Stronger Lower Bounds for Online ORAM

Pavel Hubáček(✉), Michal Koucký, Karel Král, and Veronika Slívová

Computer Science Institute of Charles University,
Malostranské náměstí 25, 118 00 Praha 1, Czech Republic
{hubacek,koucky,kralka,slivova}@iuuk.mff.cuni.cz

Abstract. Oblivious RAM (ORAM), introduced in the context of software protection by Goldreich and Ostrovsky [JACM'96], aims at obfuscating the memory access pattern induced by a RAM computation. Ideally, the memory access pattern of an ORAM should be independent of the data being processed. Since the work of Goldreich and Ostrovsky, it was believed that there is an inherent $\Omega(\log n)$ bandwidth overhead in any ORAM working with memory of size n. Larsen and Nielsen [CRYPTO'18] were the first to give a general $\Omega(\log n)$ lower bound for any *online* ORAM, i.e., an ORAM that must process its inputs in an online manner.

In this work, we revisit the lower bound of Larsen and Nielsen, which was proved under the assumption that the adversarial server knows exactly which server accesses correspond to which input operation. We give an $\Omega(\log n)$ lower bound for the bandwidth overhead of any online ORAM even when the adversary has no access to this information. For many known constructions of ORAM this information is provided implicitly as each input operation induces an access sequence of roughly the same length. Thus, they are subject to the lower bound of Larsen and Nielsen. Our results rule out a broader class of constructions and specifically, they imply that obfuscating the boundaries between the input operations does not help in building a more efficient ORAM.

As our main technical contribution and to handle the lack of structure, we study the properties of *access graphs* induced naturally by the memory access pattern of an ORAM computation. We identify a particular graph property that can be efficiently tested and that all access graphs of ORAM computation must satisfy with high probability. This property is reminiscent of the Larsen-Nielsen property but it is substantially less structured; that is, it is more generic.

Keywords: Oblivious RAM · Bandwidth overhead · Lower bound

This research was supported in part by the Grant Agency of the Czech Republic under the grant agreement no. 19-27871X, by the Charles University projects PRIMUS/17/SCI/9 and UNCE/SCI/004, Charles University grant SVV-2017-260452, and by the Neuron Fund for the support of science.

D. Hofheinz and A. Rosen (Eds.): TCC 2019, LNCS 11892, pp. 264–284, 2019.
https://doi.org/10.1007/978-3-030-36033-7_10

1 Introduction

Oblivious simulation of RAM machines, initially studied in the context of software protection by Goldreich and Ostrovsky [11], aims at protecting the memory access pattern induced by computation of a RAM from an eavesdropper. In the present day, such oblivious simulation might be needed when performing a computation in the memory of an untrusted server.[1] Despite using encryption for protecting the content of each memory cell, the memory access pattern might still leak sensitive information. Thus, the memory access pattern should be *oblivious* of the data being processed and, optimally, depend only on the size of the input.

Constructions. The strong guarantee of obliviousness of the memory access pattern comes at the cost of additional overhead. A trivial solution which scans the whole memory for each memory access induces linear *bandwidth overhead*, i.e., the multiplicative factor by which the length of a memory access pattern increases in the oblivious simulation of a RAM with n memory cells. Given its many practical applications, an important research direction is to construct an ORAM with as low overhead as possible. The foundational work of Goldreich and Ostrovsky [11] already gave a construction with bandwidth overhead $O(\log^3(n))$. Subsequent results introduced various improved approaches for building ORAMs (see [1,4–6,9,11–13,17,22,25,26,28,29] and the references therein) leading to the recent construction of Asharov et al. [2] with bandwidth overhead $O(\log n)$ for the most natural setting of parameters.

Lower-Bounds. It was a folklore belief that an $\Omega(\log n)$ bandwidth overhead is inherent based on a lower bound presented already in the initial work of Goldreich and Ostrovsky [11]. However, the Goldreich-Ostrovsky result was recently revisited in the work of Boyle and Naor [3], who pointed out that the lower bound actually holds only in a rather restricted "balls and bins" model where the ORAM is not allowed to read the content of the data cells it processes. In fact, Boyle and Naor showed that any general lower bound for *offline* ORAM (i.e., where each memory access of the ORAM can depend on the whole sequence of operations it needs to obliviously simulate) implies non-trivial lower bounds on sizes of sorting circuits which seem to be out of reach of the known techniques in computational complexity. The connection between offline ORAM lower bounds and circuit lower bounds was extended to *read-only online* ORAMs (i.e., where only the read operations are processed in online manner) by Weiss and Wichs [30] who showed that lower bounds on bandwidth overhead for read-only online ORAMs would imply non-trivial lower bounds for sorting circuits or locally decodable codes.

The first general $\Omega(\log n)$ lower bound for bandwidth overhead in *online* ORAM (i.e., where the ORAM must process sequentially the operations it has to obliviously simulate) was given by Larsen and Nielsen [18]. The core of their lower

[1] Protecting the memory access of a computation is particularly relevant in the light of the recent Spectre [16] and Meltdown [19] attacks.

bound comprised of adapting the *information transfer* technique of Patrascu and Demaine [23], originally used for proving lower bounds for data structures in the cell probe model, to the ORAM setting. In fact, the lower bound of Larsen and Nielsen [18] for ORAM can be cast as a lower bound for the oblivious Array Maintenance problem and it was recently extended to other oblivious data structures by Jacob et al. [15].

1.1 Our Results

In this work, we further develop the information transfer technique of [23] when applied in the context of online ORAMs. We revisit the lower bound of Larsen and Nielsen which was proved under the assumption that the adversarial server knows exactly which server accesses correspond to each input operation. Specifically, we prove a stronger matching lower bound in a relaxed model without any restriction on the format of the access sequence to server memory.

Note that the [18] lower bound does apply to the known constructions of ORAMs where it is possible to implicitly separate the accesses corresponding to individual input operations – since each input operation generates an access sequence of roughly the same length. However, the [18] result does not rule out the possibility of achieving sub-logarithmic overhead in an ORAM which obfuscates the boundaries in the access pattern (e.g. by translating input operations into variable-length memory accesses). We show that obfuscating the boundaries between the input operations does not help in building a more efficient ORAM. In other words, our lower bound justifies the design choice of constructing ORAMs where each input operation is translated to roughly the same number of probes to server memory (common to the known constructions of ORAMs).

Besides online ORAM (i.e., the oblivious Array Maintenance problem), our techniques naturally extend to other oblivious data structures and allow to generalize also the recent lower bounds of Jacob et al. [15] for oblivious stacks, queues, deques, priority queues and search trees.

For online ORAMs with statistical security, our results are stated in the following informal theorem.

Theorem 1 (Informal). *Any statistically secure online ORAM with internal memory of size m has expected bandwidth overhead $\Omega(\log n)$, where $n \geq m^2$ is the length of the sequence of input operations. This result holds even when the adversarial server has no information about boundaries between probes corresponding to different input operations.*

In the computational setting, we consider two definitions of computational security. Our notion of *weak computational security* requires that no polynomial time algorithm can distinguish access sequences corresponding to any two input sequences of the same length – this is closer in spirit to computational security for ORAMs previously considered in the literature. The notion of *strong computational security* requires computational indistinguishability even when the distinguisher is given the two input sequences together with an access sequence corresponding to one of them. The distinguisher should not be able to tell which one of the two input sequences produced the access sequence. Interestingly, our technique (as well as the proof technique of [18] in the model with structured access pattern) yields different lower bounds with respect to the two definitions stated in the following informal theorem.

Theorem 2 (Informal). *Any weakly computationally secure online ORAM with internal memory of size m must have expected bandwidth overhead $\omega(1)$. Any strongly computationally secure online ORAM with internal memory of size m must have expected bandwidth overhead $\Omega(\log n)$, where $n \geq m^2$ is the length of the sequence of input operations. This result holds even when the adversarial server has no information about boundaries between probes corresponding to different input operations.*

Note that even the $\omega(1)$ lower bound for online ORAMs satisfying weak computational security is an interesting result in the light of the work of Boyle and Naor [3]. It follows from [3] that any super-constant lower bound for *offline* ORAM would imply super-linear lower bounds on size of sorting circuits – which would constitute a major breakthrough in computational complexity (for additional discussion, see Sect. 5). Our techniques clearly do not provide lower bounds for offline ORAMs. On the other hand, we believe that proving the $\omega(1)$ lower bound in any meaningful weaker model would amount to proving lower bounds for offline ORAM or read-only online ORAM which would have important implications in computational complexity.

