Constant-Round Oblivious Transfer in the Bounded Storage Model

Yan Zong Ding[1], Danny Harnik[2], Alon Rosen[3] and Ronen Shaltiel[4]

[1] College of Computing, Georgia Institute of Technology. 801 Atlantic Drive, Atlanta, GA 30332-0280. Email: ding@cc.gatech.edu. Research supported by NSF Grant CCR-0205423.

[2] Dept. of Computer Science and Applied Math., Weizmann Institute of Science. Rehovot 76100, Israel. Email: harnik@wisdom.weizmann.ac.il. Research supported in part by a grant from the Israel Science Foundation.

[3] Laboratory for Computer Science, Massachusetts Institute of Technology. 200 Technology Square, Cambridge, MA 02139.[‡] Email: alon@lcs.mit.edu

[4] Dept. of Computer Science, University of Haifa, Israel.[§] Email: ronen@cs.haifa.ac.il. Research supported by the Koshland Scholarship.

Abstract. We present the first constant-round protocol for Oblivious Transfer in Maurer's *bounded storage model*. In this model, a long random string $\mathcal{R}$ is initially transmitted and each of the parties stores only a small portion of $\mathcal{R}$. Even though the portions stored by the honest parties are small, security is guaranteed against any malicious party that remembers almost the entire string $\mathcal{R}$ (but not all of it). Previous constructions for oblivious transfer in the bounded storage model required polynomially many rounds of interaction. In contrast, our protocol uses only 5 messages. In addition we also improve other parameters, such as the number of bits transferred and the probability of immaturely aborting the protocol due to failure.

Our techniques utilize explicit constructions from the theory of derandomization. In particular, we achieve the constant round complexity of our oblivious transfer protocol by constructing a novel 4-message protocol for Interactive Hashing, in place of the well-known protocol by Naor et al. (known as the NOVY protocol) which involves many rounds of interaction. Our 4-message interactive hashing protocol is constructed by use of *t-wise independent permutations*, and may be of independent interests. For achieving constant round complexity we also construct a new subset encoding scheme that is dense, namely guarantees that almost every strings in the image of the encoding function has a preimage. Other tools we employ include *randomness extractors and averaging samplers.*

Keywords: Bounded Storage Model, Oblivious Transfer, Interactive Hashing, Constant Round Complexity, Information-Theoretic Security, Almost t-wise Independent Permutations, Randomness Extractors, Averaging Samplers, Dense Subset Encoding.

[‡] Part of this work done while at the Weizmann Institute of Science, Israel.

[§] Part of this work was done while at the Weizmann Institute of Science, Israel.

1 Introduction

Oblivious transfer (OT) is one of the fundamental building blocks of modern cryptography. First introduced by Rabin [50], oblivious transfer can serve as a basis to a wide range of cryptographic tasks. Most notably, any multi-party secure computation can be based on the security of OT. This was shown for various models in several works (cf. [55, 33, 41, 13, 35, 16]). Oblivious transfer has been studied in several variants, all of which were eventually shown to be equivalent. In this paper we consider the one-out-of-two variant of OT by Even, Goldreich ad Lempel [27], which was shown to be equivalent to Rabin's variant by Crépeau [12]. One-out-of-two OT is a protocol between two players, Alice holding two secrets s_0 and s_1, and Bob holding a choice bit c. At the end of the protocol Bob should learn the secret of his choice (i.e., s_c), but learn nothing about the other secret. More precisely, the latter means that the receiver Bob (malicious or not) learns information about *at most one* secret, that is, if he obtains any (partial) information about one of the two secret, then he learns nothing about the other secret. Alice, on the other hand, should learn nothing about Bob's choice c.

Traditionally, constructions for OT have been based on strong computational assumptions, either specific assumptions such as the hardness of factoring or Diffie-Hellman problems (cf. [50, 3, 28, 45]), or generic assumptions such as the existence of enhanced trapdoor permutations (cf. [27, 33, 31, 29]). In contrast, OT cannot be reduced in a black box manner to presumably weaker primitives such as one-way functions [39]. This state of affairs motivates the construction of OT in other types of setups. Indeed, protocols for OT have been suggested in different models such as under the existence of noisy channels [14] or quantum channels [14, 5, 18].[7] In this work we follow a direction initiated by Cachin, Crépeau and Marcil [9] and construct OT in the *Bounded Storage model*.

1.1 The bounded storage model

In contrast to the usual approach in modern cryptography, Maurer's *bounded storage model* [43] bounds the *space* (memory size) of dishonest players rather than their running time. In a typical protocol in the bounded storage model a long random string $\mathcal{R}$ of length N is initially broadcast and the interaction between the polynomial-time participants is conducted based on a short portion of $\mathcal{R}$.[8] What makes such protocols interesting is that, even though the honest players store only a small fraction $k \ll N$ of the string $\mathcal{R}$, security is guaranteed

[7] We note that it is impossible to construct unconditionally secure oblivious transfer protocols in the original model for quantum cryptography (See references in [18]). Recently in [18], combining ideas from both the bounded storage model and the quantum model for cryptography, Damgård et al. introduced the so-called bounded quantum storage model, and constructed a very efficient non-interactive protocol for Rabin's original OT in this new model.

[8] One possible implementation is that $\mathcal{R}$ is broadcast at a very high rate by a trusted party. Another possibility is to have $\mathcal{R}$ transmitted from a satellite. We remark that

even against dishonest players with space K where $k \ll K < N$. Moreover, dishonest players are not restricted to be computationally bounded (This is formalized by allowing dishonest players to choose an arbitrary *memory function* $g^* : \{0,1\}^N \to \{0,1\}^K$, and store $g^*(\mathcal{R})$. From that moment on, they are not bounded in any way). Naturally, we'd like to maximize K and minimize k. In this paper we have $K = \nu N$ for an arbitrary constant $\nu < 1$ and k will be about $K^{1/2}$.

The bounded storage model has two appealing properties: (1) The security obtained is information theoretic and thus everlasting in the sense that security is guaranteed even if adversaries acquire infinite space after the protocol is executed. (2) Protocols in the bounded storage model need not rely on any assumption except the limitation on the storage capabilities of the adversary. The latter property should be contrasted with traditional works in cryptography in which, besides bounding the adversary's computational capabilities, it is also required to rely on unproven hardness assumptions (such as the existence of enhanced trapdoor permutations, or the hardness of factoring large integers). We mention that most of the previous work on the bounded storage model concentrated on private key encryption [43, 10, 2, 1, 23, 26, 42, 54] and key agreement [10].

1.2 Oblivious transfer in the bounded storage model

The first protocol for OT in the bounded storage model was given in [9]. This protocol requires $k \approx K^{2/3}$ and allows $K = \nu N$ for an arbitrary constant $\nu < 1$. The *error* ϵ in this protocol is rather large $\epsilon = k^{-O(1)}$. (Loosely speaking the error ϵ measures the probability that a dishonest receiver with storage bound K learns information of both secrets.) A modified protocol with a smaller error ϵ and smaller space k was given in [22]. For every constant $c > 0$, it achieves $k = K^{1/2+c}$ and $\epsilon = 2^{-k^{c'}}$ where $c' > 0$ is a constant that depends on c. We mention that the security of [22] is proven in a slightly different (and weaker) model, where it is assumed that two random strings $\mathcal{R}_1, \mathcal{R}_2$ of length $6K$ are transmitted one after the other and a malicious storage bounded receiver chooses what to remember about $\mathcal{R}_2$ as a function of what he remembers about $\mathcal{R}_1$. The work of [22] was subsequently extended to deal with one-out-of-k OT for any small constant $k \geq 2$ in [37].[9] All protocols mentioned above require a lot of interaction. Specifically, for $\epsilon = 2^{-k^{O(1)}}$, they require the exchange of $k^{\Omega(1)}$ messages between the two players.

1.3 Our results

We give a *constant round* OT protocol in the bounded storage model. Our protocol uses 5 messages following the transmission of the random string $\mathcal{R}$. We

in our protocol (as in many previous ones) one of the parties can transmit these bits. Furthermore, the assumption that $\mathcal{R}$ is uniformly distributed can be relaxed and it is sufficient that $\mathcal{R}$ has high min-entropy.

[9] We note that a similar extension can be easily applied to our work.

achieve parameters k and ϵ similar to that of [22] (that is, for every $c > 0$ there exists $c' > 0$ such that our protocol has $k = K^{1/2+c}$ and $\epsilon = 2^{-k^{c'}}$) while working in the stronger model of [9]. Similar to [9] we can achieve $K = \nu N$ for an arbitrary constant $\nu < 1$. In addition to being constant-round, our protocol also achieves the following improvements over [9, 22]:

- The previous protocols are designed to transfer secrets in $\{0, 1\}$. Thus, transferring long secrets requires many messages. Our protocol can handle secrets of length up to $k^{\Omega(1)}$ in one execution.[10]
- The previous protocols abort unsuccessfully with probability 1/2 even if both players are honest. Our protocol aborts only with probability $2^{-k^{\Omega(1)}}$.
- For error $\epsilon = 2^{-k^{\Omega(1)}}$, the number of bits communicated in the two previous protocols is at least $K^{1/2}$. In contrast, for error $\epsilon = 2^{-k^c}$ our protocol communicates only $O(k^c)$ bits.

We also give a precise definition for the security of oblivious transfer in the bounded storage model, and point out difficulties arising when trying to consider the more standard notion of a simulation-based definition.

1.4 Interactive Hashing

As an essential building block of our OT protocol, we construct a novel constant-round 2-to-1 *interactive hashing* protocol for unbounded parties. Loosely speaking, in such a protocol Bob holds an input $W \in \{0, 1\}^m$, and Alice and Bob want to agree on a pair W_0, W_1 such that $W_d = W$ for some $d \in \{0, 1\}$, yet Alice does not know d. It is also required that a dishonest Bob cannot "control" both W_0 and W_1. In the context of this paper, for the latter it suffices that for any fixed "bad" set S that is sufficently small, a dishonest Bob cannot force both W_0 and W_1 to be in S (See Section 5 for a precise definition). As proved in [9], the protocol of Naor, Ostrovsky, Venkatesan and Yung [44] (known as the NOVY protocol, originally used in the context of perfectly-hiding commitments) achieves 2-to-1 interactive hashing. One major drawback of the NOVY protocol, however, is that it requires $m - 1$ rounds of interaction. In this paper, relying on an explicit construction of almost t-wise independent permutations, such as the constructions presented in [40] and references therein, we construct a new 4-message protocol for 2-to-1 interactive hashing that can replace the NOVY protocol in the context of oblivious transfer in the bounded-storage model.

It is worth noting that since the security guarantees of this interactive hashing protocol hold against unbounded parties, the protocol is not restricted to the bounded storage model alone and may be of independent interest. On the other hand, our new 4-message interactive hashing protocol cannot replace the NOVY protocol in the original context in [44], that is, for constructing perfectly hiding

[10] It is also clear that OT of strings immediately implies OT of bits.

bit commitment schemes from arbitrary one-way permutations.[11] The NOVY protocol achieves a stronger simulation-based security for interactive hashing than the one defined here (see further details in Section 8). Another slight drawback of our interactive hashing protocol compared to the NOVY protocol is that it requires the knowledge of the size of the "bad" set S (this is true only when the set S is relatively large and contains a polynomial fraction of the all inputs). In the context of OT in the bounded storage model this requirement translates to knowledge of the storage bound of the malicious receiver, which is a standard requirement in the bounded storage model.

Organization. In Section 2 we present an overview of the techniques that were utilized to achieve our results. Some preliminaries are given in Section 3. Section 4 provides a precise definition of OT in the bounded storage model. In Section 5 we define the notion of "interactive hashing" and give a constant round protocol for interactive hashing. The OT protocol is presented in Section 6. Sections 6.1 and 7 are devoted to proving the correctness of the protocol. Finally, we conclude and mention some open problems in Section 8.

2 Overview of the technique

As motivation for our protocol, we begin by suggesting a simple protocol for OT in the bounded storage model which is bad in the sense that it requires large storage from the honest parties: Alice is required to store *all* of the string $\mathcal{R}$ and Bob is required to store half this string. We partition the N bit long string $\mathcal{R}$ into two equally long parts $\mathcal{R}_0, \mathcal{R}_1$ of length $N/2$. Recall that Alice has two secrets s_0, s_1 and Bob has a "choice bit" c and wants to obtain s_c. Bob will choose which of the two parts $\mathcal{R}_0, \mathcal{R}_1$ to store depending on his "choice bit" c.

Intuitively, even if Bob is dishonest and has storage bound νN then there is an $I \in \{0,1\}$ such that Bob "does not remember" $(1-\nu)N/2$ bits of information about $\mathcal{R}_I$. This can be formalized by saying that the conditional entropy of $\mathcal{R}_I$ given the memory content of Bob is roughly $(1-\nu)N/2$. (Actually, in this paper, as in [9, 22], we need to work with a variant of entropy called *min-entropy*). Let $\mathrm{Ext}(X,Y)$ (Ext for *extractor*) denote a function such that whenever X has sufficiently high min-entropy and Y is uniformly distributed then $\mathrm{Ext}(X,Y)$ is close to being uniformly distributed. (The reader is referred to [48, 53] for surveys on extractors). To complete the protocol, Alice sends $Z_i = s_i \oplus \mathrm{Ext}(\mathcal{R}_i, Y_i)$ for both $i = 0$ and $i = 1$. Note that an honest Bob can compute $\mathrm{Ext}(\mathcal{R}_c, Y_c) \oplus Z_c$ and obtain s_c. However, if Bob is dishonest then Z_I is close to uniform from Bob's point of view and reveals no information about s_I.[12] Note that even an

[11] We remark that constant-round perfectly hiding bit commitment schemes are known [47, 21, 32], but require seemingly stronger assumptions than the one-way permutations used in [44].

[12] We mention that the argument above is imprecise. Given the memory content of Bob, the strings Z_0, Z_1 are no longer independent. Thus, to prove security it is not

Input of Alice: Secrets s_0, s_1.
Input of Bob: Choice bit: $c \in \{0,1\}$.
A random string $\mathcal{R} = (\mathcal{R}_0, \mathcal{R}_1)$ is transmitted.
Alice: Store all of $\mathcal{R}$.
Bob: Store $\mathcal{R}_c$.
Alice: For $i \in \{0,1\}$, send a uniformly chosen seed Y_i, compute $V_i = \mathrm{Ext}(\mathcal{R}_i, Y_i)$ and $Z_i = V_i \oplus s_i$. Send Y_i, Z_i.
Bob: Compute $V_c = \mathrm{Ext}(\mathcal{R}_c, Y_c)$ and obtain $s_c = V_c \oplus Z_c$.

Fig. 1. A naïve protocol for OT

unbounded dishonest Alice cannot learn anything about Bob's choice bit c, as Bob does not send any messages during the protocol.

Using a setup stage before the naïve protocol. The naïve protocol above requires very large storage bounds from the honest parties. In order to instantiate it in a more efficient manner we will first apply a carefully designed *setup stage*. Our goal is that at the end of the setup stage the two players will agree on two small subsets $C_0, C_1 \subseteq [N]$ of size $\ell \ll N$, such that Alice stores $\mathcal{R}_0 = \mathcal{R}_{C_0}$ and $\mathcal{R}_1 = \mathcal{R}_{C_1}$. (We use $\mathcal{R}_C$ to denote the $|C|$ bit long string obtained by restricting $\mathcal{R}$ to the indices in C.) Bob remembers only one of $\mathcal{R}_0, \mathcal{R}_1$ and cannot remember too much information about the other string. Furthermore, Alice does not know which of the two strings is not known to Bob. Following the setup stage, the two parties can perform OT by using the naïve protocol. We call this second stage the *transfer stage*. As the sets C_0, C_1 are of size $\ell \ll N$ the storage required by the honest parties at the transfer stage is much smaller than before, and honest players can follow the naïve protocol with space $O(\ell) \ll N$.

