Sustained Space and Cumulative Complexity Trade-offs for Data-Dependent Memory-Hard Functions

Jeremiah Blocki and Blake Holman

Purdue University

Abstract. Memory-hard functions (MHFs) are a useful cryptographic primitive which can be used to design egalitarian proof of work puzzles and to protect low entropy secrets like passwords against brute-force attackers. Intuitively, a memory-hard function is a function whose evaluation costs are dominated by memory costs even if the attacker uses specialized hardware (FPGAs/ASICs), and several cost metrics have been proposed to quantify this intuition. For example, space-time cost looks at the product of running time and the maximum space usage over the entire execution of an algorithm. Alwen and Serbinenko (STOC 2015) observed that the space-time cost of evaluating a function multiple times may not scale linearly in the number of instances being evaluated and introduced the stricter requirement that a memory-hard function has high cumulative memory complexity (CMC) to ensure that an attacker's amortized space-time costs remain large even if the attacker evaluates the function on multiple different inputs in parallel. Alwen et al. (EUROCRYPT 2018) observed that the notion of CMC still gives the attacker undesirable flexibility in selecting space-time tradeoffs e.g., while the MHF `Scrypt` has maximal CMC $\Omega(N^2)$, an attacker could evaluate the function with constant $O(1)$ memory in time $O(N^2)$. Alwen et al. introduced an even stricter notion of Sustained Space complexity and designed an MHF which has $s = \Omega(N/\log N)$ sustained complexity $t = \Omega(N)$ i.e., any algorithm evaluating the function in the parallel random oracle model must have at least $t = \Omega(N)$ steps where the memory usage is at least $\Omega(N/\log N)$. In this work, we use dynamic pebbling games and dynamic graphs to explore tradeoffs between sustained space complexity and cumulative memory complexity for data-dependent memory-hard functions such as Argon2id and `Scrypt`. We design our own dynamic graph (dMHF) with the property that any dynamic pebbling strategy either (1) has $\Omega(N)$ rounds with $\Omega(N)$ space, or (2) has CMC $\Omega(N^{3-\epsilon})$ — substantially larger than N^2. For Argon2id we show that any dynamic pebbling strategy either(1) has $\Omega(N)$ rounds with $\Omega(N^{1-\epsilon})$ space, or (2) has CMC $\omega(N^2)$. We also present a dynamic version of DRSample (Alwen et al. 2017) for which any dynamic pebbling strategy either (1) has $\Omega(N)$ rounds with $\Omega(N/\log N)$ space, or (2) has CMC $\Omega(N^3/\log N)$.

Keywords: Memory Hard Function · Dynamic Pebbling · Sustained Space Complexity · Cumulative Memory Complexity

1 Introduction

Memory-hard functions (MHFs) are an important cryptographic primitive which have been used to design egalitarian proof of work puzzles [16] and to protect low entropy secrets against brute-force attacks e.g., password hashing. Intuitively, a function is "memory-hard" if for any algorithm the costs associated with evaluating this function are dominated by memory costs — even if the attacker uses specialized hardware such as Field Programmable Gate Arrays (FPGAs) or Application Specific Integrated Circuits (ASICs). Several complexity measures have been proposed to capture this intuition including space-time complexity, cumulative memory complexity (CMC) [7], and sustained space complexity (SSC) [6].

Intuitively, space-time cost considers the product of running time and the maximum space usage across the entire execution trace. For example, suppose we are given an execution trace $\sigma_1, \ldots, \sigma_t$ where σ_i denotes the state of our program at time i. The space-time costs associated with this execution trace would be $t \cdot \max_{i \leq t} |\sigma_i|$. Alwen and Serbinenko [7] observed that space-time complexity is not well suited in situations where an attacker wants to evaluate the function on multiple different inputs in parallel. In particular, the amortized space-time costs associated with multiple parallel computations can be significantly lower than the space-time costs associated with a single execution. Alwen and Blocki [2] later gave pebbling attacks on practical MHF candidates such as Catena and Argon2i demonstrating that this concern is not merely a theoretical issue. Alwen and Serbinenko [7] proposed the notion of cumulative memory complexity (CMC) to address this concern by modeling amortized space-time complexity. Intuitively, the cumulative memory cost of our execution trace $\sigma_1, \ldots, \sigma_t$ would be given by $\sum_{i=1}^{t} |\sigma_i|$. Observe that the cumulative memory cost is a lower bound for the space-time costs since $\sum_{i=1}^{t} |\sigma_i| \leq t \times \max_{i \leq t} |\sigma_i|$. Thus, requiring that a MHF has high CMC is a strictly stronger requirement than than space-time complexity.

If we adopt high CMC as our goal then we want to find a function f which satisfies the requirements that (1) the function can be evaluated in $O(N)$ steps on a sequential machine, and (2) any parallel algorithm evaluating the function has CMC at least $\Omega(N^2)$. We note that because the function can be evaluated in $O(N)$ sequential steps the CMC cannot be larger than $O(N^2)$. In fact, the `Scrypt` MHF [19] has been shown to satisfy both properties in the parallel random oracle model [8]. However, while `Scrypt` has maximal CMC the MHF also allows the attacker undesirable flexibility when selecting space-time trade-offs. For example, an attacker could evaluate `Scrypt` using constant space $O(1)$ in time $O(N^2)$ or the attacker could evaluate the function using space $O(\sqrt{N})$ and time $O(N\sqrt{N})$.

Motivated by this observation, Alwen et al. [6] introduced the stricter requirement of sustained space complexity (SSC). Returning to our example execution trace $\sigma_1, \ldots, \sigma_t$ we would say that this execution trace has s-Sustained Space complexity t' if $|\{i \ : \ |\sigma_i| \geq s\}| \geq t'$ i.e., there are at least t' steps where the memory usage exceeds s. We remark that $st' \leq \sum_i |\sigma_i|$ is a lower bound on the cumulative memory costs so the requirement that a MHF have high SSC is even stricter than requiring high CMC.

Broadly speaking there are two types of MHFs: data-independent memory-hard functions (iMHFs) and data-dependent memory-hard functions (dMHFs) [1]. In an iMHF, the memory access pattern induced by evaluating the function is not allowed to depend on the (potentially sensitive) input. By contrast, a dMHF places no restrictions on the memory access pattern. While iMHFs provide natural resistance to side-channel attacks, this comes at the cost of memory hardness. For example, `Scrypt` is a dMHF with CMC at least $\Omega(N^2)$ while any iMHF has CMC at most $O(N^2 \log\log N/\log N)$ [2]. In the context of password hashing hybrid "id" modes have been proposed to balance side-channel resistance with memory hardness. For example, the MHF Argon2id runs Argon2i (data-independent mode) for $N/2$ steps before switching to Argon2d (data-dependent mode). Optimistically, if there are no side-channel attacks we achieve stronger memory hardness. In the worst case, if there is a side-channel attack, the security of the hybrid mode (e.g., Argon2id) is downgraded to that of the data-independent mode (e.g., Argon2i).

Alwen et al. [6] gave a construction of an iMHF with $s = \Omega(N/\log N)$-sustained space complexity $\Omega(N)$ i.e., any algorithm evaluating this function in the parallel random oracle model requires at least $t' = \Omega(N)$ steps in which the space usage is at least $s = \Omega(N/\log N)$. We remark that this result is (essentially) optimal due to a pebbling result of Hopcroft [15] showing that any directed acyclic graph with N nodes and constant indegree can be pebbled using space at most $s = O(N/\log N)$. Thus, if $s = \omega(N/\log N)$ we cannot guarantee that there are any steps in which the space usage is at least s and this observation can be extended to dMHFs as well. However, the general $O(N/\log N)$-space pebbling strategy of Hopcroft [15] also requires exponential time so the cumulative cost of this pebbling strategy would be exponentially large. While the construction of Alwen et al. [6] was primarily theoretical, Blocki et al. [13] gave a practical iMHF construction with the following trade-off guarantee: any evaluation algorithm either (1) has CMC $\Omega(N^2)$ or (2) has $\Omega(N)$ rounds in which the space usage is at least $\Omega(N/\log N)$ space.

The construction of Blocki et al. [13] achieves (essentially) optimal trade-offs between CMC and SSC. While it is possible that the attacker's $s = \Omega(N/\log N)$ sustained space-complexity is lower than $\Omega(N)$ any such attack would incur a higher penalty on CMC costs. Similarly, general pebbling attacks of Alwen and Blocki [2] against any iMHF simultaneously achieve CMC $O(N^2 \log\log N/\log N)$ and there are also $o(N)$ rounds where the space usage exceeds $O(N \log\log N/\log N)$. However, trade-offs between CMC and SSC have not been explored for dMHFs where the general pebbling attacks of Alwen and Blocki [2] no longer apply. Thus, for a dMHF we might hope to achieve even stronger trade-offs e.g., it may be possible to find a dMHF with the property

[1] Ameri et al. [9] also introduced the notion of a computationally data-independent memory-hard function where the memory access pattern is allowed to depend on the input, but should be computationally bounded adversary should not be able to detect or exploit this dependence.

that any evaluation algorithm either (1) has $\Omega(N)$ rounds in which the space usage is at least $\Omega(N)$, or (2) has CMC at least $\omega(N^2)$.

In this paper, our focus will be on understanding and quantifying SSC and CMC trade-offs for data-dependent memory-hard Functions using dynamic graphs and dynamic pebbling games.

1.1 Our Results

Any attempt to solely analyze the SSC of a (dynamic or otherwise) pebbling graph will lead to weaker lower bounds. In fact, any DAG G with N nodes and constant indegree can be pebbled using at most $s = O\left(\frac{N}{\log N}\right)$ pebbles during any pebbling round [6, 15]. We observe that the pebbling strategy of Hopcroft [15] easily extends to dynamic graphs i.e., a pebbling strategy simply uses Hopcroft [15] to place a pebble on node i using at most $i/\log i$ pebbles. We can then remove pebbles from all nodes except i and repeat this method to pebble node $i+1$ etc.. Thus, if $\mathbb{G}$ is a distribution over DAGs G with constant indegree we can't hope to prove that pebbling G requires $\omega\left(\frac{N}{\log N}\right)$ pebbles for some number of steps, since its $\omega\left(\frac{N}{\log N}\right)$-SSC is zero. However, while Hopcroft's general pebbling strategy uses minimal space $O(N/\log N)$ the pebbling also runs in exponential time. Thus, we can still hope to establish stronger CMC/SSC trade-offs for dMHFs.

Ideally, we want to construct a dynamic pebbling graph in which any strategy must sustain $\Omega(N)$ nodes for $\Omega(N)$ steps, or incurs CC $\Omega(N^3)$. To show that this trade-off is the best we could hope for, we use a result from Lengauer and Tarjan.

Theorem 1 (Corollary 4.2.4 of [17]). *There exists a constant $c > 0$ such that for any DAG G with indegree δ on N nodes and any $N/\log N \leq S \leq N$, there is a legal pebbling $P = \langle P_1, \ldots, P_t \rangle$ of G which uses at most S pebbles ($\max_i\{|P_i|\} \leq S$) and takes time $t \leq S \cdot 2^{2^{c\delta N/S}}$.*

Using Theorem 1 we see that any graph with constant indegree can be pebbled with cS space in time $O(N^2)$ for any $c > 0$, resulting in the following Corollary.

Corollary 1. *For every DAG G with constant indegree δ on N nodes (with N sufficiently large) and every constant $c_1 > 0$, there exists a constant $c_2 > 0$ depending only on δ and c_1 and a pebbling P of G with c_1N-SSC $= 0$ and CC at most c_2N^2.*

Corollary 1 only deals with static graphs, but we can apply it to dynamic graphs as well. When a random node $r(i)$ is revealed upon pebbling node i, we must pebble at most the subgraph induced by nodes $1, \ldots, i$. By corollary 1, pebbling each $r(i)$ using space at most c_1N requires CC at most c_2N^2 for some constants $c_1, c_2 > 0$. Thus, pebbling any dynamic graph on N nodes with space at most c_1N requires CC at most c_2N^3. More generally, for any $c_1 > 0$, $k > 1$, and constant indegree dynamic pebbling graph $\mathbb{G}$ on N (sufficently large) nodes,

there exists a pebbling strategy for $\mathbb{G}$ which uses space at most $c_1 N/\log^k N$ and has CC at most $c_2 N^3 \log^{k-2} N$.

