CSC2125: Homework #6

Due on October 29

Fan Long

Thomas Hollis

Problem 1

Ethash and Equihash achieve ASIC resistance via memory-hard. Use your own word to explain their rationale. Do you think memory-hard problems are sufficient for achieving ASIC resistance?

Solution

The rationale of Ethash and Equihash is to prevent the dominance of ASIC miners in PoW-based cryptocurrencies by pushing miners toward GPU mining. The reason why ASIC miners are not viewed favourably within the cryptocurrency community is because they are inherently centralised. This concept goes directly against the decentralised vision of Satoshi Nakamoto.

This is because the vast majority of ASICs are created by a single Chinese firm: Bitmain. Frost & Sullivan, a highly reputable market research company, estimates that Bitmain possesses (as of 2017) around 67% of the Bitcoin mining hardware market share. In addition, Bitmain also operates Antpool which is one of the largest Bitcoin mining pools in history (currently only second to `BTC.com`).

Ethash and Equihash were therefore designed to prevent the dominance of ASIC miners by making their PoW algorithms memory-hard. The reason why memory-hardness is favoured as a method to achieve ASIC resistance is quite simple.

To understand this however we must first consider that CPU power was initially chosen by Satoshi Nakamoto as the baseline for approximating the "one-person-one-vote" method of democracy to "one-CPU-one-vote". However, ASICs can be produced to specialise the processing power for optimising hash function reversal. Therefore, general purpose CPUs can no longer compete with these single-purpose ASICs. In addition, these advances in ASIC technology cannot be transferred to general purpose computers so the "one-CPU-one-vote" doctrine is dying out.

Therefore, memory-hardness was introduced to replace the PoW bottleneck of processing power by a memory latency bottleneck. This is because memory access speeds are already highly optimised compared to general purpose CPUs vs ASICs. Indeed, theoretically this is supposed to prevent ASICs from severely outperforming general-purpose computers. The rationale behind why this works is that any improvement in speed of memory lanes can directly be implemented in general computers, unlike the specialised ASIC processors that have no general-purpose use.

However, there are arguments against this method of ASIC-resistance through memory-hard problems. Indeed, certain people claim that this is not enough as ASICs can still outperform general purpose machines. Indeed, ASIC Ethash and ASIC Equihash miners are currently available that achieve performances close to ten times better than general purpose machines. This would suggest that memory-hard problems are not sufficient. Indeed this is supported by further evidence, notably the work currently being done on blockchains which frequently change or cycle through the PoW algorithm to improve ASIC resistance.

Problem 2

How does Ethash achieve memory-hard? Suppose the current parameter of Ethash requires 3GB of memory but one want to solve Ethash with only 1.5GB memory. Can he/she do it? How much slower his/her algorithm will be? Why?

Solution

Ethash achieves memory-hardness by forcing miners to generate a directed acyclic graph (DAG) every epoch (or 30'000 blocks) using the Dagger-Hashimoto algorithm (Vitalik Buterins Dagger algorithm combined with Thaddeus Dryjas Hashimoto algorithm). More specifically the Ethash DAG algorithm structure works as follows:

DAG layer 1. A seed is computed by scanning through all block headers until the current block.

DAG layer 2. From this seed a 16MB pseudorandom cache must be computed.

DAG layer 3. From the pseudorandom cache, a 1024MB dataset is generated such that each item in this dataset corresponds to a small number of items from the cache.

DAG layer 4. To successfully mine the block the miner must use random slices of the dataset and hash them together to create a Mixhash. This is done using a sampling seed (determined by the previous header hash and nonce) to form the final result.

This above process is what causes the memory hardness of Ethash. It is interesting to note that Ethash also has a neat trick to allow light clients only to store the 16MB cache to allow light nodes and full notes to both use this PoW algorithm.

Now suppose a current Ethash parameter requires 3GB of memory and a particular user wanted to solve Ethash with 1.5GB of memory. While theoretically possible, this would take our memory-limited node much longer than other regular full nodes.

My best estimate is that it would take approximately 5 times as long. This is because of the difference in memory types and can be explained as follows:

Indeed, computer memory is divided into layers from the smallest and fastest memory storage (CPU registers) to the largest but slowest memory storage (SSD/hard drive). This is because registers are much more expensive per byte than SSD storage thus we want a hierarchy of memory with each layer caching the most commonly used data of the next layer. In the case of Ethash though, since we use random slices of the dataset, our CPU caching algorithms will not be useful and the memory access speed will be determined by the size of the RAM (since the 1024 MB dataset will not fit in the CPU cache). If the size of RAM is 1.5GB instead of 3GB thus 1.5GB of the required dataset will have to be loaded and written from the SSD rather than from the RAM. Typical DRAM speeds range from 2-20GB/s while SSD speeds range from 50-200MB/s. Let us therefore assume than for half of the dataset will be 10 times slower to load than someone with the full dataset in RAM. This corresponds to an overall performance at least 5 times worse than as stated above (for one particular iteration). However, this decline will scale with number of iterations thus could reach much higher penalties (100, 1000...) depending on number of iterations set by the Ethash difficulty (increases every year).

Problem 3

How does Equihash achieve memory-hard? What if someone tries to solve Equihash with half of the required memory? How much slower his/her algorithm will be? Why?

Solution

Equihash achieves memory-hardness by iterating through Wagners algorithm combined with a difficulty filter. More specifically the PoW works as follows:

Step 1. Select a hash function H, seed I and Equihash parameters n, k and d (set the time and memory hardness).

