On Bounded Storage Key Agreement and One-Way Functions

Chris Brzuska, Geoffroy Couteau, Christoph Egger, Willy Quach

On Bounded Storage Key Agreement and One-Way Functions

Chris Brzuska[1], Geoffroy Couteau[2], Christoph Egger[2], and Willy Quach[3]

[1] Aalto University, Finland
[2] Université Paris Cité, CNRS, IRIF, France
[3] Weizmann Institute of Science, Israel

Abstract. We study key agreement in the bounded-storage model, where the participants and the adversary can use an a priori fixed bounded amount of space, and receive a large stream of data. While key agreement is known to exist unconditionally in this model (Cachin and Maurer, Crypto'97), there are strong lower bounds on the space complexity of the participants, round complexity, and communication complexity that unconditional protocols can achieve.

In this work, we explore how a minimal use of cryptographic assumptions can help circumvent these lower bounds. We obtain several contributions:

- Assuming one-way functions, we construct a one-round key agreement in the bounded-storage model, with arbitrary polynomial space gap between the participants and the adversary, and communication slightly larger than the adversarial storage. Additionally, our protocol can achieve everlasting security using a second streaming round.
- In the other direction, we show that one-way functions are *necessary* for key agreement in the bounded-storage model with large space gaps. We further extend our results to the setting of *fully-streaming* adversaries, and to the setting of key agreement with multiple streaming rounds.

Our results rely on a combination of information-theoretic arguments and technical ingredients such as pseudorandom generators for space-bounded computation, and a tight characterization of the space efficiency of known reductions between standard Minicrypt primitives (from distributional one-way functions to pseudorandom functions), which might be of independent interest.

1 Introduction

Perhaps surprisingly, while cryptographic primitives must typically rely on hardness assumptions in the time-bounded setting (and proving their security unconditionally would entail proving $\mathsf{P} \neq \mathsf{NP}$), several cryptographic primitives of interest are known to exist *unconditionally* in the bounded-storage model (BSM). In this model, introduced by Maurer [Mau92], the participants and adversary are *space-bounded* (with a gap between the space s honest parties need and the space a the adversary needs) and have one-time read access to a huge random

string (of length $\gg a$). In the BSM, symmetric key encryption [Mau92], signatures [DQW22], key agreement [CM97], and oblivious transfer [Din01], all exist unconditionally. Yet, unconditional constructions of "public-key-style" primitives in the bounded-storage model typically suffer from strong efficiency limitations regarding the space gap between honest parties and adversaries, round complexity, and communication complexity. For example, the bounded-storage model key agreement (from now on, BSM-KA) of [CM97] requires the honest parties to use $s = \omega(\sqrt{a})$ bits of storage. More recently, the work of [DQW23] circumvented this limitation, but at the cost of requiring $r = \omega(a/s^2)$ streaming rounds and $C = \omega((a/s)^2)$ bits of communication. Unfortunately, these limitations are known to be inherent: the protocol of [CM97] was shown in [DM08] to achieve an optimal space gap $a = \tilde{\Theta}(s^2)$ when the BSM-KA uses a single streaming round, and [DQW23] further proved that the number of rounds must grow with a, and the communication must grow superlinearly with a, whenever $a \gg s^2$. Therefore, achieving unconditional security for BSM-KA requires paying a significant price either in honest parties space or in rounds and communication.

In this work, we initiate the study of cryptography in the bounded-storage model beyond the regime where the impossibility results of [DM08, DQW23] apply. That is, we ask:

Is it possible to circumvent known lower bounds on key agreements in the bounded-storage model by making a minimal use of cryptographic assumptions?

To study this question, we place ourselves in the *streaming* variant of the BSM, introduced in [DQW23], where the participants themselves can stream long strings (of length $C \gg a \gg s$) to each other. In [DQW23], it was argued that this captures more adequately the properties one wants from cryptography with bounded-storage.

1.1 Our Contributions

We provide an affirmative answer to the question. As our first contribution, we exhibit a key agreement in the streaming model tolerating an arbitrary (polynomial) gap between the space s of the honest parties and the space a of the adversary, using a single streaming round and $C = \tilde{O}(a)$ bits of communication, assuming the existence of one-way functions (OWFs).

Theorem 1 (Informal). *Let λ be a security parameter and $a = a(\lambda)$ be an arbitrary polynomial in λ. Assuming the existence of one-way functions, there is an BSM-KA protocol in the streaming model secure against an eavesdropper with space a that uses a single long stream of length $a \cdot \mathsf{poly}(\lambda)$ (followed by a single $\mathsf{poly}(\lambda)$-sized short message in the other direction), and where the honest parties use $s = \mathsf{poly}(\lambda)$ storage.*

In the Theorem above, $\mathsf{poly}(\lambda)$ denotes a fixed polynomial independent of a. The BSM-KA uses two rounds of communication with one stream; it can alternatively use a single simultaneous round of streaming (in both directions),

yielding a non-interactive key agreement in the streaming model. Eventually, the security of the BSM-KA can be strengthened to everlasting security (the shared key remains protected even if the adversary becomes all powerful after the completion of the protocol) at the cost of using an additional round of streaming.

Theorem 1 shows that OWFs are sufficient to obtain an (everlasting-secure) key agreement in the streaming model, which is essentially optimal regarding space gaps and round complexity. Then, we ask:

Are one-way functions also necessary for obtaining key agreement in the streaming model in the regime where it cannot exist unconditionally?

To approach this question, we initiate a systematic study of the relations between various forms of key agreements in the streaming model and the existence of one-way functions. We make significant progress towards answering the above question affirmatively. Our work also leaves several natural and intriguing questions open; we hope that our preliminary findings will motivate their study in future works.

In the course of our analysis, we observe that answering this question requires tightly characterizing the space efficiency of reductions between various Minicrypt primitives such as distributional OWFs, weak OWFs, standard OWFs, pseudorandom generators, and variants of pseudorandom functions. We provide some preliminary investigation in this direction, characterizing the space efficiency of existing reductions between these primitives, which we believe might be of independent interest. We believe that our work provides some additional motivation for the question of designing space-tight reduction between Minicrypt primitives, a natural question which has not received much attention so far.

Answering the question turns out to require careful considerations regarding the type of protocols and the type of adversaries that are considered. Before stating our results, we provide a brief outline of these considerations:

- key agreements in the streaming model can have a single long stream (and multiple short rounds), or multiple long streams. The distinction between these two settings was traditionally made on the basis of the desirability of minimizing the number of long rounds (see for example the discussion on the "desirable property (a)" in [DQW23]). For the question we raise, it turns out that another important distinction for single-long-stream protocols is whether the protocol starts with the long stream (a setting called the "traditional bounded-storage model" in [DQW23]), or whether it starts with short rounds.
- One can consider two types of space-bounded adversaries (we follow the naming conventions of [DQW23] for these two models and refers the reader to [DQW23] for further discussions on the distinction): "fully streaming adversaries" have space bounded by a throughout the entire protocol, while adversaries in the "unbounded processing model" are allowed unlimited short-term storage, and are only subject to keeping an a-bit state in between long rounds. Of course, building key agreement in the unbounded processing model is more desirable (our construction of Theorem 1 is in this model), while

proving impossibility results in the full streaming model yields a stronger result.

We note that our notion of unbounded processing differs from that of [DQW23] due to our use of computational assumptions: in [DQW23], an unbounded processing adversary has bounded storage during the streaming rounds, unbounded storage otherwise, and unbounded computational power throughout. We consider here a variant where the adversary remains probabilistic polynomial time (hence, in particular, always uses a polynomial amount of storage) but has no further storage bound inbetween the streaming rounds (but can only store an a-bit state after a long round). To avoid confusion, we will sometime use the terminologies "fully-streaming PPT adversary" and "unbounded-processing PPT adversary", where PPT refers to probabilistic polynomial-time. Eventually, we also consider *everlasting* security, where the adversary are (fully-streaming or unbounded-processing) PPT throughout the protocol, but become all powerful after the protocol.

In the following, we will write SM-KA to denote key agreement in the streaming model for a fully-streaming PPT adversary, and UP-KA to denote key agreement in the streaming model for an adversary in the unbounded processing PPT model. With this terminology in mind, our protocol in Theorem 1 is actually an UP-KA, secure in the unbounded-processing PPT model (the strongest adversarial model we consider). We complement this result by showing that *space-bounded* OWFs (*i.e.*, functions which are one-way against PPT adversaries with a fixed polynomially-bounded amount of storage) actually suffice for constructing SM-KA (where the adversary is fully streaming) via our construction. This requires in particular carefully tracking the space efficiency of the traditional constructions of pseudorandom generators from OWFs [HILL99], and of pseudorandom functions from pseudorandom generators [GGM84].

Corollary 2 (Informal). *Let λ be a security parameter and $a = a(\lambda)$ be an arbitrary polynomial in λ. Assuming the existence of space-bounded one-way functions with space bound $\mathsf{poly}(a)$, there is an SM-KA protocol secure against a fully-streaming PPT eavesdropper with space a that uses a single long stream of length $a \cdot \mathsf{poly}(\lambda)$ (followed by a single $\mathsf{poly}(\lambda)$-sized short message in the other direction), and where the honest parties use $s = \mathsf{poly}(\lambda)$ storage.*

We now state our main results towards showing the necessity of OWFs for streaming key agreement beyond the unconditional regime. We first focus on protocols which involve a single streaming message and a short answer.

Theorem 3 (Informal). *Assume that there exists a streaming key agreement KA against space-a PPT adversaries consisting of a single long stream from Alice to Bianca and a short message from Bianca to Alice and using $s \ll \sqrt{a}$ space for the honest parties. Then,*

- *if KA is an UP-KA, there exists one-way functions;*
- *if KA is an SM-KA, there exists space-bounded one-way functions.*

The conclusion of Theorem 3 is the best possible, as it matches exactly our positive results of Theorem 1 and Corollary 2. However, one may ask whether it could be possible to relax the requirement of one-way functions if we either restrict the adversary to be fully streaming, and/or if the protocol can have additional streaming rounds and short rounds. In this more general setting, we prove the following theorem:

Theorem 4 (Informal). *Assume that there exists a streaming key agreement against space-a (fully-streaming or unbounded-processing) PPT adversary with r rounds using $s \ll a^{1/\mathsf{polylog}(r)^r}$ space for the honest parties, for a suitably large* polylog*. Then,*

- *if* KA *is an UP-KA, there exists non-uniform, infinitely-often one-way functions;*
- *if* KA *is an SM-KA, there exists non-uniform, infinitely-often space-bounded one-way functions $f : \{0,1\}^n \to \{0,1\}^n$ with space bound $n^{\Omega(\mathsf{polylog}(r)^r)}$.*

The conclusions of Theorem 4 are weaker than that of Theorem 3 on two aspects: first we only get infinitely-often secure OWFs, and second, the conclusion requires assuming a larger space gap. The first limitation (infinitely-often security) is an unfortunate but standard consequence of the use of a disjunction argument based on the existence of a OWF inverter (a similar limitation appears in many previous works). As for the last limitation, we observe that when the number of long streams is 1, assuming only $a \gg s^2$ (up to polylog factor) suffices to achieve the weaker conclusion of space-bounded *distributional* OWFs. We view as an interesting open question the goal of obtaining space-bounded OWFs from streaming key agreement with a smaller space gap (ideally $a \gg s^2$). A natural starting point to solve this question would be to find a space-tight reduction from (space-bounded) distributional OWFs to OWFs, a question which we believe to be also of independent interest. Additionally, we note that the one-way functions obtained in Theorem 4 are non-uniform; our result can be strengthened to provide uniform one-way functions in the special case of a single long stream.

Due to the exponential dependency in r, Theorem 4 is only meaningful in the setting where r is a constant. We leave as an intriguing open question to prove (or disprove) that SM-KA with a superconstant number of rounds imply OWFs.

1.2 Discussions

One-way functions are known to be a necessary assumption for most cryptographic primitives [IL89]. Several lines of work have investigated the necessity of one-way functions for various types of cryptographic *protocols*, notably in the setting of zero-knowledge interactive proofs for NP [OW93], single-server private information retrieval [BIKM99], and constant-bias coin flipping [MPS10, HO11, BHT14]. In each case, unconditionally secure variants of these protocols can be obtained by relaxing the constraints, such as using multiple parties or servers [BOL85, CGKS95] or restricting the class of languages to SZK.

Our work fits in this broad program by studying another example of cryptographic protocol, streaming-model key-agreements, in the regime where it cannot exist unconditionally. Similar to constant-bias coin flipping and zero-knowledge for NP, we actually show that one-way functions are essentially *equivalent* to streaming key agreement. Our results nevertheless leave several gaps in the space gap between the honest parties and the adversary, most notably for protocols with a large numbers of streaming rounds. Whether these gaps can be closed, or whether some non-trivial forms of streaming key agreement beyond the unconditional regime could possibly exist without one-way functions remains an interesting open question, the main one left open by our work.

Turning to our positive result, the efficiency achieved by our protocol is essentially the best possible regarding space requirement for the honest parties (concretely, using a pseudorandom function with 128 bit keys to instantiate the protocol, the parties only need a few hundreds bits of storage) and round complexity (a single long round). However, it still requires a large amount of communication, larger than the space bound a of the adversary. A natural question is whether communicating more than a remains necessary if we assume one-way functions.[4] While this is somewhat orthogonal to our work, we still discuss it briefly.