We analyze CMC/SSC tradoffs for four dMHFs. The first dMHF that we analyze is based on a constant indegree dynamic graph that we construct. While the construction is primarily of theoretical interest it achieves (essentially) optimal CMC/SSC tradeoffs — either there are $\Omega(N)$ steps with $\Omega(N)$ pebbles or the cumulative memory complexity is $\Omega(N^{3-\epsilon})$. The second dMHF that we analyze is based on a family of depth-robust graphs [14] with indegree $O(\log N)$. We introduce dynamic edges and prove that (whp) either there are $\Omega(N)$ steps with $\Omega(N)$ pebbles or the cumulative memory complexity is $\Omega(N^3)$. While the first two dMHFs are primarily of theoretical interest we also analyze CMC/SSC tradeoffs for two practical dMHF candidates including Argon2id [1] (winner of the password hashing competition) and DRSample [4]. Our results are summarized in table 1 below.

Dynamic Graph	Space Sustained for $\Omega(N)$ steps	CC
`Scrypt` [8]	$O(1)$	$O(N^2)$
Dynamic EGS	$\Omega(N)$	$\Omega(N^3)$
Dynamic DRSample	$\Omega\left(\frac{N}{\log N}\right)$	$\Omega\left(\frac{N^3}{\log N}\right)$
Argon2id	$e \leq N$	$\tilde{\Omega}(N^4 e^{-2})$
Argon2id (Example)	$\Omega(N^{1-\epsilon})$	$\tilde{\Omega}(N^{2+2\epsilon})$
Our Construction (Example)	$\Omega(N)$	$\Omega(N^{3-\epsilon})$

Table 1. Lower Bounds: SSC vs CC Tradeoffs

Before we elaborate on each of these results, we first describe dynamic pebbling graphs and pebbling strategies in more detail.

1.2 Dynamic Graphs and Dynamic Pebbling Games

Review: Black Pebbling Games and iMHFs Before introducing dynamic graphs and dynamic pebbling games we first review the parallel black pebbling game for regular (static) graphs. The (parallel) black pebbling game is a powerful abstraction that has been used to analyze the cumulative memory complexity (or sustained space complexity) of iMHFs in the random oracle model. In the parallel black pebbling game we are given a directed acyclic graph (DAG) G which initially contains no pebbles $P_0 = \{\}$ and the goal of the pebbling game is to eventually pebble the sink node(s) of G. A legal black pebbling is a sequence $P_0, \ldots, P_t \subseteq V$ of pebbling configurations such that (0) $P_0 = \{\}$ (1) $V \subseteq \bigcup_i P_i$, and (2) for all $i < t$ and for each $v \in P_{i+1} \setminus P_i$ we have $\mathsf{parents}(v) \doteq \{u \ : \ (u, v) \in E\} \subseteq P_i$. Intuitively, each node in G corresponds to an intermediate data label, and placing a pebbling on the graph corresponds to computing the corresponding data label

and placing it in memory. We initially start with no data labels in memory (rule 0) and are not finished until we have computed all of the output labels (rule 1). We also cannot compute a new data label unless all of the dependent data labels are already available in memory (rule 2). In the sequential black pebbling game, we also require that we place at most one new pebble on the graph in each round (i.e., $|P_{i+1} \setminus P_i| \leq 1$), while no such constraint applies in the parallel black pebbling game.

An iMHF $f_{G,H}(x)$ can be viewed as a mode of operation over a DAG G and a hash function H (typically modeled as a random oracle). The source label $L_1 = H(x)$ is typically obtained by hashing the input x and the label of a node $v > 1$ is obtained by hashing the labels of v's parents in G e.g., if $\mathsf{parents}(v) = \{v-1, r(v)\}$ then we might set $L_v = H(L_{v-1}, L_{r(v)})$. The output of the function $f_{G,H}$ is simply the label L_N of the final sink node N — if there are multiple sink nodes the output can be obtained by concatenating all of these labels together. Alwen and Serbinenko [7] proved that in the parallel random oracle model the cumulative memory complexity of the function $f_{G,H}$ is completely captured by the pebbling complexity of the graph G, and Alwen et al. [6] later observed that essentially the same pebbling reduction extends to the notion of sustained space complexity. Here, the cumulative pebbling cost of a pebbling $P = (P_1, \ldots, P_t)$ is $\sum_{i=1}^{t} |P_i|$ and the s-sustained space cost is $|\{i \ : \ |P_i| \geq s\}|$ i.e., the number of pebbling rounds with <u>at least</u> s pebbles on the graph. The cumulative pebbling complexity (resp. s-sustained space complexity) of a graph G is the minimum cumulative pebbling cost (resp. s-sustained space cost) taken over all legal black pebblings of G.

dMHFs and Dynamic Graphs For an iMHF $f_{G,H}$ the data-dependency graph G is completely independent of the input x. By contrast, for a dMHF we might get a different data-dependency graph for each different input x. For example, in `Scrypt` there are $2N$ internal labels and the label for node $N+i$ is computed using the rule $L_{N+i} = H\left(L_{N+i-1} \oplus L_{1+(L_{N+i-1} \mod N)}\right)$. Thus, the data-dependence graph will contain a directed edge from node $r(N+i) = 1 + (L_{N+i-1} \mod N)$ to node $N+i$ where the value $r(N+i)$ depends on the label L_{N+i-1} and, by extension, the input x. We call this edge $(r(N+i), N+i)$ a dynamic edge since it is not fixed a priori and the value $r(N+i)$ will remain hidden until label L_{N+i-1} is computed i.e., until we place a pebble on node $N+i-1$.

In this paper we will use the notion of a dynamic pebbling game to model the complexity of a dMHF. We begin by defining a dynamic pebbling graph following the notation of [9].

Definition 1 (Dynamic Pebbling Graph [9]). *A <u>dynamic pebbling graph</u> $\mathbb{G}$ is a distribution over DAGs $G = (V, E)$ with $V = [N]$ and edges $E = E_{\mathsf{static}} \cup E_{\mathsf{dynamic}}$, consisting of <u>static edges</u>*

$$E_{\mathsf{static}} \subseteq \{(i,j) : 1 \leq i < j \leq N\}$$

and <u>dynamic edges</u>

$$E_{\mathsf{dynamic}} = \{(r(i), i) : i \in V'\},$$

for some $V' \subseteq V \setminus \{1\}$. Here, $r(i) < i-1$ is not revealed until $i-1$ is pebbled.

Strategies for pebbling such graphs can simply be thought of as algorithms, which place pebbles according to some set of instructions, possibly reacting to the dynamic edges as they are discovered. More formally, strategies are functions that output legal pebbling steps when given a partial graph.

Definition 2 (Dynamic Pebbling Strategy). *A dynamic pebbling strategy $\mathcal{S}$ is a function that takes as input*

1. *an integer $i \leq N$,*
2. *an initial pebbling configuration $P_0^i \subseteq [i]$ with $i \in P_0^i$, and*
3. *a partial graph $G_{\leq i+1}$,*

where the partial graph $G_{\leq i}$ is the subgraph of G induced by the nodes $1, \ldots, i$. The output of $\mathcal{S}(i, P_0^i, G_{\leq i+1})$ is a legal sequence of pebbling moves $P_1^i, \ldots, P_r^i$ that will be used in the next phase to place a pebble on node $i+1$, so that $i+1 \in P_{r_i}^i \subseteq [i+1]$. Given $G \sim \mathbb{G}$, we let $\mathcal{S}(G)$ denote the sequence of pebbling moves $\left\langle P_1^0, \ldots, P_{r0}^0, P_1^1, \ldots, P_{r_1}^{N-1}, \ldots, P_{r_{N-1}}^{N-1} \right\rangle$. Here, $P_1^i, \ldots, P_{r_i}^i = \mathcal{S}\left(i, P_0^i, G_{\leq i+1}\right)$, $P_0^i = P_{r_{i-1}}^{i-1}$, and $P_0^0 = \emptyset$. We call $\mathcal{S}(G)$ a pebbling (for G.)

We note that even after a the pebbling strategy $\mathcal{S}$ is fixed the final pebbling $\mathcal{S}(G)$ is not determined until the graph $G \sim \mathbb{G}$ has been chosen i.e., all of the dynamic edges have been revealed. In particular, this means that the cumulative (resp. sustained-space) cost associated with $\mathcal{S}$ can also vary depending on which dynamic edges are sampled. However, once $\mathcal{S}$ and G are fixed we can define the cumulative pebbling cost of $P = S(G) = \langle P_1, \ldots, P_T \rangle$ as $\sum_{i=1}^{T} |P_i|$. Similarly, the s-sustained space cost is $|\{i \ : \ |P_i| \geq s\}|$. Our dynamic pebbling dMHF lower-bounds will take the following form for any dynamic pebbling strategy $\mathcal{S}$ with high probability (over the sampling of $G \sim \mathbb{G}$) when $P = (P_1, \ldots, P_T) = \mathcal{S}(G)$ we either have (1) $\sum_{i=1}^{T} |P_i| \geq LB_1(N)$, or (2) $|\{i \ : \ |P_i| \geq s\}| \geq LB_2(N)$. Where the value s and the exact functions LB_1 and LB_2 will depend on the particular dMHF we are analyzing.

Open Research Challenge: Dynamic Pebbling Reductions For iMHFs it is known that, in the parallel random oracle model, the cumulative memory complexity of the function $f_{G,H}$ is fully characterized by the cumulative pebbling cost of the corresponding data-dependency graph G similar for sustained space complexity [6]. By contrast, there is no formal reduction proving that the cumulative memory complexity (resp. sustained space complexity) of a dMHF is captured by the dynamic pebbling game. In this sense a dynamic pebbling lower bound would not absolutely rule out the possibility of a more efficient attack — unless one can establish a dynamic pebbling reduction. Establishing a formal reduction between dynamic pebbling costs and the cumulative memory complexity of the associated dMHF is a major open research challenge. In the meantime, we can still interpret a dynamic pebbling lower bound as ruling out "natural" attacks and providing compelling evidence that the associated dMHF is secure.

1.3 Trad-Offs for dMHFs

Now we elaborate on the results shown in Table 1.

Our Construction We construct a dMHF (dynamic graph) with constant indegree and prove that for any pebbling strategy $\mathcal{S}$ that, except with negligible probability over the sampled graph $G \sim \mathbb{G}$, the pebbling $P = (P_1, \ldots, P_T) = \mathcal{S}(G)$ satisfies either (1) $|\{i \in [T] \ : \ |P_i| \geq c_1 N\}| \geq c_2 N$, or (2) $\sum_{i=1}^{T} |P_i| \geq c_3 N^{3-\epsilon}$. Here, $\epsilon > 0$ can be arbitrary and the constants $c_1, c_2, c_3 > 0$ depend only on ϵ. We remark that the naive sequential pebbling strategy (i.e., set $P_i = \{1, \ldots, i\}$ for each $i = 1, \ldots, N$) has $s = N/2$-sustained space complexity $N/2$ and cumulative memory cost $O(N^2)$. Our results tell us that any pebbling strategy with lower sustained space complexity must pay a massive penalty in terms of a higher CMC cost.

Dynamic EGS The second graph we examine is based on a family of depth-robust graphs constructed by Erdős et al., which we call EGS [14]. While the indegree $O(\log N)$ of these graphs is a bit larger than we might desire, the cumulative pebbling cost of the graph G is $\Omega(N^2)$ [5]. However, the sustained space-complexity of EGS has not been studied previously. We add dynamic edges to EGS to obtain a dynamic graph and show that, for suitable choices of the constants $c_1, c_2, c_3 > 0$, (whp) the pebbling $(P_1, \ldots, P_T) = \mathcal{S}(G)$ produced by any dynamic pebbling strategy satisfies either (1) $|\{i \ : \ |P_i| \geq c_1 N\}| \geq c_2 N$ or (2) $\sum_{i=1}^{T} |P_i| \geq c_3 N^3$. In particular, either there are $\Omega(N)$ rounds where the space usage is $\Omega(N)$ or the cumulative pebbling cost is massive $\Omega(N^3)$.

Dynamic DRSample We next consider DRSample, a randomized algorithm that, except with negligible probability, outputs a DAG G with cumulative pebbling cost $\Omega\left(\frac{N^2}{\log N}\right)$ and maximum indegree 2 [4]. Alwen et al. [4] implemented the corresponding iMHF and demonstrated that it is practical i.e., the execution time for a graph on N nodes is equivalent to Argon2i. While the intended use case for DRSample was to generate a static DAG G for an iMHF $f_{G,H}$ we can easily modify the definition to include dynamic (data-dependent) edges. We prove that the dynamic version $\mathbb{G}$ of DRSample achieves the following CMC/SMC trade-offs: for any dynamic pebbling strategy $\mathcal{S}$ with high probability (over the selection of $G \sim \mathbb{G}$) the pebbling $(P_1, \ldots, P_T) = \mathcal{S}(g)$ either satisfies (1) $|\{i \ : \ |P_i| \geq c_1 N/\log N\}| \geq c_2 N$, or (2) $\sum_{i=1}^{T} |P_i| \geq c_3 N^3/\log N$. In particular, either there are $\Omega(N)$ rounds with $\Omega(N/\log N)$ pebbles on the graph or we pay a massive penalty in our cumulative pebbling costs.

