Towards Optical Proof of Work

Michael Dubrovsky\textsuperscript{1}, Bogdan Penkovsky\textsuperscript{1}, Lucianna Kiffer\textsuperscript{2}, and Marshall Ball\textsuperscript{3}

\textsuperscript{1}PoWx, Cambridge, MA, USA. Email: mike@PoWx.org, bogdan@PoWx.org
\textsuperscript{2}Northeastern University Boston, MA and DAGlabs Tel Aviv, Israel. Email: lkiffer@ccs.neu.edu
\textsuperscript{3}Columbia University, New York, NY, USA. Email: marshall@cs.columbia.edu

Abstract
The current cryptocurrency Proof of Work (PoW) ecosystem relies heavily on energy cost to ensure some semblance of decentralization and, in turn, safety. However, the growing energy consumption of cryptocurrency mining has some undesirable effects. This paper proposes a novel approach to alleviate this reliance on energy cost by designing a PoW algorithm, oPoW, that is tailored to ultra-energy-efficient photonic co-processors. While the security of the system still relies on the mining process being costly, the cost in oPoW is concentrated in capital expenses (CAPEX). The oPoW scheme involves minimal modifications to Hashcash-like PoW schemes and thus inherits many properties from such schemes, including basic safety/security from the SHA hash function. The new, additional properties follow from an assumption that photonic co-processors can compute certain functions with greater energy efficiency and speed than traditional digital ASICs. We provide a general theoretical framework for such schemes in addition to suggesting a particular instantiation of this framework to realize oPoW. We provide empirical evidence to support this assertion, as well as a description of a prototype hardware implementation and its working principles.

This work is a first step toward providing a competitive PoW-based paradigm to Bitcoin’s energy-intensive ecosystem. In addition to decreased energy consumption, other advantages of oPoW over existing approaches include greater geographical decentralization and mining democratization, censorship-resistant mining, and robust growth of hashrate as well as the creation of a new application of modern analog computing.

1 Introduction
The primary function of public cryptocurrency networks, such as Bitcoin, is to maintain a decentralized electronic ledger of transactions. Crucially, this requires that there be no single authority, such as a bank, controlling or validating the contents of the ledger. A naive design of this kind of network may be achieved if users post their transactions publicly via signed messages (using public-key cryptography), and a transaction is considered complete only when the majority of nodes on the network have accepted it. However, if the network is to be trustless/permissionless and resilient to malicious actors, a mechanism must exist to prevent Sybil attacks\textsuperscript{1} and double-spending\textsuperscript{2}.

Although there were past attempts at e-cash systems, Bitcoin’s architecture (outlined in the original Nakamoto whitepaper \cite{nakamoto}) was the first to solve the double-spend and Sybil attack problems through clever use of Hashcash \cite{hashcash}. Nakamoto’s key insight was that Proofs of Work enable distributed systems to automatically impose trivially verifiable costs on participating nodes, allowing for byzantine agreement \cite{byzantine} in settings previously believed to be

\textsuperscript{1}Multiple nodes controlled by one malicious actor
\textsuperscript{2}Making two purchases with the same coin by rewriting the ledger to remove the first transaction
Proof of Work is a solution to a specific computing challenge with 40 leading zeros, then statistically 2 of the “acceptable” nonces. This difficulty setting requires the miner to find a nonce that leads to a block hash of work, which quickly becomes infeasible without control of more than half the computing power in the network (this is called a 51% attack, see Section 2.4). Moreover, any double-spending transaction becomes impossible as only the longest of the two newly created blockchains will be recognized as valid. PoW has also been applied to more complex high-throughput (in transactions per second) decentralized ledgers where blocks are in a directed acyclic graph, not simply a chain [6,7]. PoW schemes have an excellent track record of ensuring the irreversibility of transactions in the Bitcoin network. However, Proof of Work has run into severe scaling issues that may eventually undermine Bitcoin’s growth.

1.1 Proof of Work in the Context of Blockchains

Proof of Work schemes, or pricing functions, were initially proposed at Crypto 1992 by Cynthia Dwork and Moni Naor for a variety of tasks such as combating junk mail [4]. A “Proof of Work” is a solution to a specific computing challenge that unavoidably requires a certain amount of computational work to solve. This challenge is called a cryptopuzzle and is designed in such a way that it can only be brute forced by checking all possible solutions one by one until a valid solution is found. This assures that solutions are relatively rare. Solving these cryptopuzzles in the context of cryptocurrency is colloquially known as mining, because a successful solution yields rewards (known as “block rewards”). On the other hand, it is easy to verify any solution is correct, requiring only a single cycle of computation. Therefore, a Proof of Work provides a trivially verifiable guarantee that a certain amount of computation was performed to produce it.

In Bitcoin, transactions are recorded into blocks, and a linked sequence of such blocks is called a blockchain. Once a mining device (miner) compiles a block of transactions, it shuffles through random values of a special input (nonce) in the block until a cryptographic hash of the block is smaller than a predetermined threshold. The security properties of hash functions force a miner to test nonces by brute force until a satisfactory block is found. Such a block constitutes a solution to the cryptopuzzle and is itself the proof of work. Once the block is published, anyone can easily verify that the work was done by calculating the cryptographic hash of the block and checking that it is really below the predetermined threshold\(^2\). Bitcoin uses the cryptographic hash SHA-256(NIST), but various hash functions are used by different blockchain networks. Each type puts different load on the processor and memory of a miner’s computing device, but they all use the same principles. Ethereum for example, uses a cryptographic hash named Ethash, which has greater memory requirements [5].

As a result, a blockchain’s validity is based on previously performed computational work. That also means that the longest chain, implicitly corresponding to the highest amount of work, can be automatically considered to be the valid transaction history (as it accrued the majority of computing resources). Modifying any single block requires a vast amount of computation\(^3\), which quickly becomes infeasible without control of more than half the computing power in the network. This threshold is automatically adjusted by the system such that only 1 in 2,048 blocks are accepted. This threshold is designed to grow with network sources. Modifying any single block requires a vast amount of computation without the control of more than half the computing power in the network. Hence, a successful double-spend attack requires a vast amount of computational power and time, which quickly becomes infeasible without control of more than half the computing power in the network.

1.2 Challenges Faced by Bitcoin’s Proof of Work Ecosystem

As Bitcoin has grown over the past decade from a small network run by hobbyists to a global currency, the underlying Proof of Work protocol has not been updated. Initially envisioned as a global decentralized network (“one CPU-one vote”), Bitcoin transactions today are secured by a small group of corporate entities. Due to the increase in the market value of mining rewards over time and competition between miners, Bitcoin mining difficulty has grown exponentially, leading to the industrialization of mining. The enormous and growing energy use of Proof of Work has led to geographic centralization of mining in purpose-built data centers located in regions with very low energy costs and barred small entities from the mining ecosystem.

Although the exact numbers are disputed, Bitcoin’s energy use has grown steadily with its market value, and, today, Bitcoin is estimated to consume over 75 terawatt-hours per year [8]. Given that this is more than the electricity consumption of Austria [9], Bitcoin mining heavily favors economies of scale. In fact, it is only feasible for entities that can secure access to abundant, inexpensive energy [10]. The economics of mining limit profitability to places like Iceland and Western China. Besides the negative environmental externalities, which may be significant, mining today is performed primarily with the consent (and in many cases, partnership) of large public utilities and the governments that control them. Although this may not be a problem in the short term, in the long term it stands to erode the censorship resistance and security of Bitcoin and other public blockchains through potential regulation or partitioning attacks\(^6\) [13].

---

\(^2\)This threshold is automatically adjusted by the system such that only 1 block is found every \(\sim 10\) minutes; the lower the threshold the more unlikely it is to find a solution. For example: A SHA256 hash produces 256 bits, if the difficulty setting requires the miner to find a nonce that leads to a block hash with 40 leading zeros, then statistically 2\(^{256}\) trials will be required to find one of the “acceptable” nonces.

\(^3\)The proofs of work for all blocks following the altered block must be recomputed.

