The ASIC Resistance Problem:

A Systematization of Knowledge and Open Questions for Proof-of-Work Consensus

Roberto Santacroce Martins
roberto@santacroce.es
Preprint — April 2026

Abstract

We present a comprehensive systematization of knowledge on ASIC-resistant proof-of-work (PoW) designs. Beginning with Hashcash and SHA-256 as the baseline that motivated all subsequent work, we trace the hardware arms race from CPUs through GPUs, FPGAs, and ASICs, documenting efficiency gaps spanning seven orders of magnitude. We cover memory-hard functions (scrypt, Ethash, Equihash), randomized execution (RandomX, CryptoNight), algorithmic variability (X16R, ProgPoW), prime-number search (Primecoin), and zero-knowledge-based mining (Aleo PoSW). For each, we document theoretical basis, empirical outcome, and lessons learned. We synthesize the theoretical foundations—cumulative memory complexity, bandwidth hardness, fine-grained worst-case reductions, verifiable delay functions, non-amortizability, and VLSI circuit bounds—and define a threat model for ASIC resistance. We examine the underexplored role of difficulty adjustment as a resistance mechanism and propose difficulty-adaptive algorithm rotation. We formally pose five open problems and conclude that provable ASIC resistance remains an unresolved question at the intersection of complexity theory and hardware economics.

Keywords: proof of work, ASIC resistance, Hashcash, SHA-256, memory-hard functions, fine-grained complexity, difficulty adjustment, bandwidth hardness, systematization of knowledge

1. Introduction

1.1 Motivation

In 1997, Back proposed Hashcash [1] as a denial-of-service countermeasure requiring senders to compute partial hash collisions before sending email. In 2004, Finney built Reusable Proofs of Work (RPoW) [2], the first system to treat proof-of-work tokens as transferable digital artifacts—a prototype for digital cash inspired by Szabo's Bit Gold [3] and Dai's b-money [4]. In 2008, Nakamoto combined Hashcash-style PoW with a distributed timestamp server to create Bitcoin [5], launching the first decentralized cryptocurrency. Since then, PoW consensus has faced a persistent tension: the economic incentives securing the network also drive hardware specialization, concentrating mining power in entities that can afford application-specific integrated circuits (ASICs).

This paper systematizes the full landscape of attempts to resist this dynamic. We begin with the baseline—SHA-256 and the hardware arms race it spawned—because understanding *why* SHA-256 is trivially ASIC-friendly is essential to evaluating every subsequent countermeasure. We then survey memory-hard functions, randomized execution, algorithmic variability, prime-number search, and ZK-based PoW, documenting both theoretical claims and empirical outcomes. We synthesize the relevant theoretical foundations, define a threat model, examine difficulty adjustment as an underexplored mechanism, and formally state the open question of provable ASIC resistance.

1.2 Contributions

This paper makes four contributions:

(1) A unified systematization of all major ASIC resistance approaches with a comparative scorecard documenting ASIC advantage, time-to-ASIC, theoretical basis, and current status (Section 2, Table 1).

(2) A synthesis of the relevant theoretical foundations—memory complexity, bandwidth hardness, fine-grained PoW, VDFs, non-amortizability, and VLSI bounds—into a coherent framework identifying precisely where theory falls short (Section 3).

(3) A threat model for ASIC resistance and the proposal of difficulty-adaptive algorithm rotation as a novel, protocol-intrinsic defense mechanism (Section 4).

(4) Five concrete open problems whose resolution would either establish provable ASIC resistance or demonstrate its impossibility (Section 6).

2. Systematization of ASIC Resistance Approaches

2.1 The Baseline: Hashcash, SHA-256, and the Hardware Arms Race

Back's Hashcash [1] defines the simplest possible PoW: find a nonce n such that H(header || n) < T, where H is a cryptographic hash function and T is a target threshold. Bitcoin adopted SHA-256d (double SHA-256) as H. The computation consists of 64 rounds of bitwise rotations, additions mod 2^{32}, and Boolean majority/choice functions operating on eight 32-bit working variables and a 64-entry message schedule. There is no data dependency between hashes (each nonce attempt is independent), no memory requirement beyond the 256-bit state, and no branching—the dataflow is a pure, deterministic, fixed-depth pipeline. These properties make SHA-256 the *ideal* ASIC target: the entire computation maps to a static combinational circuit that can be replicated, pipelined, and clock-optimized without any of the flexibility penalties that general-purpose processors pay.