Argon2id Argon2 is a collection MHFs that won the Password Hashing Competition in 2015 [18]. There are three modes of Argon2: Argon2i, Argon2d, and Argon2id. The Argon2 designers initially recommended Argon2i (data-independent

mode) for password hashing to protect against side-channel attacks. This recommendation was later changed to Argon2id (hybrid mode) after Alwen and Blocki [2, 3] found pebbling attacks on Argon2i which reduced the cumulative memory complexity — the pebbling attacks do no extend to data-dependent modes such as Argon2id. While Argon2i has weaker theoretical guarantees than DRSample [11, 4], Argon2 is available in cryptographic libraries such as libsodium and has seen wider use in practice. In particular, the cumulative complexity of Argon2i is at most $O(N^{1.768})$ and at least $\tilde{\Omega}(N^{1.75})$. We are able to establish stronger tradeoffs for Argon2id. In particular, for any parameter e and any pebbling strategy $\mathcal{S}$ we can show that (except with negligible probability over the selection of the graph $G \sim \mathbb{G}$) the pebbling $(P_1, \ldots, P_t) = \mathcal{S}(G)$ satisfies either (1) $|\{i : |P_i| \geq e\}| \geq c_1 N$, or (2) $\sum_{i=1}^{T} |P_i| \geq c_2 N^4 e^{-2} \log^{-c_3} N$ for suitable constants $c_1, c_2, c_3 > 0$. As a concrete example if we set $e = N^{1-\epsilon}$ there are $\Omega(N)$ rounds with at least e pebbles or the cumulative memory cost is at least $\tilde{\Omega}(N^{2+2\epsilon}) = \omega(N^2)$. We remark that one can separately prove an absolute lower bound of $\Omega(N^2)$ for the cumulative pebbling complexity of Argon2id.

1.4 Technical Overview

We develop two techniques for proving CMC/SSC trade-offs for dynamic graphs. The first general technique is to define an indicator random variable $\mathsf{unlucky}_i$ for each dynamic edge $(r(i), i)$. Intuitively, we define $\mathsf{unlucky}_i = 1$ to be the event that either (1) the dynamic pebbling strategy already had a lot of pebbles (say $s = \Omega(N)$) on the graph when the edge $(r(i), i)$ was revealed, or (2) the particular choice edge $(r(i), i)$ will require us to re-pebble a lot of previously pebbled nodes. Our general strategy is to argue that the following:

1. For any sequence of bits $b_1, \ldots, b_{i-1} \in \{0, 1\}$ we have $\Pr[\mathsf{unlucky}_i \mid \forall j < i, \mathsf{unlucky}_j = b_j] \geq p$. While the events $\mathsf{unlucky}_i$ do not need to be independent, the conditional probability that $\mathsf{unlucky}_i$ is always $\geq p$ for any prior outcomes $\mathsf{unlucky}_1, \ldots, \mathsf{unlucky}_{i-1}$.
2. For some suitable constant $c \in (0, 1)$ and any $i \in [cN, N]$ with $\mathsf{unlucky}_i = 1$ either we had $s = \Omega(N)$ pebbles on the graph when $r(i)$ was revealed or the cumulative pebbling cost to place a pebble on node i will be high (say M)
3. We apply generalized concentration bounds to argue that (whp) we have $\sum_{i=cN}^{N} \mathsf{unlucky}_i \geq p(1-c)N/2$.
4. Assuming there are at least $p(1-c)N/2$ unlucky rounds $i > cN$ we either (1) have $s = \Omega(N)$ pebbles on the graph for $p(1-c)N/4$ pebbling rounds, or (2) we pay CMC cost at least M at least $p(1-c)N/4$ separate times for a total cost of $p(1-c)NM/4$.

To prove our SSC/CMC trade-offs for Argon2id we generalize and a technique introduced in [8] to analyze `Scrypt`. In particular, [8] observed that if we start with e pebbles on a line graph and are challenged to re-pebble a random node $r(i)$ on the line graph then it will take us at least $\frac{N}{4e}$ steps in expectation to place a pebble on a random node $r(i)$. Suppose that $r(i)$ is revealed at time

t_1 and a pebble is placed on node $r(i)$ at time $t_2 \geq t_1$. The challenge $r(i)$ is called "easy" if for some $t \leq t_2$ there were fewer than $|P_{t_2-t}| < \frac{N}{8t}$ pebbles on the graph at time $t_2 - t$ — in this case even if $r(i)$ had been revealed at time $t_2 - t$ we would have expected that it takes at least $\frac{N}{4\frac{N}{8t}} = 2t$ rounds to place a pebble on node $r(i)$. Thus, there is a good chance (at least $\frac{1}{2}$) that the challenge $r(i)$ is "hard" meaning that $|P_{t_2-t}| \geq \frac{N}{8t}$ for every $t \leq t_2$. Alwen et al. [8] then apply concentration bounds to argue that there are a lot of "hard" rounds which allowed them to prove that (whp) the cumulative pebbling cost for `Scrypt` is at least $\Omega(N^2)$.

We can generalize the argument of Alwen et al. [8] by exploiting the fact that Argon2i provides stronger (fractional) depth-robustness guarantees than the line graph [12]. In particular, if we start with with e pebbles on Argon2i and are challenged to place a pebble on a random node $r(i)$ we can argue that it will take us at least $\tilde{\Omega}\left((N/e)^3\right)$ steps to re-pebble node $r(i)$ in expectation. With this observation in mind we can redefine "hard" challenges to require that $|P_{t_2-t}| = \tilde{\Omega}((N/t)^3)$ for every $t \leq t_2$ — where t_2 is the time when we actually placed a pebble on node $r(i)$. Fixing $e = N^{1-\epsilon}$ we can argue that either (1) there are $\Omega(N)$ rounds with at least e pebbles on the graph, or (2) there are a lot of "hard" rounds where we started with at most e pebbles on the graph. In the second case we can argue that the cumulative pebbling cost is at least $\tilde{\Omega}(N^{2+2\epsilon})$.

Our Construction We construct a family of dynamic graphs $\mathbb{G}_D^N$ with $O(N)$ nodes and indegree 2 which has essentially optimal CMC/SSC tradeoffs. We rely on several building blocks to construct our dynamic graphs. The first building block is the notion of a maximally ST-robust graph which was recently introduced by Blocki and Cinkoske [10]. Intuitively, a maximally ST-robust graph is a DAG G with has N inputs (sources) and N outputs (sinks) with the following property: for any $k \leq N$ we can delete any subset S of k nodes from the graph and there will remain subsets A of $|A| \geq N - k$ inputs and B of $|B| \geq N - k$ outputs such that for every pair $u \in A, v \in B$ the graph $G - S$ still contains a directed path from u to v. Blocki and Cinkoske [10] gave a construction of a maximally ST-robust graph with linear size $O(N)$ and constant indegree. The second building block is a family of depth-robust graphs which we overlay on top of the source nodes of our maximally ST-robust graph. Finally, we add our data-dependent layer such that each dynamic edge $(r(i), i)$ uses a uniformly random output node $r(i)$ from our maximally ST-robust graph. Intuitively, when $r(i)$ is revealed we will get "unlucky" if either we have more than k pebbles on the graph or $r(i) \in B$, which happens with probability at least $1 - k/N$. Then whenever we get unlucky, we either have many pebbles on the graph or we will need to repebble the entire set A of $N - k$ inputs before node $r(i)$ can be pebbled. By overlaying a depth-robust graph over the input nodes we can ensure that either (1) $k = \Omega(N)$, or (2) we get unlucky with constant probability and repebbling A requires cumulative cost $\Omega(N^{2-\epsilon})$. If we get unlucky a linear number of times with respect to N (which happens with overwhelming probability) then we either sustained $\Omega(N)$ pebbles for $\Omega(N)$ steps or incurred CC $\Omega(N^{3-\epsilon})$

Dynamic EGS To prove our CMC/SSC trade-off for EGS we primarily rely on the known observation that these graphs $G = (V = [N], E)$ satisfy a key property called δ-*local expansion.* If G is a δ-local expander, then for any $S \subseteq [N]$, the graph $G - S$ contains a directed path of length $N - O(|S|)$. Intuitively, if we started with pebbles on S and we were challenged to place a pebble on one of the last cN nodes on this directed path then we would need to repebble $(1-c)N - O(|S|)$ nodes beforehand. For a suitable constant $0 < c < 1$ if $|S| = o(N)$ we can argue that the cumulative memory cost associated with repebbling $r(i)$ would be at least $\Omega(N^2)$ in this case. Observing that the probability of getting an unlucky challenge is at least $cN/N = c$ it follows that there are at least $\Omega(N)$ unlucky challenges. Thus, we either have $\Omega(N)$ challenge rounds where our initial space usage was $\Omega(N)$ or we have $\Omega(N)$ challenge rounds where we pay CMC cost $\Omega(N^2)$ — in the later case our total CMC cost is $\Omega(N^3)$.

Dynamic DRSample Our argument follows a similar pattern as our dynamic pebbling analysis of EGS. One key difference is that the DRSample graph G is less depth-robust than EGS due to the fact that DRSample has constant indegree. Instead we rely on the notion of a "metagraph" where groups of $O(\log N)$ nodes in DRSample are "merged" into a single metanode. Alwen et al. [4] showed that the metagraph G' for DRSample had $N' = O(N/\log N)$ nodes and satisfied the key-property that for every subset $S' \subseteq [N']$ of metanodes that the graph $G' - S'$ still had a path of length $(1-\eta)N' - O(|S'|)$ for some suitably small constant $\eta > 0$. A path in the metagraph extrapolates back to a path of length $O(N)$ in the original graph. At this point our argument is similar to EGS with the difference that repebbling the graph will be expensive when we begin the challenge with more than $N' = O(N/\log N)$ pebbles on the graph. Thus, we can argue that either (1) we have $\Omega(N)$ challenge rounds where we start with $\Omega(N/\log N)$ pebbles on the graph or (2) there are at least $\Omega(N)$ challenge rounds where we start with fewer than $O(N/\log N)$ pebbles on the graph and we pay CMC costs $\Omega(N^2/\log N)$ to repebble nodes while responding to the challenge complete the challenge. In the latter case the total CMC cost over all challenge rounds is $\Omega(N^3/\log N)$.

2 Preliminaries

We let $[N] = \{1, 2, \ldots, N\}$ and $[i : j] = \{i, i+1, \ldots, j-1, j\}$. For any list $A = \langle a_1, \ldots, a_n \rangle$, we let A_i denote the ith entry of A. For a DAG $G = (V, E)$ and any set S, we let $G - S$ denote the graph $G' = (V', E')$ such that $V' = V \setminus S$ and $E' = \{(u, v) \mid (u, v) \in E, u, v \in V'\}$. Whenever we implicitly refer to some $x \in \mathbb{R}$ as an integer, we always mean $\lfloor x \rfloor$. For example, $[x] = \{1, 2, \ldots, \lfloor x \rfloor\}$. For some set S, we use the notation $y \in_R S$ to indicate that y is sampled from S uniformly at random.

2.1 Dynamic Pebbling Notation

We formalize some convenient pebbling notation. Fix some dynamic pebbling strategy $\mathcal{S}$, $G = ([N], E)$, and let $P = S(G) = \langle P_1, \ldots, P_T \rangle$ be the pebbling that is produced when $G = (V, E) \sim \mathbb{G}$ is sampled. For each $v \in V$ let $\mathsf{parents}_G(i) = \{j \mid (j, i) \in E\}$ denote the parents of node v. When the graph G is clear from context we will omit G from the subscript and simply write $\mathsf{parents}(i)$. For $i \in [N]$ in which there exists a dynamic edge $(r(i), i)$, let $P(i)$ denote the pebbling configuration during the round $s(i)$ when $r(i)$ was first discovered. That is, $P(i) = P_{s(i)}$, where

$$s(i) = \begin{cases} 1 & \text{if } i = 1, \text{ and} \\ \min\{j \in [T] \mid i - 1 \in P_j\} & \text{otherwise.} \end{cases}$$

Similarly, we let $t(i) = \min_{k \geq s(i) \in [T]}\{k \mid r(i) \in P_k\}$ denotes the first round in which $r(i)$ is pebbled after $r(i)$ is revealed in round $s(i)$.

2.2 Generalized Hoeffding Inequality

the pebbling $P = \mathcal{S}(G)$ and its associated costs will depend on the particular graph $G \sim \mathbb{G}$. Thus, when analyzing the cumulative memory complexity and/or sustained space complexity of a dynamic graph we are inherently making a probabilistic claim. In particular, we would like to argue that a particular lower bound on the pebbling cost of $\mathcal{S}(G)$ holds with high probability — over the selection of $\mathcal{G} \sim \mathbb{G}$. We use the Generalized Hoeffding's Inequality to lowerbound these values [8].