\(^4\)Hashrate, and therefore energy use, is designed to grow with network value to maintain an unfeasible high cost of double-spending relative to the overall value of a successful double-spend attack.

\(^6\)Censorship at this scale is not as unlikely as one might think [11, 12],
An additional consequence of the energy-based economics of Proof of Work is the sensitivity of hashrate to block reward value. If the dollar value of block rewards falls or electricity prices rise, marginally profitable miners are forced to shut off their machines to avoid running at a loss. This leads to undesirable instability in the security of the network - especially during periods of volatility.

It is important to note that, from an algorithmic standpoint, *Bitcoin’s energy consumption is a feature, not a bug*. The network is designed to automatically incentivize an increase in PoW mining as it gets bigger in order to maintain a proportionally higher level of security. The dollar value of the mining rewards rises (by design) with the market value of the coin, leading miners to spend more resources competing for the mining rewards (which are denominated in Bitcoin), and therefore use more energy. This allows Bitcoin to scale the cost of a 51% attack as the reward associated with a successful attack increases. The Bitcoin algorithm has no direct access to information about Bitcoin’s market value, but it can indirectly infer a value increase from a hashrate increase. An alternative mining reward algorithm can be imagined that actually reduces the block reward as hashrate increases, thus limiting the incentive for miners to expend more resources and energy. This would indeed decouple the energy consumption of the network from the total value stored and the market value of each coin, however, it would mean that the cost of attacking the network would no longer grow with the incentive to do so.

New consensus mechanisms have been proposed as a means of securing cryptocurrencies whilst reducing energy cost, such as various forms of Proof of Stake and Proof of Space-Time [15, 16] being implemented by Chia and Spacemesh. While many of these alternative mechanisms offer compelling guarantees, they generally require new security assumptions, which have not been stress-tested by live deployments at any adequate scale. Consequently, we still have relatively little empirical understanding of their safety. Completely changing the bitcoin paradigm is likely to introduce new unforeseen problems. We believe that the major issues discussed above can be resolved by improving rather than eliminating Bitcoin’s fundamental security layer—Proof of Work.

1.3 A Next-Generation Proof of Work

Instead of devising a new consensus architecture to fix the scaling issues, we consider ways to shift the economics of PoW. As it is used in Bitcoin-like systems, PoW allows networks like Bitcoin to achieve consensus via economic difficulty imposed on the miners. However, the financial cost does not need to be concentrated in electricity. In fact, the situation can be significantly improved by reducing the operating expense (OPEX)—energy—as a significant cost of mining. Then, by shifting the cost towards capital expense (CAPEX)—mining hardware—the dynamics of the mining ecosystem become much less dependent on electricity prices, and much less electricity is consumed as a whole. Moreover, this automatically leads to geographically distributed mining, as mining becomes profitable even in regions with expensive electricity. Finally, lower energy consumption eliminates heating issues experienced by today’s mining operations, which further decreases operating cost as well as noise associated with fans and cooling systems. All of this means that individuals and smaller entities would be able to enter the mining ecosystem simply for the cost of a miner, without first gaining access to cheap energy or a dedicated, temperature-controlled data center. To a degree, memory-hard PoW schemes like Cuckoo Cycle [17], which increase the use of SRAM in lieu of pure computation, push the CAPEX/OPEX ratio in the right direction by occupying ASIC chip area with memory.

To fully maximize the CAPEX to OPEX ratio of mining cost, we investigate alternative Proof of Work algorithms and complementary computing hardware paradigms (beyond standard ASICs) that are difficult/expensive to produce but achieve high energy efficiency. One can observe that Artificial Intelligence (AI) hardware industry is converging to a similar goal as many companies try to commercialize exotic architectures for low-energy computing. One of the promising approaches being commercialized for AI is optical computing, specifically photonic co-processors. Due to its commercialization feasibility and long term potential for ultra-low energy use, we concluded that optical computing is a promising platform for a low energy Proof of Work.

1.4 Optical Computing

While in traditional digital hardware, we rely on electrical currents, optical computing uses light as the basis of its operations. The approach has been around for decades, however recent advances in the telecommunications industry and Artificial Intelligence (AI) have significantly contributed to optical computing development. Indeed, researchers anticipate that integrating optical processing starting with on-chip signal routing and ending with optical accelerators for AI can significantly boost processing speed, keeping energy consumption

---

7 Although this is not perfect as hashrate tends to increase due to hardware improvements.
8 Bitcoin’s block subsidy halving effectively does this to an extent, however the common argument is that transaction fees will pay for security once the block subsidy is depleted.
9 There are numerous attempted implementations, notably Ethereum’s so-far unsuccessful effort to implement some form of PoS on their mainnet. These schemes also introduce new challenges [14].
10 Demand for AI compute is growing exponentially and cannot be supported by conventional hardware without massive energy consumption [18].
levels as low as possible. Moreover, the semiconductor industry has nearly reached its fundamental limits meaning that digital computers cannot continue improving on the same trajectory they have followed for fifty years [19]. This also means that continuous technological progress will require alternative computing methods and alternative hardware. The advantages of using light for information processing can be best illustrated by the adoption of optics in the telecommunications industry. Indeed, the replacement of copper cables with optical fibers transformed intercontinental communications, including the Internet, which has become exponentially faster, and more efficient.

Optical computing has a rich history dating back to Fourier processing in the 1940s (Duffieux 1946), the first optical neural networks in the 1980s (Psalti 1984), 1980s work on the optical transistor at Bell Labs, and modern work in optical neural networks, reservoir computing and optical quantum computing. Optical computing research at Bell Labs as well as holographic computers and optical chaos communications provided the inspiration for forthcoming non-von Neumann computing hardware, including photonic deep learning and reservoir computing.

Deep Learning In recent years, photonic implementations of artificial neural networks have gained traction. As a result, several new optical hardware strategies have been developed to perform matrix-vector multiplication [20, 21], the basic operation behind those architectures. Unlike older approaches that relied on slow spatial light modulators [22], recent implementations, which are now being commercialized, use integrated silicon photonic circuits based on microring resonators [20], and Mach-Zehnder Interferometer arrays [21].

Reservoir Computing (RC) was conceived as an attempt to simplify recurrent neural networks training [23–25]. Trained in a supervised fashion, RC systems are capable of solving pattern recognition and time-series prediction challenges [26, 27] and quickly became popular for in-materio computing with various physical substrates such as water (hence “reservoir”) [28]. Researchers have shown that using optical systems for computation, photonic RC research can yield high-speed operation and low energy use. For comparison, an RC setup using off-the-shelf optical equipment is estimated to perform speech recognition tasks about three times faster than a Google TPU [29].

Despite successful demonstrations of optical computing in various forms over the years, the limiting factor for commercial adoption has always been the cost and difficulty of manufacturing the hardware on-scale. Competing digital computers have benefited from continuous improvements in fabrication technology and a fabless supply chain for new products [11]. The relatively recent advent of silicon photonics, which is compatible with standard chip manufacturing processes used for digital electronics, has opened the possibility of manufacturing scalable and reproducible optical co-processors. In the next two subsections, we provide a brief overview of silicon photonics and its application to deep learning AI processing.

Additionally, in Section 2.2, we briefly explain the working principles of matrix-vector multiplication in photonic circuits.

1.5 Silicon Photonics

Traditionally, optical systems require precise alignment and expensive, carefully manufactured bulk components. Recent advances in Photonic Integrated Circuits (PICs) have addressed these problems successfully by porting bulk optical systems to chip-scale waveguide circuits. Photonic Integrated Circuits are produced by patterning thin dielectric or semiconductor wafers using micro/nanofabrication, leveraging the incredible and ever-increasing precision of lithography to ensure alignment and enable cheap mass production. Until the early 2000s, PICs were fabricated using expensive III-V materials. Despite silicon’s inherently sub-optimal optical material properties (no commercial silicon lasers, no electro-optic effect), its ubiquitous use in electronics has created a huge fabrication ecosystem that made it advantageous to work around the problems and build on-chip silicon optical components. After breakthroughs in silicon (Si) photonic component design such as low loss optical fiber-to-chip couplers [30], fast electro-optic modulators [31], and Germanium-on-chip photodetectors [32], it became possible to take advantage of the incredible developments in silicon CMOS technology over the last six decades to produce photonic circuits in repurposed electronics foundry processes. One of the fundamental building blocks of Si photonic circuits is the nano-scale Si waveguide. Figures 2 and 3 depict typical waveguides produced in a Silicon-on-Insulator wafer, which confine light within the photonic circuit via total internal reflection (the same effect used to guide light in fiber optics).

