J. Cryptology (2003) 16: 219–237
DOI: 10.1007/s00145-002-0146-4

Journal of
CRYPTOLOGY

Oblivious Transfers and Privacy Amplification*

Gilles Brassard
Département IRO, Université de Montréal,
C.P. 6128, succursale centre-ville, Montréal,
Québec, Canada H3C 3J7
brassard@iro.umontreal.ca

Claude Crépeau
School of Computer Science, McGill University,
room 318, 3480 rue University, Montréal,
Québec, Canada H3A 2A7
crepeau@cs.mcgill.ca

Stefan Wolf
Département IRO, Université de Montréal,
C.P. 6128, succursale centre-ville, Montréal,
Québec, Canada H3C 3J7
wolf@iro.umontreal.ca

Communicated by Cynthia Dwork

Received April 2001 and revised July 2002
Online publication 5 March 2003

Abstract. Oblivious transfer (OT) is an important primitive in cryptography. In chosen one-out-of-two string OT, a sender offers two strings, one of which the other party, called the receiver, can choose to read, not learning any information about the other string. The sender on the other hand does not obtain any information about the receiver's choice. We consider the problem of reducing this primitive to OT for single bits. Previous attempts to doing this were based on self-intersecting codes. We present a new technique for the same task, based on so-called privacy amplification. It is shown that our method has two important advantages over the previous approaches. First, it is more efficient in terms of the number of required realizations of bit OT, and second, the technique even allows for reducing string OT to (apparently) much weaker primitives. An example of such a primitive is universal OT, where the receiver can adaptively choose what type of information he wants to obtain about the two bits sent by the sender subject to the only constraint that some, possibly very small, uncertainty must remain about the pair of bits.

Key words. Information-theoretic security, Oblivious transfer, Universal oblivious transfer, Reduction among information-theoretic primitives, Privacy amplification.

[*] Earlier versions of this work have appeared as [4] and [28]. This work was supported in part by Canada's CRC, NSERC, and Québec's FCAR.

1. Introduction

One-out-of-two (k-bit-) string oblivious transfer, denoted $\binom{2}{1}$-OT^k, is a primitive that originates in [27] (under the name of "multiplexing"), a paper that marked the birth of quantum cryptography. According to this primitive, one party $\mathcal{A}$ owns two secret k-bit strings w_0 and w_1, and another party $\mathcal{B}$ wants to learn w_c for a secret bit c of his choice. $\mathcal{A}$ is willing to collaborate provided that $\mathcal{B}$ does not learn any information about $w_{\bar{c}}$, but $\mathcal{B}$ will only participate if $\mathcal{A}$ cannot obtain information about c. Independently from [27] but inspired by [25], a natural restriction of this primitive was introduced subsequently in [19] with applications to contract-signing protocols: one-out-of-two bit oblivious transfer, denoted $\binom{2}{1}$-OT, concerns the case $k = 1$ in which w_0 and w_1 are single-bit secrets, generally called b_0 and b_1 in that case.

Techniques were introduced in [5] and refined in [18] and [7] to reduce $\binom{2}{1}$-OT^k to $\binom{2}{1}$-OT: several constructions were given to achieve $\binom{2}{1}$-OT^k based on the assumption of the availability of a protocol for the simpler $\binom{2}{1}$-OT. The fact that $\binom{2}{1}$-OT^k *can in principle* be reduced to $\binom{2}{1}$-OT is not surprising because $\binom{2}{1}$-OT is sufficient to implement *any* two-party computation, as has been shown by a number of authors [21], [13], [16]. Our interest in *direct* reductions is their far greater efficiency. With the exception of [17], all previous direct reductions that we are aware of [5], [18], [7] are based on a notion called *zigzag functions*, whose construction is reduced to finding particular types of error-correcting codes called *self-intersecting codes*. In a nutshell, this approach consists of selecting once and for all a suitable function f from[1] $\mathcal{F}_2^n$ to $\mathcal{F}_2^k$ for n as small as possible ($n > k$), so that if x_0 is a random pre-image of w_0 and x_1 is a random pre-image of w_1, and if $\mathcal{B}$ is given to choose via $\binom{2}{1}$-OT to see the ith bit of either x_0 or x_1, $1 \le i \le n$, then no information can be inferred on at least one of w_0 or w_1. This approach has led to various reductions with expansion factors β ranging from 4.8188 to 18: that is various polynomial-time constructible methods using $n = \beta k$ instances of $\binom{2}{1}$-OT to perform one $\binom{2}{1}$-OT^k on k-bit strings. Komlós proved that this approach cannot yield an expansion factor β that is asymptotically better than 3.5277 as reported in [10]. It was proven in 1997 by Stinson that the same bound applies even to non-linear zigzags [26]. Consult Section 9 for a discussion of the tight connection between our new protocol and the self-intersecting codes approach.

This current paper exploits a new approach to the problem using *privacy amplification*, a notion first introduced in the context of key exchange protocols [3]. The new approach allows for a solution requiring only $2(k + s + 1)$ instances of $\binom{2}{1}$-OT (where s is a security parameter) to perform one $\binom{2}{1}$-OT^k, and it can be extended to a whole range of generalizations of $\binom{2}{1}$-OT, including an extremely weak variant of bit OT, that could not be used with the reductions based on zigzag functions. Reductions related to ours were presented in [8].

An application of the simplest of our generalizations is also considered: $\binom{2}{1}$-OT^k from $\mathcal{A}$ to $\mathcal{B}$ can be reduced to $\binom{2}{1}$-OT in the other direction (from $\mathcal{B}$ to $\mathcal{A}$) by only doubling the cost of reducing to $\binom{2}{1}$-OT from $\mathcal{A}$ to $\mathcal{B}$. This improves on an earlier result of [17] by a factor of six.

[1] Throughout this paper, $\mathcal{F}_2$ denotes the field $GF(2)$ with two elements.

It is important to note that throughout this paper we are concerned with *information-theoretic* reductions between the described primitives. String OT offering only *computational* security can be realized directly and efficiently under certain *computational assumptions* (see for example [24] and references within). An information-theoretic reduction of string OT to bit OT for instance shows that in *every* security model, the two primitives are equivalent: bit OT offering *any kind* of security can be used to realize string OT with the same kind of security.

2. Privacy Amplification versus Other Methods

We describe the main idea of our new construction. Assume $\mathcal{A}$ knows a random n-bit string x about which $\mathcal{B}$ has partial information. *Privacy amplification* is a technique invented in [3] and refined in [2] that allows $\mathcal{A}$ to shrink x to a shorter string w about which $\mathcal{B}$ has an arbitrarily small amount of information even if he knows the recipe used by $\mathcal{A}$ to transform x into w. Intuitively, this can be used to implement $\binom{2}{1}$-OT$^k(w_0, w_1)(c)$ from $\binom{2}{1}$-OT because $\mathcal{A}$ can offer $\mathcal{B}$ to read one of two random strings x_0 or x_1 by a simple sequence of $\binom{2}{1}$-OT$(x_0^i, x_1^i)(c_i)$. Subsequently, $\mathcal{A}$ tells $\mathcal{B}$ how to transform x_0 into w_0 and x_1 into w_1 by way of privacy amplification. An honest $\mathcal{B}$ who accessed all the bits of x_c can reconstruct w_c from this information. However, a dishonest $\tilde{\mathcal{B}}$ who tried to access some of the bits of x_0 and some of the bits of x_1 will not have enough information on at least one of them to infer any information on the corresponding w_i or even joint information on both w_0 and w_1.

An important fact about the method based on zigzag functions considered in earlier papers is that there is no way for $\mathcal{B}$ to learn information about both w_0 and w_1 even though the zigzag function is known before he gets to choose which bits of x_0 and x_1 to obtain through the $\binom{2}{1}$-OT instances. In the new approach based on privacy amplification, $\mathcal{A}$ reveals the function to $\mathcal{B}$ *after* the necessary $\binom{2}{1}$-OTs have been performed. This allows for a protocol that is simpler, more general and more efficient, but at the cost of a vanishingly small probability of failure. (Throughout the paper, *failure* denotes the event that a dishonest $\mathcal{B}$ can collect more information than he is supposed to. In all the protocols presented an honest receiver $\mathcal{B}$ obtains no information at all about the string he did not choose.) A drawback of this approach is that a new function must be generated and transmitted at each run of the protocol.

