Secure Two-party Computation over a Z-channel

Paolo Palmieri and Olivier Pereira

Université catholique de Louvain
UCL Crypto Group, ICTEAM Institute
Place du Levant 3, B-1348 Louvain-la-Neuve, Belgium
{paolo.palmieri,olivier.pereira}@uclouvain.be

Abstract. In secure two-party computation, two mutually distrusting parties are interested in jointly computing a function, while preserving the privacy of their respective inputs. However, when communicating over a clear channel, security against computationally unbounded adversaries is impossible. Thus is the importance of noisy channels, over which we can build Oblivious Transfer (OT), a fundamental primitive in cryptography and the basic building block for any secure multi-party computation. The noisy channels commonly used in current constructions are mostly derived from the Binary Symmetric Channel (BSC), which is modified to extend the capabilities of an attacker. Still, these constructions are based on very strong assumptions, in particular on the error probability, which makes them hard to implement.
In this paper, we provide a protocol achieving oblivious transfer over a Z-channel, a natural channel model in various contexts, ranging from optical to covert communication. The protocol proves to be particularly efficient for a large range of error probabilities p (e.g., for $0.17 \leq p \leq 0.29$ when a security parameter $\varepsilon = 10^{-9}$ is chosen), where it requires a limited amount of data to be sent through the channel. Our construction also proves to offer security against unfair adversaries, who are able to select the channel probability within a fixed range. We provide coding schemes that can further increase the efficiency of the protocol for probabilities distant from the range mentioned above, and also allow the use of a Z-channel with an error probability greater than 0.5. The flexibility and the efficiency of the construction make an actual implementation of oblivious transfer a more realistic prospect.

Key words: Oblivious transfer, secure multi-party computation, information theoretic security, cryptography on noisy channels.

1 Introduction

Oblivious Transfer (OT), introduced by Rabin in 1981 [14], is a primitive of primary importance in the field of secure multi-party computation and, more generally, cryptography: any secure computation can be built on top of a secure OT protocol [10]. In this paper we present a protocol achieving OT on a *Z-channel*, a communication channel that, while being closely related to the most commonly used ones, has never been used for this purpose. We also show

the greater flexibility our construction has in comparison to current protocols, being able to accept any error probability p, and allowing an adversary large control over p itself.

Secure multi-party computation deals with the problem of mutually distrusting players interested in jointly compute a function, while preserving the privacy of their respective inputs. In the information-theoretic setting, where protocols must be secure against computationally unbounded adversaries, security cannot be generally achieved without any further assumption. However, multi-party computation can be achieved when some form of noise in the communication is available, which, in turn, allows for the construction of oblivious transfer. Since OT can be built using almost any noisy channel [5], the most interesting research question is what form of noise offers the best flexibility and efficiency, and could therefore be used in the prospect of an actual implementation.

The two players involved in an oblivious transfer protocol are generally divided into a sender and a receiver. The sender knows a set of secret input values, and is interested in communicating to the receiver a smaller subset, selected according to the latter's choice. After a successful execution of the protocol, the receiver learns the chosen inputs but gets no information about the others, while the sender remains oblivious of the receiver's selection.

The first protocol to implement oblivious transfer on a noisy channel was provided by Crépeau and Kilian in 1988 [4]. The construction is based on the *binary symmetric channel* (BSC), which flips with a fixed probability each bit sent though it. Security is guaranteed by privacy amplification, while the use of error correction codes assures the correctness of the results. Results were later improved and generalized in [3, 5, 12]. However, strong requirements are imposed by the construction: the players must have perfect knowledge of the statistics of the channel, such as the error probability, which can not change during protocol execution. Damgård et al. tried to weaken these requirements by introducing the *unfair noisy channels* in 1999 [7]. The name comes from the "unfair" advantage given to an adversary, who is able to arbitrarily choose the error probability of the channel within a certain range. Results were improved by widening the range of acceptable probabilities in a following paper in 2004 [6]. Wullschleger further extended the concept of unfair noisy channel by introducing two new channel models, the *weak erasure channel* and the *weak binary symmetric channel* [16]. The two new primitives aim to be more general by lessening the security assumptions. For instance, they take into account the possibility, for a dishonest sender, to learn with a small probability if a bit was or not affected by the channel noise (being flipped in the case of the weak BSC, or lost in the case of the weak erasure channel).

Although it has been shown that oblivious transfer can be built from almost any noisy channel, we still lack a clear understanding of what properties a noisy channel needs to be best suited for efficient protocols. This knowledge is of primary importance if we are to build better protocols and thus achieve actual implementation.

