Oblivious Transfer with a Memory-Bounded Receiver

Christian Cachin
M.I.T. *

Claude Crépeau †
McGill University ‡

Julien Marcil †
Université de Montréal §

Abstract

We propose a protocol for oblivious transfer that is unconditionally secure under the sole assumption that the memory size of the receiver is bounded. The model assumes that a random bit string slightly larger than the receiver's memory is broadcast (either by the sender or by a third party). In our construction, both parties need memory of size in $\theta(n^{2-2\alpha})$ for some $\alpha < \frac{1}{2}$, when a string of size $N = n^{2-\alpha-\beta}$ for $\alpha > \beta > 0$ is broadcast, whereas a malicious receiver can have up to γN bits of memory for any $\gamma < 1$. In the course of our analysis, we provide a direct study of an interactive hashing protocol closely related to that of Naor et al. *[NOVY98].*

1 Introduction

Oblivious transfer is an important primitive in modern cryptography. It was introduced to cryptography in several variations by Rabin and Even *et al.* [Rab81, EGL83] and had been studied already by Wiesner [Wie70] (under the name of "multiplexing"), in a paper that marked the birth of quantum cryptography. Oblivious transfer has since become the basis for realizing a broad class of cryptographic protocols, such as bit commitment, zero-knowledge proofs, and general secure multiparty computation [Yao86, GMW87, GV88, Kil88a, CvdGT95].

In a one-out-of-two oblivious transfer, denoted $\binom{2}{1}$-OT, one party Alice owns two secret bits b_0 and b_1, and another party Bob wants to learn b_c for a secret bit c of his choice. Alice is willing to collaborate provided that Bob does not learn any information about $b_{c\oplus 1}$, but Bob will not participate if Alice can obtain information about c.

Traditionally, the security of $\binom{2}{1}$-OT is based on computational assumptions, such as the hardness of factoring or the existence of trapdoor one-way permutations [EGL83, GMW87, BM90]. $\binom{2}{1}$-OT can also be implemented in terms of Rabin's OT [Rab81], in which Alice sends a bit b that is received by Bob with probability $\frac{1}{2}$ [Cré88]. The security of Rabin's protocol for OT is based on the quadratic residuosity problem.

These are relatively strong computational assumptions. However, it is also known that OT cannot likely be based on weaker assumptions: Proving that OT is secure assuming only a one-way function in a black-box reduction is as hard as proving *P≠NP* [IR89]. OT falls thus, together with key agreement, in the class of tasks that are only known how to implement using at least trapdoor one-way functions.

However, if Alice and Bob have access to a quantum channel, Oblivious Transfer can be reduced to a weaker primitive known as Bit Commitment [BBCS92, Cré84] and thus is secure assuming only a one-way function in the quantum computer model. Oblivious transfer can also be based on a noisy channel, as shown by Crépeau and Kilian [CK88, Cré97].

In this paper we describe how a bound on memory size of the receiver Bob can be used to implement oblivious transfer. We assume that there is an initial broadcast of a huge amount of random data, during which Bob is free to compute any probabilistic function with unlimited power. As long as the function's output size is bounded and does not exceed Bob's memory size (storage space), we can prove that the OT protocol is secure. No computational or memory restrictions are placed on Alice.

In order to carry out the protocol, both parties need to use some amount of memory, however. Let α, β be constants such that $0 < \beta < \alpha < \frac{1}{2}$ (e.g. a small β and $\alpha = \frac{1}{2} - \beta$). In our construction, both parties need memory of size in $\theta(n^{2-2\alpha})$ when $N = n^{2-\alpha-\beta}$ random bits are broadcast. The security of the oblivious transfer can be shown if Bob has no more than γN bits of storage for any $\gamma < 1$.

The random broadcast can be generated by a trusted random source, which needs not necessarily be an artificial device. Natural sources, such as deep-space radio sources or the cosmic background radiation could also be used. On the other hand, there is no need for a trusted third party to generate the random data. Alice can also generate the ran-

---

* MIT Laboratory for Computer Science, Cambridge, MA 02139, USA, cachin@acm.org.

† Supported in part by Canada's NSERC and Québec's FCAR.

‡ School of Computer Science, McGill University, Montréal (Québec), Canada, crepeau@cs.mcgill.ca.

§ Département d'Informatique et R.O., Université de Montréal, Montréal (Québec), Canada, marcilj@iro.umontreal.ca.

dom bits herself and send them to Bob, since no assumption about her memory limitation is made.

The study and comparison of different assumptions under which cryptographic tasks can be realized is an important aspect of research in cryptography. Perhaps the most prominent assumptions used today in the computational security model are factoring, the discrete logarithm problem, and lattice reduction problems [AD97]. However, factoring and computing discrete logarithms could be solved efficiently on a quantum computer [Sho94], while exisitng systems based on lattice reductions have been cryptanalised [NS98]. Alternatives to computational security assumptions that have been proposed include quantum cryptography, the noisy channel model, and memory bounds [CM97].

The memory bound model seems realistic in the view of current communication and high-speed networking technologies that allow transmission at rates of multiple gigabits per second. Storage systems on the order of petabytes, on the other hand, require a major investment by a potential adversary. Furthermore, the model is attractive for the following reasons: (1) the security can be based only on the assumption about the adversary's memory capacity. (2) storage costs scale linearly and can therefore be estimated accurately. (3) memory bounds offers permanent protection in the sense that future technological improvements cannot retrospectively compromise the security of messages transmitted earlier.

Smartcards provide a particularly well suited scenario to implement our protocol. In such a scenario, Alice could be a teller machine and Bob a card. Limiting the computing power of a card is a resonable assumption whereas a similar limitation on the teller machine would be much less resonable. Since $\binom{2}{1}$-OT in one direction is sufficient to implement it in both directions (see [CS91]), any two-party cryptographic task may be implemented securily in this situation from our protocol. For instance, a mutual identification scheme may be realized [CS95].

1.1 Our Construction

We provide an implementation of $\binom{2}{1}$-OT. During the initial random broadcast, Alice and Bob both store a random subset of the N bits such that their parts overlap in k positions. Then they engage in a protocol to form two sets of k bits each among the bits stored by Alice: a "good" set consisting of the bits also known to Bob and a "bad" set containing at least some bits unknown to Bob. This is done using an interactive hashing protocol similar to that of Naor *et al.* [NOVY98].

Interactive hashing is a protocol between Alice and Bob for isolating two binary strings. One string is chosen by Bob and the other one is chosen randomly, without (much) influence by Bob. However, Alice does not learn which string corresponds to Bob's input. In order to apply interactive hashing, we use two tools of independent interest.

The first tool is an efficiently computable, dense encoding of k-element subsets from $\{1, \ldots, n\}$, i.e., a mapping of k-element subsets to binary strings of length $\lg \binom{n}{k}$. It has to be efficient in the sense that encoding and decoding operate in time polynomial in n rather than $\binom{n}{k}$, even if k is proportional to n. Such a scheme has been long known in the litterature [Cov73]. The second tool is a direct analysis of interactive hashing, since the original analysis is based on simulators and not directly applicable to our setting.

Once two binary strings corresponding to the two sets are isolated, it will be the case that Bob knows all bits in the good set, but only few bits in the bad set. Then Bob asks Alice to encode b_0 and b_1 using the two sets such that b_c is encoded with the good set and $b_{c\oplus 1}$ with the bad set. Bob can recover b_c since he knows the good set, but not $b_{c\oplus 1}$.

Additional results used to show the security of the protocol are privacy amplification (or entropy smoothing) by universal hashing [BBCM95] and a theorem by Zuckerman about the min-entropy of a randomly chosen substring [Zuc96].

1.2 Related Work

For the purpose of secrecy, memory bounds have been exploited in a similar model in the cryptosystem proposed by Cachin and Maurer [CM97]. They describe a private-key cryptosystem and a protocol for key agreement by public discussion based on the assumption that an adversary's memory capacity is bounded. The security margin for their key agreement protocol is $O(n)$ memory needed for Alice and Bob versus no more than n^2 memory for an adversary.

Space bounds have also been studied with respect to interactive proof systems. Kilian [Kil88b] constructed a proof system for any language in *PSPACE*, which is zero knowledge with respect to a logarithmic space-bounded verifier. Kilian's technique can be extended to any known verifier with polynomial space bounds. In this protocol, the memory bound and interaction are interleaved in a crucial way.