12Complementary Metal–Oxide–Semiconductor (CMOS) is the nanoelectronics fabrication process which has successfully decreased the cost and size of transistors by something like a factor of 10⁶ in last 60 years. The technology is the basis for modern digital circuits, including computer processors and memory.
Silicon photonic integrated circuits have had commercial success as transceivers for various datacom applications [34]. Today, millions of silicon photonic transceivers (manufactured by companies such as Luxtera, IBM, and Intel) shuttle information between server racks at data-centers. Additionally, there are now multiple companies commercializing silicon photonics for LIDAR and bio-sensing. Crucially, major microchip foundries (including Global Foundries and TSMC) are either offering Si photonics or are in the process of launching silicon photonic manufacturing lines. In recent years, this availability of commercial silicon photonic fabrication has spurred efforts to commercialize silicon photonic chips for massively parallel computation leading to the emergence of photonic co-processors for AI.

**Integrated Photonic Co-Processors for AI**

Due to the recent success of deep learning AI algorithms, the demand for massive quantities of Multiply and Accumulate (MAC) processing has led to heavy investment in MAC processor research as well as many commercial efforts to produce specialized processors that perform these computations more efficiently from a cost and energy standpoint. In parallel to ongoing developments by GPU manufacturers like NVIDIA and Google (TPU), multiple companies such as Groq (digital), Graphcore (digital), as well as Mythic (analog) and Synthiant (analog) are pursuing innovative electronic architectures for MAC.

AI architectures have been explored in free-space optical systems [35, 36], and several companies such as LightOn, Fathom, and Optalysis are working on implementing such systems commercially. However, more recently, due to the progress made by the Princeton Neuromorphic Photonics lab [37] as well as research at MIT [21] and other academic institutions, several startups have emerged, including Lightelligence [38], Lightmatter [39], and Luminous [40], that are applying silicon photonics designs for telecommunications and quantum information processing to build MAC processing photonic circuits. The promise of the technology, as detailed by Nahmias et al. [41], is to offer 2-3 orders of magnitude better energy efficiency for MACs over electronic processors, and eventually even greater gains as optical computation has very high theoretical limits for energy efficiency [42]. In a comparison of state-of-the-art GPU performance against a model of an electronic-photonic processor based on off-the-shelf foundry components, it was found a 2.8 to 14x speedup for the same power usage when performing CNN computations [43]. An estimate for photonic co-processors by Lima et al. predicts 10fJ/MAC for a 128 channel chip vs. 1pJ/MAC for the Google TPU [44] and Nahmias et al. [41] predict that the performance can be pushed to 2.1fJ/MAC. These exciting developments in silicon photonic co-processors have created an opportunity for applying the underlying technology to low-energy applications outside AI processing.

**1.6 Optical Proof of Work**

Inspired by the recent advances in silicon photonics for low-energy computation, we envision a practical PoW system built to complement optical computing. The main goal of such a PoW approach is to achieve drastic energy savings. Although, in the long run, it is conceivable that some miners will be built based on other analog architectures, we see photonic co-processors as holding the greatest potential for high energy-efficiency combined with near-term commercial availability. As a result, we propose Optical Proof of Work (oPoW), a PoW algorithm optimized for acceleration with integrated photonic co-processors.

**2 Low Energy PoW**

Rather than attempting to compute an existing PoW algorithm using photonic hardware\(^{13}\), we chose to construct a modified PoW to favor existing photonic co-processor designs. Here we will briefly describe the co-design of a prototype photonic co-processor and PoW algorithm built to achieve our low-energy PoW goal.

It is worth noting that nearly all previous attempts to modify PoW algorithms to favor a specific hardware paradigm have focused on ASIC-resistance, meaning that rather than favoring specialized hardware, the aim is to exclude specialized hardware in favor of GPUs or CPUs\(^{14}\). Examples include Scrypt, Cryptonight, Equihash, and, more recently, ProgPoW. Besides ProgPoW, which has not been implemented yet, these experiments have more or less failed due to the inherent advantages of specialized hardware. An excellent discussion of this topic can be found in *The State of Cryptocurrency Mining* [45], where the author concludes that:

\(^{13}\)There is enormous financial incentive to do this already—many optical computing experts have looked at the possibility of using photonics for bitcoin mining, however existing PoW algorithms are ill-suited to analog computing. Hashes like SHA256 are specifically designed to be efficiently implemented by digital processors.

\(^{14}\)The end goal being democratization of hardware supply rather than energy efficiency or geographic decentralization.
“For any algorithm, there will always be a path that custom hardware engineers can take to beat out general-purpose hardware. It’s a fundamental limitation of general-purpose hardware.”

Optical PoW is fundamentally a simpler engineering problem than ASIC-resistant PoW. It is designed to be the most efficient on integrated photonics hardware, therefore giving one class of ASICs an advantage over another, rather than trying to limit the advantage of specialized hardware over general hardware.

2.1 HeavyHash

Our goal in designing oPoW was to mimic the Bitcoin PoW construction (HashCash), maintaining the cryptographic security while ensuring that the PoW crypto puzzle is optimized for our “Target Paradigm” (photonic co-processors). As the major cost of PoW is evaluating the hash function of choice, the naive solution would be to find an optically computable hash. However, a design choice was made early on to avoid all-optical hashes and Physical One Way Functions - due to issues of repeatability [46] and their poorly understood security properties. Creating a new hash optimized for photonic processing was also not considered due to the complexity and risk of deploying an untested hash function, see IOTA fiasco [47]. This leads to the selection of a hybrid design that composes digital hashing with low precision matrix-vector multiplication (intended for photonic acceleration) to produce HeavyHash. Discussed in detail in Section 3, HeavyHash is an iterated composition of an existing hash function, e.g. SHA256, and a weighting function such that the cost of evaluation of HeavyHash is dominated by the computing of the weighting function. If the weighting function is dominated by the evaluation of a real value matrix-vector multiplication of sufficient size, it can be implemented with very high efficiency by photonic co-processors [21]. The ratio of the cost of computing the hashes versus that of the weighting function is tunable within a large range due to their different complexity orders of magnitude\(15\).

2.2 Optical PoW Prototype

In its simplest form, oPoW is the Hashcash algorithm [2] with our custom hash function, HeavyHash, designed specifically to run efficiently on optical accelerators while preserving all PoW-necessary cryptographic security properties. A prototype hardware and software implementation of Optical PoW (an oPoW Bitcoin fork and a prototype oPoW silicon photonic miner) was developed with the goal of testing end-to-end functionality. Replacing the hash function in Bitcoin’s PoW code is straightforward, so here we focus on the prototype hardware. Below is a short description of the system, the silicon photonic integrated circuit at its heart and the working principle of the analog computation.

System

There are multiple known architectures for implementing an analog matrix-vector multiplier using standard silicon photonic components. The two main types of approaches are (1) the ring filter bank architecture developed at the Princeton neuromorphic photonics lab [20] and (2) various MZI interferometer meshes such as the triangular mesh used in the highly cited Shen et al. [21] paper from MIT.

Our analog photonic matrix-vector multiplier implementation is a rectangular directional coupler mesh\(16\). As seen in Figure 4 a RasPi board running our Bitcoin fork node software is paired with a driver board made by Qontrol, which communicates with a custom printed circuit board, TIA (to amplify the signal from the photodetectors) and interposer on which the silicon photonic chip is mounted\(17\). A close up of the packaged chip can be seen in Figure 1 and 5 shows a top down view of the bare chip. The RasPi performs the digital portion of the HeavyHash and offloads the analog portion to the photonic chip via the Qontrol controller.