Lemma 1 (Generalized Hoeffding's Inequality [8]). *If $V_1, \ldots, V_Q$ are binary random variables such that for any i ($0 \leq i \leq Q$) and any values $v_1, v_2, \ldots, v_i$,*

$$\Pr[V_{i+1} = 1 \mid V_1 = v_1, \ldots, V_i = v_i] \geq \rho, \tag{1}$$

then for any $\epsilon > 0$, with probability at least $1 - e^{-2\epsilon^2 Q}$, $\sum_{i=1}^{Q} V_i \geq Q(p - \epsilon)$.

2.3 Useful Graphs and Their Pebbling Complexity

Naturally, we define notions of measuring the time-space requirements for pebbling dynamic graphs. Cumulative complexity refers to the total number of pebbles used at each step to pebble a graph, while sustained space complexity describes how many steps a certain amount of pebbles were on the graph.

Definition 3 (Pebbling Complexity). *Let $\mathbb{G}$ be a dynamic pebbling graph, G be a graph in the sample space of $\mathbb{G}$, $\mathcal{P}$ be the set of legal pebblings of G, and $\mathcal{S}$ be the set of pebbling strategies for $\mathbb{G}$. We define the cumulative complexity of a*

- *pebbling $P = \langle P_1, \ldots, P_T \rangle \in \mathcal{P}$ as $\mathsf{cc}(P) = \sum_{i=1}^{T} |P_i|$,*

- *a sequence of pebbling moves* $\langle P_i, \ldots, P_j \rangle$ *as* $\mathsf{cc}(P, i, j) = \sum_{k=i}^{j} |P_k|$
- *graph* G *as* $\mathsf{cc}(G) = \min_{P \in \mathcal{P}}\{\mathsf{cc}(P)\}$*, and*
- *dynamic pebbling graph* $\mathbb{G}$ *as* $\mathsf{cc}(G) \min_{S \in \mathcal{S}}\{\mathbb{E}_{G \sim \mathbb{G}}[\mathsf{cc}(S(G))]\}$.

Likewise, we define the s-sustained space complexity of a

- *pebbling* P *as* $\mathsf{ss}(P, s) = |\{i \mid |P_i| \geq s, i \in [T]\}|$,
- *graph* G *as* $\mathsf{ss}(G, s) = \min_{P \in \mathcal{P}}\{\mathsf{ss}(P, s)\}$*, and*
- *dynamic pebbling graph* $\mathbb{G}$ *as* $\min_{S \in \mathcal{S}}\{\mathbb{E}_{G \sim \mathbb{G}}[\mathsf{ss}(S(G), s)]\}$.

For notational purposes, we also define the opposite of s-sustained space complexity called (p, ℓ)-low memory.

Definition 4 (Low Memory Pebbling). *Let* $\mathcal{S}$ *be a pebbling strategy for a graph distribution* $\mathbb{G}$ *and* G *be any graph in the sample space of* $\mathbb{G}$*. We say that* $P = S(G)$ *is a* (p, ℓ)*-low memory pebbling for* G *if*

1. *there exists* $A \subseteq [T]$ *such that* $|A| \leq \ell N$*, and*
2. *for all* $i \in A \setminus [T]$ *we have* $|P_i| \leq pN$.

The cumulative complexity of a graph is tightly correlated with the notion of depth robustness, the property of a graph having long paths even when many nodes are removed from the graph.

Definition 5 (Depth Robustness). *A DAG* $G = (V, E)$ *is* (e, d)*-depth robust if for any* $S \subseteq V$ *of size at most* e*, there exists a path of length* d *in* $G - S$.

Throughout this paper we make use of the following remark on node-deletion.

Remark 1 (of [5]). Let G be an (e, d)-depth robust graph. Then for any $S \subseteq V(G)$ of size $k \leq e$, the graph $G - S$ is $(e - k, d)$-depth robust.

We rely heavily on a lowerbound for the cumulative complexity of graphs according to their depth-robustness.

Theorem 2 (of [5]). *Let* G *be an* (e, d)*-depth-robust DAG, then* $\mathsf{cc}(G) > ed$.

3 A Theoretical MHF with Ideal Trade-Off

In this section we use an $(e = \Omega(N), d)$-depth robust graph D with constant indegree and a maximally ST-robust graph to construct a dynamic graph $\mathbb{G}_D^N$ with the property that for any pebbling strategy $\mathcal{S}$ with high probability either there are at least $\Omega(N)$ rounds with at least $\Omega(N)$ pebbles on the graph) or $\mathsf{cc}(\mathcal{S}(G)) \geq \Omega(N^2 d)$. Furthermore, if D has constant indegree than any graph G in the support of $\mathbb{G}_D^N$ also has constant indegree.

Theorem 3. *Let* D *be an* $(e = 2pN, d)$*-depth robust graph. There exist constants* $0 < c, c_1, p, \ell < 1$*, such that for any strategy* $\mathcal{S}$*, except with probability at most* $\exp(-2(1 - p - c_1)^2 N)$*, either* $\mathsf{ss}(\mathcal{S}(G), pN) > \ell N$ *or* $\mathsf{cc}(\mathcal{S}(G)) \geq cN^2 d$*, where the probability is taken over the choice of* $G \sim \mathbb{G}_D^N$.

For every constant $\epsilon > 0$ and every $N \geq 1$ Schnitger [20] gave a construction of a DAG $\mathsf{Grates}_{N,\epsilon}$ which is $(\Omega(N), \Omega(N^{1-\epsilon}))$-depth robust and has constant indegree. In particular, if we instantiate D using the $(\Omega(N), \Omega(N^{1-\epsilon}))$-depth robust graph $\mathsf{Grates}_{N,\epsilon}$ we obtain the following corollary which says that (whp) either our cc cost is at least $\Omega(N^{3-\epsilon})$ or we will have $\Omega(N)$ rounds with $\Omega(N)$ pebbles.

Corollary 2 (of Theorem 3). *For any $\epsilon > 0$, there exist constants $0 < c, c', c'', p, \ell < 1$ such that for any strategy $\mathcal{S}$, except with probability at most $\exp(-2(1-p-c')^2 N)$, either $\mathsf{ss}(\mathcal{S}(P), pN) > \ell N$ or $\mathsf{cc}(P) \geq c'' N^{3-\epsilon}$, where the probability is taken over the choice of $G \sim \mathbb{G}^N_{\mathsf{Grates}_{N,\epsilon}}$.*

3.1 The Construction

The dynamic graph $\mathbb{G}^N_D$ consists of three components. A maximally ST robust graph with input and output sets of size N, a highly depth-robust graph overlayed on the input set as seen in Figure 1, and a line graph with each node having a dynamic edge from a node sampled uniformly at random from the output set of our ST robust graph. A visualization of the complete construction of $\mathbb{G}^N_D$ is shown in Figure 2. We elaborate on each component in further detail below.

ST-robust graphs play an integral role in our construction [10]. ST-robust graphs are DAGs that have a high connectivity between the sources and sinks even when many nodes are removed. We use ST-robust graphs of the strongest variety—ones that have the maximum possible paths from the inputs to the outputs given arbitrary node deletion.

Definition 6 (ST-Robustness [10]). *Let $G = (V, E)$ be a DAG with n inputs denoted by set I and n outputs denoted by set O. Then G is $\underline{(k_1, k_2)\text{-ST robust}}$ if for all $D \subset V(G)$ with $|D| \leq k_1$ there exists a subgraph H of $G - D$ with $|I \cap V(H)| \geq k_2$ and $|O \cap V(H)| \geq k_2$ such that for all $s \in I \cap V(H)$ and $t \in O \cap V(H)$, there exists a path from s to t in H. The graph G is $\underline{\text{maximally ST-robust}}$ if G is $(k, n-k)$-ST robust for all $0 \leq k \leq n$.*

In particular, Blocki et al. [10] prove the existence of a family of maximally ST-robust graphs with size linear with respect to the size of the input and output sets.

Theorem 4 (of [5]). *For all $N > 0$, there exist maximally ST-Robust graphs on N inputs and N outputs on $O(N)$ nodes and constant indegree.*

Intuitively, suppose that when the challenge $r(L_i)$ is revealed we had pebbles on nodes S. By ST-robustness there exists a subset of $|A| \geq N - |S|$ input nodes and $|B| \geq N - |S|$ output nodes such that every $a \in A$ and $b \in B$ there is a directed path from a to b which avoids the set S entirely. In particular, this means that if the challenge $r(L_i) \in B$ is in the set B (which happens with probability at least $|B|/N \geq 1 - |S|/N$) then we will need to repebble $\underline{\text{every}}$ node in the set A before we can pebble node $r(L_i)$.

Lastly, we define a function overlay, shown in Figure 1, which we use to combine graphs as part of our construction. Intuitively, we overlay a depth-robust graph on top of the inputs of our ST-robust graph to ensure that, unless $|S|$ is sufficiently large, it will be expensive to repebble the entire set A above.

Definition 7 (Overlay). *Let $G = (V = [n], E)$ and $G' = (V' = [m], E')$ for $m > 2n$ with sources $[n]$ and sinks $[m - n + 1 : m]$. Then $\mathsf{overlay}(G, G') = (V', E \cup E')$.*

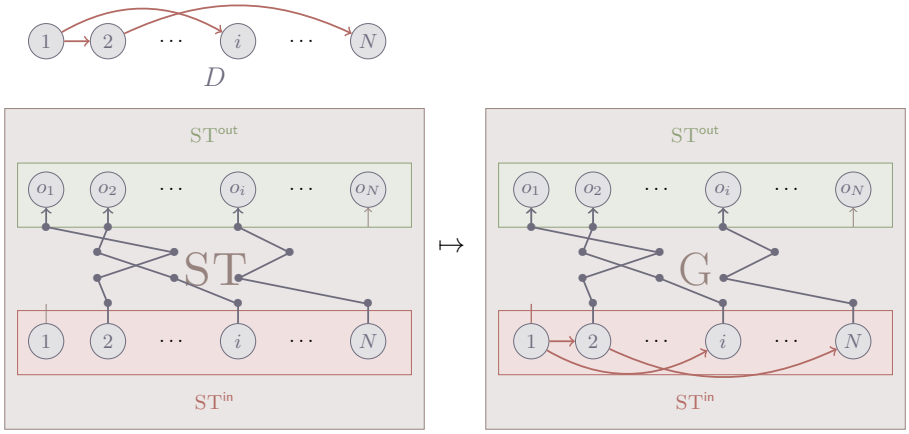

Fig. 1. Above is the visualization of the mapping overlay applied to D and ST to get the graph G. The result is an ST-robust graph with a highly depth robust input set. The red edges from $\mathrm{ST}^{\mathsf{in}}$ to $\mathrm{ST}^{\mathsf{out}}$ represent the high connectivity between the sets due to the maximal ST-robustness.

Definition 8 (The Dynamic Graph $\mathbb{G}_D^N$). *Let D be an (e_N, d_N)-depth robust graph on N nodes and ST be a maximally ST-robust graph with N inputs ST^{in} and N outputs ST^{out}. Next let $G = \mathsf{overlay}(D, ST)$ on M nodes. Let $L = \langle M + 1, \ldots, M + N \rangle$ and $G' = (V, E)$ such that $V = V(G) \cup L$ and $E = E(G) \cup \{(M + i - 1, M + i) \mid i \in [1 : N]\}$. Finally, let $\mathbb{G}_D^N$ be the distribution over the set of all G' with additional edges $\{(r(L_i), L_i) \mid i \in [N]\}$, where $r(L_i)$ maps L_i to some $j \in ST^{\mathsf{out}}$ chosen uniformly at random.*

For each P_i, let ST_{P_i} denote a subgraph of $G - L - P_i$ with paths from at least $N - |P_i|$ inputs $\mathrm{ST}_{P_i}^{\mathsf{in}}$ to at least $N - |P_i|$ outputs $\mathrm{ST}_{P_i}^{\mathsf{out}}$. Intuitively, if a strategy keeps a small number of pebbles on the graph for a large number of steps, then, upon the discovery of a dynamic edge, a large amount of inputs will likely have to be repebbled, which is expensive due to its depth robustness.

