Speak Much, Remember Little: Cryptography in the Bounded Storage Model, Revisited[⋆]

Yevgeniy Dodis[⋆⋆1], Willy Quach[⋆⋆⋆2], and Daniel Wichs[†3]

[1] New York University, New York, NY 10012, USA
[2] Northeastern University, Boston, MA 02115, USA
[3] NTT Research, Sunnyvale, CA 94085, USA

Abstract. The goal of the *bounded storage model (BSM)* is to construct unconditionally secure cryptographic protocols, by only restricting the storage capacity of the adversary, but otherwise giving it unbounded computational power. Here, we consider a *streaming* variant of the BSM, where honest parties can stream huge amounts of data to each other so as to overwhelm the adversary's storage, even while their own storage capacity is significantly smaller than that of the adversary. Prior works showed several impressive results in this model, including *key agreement* and *oblivious transfer*, but only as long as adversary's storage $m = O(n^2)$ is at most quadratically larger than the honest user storage n. Moreover, the work of Dziembowski and Maurer (DM) also gave a seemingly matching lower bound, showing that key agreement in the BSM is impossible when $m > n^2$.

In this work, we observe that the DM lower bound only applies to a significantly more restricted version of the BSM, and does not apply to the streaming variant. Surprisingly, we show that it is possible to construct key agreement and oblivious transfer protocols in the streaming BSM, where the adversary's storage can be significantly larger, and even exponential $m = 2^{O(n)}$. The only price of accommodating larger values of m is that the round and communication complexities of our protocols grow accordingly, and we provide lower bounds to show that an increase in rounds and communication is necessary.

As an added benefit of our work, we also show that our oblivious transfer (OT) protocol in the BSM satisfies a simulation-based notion of security. In contrast, even for the restricted case of $m = O(n^2)$, prior solutions only satisfied a weaker indistinguishability based definition. As an application of our OT protocol, we get general multiparty computation (MPC) in the BSM that allows for up to exponentially large gaps between m and n, while also achieving simulation-based security.

[⋆] The full version of this paper is available online [13].

[⋆⋆] `E-mail:dodis@cs.nyu.edu`. Supported by gifts from VMware Labs and Algorand, and NSF grants 2055578 and 1815546.

[⋆⋆⋆] `E-mail:quach.w@northeastern.edu`. Part of this work was completed during an internship at NTT Research.

[†] `E-mail:wichs@ccs.neu.edu`. Research supported by NSF grant CNS-1750795, CNS-2055510 and the Alfred P. Sloan Research Fellowship.

1 Introduction

It is well known that Alice and Bob cannot agree on a shared secret by communicating over public (authentic) channel, when the eavesdropper Eve has unbounded computational resources. Thus, traditional cryptography assumes that Eve is "resource bounded", and most commonly, bounds her run time. Many key agreement schemes have been constructed in this setting, starting with the seminal work of Diffie and Hellman [8], under various computational hardness assumptions. Of course, the dream of cryptography is to construct unconditionally secure protocols, without relying on any unproven assumptions, but unfortunately, this is currently beyond our reach, as it easily implies $P \neq NP$.

In contrast, the *Bounded Storage Model* (BSM), introduced in the pioneering work of Maurer [29], *only* assumes that Eve has bounded space rather than time. A long series of works [5, 4, 9, 21, 11, 1, 15, 28, 39, 10, 35, 26, 36, 17, 19] showed that it is possible to construct many kinds of unconditionally secure cryptographic schemes in this model, including key agreement and oblivious transfer over a public channel, provided that Eve's storage is not too large.

It turns out that there are several related-but-different variants of the BSM. In this work, we focus on a natural variant, which we refer to as the "streaming BSM". We first discuss this model, which will be the default throughout the paper. We will compare the streaming BSM model to other variants from the literature further below.

"Streaming" BSM. In this model, parties can generate and send huge amounts of data to each other, but only have limited local memory. The model is parameterized by two parameters: the honest parties' space capacity n, and the attacker's space capacity m, where $m \gg n$. We assume parties operate in the streaming model: they generate/receive communication one bit at a time, while only maintaining a small local memory throughout. The total communication k can be huge, say $k \gg m \gg n$, and can occur over multiple back-and-forth rounds.

For example, Alice can stream a huge random string X of length k to Bob by sampling it one bit at a time; both Alice and Bob can store some small subset of n physical locations of X, or they can store the parity of X computed in a streaming manner, but neither of them can remember all of X. The attacker Eve is also streaming, just like Alice and Bob, but has much larger memory capacity $m \gg n$. We call the resulting model the (n, m)-BSM, and it will be the default throughout the paper; sometimes, we will explicitly refer to it as the "streaming BSM" to disambiguate from other variants.

Prior Results. As with computational cryptography, in the BSM we can consider a symmetric-key setting, where honest parties can share a short secret key that can be used to encrypt arbitrarily many messages over time, or a public-key setting, where no shared key is available. In both cases the parties can freely communicate over a public channel, and the goal is to achieve unconditional, information-theoretic (IT) security, without making any additional computational assumptions.

In the symmetric-key setting, a series of beautiful papers [29, 11, 1, 15, 28, 39, 35, 26, 36, 17] showed that it is possible to achieve arbitrarily large gaps between the space of the attacker and that of the honest parties, up to exponential: $m = 2^{O(n)}$. (Of course, the price of allowing large values of m is that the ciphertext size has to grow proportionally, to ensure that we eventually overwhelm the adversary's storage capacity to overcome the Shannon lower bound. Therefore, if we want to limit ourselves to schemes with polynomial ciphertext size, then m is limited to some arbitrarily large polynomial.)

Amazingly, it is even possible to construct unconditionally secure public-key schemes in the BSM, and prior works [5, 4, 9, 21, 10, 19] constructed BSM schemes for key agreement (KA) and oblivious transfer (OT), which is then complete for all multi-party computation (MPC) [25, 23]. However, all of the prior works in the public-key setting allowed at most a quadratic gap between the adversarial and the honest users storage: $m = O(n^2)$. In fact, the work of Dziembowski and Maurer [16] seemed to suggest that this limitation is inherent, by showing there is no KA protocol in the BSM when $m > n^2$. Since OT directly implies KA, the same lower bound also extends to OT. So it may have appeared that the question of designing public-key cryptographic primitives in the BSM had been settled.

Our Question and Main Result. However, as we observe in this work, and discuss in Section 1.1, the lower bound of [16] was only shown in a *restricted* version of the BSM model, and does not apply to the more general "streaming" BSM. Most significantly, the authors critically assumed that there is *at most one* "long" communication round in the key agreement protocol, where the length k of the streamed message overwhelms the storage capacity m of the attacker. While this restriction was satisfied by many prior work in the BSM (see Section 1.1), this opens the possibility that it might be possible to break the quadratic barrier of [16] when parties use the full streaming power of the BSM, including the ability to stream *several* "long" messages to each other. This is the main question of this work:

Main Question: *Do there exist unconditionally secure key agreement (KA) and oblivious trasnfer (OT) protocols in the streaming (n, m)-BSM, when m is allowed to be much larger than n^2?*

We answer this question in the *affirmative*, and show that we can allow arbitrarily large gaps between m and n, up to exponential $m = 2^{O(n)}$. Surprisingly, this shows that unlike time-bounded public-key cryptography, — where we must rely on *additional* computational assumptions, — space-bounded public-key cryptography can be proven *unconditionally*, while supporting *arbitrary gaps* between the powers of honest parties and the attacker. The price of allowing large values of m is that the round and communication complexities of the protocols grow correspondingly and we also provide a lower bound to show that this is inherent. In particular, this means that if we want limit ourselves to protocols with polynomial (round/communication) efficiency, then m is limited to be some arbitrarily large polynomial.

Before describing our results in detail, we start by describing the different variants of the BSM, to understand the gap that we crucially exploit between the model used in the lower bound of [16] and the model for our upper bounds.

1.1 Modeling Gap: Breaking the Quadratic Barrier

Many of the prior works in the BSM, including the original work of [29] and the lower bound of Dziembowski and Maurer [16], considered a more restricted model, that we refer to as the "traditional BSM" to disambiguate from the "streaming BSM". In particular, they consider a variant where a single long random string X is broadcast by a third party, and the honest users can store a small subset of n physical locations of X (chosen non-adaptively). The adversary can store arbitrary information about X, as long as the amount of information is bounded by m bits. After this occurs, the adversary's storage becomes unbounded, and the honest parties can run some additional protocol, whose overall space and communication complexity is bounded by n. Protocols in the traditional BSM readily translate into the streaming BSM, by having one of the users stream X as the first message of the protocol.[4]

Compared to the streaming BSM, the traditional BSM can be seen as imposing additional restrictions on the honest parties Alice and Bob, and giving more power to the space-bounded attacker Eve, as follows:

(a) *Restricting Number of "Long" Rounds.* We make a distinction between "long" rounds, in which one of the parties streams a long message consisting of more than m bits of data, versus "short" rounds, consisting of fewer than m bits of data. Note that Eve can store the entire message in a short round. The traditional BSM allows only a single "long" round — the very first round of the protocol.

(b) *Uniformly Random "Long Rounds".* The traditional BSM requires that a "long" round should simply stream a uniformly random string X. When true, such X is called a *randomizer string* [29], and can also come externally (e.g., from nature) rather than being sampled by the parties.

(c) *Local Computability for Alice/Bob.* In the streaming BSM, when a party Alice streams a long string X to an honest party Bob, then Bob is allowed to arbitrarily process all of X in a streaming manner, as long as not using more than n bits of space. The traditional BSM demands a stricter property of *n-Local Computatibility* (LC) [39]: The honest parties can only access at

[4] This holds generically in the case of KA. In the case of OT, where the participants can be malicious, it may not be generically safe to allow one of the parties to chose X instead of having it sampled by a trusted third party. However, it was safe to do so for all the protocols in the literature.

most n (a-prori non-adaptively chosen) *physical* locations of each string X sent during a “long” round. [5]

(d) *Unlimited Short-term Memory for Eve.* In the streaming BSM, the adversary Eve is streaming and only has m bits of memory throughout the execution of the protocol. In the traditional BSM, we only require that Eve stores at most m bits immediately after observing each “long” round, but we allow her to use unlimited short-term memory to process the round, and do not restrict her memory during “short” rounds.

Clearly, enforcing any of the restrictions (a)-(d) makes any upper bound stronger, and hence all protocols in the traditional BSM model also apply to the streaming BSM. Indeed, most previous constructions in the traditional BSM satisfied all of these additional properties. For example, the symmetric-key results of [1, 15, 28, 39] satisfied all of (a)-(d), as did the public-key results for key agreement and oblivious transfer of [5, 4, 10]. However, there were exceptions, pointing to the fact that these restrictions were not all seen as crucial. For example, the work of [9] required two “long” rounds, and therefore did not satisfy (a). Moreover, if one wanted to use OT as a sub-protocol in general MPC, then this would require running many sequential copies, meaning that even if the OT protocol satisfied (a), the resulting MPC would not.

More recently, the ground-breaking work of Raz et al. [35, 26, 36, 17] (presented in terms of time-space tradeoffs for learning parity), constructed elegant symmetric-key encryption schemes in the streaming BSM that crucially *do not* satisfy (b)-(d); see Section 1.4. The work of [19], then lifted the techniques of Raz et al. [35, 26, 36, 17] to build key agreement, oblivious transfer and bit commitment protocols in the streaming BSM, without satisfying (b)-(d). Nevertheless, the protocols of [19] have some advantages over prior works in the traditional BSM, such as smaller number of communication rounds, and perfect correctness.