De Santis *et al.* [DPY92] introduced one message proof systems with known space verifiers, showing that no interaction is needed to exploit space bounds for zero knowledge proofs. An improved construction was given by Aumann and Feige [AF94] of a one message proof system where the ratio between the maximum space tolerated and the minimum space needed by the verifier can be arbitrarily large.

We note that our construction also uses interaction in a crucial way, but the memory bound only has to be imposed for one message at the beginning, during the broadcast of the random bits. Furthermore, the receiver in our protocol is allowed to access the complete broadcast and to compute *any function* of it before interaction starts. This is not the case for the commitment protocols by De Santis *et al.* and Aumann and Feige.

In addition, and in contrast to the proof systems with memory-bounded verifiers mentioned, the data intended to overflow a receiver's memory consists of purely random bits in our protocol. Therefore, an independent random source with very high capacity can also be used for providing the random bits.

1.3 Organization of the Paper

The dense encoding of k-subsets into binary strings is described in Section 3, where we also provide our analysis of interactive hashing. Section 4 contains the protocol construction; the security proof is given in Section 5. We start with defining terminology, assembling some tools, and introducing the notation.

2 Preliminaries

A random variable X induces a probability distribution P_X over a set $\mathcal{X}$. Random variables are denoted by capital letters. If not stated otherwise, the alphabet of a random variable is denoted by the corresponding script letter.

The *(Shannon) entropy* of a random variable X with probability distribution P_X and alphabet $\mathcal{X}$ is defined as

$$H(X) \;=\; -\sum_{x\in\mathcal{X}} P_X(x)\lg P_X(x).$$

Let $h(p) = -p\lg p - (1-p)\lg(1-p)$ stand for the *binary entropy function*. The *conditional entropy* of X conditioned on a random variable Y is

$$H(X|Y) \;=\; \sum_{y\in\mathcal{Y}} P_Y(y) H(X|Y=y)$$

where $H(X|Y=y)$ denotes the entropy of the conditional probability distribution $P_{X|Y=y}$. The *min-entropy* of a random variable X is defined as

$$H_\infty(X) = -\lg \max_{x\in\mathcal{X}} P_X(x).$$

The *variational distance* between two probability distributions P_X and P_Y over the same alphabet $\mathcal{X}$ is

$$\begin{aligned}\|P_X - P_Y\|_v &= \max_{\mathcal{X}_0\subseteq\mathcal{X}} \Big|\sum_{x\in\mathcal{X}_0} P_X(x) - P_Y(x)\Big| \\ &= \frac{1}{2}\sum_{x\in\mathcal{X}} \Big|P_X(x) - P_Y(x)\Big|.\end{aligned}$$

We say that a random variable X *is ϵ-close to* Y whenever $\|P_X - P_Y\|_v \le \epsilon$.

For a sequence $x_1,\dots,x_n$ and some set $\mathcal{S}\subseteq\{1,\dots,n\}$, we abbreviate the projection of $x_1,\dots,x_n$ onto indices in $\mathcal{S}$ by $x^{\mathcal{S}}$. Similarly, $x^{[n]}$ denotes the sequence $x_1,\dots,x_n$ with the convention that $x^{[0]}$ is the empty word. Also, $x^{[n\setminus i]}$ denotes the sequence $x_1,\dots,x_{i-1},x_{i+1},\dots,x_n$. We write $\oplus$ for addition in $GF(2)$ and $\odot$ for the inner product of two vectors over $GF(2)$.

Lemma 1. *Let X be a random variable with alphabet $\mathcal{X}$, let V be an arbitrary random variable with alphabet $\mathcal{V}$, and let $r > 0$. Then with probability at least $1-2^{-r}$, V takes on a value v for which*

$$H_\infty(X|V=v) \;\ge\; H_\infty(X) - \lg\#\mathcal{V} - r.$$

Proof. Let $p_0 = 2^{-r}/\#\mathcal{V}$. Thus, $\sum_{v:P_V(v)<p_0} P_V(v) < 2^{-r}$. It follows for all v with $P_V(v)\ge p_0$

$$\begin{aligned}H_\infty(X|V=v) &= -\lg\max_{x\in\mathcal{X}} P_{X|V=v}(x) \\ &= -\lg\max_{x\in\mathcal{X}} \frac{P_X(x)P_{V|X=x}(v)}{P_V(v)} \\ &\ge -\lg\max_{x\in\mathcal{X}} \frac{P_X(x)}{p_0} \\ &= H_\infty(X) - r - \lg\#\mathcal{V}\end{aligned}$$

which proves the lemma. □

A class $\mathcal{G}$ of functions $\mathcal{X}\to\mathcal{Y}$ is 2-*universal* if, for all distinct $x_1,x_2\in\mathcal{X}$, there are at most $\#\mathcal{G}/\#\mathcal{Y}$ functions g in $\mathcal{G}$ such that $g(x_1)=g(x_2)$.

A class $\mathcal{G}$ of functions $\mathcal{X}\to\mathcal{Y}$ is *strongly* 2-*universal* if, for all distinct $x_1,x_2\in\mathcal{X}$ and all (not necessarily distinct) $y_1,y_2\in\mathcal{Y}$, exactly $\#\mathcal{G}/\#\mathcal{Y}^2$ functions from $\mathcal{G}$ take x_1 to y_1 and x_2 to y_2.

A strongly 2-universal class of hash functions can be used to generate a sequence of pairwise independent random variables in the following way: Select $G\in\mathcal{G}$ uniformly at random and apply it to any fixed sequence $x_1,\dots,x_l$ of distinct values in $\mathcal{X}$. Let $Y_j = G(x_j)$ for $j=1,\dots,l$.

Privacy amplification [BBR86, BBR88] is a method to eliminate partial information about a random variable and extract a shorter, almost uniformly distributed value. The following theorem [ILL89, BBCM95] is formulated using min-entropy, but it can be generalized to Rényi entropy of any order $\alpha > 1$ [Cac97b].

Theorem 2 ([BBCM95]). *Let X be a random variable over the alphabet $\mathcal{X}$, let G be the random variable corresponding to the random choice (with uniform distribution) from a 2-universal class $\mathcal{G}$ of hash functions $\mathcal{X}\to\mathcal{Y}$, and let $Y = G(X)$. Then*

$$H(Y|G) \;\ge\; \lg\#\mathcal{Y} - \frac{2^{\lg\#\mathcal{Y} - H_\infty(X)}}{\ln 2}. \qquad (1)$$

The following is a result by Zuckerman [Zuc96] about the min-entropy of a randomly chosen subset $X^{\mathcal{S}}$ from a sequence $X_1, \ldots, X_n$. Intuitively, one would like to show that since $\mathcal{S}$ is chosen randomly from $\{1, \ldots, n\}$, the uncertainty about $X^{\mathcal{S}}$ is roughly $\frac{\#\mathcal{S}}{n}$ times the uncertainty about $X_1, \ldots, X_n$. The exact statement is somewhat more involved.

Theorem 3 ([Zuc96]). *Let $X^{[n]}$ be an random variable with alphabet $\{0,1\}^n$ and min-entropy at least δn, let $\mathcal{S} = \{S_1, \ldots, S_l\}$ be chosen pairwise independently as described above, let $\rho = c\delta \lg \delta^{-1}$ for some positive constant c and let $\epsilon = 3/\sqrt{\rho l}$. Then, for every value $\mathbf{s} = \{s_1, \ldots, s_l\}$ there exists a random variable $W_{\mathbf{s}}$ with alphabet $\{0,1\}^l$ and min-entropy*

$$H_\infty(W_{\mathbf{s}}) \geq \rho l$$

such that with probability at least $1 - \epsilon$ (over the choice of $\mathcal{S}$), $X^{\mathcal{S}}$ is ϵ-close to $W_{\mathcal{S}}$.