Table 1 compares the efficiency of the earlier methods to that of privacy amplification. The column "expansion factor" gives a number β such that a $\binom{2}{1}$-OT^k can be achieved with βk instances of $\binom{2}{1}$-OT, s is a security parameter, and $\varepsilon = s/k$ is arbitrarily small in the limit of large k. Thus we see that the privacy amplification method is preferable provided a probability of failure can be tolerated.

3. The New Protocol

Protocol 3.2 below realizes a randomized primitive $\binom{2}{1}$-ROT$^k(c) = \binom{2}{1}$-OT$^k(R_0, R_1)(c)$ similar to OT^k, where $\mathcal{A}$ transmits one-out-of-two uniformly distributed independent k-bit strings r_0, r_1 to $\mathcal{B}$ (R_0, R_1 are the corresponding random variables). These two random strings r_0, r_1 are then used as one-time pads to transfer the actual k-bit strings w_0, w_1.

Table 1. Efficiency of earlier methods and of privacy amplification.

Method	Expansion factor	Failure probability	Construction time
Monte Carlo Zigzag[a]	$4.8188 + \varepsilon$	2^{-s}	$O(k(k+s))$
Las Vegas Zigzag[b]	$9.6377 + \varepsilon$	0	$O(k^2)$
Zigzag à la Justesen[c]	18	0	$O(k^4)$
Zigzag à la Goppa[d]	6.4103	0	$O(k^{32})$
Privacy amplification	$2 + \varepsilon$	2^{-s}	$O(k(k+s))$

[a] Attributed to Cohen and Lempel in [7].
[b] Attributed to Kilian in [7].
[c] From [7].
[d] From [11] based on a method of [18].

Protocol 3.1. $(\binom{2}{1}\text{-OT}^k(w_0, w_1)(c))$

1. *$\mathcal{A}$ transfers a random $r_c \leftarrow \binom{2}{1}\text{-ROT}^k(c)$ to $\mathcal{B}$.*
2. *$\mathcal{A}$ sets $y_0 \leftarrow r_0 \oplus w_0$, $y_1 \leftarrow r_1 \oplus w_1$ and announces y_0, y_1 to $\mathcal{B}$.*
3. *$\mathcal{B}$ obtains $w_c \leftarrow r_c \oplus y_c$.*

Let s be a security parameter chosen by $\mathcal{A}$ and $\mathcal{B}$ so that they agree to tolerate a probability 2^{-s} of failure. Let γ be a constant to be determined later, and let $n = \gamma(k+s)$.

Privacy amplification is based on the general notion of universal classes of hash functions [9]. For sake of simplicity, we use a specific class of hash functions in our protocol to implement $\binom{2}{1}$-ROT^k from $\binom{2}{1}$-OT:

$$\{h \mid h(x) = Mx, \text{ for } M \text{ a } k \times n \text{ rank } k \text{ matrix over } \mathcal{F}_2\}.$$

Note however that our proofs are tailored for this specific class of functions and that a general result for any universal class of hash functions or similar objects such as *extractors* is left as an open problem.

Protocol 3.2. $(\binom{2}{1}\text{-ROT}^k(c))$

1. *$\mathcal{A}$ picks two random n-bit strings x_0 and x_1.*
2. **DO**$_{i=1}^{n}$ *$\mathcal{A}$ transfers $t^i \leftarrow \binom{2}{1}\text{-OT}(x_0^i, x_1^i)(c)$ to $\mathcal{B}$.*
3. *$\mathcal{A}$ picks two random $k \times n$ rank k matrices M_0 and M_1 over $\mathcal{F}_2$; she sets $r_0 \leftarrow M_0 x_0$, $r_1 \leftarrow M_1 x_1$ and announces M_0, M_1 to $\mathcal{B}$.*
4. *$\mathcal{B}$ obtains r_c by computing $M_c t$.*

In the following sections we will show that this protocol allows for reducing string OT to bit OT (Section 5) as well as to apparently much weaker primitives such as XOR-OT (Section 6), generalized OT (Section 7), and universal OT (Section 8). In all cases, we show security of Protocol 3.2 and conclude security of Protocol 3.1 by the properties of the one-time pad.

4. Information Theoretic Definition of Oblivious Transfers

A *protocol* is a multi-party synchronous program that describes for each party the computations to be performed or the messages to be sent to some other party at each point in time. The protocol terminates when no party has any message to send or information to compute. The protocols we describe in this paper all take place between two parties $\mathcal{A}$ and $\mathcal{B}$. We denote by $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ the *honest* programs to be executed by $\mathcal{A}$ and $\mathcal{B}$: honest parties behave according to $\bar{\mathcal{A}}$ and $\bar{\mathcal{B}}$ and no other program. In the following definitions of *correctness* and *privacy* we also consider alternative *dishonest* programs $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ executed by $\mathcal{A}$ or $\mathcal{B}$ in an effort to obtain illegal information from one another. The definitions specify the result of honest parties interacting together through a specific protocol as well as the possible information leakage of an honest party facing a dishonest party. We are not concerned with the situation where both parties may be dishonest as they can do anything they like in that case; we are only concerned with protecting an honest party against a dishonest party. At the end of each execution of a protocol, each party will issue an "accept" or "reject" verdict regarding their satisfaction with the behavior of the other party. Two honest parties should always issue "accept" verdicts at the end of their interactions. An honest party will issue a "reject" verdict at the end of a protocol if he received some message from the other party of improper format or some message not satisfying certain conditions specified by the protocol. We also implicitly assume certain time limits for each party to issue messages to each other: after a specified amount of time a party will give up interacting with the other party and issue a "reject" verdict.

As discussed in the Introduction, a $\binom{2}{1}$-OT is a cryptographic protocol for two participants that enables a sender $\mathcal{A}$ to transfer one of two bits b_0 or b_1 to a receiver $\mathcal{B}$ who chooses secretly which bit b_c he gets. This is done in an all-or-nothing fashion, which means that $\mathcal{B}$ cannot get partial information about b_0 and b_1 at the same time, however malicious or (computationally) powerful he is, and that $\mathcal{A}$ finds out nothing about the choice c of $\mathcal{B}$. Generalization of $\binom{2}{1}$-OT include $\binom{2}{1}$-OT^k, in which the bits b_0 and b_1 are replaced by k-bit strings w_0 and w_1, and $\binom{t}{1}$-OT^k, in which $\mathcal{A}$ has several k-bit strings $w_0, w_1, \dots, w_{t-1}$ from which $\mathcal{B}$ is given to choose one. The choice c is then from the set $T = \{0, 1, \dots, t-1\}$. Note that a simple reduction from $\binom{t}{1}$-OT^k to $2t$ calls of $\binom{2}{1}$-OT^k may be found in [5]. We thus focus solely on the latter for the rest of this paper.

Formally speaking, we describe a two-party protocol that satisfies the following constraints of *correctness* and *privacy*. These notions have been defined before for general protocols by Crépeau [14], Micali and Rogaway [23], and Beaver [1] using simulators. In this paper we use the language of information theory to express definitions similar to those introduced by Crépeau [15] and Brassard et al. [7].

Let $[P_0, P_1](a)(b)$ be the random variable (since P_0, P_1 may be probabilistic programs) that describes the outputs obtained by $\mathcal{A}$ and $\mathcal{B}$ when they execute together the programs P_0 and P_1 on respective inputs a and b. Similarly, let $[P_0, P_1]^*(a)(b)$ be the random variable that describes the total information (including not only messages received and issued by the parties but also the result of any local random sampling they may have performed) acquired during the execution of protocol $[P_0, P_1]$ on inputs a, b. Let $[P_0, P_1]_P(a)(b)$ and $[P_0, P_1]^*_P(a)(b)$ be the marginal random variables obtained by restricting the above to only one party P. The latter is often called the *view* of P [20]. In the following definition, the equality sign (=) means that the distributions on the left-hand side and the right-hand side are the same.