Implementing the setup stage. An implementation for such a setup stage was suggested in [9, 22]: Alice and Bob each choose a random subset of $[N]$ of size $n = \sqrt{2N\ell}$. We denote them by A and B respectively. When the string $\mathcal{R}$ is transmitted Alice and Bob store $\mathcal{R}_A$ and $\mathcal{R}_B$ respectively. Alice then sends A to Bob. By the birthday paradox, with high probability $C = A \cap B$ is of size at least ℓ. Note that Bob remembers $\mathcal{R}_C$, and Alice does not know C. To complete the setup stage, Alice and Bob play an interactive hashing protocol where Bob uses $W = C$ as input. We give a precise definition of "interactive hashing" later on. Following the interactive hashing protocol, Alice and Bob obtain sets $C_0, C_1 \subseteq A$ such that $C = C_d$ for some $d \in \{0,1\}$. The security of the interactive

sufficient to prove that Z_I is uniformly distributed given the memory content of Bob. In the technical proof we prove that Z_I is uniformly distributed given the memory content of Bob, Z_{1-I} and Y_0, Y_1.

A long random string $\mathcal{R}$ of length N is transmitted.
Alice: Choose random $A \subseteq [N]$ of size n and store $\mathcal{R}_A$.
Bob: Choose random $B \subseteq [N]$ of size n and store $\mathcal{R}_B$.
Alice: Send A to Bob.
Bob: Verify that $C = A \cap B$ is of size at least $\ell = n^2/2N$. If C is large then randomly truncate it to size exactly ℓ.
Alice and Bob: Play an interactive hashing protocol where Bob's input is C. Both Alice and Bob obtain $C_0, C_1 \subseteq A$ such that $C \in \{C_0, C_1\}$.
At this point, Alice and Bob use the naïve protocol with $\mathcal{R}_0 = \mathcal{R}_{C_0}$ and $\mathcal{R}_1 = \mathcal{R}_{C_1}$.

Fig. 2. The protocol for the setup stage

hashing protocol guarantees the following properties: (1) Alice does not know d. (2) Bob "does not remember a lot of information" about one of the strings $\mathcal{R}_{C_0}, \mathcal{R}_{C_1}$. Thus, the two sets C_0, C_1 satisfy the properties required above and the parties can complete the OT protocol by using the naïve protocol.[13] Note that the setup stage requires the honest parties to store only $k = n = O(\sqrt{N\ell})$ bits. In this presentation, we did not discuss the security of Bob, however it is easy to see that even an unbounded Alice, who remembers all of $\mathcal{R}$, cannot learn any information about c.

Previous protocols. The protocols of [9, 22] both use the setup stage described above. They implement interactive hashing using the NOVY protocol from [44] which takes $\ell = k^{\Omega(1)}$-rounds. Following the setup stage they perform what can be seen in retrospect as variants of our naïve protocol. (Both papers do not use extractors explicitly, however their strategies can be viewed as some (weak) implementations of extractors.)

Our improvements. Our main improvement comes from replacing the NOVY protocol for interactive hashing by a new 4-message protocol. This protocol is based on explicit constructions of almost t-wise independent permutations. Some of the additional improvements are given by using competitive explicit constructions of extractors for the naïve protocol above. Another source of improvement comes from allowing Alice to choose the set A using an *averaging sampler* (The reader is referred to [30] for a survey on samplers). Choosing the set A using a competitive averaging sampler reduces the memory requirements of Alice and Bob, as well as the overall communication. We note that using a sampler to choose the set B as well, we can further improve the total communication

[13] A subtlety is that Bob has no control whether $C = C_0$ or $C = C_1$. In the actual protocol we allow Bob to ask Alice to "switch" between the roles of C_0, C_1 in order to receive the desired secret.

and memory requirements. We remark that the usefulness of extractors in the bounded storage model was demonstrated in [42] and [54], and that of averaging samplers was demonstrated in [54].[14] Our paper is another example of the usefulness of ideas from the theory of derandomization in the bounded storage model.

2.1 The improved interactive hashing protocol

In an interactive hashing protocol Bob holds an input $W \in \{0,1\}^m$ and at the end of the protocol both parties should agree on W_0, W_1. It is required that there is a $d \in \{0,1\}$ such that $W = W_d$ and that a dishonest Alice cannot learn d. The main requirement is that a dishonest Bob cannot "control" both W_0 and W_1. This is captured by the following condition: For every strategy of Bob and every set S of size 2^s (where s is a parameter), if Alice is honest then with high probability Bob cannot force that both W_0 and W_1 are in S.

A naïve solution. A naïve solution to this problem is that Alice sends a random 2-to-1 "hash function" $h : \{0,1\}^m \to \{0,1\}^{m-1}$ and Bob replies with $z = h(W)$. Then the two parties compute the two preimages W_0, W_1 of z under h. Note that for $s > m/2$ this protocol fails even if Alice sends a completely random function $h : \{0,1\}^m \to \{0,1\}^{m-1}$ (By the birthday paradox, for every S of size $2^s > 2^{m/2}$ with high probability over h there are $W_0, W_1 \in S$ such that $h(W_1) = h(W_2)$).

The NOVY protocol. The NOVY protocol [44] for interactive hashing can be thought of as a variant of the naïve solution described above in which Alice does *not* send "all" of the hash function at once. Alice chooses a random $(m-1) \times m$ matrix A with entries in $\{0,1\}$ subject to the restriction that all the $m-1$ rows of A are linearly independent. Every such A defines a function $h_A(x) = A \cdot x$, which is clearly 2-to-1. The protocol consists of $m-1$ rounds. In round i, Alice sends A_i (the i'th row of A), and Bob replies with the $z_i = \langle A_i, W \rangle = h_A(W)_i$. Intuitively, revealing h_A slowly in return to bits z_i restricts Bob in the sense that he has to "choose at least part of his input" before seeing all of h_A.

The new protocol. In contrast to the naïve protocol which sends all of the 2-to-1 hash function at once, the NOVY protocol sends $m-1$ hash functions which together form a 2-to-1 function. Our improved protocol will use two hash functions that together form a 2-to-1 hash function. The first hash function is chosen from a family of permutations over $\{0,1\}^m$ with stronger independence properties. Namely, Alice chooses π at random from a family of m-wise independent permutations. She then sends π to Bob and in exchange Bob sends at once $z_1, \cdots, z_v$ where $z_i = \pi(W)_i$ and v is close to m. We can show that

[14] It should be noted that the seminal paper of Nisan and Zuckerman [49] which defined extractors, already used them in a very related context to construct pseudorandom generators against bounded space machines.

the strong independence properties of π "protect Alice" and allow the parties to engage in a new interactive hashing protocol for sending the remaining $m - v - 1$ bits. It turns out that by choosing the parameters appropriately, the two parties can use the naïve solution (with a pairwise independent hash function $g : \{0,1\}^{m-v} \rightarrow \{0,1\}^{m-v-1}$) after the first round. As a result of that we obtain a 2-round (4-messages) protocol. The precise protocol is described in Section 5.4.

Unfortunately, we are not aware of any explicit construction of a small sample space of exact t-wise independent permutations for $t > 3$. Nevertheless, it has been shown how to construct a sample space of permutations in which every t elements are close to being independent (see [40] and references therein), and we can carry out the argument with this weaker property.

3 Preliminaries

We use $[N]$ to denote the set $\{1, \dots, N\}$. We use $X \stackrel{\mathrm{R}}{\leftarrow} S$ to denote uniformly choosing X from S. For a set $A \subseteq [N]$ and a string $\mathcal{R} \in \{0,1\}^N$ we let $\mathcal{R}_A$ denote the substring of $\mathcal{R}$ consisting of the bits indexed by A. For a set S and $\ell \leq |S|$, we use $\binom{S}{\ell}$ to denote the set of all subsets $T \subseteq S$ with $|T| = \ell$.

Definition 3.1 (2^k-to-1 functions) *A function $h : \{0,1\}^m \rightarrow \{0,1\}^{m-k}$ is 2^k-to-1 if for every output of h there are exactly 2^k pre-images. That is, $|h^{-1}(z)\}| = 2^k$ for every $z \in \{0,1\}^{m-k}$.*

3.1 Encoding subsets.

We use a method of encoding sets in $\binom{[n]}{\ell}$ into binary strings. The following method was used in [9]:

Theorem 3.2 ([11]) *For every integers $\ell \leq n$ there is a one to one mapping $F : \binom{[n]}{\ell} \rightarrow [\binom{n}{\ell}]$ such that both F and F^{-1} can be computed in time polynomial in n and space $O(\log \binom{n}{\ell})$.*

Using Theorem 3.2 we can encode $\binom{[n]}{\ell}$ by binary strings of length $\lceil \log \binom{n}{\ell} \rceil$. However, it could be the case that images of subsets constitute only slightly more than half of the strings above. This is exactly what causes the protocols of [9, 22] to unsuccessfully abort with probability 1/2 (and is solved by repeating the protocol until the execution succeeds). Since in this work we are aiming for low round complexity, it would be beneficial to have the probability of unsuccessful abort to be significantly smaller than 1/2. To achieve this, we will use a more redundant encoding. This encoding is more "dense" than the original one and thus guarantees that most strings can be decoded.

Definition 3.3 (Dense encoding of subsets) *For every integers $\ell \leq n$ let F be the mapping from Theorem 3.2. Given an integer $m \geq \lceil \log \binom{n}{\ell} \rceil$ we set $t_m = \lfloor 2^m / \binom{n}{\ell} \rfloor$. Define the mapping $F_m : \binom{[n]}{\ell} \times [t_m] \rightarrow [2^m]$ as $F_m(S,i) = (i-1)\binom{n}{\ell} + F(S)$ (every subset S is mapped to t_m different m bit strings).*

We now have the following Lemma.

Lemma 3.4 *For every $\ell \leq n$ and $m \geq \lceil \log \binom{n}{\ell} \rceil$, the encoding F_m is a one-to-one mapping that is computable in time poly$(n, \log m)$ and space $O(\log \binom{n}{\ell}) + \log m$. Furthermore: (1) Let D denote the image of F_m. The function F_m^{-1} (defined over D) is also computable in time poly$(n, \log m)$ and space $O(\log \binom{n}{\ell}) + \log m$. (2) $\frac{|D|}{2^m} > 1 - \binom{n}{\ell}/2^m$.*

We call the image D of F_m the set of valid encodings.

Proof: Note that $D = [t_m \cdot \binom{n}{\ell}]$. The encoding F_m is one-to-one since for every $W \in D$, $F_m^{-1}(W) = F^{-1}(((W-1) \bmod \binom{n}{\ell})+1)$. It is also easy to verify for each $W \in [2^m]$ whether $W \in D$, that is, whether W is a valid encoding, by checking whether $W \leq t_m \cdot \binom{n}{\ell}$. Property (1) follows from the formula for F_m^{-1} and the guarantee on F_m (in definition 3.3). Also, the number of distinct encodings in D equals $t_m \cdot \binom{n}{\ell} = \lfloor 2^m / \binom{n}{\ell} \rfloor \cdot \binom{n}{\ell} \geq (2^m/\binom{n}{\ell} - 1) \cdot \binom{n}{\ell} = 2^m - \binom{n}{\ell}$ and property (2) follows. ■

3.2 Min-entropy and Extractors.

Min-entropy is a variant of Shannon's entropy that measures the randomness of a probability distribution or a random variable in the *worst case*. A distribution has high min-entropy if the probability mass it assigns to every element of the probability space is small.

Definition 3.5 (Min-entropy) *For a distribution X over a probability space Ω the* min-entropy *of X is defined as $H_\infty(X) = \min_{x \in \Omega} \log(1/\Pr[X = x])$. We say that X is a k-source if $H_\infty(X) \geq k$, that is, for every $x \in \Omega$, $\Pr[X = x] \leq 2^{-k}$.*

Definition 3.6 (Statistical distance) *Two distributions P and Q over Ω are ϵ-close (also denoted $P \stackrel{\epsilon}{\equiv} Q$) if for every $A \subseteq \Omega$, $|\Pr_{x \leftarrow P}(A) - \Pr_{x \leftarrow Q}(A)| \leq \epsilon$.*

Thus, that two distributions P and Q are ϵ-close means that the maximum advantage of any unbounded *distinguisher* in distinguishing between P and Q, is at most ϵ.

An extractor is a function that "extracts" randomness from arbitrary distributions which "contain" sufficient (min)-entropy [49].

Definition 3.7 (Strong extractor) *A function* $\mathrm{Ext} : \{0,1\}^{n_E} \times \{0,1\}^{d_E} \to \{0,1\}^{m_E}$ *is a (k_E, ϵ_E)-strong extractor if for every k_E-source X over $\{0,1\}^{n_E}$ the distribution $(\mathrm{Ext}(X,Y), Y)$, where Y is uniform over $\{0,1\}^{d_E}$, is ϵ_E-close to (U_{m_E}, Y), where U_{m_E} is uniform over $\{0,1\}^{m_E}$ and independent of Y.*

We remark that a *regular* (non-strong) extractor is defined in a similar way, with the random variables $(\mathrm{Ext}(X,Y), Y)$ and (U_{m_E}, Y) above replaced by $\mathrm{Ext}(X,Y)$ and U_{m_E} respectively.

3.3 Averaging Samplers and Min-Entropy Samplers.

A fundamental lemma by Nisan and Zuckerman [49] asserts that given a δv-source X on $\{0,1\}^v$, with high probability over choosing a subset $T \subseteq [v]$ of size t, X_T is roughly a δt-source. Or in other words, that sampling a random piece from a source preserves the "min-entropy rate" of the source. As shown in [49, 52, 54] the lemma does not require a uniformly chosen subset. It is sufficient that T is chosen using a "good averaging sampler".

Definition 3.8 (Averaging sampler) *A function* $\mathrm{Samp} : [L] \to [v]^t$ *is a* (μ, θ, γ)-averaging sampler *if for every function* $f : [v] \to [0,1]$ *with average value* $\frac{1}{v}\sum_i f(i) \geq \mu$,

$$\Pr_{p \in [L]}\left[\frac{1}{t}\sum_{1 \leq i \leq t} f(\mathrm{Samp}(p)_i) < \mu - \theta\right] \leq \gamma$$

The function Samp *is said to have* distinct samples *if for every* $p \in [L]$*, the* t *outputs of* $\mathrm{Samp}(p)$ *are distinct.*

Averaging samplers have been a subject of a line of studies starting with [4]. For a survey of averaging samplers, see [30]. The following Lemma asserts that using an averaging sampler to sample a subset from a source with min-entropy rate δ roughly preserves the min-entropy rate of the source.

