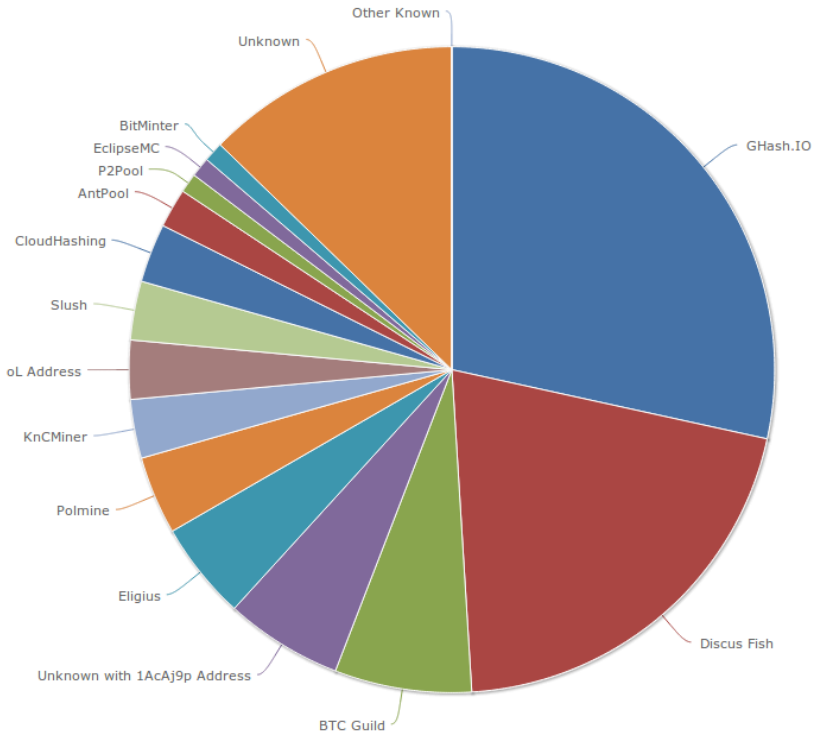


Figure 5.14 (b) Hash power by mining pool, via blockchain.info (August 2014)

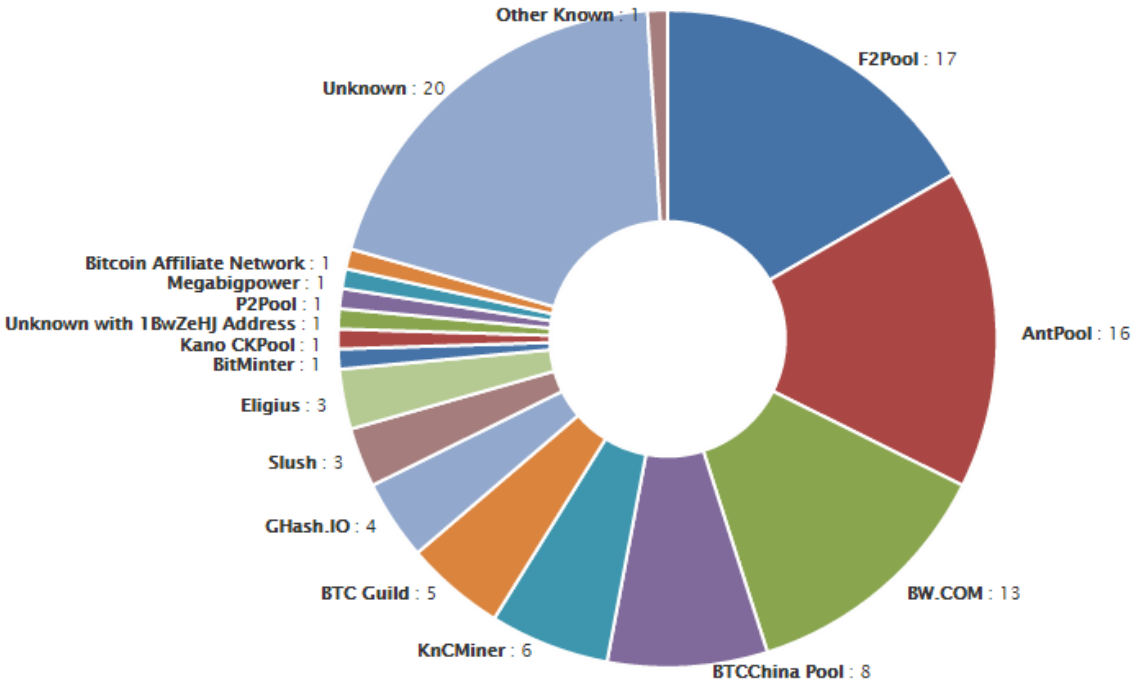


Figure 5.14 (c) Hash power by mining pool, via blockchain.info (April 2015)