Alternative Definitions of ORAM. Previous works considered various alternative definitions of ORAM. We clarify the ORAM model in which our techniques yield a lower bound in Sect. 2.1 and discuss its relation to other models in Sect. 5. As an additional contribution, we demonstrate an issue with the definition of ORAM appearing in Goldreich and Ostrovsky [11]. Specifically, we show that the definition can be satisfied by a RAM with constant overhead and no meaningful security. The definition of ORAM in Goldreich and Ostrovsky [11] differs from the original definition in Goldreich [10] and Ostrovsky [21], which do not share the issue we observed in the definition from Goldreich and Ostrovsky [11]. Given that the work of Goldreich and Ostrovsky [11] might serve as a primary reference for our community, we explain the issue in Sect. 5 to help preventing the use of the problematic definition in future works.

Persiano and Yeo [24] recently adapted the chronogram technique [8] from the literature on data structure lower bounds to prove a lower bound for *differentially private RAMs* (a relaxation of ORAMs in the spirit of differential privacy [7] which ensures indistinguishability only for input sequences that differ in a single operation). Similarly to the work of Larsen and Nielsen [18], the proof in [24] exploits the fact that the distinguisher knows exactly which server accesses correspond to each input operation. However, as the chronogram technique significantly differs from the information transfer approach, we do not think that our techniques would directly allow to strengthen the [24] lower bound for differentially private RAMs and prove it in the model with an unstructured access pattern.

1.2 Our Techniques

The structure of our proof follows a similar blueprint as the work of Larsen and Nielsen [18]. However, we must handle new issues introduced by the more general adversarial model. Most significantly, our proof cannot rely on any formatting of the access pattern, whereas Larsen and Nielsen leveraged the fact that the access pattern is split into blocks corresponding to each read/write operation. To handle the lack of structure in the access pattern, we study the properties of the *access graph* induced naturally by the access pattern of an ORAM computation. We identify a particular graph property that can be efficiently tested and that all access graphs of ORAM computation must satisfy with high probability. This property is reminiscent of the Larsen-Nielsen property but it is substantially less structured; that is, it is more generic.

The access graph is defined as follows: the vertices are timestamps of server probes and there is an edge connecting two vertices if and only if they correspond to two subsequent accesses to the same memory cell. We define a graph property called *ℓ-dense k-partition*. Roughly speaking, graphs with ℓ-dense k-partitions are graphs which may be partitioned into k disjoint subgraphs, each subgraph having at least ℓ edges. We show that this property has to be satisfied (with high probability) by access graphs induced by an ORAM for any k and an appropriate ℓ. To leverage this inherent structure of access graph towards a lower bound on bandwidth overhead, we prove that if a graph has $\frac{\ell}{k}$-dense k-partition for some ℓ and K different values of k then the graph must have at least $\Omega(\ell \log K)$ edges. In Sect. 3, we provide the formal definition of access graph and ℓ-dense k-partitions and prove a lower bound on the expected number of edges for a graph that has many ℓ-dense k-partitions.

In Sect. 4, we prove that access graphs of ORAMs have many dense partitions. Specifically, using a communication-type argument we show that for $\Omega(n)$ values of k, there exist input sequences for which the corresponding graph has $\Omega(\frac{n}{k})$-dense k-partition with high probability. Applying the indistinguishability of sequences of probes made by ORAM, we get one sequence for which its access graph satisfies $\frac{n}{k}$-dense k-partition for $\Omega(n)$ values of k with high probability. Combining the above results from Sect. 4 with the results from Sect. 3, we get that the graph of such a sequence has $\Omega(n \log n)$ edges, and thus by definition, $\Omega(n \log n)$ vertices in expectation. This implies that the expected number of probes made by the ORAM on any input sequence of length n is $\Omega(n \log n)$.

2 Preliminaries

In this section, we introduce some basic notation and recall some standard definitions and results. Throughout the rest of the paper, we let $[n]$ for $n \in \mathbb{N}$ to denote the set $\{1, 2, \ldots, n\}$. A function $\mathsf{negl}(n) \colon \mathbb{N} \to \mathbb{R}$ is *negligible* if it approaches zero faster than any inverse polynomial.

Definition 1 (Statistical Distance). *For two probability distributions X and Y on a discrete universe S, we define* statistical distance of X and Y *as*

$$\mathrm{SD}(X, Y) = \frac{1}{2} \sum_{s \in S} |\Pr[X = s] - \Pr[Y = s]| \ .$$

We use the following observation, which characterizes statistical distance as the difference of areas under the curve (see Fact 3.1.9 in Vadhan [27]).

Proposition 1. *Let X and Y be probability distributions on a discrete universe S, let $S_X = \{s \in S \colon \Pr[X = s] > \Pr[Y = s]\}$, and define S_Y analogously. Then*

$$\mathrm{SD}(X, Y) = \Pr[X \in S_X] - \Pr[Y \in S_X] = \Pr[Y \in S_Y] - \Pr[X \in S_Y] \ .$$

We also use the following data-processing-type inequality.

Proposition 2. *Let X and Y be probability distributions on a discrete universe S. Then for any function $f \colon S \to \{0,1\}$, it holds that $|\Pr[f(X) = 1] - \Pr[f(Y) = 1]| \le \mathrm{SD}(X, Y)$.*

Definition 2 (Computational indistinguishability). *Two probability ensembles, $\{X_n\}_{n \in \mathbb{N}}$ and $\{Y_n\}_{n \in \mathbb{N}}$, are* computationally indistinguishable *if for every polynomial-time algorithm D there exists a negligible function* $\mathsf{negl}(\cdot)$ *such that*

$$|\Pr[D(X_n, 1^n) = 1] - \Pr[D(Y_n, 1^n) = 1]| \le \mathsf{negl}(n) \ .$$

2.1 Online ORAM

In this section, we present the formal definition for online oblivious RAM (ORAM) we consider in our work – we build on the oblivious cell-probe model of Larsen and Nielsen [18].

Definition 3 (Array Maintenance Problem [18]). *The* Array Maintenance problem with parameters (ℓ, w) *is to maintain an array B of ℓ w-bit entries under the following two operations:*

- *$(W, a,\ d)$: Set the content of $B[a]$ to d, where $a \in [\ell]$, $d \in \{0,1\}^w$.* (Write operation)
- *$(R, a,\ d)$: Return the content of $B[a]$, where $a \in [\ell]$ (note that d is ignored).* (Read operation)

We say that a machine $\mathcal{M}$ implements the Array Maintenance problem with parameters (ℓ, w) and probability p, *if for every input sequence of operations*

$$y = (o_1, a_1, d_1), \ldots, (o_n, a_n, d_n),\ \textit{where each } o_i \in \{R, W\}, a_i \in [\ell], d_i \in \{0,1\}^w ,$$

and for every read operation in the sequence y, $\mathcal{M}$ returns the correct answer with probability at least p.