Intuitively, if the adversary can store the entire stream, we would expect them to break the key agreement by virtue of the inexistence of key agreement from one-way functions [IR89]. And indeed, if the total communication c is below $\sqrt{a}$, the protocol can be broken in time roughly quadratic in the honest parties' runtime by the attack of Barak and Mahmoody [BM09] (the attack is only efficient in the number of oracle queries, but it can be made concretely efficient given a one-way function inverter, see e.g. [CFM21]). However, if the total communication c is much closer to a (e.g. $\delta \cdot a$ for some constant $\delta < 1$), the question becomes equivalent to the following problem: is it possible to build key agreement from one-way functions against polynomial-time *linear space* adversaries? Interestingly, this question remains wide open as of today, even if we model the one-way function with a random oracle: all known attacks on key agreements from random oracles [IR89, BM09] appear to inherently require a quadratic amount of space (in the runtime of the honest parties, hence in particular in the communication overhead of the protocol), yet all known variants of Merkle's seminal key agreement protocol in the ROM [Mer74, Mer78] can be broken in linear space.

1.3 Our Techniques

OWFs are sufficient for streaming key agreement. We start with our constructive results. Our construction of gap-optimal and round-optimal UP-KA from OWFs

[4] Of course, in the unbounded processing model, the question is meaningless as the adversary can store everything and get unbounded space afterwards, which makes it essentially an unbounded-space polytime adversary. The question makes sense, however, in the fully-streaming model where the adversarial storage remains bounded after the computation.

is suprisingly simple and conceptually quite natural in hindsight. Our starting point is the unconditional key agreement protocol of Cachin and Maurer [CM97]: in this protocol, Alice streams $C > a$ bits to Bianca, who stores $s \approx \sqrt{C}$ bits of the stream while Alice does the same. By the birthday paradox, the parties get a collision with noticeable probability, and after exchanging the positions of the bits they stored, agree on a key.

We observe that if Alice has the ability to *recompute* the stream, then the parties can store considerably less data: Bianca can store $s \approx \lambda$ bits (where λ is a fixed security parameter) and send her positions to Alice, who recomputes the stream and stores the same bits. The common key is extracted from these bits. This suggests a simple methodology: in our protocol, Alice stores a pseudorandom function (PRF) key $k \in \{0,1\}^\lambda$ and then streams $C > a$ many bits $\mathsf{PRF}(k,1),..,\mathsf{PRF}(k,C)$. Bianca receives the stream and stores λ many of the bits at random locations $\ell_1,..,\ell_\lambda$ which she sends to Alice once her streaming phases ended. They both set their key to be

$$\mathsf{key} := \mathsf{Ext}(\mathsf{PRF}(k,\ell_1)\|..\|\mathsf{PRF}(k,\ell_\lambda)). \tag{1}$$

Except for an additional game-hop based on the PRF, the security analysis of this protocol is analogous to [CM97] and, conceptually, captures that in space $a \ll C$, the adversary only has a small probability p to have stored the information about $\mathsf{PRF}(k,\ell_i)$ and thus, its advantage is upper-bounded by p^λ. Furthermore, the protocol can be made everlasting secure using the bounded storage extractor of Vadhan [Vad04]: instead of outputting the key, Alice creates a stream of length $2a$ and both parties use the key k obtained from the computational protocol as extraction seed.[5]

Eventually, in the fully-streaming PPT model (where the adversary remains space-bounded after the protocol), it is clear that it suffices for the PRF above to be secure against space-bounded PPT adversaries. However, while PRFs are known to be equivalent to OWFs [HILL99, GGM84], it is not immediately obvious that *space-bounded* PRFs should be equivalent to *space-bounded* OWFs – and indeed, this does not appear to follow from existing reductions! Nevertheless, by carefully tracking down the space efficiency of the OWF-to-PRG and PRG-to-PRF reductions, we oberve that space-bounded OWFs are actually sufficient (albeit with a loss in space) to build space-bounded *consecutive* PRFs, a simple variant of PRF which restricts the queries to be consecutive integers (which clearly suffices to instantiate our protocol above).

OWFs are necessary for stream-first UP-KA. Conversely, we show that the existence of UP-KA beyond the unconditional

[5] This is very close in spirit to the hybrid-BSM approach discussed in [DM04], where a similar idea is used to convert a "standard" computational key exchange into an everlasting one. However, several works [DM04, HN06] have pointed out that this strategy fails in general. Our setting is slightly different, and is in particular not captured by the impossibility results in [DM04, HN06], and our concrete instantiation can actually be proven secure formally.

regime implies a one-way function. We start with stream-first UP-KA, and show that any such protocol implies a OWF. Our OWF construction follows Impagliazzo and Luby's approach [IL89] who, given a key agreement protocol KA, construct a distributional OWF (dOWF), i.e., a OWF where it is hard to sample a uniformly random pre-image, see Fig. 1. f_{IL} generates a $(\mathsf{transcript}, \mathsf{key})$ from the distribution induced by the key agreement protocol, and then replaces key with a uniform key with probability $\frac{1}{2}$. f_{IL} is a dOWF because a uniformly random pre-image of $(\mathsf{transcript}, \mathsf{key})$ would reveal the bit b, i.e., whether the key key is real or random. Unfortunately, when KA is a streaming key agreement, we cannot claim that f_{IL} is a dOWF, since accessing the entire protocol transcript might allow trivial inversion attacks: For example, in our protocol, described in and before Equation (1), given the entire transcript, one can simply take the indices $\ell_1, .., \ell_\lambda$ which Bianca sent to Alice and then look up the values of $\mathsf{PRF}(k, \ell_i)$ for all $1 \leq i \leq \lambda$ in Alice's message.

$f_{\mathsf{IL}}(b, r_A, r_B, \mathsf{key}')$

$(\mathsf{transcript}, \mathsf{key}) \leftarrow \mathsf{KA}(r_A, r_B)$
if $b = 1$:
  $\mathsf{key} \leftarrow \mathsf{key}'$
return $(\mathsf{transcript}, \mathsf{key})$

Fig. 1. f_{IL}

To circumvent this issue, we rely on the information-theoretic attacker of Dziembowski and Maurer [DM08] (we call it Eve). At a high level, the attacker sample $\mathcal{O}(s)$ views for Bianca consistent with the long stream. The main Theorem of Dziembowski and Maurer (stated in a re-phrased, weaker version as Theorem 21 in our work) states that the view of Eve has large mutual information with the shared key. Equivalently, the distributions induced by the sampling of Eve's view (for a random stream) together with the *short* message (after the long stream) and *either* the shared key or the random key are statistically far. If the protocol is secure, these distributions must be computationally indistinguishable; this suggests a modified distributional OWF f_{DM} (represented on Figure 2) that replaces Alice's stream by the adversary's $\mathcal{E}'_{\mathsf{DM}}s$ view. Here, $\mathsf{short\text{-}transcript}$ denotes the short message from Bianca to Alice.

By Dziembowski-Maurer, when state and key come from a real protocol execution, $f(0, r_A, r_B, (r_{B,1}, .., r_{B,400s}), \mathsf{key}')$ and $f(1, ..)$ are statistically far from one another, so that a uniform inverter of f_{DM} directly yields a distinguisher for the key agreement protocol. Now, given a dOWF, we obtain a PRF via a sequence of MiniCrypt reductions:

$f_{\mathsf{DM}}(b, r_A, r_B, (r_{B,1}, .., r_{B,s}), \mathsf{key}')$

$(\mathsf{stream}, \mathsf{short\text{-}transcript}, \mathsf{key}) \leftarrow \mathsf{KA}(r_A, r_B)$
for $i = 1..400s$:
  $\mathsf{state}_{B,i} \leftarrow B_1(\mathsf{stream}, r_{B,i})$
if $b = 1$:
  $\mathsf{key} \leftarrow \mathsf{key}'$
return $(\mathsf{short\text{-}transcript}, \mathsf{key}, \mathsf{state}_{B,1}, .., \mathsf{state}_{B,400s})$

Fig. 2. The dOWF candidate f_{DM}

$$\text{distributional OWF} \overset{[\text{IL89, Yao82}]}{\Rightarrow} \text{OWF} \overset{[\text{HILL99}]}{\Rightarrow} \text{PRG} \overset{[\text{GGM84}]}{\Rightarrow} \text{PRF}$$

It remains to argue that the resulting PRF is *space-efficient*, e.g., in addition to storing the key k, the PRF uses only uses $|k|$ bits additional space. Since the

original dOWF internally computes the stream of an UP-KA, a straightforward implementation of dOWF might indeed consume a lot of space and so might the PRF constructed from it. However, given any pseudorandom function PRF with key length λ, we know that it consumes at most space $\mathsf{poly}(\lambda)$ for some fixed polynomial poly. Now, based on PRF, define the following space-efficient $\mathsf{PRF}_{\mathsf{SE}}$ with key-length $\lambda_{\mathsf{SE}} := \lambda + \mathsf{poly}(\lambda)$

$$\mathsf{PRF}_{\mathsf{SE}}(k, x) := \mathsf{PRF}(k_{1..\lambda}, x),$$

where $k_{1..\lambda}$ are the first bits of key k. In addition to space $|k|$ to store the key k, $\mathsf{PRF}_{\mathsf{SE}}$ indeed only uses space $|k| \geq \mathsf{poly}(\lambda)$. .

To obtain SM-KA (*i.e.* KA in the fully-streaming PPT model), it suffices to assume a *space-bounded* one-way function (SB-OWF) that is secure against space-bounded PPT adversaries. In order to prove that SM-KA implies SB-OWF, we need to further modify f_{DM} once more for this purpose. Namely, if Alice and Bianca use a lot of randomness, they receive this randomness as a stream[6]. However, the function f_{DM} needs to take all of this randomness as input – which can be potentially larger than the space bound a of the adversary! In contrast, a SB-OWF should be computable using much less space than the space a allocated to the adversary.

A natural idea to circumvent this limitation is to derandomize the input of f_{DM} via a pseudorandom generator. Of course, since we seek to prove the existence of a SB-OWF, we cannot assume a PRG which is already a stronger primitive. Fortunately, it turns out that in this setting, it suffices to rely on a *non-cryptographic* pseudorandom generator for space-bounded algorithms, such as Nisan's PRG for read-once branching programs [Nis90]. A slight technicality remains: we need to argue that the distribution $\{f_{\mathsf{DM}}(b, r_A, r_B, (r_{B,1}, .., r_{B,400s}), \mathsf{key}') : (r_A, r_B) \leftarrow\!\!\$ \{0,1\}^*\}$ is statistically close to the distribution obtained by replacing (r_A, r_B) by the output of a PRG for space-bounded algorithms. Unfortunately, this is not implied by the security of the PRG, since PRG security only implies that it fools distinguishers outputting a *single bit* – that is, it only guarantees that the marginal distributions of each of the output bits are statistically close, but not that the distributions themselves are statistically close (a property called *non-boolean* pseudorandomness in [DI06]). Fortunately, a closer look at the security analysis of Nisan's PRG [Nis90] (with minor modifications of the parameters of the proof) reveals that it actually already is an unconditionally secure non-boolean PRG for space-bounded algorithm, which allows us to conclude.

OWFs versus general streaming key agreement. Eventually, we turn to our last result, summarized in Theorem 4. We follow the round-reduction method introduced in DQW [DQW23] to prove a lower bound on multi-round streaming protocols. Essentially, their approach recursively uses (a variant of) the unconditional attacker of Dziembowski and Maurer [DM08] to convert an ℓ-long-round

[6] this is equivalent to having one-time read access to their random tape, which is the standard way to model probabilistic space-bounded algorithms.

UP-KA KA_ℓ into an $(\ell-1)$-long-round streaming key agreement $\mathsf{KA}_{\ell-1}$, as follows:

- One party, say, Bianca, locally samples $s+1$ states $(\mathsf{st}_1^B, \cdots, \mathsf{st}_s^B, \mathsf{st}_{s+1}^B)$ consistent with her state after the first long round of KA_ℓ. She sends $(\mathsf{st}_1^B, \cdots, \mathsf{st}_s^B)$ to Alice.
- Alice samples an "Alice view" st^A of KA_ℓ consistent with the s states $(\mathsf{st}_1^B, \cdots, \mathsf{st}_s^B)$ received from Bianca.
- Both parties execute the rest of KA_ℓ using st^A and st_{s+1}^B as their state.

It is easy to see that the above yields a correct $(\ell-1)$-long-round protocol $\mathsf{KA}_{\ell-1}$; the crux in the analysis of DQW lies in showing that this round-reduction also preserves security.

Now, to show that a streaming key agreement beyond the unconditional regime implies OWFs, we show that a one-way function inverter can be used to make the DQW round-reduction efficient. At a high level,

- Bianca locally samples a valid transcript T for all the short rounds of KA_ℓ. Then, she samples $(s+1)$ pre-long-round states $(\mathsf{prestate}_1^B, \cdots, \mathsf{prestate}_{s+1}^B)$ consistant with T (using the efficient inverter for distributional OWFs), locally simulates the long stream, and computes in parallel the $s+1$ resulting states $(\mathsf{st}_1^B, \cdots, \mathsf{st}_s^B, \mathsf{st}_{s+1}^B)$, and sends the s first states to Alice.
- Alice samples st^A consistent with $(\mathsf{st}_1^B, \cdots, \mathsf{st}_s^B)$, using again the distributional OWF inverter, and both parties execute the rest of KA_ℓ using st^A and st_{s+1}^B as their state.