In April 2015, the situation looks very different, and less concentrated. The possibility of a pool acquiring 51% is still a concern in the community, but perhaps less so. Due to new miners and pools entering the market and the ease of switching between pools for miners, the market share of different pools remains fluid. It remains to be seen how things will evolve in the long run.

Are mining pools a good thing? The advantages of mining pools are that they make mining much more predictable for the participants and they make it easier for smaller miners to get involved in the game. Without mining pools, the variance would make mining infeasible for small miners.

Another advantage of mining pools is that since there's one central pool manager who is sitting on the network and assembling blocks it makes it easier to upgrade the network. By upgrading the software that the mining pool manager is running that effectively updates all of the software that all the pool members are running.

The main disadvantage of mining pools, of course, is that they lead to centralization. It's an open question how much power the operators of a large mining pool actually have. Of course miners are free in theory to leave a pool if it is perceived as too powerful, but it’s unclear how often miners do so in practice.

Another disadvantage of mining pools is that it lowers the population of people actually running a fully validating Bitcoin node. Previously all miners, no matter how small, had to run their own fully

validating node. They all had to store the entire block chain and validate every transaction. Now, most miners offload that task to their pool manager, and this is one reason why as we mention in Chapter 3, the number of fully validated nodes may actually be going down in the Bitcoin network.

If you're concerned about the level of centralization introduced by mining pools, you might ask: could we redesign the mining process so that we don't have any pools and everybody has to mine for themselves? We'll consider this question in Chapter 8.

5.5 Mining incentives and strategies

We've spent most of this chapter describing how the main challenge of being a miner is getting good hardware, finding cheap electricity, getting up and running as fast as you can, and hoping for some good luck. But it turns out that there are also some interesting strategic considerations that every miner has to make before they pick which blocks to work on.

1. *Which transactions to include.* Miners get to choose which transactions they want to include in a block. The default strategy is to include any transaction that includes higher than some minimum transaction fee.
2. *Which block to mine on.* Miners also get to decide on top of which block they want to mine. The default behavior for this decision is to extend the longest valid chain.
3. *Choosing between blocks at the same height.* If two different blocks are mined and announced at around the same time, it results in a 1-block fork, with either block admissible under the longest valid chain policy. Miners then have to decide which block to extend. The default behavior is to build on top of the block that they heard about first.
4. *When to announce new blocks.* When they find a block, miners have to decide when to announce this to the Bitcoin network. The default behavior is to announce it immediately, but they can choose to wait some time before announcing it.

As we see, miners are faced with many decisions. For each decision we mentioned a default strategy. This is the strategy employed by the Bitcoin reference client, which is run by the vast majority of miners at the time of this writing.

But depending on the fraction of mining power controlled by a miner, it may be possible that a non-default strategy is more profitable. Finding such scenarios and strategies is an active area of research. Let’s look at several such potentially profitable deviations. In the following discussion, we’ll assume there’s a deviant miner who controls some fraction of mining power which we’ll denote by α.

Forking attack. The simplest attack is a forking attack, and the obvious way to profit from this attack is to perform a double spend. The miner sends some money to a victim, Bob, in payment for some good or service. Bob waits and sees that the transaction paying him has indeed been included in the

block chain (perhaps he follows the common heuristic and even waits for six confirmations to be sure). Convinced that he has been paid, Bob ships the good or performs the service.

The miner now goes ahead and begins working on an earlier block — before the block that contains the transaction to Bob. In this forked chain, the miner inserts an alternate transaction — or a double spend — of the coins paid to Bob back to the miner's own address.