Finney's RPoW [2], launched in August 2004, deserves attention as the missing link between Hashcash and Bitcoin. RPoW accepted Hashcash tokens and exchanged them for RSA-signed reusable tokens that could be transferred from person to person, with double-spending prevented by a trusted server running on an IBM 4758 secure cryptographic coprocessor. The system's security model relied on transparent remote attestation—anyone could verify the server code via the 4758's hardware attestation, ensuring not even the server operator could forge tokens. RPoW was the first functioning digital cash system based on proof-of-work and a direct intellectual precursor to Bitcoin. Nakamoto's key innovation was replacing Finney's trusted server with a decentralized blockchain, making the PoW both the consensus mechanism and the token-generation process.

The Bitcoin mining hardware evolution proceeded in four phases, each delivering roughly an order of magnitude efficiency improvement [6, 7]:

CPU era (2009–2010): Satoshi's original client mined on standard x86 CPUs at ~5,000 J/GH. A typical desktop could produce ~2–5 MH/s. Mining was accessible to anyone with a personal computer, fulfilling the 'one-CPU-one-vote' vision of the whitepaper.

GPU era (2010–2012): In mid-2010, miners discovered that GPUs' massively parallel architectures could perform SHA-256 50–100× faster than CPUs. A single Radeon HD 5870 achieved ~400 MH/s. GPU mining concentrated power among those with gaming-grade hardware and cheap electricity, beginning the centralization dynamic.

FPGA era (2011–2013): Field-programmable gate arrays offered ~50 J/GH through pipelined SHA-256 implementations. The key innovation was deep pipelining of SHA-256's algebra chain, doubling clock frequency at the cost of latency. FPGAs were a transitional technology; their programmability was unnecessary for a fixed algorithm, making dedicated silicon the logical next step.

ASIC era (2013–present): Canaan Creative's Avalon (January 2013) shipped the first consumer Bitcoin ASIC at 60 GH/s on 130nm silicon [8]. Bitmain entered shortly after, and the ensuing arms race drove chip geometries from 130nm through 65nm, 28nm, 16nm, 7nm, to 5nm. Modern ASICs (Bitmain Antminer S21, ~200 TH/s at ~17.5 J/TH; MicroBT WhatsMiner M60S) achieve roughly $10^7\times$ efficiency over the original CPUs [6]. By removing all unnecessary circuitry—graphics pipelines, branch predictors, cache hierarchies, operating system support—and replacing standard-cell flip-flops with dynamic circuits (reducing transistor count from ~22T to ~4T per latch), ASIC designers achieved efficiency gains fundamentally unattainable by general-purpose hardware [7].

As of early 2026, the Bitcoin network hashrate exceeds 800 EH/s, produced entirely by ASIC silicon [9]. The total elimination of CPU, GPU, and FPGA mining from Bitcoin demonstrates the endpoint of the specialization dynamic: **for any sufficiently simple, fixed, deterministic computation, purpose-built hardware will always dominate.** Every ASIC resistance approach described in the following sections can be understood as an attempt to make the PoW computation complex, variable, or memory-bound enough to prevent this outcome.

2.2 Memory-Hard Functions

The memory-hardness paradigm begins with Percival's scrypt [10], which introduced sequential memory-hard key derivation with a conjectured $\Omega(N^2)$ area-time lower bound. Alwen, Chen, Pietrzak, Reyzin, and Tessaro [11] proved scrypt achieves optimal cumulative memory complexity $\Omega(N^2 \cdot w / \log^2 N)$ in the parallel random oracle model. However, Litecoin's conservative parameters (N=1024, 128 KB) left the bound vacuous in practice; Bitmain's Antminer L3+ (2017) and L7 (2021) achieve ~100× CPU efficiency [12].

Ethash [13] combined a ~1 GB DAG with pseudorandom memory reads. Bitmain's Antminer E3 (2018) achieved 5–10× GPU efficiency; the E9 (2022) pushed to ~10–20× [14]. Equihash [15], grounded in the Generalized Birthday Problem, was breached within two years by Bitmain's Z9 mini at ~15× GPU efficiency. Alcock and Ren [16] critically showed that no tight tradeoff-resistance bound is proven for Equihash.