Using a dedicated analysis (building upon the methods of DQW), we prove that the above protocol is an $(\ell-1)$-long-round secure streaming key agreement $\mathsf{KA}_{\ell-1}$, with the same adversarial space bound. However, there is a degradation in the honest parties space, which increased from s in KA_ℓ to $\Omega(s^2)$ in $\mathsf{KA}_{\ell-1}$. After ℓ rounds of round-reduction, we obtain a protocol KA_0 with space bound $s' = s^{2^\ell}$ and no long rounds. If $s' < a$, this yields a contradiction. One intuitively expects this strategy to rule out the existence of KA_ℓ with adversarial storage $a > s^{2^\ell}$, which is polynomial as long as ℓ is a constant.

The above high-level sketch leaves several important details under the rug. In particular, for technical reasons, the space loss of our reduction actually grows with the total number r of rounds of the protocol rather than the number ℓ of long rounds; the loss is of the form $s^{\mathsf{polylog}(r)^r} < a$, which remains polynomial as long as r is a constant. Eventually, in the fully-streaming setting, we rely on the inexistence of *space-bounded* OWFs to perform the round-reduction and use in addition an information-theoretic PRG of Nisan [Nis90] to derandomize the space-bounded OWF constructed, which introduces additional technicalities and yields a worse gap (though still polynomial when r is a constant).

1.4 Related Works

The bounded storage model has received significant attention since its introduction by Maurer [Mau92], both in the symmetric setting [Lu02, DR02, ADR02,

DM02, Vad04] and in the public-key setting [CM97, CCM98, Din01, DHRS04, HCR02, DQW23]. Recently, a breakthrough result of Raz on space lower bounds for learning parities [Raz16, Raz17] has led to a renewal of interest for the model [KRT17, GRT18, GZ19, DQW22].

A closely-related, but distinct model compared to our work is the hybrid bounded-storage model (hybrid BSM), introduced in [DM04] and further studied in [HN06]. In the hybrid BSM as in our model, the adversary is space-bounded *and* computationally bounded throughout the execution of the protocol. However, the setting and goal are quite different: in the hybrid BSM, the parties first agree on a shared key via a "standard" computational key-exchange (e.g. the Diffie-Hellman key exchange), and then use the shared key K to agree on which positions to read from a long stream to generate a new key K'. The hope is that even if the standard key-exchange is only computationally secure, since the long stream disappears afterwards, the scheme will enjoy everlasting security, and K' will remain private even if the adversary becomes all powerful afterwards. The work of [DM04] showed (via a contrived counter-example) that this intuition fails to hold in general, and [HN06] proved a general black-box impossibility result for the hybrid-BSM, as well as a positive result in the bounded-storage + random oracle models. We note that, while we also consider everlasting security and computationally bounded adversaries, our setting is different in that we do not use a classical (computational) key agreement combined with an unconditional BSM key agreement; rather, we directly build a streaming key agreement from one-way functions. Other works that discuss combinations of the bounded-storage model with computational assumptions in a different setting include [MST04] (on timestamping in the BSM), [GZ21] (achieving primitives that are impossible to achieve classically by combining the BSM with computational assumptions), and [BS23] (combining BSM with grey-box obfuscation to obtain simulation-secure functional encryption).

Eventually, as we discussed earlier, our work fits in the general program of demonstrating the necessity of one-way functions for various cryptographic protocols in the regime where they cannot exist unconditionally, such as zero-knowledge interactive proofs for NP [OW93] and constant-bias coin flipping [MPS10, HO11, BHT14].

1.5 Organization

In Section 2, we introduce some technical definitions and lemmas. Section 3 introduces our models for streaming key agreement, with either fully-streaming PPT adversaries (SM-KA) or unbounded processing PPT adversaries (UP-KA). Section 4 introduces our construction of UP-KA with small honest space requirement from pseudorandom functions, using a single streaming round, and extends this construction to show a stream-first SM-KA from space-bounded (consecutive) pseudorandom functions. Section 5 provides two converse of our construction, showing that *stream-first* UP-KA beyond the unconditional regime implies one-way functions, and that general UP-KA with a constant number of streaming rounds imply infinitely-often OWFs. Section 6 extends our analysis to

SM-KA using information-theoretic pseudorandom generators for space-bounded computations, obtaining space-bounded OWFs and infinitely-often space-bounded OWFs for stream-first and general SM-KA respectively; it relies on a derandomization lemma which had been observed before, but without a precise quantitative statement. A self-contained proof of this derandomization lemma is included in Appendix C of the full version of this paper Eventually, in Section 7, we fill the remaining gap with respect to our construction by proving that space-bounded one-way functions imply space-bounded consecutive pseudorandom functions.

2 Preliminaries

Definition 5 (Infinitely Often Distributional One-Way Functions). *A function f is a ε infinitely often distributional one-way functions (ε-io-dOWF), if it can be computed in time polynomial in its input size and for infinitely many $\lambda_1 < \lambda_2 < ..$, it holds that for all PPT algorithms $\mathcal{A}$ and large enough j*

$$\mathsf{SD}\left((U_{\lambda_j}, f(U_{\lambda_j})), (\mathcal{A}(1^{\lambda_j}, f(U_{\lambda_j})), f(U_{\lambda_j}))\right) > \varepsilon(\lambda_j),$$

where U_{λ_j} denotes the uniform distribution over $\{0,1\}^{\lambda_j}$.

Remark. We will also use non-uniform ε-io-dOWFs where f can be computed by a non-uniform sequence of polynomial-size circuits.

2.1 Information-Theoretic Tools

Definition 6 (Extractor [NZ96]). *We say that an efficient function $\mathsf{Ext} : \{0,1\}^{\mathsf{SEED}} \times \{0,1\}^n \to \{0,1\}^\ell$ is an (α, ϵ)-extractor if for all random variables (X, Z) such that X is supported over $\{0,1\}^n$ and $\mathrm{H}_\infty(X \mid Z) \geq \alpha$ we have $\mathsf{SD}((Z, S, \mathsf{Ext}(S; X)), (Z, S, U_\ell)) \leq \epsilon$ where S, U_ℓ are uniformly random and independent bit-strings of length d, ℓ respectively.*

Lemma 7 (Extractor [ILL89]). *For $\alpha \geq \ell + 2\log(1/\epsilon)$ and $\mathsf{SEED} \geq n + \ell$, there exist an (α, ϵ)-extractor $\mathsf{Ext} : \{0,1\}^{\mathsf{SEED}} \times \{0,1\}^n \to \{0,1\}^\ell$. Furthermore, such an extractor can be computed in $\mathcal{O}(n)$ time and space.*

Let $h(p)$ be the binary entropy function and h^{-1} its inverse s.th. $p \geq \frac{1}{2}$

Lemma 8 (Bit-Entropy [DQW23] Lemma 3.1). *For $1 \leq \delta \leq 1$, 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 $\mathrm{H}_\infty(X \mid Y) \geq \delta k$, and I is uniformly random over [k] and independent of X, Y then $\mathrm{H}_\infty(X[I] \mid Y, I) \geq -\log(h^{-1}(\delta))$*

Lemma 9 (Jensen). *For all random variables X, $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$.*

3 Key Agreement in the Streaming Model

In this section, we will introduce the notion of key agreement in the streaming model. We start by introducing the notion of streaming algorithm we will use throughout the paper as well as some notational conventions. We further provide security notions both in the fully streaming setting—all parties remain space-restricted at all times—and the unbounded processing setting where parties may temporarily use arbitrary (polynomial) space for processing messages.

Notation and Conventions. An algorithm $\mathcal{A}$ may have input to one or more streamed inputs. We write $\mathcal{A}(a, b)$ to indicate (streaming) access to the ordered tuple (a, b) and $\mathcal{A}(a; b)$ if $\mathcal{A}$ can read independently from streams a and b. Concretely, $\mathcal{A}(\mathsf{st}, x; r)$ indicates that $\mathcal{A}$ can read from a stream containing first the state st and then the transcript x as well as independently read random coins from r. Additionally, we annotate inputs which exceed the memory limit and thus need to be read in a streaming fashion by superscript $^{\mathsf{str}}$, e.g., r^{str}. We write $\mathsf{str}^{\mathsf{str}}.\mathsf{read}(\mathsf{len})$ for reading len bits from a stream and $\mathsf{str}^{\mathsf{str}}.\mathsf{write}(\mathsf{val})$ for writing the value val to the stream. Finally, throughout this paper we stick to the conventions on variable names outlined in Table 1. As is the tradition in key agreement, we denote the adversary by $\mathcal{E}$ (Eve) to avoid confusion with Alice who is abbreviated with A. Note that we often omit the security parameter for succinctness of notation.

λ	: security parameter
s	: space bound for honest parties
C	: Communication stream length
r	: rounds, i.e., nbr. of messages
a	: adversary's space bound

Table 1. Conventions on variable names

3.1 Fully Streaming Model

In the streaming model, algorithms are restricted in the space they use throughout their executions. They can still read from input streams and write to output streams larger than their space bound.

Definition 10 (Streaming algorithm). *Let $s : \mathbb{N} \to \mathbb{N}$ and $c : \mathbb{N} \to \mathbb{N}$ be polynomials in λ. An algorithm $\mathcal{A}$ is an (s, C)-streaming PPT, if it gets the security parameter 1^λ, some input x with $|x| \leq C(\lambda)$ as well as two parallel streams $(\mathsf{str}^{\mathsf{str}}; r^{\mathsf{str}})$ with $|\mathsf{str}^{\mathsf{str}}| \leq C(\lambda)$, outputs a value y and a stream $\mathsf{str}^{\mathsf{str}}_{\mathcal{A}}$ such that*

Efficiency. *$\mathcal{A}$ runs in time polynomial in λ,*
Space-bound. *$\mathcal{A}$ uses at most $s(\lambda)$ bits of storage at any point of time and, in particular, $|y| \leq s(\lambda)$, and*
Stream-bound. $|\mathsf{str}^{\mathsf{str}}_{\mathcal{A}}| \leq C(\lambda)$.

Note that $\mathcal{A}$ does not have further randomness beyond the randomness received as a r^{str}. Further, as $\mathcal{A}$ receives multiple streams as input, it can independently

read from the randomness and is not required to fully read any of the streams it receives.

Definition 11 (Key Agreement in the Streaming Model (SM-KA)). *Let s, C, and r be polynomials in λ. A (s, C, r)-SM-KA protocol KA consists of r (s, C)-streaming PPT $(A_1, B_1, \ldots, A_{r/2}, B_{r/2})$, such that each of the PPT P has syntax*

$$(\mathsf{st}', x'^{\mathsf{str}}, \mathsf{key}) \leftarrow P(\mathsf{st}, x^{\mathsf{str}}; r^{\mathsf{str}})$$

with $|\mathsf{key}| = \lambda$ and together, they satisfy correctness (cf. Definition 12).

When running a (s, C, r)-SM-KA protocol, A_1 and B_2 take as input an empty state, and since A_1 sends the first message, A_1 also takes as input an empty x^{str}. Only the last stages $A_{r/2}$ and $B_{r/2}$ return a key, but for uniformity of syntax and w.l.o.g., we let all protocol stages return a key. With this understanding of the syntax, we define a protocol as follows:

$$\frac{\mathsf{KA}({r_A}^{\mathsf{str}}, {r_B}^{\mathsf{str}}) = (r^{\mathsf{str}}_{A,1}, ..r^{\mathsf{str}}_{A,r/2}, r^{\mathsf{str}}_{B,1}, .., r^{\mathsf{str}}_{B,r/2})}{}$$

$\mathsf{st}_A \leftarrow []; \quad \mathsf{st}_B \leftarrow []; \quad x^{\mathsf{str}}_{B,0} \leftarrow []$

for $i = 1, .., r/2$ **do**

$(\mathsf{st}_A, x^{\mathsf{str}}_{A,i}, \mathsf{key}_{A,i}) \leftarrow A_i(\mathsf{st}_A, x^{\mathsf{str}}_{B,i}; r^{\mathsf{str}}_{A,i})$

$(\mathsf{st}_B, x^{\mathsf{str}}_{B,i}, \mathsf{key}_{B,i}) \leftarrow B_i(\mathsf{st}_B, x^{\mathsf{str}}_{A,i}; r^{\mathsf{str}}_{B,i})$

$\mathsf{key}_A \leftarrow \mathsf{key}_{A,r/2}$

$\mathsf{key}_B \leftarrow \mathsf{key}_{B,r/2}$

$x^{\mathsf{str}} \leftarrow (x^{\mathsf{str}}_{A,1}, x^{\mathsf{str}}_{B,1}, .., x^{\mathsf{str}}_{A,r/2}, x^{\mathsf{str}}_{B,r/2})$

return $(x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B)$

Definition 12 (Correctness). *Let s, C, and r be polynomials in λ. An (s, C, r)-SM-KA is ϵ_{KA}-correct if for all but finitely many λ*

$$\Pr_{r^{\mathsf{str}}}\left[\mathsf{key}_A = \mathsf{key}_B : (x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B) \leftarrow \mathsf{KA}(r^{\mathsf{str}})\right] = 1 - \epsilon_{\mathsf{KA}}.$$

If ϵ_{KA} is negl we sometimes omit it.

Security of a (s, C, r)-SM-KA protocol has an additional parameter a which bounds the length of the adversary's storage and requires that Alice's key is indistinguishable from random (and thus, by correctness, so is Bianca's key).