Overall, looking at the literature, it appears that many works implicitly viewed the streaming BSM as the real conceptual goal, but ended up satisfying additional properties (a)-(d) that they incorporated into their formal model. This view seems to be shared by the more recent works of [35, 26, 36, 17, 19] that did not satisfy the additional properties, but still continued to refer to their model as the BSM, without carefully distinguishing between the variants. We continue in this vein, and view the streaming BSM as the main notion to strive for, while achieving the additional restrictions (a)-(d) can be seen as a nice bonus, but is not essential.

Moving to the lower bound of Dziembowski and Maurer [16], it turns out it critically used restriction (a), namely that there is only a single long round having large communication. Hence, to overcome the quadratic barrier imposed by [16], our protocols must use multiple long rounds.

Interestingly, we will be able to do so while still satisfying the additional restrictions (b)-(d). In particular, our protocols contains many long rounds, each

[5] For example, if local computability is demanded, parties cannot compute the parity of all the bits of X.

of which involves generating a long uniformly random string X, while the honest parties store some small set of at most n physical locations of X. The adversary is only restricted to storing at most m bits of information about each X sent in a long round, but gets unlimited memory otherwise (i.e. during the short rounds and for computing the functions that compresses each X into m bits). However, we will mostly view these additional features as secondary, and focus most of our discussion on the fully unrestricted streaming BSM. If follow-up works manage to get further improvements by also dropping the restrictions (b)-(d), much like the works of [35, 26, 36, 17, 19], this would be "fair game" and satisfy the main goal from our point of view.

To sum up, even though many prior works already departed from the traditional BSM and considered the streaming BSM as the main model, when it comes to public-key schemes, all prior works in the BSM were stuck at the quadratic gap between honest and adversarial storage. On the other hand, the quadratic lower bound of [16] does not extend to the streaming BSM, which opens the door for our results.

1.2 Our Results

As our main positive results, we design protocols for key agreement (KA), oblivious transfer (OT) and general multiparty computation (MPC) in the (n, m)-BSM, supporting up to an exponential gap between the honest user and adversary storage: $m = 2^{O(n)}$. This *qualitatively matches the positive results in the space-bounded symmetric-key setting*, albeit in (substantially) more rounds. In fact, we also show that large number of long rounds (and also overall large communication complexity) is essential when $m \gg n^2$, by non-trivially extending the lower bound of [16] to general BSM protocols. Details follow.

Key Agreement in BSM. Recall, the goal of a KA protocol is for Alice and Bob agree on a ℓ-bit key while talking over an authenticated-but-public channel. In Section 5, we show the following result in the (n, m)-BSM:

Theorem 1 (informal). *For any m, λ, there exists some $n_{min} = O(\log m + \lambda)$ such that for all $n \geq n_{min}$ there is an unconditionally secure key agreement protocol in the (n, m)-BSM that outputs an $\Omega(n)$-bit key and achieves security $2^{-\Omega(\lambda)}$. Furthermore:*

- *The number of rounds is $\widetilde{O}(\lceil m/n^2 \rceil \cdot \lambda)$.*
- *The communication complexity is $\widetilde{O}(m \lceil m/n^2 \rceil \cdot \lambda)$.*

Note that, although the adversary's storage bound m can even be exponentially larger than n, this comes at the cost of increasing the number of rounds and bits of communication. If we want the overall protocol to be polynomially efficient, then we must restrict m to be some arbitrarily large polynomial.

Oblivious Transfer and Beyond. As our second main result, we build an OT-protocol in (n, m)-BSM, achieving nearly the same parameters as our KA protocol from Theorem 1. Recall, in an OT protocol, sender Alice has two ℓ-bit messages $(\mathsf{msg}_0, \mathsf{msg}_1)$, and receiver Bob has a single choice bit $c \in \{0, 1\}$. At the end of the protocol, Alice should learn nothing, while Bob should learn msg_c, and get no information about msg_{1-c}. When ported to (n, m)-BSM, (1) honest Alice and Bob should use space at most n, (2) the privacy of choice bit c should hold even against malicious Alice with storage m, and (3) the privacy of m_{1-c} should hold even against malicious Bob with storage m.

In our work we will achieve receiver privacy guarantee (2) even against *unbounded* space sender, so we only rely on the BSM for sender privacy (3). Moreover, our protocol satisfies *simulation-based* security, with an efficient simulator. This means that our simulator only uses the attacker as a black-box and is efficient relative to the corresponding attacker. In contrast, prior OT works in the BSM [4, 9, 10, 19] all satisfied a weaker indistinguishability-based variant of sender-privacy, which roughly corresponds to inefficient simulation. The problem of having an efficient simulator was explicitly stated as an interesting and challenging open problem in [10]. Our result, formally proven in Section 6, is summarized below:

Theorem 2 (informal). *For any m, λ, there exists some $n_{min} = O(\log m + \lambda)$ such that for all $n \geq n_{min}$ there is an unconditionally secure OT protocol with efficient simulator in the (n, m)-BSM with message size $\Omega(n)$ and security (and correctness) errors $2^{-\Omega(\lambda)}$. Furthermore:*

- *The number of rounds is $\widetilde{O}(\lceil m/n^2 \rceil \cdot \mathrm{poly}(\lambda))$.*
- *The communication complexity is $\widetilde{O}(m \cdot \lceil m/n^2 \rceil \cdot \mathrm{poly}(\lambda))$.*
- *Receiver security holds even against a malicious sender with unbounded space.*

To generalize our result to general MPC, recall that OT is information-theoretically complete for general MPC [25, 23]. In Section 6.4, we observe that this result also extends to the (n, m)-BSM, provided we allow the honest parties' storage n, round complexity R, and communication complexity C to also polynomially-depend on the circuit size of the corresponding MPC functionality. Note that these parameters are completely independent of the adversary's storage bound m, which can still be arbitrarily (up to exponentially) larger than n. A similar observation that OT implies MPC in the BSM was already made in [10] and expanded on in [27], albeit in the setting where both the OT and the MPC only satisfy inefficient simulation.

We emphasize that the efficient simulation of our OT protocol is *critical* to achieve efficient simulation of the resulting MPC. If we apply our MPC to the special case of the zero-knowledge (ZK) functionality, we get the first ZK protocol in (n, m)-BSM with an *efficient simulator* and arbitrary gap between m and n. In contrast, if we only had indistinguishability-based OT, we would get ZK with an inefficient simulator (which is equivalent to witness indistinguishability), which is insufficient/uninteresting in many situations when the witness is unique. Indeed the prior works of [37, 2] constructed (non-interactive) witness indistinguishable proofs in the BSM, and explicitly left zero-knowledge as an open problem, which we resolve here in the interactive setting.

Round and Communication Lower Bound. As we already mentioned, circumventing the lower bound [16] requires more than one long round. Also, any protocol in the (n, m)-BSM clearly requires more than m bits of communication. However, our protocols in Theorems 1 and 2 are noticeably less efficient: they use $\Omega(m/n^2)$ rounds and $\Omega(m^2/n^2)$ communication. This begs the question of whether large round and communication complexities of our protocols are inherent. In particular, when $m \gg n^2$, should the number of rounds R grow with m and should the communication C be super-linear in m?

Unfortunately, we show that the answer is affirmative (see Theorem 22). Specifically, we show that any KA and OT protocols must satify $R \geq \Omega((m/n^2)^{1/2})$ and $C \geq \Omega(m \cdot (m/n^2)^{1/2})$. While leaving a non-trivial gap with our upper bounds $R = \widetilde{O}(m/n^2)$ and $C = \widetilde{O}(m^2/n^2)$ when $m \gg n^2$, it still shows that the number of rounds grows with m, and the communication must be super-linear in m. It is an interesting open question to close this quantitative gap between our lower and upper bounds.

Our basic lower bound above only holds for BSM protocols where the attacker Eve is allowed unlimited short-term memory, and is only subject to keeping an m-bit state in between rounds (i.e., condition (d)). However, we also non-trivially extend our lower bound to show that it can even handle fully streaming adversaries that are restricted to m-bits of memory throughout the protocol execution, at the cost of a weaker quantitative bound: $R \geq \Omega((m/n^2)^{1/3})$, $C \geq \Omega(m \cdot (m/n^2)^{1/3})$. It is also an interesting open question to close the quantitative gap between this bound and the previous one.

1.3 Our Techniques

Bit-Entropy Lemma. As a crucial tool in our KA and OT constructions, we rely on a new technical lemma for min-entropy (Lemma 12). On a high level, the lemma says that if a long string $X \in \{0,1\}^k$ has high min-entropy (e.g., because it was chosen uniformly at random and the adversary could only remember $m \ll k$ bits of information about it, in which case X denotes the conditional distribution), then many individual bits $X[i]$ of X must have non-trivial min-entropy. Specifically, if $\mathbf{H}_\infty(X) \geq \delta \cdot k$, we show that $\sum_{i \in [k]} \mathbf{H}_\infty(X[i]) \geq \rho \cdot k$, where we (optimally) relate ρ to δ. For example, when $\delta = \Omega(1)$, then $\rho = \Omega(1)$. The technical lemma relates to conceptually similar lemmas in [33, 39, 3], showing that random subsets of bits in X have a high entropy rates. It also relates to quasi chain-rules for min-entropy [14, 38]. However, to our knowledge, the single-bit version does not appear to follow easily from the prior results.

Key Agreement Protocol. The high-level idea for our KA protocol from Section 5.2 is surprisingly simple. For readers familiar with prior work on the bounded storage model, our protocol builds on a core template, introduced in [5] and further used in [4, 9, 10], which we adapt and extend to the interactive setting. The protocol consists of many rounds i, where Alice streams a $(k = 2m)$-bit random string X to Bob and remembers a single random location in the string $X[a]$. Similarly, as Bob receives the string X, he remembers a single random location $X[b]$. At the end of each round, Alice and Bob exchange their choice of locations a, b with each other; if $a = b$, they set $X[a] = X[b]$ as their shared key and terminate, else they erase all of their memory so far and go to the next round. Their storage only consists of a single index and is therefore $n = O(\log m)$. The probability of Alice and Bob agreeing in any round is $1/(2m)$ and therefore after $O(m)$ rounds they are likely to terminate. In the round i^* where they agree, the attacker can only remember m out of $2m$ bits of arbitrary information about the string X that was sent, and the choice of what information to remember is made before seeing Alice's and Bob's locations a, b. Therefore, the agreed upon location $X[a] = X[b]$ in that round has some constant amount of entropy from Eve's point of view.

The simple template above only outputs a 1-bit shared key, only guarantees that it has some low but non-trivial entropy from the point of view of the attacker (but does not guarantee that it is uniformly random), has a constant correctness error and and requires $O(m)$ rounds. However, it is easy to address these deficiencies. First, Alice/Bob can store $O(n/\log m)$ random locations (not just 1), which means they improve their odds of agreement in a given round from $1/m$ to roughly $O(n^2/m)$, to get round complexity $O(m/n^2)$ and communication complexity $O(m^2/n^2)$, respectively. Second, we can amplify security (and correctness) to ensure that the agreed upon key is $2^{-\lambda}$-statistically close to uniform, while simultaneously making the key longer (say, λ bits), by repeating the above $O(\lambda)$ times, and applying a randomness extractor to the $O(\lambda)$ agreed upon bit locations. Finally, once the symmetric-key is $O(\lambda)$ bits long, we amplify it to be $\Omega(n)$ bits, by adding an additional round, and using any of the optimal symmetric-key BSM protocols (e.g., [39]).

One crucial difference with the template of [5] and any single round in our interactive protocol is that, in our case, Alice and Bob agree on bits in a given round with very small probability $O(n^2/m) \ll 1$, as opposed to almost always agreeing in [5]. Our analysis is consequently significantly different, and builds on our bit-entropy lemma.