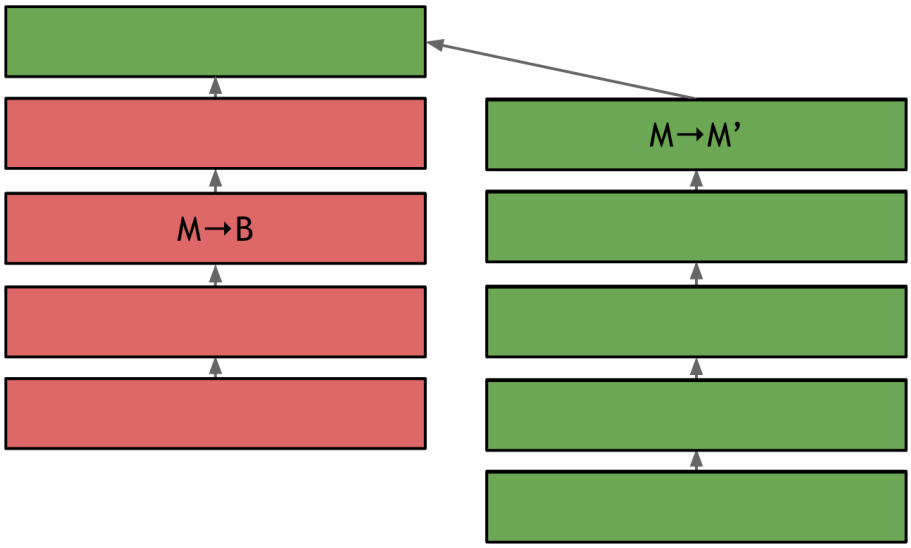


Figure 5.15 Forking attack.

When the miner initially goes back and works on an earlier point in the chain, the doesn't immediately succeed since the forked chain is not the longest chain. However, if the miner has a majority of the hash power — that is, if $\alpha > 0.5$ — the alternate chain eventually become the longest chain, and hence the valid block chain. Once this occurs, the transaction paying Bob no longer exists on the consensus block chain. Moreover, since those coins have already been spent (on the new consensus chain), that transaction can no longer make its way onto the block chain.

Is 51% necessary? Launching a forking attack is certainly possible if $\alpha > 0.5$. In practice, it might be possible to perform this attack with a bit less than that because of other factors like network overhead. The non-attacker chain will have some stale blocks for the usual reason: there is a latency for miners to hear about each others' blocks. But the attacker can avoid most of this latency within his chain. Similarly, the attack gets easier the further over 50 percent you go. People often talk about a 51 percent attacker as if 51% is a magical threshold that suddenly enables a forking attack. In reality, it's more of a gradient.

It's not clear whether a forking attack can actually succeed in practice. The attack is detectable, and it's possible that that the community would decide to reverse the attack by refusing to accept the alternate chain even though it is longer. Moreover, it's possible that such an attack occurring would

completely crash the Bitcoin exchange rate. If a miner carried out such an attack, people might lose confidence in the system and refrain from buying bitcoins causing the exchange rate to fall.

For these reasons, the most likely motivation for a forking attack is to destroy the currency by a dramatic loss of confidence. This has been referred to as a Goldfinger attack after the Bond villain that tried to irradiate all the gold in Fort Knox to make it valueless. A Goldfinger attacker's goal might be to destroy the currency, or possibly to profit by either having shorted Bitcoin having significant holdings in some competing currency.

Forking attack via bribery. Buying enough hardware to control the majority of the hash power seems like quite an expensive and difficult task. But's possible that there is an easier way to launch a forking attack. Whereas it would be really expensive to buy enough mining capacity to have more than everybody else in the world, it might be possible to bribe the people who do control all that capacity to work on your behalf

There are a few ways that you could bribe miners. One way is to do this "out of band" — perhaps locate some large miners and hand them an envelope of cash for working on your chain. A more clever technique is to declare yourself to be a new mining pool and run it at a loss. You could offer greater incentives than other pools and cause many miners to join your pool. Even though the incentives you offer will not be sustainable, you may be able to keep them going for long enough to successfully launch a forking attack and perhaps profit. A third technique for bribing is to leave big tips in your forking blocks — big enough to cause miners to leave the longest chain and work on your chain in hopes that it will become the longest chain and they will collect the tip.

However you actually go about doing the bribing, the idea is the same: instead of actually acquiring all the mining capacity yourself, you just pay the people who already have it to work on your fork.

Perhaps miners won't help out your attack because to do so would hurt the currency in which they have invested so much money and mining equipment. On the other hand, while miners as a group might want to keep the currency solvent, they don't act collectively. Individual miners might defect and accept a bribe if they thought they could make more money in the short term. This would be a classic tragedy of the commons from an economic perspective.