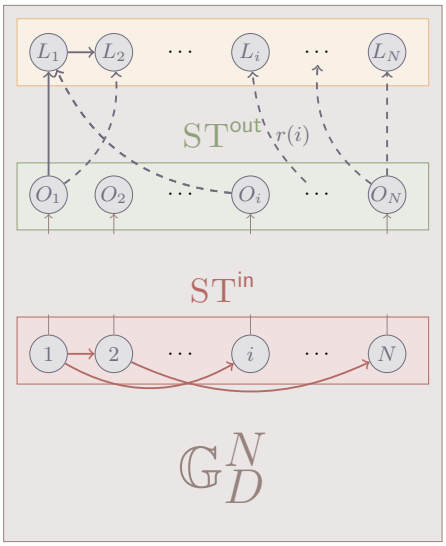

Fig. 2. Above is the final construction of $\mathbb{G}_D^N$, which combines $\mathsf{overlay}(D, \mathrm{ST})$ with a line graph on nodes L. For each L_i there is a dynamic edge from $r(L_i) \in_R \mathrm{ST}^{\mathsf{out}}$.

3.2 Lowerbounding Costly Edges

The first step in describing the trade-off between sustained space and cumulative complexity of $\mathbb{G}_D^N$ is describing how often a low-memory pebbling encounters a costly edge, one that requires a large amount of repebbling. Even if a pebbling keeps only a small number of pebbles on the graph, it's possible that it gets "lucky" and avoids costly edges. In this section we show that a pebbling can't get lucky many times, except with negligible probability.

Fix some parameters $0 < \ell < 1-p < 1$. Let $\mathsf{unlucky}_1, \ldots, \mathsf{unlucky}_N$ be random variables such that $\mathsf{unlucky}_i = 1$ if $|P(L_i)| > pN$ or $r(L_i) \in \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$ i.e., we have at most pN pebbles on the graph when the challenge edge $r(L_i)$ is revealed or we will need to repebble the entire set $\mathrm{ST}^{\mathsf{in}}_{P(L_i)}$ since $r(L_i) \in \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$. Let $\mathsf{unlucky} = \sum_{i \in [N]} \mathsf{unlucky}_i$. Getting "lucky" during a round in which some $r(L_i)$ is discovered refers to the event that there are a small amount of pebbles on the graph, yet $r(L_i) \notin \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$ and isn't guaranteed to be costly. Intuitively, the challenge $r(L_i)$ is guaranteed to be costly if it happens to be in $\mathrm{ST}^{\mathsf{out}}_{P(L_i)}$. The exact penalty for being unlucky will be described later. Here, we find that the probability of getting lucky is simply upperbounded by p, since there are at most pN nodes of $\mathrm{ST}^{\mathsf{out}}$ that are not in $\mathrm{ST}^{\mathsf{in}}_{P(L_i)}$.

Lemma 2. *Let $\mathcal{S}$ be any strategy and let $P = \langle P_1, \ldots, P_T \rangle = S(G)$, where $G \sim \mathbb{G}_D^N$. Then for any fixed $b_1, \ldots, b_{i-1} \in \{0,1\}$ we have*

$$q \coloneqq \Pr\left[\mathsf{unlucky}_i \,\middle|\, \bigwedge_{j \in [i-1]} \mathsf{unlucky}_j = b_j\right] \geq 1 - p,$$

where the probability is taken over the selection of $G \sim \mathbb{G}_D^N$.

Proof. If $|P(L_i)| > pN$, then $q = \mathsf{unlucky}_i = 1$. If $|P(L_i)| \leq pN$. Then regardless of any prior pebbling steps, we have that there are at most pN nodes in $\mathrm{ST}^{\mathsf{out}} \setminus \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$ by construction. Since $r(L_i)$ is chosen uniformly at random, it follows by Theorem 2 that $r(L_i) \notin \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$ with probability at most

$$\frac{\left|\mathrm{ST}^{\mathsf{out}} \setminus \mathrm{ST}^{\mathsf{out}}_{P(L_i)}\right|}{N} \leq p,$$

so $q \geq 1 - p$. □

Ultimately, our goal is to show that, with overwhelming probability, any low-memory pebbling gets unlucky so often that it incurs an unreasonable time cost, because each time the pebbling gets unlucky it incurs a high cost while repebbling $r(L_i)$. So, we must show that it's very unlikely that such pebblings get unlucky only a relatively few amount of times. From Lemmas 1 and 2, Lemma 3 immediately follows, and so the proof is left to the full version of this paper.

Lemma 3. *Let* $\mathcal{S}$ *be any pebbling strategy, and let* $P = S(G)$ *for* $G \sim \mathbb{G}_D^N$. *Then for all* $\epsilon > 0$ $\Pr\left[\sum_{i \in [N]} \mathsf{unlucky}_i < N(1 - p - \epsilon)\right] \leq \exp(-2\epsilon^2 N)$

3.3 The Trade-off between Sustained Space and Cumulative Complexity

We now argue that whenever $\mathsf{unlucky}_i = 1$ and we have at most $|P(L_i)| \leq pN$ pebbles on the graph when the challenge $r(L_i)$ is revealed that the cumulative pebbling cost incurred between rounds $s(L_i)$ (when the challenge $r(L_i)$ is revealed) and $t(L_i)$ (when we place a pebble on node $r(L_i)$) is at least $\mathsf{cc}(P, s(L_i), t(L_i)) \geq pNd$. We conclude with the proof of our main result from this section.

If few pebbles are on the graph at step $s(L_i)$, then the pebbling can get lucky in the sense that $r(L_i)$ isn't expensive to pebble; otherwise, the configuration necessitates some costly pebbling moves from step $s(L_i)$ to step $t(L_i)$. More concretely, if $\mathsf{unlucky}_i = 1$ and $|P(L_i)| \leq pN$ then $r(L_i)$ is in $\mathrm{ST}^{\mathsf{out}}_{P(i)}$ and pebbling $r(L_i)$ requires pebbling at least $N(1-p)$ nodes of $\mathrm{ST}^{\mathsf{in}}_{P(i)}$, which is $(e - pN, d)$-depth robust by Remark 1.

Lemma 4. *For any pebbling strategy* $\mathcal{S}$ *and* $P = S(G)$ *for* $G \sim \mathbb{G}_D^N$. *If* $\mathsf{unlucky}_i = 1$, $|P(L_i)| \leq pN$, *and* D *is* $(2pN, d)$*-depth robust, then* $\mathsf{cc}(P, s(L_i), t(L_i)) \geq pNd$. *We call such* $(r(L_i), L_i)$ *costly edges.*

Proof. If $\mathsf{unlucky}_i = 1$, and $|P(L_i)| \leq pN$ then $r(L_i) \in \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$. Since $\mathrm{ST}_{P(L_i)}$ is maximally ST-robust, there are paths from at least $N(1-p)$ inputs to $r(L_i)$. That is, $\mathrm{ST}^{\mathsf{in}}_{P(L_i)}$ must be pebbled by round $t(L_i)$. Since $\mathrm{ST}^{\mathsf{in}}$ is $(2pN, d)$-depth robust, $\mathrm{ST}^{\mathsf{in}}_{P(L_i)}$ is (pN, d)-depth robust by Remark 2. It follows that

$$\mathsf{cc}(P, s(L_i), t(L_i)) \geq \mathsf{cc}\left(\mathrm{ST}^{\mathsf{in}}_{P(L_i)}\right) \geq pNd.$$

□

Now we have the tools to prove Theorem 3, which is a straight-forward consequence of Lemmas 3 and 4.

Proof of Theorem 3. If P is not (p, ℓ)-low memory, then $\mathsf{ss}(P, pN) > \ell N$. Suppose P is (p, ℓ)-low memory. By Lemma 3, except with probability at most $e^{-2(1-p-c_1)^2 N}$ (for $\ell < c_1 < 1 - p$), there are at least $c_1 N$ pebbling moves in which either $|P(L_i)| > pN$ or $r(L_i) \in \mathrm{ST}^{\mathsf{out}}_{P(L_i)}$, so for $c_2 = c_1 - \ell$, there are nodes $i_1, \ldots i_{c_2 N}$ in which $\mathsf{unlucky}_{i_j} = 1$ and $|P(L_{i_j})| \leq pN$. It follows by Lemma 4 that $\mathsf{cc}(P) \geq \sum_{j \in [c_2 N]} \mathsf{cc}\left(P, s(L_{i_j}), t(L_{i_j})\right) \geq c_2 p N^2 d$. □

3.4 An Instance with High Trade-Off

First we give a construction of $\mathbb{G}^N_D$, which requires the family of graphs $\mathsf{Grates}_{N,\epsilon}$ for $\epsilon > 0$. The graph $\mathsf{Grates}_{N,\epsilon}$ is ideal for this construction, as it is $(\Omega(N), \Omega(N^{1-\epsilon})$-depth robust.

Definition 9 (Grates [20]). *For all $\epsilon > 0$, there exist constants $\gamma, c > 0$, depending only on ϵ such that the graph $\mathsf{Grates}_{N,\epsilon}$ on N nodes is $(\gamma N, cN^{1-\epsilon})$-depth robust and has constant indegree.*

Now, Theorem 3 shows that that $\mathbb{G}^N_D$ with $D = \mathsf{Grates}_{N,\epsilon}$ gives us the best known SSC-CC trade-off for constant indegree dynamic graphs, resulting in Corollary 2.

We remark it is better to instantiate $\mathbb{G}^N_D$ with $D = \mathsf{Grates}_{N,\epsilon}$ instead of another depth-robust graph like DRSample [4] is $(\Omega(N/\log N), \Omega(N))$-depth robust. This is an interesting observation because if we considered these DAGs as a standalone iMHF then we would prefer to use DRSample. In particular, DRSample has $\mathsf{cc} = \Omega(N^2/\log N)$ in comparison to $\mathsf{cc} = \Omega(N^{2-\epsilon})$ for grates. The reason why DRSample is not suitable is that we do not have any guarantees that the graph is (e, d)-depth robust when $e = \Omega(N)$.

4 Dynamic EGS

The next graph family we hybridize is a construction by Erdős et al., which we will call EGS. EGS achieves the maximum possible cumulative complexity of $\Omega(N^2)$ [14]. While EGS achieves the highest possible cumulative complexity, it is the least practical of the graphs we're considering, as it has indegree $\Omega(\log N)$ [14]. In this section we construct a simple, dynamic version of this graph and show that it achieves the maximal sustained space and cumulative memory trade-off. The precise details of this construction are unnecessary, as we rely only on the fact that the graph satisfies the properties of local expansion.

Definition 10 (Local Expansion [6]). *Let I_r and I^*_r be defined such that $I_r(x) = \{x - r - 1, \ldots, x\}$ and $I^*_r(x) = \{x + 1, \ldots, x + r\}$. We say that a DAG $G = (V = [N], E)$ is a δ-local expander if for all $i \in V$, $r \leq i, N - i$, and $A \subseteq I^*_r(x)$ and $B \subseteq I_r(x)$ each of size at least δr, there exists an edge from A to B.*

Local expansion naturally gives guarantees on connectivity and depth-robustness after arbitrary node deletion and is a key property of EGS.

Theorem 5 (of [6, 14]). *For any $0 < \delta < 1$, there exists a family of graphs $\{\mathsf{EGS}_N^\delta\}_{N=1}^\infty$ such that EGS_N^δ is a δ-local expander on N nodes. For some constants $c, \eta, \eta' > 0$, depending only on δ, each EGS_N^δ has indegree $c \log N$ and is $(\eta N, \eta' N)$-depth robust. Furthermore, for each $i \in [N]$, $\mathsf{EGS}_N^\delta([i])$ is $(\eta i, \eta' i)$-depth robust.*

In constructing a hybrid extension of EGS, we want to add dynamic edges that require the adversary to repebble many nodes. For each node, we simply select an incoming edge from a prior node chosen uniformly at random. We'll show that if an adversary doesn't keep sufficiently many pebbles on the graph, then it will have to repebble maximally depth-robust subgraphs of EGS many times.

Definition 11 (Dynamic EGS). *The dynamic pebbling graph DEGS_N^δ is the graph EGS_N^δ with additional dynamic edges $\{(r(i), i) \mid i \in [3 : N]\}$ with $r(i) \in_R [i-2]$.*

We'll show that with overwhelming probability, any dynamic pebbling strategy either maintains pN nodes on the graph for more than ℓN steps, or has cumulative complexity $\Omega(N^3)$.

Theorem 6. *There exist constants $0 < c, c', c_1, \rho, p, \ell < 1$ such that for any strategy $\mathcal{S}$, except with probability at most $\exp(-2\left(\rho - \frac{c}{1-c_1}\right)^2 (1-c_1)N)$, we either have $\mathsf{ss}(\mathcal{S}(G), pN) > \ell N$ or $\mathsf{cc}(\mathcal{S}(G)) \geq c' N^3$ where the probability is taken over the selection of $G \sim \mathsf{DEGS}_N^\delta$.*

The last tool we'll use to prove Theorem 6 are good nodes. If a pebbling strategy has pebbles S on a graph, then it's useful to know whether a given node is surrounded by only relatively few pebbles, since that way the node is more likely to be a part of a long path.