\(15\) An N increase in the output size of the hash, corresponding to an N increase in hash computation cost, leads to an \(N^2\) increase in computation cost for the weighting function due to the properties of matrix multiplication.

\(16\) Mesh design was generated using an algorithm provided by Sunil Pai based on his work at Stanford [48].

\(17\) The PCB, TIA, and silicon photonic chip were fabricated in partnership with SiEPIC kits, an integrated photonics engineering firm affiliated with University of British Columbia.
Silicon Photonic Chip

As seen in Figure 5, the chip consists of a single surface grating coupler input, split into 16 outputs. Each output is modulated (according to data provided by the RasPi) individually by a balanced thermal Mach-Zehnder Modulator\(^{18}\). The outputs of the modulators are then fed into the matrix multiplication directional coupler mesh\(^{19}\), and the outputs of the matrix multiplication network are collected into fibers via grating couplers and converted into electrical signals by the photodiodes and TIA. In a commercial setup, the photodetectors would be on-chip, making the entire package much simpler and eliminating the need for fiber optic connections (a laser input can be coupled via flip-chip bonding).

Working Principle of Unitary Matrix Multiplication in a Mesh of Directional Couplers

A generalized discussion of matrix multiplication setups using photonics/interference can be found in Reck et al., Russell et al. [49, 50]. A detailed discussion of several integrated photonic architectures for matrix multiplication and corresponding tuning algorithms can be found in Pai et al. [51]. In this section, we will provide a basic intuition of the working principle of the approach we used.

As seen in Figure 6, a single laser input is split evenly into multiple waveguides, each waveguide feeds into a modulator that can decrease the intensity of the light.

Figure 6: General Block diagram of the device’s components (metal in blue). N = 16 in our design and the metal wirebonding pads provide electrical access to MZM modulators. A second set of pads provide access to tuning heaters in the directional coupler mesh.

We chose a Mach-Zender Modulator, as seen in Figure 7, which splits the input light into two waveguides, and recombines them again with a phase shift. The phase shift is accomplished using a heater\(^{20}\) which changes the refractive index of one of the waveguides in the modulator.

Figure 7: Mach-Zender Modulator

In a perfect device, a full \(\pi\) shift results in complete destructive interference, and smaller shifts can be used to get

---

\(^{18}\)A brief explanation of the MZM is given in Section 2.2.3. For more details, an excellent tutorial on typical silicon photonics components can be found in *Silicon Photonics Design: From Devices to Systems* by Lukas Chrostowski and Michael Hochberg [33]

\(^{19}\)A brief explanation of directional couplers is given in 2.2.3

\(^{20}\)There are PN junction-based phase shifters with much better speed and efficiency.
partial destructive interference. This is how a digital signal used to drive the modulator can be converted into an analog optical intensity. For example, in a 4-bit system, a $\pi$ shift would correspond to 0000, and zero phase shift would correspond to 1111, with partial interference providing the levels in between. The outputs of the modulators are then fed into a mesh of directional couplers (see Figure 8 below), whose splitting ratios depend on the phases $\phi$ and $\phi'$ of the light entering at each input and the effective optical geometry of the coupling region, depending on the physical geometry (length of coupler and the gap between the waveguides) as well as refractive index which can be tuned with a heating element.

![Figure 8: Basic directional coupler design](image)

By tuning the phase delays of each waveguide at each layer of the directional coupler mesh and the coupling region’s effective optical length using heaters it should be possible to achieve an arbitrary unitary transfer matrix\(^21\). If we call the vector of amplitudes and phases exiting each modulator the input vector $[I]\(^22\)$, and the transfer matrix of the mesh $[U]$, the output is mathematically equal to the matrix-vector operation $[U]^*[I]$. However, the actual signal detected at the photodetectors corresponds to the intensity of light, not the phase. Therefore, the detected analog electrical signal actually returns the absolute value of the output vector $[O]$. Pai et al. recently published a detailed discussion of this architecture and several other similar architectures with different trade-offs [48]. It is worth noting, that although unitary (or orthogonal if detecting only amplitude) matrices are ideal for this kind of architecture, any real value matrix (with values ranging 0-1) can be implemented with some additional tweaks left out in the interest of brevity. The advantage of using photonics to perform this operation, assuming a low precision AD-DA conversion is compatible with the use case, here is tremendous: all the light beams perform inherently parallel processing.

We anticipate that adopting a PoW algorithm designed for photonic hardware will provide blockchain networks with various benefits, such as better security (higher 51% attack resistance) and lower overall network energy consumption for equivalent networks. Although detailed analysis of the economics and security implications of a CAPEX-dominated PoW will be published separately, in the following sections we provide a brief summary.

### 2.3 Energy Savings

The total energy use of a PoW blockchain does not depend on the energy cost of a single hash/trial but on the total amount of energy used by the miners performing work in the system. For example, if SHA256 was replaced by a hash with lower computational difficulty (and therefore lower energy consumption per hash), but the same optimal hardware paradigm (ASICs), then miners would simply be forced to perform more hashes via difficulty adjustment.

However, if the cost of PoW computation is biased towards a hardware paradigm which is more energy-efficient (but may be more expensive in terms of hardware cost), a blockchain built on such a PoW scheme will be more energy-efficient overall even if, somewhat paradoxically, an individual hash is more computationally expensive. The reason that this is possible is that the PoW difficulty adjusts so that the relative cost per block of the schemes is equivalent, even if the cost per single evaluation of the underlying hash function is significantly different. Because the overall cost of the schemes depend on the value of the block reward, not the number of hashes required to get the reward, we can directly compare the relative factors that make up the cost of each hashing scheme (energy cost, OPEX, and hardware depreciation, CAPEX).\(^23\)

So long as the more energy-efficient hardware paradigm provides some marginal cost advantage over others, we assume that rational agents will adopt this paradigm to maximize their utility in a given mining ecosystem.

In summary, a low energy PoW can be achieved by tailoring a PoW algorithm to a hardware paradigm with a CAPEX dominated cost per hash/trial.

#### 2.3.1 Modeling Energy Savings

We will compare the energy use of oPoW secured network with an equivalent Hashcash secured network, assuming energy costs are the same for miners in both network. Additionally, we will assume that the two systems have the same block reward dollar value and hardness-parameters are scaled such that the dollar cost of mining a block is identical in both systems. The dollar cost of a single hash is a function of both the cost of hardware (CAPEX amortized over hardware lifetime) and the cost of energy per evaluation (OPEX). The relative contributions of the amortized CAPEX and OPEX directly determine the breakdown of the ecosystem’s overall mining expenditure.

Assuming a system has block reward, $R$, this is equal to amortized CAPEX per block + OPEX per block + some aver-

\(^{21}\) Practically, our prototype can only achieve a subset of unitary matrices, due to limited number of electrical inputs but this is not a fundamental problem for commercial systems.

\(^{22}\) By design the input phases are all the same.

\(^{23}\) We are assuming here that the hashrate will be adjusted by market forces so that the relative cost of mining is identical in either framework.
age profit margin \( \epsilon \), thus:

\[
R = \text{CAPEX} + \text{OPEX} + \epsilon
\]

Taking a digital system where OPEX is kWh per hash \times dollars per kWh = \( O_{\text{digital}} \) and amortized CAPEX per hash = \( C_{\text{digital}} \), a digital system has \( K_{\text{digital}} \) hashes per block such that:

\[
R = K_{\text{digital}} \ast (O_{\text{digital}} + C_{\text{digital}}) + \epsilon
\]

Equally, if we assume that the block rewards and profit margin of miners are identical in both digital and optical systems, an oPoW system has \( K_{\text{optical}} \) hashes per block such that:

\[
R = K_{\text{optical}} \ast (O_{\text{optical}} + C_{\text{optical}}) + \epsilon
\]

Thus the total spend on energy per block of each system is \( K_{\text{digital}} \ast O_{\text{digital}} \) and \( K_{\text{optical}} \ast O_{\text{optical}} \) respectively.