We also notice that, while our protocol takes many rounds (which we show to be inherent) and therefore does not satisfy restriction (a), it does satisfy the additional restrictions (b)-(d): each long string is truly random, Alice and Bob are "locally computatable", and security holds even if Eve has an unrestricted amount of short-term local memory, as long as she can only remember at most m bits of information after seeing each string X.

Oblivious Transfer Protocol. In an OT protocol, sender Alice has two messages $(\mathsf{msg}_0, \mathsf{msg}_1)$, and receiver Bob has a single choice bit $c \in \{0, 1\}$. At the end of the protocol, Alice should learn nothing, while Bob learns msg_c, and gets no information about msg_{1-c}.

Our oblivious transfer crucially relies on a tool called *interactive hashing* [31, 10]. This tool was also used to construct OT in the BSM by prior works [4, 9, 10] achieving a quadratic gap between the honest and adversarial storage. However, our protocol uses it in a substantially different way. In an interactive hashing protocol, a sender Bob has a random input $b \in [k]$, and at the end of the protocol, Alice can narrow down Bob's input to one of two possible choices b_0, b_1 such that $b \in \{b_0, b_1\}$, but Alice does not learn which of them it is; both options are equally likely. On the other hand, Bob cannot simultaneously control both of the values b_0, b_1 that Alice ends up with, and in particular he cannot cause both of them to land in some sparse subset $B \subseteq [k]$. Such interactive hashing protocols can be performed with 4 rounds of interactions and polylog(k) time/space. The security properties hold information-theoretically, even if the parties have unbounded computation and memory.

We now describe a simplified version of our OT protocol, which roughly corresponds to the case where honest users have $n = O(\log m)$ storage. We first rely on a component sub-protocol, which one can think of as an (imperfect) form of Rabin OT [34]: Alice outputs some bit r, and Bob either also outputs r or $\perp$, but Alice does not learn which of these occurred. We set the length of "long rounds" to $k = O(m \log(m))$:

- Alice and Bob choose random indices $a, b \leftarrow [k]$ respectively. Alice samples a random string $X \leftarrow \{0,1\}^k$ and sends it to Bob. Alice stores $X[a]$ and Bob stores $X[b]$.
- Alice and Bob run interactive hashing where Bob uses his index b. Alice learns that it is one of b_0, b_1.
- Alice checks if $a \in \{b_0, b_1\}$, and if not, then the parties go back to the beginning and try again. Else Alice sends a to Bob and outputs $r = X[a]$. Bob checks if $a = b$ and if so he outputs $r = X[b]$ else he outputs $\perp$.

The interactive hashing security ensures that even if Alice is malicious, she does not learn whether Bob outputs $\perp$ or r. On the other hand, even if Bob is malicious and has storage m, there is only a small $O(k/\log k)$ set of bad indices $B \subseteq [k]$ that he "knows" (have very small entropy given his state). The interactive hashing ensures that it's unlikely that both b_0, b_1 are in B, and Alice selects one of them at random (the one that matches her a). Therefore, in the execution where Alice accepts, with probability $\approx 1/2$, Alice's index satisfies $a \notin B$ and therefore Bob does not know $r = X[a]$.

To go from the above sub-protocol to full OT, we employ a variant of the trick of [7] to go from Rabin OT to the more standard 1-out-of-2 OT. The parties run the above sub-protocol for $t = 3\lambda$ iterations, where Alice outputs bits $(r_1, \ldots, r_t) \in \{0,1\}^t$ and Bob outputs $(r'_1, \ldots, r'_t) \in \{0, 1, \perp\}^t$ such that $r'_i \in \{r_i, \perp\}$ and roughly 1/2 of them are $\perp$, but Alice does not know which. Bob selects two disjoint subsets $I_0, I_1 \subseteq [t]$ of size λ each at random, subject to I_c only containing values i for which $r'_i \neq \perp$. Alice applies an extractor on the values r_{I_0}, r_{I_1} and uses the outputs to one-time-pad her messages $\mathsf{msg}_0, \mathsf{msg}_1$. This allows Bob to recover msg_c. It's easy to see that the sets I_0, I_1 look identically distributed to Alice and so she does not learn Bob's choice bit c. On the other hand, since Bob only knows roughly $\frac{t}{2} = \frac{3\lambda}{2}$ of the values r_i, at least one of r_{I_0}, r_{I_1} must contain roughly $\frac{\lambda}{2}$ values that Bob does not know, and hence the corresponding extracted string will blind the message.

Note that in our scheme, security against an adversarial Bob (receiver) relies on him having bounded storage m, but security against an adversarial Alice (sender) does not impose any restrictions on her storage. The overall protocol requires $\widetilde{O}(m \cdot \lambda)$ rounds to terminate and $\widetilde{O}(m^2\lambda)$ communication. Our full protocol generalizes the above to settings where honest users have larger storage n to get $\widetilde{O}(\lceil m/n^2 \rceil \cdot \lambda)$ rounds and $\widetilde{O}(m \cdot \lceil m/n^2 \rceil \cdot \lambda)$ communication. This requires additional technical ideas to perform interactive hashing on sets of indices rather than just a single index; see Section 6.

One issue with the above idea, and indeed all prior constructions of OT in the BSM [4, 9, 10, 19], is that it only satisfies a weak form of indistinguishability-based security, which is equivalent to security with an inefficient simulator. In particular, to simulate an adversarial Bob, we need to figure out his choice bit c, which requires figuring out which locations $X[a]$ he "knows" and which he does not. This can be done inefficiently (and non-black-box) by looking at Bob's state after processing X and figuring out the conditional entropy of each bit of X given the state; but there seems to be no hope to make this process efficient. We show how to overcome this via an efficient rewinding-based simulation strategy. The simulator forks off many copies of the interactive hashing protocol and figures out which indices show up as one of Alice's outputs with *high frequency*. We show that this serves as a good proxy for the indices that Bob knows – since he only knows $X[a]$ for very few locations a, he has to "play" such locations with high frequency if he wants to have a good chance of Alice selecting them. Therefore, by using the efficiently computable set of high-frequency indices as a proxy for the inefficiently computable set of indices that Bob knows, we can efficiently extract Bob's choice bit c.

Lower Bound. We prove a lower-bound for KA and, since OT directly gives KA, this also implies an identical lower bound for OT. Let us first recall the main intuition of the DM lower bound [16]. Let $m > n$ be the storage size of the adversary, and suppose the first message of the protocol is some large message M, potentially of size $|M| \gg m$ much larger than the adversary's storage. In the real protocol, the honest parties Alice and Bob respectively compute states s_A and s_B after processing M. DM shows that there exists some compact information s_E^* of size m, which (1) is publicly-computable given M, and (2) *decorrelates* the states s_A and s_B of Alice and Bob in the following sense: conditioned on s_E^*, the users' states s_A and s_B only share a low amount of mutual information, bounded by n^2/m. Therefore, if $m = O(n^2)$ is sufficiently large, the information shared between Alice and Bob conditioned on the adversary's view becomes too small (much less than 1 bit) for them to agree on a shared random key.

One obstacle towards extending DM to the interactive setting is that, even if the mutual information created in each round is very small $O(n^2/m)$, with sufficiently many rounds it can add up. Indeed, this is exactly what our upper bound exploits, and why one can allow large gaps between m and n with a large numbers of rounds! For our lower bound on rounds and communication, we want to show that this is essentially the best that one can do. There are two main obstacles. Firstly, the DM approach only works if Alice and Bob do not share any mutual information in the first place. This is true at the beginning of the protocol, which results in a candidate adversarial strategy for the first round of the protocol. But it is not clear whether it extends to any intermediate round within the protocol execution, where Alice and Bob managed to already get some, albeit small, amount of mutual information.[6] Moreover, a naive attempt would be to have Eve compute an appropriate s_E^* for every round, but then Eve would need to store all of these values throughout the duration of the protocol, thus blowing up her storage.

Instead, we approach the DM core idea from a different angle, by thinking of it as a *round reduction* step that allows us to convert an R round protocol into an $R-1$ round protocol, with only a small loss in correctness and security. In particular, instead of having Alice send the long message M to Bob in the first round, we remove the first round entirely, and have Bob do the following: (1) sample M as Alice would, (2) (inefficiently) sample s_E^* given M as the adversary in DM, and (3) sample his state s_B conditioned on s_E^*; (4) use it to compute the next message M', and (5) send (s_E^*, M') as the new message to Alice. Alice then: (6) samples s_A conditioned on s_E^*; and (7) processes M' using s_A, as she would have done originally. Note that Alice and Bob are now inefficient, with unlimited short-term memory to process each round, but only keep short n-bit states between rounds, similar to feature (d) of Eve.[7]

We claim that the round reduction step preserves correctness and security up to some small loss. This holds because the original states s_A, s_B had small mutual information conditioned on s_E^*, which implies that they are statistically close to independent. Therefore, the new way of sampling s_A, s_B truly independently conditioned on s_E^* only introduces a small statistical error. On the other hand, any attack Eve can perform on the new protocol by observing both of the values (s_E^*, M') sent by Bob at the same time, she could have also performed originally by computing s_E^* from Alice's original message M, storing s_E^* locally in her m-bit state (here we crucially rely on it being small), and then performing the same computation on the values (s_E^*, M'), once Bob sends M'.

By performing the round-reduction steps iteratively, we eventually get a 0-round key agreement protocol, which leads to a contradiction. However, each time we perform the round-reduction step we incur some statistical error $\sqrt{n^2/m}$. The square-root comes from using Pinsker's inequality to convert from mutual information to statistical distance. Therefore, we only end up with a secure pro-

[6] Indeed, it is not true in general that their mutual information can only increase by a small amount in each round; once Alice and Bob share even a small amount of mutual information (e.g., they share a short extractor seed, perhaps even only with small probability), they may be able to leverage it to derive much more mutual information in just one additional round (e.g., send a long message and extract).

[7] Note that allowing Alice and Bob to be stronger makes the resulting lower bound stronger as well.

tocol at the end, if the original protocol has $R = O(\sqrt{m/n^2})$ rounds, which gives our lower bound on rounds $R \geq \Omega(\sqrt{m/n^2})$. Note that this also gives a lower bound on communication C since $C \geq R$. However, we can get a stronger lower bound of $C \geq \Omega(m \cdot \sqrt{m/n^2})$ by showing how to remove "small" rounds (i.e., having communication smaller than m) for free, without any loss in correctness/security. We refer to Section 7.2 for more details.

As mentioned previously, this lower bound only rules out protocols secure against strong attackers Eve who have access to unbounded short-term memory to process each round, while storing m bits between rounds. We further adapt the techniques above to handle *fully streaming* adversaries, that are restricted to m bits of memory throughout the protocol. The main observation is that the only step in the round reduction procedure that requires Eve to have unbounded short-term memory is sampling s_E^* given M. We first observe that this step can be performed in a streaming manner using small local memory, as long as Alice and Bob are streaming algorithms with small local memory. However, even if the latter was the case in the initial protocol, once we start removing rounds, we required Alice and Bob to have large local memory to run Eve's attack. This turns into a recursive analysis, where the memory that Alice and Bob need to run the protocol after removing R rounds, depends on the memory Eve needs to attack on the protocol after removing $R-1$ rounds, which depends on the memory Alice and Bob need to run the protocol after removing $R-1$ rounds etc. By carefully analyzing this recursion, we show that Eve's short-term memory can be bounded to only be a factor of R larger than the previous bound we had on her long-term memory, which yields our new new quantitatively weaker bounds of $R \geq \Omega((m/n^2)^{1/3})$ and $C \geq \Omega(m \cdot (m/n^2)^{1/3})$ for the fully streaming model. We refer to Section 7.4 for more details.

