Oblivious Transfer is Symmetric

Stefan Wolf Jürg Wullschleger

Computer Science Department, ETH Zürich, Switzerland.
{wolf,wjuerg}@inf.ethz.ch

Abstract. We show that oblivious transfer of bits from A to B can be obtained from a single instance of the same primitive from B to A. Our reduction is perfect and shows that oblivious transfer is in fact a symmetric functionality. This solves an open problem posed by Crépeau and Sántha in 1991.

1 Introduction

Modern cryptography is an increasingly broad discipline and deals with many subjects besides the classical tasks of encryption or authentication. An example is *multi-party computation*, where two or more parties, mutually distrusting each other, want to collaborate in a secure way in order to achieve a common goal, for instance, to carry out an electronic election. An example of a specific multi-party computation is *secure function evaluation*, where every party holds an input to a function, and the output should be computed in a way such that no party has to reveal unnecessary information about her input.

A primitive of particular importance in the context of two- and multi-party computation is *oblivious transfer*. In classical *Rabin oblivious transfer* [19] or *Rabin OT* for short, one of the parties—*the sender*—sends a bit b which reaches *the receiver* with probability 1/2; the sender hereby remains ignorant of about whether the message has arrived or not. In other words, Rabin OT is nothing else than a binary erasure channel. Another variant of oblivious transfer is *chosen one-out-of-two oblivious transfer*—$\binom{2}{1}-\mathrm{OT}$ for short—, where the sender sends two bits b_0 and b_1 and the receiver's input is a choice bit c; the latter then learns b_c but gets no information about the other bit b_{1-c}. Chosen one-out-of-two oblivious transfer can be generalized to a primitive where the sender sends n messages, k of which the receiver can choose to read: *chosen k-out-of-n l-bit string oblivious transfer* or $\binom{n}{k}-\mathrm{OT}^l$. One reason for the importance of oblivious transfer is its *universality*, i.e., it allows, in principle, for carrying out *any* two-party computation [14].

Besides *computational* cryptographic security, which is based on the assumed hardness of certain computational problems and a limitation on the adversary's computing power, there also exists *unconditional* security, which is based on the fact that the *information* the potential adversary obtains is limited. This latter type of security withstands attacks even by a computationally unlimited adversary; clearly, it is, *a priori*, more desirable to realize cryptographic primitives in such an unconditionally secure way. Unfortunately, oblivious transfer is impossible to achieve in an unconditionally secure way from scratch, i.e., between parties connected by a noiseless channel; in fact, not even if this is a *quantum* channel over which the parties can exchange not only "classical" bits but quantum states [15]. However, if some additional weak and realistic primitives are available such as noisy channels and noisy correlations, then unconditional security *can* be often achieved [7], [6], [8], [13], [22], [23].

Another way of realizing unconditionally secure oblivious transfer is from (a weaker form of) oblivious transfer itself: All the variants of oblivious transfer have been shown equivalent to different extents. For instance, $\binom{2}{1}-\mathrm{OT}$ can be reduced to m realizations of Rabin OT as long as a failure probability of 2^{-m} can be accepted [5]. On the other hand, $\binom{2}{1}-\mathrm{OT}^l$ can be reduced to $\Theta(l)$ realizations of $\binom{2}{1}-\mathrm{OT}$—with or without failure probability, where the reduction can be made more efficient in terms of the hidden constant if a small probability of failure can be accepted [4]. In [2], a protocol was presented that reduces $\binom{2}{1}-\mathrm{OT}$ to a $\binom{2}{1}-\mathrm{OT}$ being available at an earlier point in time. This means that $\binom{2}{1}-\mathrm{OT}$ can be *precomputed* (or *stored* and used at any time later).

In [18] and [9], methods were proposed for obtaining $\binom{2}{1}-\mathrm{OT}$ from A to B from n instances of $\binom{2}{1}-\mathrm{OT}$ from B to A, where a failure probability of $2^{-\Theta(n)}$ has to be tolerated. The protocol of [18] is based on the realization of so-called "XOT" (i.e., the receiver can also choose to receive the XOR of the two bits sent) from two realizations of $\binom{2}{1}-\mathrm{TO}$—the *reversed* version of $\binom{2}{1}-\mathrm{OT}$. Note, however, that the resulting reduction of $\binom{2}{1}-\mathrm{OT}$ to $\binom{2}{1}-\mathrm{TO}$ of [18] also requires $\Theta(\log(1/\varepsilon))$ realizations of $\binom{2}{1}-\mathrm{TO}$ if ε is the tolerated failure probability.