Lemma 3.9 *[54] Suppose that* $\mathrm{Samp} : [L] \to [v]^t$ *is a* (μ, θ, γ)*-averaging sampler with distinct samples for* $\mu = (\delta - 2\tau)/\log(1/\tau)$ *and* $\theta = \tau/\log(1/\tau)$*. Then for every* δv*-source* X *on* $\{0,1\}^v$*, the random variable* $(P, X_{\mathrm{Samp}(P)})$ *where* P *is uniform over* $[L]$ *is* $\gamma + 2^{-\Omega(\tau v)}$*-close to a random variable* (P, Q) *such that for every* $p \in [L]$*, the random variable* $Q|_{P=p}$ *is a* $(\delta - 3\tau)t$*-source.*

It will be convenient for us to state Lemma 3.9 in different form. For this purpose we introduce the notion of a *min-entropy sampler.*

Definition 3.10 (Min-entropy sampler) *A function* $\mathrm{Samp} : [L] \to [v]^t$ *with distinct samples is a* $(\delta, \delta', \phi, \epsilon)$-min-entropy sampler *if for every* δv*-source* X *over* $\{0,1\}^v$ *there is a set* $G \subseteq [L]$ *of density* $1 - \phi$ *such that for every* $p \in G$ *the distribution* $X_{\mathrm{Samp}(p)}$ *is* ϵ*-close to a* $\delta' t$*-source.*

With this notation we can restate Lemma 3.9 as follows:

Corollary 3.11 *Suppose that* $\mathrm{Samp} : [L] \to [v]^t$ *is a* (μ, θ, γ)*-averaging sampler with distinct samples for* $\mu = (\delta - 2\tau)/\log(1/\tau)$ *and* $\theta = \tau/\log(1/\tau)$*. Then there is a constant* $c > 0$ *such that for every* $0 < \alpha < 1$*,* Samp *is a* $(\delta, \delta - 3\tau, (\gamma + 2^{-c\tau v})^{1-\alpha}, (\gamma + 2^{-c\tau v})^{\alpha})$*-min-entropy sampler.*

Proof: Let c be the constant hidden in the statement of Lemma 3.9. That is, $(U_r, X_{\mathrm{Samp}(U_r)})$ is $\gamma + 2^{-c\tau v}$-close to a random variable (P, Q) where P is uniform over $\{0,1\}^r$ and for every $p \in \{0,1\}^r$, the random variable $Q|_{P=p}$ is a

$(\delta - 3\tau)t$-source. Let B be the set of $p \in \{0,1\}^r$ such that the $X_{\mathrm{Samp}(p)}$ is not $(\gamma + 2^{-c\tau v})^\alpha$-close to a $(\delta - 3\tau)t$-source. It follows that the density ϕ of B is at most $(\gamma + 2^{-c\tau v})^{1-\alpha}$. This is because otherwise the statistical distance between $(U_r, X_{\mathrm{Samp}(U_r)})$ and any (P, Q) that satisfy the condition in Lemma 3.9 is at least $\phi(\gamma + 2^{-c\tau v})^\alpha > \gamma + 2^{-c\tau v}$. Let G be the complement of B. The Lemma follows. ∎

3.4 The random subset sampler

An averaging sampler is a way of choosing a set with certain "random properties". Naturally, choosing a set uniformly at random gives an averaging sampler. We now state the parameters of this sampler.

Definition 3.12 (The random subset sampler) *Given integers $v \geq t$, fix a mapping E from $[\binom{v}{t}]$ to $\binom{[v]}{t}$, let $RS : [\binom{v}{t}] \to \binom{[v]}{t}$ be $RS(p) = E(p)$.*

The following lemma about random subset samplers is standard. We give a proof for completeness.

Lemma 3.13 *There is a constant c such that for every $v \geq t$ and every $\mu, \lambda > 0$, RS is a $(\mu, \lambda\mu, 2e^{-(c\lambda^2 t\mu)})$-averaging sampler with distinct samples.*

Proof: Consider the following probability space for choosing a subset $S \subseteq [v]$. For every $i \in [v]$ we independently choose whether $i \in S$ with probability t'/v. For every $i \in [v]$ we define X_i to be $f(i)$ if $i \in S$ and zero otherwise. Note that $E(X_i) = f(i)t'/v$. Let $X = \sum X_i$, by linearity of expectation $E(X) = t'\mu$. By Chernoff's inequality, there is a constant $c > 0$ such that $\Pr[|X - t'\mu| > \delta t'\mu] < e^{-c\delta^2 t'\mu}$. Furthermore, another application of the inequality gives that $\Pr[||S| - t'| < \delta t'] < e^{-c\delta^2 t'}$. We now randomly add indices to S to give a subset of size $t = (1+\delta)t'$. Note that this sample space is $e^{-c\delta^2 t'}$-close to the sample space of choosing a random subset T of size t. Consider $X' = \frac{t'}{v}\sum_{i \in S} f(i)$ for the enlarged S. It follows that $X' \geq X$. We have that $\Pr[X' < t'\mu - \delta t'\mu] < e^{-c\delta^2 t'\mu}$. We now want this event to hold whenever $\{X' < t\mu - t\lambda\mu\}$ holds. A calculation shows that $\lambda \geq 2\delta/1 + \delta$, which is satisfied for $\delta = \lambda/2$. Thus, when choosing a random subset T of size t, $\Pr[\frac{1}{t}\sum_{i \in T} f(i) < \mu - \lambda\mu] < 2e^{\frac{c\lambda^2 t\mu}{4(1+\lambda/2)}} \leq 2e^{-\Omega(\lambda^2 t\mu)}$, where the last inequality follows because $\lambda < 1$. ∎

Corollary 3.14 *For every $v \geq t$, and $\delta > 0$, RS is a $(\delta, \delta/2, \phi, \epsilon)$-min-entropy sampler, for $\phi = \epsilon = 3 \cdot 2^{-\Omega(\delta t / \log(1/\delta))}$.*

Proof: Let $\tau = \delta/6$, $\mu = (\delta - 2\tau)/\log(1/\tau) = 4\delta/6\log(6/\delta)$ and $\theta = \tau/\log(1/\tau)$. We choose $\lambda = \theta/\mu = 1/4$ and apply Lemma 3.13. It follows that there exists a constant d such that RS is a (μ, θ, γ)-averaging sampler for $\gamma = 2e^{-dt\mu}$. By applying Corollary 3.11 with $\alpha = 1/2$, there is a constant $c > 0$ such that RS is a $(\delta, \delta', \phi, \epsilon)$-min-entropy sampler for $\delta' = \delta - 3\tau = \delta/2$ and

$$\begin{aligned}\phi = \epsilon = (\gamma + 2^{-c\tau v})^{1/2} &\leq \gamma^{1/2} + 2^{-c\tau v/2} \\ &\leq 2 \cdot 2^{-\Omega(\delta t/\log(1/\delta))} + 2^{-\Omega(\delta v)} \\ &\leq 3 \cdot 2^{-\Omega(\delta t/\log(1/\delta))},\end{aligned}$$

where the last inequality follows because $t \leq v$. ■

3.5 Some useful technical Lemmas

The following two lemmata measure the min-entropy in a k-source X when conditioned on the event $\{Y = y\}$ where Y is a (dependent) random variable over short strings. It is very useful for us as we often consider the information in the long random string $\mathcal{R}$ from the point of view of a bounded storage machine that stores only few bits of information about $\mathcal{R}$.

The following lemma is standard. We include a proof for completeness, and it may be instructive for understanding the proof of the next related yet much more complicated lemma.

Lemma 3.15 *If X is a k-source and Y is over $\{0,1\}^r$ then with probability $1-\beta$ (over the choice of $y \leftarrow Y$) $(X \mid Y = y)$ is a $(k - r - \log(1/\beta))$-source.*

Proof: Let $B = \{y \mid \Pr[Y = y] < 2^{-r-\log(1/\beta)}\}$. It follows that $\Pr[Y \in B] < 2^r \cdot 2^{-r-\log(1/\beta)} \leq \beta$. For every $y \notin B$ and x,

$$\Pr[X = x \mid Y = y] \leq \frac{\Pr[X = x]}{\Pr[Y = y]} \leq \frac{2^{-k}}{2^{-r-\log(1/\beta)}} = 2^{-(k-r-\log(1/\beta))}.$$

■

The next Lemma is a strengthening of Lemma 3.15 and handles the case in which X is only close to having high min-entropy.

Lemma 3.16 *If X is ξ-close to a k-source and Y is over $\{0,1\}^r$ then with probability $1 - \beta - \sqrt{2\xi}$ (over the choice of $y \leftarrow Y$) $(X \mid Y = y)$ is $\sqrt{2\xi}$-close to a $(k - r - \log(1/\beta) - 1)$-source.*

Proof: Let $T = \{x \mid \Pr[X = x] > 2^{-(k-1)}\}$. We first show that

$$\Pr[X \in T] \leq 2\xi \tag{1}$$

Let X' be a k-source that is ξ-close to X, and let $P_X = \Pr[X \in T]$ and $P_{X'} = \Pr[X' \in T]$. We have that $|P_X - P_{X'}| < \xi$. On the other hand we have that $P_X \geq |T|2^{-(k-1)}$ and $P_{X'} \leq |T|2^{-k}$. Therefore,

$$P_X - P_{X'} \geq |T|2^{-(k-1)} - |T|2^{-k} = |T|2^{-k} \geq P_{X'}$$

Thus, $P_{X'} \leq \xi$. We now conclude that $P_X \leq P_{X'} + \xi \leq 2\xi$. Thus, equation (1) follows. Let $H = \{(x, y) \mid \Pr[X = x, Y = y] > 2^{-(k-1)}\}$ and $H_y = \{x \mid (x, y) \in H\}$. Note that if a pair $(x, y) \in H$ then $x \in T$. Thus,

$$\Pr[(X, Y) \in H] \leq \Pr[X \in T] \leq 2\xi$$

Let $v_y = \Pr[X \in H_y \mid Y = y]$. Let $A = \{y \mid v_y \geq \sqrt{2\xi}\}$. We now bound the probability that $\{Y \in A\}$. We have that

$$\begin{aligned}
2\xi \geq \Pr[(X, Y) \in H] &= \sum_y \Pr[(X, Y) \in H \mid Y = y] \Pr[Y = y] \\
&= \sum_y \Pr[X \in H_y \mid Y = y] \Pr[Y = y] \\
&= \sum_y v_y \Pr[Y = y] \geq \sum_{y \in A} v_y \Pr[Y = y] \geq \Pr[Y \in A] \sqrt{2\xi}.
\end{aligned}$$

Thus, we have that $\Pr[Y \in A] \leq \sqrt{2\xi}$. Let $B = \{y \mid \Pr[Y = y] < 2^{-r - \log(1/\beta)}\}$. It follows that $\Pr[B] < 2^r \cdot 2^{-r - \log(1/\beta)} \leq \beta$. Let $C = A \cup B$. We have that $\Pr[Y \in C] \leq \beta + \sqrt{2\xi}$. Finally, we show that for every $y \notin C$, the distribution $(X \mid Y = y)$ is $\sqrt{2\xi}$-close to a $k - 1 - r - \log(1/\beta)$-source. To see this we check that for every such y and $x \notin H_y$,

$$\Pr[X = x \mid Y = y] = \frac{\Pr[X = x, Y = y]}{\Pr[Y = y]} \leq \frac{2^{-(k-1)}}{2^{-(r + \log(1/\beta))}} = 2^{-(k - 1 - r - \log(1/\beta))}.$$

The Lemma follows as $\Pr[X \in H_y \mid Y = y] = v_y < \sqrt{2\xi}$ and the weight of heavy elements is small. ■

The following Lemma asserts that running a strong extractor on a k-source gives a distribution which is close to uniform for most choices of $y \in \{0,1\}^{d_E}$. We will use this to argue that if a random variable has high min-entropy from the point of view of some bounded storage machine B^*, then for most seeds y applying an extractor gives a distribution which is (close to) uniform from the point of view of B^* *even if* y *is revealed to* B^*.

Lemma 3.17 *If* $\mathrm{Ext} : \{0,1\}^{n_E} \times \{0,1\}^{d_E} \to \{0,1\}^{m_E}$ *is a* (k_E, ϵ_E)*-strong extractor then for every* k*-source* X *over* $\{0,1\}^{n_E}$*, with probability* $1 - 2\sqrt{\epsilon_E}$ *over the choice of* $y \xleftarrow{\mathrm{R}} \{0,1\}^{d_E}$ *the distribution* $E(X, y)$ *is* $\sqrt{\epsilon_E}$*-close to uniform.*

Proof: Assume by contradiction that for at least a $2\sqrt{\epsilon_E}$-fraction of y's, there were a distinguisher D_y which distinguishes $E(X, y)$ from U_{m_E} with advantage greater than $\sqrt{\epsilon_E}$. W.l.o.g. assume that for at least a $\sqrt{\epsilon_E}$-fraction of y's, $\Pr[D_y(E(X, y)) = 1] - \Pr[D_y(U_{m_E}) = 1] > \sqrt{\epsilon_E}$. It follows that there is a distinguisher for distinguishing $(E(X, Y), Y)$ from (U_{m_E}, Y) with advantage greater $\sqrt{\epsilon_E} \cdot \sqrt{\epsilon_E} = \epsilon_E$, a contradiction. ■

4 Oblivious Transfer in the Bounded Storage Model

We now turn to formally define oblivious transfer in the bounded storage model. We first note that formal definitions of oblivious transfer in information-theoretic settings have been previously proposed (c.f. [6, 24, 7]). Shortcomings of those previous definitions have been pointed out in a recent paper [15] which initiated a systematic study of general information-theoretically secure two-party protocols.

In this section we give an adequate definition of oblivious transfer in the bounded storage model. Our definition characterizes malicious strategies for Alice and Bob. Note that in the definitions below the malicious strategies are asymmetric. We restrict malicious strategies for Bob to have bounded storage while no bounds are placed on malicious strategies for Alice. Clearly, if a protocol is secure against unbounded strategies for Alice, it is also secure against bounded strategies. Thus, the security defined here is even stronger than that explained in the introduction.

Definition 4.1 (Malicious Strategy for Alice) *A (malicious) strategy $A^*$for Alice is an unbounded interactive machine with inputs $\mathcal{R} \in \{0,1\}^N$ and $s_0, s_1 \in \{0,1\}^u$. That is, A^* receives $\mathcal{R}$ and s_0, s_1, and interacts with B. In each stage, it may compute the next message as any function of its inputs, its randomness and the messages it received thus far. The view of A^* when interacting with B that holds input c (denoted* $\mathrm{view}_{\mathrm{A}^*}^{\langle \mathrm{A}^*,\mathrm{B}\rangle}(\mathrm{s}_0,\mathrm{s}_1;\mathrm{c})$) *consists of its local output.* [15]

The following definition captures a bounded storage strategy with storage bound K. Loosely speaking, the only restriction made on a bounded storage strategy B^* is that it has some (possibly probabilistic) *memory function* $g^* : \{0,1\}^N \rightarrow \{0,1\}^K$ and its actions depend on $\mathcal{R}$ only through $g^*(\mathcal{R})$. This formally captures that B^* remembers only K bits about $\mathcal{R}$.

Definition 4.2 (Bounded storage strategy for Bob) *A bounded storage strategy B^* for Bob with memory bound K is a pair $(g^*, \widehat{B}^*)$ where:*

- *$g^* : \{0,1\} \times \{0,1\}^N \rightarrow \{0,1\}^K$ is an arbitrary (not necessarily efficiently computable) probabilistic function with input c and $\mathcal{R}$.*
- *$\widehat{B}^*$ is an unbounded interactive machine with inputs $c \in \{0,1\}$ and $b^* \in \{0,1\}^K$.*

The behavior described by a strategy B^* with input c is the following: When given the string $\mathcal{R} \in \{0,1\}^N$, B^* computes $b^* = g^*(c, R)$. B^* then interacts with A using the interactive machine $\widehat{B}^*$ receiving inputs c and b^*. The view of B^* with input c when interacting with A with inputs s_0, s_1 (denoted $\mathrm{view}_{\mathrm{B}^*}^{\langle \mathrm{A},\mathrm{B}^*\rangle}(\mathrm{s}_0,\mathrm{s}_1;\mathrm{c})$) *is defined as the view of $\widehat{B}^*$ when interacting with A.*

We now turn to the definition of oblivious transfer in the bounded storage model. The security of Bob asks that for any malicious strategy for Alice, its view is

[15] The view of A may be thought of as also containing the party's randomness, inputs and outputs, as well as the messages received from B. This more intuitive "view" is possible since w.l.o.g. the malicious party may copy this view to his output.

identically distributed whether Bob inputs $c = 0$ or $c = 1$. The definition of Alice's security is a bit more complex because one of her secrets is passed to Bob. For this definition, we partition *every* protocol that implements OT into two stages. The first stage is called the *Setup Stage*, and includes the transmission of the long string $\mathcal{R}$ and all additional messages sent by Alice and Bob until the point where Alice first makes use of her input s_0, s_1. The remaining steps in the protocol are called the *Transfer Stage*. We also need the following definition.

Definition 4.3 *Two pairs $\bar{s}=(s_0, s_1)$ and $\bar{s}'=(s'_0, s'_1)$ are c-consistent if $s_c=s'_c$.*

The security of Alice asks that following the setup stage (which does not depend on the secrets), there is an index Γ (possibly a random variable which depends on $\mathcal{R}$ and the messages sent by the two parties in the setup stage) such that Bob's view is (close to) identically distributed for every two Γ-consistent pairs. In other words, Bob's view is (almost) independent of one of the secrets (defined by $1 - \Gamma$). We next present the actual definition.