3 Tools

3.1 Encoding k-Element Subsets

Let $\mathcal{S} = \{1, 2, \ldots, n\}$. A set $\mathcal{Q}$ is a *k-element subset* of $\mathcal{S}$ if $\mathcal{Q} \subseteq \mathcal{S}$ and $\#\mathcal{Q} = k$. We now give an efficient encoding of the k-element subsets as binary strings, that is, a mapping σ from the set of all k-element subsets given by a list of k integers from $\{1, \ldots, n\}$ into binary strings of length $\lceil \lg \binom{n}{k} \rceil$. Such a scheme may be found in [Cov73]. The encoding as described associates an integer in $\{0, \ldots, \binom{n}{k} - 1\}$ with the k-element subset. The corresponding string is simply the binary representation of that integer.

Without lost of generality let $\mathcal{Q} = \{e_1, e_2, \ldots, e_k\}$ be a k-element subset of $\mathcal{S}$ such that $e_i \in \mathcal{S}$ and $e_i > e_{i-1}$ for $i = 1, 2, \ldots, k$. For convenience, we use $e_0 = 0$. The $e_1, \ldots, e_k$ are the positions of 1's in the binary string of weight k starting from the left. The k-subsets of $\mathcal{S}$ correspond naturally to the binary strings of length n and weight k.

The integer associated with a binary string q of weight k is the number of strings that precede q in the list of all such strings according to the inverse lexicographical order (e.g. 11100, 11010, 11001, 10110, …). Let us count the number of strings preceding some particular string s given by $e_1, \ldots, e_k$. The leftmost 1 of s is preceded by $e_1 - 1$ zeros. Thus, for every position j, $1 \leq j \leq e_1 - 1$, there are $\binom{n-j}{k-1}$ strings of weight k with their first one in position j, each prior to s. Continuing this way of reasoning, the i-th 1 of s is preceded by 0's in the positions $e_{i-1} + 1$ to $e_i - 1$. For every position j, $e_{i-1} + 1 \leq j \leq e_i - 1$, there are $\binom{n-j}{k-i}$ strings of weight k identical to s up to position e_{i-1} with their i-th one in position j, each prior to s in the list. Summing this up over all $i = 1, \ldots, k$, we obtain the index $\sigma(\mathcal{Q})$ corresponding to s and $e_1, \ldots, e_k$. Thus, the encoding σ is given by

$$\sigma_{n,k}(\mathcal{Q}) = \sum_{i=1}^{k} \sum_{j=e_{i-1}+1}^{e_i - 1} \binom{n-j}{k-i}.$$

The decoding is done by the following procedure that takes as input an integer m and outputs the corresponding set $\mathcal{Q}$, represented by $e_1, \ldots, e_k$. It is easy to see that σ and σ^{-1} are computable in time polynomial in n. This procedure is implemented in linear space using a standard dynamic programming algorithm [BB88].

Algorithm 1 Calculate $\mathcal{Q} = \sigma_{n,k}^{-1}(m)$

for $i = 1$ **to** k **do**

$e_i \leftarrow$ biggest l such that $\sum_{j=e_{i-1}+1}^{l-1} \binom{n-j}{k-i} \leq m$

$m \leftarrow m - \sum_{j=e_{i-1}+1}^{e_i - 1} \binom{n-j}{k-i}$

end for

3.2 Interactive Hashing

Interactive hashing [NOVY98] is a protocol between a challenger Alice (with no input) and a responder Bob with input $s \in \{0,1\}^m$ and provides a way to isolate two strings. One of the strings is Bob's input s and the other one is chosen randomly; Alice does not learn which one is s. Define the 2-universal class of hash functions from $\{0,1\}^m$ to $\{0,1\}$ as

$$\mathcal{G} \;=\; \{g(x) = a \odot x \,|\, a \in \{0,1\}^m\}.$$

The protocol operates in $m - 1$ rounds. Round j, for $j = 1, \ldots, m-1$, consists of the following steps:

1. Alice chooses a function $g_j \in \mathcal{G}$ with uniform distribution. Let $a_j \in \{0,1\}^m$ be the description of g_j. If a_j is linearly dependent on $a_1, \ldots, a_{j-1}$, then Alice repeats this step until it is independent, else she announces g_j to Bob.

2. Bob computes $b_j = g_j(s) = a_j \odot s$ and sends b_j to Alice.

At the end, Alice knows $m - 1$ linear equations satisfied by s. Since the a_j's are linearly independent, the system has exactly two m-bit strings s_0, s_1 as solutions that can be found by standard linear algebra. In our application of interactive hashing, Bob can cheat if he can answer Alice's queries in such a way that both s_0, s_1 are elements of a fixed set $\mathcal{S}$.

This specific way of hashing will be the limiting factor of our construction in terms of the memory required by the participants. In order to check dependencies among the a_j's, Alice must store them all and thus memory size in $\theta(m^2)$ is necessary. Moreover, the matrix is also necessary to calculate s_0, s_1 by both parties.

If a noninteractive hash function were used, Bob could produce a collision if $\#\mathcal{S} \approx 2^{m/2}$. In contrast, Bob can only cheat in interactive hashing if the size of $\mathcal{S}$ is close to 2^m. This is shown in the remainder of this section.

Lemma 4. *Let $\mathcal{S} \subseteq \{0,1\}^m$ with $\#\mathcal{S} = 2^{\nu m}$ for $0 < \nu < 1$ and let c be a positive integer such that $c \leq \nu m/3$. Let $\mathcal{G}$ be the 2-universal class of hash functions defined above, mapping $\{0,1\}^m$ to $\{0,1\}$. Let G be a random variable with uniform distribution over $\mathcal{G}$. Then for any $b \in \{0,1\}$, G takes on a value g such that*

$$\frac{\#\{s \in \mathcal{S} | g(s) = b\}}{\#\mathcal{S}} \ < \ \frac{1}{2} + 2^{-c}$$

with probability at least $1 - 2^{-c}$.

Proof. Consider the indicator random variables for $s \in \mathcal{S}$

$$Z_s \ = \ \begin{cases} 1 & \text{if } G(s) = 0 \\ 0 & \text{otherwise} \end{cases}$$

and the sum $Z = \sum_{s \in \mathcal{S}} Z_s = \#\{s \in \mathcal{S} | G(s) = 0\}$. Similarly, let $\overline{Z} = \#\mathcal{S} - Z = \#\{s \in \mathcal{S} | G(s) = 1\}$. Let $X = \max\{Z, \overline{Z}\}$ and let $Y = \left\{ \begin{array}{ll} Z & \text{with prob. } 1/2 \\ \overline{Z} & \text{with prob. } 1/2 \end{array} \right.$.

Our goal is to show that X takes on a value x such that

$$\frac{x}{\#\mathcal{S}} \ < \ \frac{1}{2} + 2^{-c}$$

with probability at least $1 - 2^{-c}$.

But notice that $|Z - \frac{\#\mathcal{S}}{2}| = |\overline{Z} - \frac{\#\mathcal{S}}{2}|$ and therefore $|X - \frac{\#\mathcal{S}}{2}| = |Y - \frac{\#\mathcal{S}}{2}|$. In consequence, for all $\sigma > 0$ we have $\mathrm{P}\left[|X - \frac{\#\mathcal{S}}{2}| \geq \sigma\right] = \mathrm{P}\left[|Y - \frac{\#\mathcal{S}}{2}| \geq \sigma\right]$.

Therefore, it is sufficient to show that

$$|Y - \frac{\#\mathcal{S}}{2}| \geq 2^{-c}$$

with probability at most 2^{-c}.