Definition 13 (Fully Streaming Security). *Let s, C, r and a be polynomials in λ. KA is a (s, C, r, a)-SM-KA δ_{KA}-secure protocol if it is a (s, C, r)-SM-KA and for all but finitely many λ and all (a, Cr)-streaming PPT $\mathcal{E}$, the advantage* $\mathsf{Adv}^{\mathsf{stream}}_{\mathsf{KA},\mathcal{E}}(\lambda) :=$

$$\left| \begin{array}{c} \Pr_{r^{\mathsf{str}}, r_{\mathcal{E}}{}^{\mathsf{str}}}\left[1 = \mathcal{E}(1^\lambda, \mathsf{key}_A; x^{\mathsf{str}}; r_{\mathcal{E}}) : (x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B) \leftarrow \mathsf{KA}(r^{\mathsf{str}})\right] \\ - \Pr_{r^{\mathsf{str}}, r_{\mathcal{E}}{}^{\mathsf{str}}, \mathsf{key}}\left[1 = \mathcal{E}(1^\lambda, \mathsf{key}; x^{\mathsf{str}}; r_{\mathcal{E}}) : (x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B) \leftarrow \mathsf{KA}(r^{\mathsf{str}})\right] \end{array} \right|$$

is upper bounded by δ_{KA}. If δ_{KA} is negl in λ we sometimes omit it.

3.2 Unbounded Processing Model

In addition, we relax the space-bound and define *unbounded processing* algorithms. Unbounded processing algorithms may use arbitrary (polynomial in λ) space, however their output y still has to satisfy $|y| \leq s(\lambda)$.

Definition 14 (Unbounded Processing Protocol). *Let $s : \mathbb{N} \to \mathbb{N}$ and $c : \mathbb{N} \to \mathbb{N}$ be polynomials in λ. An protocol Π is an (s, C)-unbounded-processing PPT, if it consists of rounds $(\mathsf{st}, \mathsf{str}^{\mathsf{str}}, x) \leftarrow \mathsf{send}(1^\lambda, \mathsf{st}), (\mathsf{st}, x) \leftarrow \mathsf{receive}(1^\lambda, \mathsf{st}, \mathsf{str}^{\mathsf{str}})$ where Alice and Bianca alternate in running the* send *and* receive *algorithms such that*

Efficiency. send *and* receive *run in time polynomial in λ,*
Stream-bound. $|\mathsf{str}^{\mathsf{str}}| \leq C(\lambda)$.
Small State and Output. *The state* st *and output x is bounded by $s(\lambda)$*

Definition 15 (Key Agreement in the Unbounded Processing (UP-KA)). *Let s, C, and r be polynomials in λ. A (s, C, r)-UP-KA protocol* KA *consists of r UP round functions PPT $(\mathsf{send}_{A,1}, \mathsf{receive}_{B,1}), \ldots, (\mathsf{send}_{B,r}, \mathsf{receive}_{A,r})$, with syntax*

$$(\mathsf{st}', x'^{\mathsf{str}}, \mathsf{key}) \leftarrow \mathsf{send}(\mathsf{st}; r^{\mathsf{str}})$$
$$(\mathsf{st}', \mathsf{key}) \leftarrow \mathsf{receive}(\mathsf{st}, x^{\mathsf{str}}; r^{\mathsf{str}})$$

with $|\mathsf{key}| = \lambda$ and together, they satisfy correctness (cf. Definition 12). Regrouping, we also consider the sequence $(A_1 := (\mathsf{receive}_{A,1}, \mathsf{send}_{A,1}), B_1 := (\mathsf{receive}_{B,1}, \mathsf{send}_{B,1})), \ldots, (A_{r/2} := (\mathsf{receive}_{A,r/2}, \mathsf{send}_{A,r/2}), B_{r/2} := (\mathsf{receive}_{B,r/2}, \mathsf{send}_{B,r/2}))$ where the first receive algorithm and the last send algorithm is empty.

For security in the Unbounded Processing setting, we need to split the adversary in one instance per round $\mathcal{E}_1, \ldots, \mathcal{E}_r$ and final distinguishing adversary $\mathcal{E}$. Similarly to the round algorithms, $\mathcal{E}$ are required to be PPT in λ and follow the syntax $\mathsf{st} \leftarrow \mathcal{E}(\mathsf{st}, \mathsf{str}^{\mathsf{str}}; r^{\mathsf{str}})$ where $|\mathsf{st}| \leq a(\lambda)$.

Definition 16 (Unbounded Processing (UP-KA) security). *Let s, C, and r be polynomials in λ.* KA *is a (s, C, r, a)-UP-KA δ_{KA} secure protocol if it is a (s, C, r)-UP-KA and for all but finitely many λ and for all PPT $\mathcal{E}_1, \ldots, \mathcal{E}_r$ outputting a state $\mathsf{st}_{\mathcal{E}_i}$ with $\left|\mathsf{st}_{\mathcal{E}_i}\right| \leq a(\lambda)$ and all PPT $\mathcal{E}$, the advantage*

$$\mathsf{Adv}^{\mathsf{unbound}}_{\mathsf{KA},\mathcal{E}_r,\mathcal{E}}(\lambda) := \left|\Pr\left[1 = \mathcal{E}(1^\lambda, \mathsf{key}_A; \mathsf{st}_{\mathcal{E}_r}; r_{\mathcal{E}})\right] \Pr\left[1 = \mathcal{E}(1^\lambda, \mathsf{key}; \mathsf{st}_{\mathcal{E}_r}; r_{\mathcal{E}})\right]\right|$$

is upper bounded by δ_{KA}, where the probabilities are taken over sampling r^{str}, the (implicit) randomness of $\mathcal{E}, \mathcal{E}_1, \ldots, \mathcal{E}_r$ and, for the second probability, $\mathsf{key}, x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B) \leftarrow \mathsf{KA}(r^{\mathsf{str}})$ and $\mathsf{st}^{\mathsf{str}}_{\mathcal{E}_i} \leftarrow \mathcal{E}_i(\mathsf{st}_{\mathcal{E}_{i-1}}, x^{\mathsf{str}}_i)$ and $(x^{\mathsf{str}}_1, \cdots, x^{\mathsf{str}}_{2r}) \leftarrow x^{\mathsf{str}})$.If δ_{KA} is negl in λ we sometimes omit it.

Normal Form

We additionally place the following additional constraints on protocols in both the fully streaming and unbounded processing model:

Short Rounds are Short. In particular, all short messages in a protocol fit within honest parties space $s(\lambda)$

No Consecutive Long Rounds. Between two long (streaming) messages, we require at least one short message

4 Constructing Key Agreement

In this section we present our $(s, C, r = 2)$-SM-KA and $(s, C, r = 2)$-UP-KA protocols. Recall, that in contrast to [CM97] the stream is generated by Alice using a PRF. Consequently, Bianca can choose a single index and send it to Alice who can reconstruct the bit using the (small) PRF key. To produce a large, uniform key, we parallel compose the basic protocol $\widetilde{\mathcal{O}}(\lambda)$ times and extract the key using a seed chosen by Bianca. In our proof, we rely on consecutive PRFs—a weaker notion of PRFs which can only be accessed on consecutive values—as this notion suffices for our proofs and can be constructed *space efficient* from one-way functions.

4.1 Consecutive PRFs

While the reduction for the GGM construction of PRFS [GGM84] requires space linear in the number of queries, the reduction can be made space-efficient under the restriction of only allowing sequential queries. We formally discuss this reduction in the last section of the full version and use consecutive PRFs in our construction.

Definition 17. *A function* $f : \{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}$ *is a sequential PRF if for all probabilistic adversaries* $\mathcal{A}$ *running in time* $\mathsf{poly}(\lambda)$

$$\left| \begin{array}{r} \Pr_{k \leftarrow\$ \{0,1\}^\lambda}\left[1 = \mathcal{A}^{\mathsf{EVAL}^0_{f,k}(\cdot)}(1^\lambda)\right] \\ -\Pr_{k \leftarrow\$ \{0,1\}^\lambda}\left[1 = \mathcal{A}^{\mathsf{EVAL}^1_{f,k}(\cdot)}(1^\lambda)\right] \end{array} \right| \leq \mathsf{negl}(\lambda)$$

$\mathsf{EVAL}^b_{f,k}(i)$

if $\mathsf{ctr} = \bot$ **then** $\mathsf{ctr} \leftarrow 0$
assert $i = \mathsf{ctr} + 1$
$\mathsf{ctr} \leftarrow i$
if $b = 0$ **then** $y \leftarrow f(k, i)$
else $y \leftarrow\$ \{0,1\}$
return y

4.2 SB-PRF ⇒ Fully Streaming Key-Agreement

For simplicity we set the desired length of the produced keys to λ matching the security parameter of the consecutive PRF.

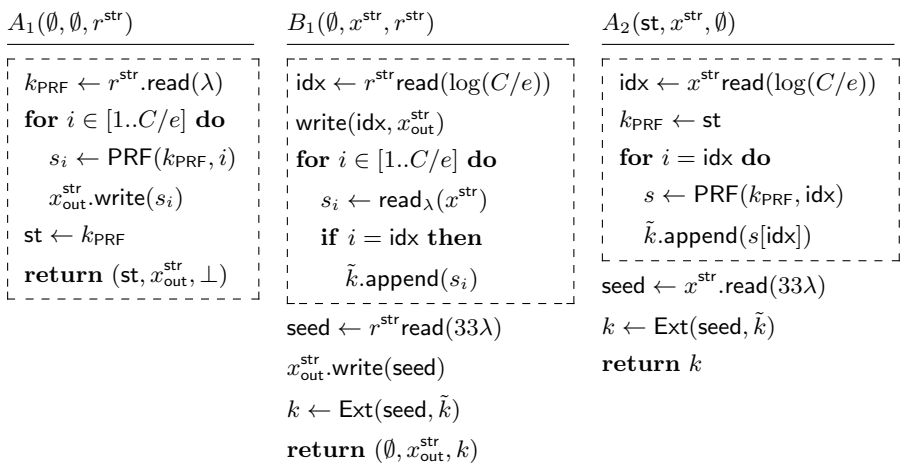

Fig. 3. Honest Protocol $\mathsf{KA} := ((A_1, B_1), (A_2))$ where boxed parts are repeated $e := 30\lambda$ times in parallel

Theorem 18 (SB-PRFs ⇒ fully streaming key-agreement (SM-KA)). *Let* PRF *be a Consecutive SB-PRF* $\{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}$ *which can be evaluated in space* s_{PRF} *and* Ext *and* $(3\lambda, \lambda)$*-extractor* $\{0,1\}^{30\lambda} \times \{0,1\}^{31\lambda} \to \{0,1\}^\lambda$*. Then* $\mathsf{KA}_{Fig.3}$ *is a* $(s, C, r = 2)$*-SM-KA protocol with perfect correctness and honest user space* $s = \mathcal{O}(\lambda \cdot \log(C) + s_{\mathsf{PRF}})$ *and* $(s, C, r = 2, a = \frac{C}{60\lambda})$*-SM-KA security*

The proof is fairly standard and omitted in the conference version but included in the full version of this paper.

4.3 PRF ⇒ Unbounded Processing Key-Agreement

Theorem 19 (PRFs ⇒ unbounded processing key-agreement (UP-KA)). *Let* PRF *be a PRF* $\{0,1\}^\lambda \times \{0,1\}^\lambda \to \{0,1\}$ *and* Ext *and* $(3\lambda, \lambda)$*-extractor* $\{0,1\}^{30\lambda} \times \{0,1\}^{31\lambda} \to \{0,1\}^\lambda$*. Then* $\mathsf{KA}_{Fig.3}$ *is a* $(s, C, r = 2)$*-UP-KA protocol with perfect correctness and honest user space* $s = \mathcal{O}(\lambda)$ *and* $(s, C, r = 2, a = \frac{C}{60\lambda})$*-UP-KA security*

Proof Sketch. The unbounded processing model places fewer restrictions on the honest parties, and thus we can avoid the requirement for the PRF to allow evaluation in restricted space. For security, observe that the only point where we used the space restriction on the adversary was to bound the size of the adversary's space after receiving the stream from Alice. As Eve is space bounded between rounds in the unbounded processing model as well, the same argument applies. □

4.4 Arbitrary Output Length and Everlasting Security

[DQW23] show that it is possible to obtain large keys at the cost of one additional round: Alice streams C uniform bits and both parties use their derived key k as seed to extract a large key K using a bounded storage extractor [Vad04] with good locality. This transformation applies directly to our construction as well, with additional space cost $\mathcal{O}(|K|)$ for the honest parties. We further observe that this step is secure against unbounded adversaries and the seed k can be published after the protocol terminates thus resulting in a protocol with *everlasting* security.

5 Unbounded Processing: UP-KA implies dOWFs

5.1 Stream-first key agreement ⇒ dOWF

We start by considering the *stream first* setting, where Alice first sends a long streaming message to Bianca, and afterwards, Bianca sends a short message to Alice. As outlined in Section 1.3, if KA is a strong stream-first UP-KA protocol that is secure against adversaries with large enough space, then f_{DM} (cf. Fig. 2) is a dOWF.

Theorem 20 (Stream-first UP-KA ⇒ dOWF). *Let* KA *be a stream-first* (s, C, r, a)*-UP-KA protocol with* $a \geq 400s^2$*, correctness error* $\epsilon_{\mathsf{KA}} \leq \frac{1}{400}$ *and security gap* $\delta_{\mathsf{KA}} \leq \frac{1}{5}$*, then* f_{DM} *is an* $\epsilon_{\mathcal{I}}$*-dOWF for any constant* $\epsilon_{\mathcal{I}} \leq \frac{1}{10}$*.*

The proof of Theorem 20 builds on the following Dziembowski-Maurer (DM) theorem on the function f_{DM} which is induced by a key agreement protocol KA. For $b \in \{0,1\}$, we define the distributions $f_{\mathsf{DM}}(b, R)$ by sampling $r = (r_A, r_B, r_{B,1}, .., r_{B,s}, \mathsf{key}')$ uniformly at random and returning $f_{\mathsf{DM}}(b, r)$.