Definition 4.4 (Oblivious Transfer) *A protocol $\langle A, B\rangle$ is said to implement* $(1-\epsilon)$-oblivious transfer (OT) *against storage K if it is a protocol in which Alice inputs two (secrets) $s_0, s_1 \in \{0,1\}^u$, Bob inputs a choice bit $c \in \{0,1\}$, and that satisfies:*

Functionality : *If Alice and Bob follow the protocol then for any s_0, s_1 and c,*
1. *The protocol does not abort with probability at least $1 - \epsilon$.*
2. *If the protocol ends then Bob outputs s_c, whereas Alice outputs nothing.*

Security for Bob*: The view of any strategy A^* is independent of c. Namely, for every s_0, s_1:*

$$\left\{\mathrm{view}_{A^*}^{\langle A^*,B\rangle}(s_0, s_1; c) \mid c = 0\right\} \;\equiv\; \left\{\mathrm{view}_{A^*}^{\langle A^*,B\rangle}(s_0, s_1; c) \mid c = 1\right\}$$

(K,ϵ)**-Security for Alice***: For every bounded storage strategy B^* for Bob with memory bound K and input c there is a random variable Γ defined by the end of the setup stage such that for every two pairs $\bar{s}$ and $\bar{s}'$ that are Γ-consistent:*

$$\left\{\mathrm{view}_{B^*}^{\langle A,B^*\rangle}(\bar{s}; c)\right\} \;\stackrel{\epsilon}{\equiv}\; \left\{\mathrm{view}_{B^*}^{\langle A,B^*\rangle}(\bar{s}'; c)\right\}$$

It is instructive to first consider an adversary Bob that is "semi-honest". Such an adversary follows the protocol, yet he stores additional information on R and tries to use this information to learn both secrets. In this case, the definition above can be simplified and the random variable Γ can be replaced by the choice bit c. However, in general Bob may ignore c and decide to play using a choice bit Γ that is chosen as a function of $\mathcal{R}$ and the messages in the setup stage. Thus, letting Γ depend on $\mathcal{R}$ and the messages in the setup stage is unavoidable. The definition above guarantees that no matter how Bob decided to play there is always a secret which he does not learn. We stress that the security achieved in this definition is *information theoretic*. We remark that the definition would be meaningless if Γ was allowed to depend on the secrets s_0, s_1, and indeed, this is the reason we partition protocols into a setup stage and a transfer stage. We require that the random variable Γ is defined at the end of the setup stage and therefore it does not depend on s_0, s_1.

4.1 Using OT in the bounded storage model as a sub-protocol

We remark that it does not immediately follow from the definition above that all the "standard" applications of OT can be performed in the bounded storage model (this is also the case for the previous protocols in this model [9, 22]). Nevertheless, we now explain how this protocol can be used as a sub-protocol to perform other cryptographic tasks. For example, our OT protocol can be used in the construction of Kilian [41], to give a protocol for secure two-party computation in the bounded storage model. The security achieved guarantees that an unbounded party learns nothing about the input of the other party. In order to use our protocol as a sub-protocol inside other protocols we note that our security definition implies security by a *simulation argument* (although the simulator is not necessarily efficient). Loosely speaking, the simulation paradigm requires that any attack of a malicious party can be simulated in an ideal setting where the parties interact only through a trusted party. This insures that the protocol is as secure as an interaction in the ideal setting. The ideal setting for OT is that both players send their inputs $(s_0, s_1; c)$ to a trusted party, and the trusted party sends s_c to Bob. We need to show that any malicious bounded storage adversary in the bounded storage model can be simulated in a way that gives him the same information when interacting through the trusted party. A weakness of the argument we are about to present is that the simulation we give is not necessarily efficient. We stress that typically one requires that the simulators should run with essentially the same efficiency as the attack being simulated, and that this would provide a stronger notion of security. We now give a sketch of the simulator for a malicious receiver strategy B^*. (likewise, a simulator for a malicious sender can be given). The simulator plays the roles of both B^* and A in the protocol up to the transfer stage. At this point the simulator computes the random variable Γ and calls the trusted party asking for secret s_Γ. It continues by simulating A with inputs s_Γ as received from the trusted party and a random $s_{1-\Gamma}$. By our security definition this turns out to be a valid simulation. We remark that the protocol presented in this paper does not suggest how to efficiently compute Γ and this is why the simulation is not necessarily efficient. It now follows that any application of OT in which security of the protocol is proven using a simulation argument (as the aforementioned secure two-party computation of [41]) can be performed in the bounded storage model with information theoretic security. Nevertheless, as our simulator is not efficient it does not follow that one can "mix" OT in the bounded storage model with cryptography based on computational hardness assumptions. It is an interesting open problem to come up with a protocol that can be efficiently simulated. A related problem is pointed out by Dziembowski and Maurer [25].[16]

[16] In [25], Dziembowski and Maurer showed that using a computationally secure key agreement protocol in order to agree on a secret key for a private-key encryp-

5 Interactive Hashing

One of the main tools we use in this paper is the interactive hashing protocol. While useful in the bounded storage model, it is important to note that interactive hashing is not necessarily related to this model. As a matter of fact, the definitions and protocols given here achieve security against all powerful adversaries with no storage bounds at all.

5.1 Definition of Interactive Hashing

Interactive hashing is a protocol between Alice with no input and Bob with an input string. At the end of the protocol Alice and Bob should agree on two strings: One should be Bob's input and intuitively the other should be random. Moreover, Alice should not be able to distinguish which of the two is Bob's input and which is the random string.

Definition 5.1 (Interactive Hashing) *A protocol $\langle A, B\rangle$ is called an* interactive hashing protocol *if it is an efficient protocol between Alice with no input and Bob with input string $W \in \{0,1\}^m$. At the end of the protocol both Alice and Bob output a (succinct representation of a) 2-to-1 function $h : \{0,1\}^m \to \{0,1\}^{m-1}$ and two values $W_0, W_1 \in \{0,1\}^m$ (in lexicographic order) so that $h(W_0) = h(W_1) = h(W)$. Let $d \in \{0,1\}$ be such that $W_d = W$. Furthermore, if the distribution of the string W_{1-d} over the randomness of the two parties is η-close to uniform on all strings not equal to W_d, then the protocol is called* η-uniform interactive hashing *(or simply uniform interactive hashing if $\eta = 0$).*

Definition 5.2 (Security of Interactive Hashing) *An interactive hashing protocol is* **secure for B** *if for every unbounded strategy A^*, and every W, if h, W_0, W_1 are the outputs of the protocol between an honest Bob with input W and A^*, then*

$$\left\{\mathrm{view}_{\mathrm{A}^*}^{\langle \mathrm{A}^*,\mathrm{B}\rangle}(\mathrm{W}) \mid \mathrm{W} = \mathrm{W}_0\right\} \equiv \left\{\mathrm{view}_{\mathrm{A}^*}^{\langle \mathrm{A}^*,\mathrm{B}\rangle}(\mathrm{W}) \mid \mathrm{W} = \mathrm{W}_1\right\},$$

where $\mathrm{view}_{\mathrm{A}^*}^{\langle \mathrm{A}^*,\mathrm{B}\rangle}(\mathrm{W})$ is A^*'s view of the protocol when B's input is W. An interactive hashing protocol is (s,ρ)**-secure for A** *if for every $S \subseteq \{0,1\}^m$ of size at most 2^s and every unbounded strategy B^*, if W_0, W_1 are the outputs of the protocol, then*

$$\Pr[W_0, W_1 \in S] < \rho$$

where the probability is taken over the coin tosses of A and B^*. An interactive hashing protocol is (s,ρ)**-secure** *if it is secure for B and (s,ρ)-secure for A.*

tion scheme in the bounded storage model (BSM), does *not* necessarily yield an information-theoretically secure combined (hybrid) scheme. Dziembowski and Maurer proved this by giving an explicit computationally secure key agreement protocol and a secure BSM private-key encryption scheme for which the resulting hybrid scheme is insecure.

Remark. The definition above does not deal with the case that dishonest players abort before the end of the execution. Intuitively, such a definition is sufficient for our purposes since in our OT protocol, the interactive hashing is used before the players send any message that depends on their secrets, and thus their secrets are not compromised.

5.2 t-wise Independent Permutations and Hash Functions

In our interactive hashing protocol we would like to use a random permutation on m bit strings. However, a description of such a permutation would be exponentially long since there are $(2^m)!$ such permutations. The solution is to use a permutation that falls short of being truly random but still has enough randomness to it. Specifically we want to efficiently sample a permutation π out of a small space of permutations such that when applied on any t points in $\{0,1\}^m$, π behaves as a truly random permutation. Such a space is called a t-wise independent permutation space. Unlike in the case of functions, where there are extremely randomness efficient constructions of t-wise independent functions, we are unaware of such constructions for permutations. Instead we further relax our demands and ask the construction to be *almost* t-wise independent, that is, the distribution induced by the permutation π on any t points is statistically close to the distribution induced on these points by a truly random permutation. Formally:

Definition 5.3 *An η-almost t-wise independent permutation space is a procedure that takes as input a seed of l bits and outputs a description of an efficiently computable permutation in S_{2^m},*[17] *with the property that a uniformly chosen seed induces a distribution $\Pi_{t,\eta}$ on permutations such that for any t strings $x_1, \ldots x_t \in \{0,1\}^m$:*

$$\{\pi(x_1), \ldots \pi(x_t)\}_{\pi \xleftarrow{\text{R}} \Pi_{t,\eta}} \stackrel{\eta}{\equiv} \{\pi(x_1), \ldots \pi(x_t)\}_{\pi \xleftarrow{\text{R}} S_{2^m}}$$

Such a construction was presented by Gowers [36] based on simple 3-bit permutations (another approach is based on a Feistel structure, see [46]). Recently, the efficiency of such constructions was improved in Hoory et al. [38] and Brodsky and Hoory [8] and ultimately by Kaplan et al. [40].

Theorem 5.4 ([40]) *There exists an η-almost t-wise independent permutation space $\Pi_{t,\eta}$ with $t = m$, $\eta = \left(\frac{1}{2^m}\right)^t$ and seed length $l = O(tm + \log(\frac{1}{\eta})) = O(m^2)$. Furthermore, $\Pi_{t,\eta}$ runs in time and space polynomial in the seed length.*

Pairwise Independent Permutations. A widely used tool is a pairwise independent permutation of strings of m bits. This is simply a 2-wise independent permutation as defined above (i.e., a 0-almost 2-wise independent permutation).

[17] S_{2^m} denotes the family of all permutations on m bit strings

The construction that we use identifies $\{0,1\}^m$ with the field $GF(2^m)$. A permutation is sampled by randomly choosing two elements $a, b \in GF(2^m)$ with the restriction that $a \neq 0$. The permutation is then defined by $g_{a,b}(x) = ax+b$ (where all operations are in the field). A pairwise independent permutation is therefore specified by $2m$ random bits. We note that to construct a *pairwise independent 2-to-1 hash function*, one simply takes a pairwise independent permutation and omit the last bit of its output.

5.3 Partial Result: A Two Message Interactive Hashing

We start by showing that when the bad set S is small enough then the following naïve protocol is sufficiently good. In this 2 message protocol called 2M-IH, Alice sends a random 2-to-1 hash function $h : \{0,1\}^m \rightarrow \{0,1\}^{m-1}$ and Bob replies with $z = h(W)$.

Claim 5.5 *For all u, the* 2M-IH *protocol is a $(s, 2^{-(m-2s+1)})$-secure uniform interactive hashing.*

Proof: The 2M-IH is clearly an interactive hashing protocol, and since h is pairwise independent, then it is also uniform (W_{1-d} is uniformly distributed over all strings barring W_d). The 2M-IH is also secure for B since all that Bob sends to Alice is $h(W)$, which is the exact same view whether Bob has input $W = W_1$ or $W = W_0$. On the other hand, since h is a pairwise independent hash function, then the probability over the choice of h for any two strings W_0, W_1 to be mapped to a certain cell $z \in \{0,1\}^{m-1}$ is perfectly random, that is:

$$\Pr_h[h(W_0) = h(W_1) = z] = 2 \cdot \frac{1}{2^m} \cdot \frac{1}{2^m - 1}$$

Denote $X_z = 1$ if both strings mapped to cell z are from the set S and $X_z = 0$ otherwise. Then:

$$\Pr_h[X_z = 1] \leq \binom{2^s}{2} \Pr_h[h(W_0) = h(W_1) = z] \leq \frac{2^s}{2^m} \cdot \frac{2^s - 1}{2^m - 1} \leq \frac{2^{2s}}{2^{2m}}$$

Denote by X the number of cells z such that both values mapped into z are from the set S, then:

$$E(X) = E\big(\sum_z X_z\big) = \sum_z E\left(X_z\right) \leq 2^{m-1} \cdot \frac{2^{2s}}{2^{2m}} \leq 2^{-(m-2s+1)}$$

The protocol is insecure only if Bob finds a cell z with two bad values, that is only if $X \geq 1$. But using Markov's inequality we have that $\Pr[X \geq 1] \leq E(X) \leq 2^{-(m-2s+1)}$. Thus this protocol is $(s, 2^{-(m-2s+1)})$-secure for Alice. $\blacksquare$

5.4 A Four Message Protocol for Interactive Hashing

The two message protocol is useful when the bad set S is very small. However, if S is large (for example, if $|S| > 2^{m/2}$) then this protocol does not suffice. We need to deal with a large set S in our application for OT in the bounded storage model. It was observed in [9] that the protocol of [44] can handle large sets. However, the interactive hashing protocol of [44] requires $m-1$ rounds of communication. We present a 4 message protocol in Figure 3.

4M-IH (4 Message Interactive Hashing)

Common Input: Parameters m and s.

Let $v = s - \log m$.

A family Π of η-almost t-wise independent permutations $\pi : \{0,1\}^m \to \{0,1\}^m$

Set $t = m$ and $\eta = \left(\frac{1}{2^v}\right)^t$.

A family G of 2-wise independent 2-1 hash functions $g\colon \{0,1\}^{m-v} \to \{0,1\}^{m-v-1}$

A family H (induced by Π, G) of 2-1 hash functions $h : \{0,1\}^m \to \{0,1\}^{m-1}$ defined as:

$$h(x) \stackrel{\text{def}}{=} \pi(x)_1, \ldots, \pi(x)_v, g\left(\pi(x)_{v+1} \ldots, \pi(x)_m\right)$$

where $\pi(x)_i$ denotes the i^{th} bit of $\pi(x)$.

Input of Alice: $\perp$.

Input of Bob: $W \in \{0,1\}^m$.

- **Alice:** Choose $\pi \stackrel{\text{R}}{\leftarrow} \Pi$. Send π to Bob.
- **Bob:** Compute $z_1, \ldots z_m = \pi(W)$. Send $\pi'(W) = z_1, \ldots, z_v$ to Alice (let π' denote π when truncated to its first v bits).
- **Alice:** Choose $g \stackrel{\text{R}}{\leftarrow} G$. Send g to Bob.
- **Bob:** Send $g(z_{v+1}, \ldots, z_m)$ to Alice.
- **Alice and Bob:** Output W_0, W_1 s.t. $h(W_0) = h(W_1) = h(W)$.

Fig. 3. The four message protocol for interactive hashing.