1.1 Our Contribution

In [9], Crépeau and Sántha raised the question of whether it is possible to implement oblivious transfer in one direction using fewer instances of oblivious transfer in the other. In this paper, we answer this question with *yes* by presenting a protocol that needs *one* instance of oblivious transfer, *one* bit of communication and *one* bit of additional (local) randomness. All these parameters are optimal. Our reduction is very simple; in other words, the reversed version of oblivious transfer is basically just another way of *looking* at it. The symmetry is already there, *oblivious transfer is symmetric*.

Our reduction can be used to transform *any* protocol for $\binom{2}{1}-\mathrm{OT}$—offering either computational or information-theoretic security for A and B, respectively—into a protocol for oblivious transfer from B to A having exactly the same security both for A and B as the original protocol; no additional failure can occur.

1.2 Outline

In Section 3, we first present protocols from [2] that allow $\binom{2}{1}-\mathrm{OT}$ to be "stored", i.e., to transform oblivious transfer into an *oblivious key*. Then, we will show that such an oblivious key can very easily be reversed—by a simple XOR executed by both players on their local data. It follows that oblivious transfer can be reversed equally easily. In Section 4 we present an even simpler protocol for reversing oblivious transfer and prove its security.

2 Definitions and Security

We define $\binom{2}{1}-$OT as a *black-box* (see Figure 1).

Definition 1. By $\binom{2}{1}-$OT or *chosen one-out-of-two oblivious transfer* we denote the following primitive between a sender A and a receiver B. A has two inputs b_0 and b_1 and no output, and B has input c and output y such that $y = b_c$.

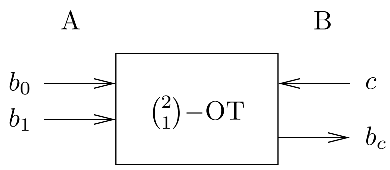


Fig. 1. Chosen one-out-of-two oblivious transfer.

For the *reversed* version of $\binom{2}{1}-$OT, where B is the sender and A is the receiver, we will write $\binom{2}{1}-$TO.

This black-box model of oblivious transfer is called the *ideal model*, and it is how the world is supposed to be: The players have no other way of accessing the box than by the defined inputs and outputs: cheating is impossible. However, in reality such a perfect box does normally not exist. It must be *simulated* by a *protocol*. In this *real model*, the players can cheat in principle by not following the rules. A protocol is called a *secure implementation* of oblivious transfer if an adversary cannot do anything in the real model that he could not just as well have done in the ideal model. Thus, it must be shown that for any adversary in the real model, there exists an equivalent adversary in the ideal model: he gets the same information and the honest player obtains the same outputs as in the real model; the resulting views are *indistinguishable*.

We follow the formalism of [16] and [1] (see also [11]) to define when a protocol perfectly securely evaluates a function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{U} \times \mathcal{V}$. A *protocol* is a pair of algorithms $A = (A_1, A_2)$ that can interact by two-way message exchange. A pair $(\overline{A_1}, \overline{A_2})$ of algorithms is *admissible* for protocol A if at least one of the parties is honest, i.e., if $\overline{A_1} = A_1$ or $\overline{A_2} = A_2$ holds. (Note that in the case where both parties are cheaters, no security is required.) By z, we denote some additional auxiliary input that can potentially be used by both parties. For instance, z could include information about previous executions of the protocol. Note, however, that an honest party never makes use of z.

The Ideal Model. In the *ideal* model, the two parties can make use of a trusted party to calculate the function. The algorithms $\overline{B_1}$ and $\overline{B_2}$ of the protocol $\overline{B} = (\overline{B_1}, \overline{B_2})$ receive the inputs x and y, respectively, and the auxiliary input z. They send values x' and y' to the trusted party, who sends them back the values u' and v'—satisfying $(u', v') = f(x', y')$. Finally, $\overline{B_1}$ and $\overline{B_2}$ output the values u and v. The two *honest* algorithms B_1 and B_2 always send $x' = x$ and $y' = y$ to the trusted party, and always output $u = u'$ and $v = v'$. Now, if $\overline{B} = (\overline{B_1}, \overline{B_2})$ is an admissible pair of algorithms for protocol $B = (B_1, B_2)$, the *joint execution of f under $\overline{B}$ in the ideal model*,

$$\mathbf{ideal}_{f,\overline{B}(z)}(x, y) ,$$

is the resulting output pair, given the inputs x and y and the auxiliary input z.