Definition 4 (Online Oblivious RAM). *For $m, w \in \mathbb{N}$, let RAM*(m, w) denote a probabilistic random access machine $\mathcal{M}$ with m cells of internal memory, each of size w bits, which has access to a data structure, called* server*, implementing the Array Maintenance problem with parameters $(2^w, w)$ and probability 1. In other words, in each step of computation $\mathcal{M}$ may* probe *the server on a triple $(o, a, d) \in \{R, W\} \times [2^w] \times \{0, 1\}^w$ and on every input (R, a, d) the server returns to $\mathcal{M}$ the data last written in $B[a]$. We say that RAM** probes *the server whenever it makes an Array Maintenance operation to the server.*

Let m, M, w be any natural numbers such that $M \leq 2^w$. An online Oblivious RAM $\mathcal{M}$ with address range M, cell size w bits and m cells of internal memory *is a RAM*(m, w) satisfying online access sequence, correctness, and statistical (resp. computational) security as defined below.*

Online Access Sequence: *For any input sequence $y = y_1, \ldots, y_n$ the RAM* machine $\mathcal{M}$ gets y_i one by one, where each $y_i \in \{R, W\} \times [M] \times \{0, 1\}^w$. Upon the receipt of each operation y_i, the machine $\mathcal{M}$ generates a possibly empty sequence of* server probes $(o_1, a_1, d_1), \ldots, (o_{\ell_i}, a_{\ell_i}, d_{\ell_i})$*, where each $(o_i, a_i, d_i) \in \{R, W\} \times [2^w] \times \{0, 1\}^w$, and updates its internal memory state in order to correctly implement the request y_i. We define the* access sequence corresponding to y_i *as $A(\mathcal{M}, y_i) = a_1, a_2, \ldots, a_{\ell_i}$. For the input sequence y, the* access sequence $A(\mathcal{M}, y)$ *is defined as*

$$A(\mathcal{M}, y) = A(\mathcal{M}, y_1), A(\mathcal{M}, y_2), A(\mathcal{M}, y_3), \ldots, A(\mathcal{M}, y_n).$$

Note that the definition of the machine $\mathcal{M}$ is online*, and thus for each input sequence $y = y_1, \ldots, y_n$ and each $i \in [n-1]$, the access sequence $A(\mathcal{M}, y_i)$ does not depend on $y_{i+1}, \ldots, y_n$.*

Correctness: *$\mathcal{M}$ implements the Array Maintenance problem with parameters (M, w) with probability at least $1 - p_{\text{fail}}$.*

Statistical Security: *For any two input sequences y, y' of the same length, the statistical distance of the distributions of access sequences $A(\mathcal{M}, y)$ and $A(\mathcal{M}, y')$ is at most $\frac{1}{4}$.*

Computational Security: *For computational security, we consider infinite families of ORAM where we allow m, M, w to be functions of the length n of the input sequence. We distinguish between the following two notions:*

Weak Computational Security: *For any infinite families of input sequences $\{y_n\}_{n \in \mathbb{N}}$ and $\{y'_n\}_{n \in \mathbb{N}}$ such that $|y_n| = |y'_n| \geq n$ for all $n \in \mathbb{N}$, the probability ensembles $\{A(\mathcal{M}, y_n)\}_{n \in \mathbb{N}}$ and $\{A(\mathcal{M}, y'_n)\}_{n \in \mathbb{N}}$ are computationally indistinguishable.*

Strong Computational Security: *For any infinite families of input sequences $\{y_n\}_{n \in \mathbb{N}}$ and $\{y'_n\}_{n \in \mathbb{N}}$ such that $|y_n| = |y'_n| \geq n$ for all $n \in \mathbb{N}$, the probability ensembles $\{(y_n, y'_n, A(\mathcal{M}, y_n))\}_{n \in \mathbb{N}}$ and $\{(y_n, y'_n, A(\mathcal{M}, y'_n))\}_{n \in \mathbb{N}}$ are computationally indistinguishable.*

The parameters of our ORAM model from Definition 4 are depicted in Fig. 1. We use different sizes of arrows on server and RAM side to denote the asymmetry of the communication (the RAM sends type of operation, address, and data and the server returns requested data in case of a read operation and dummy value in case of a write operation). Note that the input sequence y of ORAM consists of a sequence of all operations, whereas the access sequence $A(\mathcal{M}, y)$ consists of a sequence of addresses of all probes.

Arguably, a user of an ORAM might want the stronger notion of computational security whereas the weaker notion is closer to the past considerations. Note that in the case of weak computational security, the adversarial distinguisher does not have access to the input sequences. Thus, it is restricted to contain only constant amount of information about the whole families of input sequences $\{y_n\}_n$ and $\{y'_n\}_n$. In contrast, in the case of strong computational security, the adversarial distinguisher is given also the input sequences. Thus, it is able to compute any polynomial time computable information about the input sequences. This distinction is crucial for our results, as we are able to prove only an $\omega(1)$ lower bound for weak security as opposed to the $\Omega(\log n)$ lower bound for strong security (see Theorems 5 and 4). Nevertheless, we believe that the known constructions of ORAM satisfy the notion of strong computational security.

For ease of exposition, in the rest of the paper we assume perfect correctness of the ORAM (i.e., $p_{\text{fail}} = 0$). However, our lower bounds can be extended also to ORAMs with imperfect correctness (see Remark 1). Finally, our lower bounds hold also for *semi-offline* ORAMs where the ORAM machine $\mathcal{M}$ receives the type and address of each operation in advance and it has to process in online manner only the data to be written during each write operation (see Remark 2).

3 Dense Graphs

In this section, we define an efficiently testable property of graphs that we show to be satisfied by graphs induced by the access pattern of any statistically secure ORAM. This property implies that the overhead of such ORAM must be logarithmic.

We say a directed graph $G = (V, E)$ is *ordered* if V is a subset of integers and for each edge $(u, v) \in E$, $u < v$. For a graph $G = (V, E)$ and $S, T \subseteq V$, we let $E(S, T) \subseteq E$ be the set of edges that start in S and end in T, and for integers $a \le m \le b \in V$ we let $E(a, m, b) = E(\{a, a+1, \ldots, m-1\}, \{m, m+1, \ldots, b-1\})$.

Definition 5. *A* k-partition *of an ordered graph* $G = (V = \{0, 1, 2, \ldots, N-1\}, E)$ *is a sequence* $0 = b_0 \le m_0 \le b_1 \le m_1 \le \cdots \le b_k = N$. *We say that the* k*-partition is* ℓ*-dense if for each* $i \in \{0, \ldots, k-1\}$, $E(b_i, m_i, b_{i+1})$ *is of size at least* ℓ.

There is a simple greedy algorithm running in time $\mathcal{O}(|V|^2 \cdot |E|)$ which tests for given integers k, ℓ whether a given ordered graph $G = (V, E)$ has an ℓ-dense k-partition. (The algorithm looks for the k parts one by one greedily from left to right.)

Lemma 1. *Let* $K \subseteq \mathbb{N}$ *be a subset of powers of 4. Let* $\ell \in \mathbb{N}$ *be given. Let* $G = (\{0, \ldots, N-1\}, E)$ *be an ordered graph which for each* $k \in K$ *has an* (ℓ/k)*-dense* k*-partition. Then* G *has at least* $\frac{\ell}{2} \cdot |K|$ *edges.*

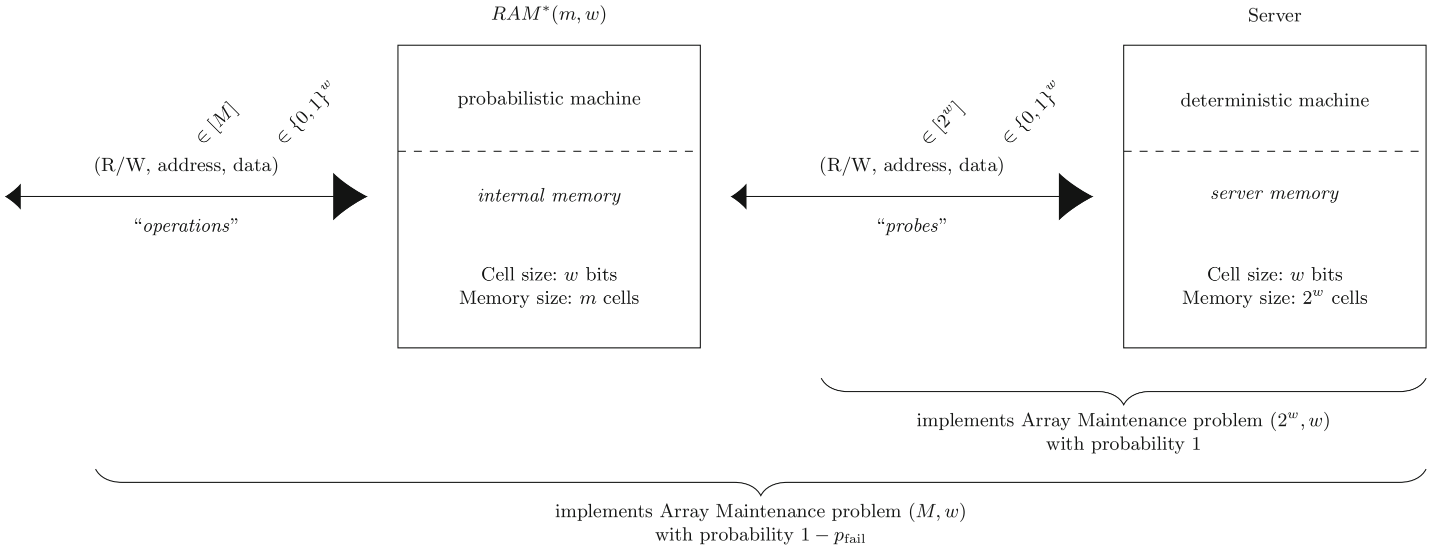

Fig. 1. Schema of online ORAM from Definition 4.