Definition 12 (Good Nodes [4]). *Let $\gamma > 0$, $G = ([N], E)$ be a DAG, and $S \subseteq V$. The node $i \in [N]$ is γ-good with respect to S if (1) for all $r \in [i]$ $|I_r(i) \cap S| \leq \gamma r$ and (2) for all $r \in [m - i + 1]$ $|I_r^*(i) \cap S| \leq \gamma r$.*

We first show that if a strategy keeps some sufficiently small amount of pebbles on the graph, then there will be large paths in the remaining graph. The good nodes form such paths.

Lemma 5 (Lemma 5 of [6]). *Let $G = ([N], E)$ be a δ-local expander and $x < y \in [N]$. For any $S \subseteq [N]$ and γ such that $\delta < \min\{\gamma/2, 1/4\}$, the graph $G - S$ contains a directed path through all nodes in G which are γ-good with respect to S.*

Prior work gave a lowerbound on the number good nodes in arbitrary DAGs with respect to an arbitrarily-sized subset of nodes. This immediately allows us to lowerbound the probability that $r(i)$ is a good node.

Lemma 6 (Lemma 6 of [6]). *For any DAG $G = ([N], E)$, $\gamma > 0$, and $S \subseteq [N]$, there are at least $N - \frac{1+\gamma}{1-\gamma}|S|$ nodes in G which are γ-good with respect to S.*

4.1 Lowerbounds on Getting Unlucky

We define events that are costly for strategies that employ low-memory pebblings and show that they happen with reasonably large probability. First, we must characterize events that may lead to high cumulative cost if an adversary has relatively few pebbles on the graph upon discovering $r(i)$. We know by Lemma 5, there's a good chance that $r(i)$ is a γ-good node. If that's the case, then there's a long path that includes $r(i)$. Eventually, we show that if $r(i)$ is good and sufficiently large, then the subgraph of good nodes prior to $r(i)$ is highly depth robust and must be repebbled.

To quantify what it means for i and $r(i)$ to be "large" enough that pebbling the subgraph of good nodes is sufficiently costly, we require the assignment of several constants with various constraints in agreement with Lemma 7.

Lemma 7. *Fix any $0 < \eta < 1$ according to Theorem 5. There exists an assignment of p, ℓ, c_1, c_2, and c_3 such that for all $0 < \gamma < 1$,*

1. $1 - \eta < c_3 < c_2 = c_1\left(1 - p\frac{1+\gamma}{1-\gamma}\right)$,
2. $0 < \ell < c_2(c_2 - c_3)(1 - c_1)$, *and*
3. $0 < c_2 < c_1 < 1 - \ell$.

Proof. To satisfy $1 - \eta < c_2 < 1$, we first pick $0 < p < \frac{\eta(1-\gamma)}{1+\gamma}$. Since $c_2 = c_1(1 - p\frac{1-\gamma}{1+\gamma})$, it follows that $c_2 < c_1$. Fix ℓ such that $0 < \ell < c_2(c_2 - c_3)(1 - c_1)$. Then (3) is satisfied by the fact that $0 < c_1, c_2, c_3 < 1$. Finally, fix any c_3 such that $1 - \eta < c_3 < c_2$. □

Let $0 < \delta < 1$, assign $0 < \gamma < 1$ satisfying Lemma 5, and fix p, ℓ, c_1, c_2, and c_3 according to Lemma 7. Let $\mathcal{S}$ be any pebbling strategy for $G \sim \mathsf{DEGS}_N^\delta$ and $P = \mathcal{S}(P)$. Next we define an indicator random variable for the whether or not $r(i)$ is good. Let good_i be the random variable such that $\mathsf{good}_i = 1$ if $r(i)$ is γ-good with respect to $P(i)$ and $\mathsf{good}_i = 0$ otherwise. The function rank_i determines how far $r(i)$ is along the path of good nodes. The higher the value of $\mathsf{rank}_i(r(i))$, the more expensive it will be to pebble $r(i)$. More formally, $\mathsf{rank}_i(v) = j$ if v is topologically the jth γ-good node respect to $P(i)$, and $\mathsf{rank}_i(v) = 0$ otherwise.

Finally we say that an adversary is unlucky at a step $s(i)$ if $r(i)$ is good and sufficiently far along the path of good nodes. For $i \geq c_1 N + 2$, let $\mathsf{unlucky}_i$ be the random variable such that $\mathsf{unlucky}_i = 0$ if $P(i) \leq pN$ and either $\mathsf{good}_i = 0$ or $\mathsf{rank}_i(v) < c_3 N$ and $\mathsf{unlucky}_i = 0$ otherwise. Let $\mathsf{unlucky} = \sum_{i \in [c_1 N + 2, N]} \mathsf{unlucky}_i$.

Intuitively, an adversary that has few pebbles on the graph at step $s(i)$ is unlucky if $r(i)$ is a γ-good node with respect to $P(i)$ and of large depth. We show that any strategy gets unlucky at step $s(i)$ with some constant probability.

Lemma 8. *There exists a constant $\rho > 0$ such that for each for each $i \in [c_1 N + 2 : N]$ and $b_{c_1 N+2}, \ldots, b_{i-1} \in \{0, 1\}$.*

$$\Pr\left[\mathsf{unlucky}_i \middle| \bigwedge_{j \in [c_1 N + 2 : i-1]} \mathsf{unlucky}_j = b_j\right] \geq \rho.$$

Proof. If $|P(i)| > pN$ then $\mathsf{unlucky}_i = 1$, so assume $|P(i)| \leq pN$. Then $G([i-2])$ is a δ-local expander on at least $c_1 N$ nodes, and there are at least $c_2 N = c_1 N - c_1 pN\frac{1+\gamma}{1-\gamma}$ γ-good nodes with respect to $P(i)$ in $G([i-2])$ by Theorem 6. So, the probability that $\mathsf{good}_i = 1$ is at least $\frac{c_2 N}{i-2} \geq \frac{c_2 N}{N} = c_2$

For c_3 assigned according to Lemma 7, we have

$$\Pr[\mathsf{rank}_i(r(i)) \geq c_3 N \mid \mathsf{good}_i] \geq c_2 - c_3,$$

then by conditional probability $\Pr[\mathsf{rank}_i(r(i)) \geq c_3 N] \geq c_2(c_2 - c_3)$. Then for $\rho = c_2(c_2 - c_3)$, $\Pr\left[\mathsf{unlucky}_i \middle| \bigwedge_{j \in [c_1 N + 2 : i-1]} \mathsf{unlucky}_j = b_j\right] \geq \rho$. □

Just as before, combining Lemmas 1 and 8 immediately implies Lemma 9.

Lemma 9. *For some constant $c > 0$,*

$$\Pr[\mathsf{unlucky} < cN] \leq \exp\left(-2\left(\rho - \frac{c}{1-c_1}\right)^2 (1-c_1)N\right).$$

4.2 The Cost of Getting Unlucky

Next we examine the cost associated with $\mathsf{unlucky}_i$. Theorem 5 implies that being unlucky results in high cumulative cost from step $s(i)$ to step $t(i)$.

Lemma 10. *If $\mathsf{unlucky}_i = 1$ and $|P(i)| < pN$ then $\mathsf{cc}(P, s(i), t(i)) \geq c_5 N^2$ for some constant $c_5 > 0$.*

Proof. If $\mathsf{unlucky}_i = 1$ and $|P(i)| < pN$, then all of the γ-good nodes of $G([r(i)])$ with respect to $P(i)$ must be repebbled before or on step $t(i)$. It follows by Theorem 5 that this subgraph is $(\eta c_3 N, \eta' c_3 N)$-depth robust for some constants $\eta, \eta' > 0$. Since by Lemma 7 $c_3 > 1 - \eta$, we can apply Theorem 2 to get

$$\begin{aligned}\mathsf{cc}(i, s(i), t(i)) &\geq (c_3 + \eta - 1)\eta' c_3 N^2 \\ &\geq (c_3 + \eta - 1)\eta' c_3 N^2 \\ &= c_5 N^2,\end{aligned}$$

for $c_5 = (c_3 + \eta - 1)\eta' c_3$ □

As with Theorem 3, Theorem 6 directly follows from Lemma 10. This is because Lemma 9 implies that, except with negligible probability, there are $\Omega(N)$ steps in which $\mathsf{unlucky}_i = 1$ and $|P(i)| \leq pN$. Then Lemma 10 implies that such a strategy incurs CC $\Omega(N^2)$ for each of these incidences, resulting in a total CC of $\Omega(N^3)$.

5 Dynamic DRSample

DRSample is a randomized algorithm that, except with negligible probability, outputs an $\left(\Omega\left(\frac{N}{\log N}\right), \Omega(N)\right)$-depth robust graph on N nodes [4]. While losing a $\log N$ factor in depth robustness, output graphs of DRSample contrast EGS by having indegree 2 and being practical for common applications of MHFs. To prove this section's main result, Theorem 7, we use a stronger version of depth-robustness, where we are guaranteed sufficiently long paths even after the deletion of blocks of consecutive nodes.

Definition 13 (Block-Depth Robust [4]). *Let $N \in \mathbb{N}$ and $G = (V = [n], E)$ be a DAG. For a node v, let $N(v, b) = \{v - b + 1, \ldots, v\}$, and for $S \subseteq V$, let $N(S, b) = \bigcup_{v \in S} N(v, b)$. The graph G is (e, d, b)-block-depth robust if for every set $S \subseteq V$ of size at most e, there exists a path of length d in $G - N(S, b)$.*

We also use a more general form of local expansion, which implies high connectivity between the nodes after node deletion.

Definition 14 (Local Expansion Node [4]). *For a graph $G = (V = [N], E)$, $c > 0$ and $r^* \in \mathbb{Z}^+$, we say that a node $v \in V$ is a (c, r^*)-local expander if for all $r \geq r^*$ we have*

- *for all $A \subset I_v^*(r)$ and $B \subseteq I_{v+r}^*(r)$ of size $|A|, |B| \geq cr$ there exists an edge from A to B, and*
- *for all subsets $A \subseteq I_v(r)$ and $B \subseteq I_{v-r}^*(r)$ of size $|A|, |B| \geq cr$, there exists an edge from A to B.*

In our analysis of Dynamic DRSample, we will often examine its metagraph. The metagraph of a DAG $G = ([N], E)$ with parameter m simply maps each block $[mi + 1 : m(i + 1)]$ to a node. Two nodes of the metagraph u and v are connected if in the original graph there's an edge from the "last part" of the u block to the "first part" of the v block.

Definition 15 (Metagraph [4]). *For a graph $G = ([N], E)$ and $m > 0$, we define the metagraph $G_m = (V_m, E_m)$ as follows. Let $N' = \lfloor N/m \rfloor$ and $V_m = [N']$. Let*

- $M_i = [(i - 1)m + 1 : im]$,
- $M_i^F = \left[(i - 1)m + 1 : (i - 1)m + \left\lfloor m\frac{1-1/10}{2} \right\rfloor\right]$, *and*
- $M_i^L = \left[(i - 1)m + 1 + \left\lceil m\frac{1+1/10}{2} \right\rceil : im\right]$.

Then $E_m = \{(i, j) \mid M_i^L \times M_j^F \cap E \neq \emptyset\}$.

There is a natural correspondence between the depth-robustness and block-depth robustness of graphs and metagraphs.

Remark 2 (Claim 1 of [4]). Let G be a DAG. If G_m is (e, d)-depth robust, then G is $(e/2, md/10, m)$-block depth robust.

DRSample has a number of tunable parameters. We exclusively refer to DRSample with the recommended parameters from [4].

Definition 16 (DRSample [4]). *The randomized algorithm* $\mathsf{DRSample}$ *on input* N *outputs a graph* $\mathsf{DR} = (V, E)$ *on* N *nodes with the following properties. Fix any* $0 < p, \epsilon < 1$ *and let*

- $a = 160$
- $m = a \log N$,
- $N' = \lfloor N/m \rfloor$
- $\gamma = 0.1$,
- $\sigma = 0.125$
- $x = 0.00861$,
- $\alpha = 0.2916$,
- $r^* = 8$,
- $c_{10} = 1 - \frac{2p}{\sigma} - x - \epsilon$,
- $\eta = 0.038945$, *and*
- $\eta' = 0.3$.