Definition 1 (Correctness). Protocol $[\bar{\mathcal{A}}, \bar{\mathcal{B}}]$ is *correct* for $\binom{2}{1}$-OT^k if

- $\forall w_0, w_1 \in \mathcal{F}_2^k, c \in \mathcal{F}_2$,

$$[\bar{\mathcal{A}},\bar{\mathcal{B}}](w_0, w_1)(c) = (\varepsilon, w_c), \tag{1}$$

- $\forall \tilde{\mathcal{A}} \, \exists \, \tilde{\mathcal{A}}'$ s.t. $\forall w_0, w_1 \in \mathcal{F}_2^k, c \in \mathcal{F}_2$,

$$([\tilde{\mathcal{A}}, \bar{\mathcal{B}}]_{\mathcal{B}}(w_0, w_1)(c), \mathcal{B} \text{ accepts}) = ((\tilde{\mathcal{A}}'(w_0, w_1))_c, \mathcal{B} \text{ accepts}). \tag{2}$$

Intuitively, condition (1) means that if the protocol is executed as described, it will accomplish the task it was designed for: $\mathcal{B}$ receives word w_c and $\mathcal{A}$ receives nothing. Condition (2) means that in situations in which $\mathcal{B}$ does not abort, $\mathcal{A}$ cannot induce a distribution on $\mathcal{B}$'s output using a dishonest $\tilde{\mathcal{A}}$ that she could not induce simply by changing the input words and then being honest (which she can always do without being detected).

Let (W_0, W_1) and C be the random variables taking values over $\mathcal{F}_2^{2k}$ and $\mathcal{F}_2$ (later denoted $RV(\mathcal{F}_2^{2k})$ and $RV(\mathcal{F}_2)$) that describe $\mathcal{A}$'s and $\mathcal{B}$'s inputs. We assume that both $\mathcal{A}$ and $\mathcal{B}$ are aware of the arbitrary joint probability distribution of these random variables $P_{W_0,W_1,C}$. A sample w_0, w_1, c is generated from that distribution and w_0, w_1 is provided as $\mathcal{A}$'s secret input while c is provided as $\mathcal{B}$'s secret input.

We assume that the reader is familiar with the notion of *entropy* $\mathbf{H}(X)$ of a random variable X. The mutual *information* of two random variables X, Y is given by $\mathbf{I}(X; Y) = \mathbf{H}(X) - \mathbf{H}(X \mid Y)$ and conditioned by a third random variable Z, $\mathbf{I}(X; Y \mid Z) = \mathbf{H}(X \mid Z) - \mathbf{H}(X \mid Y, Z)$.

Definition 2 (Privacy). Protocol $[\bar{\mathcal{A}}, \bar{\mathcal{B}}]$ is *private* for $\binom{2}{1}$-OT^k if $\forall (W_0, W_1) \in RV(\mathcal{F}_2^{2k}), C \in RV(\mathcal{F}_2)$,

- $\forall w_0, w_1 \in \mathcal{F}_2^k, \forall \tilde{\mathcal{A}}$,

$$\mathbf{I}(C; [\tilde{\mathcal{A}}, \bar{\mathcal{B}}]^*_{\mathcal{A}}(W_0, W_1)(C) \mid (W_0, W_1) = (w_0, w_1)) = 0, \tag{3}$$

- $\forall c \in \mathcal{F}_2, \forall \tilde{\mathcal{B}}, \exists \tilde{C} \in RV(\mathcal{F}_2)$ s.t.

$$\mathbf{I}(W_{\neg \tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^*_{\mathcal{B}}(W_0, W_1)(C) \mid W_{\tilde{C}}, C = c) = 0. \tag{4}$$

The above two conditions are designed to guarantee that each party is limited to the information he or she should get according to the honest task definition. Condition (3) means that $\tilde{\mathcal{A}}$ cannot acquire any information about C through the protocol. On the other hand, condition (4) means that $\tilde{\mathcal{B}}$ may acquire information about only one of W_0, W_1 through the protocol. In particular, no joint information about the two words may be obtained by the protocol. This is why our condition assumes that $\tilde{\mathcal{B}}$ is given one of the words. (We do not require that $\tilde{\mathcal{B}}$ be given W_C because there is no way to prevent him from obtaining any other $W_{\tilde{C}}$ through otherwise honest use of the protocol.)

Definition 3. A protocol for $\binom{2}{1}$-OT^k with security s ($\binom{2}{1}$-OT^k_s for short) is correct (Definition 1) and has the property that there exists an event $\mathcal{S}$ with probability at least $1 - 2^{-s}$, taken over all possible choices of $\tilde{\mathcal{B}}$ and over all the coin tosses of $\bar{\mathcal{A}}$, such that given that $\mathcal{S}$ occurs, the receiver $\tilde{\mathcal{B}}$ obtains no information about one of the k-bit strings, even when given the other (Definition 2).

In the following sections we focus only on the non-trivial aspects of the above definitions. In particular, we do not demonstrate correctness which immediately follows from the design of each protocol. Also, each protocol is such that the only information $\mathcal{A}$ might obtain throughout the protocol is via the use of the bit primitive ($\binom{2}{1}$-OT, $\binom{2}{1}$-XOT, $\binom{2}{1}$-GOT, ...). Since we assume they all satisfy condition (3), it follows immediately that our protocols also satisfy condition (3).

Thus our proofs of security focus solely on demonstrating that our protocols satisfy condition (4) with probability at least $1 - 2^{-s}$.

5. Reducing String OT to Bit OT

We show first that Protocol 3.1 combined with Protocol 3.2 (in the following referred to simply as Protocol 3.1) allows for reducing string OT to ordinary bit OT, where the number of required realizations of bit OT is only twice the length of the strings plus the security parameter s. We assume bit OT to be given as a black-box where the only thing the parties can do is to provide legitimate inputs at their choosing and get the corresponding outputs.

Theorem 1. *Protocol* 3.1 *allows for reducing* $\binom{2}{1}$-OT^k_s *to n realizations of* $\binom{2}{1}$-OT *for any*

$$n \geq 2(k + s + 1). \tag{5}$$

Before proving Theorem 1, we show that for the security of string OT it is sufficient that the receiver is not able to get any (non-negligible) information about any non-trivial linear function from one of the strings to a single bit, and additionally any such function from the pair of strings to one bit that depends non-trivially on *both* strings (Theorem 4).

Lemma 2. *Let S be a random variable taking k-bit strings as values, i.e.,* $\mathcal{S} \subseteq \mathcal{F}_2^k$. *Assume that for all non-constant linear functions g mapping $\mathcal{F}_2^k$ to $\mathcal{F}_2$, the bit $g(S)$ is symmetric, i.e.,* $\text{Prob}[g(S) = 1] = 1/2$. *Then S is uniformly distributed over* $\mathcal{F}_2^k$.

Proof. Let $(g_1, g_2, \ldots, g_{2^k-1})$ and $(s_1, s_2, \ldots, s_{2^k-1})$ be lists of all non-constant linear functions from $\mathcal{F}_2^k$ to $\mathcal{F}_2$ and of all non-zero k-bit strings, respectively. We consider the following mapping from distributions P_S over $\mathcal{F}_2^k$ to lists of probabilities

$$(\text{Prob}[g_1(S) = 1], \text{Prob}[g_2(S) = 1], \ldots, \text{Prob}[g_{2^k-1}(S) = 1]).$$

This is a mapping from $\mathbf{R}^{2^k-1}$ to $\mathbf{R}^{2^k-1}$:

$$\begin{pmatrix} P_S(s_1) \\ P_S(s_2) \\ \vdots \\ P_S(s_{2^k-1}) \end{pmatrix} \mapsto \begin{pmatrix} \text{Prob}[g_1(S) = 1] \\ \text{Prob}[g_2(S) = 1] \\ \vdots \\ \text{Prob}[g_{2^k-1}(S) = 1] \end{pmatrix}.$$

It is clear that this mapping is linear and that the corresponding real $(2^k - 1) \times (2^k - 1)$ matrix has the property that all the row vectors consist of $2^{k-1} - 1$ zeros and 2^{k-1} ones, and every pair of row vectors has a pair of ones at exactly 2^{k-2} of the positions. (The row vectors are the non-zero codewords of the dual code to a Hamming code.) The described matrix is called the Hadamard matrix and is well known to be invertible. For the sake of transparence, however, we give a short proof of this fact.

We show that the row vectors of the matrix are linearly independent over $\mathbf{R}$, and that hence the matrix is invertible.