To show this, notice that

$$\mathrm{E}[Y] = \frac{\mathrm{E}[Z]}{2} + \frac{\mathrm{E}[\overline{Z}]}{2} = \frac{\mathrm{E}[Z + \overline{Z}]}{2} = \frac{\#\mathcal{S}}{2}.$$

It follows from the Chebychev Inequality that

$$\mathrm{P}\left[|Y - \frac{\#\mathcal{S}}{2}| \geq \sigma\right] = \mathrm{P}\left[|Y - \mathrm{E}[Y]| \geq \sigma\right] \ \leq \ \frac{\mathrm{Var}[Y]}{\sigma^2}$$

for $\sigma > 0$. Thus we must also find $\mathrm{Var}[Y]$. By definition

$$\begin{aligned}
\mathrm{Var}[Y] &= \mathrm{E}[Y^2] - \mathrm{E}[Y]^2 \\
&= \mathrm{E}[Y^2] - \left(\frac{\#\mathcal{S}}{2}\right)^2 \\
&= \frac{\mathrm{E}[Z^2]}{2} + \frac{\mathrm{E}[\overline{Z}^2]}{2} - \frac{\#\mathcal{S}^2}{4} \\
&= \frac{\mathrm{E}[(Z + \overline{Z})^2 - 2Z\overline{Z}]}{2} - \frac{\#\mathcal{S}^2}{4} \\
&= \frac{\#\mathcal{S}^2}{2} - \frac{\#\mathcal{S}^2}{4} - \mathrm{E}[Z\overline{Z}] \\
&= \frac{\#\mathcal{S}^2}{4} - \mathrm{E}[Z\overline{Z}].
\end{aligned}$$

Define the indicator random variable

$$K_{s_1,s_2} \ = \ \begin{cases} 1 & \text{if } G(s_1) \neq G(s_2) \\ 0 & \text{otherwise} \end{cases}$$

and the sum

$$K = \sum_{s_1,s_2 \in \mathcal{S}^2} K_{s_1,s_2} = \#\{s_1, s_2 \in \mathcal{S}^2 | G(s_1) \neq G(s_2)\}.$$

Notice that $K = Z\overline{Z} + \overline{Z}Z$ as for every such pair (s_1, s_2) we have Z choices for s_1 and $\overline{Z}$ choices for s_2 if $G(s_1) = 0$ and $G(s_2) = 1$ and inversely when $G(s_1) = 1, G(s_2) = 0$. Therefore $\mathrm{E}[Z\overline{Z}] = \mathrm{E}[K]/2$.

However, notice that $\mathrm{E}[K] = \sum_{s_1,s_2 \in \mathcal{S}^2} \mathrm{E}[K_{s_1,s_2}]$ and that $\mathrm{E}[K_{s_1,s_2}] = 0$ if $s_1 = s_2$, while $\mathrm{E}[K_{s_1,s_2}] \geq 1/2$ when $s_1 \neq s_2$ since G is repeatedly chosen from a 2-universal class of hash functions.

In conclusion, $\mathrm{E}[K] \geq \#\{s_1, s_2 \in \mathcal{S}^2 | s_1 \neq s_2\}/2$ and $\mathrm{E}[Z\overline{Z}] \geq \frac{\#\mathcal{S}^2 - \#\mathcal{S}}{4}$, leading to

$$\mathrm{Var}[Y] = \frac{\#\mathcal{S}^2}{4} - \mathrm{E}[Z\overline{Z}] \leq \frac{\#\mathcal{S}^2}{4} - \frac{\#\mathcal{S}^2 - \#\mathcal{S}}{4} = \frac{\#\mathcal{S}}{4}.$$

Substituting $\sigma = \sqrt{2^c \#\mathcal{S}/4}$ we get

$$\mathrm{P}\left[|Y - \frac{\#\mathcal{S}}{2}| \geq 2^{\frac{(c+\nu m-2)}{2}}\right] \ \leq \ 2^{-c}.$$

But then, the reduction factor satisfies

$$\begin{aligned}
\frac{Y}{\#\mathcal{S}} &\leq \frac{1}{2} + 2^{\frac{c+\nu m-2}{2} - \nu m} \\
&< \frac{1}{2} + 2^{-c}
\end{aligned}$$

except with probability 2^{-c} and the lemma follows. □

The lemma shows that each round of interactive hashing reduces the size of $\mathcal{S}$ by a factor of almost 2, as long as $\mathcal{S}$ is large (compared to 2^c). To keep track of the overall reduction, however, we need also the following standard lemma.

Lemma 5. *Let $\mathcal{S} \subseteq \{0,1\}^m$ with $\#\mathcal{S} = 2^{\nu m}$ for $0 < \nu < 1$ and let $c, d \leq m$ be positive integers such that $2\nu m < d - c$. Let $\mathcal{G}$ be a 2-universal class of hash functions mapping $\{0,1\}^m$ to $\{0,1\}^d$. Let G be a random variable with uniform distribution over $\mathcal{G}$. The probability that G takes on a value g such that there are distinct $s_1, s_2 \in \mathcal{S}$ with $g(s_1) = g(s_2)$ is at most 2^{-c}.*

Proof. Define the function $a : \mathcal{G} \to \mathbb{N}$ to give the number of collisions in $\mathcal{S}$ for a particular g, that is,

$$a(g) \ = \ \#\{(s_1, s_2) \in \mathcal{S}^2 | g(s_1) = g(s_2), s_1 < s_2\}$$

and let $A = a(G)$. Let

$$c(s_1, s_2) \ = \ \begin{cases} \#\{g \in \mathcal{G} \mid g(s_1) = g(s_2)\} & \text{if } s_1 < s_2 \\ 0 & \text{otherwise.} \end{cases}$$

Since $\mathcal{G}$ is 2-universal, we have $c(s_1, s_2) \leq \frac{\#\mathcal{G}}{2^d}$ for all s_1, s_2. Now it is easy to see that

$$\sum_{g \in \mathcal{G}} a(g) \ = \ \sum_{(s_1,s_2) \in \mathcal{S}^2} c(s_1, s_2) \ \leq \ \frac{1}{2} \#\mathcal{S}^2 \frac{\#\mathcal{G}}{2^d}$$

and therefore $\mathrm{E}[A] \leq \frac{\#\mathcal{S}^2}{2^{d+1}} = 2^{2\nu m - d - 1}$. By the Markov Inequality, we get

$$\mathrm{P}[A \geq 1] \ \leq \ \mathrm{P}[A \geq 2^{c + 2\nu m - d - 1}] \ \leq \ 2^{-c}$$

since $2\nu m < d - c$. □

Lemma 6. *Suppose Alice and Bob engage in interactive hashing of an m-bit string held by Bob to $m - 1$ bits as described above and let $r \geq \lg m$. Let $\mathcal{S} \subseteq \{0,1\}^m$ be any subset of the inputs with cardinality $2^{\nu m}$. If $\nu < 1 - \frac{8r+4}{m}$, then the probability that Bob can answer Alice's queries such that two distinct elements s_1, s_2 of $\mathcal{S}$ are consistent with his answers is at most 2^{-r}.*

Proof. Let $\mathcal{S}_0 = \mathcal{S}$ and, for $j = 1, \ldots, m-1$, define

$$\mathcal{S}_j \ = \ \{s \in \mathcal{S}_{j-1} | g_j(s) = b_j\}.$$

As long as $\mathcal{S}_j$ is large enough, the size of $\mathcal{S}_{j+1}$ can be bounded using Lemma 4. Afterwards, we apply Lemma 5 once for the remaining rounds. Let $c = 2r$ and let $j_t = \lfloor \nu m - 3c + 1 \rfloor$ be the integer that will mark the transition. It follows from Lemma 4 by induction on j, for $1 \leq j < j_t - 1$ that

$$\#\mathcal{S}_j \ \leq \ \#\mathcal{S}\big(\frac{1}{2} + 2^{-c}\big)^j$$

except with probability at most $j2^{-c}$. In consequence, $\#\mathcal{S}_{j_t} \leq 2^{\nu m - j_t}(1 + 2^{-c+1})^{j_t}$ and we have

$$\lg(\#\mathcal{S}_{j_t}) \ \leq \ (\nu m - j_t) + j_t \lg(1 + 2^{-c+1}) \ < \ 3c + 1 \qquad (2)$$

from the definition of j_t and the fact that $j_t \lg(1 + 2^{-c+1}) < m2^{-c} < 1$. In order to apply Lemma 5 for step j_t (rounds j_t through $m - 1$ collectively) using $\mathcal{S}_{j_t}$, we need to establish

$$2 \lg(\#\mathcal{S}_{j_t}) \ \leq \ (m - 1 - j_t) - c. \qquad (3)$$

From (2) and since $\nu < 1 - \frac{4c+4}{m}$ implies that $4c < m - \nu m - 4$, we get

$$2 \lg(\#\mathcal{S}_{j_t}) \ < \ 6c + 2 \ < \ 2c + m - \nu m - 2 \qquad (4)$$

Thus, since $\nu m - 3c < j_t$ we have $2c + m - \nu m - 2 < m - 1 - j_t - c$ and (3) holds. The overall failure probability is at most $(j_t + 1)2^{-c} < m2^{-c} \leq 2^{-r}$ and the lemma follows. □

4 The Protocol

Suppose a large amount of random data (N uniformly distributed random bits) is sent from Alice to Bob over a high-capacity channel. Alternatively, the random data can be produced and broadcast by a random source R that both Alice and Bob trust to output random bits. The only assumption needed to prove the security of the protocol is that N must exceed Bob's storage capacity. If both participants are honest, they need much less memory than can be tolerated against malicious Bob. Thus, even if Alice produces the random bits, she saves only a small part of them.