Proof. We use the following claim to bound the number of edges.

Claim. Let $k > k' > 0$ be integers. Let $0 = b_0 \leq m_0 \leq b_1 \leq m_1 \leq \cdots \leq b_k = N$ be a k-partition of G, and $0 = b'_0 \leq m'_0 \leq b'_1 \leq m'_1 \leq \cdots \leq b'_{k'} = N$ be a k'-partition of G. Then for at least $k - k'$ distinct $i \in \{0, \dots, k-1\}$

$$E(b_i, m_i, b_{i+1}) \cap \bigcup_{j \in \{0,\dots,k'-1\}} E(b'_j, m'_j, b'_{j+1}) = \emptyset. \tag{1}$$

Proof. For any $j \in \{0, \dots, k'-1\}$ and $(u, v) \in E(b'_j, m'_j, b'_{j+1})$, if $(u, v) \in E(b_i, m_i, b_{i+1})$ for some i then $b_i < m'_j < b_{i+1}$ (as $b_i \leq u < m'_j \leq v \leq b_{i+1}$.) Thus, i is uniquely determined by j. Hence, $E(b_i, m_i, b_{i+1})$ may intersect $\bigcup_{j \in \{0,\dots,k'-1\}} E(b'_j, m'_j, b'_{j+1})$ only if $b_i \leq m'_j < b_{i+1}$, for some $j \in \{0, \dots, k'-1\}$. Thus, such an intersection occurs only for at most k' different i. The claim follows. □

Now we are ready to prove Lemma 1. For each $k \in K$, pick an (ℓ/k)-dense k-partition $0 = b_0 \leq m_0 \leq b_1 \leq m_1 \leq \cdots \leq b_k = N$ of G and define the set of edges E_k:

$$E_k = \bigcup_{i \in \{0,\dots,k-1\}} E(b_i, m_i, b_{i+1}).$$

For each $k \in K$, we lower-bound $\left| E_k \setminus \bigcup_{k' \in K, k' < k} E_{k'} \right|$ by $\ell/2$. Since K contains powers of 4, $\sum_{k' \in K, k' < k} k' \leq k/2$. By the above claim, for at least $k - \sum_{k' \in K, k' < k} k' \geq k/2$ different $i \in \{0, \dots, k-1\}$, $E(b_i, m_i, b_{i+1}) \cap \bigcup_{k' \in K, k' < k} E_{k'} = \emptyset$. By density, $|E(b_i, m_i, b_{i+1})| \geq \ell/k$, so $\left| E_k \setminus \bigcup_{k' \in K, k' < k} E_{k'} \right| \geq \frac{\ell}{k} \cdot \frac{k}{2} = \ell/2$. Hence, $\left| \bigcup_{k \in K} E_k \right| = \sum_{k \in K} \left| E_k \setminus \bigcup_{k' \in K, k' < k} E_{k'} \right| \geq |K| \cdot \frac{\ell}{2}$. □

In the following corollary, we show that the property of having many dense partitions with some probability implies proportionally many edges. (Note that the $\lfloor \log_4 t \rfloor - \lceil \log_4 s \rceil$ term corresponds exactly to the number of powers of four between s and t.)

Corollary 1. *Let ℓ, s, t be natural numbers, where $s \leq t$. Let $p \in [0, 1]$ be a real. Let G be an ordered graph picked at random from a distribution such that for each integer k, $s \leq k \leq t$, the randomly chosen ordered graph G has (ℓ/k)-dense k-partition with probability at least p. Then the expected number of edges in G is at least $\frac{p\ell}{2} \cdot (\lfloor \log_4 t \rfloor - \lceil \log_4 s \rceil)$.*

Proof. Let K be the set of integers such that $k \in K$ if and only if k is a power of 4 and G has an (ℓ/k)-dense k-partition. K is a random variable. The expected size of K is at least $p(\lfloor \log_4 t \rfloor - \lceil \log_4 s \rceil)$. By Lemma 1, the expected number of edges in G is at least $\frac{\ell}{2} \cdot p \cdot (\lfloor \log_4 t \rfloor - \lceil \log_4 s \rceil)$. □

4 ORAM Lower Bound

In this section, we fix integers $n, m, M, w \geq 1$ such that $m \leq \sqrt{n}$, $n \leq M \leq 2^w$, and an ORAM $\mathcal{M}$ with address range M, cell size w and m cells of internal memory (see Definition 4). We argue that any statistically secure ORAM $\mathcal{M}$ must make $\Omega(n \log n)$ server probes in expectation in order to implement a sequence of n input operations. We also show that any ORAM $\mathcal{M}$ satisfying Weak Computational Security must make $\omega(n)$ server probes in expectation on any input sequence of length n.

Definition 6. *Let $A(\mathcal{M}, y) = a_0, \ldots, a_{N-1}$ be an access sequence of $\mathcal{M}$ for some input sequence y. We define a directed graph $G(A(\mathcal{M}, y)) = (V, E)$ called* access graph *as follows: $V = \{0, \ldots, N-1\}$ and $(i, j) \in E$ iff $i < j$ and $a_i = a_j$ and for each $k \in \{i+1, \ldots, j-1\}$, $a_k \neq a_i$.*

Notice that every vertex of an access graph has outdegree as well as indegree at most one.

In the following, we consider input sequences of even length $n \in \mathbb{N}$. First, we define a sequence of alternating writes and reads at address $a = 1$ with data $d = 0^w$ as $Y_{n,0} = [(W, 1, 0^w), (R, 1, 0^w)]^{n/2}$. Second, for each $k \in \left\{1, 2, \ldots, \frac{n}{2}\right\}$, let $\ell = \left\lfloor \frac{n}{2k} \right\rfloor$, we define a distribution $Y_{n,k}$ of input sequences as

$$\begin{aligned} Y_{n,k} =& (W, 1, b_{1,1}), (W, 2, b_{1,2}), \ldots, (W, \ell, b_{1,\ell}), (R, 1, 0^w), (R, 2, 0^w), \ldots, (R, \ell, 0^w), \\ & (W, 1, b_{2,1}), (W, 2, b_{2,2}), \ldots, (W, \ell, b_{2,\ell}), (R, 1, 0^w), (R, 2, 0^w), \ldots, (R, \ell, 0^w), \\ & \ldots, \\ & (W, 1, b_{k,1}), (W, 2, b_{k,2}), \ldots, (W, \ell, b_{k,\ell}), (R, 1, 0^w), (R, 2, 0^w), \ldots, (R, \ell, 0^w), \\ & (W, 1, 0^w), (R, 1, 0^w), (W, 1, 0^w), \ldots, (R, 1, 0^w) \ , \end{aligned}$$

where each $b_{i,j} \in \{0, 1\}^w$ is an independently uniformly chosen bit string. We define the i-th block of writes $W_i = (W, 1, b_{i,1}), (W, 2, b_{i,2}), \ldots, (W, \ell, b_{i,\ell})$ and the i-th block of reads R_i to be the sequence of operations $(R, 1, 0^w), (R, 2, 0^w), \ldots, (R, \ell, 0^w)$ following right after W_i. Note that after the k-th block of reads the sequence is padded to length n by a sequence of alternating writes and reads. For an ORAM $\mathcal{M}$, we use the notation $G_{n,k} = G(A(\mathcal{M}, Y_{n,k}))$ and $G_{n,0} = G(A(\mathcal{M}, Y_{n,0}))$ when $\mathcal{M}$ is clear from the context.

The following lemma uses only correctness of ORAM and does not depend on its security. The proof of the lemma uses the information transfer technique similarly to Lemma 2 in [18].