First, all the row vectors v_i have the same norm $|v_i| = \sqrt{2^{k-1}}$ in $\mathbf{R}^{2^k-1}$, and secondly, every pair of such vectors has the same scalar product $\langle v_i, v_j\rangle = 2^{k-2}$ $(i \neq j)$. We show that any set of vectors with these properties must be linearly independent. It is sufficient to show this for a set of vectors v_i with the property

$$\langle v_i, v_i\rangle = 1, \qquad \langle v_i, v_j\rangle = \alpha \quad (\text{if } i \neq j)$$

for some $0 < \alpha < 1$.

Assume that we have a set of $r + 1$ such vectors with

$$v_{r+1} = \sum_{i=1}^{r} \lambda_i v_i .$$

Then we get for $1 \leq i \leq r$,

$$\alpha = \langle v_{r+1}, v_i\rangle = \lambda_i + \sum_{j \neq i} \lambda_j \alpha = (1 - \alpha)\lambda_i + \sum_{j=1}^{r} \lambda_j \alpha,$$

hence $\lambda_i = \alpha(1 - \sum_j \lambda_j)/(1 - \alpha)$ for all i, as a result all λ_i are the same (because it does not depend on i) and thus $\lambda_i = \alpha/(1 + (r - 1)\alpha)$ for all i. Then

$$1 = \langle v_{r+1}, v_{r+1}\rangle = \left(\frac{\alpha}{1 + (r - 1)\alpha}\right)^2 (r + r(r - 1)\alpha)$$

implies $r = -1/\alpha$, which is a contradiction. Hence the row vectors of the described matrix must be linearly independent, and the matrix itself is thus invertible. Therefore, the distribution P_S satisfying $\text{Prob}[g_i(S) = 1] = 1/2$ for all i is uniquely determined.

This concludes the proof, since the uniform distribution is clearly a distribution for which all the bits $g_i(S)$ are unbiased. □

The proof of Lemma 2 actually shows the following stronger statement on general distributions and "linear-functional characteristics."

Lemma 3. *Let S be a random variable taking k-bit strings as values, i.e., $\mathcal{S} \subseteq \mathcal{F}_2^k$. Then P_S is uniquely determined by the values* $\mathrm{Prob}[g(S) = 1]$ *for all linear functions g mapping $\mathcal{F}_2^k$ to $\mathcal{F}_2$.*

Theorem 4. *Let S be a random variable taking as values $2k$-bit strings, $\mathcal{S} \subseteq \mathcal{F}_2^{2k}$, and let S_1 and S_2 denote the first and second halves of S, respectively. Assume that the distribution P_S (i.e., the joint distribution of S_1 and S_2) has the following two properties*:

1. *For every linear function h mapping $\mathcal{F}_2^k$ to $\mathcal{F}_2$, $h(S_1)$ is a symmetric bit, i.e.,* $\mathrm{Prob}[h(S_1) = 1] = 1/2$, *and*
2. *for all linear functions $g(\cdot,\cdot)$ mapping $\mathcal{F}_2^k \times \mathcal{F}_2^k$ to $\mathcal{F}_2$ and such that g depends non-trivially on both inputs, $g(S_1, S_2)$ is a symmetric bit.*

Then S_1 and S_2 are independent, i.e.,

$$P_{S_1 S_2}(s_1, s_2) = P_{S_1}(s_1) \cdot P_{S_2}(s_2),$$

and S_1 is uniformly distributed.

Proof. First, it is straightforward to see that the product distribution $P_{S_1} \cdot P_{S_2}$, where P_{S_1} is the uniform distribution over $\mathcal{F}_2^k$, is a particular distribution with the given "linear-functional characteristic". Here, this characteristic is completed by the functionals that are non-trivial only on the second input which uniquely determine, and are uniquely determined by, the marginal distribution P_{S_2}. By Lemma 3, the distribution $P_{S_1 S_2}$ is uniquely determined by this characteristic, and this concludes the proof. □

We are now ready to prove Theorem 1. In fact, we even prove a statement stronger than Theorem 1, since we will give $\mathcal{B}$ more possibility of choice: instead of choosing one of the bits sent, $\mathcal{B}$ is also allowed to obtain the XOR of the two bits. (In this case, $\mathcal{B}$'s choice "trit" is equal to "$\oplus$", and $x_\oplus$ stands for $x_0 \oplus x_1$. However, an honest Bob would never choose $\oplus$. See Section 6 for a detailed discussion of so-called XOR-OT.)

Proof of Theorem 1. We first show that with high probability, one of the two strings r_0 and r_1 is perfectly uniformly distributed from $\tilde{\mathcal{B}}$'s point of view. First, it is clear that for (at least) one of the strings x_0 and x_1, $\tilde{\mathcal{B}}$ has no information about at least half the bits $x_0^1, x_0^2, \ldots, x_0^n$ or $x_1^1, x_1^2, \ldots, x_1^n$, respectively. More precisely, there exists a bit $\tilde{c} \in \{0, 1\}$ and a subset $S \subseteq \{1, \ldots, n\}$ of size at least $n/2$ such that for all $i \in S$, we have

$$\mathbf{H}(X_{\tilde{c}}^i \mid X_{c_1}^1 X_{c_2}^2 \cdots X_{c_n}^n) = 1,$$

where $X_{c_1}^1 X_{c_2}^2 \cdots X_{c_n}^n$ summarizes the entire information $\tilde{\mathcal{B}}$ has obtained during the execution of the n bit OTs (where every choice c_i is in $\{0, 1, \oplus\}$). We can assume without loss of generality that this string is x_0, i.e., $\tilde{c} = 0$.

Let now h be any specific non-constant linear function mapping $\mathcal{F}_2^k$ to $\mathcal{F}_2$. Then we have

$$h(r_0) = hM_0x_0 = m_0 \odot x_0 = \bigoplus_i m_0^i x_0^i,$$

where the second h is the name for the vector that represents the function h, which yields a random non-zero vector of bits $m_0 = hM_0$ (where the sum is modulo 2). Since $\tilde{\mathcal{B}}$ has no information at all about at least $n/2$ bits among the x_0^i, he has, with probability at least $1 - (1/2)^{n/2}$, no information at all about the bit $h(r_0)$. By the union bound, we can conclude that with probability at least $1 - 2^k(1/2)^{n/2}$, $\tilde{\mathcal{B}}$ has no information about the bit $h(r_0)$ for any linear function h. By Lemma 2, this event implies that in $\tilde{\mathcal{B}}$'s view, r_0 is perfectly uniformly distributed.

We now describe the condition under which $\tilde{\mathcal{B}}$ learns $g(r_0, r_1)$ at Step 3 of Protocol 3.2 for some specific non-trivial linear function g defined by two strings g_0 and g_1: $g(r_0, r_1) := g_0 \odot r_0 \oplus g_1 \odot r_1$. By definition

$$g(r_0, r_1) = g_0 \odot r_0 \oplus g_1 \odot r_1 = g_0 M_0 x_0 \oplus g_1 M_1 x_1 = z_0 \odot x_0 \oplus z_1 \odot x_1,$$

where $z_0 = g_0 M_0$ and $z_1 = g_1 M_1$. Because x_0 and x_1 are random, $\tilde{\mathcal{B}}$ cannot learn anything about $g(r_0, r_1)$ at Step 3 unless he is lucky enough that his choices c_i simultaneously follow

$$c_i = \begin{cases} 0 & \text{when} \quad (z_0^i, z_1^i) = (1, 0), \\ 1 & \text{when} \quad (z_0^i, z_1^i) = (0, 1), \\ \oplus & \text{when} \quad (z_0^i, z_1^i) = (1, 1) \end{cases}$$

in all the instances of $\binom{2}{1}$-OT such that z_0^i and z_1^i are not both zero. (The value of c_i is unimportant when $(z_0^i, z_1^i) = (0, 0)$ since neither x_0^i nor x_1^i nor $x_0^i \oplus x_1^i$ is required in that case to compute $g(r_0, r_1)$.)