In $\binom{2}{1}$-OT, which our construction implements, Alice has two input bits b_0, b_1 and Bob chooses c and obtains b_c, but Alice does not learn c. The protocol operates in the following steps. During the initial random broadcast, Alice and Bob both store a random subset of the N bits such that their parts overlap in ℓk positions. Then they engage in a way to form two sets among the bits stored by Alice, a "good" set and a "bad" set, of ℓk bits each. This is done using the interactive hashing protocol of Section 3.2 such that Alice does not learn which set is good. Bob knows all bits in the good set, but not all of the bad set. Then Bob asks Alice to encode b_0 and b_1 using the two sets such that b_c is encoded with the good set and $b_{c \oplus 1}$ with the bad set. Bob can recover b_c since he knows the bits from the good set, but cannot $b_{c \oplus 1}$ because some bits from the bad set are missing.

Included in the protocol is an additional distilliation step: the bits stored by Alice are first divided into blocks of ℓ bits each and then each block is hashed to one bit. The two sets are then formed on the level of bits and consist of k bits each.

Alice and Bob agree on the following parameters (rounding is implicit).

1. α, β such that $0 < \beta < \alpha < \frac{1}{2}$: these parameters determine the memory requirements.

2. n: number of bits that Alice and Bob store from the random broadcast.

3. $N = n^{2-\alpha-\beta}$: number of bits in the random broadcast.

4. $m = n^{1-\alpha}$: number of blocks (and bits $T_1, \ldots, T_m$).

5. $\ell = n^{\alpha}$: length of one block.

6. $k = n^{\beta}$: bound on the estimated number of blocks (and bits from $T_1, \ldots, T_m$) that overlap.

7. $\mathcal{G} = \{g | g : \{0,1\}^{\ell} \rightarrow \{0,1\}\}$: 2-universal class of hash functions for compressing blocks to bits.

The security margin in terms of memory will be that the maximum memory for a malicious Bob that can be tolerated is γN for $\gamma < 1$, versus the Nk/ℓ memory size needed for the honest players. A typical choice of the parameters could be a small β and $\alpha = \frac{1}{2} - \beta$, yielding $N = n^{1.5-2\beta}$ and $Nk/\ell = n^{1+2\beta}$.

The Protocol for $\binom{2}{1}$-OT$(b_0, b_1)(c)$:

1. Alice (or an independent source) broadcasts N random bits $r_1, \ldots, r_N$, abbreviated by $r^{[N]}$. Alice stores the n bits at positions $\mathcal{A} = \{a_1, \ldots, a_n\}$, where $\mathcal{A}$ consists of n uniformly random values from $\{1, \ldots, N\}$. Bob stores the bits at positions $\mathcal{B} = \{b_1, \ldots, b_n\}$, where $\mathcal{B}$ is chosen by Bob with uniform distribution. The substrings of $r^{[N]}$ are denoted by $r^{\mathcal{A}}$ and $r^{\mathcal{B}}$, respectively.

2. Alice sends $\mathcal{A}$ to Bob. With the bits in $r^{\mathcal{A}}$ Bob forms m blocks $x_1, \ldots, x_m$ of length $\ell = n^{\alpha}$ bits each such that the overlap $\mathcal{A} \cap \mathcal{B}$ spans at least $k = n^{\beta}$ complete blocks. If this is not possible (because the overlap is less than ℓk bits) he aborts. Let $\mathcal{S} \subseteq \{1, \ldots, m\}$ denote a set of k blocks that Bob knows completely. Formally, Bob constructs a permutation of $\mathcal{A}$, denoted by $\pi : \{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$, and the sets $\mathcal{C}_j = \{a_{\pi((j-1)\ell+1)}, \ldots, a_{\pi(j\ell)}\}$ for $j = 1, \ldots, m$ such that for all $j \in \mathcal{S}$ $\mathcal{C}_j \subset \mathcal{A} \cap \mathcal{B}$. He announces π to Alice.

3. Alice groups her stored n bits $r^{\mathcal{A}}$ into blocks $r^{\mathcal{C}_1}, \ldots, r^{\mathcal{C}_m}$ as announced by Bob. Then Alice chooses m hash functions $g_1, \ldots, g_m$ independently and uniformly at random from $\mathcal{G}$ and announces them to Bob. She applies them to the blocks $r^{\mathcal{C}_1}, \ldots, r^{\mathcal{C}_m}$ and obtains the bits $t_1, \ldots, t_m$, where $t_j = g_j(r^{\mathcal{C}_j})$.

4. Bob computes $y_j = g_j(r^{\mathcal{C}_j})$ for $j \in \mathcal{S}$. He also computes the binary encoding $s = \sigma(\mathcal{S})$ of $\mathcal{S}$ of length $M = \lceil \lg \binom{m}{k} \rceil = \lceil \lg \binom{n^{1-\alpha}}{n^{\beta}} \rceil \leq n^{1-\alpha} h(n^{\alpha+\beta-1})$.

5. Alice and Bob engage in the interactive hashing of s into a bit string w of length $m - 1$, as described in Section 3.2. Alice computes the two sets $\mathcal{U}_0, \mathcal{U}_1 \subset \{1, \ldots, m\}$ such that $\sigma(\mathcal{U}_0)$ and $\sigma(\mathcal{U}_1)$ both hash to w and $\mathcal{U}_0 < \mathcal{U}_1$ according to some fixed order. If this is not possible because one of the strings that hashes to w is not a valid encoding of a subset, Bob aborts.

6. Bob also knows $\mathcal{U}_0, \mathcal{U}_1$. He chooses the bit c' such that $\mathcal{U}_{c' \oplus c} = \mathcal{S}$ and sends c' to Alice. Alice computes $z_0 = b_0 \oplus \left(\bigoplus_{j \in \mathcal{U}_{c'}} t_j\right)$ and $z_1 = b_1 \oplus \left(\bigoplus_{j \in \mathcal{U}_{c' \oplus 1}} t_j\right)$.

7. Bob recovers $b_c = z_c \oplus \left(\bigoplus_{j \in \mathcal{S}} y_j\right)$.

The descriptions of $\mathcal{A}$ and $\mathcal{B}$ have size in $O(n \lg n)$, which could be reduced by choosing the sets with pairwise independent distribution. The expected number of common indices is $k\ell = n^2/N = n^{\alpha+\beta}$. By storing a few extra bits, Alice and Bob can ensure that the overlap is $k\ell$ bits except with small probability. As mentioned earlier, this version of the protocol requires both party to memorize a $\theta(M^2)$ size matrix in order to calculate the values of $\mathcal{U}_0, \mathcal{U}_1$ in step 6.

It is easy to see that the protocol is complete and succeeds with probability at least $\frac{1}{2}$ if Alice and Bob are honest since more than half the string of length M are valid subset encoding (aborts can occur in step 2 if the overlap is not large enough and in step 5 if hashing yields an invalid subset encoding.) In order to prevent Bob from cheating by inducing aborts too often (e.g. while waiting until the overlap of $\mathcal{A}$ and $\mathcal{B}$ is much larger than expected), Alice will only cooperate for at most n' repetitions of the protocol for some $n' \ll n$. If Bob aborts more often, she concludes that he must be cheating, since the abort probability of an honest Bob is at most $O(2^{-n'})$.

5 Security Proof

We note first that if the protocol does not abort, then Alice obtains no information about which one of $\mathcal{U}_0, \mathcal{U}_1$ corresponds to Bob's set $\mathcal{S}$ and therefore the protocol is secure for Bob. If the protocol aborts, then no information depending on Alice's inputs b_0, b_1 or Bob's input c has been disclosed yet and therefore, we need not worry further about aborts.

Thus, the security of the protocol is established by the next theorem.