Lemma 2. *Let $n, m, M, w, \mathcal{M}$ be as in the beginning of this section, moreover suppose $n \geq 10$ is an even integer. Let $k \geq 1$ be an integer such that $k \leq \frac{n}{10(m+2\log n+11)}$. Let $A(\mathcal{M}, Y_{n,k})$ be the access sequence of $\mathcal{M}$ and $G_{n,k}$ be the corresponding access graph. ($G_{n,k}$ is a random variable that depends on $Y_{n,k}$ and the internal randomness of $\mathcal{M}$.) With probability at least $1 - \frac{1}{n}$, $G_{n,k}$ has $(n/5k)$-dense k-partition.*

Proof. By our assumption from the beginning of this section, $n \leq M$, and thus for any $k \in \{1, 2, \ldots, \frac{n}{2}\}$ all sequences $Y_{n,k}$ have all addresses in the correct range. Fix any k satisfying the assumptions of this lemma and set $\ell = \left\lfloor \frac{n}{2k} \right\rfloor$. As defined before let W_i and R_i be the i-th block of writes and reads in $Y_{n,k}$, respectively. Let U_i be the vertices of $G_{n,k}$ corresponding to W_i, and V_i be the vertices corresponding to R_i. It suffices to prove that for each $i \in \{1, \ldots, k\}$, the probability that there are fewer than $n/5k$ edges between U_i and V_i is less than $1/n^2$. If this holds then by the union bound the lemma follows.

For contradiction, assume there exists $i \in \{1, \ldots, k\}$ such that the probability that there are fewer than $n/5k$ edges between U_i and V_i is at least $1/n^2$. Here, the randomness is taken over the choice of an input sequence $y \leftarrow Y_{n,k}$ and the internal randomness of $\mathcal{M}$. Fix such an i. Fix all the randomness except for the choice of $b_{i,1}, \ldots, b_{i,\ell}$ in $Y_{n,k}$ so that $G_{n,k}$ obtained from this restricted distribution has fewer than $n/5k$ edges between U_i and V_i with probability $\geq 1/n^2$ over the choice of $b_{i,1}, \ldots, b_{i,\ell}$. (This is possible by an averaging argument.) Let $B \subseteq \{0,1\}^{w \times \ell}$ be the set of choices for $b_{i,1}, \ldots, b_{i,\ell}$ which give fewer than $n/5k$ edges between U_i and V_i in $G_{n,k}$. Clearly, $|B| \geq 2^{w\ell}/n^2$.

We use $\mathcal{M}$ to construct a deterministic protocol that transmits any string from B from Alice to Bob, two communicating parties, using at most $\log |B| - 10$ bits. That gives a contradiction as such an efficient transmission violates the pigeon-hole principle.

On input $b \in B$ to Alice, Alice sends a single message to Bob who can determine b from the message. They proceed as follows. Both Alice and Bob simulate $\mathcal{M}$ on $Y_{n,k}$ up until reaching W_i. All the randomness used before the i-th block of writes W_i is fixed and known both to Alice and Bob. Then Alice continues with the simulation of $\mathcal{M}$ on W_i with data $b_{i,1}, b_{i,2}, \ldots, b_{i,\ell}$ set to b. Once she finishes it, she sends the content of the internal memory of $\mathcal{M}$ to Bob using wm bits. Then Alice continues with the simulation of $\mathcal{M}$ on R_i and whenever $\mathcal{M}$ makes a server probe to read from a location that was written last time during the simulation of W_i, Alice sends over the address and the content of that cell to Bob. Overall, Alice sends at most $mw + 2wn/5k$ bits of communication to Bob that can be concatenated into a single message of this size.

On receiving side, Bob uses the internal state of $\mathcal{M}$ communicated by Alice to continue with the computation on R_i, while he uses the state of the server he obtained initially before reaching W_i. He simulates all server probes by himself, except for read operations that match the list sent by Alice, where he initially uses the content provided by Alice. Clearly, Bob can determine b from the simulation.

As $k \leq \frac{n}{10(m+2\log n+11)}$, $mw + 2wn/5k \leq (n/2k - 2\log n - 11)\, w$, so $mw + 2wn/5k \leq (\ell - 2\log n - 10)w$, hence, the number of communicated bits is $mw + 2wn/5k \leq \log |B| - (2w-2)\log n - 10w$, which is a contradiction. □

Remark 1. Using good error-correcting codes (see for instance [20]), this lemma could be generalized to the case when $\mathcal{M}$ implements Array Maintenance problem with probability $1 - p_{\text{fail}} < 1$, i.e., $\mathcal{M}$ is allowed to return a wrong value for each of its input read operations with a small constant probability p_{fail}. The graph $G_{n,k}$ would still have $(\epsilon n/k)$-dense k-partition with $1 - 1/n$ probability for some $\epsilon > 0$ which depends only on the allowed failure probability p_{fail}.

Remark 2. Note that the randomness of input sequence $Y_{n,k}$ is used only for the data to be written. Moreover, the proof relies only on incompressibility of a random string stored during the write block and it does not rely on the addresses used to store this data. Thus, the same proof goes through even for *semi-offline ORAMs*, i.e., if we allow the ORAM to know the type and address of each input operation in y in advance. On the other hand, as our proof uses interleaved sequences of write blocks and read blocks, it is unlikely that it would be possible to extend it to the *read-only online ORAM* model of Weiss and Wichs [30].

Note that using an averaging argument we can assume that the probability in Lemma 2 is only over the randomness of $\mathcal{M}$. Thus we get the following corollary proving for every k the existence of a single input sequence whose corresponding access graph has $\frac{n}{5k}$-dense k-partition with high probability.

Corollary 2. *For any even integer $n \geq 10$ and an integer $k \geq 1$ such that $k \leq \frac{n}{10(m+2\log n+11)}$ there is an input sequence $y_{n,k}$ of length n such that $G(A(\mathcal{M}, y_{n,k}))$ has a $(n/5k)$-dense k-partition with probability at least $1 - \frac{1}{n}$.*

We show that by statistical security of $\mathcal{M}$, this property holds for a single input sequence and many different values of k.

Lemma 3. *Let $n, m, M, w, \mathcal{M}$ be as in the beginning of this section, and assume n is even and $n \geq 10$. Let y be an input sequence to $\mathcal{M}$ of length n. If $\mathcal{M}$ is a statistically secure online ORAM then for every $k \in \left\{1, 2, \ldots, \left\lfloor \frac{n}{10(m+2\log n+11)} \right\rfloor \right\}$*

$$\Pr\left[G(A(\mathcal{M}, y)) \text{ has an } (n/5k)\text{-dense } k\text{-partition}\right] \geq \frac{3}{5}.$$

Proof. For contradiction, suppose that for some k the probability is less than 3/5. From the statistical security of $\mathcal{M}$ we know that the statistical distance $\mathrm{SD}\left(A(\mathcal{M}, y), A(\mathcal{M}, y_{n,k})\right) \leq \frac{1}{4}$ where $y_{n,k}$ is given by Corollary 2. By Corollary 2 the sequence $y_{n,k}$ gives us a graph $G(A(\mathcal{M}, y_{n,k}))$ which has an $(n/5k)$-dense k-partition with probability at least $1 - 1/n \geq 9/10$. Define a function $f_{\ell,k}$ on ordered graphs that is an indicator of having an ℓ-dense k-partition. Applying Proposition 2 with $X \leftarrow G(A(\mathcal{M}, y))$, $Y \leftarrow G(A(\mathcal{M}, y_{n,k}))$, and $f = f_{n/5k,k}$, we can conclude that $G(A(\mathcal{M}, y))$ has an $(n/5k)$-dense k-partition with probability at least $3/4 - 1/10 \geq 3/5$. □

We are ready to prove our main theorem for statistically secure ORAM.