The critical theoretical limitation was identified by Alwen and Blocki [17]: *any* data-independent memory-hard function has cumulative memory complexity at most $O(n^2 \cdot \log \log n / \log n)$—an impossibility result showing that data-independent MHFs are strictly weaker than data-dependent ones like scrypt. Blocki, Lee, and Zhou [18] further proved that certifying the cumulative memory complexity of a given DAG construction is Unique-Games hard to approximate within any constant factor. **Verifying ASIC resistance is itself computationally intractable.**

2.3 Randomized Execution: RandomX

RandomX [19] (tevador et al., 2019; audited by Trail of Bits [20], Kudelski, X41 D-Sec, and Quarkslab) took a structural departure. Rather than binding a fixed function to memory, it executes randomly generated programs on a virtual machine with 29 instructions, 8 programs per hash, each 256 instructions over 2048 iterations, operating on a 2 MiB scratchpad backed by a 2080 MiB dataset generated via Argon2d and SuperscalarHash. The design thesis was to force any ASIC to converge toward a general-purpose CPU.

The Bitmain Antminer X5, announced September 2023 and mass-shipping 2024–2025, tests this thesis [21, 22]. Built on a custom RISC-V SoC exploiting RandomX's reference RISC-V JIT target, the X5 achieves 212 kH/s at 1,350 W (6.37 J/kH) compared to ~7.7 J/kH for a Ryzen 9 7950X—a **~1.3× efficiency advantage**, qualitatively different from SHA-256's $\sim 10^7\times$ or scrypt's $\sim 10^3\times$. The Antminer X9 (shipping 2026) claims ~2.45 J/kH (3–4×), though independently unvalidated [23]. A RandomX v2 upgrade targets only ~30% ASIC efficiency reduction [24].

RandomX represents the high-water mark of practical ASIC resistance. Its ~1.3× gap validates the CPU-emulation strategy—but the gap's existence validates the skeptics too. Bitmain built a stripped-down

RISC-V CPU, removing everything RandomX doesn't use. **The lesson: any fixed instruction set, however broad, leaves room for stripping.**

2.4 Algorithm Variability and Fork-as-Defense

CryptoNight's variant treadmill—CNv1 (April 2018), CNv2 (October 2018), CN/R (March 2019) [25, 26]—established fork-as-defense before RandomX displaced it. Each fork invalidated committed ASICs but imposed ~6-month coordination costs on the entire ecosystem.

X16R [27] (Ravencoin, 2018) chained 16 hash functions in block-hash-determined order; covert ASICs emerged by mid-2019, triggering X16Rv2 and KawPoW [28]. ProgPoW [29] attempted 'proof-of-GPU' with random programs every ~50 blocks but was never deployed on Ethereum due to governance deadlock. The lesson: *permuting a known instruction set is far weaker than randomizing the program itself*—ASICs can multiplex fixed primitives with small control overhead.

2.5 Alternative Approaches: Primes and ZK-Based PoW

Primecoin [30] (King, 2013) searches for Cunningham and bi-twin prime chains using variable-length big-integer arithmetic with unpredictable branching. No production ASIC has emerged after 11+ years, making it the longest-surviving PoW without dedicated hardware—though this likely reflects market-cap economics rather than proven resistance.

Aleo's Proof of Succinct Work [31, 32] reframes the problem: coinbase puzzles require partial SNARK proofs (Marlin over BLS12-377), explicitly inviting GPU and FPGA specialization because the prover hardware also accelerates transaction privacy. Dedicated ASIC miners (IceRiver, Goldshell) already ship. Aleo has abandoned ASIC resistance as a design goal.