Theorem 7. *Suppose malicious Bob's memory is not more than γN bits for some $\gamma < 1$. Then, for sufficiently large n, the probability that Bob learns information about both bits b_0, b_1 can be made inverse polynomially small.*

During the random broadcast, a malicious Bob can compute any probabilistic function from $\{0,1\}^N$ to $\mathcal{V}$ with output V such that $\lg \#\mathcal{V} \leq \gamma N$. Let the random variable $R^{[N]}$

correspond to $r^{[N]}$ and let $X^{[m]} = X_1, \ldots, X_m$ correspond to the distribution of the blocks $x_1, \ldots, x_m$ in Alice's subset conditioned on Bob's knowledge $V = v$, or

$$P_{X^{[m]}}(x_i) = P_{R^{\mathcal{A}}|V=v}(r^{C_i}).$$

for $i = 1, \ldots, m$. ($X^{[m]}$ is a random variable over n-bit strings.)

The proof consists of three major steps. First, a lower bound on Bob's min-entropy about $R^{\mathcal{A}}|V = v$, the bits stored by Alice, is obtained (Lemma 8). Second, the hashing of blocks to bits is examined in Lemma 9 and it is shown that Bob can know at most about $(1-\rho)N$ of the bits. The third step (proof of Theorem 7) uses the analysis of interactive hashing from Lemma 6 to show that a malicious Bob cannot learn information about both bits.

Lemma 8. *Let $\epsilon_1 > 0$, let $\delta = (1 - \gamma - \frac{1}{N}\lg\frac{1}{\epsilon_1})$, let $\rho = c\delta/\lg\delta^{-1}$ for some $c > 0$, and let $\epsilon_2 = 3/\sqrt{\rho n}$. Then, except with probability $\epsilon_1 + \epsilon_2$, the distribution of $X^{[m]}$ is ϵ_2-close to a random variable with min-entropy at least ρn.*

Proof. Because $R^{[N]}$ is assumed to be uniformly distributed, it has min-entropy $H_\infty(R^{[N]}) = N$. Lemma 1 roughly shows that the min-entropy of $R^{[N]}$ given V is at least $(1-\gamma)N$ with probability $1-\epsilon_1$. More precisely, for any $\epsilon_1 > 0$, the particular value v that V takes on satisfies $H_\infty(R^{[N]}|V = v) \geq (1-\gamma)N - \lg\frac{1}{\epsilon_1}$, except with probability at most ϵ_1.

We now invoke Theorem 3 and obtain that the distribution of $X^{[m]} = X_1, \ldots, X_m$ (a random variable on n-bit strings) is ϵ_2-close to a distribution with min-entropy ρn except with probability ϵ_2. □

The next lemma shows that Bob lacks knowledge of at least about ρm bits from $T_1, \ldots, T_m$ with high probability. It involves a *spoiling knowledge* argument that is often used in connection with privacy amplification [BBCM95, Cac97a]: Suppose side information is made available to Bob by an oracle. The side information is tailored for Bob's distribution and serves the purpose of increasing his entropy and to obtain better results. Note that the oracle giving spoiling knowledge is used only as a proof technique and not for carrying out privacy amplification.

Lemma 9. *Let $\epsilon_4, \epsilon_5 > 0$ and suppose $X^{[m]}$ has min-entropy at least ρn. There is a subset $\mathcal{Q} \subseteq \{1, \ldots, m\}$ of cardinality $q = (\rho n - m(\lg n + 2) - \lg\frac{1}{\epsilon_4} - 2m\lg\frac{1}{\epsilon_5})/\ell$ such that Bob's distribution of $T^{\mathcal{Q}}$, conditioned on particular values v, $f_1, \ldots, f_m$, and x_j for $j \notin \mathcal{Q}$, is ϵ_6-close to the uniform distribution over bit strings of length q, where $\epsilon_6 = m2^{-2\ell} + \epsilon_4 + \epsilon_5 + \sqrt{2q\epsilon_5}$.*

Proof. The main part of the proof is to construct spoiling knowledge such that min-entropies of the blocks $X_1, \ldots, X_m$ add up and then applying privacy amplification for hashing the blocks to bits $T_1, \ldots, T_m$.

Suppose side information $u_1, \ldots, u_m$ with $u_j \in \{0, \ldots, 2j\ell\}$ for $j = 1, \ldots, m$ is made available to Bob. Let the random variable $U^{[m]}$ correspond to the distribution of $u^{[m]}$. It is defined for $j = 1, \ldots, m$ as $U_j = \lambda_j(X^{[j]})$, where

$$\lambda_j(x^{[j]}) \;=\; \begin{cases} 2j\ell & \text{if } P_{X^{[j]}}(x^{[j]}) \leq 2^{-2j\ell} \\ \lfloor -\lg P_{X^{[j]}}(x^{[j]}) \rfloor & \text{otherwise.} \end{cases}$$

(Side information U_j of this type has also been called *log-partition spoiling knowledge* [Cac97a]). U_j partitions the values of $X^{[j]}$ into sets of approximately equal probability under $P_{X^{[j]}|U_j=u_j}$. For all u_j except $u_j = 2j\ell$, the values of the probability distributions $P_{X^{[j]}|U_j=u_j}$ differ by less than a factor of two and we have

$$\frac{1}{2}\max_{x^{[j]}} P_{X^{[j]}|U_j=u_j}(x^{[j]}) \;\leq\; \min_{x^{[j]}} P_{X^{[j]}|U_j=u_j}(x^{[j]}). \quad (5)$$

The probability that there exists a j s.t. $U_j = 2j\ell$ is no more than $\epsilon_3 = m2^{-2\ell}$ and we assume $U_j \neq 2j\ell$ for the rest of the proof.

The size of $U^{[m]}$ (in bits) is less than $m\lg(2m\ell) = m\lg(2n)$. Therefore, $U^{[m]}$ satisfies

$$\begin{aligned} &H_\infty(X^{[m]}|U^{[m]} = u^{[m]}) \\ &\qquad \geq H_\infty(X^{[m]}) - m(\lg n + 1) - \lg\frac{1}{\epsilon_4} \end{aligned} \quad (6)$$

except with probability ϵ_4 by Lemma 1. We assume that (6) holds in the remainder of the proof.

Claim. *For all $x^{[1]}, \ldots, x^{[m-1]}$, we have*

$$\begin{aligned} &\sum_{j=1}^{m} H_\infty(X_j|U^{[m]} = u^{[m]}, X^{[j-1]} = x^{[j-1]}) \\ &\qquad \geq \rho n - m(\lg n + 2) - \lg\frac{1}{\epsilon_4}. \end{aligned} \quad (7)$$

Note that this implies Bob's min-entropies of at least

$$q \;=\; \big(\rho n - m(\lg n + 2) - \lg\frac{1}{\epsilon_4} - 2m\lg\frac{1}{\epsilon_5}\big)/\ell \quad (8)$$

blocks from $X_1, \ldots, X_m$ exceed $2\lg\frac{1}{\epsilon_5}$, conditioned on any particular values of the other blocks. (There are m blocks for which the sum of the min-entropies is bounded from below in (7), and the min-entropy of each block is at most ℓ.)