Theorem 5.6 *For all s, m such that $s \geq \log m + 2$, the* 4M-IH *protocol is an $(s, 2^{-(m-s)+O(\log m)})$-secure η'-uniform interactive hashing protocol for $\eta' = \left(\frac{1}{2^{s-\log m-1}}\right)^m < 2^{-m}$.*

Proof: We start by noting that the protocol is efficient for both parties due to the efficiency of the permutations used. Furthermore, they can run in small space. This is an η'-uniform interactive hashing protocol since h is η close to pairwise independent. Therefore for any value w' and input W, the distribution of the pair $(h(W), h(w'))$ is η-close to that of a random permutation. The probability that $W_{1-d} = w'$ is therefore η-close to $\frac{1}{2^m-1}$. Summing the distance to uniformity over all possible values w', we get that W_{1-d} is distributed $\eta' = 2^m \cdot \eta$ close to uniform. The 4M-IH protocol is secure for B since no matter what strategy

A^* Alice uses, the messages that Bob sends are identical whether his input is $W = W_0$ or $W = W_1$ (recall that $h(W_0) = h(W_1)$). This protocol has two stages of question and answer (4 messages), and in order to prove the security for A we view each of these two parts separately. In the first part, all strings $W \in \{0,1\}^m$ are divided by π' into 2^v cells (according to the value of $\pi'(W)$). Our goal is to show that no cell $z' \in \{0,1\}^v$ has too many strings from the bad set S mapped to it. The second part of the protocol can then be viewed as implementing the 2M-IH protocol on strings in the cell z', yielding the security of the combined protocol (the portion of bad strings in the cell z' is reduced to less than a square root of the strings in the cell). We start by bounding the probability that a specific set of t strings are mapped by π' to the same cell z.

Claim 5.7 *For every $z \in \{0,1\}^v$ and all $x_1, \ldots, x_t \in \{0,1\}^m$ we have that:*

$$\rho = \Pr_{\pi \in \Pi}[\pi'(x_1) = \pi'(x_2) = \ldots = \pi'(x_t) = z] \leq \left(\frac{1}{2^v}\right)^t + \eta.$$

Proof: Suppose that π were a t-wise independent function (and not permutation), then for every $x_i \in \{0,1\}^m$ we have that the probability that $\pi'(x_i) = z$ is exactly $\frac{1}{2^v}$ and the probability that this is the case for t different values is exactly $\left(\frac{1}{2^v}\right)^t$. But since π is a permutation, this probability is smaller since for every i we have $\Pr[\pi'(x_i) = z \mid \pi'(x_1) = \pi'(x_2) = \ldots \pi'(x_{i-1}) = z] \leq \frac{1}{2^v}$. But as π is actually an almost t-wise independent permutation, the probability on t elements may deviate by up to η from the truly random permutation, and therefore $\rho \leq \left(\frac{1}{2^v}\right)^t + \eta$. ■

Let us focus on a specific cell $z \in \{0,1\}^v$. For every set of t elements $x_1, \ldots, x_t \in S$, denote by $Y_z^\pi(x_1, \ldots, x_t)$ the indicator random variable that indicates whether or not all x_i are mapped to z by π'. That is:

$$Y_z^\pi(x_1, \ldots, x_t) = \begin{cases} 1 & \pi'(x_1) = \pi'(x_2) = \ldots = \pi'(x_t) = z \\ 0 & \text{otherwise.} \end{cases}$$

Let Y_z^π denote the number of strings from S mapped to cell z by π'. Let $E = \frac{2^s}{2^v}$, which is the expected number of strings from S in each cell, if the cells were divided uniformly among all strings in S. We claim that with high probability, Y_z^π does not deviate much from E.

Lemma 5.8 *Let $t - 1 \leq E$. Then for all $z \in \{0,1\}^v$,*

$$\Pr_{\pi \in \Pi}[Y_z^\pi \geq 3E] \leq 2^{-(t-1)}.$$

Proof: Consider the table of all possible $Y_z^\pi(x_1, \ldots, x_t)$, where each row stands for a specific set $\{x_1, \ldots, x_t\}$ where each $x_i \in S$, and each column stands for a choice of π. By Claim 5.7, the fraction of ones in each row, and hence the fraction of ones in the whole table, is at most $\left(\frac{1}{2^v}\right)^t + \eta$. On the other hand, for each π such that $Y_z^\pi \geq 3E$ there are at least $\binom{3E}{t}$ sets of t elements for which $Y_z^\pi(x_1,\ldots,x_t)=1$, therefore the fraction of ones in the table is at least $\Pr_{\pi\in\Pi}[Y_z^\pi \geq 3E]\cdot\binom{3E}{t}/\binom{2^s}{t}$. Therefore we get that:

$$\Pr_{\pi\in\Pi}\left[Y_z^\pi \geq 3E\right] \leq \frac{\binom{2^s}{t}}{\binom{3E}{t}}\left(\left(\frac{1}{2^v}\right)^t+\eta\right).$$

Recall that $\eta=\left(\frac{1}{2^v}\right)^t$ and using the fact that $\binom{a}{c}/\binom{b}{c}\leq\left(\frac{a}{b-c+1}\right)^c$ we get:

$$\Pr_{\pi\in\Pi}\left[Y_z^\pi \geq 3E\right] \leq \left(\frac{2^s}{3E-t+1}\right)^t\cdot 2\cdot\left(\frac{1}{2^v}\right)^t.$$

Since $t-1\leq E$ and $E=\frac{2^s}{2^v}$,

$$\begin{aligned}\Pr_{\pi\in\Pi}\left[Y_z^\pi \geq 3E\right] &\leq 2\cdot\left(\frac{2^s}{2E2^v}\right)^t\\ &\leq 2\cdot\left(\frac{2^s}{2\frac{2^s}{2^v}2^v}\right)^t = 2\cdot 2^{-t}.\end{aligned}$$

This completes the proof of Lemma 5.8. ■

As a corollary of Lemma 5.8 we get that with high probability there is no cell that contains a large number of bad elements. Applying a union bound gives:

$$\Pr_{\pi\in\Pi}\left[\exists z \text{ s.t. } Y_z^\pi \geq 3E\right] \leq 2^{-(t-1-v)}.$$

Recall that $t=m$ and $v=s-\log m$. So the condition $t-1\leq E$ in Lemma 5.8 holds, and thus the probability of error here is $2^{-(m-s)-\log m+1}$. Assuming that indeed for all cells z we have $Y_z^\pi<3E$, then the second part of the protocol is actually running the 2M-IH on the strings in a specific cell z'. This cell contains all the possible extensions of z' into an m bit string. Therefore, the 2M-IH is run on strings of length $m'=m-v$. There are no more than $2^{s'}=4\cdot 2^{s-v}$ strings that belong to the bad set S. According to Claim 5.5 the second part of the protocol is an $(s',2^{-(m'-2s'+1)})$-interactive hashing protocol. The probability that Bob can choose a cell with two string from the bad set is therefore $2^{-(m'-2s'+1)}=2^{-(m-v-2(s-v+2)+1)}=2^{-(m-s)+\log m+3}$. Combined with the probability that there exist a z with $Y_z^\pi\geq 3E$ we get that the probability that any strategy B^* that Bob plays succeeds in choosing both W_0 and W_1 in the set S is at most $2^{-(m-s)+O(\log m)}$. ■

6 The Oblivious Transfer protocol

Our BS-OT protocol is presented in figure 4. The protocol relies on three ingredients: An extractor, a min-entropy sampler, and an interactive hashing protocol. The precise requirements from the ingredients are presented in figure 5.

In our suggested implementation of BS-OT we choose Samp_A to be the sampler from [54], Ext to be an extractor from [51] and use the 4M-IH interactive

Input of Alice: Secret strings $s_0, s_1 \in \{0,1\}^u$.
Input of Bob: Choice bit $c \in \{0,1\}$.
Setup Stage:
Subsets Stage: Alice and Bob store subsets of the string $\mathcal{R} \in \{0,1\}^N$.

- **Alice:** Choose $P \overset{\text{R}}{\leftarrow} [L_A]$. Compute $A \subset [N]$ of size n by $A = \mathrm{Samp}_A(P)$ and store the bits $\mathcal{R}_A$.
- **Bob:** Choose random $B \subset [N]$ of size n and store the bits $\mathcal{R}_B$.
- **Alice:** Send A to Bob by sending P.
- **Bob:** Determine $C = A \cap B$. If $|C| < \ell$ abort. If $|C| > \ell$, randomly truncate it to be of size ℓ.
- **Bob:** Compute t_m as in Definition 3.3. Choose $Q \overset{\text{R}}{\leftarrow} [t_m]$ and compute $W = F_m(C, Q)$.*

Interactive Hashing Stage: Interactively hash W.

- **Bob:** Input W into the interactive hashing protocol.
- **Alice and Bob:** Interactively obtain h and W_0, W_1 s.t. $h(W_0) = h(W_1) = h(W)$. Compute the subsets C_0, C_1 encoded by W_0, W_1. If W_0 or W_1 isn't a valid encoding then abort.

Choice Stage:

- **Bob:** Let $d \in \{0,1\}$ be such that $W_d = W$. Send $e = c \oplus d$.
- **Alice:** For $i \in \{0,1\}$ send $Y_i \overset{\text{R}}{\leftarrow} \{0,1\}^{d_E}$.

Transfer Stage:

- **Alice:** Set $X_0 = \mathcal{R}_{C_0}$ and $X_1 = \mathcal{R}_{C_1}$.
- **Alice:** Send "encrypted" values of s_0 and s_1: For $i \in \{0,1\}$, Send $Z_i = s_{i \oplus e} \oplus \mathrm{Ext}(X_i, Y_i)$.
- **Bob:** Compute $X = \mathcal{R}_C$. Bob's output is given by $\mathrm{Ext}(X, Y_{c\oplus e}) \oplus Z_{c \oplus e}$

* The range of F_m is $[n]$ and not $A = \mathrm{Samp}_A(P)$. For simplicity, we treat C as a subset of A.

Fig. 4. Protocol BS-OT for 1-2 OT in the bounded storage model.

hashing protocol from the previous section. The precise choices of parameters for these ingredients appear in Section 6.1. These choices meet the requirements of figure 5 with $\epsilon = 2^{-\Omega(\ell)}$.The main theorem of this paper asserts that this implementation of BS-OT is a constant round protocol for oblivious transfer in the bounded storage model. At first reading, the reader may safely ignore the sampler and assume that the set A is chosen uniformly at random. That is assume that Samp_A is the identity mapping on $\binom{[N]}{n}$.[18]

Theorem 6.1 *There is a constant $\alpha > 0$ such that if N, n and ℓ satisfy $\log n \leq \ell \leq n^\alpha$ then for every constant $\nu < 1$ let protocol* BS-OT *use the ingredients described in Section 6.1. Protocol* BS-OT *is a $(1-\epsilon)$-oblivious transfer protocol against storage $K = \nu N$, with $\epsilon = 2^{-\Omega(\ell)}$. Furthermore:*

- *The protocol has 5 messages.*
- *The strategies for Alice and Bob runs in time $\mathrm{poly}(n)$ and space $k = O(n \log n)$.*
- *The protocol passes secrets of length $u = \Omega(\ell)$.*

[18] Using different samplers allows choosing a "random" set A which has a shorter description. Specifically, using the sampler from Section 6.1 reduces the description size of A from $\log \binom{N}{n} = \Theta(n \log n)$ to $O(\ell)$.

Parameters:

- N - the length of the long random string $\mathcal{R}$.
- n - the number of bits honest players remember about $\mathcal{R}$.
- u - the length of the secrets.
- $\ell = n^2/2N$ - the size of the intersection set.
- ν - the dishonest receiver remembers at most νN bits about $\mathcal{R}$.
- ϵ - the error of the protocol. The protocol works as long as ϵ is not too small. See the precise requirement on ϵ under "additional requirements" below.

Ingredients:

- A $(\delta_A, \delta'_A, \phi_A, \epsilon_A)$-min-entropy sampler $\mathrm{Samp}_A : [L_A] \to [N]^n$ with:
  - $\delta_A \leq (1-\nu)/2$.
  - $\delta'_A = \delta_A/8$.
  - $\phi_A \leq \epsilon/20$.
  - $\epsilon_A \leq (\epsilon/20)^2$.
  - L_A determines the length of the first message sent by Alice.
- A (k_E, ϵ_E)-strong extractor $\mathrm{Ext} : \{0,1\}^{n_E} \times \{0,1\}^{d_E} \to \{0,1\}^{m_E}$ with:
  - $n_E = \ell$
  - $d_E \leq \delta'_A \ell/12$
  - $m_E = u \leq \delta'_A \ell/12$.
  - $k_E \leq \delta'_A \ell/6$.
  - $\epsilon_E \leq (\epsilon/20)^2$.
- An (s, ρ)-secure (2^{-m})-uniform interactive hashing protocol for strings of length $m = 10\ell \log n$ with:
  - $s \leq m - c_{IH}\delta'_A \ell/\log(1/\delta'_A) + 1$ ($c_{IH} > 0$ is a universal constant chosen in the proof).
  - $\rho \leq \epsilon/20$.

Additional requirements:

- $\epsilon \geq 2^{-c\delta'_A \ell/\log(1/\delta'_A)}$ where $c > 0$ is a universal constant chosen in the proof.

Fig. 5. Ingredients and requirements for Protocol BS-OT.

- *The overall number of bits exchanged is* $TC = O(\ell^{O(1)})$.

The constants hidden in ϵ, s, u *and* TC *above depend on* ν.[19]

The results mentioned in the introduction can be obtained by choosing $n = N^{1/2+a}/\log N$ for some small constant $a > 0$. Note that if a is sufficiently small then the space of honest players satisfies $k = O(n \log n) = O(N^{1/2+a}) \leq O(K^{1/2+a})$, where the last inequality follows for every constant fraction $\nu < 1$. As $\ell = n^2/2N$ we have that $\ell = n^{2a}/2\log N \geq k^a$ for large enough n, and we have that $\epsilon = 2^{-\Omega(\ell)} = 2^{-\Omega(k^a)}$. The remainder of the paper is devoted to proving Theorem 6.1. In Section 6.1 we explain how to choose the ingredients in a way that satisfies the requirements in Figure 5. In Section 7 we prove the

[19] Tracing this dependency gives that for $\delta = (1-\nu)$: $\epsilon = 2^{-\Omega(\delta\ell/\log(1/\delta))}$, $s = m - O(\delta\ell/\log(1/\delta))$, and $u = \Omega(\delta\ell)$. This holds even when ν isn't a constant as long as $n \geq \ell/\delta^4$. That is, the Theorem holds even for $\nu \approx 1 - (\ell/n)^4$.

functionality and security of the protocol. The most challenging part is proving Alice's security.

6.1 Choosing the ingredients

We now turn to choose the ingredients for BS-OT to get the parameters guaranteed in Theorem 6.1. Given n, N, u, ν, we shoot for $\epsilon = 2^{-\Omega(\ell)}$. We need to show an extractor, a sampler and an interactive hashing protocol that satisfy the conditions specified in figure 5.

The extractor. In [51] it was shown how to construct a (k_E, ϵ_E)-strong extractor, $\mathrm{Ext}\colon\{0,1\}^\ell \times \{0,1\}^{d_E} \to \{0,1\}^u$, for every k_E, $u \le k_E - 2\log(1/\epsilon_E) - O(1)$, and $d_E = c\log(1/\epsilon_E)$ for some constant c, as long as $\log(1/\epsilon) > \log^4 \ell$. Setting $k_E = \delta'_A \ell/6$, we can get $u = \delta'_A \ell/12$ for $d_E \le \delta'_A \ell/6$ and $\epsilon_E = 2^{-c'\delta'_A \ell}$ for some constant $c' > 0$ (which depends on c). This choice satisfies the requirements in figure 5. We note that the above extractor can be computed in time and space polynomial in ℓ.