Except with negligible probability $\mu(N)$*, for any subset of metanodes* S *of size at most* pN'*,* $\mathsf{DR}_m - S$ *contains at least* $c_{10}N'$ (α, r^*)*-local expanders that are* γ*-good with respect to* S*. Each of the these nodes are connected, and the metagraph* DR_m *is* $(\eta N', \eta' N')$*-depth robust.*

Next we hybridize DRSample by adding dynamic edges for each node. Here, we make the block parameter m inherent in the construction, as each node has a random edge to the "end" of a random metanode. For the ease of notation, we let $\mathsf{FromMeta}$ be the function mapping metanodes to nodes in the original graph, meaning $\mathsf{FromMeta}(i) = (i-1)m + 1$. Likewise, for $v \in V$, we let $\mathsf{ToMeta}(v) = \lfloor (v-1)/m \rfloor + 1$.

Definition 17 (Dynamic DRSample). *Dynamic DRSample is the dynamic pebbling graph* DDR_N^m*, constructed as follows. Let* $\mathsf{DR} = (V, E) \leftarrow \mathsf{DRSample}(N)$*. Then* DDR_N^m *is* DR *with additional dynamic edges* $\{(r(i), i) \mid i \in [\mathsf{FromMeta}(3) : N]\}$*, where* $r(i)$ *is chosen from* $\{km \mid 1 \leq k \leq \mathsf{ToMeta}(i) - 2\}$ *uniformly at random. That is, for each node* $i \geq 2m + 1$ *of* DDR_N^m*, there's a dynamic edge to* i *from the end of a random metanode.*

The main result of this section is that any pebbling strategy, except with negligible probability, either sustains $p\frac{N}{a \log N}$ pebbles for ℓN steps, or has cumulative complexity $\Omega\left(\frac{N^3}{\log N}\right)$.

Theorem 7. *There exists constants* $0 < p, \ell, a, c_{13}, c_{14} < 1$ *and negligible function* μ*, such that for* $m = a \log N$*, any strategy* $\mathcal{S}$*, except with probability at most* $\mu(N)$*, we have either* $\mathsf{ss}\left(S(G), \frac{pN}{a \log N}\right) > \ell N$ *or* $\mathsf{cc}(S(G)) \geq \frac{c_{14} N^3}{\log N}$*, where the probability is taken over the selection of* $G \sim \mathsf{DDR}_N^m$*.*

Before we begin proving Theorem 7, we need to setup some useful variables and notation. Let $G \sim \mathsf{DDR}_N^m$, $\mathcal{S}$ be any strategy, and $P = \mathcal{S}(G)$, and assign a, m, N', γ, α, r^*, c_{10}, η, and η' according to Definition 16. Let $r_m(i) = \mathsf{ToMeta}(r(i))$ and $P_m(i) = \mathsf{ToMeta}(P(i))$. When i and P are known, we say that v is a good expander when v is γ-good with respect to $P_m(i)$ in G_m and is a (δ, r^*)-local expander in G_m. We'll heavily use fact from Definition 16 that all good expanders are connected.

We'll want to show that for some chosen $0 < c_{11} < 1$ and $i \geq c_{11}N + 2$, if $|P_m(i)| \leq pN'$ then the subgraph induced by the good expanders less than $i-1$ is still $\left(\Omega\left(\frac{N}{\log N}, N\right), \Omega(N)\right)$-depth robust. While there are $c_{10}N'$ good expanders in $G_m - P_m(i)$, there could be as many as $(1-c_{11})N'$ good expanders that have never been pebbled (and thus are not candidates for $r_m(i)$). So, we need to show that c_{10} can take values greater than $1 - \eta + (1 - c_{11})$, yet still be less than 1. Recall from Definition 16 that c_{10} is an implicit function of p, so we can only achieve this by assigning p and c_{11} the appropriate values. Namely, we need

$$\eta - (1 - c_{10}) - (1 - c_{11}) > 0. \tag{2}$$

It suffices for $c_{11} = 0.97$, $p = 2 \times 10^{-5}$, and $c_{10} = 0.99106$.

Until the proof of Theorem 7, we assume that G is $(\eta N', \eta' N')$-depth robust, and for any set S of size at most pN', $G - S$ contains at least $c_{10}N'$ good expanders.

5.1 Lowerbounds on Getting Unlucky

We want to determine the number of times an adversary could be "unlucky." For a step to be unlucky, we need it to be sufficiently large, so that it may be costly to rectify. Specifically, we want the metanode corresponding to this step to be at least $c_{11}N' + 2$. Moreover, if $|P(i)| \leq pN'$, we say the adversary is unlucky if $r(i)$ is a good expander and large. As before, let $\mathsf{rank}_i(v) = j$ when v is topologically the jth good expander in G.

These trials consist of the nodes starting from $\mathsf{FromMeta}(c_{11}N' + 2)$ to N. There are $K \geq N - \mathsf{FromMeta}(c_{11}N' + 2) \geq N(1 - c_{11}) + m - 2$ such nodes. We'll assume that $N > 200$ and fix $\kappa = 1 - c_{11} - \frac{1}{100}$ so that $0 < \kappa N \leq K$. For $i \in [(1-\kappa)N : N]$, we define the random variable $\mathsf{unlucky}_i$ such that

$$\mathsf{unlucky}_i = \begin{cases} 0 & \text{if } |P(i)| \leq p, \text{ but either } r_m(i) \text{ isn't a good expander} \\ & \text{or } \mathsf{rank}_i(r_m(i)) < c_{12}N', \text{ and} \\ 1 & \text{otherwise,} \end{cases}$$

for some constant c_{12} such that

$$1 - \eta < c_{12} < c_{10} - (1 - c_{11}). \tag{3}$$

See that c_{12} can take such values since c_{10} and c_{11} satisfy Equation 2. Then when we take out all nodes but the $c_{12}N'$ good expanders that must be repebbled, G will still be adequately depth-robust since the $c_{12}N'$ good expanders account for almost all nodes of G. Finally, let $\mathsf{unlucky} = \sum_{i \in [(1-\kappa)N:N]} \mathsf{unlucky}_i$. We show that such steps are unlucky with constant probability.

Lemma 11. *For any $i \in [(1-\kappa)N : N]$ and $b_1, \ldots, b_{i-1} \in \{0, 1\}$,*

$$\Pr\left[\mathsf{unlucky}_i \middle| \bigwedge_{j \in [i-1]} \mathsf{unlucky}_j = b_j\right] \geq \rho,$$

for some constant $\rho > 0$.

Proof. If $|P_m(i)| > pN'$ then $\mathsf{unlucky}_i = 1$, so assume otherwise. There are at least $(c_{10} - (1 - c_{11}))N'$ good expanders in $G_m([i-2])$, so the probability that $r_m(i)$ is a good expander out of at most N' total nodes is at least $c_{10} - (1 - c_{11})$.

If $r_m(i)$ is a good expander, then $\mathsf{rank}_i(r_m(i)) \geq c_{12}N'$ with probability at least $1 - c_{12}$, so by conditional probability, $r_m(i)$ is a good expander and the $c_{12}N'$th or higher good expander with probability at least $\rho = (c_{10} - (1 - c_{11}))(1 - c_{12})$. Finally, $\rho > 0$ since $c_{10} > 1 - \eta + (1 - c_{11})$ by Equation 2. □

5.2 The Cost of Being Unlucky

Intuitively, we want to show that a costly node requires high cumulative cost to repebble since all of the good expanders are connected.

Lemma 12. *If* $\mathsf{unlucky}_i = 1$ *and* $|P(i)| < pN'$*, then* $\mathsf{cc}(P, s(i), t(i)) \geq \frac{c_{15}N^2}{\log N}$ *for some* $c_{15} > 0$*.*

Proof. First we have $|P_m(i)| \leq |P(i)| \leq pN'$. Let $i_m = \mathsf{ToMeta}(i) - 2$. If the above assumptions hold, then $i_m \geq c_{11}N'$ and $r_m(i)$ is a good expander with $\mathsf{rank}_i(r_m(i)) \geq c_{12}N'$. Then there are nodes $v_1, \ldots, v_{c_{12}N'}$ which are good expanders and connected in $G_m[i-2] - P_m(i)$. Since c_{12} and c_{11} satisfy Equation 3, $c_{12} > 1 - \eta + (1 - c_{11})$ it follows that the subgraph $G_m(\{v_1, \ldots, v_{c_{12}N'}\})$ is $((\eta + c_{12} - 1)N', \eta'N')$-depth robust. To pebble $r(i)$, all of the nodes that comprise each metanode v_j must be pebbled. By Remarks 1 and 2, $G\left(\bigcup_{j \in [c_{12}N']} I_m^*(\mathsf{FromMeta}(v_j))\right)$ is

$$\left(\frac{(\eta + c_{12} - 1)N'}{2}, c_{11}\eta' N/10, m\right)\text{-block depth robust.}$$

Since this subgraph has no pebbles on it on step $s(i)$ and must be repebbled by step $t(i)$, we have $\mathsf{cc}(P, s(i), t(i)) \geq \frac{c_{15}N^2}{\log(N)}$ for $c_{15} = \frac{\eta + c_{12} - 1}{20} c_{11}\eta'$.

□

The proof of Theorem 7 closely follows the proof of Theorem 3, and a formal proof is deferred to the full version of this paper The main difference has to do with arguments on the metagraph G_m, which is covered by the proof of Lemma 16. More closely, if $\mathsf{ss}\left(\mathcal{S}(G), \frac{pN}{a \log N}\right) \leq \ell N$ then with overwhelming probability there are $c_{13}N$ nodes in which $\mathsf{unlucky}_j = 1$ for $\ell < c_{13} < \kappa\rho$. Then there are nodes $i_1, \ldots, i_{(c_{13} - \ell)N}$ in which $\mathsf{unlucky}_{i_j} = 1$ *and* $|P(i_j)| \leq pN'$. Then by Lemma 16, it follows that $\mathsf{cc}\left(\mathcal{S}(G)\right) = \Omega\left(\frac{N^3}{\log N}\right)$.

6 Argon2id

Argon2id is a hybrid MHF that's currently deployed in several cryptographic libraries, so it is necessary to understand its sustained space guarantees [1].

The first half of the evaluation is data-independent, while the second half is data-dependent. In the pebbling model, this corresponds to the first half of the nodes having fixed edges that are known at the start of Argon2id's evaluation, while the random edges in the second half are dynamic. Intuitively, the data-dependent phase induces high cumulative cost, while the data-independent phase has weaker, yet still significant, cumulative cost to fall back on in the presence of a side-channel attack.

Definition 18 (Argon2id [1]). *The dynamic pebbling graph* $\mathsf{Argon2id}_N$ *consists of the vertex set* $V = [2N]$ *and edge set*

$$E = \{(i, i+1) \mid i \in [2N-1]\} \cup \{(r(i), i) \mid i \in [2N]\},$$

where $r(i)$ *is a random value distributed as follows:*

$$Pr[r(i) = j] = \Pr_{x \in_R [M]}\left[i\left(1 - \frac{x^2}{M^2}\right) \in (j-1, j]\right]$$

for some $M \gg N$*. The edges* $(r(i), i)$ *are only dynamic when* $i > N$*. When* $i \leq N$*,* $(r(i), i)$ *is static and known prior to pebbling.*

In particular, we show the following results.

Theorem 8. *There exists some constants* $\delta, \gamma < 0 < f, u, \ell, \delta', \gamma' < 1$ *such that for any pebbling strategy* $\mathcal{S}$*, with high probability, either* $\mathsf{ss}(\mathcal{S}(G), \delta' \log^{\delta} Ne) > \ell N$ *or* $\mathsf{cc}(\mathcal{S}(G)) \geq \gamma' N^4 e^{-2} \log^{\gamma} N$*, where the probability is taken over the choice of* $G \sim \mathsf{Argon2id}_N$*.*

Corollary 3. *Let* $\mathcal{S}$ *be any strategy and* $G \sim \mathsf{Argon2id}_N$*. Then there exists constants* $\delta, \gamma < 0 < f, u, \ell, \delta', \gamma' < 1$ *such that for all* $\epsilon > 0$ *and with high probability, either* $\mathsf{ss}(S(G), \delta' N^{1-\epsilon} \log^{\delta} N) > \ell N$ *or* $\mathsf{cc}(S(G)) = \gamma' N^{2+2\epsilon} \log^{\gamma} N$*.*

The techniques used to prove Theorem 8 are completely different than the other three trade-off proofs in this paper. We start by arguing if an strategy has e pebbles on the graph on step $s(i)$, then with some reasonably large probability the depth of $r(i)$ is $d = \tilde{\Omega}(N^3/e^3)$. For this argument, we use a new graph property called fractional-depth robustness, which says that if a limited amount of nodes are deleted from the graph, then there are some fraction of nodes still with large depth. From then on, the proof of Theorem 8 uses techniques from the proof that the dynamic pebbling graph $\mathtt{Scrypt}_N$ has CC $\Omega(N^2)$ [8]. Specifically, if $r(i)$ has depth d in $G - P(i)$, then the minimum required steps to pebble $r(i)$ is d. For this to happen, e must have been sufficiently large (otherwise d would necessarily be larger). The argument is repeated for all steps between $s(i-1)+1$ and $s(i)$ to lowerbound its CC.