Remember that M_0 and M_1 are picked at random among rank k matrices and neither g_0 nor g_1 is zero. Therefore $z_0 = g_0 M_0$ and $z_1 = g_1 M_1$ are random non-zero binary strings of length n chosen independently according to the uniform distribution. In particular, z_0 and z_1 are independent of $\tilde{\mathcal{B}}$'s choice of the c_i's. It follows that, for each i, the probability that either $(z_0^i, z_1^i) = (0, 0)$ or $\tilde{\mathcal{B}}$'s choice c_i turns out to have been appropriate according to the above case analysis is at most

$$1 - \tfrac{3}{4} \cdot \tfrac{2}{3} = \tfrac{1}{2}.$$

Since $\tilde{\mathcal{B}}$ must have been lucky for each i, $1 \le i \le n$,

$$\text{Prob}[\tilde{\mathcal{B}} \text{ learns } g(r_0, r_1)] \le 2^{-n}$$

for each non-trivial linear function g, whatever choices $\tilde{\mathcal{B}}$ makes for the c_i's. Finally, given that there are less than 2^{2k} such linear functions, we conclude that

$$\text{Prob}[\text{there exists a non-trivial } g \text{ such that } \tilde{\mathcal{B}} \text{ learns } g(r_0, r_1)] < 2^{2k-n}.$$

Here, the fact that $\tilde{\mathcal{B}}$ *does not learn* $g(r_0, r_1)$ means that a dishonest receiver does not get any information at all about this bit.

Altogether, we get that the probability that both strings r_0 and r_1 are *not* uniformly distributed in $\tilde{\mathcal{B}}$'s view or that $\tilde{\mathcal{B}}$ has some information about $g(r_0, r_1)$ for any non-trivial g is upper bounded by

$$2^{k-n/2} + 2^{2k-n} \le 2^{k-n/2+1} \le 2^{-s}$$

if inequality (5) is satisfied. We thus conclude that except with probability 2^{-s} for uniformly distributed independent $R_0, R_1, \forall c \in \mathcal{F}_2, \forall \tilde{\mathcal{B}}, \exists \tilde{C} \in RV(\mathcal{F}_2)$ s.t.

$$\mathbf{I}(R_{\neg\tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^{*}_{\mathcal{B}}(R_0, R_1)(C) \mid R_{\tilde{C}}, C = c) = 0.$$

Finally, since these two strings R_0, R_1 are used as one-time pads for W_0, W_1 the same property transfers to these as well:

$$\mathbf{I}(W_{\neg\tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^{*}_{\mathcal{B}}(W_0, W_1)(C) \mid W_{\tilde{C}}, C = c) = 0. \qquad \square$$

6. XOR-OT and Reversing OT

A $\binom{2}{1}$-XOT is an extension of $\binom{2}{1}$-OT that enables a sender $\mathcal{A}$ to transfer to a receiver $\mathcal{B}$ either one bit among b_0 and b_1 or their exclusive-or, at $\mathcal{B}$'s choice. More formally, $\mathcal{A}$ inputs b_0 and b_1 into the protocol, $\mathcal{B}$ inputs $c \in \{0, 1, \oplus\}$, and $\mathcal{B}$ learns b_c while $\mathcal{A}$ learns nothing, where, again, for convenience we use $b_\oplus$ to denote $b_0 \oplus b_1$. As usual, this is done in an all-or-nothing fashion: $\mathcal{B}$ cannot get more information about b_0 and b_1 than b_0, b_1, or $b_\oplus$, however malicious or computationally powerful he is. Note that in our application of $\binom{2}{1}$-XOT, which is to use it instead of $\binom{2}{1}$-OT inside Protocol 3.2, an honest $\mathcal{B}$ would never request $b_\oplus$. Therefore we can safely use any protocol in which it is merely *tolerated* that $\tilde{\mathcal{B}}$ might learn $b_\oplus$ in cheating attempts even though $\mathcal{A}$ is not required to provide it upon request.

The $\binom{2}{1}$-XOT comes naturally in a specific implementation of $\binom{2}{1}$-OT: in [6] a protocol for $\binom{2}{1}$-OT is given under the assumption that deciding quadratic residuosity modulo a composite number is hard. In that implementation, the possibility that $\tilde{\mathcal{B}}$ obtains $b_\oplus$ arises naturally and some effort is made to prevent it. The current paper shows that this effort was unnecessary if the final goal is to implement $\binom{2}{1}$-OT^k rather than simply $\binom{2}{1}$-OT. Indeed, the proof of Theorem 1 already shows that Protocol 3.1 reduces string OT to $\binom{2}{1}$-XOT, so no additional proof is required. We assume $\binom{2}{1}$-XOT is given as a black-box where the only thing the parties can do is to provide legitimate inputs at their choosing and get the corresponding outputs.

Theorem 5. *Protocol* 3.1 *allows for reducing* $\binom{2}{1}$-OT^k_s *to n realizations of* $\binom{2}{1}$-XOT *for any*

$$n \geq 2(k + s + 1).$$

As an application of Theorem 5 we consider the problem of inverting the direction of an OT. More precisely, consider that $\mathcal{A}$ wants to send one of two words w_0 or w_1 to $\mathcal{B}$ when they only have a $\binom{2}{1}$-OT channel running from $\mathcal{B}$ to $\mathcal{A}$. A very efficient protocol for sending one of two *bits* from $\mathcal{A}$ to $\mathcal{B}$ is given in [17] provided $\mathcal{A}$ does not mind the possibility that $\mathcal{B}$ might learn the exclusive-or of her two bits: two instances of reversed $\binom{2}{1}$-OT are sufficient to implement $\binom{2}{1}$-XOT. For completeness we include this very simple protocol:

Protocol 6.1. $(\binom{2}{1}\text{-XOT}(b_0, b_1)(c))$

1. *$\mathcal{B}$ picks four random bits u_0, u_1, v_0, v_1 such that $u_i = v_i$ iff $\neg i = c$.*
2. *$\mathcal{B}$ transfers $t_i \leftarrow \binom{2}{1}\text{-OT}(u_i, v_i)(b_i)$ to $\mathcal{A}$, $i \in \{0, 1\}$.*
3. *$\mathcal{A}$ announces $t \leftarrow t_0 \oplus t_1$ to $\mathcal{B}$.*
4. *$\mathcal{B}$ recovers b_c by computing $t \oplus u_0 \oplus u_1$.*

We leave it as an easy exercise to the reader to check that the following table is correct:

u_0, v_0	u_1, v_1	$t_0 \oplus t_1 \oplus u_0 \oplus u_1$	c
$\neq$	$=$	b_0	0
$=$	$\neq$	b_1	1
$\neq$	$\neq$	$b_0 \oplus b_1$	$\oplus$
$=$	$=$	0	

No known construction efficiently implements $\binom{2}{1}$-OT from so few instances of reversed $\binom{2}{1}$-OT. In other words, it is currently much easier to implement $\binom{2}{1}$-XOT rather than $\binom{2}{1}$-OT from $\mathcal{A}$ to $\mathcal{B}$ given a $\binom{2}{1}$-OT channel from $\mathcal{B}$ to $\mathcal{A}$. This is fine because we just showed that $\binom{2}{1}$-XOT is just as good as $\binom{2}{1}$-OT for the purpose of implementing $\binom{2}{1}$-OT^k. Therefore, $\binom{2}{1}$-OT^k from $\mathcal{A}$ to $\mathcal{B}$ can be implemented from slightly more than $4k$ instances of $\binom{2}{1}$-OT from $\mathcal{B}$ to $\mathcal{A}$. This is a sixfold improvement over [17].

Corollary 6. $\binom{2}{1}$-OT$_s^k$ *can be reduced to n realizations of* $\binom{2}{1}$-OT *from $\mathcal{B}$ to $\mathcal{A}$ for any*

$$n \geq 4(k + s + 1).$$

7. Generalized OT: Uncertainty Concentration and Erasure Channels

A $\binom{2}{1}$-GOT is a cryptographic primitive for two participants that enables a sender $\mathcal{A}$ to transfer a one-bit function evaluated on (b_0, b_1) to a receiver $\mathcal{B}$ who chooses secretly which one-bit function f he gets from her input bits. This is, again, done in an all-or-nothing fashion: $\mathcal{B}$ cannot get more information about b_0 and b_1 than $f(b_0, b_1)$ for some f, however malicious or computationally powerful he is, and $\mathcal{A}$ finds out nothing about the choice f of $\mathcal{B}$. As was the case with $\binom{2}{1}$-XOT in Section 6, one may think of a $\binom{2}{1}$-GOT protocol as merely tolerating the fact that a cheating $\tilde{\mathcal{B}}$ might learn $f(b_0, b_1)$ for some f rather than specifying that any such f can be learned at $\mathcal{B}$'s whim.