We can then calculate the energy savings of the optical system over the traditional digital system as follows:

Energy savings = \( \frac{\text{digital energy cost} - \text{optical energy cost}}{\text{digital energy cost}} \)

\[
= \frac{K_{\text{digital}} \ast O_{\text{digital}} - K_{\text{optical}} \ast O_{\text{optical}}}{K_{\text{digital}} \ast O_{\text{digital}}}
\]

\[
= 1 - \frac{K_{\text{optical}} \ast O_{\text{optical}}}{K_{\text{digital}} \ast O_{\text{digital}}}
\]

We can reduce the above to a simple function of the relative contributions of energy cost and amortized hardware cost per hash for optical and digital hardware. Namely, define \( \text{MIR}^O \) and \( \text{MIR}^D \) to be the miner inefficiency ratio of the average optical and digital mining hardware in the systems (respectively),

\[
\text{MIR}^O := \frac{\text{energy cost per optical eval}}{\text{total cost per optical eval}} = \frac{O_{\text{optical}}}{C_{\text{optical}} + O_{\text{optical}}},
\]

\[
\text{MIR}^D := \frac{\text{energy cost per digital eval}}{\text{total cost per digital eval}} = \frac{O_{\text{digital}}}{C_{\text{digital}} + O_{\text{digital}}}
\]

Then, substituting in the above we get:

Energy Savings = 1 - \( \frac{\text{MIR}^O}{\text{MIR}^D} \).

### 2.4 Security Budget Implications

The Bitcoin network pays miners approximately $5B yearly to secure its ledger. In the end, this cost is borne by the holders of Bitcoin via inflation. When analyzing any proposed security/consensus algorithm for decentralized cryptocurrencies, the key question is: *How much real security does the security budget buy?*

To examine the economic security of implementing oPoW in a blockchain protocol, we consider the classic 51% attack [1], as well as hashrate behavior over time.

#### 2.4.1 Fifty-One Percent Attack Security in a Low-OPEX PoW

In a 51% attack, an adversary is interested in acquiring more than half of the *hash-power* of the system in order to break consensus, double-spend payments, censor specific payments and so on. The cost of such an attack should ideally match (and surpass) the total CAPEX (hardware controlled by honest nodes) in the system and pay for the OPEX (energy) cost for the duration of the attack. Any PoW blockchain system’s security is predicated on a high cost for such an attack.

Assuming a single system using a particular implementation of the oPoW algorithm\[24\], an attacker willing to acquire 51% of the hashrate for an attack likely can’t rent the hardware necessary for this attack. Miners in the system are unlikely to simultaneously rent out such a large portion of their hashrate\[25\], and since no other system is using the hardware there is no secondary source (a system with generic hardware, such as GPUs, doesn’t have this advantage). Thus the attacker must purchase (or produce) close to the total CAPEX of the system to gain 51% of the computing power. Note that by attacking the system, the attacker potentially makes the resale value of the hardware negligible.

Since 51% attacks also incur OPEX costs for the duration of the attack, an attacker would try to minimize the duration of an attack. For a double spend, for example, the attack would only need to last as long as the confirmation time of the victim, i.e. no more than one day. Assuming the attack lasts for \( T \) blocks and a system with hashrate \( H \) and CAPEX cost per hash \( C_x \) and OPEX cost per hash per expected time to find a block (i.e. 10 minutes in Bitcoin) is \( O_x \) we have that the cost of attack is \( H \ast C_x + T \ast H \ast O_x \). In any system where the OPEX cost for a day of mining is \( << \) the total CAPEX of the system, the cost of attack is highly dominated by the cost to acquire all the hardware for the attack.

This analysis also holds true for Bitcoin. Although Bitcoin has high OPEX, the cost of a short attack (on the order of days/weeks) is dominated by the cost of acquiring the necessary hardware. Take the Antminer S9 with retail value around $700 and hashrate 14TH/s which can mine approximately 1.5e-7 of the current Hashrate of Bitcoin, i.e. an expected $0.019 mining reward. Using the mining reward as a cap on the OPEX cost, otherwise mining would not be profitable.

\[24\] The analysis gets complicated if multiple networks are using the same PoW algorithm, because miners from one can attack the other.

\[25\] Although not impossible, a coordinated effort to rent approximately half of the hardware on a network would be difficult to hide and owners of hardware have an incentive not to rent to attackers, as the hardware is likely to lose value if the network is attacked.
this means the CAPEX of the Antminer S9 is at least 157,000 (or 36,800) times its maximum. OPEX. Thus in the cost of attack formulation, for an attack lasting a few days (or few hundred blocks) the CAPEX is the dominating factor. Therefore, as we shift the hardware cost from OPEX to CAPEX to be even more CAPEX heavy, we increase the cost of these short-term attacks.

Overall, we believe oPoW may provide increased “51% security” than an OPEX-heavy PoW in the long run, as there are indications it will lead to faster hashrate growth and greater hashrate resilience to decreases in the value of the block rewards. 27

2.4.2 Hashrate Growth and Resilience in a Low-OPEX PoW

Shifting mining cost from OPEX to CAPEX, increases the total effective investment made by the network (via block rewards and transaction fees) into long term security. Any OPEX costs the miners incur do not contribute to the hashrate growth and, therefore, do not contribute to the long term security either. As more funds flow to CAPEX, the network builds up a larger and larger cache of specialized security hardware, making the barrier for attack higher. In a related positive effect of CAPEX dominance, miners running low-OPEX hardware have less incentive to turn it off when the coin price (and therefore mining reward value), or the price of electricity fluctuates. Bitcoin’s hashrate growth is not nearly as impressive as it looks when hardware performance improvements are accounted for. Analyzing something like Specific Hashrate (we can loosely define Specific Hashrate as hashrate divided by the dollar cost of performing a single hash) shows that Bitcoin’s security is very sensitive to price. In Q4 2018, Bitcoin prices were volatile, and the coin temporarily lost around 45% of its market value. As a result of miners shutting off their machines to avoid paying for electricity, the hashrate dropped from 60 EH/s to 35 EH/s [52] (despite Bitmain releasing a new high performance 7nm miner [53] and other hardware manufacturers joining the fray). oPoW’s economies can create a faster-growing, more stable, and more committed community of miners.

3 Modifying Hashcash

In this section, we show that a simple modification to the hash function at the core of the Hashcash proof of work system, can increase the cost of producing a proof of work.

Recall that if this increase in cost scales differently across different hardware platforms it is reasonable to assume that miners will shift to the more efficient platform. If additionally, the modified proof of work is such that energy makes up less of the relative mining cost in this new ecosystem (based on the platform that is cheaper overall) vs the old, then energy will be saved.

However, in this section we simply introduce a generic framework for modifying Hashcash-like Proof of Work Schemes to achieve these ends, without compromising safety. Our transformation is incredibly simple, requiring only a simple black-box transformation of the underlying cryptographic hash function. And as a result integrating to existing platforms involves minimal overhead.

To instantiate this framework two ingredients are required: (1) a function that is hard to evaluate, (2) a cryptographic hash function. We introduce a complexity-theoretic property we call “minimal effective hardness” that suffices to realize the first ingredient. Following the paradigm of Bellare and Rogaway [54], we prove that our scheme achieves the desired soundness by modeling the underlying cryptographic hash function as a public random function. In practice, the cryptographic hash function can be concretely instantiated as SHA256, or others.

3.1 Review of Hashcash

In general, in a proof of work system we want the guarantee that any possible prover, given a random challenge c, is required to perform some amount of computational labor in order to produce a valid proof, p, corresponding to c.

Hashcash followed from a simple observation that given a random oracle, the number of calls required to find an input that hits a sparse set (a $2^{-k}$ fraction of the codomain), behaves like the geometric distribution (with parameter $2^{-k}$).