1.1 The Z-channel

In a 1980 correspondence on the IEEE Transactions on Information Theory, Golomb succinctly describes the mutual information and capacity of the Z-channel [8]. A Z-channel is a discrete memoryless channel with two input symbols (x_1, x_2) and two output symbols (y_1, y_2). The noise in the channel is determined by its error probability p. The conditional probabilities of receiving a given output based on a given input are expressed by the following matrix:

$$\begin{pmatrix} P(y_1|x_1) & P(y_2|x_1) \\ P(y_1|x_2) & P(y_2|x_2) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ p & 1-p \end{pmatrix} .$$

In the following, we use the name Z-channel to refer to a binary Z-channel, where $(x_1, x_2) = (0, 1)$ and $(y_1, y_2) = (0, 1)$. Using these symbols, the behavior of the Z-channel closely resemble that of a binary symmetric channel, except for the fact that the noise is affecting the communication asymmetrically (see Fig. 1). In practice, we can transmit a "0" through the channel noiselessly, while sending a "1" we have a probability p that it will be received as a "0".

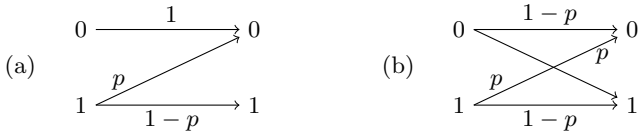

Fig. 1. A Z-channel (a) and a standard Binary Symmetric Channel (b).

We observe that using the correct encoding, a Z-channel with crossover probability $p = \frac{1}{2}$ is in fact equivalent to a Rabin OT (which in turn is equivalent to an erasure channel with $p = \frac{1}{2}$), where the sender sends a secret bit b, which is received with probability $\frac{1}{2}$ and lost otherwise, and remains oblivious as to whether the bit reached the receiver or not. We can achieve this by sending the bit string "01"for $b = 0$ and "10" for $b = 1$. The receiver is able to decode the bit if the string is received correctly, but learns no information about it if she receives the string "00" instead.

The Z-channel appears as a natural model of various types of communications, from optical communications [1], to various forms of covert communications [11].

1.2 Contribution

While we know that noise is essential to achieve security in multi-party computation, we still lack the understanding of how the properties of a particular noisy channel affect the efficiency and security of oblivious transfer protocols. This paper aims at showing how a careful selection of a noisy channel is needed in order to achieve secure multi-party computation efficiently.

To do so we provide a protocol implementing oblivious transfer on a simple channel model, the Z-channel. This channel has interesting properties: contrary to the binary symmetric and erasure channels, only part of the input symbols are affected by the noise, allowing the receiver to interpret some (but not all) of the output symbols with certainty. For the other symbols, the receiver has no way to recover the original information with certainty. This provides us with the ambiguity needed for the security of the construction.

We show how these channel properties can be exploited to build an oblivious transfer protocol that can accommodate any non-zero value of the flipping probability p of the Z-channel and, interestingly, can offer security as long as this probability is known to lie in any fixed range. This contrasts with the case of the unfair BSC (UNC) channel studied by Damgård et al. [7, 6], for which it is known that OT is impossible to achieve as soon as the range of acceptable flipping probabilities increases too much. In general, we show that an unfair Z-channel, which allows the adversary to choose the probability of the channel within a range, behaves much better than an unfair BSC: oblivious transfer can be achieved for any fixed range, and our protocol is efficient for ranges larger than the possibility range of a UNC.

Following the terminology of Imai et al. [9], our protocol achieves an oblivious transfer rate of $\frac{1}{4}$. We also demonstrate how this efficiency and flexibility can be further improved for specific channel parameters and concrete security bounds, through the use of different coding strategies and protocol modifications.

1.3 Outline of the Paper

Section 2 contains some useful definitions and preliminary notions that are needed in the following. In particular, we give a security definition for oblivious transfer. Section 3 describes the basic version of our protocol for achieving OT on a Z-channel, and proves the security of this protocol in the semi-honest model. In Section 4, we introduce the unfair version of the Z-channel, and we analyze the efficiency of our construction. We also demonstrate how different coding strategies and protocol modifications can be used to improve the efficiency of concrete instances of our protocol.

2 Preliminaries

Many different flavors of oblivious transfer exist, and they have all been proved equivalent by Crépeau [2]. In the following, when using the name oblivious transfer, we will be referring to a *binary 1-out-of-2 oblivious transfer* protocol, where a sender, Sam, has two bits b_0 and b_1, while a receiver, Rachel, has a choice bit c. After a successful execution of such a protocol, three conditions must be satisfied: the receiver party knows the value of the selected bit b_c (correctness); the receiver party learns nothing about the value of the other bit b_{1-c} (security for Sam); the sender party learns nothing about the choice bit c (security for Rachel) [4].

A useful way of measuring the knowledge an adversary has on a secret bit of information is the *prediction advantage*. Prediction advantage measures the advantage the adversary has in guessing the secret bit by using all the available information.

Definition 1. [15] *Let P_{XY} be a distribution over $\{0,1\} \times \mathcal{Y}$. The* maximal bit prediction advantage *of X from Y for a function f is*

$$\mathrm{PredAdv}\left(X \mid Y\right) = 2 \cdot \max_{f} \Pr\left[f\left(Y\right) = X\right] - 1 \ . \tag{1}$$

All the information that is available to a player during an execution of the protocol is called the *view* of the player.