Theorem 21 (Dziembowski-Maurer). *If* KA *is a stream-first* (s, C, r, a)*-UP-KA or SM-KA protocol with* $\epsilon_{\mathsf{KA}} \leq \frac{1}{400}$*-correctness error. Then for all large enough* λ*,*

$$\mathsf{SD}(f_{\mathsf{DM}}(0, R), f_{\mathsf{DM}}(1, R)) \geq \frac{9}{10}$$

Remark. Dziembowski and Maurer prove a stronger version of Theorem 21 which precisely characterizes the entropy of the key rather than only its statistical distance from a uniformly random key. The above is a re-statement of Dziembowski-Maurer (DM) in the DQW fully streaming/unbounded processing model, simplified for our application. A self-contained proof of Theorem 21 is included in Appendix A of the full version of this paper

In addition to DM, we will use the following useful claim throughout this and the next section to lower bound the advantage of a distinguisher induced by a uniform inverter.

Claim 1. *For* $b \in \{0,1\}$*, let* X^b *be two arbitrary distributions, and let* Y *be the distribution which samples* b *uniformly and then returns* $z \leftarrow\!\!\$\ X^b$*. Then,*

$$\mathbb{E}_{b, z \leftarrow\!\!\$ Y}[\Pr_{b', z' \leftarrow\!\!\$ Y}[b' = b \mid z' = z]] \geq (\mathsf{SD}(X^0, X^1))^2$$

We prove Claim 1 in Appendix B of the full version and now use Claim 1 to prove Theorem 20.

Proof of Theorem 20. Assume towards contradiction that f_{DM} is not an $\epsilon_{\mathcal{I}}$-dOWF for $\epsilon_{\mathcal{I}} = \frac{1}{10}$. Then, there exists a PPT inverter $\mathcal{I}$ such that for infinitely many security parameters

$$\mathsf{SD}((B, R, f_{\mathsf{DM}}(B,R)), (\mathcal{I}(f_{\mathsf{DM}}(B,R)), f_{\mathsf{DM}}(B,R))) < \epsilon_{\mathcal{I}}, \tag{2}$$

where R is the uniform input $(r_A^{\mathsf{str}}, r_B^{\mathsf{str}}, (r_{B,1}^{\mathsf{str}}, .., r_{B,400s}^{\mathsf{str}}), \mathsf{key}')$. Let $\mathcal{E}_{\mathcal{I}}$ be the distinguisher which given z, runs

$$(b, r) \leftarrow\!\$\, \mathcal{I}(z);\ \textbf{return}\ b.$$

We construct the the following adversary $\mathcal{E}_{\mathcal{I}}$ against the stream-first (s, C, r, a)-UP-KA protocol KA: Adversary $\mathcal{E}_{\mathcal{I}}$ prepares the running of $400s$ different copies of Bianca, each with its own randomness stream $r_{B,j}^{\mathsf{str}}$ which $\mathcal{E}_{\mathcal{I}}$ does not store, but instead generates (in parallel) on the fly as needed. When $\mathcal{E}_{\mathcal{I}}$ receives stream, adversary $\mathcal{E}_{\mathcal{I}}$ computes $\mathsf{st}_{B,j} \leftarrow B_1(\mathsf{stream}, r_{B,j}^{\mathsf{stream}})$ in parallel for all $1 \le j \le 400s$ and stores $\mathsf{st}_{B,1}, .., \mathsf{st}_{B,400s}$. Next, $\mathcal{E}_{\mathcal{I}}$ receives $\mathsf{short\text{-}transcript}, \mathsf{key}$, runs

$$(b^*, r) \leftarrow\!\$\, \mathcal{I}(\mathsf{st}_{B,1}, .., \mathsf{st}_{B,400s}, \mathsf{short\text{-}transcript}, \mathsf{key})$$

and returns b^*. In the proof, we denote by $\mathcal{E}_{\mathcal{U}}$ the analogous (inefficient) adversary which, instead of the (efficient) $\mathcal{I}$, runs $\mathcal{U}$ that returns a *perfectly* uniform pre-image of z under f_{DM}.

Space. The adversary $\mathcal{E}_{\mathcal{I}}$ samples $400s$ Bianca states, each of which requires space s. Thus, in the streaming phase, $\mathcal{E}_{\mathcal{I}}$ runs in space $400s^2$. Note that $\mathcal{I}$ is run after receiving the stream has terminated, so that its space consumption does not affect $\mathcal{E}$'s space limitation while receiving.

Advantage. Now, we can lower bound the advantage $\mathsf{Adv}^{\mathsf{unbound}}_{\mathsf{KA},\mathcal{E}_{\mathcal{I}}}(\lambda)$ as follows:

$$\begin{aligned}
&|\Pr_{r_A{}^{\mathsf{str}}, r_B{}^{\mathsf{str}}, r_{\mathcal{E}_{\mathcal{I}}}{}^{\mathsf{str}}, \mathsf{key}}\left[1 = \mathcal{E}_{\mathcal{I}}(1^\lambda, \mathsf{key}; x^{\mathsf{str}}; r_{\mathcal{E}_{\mathcal{I}}}) : (x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B) \leftarrow \mathsf{KA}(r_A{}^{\mathsf{str}}, r_B{}^{\mathsf{str}})\right] \\
&- \Pr_{r_A{}^{\mathsf{str}}, r_B{}^{\mathsf{str}}, r_{\mathcal{E}_{\mathcal{I}}}{}^{\mathsf{str}}}\left[1 = \mathcal{E}_{\mathcal{I}}(1^\lambda, \mathsf{key}_A; x^{\mathsf{str}}; r_{\mathcal{E}_{\mathcal{I}}}) : (x^{\mathsf{str}}, \mathsf{key}_A, \mathsf{key}_B) \leftarrow \mathsf{KA}(r_A{}^{\mathsf{str}}, r_B{}^{\mathsf{str}})\right]| \\
&= |\Pr_{r,\mathcal{I}}[(1,*) = \mathcal{I}(f_{\mathsf{DM}}(1,r))] - \Pr_{r,\mathcal{I}}[(1,*) = \mathcal{I}(f_{\mathsf{DM}}(0,r))]| \\
&\overset{(\dagger)}{\ge} |\Pr_{r,\mathcal{U}}[(1,*) = \mathcal{U}(f_{\mathsf{DM}}(1,r))] - \Pr_{r,\mathcal{U}}[(1,*) = \mathcal{U}(f_{\mathsf{DM}}(0,r))]| - 4\epsilon \\
&= |\Pr_{r,\mathcal{U}}[(1,*) = \mathcal{U}(f_{\mathsf{DM}}(1,r))] + \Pr_{r,\mathcal{U}}[(0,*) = \mathcal{U}(f_{\mathsf{DM}}(0,r))] - 1| - 4\epsilon \\
&\ge 2\mathbb{E}_{b,r}[\Pr_{r',b'}[b' = b \mid f_{\mathsf{DM}}(b',r') = f_{\mathsf{DM}}(b,r)]] - 1 - 4\epsilon_{\mathcal{I}} \\
&\overset{\text{Cl. 1}}{\ge} 2\mathsf{SD}(f_{\mathsf{DM}}(1,R), f_{\mathsf{DM}}(0,R))^2 - 1 - 4\epsilon_{\mathcal{I}} \\
&\overset{\text{T. 21}}{\ge} 2\left(\frac{9}{10}\right)^2 - 1 - 4\epsilon_{\mathcal{I}} > 2\left(\frac{9}{10}\right)^2 - 1 - \frac{4}{10} > \frac{3}{5} - \frac{2}{5} = \frac{1}{5} \ge \delta_{\mathsf{KA}}
\end{aligned}$$

where (†) follows, because $\mathcal{I}$ approximates the uniform distribution ϵ well, but since the statistical distance in (2) is over the choice of b as well, the loss is doubled, and then, it is further doubled since we have a loss for each term. □

5.2 Sampling st conditioned on q copies of itself

Let us open up one of the ideas of Dziembowski-Maurer (DM) underlying their proof of Theorem 21, since it is a useful tool for generalizing Section 5.1 to key agreement protocols with multiple streaming rounds.

DM show that q equally distributed Bianca states already contain most of the information of Bianca's *actual* state. Using DM's ideas, Dodis, Quach and Wichs (DQW) strengthen the lemma into stating that, in fact, *sampling* a Bianca state conditioned on q of his own states will yield an almost equally distributed state. Both DM and DQW state their lemmas in more general terms and we follow their tradition here. Namely, consider a pair of *jointly* distributed random variables (Z, Y). First sample Y and then q random variables $Z_1, .., Z_q$, each of which is sampled according to the distribution of Z conditioned on Y. Now, the claim is that if we sample Z' according to Z conditioned on $Z_1, .., Z_i$ (rather than on Y) for a suitable $1 \leq i \leq q$, then these two distribution are close. In the lemma below, X is equal to $f(Y)$ for some (potentially probabilistic) function f.

Lemma 22 (DQW). $\exists i : 1 \leq i \leq q$ *such that*

$$\mathsf{SD}((X, Z, Z_1, .., Z_i), (X, Z', Z_1, .., Z_i)) \leq \sqrt{\frac{H(X)}{2(q+1)}}$$

Remark. Intuitively, sampling Z' conditioned on *more* information about Y should be useful to decrease the statistical distance and hence, one might think that choosing $i = q$ is always a valid choice. However, the proof of Lemma 22 currently just relies on the chain rule for mutual information and only shows that such an i *exists*. Note that we stated Theorem 21 with $400s$ instead of i, because statistical distance can only increase when adding more variables, but the same argument does not directly apply here.[7]

When we apply Lemma 22 in Section 5.3, X is Alice's state, $Z_1, .., Z_q$ are Bianca states and Z and Z' are also Bianca states. Since Alice's state size is upper-bounded by s, we also have $H(X) \leq s$, and choosing $q = \frac{s^{1+2m}}{2}$ yields an upper bound of $\sqrt{\frac{H(X)}{2(q+1)}} \leq \sqrt{\frac{s}{s^{1+2m}}} \leq \frac{1}{s^m}$.

5.3 dOWFs via round reduction

Section 5.1 shows that stream-first UP-KA (with large enough space gap) implies a dOWF. This result is of interest on its own and didactically meaningful, since all subsequent analyses of success probability follow a similar template, but have additional steps or additional conceptual ideas. Nevertheless, the most important

[7] More precisely, for any function g, $\mathsf{SD}((X, X'), (Y, Y')) \geq \mathsf{SD}(g(X, X'), g(Y, Y'))$ and choosing g to be a projection on the first variable shows $\mathsf{SD}((X, X'), (Y, Y')) \geq \mathsf{SD}(X, Y)$. Unfortunately, $(X, Z', Z_1, .., Z_i)$ is not a projection of $(X, Z', Z_1, .., Z_q)$, since Z' is conditioned on $Z_1, .., Z_i$ and $Z_1, .., Z_q$, respectively.

role of the result for stream-first UP-KA is that it establishes as *base case* for an inductive argument that we carry out in this section.

Concretely, we follow the DQW round reduction template: DQW prove that if there is an r-message UP-KA/SM-KA protocol, then there is also an $r-1$-message UP-KA/SM-KA protocol with slightly worse parameters. Arguing by induction, we then obtain that any r-message UP-KA/SM-KA protocol with large enough parameters implies a stream-first UP-KA/SM-KA protocol (possibly with an empty first stream, if all messages end up being short), which we already know implies a dOWF.

The DQW round reduction technique operates in the information-theoretic setting, and we would like to adopt their technique to the computational setting. Unfortunately, several sampling operations in the DQW round reduction are inefficient. Thus, we prove that an r-message UP-KA/SM-KA protocol can be transformed into an $r-1$-message SM-KA protocol with slightly worse parameters *or* that an infinitely-often dOWF exists. Applying the argument iteratively, we obtain that an r-message UP-KA/SM-KA protocol implies an *or* statement over $r+1$ possible candidates for an infinitely often (io) dOWF.

Conceptual idea. To present the conceptual idea behind the DQW round reduction technique and our variant of it, we now describe the protocol transformation using inefficient reverse sampling and then subsequently replace inefficient reverse sampling by an inverter $\mathcal{I}$ similarly as in the previous section.

We denote $A_1, B_1, A_2, B_2, ..$ the code of Alice and Bianca in the original protocol and add an overline for the transformed protocol $\overline{A}_1, \overline{B}_1, \overline{A}_2, \overline{B}_2, ...$ Assume w.l.o.g. that Alice sends the first message x_A.

Short messages. If the message x_A is short, then we can just "move it into Bianca's computation" and have Alice perform reverse sampling to compute her state later, i.e., we obtain a protocol where Bianca sends the following first message:

$\overline{B}_1$	$\overline{A}_1(\overline{x_B})$	$A_1^{\mathsf{mess}}(r_A{}^{\mathsf{str}})$
$(\mathsf{st}_A, x_A) \leftarrow\!\!\$\ A_1$	$\mathsf{parse}\ (x_A, x_B) \leftarrow \overline{x_B}$	$(\mathsf{st}_A, x_A) \leftarrow A_1(r_A{}^{\mathsf{str}})$
// The randomness r_A is implicit.	$r_A{}^{\mathsf{str}} \leftarrow\!\!\$\ (A_1^{\mathsf{mess}})^{-1}(x_A)$	**return** x_A
$(\mathsf{st}_B, x_B) \leftarrow\!\!\$\ B_1(x_A)$	$(\mathsf{st}_A, x_A) \leftarrow A_1(r_A)$	
$\overline{x_B} \leftarrow (x_A, x_B)$	$(\mathsf{st}'_A, x'_A) \leftarrow\!\!\$\ A_2(\mathsf{st}_A, x_B)$	
return $(\mathsf{st}_B, \overline{x_B})$	**return** (st'_A, x'_A)	

We now prove that the function A_1^{mess} which maps Alice's randomness to Alice's message x_A is a dOWF—or that we have a protocol with one round less. Namely, if A_1^{mess} is *not* a dOWF, then we obtain a new $r-1$ message protocol where we replace the inefficient inverse sampling $r_A{}^{\mathsf{str}} \leftarrow\!\!\$\ (A_1^{\mathsf{mess}})^{-1}(x_A)$ of Alice's state by an efficient sampler. W.l.o.g., we consider protocols in a normal form, where no two streaming rounds follow onto each other, but rather, there are always short rounds in between.