Table 1. *Empirical ASIC resistance outcomes across deployed PoW algorithms.*

Algorithm	Strategy	ASIC Gap	Years to ASIC	Theoretical Basis	Status
SHA-256	None	~10^7×	~2	None	Dominated
Scrypt	Memory-hard	~10^3×	~2	Proven cmc [11]	Dominated
Ethash	Memory-hard	5–20×	~3	Heuristic DAG	PoS migration
Equihash	Memory-hard	~15×	~2	GBP; disputed [16]	Dominated
CryptoNight	Fork treadmill	Reset each fork	~1 ea.	None (reactive)	Replaced
X16R	Algo variability	Broken	~1.5	None	Replaced
RandomX	CPU emulation	~1.3×	~4	Heuristic [19, 20]	**Active**
Aleo PoSW	ZK-based	N/A	<1	Not a goal [31]	Active (no resist.)
Primecoin	Big-int arith.	None yet	11+	None formal	Active (low cap)

3. Theoretical Foundations

3.1 Memory Complexity and Pebbling

The formal study of memory-hard functions proceeds through the pebbling framework. Alwen and Serbinenko [33] introduced cumulative memory complexity (cmc) in the parallel random oracle model and reduced

memory-hardness to graph-theoretic properties of the underlying DAG. Alwen, Blocki, and Pietrzak [34] strengthened this via sustained space complexity and depth-robust graph constructions.

Alwen and Blocki's impossibility result [17] shows that *any* data-independent MHF has cmc at most $O(n^2 \cdot \log \log n / \log n)$—provably weaker than data-dependent constructions. This creates a dilemma: data-dependent MHFs like scrypt are stronger but vulnerable to cache-timing attacks in password-hashing contexts (less relevant for PoW). Blocki, Lee, and Zhou [18] compound the difficulty by proving that computing exact cmc is Unique-Games hard.

3.2 Bandwidth Hardness

Ren and Devadas [35] make a crucial observation: memory complexity captures only the *area* cost of custom hardware, not the *energy* cost. Since off-chip DRAM access energy is roughly constant across platforms (~20 pJ/bit), functions that are *bandwidth-hard*—requiring high sustained memory bandwidth—equalize energy costs more directly. They prove scrypt and Balloon Hashing [36] are bandwidth-hard under explicit assumptions. This is the closest the literature comes to a hardware-realistic cost model for ASIC resistance.

3.3 Fine-Grained Complexity and Provable PoW

Ball, Rosen, Sabin, and Vasudevan [37, 38] construct PoW families whose average-case hardness reduces to worst-case conjectures from fine-grained complexity theory: the Orthogonal Vectors (OV), 3SUM, and All-Pairs Shortest Paths (APSP) conjectures. The resulting PoW inherits $n^{2-o(1)}$ or $n^{3-o(1)}$ hardness via quantitative self-reducibility. The algebraic structure permits zero-knowledge proofs of work and distributed proof generation.

Key limitations: only polynomial prover-verifier gaps (not exponential as in hash-based PoW); non-amortizability required repair via direct-sum arguments [38]; moderate hardness makes Bitcoin-scale consensus impractical. Vassilevska Williams [39] provides the canonical statement of the underlying hypotheses. Komargodski, Schen, and Weinstein [40] recently revisited PoUW from arbitrary matrix multiplication.

3.4 Non-Amortizability and Verifiable Delay Functions

Non-amortizability—the property that computing k proofs costs k times as much as one proof, with no economies of scale—is essential for ASIC resistance but often overlooked. Without it, a specialized pipeline can amortize setup costs across many parallel hash attempts. SHA-256's perfect parallelism (each nonce attempt is independent) is why ASICs achieve $10^7\times$: they simply replicate pipelines.

Verifiable delay functions (VDFs), formalized by Boneh, Bonneau, Bunz, and Fisch [41], provide the strongest form of non-amortizability via inherently sequential computation (repeated squaring in groups of unknown order). Wesolowski [42] and Pietrzak [43] gave efficient constructions. Rotem, Segev, and Shahaf [44] proved generic-group lower bounds. ASIC advantages for modular squaring are limited to small constants, consistent with the sequentiality conjecture. VDFs provide only sequentiality, not memory or bandwidth hardness, making them complementary to—not substitutive for—memory-hard PoW.