The Real Model. In the *real* model, the parties have to compute f by a protocol $\Pi = (A_1, A_2)$ without the help of a trusted party. Let $\overline{A} = (\overline{A_1}, \overline{A_2})$ be an admissible pair for A. Then the *joint execution of Π under $\overline{A}$ in the real model*,

$$\mathbf{real}_{\Pi,\overline{A}(z)}(x, y) ,$$

is the resulting output pair, given the inputs x and y and the auxiliary input z.

Perfect Security: "Real = Ideal". A protocol Π computes a function f *perfectly securely* if, intuitively speaking, every "real" cheater has an equally powerful counterpart in the ideal model. Definition 2 also applies to *reduction* protocols from one functionality to another; here, the algorithms are allowed to call an oracle which perfectly implements the given functionality.

Definition 2. A protocol Π *computes f perfectly securely* if for every admissible $\overline{A} = (\overline{A_1}, \overline{A_2})$ there exists an admissible $\overline{B} = (\overline{B_1}, \overline{B_2})$—as efficient as $\overline{A}$[1] and with identical set of honest players—such that for all $x \in \mathcal{X}$, $x \in \mathcal{Y}$, and $z \in \mathcal{Z}$,

$$\mathbf{real}_{\Pi,\overline{A}(z)}(x, y) \equiv \mathbf{ideal}_{f,\overline{B}(z)}(x, y)$$

holds, where $\equiv$ means that the distributions are identical.

3 Storing and Reversing Oblivious Transfer

Oblivious transfer protocols rely either on tools borrowed from public-key cryptography [19], [10] or on additional assumptions [3], [6], [20], [4], [23]. In the first case, we have to deal with relatively slow algorithms which may be the bottleneck of the protocol execution. In the second case, one depends on these additional

[1] The running time of $\overline{B}$ must by polynomial in the running time of $\overline{A}$.

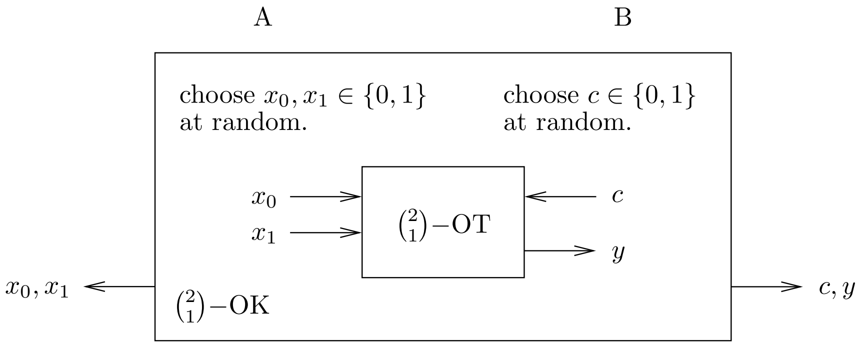
assumptions being present at the time of the execution of the protocol. In both cases it is, therefore, desirable to carry out as much of the computation as possible *in advance*, and to make the actual execution of oblivious transfer as fast and simple as possible, based on this pre-computation. Actually, almost the *entire* computation can be done beforehand: Protocols 1 and 2, proposed in [2], show how $\binom{2}{1}$−OT can be transformed into a so-called *oblivious key*, and *vice versa*.

An *oblivious key* is, intuitively speaking, the distribution that arises when A and B choose their inputs at random and execute $\binom{2}{1}$−OT.

Definition 3. By an *oblivious key*, $\binom{2}{1}$−OK, we denote the primitive where a sample of two random variables $U = (X_0, X_1)$ and $V = (C, Y)$ is given to A and B, respectively, where X_0, X_1, and C are independently and uniformly distributed bits, and where $Y = X_C$ holds.

Note that $\binom{2}{1}$−OK is a *key* for oblivious transfer in very much the same sense as a shared secret bit is an encryption key in the one-time pad.