Lemma 23 (Short messages). *Let $m > 0$ be a constant. Let KA be a (s, C, r, a)-UP-KA with $a > \frac{s^{2+2m}}{2}$, correctness error ϵ_{KA}, security δ_{KA}, where the length of the first message is bounded by $\frac{s^{2+2m}}{2}$. Then, either A_1^{mess} is an $\epsilon_{\mathcal{I}}$-io-dOWF, or there exists an inverter $\mathcal{I}$ for A_1^{mess} such that for all but finitely many λ,*

$$\mathsf{SD}((A_1^{\mathsf{mess}}(R), R), (A_1^{\mathsf{mess}}(R), \mathcal{I}(A_1^{\mathsf{mess}}(R)))) < \epsilon_{\mathcal{I}}, \tag{3}$$

where R is a uniform sample of $r_A{}^{\mathsf{str}}$. Moreover, $\overline{\mathsf{KA}}$ defined by $\overline{A_1}$ (replacing $(A_1^{\mathsf{mess}})^{-1}$ by $\mathcal{I}$), $\overline{B_1}$ and $\overline{A_j} := A_{j+1}$, $\overline{B_j} := B_j$ for $j > 1$ is a $(s^{2+2m}, C, r-1, a)$-UP-KA with correctness error $\epsilon_{\overline{\mathsf{KA}}} = \epsilon_{\mathsf{KA}} + \epsilon_{\mathcal{I}}$ and security $\delta_{\overline{\mathsf{KA}}} = \delta_{\mathsf{KA}} + \epsilon_{\mathcal{I}}$.

Remark. We only obtain an infinitely often (io) dOWF rather than a dOWF, because we need $\mathcal{I}$ to successfully invert A_1^{mess} on all but finitely many λ for security and correctness to hold.

Proof. Communication length C. Note that for $\overline{\mathsf{KA}}$, the transcript

$$(\overline{x_{B,1}}, \overline{x_{A,1}}, \overline{x_{B,2}}, ..)$$

is equal to

$$((x_{A,1}, x_{B,1}), x_{A,2}, x_{B,2}, ..$$

and thus, the communication complexity of the two protocols are identical (we omit constant costs for bracketing $(x_{A,1}, x_{B,1})$).

Normal form. The protocol is still in normal form: If $x_{B,1}$ in KA was a stream, then $x_{A,2}$ is short. Now, $\overline{x_{B,1}} = (x_{A,1}, x_{B,1})$ is a stream, too, and $\overline{x_{A,1}} = x_{A,2}$ is still short. If $x_{B,1}$ in KA was short, then $\overline{x_{B,1}} = (x_{A,1}, x_{B,1})$ is still short, since $|x_{A,1}| + |x_{B,1}| \leq \frac{s^{2+2m}}{2} + s < s^{2+2m}$.

Space bounds of honest parties. Since we only modified the behaviour of the parties on non-streaming rounds, their behaviour in streaming rounds remains the same, using the same space bounds as before. Moreover, the $\overline{B_1}$ only stores a state $\overline{\mathsf{st}_B} = \mathsf{st}_B$ of size s. Finally, for $\overline{A_1}$, the $\mathsf{receive}_{\overline{A_1}}$ can just store the message $\overline{x_{B,1}} = (x_{A,1}, x_{B,1})$ because $|x_{A,1}| + |x_{B,1}| \leq \frac{s^{2+2m}}{2} + s < s^{2+2m}$ is lower than its space bound.

Correctness. The distribution of Bianca's key in KA and $\overline{KA}$ is identical, but the distribution of Alice's key might change by at most $\epsilon_{\mathcal{I}}$ due to the statistical distance of the sampler.

Security. Let $\mathcal{E}$ be a PPT adversary against $\overline{\mathsf{KA}}$ and assume towards contradiction that $\mathcal{E}$'s advantage $\delta_{\mathcal{E}} > \delta_{\mathsf{KA}} + \epsilon_{\mathcal{I}}$. Since the transcript $(\overline{x_{B,1}}, \overline{x_{A,1}}, \overline{x_{B,2}}, ..)$ of $\overline{\mathsf{KA}}$ is equal to $((x_{A,1}, x_{B,1}), x_{A,2}, x_{B,2}, ..)$ and since $|x_{A,1}| \leq a$, the reduction $\mathcal{R}_{\mathcal{E}}$ against KA can store $x_{A,1}$ and then run the first stage of $\mathcal{E}$ only once $\mathcal{R}_{\mathcal{E}}$ also receives $x_{B,1}$. Subsequently, $\mathcal{R}_{\mathcal{E}}$ proceeds exactly as $\mathcal{E}$.

$\mathcal{R}_{\mathcal{E}}$'s simulation of KA is up to $\epsilon_{\mathcal{I}}$-far from the distribution of $\overline{\mathsf{KA}}$, since Alice's state in $\overline{\mathsf{KA}}$ has statistical distance at most $\epsilon_{\mathcal{I}}$ from her state in KA. Hence, we obtain that $\mathcal{R}_{\mathcal{E}}$ has advantage

$$\delta_{\mathcal{R}_{\mathcal{E}}} \geq \delta_{\mathcal{E}} - \epsilon_{\mathcal{I}} > \delta_{\mathsf{KA}} - \epsilon_{\mathcal{I}} + \epsilon_{\mathcal{I}} = \delta_{\mathsf{KA}}$$

against KA and we reach a contradiction. □

Long messages. Now, in the case that Alice's first message x_A is long, Bianca cannot generate Alice's first message x_A, send it to her together with his own message x_B and then Alice performs reverse sampling given x_A, because this would destroy the normal form of the protocol, since x_B is a short message and might be followed by a long message. Therefore, we would like to replace x_B by a message which is also *short*.

Lemma 22 gives us a tool how Alice can sample an almost well-distributed state given something short, namely s copies of her own state. Indeed, Bianca could sample s copies of Alice's state in the unbounded pre-processing model (and assuming a suitably efficient inverter). However, for consistency with the next section, we implement a different strategy here that will also work in the fully streaming setting.

As we have already seen in f_{DM}, Bianca can efficiently sample several of *her own* states. Very surprisingly, DQW show that if Alice samples her state conditioned on i copies of *Bianca's* state $\mathsf{st}_{B,1}$,..,$\mathsf{st}_{B,i}$, her state is actually well-distributed, yielding the following transformed protocol, where i is the index guaranteed by Lemma 22.

$\overline{B}_1$

$(\mathsf{st}_A, x_A^{\mathsf{str}}) \leftarrow\!\$\ A_1$
  // Running Alice.
for $j = 1..i$:
  $(\mathsf{st}_{B,j}, x_{B,j}) \leftarrow\!\$\ B_1(x_A^{\mathsf{str}})$
  // Sampling $i \leq s^{a+2c}$ Bianca states.
$z \leftarrow (\mathsf{st}_{B,1}, x_{B,1}, .., \mathsf{st}_{B,i}, x_{B,i})$
$(r_A^{\mathsf{str}}, \_) \leftarrow\!\$\ (\mathbf{B_1} \circ A_1)^{-1}(z)$
  // Re-sample conditional
  // randomness for Alice.
$(\mathsf{st}'_A, x_A^{'\mathsf{str}}) \leftarrow A_1(r_A^{\mathsf{str}})$
  // Running Alice.
$(\overline{\mathsf{st}_B}, x_B) \leftarrow\!\$\ B_1(x_A^{'\mathsf{str}})$
  // Sampling a fresh Bianca state.
$\overline{x_B} \leftarrow (z, x_B)$
return $(\overline{\mathsf{st}_B}, \overline{x_B})$

$\overline{A}_1(\overline{x_B})$

$(z, x_B) \leftarrow \mathsf{parse}\ \overline{x_B}$
$(r_A^{\mathsf{str}}, \_) \leftarrow\!\$\ (\mathbf{B_1} \circ A_1)^{-1}(z)$ // Re-sample cond.
  // rand. for Alice.
$(\mathsf{st}_A, x_A^{\mathsf{str}}) \leftarrow A_1(r_A^{\mathsf{str}})$ // Running Alice.
$(\mathsf{st}'_A, x'_A) \leftarrow\!\$\ A_2(\mathsf{st}_A, x_B)$
$\overline{\mathsf{st}_A} \leftarrow \mathsf{st}'_A$
$\overline{x_A} \leftarrow x'_A$
return $(\overline{\mathsf{st}_A}, \overline{x_A})$

$(\mathbf{B_1} \circ A_1)(r_A^{\mathsf{str}}, (r_{B,1}, .., r_{B,i}))$

$(\mathsf{st}_A, x_A^{\mathsf{str}}) \leftarrow A_1(r_A^{\mathsf{str}})$ // Running Alice.
for $j = 1..i$:
  $(\mathsf{st}_{B,j}^{\mathsf{str}}, x_{B,j}^{\mathsf{str}}) \leftarrow B_1(x_A^{\mathsf{str}}; r_{B,j})$
  // Computing $i \leq s^{1+2m}$ Bianca states.
return $(\mathsf{st}_{B,1}, x_{B,1}, .., \mathsf{st}_{B,i}, x_{B,i})$

The function $(\mathbf{B_1} \circ A_1)$ is a natural candidate for a dOWF since the above protocol transformation only works if $(\mathbf{B_1} \circ A_1)$ is *not* an (infinitely often) dOWF. Before turning to an efficient implementation of the protocol using an efficient inverter for $\mathbf{B_1} \circ A_1$, let us briefly consider why these inefficient versions of $\overline{A}_1$ and $\overline{B}_1$ would yield a good *joint* distribution of Alice and Bianca states. The DQW key idea here is that *both* Alice and Bianca, sample their state conditioned

on $x_{B,1}, \mathsf{st}_{B,1}, .., x_{B,i}, \mathsf{st}_{B,i}$ *only.* Therefore, Alice's state is *perfectly* distributed as in the original protocol by the definition of conditional sampling. Now, to argue that the *joint* distribution of Alice's and Bianca's state is close to the original distribution, we invoke Lemma 22 on Bianca's state to conclude that sampling conditioned on $x_{B,1}, \mathsf{st}_{B,1}, .., x_{B,i}, \mathsf{st}_{B,i}$ yields a sample that is statistically close to the original distribution. We now make these arguments formal.

Lemma 24 (Long messages). *Let $m > 0$ be a constant. Let KA be a (s, C, r, a)-UP-KA with $a > s^{2+2m}$, correctness error ϵ_{KA} and security δ_{KA}, where the length of the first message is greater than $\frac{s^{2+2m}}{2}$. $\mathbf{B_1} \circ A_1$ is a non-uniform $\epsilon_{\mathcal{I}}$-io-dOWF, or there exists an inverter $\mathcal{I}$ for $\mathbf{B_1} \circ A_1$ such that for all but finitely many λ,*

$$\mathsf{SD}((\mathbf{B_1} \circ A_1(R), R), (\mathbf{B_1} \circ A_1(R), \mathcal{I}(\mathbf{B_1} \circ A_1(R)))) < \epsilon_{\mathcal{I}}, \tag{4}$$

where R is a uniform sample of $r_A^{\mathsf{str}}, (r_{B,1}, .., r_{B,i})$ and i is the index guaranteed by Lemma 22. Moreover, $\overline{\mathsf{KA}}$ defined by $\overline{A_1}$ (replacing $(\mathbf{B_1} \circ A_1)^{-1}$ by $\mathcal{I}$), $\overline{B_1}$ and $\overline{A_j} := A_{j+1}$, $\overline{B_j} := B_j$ for $j > 1$ is a $(s^{2+2m}, C, r-1, a)$-UP-KA with correctness error $\epsilon_{\overline{\mathsf{KA}}} = \epsilon_{\mathsf{KA}} + 2\epsilon_{\mathcal{I}} + \frac{1}{s^m}$ and security $\delta_{\overline{\mathsf{KA}}} = \delta_{\mathsf{KA}} + 2\epsilon_{\mathcal{I}} + \frac{1}{s^m}$.

Remark. The non-uniformity is induced by the need to know the index i, which cannot be computed efficiently and which might be a different index for each security parameter. Thus, the non-uniform advice is $\mathcal{O}(\log \lambda)$ when Lemma 24 is applied once and $\mathcal{O}(r \log \lambda)$ when Lemma 24 is applied recursively r times.

Proof. Communication length C. Since we assumed that $|x_A| > \frac{s^{2+2m}}{2}$, the communication complexity of the protocol decreased, since instead of x_A, we now send up to $\frac{s^{1+2m}}{2}$ Bianca states each of which has size at most s, so overall, we replaced a message of size $|x_A| > \frac{s^{2+2m}}{2}$ by a message of size $\leq \frac{s^{2+2m}}{2}$.