Proof of the claim: First note that

$$\begin{aligned} &H_\infty(X^{[m]}|U^{[m]} = u^{[m]}) - m \\ &\qquad \geq \rho n - m(\lg n + 2) - \lg\frac{1}{\epsilon_4} \end{aligned}$$

from (6) and therefore, proving the claim reduces to

$$\sum_{j=1}^{m} H_\infty(X_j|U^{[m]}=u^{[m]},X^{[j-1]}=x^{[j-1]}) \geq H_\infty(X^{[m]}|U^{[m]}=u^{[m]})-m. \quad (9)$$

We do this by induction on $j=0,\ldots,m-1$ using the following induction hypothesis: For all $x^{[m-j+1]},\ldots,x^{[m]}$, it holds

$$H_\infty(X^{[m-j]}|U^{[m]}=u^{[m]}) + \sum_{j'=m-j+1}^{m} H_\infty(X_{j'}|X^{[m-j'+1]}=x^{[m-j'+1]},U^{[m]}=u^{[m]}) \geq H_\infty(X^{[m]}|U^{[m]}=u^{[m]})-j. \quad (10)$$

Clearly (10) holds for $j=0$. Suppose that (10) holds for some $j\leq 0$. Let $P_j = P_{X_{m-j}|X^{[m-j-1]}=x^{[m-j-1]},U^{[m]}=u^{[m]}}(x_{m-j})$. Then we have for all $x^{[m-j-1]}$ and all x_{m-j} that

$$\begin{aligned}
&\max_{\tilde{x}^{[m-j]}} P_{X^{[m-j]}|U^{[m]}=u^{[m]}}(\tilde{x}^{[m-j]})\\
&\geq P_{X^{[m-j-1]}|U^{[m]}=u^{[m]}}(x^{[m-j-1]})\cdot P_j\\
&\geq \min_{\tilde{x}^{[m-j-1]}} P_{X^{[m-j-1]}|U^{[m]}=u^{[m]}}(\tilde{x}^{[m-j-1]})\cdot P_j\\
&\geq \frac{1}{2}\max_{\tilde{x}^{[m-j-1]}} P_{X^{[m-j-1]}|U^{[m]}=u^{[m]}}(\tilde{x}^{[m-j-1]})\cdot P_j
\end{aligned}$$

by the property (5) of the side information $U^{[m]}$. Since this holds for all x_{m-j} (and thus for the max), we obtain

$$\begin{aligned}
&H_\infty(X_{m-j}|X^{[m-j-1]}=x^{[m-j-1]},U^{[m]}=u^{[m]})\\
&\quad + H_\infty(X^{[m-j-1]}|U^{[m]}=u^{[m]})\\
&\quad\quad \geq H_\infty(X^{[m-j]}|U^{[m]}=u^{[m]})-1 \quad (11)
\end{aligned}$$

for all $x^{[m-j-1]}$. It follows now from (10) and from (11) that for all $x^{[m-j]},\ldots,x^{[m]}$,

$$H_\infty(X^{[m-j-1]}|U^{[m]}=u^{[m]}) + \sum_{j'=m-j}^{m} H_\infty(X_{j'}|X^{[m-j']}=x^{[m-j']},U^{[m]}=u^{[m]}) \geq H_\infty(X^{[m]}|U^{[m]}=u^{[m]})-j-1 \quad (12)$$

This establishes the induction step and thus 9 and the claim (7). □

For second step in the proof of Lemma 9, we apply Theorem 2 (privacy amplification). Let $\mathcal{Q}\subseteq\{1,\ldots,m\}$ be a set of q indices j such that, for all $j\in\mathcal{Q}$,

$$H_\infty(X_j|U^{[m]}=u^{[m]},X^{[j-1]}=x^{[j-1]}) \geq 2\lg\frac{1}{\epsilon_5}.$$

Such a set exists according to the claim (7). Using Theorem 2, we obtain for $j\in\mathcal{Q}$,

$$H(T_j|F_j=f_j,U^{[m]}=u^{[m]},X^{[j-1]}=x^{[j-1]}) \geq 1-2{\epsilon_5}^2/\ln 2,$$

where F_j for $j=1,\ldots,m$ denotes the random variable corresponding to the choice of the hash function f_j with uniform distribution. Let $q_{\max}$ be the largest element of $\mathcal{Q}$ and let $\bar{\mathcal{Q}}=\{1,\ldots,q_{\max}\}\setminus\mathcal{Q}$. By summing up the entropies, we have

$$H(T^{\mathcal{Q}}|F^{\mathcal{Q}},U^{[m]}=u^{[m]},X^{\bar{\mathcal{Q}}}=x^{\bar{\mathcal{Q}}}) \geq q-2q{\epsilon_5}^2/\ln 2.$$

Thus, except with probability ϵ_5, $F^{\mathcal{Q}}$ takes on a value $f^{\mathcal{Q}}$ such that

$$H(T^{\mathcal{Q}}|F^{\mathcal{Q}}=f^{\mathcal{Q}},U^{[m]}=u^{[m]},X^{\bar{\mathcal{Q}}}=x^{\bar{\mathcal{Q}}}) \geq q-2q\epsilon_5/\ln 2.$$

In this case, it follows from the standard inequality $\lg\#\mathcal{X}-H(X)\geq\frac{1}{\ln 2}{\|P_X-P_U\|_v}^2$ that

$$\|P_{T^{\mathcal{Q}}|F^{\mathcal{Q}}=f^{\mathcal{Q}},U^{[m]}=u^{[m]},X^{\bar{\mathcal{Q}}}=x^{\bar{\mathcal{Q}}}}-P_U\|_v \leq \sqrt{2q\epsilon_5},$$

where P_U denotes the uniform distribution over q bits. Accounting for all the cases excluded above, it follows

$$\|P_{T^{\mathcal{Q}}}-P_U\|_v \leq \epsilon_3+\epsilon_4+\epsilon_5+\sqrt{2q\epsilon_5}.$$

ϵ_3 and ϵ_4 are used in for spoiling knowledge and ϵ_5 is needed to remove the expectation from the conditional entropy of $T^{\mathcal{Q}}$. □

Proof of Theorem 7. Let $\mu>0$ be a small constant. Then, for all sufficiently large n, we have $q\geq(\rho-\mu)m$ from Lemma 9.

For the analysis of interactive hashing in step 5, we will use Lemma 6. There are $\binom{m}{k}=\binom{n^{1-\alpha}}{n^\beta}$ subsets and inputs for Bob in total, thus $M=\lg\binom{m}{k}$ for Lemma 6. Suppose Bob lacks knowledge about at least q bits from $T_1,\ldots,T_m$, i.e., he has complete knowledge about not more than $\binom{m-q}{k}\leq\binom{\zeta n^{1-\alpha}}{n^\beta}$ of the subsets, corresponding to the set $\mathcal{S}$ of Lemma 6, where $\zeta=1-\rho+\mu$.

In order to apply the lemma setting $r=\lg M$ we need to make sure that $\nu=\frac{1}{M}\lg\binom{m-q}{k}<1-\frac{8\lg M+4}{M}$, which is equivalent to

$$\lg\binom{n^{1-\alpha}}{n^\beta}-\lg\binom{\zeta n^{1-\alpha}}{n^\beta}-8\lg\lg\binom{n^{1-\alpha}}{n^\beta} > 4.$$

This can be satisfied by choosing n sufficiently large, since ζ is a constant smaller than 1. It follows that Bob has probability not more than $\epsilon_7=1/M=\left(\lg\binom{n^{1-\alpha}}{n^\beta}\right)^{-1}\approx n^{\alpha-1}/h(n^{\alpha+\beta-1})$ of knowing all bits of both sets and therefore of recovering both bits b_0,b_1.

The overall failure probability is at most $\epsilon_1 + 2\epsilon_2 + \epsilon_6 + \epsilon_7$, where ϵ_1, ϵ_2 are from Lemma 8 and ϵ_6, ϵ_7 are from Lemma 9. More precisely, $\epsilon_1, \epsilon_4, \epsilon_5$ are parameters fixed above and

1. $\epsilon_2 = 3/\sqrt{\rho n}$,
2. $\epsilon_3 = n^{(1-\alpha)} 2^{-2n^\alpha}$,
3. $\epsilon_6 = n^{(1-\alpha)} 2^{-2n^\alpha} + \epsilon_4 + \epsilon_5 + \sqrt{2q\epsilon_5}$,
4. $\epsilon_7 = \lg \binom{n^{1-\alpha}}{n^\beta}^{-1} \approx n^{\alpha-1}/h(n^{\alpha+\beta-1})$.

□

6 Discussion

The error probability of the security proof guaranteed by Theorem 7 is inverse polynomial in n, which may not be enough for some applications (even if n is generally large). However, by repeating the protocol l times the error can be reduced to an exponentially small quantity. Alice selects $2l$ random bits $b_1^0, \ldots, b_l^0$ and $b_1^1, \ldots, b_l^1$ such that $b_0 = \bigoplus_{j=1}^l b_j^0$ and $b_1 = \bigoplus_{j=1}^l b_j^1$ and they perform $\binom{2}{1}$-OT$(b_j^0, b_j^1)(c)$ for $j = 1, \ldots, l$. It is easy to see that now the probability that a malicious Bob obtains any information about $b_{c\oplus 1}$ is $O(2^{-l})$.