Protocol 1: $\binom{2}{1}$−OK from $\binom{2}{1}$−OT



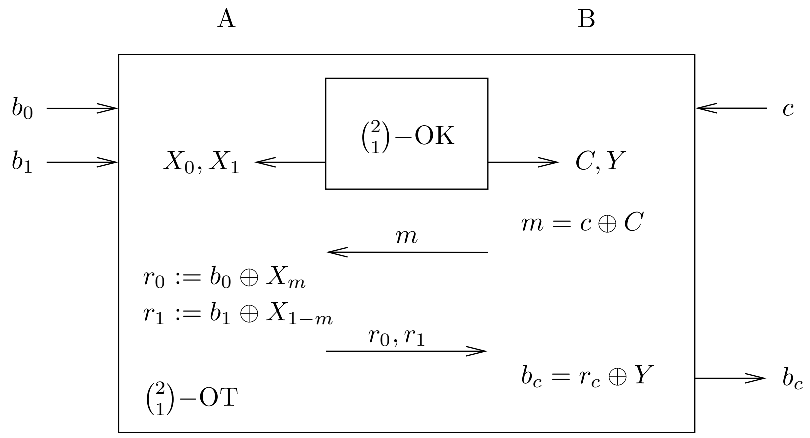
The proofs that Protocols 1 and 2 are perfect single-copy reductions between the primitives $\binom{2}{1}$−OT and $\binom{2}{1}$−OK are given in [2] (and straight-forward). Note that both Protocols 1 and 2 work in the *honest-but-curious* model, whereas their combination is even perfectly secure in the *malicious* model.

The distribution of $\binom{2}{1}$−OK is given and illustrated on the left hand side of Figure 2.

$$P_{UV}((x_0, x_1), (c, y)) = \begin{cases} 1/8 & \text{if } y = x_c \\ 0 & \text{otherwise} . \end{cases}$$

When the symbols of U and V are renamed in a suitable way, the distribution corresponds to the one arising when Shannon's so-called *"noisy-typewriter*

Protocol 2: $\binom{2}{1}-\mathrm{OT}$ from $\binom{2}{1}-\mathrm{OK}$



channel" [21] is used with random input (see on the right hand side of Figure 2). Obviously, this distribution is *symmetric*. On the other hand, $\binom{2}{1}-\mathrm{OK}$ is equivalent to $\binom{2}{1}-\mathrm{OT}$, which is, hence, symmetric as well: *A single* instance of $\binom{2}{1}-\mathrm{TO}$ allows for generating a realization of $\binom{2}{1}-\mathrm{OT}$. The reduction is not only single-copy but also *perfect*, i.e., unconditionally secure without any error. This solves an open problem posed in [9] in a very simple way. Lemma 1 shows how the values in the distribution of a $\binom{2}{1}-\mathrm{OK}$ must be renamed in order for the oblivious key to be reversed.

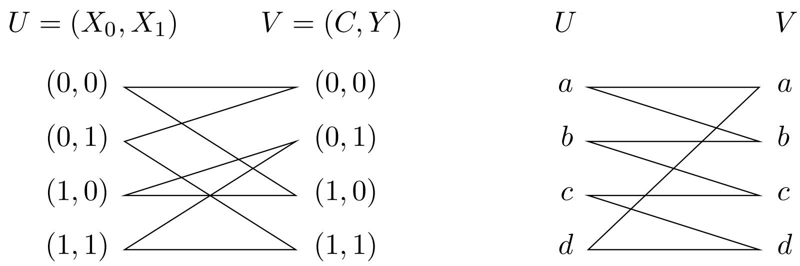


Fig. 2. Left hand side: The distribution of $\binom{2}{1}-\mathrm{OK}$. (Each edge is a possible combination with probability 1/8.) Right hand side: The distribution arising from the "noisy-typewriter channel." Obviously, the two distributions are equivalent.

Lemma 1. *Let X_0, X_1, C, and Y be binary random variables and let $(U, V) = ((X_0, X_1), (C, Y))$ be a $\binom{2}{1}-\mathrm{OK}$. Then $((\overline{X_0}, \overline{X_1}), (\overline{C}, \overline{Y})) := ((Y, C \oplus Y), (X_0 \oplus X_1, X_0))$ is a $\binom{2}{1}-\mathrm{OK}$ as well.*