Putting this in language of PoWs, we can view a random string $c$ as specifying a random oracle. In particular, let $H_c(c, \cdot) := H(c, \cdot)$. Then, a proof in hash cash is simply a string $x$ such that

$$H_c(x) \text{ has } \lambda \text{ leading zeros.}$$

In its use in Bitcoin, the block header is effectively provides a random challenge. The random oracle is instantiated as SHA256.

Note that throughout the remainder of this paper we will simply use $H$ instead of $H_c$ or $H(c, \cdot)$.

3.2 The Heavy Hash Framework

The intuition behind our framework is very elementary. Assume that there is a function that has a differing evaluation cost relative to two different hardware paradigms, and that the paradigm with the lower cost (the “target paradigm”) meets our other requirements (e.g. the function requires little energy
to evaluate in the target paradigm). Then, a cryptographic hash by composed with such a function (a) retains many of the security properties of the cryptographic hash, but (b) the relative cost of evaluating the composed hash is lower in the target paradigm. We call such a composed hash function a “heavy hash.”

Slightly more concretely, suppose $H$ is a cryptographic hash with computational cost $C_H$ in some paradigm $P$. Ideally, such a measure of cost is simply the average cost to evaluate the hash function according to the current state of the art (and a conservative estimate on energy prices). Additionally, suppose $f$ is a permutation with cost $C_f$ in hardware paradigm $P_1$ and cost $C_{f,2}$ in paradigm $P_2$ (assume these costs include converting to and from paradigm $P$). Then consider the modified hash function:

$$H'(x) := f(H(x)).$$

Intuitively, $H'$ has cost $C_H + C_f$ in Paradigm $P_1$ and $C_H + C_{f,2}$ in Paradigm $P_2$. We will indeed show that such an $f \circ H$ where $f$ is hard for boolean circuits in a very specific way yields a PoW where the work required scales according to the hardness of $f$. For the proof of work scheme to be effective we will additionally desire that $f$ is not too hard for boolean circuits, else verifying the proof with boolean circuits may be prohibitively expensive. But for this section we will focus simply on the work required to produce a proof.

**Permutations preserve security.** Before elaborating on the application of proofs of work, we give some very straightforward intuition for why one might expect composing certain functions (in particular, those that “preserve min-entropy”) with cryptographic hash functions to at least preserve the security properties of the cryptographic hash.

While this is not actually true in general, something can be said for the specific case that the function composed with the hash is a permutation. In particular, a random oracle, $H$, composed with any permutation, $\Pi$, remains a random oracle. In some sense, this tells us that the composed function $\Pi \circ H$ “inherits” properties of $H$.

**Fact 3.1 (Permuting “preserves” random oracles). If $H : \{0,1\}^n \rightarrow \{0,1\}^m$ is a uniformly random function and $\Pi$ a permutation on $\{0,1\}^m$, then $H' \equiv \Pi \circ H$ is distributed identically to $H$.**

It is not difficult to see that this follows from the fact that composition with any permutation induces an automorphism on the set of all functions.

**Hard permutations from hard functions.** We additionally note that any function can be turned into a permutation via the feistel transformation. For any $g : \{0,1\}^n \rightarrow \{0,1\}^m$, define $\Pi_g : \{0,1\}^{n+m} \rightarrow \{0,1\}^{n+m}$ as follows:

$$\Pi_g(x, y) := (x \oplus g(y), y)$$

Moreover, computing $\Pi_g$ is at least as hard as computing $g$ (modulo the bit-wise XOR), even when amortizing over many random instances.

However, we note that for the specific case of Hashcash-like PoWs, it is not necessary that the hard function, $f$, is permutation. (However, it may be helpful to put the hard function in the form of a permutation if $H$ is “like” a random function and one would like $f \circ H$ to inherit this property.)

**Heavy Hash.** We now outline our construction for modifying cryptographic hash functions with hard functions, before delving into the particular hardness we require of the underlying function.

$$H'(x) := H(f(H(x)), H(x))$$

As mentioned above, we will force the evaluator to evaluate the hard function, $f$, on the output of the hash, $H$. Intuitively, this gives a fresh random input to $f$ on every new input to the hash function. Additionally, to deal with the possibility that $f$ itself may have some nice structure, we require that $H$ is evaluated again on the output and input of $f$. Evaluating $H$ again on the input and output of $f$ effectively specifies a random sparse relation on the inputs and outputs of $f$.

To see why this is helpful, recall the (loose) specification of the Hashcash with respect to some hash function $F$:

For statement $x$, find proof $y$ s.t. $F(x, y)$ has $k$ leading $0$s.

In particular, it involves finding a point which satisfies a sparse relation (find a point on the sub-cube of the domain, specified by a [random] $x$, whose output under $F$ satisfies some sparse predicate, $P$ that is true on any input with $k$ leading zeros).

There is a danger in the above intuition in that $f$ may be tailor made such that determining if an output of $H$, $z^*$, is in the set $\{z : P(f(z)) = 1\}$ may be much easier than computing $f$ itself. To handle this situation, we can simply destroy any

---

28However, such a notion of complexity is more or less intractable theoretically and our propositions will be stated in a more traditional oracle-aided-circuits model. In this setting differing hardware paradigms would correspond to different classes of circuits (topologies and gate functionalities) accompanied with different complexity measures.

29Moreover, if $C < C_f^1 + C_{f,2}^2$, and we extend the domain of $H'$ via the Merkel-Damgard transformation, the relative cost in $P_1$ vs $P_2$ will tend towards $C_f^1/C_{f,2}^2$.
structure in \( P \) using \( H \): look for outputs \( z \) such that \( P(H(z)) = 1 \) (or if we think of \( H \) as the keyed function \( H_t \), then \( z \) such that \( P(H_t(z)) = 1 \)). In other words, the Bitcoin PoW with respect to the function \( H \circ f \circ H \).

### 3.3 Minimal Effective Hardness

To instantiate the Heavy Hash we need a function that satisfies a fairly robust complexity condition. The flavor of lower bound on evaluation complexity we outline in this subsection is not the only requirement we would like, in fact we would also like that this bound is tight, so verification of the proofs of work is not too expensive for digital machines. However, the thrust of this section will be outlining the lower bound requirements that make generating proofs of work on a given hardware platform expensive to perform at scale.

#### 3.3.1 The Complexity Condition

Let \( f : \{0, 1\}^n \rightarrow \{0, 1\}^m \) be some candidate function. While we have the above proof system in mind, we will present the hardness condition in full generality. Before giving the definition, we give an informal description of the various parameters involved in the definition.

- **Input length of** \( f \): \( n \).
- **Output length of** \( f \): \( m \).
- **Bound on work required per successful evaluation overall**: \( w \).
- **Maximum number of successful evaluations required to be “hard”:** \( T \).
- **Probability things do not behave as expected:** \( \varepsilon \).

**Definition 1.** We say \( f \) is \((\varepsilon, w, T)\)-minimal effectively hard (MEH) for circuits if the following holds:

Let \( A \) be a circuit that can “push a button” to receive a fresh uniformly random \( x \in \{0, 1\}^n \). A may do this an arbitrary number of times (pushing the button has some low, fixed cost). \( A \) can also output “candidate solutions:” tuples \((x, y)\). Any output \( x, y \) where \( x \) was not received from pushing the button or \( y \neq f(x) \) is ignored.

Each of \( t = 1, 2, 3, \ldots, T \), and any such \( A \), the probability that \( A \) does less than \( t \cdot w \) “work” yet outputs at least \( t \) distinct tuples \((x_i, f(x_i))\) is at most \( \varepsilon \).

---

31Because we are in a fixed parameter regime, one can always build a look up table for every single input (paying roughly the cost of evaluating every input), subsequent evaluations will be cheap and the amortized cost will tend towards the cost of a look up. This parameter is intended to make irrelevant any “attack” of this form.

32In the context of the PoW, this corresponds to evaluating SHA2.

33For ASICs, work can likely be coarsely interpreted as chip area times clock cycles. Ultimately, what we care about is dollar cost, which correlates to cost of manufacturing and energy cost of running the chip.

34Note that there may be a better cleaner formulation of the necessary conditions to instantiate the heavy hash paradigm. This is just one suggestion.