Table 2 enumerates all 14 possible non-constant functions from two bits to one. (We ignore the two constant function since they would yield no information if used.) The symbols used refer to the common boolean functions. Example: $\overline{\wedge}$ stands for $\overline{b_0 \wedge b_1}$. The notations 0 and 1 are used for the projection functions $b_0 0 b_1 = b_0$ and $b_0 1 b_1 = b_1$. We say that a function $f(b_0, b_1)$ is *biased* if the probability that $f(b_0, b_1) = 1$ is not

Table 2. Enumeration of non-constant functions.

		Function													
b_0	b_1	$\overline{\vee}$	$\overline{\rightarrow}$	$\overline{1}$	$\overline{\leftarrow}$	$\overline{0}$	$\oplus$	$\overline{\wedge}$	$\wedge$	$\overline{\oplus}$	0	$\leftarrow$	1	$\rightarrow$	$\vee$
0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
0	1	0	0	0	1	1	1	1	0	0	0	0	1	1	1
1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1
Biased		√	√		√			√	√			√		√	√

$1/2$ when b_0 and b_1 are chosen randomly and independently according to the uniform distribution. The ordinary $\binom{2}{1}$-OT is a special case of $\binom{2}{1}$-GOT where $\mathcal{B}$ is limited to the functions 0 and 1.

It has been shown in [5] that $\binom{2}{1}$-GOT is a sufficient primitive to implement $\binom{2}{1}$-OT. The reduction they presented uses $\Theta(s)$ runs of $\binom{2}{1}$-GOT to achieve a single $\binom{2}{1}$-OT in such a way that the reduction may fail and give both bits to $\mathcal{B}$ with probability 2^{-s}. If this protocol is combined with a standard reduction of $\binom{2}{1}$-OT^k we obtain a global cost of $\Theta(ks)$ runs of $\binom{2}{1}$-GOT per $\binom{2}{1}$-OT^k. Contrary to reductions to $\binom{2}{1}$-OT, reductions to $\binom{2}{1}$-GOT *must* involve a failure probability since it is *always* possible to get all the information sent by $\mathcal{A}$ by selecting the appropriate biased function at each transfer by sheer luck. For example, if $\mathcal{B}$ requests $x_0^i \wedge x_1^i$ at Step 2 of Protocol 3.2 for some i, and if he obtains the value 1, then he knows that both x_0^i and x_1^i are equal to 1. Using the new privacy amplification method we obtain a direct reduction of $\binom{2}{1}$-OT^k at a cost of only slightly more than $4.8188\,k$ instances of $\binom{2}{1}$-GOT.

We consider the variation of Protocol 3.2 in which $\binom{2}{1}$-OT is replaced by $\binom{2}{1}$-GOT. We assume $\binom{2}{1}$-GOT is given as a black-box where the only thing the parties can do is to provide legitimate inputs at their choosing and get the corresponding outputs. We show that the reduction still works, where the number of required realizations of the $\binom{2}{1}$-GOT is greater by a factor of roughly 2.4094 than for the reduction to $\binom{2}{1}$-XOT. This is an improvement to the analysis of [4]. Moreover, the proof given below is considerably simpler.

Theorem 7. *Protocol* 3.1 *allows for reducing* $\binom{2}{1}$-OT$_s^k$ *to* n *realizations of* $\binom{2}{1}$-GOT *for any*

$$n \geq \frac{2}{2 - \log_2 3}\,(k + s + 1) \approx 4.8188\,(k + s + 1). \tag{6}$$

For the proof of Theorem 7 we need the following lemma which states that among all possible types of (partial) information about a bit which lead to the same error probability when guessing the bit, the particular information that is obtained by sending the bit over a symmetric erasure channel provides the largest amount of information about the bit.

More explicitly, and even stronger than that, we show that for every other type of information U about a bit B, there exists *additional* information V (that can be thought of as being provided by an oracle) such that given V together with U, the situation perfectly corresponds to information resulting when B is sent over an erasure channel, and the probability of guessing B correctly given the additional information is unchanged.

This "additional-information argumentation" leads to a partial order on all possible types of side information about a random variable in a very strict sense: the "stronger" side information is "more powerful" than the weaker one in every respect because the stronger information *contains* the weaker one.

Intuitively speaking, Lemma 8 states that the uncertainty about the bit B can be concentrated in an event (called $\{V = \Delta\}$ here, where Δ stands for the erasure symbol of the erasure channel) of probability $2p$: given this event, B is symmetrically distributed, i.e., its uncertainty is maximal.

Lemma 8. *Let B be a symmetric binary random variable (i.e., its range is $\{0, 1\}$ and $P_B(0) = P_B(1) = 1/2$ holds), and let U be a random variable such that B and U have joint distribution P_{BU}. Let p be the average error probability of guessing B when given U, using the optimal guessing strategy. Then there exists a random variable V with the following properties*:

1. *The range of V is $\mathcal{V} = \{0, 1, \Delta\}$*,
2. $P_V(\Delta) = 2p$,
3. *for every $u \in \mathcal{U}$, we have*

$$P_{B|U=u,V=\Delta}(0) = P_{B|U=u,V=\Delta}(1).$$

Proof. Let $u \in \mathcal{U}$, and assume without loss of generality that $a = P_{B|U=u}(0) \geq P_{B|U=u}(1) = b$. Let V be defined by

$$\begin{aligned} P_{V|B=0,U=u}(0) &= (a-b)/a, \\ P_{V|B=0,U=u}(\Delta) &= b/a, \\ P_{V|B=1,U=u}(\Delta) &= 1. \end{aligned}$$

Note that $P_{V|U=u}(\Delta) = 2p$, i.e., twice the error probability for guessing B when given $U = u$. This concludes the proof. □

Proof of Theorem 7. First, we observe that for all i, $\tilde{\mathcal{B}}$'s expected error probability about at least one of the two bits x_0^i and x_1^i is $1/4$. (This holds with equality if $\tilde{\mathcal{B}}$ chooses a *biased* function in the ith realization of GOT.) Hence we can assume, according to Lemma 8, without loss of generality that $\tilde{\mathcal{B}}$ receives the corresponding bit over a symmetric erasure channel with erasure probability $1/2$. Hence at least one of the two strings x_0 and x_1 (say x_0 without loss of generality) contains at least $n/2$ bits about which $\tilde{\mathcal{B}}$ has no information at all with probability at least $1/2$. Consequently, if h is a fixed non-constant linear function mapping $\mathcal{F}_2^k$ to $\mathcal{F}_2$, then $\tilde{\mathcal{B}}$ has no information about $h(r_0)$ with probability at least $1 - (3/4)^{n/2}$ (the probability that a particular one of these $n/2$ bits appears in the sum, i.e., has to be known to get any information about the sum, but is not known to $\tilde{\mathcal{B}}$, is $1/4$). By the union bound, $\tilde{\mathcal{B}}$ has no information about *any* $h(r_0)$

with probability at least

$$1 - 2^k \left(\frac{3}{4}\right)^{n/2} \geq 1 - \frac{2^{-s}}{2}.$$

Let now $g(\cdot, \cdot)$ be a fixed non-trivial linear function mapping $(\mathcal{F}_2^k)^2$ to $\mathcal{F}_2$. As shown already in the proof of Theorem 1, $\tilde{\mathcal{B}}$ needs, in order to obtain information about $g(r_0, r_1)$, one of the bits x_0^i, x_1^i, or $x_\oplus^i$ (or nothing with probability at most $1/4$), depending on the choice of the random linear functions, for all i. We have seen already that $\tilde{\mathcal{B}}$'s expected error probability about the bit he needs is at least $1/4$ if he chooses to see an unbiased function of the two bits. If he chooses a (non-constant) *biased* function, this probability is as follows. For fixed i, $\tilde{\mathcal{B}}$ needs one of the bits x_0, x_1, or $x_\oplus$ with probability at least $3/4$. Given the choice of a biased function, with probability $1/4$, the obtained value tells him both bits x_0 and x_1, and otherwise (hence with probability $3/4$), his error probability about the required bit is $1/3$. Altogether, the *expected* error probability is at least

$$\tfrac{3}{4} \cdot \tfrac{3}{4} \cdot \tfrac{1}{3} = \tfrac{3}{16}$$

(hence slightly smaller than when choosing an unbiased function).