Normal form. The protocol is still in normal form. Since $x_{A,1}$ is long, $x_{B,1}$ in KA is short. Now, $\overline{x_{B,1}}$ is still short, since it contains $|x_{B,1}| \leq s$ bits as well as up to $\frac{s^{1+2m}}{2}$ many Bianca states, each of which are of size at most s, so the overall length of $\overline{x_{B,1}}$ is bounded by $\frac{s^{2+2m}}{2} + s < s^{2+2m}$ and thus below the new space bound for honest parties.

Space bounds of honest parties. Analogously to the short message case, the parties' behaviour in rounds other than the first remains the same, using the same space bounds as before. Moreover, $\overline{B_1}$ only stores a state $\overline{\mathsf{st}_B} = \mathsf{st}_B$ of size s. And similarly to before, for $\overline{A_1}$, the $\mathsf{receive}_{\overline{A_1}}$ can just store the message $\overline{x_{B,1}} = (x_{A,1}, x_{B,1})$ because $\frac{s^{2+2m}}{2} + s < s^{2+2m}$ is lower than its space bound.

Security. Let $\mathcal{E}$ a PPT adversary with space-bound a against $\overline{\mathsf{KA}}$ and assume towards contradiction that $\mathcal{E}$ has advantage $\delta_{\overline{\mathsf{KA}}} > \delta_{\mathsf{KA}} + 2\epsilon_{\mathcal{I}} + \frac{1}{s^m}$. We construct a new PPT adversary $\mathcal{R}_{\mathcal{E}}$ against KA. As in the previous section, after the first message of $\overline{\mathsf{KA}}$, the reduction $\mathcal{R}_{\mathcal{E}}$ just runs $\mathcal{E}$, we thus now focus on $\mathcal{R}_{\mathcal{E}}$'s simulation of the first message $\overline{x_{B,1}}$ of $\overline{\mathsf{KA}}$. Upon receiving x_A^{str} as a stream, $\mathcal{R}_{\mathcal{E}}$ computes $\frac{s^{1+2m}}{2}$ many Bianca states in parallel as follows:

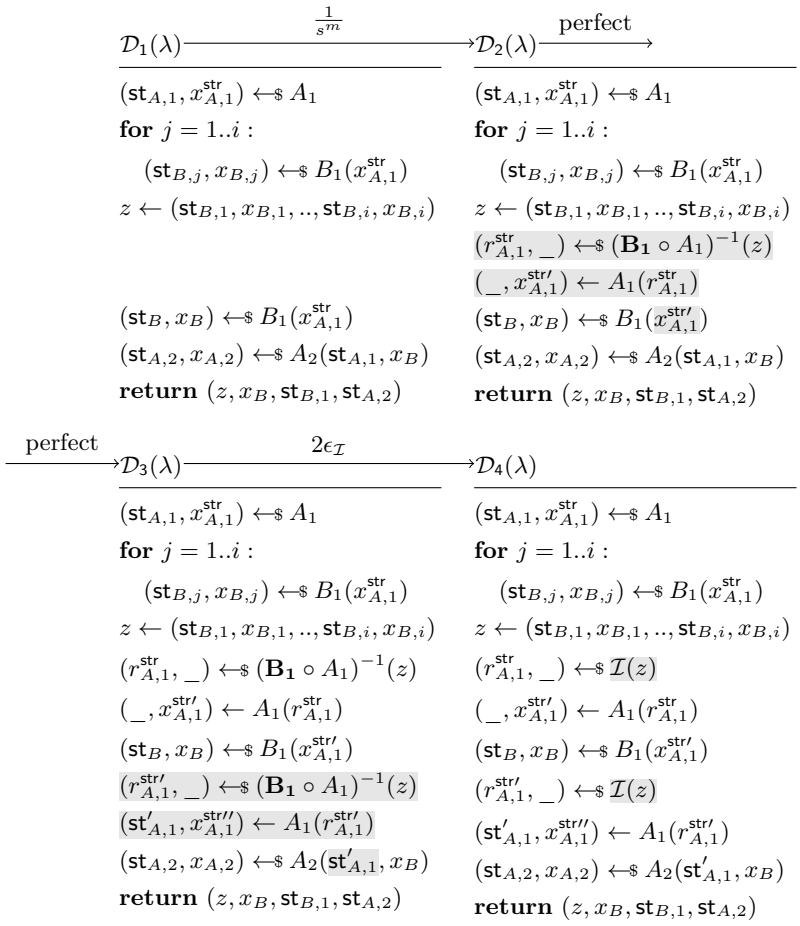

Fig. 4. Hybrids for Lemma 24

for $j = 1..i$:
$(\mathsf{st}_{B,j}, x_{B,j}) \leftarrow\$ B_1(x^{\mathsf{str}}_A)$
// Sampling $i \leq s^{1+2m}$ Bianca states.
$z \leftarrow (\mathsf{st}_{B,1}, x_{B,1}, .., \mathsf{st}_{B,i}, x_{B,i})$

Since $a > \frac{s^{2+2m}}{2}$, $\mathcal{R}_{\mathcal{E}}$ can store those. Next, upon receiving receiving Bianca's message x_B, $\mathcal{R}_{\mathcal{E}}$ runs $\mathcal{E}$ on (z, x_B), and from there just runs $\mathcal{E}$. We argue about the statistical distance of $\mathcal{R}_{\mathcal{E}}$'s simulation by game-hopping. In Fig. 4, the upper-left column describes how the joint distribution of $(z, x_B, \mathsf{st}_B, \mathsf{st}_{A,2})$ is generated in $\mathcal{E}$'s simulation, and the lower-right column describes how the joint distribution of $(z, x_B, \mathsf{st}_B, \mathsf{st}_{A,2})$ is generated in $\overline{\mathsf{KA}}$.

From the 1st to 2nd column, we replace sampling of Bianca's state and message by conditional inverse sampling. By Lemma 22, the statistical distance

is at most $\frac{1}{s^m}$, cf. discussion in Section 5.2. From the 2nd to 3rd column, we sample Alice's state conditionally. This step is perfect. Finally, from the 3rd to 4th column, we replace the 2 perfect inverse samplings by inverse samplings by $\mathcal{I}$; the statistical distance is $< 2\epsilon_{\mathcal{I}}$. Thus, we obtain that $\mathcal{R}_{\mathcal{E}}$ has advantage greater than

$$\delta_{\overline{\mathsf{KA}}} - \frac{1}{s^m} - 2\epsilon_{\mathcal{I}} > \delta_{\mathsf{KA}} + \frac{1}{s^m} + 2\epsilon_{\mathcal{I}} - \frac{1}{s^m} - 2\epsilon_{\mathcal{I}} = \delta_{\mathsf{KA}}$$

and we reach a contradiction.

Correctness. As we analyzed security via a *statistical* sequence of game-hops, the analysis implies that the overall distribution of the protocol's behaviour changes by $2\epsilon_{\mathcal{I}} + \frac{1}{s^m}$ and thus, the correctness error increases by the same amount.

□

5.4 Conclusion

We proved in the unbounded processing PPT model that, when space gaps are large enough, r-message UP-KA can be transformed into $r-1$-message UP-KA, or an io-dOWF exists. Moreover, we proved that stream-first UP-KA implies dOWFs. Now, we put these transformations together into the following theorem which states that r-message UP-KA with large enough space gaps implies io-dOWF. Note that for the following theorem, no efforts have been made to optimize the parameters.

Theorem 25 (UP-KA ⇒ io-dOWF). *Let r be a constant. Let KA be a (s, C, r, a)-UP-KA with $a \geq s^{(3m)^r}$, where m is a constant such that $m \geq \log_s 10^4 r$ for large enough security paramters. Then, there exists a non-uniform ϵ-io-dOWF with $\epsilon = \frac{1}{10^4 r}$.*

Proof. Space of honest parties. Lemma 23 and Lemma 24 both increase the space of honest parties from s to s^{2+2m}. Thus, after $r-1$ applications of either of the lemmas, we obtain space $s^{(2+2m)^{r-1}} \leq s^{(3m)^{r-1}}$. Now, Theorem 20 requires the adversary to have space at least $(s^{(3m)^{r-1}})^2$. which is indeed lower than $a = s^{(3m)^r}$.

Correctness and Security. Each application of Lemma 23 and Lemma 24 reduces correctness and security by at most $2\epsilon_{\mathcal{I}} + \frac{1}{s^m}$. Theorem 20 requires the correctness error of the stream-first protocol to be at most $\frac{1}{400}$ and the security gap to be at most $\frac{1}{5}$. The increase of the correctness error and security gap are both dominated by $2r \cdot \epsilon_{\mathcal{I}} = \frac{2}{10^4}$. Additionally, we get a term that is upper bounded by $r \cdot \frac{1}{s^m} \leq \frac{1}{10^4}$, and $\frac{3}{10^4} < \frac{1}{400}$, which is also smaller than $\frac{1}{5}$.

□

6 Fully streaming: SM-KA implies SB-dOWFs

6.1 A derandomization lemma

We start by stating a derandomization lemma, which states (in essence) that if an algorithm $\mathcal{A}$ takes as input a random stream r of length $|r| \gg s$ (and

possibly some additional short input), runs in time t, uses space at most s, and returns an output of size s, then this algorithm can be derandomized into an algorithm $\mathsf{Der}(\mathcal{A})$ that uses slightly larger space $\Theta(s \cdot \log t)$ but takes as input only $O(s \cdot \log t)$ bits of randomness, such that the output distribution of $\mathsf{Der}(\mathcal{A})$ is statistically close to that of $\mathcal{A}$. Looking ahead, our results in the fully-streaming model will build upon this lemma to convert the OWFs constructed in Section 5 into SB-OWFs.

Lemma 26 (Derandomization). *There exist a global constant c and a transformation* Der *such that the following holds: Let $\mathcal{A}$ be a deterministic algorithm, taking as input a uniformly random string $r \in \{0,1\}^t$ (its randomness), running in time t and space s and producing an output of length $\leq s$. Then if $2^s \geq 8t^2 \log t$,*

$$\mathsf{SD}(\mathcal{A}(r), \mathsf{Der}(\mathcal{A})(r_{\mathsf{short}})) \leq 2^{-s},$$

where $r \leftarrow\!\!\$\ \{0,1\}^t$, $r_{\mathsf{short}} \leftarrow\!\!\$\ \{0,1\}^{s(\log t + c)}$, and $\mathsf{Der}(\mathcal{A})$ runs runs in time at most $c \cdot t \cdot \log t \cdot s^2$ and uses space at most $56 \log t \cdot s + c \cdot s$

In Appendix C of the full version , we prove Lemma 26. We stress that the proof is not from us: it basically follows the analysis of Nisan from [Nis90]. However, Lemma 26 does not follow from any Theorem in Nisan's paper, but rather follows from the *proof* of Theorem 1 in Nisan's paper. For completeness, we therefore reproduce this proof here, following the presentation given in the lecture notes of Ryan O'Donnell[8], with some suitable adaptation of the parameters to derive Lemma 26. In essence, the core observation is that Nisan's pseudorandom generator for low-space algorithms satisfies a stronger property: it fools *non-boolean distinguishers* that output a string $x \in \{0,1\}^s$ (where fooling means that the output distribution of the distinguishers given outputs of Nisan's PRG is statistically close to their output distribution given true random coins). We also note that this property has been observed before: it was mentioned in passing in the works of Nisan and Zuckerman [NZ96] and of Dubrov and Ishai [DI06].

6.2 Stream-first key agreement ⇒ SB-dOWF

We now adapt the proof of Theorem 20 to the fully streaming setting. Naturally, the resulting dOWF is only secure against space-bounded adversaries.

Definition 27 (Space-bounded Distributional One-Way Functions). *A function $f : \{0,1\}^n \to \{0,1\}^m$ is a (s, a, ε)-*space-bounded distributional one way function *(SB-dOWF), if the following conditions hold.*

Space-bounded Efficiency. *f can be computed in time polynomial in λ and in space $s(\lambda)$. Furthermore, we impose $m(\lambda) \leq s(\lambda)$.*

Security. *For every polynomial-time adversary $\mathcal{A}$ which uses at most $s(\lambda)$ bits of storage, we have that for all large enough λ,*

$$\mathsf{SD}\left((U_n, f(U_n)), (\mathcal{A}(1^\lambda, f(U_n)), f(U_n))\right) \geq \varepsilon(\lambda). \tag{5}$$

[8] https://www.cs.cmu.edu/~odonnell/complexity/docs/lecture16.pdf

Remark. Analogously to infinitely often OWFs (Definition 5), we will later also use (s, a, ε)-io-SB-dOWFs, where (5) only holds for infinitely many λ. Jumping ahead, the infinitely often property will later be needed In the long message lemma (included in the full version), which is the analogous statement to Lemma 24. Again, we will use a dOWF inverter to construct a protocol, and correctness requires the inverter to invert correctly on all but finitely many λ. We will also consider non-uniform versions of (s, a, ε)-io-SB-dOWFs, where f can be computed by a non-uniform sequence of polynomial-size circuits of width $\leq a$. Again, as for Lemma 24, the non-uniform advice will be the index i guaranteed by Lemma 22.

Different from Theorem 20, we first need to modify f_{DM}, since it encodes the protocol into a deterministic function and the participants could use (much) more randomness than space, increasing the input length of f_{DM} beyond the adversary's space bound. We thus invoke Lemma 26 (derandomization) and, instead consider $f_{\mathsf{stream-1st}}(b, r_{\mathsf{short}})$ which is a derandomized version of f_{DM} and has (almost) the same output distribution despite using significantly less randomness.