3.5 VLSI Complexity and Circuit Lower Bounds

Thompson's VLSI complexity theory [45] established $AT^2 \geq c \cdot N^2 \log^2 N$ for sorting and DFT; Brent and Kung [46] showed $AT^2 = \Omega(n^2)$ for multiplication. **No Thompson-style bound is known for any PoW-useful function.** Proving such bounds for cryptographic hashes would require major complexity-theoretic breakthroughs (likely $P \neq NC$ or stronger). No theorem certifies that specialized hardware cannot achieve more than constant-factor advantage over general-purpose computation for any problem of cryptographic interest.

3.6 Synthesis: What Theory Does and Does Not Provide

Provable ASIC resistance is currently unattained and no impossibility theorem rules it out. The strongest proxies—cumulative memory complexity, sustained space, bandwidth hardness—are model-dependent, capture only necessary conditions, and are frequently incomparable. A formal definition of ASIC resistance remains absent: candidates along the lines of 'every circuit computing f has AT product at most $c \cdot AT_{CPU}(f)$' encounter the problem that any deterministic function can be hardwired—the question is quantitative, not qualitative, and the right cost model is not a clean cryptographic object.

4. Threat Model and Difficulty Adjustment

4.1 Threat Model for ASIC Resistance

We define the ASIC resistance threat model as follows. The **honest miners** use commodity general-purpose hardware (CPUs, GPUs, or consumer-grade accelerators) available on the open market. The **adversary** is a well-funded entity capable of designing and fabricating custom silicon (ASICs or FPGAs) optimized for the specific PoW algorithm. The adversary has access to state-of-the-art semiconductor fabrication (currently 5nm and below) and can invest tens of millions of dollars in chip design and tape-out.

We define the **ASIC advantage ratio** α as the ratio of energy efficiency (J/hash) between the best commodity hardware and the best custom hardware: $\alpha = E_{commodity} / E_{ASIC}$. For SHA-256, $\alpha \approx 10^7$; for RandomX, $\alpha \approx 1.3$. An ASIC-resistant PoW aims to minimize α, ideally achieving $\alpha \leq c$ for some small constant c (e.g., c = 2). A PoW is **strongly ASIC-resistant** if α can be proven bounded by a constant under explicit computational assumptions.

4.2 Current Difficulty Adjustment: Threshold-Only

Standard difficulty adjustment algorithms—Bitcoin's 2016-block retarget [5], Kimoto Gravity Well [47], Dark Gravity Wave [48], DigiShield [49], and Zawy's LWMA [50]—adjust only the *target threshold*. The computational task remains identical regardless of difficulty level; only the acceptance criterion changes. No widely deployed protocol ties the choice of PoW primitive to the current difficulty.

4.3 Partial Precedents

Two partial exceptions exist. RandomX rotates its dataset key K every ~2048 blocks [19]. ProgPoW rotates its random program every ~50 blocks [29]. Neither couples the *computational primitive* to the difficulty metric. Monero's historical strategy of hard-forking every six months [25, 26] demonstrated that algorithmic churn works operationally but is unsustainable due to governance costs.

4.4 Proposal: Difficulty-Adaptive Algorithm Rotation

We propose that difficulty adjustment should modulate not just the target but the *computational profile* of the PoW itself. Concretely, as difficulty increases, the protocol could shift the ratio of operation types (e.g., memory-bound vs. compute-bound, integer vs. floating-point), adjust working-set sizes, or rotate between distinct operator families.

The key insight: if the optimal hardware substrate changes with difficulty, an ASIC optimized for one regime becomes suboptimal at another. This creates a *co-evolutionary pressure* where hardware must remain general. Unlike fork-based churn, difficulty-adaptive rotation is protocol-intrinsic and requires no governance coordination.

Technical challenges: (a) **Grinding resistance**—the operator-mix function must be deterministic from public chain state and non-manipulable. (b) **Verification efficiency**—verifiers must check proofs without replaying the full operator mix. (c) **Security model compatibility**—formalizability within existing frameworks like the Bitcoin

backbone model [51]. (d) **Smooth transitions**—abrupt changes could cause hashrate instability.

5. Related Work

5.1 Proof of Useful Work

Fitzi, Kiayias, Panagiotakos, and Russell's Ofelimos [52] (CRYPTO 2022) provides the first PoUW with formal Bitcoin-backbone security proofs. However, the useful work is ASIC-accelerable by construction. Shoker's Proof of Exercise [53] uses matrix multiplication—the canonical ASIC workload. Neither addresses ASIC resistance as a security goal.