The sampler. In [54] it was shown how to construct a (μ, θ, γ)-averaging sampler $\mathrm{Samp} : [L] \to [v]^t$ with distinct samples for every $\mu > \theta > 0$ and $\gamma > 0$ as long as $t \ge \Omega(\log(1/\gamma)/\theta^2)$. This sampler has seed length $\log L \le \log(v/t) + \log(1/\gamma)(1/\theta)^{O(1)}$. Let $v = N$ and $t = n$. By Corollary 3.11, for every δ, γ such that $\log(1/\gamma)/\delta^4 \le n$, this sampler yields a $(\delta, \delta/2, (\gamma+2^{-\Omega(\delta N)})^{1/2}, (\gamma+2^{-\Omega(\delta N)})^{1/2})$-min-entropy sampler $\mathrm{Samp}_A : [L_A] \to [N]^n$. Setting $\gamma = 2^{-\ell}$ we have that as long as $n \ge \ell/\delta^4$, this sampler has $\phi = \epsilon = 2^{-\Omega(\ell)}$, and $\log L_A \le \log(N/n) + \ell(1/\delta)^{O(1)} \le \log n + \ell(1/\delta)^{O(1)}$ for $n \ge \sqrt{N}$. Note that the condition $n \ge \ell/\delta^4$ is satisfied when ν is a constant (as in this case $\delta = \delta_A$ is also a constant).[20] We also note that the above sampler can be computed in time polynomial in n and space $O(n)$.

The interactive hashing protocol. We need to show that protocol 4M-IH satisfies the requirements of figure 5. It is required there that 4M-IH is $(s, 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))})$-secure for $s \le m - c_{IH}\delta'_A \ell/\log(1/\delta'_A) + 1$ where $c_{IH} > 0$ is some universal constant and $\delta'_A = \alpha(1-\nu)$ for some $\alpha > 0$. By Theorem 5.6, we have that

$$\rho \le 2^{-(m-s)+O(\log m)} \le 2^{-c_{IH}\delta'_A \ell/\log(1/\delta'_A)+O(\log m)} \le 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}$$

as $m = 10\ell\log n$ and $\ell \ge \log n$. When ν is a constant, δ'_A is also a constant and we have that $\rho = 2^{-\Omega(\ell)}$ as required. We note that Protocol 4M-IH requires time and space polynomial in ℓ.

[20] We remark that we don't have to require that ν is a constant. Our protocol also works for $\nu = 1 - o(1)$ as long as the condition above ($n \ge \ell/\delta^4$) is satisfied.

7 Proof of Main Theorem

7.1 The functionality of the OT protocol

The following Lemma asserts that protocol BS-OT indeed implements oblivious transfer.

Lemma 7.1 *For every choice of ingredients for* BS-OT *and every s_0, s_1, c, if Alice and Bob follow protocol* BS-OT *then*

- *With probability $1 - 2^{-\Omega(\ell)}$ the protocol does not abort.*
- *If the protocol does not abort then Bob's output is indeed s_c.*

Proof: We first show that with high probability $|A \cap B| \geq \ell$. This is because for every fixed A, as B is a random set the expected size of $A \cap B$ is $n^2/N \geq 2\ell$. A standard Lemma (see for example Corollary 3 in [22]) can be used to show that there exists a constant $0 < \alpha < 1$ such that probability that $|A \cap B| < \ell$ is at most $2e^{-\alpha\ell}$. We now show that the probability that one of W_0, W_1 is not a valid encoding of a subset is small. W_d was chosen by Bob and is certainly a valid encoding. By the definition of Interactive Hashing, the other string W_{1-d} is η-close to uniformly distributed in $\{0,1\}^m$, for $\eta < 2^{-m}$. By Lemma 3.4 the probability that a random string $W \in \{0,1\}^m$ is not a valid encoding is at most $\binom{n}{\ell}2^{-m} \leq 2^{\ell \log n - m} \leq 2^{-\ell-1}$ as $m = 10\ell \log n$. It follows that the probability of abort is bounded by $2^{-m} + 2^{-\ell-1} \leq 2^{-\ell}$. To see that whenever the protocol does not abort Bob indeed outputs s_c, we observe that $X = \mathcal{R}_C$ is known to Bob (since $C = A \cap B \subseteq B$ and Bob has stored all the bits $\mathcal{R}_B$). In particular, Bob is always able to compute $\mathrm{Ext}(X, Y_{c\oplus e})$ and subsequently use it in order to "decrypt" the value $Z_{c\oplus e}$. By the definition of the protocol we then have:

$$\begin{aligned}
\mathrm{Ext}(X, Y_{c\oplus e}) \oplus Z_{c\oplus e} &= \mathrm{Ext}(X, Y_d) \oplus (s_c \oplus \mathrm{Ext}(X_d, Y_d)) \\
&= \mathrm{Ext}(X, Y_d) \oplus (s_c \oplus \mathrm{Ext}(X, Y_d)) \qquad (2) \\
&= s_c.
\end{aligned}$$

where Eq. (2) follows from the fact that X_d equals $\mathcal{R}_C$ $(= X)$, which in turns follows from the fact that $C_d = C$ (since $W_d = W$ and the encoding F_m is one-to-one). The lemma follows. ■

We next verify that the protocol indeed meets the promised efficiency properties.

Lemma 7.2 *Let* Ext, Samp_A *and* IH *be chosen as in Theorem 6.1. Then the properties in the itemized list in Theorem 6.1 are satisfied by protocol* BS-OT*.*

Proof: It is easy to verify that the protocol has 5 messages (not including the transmission of $\mathcal{R}$). By section 6.1 the extractor and sampler run in time polynomial in n and space $\ell^{O(1)} + O(n)$. Protocol 4M-IH runs in time and space polynomial in $m = 10\ell \log n$. Thus, both parties run in time polynomial in n. Both parties require space n to store $\mathcal{R}_A$ and $\mathcal{R}_B$ and space $m^{O(1)}$ to play 4M-IH. Alice's set A is chosen by a sampler with seed length $\log L_A = O(\ell)$, thus it can be stored in space $O(\ell)$. Overall, Alice's space is bounded by $O(n) + poly(\ell)$.

Bob's set B is a random set, and thus takes $O(n \log n)$ bits to store. We conclude that both players can run their strategies in space $O(n \log n) + poly(\ell)$ which is bounded by $O(n)$ for sufficiently small α as required. The protocol passes secrets of length m_E where $m_E = \Omega(\ell)$. Finally, the longest message sent in the protocol is the description of the permutation π in the interactive which is of length at most $\ell^{O(1)}$. ■

7.2 Security for Bob

The following Theorem guarantees the security of Bob.

Theorem 7.3 *For every choice of ingredients of* BS-OT*, the protocol is secure for Bob.*

Proof: We show that for any strategy A^*, the view of A^* is independent of the bit c. This is shown by the following argument: Fix the randomness of A^* and $\mathcal{R}$. We show a perfect bijection between possible pairs of B's randomness r_B and input c. That is, for each pair (r_B, c) that is consistent with the view V of A^*, there exists a unique pair $(r'_B, 1-c)$ such that r'_B and $1-c$ are consistent with the same view V. There are two possible options for a $V = \text{view}_{A^*}^{\langle A^*, B\rangle}$:

- The protocol aborts before the choice stage where Bob sends Alice the value $e = c \oplus d$. In such a case, the view V is totally independent of c and we map every consistent r_B to itself ($r'_B = r_B$). Clearly r_B is consistent with both $c = 0$ and $c = 1$.
- V includes the message $e = c \oplus d$ sent by Bob. In such a case, suppose that (r_B, c) is consistent V. That is, r_B is the randomness that chooses the random set B so that $C = A \cap B$ is encoded by the string W_d. By the fact that the protocol did not abort, we are assured that also W_{1-d} encodes a legal set C'. Then we choose r'_B to be the randomness that chooses $B' = (B \setminus C) \cup C'$ and encodes C' by W_{1-d}. This perfectly defines $(r'_B, 1-c)$ that is consistent with the view V. Furthermore, $(r'_B, 1-c)$ is mapped by the same process back to (r_B, c), hence we get a perfect bijection.

Theorem 7.3 follows. ■

7.3 Security for Alice

The following theorem (which is technically the most challenging theorem of this paper) guarantees Alice's security against bounded storage receivers. This theorem refers to a list of requirements on the parameters of the ingredients which appears in figure 5.

Theorem 7.4 *For every $\nu < 1$ (not necessarily constant), if all the requirements in figure 5 are met, then protocol* BS-OT *is $(\nu N, \epsilon)$-secure for Alice.*

We prove Theorem 7.4 in Section 7.5. In section 6.1 we showed that Ext and Samp_A and 4M-IH satisfy all the requirements in figure 5 for $\epsilon = 2^{-\Omega(\ell)}$. Theorem 7.4 thus implies the following corollary which in turn completes the proof of Theorem 6.1.

Corollary 7.5 *Let* Ext, Samp_A *and* IH *be chosen as in Theorem 6.1. Protocol* BS-OT *is $(\nu N, \epsilon)$-secure for Alice, for $\epsilon = 2^{-a\ell}$ where $a > 0$ is a constant that depends on ν.*

7.4 Overview of the proof

Theorem 7.4 regarding Alice's security is somewhat technical and involves many parameters. We find it instructive to first give a sketch of the proof while ignoring the precise parameters. The formal proof appears in the next section. Fix some bounded storage strategy B^* with storage bound νN for some $\nu < 1$, and an input c. We need to show that there exists a random variable Γ determined in the setup stage such that for every two pairs of secrets s, s' which are Γ-consistent the view of B^* is distributed roughly the same whether Alice's input is s or s'. Recall that in the protocol, the secrets s_0, s_1 are only involved in the transfer stage where $z_i = \mathrm{Ext}(X_i, Y_i) \oplus s_i$ for $i \in \{0, 1\}$. Our goal is to show that there exists a random variable I determined in the setup stage such that for every choice of secrets s_0, s_1, the string Z_I is close to uniformly distributed from B^*'s point of view. More precisely, for every $i \in \{0, 1\}$ we split Alice's messages into Z_i and all the rest of the messages which we denote by MSG_i. For every fixing of r of $\mathcal{R}$ and msg_i of MSG_i, B^*'s point of view on Z_i is captured by considering the distribution $Z'_i = (Z_i \mid g^*(\mathcal{R}) = g^*(r), MSG_i = msg_i)$. We show that for most fixings r and msg_I, the random variable Z'_I is close to uniformly distributed. We now explain how we achieve this goal. It is instructive to first consider a simplified scenario in which B^* chooses to remember the original content of $\mathcal{R}$ at νN indices. We call these indices "bad" indices, and the remaining $(1-\nu)N$ indices "good" indices. Let $\delta = (1 - \nu)$. The proof proceeds as follows:

1. We note that B^* does not remember the δN good indices.
2. When Alice uses a sampler to choose A, with high probability she hits a large fraction (say $\delta n/2$) of the good indices.
3. We have that the set A contains many good indices. If we were to choose a random subset of A with ℓ indices, then with high probability we will hit many (say $\delta\ell/4$) good indices. Let S be the set of all such subsets which hit less indices. By the above argument S is a small set.
4. It follows that when Alice and Bob use interactive hashing to determine the subsets C_0 and C_1, at least one of the subsets is not in S. We define the random variable I to be the index of this subset. It follows that C_I contains many good indices.
5. We now consider $X_I = \mathcal{R}_{C_I}$ given MSG_I. As it contains many good indices, it has high min-entropy. It follows that with high probability over the choice of Y_I, $\mathrm{Ext}(X_I, Y_I)$ is close to uniformly distributed even given MSG_I. Thus,

Z_I is close to uniformly distributed as required. However, a subtlety is that $\mathcal{R}_{C_0}$ and $\mathcal{R}_{C_1}$ are *not* independent, consequently Z_{1-I} may give information about $\mathcal{R}_{C_I}$, and hence may potentially give some information about Z_I. We explain how to resolve this issue below in the proof sketch for the general case where B^* computes an arbitrary memory function of $\mathcal{R}$.

We now sketch how to make this argument work when B^* is allowed to remember an arbitrary function $g^* : \{0,1\}^N \to \{0,1\}^{\nu N}$ of $\mathcal{R}$. Intuitively, the notion of "min-entropy" replaces that of "good bits" in this case.

1. It is easy to see that for most fixings r of $\mathcal{R}$, the random variable $(\mathcal{R} \mid g^*(\mathcal{R}) = g^*(r))$ has high min-entropy (say $\Omega(\delta N)$).
2. When Alice uses a min-entropy sampler for most fixings p of P she obtains a set A such that $(\mathcal{R}_A \mid g^*(\mathcal{R}) = g^*(r), P = p)$ has high min-entropy.
3. Choosing a random subset is a min-entropy sampler, and thus for most choices of a subset of C of size ℓ, $(\mathcal{R}_C \mid g^*(\mathcal{R}) = g^*(r), P = p)$ has high min-entropy.
4. As before it follows that following the interactive hashing with high probability there exists an I such that $(\mathcal{R}_{C_I} \mid g^*(\mathcal{R}) = g^*(r), P = p)$ has high min-entropy.
5. We note that $\mathcal{R}_{C_0}$ and $\mathcal{R}_{C_1}$ are *not* independent, as mentioned earlier. Thus, it may be the case that Z_{1-I} gives information about $\mathcal{R}_{C_I}$. However, we set the parameters so that $\mathcal{R}_{C_I}$ has min-entropy much larger than the length of the pair (Z_{1-I}, Y_{1-I}). As a consequence we can argue that for most fixings z_{1-I} and y_{1-I}, $(\mathcal{R}_{C_I} \mid g^*(\mathcal{R}) = g^*(r), P = p, Z_{1-I} = z_{1-I}, Y_{1-I} = y_{1-I})$ has high min-entropy. Thus, running an extractor, with high probability over Y_I we obtain a distribution which is close to uniform given MSG_I just as before.

7.5 The Full Proof of Alice's Security

In this section we assume that the parameters of the Extractor, Sampler and Interactive Hashing protocol satisfy the requirements in figure 5. We fix some bounded storage strategy B^* with storage bound νN for some $\nu < 1$. We also fix some secrets s_0, s_1 and choice bit c. Consider the probability space obtained in the interaction of $Alice(s_0, s_1)$ and $B^*(c)$ when receiving a random $\mathcal{R}$. We can assume w.l.o.g. that B^* is deterministic, as otherwise we consider every possible fixing of his random coins. The view of B^* depends on $g^*(\mathcal{R})$ and Alice's messages. We denote the random variable of Alice's messages by $MSG = (P, MSG_{IH}, Y_0, Y_1, Z_0, Z_1)$ where P is the seed for the sampler used to choose A, MSG_{IH} is the messages sent by Alice in the Interactive Hashing protocol, and Y_0, Y_1, Z_0, Z_1 are defined in protocol BS-OT. The security of the protocol follows from the following technical Lemma: (We say that a bounded storage strategy B^* is non-aborting if it always sends messages which are syntactically consistent with the protocol and in addition it never claims that the intersection $C = A \cap B$ is too small in the subset stage.)

Lemma 7.6 *For every s_0, s_1 and c, and every non-aborting bounded storage strategy B^* with storage bound νN, there exists a subset $G_{7.6}$ of 6-tuples $(b, p, msg_{IH}, z', y', y'')$ and a random variable I such that:*

- *I is determined in the setup stage (it does not depend on messages sent in the choice stage).*
- $\Pr[(g^*(\mathcal{R}), P, MSG_{IH}, Z_{1-I}, Y_{1-I}, Y_I) \in G_{7.6}] \geq 1 - \beta_{7.6}$ *for* $\beta_{7.6} = 2^{-(1-\nu)N/2} + \phi_A + \rho + 2^{-\delta'_A \ell/6} + \sqrt{2(\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))})} + 2\sqrt{\epsilon_E}$.
- *For every $(b, p, msg_{IH}, z', y', y'') \in G_{7.6}$, $(Z_I \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y', Y_I = y'')$ is $\epsilon_{7.6}$-close to uniform for $\epsilon_{7.6} = \sqrt{2(\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))})} + \sqrt{\epsilon_E}$.*

Loosely speaking, this means that there is a bit I, (which is determined *before* Alice uses s_0, s_1) such that Bob's view does not depend on the secret "encrypted" in Z_I. Indeed this Lemma suffices to prove Theorem 7.4.