1.4 Related Work

We already extensively mentioned the prior work on the symmetric-key BSM [29, 1, 15, 28, 39, 35, 26, 36, 17] and the public-key BSM models [5, 4, 9, 21, 10, 19]. In particular, the work of [35, 26, 36, 17] constructed "reusable" n-bit-key symmetric-key encryption schemes, capable of encrypting exponentially many b-bit messages, where an individual ciphertext is "only" $O(mb/n)$ bits long.[8] When $b \ll n$, this is a huge saving compared to the prior symmetric-key schemes in the BSM, where each individual ciphertext had size greater than m, irrespective of message length. Interestingly, these works did not satisfy restrictions (b)-(d), critically using full features of the streaming BSM.

In the context of proof systems, [37, 2] constructed non-interactive witness indistinguishable proofs secure against memory-bounded streaming verifiers, allowing arbitrary gap between the values n and m. In contrast, the proofs systems constructed in this work are full zero-knowledge, with efficient simulation, but

[8] This is optimal, as otherwise Eve is capable of storing more than n/b ciphertexts in its memory, allowing the parties to encrypt more than $b \cdot n/b = n$ bits of information using an n-bit key, contradicting Shannon lower bound.

use many rounds of interaction. In a related vein, a very recent work of [20] considered the notion of "disappearing cryptography" in the (streaming) BSM. Here, a component of the scheme (e.g., a ciphertext, signature, proof or program) is streamed bit by bit. The space-bounded receiver can get the functionality of the system once, after which the object "disappears" for subsequent use.

The work of [30] designed novel "timestamping" schemes in the (traditional) BSM. Here space-bounded sender and receiver have access to a long randomizer string X: the sender will timestamp a given document D at time t, and the receiver will prepare to verify D (which is yet unknown). The sender can then prove the timestamping of D to the receiver at a much later time, and the receiver is guaranteed that the sender is unable to timestamp a "very different" (i.e., high-entropy) document D'.

Finally, we mention the seminal works of [32, 33] in the context of designing pseudorandom generators fooling space-bounded distinguishers. Unlike the BSM setting, the memory n of the generator must be necessarily higher than the memory m of the the distinguisher, and the works of [32, 33] come very close to this bound, unconditionally. In a similar vein, the work of [24] constructs deterministic randomness extractors for space-bounded sources of randomness.

2 Preliminaries

Notation. When X is a distribution, or a random variable following this distribution, we let $x \leftarrow X$ denote the process of sampling x according to the distribution X. If X is a set, we let $x \leftarrow X$ denote sampling x uniformly at random from X. We use the notation $[k] = \{1, \ldots, k\}$. If $x \in \{0,1\}^k$ and $i \in [k]$ then we let $x[i]$ denote the i'th bit of x. If $s \subseteq [k]$, we let $x[s]$ denote the list of values $x[i]$ for $i \in s$.

Statistical Distance. Let X, Y be random variables with supports S_X, S_Y, respectively. We define their *statistical difference* as

$$\mathbf{SD}(X,Y) = \frac{1}{2} \sum_{u \in S_X \cup S_Y} \left| \Pr[X = u] - \Pr[Y = u] \right| .$$

We write $X \approx_\varepsilon Y$ to denote $\mathbf{SD}(X,Y) \leq \varepsilon$.

Predictability and Entropy. The *predictability* of a random variable X is $\mathbf{Pred}(X) \stackrel{\mathsf{def}}{=} \max_x \Pr[X = x]$. The *min-entropy* of a random variable X is $\mathbf{H}_\infty(X) = -\log(\mathbf{Pred}(X))$. Following Dodis et al. [12], we define the conditional predictability of X given Y as $\mathbf{Pred}(X|Y) \stackrel{\mathsf{def}}{=} \mathbb{E}_{y \leftarrow Y}\left[\mathbf{Pred}(X|Y = y)\right]$ and the (average) conditional min-entropy of X given Y as: $\mathbf{H}_\infty(X|Y) = -\log\left(\mathbf{Pred}(X|Y)\right)$. Note that $\mathbf{Pred}(X|Y)$ is the success probability of the optimal strategy for guessing X given Y.

Lemma 3 ([12]). *For any random variables X, Y, Z where Y is supported over a set of size T we have $\mathbf{H}_\infty(X|Y,Z) \leq \mathbf{H}_\infty(X|Z) - \log T$.*

Lemma 4 ([12]). *For any random variables X, Y,for every $\varepsilon > 0$ we have*

$$\Pr_{y \leftarrow Y}[\mathbf{H}_\infty(X|Y = y) \geq \mathbf{H}_\infty(X|Y) - \log(1/\varepsilon)] \geq 1 - \varepsilon.$$

Lemma 5. *If X and Y are independent conditioned on Z then $\mathbf{H}_\infty(X|Y) \geq \mathbf{H}_\infty(X|Y, Z) \geq \mathbf{H}_\infty(X|Z)$.*

Lemma 6. *If X and Y are independent conditioned on Z then $\mathbf{H}_\infty(X, Y|Z) \geq \mathbf{H}_\infty(X|Z) + \mathbf{H}_\infty(Y|Z)$.*

Shannon Entropy. The Shannon entropy of a random variable X is $\mathbf{H}(X) \overset{\mathsf{def}}{=} \mathbf{E}_{x \leftarrow X}[-\log(\Pr[X = x])]$. The conditional Shannon entropy of X given Y is $\mathbf{H}(X|Y) \overset{\mathsf{def}}{=} \mathbf{E}_{y \leftarrow Y}\,\mathbf{H}(X|Y = y) = \mathbf{E}_{(x,y) \leftarrow (X,Y)}[-\log(\Pr[X = x|Y = y])]$.

For $0 \leq p \leq 1$ we define the binary entropy function $\mathbf{h}(p) \overset{\mathsf{def}}{=} \mathbf{H}(B_p)$, where B_p is a Bernoulli variable that outputs 1 with probability p and 0 with probability $1 - p$.

Lemma 7. *For any random variables X, Y, we have: $\mathbf{H}_\infty(X|Y) \leq \mathbf{H}(X|Y)$.*

Extractors. We review the notion of randomness extractors and known parameters.

Definition 8 ((Strong, Average-Case) Seeded Extractor [33]). *We say that an efficient function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ *is an* (α, ε)-extractor *if for all random variables (X, Z) such that X is supported over $\{0,1\}^n$ and $\mathbf{H}_\infty(X|Z) \geq \alpha$ we have $\mathbf{SD}((Z, S, \mathsf{Ext}(X; S))\ ,\ (Z, S, U_\ell)) \leq \varepsilon$ where S, U_ℓ are uniformly random and independent bit-strings of length d, ℓ respectively.*

Theorem 9 ([22]). *There exist an (α, ε)-extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^\ell$ as long as $\alpha \geq \ell + 2\log(1/\varepsilon)$. Furthermore, such an extractor can be computed in $O(n)$ time and space.*

Definition 10 (BSM Extractor [39]). *We say that an efficient function* $\mathsf{BSMExt} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^\ell$ *is an* (n, m, ε)-BSM extractor *if:*

- *Given* $\mathsf{seed} \in \{0,1\}^d$ *initially stored in memory, it is possible to compute* $\mathsf{BSMExt}(x; \mathsf{seed})$ *given streaming access to $x \in \{0,1\}^k$ using at most n bits of total memory. Moreover, it can be done while only accessing at most n locations (chosen non-adaptively) in the string x.*
- BSMExt *is an (α, ε)-extractor (Definition 8) for $\alpha = k - m$.*

Note that a BSM Extractor gives a simple one-round protocol (n, m)-BSM protocol where Alice and Bob start with a uniformly random shared key key_0 of some small size d and derive a new shared key $\mathsf{key}_1 \in \{0,1\}^\ell$ of a larger size $\ell > d$. Alice just streams a random $x \in \{0,1\}^k$ to Bob and both parties compute $\mathsf{key}_1 = \mathsf{BSMExt}(x; \mathsf{key}_0)$. Security holds since the adversary can only store m-bits of information about x so it has $\alpha \geq k - m$ bits of entropy conditioned on the adversary's view, and key_0 acts as a random seed which is a-priori unknown to the adversary. Therefore $\mathsf{key}_1 = \mathsf{BSMExt}(x; \mathsf{key}_0)$ is ε-close to uniform given the adversary's view of the protocol.

Theorem 11 ([39]). *For any $m \geq \ell,\lambda$, there is a (n,m,ε)-BSM extractor $\mathsf{BSMExt} : \{0,1\}^k \times \{0,1\}^d \to \{0,1\}^\ell$ with $n = O(\ell + \lambda + \log m)$, $\varepsilon = 2^{-\Omega(\lambda)}$, $k = O(m + \lambda\log(\lambda))$, $d = O(\log m + \lambda)$.*

3 Bit-Entropy Lemma

We prove a new lemma showing that if X has sufficiently high min-entropy, then many individual bits $X[i]$ have sufficiently high min-entropy as well.

For $q \in [0,1]$, we define $\mathbf{h}_+^{-1}(q)$ to be the unique value p such that $.5 \leq p \leq 1$ and $\mathbf{h}(p) = q$, where $\mathbf{h}$ is the binary entropy function defined above.

Lemma 12. *Assume X, Y are random variables, where X is distributed over $\{0,1\}^k$. Let $X[i]$ denote the i'th bit of X. If $\mathbf{H}_\infty(X|Y) \geq \delta k$ the following 3 statements hold:*

1. $\sum_i \mathbf{Pred}(X[i] \mid Y) \leq \mathbf{h}_+^{-1}(\delta)k$.
2. $\sum_i \mathbf{H}_\infty(X[i] \mid Y) \geq -\log(\mathbf{h}_+^{-1}(\delta))k$.
3. *If I is uniformly random over $[k]$ and independent of X, Y then* $\mathbf{H}_\infty(X[I] \mid Y, I) \geq -\log(\mathbf{h}_+^{-1}(\delta))$.