None of this has actually happened. It's an open question if a bribery attack like this could actually be viable.

Block-withholding attacks. Say that you just found a block. The default behavior is to immediately announce it to the network, but if you're carrying out a block-withholding attack, you do not announce it right away. Instead you try to get ahead by doing some more mining on top of this block in hopes of finding two blocks in a row before the rest of network finds even one, and you keep these blocks to yourself as a secret.

If you're ahead of the public block chain by two blocks, all of the mining effort of the rest of the network will be wasted. Other miners will mine on top of what they think is the longest chain, but as soon as they find a valid block, you can announce the two blocks that you were withholding. That would instantly be the new longest valid chain and the block that the rest of the network worked so hard to find would immediately be orphaned, and cut off from the longest chain. This is known as ***selfish mining.*** By wasting some of the hash power of the rest of the network, you hope to increase your effective share of mining rewards.

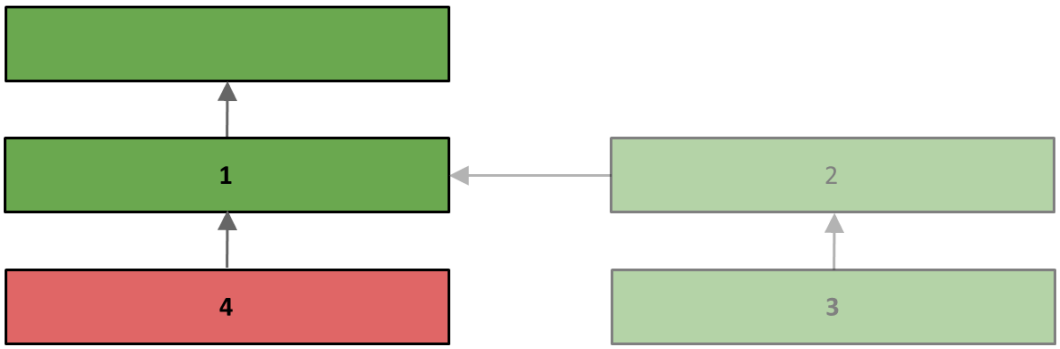


Figure 5.16: Illustration of selfish mining. This shows one of several possible ways in which the attack could play out. (1) Block chain before attack. (2) Attacker mines a block, withholds it, starts mining on top of it. (3) Attacker gets lucky, finds a second block before the rest of the network, continues to withhold blocks. (4) Non-attacker finds a block and broadcasts it. In response, the attacker broadcasts both his blocks, orphaning the red block and laying waste the mining power that went into finding it.

The problem is that you need to get lucky to find two blocks in a row. Chances are that someone else in the network announces a valid block when you're only one block ahead. If this happens, you'll want to immediately announce your secret block yourself. This creates a 1-block fork and every miner will need to make a decision about which of those blocks to mine on. Your hope is that a large fraction of other miners will hear about your block first and decide to work on it. The viability of this block withholding approach is going to depend very heavily on your ability to win these races. So your network position is of key importance here. You could try to peer with every node so that your block will reach most nodes first.

As it turns out, if you assume that you only have a 50 percent chance of winning these races, selfish mining is an improvement over the default strategy if $\alpha > .25$. The existence of this attack is quite surprising, and it's contrary to the original widely-held belief that without a majority of the network — that is with $\alpha \leq .5$, there was no better mining strategy then the default. So it's not safe to assume that a miner who doesn't control 50 percent of the network doesn't have anything to gain by switching to an alternate strategy.

At this point selfish mining is just a theoretical attack and hasn't been observed in practice. Selfish mining would pretty easy to detect because it would increase the rate of near-simultaneous block announcements.

Blacklisting and punitive forking. Say a miner wants to blacklist transactions from address *X*. In other words, they want to freeze the the money held by that address making it unspendable. Perhaps you intend to profit off of this by some sort of ransom or extortion demanding that the person you're blacklisting pay you in order to be taken off of your blacklist. Blacklisting also might be something that you want to do for legal reasons. Maybe certain addresses are designated as bad by the government, and law enforcement may demand that all miners operating in their jurisdiction blacklist those addresses.