Proof of theorem 7.4: Let B^* be some bounded storage strategy with storage bound νN. We assume w.l.o.g. that B^* is non-aborting. If B^* chooses to abort at some stage we modify it to answer randomly from that point on. Let $G_{7.6}$ and I be the set and random variable guaranteed in Lemma 7.6. We define $\Gamma = (1 - I) \oplus e$ where e is the bit sent by B^* at the choice stage. Let $\bar{s}$ and $\bar{s}'$ be two pairs of secrets which are Γ-consistent. Let $V(s_0, s_1, c)$ denote the view of B^* when given input c and interacting with $Alice(s_0, s_1)$. We need to show that $V(\bar{s}, c)$ and $V(\bar{s}', c)$ are statistically close. We first observe that for every 6-tuples $(b, p, msg_{IH}, z', y', y'') \in G_{7.6}$, the probability that $g^*(\mathcal{R}) = b$ and Alice sends $P = p$, $MSG_{IH} = msg_{IH}$, $Z_{1-I} = z'$, $Y_{1-I} = y'$ and $Y_I = y''$ is identical no matter whether Alice plays with $\bar{s}$ or $\bar{s}'$. This is trivial for all messages except Z_{1-I} as they are sent in the setup stage and do not depend on the secrets. It holds for Z_{1-I} because Z_{1-I} depends only on $s_{(1-I)\oplus e}$. Note that $s_{(1-I)\oplus e}$ is identical in the two cases as $\bar{s}$ and $\bar{s}'$ are Γ-consistent and $\Gamma = (1 - I) \oplus e$. Furthermore we observe that:

$$(V(\bar{s}, c) \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y', Y_I = y'') \text{ is} \tag{3}$$

$$2\epsilon_{7.6}\text{-close to}$$

$$(V(\bar{s}', c) \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y', Y_I = y'').$$

This holds because $V(s_0, s_1, c)$ is determined by $g^*(\mathcal{R})$ and Alice's messages $MSG = (P, MSG_{IH}, Y_0, Y_1, Z_0, Z_1)$, and by Lemma 7.6, $(Z_I \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y', Y_I = y'')$ is $\epsilon_{7.6}$-close to uniform. Thus, (3) follows by the triangle inequality. By (3) above the two random variables $V(\bar{s}, c)$ and $V(\bar{s}', c)$ are $2\epsilon_{7.6}$-close on a set $G_{7.6}$ which has probability $1 - \beta_{7.6}$. Therefore, they are $(2\epsilon_{7.6} + \beta_{7.6})$-close. Thus the final error is given by

$$2\epsilon_{7.6} + \beta_{7.6} \leq 2\left(\sqrt{2(\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))})} + \sqrt{\epsilon_E}\right)$$

$$+2^{-(1-\nu)N/2}+\phi_A+\rho+2^{-\delta'_A\ell/6}+\sqrt{2(\epsilon_A+3\cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))})}+2\sqrt{\epsilon_E}$$

$$\le 2^{-(1-\nu)N/2}+\phi_A+\rho+2^{-\delta'_A\ell/6}+3\cdot(\sqrt{2(\epsilon_A+3\cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))})}+2\sqrt{\epsilon_E})$$

$$\le 2^{-(1-\nu)N/2}+\phi_A+\rho+2^{-\delta'_A\ell/6}+6\sqrt{\epsilon_A}+9\cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))}+6\sqrt{\epsilon_E})$$

$$\le 2^{-\delta'_A\ell}+\epsilon/20+\epsilon/20+2^{-\delta'_A\ell/6}+6\epsilon/20+9\cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))}+6\epsilon/20$$

$$\le \frac{14\epsilon}{20}+2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))}\le\epsilon$$

where all the inequalities above except the last one follow by the requirements in Figure 5. For the last inequality we choose the constant c that appears in the requirement that $\epsilon\ge 2^{-c\delta'_A\ell/\log(1/\delta'_A)}$ in Figure 5 to be small enough so that the inequality holds. ■

The remainder of this section is devoted to proving Lemma 7.6. We will follow the messages of the protocol one by one to obtain the set G and random variable I. We start by showing that with high probability B^* forgets at least $(1-\nu)N/2$ bits of information about $\mathcal{R}$.[21]

Lemma 7.7 *There exists $G_{7.7}\subseteq\{0,1\}^{\nu N}$ such that:*

- $\Pr[g^*(\mathcal{R})\in G_{7.7}]\ge 1-\beta_{7.7}$ *for* $\beta_{7.7}=2^{-(1-\nu N)/2}$.
- *For every $b\in G_{7.7}$, $(\mathcal{R}\mid g^*(\mathcal{R})=b)$ is a $((1-\nu)N/2)$-source.*

Proof: We apply Lemma 3.15 where R plays the role of X, $g^*(R)$ plays the role of Y, and $\beta=2^{-(1-\nu)N/2}$. It follows that there is a subset $G_{7.7}\subseteq\{0,1\}^{\nu N}$ such that $\Pr[g^*(R)\in G_{7.7}]\ge 1-\beta$ and for every $b\in G_{7.7}$, $(\mathcal{R}\mid g^*(\mathcal{R})=b)$ is a $((1-\nu)N/2)$-source. ■

Our next step is to show that when Alice chooses A, with high probability B^* does not remember a lot about $\mathcal{R}_A$.

Lemma 7.8 *There exists $G_{7.8}\subseteq\{0,1\}^{\nu N}\times[L_A]$ such that*

- $\Pr[(g^*(\mathcal{R}),P)\in G_{7.8}]\ge 1-\beta_{7.8}$ *for* $\beta_{7.8}=\beta_{7.7}+\phi_A=2^{-(1-\nu)N/2}+\phi_A$.
- *For every $(b,p)\in G_{7.8}$, the random variable $(\mathcal{R}_{\mathrm{Samp}_A(P)}\mid g^*(\mathcal{R})=b,P=p)$ is ϵ_A-close to a $\delta'_A n$-source. .*

Proof: Let $G_{7.7}$ be the set from Lemma 7.7. As $\delta_A N\le(1-\nu)N/2$, we have that for every $b\in G_{7.7}$, $(R\mid g^*(R)=b)$ is a $\delta_A N$-source. By the properties of Samp_A it follows that for every $b\in G_{7.7}$ there exists a set $G_b\subseteq[L_A]$ with density $1-\phi_A$ such that for $p\in G_b$, $(R_{\mathrm{Samp}_A(p)}\mid g^*(\mathcal{R})=b)$ is ϵ_A-close to a $\delta'_A n$-source. Let $G_{7.8}=\{(b,p)\mid b\in G_{7.7},p\in G_b\}$. The lemma follows. ■

We now show that B^* does not remember a lot about $\mathcal{R}_C$ for a random subset $C\subseteq A$.

[21] It is instructive to consider a player B^* that sets $g^*(\mathcal{R})=0^{\nu N}$ if and only if $\mathcal{R}=0^N$. In the case that $\mathcal{R}=0^N$, B^* remembers all of $\mathcal{R}$. The lemma says that such a case happens with very small probability.

Lemma 7.9 *For every* $(b,p) \in G_{7.8}$*, let* $G_{(b,p)}$ *be the set of all* $C \subseteq \mathrm{Samp}_A(p)$ *such that* $|C| = \ell$ *and* $(\mathcal{R}_C \mid g^*(R) = b, P = p)$ *is* $\epsilon_A + 3 \cdot 2^{-\Omega(\delta_A \ell / \log(1/\delta_A))}$*-close to a* $\delta'_A \ell/2$*-source. Then for every* $(b,p) \in G_{7.8}$*, the density of* $G_{(b,p)}$ *in the set* $\binom{[n]}{\ell}$ *is at least* $1 - 3 \cdot 2^{-\Omega(\delta'_A \ell / \log(1/\delta'_A))}$*.*

Proof: Let $(b,p) \in G_{7.8}$. By Lemma 7.8 we have that $(\mathcal{R}_{\mathrm{Samp}_A(p)} \mid g^*(\mathcal{R}) = b, P = p)$ is ϵ_A-close to a $\delta'_A n$-source. Consider the random subset sampler RS which samples sets of size ℓ in a universe of size n. We think of this sampler as sampling subsets of $\mathrm{Samp}_A(p)$. By Corollary 3.14 RS is a $(\delta'_A, \delta'_A/2, 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}, 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))})$-min-entropy sampler. It follows that there exists a subset $G_{(b,p)} \subseteq \binom{[n]}{\ell}$ of density $1 - 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}$ such that for every $C \in G_{(b,p)}$ such that $(\mathcal{R}_{\mathrm{Samp}_A(p) \cap C} \mid g^*(\mathcal{R}) = b, P = p)$ is $\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}$-close to a $\delta'_A \ell/2$-source. ■

We now show that after the interactive hashing phase, with high probability there exists an i such that B^* does not remember a lot about $\mathcal{R}_{C_i}$.

Lemma 7.10 *There exists a set* $G_{7.10}$ *of triples* (b, p, msg_{IH}) *such that:*

- $\Pr[(g^*(\mathcal{R}), P, MSG_{IH}) \in G_{7.10}] \geq 1 - \beta_{7.10}$ *for* $\beta_{7.10} = \beta_{7.8} + \rho = 2^{-(1-\nu)N/2} + \phi_A + \rho$*.*
- *For every* $(b, p, msg_{IH}) \in G_{7.10}$ *there exists an* $i \in B$ *such that* $(\mathcal{R}_{C_i} \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH})$ *is* $\epsilon_{7.10}$*-close to a* $\delta'_A \ell/2$*-source for* $\epsilon_{7.10} = \epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}$

Proof: Let $G_{7.8}$ be the set from Lemma 7.8. For every $(b,p) \in G_{7.8}$, let $G_{(b,p)}$ be the set defined in Lemma 7.9. For every pair $(b,p) \in G_{7.8}$ we define set $S_{(b,p)} \subseteq \{0,1\}^m$ as follows: Recall the encoding F_m (defined in definition 3.3) which encodes pairs (C, Q), we think of C as a subset of $\mathrm{Samp}_A(P)$ of size ℓ and $Q \in [t_m]$ (where t_m is defined in definition 3.3). We use D to denote the image of F_m. We define

$$S'_{(b,p)} = \{W \in \{0,1\}^m \mid \ W \notin D\}$$

(i.e. those for which F_m^{-1} is undefined). We define

$$S''_{(b,p)} = \{W \in \{0,1\}^m \mid \text{ such that } (C,Q) = F_M^{-1}(W) \text{ is defined and } C \notin G_{(b,p)}\}.$$

Finally, we define $S_{(b,p)} = S'_{(b,p)} \cup S''_{(b,p)}$. We now show that $S_{(b,p)}$ is small. By Lemma 3.4 we have that $|D|/2^m \geq 1 - \binom{n}{\ell}2^{-m}$. Thus, $|S'_{(b,p)}| \leq \binom{n}{\ell} \leq 2^{\ell \log n} \leq 2^{m/10}$ as $m = 10\ell \log n$. We now bound $S''_{(b,p)}$. We have that the density of $G_{(b,p)}$ is at least $1 - 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}$. As F_m outputs any set C the same number of times, we have that $|S''_{(b,p)}| \leq 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))} \cdot 2^m$. Overall, we have that

$$|S_{(b,p)}| \leq 2^{m/10} + 2^m \cdot 3 \cdot 2^{-\Omega(\delta'_A \ell/\log(1/\delta'_A))}$$

We now choose the constant c_{IH} which appear in Figure 5 on the requirement from the interactive hashing protocol. We define c_{IH} to be the constant hidden in the $\Omega(\cdot)$ notation above. We thus have that:

$$|S_{(b,p)}| \leq 2^{m/10} + 2^{m - c_{IH}\delta'_A \ell / \log(1/\delta'_A)}$$

Note that $m = 10\ell \log n$ and therefore the second term is larger and we conclude that

$$|S_{(b,p)}| \leq 2^{m - c_{IH}\delta'_A \ell / \log(1/\delta'_A) + 1}.$$

As the interactive hashing protocol is (s, ρ)-secure for $s \geq m - c_{IH}\delta'_A\ell/\log(1/\delta'_A) + 1$, we have that for each $(b,p) \in G_{7.8}$, there is a subset $G'_{(b,p)}$ of messages msg_{IH} of probability measure at least $1 - \rho$ such that for each $msg_{IH} \in G'_{(b,p)}$, there exists an $i \in \{0,1\}$ such that $C_i \in G_{(b,p)}$. Define $G_{7.10}$ to be the set of triples (b, p, msg_{IH}) such that $(b,p) \in G_{7.8}$ and $msg_{IH} \in G'_{(b,p)}$. It follows that for each $(b, p, msg_{IH}) \in G_{7.10}$, $(\mathcal{R}_{C_i} \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH})$ is $\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))}$-close to a $\delta'_A\ell/2$-source. ∎

We can now identify the secret that B^* does not learn.

Definition 7.11 (Identifying the secret) *We define a random variable $I = I(R, P, MSG_{IH})$ as the bit i from Lemma 7.10. I is defined arbitrarily when $(g^*(R), P, MSG_{IH}) \notin G_{7.10}$. Note that I is determined in the setup stage.*

It follows that by running an extractor on $\mathcal{R}_{C_i}$ Alice obtains a string Z_i which is (close to) uniform from B^*'s point of view. Thus, intuitively the message Z_i does not give information about the secret encoded in it. However, this guarantee is not sufficient to prove the security of the protocol. This is because at the last step of the protocol Alice sends Z_{1-i} which may depend on Z_i and give B^* more information. We therefore prove that with high probability B^* does not remember a lot about $\mathcal{R}_{C_i}$ even when given Z_{i-1}, Y_{i-1}.

Lemma 7.12 *There exists a subset $G_{7.12}$ of 5-tuples (b, p, msg_{IH}, z', y') such that*

- $\Pr[(g^*(\mathcal{R}), P, MSG_{IH}, Z_{1-I}, Y_{1-I}) \in G_{7.12}] \geq 1 - \beta_{7.12}$ *for* $\beta_{7.12} = \beta_{7.10} + 2^{-\delta'_A\ell/6} + \sqrt{2\epsilon_{7.10}} = 2^{-(1-\nu)N/2} + \phi_A + \rho + 2^{-\delta'_A\ell/6} + \sqrt{2(\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))})}$.
- *For every $(b, p, msg_{IH}, z', y') \in G_{7.12}$, the random variable $(\mathcal{R}_{C_I} \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y')$ is $\epsilon_{7.12}$-close to a $\delta'_A\ell/6$-source, for* $\epsilon_{7.12} = \sqrt{2\epsilon_{7.10}} = \sqrt{2(\epsilon_A + 3 \cdot 2^{-\Omega(\delta'_A\ell/\log(1/\delta'_A))})}$

Proof: (of lemma 7.12) Let $G_{7.10}$ be the set from Lemma 7.10. Fix some $(b, p, msg_{IH}) \in G_{7.10}$ we have that $(\mathcal{R}_{C_I} \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH})$ is $\epsilon_{7.10}$ -close to a $\delta'_A\ell/2$-source. Note that the total length of (Z_{1-I}, Y_{1-I}) is no more than $\delta'_A\ell/6$. We now use Lemma 3.16 with the following parameters: The probability space for the Lemma is the initial probability space of the protocol conditioned on the event $E = \{g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}\}$. $\mathcal{R}_{C_I}$ plays the role of X and (Z_{1-I}, Y_{1-I}) play the role of Y. We use $\beta = 2^{-\delta'_A \ell/6}$ and $\xi = \epsilon_{7.10}$. We conclude that with probability $1 - \beta - 2\sqrt{\epsilon_{7.10}}$ over choosing y', z' from $(Y_{1-I}, Z_{1-I} \mid E)$, $(\mathcal{R}_{C_I} \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y')$ is $\epsilon_{7.12} = \sqrt{2\epsilon_{7.10}}$ close to a $\delta'_A\ell/6$-source. We define $G_{7.12}$ to be the set of all 5-tuples (b, p, msg_{IH}, z', y') such that $(b, p, msg_{IH}) \in G_{7.10}$ and z', y' are good in the sense explained above. ■

It follows that when Alice applies an extractor to $\mathcal{R}_{C_I}$, she obtains a string which is (close to) uniform even when conditioned on the rest of Alice's messages. We are finally ready to prove Lemma 7.6.