Proof. We have:

$$\delta k \leq \mathbf{H}_\infty(X|Y) \leq \mathbf{H}(X|Y) = \sum_{i\in[k]} \mathbf{H}(X[i] \mid X[1], ..., X[i-1], Y) \leq \sum_{i\in[k]} \mathbf{H}(X[i] \mid Y).$$

Therefore

$$\delta \leq \mathop{\mathbf{E}}_{i\leftarrow[k], Y\leftarrow y} \mathbf{H}(X[i] \mid Y = y).$$

Since $\mathbf{h}_+^{-1}$ is a decreasing and concave function, this means:

$$\begin{aligned}
\mathbf{h}_+^{-1}(\delta) &\geq \mathbf{h}_+^{-1}\left(\mathop{\mathbf{E}}_{i\leftarrow\{0,1\}^k, y\leftarrow Y} \mathbf{H}(X[i] \mid Y = y)\right) \\
&\geq \mathop{\mathbf{E}}_{i\leftarrow[k], y\leftarrow Y} \mathbf{h}_+^{-1}(\mathbf{H}(X[i] \mid Y = y)) \\
&\geq \mathop{\mathbf{E}}_{i\leftarrow[k], y\leftarrow Y} (\max_{b\in\{0,1\}} \Pr[X[i] = b \mid Y = y]) \\
&\geq \mathop{\mathbf{E}}_{i\leftarrow[k]} \mathbf{Pred}(X[i]|Y).
\end{aligned}$$

This proves the first part of the theorem. Also the third part of the theorem follows since $\mathbf{H}_\infty(X[I] \mid Y, I) = -\log(\mathbf{E}_{i\leftarrow I} \mathbf{Pred}(X[I] \mid Y, I = i)) = -\log(\mathbf{E}_{i\leftarrow[k]} \mathbf{Pred}(X[i] \mid Y))$. The second part follows since $(-\log)$ is a decreasing and convex function so

$$\begin{aligned}
-\log(\mathbf{h}_+^{-1}(\delta)) &\leq -\log\left(\mathop{\mathbf{E}}_{i\leftarrow[k]} \mathbf{Pred}(X[i] \mid Y)\right) \\
&\leq \mathop{\mathbf{E}}_{i\leftarrow[k]} -\log(\mathbf{Pred}(X[i] \mid Y)) \\
&\leq \mathop{\mathbf{E}}_{i\leftarrow[k]} \mathbf{H}_\infty(X[i] \mid Y)
\end{aligned}$$

□

Remark 13. To the best of our knowledge, the "bit-prediction" lemma above is new, as it talks about individual bit prediction; as opposed to "subkey-predcition" lemma studied in prior BSM literature [33, 39, 3], which talked about simultaneously predicting a large subset of bits. It also does not appear to follow directly from quasi chain-rules for min-entropy [14, 38] that have a large loss in parameters that does not appear to give any non-trivial bounds in the bit setting. We also remark that our parameters are tight, as can be seen by taking X to be the uniform distribution over a hamming ball of radius pk, where $p = 1 - \mathbf{h}_+^{-1}(\delta) \leq 1/2$. The volume of this ball is roughly $2^{\mathbf{h}(p)k} = 2^{\delta k}$, so $\mathbf{H}_\infty(X) = \delta k$. Yet, each bit of X can be predicted with probability at least $1 - p = \mathbf{h}_+^{-1}(\delta)$.

Lemma 14. *For any* $0 < \varepsilon \leq 1$ *there is a* $\delta = \Omega(\varepsilon^2)$ *such that* $-\log(\mathbf{h}_+^{-1}(1 - \delta)) = (1 - \varepsilon)$*.*

Proof. Given ε we can solve:

$$\begin{aligned} & -\log(\mathbf{h}_+^{-1}(1-\delta)) = 1 - \varepsilon \\ \Rightarrow \quad & \mathbf{h}_+^{-1}(1-\delta) = 2^\varepsilon/2 \\ \Rightarrow \quad & 1 - \delta = \mathbf{h}(2^\varepsilon/2) = \mathbf{h}(1/2 + \Theta(\varepsilon)) = 1 - \Theta(\varepsilon^2). \end{aligned}$$

where we rely on the bound $2^\varepsilon = (1 + \Theta(\varepsilon))$ and $\mathbf{h}(1/2 + \Theta(\varepsilon)) = 1 - \Theta(\varepsilon^2)$ (e.g. [6, Theorem 2.2]). □

4 Bounded Storage Model

A (n, m)-*bounded storage model (BSM)* protocol, is parametrized by a bound n on the memory of the honest parties, and a bound $m > n$ on the memory of the adversary. Communication between parties occurs in rounds where one party sends data to another party. Honest parties send and receive data in a *streaming* manner, by generating/reading the stream one bit at a time, while only using n bits of memory overall. The adversary is also a streaming algorithm with m bits of memory.

For all our constructions, we will satisfy additional properties, corresponding to properties (b)-(d) discussed in the introduction. The protocol consists of two types of rounds: "short rounds" are of size is $< n$, and can be fully generated, sent, and processed by the honest parties using only n bits bits of memory, without needing to be streamed one bit at a time, while "long rounds" are of size $> m$.[9] Our protocols satisfy the following additional properties:

[9] We will allow ourselves to split up the protocol into rounds arbitrarily, and may have two (or more) adjacent rounds where the same party A talks to party B.

- *Uniformly Random "Long Rounds".* Each long round consists of a uniformly random string x generated by some party A and sent to party B.
- *Local Computability for Honest Parties.* In each long round, the honest parties only read a small set of $< n$ locations of x and use these to update their state, while using only n bits of memory in total. Furthermore, the set of locations accessed is chosen non-adaptively at the beginning of the round, before seeing any bits of x.
- *Unlimited Short-term Memory for Adversary.* The adversary can generate and read the entire long round of communication at once, and can use unlimited amounts of short-term memory during this process, but can only store a compressed m-bit state immediately after the end of each long round. There are no restrictions on the adversary's memory during/after short rounds.

5 Key Agreement

5.1 Definition

A key agreement protocol in the (n, m)-BSM with security ε is a protocol between two honest users Alice and Bob with memory bound n. At the end of the protocol Alice and Bob outputs values $\mathsf{key}_A, \mathsf{key}_B \in \{0,1\}^\ell$ respectively. For correctness, we require that when the protocol is executed honestly then $\Pr[\mathsf{key}_A = \mathsf{key}_B] = 1$. For security, we consider a passive BSM adversary Eve with memory bound m. Let view_{Eve} denote Eve's final state at the end of the protocol execution. We require that

$$(\mathsf{view}_{Eve}, \mathsf{key}_A) \approx_\varepsilon (\mathsf{view}_{Eve}, \mathsf{key}^*)$$

where $\mathsf{key}^* \leftarrow \{0,1\}^\ell$ is chosen uniformly at random and independently of the protocol execution.

5.2 Construction

Theorem 15. *For any $m \geq \ell, \lambda$ there is some $n_{min} = O(\lambda + \ell + \log m)$ such that for all $n > n_{min}$ there is a key agreement protocol in the (n, m)-BSM that outputs an ℓ-bit key and has security $\varepsilon = 2^{-\Omega(\lambda)}$. The round complexity of the protocol is $O(\lceil (m/n^2) \rceil \cdot \lambda \cdot \mathrm{polylog}(m))$ and the communication complexity $O(m \cdot \lceil (m/n^2) \rceil \cdot \lambda \cdot \mathrm{polylog}(m + \lambda))$.*

Proof. We present the key agreement protocol between Alice and Bob. We refer to the full version [13]for a proof.

Construction. Given m, λ, ℓ we define additional parameters as follows.

- Let $k = 2m$.
- Let $d_1 = O(\lambda + \log m)$ and $n_1 = O(\lambda + \log m + \ell)$ and $k' = O(m + \lambda \log(\lambda))$ be some values such that there is a $(n_1, m, \varepsilon = 2^{-\Omega(\lambda)})$-*BSM extractor* $\mathsf{BSMExt} : \{0,1\}^{k'} \times \{0,1\}^{d_1} \to \{0,1\}^\ell$ per Theorem 11.

- Let $t = O(\lambda + \log m)$ and $d_0 = O(t), n_0 = O(t)$ be some value such that there is a $(t/10, \varepsilon = 2^{-\Omega(\lambda)})$-extractor $\mathsf{Ext} : \{0,1\}^t \times \{0,1\}^{d_0} \to \{0,1\}^{d_1}$ that can be computed using n_0 space per Theorem 9.
- Define $n_{min} = \max(n_0, n_1, 2t + \lceil \log k \rceil + 1) = O(\lambda + \log m + \ell)$.
- For any $n \geq n_{min}$, define $\tilde{n} = \lfloor (n-t)/(\lceil \log k \rceil + 1) \rfloor = \Omega(n/\log m)$.

The protocol works as follows.

1. Set $i := 0$. Repeat the following until $i = t$:
   (a) Alice and Bob select uniformly random subsets $s_A, s_B \subseteq [k]$ of size $|s_A| = |s_B| = \widetilde{n}$ respectively.
   Alice streams a uniformly random string $x \leftarrow \{0,1\}^k$ to Bob.
   Alice stores $x[s_A]$ while Bob stores $x[s_B]$.
   (b) Bob sends s_B to Alice.
   (c) If $s_A \cap s_B \neq \emptyset$ then Alice selects a random index $j \leftarrow s_A \cap s_B$ and sends j to Bob.
   Both Alice and Bob set $r_i = x[j]$ and increment $i := i + 1$.
   Else if $s_A \cap s_B = \emptyset$ then Alice simply sends $j = \bot$ to Bob.
2. Alice and Bob set $r := (r_1, \ldots, r_t)$.
   Alice sends a random $\mathsf{seed}_0 \leftarrow \{0,1\}^{d_0}$ to Bob and both of them compute $\mathsf{seed}_1 = \mathsf{Ext}(r; \mathsf{seed}_0)$.
3. Alice streams a uniformly random string $x \leftarrow \{0,1\}^{k'}$ to Bob and both parties compute $\mathsf{key} = \mathsf{BSMExt}(x; \mathsf{seed}_1)$.

□

6 Oblivious Transfer and Multiparty Computation

6.1 Definition of Oblivious Transfer

We define oblivious transfer (OT) in the BSM via a real/ideal framework. In the ideal model the sender (Alice) gives two messages $(\mathsf{msg}_0, \mathsf{msg}_1) \in (\{0,1\}^\ell)^2$ to an ideal functionality $\mathcal{F}_{OT}$ and the receiver (Bob) gives a bit $c \in \{0,1\}$. The ideal functionality $\mathcal{F}_{OT}$ gives msg_c to the receiver and gives nothing to the sender.

A protocol Π realizes $\mathcal{F}_{OT}$ in the (n,m)-BSM with security ε if:

- Π can be executed by honest parties with n-bit memory.
- There exists an efficient black-box simulator $\mathsf{Sim}^{\mathcal{A}}$ that runs in time $\mathrm{poly}(n, m, \lambda = \log(1/\varepsilon))$ with black-box (rewinding) access to the adversary $\mathcal{A}$, such that for any (inefficient) BSM-adversary $\mathcal{A}$ with m-bit state corrupting either the sender or the receiver and for any choice of inputs $\mathcal{Z} = (\mathsf{msg}_0, \mathsf{msg}_1, c)$ from the environment, we have

$$\mathsf{REAL}_{\mathcal{A},\Pi,\mathcal{Z}} \approx_\varepsilon \mathsf{IDEAL}_{\mathsf{Sim}^{\mathcal{A}},\mathcal{F}_{OT},\mathcal{Z}}$$

where we define the distributions:

$\mathsf{REAL}_{\mathcal{A},\Pi,\mathcal{Z}}$: denotes the real execution of Π with the adversary $\mathcal{A}$ taking on the role of either the sender or the receiver while the honest party uses the input specified by $\mathcal{Z}$; the output of the distribution consists of the output of $\mathcal{A}$ together with the inputs/outputs of the honest party.

$\mathsf{IDEAL}_{\mathsf{Sim}^{\mathcal{A}},\mathcal{F}_{OT},\mathcal{Z}}$: denotes the ideal execution of $\mathcal{F}_{OT}$ with an ideal-adversary $\mathsf{Sim}^{\mathcal{A}}$ taking on the same role as $\mathcal{A}$, while the honest party uses the input specified by $\mathcal{Z}$; the output of the distribution consists of the output of $\mathsf{Sim}^{\mathcal{A}}$ together with the inputs/outputs of the honest party.

We further say that the protocol is secure against an *unbounded-memory* sender (resp. receiver) if we can drop the requirement on the storage of $\mathcal{A}$ when it corrupts the sender (resp. receiver).