Theorem 3. *There are constants $c_0, c_1 > 0$ such that for any integers $m, w \geq 1$ and $M \geq n \geq c_0$ where $m \leq \sqrt{n}$ and $M \leq 2^w$, any statistically secure online ORAM $\mathcal{M}$ with address range M, cell size w bits and m cells of internal memory must perform at least $c_1 n \log n$ server probes in expectation (the expectation is over the randomness of $\mathcal{M}$) on any input sequence of length n.*

Proof. Fix an ORAM machine $\mathcal{M}$. Consider any input sequence y to $\mathcal{M}$ of length n. By Lemma 3 for every k, such that $1 \leq k \leq \left\lfloor \frac{n}{10(m+2\log n+11)} \right\rfloor$, we get that

$$\Pr\left[G(A(\mathcal{M},y)) \text{ has an } (n/5k)\text{-dense } k\text{-partition}\right] \geq \frac{3}{5}.$$

Applying Corollary 1 with $s = 1$, $t = \left\lfloor \frac{n}{10(m+2\log n+11)} \right\rfloor$, $\ell = \left\lfloor \frac{n}{5} \right\rfloor$, and $p = 3/5$, we can lower bound the expected number of edges in $G(A(\mathcal{M},y))$ by

$$\frac{3n}{50}\left\lfloor \log_4 \left\lfloor \frac{n}{10(m+2\log n+11)} \right\rfloor \right\rfloor.$$

For $n \geq 1000$, $\left\lfloor \frac{n}{10(m+2\log n+11)} \right\rfloor \geq \frac{\sqrt{n}}{40}$. Hence, the expected number of edges in $G(A(\mathcal{M},y))$ is at least $\frac{3}{100} \cdot n \log \frac{\sqrt{n}}{40} \geq \frac{1}{100} \cdot n \log n$, provided c_0 is large enough. Since the indegree of each vertex of an access graph is at most one, the expected number of vertices in $G(A(\mathcal{M},y))$, which is the same as the expected number of probes in $A(\mathcal{M},y)$, is at least $\frac{1}{100} \cdot n \log n$. □

Next, we prove $\Omega(\log n)$ lower bound for ORAMs satisfying strong computational security from Definition 4.

Lemma 4. *Let $m, M, w \colon \mathbb{N} \to \mathbb{N}$ be non-decreasing functions such that for all n large enough: $m(n) \leq \sqrt{n}$ and $n \leq M(n) \leq 2^{w(n)}$. Let $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$ be a sequence of online ORAMs with address range $M(n)$, cell size $w(n)$ bits and $m(n)$ cells of internal memory which satisfy strong computational security. Let $\{y_n\}_{n\in\mathbb{N}}$ be an infinite family of input sequences where $|y_n| = n$, for each $n \in \mathbb{N}$.*

Then there exists n_0 such that for every $n \geq n_0$ and for every k in the set $\left\{1, 2, \ldots, \left\lfloor \frac{n}{10(m(n)+2\log n+11)} \right\rfloor \right\}$,

$$\Pr\left[G(A(\mathcal{M}_n,y_n)) \textit{ has an } (n/5k)\textit{-dense } k\textit{-partition}\right] \geq \frac{3}{5}.$$

Proof. For contradiction, assume there are infinitely many pairs of integers (n,k), s.t. $k \leq \left\lfloor \frac{n}{10(m(n)+2\log n+11)} \right\rfloor$ and that the probability that y_n has an $(n/5k)$-dense k-partition is less than $3/5$.

Let $\mathcal{D}$ be an algorithm which given two input sequences y and y' of length n and an access sequence $A(\mathcal{M}_n, z)$, where $z \in \{y, y'\}$, does the following:

1. Compute n.
2. Compute k' to be the number of blocks of consecutive reads of length $\lfloor n/k' \rfloor$ in the input sequence y'.
3. If $A(\mathcal{M}_n, z)$ does not have $(n/5k')$-dense k'-partition $\mathcal{D}$ returns "1" (i.e. D guesses that $z = y$).
4. Otherwise $\mathcal{D}$ returns "1" with probability 1/2 and "2" with probability 1/2 (i.e. D guesses at random).

There is a polynomial time greedy algorithm determining whether the graph $G(A(\mathcal{M}_n, z))$ contains an ℓ-dense k-partition. Thus algorithm $\mathcal{D}$ runs in time polynomial in the length of the access sequence $A(\mathcal{M}_n, z)$.

Let $y_{n,k}$ be a sequence from Corollary 2. So, $G(A(\mathcal{M}_n, y_{n,k}))$ has an $(n/5k)$-dense k-partition with probability at least $1-1/n \geq 9/10$. Observe that if $y = y_n$ and $y' = y_{n,k}$ then:

$$\begin{aligned}|\Pr[\mathcal{D}(y_n, y_{n,k}, A(\mathcal{M}_n, y_n)) = 1] - \Pr[\mathcal{D}(y_n, y_{n,k}, A(\mathcal{M}_n, y_{n,k})) = 1]| \\ \geq \left(\frac{2}{5} + \frac{3}{5} \cdot \frac{1}{2}\right) - \left(\frac{1}{10} + \frac{9}{10} \cdot \frac{1}{2}\right) = \frac{3}{20}.\end{aligned}$$

By the assumption $\mathcal{D}$ returns “1” in step 3 on $A(\mathcal{M}_n, y_n)$ with probability at least 2/5. By Corollary 2 $\mathcal{D}$ answers “1” on $A(\mathcal{M}_n, y_{n,k})$ with probability at most 1/10.

This contradicts the strong computational security of $\mathcal{M}_n$ as $\mathcal{D}$ should not distinguish between y and y' with non-negligible probability. □

Theorem 4. *Let $m, M, w \colon \mathbb{N} \to \mathbb{N}$ be non-decreasing functions such that for all n large enough: $m(n) \leq \sqrt{n}$ and $n \leq M(n) \leq 2^{w(n)}$. Let $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$ be a sequence of online ORAMs with address range $M(n)$, cell size $w(n)$ bits and $m(n)$ cells of internal memory which satisfy strong computational security. Let $\{y_n\}_{n\in\mathbb{N}}$ be an infinite family of input sequences where $|y_n| = n$, for each $n \in \mathbb{N}$.*

There are constants $c_0, c_1 > 0$, such that for any $n \geq c_0$, $\mathcal{M}_n$ must perform in expectation at least $c_1 n \log n$ server probes on the input sequence y_n.

Proof. The proof is identical to the proof of Theorem 3 but we use Lemma 4 instead of Lemma 3. Note that the different order of quantifiers is caused by different order of quantifiers in Lemma 3 and in Lemma 4. □

In the rest of this section, we prove an $\omega(1)$ lower bound for ORAMs satisfying weak computational security from Definition 4. Note that in the case of weak computational security it is unclear which k should the adversary use to distinguish y and y'. Thus, we cannot directly conclude that y has $\frac{n}{5k}$-dense k-partition for every n and $k \leq \left\lfloor \frac{n}{10(m(n)+2\log n+11)} \right\rfloor$. On the other hand, for every k there could be only finitely many values n such that there is an input sequence of length n which has no $\frac{n}{5k}$-dense k-partition. This fact allows us to prove the $\omega(1)$ lower bound for weak computational security.

Theorem 5. *Let $m, M, w \colon \mathbb{N} \to \mathbb{N}$ be non-decreasing functions such that for all n large enough: $m(n) \leq \sqrt{n}$ and $n \leq M(n) \leq 2^{w(n)}$. Let $\{\mathcal{M}_n\}_{n\in\mathbb{N}}$ be a sequence of online ORAMs with address range $M(n)$, cell size $w(n)$ bits and $m(n)$ cells of internal memory which satisfy weak computational security. Let $\{y_n\}_{n\in\mathbb{N}}$ be a sequence of input sequences where $|y_n| = n$, for each $n \in \mathbb{N}$.*

For any constant $c_1 > 0$ there is a constant $c_0 > 0$, such that for any $n \geq c_0$, $\mathcal{M}_n$ must perform in expectation at least $c_1 n$ server probes on the input sequence y_n.

In particular there is no computationally secure online ORAM with constant bandwidth overhead $\mathcal{O}(1)$.

Proof. For each $n \in \mathbb{N}$, define $k(n)$ to be the smallest k such that

$$\Pr[G(A(\mathcal{M}_n, y_n)) \text{ has } (n/5k)\text{-dense } k\text{-partition}] < 1/2.$$

Using Corollary 1 we get for each n large enough that the expected number of edges in $G(A(\mathcal{M}_n, y_n))$ is at least $c \cdot n \log k(n)$, for some absolute constant $c > 0$. It suffices to show that $k(n) \to \infty$ as $n \to \infty$. There cannot exist a constant k such that Y_n has $(n/5k)$-dense k-partition with probability less than $\frac{1}{2}$ for infinitely many n. Otherwise $\{y_n\}_n$ would be computationally distinguishable from $\{Y_{n,k}\}_n$ (by the greedy algorithm which has k hard-wired). So, $k(n) \to \infty$ as $n \to \infty$. □

5 Alternative Definitions for Oblivious RAM

In this section, we recall some alternative definitions for ORAM which appeared in the literature and explain the relation of our lower bound to those models.