According to Lemma 8, we can assume that $\tilde{\mathcal{B}}$ obtains the bit over a binary and symmetric erasure channel with erasure probability $3/8$. In this case, the probability that he learns all the n bits he needs is $(5/8)^n$. (Note that otherwise, he has *no information at all* about the corresponding bit.) Since

$$n \geq \frac{2}{2 - \log_2 3}(k + s + 1) \geq \frac{2k + s + 1}{\log_2(8/5)},$$

the probability that he learns some information about at least one of the values $g(r_0, r_1)$ is, by the union bound, at most

$$2^{2k} \cdot \left(\frac{5}{8}\right)^n \leq \frac{2^{-s}}{2}.$$

Using the union bound and Theorem 4 we thus conclude that except with probability $2^{-s}/2$ for uniformly distributed independent R_0, R_1, $\forall c \in \mathcal{F}_2$, $\forall \tilde{\mathcal{B}}$, $\exists \tilde{C} \in RV(\mathcal{F}_2)$ s.t.

$$\mathbf{I}(R_{\neg\tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^*_{\mathcal{B}}(R_0, R_1)(C) \mid R_{\tilde{C}}, C = c) = 0.$$

Finally, since these two strings R_0, R_1 are used as one-time pads for W_0, W_1 the same property transfers to these as well:

$$\mathbf{I}(W_{\neg\tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^*_{\mathcal{B}}(W_0, W_1)(C) \mid W_{\tilde{C}}, C = c) = 0. \qquad \square$$

8. Universal OT and Fano's Lemma

The most general (i.e., weakest) primitive in the described context appears to be the so-called *universal* OT proposed in [4]. Here, $\mathcal{B}$ is allowed to choose *any* type of information, in particular probabilistic information, about the bits sent by $\mathcal{A}$, not exceeding a certain bound on Shannon entropy. Obviously, this primitive is much more general than GOT. For instance, $\mathcal{B}$ can choose here to receive slightly noisy versions of both bits b_0 and b_1 (with some arbitrarily small error probability ε).

Definition 4. Let $\alpha > 0$. A *universal oblivious transfer with parameter* α (α-UOT for short) is a cryptographic primitive involving two parties $\mathcal{A}$ (called the *sender*) and $\mathcal{B}$ (the *receiver*). The sender $\mathcal{A}$'s input is a pair of bits (b_0, b_1). The receiver $\mathcal{B}$ on the other hand inputs an arbitrary discrete memoryless channel Ω with input alphabet $\{(0,0), (0,1), (1,0), (1,1)\}$ that must satisfy

$$\mathbf{H}((B_0, B_1) \mid \Omega(B_0, B_1)) \geq \alpha,$$

where $\Omega(B_0, B_1)$ is the random variable representing the channel's output when its input is the pair (B_0, B_1) of uniformly distributed independent bits. The receiver obtains $\Omega(B_0, B_1)$, but no additional information about (B_0, B_1). Finally, $\mathcal{A}$ learns nothing about $\mathcal{B}$'s choice of the channel Ω.

It was stated as an open problem in [4] whether this primitive is as strong as string OT, i.e., whether it is also possible to reduce $\binom{2}{1}$-OT^k efficiently to general UOT. We assume that α-UOT is given as a black-box where the only thing the parties can do is to provide legitimate inputs at their choosing and get the corresponding outputs. Theorem 9 shows that the answer to this question is *yes*, and that the number of required realizations of α-UOT (for any fixed $\alpha > 0$) is of order $O(k + s)$.

Theorem 9. *Protocol* 3.1 *reduces* $\binom{2}{1}$-OT$_s^k$ *to n realizations of* α-UOT *for every*

$$n \geq \frac{(k + s + 1) \cdot 4 \ln 2}{p_{\mathrm{e}}}, \tag{7}$$

where p_{e} *is the unique solution in* $(0, \frac{1}{2}]$ *to the equation*

$$\mathbf{h}(p_{\mathrm{e}}) + p_{\mathrm{e}} \log_2 3 = \alpha,$$

and where $\mathbf{h}(\cdot)$ *is the binary entropy function*.

The crucial point in the proof of Theorem 9 is to apply Fano's inequality which gives a lower bound on the error probability of guessing the outcome of a random variable, given its (conditional) entropy, and to apply Lemma 8 to the resulting situation.

Fano's Lemma (see [12]). *Let X and Y be two random variables, and let* p_{e} *be the error probability when guessing X with any strategy, given the outcome of Y. Then*

$$\mathbf{H}(X \mid Y) \leq \mathbf{h}(p_{\mathrm{e}}) + p_{\mathrm{e}} \cdot \log_2(|\mathcal{X}| - 1)$$

(*where* $\mathcal{X}$ *is the range of* X).

Proof of Theorem 9. Let n be the length of the strings x_0 and x_1 in Protocol 3.2. According to Fano's inequality, the expected error probability, given $\Omega_i(x_0^i, x_1^i)$, about the pair of bits (x_0^i, x_1^i) is at least p_{e}, where p_{e} stands for the unique solution in $(0, \frac{1}{2}]$ to the equation $\mathbf{h}(p_{\mathrm{e}}) + p_{\mathrm{e}} \cdot \log_2 3 = \alpha$. This means, by the union bound and since two of the bits determine the third one, that for at least two of the bits x_0^i, x_1^i, $x_\oplus^i$ $(:= x_0^i \oplus x_1^i)$, the expected error probability is at least $p_{\mathrm{e}}/2$.

From this we can conclude by Lemma 8 that in at least one of the strings x_0 and x_1 (say x_0) there are at least $n/2$ bits x_0^i about which $\tilde{\mathcal{B}}$ has no information with probability at least p_e. Let h be a fixed non-constant linear function mapping $\mathcal{F}_2^k$ to $\mathcal{F}_2$. Then $\tilde{\mathcal{B}}$ does not get any information about the bit $h(r_0)$ with probability at least $1 - (1 - p_e/2)^{n/2}$. By the union bound, he does not learn any of the bits $h(r_0)$ with probability at least

$$1 - 2^k(1 - p_e/2)^{n/2}.$$

From condition (7) we conclude that this probability is at least $1 - 2^{-s}/2$.

Let now $g(\cdot, \cdot)$ be a linear function mapping $[\mathcal{F}_2^r]^2$ to $\mathcal{F}_2$ depending non-trivially on both inputs. We consider the probability that $\tilde{\mathcal{B}}$ gets some information about the bit $g(r_0, r_1)$. For every $i = 1, \dots, n$, the bit $g(r_0, r_1)$ can be written as

$$a_0 x_0^i \oplus a_1 x_1^i \oplus L_i(x_0^1, \dots, x_0^{i-1}, x_0^{i+1}, \dots, x_0^n, x_1^1, \dots, x_1^{i-1}, x_1^{i+1}, \dots, x_1^n),$$

where $a_0, a_1 \in \mathcal{F}_2$ are independent and random (given that g depends non-trivially on both inputs and that M_0 and M_1 are independent and random among rank k matrices), and where L_i is a linear function mapping to a bit.

We conclude that with probability at least $1 - (1/4 + 3/4 \cdot 1/3) = 1/2$, $\tilde{\mathcal{B}}$'s expected error probability about the bit $a_0 x_0^i \oplus a_1 x_1^i$ he needs is at least $p_e/2$, hence his overall expected error probability is at least $(p_e/2)/2 = p_e/4$. As Lemma 8 shows, the worst case (for $\mathcal{A}$) is when $\tilde{\mathcal{B}}$ has full information about the required bit with conditional probability $1 - 2(p_e/4) = 1 - p_e/2$, and no information otherwise. Thus $\tilde{\mathcal{B}}$ will in this case have *no information at all* about $g(r_0, r_1)$ with probability

$$1 - (1 - p_e/2)^n.$$

Hence the probability $\text{Prob}[\mathcal{S}]$ of the event $\mathcal{S}$ that there exists a non-trivial bilinear function g such that $\tilde{\mathcal{B}}$ has some information about $g(r_0, r_1)$ is, by the union bound, bounded by

$$\text{Prob}[\mathcal{S}] < 2^{2k}(1 - p_e/2)^n < 2^{-s}/2$$

(we have used condition (7) here). Altogether, we can conclude by the union bound and by Theorem 4 that with probability at least $1 - 2^{-s}$, for uniformly distributed independent $R_0, R_1, \forall c \in \mathcal{F}_2, \forall \tilde{\mathcal{B}}, \exists \tilde{C} \in RV(\mathcal{F}_2)$ s.t.

$$\mathbf{I}(R_{\neg\tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^*_{\mathcal{B}}(R_0, R_1)(C) \mid R_{\tilde{C}}, C = c) = 0.$$

Finally, since these two strings R_0, R_1 are used as one-time pads for W_0, W_1 the same property transfers to these as well:

$$\mathbf{I}(W_{\neg\tilde{C}}; [\bar{\mathcal{A}}, \tilde{\mathcal{B}}]^*_{\mathcal{B}}(W_0, W_1)(C) \mid W_{\tilde{C}}, C = c) = 0. \qquad \square$$

9. Concluding Remarks

We have studied the problem of reducing string OT to bit OT and weaker primitives. The key technique we used is privacy amplification, which was shown useful earlier in the context of information-theoretic key agreement, in particular quantum key agreement.