We say that the protocol is only secure with *inefficient simulation*, if we drop the requirement on the efficiency of the simulator. Our default notion will be *efficient simulation.*

On Efficient Simulation. Note that we require efficient simulation even though the adversary may be computationally unbounded. This may seem strange at first, but is natural and is analogous to (e.g.,) requiring an efficient simulator for statistical Zero Knowledge proofs [18] or for information-theoretically secure MPC protocols. In particular, the definition is agnostic to whether or not the adversary is efficient, but ensures that the adversary cannot learn anything in the real world that it could not also learn with only polynomially *more* computational power in the ideal world. The need for an efficient simulator is crucial when leveraging OT to construct other more complex functionalities, as we will do in Section 6.4. For example, we can use our OT in the BSM to construct zero-knowledge (ZK) proofs in the BSM. If the OT simulator were inefficient, the resulting ZK proof would only be inefficiently simulatable (equivalently, would only be witness indistinguishable), which is completely meaningless in many scenarios where the witness is unique; the prover may as well just send the witness in the clear.

On the other hand, our simulator does not have bounded storage and can use more memory than the adversary. This naturally corresponds to the idea that having some a-priori (polynomial) bound on storage is only assumed to be a limitation in the real world, and is a useful limitation in helping us build secure protocols, but is not a fundamental restriction that we need to also preserve for the ideal-world adversary interacting with the ideal functionality.

6.2 Interactive Hashing

Basic Interactive Hashing. In an interactive hashing protocol a sender Bob has an input $u \in [k]$. The goal of the protocol is for Alice to narrow down Bob's input to one of two possible choices u_0, u_1 such that $u = u_b$ for one of $b = 0$ or $b = 1$, but Alice does not learn which. In particular, even if Alice acts maliciously, when Bob chooses his input $u \leftarrow [k]$ at random, then both choices of b appear equally likely from Alice's point of view. On the other hand, although Bob can choose an arbitrary input u and ensures $u = u_b$ for some $b \in \{0, 1\}$, he cannot control the "other" value u_{1-b} too much. In particular, even if Bob is malicious during the protocol, for any sufficiently sparse subset $B \subseteq [k]$, it is highly unlikely that *both* of u_0, u_1 are contained in B.

Definition 16. *An* interactive hashing protocol *is a protocol between a public-coin randomized Alice (receiver) and a deterministic Bob (sender). Alice has no input and Bob has some input $u \in [k]$. At the end of the protocol, we denote the transcript (h, v), consisting of all the random messages h sent by Alice and all the responses v sent by Bob. We can think of h as defining a hash function that maps Bob's input u to his set of responses $v = h(u)$. The protocol has the following properties:*

- *2-to-1 Hash: Every possible choice of Alice's messages results in a hash function h which is 2-to-1, meaning that for every v in the image of h has exactly two pre-images: $|h^{-1}(v)| = 2$.*
- *(α, β)-Security: For any set $B \subseteq [k]$ of size $|B| \leq \beta \cdot k$, if Alice follows the protocol honestly and Bob acts arbitrarily resulting in some transcript (h, v) such that $\{u_0, u_1\} = h^{-1}(v)$ then $\Pr[\{u_0, u_1\} \subseteq B] \leq \alpha$.*

Note: the 2-to-1 hash property ensures that, if Bob chooses $u \leftarrow [k]$ uniformly at random and acts honestly during the protocol, then even if Alice acts maliciously resulting in some transcript (h, v) at the end of the protocol, if we define $\{u_0, u_1\} = h^{-1}(v)$ such that $u = u_b$, Alice cannot distinguish between b and $1 - b$.

We have constructions of interactive hashing (with security against arbitrary Alice and Bob, without any bound on their memory):

Theorem 17 **([31, 10]).** *There is an 4-round interactive hashing protocol with (α, β)-security for any $\beta < 1$ with $\alpha = O(\beta \log k)$. Furthermore, the execution of the protocol and the computation of h^{-1} can be done in* polylog$k$ *time and space.*

Definition of Set Interactive Hashing. Here, we extend the notion of interactive hashing to the case where the sender Bob has an entire set of inputs $s_B \subseteq [k]$. Alice has her own set of inputs $s_A \subseteq [k]$. The goal of the protocol is to ensure that when there is a value in the intersection $s_A \cap s_B$ then there is a good chance that Alice will accept and output some value $u \in S_A$, in which case it then holds with probability 1/2, that $u \in S_B$ and Bob accepts and outputs it, while with probability 1/2 Bob rejects. Alice should not learn which of these two cases occur, even if she acts maliciously. On the other hand, even if Bob is malicious, he cannot have too much control over the value that Alice outputs: for any sufficiently sparse set $B \subseteq [k]$, he cannot ensure that the value u that Alice outputs (conditioned on her accepting) is in the set B with probability much higher than 1/2.

Definition 18. *In a set interactive hashing protocol, Alice and Bob have sets $s_A, s_B \subseteq [k]$ of size $|s_A| = |s_B| = n$. At the end of the protocol, Alice either rejects* *by sending a special $\perp$ message to Bob, or she accepts and sends some $u \in S_A$ to Bob. If Alice sends $\perp$, then Bob always rejects and outputs $\perp$. Otherwise, Bob can either accept, in which case he outputs the same u as Alice and it must hold that $u \in S_B$, or he rejects. The protocol has (α, β)-security if it satisfies the following properties:*

- *Correctness: If Alice and Bob both execute the protocol honestly using random subsets $s_A, s_B \subseteq [k]$ of size $|s_A| = |s_B| = n$ then:*
  $$\Pr[\textit{Alice accepts}] \geq \Omega(\min(n^2/k, 1)) \quad , \quad \Pr[\textit{Bob accepts} \mid \textit{Alice accepts}] = \frac{1}{2}.$$
  Furthermore whenever Alice accepts with some value u, then it must be the case that $u \in S_A$ and if Bob also accepts then it must be the case that $u \in S_A \cap S_B$.
- *Security for Honest Bob: If Bob follows the protocol honestly using a random subset $s_B \subseteq [k]$ of size $|s_B| = n$ and Alice follows the protocol arbitrarily, then, even condition on any arbitrary protocol transcript in which Alice accepts (i.e., does not send $\perp$ to Bob as the last message) we have:*
  $$\Pr[\textit{Bob accepts}] = \frac{1}{2}.$$
- *(α, β)-Security for honest Alice: Let $B \subseteq [k]$ be a set of size $|B| \leq \beta \cdot k$. If Alice follows the protocol honestly using a random subset $s_A \subseteq [k]$ of size $|s_A| = n$ and Bob follows the protocol arbitrarily, then*
  $$\Pr[\textit{Alice outputs } u \in B \mid \textit{Alice accepts}] \leq \frac{1}{2} + \alpha.$$

6.3 OT Construction

Theorem 19. *For any $m \geq \ell, \lambda$ there is some $n_{min} = \Omega(\log m + \ell + \lambda)$ such that for all $n \geq n_{min}$ there is an oblivious transfer protocol in the (n, m)-BSM with ℓ-bit messages and security $\varepsilon = 2^{-\Omega(\lambda)}$. The protocol is secure with efficient simulation, and it achieves security against an unbounded-memory sender. The round complexity is $O(\lceil m/n^2 \rceil \cdot \mathrm{poly}(\lambda, \log m))$ and the communication complexity $O(m \cdot \lceil m/n^2 \rceil \cdot \mathrm{poly}(\lambda, \log(m)))$.*

Proof. We describe the OT protocol between sender Alice and receiver Bob, and refer to the full version [13] for a proof.

Construction. Given m, λ, ℓ we define additional parameters as follows:

- Let $d_1 = O(\lambda + \log m)$ and $n_1 = O(\lambda + \log m + \ell)$ and $k' = O(m + \ell + \lambda \log(\lambda))$ be some values such that there is a $(n_1, m + \ell, \varepsilon = 2^{-\Omega(\lambda)})$-*BSM extractor* $\mathsf{BSMExt} : \{0,1\}^{k'} \times \{0,1\}^{d_1} \to \{0,1\}^{\ell}$ per Theorem 11.
- Let $t = O(\lambda + \log m)$ and $d_0 = O(t), n_0 = O(t)$ be some value such that there is a $(t/40, \varepsilon = 2^{-\Omega(\lambda)})$-extractor $\mathsf{Ext} : \{0,1\}^t \times \{0,1\}^{d_0} \to \{0,1\}^{d_1}$ that can be computed using n_0 space per Theorem 9.

- Set $k = (m + \lambda)\log^3(m + \lambda)$.
- Set $\alpha = \frac{1}{20}$ and let $\beta = 1/O(\log k)$ be such that (α, β)-security for interactive set hashing holds.
- Set $\delta = \Omega(\beta^2) = 1/O(\log^2 k))$ be such that $-\log(\mathbf{h}_+^{-1}(1 - \delta)) = (1 - \beta/2)$ by Lemma 14.
  This ensures that $\delta k \geq \Omega((m + \lambda)\log(m + \lambda))$.
- Let $g(k) = \text{poly}\log k$ be a parameter associated with set interactive hashing, such that an execution of the set interactive hashing protocol with parameters n, k can be done in $n \cdot g(k)$ time and space(see the full version [13] for more details). Assume $g(k) \geq \lceil \log k + 1 \rceil$.
- Define $n_{min} = \max(2n_0, 2n_1, 3t + g(k)) = O(\lambda + \log m + \ell)$.
- For any $n \geq n_{min}$, define $\tilde{n} = \lfloor (n - 2t)/g(k) \rfloor = \Omega(n/\text{polylog}(m + \lambda))$.
- Let $p = \Omega(\min(\widetilde{n}^2/k, 1))$ be the correctness probability of Alice accepting during an honest execution of the set interactive hashing protocol with parameters $\widetilde{n}, k$,per Definition 18.
  Set $R_{max} = 2t/p = O(t \cdot \lceil k/\widetilde{n}^2 \rceil) = O(\lceil m/n^2 \rceil \cdot \text{poly}(\lambda, \log m)$.

The protocol works as follows.

Bob has a choice bit $c \in \{0, 1\}$ and Alice has two messages $\mathsf{msg}_0, \mathsf{msg}_1 \in \{0, 1\}^\ell$.

1. Alice and Bob initiate vectors $r_A \in \{0, 1\}^t, r_B \in \{0, 1, \perp\}^t$ respectively. They set $i := 0$.
   Repeat the following until $i = t$:
   (a) Alice and Bob select uniformly random subsets $s_A, s_B \subseteq [k]$ of size $|s_A| = |s_B| = \widetilde{n}$ respectively.
   Alice streams a uniformly random string $x \leftarrow \{0, 1\}^k$ to Bob.
   Alice stores $x[s_A]$, and Bob stores $x[s_B]$.
   (b) Alice and Bob perform set interactive hashing, with Bob's input being s_B.
      - If Alice rejects, then both parties move to the next iteration.
      - Else, if Alice accepts with some value $u \in s_A$, then she sets $r_A[i] = x[u]$.
         - If Bob also accepts then it must be the case that $u \in s_B$ and he sets $r_B[i] = x[u]$.
         - Else, Bob sets $r_B[i] = \perp$.

        Both parties increment $i := i + 1$.

   If the number of iterations reaches R_{max} before $i = t$, the parties abort.
2. Bob sets $I := \{i \in [t] : r_B[i] \neq \perp\}$. If $|I| < \frac{2 \cdot t}{5}$ then Bob aborts. Else he chooses two sets I_0, I_1 of size $|I_0| = |I_1| = \frac{2 \cdot t}{5}$ by sub-selecting $I_c \subseteq I$ and $I_{1-c} \subseteq [t] \setminus I_c$ uniformly at random.
   Bob sends I_0, I_1 to Alice.
3. Alice checks that $|I_0| = |I_1| = \frac{2 \cdot t}{5}$ and $I_0 \cap I_1 = \emptyset$ and aborts otherwise.
   She chooses an extractor seed $\mathsf{seed} \leftarrow \{0, 1\}^{d_0}$ and sends seed to Bob.
   Alice computes $\mathsf{seed}_0 = \mathsf{Ext}(r_A[I_0]; \mathsf{seed})$, $\mathsf{seed}_1 = \mathsf{Ext}(r_A[I_1]; \mathsf{seed})$.
   Bob computes $\mathsf{seed}_c = \mathsf{Ext}(r_B[I_c]; \mathsf{seed})$.