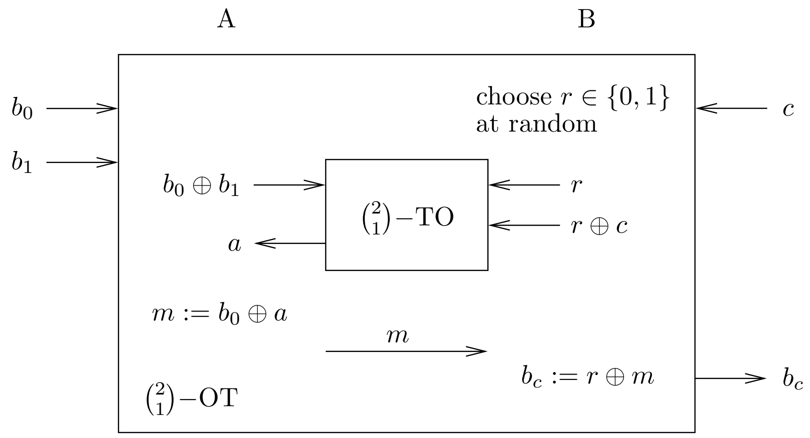
Proof. $\overline{Y} = X_0 = X_C \oplus C(X_0 \oplus X_1) = Y \oplus C(X_0 \oplus X_1) = \overline{X_0} \oplus (\overline{X_0} \oplus \overline{X_1})\overline{C} = \overline{X}_{\overline{C}}$.

A formal proof of the security of this transformation is omitted here. Intuitively, the privacy of both players is preserved since the ignorance of one player about the XOR of X_0 and X_1 is transformed into the ignorance of C, and *vice versa*.

4 Optimally Reversing Oblivious Transfer

The protocol outlined in the end of Section 3 requires *three* bits of additional communication. We present an even simpler protocol, Protocol 3, using only *one* bit of additional communication from A to B; this is optimal.

Protocol 3: $\binom{2}{1}$−OT from $\binom{2}{1}$−TO



Theorem 1. *Protocol 3 perfectly securely reduces* $\binom{2}{1}$−OT *to one realization of* $\binom{2}{1}$−TO.

Proof. Let first both parties be honest, i.e., $\overline{A} = (A_1, A_2)$ in Protocol 3. Then we have, for all $(b_0, b_1) \in \{0,1\}^2$, $c \in \{0,1\}$, and $z \in \mathcal{Z}$,

$$\begin{aligned}\mathbf{real}_{3,\overline{A}(z)}((b_0,b_1),c) &= (\perp, r \oplus (b_0 \oplus a)) \\ &= (\perp, b_0 \oplus (b_0 \oplus b_1)c) \\ &= (\perp, b_c) \\ &= \mathbf{ideal}_{\binom{2}{1}-\mathrm{OT},B(z)}((b_0,b_1),c)\ .\end{aligned}$$

Let now the first party be honest, i.e., $\overline{A} = (A_1, \overline{A_2})$. In the real model, $\overline{A_2}$ receives (c, z) and sends $(a_0, a_1) = a(c, z)$ to $\binom{2}{1}$−TO. Then he receives

$m = b_0 \oplus a_{b_0 \oplus b_1}$, and outputs $v(c, z, a_0, a_1, m)$. Let the adversary $\overline{B_2}$ in the ideal model be defined as follows: On inputs (c, z), he sends (c, z) to $\overline{A_2}$, and gets $(\overline{a_0}, \overline{a_1}) = a(c, z)$ back. He sends $\overline{c} := \overline{a_0} \oplus \overline{a_1}$ to $\binom{2}{1}-\mathrm{OT}$ and gets $b_{\overline{c}}$ back. Then he sends $\overline{m} := b_{\overline{c}} \oplus \overline{a_0}$ to $\overline{A_2}$, gets $\overline{v} = v(c, z, \overline{a_0}, \overline{a_1}, \overline{m})$ back and outputs $\overline{v}$.