Theorem 28 (Stream-first SM-KA $\Rightarrow$ SB-dOWF). *Let* KA *be a stream-first (s, C, r, a) SM-KA in with $a = \mathcal{O}(s^{4+k})$ for some constant $k > 0$, correctness error $\epsilon_{\mathsf{KA}} \leq \frac{1}{400}$, security gap $\delta_{\mathsf{KA}} < \frac{1}{10^3}$ and A and B running in overall time t. Additionally, we assume (for convenience) that $56 \cdot 400 \log(400ts) + 400c) \leq s$, and $400^3 c \log(400st) \leq s$. Then,*

$$f_{\mathsf{stream-1st}}(b, r_{\mathsf{short}}) := \begin{cases} \mathsf{Der}(f_{\mathsf{DM}}(0, \cdot))(r_{\mathsf{short}}) & \text{if } b = 0 \\ \mathsf{Der}(f_{\mathsf{DM}}(1, \cdot))(r_{\mathsf{short}}) & \text{if } b = 1 \end{cases}$$

with $r_{\mathsf{short}} \in \{0,1\}^{s^3}$, is an $(s', a, \epsilon_{\mathcal{I}})$-SB-dOWF with space $s' = s^3$, time ts^3 for any $\epsilon_{\mathcal{I}} \leq \frac{1}{5}$.

The proof of Theorem 28 is analogous to the proof of Theorem 20, with an additional (small) loss for the derandomization inaccuracy as well as an additional increase in space due to the derandomization. The details can be found in the full version of this paper.

6.3 Conclusion

We proved in the fully streaming PPT model that, when space gaps are large enough, r-message SM-KA can be transformed into $r-1$-message SM-KA, or a non-uniform io-SB-dOWF exists. Moreover, we proved that stream-first SM-KA implies SB-dOWFs. Now, analogously to Theorem 25, we put these transformations together into the following theorem which states that r-message SM-KA with large enough space gaps implies a non-uniform io-SB-dOWF. Note that for the following theorem, no efforts have been made to optimize the parameters.

Theorem 29 (SM-KA $\Rightarrow$ io-SB-dOWF). *Let r and w be constants. Let* KA *be a (s, C, r, a)-SM-KA with $a \geq s^{(3mw)^r}$, where m is a constant such that $m \geq \log_s 10^4 wr$ for large enough security parameters. Then, there exists a non-uniform (s_f, a_f, ϵ_f)-io-SB-dOWF f with $a_f = s_f^w$ and with $\epsilon_f = \frac{1}{10^4 r}$.*

Discussion. DQW use a (short) common reference string (CRS) as a technical tool in their round reduction arguments for the fully streaming protocol, which allows them to rely on setup routines that are not necessarily space-bounded—note that this is the only reason that the CRS is useful, because else, the CRS could just be generated and sent by the party who generates the first message. In addition to being a technical tool, including a CRS makes their result stronger, since DQW also rule out protocols where the CRS is not (space-)efficiently computable. We, in turn, do not achieve such a stronger result, since we seek to build (space-)efficiently computable SB-dOWF. Thus, we cannot use a CRS as a technical tool where we move (space-)inefficient computations that the transformation incurs. However, using derandomization (Lemma 26) as well as efficient inverters (which exist assuming that a certain function is *not* an SB-dOWF), our results also show that in our setting, all transformations can be implemented in a space-efficient manner. It is conceivable that analogous derandomization arguments also apply to DQW (using inefficient inverters), but we did not investigate this question in sufficient depth to make this claim.

7 SB-dOWFs implies SB-PRFs

Impagliazzo and Luby (IL [IL89]) show that distributional OWFs imply weak OWFs via universal hashing, and that Yao shows that weak OWFs imply standard OWFs via parallel repetition, cf. [Yao82, Gol01], then several constructions transform OWFs into PRGs [HILL99, HRV13, VZ12], and finally, Goldreich, Goldwasser and Micali transform PRGs into PRFs [GGM84]. The goal of this section is to show that the aforementioned reductions are sufficiently tight in space so that, together with Theorem 29, we obtain the following theorem for SM-KA.

Theorem 30 (SM-KA $\Rightarrow$ SB-PRFs). *There exists a universal constant u such that the following holds: let r and w be arbitrary constants. Let* KA *be a (s, C, r, a)-SM-KA with $a \geq s^{u\cdot(3mw)^r}$, where m is a constant such that $m \geq \log_s 10^4 wr$ for large enough security parameters. Then, there exists a non-uniform (s_f, a_f)-io-SB-consecutive-PRF F with $a_f = s_f^w$.*

Theorem 30 follows mainly by inspection, and observing that the reductions mentioned above preserve the fine-grained space hardness of the notions pretty well. Due to space constraints the proof is only included in the full version of the paper.

Acknowledgments

Chris Brzuska was supported by the Research Council of Finland grant No. 358950. Geoffroy Couteau was supported by the French Agence Nationale de la Recherche (ANR), under grant ANR-20-CE39-0001 (project SCENE), by the France 2030 ANR Project ANR22-PECY-003 SecureCompute, and by ERC project OBELiSC (Grant 101115790). Christoph Egger was supported by the European Commission

under the Horizon2020 research and innovation programme, Marie Sklodowska-Curie grant agreement No 101034255. Willy Quach was supported by the Israel Science Foundation (Grant No. 3426/21), and by the Horizon Europe Research and Innovation Program via ERC Project ACQUA (Grant 101087742).

References

ADR02. Yonatan Aumann, Yan Zong Ding, and Michael O Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48(6):1668–1680, 2002.

BHT14. Itay Berman, Iftach Haitner, and Aris Tentes. Coin flipping of *any* constant bias implies one-way functions. In David B. Shmoys, editor, *46th ACM STOC*, pages 398–407. ACM Press, May / June 2014.

BIKM99. Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin. One-way functions are essential for single-server private information retrieval. In *31st ACM STOC*, pages 89–98. ACM Press, May 1999.

BM09. Boaz Barak and Mohammad Mahmoody-Ghidary. Merkle puzzles are optimal - an $O(n^2)$-query attack on any key exchange from a random oracle. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 374–390. Springer, Berlin, Heidelberg, August 2009.

BOL85. Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 408–416. IEEE, 1985.

BS23. Mohammed Barhoush and Louis Salvail. Functional encryption in the bounded storage models. *CoRR*, abs/2309.06702, 2023.

CCM98. Christian Cachin, Claude Crépeau, and Julien Marcil. Oblivious transfer with a memory-bounded receiver. In *39th FOCS*, pages 493–502. IEEE Computer Society Press, November 1998.

CFM21. Geoffroy Couteau, Pooya Farshim, and Mohammad Mahmoody. Black-box uselessness: Composing separations in cryptography. In James R. Lee, editor, *ITCS 2021*, volume 185, pages 47:1–47:20. LIPIcs, January 2021.

CGKS95. Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *36th FOCS*, pages 41–50. IEEE Computer Society Press, October 1995.

CM97. Christian Cachin and Ueli M. Maurer. Unconditional security against memory-bounded adversaries. In Burton S. Kaliski, Jr., editor, *CRYPTO'97*, volume 1294 of *LNCS*, pages 292–306. Springer, Berlin, Heidelberg, August 1997.

DHRS04. Yan Zong Ding, Danny Harnik, Alon Rosen, and Ronen Shaltiel. Constant-round oblivious transfer in the bounded storage model. In Moni Naor, editor, *TCC 2004*, volume 2951 of *LNCS*, pages 446–472. Springer, Berlin, Heidelberg, February 2004.

DI06. Bella Dubrov and Yuval Ishai. On the randomness complexity of efficient sampling. In Jon M. Kleinberg, editor, *38th ACM STOC*, pages 711–720. ACM Press, May 2006.

Din01. Yan Zong Ding. Oblivious transfer in the bounded storage model. In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 155–170. Springer, Berlin, Heidelberg, August 2001.

DM02. Stefan Dziembowski and Ueli M. Maurer. Tight security proofs for the bounded-storage model. In *34th ACM STOC*, pages 341–350. ACM Press, May 2002.

DM04. Stefan Dziembowski and Ueli M. Maurer. On generating the initial key in the bounded-storage model. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 126–137. Springer, Berlin, Heidelberg, May 2004.

DM08. Stefan Dziembowski and Ueli Maurer. The bare bounded-storage model: The tight bound on the storage requirement for key agreement. *IEEE Transactions on Information Theory*, 54(6):2790–2792, 2008.

DQW22. Yevgeniy Dodis, Willy Quach, and Daniel Wichs. Authentication in the bounded storage model. In Orr Dunkelman and Stefan Dziembowski, editors, *EUROCRYPT 2022, Part III*, volume 13277 of *LNCS*, pages 737–766. Springer, Cham, May / June 2022.

DQW23. Yevgeniy Dodis, Willy Quach, and Daniel Wichs. Speak much, remember little: Cryptography in the bounded storage model, revisited. In Carmit Hazay and Martijn Stam, editors, *EUROCRYPT 2023, Part I*, volume 14004 of *LNCS*, pages 86–116. Springer, Cham, April 2023.

DR02. Yan Zong Ding and Michael O Rabin. Hyper-encryption and everlasting security. In *STACS 2002: 19th Annual Symposium on Theoretical Aspects of Computer Science Antibes-Juan les Pins, France, March 14–16, 2002 Proceedings 19*, pages 1–26. Springer, 2002.

GGM84. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions (extended abstract). In *25th FOCS*, pages 464–479. IEEE Computer Society Press, October 1984.

Gol01. Oded Goldreich. *Foundations of Cryptography: Basic Tools*, volume 1. Cambridge University Press, Cambridge, UK, 2001.

GRT18. Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *50th ACM STOC*, pages 990–1002. ACM Press, June 2018.

GZ19. Jiaxin Guan and Mark Zhandry. Simple schemes in the bounded storage model. In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part III*, volume 11478 of *LNCS*, pages 500–524. Springer, Cham, May 2019.

GZ21. Jiaxin Guan and Mark Zhandry. Disappearing cryptography in the bounded storage model. In Kobbi Nissim and Brent Waters, editors, *TCC 2021, Part II*, volume 13043 of *LNCS*, pages 365–396. Springer, Cham, November 2021.

HCR02. Dowon Hong, Ku-Young Chang, and Heuisu Ryu. Efficient oblivious transfer in the bounded-storage model. In Yuliang Zheng, editor, *ASIACRYPT 2002*, volume 2501 of *LNCS*, pages 143–159. Springer, Berlin, Heidelberg, December 2002.

HILL99. Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

HN06. Danny Harnik and Moni Naor. On everlasting security in the hybrid bounded storage model. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP 2006, Part II*, volume 4052 of *LNCS*, pages 192–203. Springer, Berlin, Heidelberg, July 2006.

HO11. Iftach Haitner and Eran Omri. Coin flipping with constant bias implies one-way functions. In Rafail Ostrovsky, editor, *52nd FOCS*, pages 110–119. IEEE Computer Society Press, October 2011.

HRV13. Iftach Haitner, Omer Reingold, and Salil P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. *SIAM J. Comput.*, 42(3):1405–1430, 2013.

IL89. Russell Impagliazzo and Michael Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *30th FOCS*, pages 230–235. IEEE Computer Society Press, October / November 1989.

ILL89. Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *21st ACM STOC*, pages 12–24. ACM Press, May 1989.

IR89. Russell Impagliazzo and Steven Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM STOC*, pages 44–61. ACM Press, May 1989.

KRT17. Gillat Kol, Ran Raz, and Avishay Tal. Time-space hardness of learning sparse parities. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *49th ACM STOC*, pages 1067–1080. ACM Press, June 2017.

Lu02. Chi-Jen Lu. Hyper-encryption against space-bounded adversaries from on-line strong extractors. In Moti Yung, editor, *CRYPTO 2002*, volume 2442 of *LNCS*, pages 257–271. Springer, Berlin, Heidelberg, August 2002.

Mau92. Ueli M. Maurer. A universal statistical test for random bit generators. *Journal of Cryptology*, 5(2):89–105, January 1992.

Mer74. R. Merkle. C.s. 244 project proposal. In *Facsimile available at http://www.merkle.com/1974.*, 1974.

Mer78. Ralph C Merkle. Secure communications over insecure channels. *Communications of the ACM*, 21(4):294–299, 1978.

MPS10. Hemanta K. Maji, Manoj Prabhakaran, and Amit Sahai. On the computational complexity of coin flipping. In *51st FOCS*, pages 613–622. IEEE Computer Society Press, October 2010.

MST04. Tal Moran, Ronen Shaltiel, and Amnon Ta-Shma. Non-interactive timestamping in the bounded storage model. In Matthew Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 460–476. Springer, Berlin, Heidelberg, August 2004.

Nis90. Noam Nisan. Psuedorandom generators for space-bounded computation. In *22nd ACM STOC*, pages 204–212. ACM Press, May 1990.

NZ96. Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

OW93. Rafail Ostrovsky and Avi Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *[1993] The 2nd Israel Symposium on Theory and Computing Systems*, pages 3–17. IEEE, 1993.

Raz16. Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In Irit Dinur, editor, *57th FOCS*, pages 266–275. IEEE Computer Society Press, October 2016.

Raz17. Ran Raz. A time-space lower bound for a large class of learning problems. In Chris Umans, editor, *58th FOCS*, pages 732–742. IEEE Computer Society Press, October 2017.

Vad04. Salil P. Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17(1):43–77, January 2004.

VZ12. Salil P. Vadhan and Colin Jia Zheng. Characterizing pseudoentropy and simplifying pseudorandom generator constructions. In Howard J. Karloff and Toniann Pitassi, editors, *44th ACM STOC*, pages 817–836. ACM Press, May 2012.

Yao82. Andrew Chi-Chih Yao. Theory and applications of trapdoor functions (extended abstract). In *23rd FOCS*, pages 80–91. IEEE Computer Society Press, November 1982.