The Definition of Larsen and Nielsen. Larsen and Nielsen (see Definition 4 in [18]) required that for any two input sequences of equal length, the corresponding distributions of access sequences cannot be distinguished with probability greater than $1/4$ by any algorithm running in polynomial time in the sum of the following terms: the length of the input sequence, logarithm of the number of memory cells (i.e., $\log n$), and the size of a memory cell (i.e., $\log n$ for the most natural parameters). We show that their definition implies statistical closeness as considered in our work (see the statistical security property in Definition 4). Therefore, any lower bound on the bandwidth overhead of ORAM satisfying our definition implies a matching lower bound w.r.t. the definition of Larsen and Nielsen [18].

To this end, let us show that if two distributions of access sequences are not statistically close, then they are distinguishable in the sense of Larsen and Nielsen. Assume there exist two input sequences y and y' of equal lengths, for which the access sequences $A(\mathcal{M}, y)$ and $A(\mathcal{M}, y')$ have statistical distance greater than $1/4$. We define a distinguisher algorithm D that on access sequence x outputs 1 whenever $\Pr[A(\mathcal{M}, y) = x] > \Pr[A(\mathcal{M}, y') = x]$, outputs 0 whenever $\Pr[A(\mathcal{M}, y) = x] < \Pr[A(\mathcal{M}, y') = x]$, and outputs a uniformly random bit whenever $\Pr[A(\mathcal{M}, y) = x] = \Pr[A(\mathcal{M}, y') = x]$. It follows from definition of D, basic properties of statistical distance (see proposition 1), and our assumption about the statistical distance of $A(\mathcal{M}, y)$ and $A(\mathcal{M}, y')$ that

$$|\Pr[D(A(\mathcal{M}, y)) = 1] - \Pr[D(A(\mathcal{M}, y')) = 1]| = \mathrm{SD}\left(A(\mathcal{M}, y), A(\mathcal{M}, y')\right) > \frac{1}{4}\ .$$

Note that D can be specific for the pair of the two input sequences y and y' and it can have all the significant information about the distributions $A(\mathcal{M}, y)$ and

$A(\mathcal{M}, y')$ hardwired. For example, it is sufficient to store a string describing for each access sequence x whether it is more, less, or equally likely under $A(\mathcal{M}, y)$ or $A(\mathcal{M}, y')$. Even though such string is of exponential size w.r.t. the length of the access pattern, D needs to simply access the position corresponding to the observed access pattern to output its decision as described above. Thus, D can run in linear time in the length of the access sequence (which is polynomial in the length of the input sequence) and distinguishes the two access sequences with probability greater than $1/4$.

The Definition of Goldreich and Ostrovsky. Unlike the original definition of ORAM from Goldreich [10] and Ostrovsky [21], the definition of ORAM presented in Goldreich and Ostrovsky [11] postulates an alternative security requirement. However, the alternative definition suffers from an issue which is not present in the original definition and which, to the best of our knowledge, was not pointed out in the literature. In particular, the definition in [11] can be satisfied by a dummy ORAM construction with only a constant overhead and without achieving any indistinguishability of the access sequences. Given that Goldreich and Ostrovsky [11] might serve as a primary reference for our community, we explain the issue in the following paragraph to help preventing the use of the problematic definition in future works.

Recall the definition of ORAM with perfect security from Goldreich and Ostrovsky (Definition 2.3.1.3 in [11]):

Goldreich-Ostrovsky Security: *For any two input sequences y and y', if the length distributions $|A(\mathcal{M}, y)|$ and $|A(\mathcal{M}, y')|$ are identical, then $A(\mathcal{M}, y)$ and $A(\mathcal{M}, y')$ are identical.*

As we show, this requirement can be satisfied by creating an ORAM that makes sure that on any two distinct sequences y, y', the length distributions $|A(\mathcal{M}, y)|$ and $|A(\mathcal{M}, y')|$ differ. Note that no indistinguishability is required in that case and the ORAM can then reveal the access pattern of the input sequence.

To this end, we describe an ORAM with a constant overhead so that the length $|A(\mathcal{M}, y)|$ is either $2|y|$ or $2|y|+1$ and the distribution $|A(\mathcal{M}, y)|$ encodes the sequence y. The ORAM proceeds by performing every operation y_i directly on the server followed by a read operation from address 1. After the last instruction in y, the ORAM selects a random sequence of operations r of length $|y|$ and if r is lexicographically smaller than y then the ORAM performs an extra read from address 1 before terminating. Note that this ORAM can be efficiently implemented using constant amount of internal memory by comparing the input sequence to the randomly selected one online. Also, the machine does not need to know the length of the sequence in advance. Finally, the length distribution $|A(\mathcal{M}, y)|$ is clearly different for each input sequence y. Given that the above definition of ORAM of Goldreich and Ostrovsky allows the dummy construction with a constant overhead, we do not hope to extend our lower bound towards this definition.

One could object that the above dummy ORAM exploits the fact that indistinguishability of access sequences must hold only if the length distributions are identical. However, it is possible to construct a similar dummy ORAM with low overhead satisfying even the following relaxation of the definition requiring indistinguishability of access sequences corresponding to any pair of y and y' for which $|A(M, y)|$ and $|A(M, y')|$ are statistically close (i.e., the indistinguishability is required for a potentially larger set of access patterns):

Relaxation of Goldreich-Ostrovsky Security: *For any two input sequences y and y', if the length distributions $|A(\mathcal{M}, y)|$ and $|A(\mathcal{M}, y')|$ are statistically close, then $A(\mathcal{M}, y)$ and $A(\mathcal{M}, y')$ are statistically close.*

We show there is a dummy ORAM $\mathcal{M}$ with a constant overhead such that for any two input sequences y and y' which differ in their accessed memory locations, the statistical distance $\mathrm{SD}\left(|A(\mathcal{M}, y)|, |A(\mathcal{M}, y')|\right)$ is at least $\frac{1}{nM}$ (where $n = |y| = |y'|$ and M is the size of address range).

The ORAM $\mathcal{M}$ works as follows. At the beginning, the ORAM picks $i \in [n]$ and $r \in [M]$ uniformly at random. Then for $j = 1, \ldots n$, it executes each of the input operations (o_j, a_j, d_j) directly on the server. For each $j < i$, it performs two additional reads from address 1 after executing the j-th input operation. For $j = i$, after the i-th input operation it performs two additional reads from address 1 if $r \leq a_i$, and it performs one additional read from address 1 if $r > a_i$. For $j > i$, it performs each of the input operations without any additional read.

It is straightforward to verify that the distribution of $|A(\mathcal{M}, y)|$ satisfies: for each $i \in [n]$, $\Pr[|A(\mathcal{M}, y)| = n + 2i] = \frac{a_i}{nM}$. Hence, for any pair y and y' of two input sequences of length n, if the sequences of addresses accessed by them differ then the statistical distance between the distributions of $|A(\mathcal{M}, y)|$ and $|A(\mathcal{M}, y')|$ is at least $1/nM$. If M is polynomial in n this means that their distance is at least $\frac{1}{\mathrm{poly}(n)}$. Thus, $\mathcal{M}$ satisfies even the stronger variant of the definition from [11] even though its access sequence leaks the addresses from the input sequence.

It was previously shown by Haider, Khan and van Dijk [14] that there exists an ORAM construction which reveals all memory accesses from the input sequence while satisfying the definition of Goldreich and Ostrovsky from [11]. However, their construction has an *exponential* bandwidth overhead which makes it insufficient to demonstrate any issue with the definition of Goldreich and Ostrovsky. Clearly, any definition of ORAM can disregard constructions with super-linear overhead as a perfectly secure ORAM (with linear overhead) can be constructed by simply passing over the whole server memory for each input operation. Unlike the construction of [14], our constructions of the dummy ORAMs with constant bandwidth overhead exemplify that the definition of Goldreich and Ostrovsky from [11] is problematic in the interesting regime of parameters.