4. Alice streams a uniformly random string $x \leftarrow \{0,1\}^{k'}$ to Bob.
   Alice computes $\mathsf{key}_0 = \mathsf{BSMExt}(x; \mathsf{seed}_0), \mathsf{key}_1 = \mathsf{BSMExt}(x; \mathsf{seed}_1)$.
   Bob computes $\mathsf{key}_c = \mathsf{BSMExt}(x; \mathsf{seed}_c)$.
5. Alice sends to Bob:

$$\mathsf{ct}_0 = \mathsf{key}_0 \oplus \mathsf{msg}_0, \quad \mathsf{ct}_1 = \mathsf{key}_1 \oplus \mathsf{msg}_1$$

   and Bob outputs $\mathsf{msg} = \mathsf{ct}_c \oplus \mathsf{key}_c$.

□

6.4 Multiparty Computation from OT

It is known that one can use the oblivious transfer (OT) ideal functionality as a black box to achieve general multi-party computation in the OT-hybrid model [25, 23]. By plugging in our construction of OT in the BSM, one therefore gets general multiparty computation in the BSM with efficient simulation. A similar observation that OT implies MPC in the BSM was already made in [10] and expanded on in [27], albeit in the setting where both the OT and the MPC only satisfy inefficient simulation.

We provide some additional details. Assume we want to perform a multiparty computation of some circuit C with N parties and security parameter λ.

- Honest user storage: If we start with an OT protocol in the (n, m)-BSM and use it to construct MPC, the honest users need to keep in memory all of the intermediate state of the external MPC protocol in the OT-hybrid model. The size of this state is some $\text{poly}(|C|, N, \lambda)$ completely independent of n, m. Therefore the resulting protocol will be in the (n', m)-BSM model with $n' = n + \text{poly}(|C|, N, \lambda)$, which can still be arbitrarily smaller than the adversarial storage m.
- Adversary storage: We note that the MPC protocol only executes copies of the OT protocol sequentially. When the "outer" simulator of the overall MPC needs to simulate each OT execution, it can spawn of a fresh copy of an "inner" OT simulator. Although the outer simulator may need to store some additional state related to the outer MPC execution, this is completely unrelated to the inner OT. Therefore, the OT protocol only needs to achieve security against an OT adversary with the same storage bound m as the overall MPC adversary.

Summarizing we get the following theorem as a corollary of our OT protocol (Theorem 19) and the works of [25, 23].

Theorem 20. *For any m, λ and any N-party ideal functionality $\mathcal{F}$ having circuit size $|\mathcal{F}|$, there is some $n_{min} = O(\log m) + \text{poly}(|\mathcal{F}|, N, \lambda)$ such that for all $n \geq n_{min}$ there is a secure MPC protocol in the (n, m)-BSM with $\varepsilon = 2^{-\Omega(\lambda)}$ security against an adversary that can maliciously corrupt any number of parties. The round complexity is $O(\lceil m/n^2 \rceil \cdot \text{poly}(|\mathcal{F}|, N, \lambda))$ and the communication complexity $O(m \cdot \lceil m/n^2 \rceil \cdot \text{poly}(|\mathcal{F}|, N, \lambda))$.*

7 Lower Bounds on Rounds and Communication

In this section, we prove that achieving large memory gaps between adversaries and honest parties in the bounded storage model inherently requires large round complexity and communication. In Section 7.1, we introduce the specific BSM we use for our lower bound. In Section 7.2, we prove a lower bound on round complexity and communication in this model. Looking ahead, one drawback of this lower bound is that it only rules out protocols secure against somewhat strong, non-streaming adversaries. In Section 7.3, we introduce another variant of the BSM where adversaries are streaming, and prove an associated lower bound in Section 7.4.

7.1 Model for the Lower Bound: the Unbounded Processing Model

As mentioned in the introduction, our lower bound holds in a stronger model than the variant of streaming BSM we use for our positive results in Section 4. The main conceptual difference is that both the honest parties are only bound by their storage used between the rounds, but could compute its contents using unbounded temporary memory. We describe that model, and introduce notation in more details below. We develop in more details the relation with previously discussed notions of BSM in Remark 21.

A (n, m)-bounded storage model protocol Π in the *unbounded processing model*, is parametrized by a bound n on the storage of honest parties and a bound m on the storage of the adversary. In the case of two parties, Alice and Bob send (potentially large) messages to each other at every round. Every round i consists of one party, say Alice, sending a message to the other, say Bob, as follows: she computes

$$(s_A^{(i)}, M^{(i)}) \leftarrow \mathsf{send}_A^{(i)}(s_A^{(i-1)})$$

and Bob computes

$$s_B^{(i)} \leftarrow \mathsf{receive}_B^{(i)}(s_B^{(i-1)}, M^{(i)}),$$

and vice-versa if Bob sends the message in round i. $s_A^{(i)}$ and $s_B^{(i)}$ denote the local states kept by Alice and Bob respectively after round i, and $M^{(i)}$ denotes the message sent at round i. By convention their starting states are $s_A^{(0)} = s_B^{(0)} = \emptyset$. We require the states s_A and s_B to be of bounded size, namely $|s_A^{(i)}|, |s_B^{(i)}| \leq n$ for all i. There are however no restrictions on the complexity of the functions $\mathsf{send}_A^{(i)}, \mathsf{receive}_B^{(i)}$ in rounds i where Alice sends a message, or $\mathsf{send}_B^{(i)}, \mathsf{receive}_A^{(i)}$ in rounds i where Bob sends a message. We'll assume for convenience of notation that parties speak turn by turn, namely Alice sends messages in odd rounds and Bob sends messages in even rounds, or vice-versa.

Adversaries Adv in this model are similarly modeled as functions $\mathsf{Adv}^{(i)}$: $(s_E^{(i-1)}, M^{(i)})^{(i)} \mapsto s_E^{(i)}$, where there are no restrictions on the complexity of $\mathsf{Adv}^{(i)}$, up to the state $s_E^{(i)}$ having size at most m for all i.

We will respectively denote by C and R some upper bounds on the total communication and the number of rounds of Π, which hold over all possible executions of Π. In the case of key agreement, this is without loss of generality up to a constant loss in either security (having both parties abort and output 0) or correctness (having both parties abort and output a random value), using Markov's inequality.

We will furthermore suppose that the length of the message sent in any fixed round i is fixed by the protocol, and in particular does not depend on its internal randomness. We discuss how to relax this requirement in the full version [13].

We now define key agreement (with 1-bit output) in this model. A key agreement protocol Π in the (n, m)-BSM is a protocol with two parties, Alice and Bob, which results in a single-bit final state $s_A^{(R)}, s_B^{(R)} \in \{0, 1\}$. We require the following properties:

- δ-correctness: We have
$$\Pr[s_A^{(R)} = s_B^{(R)}] \geq 1/2 + \delta.$$
for some constant $\delta \leq 1/2$.
- (m, ε)-Sscurity: No adversary Adv with memory m (with the specifications above) can guess Alice's output $s_A^{(R)}$ at the end of the protocol:
$$\forall \mathsf{Adv}, \Pr[s_E^{(R)} = s_A^{(R)}] \leq 1/2 + \varepsilon,$$
for constant $\varepsilon \leq 1/2$.
- We furthermore require $\delta - \varepsilon = \Theta(1)$.

The last requirement enforces that adversaries have strictly smaller probability of guessing the output of the honest parties than the other honest party.

Remark 21 (Comparison with previously discussed models). As mentioned before, this defines a more expressive model than the one in Section 4, as honest users for the definition above are stronger than in Section 4. The main differences are (1) there are no restrictions on the computational power of the *honest users* to compute their states kept between the rounds of the protocol, who can in particular use arbitrary large temporary memory, (2) they are neither bound to send uniformly random "long" messages, nor restricted to have local access to it. In the terminology we used in the introduction, the lower bound holds for honest users without restrictions (b), (c), but with the same capability (d) as Eve. All these capabilities make the resulting lower bound stronger.

However, we only consider strong adversaries with unlimited short-term memory (restriction (d) in the introduction). This does make our lower bound weaker than ideal, and leaves open the possibility of a tighter lower bound for more restricted classes of "streaming" adversaries. Looking ahead, in Sections 7.3 and 7.4, we adapt this model and the subsequent lower bound to restrict adversaries to be streaming, albeit at the cost of slightly worse quantitative bounds.

To sum up, in this new model, the honest users have the same capabilities as the adversary, up to a smaller storage between rounds.

In terms of key agreement, we relax correctness and security to only be constants, as long as honest users have some non-trivial advantage in agreeing on the output bit compared to an adversary; this again makes our lower bound stronger.

7.2 Lower Bound in the Unbounded Processing Model

Theorem 22. *Let Π be a key agreement protocol in the unbounded processing model (Section 7.1), with honest storage n, satisfying δ-correctness and (m, ε) security, where $\delta - \varepsilon = \Omega(1)$. Suppose furthermore that for any execution of Π, the total communication between Alice and Bob is at most C and consists of at most R rounds. Then $C \geq \Omega\left(\frac{m^{3/2}}{n}\right)$, and $R \geq \Omega\left(\frac{\sqrt{m}}{n}\right)$.*

Remark 23 (Lower Bound for OT). Because any OT protocol directly induces a key agreement protocol with identical round complexity and communication, the theorem directly extends to an identical lower bound for OT.

We refer to the full version [13] for a proof.

7.3 Model for a Lower Bound against Streaming Adversaries

In the unbounded processing model for our lower bounds of Sections 7.1 and 7.2, the only restriction, both for the honest parties and the adversary, is that their maintained state between rounds of communications has bounded size. In particular, they all can process messages from the protocol using potentially *unbounded* temporary memory, so long as they compress it to some limited amount of storage afterwards.

One natural setting left open, however, is the case where the adversary has bounded storage throughout the entire attack and only streaming access to messages sent. This makes the adversary *weaker* than in the model of Section 7.1, and it is not clear whether the subsequent lower bound extends. In this section, along with Section 7.3, we extend the lower bound of Sections 7.1 and 7.2 to such adversaries, albeit at the cost of slightly worse parameters. Another difference is that while Sections 7.1 and 7.2 also rule out protocols with unbounded processing *honest parties*, the model of this section and the subsequent lower bound in Section 7.3 only rule out *streaming* honest parties.

We first describe our model, that we call the *streaming model with CRS* . Honest parties send and receive messages in a *streaming manner*, using some bounded memory n, without any other restriction on the messages sent nor on the receiving algorithm. We will also consider adversaries which are similarly treating messages sent between the parties in a streaming manner using bounded memory $m > n$.

For comparison with Section 4, honest users are still more powerful, as having general streaming access to messages (as opposed to local access), and are not required to send uniformly random messages. In other words, honest parties neither have restriction (b) nor (c), but are still required now to be treating messages in a streaming manner. The adversary, however, is *weakened* to only have streaming access to the messages of the protocol, similar to honest users. Doing so makes the resulting lower bound stronger.

Compared with our previous lower bound (Sections 7.1 and 7.2), both the honest parties and the adversary are weaker, as they are now both streaming, as opposed to having unbounded preprocessing. As a result, the resulting lower bounds are technically incomparable. Still, we believe that restricting ourselves to protocols where honest parties use bounded memory during the whole execution of the protocol is an extremely natural setting for protocols in the bounded storage model.