Since

$$\overline{m} = \overline{a_0} \oplus b_{\overline{c}} = \overline{a_0} \oplus b_{\overline{a_0} \oplus \overline{a_1}} = b_0 \oplus \overline{a_0} \oplus (b_0 \oplus b_1)(\overline{a_0} \oplus \overline{a_1}) = b_0 \oplus \overline{a_{b_0 \oplus b_1}},$$

we have, for all $(b_0, b_1) \in \{0,1\}^2$, $c \in \{0,1\}$, and $z \in \mathcal{Z}$,

$$\begin{aligned}
\mathbf{real}_{3,\overline{A}(z)}((b_0, b_1), c) &= (\perp, v(c, z, a_0, a_1, m)) \\
&= (\perp, v(c, z, a_0, a_1, b_0 \oplus a_{b_0 \oplus b_1})) \\
&\equiv (\perp, v(c, z, \overline{a_0}, \overline{a_1}, b_0 \oplus \overline{a_{b_0 \oplus b_1}})) \\
&= (\perp, v(c, z, \overline{a_0}, \overline{a_1}, \overline{m})) \\
&= \mathbf{ideal}_{\binom{2}{1}-\mathrm{OT},(B_1,\overline{B_2})(z)}((b_0, b_1), c) \ .
\end{aligned}$$

Assume now that the second party is honest, i.e., $\overline{A} = (\overline{A_1}, A_2)$. In the real model, $\overline{A_1}$ receives $((b_0, b_1), z)$ and sends $d = d((b_0, b_1), z)$ to $\binom{2}{1}-\mathrm{TO}$, which returns $l = r \oplus dc$. Then, he sends $m = m((b_0, b_1), z, l)$ to A_2 and outputs $u((b_0, b_1), z, d, l, m)$. Let the adversary $\overline{B_1}$ in the ideal model be defined as follows: On inputs $((b_0, b_1), z)$, $\overline{B_1}$ sends $((b_0, b_1), z)$ to $\overline{A_1}$ and gets $\overline{d} = d((b_0, b_1), z)$ back. He chooses $\overline{l}$ uniformly at random and sends it to $\overline{A_1}$, who sends $\overline{m} = m((b_0, b_1), z, \overline{l})$ and $\overline{u} = u((b_0, b_1), z, \overline{d}, \overline{l}, \overline{m})$ back. He sends $(\overline{l} \oplus \overline{m}, \overline{l} \oplus \overline{m} \oplus d)$ to $\binom{2}{1}-\mathrm{OT}$ and outputs $\overline{u}$.

The honest player will output $\overline{l} \oplus \overline{m} \oplus c\overline{d}$. Since $l = r \oplus dc$ and r is uniform and independent of everything else, l is uniform and independent as well, which means that it has the same joint distribution as $\overline{l}$ with everything else. Therefore we have, for all $(b_0, b_1) \in \{0,1\}^2$, $c \in \{0,1\}$, and $z \in \mathcal{Z}$,

$$\begin{aligned}
\mathbf{real}_{3,\overline{A}(z)}((b_0, b_1), c) &= (u((b_0, b_1), z, d, l, m), r \oplus m) \\
&= (u((b_0, b_1), z, d, l, m, l \oplus dc \oplus m) \\
&\equiv (u((b_0, b_1), z, \overline{d}, \overline{l}, \overline{m}, \overline{l} \oplus \overline{d}c \oplus \overline{m}) \\
&= \mathbf{ideal}_{\binom{2}{1}-\mathrm{OT},\overline{B}(z)}((b_0, b_1), c) \ .
\end{aligned}$$

Obviously, the simulated adversary is as efficient as the real adversary. □

Our protocol is optimal: First of all, since it is impossible to construct unconditionally secure oblivious transfer from scratch, using *a single* instance of $\binom{2}{1}-\mathrm{TO}$ is optimal. Since $\binom{2}{1}-\mathrm{TO}$ does not allow any communication from Bob to Alice, but $\binom{2}{1}-\mathrm{OT}$ does allow one bit of communication, any protocol must communicate at least one bit. Furthermore, there cannot exist a protocol where Bob does not use any randomness, because then his inputs to $\binom{2}{1}-\mathrm{TO}$ would be deterministic functions of c. These functions could not both be constant, since then the output of $\binom{2}{1}-\mathrm{TO}$ would not depend on c and be useless, and therefore no oblivious transfer would be possible. But if the functions are not constant, A is able to obtain information about c.

5 Oblivious Linear-Function Evaluation

In contrast to $\binom{2}{1}-\mathrm{OT}$, all the other forms of oblivious transfer cannot be reversed without loss, i.e., in the perfect single-copy sense of Sections 3 and 4. This can easily be seen from the *monotones*, defined in [24]: A primitive can only be reversed without loss if

$$H(Y \searrow X|X) = H(X \searrow Y|Y),$$

and $\binom{2}{1}-\mathrm{OT}$ is the only example of $\binom{n}{k}-\mathrm{OT}^l$ having this property.