### 3.4 Analysis in the Random Oracle Model

We now show this transformation, when the modifying function is a permutation increases the cost of a RO-based PoW scheme, relative to a given hardware paradigm.

**Fixed-Cost Random Oracle** In our model, we will assume that invoking the oracle, \( H \), has a single fixed cost, \( C_H \), independent of the hardware paradigm. We assume a non-uniform model computation where the computational measure of cost that obeys standard composition properties.

The following proposition states that the Hashcash scheme, when instantiated with appropriately heavy hash, requires an amount of work that scales with hardness of the underlying function.

**Proposition 3.2.**

If \( f \) is \((\varepsilon, w, T)\)-MEH for circuits, then the Hashcash PoW scheme, when instantiated with the function, \( H' \), defined by \( x \in \{0, 1\}^n \leftrightarrow H'_1(H_1(x)), H'_2(x) \) (where \( H_1 : U \rightarrow U \) and \( H_2 : U \times U \rightarrow \{0, 1\} \) are random functions) and security/sparsity parameter \( \lambda \), is sound: the distribution of an optimal prover’s work that makes non-adaptive calls to \( H_1 \) is statistically close \((\varepsilon + 2^{-\lambda})\) to a scaled exponential/geometric distribution (probability of success with \( w(C_f + 2C_H) \) work is \(-\ln(1 - 2^{-\lambda})e^{\ln(1 - 2^{-\lambda})w}\)).

**Proof.** Let \( P^* \) be a prover. Consider the event, \( E \), that \( P^* \) submits a successful proof \( x^* \) such that \( H' \) has \( \lambda \) leading zeros without calling \( H \) on at least one of \( x^* \) or \( (f(H'(x^*)), H'(x^*)) \). The probability of such an event is \( 2^{-\lambda} \) over the choice of \( H \).

Otherwise (with probability at least \( 1 - 2^{-\lambda} \)), \( P^* \) must have called \( H \) on both \( x^* \) and \( (f(H'(x^*)), H'(x^*)) \). Let \( S \) denote the set of such that \( P^* \) calls \( H \) on both \( x \) and \( (f(H(x)), H(x)) \). Conditioned on \( E \) not happening, the probability that \( P^* \) outputs any viable proof is at most \( 1 - (1 - 2^{-\lambda})^{|S|} \).

Clearly, \( P^* \) has performed at least \( 2|S|C_H \) work. We wish to show that it must have performed \( |S|C_f \) additional work. To do so, we will construct a circuit \( A' \) from that finds \( |S| \) solutions to \( f \) on a (random) batch of inputs.

To construct such a circuit we will simply consider the queries made by \( P^* \). We can organize queries made to \( H \) into two types: (a) those from \( U \) and (b) those from \( U \times U \). Fresh (not-previous queried) queries of type (a) will correspond to “button” presses in the MEH game, and queries of type (b) will correspond to candidate solutions.

If queries of type (a) are made adaptively, we \( A \) will maintain a dictionary to determine if a given query is fresh. This has overhead \( c \cdot q_a \), where \( c \) is some constant and \( q_a \) is the number of queries of type a. By our assumption, if \( P^* \) produces a set \( S \) as described above (of size \(|S|\)) valid solutions then with probability at most \( \varepsilon \), \( P^* \) did at least \( w|S| \) work. \( \square \)
3.5 The conjectured “hard” problem

Here we discuss a concrete implementation of the above paradigm that we conjecture will be cheaper overall to evaluate via PICs hardware (as opposed to PICs). Due to the particularities of the hardware (in particular, the difficulty in getting many bits of reliable output using photonics) we will deviate somewhat from the simple paradigm described above.

3.5.1 Candidate: Random Linear Transform

Informally, we conjecture that Strassen matrix multiplication is essentially optimal (in that it nearly matches the minimal effective hardness), for reasonable parameters, for evaluating the product (with fixed point arithmetic) of most matrices with random vectors.

Our reason for conjecturing that this problem is minimally effective hard (where the true complexity measure is the amortized cost, in dollars, of evaluation) is simply that fixed precision linear algebra is central a variety of domains in computer science particularly computer graphics and machine learning. These computations are performed at incredibly high (and rapidly increasing) throughput, particularly due to the explosive growth of machine learning. Because it is a critical ingredient in so many applications, it is not unreasonable to expect that true cost of fixed-precision linear algebra via ASICs/FPGAs, barring some revolutionary innovation, is somewhat stable and its projected decay will be fairly well understood [55, 56]. On the other hand, in past couple years a number of high-profile start-ups have launched on the premise that ASPICs will be a cheaper means of performing these computations.

Recall that we wish our function to meet the following specifications:

1. $F$ is Minimum Effective Hard for digital hardware
2. $F$ can be accelerated using an analog photonic low precision vector-matrix multiplier

The candidate hard function $F_Q$ is specified in Figure 9. In fact, $F$ is simply a distribution over functions. We conjecture that a random function is minimally effective hard with high probability.

3.5.2 On “Hard” Linear Transformations

At the core of our approach is a linear transformation that requires more work to apply on digital hardware than on optical hardware. Unfortunately, the best we can do here is conjecture a given linear transformation is hard.

Luckily, as mentioned above, matrix vector multiplication is so extensively studied in practice that bounds derived from empirical testing are likely to be quite robust. Moreover, the optical hardware is such that the matrix $Q$ need not be “baked-in” for all time, but can be adjusted in milliseconds to evaluate arbitrary linear transforms. Unfortunately, this is too slow to generate a random matrix with each nonce. (Additionally, large matrices would also require many pseudorandom bits.) But, it is quick enough to change the matrix on every block or every couple of blocks, by say pseudorandomly expanding older blocks, or otherwise. While this appears to make the problem more robust, we note though that this is likely to require either (a) a more robust conjecture on the density of “hard” matrices or (b) new block-chain analysis. This is because such a technique could admit an attack, related to the block withholding attack of Eyal and Sirer [57], whereby a miner searches for block prefixes that lead to “favorable” matrices.

We additionally note that photonics hardware is such that it can perform complex matrix multiplication, provided the matrix $Q$ is unitary. Unfortunately, there are currently obstacles to manufacturing the current hardware prototypes at scale. However, if these difficulties were ironed out in the future we expect that this would yield an even more robust candidate problem.

Finally, it is worth mentioning that when considering linear transformations over finite fields the theoretical situation is somewhat simpler. There, random linear transformations are as hard as any, up to a factor of 2.\textsuperscript{35}

For any transformation $T$, sample random matrices $A$ and $B$

\textsuperscript{35}The problem with embedding such calculations in the real arithmetic computed (e.g. outputting the parities of the result to simulate arithmetic over $\mathbb{F}_2$), is that the hardware is only capable of reliably producing higher order bits.
such that $T = A + B$. Because, marginally, $A$ and $B$ are uniform we get that most matrices are at most a constant factor easier than the hardest one by the simple decomposition:

$$Tx = (A + B)x = Ax + Bx.$$ 

4 Considerations for a Practical Implementation of oPoW

Below, we discuss some of the basic factors required for successful real-world oPoW implementation. We do not intend this list to be exhaustive, but instead, highlight some of the key considerations.

4.1 Energy Savings

For an oPoW implementation to deliver the drastic energy savings relative to equivalent conventional Proof of Work blockchains, two key requirements will need to be satisfied:

1. Miners using photonic co-processors must have a lower total cost per hash (amortized CAPEX + OPEX) than competing hardware (e.g. ASICs, GPUs).

2. The ratio of CAPEX to OPEX in the cost per hash for photonic oPoW miners must be an order of magnitude higher than it is for ASICs and GPUs currently running on networks like Bitcoin and Ethereum.

Based on internal engineering and extensive discussions with researchers and hardware companies working in photonic computing, we believe that these are achievable goals given the state of the art today, however, a live implementation of oPoW will provide an empirical test.

4.2 Decentralization

Supposing drastic energy savings are achieved, it can be argued that mining decentralization would necessarily be a direct result. Below we briefly discuss two aspects of the issue.