We have shown that privacy amplification not only allows for a reduction of string OT to bit OT in a more efficient way than previously described approaches, but that it has the additional advantage that string OT can be reduced to apparently much weaker primitives such as generalized OT or universal OT. In conclusion, the privacy amplification method is better than previous methods in any situation as long as one is willing to accept an exponentially small probability of failure.

To emphasize the similarity of the protocol used in this paper to earlier proposals, consider the following variation on the combination of Protocols 3.1 and 3.2 (we leave it as an exercise to the reader to verify that this protocol is equivalent):

Protocol 9.1. $(\binom{2}{1}\text{-OT}^k(w_0, w_1)(c))$

1. *$\mathcal{A}$ picks two random $k \times n$ rank k matrices M_0 and M_1 over $\mathcal{F}_2$ and two random n-bit strings x_0 and x_1 such that $M_0 x_0 = w_0$ and $M_1 x_1 = w_1$.*
2. $\mathbf{DO}_{i=1}^{n}$ *$\mathcal{A}$ transfers $t^i \leftarrow \binom{2}{1}\text{-OT}(x_0^i, x_1^i)(c)$ to $\mathcal{B}$.*
3. *$\mathcal{A}$ announces M_0, M_1 to $\mathcal{B}$.*
4. *$\mathcal{B}$ recovers $w_c \leftarrow M_c t$.*

Notice that this protocol is *identical* to the so-called Monte Carlo Zigzag method from [7] except for the fact that $\mathcal{B}$ only learns M_0, M_1 *after* the $\binom{2}{1}$-OTs have taken place, whereas in the Zigzag method $\mathcal{B}$ learns M_0, M_1 *before* the $\binom{2}{1}$-OTs take place. It is known however that choosing a single $M_0 = M_1$ in the Zigzag method does not change the asymptotic probability that the linear code, with generating matrix M_0, (self-) intersects.

Finally, although it is tempting to adopt generalizations of this apparently simpler protocol, we believe that generalizing our main Protocols 3.1 and 3.2 is easier because in the case of a general hash function h finding x_0 and x_1 such that $h(x_0) = w_0$ and $h(x_1) = w_1$ may be much more time consuming than the forward calculations involved in Protocol 3.2.

Acknowledgments

We thank Christian Cachin, Ueli Maurer, Dominic Mayers, and Louis Salvail for their help, comments, suggestions, and support. We are also grateful to two anonymous referees for many helpful remarks and suggestions.

References

[1] D. Beaver, Foundations of secure interactive computing, *Advances in Cryptology – CRYPTO '91 Proceedings*, Springer-Verlag, Berlin, 1992, pp. 377–391.

[2] C. H. Bennett, G. Brassard, C. Crépeau, and U. M. Maurer, Generalized privacy amplification, *IEEE Transactions on Information Theory*, Vol. 41, no. 6, 1995, pp. 1915–1923.

[3] C. H. Bennett, G. Brassard, and J.-M. Robert, Privacy amplification by public discussion, *SIAM Journal on Computing*, Vol. 17, 1988, pp. 210–229.

[4] G. Brassard and C. Crépeau, Oblivious transfers and privacy amplification, *Advances in Cryptography – EUROCRYPT'* 97, LNCS, Vol. 1233, Springer-Verlag, Berlin, 1997. pp. 334–345.

[5] G. Brassard, C. Crépeau, and J.-M. Robert, Information theoretic reductions among disclosure problems, *Proceedings of* 27*th Annual IEEE Symposium on Foundations of Computer Science*, 1986, pp. 168–173.

[6] G. Brassard, C. Crépeau, and J.-M. Robert, All-or-nothing disclosure of secrets, *Advances in Cryptology*: *Proceedings of Crypto* '86, Springer-Verlag, Berlin, 1987, pp. 234–238.

[7] G. Brassard, C. Crépeau, and M. Sántha, Oblivious transfers and intersecting codes, *IEEE Transactions on Information Theory*, Vol. 42, no. 6, November 1996, pp. 1769–1780.

[8] C. Cachin, On the foundations of oblivious transfer, *Advances in Cryptography – EUROCRYPT'* 98, LNCS, Vol. 1403, Springer-Verlag, Berlin, 1998, pp. 361–374.

[9] J. L. Carter and M. N. Wegman, New hash functions and their use in authentication and set equality, *Journal of Computer and System Sciences*, Vol. 22, 1981, pp. 265–279.

[10] G. D. Cohen and A. Lempel, Linear intersecting codes, *Discrete Mathematics*, Vol. 56, 1985, pp. 35–43.

[11] G. D. Cohen and G. Zémor, Intersecting codes and independent families, *IEEE Transactions on Information Theory*, Vol. 40, no. 6, November 1994, pp. 1872–1881.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Series in Telecommunications, Wiley, New York, 1992.

[13] C. Crépeau, Verifiable disclosure of secrets and application, *Advances in Cryptology*: *Proceedings of Eurocrypt* '89, Springer-Verlag, Berlin, 1990, pp. 181–191.

[14] C. Crépeau, Correct and private reductions among oblivious transfers, Ph.D. thesis, MIT, 1990.

[15] C. Crépeau, Quantum oblivious transfer, *Journal of Modern Optics*, Vol. 41, no. 12, December 1994, pp. 2455–2466.

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

[17] C. Crépeau and M. Sántha, On the reversibility of oblivious transfer, *Advances in Cryptology*: *Proceedings of Eurocrypt* '91, Springer-Verlag, Berlin, 1991, pp. 106–113.

[18] C. Crépeau and M. Sántha, Efficient reductions among oblivious transfer protocols based on new self-intersecting codes, in *Sequences II*, *Methods in Communications*, *Security and Computer Science*, Springer-Verlag, Berlin, 1991, pp. 360–368.

[19] S. Even, O. Goldreich, and A. Lempel, A randomized protocol for signing contracts, *Proceedings of Crypto* 82, Plenum, New York, 1983, pp. 205–210.

[20] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof-systems, *SIAM Journal on Computing*, Vol. 18, 1989, pp. 186–208.

[21] J. Kilian, Founding cryptography on oblivious transfer, *Proceedings of* 20*th Annual ACM Symposium on Theory of Computing*, 1988, pp. 20–31.

[22] U. Maurer, Information-theoretic cryptography, *Advances in Cryptography – CRYPTO'* 99, LNCS, Vol. 1666, Springer-Verlag, Berlin, 1999, pp. 47–64.

[23] S. Micali, and P. Rogaway, Secure computation, *Advances in Cryptology – CRYPTO* '91 *Proceedings*, Springer-Verlag, Berlin, 1991, pp. 392–404.

[24] M. Naor, and B. Pinkas, Efficient oblivious transfer protocols, *Proceedings of the Twelfth Annual Symposium on Discrete Algorithms*, January 7–9, 2001, Washington, DC, pp. 448–457.

[25] M. O. Rabin, How to exchange secrets by oblivious transfer, Technical Memo TR-81, Aiken Computation Laboratory, Harvard University, 1981.

[26] D. R. Stinson, Some results on nonlinear zigzag functions, *Journal of Combinatorial Mathematics and Combinatorial Computing*, Vol. 29, 1999, pp. 127–138.

[27] S. Wiesner, Conjugate coding, *Sigact News*, Vol. 15, no. 1, 1983, pp. 78–88. Original manuscript written circa 1970.

[28] S. Wolf, Reducing string oblivious transfer to universal oblivious transfer, *Proceedings of ISIT* 2000, 2000.