In this section, we present another natural generalization of oblivious transfer to strings that *can* be reversed perfectly: *oblivious linear-function evaluation over* $GF(q)$ or $GF(q)-\mathrm{OLFE}$ for short. Roughly speaking, the sender's input is a linear function $f : x \mapsto y = a_0 + a_1 x$, where $a_0, a_1, x, y \in GF(q)$, and the receiver's input is an argument $x \in GF(q)$ for which he then learns the evaluation of the function, $y = f(x)$ (see Figure 3). $GF(q)-\mathrm{OLFE}$ is a special case of oblivious polynomial evaluation [17]. It can easily be verified that $GF(2)-\mathrm{OLFE}$ is equivalent to $\binom{2}{1}-\mathrm{OT}$. Furthermore, [20] shows that with one instance of $GF(q)-\mathrm{OLFE}$ a very simple commitment scheme can be implemented, which allows to commit to a value $x \in GF(q)$. The scheme is perfectly hiding and $1/q$-binding.

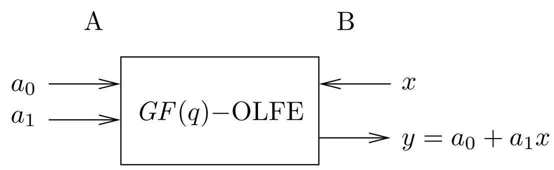
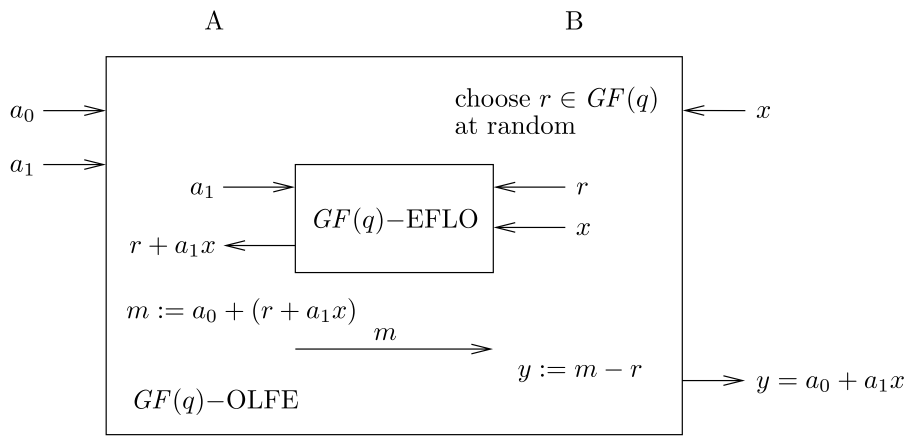


Fig. 3. Oblivious linear-function evaluation over $GF(q)$.

The protocols of Sections 3 and 4 generalize to $GF(q)-\mathrm{OLFE}$ in a straightforward way: $GF(q)-\mathrm{OLFE}$ is, as oblivious transfer, equivalent to a non-interactive key—and can, therefore, be stored in the same sense. Moreover, this key is, as $\binom{2}{1}-\mathrm{OK}$, symmetric. Hence, $GF(q)-\mathrm{OLFE}$ from A to B can be reduced to $GF(q)-\mathrm{OLFE}$ from B to A—$GF(q)-\mathrm{EFLO}$ for short—in a perfect and single-copy sense. Protocol 4 is, in addition, optimal with respect to the required communication.

Protocol 4: $GF(q)-$OLFE from $GF(q)-$EFLO



6 Concluding Remarks

We have shown that chosen one-out-of-two bit oblivious transfer can be optimally reversed *very* easily. Furthermore, we have have presented a more general primitive with the same property: oblivious linear-function evaluation.

Acknowledgments

This work was carried out while both authors were with Université de Montréal, Canada. This research was supported by Canada's NSERC, Québec's FQRNT, and Switzerland's SNF.