The traditional wisdom is that there's no effective way to blacklist addresses in Bitcoin. Even if some miners refuses to include some transactions in blocks, other miners will. If you're a miner trying to blacklist, you could try something stronger, namely, punitive forking. You could announce that you'll refuse to work on a chain containing a transaction originating from this address. This is quite an extreme strategy if you have less than the majority of the network hash power. By announcing that you'll refuse to mine on any chain that has certain transactions, if such a chain does come into existence and is accepted by the rest of the network as the longest chain, you will have cut yourself off from the consensus chain forever, and all of the mining that you're doing is essentially wasted.

Feather-forking. In other words, a threat to blacklist certain transactions via punitive forking in the above manner is not credible as far as the other miners are concerned. But there's a much more clever way to do it. Instead of announcing that you're going to fork forever as soon as you seen an a transaction originating from address *X*, you announce that you're going to fork if you see a block that has a transaction from address *X*, but you will give up after a while — typically after one or two blocks confirm the transaction from address *X,* you'll go back to the longest chain.

If you give up after one confirmation, your chance of orphaning the block with the transaction from *X* is α^2. The reason for this is that you'll have to find two consecutive blocks to get rid of the block with the transaction from address *X* before the rest of the network finds a block, and α^2 is the chance that you will get lucky twice.

A chance of α^2 might not seem very good. If you control 20% of the hash power, there's only a 4% chance of actually getting rid of that transaction that you don't want to see in the block chain. But it's better than it might seems as you might motivate other miners to join you. As long as you've been very public about this, other miners know that if they include a transaction from address *X*, they have an α^2 chance that the block that they find will end up being orphaned because of your feather-forking attack. If they don't have any strong motivation to include that transaction from address X and it doesn't have a high transaction fee, the α^2 chance of losing their mining reward might be a much bigger incentive than collecting the transaction fee.

It emerges then that other miners may rationally decide to join you in enforcing the blacklist, and you can therefore enforce a blacklist even if $\alpha < .5$. If you have less than a majority of the mining capacity, the success of this attack is going to depend heavily on how convincing you are to the other miners that you're definitely going to fork.

Transitioning to mining rewards dominated by transaction fees. As of 2015 transaction fees don't matter that much since block rewards provide the vast majority — well over 99% — of all the revenue that miners are making. But every four years the block reward is cut in half, and eventually, the block reward will be low enough that transactions fees are going to be the main source of revenue for miners. It's an open question as to how exactly miners will operate when transaction fees become their main source of income. Are miners going to be more aggressive in enforcing minimum transaction fees, and how are they going to cooperate to enforce that?

In summary, miners are free to implement any strategy that they want although in practice we've seen very little behavior of anything other than the default strategy. There's no complete model for miner behavior that says that entire default strategy is optimal, and in this chapter we've seen specific examples of deviations that may be profitable for miners with sufficient hash power. Mining strategies is an area in which the practice is ahead of the theory. Empirically, we've seen that in a world where most miners do choose the default strategy, Bitcoin seems to work well. But we're not sure if it works in theory yet.

We also can't be sure that it will always continue to work well in practice. The facts on the ground are going to change for Bitcoin. Miners are becoming more centralized and more professional, and the network capacity is increasing. Besides, in the long run Bitcoin must contend with the transition from fixed mining rewards to transaction fees. We don't really know how this will play out, and using game-theoretic models to try to predict it is a very interesting current area of research.

Further reading

An excellent paper on the evolution of mining hardware:

Taylor, Michael Bedford. Bitcoin and the age of bespoke Silicon. Proceedings of the 2013 International Conference on Compilers, Architectures and Synthesis for Embedded Systems. IEEE Press, 2013.

The "systematization of knowledge" paper on Bitcoin and cryptocurrencies, especially Section III on Stability:

Bonneau, Joseph, Andrew Miller, Jeremy Clark, Arvind Narayanan, Joshua A. Kroll, and Edward W. Felten. Research Perspectives and Challenges for Bitcoin and Cryptocurrencies. Proceedings of 2015 IEEE Security and Privacy Conference, 2015.

A comprehensive 2011 paper analyzing different reward systems for pooled mining (some of the information is a bit out of date, but overall it's still a good resource):

Rosenfeld, Meni. Analysis of bitcoin pooled mining reward systems. arXiv preprint arXiv:1112.4980 (2011).

Several papers that analyze mining strategy:

Eyal, Ittay, and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014.

Kroll, Joshua A., Ian C. Davey, and Edward W. Felten. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. Proceedings of WEIS. Vol. 2013.

Eyal, Ittay. The Miner's Dilemma. Proceedings of 2015 IEEE Security and Privacy Conference, 2015.