5.2 Proof of Learning and Verification

Jia et al.'s Proof-of-Learning [54] was comprehensively broken by Zhang et al. [55] and shown to be fundamentally unverifiable by Fang et al. [56]. The general lesson: any scheme reducing to 'replay a claimed training trajectory' inherits spoofing attacks because the hypothesis class of plausible training paths to any final weight set is too large.

5.3 Egalitarian Computing

Biryukov and Khovratovich [57] proposed memory-hard-everything (MTP); Dinur and Nadler [58] broke it via tradeoff attacks on Argon2d's data-dependent structure. Karakostas, Kiayias, Nasikas, and Zindros [59] offer the only economic-level formalization of mining egalitarianism, but provide frameworks rather than constructions achieving their own definitions.

6. Open Questions

We identify five concrete research problems:

Problem 1 (Formalization). Define ASIC resistance as a complexity-theoretic property. Candidates include: (i) bounded AT ratio between any circuit and a reference RAM machine, (ii) bounded energy ratio formalizing Ren-Devadas bandwidth hardness [35], or (iii) bounded efficiency ratio in a parameterized hardware cost model. Which formalization best captures real-world ASIC advantage?

Problem 2 (Impossibility). Does there exist a formal impossibility theorem of the form 'no PoW function has polynomially bounded ASIC advantage under assumption X'? The Thompson VLSI framework [45] suggests model-dependence, but no proof rules out a model-independent result.

Problem 3 (Simultaneous properties). Can a single PoW simultaneously achieve non-amortizability (VDF-style [41]), memory/bandwidth hardness [11, 35], provable worst-case hardness [37, 38], and polylog-time verification? No current primitive achieves more than two.

Problem 4 (Difficulty-adaptive rotation). Can difficulty-adaptive algorithm rotation (Section 4.4) be formalized within the Bitcoin backbone model [51] without enabling grinding or selfish-mining attacks?

Problem 5 (The 1.3× question). Is the empirical ~1.3× RandomX/Antminer X5 gap a fundamental lower bound on what CPU-emulating PoW can achieve? If so, can it be proved? If not, what designs can drive it closer to unity?

7. Conclusion and Future Directions

A decade of deployed PoW experimentation and two decades of complexity theory converge on a single diagnosis: **ASIC resistance is currently an empirical property defended by ongoing engineering, not a**

proven cryptographic property. Every deployed 'ASIC-resistant' PoW has either been overtaken (SHA-256, scrypt, Ethash, Equihash, CryptoNight, X16R) or compressed the gap without eliminating it (RandomX, now at ~1.3×). Every theoretical proxy captures a necessary condition in an idealized model but does not certify real-silicon behavior.

The empirical trajectory, however, is encouraging. Each generation of ASIC resistance design has compressed the advantage ratio by roughly an order of magnitude: from 10^7 (SHA-256) to 10^3 (scrypt) to 10^1 (Ethash/Equihash) to 10^0 (RandomX). Whether this trend can reach its logical limit—$\alpha = 1$, where no specialization advantage exists—is the central open question.

We have proposed difficulty-adaptive algorithm rotation as one concrete direction and identified five open problems that, if resolved, would either establish provable ASIC resistance or demonstrate its impossibility. In a companion paper, we explore a specific instantiation direction inspired by the computational properties of neural network inference, which exhibits precisely the combination of memory-wall binding, operator heterogeneity, and precision diversity that the PoW literature has struggled to achieve. We invite the community to pursue both the theoretical foundations and practical constructions.

References

[1] A. Back, "Hashcash — A Denial of Service Counter-Measure," Tech Report, 2002. http://www.hashcash.org/papers/hashcash.pdf

[2] H. Finney, "RPOW — Reusable Proofs of Work," August 2004. https://nakamotoinstitute.org/finney/rpow/

[3] N. Szabo, "Bit Gold," Unenumerated blog, December 2005. https://unenumerated.blogspot.com/2005/12/bit-gold.html

[4] W. Dai, "b-money," 1998. http://www.weidai.com/bmoney.txt

[5] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. https://bitcoin.org/bitcoin.pdf

[6] M. B. Taylor, "The Evolution of Bitcoin Hardware," *IEEE Computer*, vol. 50, no. 9, pp. 58–66, 2017.