Proof: (of Lemma 7.6) Let $G_{7.12}$ be the set from Lemma 7.12 and I be the random variable defined in definition 7.11. We have that for every $(b, p, msg_{IH}, z', y') \in G_{7.12}$, the random variable $(\mathcal{R}_{C_I} \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y')$ is $\epsilon_{7.12}$-close to a $\delta'_A\ell/6$-source. As Ext is a $(\delta'_A\ell/6, \epsilon_E)$-strong extractor it follows from Lemma 3.17 that with probability $1-2\sqrt{\epsilon_E}$ over choosing y from $(Y_I \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y')$, $(\mathrm{Ext}(\mathcal{R}_{C_I}, Y_I) \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y', Y_I = y)$ is $\epsilon_{7.12} + \sqrt{\epsilon_E}$-close to uniform. We define $G_{7.6}$ to be the set of all tuples whose prefix is in $G_{7.12}$ and y is good in the sense explained above. It follows that $Z_I = s_{I \oplus e} \oplus \mathrm{Ext}(R_{C_I}, Y_I)$ satisfies that $(Z_I \mid g^*(\mathcal{R}) = b, P = p, MSG_{IH} = msg_{IH}, Z_{1-I} = z', Y_{1-I} = y', Y_I = y)$ is $\epsilon_{7.6}$-close to uniform. ■

8 Conclusions and open problems

We have constructed the first constant-round protocol for oblivious transfer in the bounded storage model. Our protocol involves only 5 messages. As a main building block, we have constructed a novel 4-message interactive hashing protocol using almost t-wise independent permutations. Our interactive hashing protocol may be of independent interests.

Our OT protocol also has some additional improvements over previous work [9, 22], including total communication efficiency, memory requirement, probability of aborting, and handling of long secrets. Our protocol achieves $k \approx \sqrt{K} \approx \sqrt{N}$, where k is the space requirement of honest parties, K is the space bound of a malicious receiver, and N is the length of the public random string $\mathcal{R}$. In words, the space of the honest parties is about a square root of the space allowed for the malicious parties. This space requirement has recently been proved to be *optimal* in [25].[22]

[22] In [25], it is shown that for secure key agreement (KA) against a passive eavesdropper in the bounded storage model, the product of the space required of Alice and Bob must be at least $\Omega(K)$, where K the adversay's storage bound. It is well known (c.f. [29]) that a secure OT protocol yields a secure KA protocol. Moreover, the standard reduction from KA to OT has the following properties: Suppose that one is given an OT protocol that requires honest Alice and Bob of space k_A and k_B

Our 5-message OT protocol attains a very small error $\epsilon = 2^{-\Omega(\ell)}$ against a malicious receiver, and has a total communication of $\ell^{O(1)}$ bits, where $\ell = |A \cap B|$ is size of the intersection of the sets of indices in $[N]$ sampled by Alice and Bob respectively, and $\ell > u$ where u is the length of the secrets. In the case of a large secret length u, if one settles for a larger yet still negligibly small error, e.g. $\epsilon = 2^{-\log^c N}$, the communication complexity can be further reduced, by using a randomness efficient averaging sampler $\mathrm{Samp}_B : [L_B] \to [N]^n$ for Bob in choosing his set B. That is, instead of choosing B uniformly at random, Bob chooses a random seed $y \leftarrow [L_B]$ for Samp_B, and computes $B = \mathrm{Samp}_B(y)$. Then the seed y is input to the interactive hashing protocol, instead of an encoding of B. The rest of the protocol is essentially the same. The security of the resulting protocol variant can be proved by slightly modifying the proof in Section 7. Since the density of A is subconstant, we need a sampler Samp_B for a subconstant average μ (recall Definition 3.8). Although optimal averaging samplers for a subconstant average have not been constructed, a short seed length of say $O(\log N \log (1/\epsilon))$ can be achieved by constructions based on t-wise independence (c.f. [4]). Using such a sampler, the communication complexity of the resulting protocol becomes $O(u + (\log N \log (1/\epsilon))^{O(1)})$. When $\log (1/\epsilon) \ll u$, the saving would be significant.

Our new 4-message interactive hashing protocol can replace the NOVY protocol of [44] in our setting. A similar phenomena was observed also in the context of Zero-Knowledge. Damgård [17] used the NOVY protocol to give certain transformations of "honest verifier" Zero-Knowledge protocols into general Zero-Knowledge protocols. Later works [19, 34] replaced the NOVY protocol with a constant-round protocol. This raises the interesting question of whether the NOVY protocol can be replaced by a constant-round protocol for the application in [44], that is, for constructing perfectly hiding bit commitment schemes from arbitrary one-way permutations. We remark that constant-round perfectly hiding bit commitment schemes are known only using seemingly stronger assumptions [47, 21, 32]. The NOVY protocol achieves a stronger security for interactive hashing than the one defined here. This stronger security allows its use in the application of [44]. Loosely speaking, it is shown in [44] that their protocol is secure in the following sense: For every malicious strategy B^* for Bob, there is a "simulator" $A_{B^*}(W')$ with running time polynomial in that of B^*, such that for most $W' \in \{0,1\}^m$, the simulator $A_{B^*}(W')$ can run B^*, play Alice's role, and generate a perfectly simulated random transcript in which one of the two outputs is W'. Intuitively, this is a stronger and *computational* form of the notion that Bob does not "control" the two outputs. Obtaining this stronger property with a constant number of rounds seems hard. A very related open problem was raised in [20] in the context of Zero-Knowledge.

---

respectively, and is secure against a malicious party with space bound K. Then the resulting KA protocol invokes the given OT protocol exactly once, requires the two communicating parties Alice and Bob of the same space k_A and k_B, and achieves the given OT protocol's security against an eavesdropper with space bound $K - k_B$. Thus by the lower bound of [25], $k_A k_B \geq \Omega(K - k_B)$.

It is interesting to further study simulation-based definitions for protocols in the bounded storage model, and investigate whether it is possible to construct efficiently simulatable protocols. Positive results may allow composition of such protocols with secure protocols in the standard complexity-based model, with the benefit of combining the salient features of both worlds.

Acknowledgements. We thank Yuval Ishai for bringing [9] to our attention and Oded Goldreich for helpful discussions and for pointing out some relevant work. We are also grateful to Moni Naor for insights on the NOVY protocol. Thanks also to Claude Crépeau, Salil Vadhan and the reviewers for their helpful remarks.

References

1. Y. Aumann, Y.Z. Ding, and M. O. Rabin. Everlasting security in the bounded storage model. *IEEE Transactions on Information Theory*, 48, 2002.
2. Y. Aumann and M. O. Rabin. Information theoretically secure communication in the limited storage space model. In *Advances in Crypology – CRYPTO '99*, volume 1666, pages 65–79, 1999.
3. M. Bellare and S. Micali. Non-interactive oblivious transfer and applications. In *Advances in Cryptology - CRYPTO '89, Lecture Notes in Computer Science*, volume 435, pages 547–557. Springer, 1989.
4. M. Bellare and J. Rompel. Randomness-efficient oblivious sampling. In *35th IEEE Symposium on Foundations of Computer Science*, pages 276–287, 1994.
5. C.H. Bennett, G. Brassard, C. Crépeau, and M.H. Skubiszewska. Practical quantum oblivious transfer protocols. In *Advances in Cryptology - CRYPTO '91, Lecture Notes in Computer Science*, volume 576, pages 351–366. Springer, 1992.
6. B. Brassard, C. Crépeau, and M. Santha. Oblivious transfers and intersecting codes. *IEEE Transactions on Information Theory*, 42(6):1769–1780, 1996.
7. B. Brassard, C. Crépeau, and S. Wolf. Oblivious transfers and privacy amplification. *Journal of Cryptology*, 16(4):219–237, 2003.
8. A. Brodsky and S. Hoory. Simple permutations mix even better. In *Arxiv Math.CO/0411098*, 2004.
9. C. Cachin, C. Crépeau, and J. Marcil. Oblivious transfer with a memory-bound receiver. In *39th IEEE Symposium on Foundations of Computer Science*, pages 493–502, 1998.
10. C. Cachin and U. Maurer. Unconditional security against memory-bound adversaries. In *Advances in Cryptology – CRYPTO '97*, pages 292–306, 1997.
11. T.M. Cover. Enumerative source encoding. *IEEE Transaction on Information Theory*, 19(1):73–77, 1973.
12. C. Crépeau. Equivalence between two flavours of oblivious transfers. In *Advances in Cryptology - CRYPTO '87, Lecture Notes in Computer Science*, volume 293, pages 350–354. Springer-Verlag, 1987.
13. C. Crépeau. Verifiable disclosure of secrets and applications. In *Advances in Cryptology - EUROCRYPT '89, Lecture Notes in Computer Science*, volume 434, pages 150–154. Springer-Verlag, 1989.

14. C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions. In *29th IEEE Symposium on Foundations of Computer Science*, pages 42–52, 1988.
15. C. Crépeau, G. Savvides, C. Schaffner, and J. Wullschleger. Information-theoretic conditions for two-party secure function evaluation. In *Advances in Cryptology - EUROCRYPT '06, Lecture Notes in Computer Science*. Springer-Verlag, 2006.
16. C. Crépeau, J. van de Graaf, and A. Tapp. Committed oblivious transfer and private multi-party computation. In *Advances in Cryptology - CRYPTO '89, Lecture Notes in Computer Science*, volume 965, pages 110–123. Springer-Verlag, 1995.
17. I. Damgård. Interactive hashing can simplify zero-knowledge protocol design without computational assumptions. In *Advances in Cryptology - CRYPTO '93, Lecture Notes in Computer Science*, volume 773, pages 100–109. Springer, 1993.
18. I. Damgård, S. Fehr, L. Salvail, and C. Schaffner. Cryptography in the bounded quantum-storage model. In *46th IEEE Symposium on Foundations of Computer Science*, pages 449–458, 2005.
19. I. Damgård, O. Goldreich, T. Okamoto, and A. Wigderson. Honest verifier vs dishonest verifier in public cain zero-knowledge proofs. In *Advances in Cryptology - CRYPTO '95, Lecture Notes in Computer Science*, volume 963, pages 325–338. Springer, 1995.
20. I. Damgård, O. Goldreich, and A. Wigderson. Information theory versus complexity theory: Another test case, 1995.
21. I. Damgård, T. Pedersen, and B. Pfitzmann. On the existence of statistically hiding bit commitment schemes and fail-stop signatures. In *Advances in Cryptology - CRYPTO '93, Lecture Notes in Computer Science*, volume 773, pages 250–265. Springer, 1993.
22. Y.Z. Ding. Oblivious transfer in the bounded storage model. In *Advances in Cryptology - CRYPTO '01, Lecture Notes in Computer Science*, volume 2139, pages 155–170. Springer, 2001.
23. Y.Z. Ding and M.O. Rabin. Hyper-encryption and everlasting security. In *Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 1–26, 2002.
24. Y. Dodis and S. Micali. Lower bounds for oblivious transfer reductions. In *Advances in Cryptology - EUROCRYPT '99, Lecture Notes in Computer Science*, volume 1592, pages 42–55. Springer-Verlag, 1999.
25. S. Dziembowski and U. Maurer. On generating the initial key in the bounded storage model. In *Advances in Cryptology - Eurocrypt '04, Lecture Notes in Computer Science*, volume 3027, 2004.
26. S. Dziembowski and U. Maurer. Optimal randomizer efficiency in the bounded-storage model. *Journal of Cryptology*, 17(1):5–26, 2004.
27. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.
28. M. Fischer, S. Micali, and C. Rackoff. A secure protocol for the oblivious transfer. *Journal of Cryptology*, 9(3):191–195, 1996.
29. Y. Gertner, S. Kannan, T. Malkin, O. Reingold, and M. Viswanathan. The relationship between public key encryption and oblivious transfer. In *41st IEEE Symposium on Foundations of Computer Science*, pages 325–335, 2000.
30. O. Goldreich. A sample of samplers - a computational perspective on sampling (survey). In *Electronic Colloquium on Computational Complexity (ECCC) (20)*, volume 4, 1997.

31. O. Goldreich. *Foundations of Cryptography - Volume II Basic Applications*. Cambridge University Press, 2004.
32. O. Goldreich and A. Kahan. How to construct constant-round zero-knowledge proof systems for NP. *Journal of Cryptology*, 9(2):167–189, 1996.
33. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game - a completeness theorem for protocols with honest majority. In *19th ACM Symposium on the Theory of Computing*, pages 218–229, 1987.
34. O. Goldreich, A. Sahai, and S. Vadhan. Honest-verifier statistical zero-knowledge equals general statistical zero-knowledge. In *30th ACM Symposium on the Theory of Computing*, pages 399–408, 1998.
35. S. Goldwasser and L. Levin. Fair computation of general functions in presence of immoral majority. In *Advances in Cryptology - CRYPTO '90, Lecture Notes in Computer Science*, volume 537, pages 77–93. Springer-Verlag, 1990.
36. W.T. Gowers. An almost m-wise independent random permutation of the cube. *Combinatorics, Probability and Computing*, 5:119–130, 1996.
37. D. Hong, K.Y. Chang, and H. Ryu. Efficient oblivious transfer in the bounded-storage model. In *Advances in Cryptology – ASIACRYPT '02, Lecture Notes in Computer Science*, pages 143–159. Springer-Verlag, December 2002.
38. S. Hoory, A. Magen, S. Myers, and C. Rackoff. Simple permutations mix well. In *ICALP*, pages 770–781, 2004.
39. R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *21st ACM Symposium on the Theory of Computing*, pages 44–61, 1989.
40. E. Kaplan, M. Naor, and O. Reingold:. Derandomized constructions of k-wise (almost) independent permutations. In *RANDOM-APPROX '05, Lecture Notes in Computer Science*, volume 3624, pages 354–365, 2005.
41. J. Kilian. Founding cryptography on oblivious transfer. In *20th ACM Symposium on the Theory of Computing*, pages 20–31, 1988.
42. C. Lu. Encryption against space-bounded adversaries from on-line strong extractors. *Journal of Cryptology*, 17(1):27–42, 2004.
43. U. Maurer. Conditionally-perfect secrecy and a provably-secure randomized cipher. *Journal of Cryptology*, 5(1):53–66, 1992.
44. M. Naor, R. Ostrovsky, R. Venkatesan, and M. Yung. Perfect zero-knowledge arguments for np using any one-way permutation. *Journal of Cryptology*, 11(2):87–108, 1998. preliminary version in CRYPTO 92.
45. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *SIAM Symposium on Discrete Algorithms (SODA 2001)*, pages 448–457, 2001.
46. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-rackoff revisited. *Journal of Cryptology*, 12(1):29–66, 1999.
47. M. Naor and M. Yung. Universal one-way hash functions and their cryptographic applications. In *21st ACM Symposium on the Theory of Computing*, pages 33–43, 1989.
48. N. Nisan. Extracting randomness: How and why, a survey. *IEEE Conference on Computational Complexity*, pages 44–58, 1996.
49. N. Nisan and D. Zuckerman. Randomness is linear in space. *JCSS*, 52(1):43–52, 1996.
50. M. O. Rabin. How to exchange secrets by oblivious transfer. TR-81, Harvard, 1981.
51. R. Raz, O. Reingold, and S. Vadhan. Error reduction for extractor. In *40th IEEE Symposium on Foundations of Computer Science*, pages 191–201, 1999.

52. O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. In *41st IEEE Symposium on Foundations of Computer Science*, pages 22–31, 2000.
53. R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the EATCS*, 77:67–95, 2002.
54. S.P. Vadhan. On constructing locally computable extractors and cryptosystems in the bounded storage model. *Journal of Cryptology*, 17(1):43–77, 2004.
55. A. C. Yao. How to generate and exchange secrets. In *27th IEEE Symposium on Foundations of Computer Science*, pages 162–167, 1986.