4.2.1 Geographic Decentralization

Although there will always be a small energy cost, and therefore some kind of savings associated with operating in cheap-energy regions, energy will no longer be the deciding factor in profitability. There is a pent-up demand for mining participation in big cities and other areas with expensive energy but crypto-friendly laws (e.g., Malta). Currently, potential miners in these areas have no way to get a return on capital because their operating costs would be higher than the value of the cryptocurrency rewards they would be able to capture via mining. oPoW will democratize mining and provide miners an opportunity to operate in more crypto-friendly jurisdictions with lower risks, the rule of law, and lower costs of capital.

We expect to see big miners in low-energy regions continue mining conventional PoW coins, where they will have less competition, as new players emerge in other regions to mine on oPoW networks.

4.2.2 Hardware Manufacturing Decentralization

In addition to energy efficiency, silicon photonics as a platform has the advantage of lower NREs (non-recurring engineering expenses) as silicon photonic circuits are fabricated using older process nodes (e.g., 200nm SOI [58], 90nm SOI [59] vs. 7nm for Bitcoin ASICs [60]). Low NREs will work to decrease barriers to entry and ensure a healthy, competitive supplier market for oPoW miners in the long run. Additionally because oPoW is based on a photonic co-processor architecture that is being applied more generally to AI processing, we expect there to be robust supplier competition in oPoW mining hardware. Not only are there multiple companies commercializing AI photonic co-processors as discussed in the introduction, but there are also other approaches to analog matrix-vector multiplication being investigated, such as crossbar memristor arrays and other electronic brain-inspired architectures which could eventually deliver competing miners to the market. A broader intuition worth mentioning: it is much easier for a single manufacturer to dominate the market for a hash like SHA256 that has no high-performance computing use-case outside cryptocurrency than it is for a single manufacturer to dominate the market for computing a more general operation that is used beyond a particular coin’s PoW.

4.2.3 Fault Tolerance for Maximum Energy Efficiency

Noise generated in photonic co-processors from digital to analog conversions, analog to digital conversions, shot noise, temperature fluctuations and other sources, means that accuracy comes with an energy/performance penalty [41]. In order to maximize the energy-efficiency of analog photonic compute, oPoW must be fault tolerant.

Assuming a matrix size of $256 \times 256$, the HeavyHash design proposed in Section 3 begins to become impractical with significantly more than a $10^{-3}$ error rate in the LSB (least significant bit) of each 4 bit value in the output vector. Error

\[ \text{Error} \]

Moreover, any transform is at most twice as easy with respect to a random input because one can always sample $r, s$ uniformly such that $r + s = x$ and then $Tx = T(r + s) = Tr + Ts$, the sum of the transform of two transformation of (marginally) random inputs.

\[ \text{Moreover, any transform is at most twice as easy with respect to a random input because one can always sample } r, s \text{ uniformly such that } r + s = x \text{ and then } Tx = T(r + s) = Tr + Ts, \text{ the sum of the transform of two transformation of (marginally) random inputs.} \]

\[ 36 \text{Moreover, any transform is at most twice as easy with respect to a random input because one can always sample } r, s \text{ uniformly such that } r + s = x \text{ and then } Tx = T(r + s) = Tr + Ts, \text{ the sum of the transform of two transformation of (marginally) random inputs.} \]

\[ 37 \text{A reasonable medium-term target for commercial photonic co-processors based on the architectures referenced in this paper. It becomes hard to increase matrix size far beyond } 512 \times 512 \text{ as the number of control circuits for tuning increases with the square of matrix size and in some architectures each row added to the matrix requires the addition of another multiplexed wavelength.} \]

\[ 38 \text{There’s a } 0.9999256 = 77.4\% \text{ probability that none of the } 256 \text{ values in the output vector of the matrix-vector operation has no LSB errors, meaning that } 22.6\% \text{ of trials will be wasted, which is acceptable.} \]
rate of $10^{-3}$ is practical in commercial photonics hardware, however, in future iterations of the algorithm, we hope to improve fault tolerance with a long term target of tolerating a $10^{-1}$ error rate in the LSB. This target is consistent with error rates where deep neural networks appear to no longer show resilience to noise [41, 61], therefore it is reasonable to expect that broadly available photonic co-processors for MACs will not be designed to operate at higher noise levels (whether or not there are potential energy savings).

5 Long Term Outlook

**Cryptocurrency** Cryptocurrencies have progressed in the past five years from the concept stage to early commercialization. It is hard to estimate the true long term potential of the technology, however, it is clear that there is an opportunity to increase the efficiency and fairness of the global financial system. Access to cryptocurrency markets can act as a safety valve in crisis situations, and we have already seen this happen as fiat currency crashes in countries like Zimbabwe and Venezuela have led to local spikes in Bitcoin demand. As cryptocurrencies become more stable, functional and user-friendly they will be able to compete with traditional financial services more broadly.

**Silicon Photonic Co-Processors** There is a lot of hope in the photonics industry that the success of silicon photonics in data communications will translate to computational use cases. It is clear that computing with photons instead of electrons offers attractive fundamental advantages, however, numerous practical engineering challenges must be met to apply optical computing broadly. Taking advantage of the standard semiconductor fabrication supply chain by using silicon photonics addresses many of the major problems, however there are some still remaining, such as the absence of a silicon laser source [39]. In the case of AI processing, suitable optical nonlinear components that are compatible with semiconductor foundries are an active area of research, however not yet commercially available. Additionally, components in silicon photonic circuits are typically tuned to adjust for manufacturing variance using microheaters which, leads to an increase in the overall energy consumption of any photonic circuit [40]. We anticipate that the simplicity of the Optical Proof of Work use case (brute force computation designed for photons, with nearly no memory requirements or variability) will prove to be an excellent stepping stone for photonic co-processor technology on it’s way to mainstream commercialization. **Optical PoW** Scaling store-of-value cryptocurrencies, Bitcoin and others, to meet global demand will require both technical and social innovations. Numerous researchers and developers are working to make improvements via off-chain developments such as the Lightning Network [41] and fundamental blockchain innovations such as MimbleWimble / Zcash / Monero (privacy), and DAGs (scalability). Entrepreneurs are improving new-user onboarding and generally smoothing out the experience for non-technical users. However, besides interesting efforts (albeit very centralized) to use renewable energy for Bitcoin mining, Proof of Work has not seen much innovation since the advent of Bitcoin mining ASICs in 2012.

Our goal at PoWx is to change that by taking advantage of next-generation computing. A fundamental shift in the PoW ecosystem is needed to support another order-of-magnitude increase in decentralized store-of-value. While requiring minimal modifications to existing Proof of Work schemes and thus inheriting desirable security properties, Optical Proof of Work has the potential to solve some of the deepest issues faced by Bitcoin and other cryptocurrencies today. oPoW has the promise to unether cryptocurrencies from power plants, enabling geographically decentralized mining and therefore improved security with additional benefits of eliminating the sensitivity of hashrate to coin price, and democratizing issuance. The implementation of oPoW will help accelerate the development of energy-efficient photonics co-processors, acting as a stepping stone to other applications.

Acknowledgements

The authors acknowledge partial financial support from PoWx. We thank Sunil Pai (Stanford) for assistance in prototype chip design and Mustafa Hamood, Stephen Lin, and Jaspreet Joha (SiEPIC/University of British Columbia) for assistance in prototype fabrication and testing. Additionally we would like to thank Guy Corem (Beam), Bram Cohen (Chia), Tom Brand (Starkware), Yichen Shen (Lightelligence), Mitchell Nahmias (Luminous Computing), Yonatan Sampolinsky (DAGlabs), and John Tromp (Grin) for helpful discussions and feedback.

References


[4] Cynthia Dwork and Moni Naor. Pricing via processing or combating junk mail. In CRYPTO, volume 740 of
Archived at http://archive.is/ZvgcL.


https://www.cbeci.org/.


[18] Dario Amodei. AI and compute. 


[56] Jeff Johnson. Making floating point math highly efficient for AI hardware.