Simulation-Based Definitions. The recent work of Asharov et al. [2] employs a simulation-based definition parameterized by a functionality which implements an oblivious data structure. Our lower bounds directly extend to their stronger definition when the functionality implements Array Maintenance. Moreover, our techniques can be adapted to give lower bounds for functionalities implementing stacks, queues and others considered in [15].

Weak vs. Strong Computational Security. In this work, we distinguish between weak and strong computational security (see Definition 4). Our techniques do not allow to prove matching bounds for ORAMs satisfying the two notions and we show $\Omega(\log n)$ lower bound only w.r.t. strong computational security. Though, as we noted in Sect. 1.1, even the $\omega(1)$ lower bound for online ORAMs satisfying weak computational security is an interesting result in the light of the work of Boyle and Naor [3]. It follows from [3] that any super-constant lower bound for *offline* ORAM would imply super-linear lower bounds on size of sorting circuits – which would constitute a major breakthrough in computational complexity. The main result from Boyle and Naor [3] can be rephrased using our notation as follows.

Theorem 6 (Theorem 3.1 [3]**).** *Suppose there exists a Boolean circuit ensemble $C = \{C(n, w)\}_{n,w}$ of size $s(n, w)$, such that each $C(n, w)$ takes as input n words each of size w bits, and outputs the words in sorted order. Then for word size $w \in \Omega(\log n) \cap n^{o(1)}$ and constant internal memory $m \in \mathcal{O}(1)$, there exists a secure offline ORAM (as per Definition 2.8 [3]) with total bandwidth and computation $\mathcal{O}(n \log w + s(2n/w, w))$.*

Moreover, the additive factor of $\mathcal{O}(n \log w)$ follows from the transpose part of the algorithm of [3] (see Figs. 1 and 2 in [3]). As Boyle and Naor showed in their appendix (Remark B.3 [3]) this additive factor in total bandwidth may be reduced to $\mathcal{O}(n)$ if the size of internal memory is $m \geq w$. Thus, sorting circuit of size $\mathcal{O}(nw)$ implies offline ORAM with total bandwidth $\mathcal{O}(n+2\frac{n}{w}w) = \mathcal{O}(n)$. Or the other way around, lower bound $\omega(n)$ for total bandwidth of offline ORAM implies $\omega(nw)$ lower bound for circuits sorting n words of size w bits, each.

We leave it as an intriguing open problem whether it is possible to prove an $\Omega(\log n)$ lower bound for online ORAMs satisfying weak computational security.

Acknowledgements. We wish to thank Oded Goldreich for clarifications regarding the ORAM definitions in [10,11,21] and Jesper Buus Nielsen for clarifying the details of the lower bound for computationally secure ORAMs from [18]. We are also thankful to the anonymous TCC 2019 reviewers for insightful comments that helped us improve the presentation of our results.

References

1. Ajtai, M.: Oblivious RAMs without cryptographic assumptions. In: Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5–8 June 2010, pp. 181–190 (2010)
2. Asharov, G., Komargodski, I., Lin, W., Nayak, K., Peserico, E., Shi, E.: OptORAMa: optimal oblivious RAM. IACR Cryptology ePrint Archive 2018/892 (2018)
3. Boyle, E., Naor, M.: Is there an oblivious RAM lower bound? In: Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, 14–16 January 2016, pp. 357–368 (2016)

4. Chung, K.-M., Liu, Z., Pass, R.: Statistically-secure ORAM with $\tilde{O}(\log^2 n)$ overhead. In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 62–81. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45608-8_4
5. Chung, K., Pass, R.: A simple ORAM. IACR Cryptology ePrint Archive 2013/243 (2013)
6. Damgård, I., Meldgaard, S., Nielsen, J.B.: Perfectly secure oblivious RAM without random oracles. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 144–163. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_10
7. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006). https://doi.org/10.1007/11681878_14
8. Fredman, M.L., Saks, M.E.: The cell probe complexity of dynamic data structures. In: Proceedings of the 21st Annual ACM Symposium on Theory of Computing, Seattle, Washigton, USA, 14–17 May 1989, pp. 345–354 (1989)
9. Gentry, C., Goldman, K.A., Halevi, S., Julta, C., Raykova, M., Wichs, D.: Optimizing ORAM and using it efficiently for secure computation. In: De Cristofaro, E., Wright, M. (eds.) PETS 2013. LNCS, vol. 7981, pp. 1–18. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39077-7_1
10. Goldreich, O.: Towards a theory of software protection and simulation by oblivious RAMs. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, USA, pp. 182–194 (1987)
11. Goldreich, O., Ostrovsky, R.: Software protection and simulation on oblivious RAMs. J. ACM **43**(3), 431–473 (1996)
12. Goodrich, M.T., Mitzenmacher, M.: Privacy-preserving access of outsourced data via oblivious RAM simulation. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part II. LNCS, vol. 6756, pp. 576–587. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22012-8_46
13. Goodrich, M.T., Mitzenmacher, M., Ohrimenko, O., Tamassia, R.: Oblivious RAM simulation with efficient worst-case access overhead. In: Proceedings of the 3rd ACM Cloud Computing Security Workshop, CCSW 2011, Chicago, IL, USA, 21 October 2011, pp. 95–100 (2011)
14. Haider, S.K., Khan, O., van Dijk, M.: Revisiting definitional foundations of oblivious RAM for secure processor implementations. CoRR abs/1706.03852 (2017). http://arxiv.org/abs/1706.03852
15. Jacob, R., Larsen, K.G., Nielsen, J.B.: Lower bounds for oblivious data structures. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, 6–9 January 2019, pp. 2439–2447 (2019)
16. Kocher, P., et al.: Spectre attacks: exploiting speculative execution. CoRR abs/1801.01203 (2018). http://arxiv.org/abs/1801.01203
17. Kushilevitz, E., Lu, S., Ostrovsky, R.: On the (in) security of hash-based oblivious RAM and a new balancing scheme. In: Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, 17–19 January 2012, pp. 143–156 (2012)
18. Larsen, K.G., Nielsen, J.B.: Yes, there is an oblivious RAM lower bound!. In: Shacham, H., Boldyreva, A. (eds.) CRYPTO 2018, Part II. LNCS, vol. 10992, pp. 523–542. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96881-0_18
19. Lipp, M., et al.: Meltdown: reading kernel memory from user space. In: 27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, 15–17 August 2018, pp. 973–990 (2018)

20. MacWilliams, F., Sloane, N.: The Theory of Error-Correcting Codes. North-Holland, Amsterdam (1977)
21. Ostrovsky, R.: Efficient computation on oblivious RAMs. In: Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, Baltimore, Maryland, USA, 13–17 May 1990, pp. 514–523 (1990)
22. Patel, S., Persiano, G., Raykova, M., Yeo, K.: PanORAMa: oblivious RAM with logarithmic overhead. In: 59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, 7–9 October 2018, pp. 871–882 (2018)
23. Patrascu, M., Demaine, E.D.: Logarithmic lower bounds in the cell-probe model. SIAM J. Comput. **35**(4), 932–963 (2006)
24. Persiano, G., Yeo, K.: Lower bounds for differentially private RAMs. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part I. LNCS, vol. 11476, pp. 404–434. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17653-2_14
25. Ren, L., Fletcher, C.W., Kwon, A., Stefanov, E., Shi, E., van Dijk, M., Devadas, S.: Ring ORAM: closing the gap between small and large client storage oblivious RAM. IACR Cryptology ePrint Archive 2014/997 (2014)
26. Stefanov, E., et al.: Path ORAM: an extremely simple oblivious RAM protocol. J. ACM **65**(4), 18:1–18:26 (2018)
27. Vadhan, S.P.: A Study of Statistical-Zero Knowledge Proofs. Ph.D. thesis, Massachusetts Institute of Technology, September 1999
28. Wang, X., Chan, T.H., Shi, E.: Circuit ORAM: on tightness of the Goldreich-Ostrovsky lower bound. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015, pp. 850–861 (2015)
29. Wang, X.S., Huang, Y., Chan, T.H., Shelat, A., Shi, E.: SCORAM: oblivious RAM for secure computation. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, 3–7 November 2014, pp. 191–202 (2014)
30. Weiss, M., Wichs, D.: Is there an oblivious RAM lower bound for online reads? In: Beimel, A., Dziembowski, S. (eds.) TCC 2018, Part II. LNCS, vol. 11240, pp. 603–635. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-03810-6_22