References

1. D. Beaver, Foundations of Secure Interactive Computing, *Advances in Cryptology—Proceedings of CRYPTO '91*, LNCS, Vol. 576, pp. 377–391, Springer-Verlag, 1992.
2. D. Beaver, Precomputing oblivious transfer, *Advances in Cryptology—Proceedings of CRYPTO '95*, LNCS, Vol. 963, pp. 97–109, Springer-Verlag, 1995.
3. C. H. Bennett, G. Brassard, C. Crépeau, and H. Skubiszewska, Practical quantum oblivious transfer, *Advances in Cryptology—Proceedings of EUROCRYPT '91*, LNCS, Vol. 576, pp. 351–366, Springer-Verlag, 1992.
4. G. Brassard, C. Crépeau, and S. Wolf, Oblivious transfers and privacy amplification, *Journal of Cryptology*, Vol. 16, No. 4, pp. 219–237, 2003.
5. C. Crépeau, *Correct and private reductions among oblivious transfers*, Ph. D. Thesis, Massachusetts Institute of Technology, 1990.
6. C. Crépeau, Efficient cryptographic protocols based on noisy channels, *Advances in Cryptology—Proceedings of CRYPTO '97*, LNCS, Vol. 1233, pp. 306–317, Springer-Verlag, 1997.

7. C. Crépeau and J. Kilian, Achieving oblivious transfer using weakened security assumptions, *Proceedings of the* 28^{th} *Symposium on Foundations of Computer Science (FOCS '88)*, pp. 42–52, IEEE, 1988.
8. C. Crépeau, K. Morozov, and S. Wolf, Efficient unconditional oblivious transfer from almost any noisy channel, *Proceedings of Fourth Conference on Security in Communication Networks (SCN) '04*, LNCS, Springer-Verlag, 2004.
9. C. Crépeau and M. Sántha, On the reversibility of oblivious transfer, *Advances in Cryptology—Proceedings of EUROCRYPT '91*, LNCS, Vol. 547, pp. 106–113, Springer-Verlag, 1991.
10. S. Even, O. Goldreich, and A. Lempel, A randomized protocol for signing contracts, *Communications of the ACM*, Vol. 28, No. 6, pp. 637–647, 1985.
11. O. Goldreich, *Foundations of Cryptography, Volume II: Basic Applications*, Cambridge University Press, 2004.
12. H. Imai, J. Müller-Quade, A. Nascimento, and A. Winter, Rates for bit commitment and coin tossing from noisy correlation, *Proceedings of the IEEE International Symposium on Information Theory (ISIT) '04*, IEEE, 2004.
13. H. Imai, A. Nascimento, and A. Winter, Oblivious transfer from any genuine noise, *unpublished manuscript*, 2004.
14. J. Kilian, Founding cryptography on oblivious transfer, *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing (STOC '88)*, pp. 20–31, 1988.
15. D. Mayers, Unconditionally secure quantum bit commitment is impossible, *Phys. Rev. Lett.*, Vol. 78, pp. 3414–3417, 1997.
16. S. Micali and P. Rogaway, Secure computation, *Advances in Cryptology—Proceedings of CRYPTO '91*, LNCS, Vol. 576, pp. 392–404, Springer-Verlag, 1992.
17. M. Naor and B. Pinkas. Oblivious transfer and polynomial evaluation, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing (STOC '99)*, pp. 245-354, 1999.
18. R. Ostrovsky, R. Venkatesan, and M. Yung, Fair games against an all-powerful adversary, *AMS DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, Vol. 13, pp. 155–169, 1990.
19. M. Rabin, How to exchange secrets by oblivious transfer, *Technical Report TR-81, Harvard Aiken Computation Laboratory*, 1981.
20. R. L. Rivest, Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer, *unpublished manuscript*, 1999.
21. C. E. Shannon, A mathematical theory of communication, *Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656, 1948.
22. A. Winter, A. Nascimento, and H. Imai, Commitment capacity of discrete memoryless channels, *Cryptography and Coding*, LNCS, Vol. 2898, pp. 35–51, Springer-Verlag, 2003.
23. S. Wolf and J. Wullschleger, Zero-error information and applications in cryptography, *Information Theory Workshop (ITW) '04*, IEEE, 2004.
24. S. Wolf and J. Wullschleger. New monotones and lower bounds in unconditional two-party computation. In *Advances in Cryptology—Proceedings of CRYPTO '05*, LNCS, Vol. 3621, pp. 467–477, Springer-Verlag, 2005.
25. A. D. Wyner, The wire-tap channel, *Bell System Technical Journal*, Vol. 54, No. 8, pp. 1355–1387, 1975.
26. R. W. Yeung, A new outlook on Shannon's information measures, *IEEE Transactions on Information Theory*, Vol. 37, No. 3, pp. 466–474, 1991.