Optionally, we will consider a streaming model *in the common reference string model*, where a common reference string is available prior to protocol execution. The CRS is used to (independently) derive starting states for the parties of the protocol. We consider these processes (namely, the CRS generation and the user state generation) to be performed by a trusted party, which can potentially run in memory larger than n. Honest parties do not require knowledge of the CRS to execute the remainder protocol, but adversaries do have access to the CRS to mount attacks. For simplicity, we will only consider CRS that directly fit in the adversary's memory.

We define key agreement (with one-bit output) in a very similar way as in Section 7.1: we refer to that section for our notion of δ-correctness and (m, ε)-security. We further consider security against *non-uniform attacks* where adversaries obtain some non-uniform advice that can be generated using unbounded memory.

7.4 Lower Bound against Streaming Adversaries

Theorem 24. *Let Π be a key agreement protocol in the streaming model (Section 7.3) with honest storage n, satisfying δ-correctness and (m, ε) security against non-uniform attacks, where $\delta - \varepsilon = \Omega(1)$. Suppose furthermore that for any execution of Π, the total communication between Alice and Bob is at most C and consists of at most R rounds. Then $C \geq \Omega\left(m \cdot \left(\frac{m}{n^2}\right)^{1/3}\right)$, and $R \geq \Omega\left(\left(\frac{m}{n^2}\right)^{1/3}\right)$.*

Remark 25 (Lower Bound for OT). Because any OT protocol directly induces a key agreement protocol with identical round complexity and communication, the theorem directly extends to an identical lower bound for OT.

We refer to the full version [13] for a proof.

References

1. Aumann, Y., Ding, Y.Z., Rabin, M.: Everlasting security in the bounded storage model. IEEE Transactions on Information Theory **48**(6), 1668–1680 (2002). https://doi.org/10.1109/TIT.2002.1003845

2. Aumann, Y., Feige, U.: On message proof systems with known space verifiers. In: Stinson, D.R. (ed.) CRYPTO'93. LNCS, vol. 773, pp. 85–99. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 22–26, 1994). https://doi.org/10.1007/3-540-48329-2_8
3. Bellare, M., Kane, D., Rogaway, P.: Big-key symmetric encryption: Resisting key exfiltration. In: Robshaw, M., Katz, J. (eds.) CRYPTO 2016, Part I. LNCS, vol. 9814, pp. 373–402. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 14–18, 2016). https://doi.org/10.1007/978-3-662-53018-4_14
4. Cachin, C., Crépeau, C., Marcil, J.: Oblivious transfer with a memory-bounded receiver. In: 39th FOCS. pp. 493–502. IEEE Computer Society Press, Palo Alto, CA, USA (Nov 8–11, 1998). https://doi.org/10.1109/SFCS.1998.743500
5. Cachin, C., Maurer, U.M.: Unconditional security against memory-bounded adversaries. In: Kaliski Jr., B.S. (ed.) CRYPTO'97. LNCS, vol. 1294, pp. 292–306. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 1997). https://doi.org/10.1007/BFb0052243
6. Calabro, C.: The exponential complexity of satisfiability problems. Ph.D. thesis, University of California, San Diego, USA (2009), http://www.escholarship.org/uc/item/0pk5w64k
7. Crépeau, C.: Equivalence between two flavours of oblivious transfers. In: Pomerance, C. (ed.) CRYPTO'87. LNCS, vol. 293, pp. 350–354. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 1988). https://doi.org/10.1007/3-540-48184-2_30
8. Diffie, W., Hellman, M.E.: New directions in cryptography. IEEE Transactions on Information Theory **22**(6), 644–654 (November 1976)
9. Ding, Y.Z.: Oblivious transfer in the bounded storage model. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 155–170. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 19–23, 2001). https://doi.org/10.1007/3-540-44647-8_9
10. Ding, Y.Z., Harnik, D., Rosen, A., Shaltiel, R.: Constant-round oblivious transfer in the bounded storage model. Journal of Cryptology **20**(2), 165–202 (Apr 2007). https://doi.org/10.1007/s00145-006-0438-1
11. Ding, Y.Z., Rabin, M.O.: Hyper-encryption and everlasting security. In: Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science. p. 1–26. STACS '02, Springer-Verlag, Berlin, Heidelberg (2002)
12. Dodis, Y., Ostrovsky, R., Reyzin, L., Smith, A.D.: Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM J. Comput. **38**(1), 97–139 (2008). https://doi.org/10.1137/060651380, https://doi.org/10.1137/060651380
13. Dodis, Y., Quach, W., Wichs, D.: Speak much, remember little: Cryptography in the bounded storage model, revisited. Cryptology ePrint Archive, Paper 2021/1270 (2021), https://eprint.iacr.org/2021/1270, https://eprint.iacr.org/2021/1270
14. Dziembowski, S., Kazana, T., Zdanowicz, M.: Quasi chain rule for min-entropy. Information Processing Letters **134**, 62–66 (2018). https://doi.org/https://doi.org/10.1016/j.ipl.2018.02.007, https://www.sciencedirect.com/science/article/pii/S002001901830036X
15. Dziembowski, S., Maurer, U.M.: Tight security proofs for the bounded-storage model. In: 34th ACM STOC. pp. 341–350. ACM Press, Montréal, Québec, Canada (May 19–21, 2002). https://doi.org/10.1145/509907.509960

16. Dziembowski, S., Maurer, U.M.: On generating the initial key in the bounded-storage model. In: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 126–137. Springer, Heidelberg, Germany, Interlaken, Switzerland (May 2–6, 2004). https://doi.org/10.1007/978-3-540-24676-3_8
17. Garg, S., Raz, R., Tal, A.: Extractor-based time-space lower bounds for learning. In: Diakonikolas, I., Kempe, D., Henzinger, M. (eds.) 50th ACM STOC. pp. 990–1002. ACM Press, Los Angeles, CA, USA (Jun 25–29, 2018). https://doi.org/10.1145/3188745.3188962
18. Goldwasser, S., Micali, S., Rackoff, C.: The knowledge complexity of interactive proof systems. SIAM J. Comput. **18**(1), 186–208 (Feb 1989). https://doi.org/10.1137/0218012, https://doi.org/10.1137/0218012
19. Guan, J., Zhandry, M.: Simple schemes in the bounded storage model. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019, Part III. LNCS, vol. 11478, pp. 500–524. Springer, Heidelberg, Germany, Darmstadt, Germany (May 19–23, 2019). https://doi.org/10.1007/978-3-030-17659-4_17
20. Guan, J., Zhandry, M.: Disappearing cryptography in the bounded storage model. In: Nissim, K., Waters, B. (eds.) Theory of Cryptography. pp. 365–396. Springer International Publishing, Cham (2021)
21. Hong, D., Chang, K.Y., Ryu, H.: Efficient oblivious transfer in the bounded-storage model. In: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2501, pp. 143–159. Springer, Heidelberg, Germany, Queenstown, New Zealand (Dec 1–5, 2002). https://doi.org/10.1007/3-540-36178-2_9
22. Impagliazzo, R., Levin, L.A., Luby, M.: Pseudo-random generation from one-way functions. In: Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing. p. 12–24. STOC '89, Association for Computing Machinery, New York, NY, USA (1989). https://doi.org/10.1145/73007.73009, https://doi.org/10.1145/73007.73009
23. Ishai, Y., Prabhakaran, M., Sahai, A.: Founding cryptography on oblivious transfer - efficiently. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 572–591. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 17–21, 2008). https://doi.org/10.1007/978-3-540-85174-5_32
24. Kamp, J., Rao, A., Vadhan, S., Zuckerman, D.: Deterministic extractors for small-space sources. Journal of Computer and System Sciences **77**(1), 191–220 (2011). https://doi.org/https://doi.org/10.1016/j.jcss.2010.06.014, https://www.sciencedirect.com/science/article/pii/S002200001000098X, celebrating Karp's Kyoto Prize
25. Kilian, J.: Founding cryptography on oblivious transfer. In: 20th ACM STOC. pp. 20–31. ACM Press, Chicago, IL, USA (May 2–4, 1988). https://doi.org/10.1145/62212.62215
26. Kol, G., Raz, R., Tal, A.: Time-space hardness of learning sparse parities. In: Hatami, H., McKenzie, P., King, V. (eds.) 49th ACM STOC. pp. 1067–1080. ACM Press, Montreal, QC, Canada (Jun 19–23, 2017). https://doi.org/10.1145/3055399.3055430
27. Liu, J., Vusirikala, S.: Secure multiparty computation in the bounded storage model. In: Paterson, M.B. (ed.) Cryptography and Coding. pp. 289–325. Springer International Publishing, Cham (2021)
28. Lu, C.J.: Hyper-encryption against space-bounded adversaries from on-line strong extractors. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 257–271. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 18–22, 2002). https://doi.org/10.1007/3-540-45708-9_17

29. Maurer, U.M.: Conditionally-perfect secrecy and a provably-secure randomized cipher. Journal of Cryptology **5**(1), 53–66 (Jan 1992). https://doi.org/10.1007/BF00191321
30. Moran, T., Shaltiel, R., Ta-Shma, A.: Non-interactive timestamping in the bounded-storage model. Journal of Cryptology **22**(2), 189–226 (Apr 2009). https://doi.org/10.1007/s00145-008-9035-9
31. Naor, M., Ostrovsky, R., Venkatesan, R., Yung, M.: Perfect zero-knowledge arguments for NP can be based on general complexity assumptions (extended abstract). In: Brickell, E.F. (ed.) CRYPTO'92. LNCS, vol. 740, pp. 196–214. Springer, Heidelberg, Germany, Santa Barbara, CA, USA (Aug 16–20, 1993). https://doi.org/10.1007/3-540-48071-4_14
32. Nisan, N.: Pseudorandom generators for space-bounded computations. In: Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing. p. 204–212. STOC '90, Association for Computing Machinery, New York, NY, USA (1990). https://doi.org/10.1145/100216.100242, https://doi.org/10.1145/100216.100242
33. Nisan, N., Zuckerman, D.: Randomness is linear in space. J. Comput. Syst. Sci. **52**(1), 43–52 (1996). https://doi.org/10.1006/jcss.1996.0004, https://doi.org/10.1006/jcss.1996.0004
34. Rabin, M.O.: How to Exchange Secrets with Oblivious Transfer (1981), harvard Aiken Computational Laboratory TR-81
35. Raz, R.: Fast learning requires good memory: A time-space lower bound for parity learning. In: Dinur, I. (ed.) 57th FOCS. pp. 266–275. IEEE Computer Society Press, New Brunswick, NJ, USA (Oct 9–11, 2016). https://doi.org/10.1109/FOCS.2016.36
36. Raz, R.: A time-space lower bound for a large class of learning problems. In: Umans, C. (ed.) 58th FOCS. pp. 732–742. IEEE Computer Society Press, Berkeley, CA, USA (Oct 15–17, 2017). https://doi.org/10.1109/FOCS.2017.73
37. Santis, A.D., Persiano, G., Yung, M.: One-message statistical zero-knowledge proofs and space-bounded verifier. In: Proceedings of the 19th International Colloquium on Automata, Languages and Programming. p. 28–40. ICALP '92, Springer-Verlag, Berlin, Heidelberg (1992)
38. Skorski, M.: Strong chain rules for min-entropy under few bits spoiled. In: 2019 IEEE International Symposium on Information Theory (ISIT). pp. 1122–1126 (2019). https://doi.org/10.1109/ISIT.2019.8849240
39. Vadhan, S.P.: Constructing locally computable extractors and cryptosystems in the bounded-storage model. Journal of Cryptology **17**(1), 43–77 (Jan 2004). https://doi.org/10.1007/s00145-003-0237-x