Step 2. Generate 2^{20} hashes (for Equihash $N = 200$, $K = 9$ as in Zcash) and split them in the middle to create 2^{21} strings of equal size (25 bytes). Then solve Wagners algorithm (enumerate a list, sort the list and find unordered pairs of collisions). This will require checking 2^k different strings such that the XOR of two strings is zero. In order to do this at least 50 MB of memory is needed to store these intermediate strings.

Step 3. Repeat steps 1 and 2 until you meet the difficulty target T. This means a total minimum memory of 522MB is needed (again for $N = 200$, $K = 9$).

This above process is what causes the memory hardness of Equihash.

Now suppose a current Equihash parameter requires X GB of memory and a particular user wanted to solve Ethash with $X/2$ memory. While theoretically possible, this would take our memory-limited node much longer than other regular full nodes. Again, this is due to the hierarchy of memory as described above. However, here in Equihash unlike in Ethash there is no dataset being generated but rather smaller chunks being used repeatedly. Therefore the penalty incurred by having limited memory depends how high X is. The authors of Equihash estimate on page two of the original whitepaper that if someone is trying to use 250MB for a 700MB-proof (roughly $X/2$ for a proof of X) they would pay a 1000-fold penalty in computations. I believe it would therefore take approximately 1000 times as long but this would depend on the magnitude of X.

Problem 4

Why equihash claims to be ASIC-resistant but people have developed an ASIC miner that mines ZCash equihash 10X faster? (Hint: ZCash uses equihash with the parameter N=200 and K=9. Estimate how much memory it would require to solve ZCash equihash? ASIC uses SRAM which is roughly $5 per MB.)

Solution

As seen in the question above, using equihash with parameters $N = 200$ and $K = 9$ would require a minimum memory of 522MB (Equihash whitepaper page 7) since we need to store 2^{21} strings of 25 bytes each over T iterations. This would cost an ASIC manufacturer around $2600 just in terms of raw SRAM memory cost without taking into account all other costs (including design cost).

However, the current price of Bitmains ZCash miner (Antminer Z9) is around $850. The reason behind this is that 522MB is too big to be stored solely on SRAM / CPU caches. Indeed, since they cannot rely solely on SRAM Bitmain needed to instead add a slower but larger level of DRAM memory (as in general purpose computers). Indeed, by examining board view files of the Antminer Z9, I can clearly identify a 2GB DRAM module. At this price, the Antminer Z9 is essentially a glorified general-purpose computer (with a processor and memory) stripped down to only include the bear necessary items to mine ZCash as fast as possible. No unnecessary display lanes, IO or fancy RGB lighting (OK maybe a bit of RGB lighting because everything needs RGB nowadays for some reason). Indeed by focusing the cost on using the best possible memory components (maybe even a larger on-chip CPU cache) this allows ZCash ASIC miners (like the Antiminer Z9) to achieve an approximately 10 times improvement on regular general-purpose machines for the same price.

Problem 5

Do you think it is possible to stop specialized hardware design with proof-of-work puzzles like Ethash and Equihash? Why?

Solution

Let us start off this philosophical open-ended question with a cautionary statement: Equihash and Ethash are broken. Equihash and Ethash are broken in the sense that they have failed to achieve their expected goal of high ASIC-resistance. In fact, the Equihash inventors themselves confess on their white paper (p12) than the cost advantage of ASICs would be at most a factor of 150 which is not good enough in my opinion.

In 2018, Bitmain successfully released ASIC-optimised miners for Zcash, Bitcoin Gold and other cryptocurrencies that use Equihash for PoW (the product was named the AntminerZ9). This makes the usage of GPUs for mining Equihash-based PoW cryptocurrencies increasingly expensive.

Similarly, also in 2018, Bitmain successfully released ASIC-optimised miners for Ethereum and other cryptocurrencies that use Ethash for PoW. This time the product was named the Antminer E3. Once again this makes the usage of GPUs for miner Ethash-based cryptocurrencies increasingly expensive.

However, both Equihash and Ethash have made the advantage of ASIC miners less important. These Bitmain machines have not magically broken the memory hard requirements of the PoW functions but are instead ASICs with particularly good memory devices and everything else non-essential removed. Indeed, the CPU advantage of ASICs over general purpose machines was something like a 100-time speed-up to the dollar while it is now only around a 10-time speedup to the dollar due to the memory-hard PoW algorithms. In this sense Ethash and Equihash have both somewhat helped but are not sufficient in the war against ASIC mining. The threshold of how much memory-hardness should be used is a delicate one because if the algorithms are too memory costly, they will be very much out of reach of regular general-purpose computers (and therefore continue to push toward centralisation).

It is worth noting at this stage that, theoretically, buying a general-purpose computer for the purpose of mining will always be less efficient from a dollar-hashrate perspective than buying a minimally designed ASIC. However, designing ASICs comes at a great cost to Bitmain and other ASIC manufacturers since the design time of engineers is ultimately reflected in the final price of the ASIC. I am indeed biased to believe that it is possible to stop specialized hardware as I would like to live in such a libertarian utopia. I look forward to seeing what kind of new inventive algorithms will be designed to solve this issue but I personally believe that PoW is fundamentally flawed. This is supported by the fact that Ethereum developers are currently heading toward implementing the Proof-of-Stake (PoS) alternative in the next version of ETH (codename: Casper). Unfortunately, PoS is also fundamentally flawed in my mind as it will encourage centralisation (only the richest can mint their coins effectively). I believe, cryptocurrencies like Gridcoin with their Proof-of-Research consensus offer a much more interesting alternative.

However, I think that only time will tell if the cost efficiencies of redesigning ASICs for each new PoW algorithm are profitable enough for Bitmain and other manufacturers to continue fighting in the ASIC-resistance war of PoW.