6.1 The Trade-Off and Cumulative Complexity

We prove the CC penalty for low-memory pebblings using a graph property called fractional depth-robustness.

Definition 19 (Fractional Depth-Robustness [11]). *For a vertex v in a graph G, let $\mathsf{depth}(v, G)$ denote the longest path to v in G. A DAG $G = (V = [N], E)$ (e, d, f)-fractionally depth robust if for all $S \subseteq V$ with $|S| \leq e$, we have $|\{v \mid, v \in V, \mathsf{depth}(v, G) \geq d\}| \geq fN$.*

Next we'll use the following facts about the graph underlying Argon2id.

Lemma 13 (of [11]). *Let $G \sim \mathsf{Argon2id}_N$. There exists $0 < \alpha', f < 1$ and $\alpha \leq 0$ such that, with probability $1 - o\left(\frac{1}{N}\right)$, $G([N])$ is $\left(e, \frac{\alpha' N^3 \log^\alpha N}{e^3}, f\right)$-fractionally depth robust.*

Let $\mathcal{S}$ be a pebbling strategy, $G \sim \mathsf{Argon2id}_N$, and $P = S(G)$. For the ease of notation, $e_i = |P_i|$ and d_i denote the minimum required steps from $s(i)$ to pebble $r(i)$. For now we'll assume from Lemma 13 that G is $\left(e, \frac{\alpha' N^3 \log^\alpha N}{e^3}, f\right)$-fractionally depth-robust for some $0 < \alpha', f < 1$ and $\alpha \leq 0$. Immediately, this says that if $|P(i)| \leq e$ then there are fN nodes of depth at least $\frac{\alpha' N^3 \log^\alpha N}{e^3}$ in $G - P(i)$. By Definition 18 $r(i)$ is not chosen uniformly at random, as the distribution slightly shifts probability mass to nodes closer to i. However, this shift isn't significant enough for our arguments. This is formalized by Lemma 14. This claim is inherent by the work of [11], but we include a proof in the full version of this paper.

Lemma 14 (of [11]). *Let $G \sim \mathsf{Argon2id}_N$, $i > N$, and $j \leq N$. Then* $\Pr[r(i) = j] \geq \frac{1}{8N}$.

Immediately from Lemma 14, we have

$$\Pr\left[d_i \geq \frac{\alpha' N^3 \log^\alpha N}{e_{s(i)}^3}\right] \geq f/8. \tag{4}$$

This is the probability that the adversary, upon discovering $r(i)$ at step $s(i)$, must take at least $\frac{\alpha' N^3 \log^\alpha N}{e_{s(i)}^3}$ steps to pebble $r(i)$. From any $j \leq s(i)$, the minimum required steps to pebble $r(i)$ is at least $s(i) - j + d_i$. Then even if the adversary knew $r(i)$ on step $s(i) - j$, it would have to take at least $d_i + j \geq \frac{\alpha' N^3 \log^\alpha N}{e_{s(i)-j}^3}$ steps with probability at least $f/8$. Intuitively, this is because each $r(i)$ is independent of the strategy employed by $\mathcal{S}$, meaning we can take $r(i)$ to be chosen before the pebbling begins. Then even if $f(i)$ was discovered on step $s(i) - j$, Equation 4 applies. Let $s(i) - h_i$ be a step that maximizes this bound on d_i. Then for all $k \leq s(i)$, $d_i \geq \frac{\alpha' N^3 \log^\alpha N}{e_{s(i)-h_i}^3} - h_i \geq \frac{\alpha' N^3 \log^\alpha N}{e_{s(i)-k}^3} - k$, so $e_{s(i)-k} \geq \frac{\alpha'^{1/3} N \log^{\alpha/3} N}{(d_i+k)^{1/3}}$ by the construction of h_i. For $i \in [N+1 : 2N]$, we define the random variables $\mathsf{hard}_i = 1$ if $d_i \geq \frac{\alpha' N^3 \log^\alpha N}{e_{s(i)-h_i}^3} - h_i$, and $\mathsf{hard}_i = 0$ otherwise. If $\mathsf{hard}_i = 1$, then for all $k \leq s(i)$, $e_{s(i)-k} \geq \frac{\alpha'^{1/3} N \log^{\alpha/3} N}{(d_i+k)^{1/3}}$ by the construction of h_i. This allows us to lowerbound the cumulative cost

associated with steps $s(i-1)+1$ to $s(i)$. Next we define the random variables $\mathsf{unlucky}_i^e = \begin{cases} 1 & \text{if either } e_{s(i)} > e \text{ or both } e_{s(i)} \leq e \text{ and } \mathsf{hard}_i = 1, \text{ and} \\ 0 & \text{otherwise.} \end{cases}$

Lemma 15. *For any* $b_1, \ldots, b_{i-1} \in \{0,1\}$, $\Pr\left[\mathsf{unlucky}_i^e \middle| \bigwedge_{j\in[i-1]} \mathsf{unlucky}_j^e = b_j\right] \geq f/8$.

Proof. If $e_i > e$, then $\mathsf{unlucky}_i^e = 1$, so assume otherwise. By the fractional-depth robustness of Argon2id, even if $r(i)$ was discovered on round $s(i) - h_i$, $r(i)$ has depth at least $\frac{\alpha' N^3 \log^\alpha N}{e^3_{s(i)-h_i} - (s(i)-h_i)}$ with probability at least $f/8$ by Equation 4. □

Next we show that there is high cost associated with being unlucky. This argument closely follows Claim 8 of [8].

Lemma 16. *If* $\mathsf{unlucky}_i^e = \mathsf{unlucky}_j^e = 1$ *and* $|P(i)|, |P(j)| \leq pe$ *for some* $j < i$, *then* $\mathsf{cc}(s(j)+1, s(i)) \geq \beta' N^3 e^{-2} \log^\beta N$ *for some* $0 < \beta' < 1$ *and* $\beta \leq 0$.

Proof. We have

$$\begin{aligned}
\mathsf{cc}(s(j)+1, s(i)) &\geq \mathsf{cc}(s(i) - d_j + 1, s(i)) \\
&= \sum_{k=0}^{d_j - 1} e_{i-k} \\
&= \sum_{k=0}^{d_j - 1} \frac{{\alpha'}^{1/3} N \log^{\alpha/3} N}{(d_i + k)^{1/3}} && \mathsf{hard}_i = 1 \\
&\geq {\alpha'}^{1/3} N \log^{\alpha/3} N \int_{d_i}^{d_i + d_{j_{i-1}}} \frac{1}{x^{1/3}} dx \\
&= 3{\alpha'}^{1/3} N \log^{\alpha/3} N / 2((d_i + d_j)^{2/3} - {d_i}^{2/3}) \\
&\geq \beta' N^3 e^{-2} \log^\beta N && (5)
\end{aligned}$$

for some $0 < \beta' < 1$ and $\beta \geq 0$. Step 5 follows from a simple argument, which is detailed in the full version of this paper. □

Just as with Theorems 6 and 7, Theorem 8 directly follows from Lemma 16, so the proof has been deferred to the full version of this paper. Corollary 3 directly follows.

7 Open Problems

We conclude with several open question for future work. Of course, the most pressing question is whether or not there exists a dynamic pebbling reduction for dMHFs in an idealized model of computation — similar to the pebbling reduction for iMHFs in parallel random oracle model [7]. Such a pebbling reduction would

greatly simplify the design and analysis of future dMHFs. Another interesting direction would be to try to find direct proofs of CMC/SSC trade-offs for one or more of the dMHFs considered in this paper. For example, while [8] used dynamic pebbling to build intuition about the cumulative memory complexity of `Scrypt` the final security proof was direct and did not rely on pebbling arguments. Another natural question is the development of dynamic pebbling attacks. For example, fixing $s = o(N/\log N)$ we could ask what is the minimum `cc` pebbling strategy which is guaranteed to have s-sustained space complexity $o(N)$.

References

[1] Dmitry Khovratovich Alex Biryukov Daniel Dinu. Argon2: the memory-hard function for password and other applications. Tech. rep. 2017.

[2] Joël Alwen and Jeremiah Blocki. "Efficiently Computing Data-Independent Memory-Hard Functions". In: Advances in Cryptology – CRYPTO 2016, Part II. Ed. by Matthew Robshaw and Jonathan Katz. Vol. 9815. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, Aug. 2016, pp. 241–271.

[3] Joël Alwen and Jeremiah Blocki. "Towards Practical Attacks on Argon2i and Balloon Hashing". In: 2017 IEEE European Symposium on Security and Privacy (EuroS P). 2017, pp. 142–157. DOI: 10.1109/EuroSP.2017.47.

[4] Joël Alwen, Jeremiah Blocki, and Ben Harsha. "Practical Graphs for Optimal Side-Channel Resistant Memory-Hard Functions". In: ACM CCS 2017: 24th Conference on Computer and Communications Security. Ed. by Bhavani M. Thuraisingham et al. ACM Press, Oct. 2017, pp. 1001–1017.

[5] Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. "Depth-Robust Graphs and Their Cumulative Memory Complexity". In: Advances in Cryptology – EUROCRYPT 2017, Part III. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10212. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, Apr. 2017, pp. 3–32.

[6] Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak. "Sustained Space Complexity". In: Advances in Cryptology – EUROCRYPT 2018, Part II. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10821. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, Apr. 2018, pp. 99–130.

[7] Joël Alwen and Vladimir Serbinenko. "High Parallel Complexity Graphs and Memory-Hard Functions". In: 47th Annual ACM Symposium on Theory of Computing. Ed. by Rocco A. Servedio and Ronitt Rubinfeld. ACM Press, June 2015, pp. 595–603.

[8] Joël Alwen et al. "Scrypt Is Maximally Memory-Hard". In: Advances in Cryptology – EUROCRYPT 2017, Part III. Ed. by Jean-Sébastien Coron and Jesper Buus Nielsen. Vol. 10212. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, Apr. 2017, pp. 33–62.

[9] Mohammad Hassan Ameri, Jeremiah Blocki, and Samson Zhou. "Computationally Data-Independent Memory Hard Functions". In: ITCS 2020: 11th Innovations in Theoretical Computer Science Conference. Ed. by Thomas Vidick. Vol. 151. LIPIcs, Jan. 2020, 36:1–36:28.

[10] Jeremiah Blocki and Mike Cinkoske. "A New Connection Between Node and Edge Depth Robust Graphs". In: ITCS 2021: 12th Innovations in Theoretical Computer Science Conference. Ed. by James R. Lee. Vol. 185. LIPIcs, Jan. 2021, 64:1–64:18.

[11] Jeremiah Blocki and Samson Zhou. On the Depth-Robustness and Cumulative Pebbling Cost of Argon2i. TCC. `https://ia.cr/2017/442`. 2017.

[12] Jeremiah Blocki and Samson Zhou. "On the Depth-Robustness and Cumulative Pebbling Cost of Argon2i". In: TCC 2017: 15th Theory of Cryptography Conference, Part I. Ed. by Yael Kalai and Leonid Reyzin. Vol. 10677. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, Nov. 2017, pp. 445–465.

[13] Jeremiah Blocki et al. "Data-Independent Memory Hard Functions: New Attacks and Stronger Constructions". In: Advances in Cryptology – CRYPTO 2019, Part II. Ed. by Alexandra Boldyreva and Daniele Micciancio. Vol. 11693. Lecture Notes in Computer Science. Springer, Heidelberg, Germany, Aug. 2019, pp. 573–607.

[14] Paul Erdös, Ronald L Graham, and Endre Szemerédi. "On sparse graphs with dense long paths". In: Computers & Mathematics with Applications 1.3-4 (1975), pp. 365–369.

[15] John Hopcroft, Wolfgang Paul, and Leslie Valiant. "On time versus space". In: Journal of the ACM (JACM) 24.2 (1977), pp. 332–337.

[16] Charles Lee. Litecoin. 2011.

[17] Thomas Lengauer and Robert Endre Tarjan. "Upper and Lower Bounds on Time-Space Tradeoffs". In: Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing. STOC '79. Atlanta, Georgia, USA: Association for Computing Machinery, 1979, pp. 262–277. DOI: `10.1145/800135.804420`.

[18] Password Hashing Competition. `https://www.password-hashing.net/`. 2015.

[19] Colin Percival. "Stronger key derivation via sequential memory-hard functions". In: (Jan. 2009).

[20] Georg Schnitger. "On Depth-Reduction and Grates". In: 24th Annual Symposium on Foundations of Computer Science. IEEE Computer Society Press, Nov. 1983, pp. 323–328.