[7] Linzhi ASICs, "History of Bitcoin Mining Hardware," Medium, November 2019. https://medium.com/@Linzhi

[8] V. Buterin, "Avalon Ships Bitcoin's First Consumer ASICs," *Bitcoin Magazine*, January 2013.

[9] D-Central Technologies, "ASIC Miners: The Complete Guide to Bitcoin Mining Hardware in 2026," February 2026.

[10] C. Percival, "Stronger Key Derivation via Sequential Memory-Hard Functions," BSDCan 2009. RFC 7914, 2016.

[11] J. Alwen, B. Chen, K. Pietrzak, L. Reyzin, and S. Tessaro, "Scrypt is Maximally Memory-Hard," in *Proc. EUROCRYPT 2017*, LNCS 10212. IACR ePrint 2016/989.

[12] Bitmain Technologies, "Antminer L3+ / L7 Product Specifications," 2017/2021.

[13] V. Buterin and M. Dryja, "Ethash Design Rationale," Ethereum Wiki, 2014–2015. See also G. Wood, *Yellow Paper*, Appendix J.

[14] Bitmain Technologies, "Antminer E3 / E9 Product Specifications," 2018/2022.

[15] A. Biryukov and D. Khovratovich, "Equihash: Asymmetric Proof-of-Work Based on the Generalized Birthday Problem," in *Proc. NDSS 2016*. IACR ePrint 2015/946.

[16] L. Alcock and L. Ren, "A Note on the Security of Equihash," unpublished, 2017.

[17] J. Alwen and J. Blocki, "Efficiently Computing Data-Independent Memory-Hard Functions," in *Proc. CRYPTO 2016*, LNCS 9815.

[18] J. Blocki, S. Lee, and S. Zhou, "On the Computational Complexity of Minimal Cumulative Cost Graph Pebbling," in *Proc. ITCS 2020*.

[19] tevador, hyc, vielmetti, antanst, and SChernykh, "RandomX: Proof of Work Based on Random Code Execution," 2019. https://github.com/tevador/RandomX

[20] Trail of Bits, "Security Assessment of RandomX," Audit Report, 2019. https://blog.trailofbits.com/2019/07/02/state/

[21] D-Central Technologies, "The Antminer X5 Controversy: Verdict on Monero's Newest Miner," 2024.

[22] Bitmain Technologies, "Antminer X5 Specifications: 212 kH/s, 1350W," 2023.

[23] OneMiners, "The Ultimate Guide to Antminer X9," 2026.

[24] CoinBuzzNow, "RandomX v2 Update: Efficiency Boost for Modern CPUs," December 2025.

[25] Monero Project, "CryptoNight Variant History," GitHub monero-project/monero.

[26] S. Chernykh, "Cryptonight Variant 4 aka CryptonightR," PR #5126, monero-project/monero, 2019.

[27] T. Black and S. Weight, "X16R Algorithm," Ravencoin Whitepaper, January 2018.

[28] Ravencoin Project, "X16Rv2 and KawPoW Fork History," 2019–2020.

[29] IfDefElse, "ProgPoW: A Programmatic Proof-of-Work," EIP-1057, May 2018. GitHub: ifdefelse/ProgPOW.

[30] S. King, "Primecoin: Cryptocurrency with Prime Number Proof-of-Work," Whitepaper, July 2013.

[31] Aleo Systems, "Proof of Succinct Work," Aleo Developer Documentation, 2024.

[32] Aleo Systems, "Proof of Stake vs Proof of Work: Aleo's PoSW," Aleo Blog, April 2025.

[33] J. Alwen and V. Serbinenko, "High Parallel Complexity Graphs and Memory-Hard Functions," in *Proc. STOC 2015*. IACR ePrint 2014/238.

[34] J. Alwen, J. Blocki, and K. Pietrzak, "Depth-Robust Graphs and Their Cumulative Memory Complexity," in *Proc. EUROCRYPT 2017*. See also "Sustained Space Complexity" (EUROCRYPT 2018).

[35] L. Ren and S. Devadas, "Bandwidth Hard Functions for ASIC Resistance," in *Proc. TCC 2017*, LNCS 10677. IACR ePrint 2017/225.