In our construction, both parties need $\theta(n^{2-2\alpha})$ memory size if they are honest and the security can be guaranteed if Bob has not more than $\gamma n^{2-\alpha-\beta}$ memory size for some small $\beta > 0$ and $\gamma < 1$, typically. It is an interesting open problem whether this difference can be enlarged. For example, in the cryptosystem by Cachin and Maurer based on memory bounds [CM97], the security margin is about $O(n)$ vs. n^2 for the public key agreement protocol. We believe that this should also be achievable for oblivious transfer.

Acknowledgment

We are greatful to Adam Smith and Alain Tapp for helpful discussions.

References

[AD97] Miklós Ajtai and Cynthia Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, Proc. 29th Annual ACM Symposium on Theory of Computing (STOC), 1997, pp. 284–293.

[AF94] Yonatan Aumann and Uriel Feige, *One message proof systems with known space verifiers*, Advances in Cryptology: CRYPTO '93 (Douglas R. Stinson, ed.), Lecture Notes in Computer Science, vol. 773, Springer, 1994, pp. 85–99.

[BB88] G. Brassard and P. Bratley, *Algorithmics: Theory and practice*, Prentice-Hall, Englewoods Cliffs, NJ, 1988.

[BBCM95] Charles H. Bennett, Gilles Brassard, Claude Crépeau, and Ueli M. Maurer, *Generalized privacy amplification*, IEEE Transactions on Information Theory **41** (1995), no. 6, 1915–1923.

[BBCS92] C.H. Bennett, G. Brassard, C. Crépeau, and M.-H. Skubiszewska, *Practical quantum oblivious transfer protocols*, Advances in Cryptology: Proceedings of Crypto '91, Lecture Notes in Computer Science, vol. 576, Springer-Verlag, 1992, pp. 351–366.

[BBR86] Charles H. Bennett, Gilles Brassard, and Jean-Marc Robert, *How to reduce your enemy's information*, Advances in Cryptology: CRYPTO '85 (Hugh C. Williams, ed.), Lecture Notes in Computer Science, vol. 218, Springer, 1986, pp. 468–476.

[BBR88] C. H. Bennett, G. Brassard, and J.–M. Robert, *Privacy amplification by public discussion*, SIAM J. Computing **17** (1988), no. 2, 210–229.

[BM90] Mihir Bellare and Silvio Micali, *Non-interactive oblivious transfer and applications*, Advances in Cryptology: CRYPTO '89 (Gilles Brassard, ed.), Lecture Notes in Computer Science, vol. 435, Springer, 1990, pp. 547–557.

[Cac97a] Christian Cachin, *Entropy measures and unconditional security in cryptography*, ETH Series in Information Security and Cryptography, vol. 1, Hartung-Gorre Verlag, Konstanz, Germany, 1997, ISBN 3-89649-185-7 (Reprint of Ph.D. dissertation No. 12187, ETH Zürich).

[Cac97b] Christian Cachin, *Smooth entropy and Rényi entropy*, Advances in Cryptology: EUROCRYPT '97 (Walter Fumy, ed.), Lecture Notes in Computer Science, vol. 1233, Springer-Verlag, 1997, pp. 193–208.

[CK88] Claude Crépeau and Joe Kilian, *Achieving oblivious transfer using weakened security assumptions*, Proc. 29th IEEE Symposium on

Foundations of Computer Science (FOCS), 1988.

[CM97] Christian Cachin and Ueli Maurer, *Unconditional security against memory-bounded adversaries*, Advances in Cryptology: CRYPTO '97 (Burt Kaliski, ed.), Lecture Notes in Computer Science, vol. 1294, Springer-Verlag, 1997, pp. 292–306.

[Cov73] T.M. Cover, *Enumerative source encoding*, IEEE Transactions on Information Theory **19** (1973), no. 1, 73–77.

[Cré84] C. Crépeau, *Quantum oblivious transfer*, Journal of Modern Optics **41** (1984), no. 12, 2445–2454.

[Cré88] Claude Crépeau, *Equivalence between two flavours of oblivious transfer*, Advances in Cryptology: CRYPTO '87 (Carl Pomerance, ed.), Lecture Notes in Computer Science, vol. 293, Springer, 1988, pp. 350–354.

[Cré97] Claude Crépeau, *Efficient cryptographic protocols based on noisy channels*, Advances in Cryptology: EUROCRYPT '97 (Walter Fumy, ed.), Lecture Notes in Computer Science, vol. 1233, Springer, 1997, pp. 306–317.

[CS91] C. Crépeau and M. Sántha, *On the reversibility of oblivious transfer*, Advances in Cryptology: Proceedings of Eurocrypt '91, Lecture Notes in Computer Science, vol. 547, Springer-Verlag, 1991, pp. 106–113.

[CS95] C. Crépeau and L. Salvail, *Quantum oblivious mutual identification*, Advances in Cryptology: Proceedings of Eurocrypt '95, Lecture Notes in Computer Science, vol. 921, Springer-Verlag, 1995, pp. 133–146.

[CvdGT95] C. Crépeau, J. van de Graaf, and A. Tapp., *Committed oblivious transfer and private multi-party computations*, Advances in Cryptology: Proceedings of Crypto '95, Lecture Notes in Computer Science, vol. 963, Springer-Verlag, 1995, pp. 110–123.

[DPY92] Alfredo De Santis, Giuseppe Persiano, and Moti Yung, *One-message statistical zero-knowledge proofs with space-bounded verifier*, Proc. 19th ICALP, Lecture Notes in Computer Science, vol. 623, Springer, 1992, pp. 28–40.

[EGL83] Shimon Even, Oded Goldreich, and Abraham Lempel, *A randomized protocol for signing contracts*, Proc. CRYPTO '82 (R. L. Rivest, A. Sherman, and D. Chaum, eds.), Plenum Press, 1983, pp. 205–210.

[GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson, *How to play any mental game or a completeness theorem for protocols with honest majority*, Proc. 19th Annual ACM Symposium on Theory of Computing (STOC), 1987, pp. 218–229.

[GV88] Oded Goldreich and Ronen Vainish, *How to solve any protocol problem – an efficiency improvement*, Advances in Cryptology: CRYPTO '87 (Carl Pomerance, ed.), Lecture Notes in Computer Science, vol. 293, Springer, 1988, pp. 73–86.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby, *Pseudo-random generation from one-way functions*, Proc. 21st Annual ACM Symposium on Theory of Computing (STOC), 1989, pp. 12–24.

[IR89] Russell Impagliazzo and Steven Rudich, *Limits on the provable consequences of one-way permutations*, Proc. 21st Annual ACM Symposium on Theory of Computing (STOC), 1989, pp. 186–208.

[Kil88a] Joe Kilian, *Founding cryptography on oblivious transfer*, Proc. 20th Annual ACM Symposium on Theory of Computing (STOC), 1988, pp. 20–31.

[Kil88b] Joe Kilian, *Zero-knowledge with log-space verifiers*, Proc. 29th IEEE Symposium on Foundations of Computer Science (FOCS), 1988, pp. 25–35.

[NOVY98] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung, *Perfect zero-knowledge arguments for* NP *using any one-way function*, Journal of Cryptology **11** (1998), no. 2, 87–108, Preliminary version presented at CRYPTO '92.

[NS98] P. Nguyen and J. Stern, *Cryptanalysis of the ajtai-dwork cryptosystem*, Advances in Cryptology: Proceedings of Crypto '98, Lecture Notes in Computer Science, Springer-Verlag, 1998.

[Rab81] Michael O. Rabin, *How to exchange secrets by oblivious transfer*, Tech. Report TR-81, Harvard, 1981.

[Sho94] Peter W. Shor, *Algorithms for quantum computation: Discrete log and factoring*, Proc. 35th IEEE Symposium on Foundations of Computer Science (FOCS), 1994, pp. 124–134.

[Wie70] Stephen Wiesner, *Conjugate coding*, Reprinted in SIGACT News, vol. 15, no. 1, 1983, original manuscript written ca. 1970.

[Yao86] Andrew C.-C. Yao, *How to generate and exchange secrets*, Proc. 27th IEEE Symposium on Foundations of Computer Science (FOCS), 1986, pp. 162–167.

[Zuc96] David Zuckerman, *Simulating BPP using a general weak random source*, Algorithmica **16** (1996), 367–391, Preliminary version presented at 32nd FOCS (1991).