[36] D. Boneh, H. Corrigan-Gibbs, and S. Schechter, "Balloon Hashing: A Memory-Hard Function Providing Provable Protection Against Sequential Attacks," in *Proc. ASIACRYPT 2016*.

[37] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Average-Case Fine-Grained Hardness," in *Proc. STOC 2017*. IACR ePrint 2017/202.

[38] M. Ball, A. Rosen, M. Sabin, and P. N. Vasudevan, "Proofs of Work from Worst-Case Assumptions," in *Proc. CRYPTO 2018*, LNCS 10991. IACR ePrint 2018/559.

[39] V. Vassilevska Williams, "On Some Fine-Grained Questions in Algorithms and Complexity," in *Proc. ICM 2018*.

[40] I. Komargodski, S. Schen, and O. Weinstein, "Proofs of Useful Work from Arbitrary Matrix Multiplication," arXiv:2504.09971, 2025.

[41] D. Boneh, J. Bonneau, B. Bunz, and B. Fisch, "Verifiable Delay Functions," in *Proc. CRYPTO 2018*, LNCS 10991.

[42] B. Wesolowski, "Efficient Verifiable Delay Functions," in *Proc. EUROCRYPT 2019*. *J. Cryptology*, 2020.

[43] K. Pietrzak, "Simple Verifiable Delay Functions," in *Proc. ITCS 2019*.

[44] L. Rotem, G. Segev, and I. Shahaf, "Generic-Group Delay Functions Require Hidden-Order Groups," in *Proc. EUROCRYPT 2020*.

[45] C. D. Thompson, "A Complexity Theory for VLSI," PhD thesis, Carnegie Mellon University, 1980. Earlier version in *Proc. STOC 1979*.

[46] R. P. Brent and H. T. Kung, "The Area-Time Complexity of Binary Multiplication," *J. ACM*, vol. 28, no. 3, pp. 521–534, 1981.

[47] Kimoto, "Kimoto Gravity Well," Megacoin, 2013.

[48] E. Duffield, "Dark Gravity Wave," Dash (Darkcoin), 2014.

[49] DigiShield, "DigiShield v3 Difficulty Adjustment," DigiByte, 2014. Adopted by Zcash.

[50] zawy12, "Linearly Weighted Moving Average (LWMA) Difficulty Algorithm," GitHub zawy12/difficulty-algorithms, 2017.

[51] J. Garay, A. Kiayias, and N. Leonardos, "The Bitcoin Backbone Protocol: Analysis and Applications," in *Proc. EUROCRYPT 2015*, LNCS 9057, pp. 281–310.

[52] M. Fitzi, A. Kiayias, G. Panagiotakos, and A. Russell, "Ofelimos: Combinatorial Optimization via Proof-of-Useful-Work," in *Proc. CRYPTO 2022*. IACR ePrint 2021/1379.

[53] A. Shoker, "Sustainable Blockchain through Proof of Exercise," in *Proc. IEEE NCA 2017*.

[54] H. Jia et al., "Proof-of-Learning: Definitions and Practice," in *Proc. IEEE S&P 2021*. arXiv:2103.05633.

[55] R. Zhang et al., "'Adversarial Examples' for Proof-of-Learning," in *Proc. USENIX Security 2022*. arXiv:2108.09454.

[56] C. Fang et al., "On the Fundamental Limits of Formally (Dis)Proving Robustness in Proof-of-Learning," arXiv:2208.03567, 2022.

[57] A. Biryukov and D. Khovratovich, "Egalitarian Computing," in *Proc. USENIX Security 2016*. arXiv:1606.03588.

[58] I. Dinur and N. Nadler, "Time-Space Tradeoffs for Argon2d," in *Proc. CRYPTO 2017*.

[59] A. Karakostas, A. Kiayias, C. Nasikas, and D. Zindros, "Cryptocurrency Egalitarianism: A Quantitative Approach," in *Proc. Tokenomics 2019*. arXiv:1907.02434.

[60] S. Asadi, A. Golovnev, T. Gur, and I. Shinkar, "Worst-Case to Average-Case Reductions via Additive Combinatorics," in *Proc. STOC 2022*.