2.1 A Security Definition for Oblivious Transfer

Using the concept of prediction advantage, the three conditions that form the security of an oblivious transfer protocol can be formally defined, leading to a security definition for OT. Such a definition can be found in [13]. We will use it in the following to prove the security of our construction.

Definition 2. [13] *A protocol Π between a sender and a receiver, where the sender inputs $(b_0, b_1) \in \{0,1\}$ and outputs nothing, and the receiver inputs $c \in \{0,1\}$ and outputs S, securely computes 1-2 oblivious transfer with an error of at most ε, assuming that U and V represent the sender and receiver views respectively, if the following conditions are satisfied:*

- (Correctness) *If both players are honest, we have*

$$Pr\left[S = b_c\right] \geq 1 - \varepsilon \ . \tag{2}$$

- (Security for Sam) *For an honest sender and an honest (but curious) receiver we have*

$$\mathrm{PredAdv}\left(b_{1-c} \mid V, c\right) \leq \varepsilon \ . \tag{3}$$

- (Security for Rachel) *For an honest receiver and an honest (but curious) sender we have*

$$\mathrm{PredAdv}\left(c \mid U, b_0, b_1\right) \leq \varepsilon \ . \tag{4}$$

3 Oblivious Transfer over a Z-channel

Our construction differs from standard oblivious transfer protocols for the binary symmetric channel for the fact that it does not require the use of error correcting codes (ECC's). This is due to the fact that some output symbols can always be interpreted correctly, thanks to the Z-channel properties (a "1" output symbol always come from a "1" input). The ambiguity needed to assure the security of the construction is however guaranteed by the fact that we cannot correct errors in other output symbols (a "0" output symbol can come either from a "0" or a "1" input).

3.1 Protocol

The protocol is a sequence of three different stages. The first stage is called *precomputation*, and is performed by the sender. During this stage, the sender selects, according to some prescribed distribution, a set of bit pairs to be sent on the Z-channel.

The communication between the sender and the receiver takes place in the second stage (*communication*). The precomputed set is sent to the receiver through the Z-channel. The interaction then proceeds by exchanging on a clear channel the information needed to construct the encoded version of the secret bits b_0 and b_1, that are subsequently sent to the receiver.

During the third and last stage, called *postcomputation*, the receiver computes the value of the chosen bit b_c.

Protocol 1. The three phases described below happen sequentially.

Precomputation The sender Sam selects n pairs of bits $s_i := (c_i, c'_i)$ such that $c_i \oplus c'_i = 1$ for all $i \in [1, n]$. Knowledge of the value of n is shared between the parties.

Communication The sender Sam and the receiver Rachel can communicate over a Z-channel with error probability p and over a clear channel.

1. Sam sends the pairs $s_1, \dots s_n$ to Rachel through the Z-channel.
2. Rachel receives a sequence of n pairs of bits $r_i := (d_i, d'_i)$, with $i \in [1, n]$, somehow similar to the s_i pairs.
3. Rachel arbitrarily selects two sequences I_c and I_{1-c}, where c is her choice bit, each composed of $\frac{n}{2}$ unique indices $i \in [1, n]$, where $i \in I_c$ if and only if $d_i \neq d'_i$, that is, if the pair (c_i, c'_i) has not been modified during the transmission. When she selected enough elements for I_c, Rachel puts all the other indices i in I_{1-c}, regardless of the content of (d_i, d'_i) and sends the two sets back to Sam on a clear channel. If instead there are less than $\frac{n}{2}$ indices that she can put in I_c, she aborts the protocol.
4. Sam receives the two sequences of indices (I_0, I_1) and builds two strings e_0, e_1 such that the i-th bit of e_b has value 0 if and only if $\left(c_{I_{b[i]}}, c'_{I_{b[i]}}\right) = (0, 1)$ where $I_{b[i]}$ is the i-th index in I_b. Then Sam chooses two universal hash functions h_0 and h_1 whose output is 1-bit long for any input. He computes

$$f_0 = (b_0 \oplus h_0(e_0)) \quad , \quad f_1 = (b_1 \oplus h_1(e_1)) \quad , \tag{5}$$

   and sends f_0, f_1, h_0, h_1 to Rachel via a clear channel.

Postcomputation The receiver Rachel builds the string e_c using the same procedure used by Sam when building (e_0, e_1) but using $\left(d_{I_{c[i]}}, d'_{I_{c[i]}}\right)$ instead of $\left(c_{I_{b[i]}}, c'_{I_{b[i]}}\right)$. Then she computes

$$b_c = (f_c \oplus h_c(e_c)) \quad . \tag{6}$$

We observe that this protocol follows the general pattern of OT-protocols from noisy channels [3, 5, 12]. First, it somehow builds an erasure channel (this is the purpose of the precomputation stage). Then this erasure channel is used a number of times to realize OT.

3.2 Security in the Semi-honest Scenario

In the *semi-honest model* the players act in a honest-but-curious way. In practice, they follow the protocol, but try to use all the information they can get during the protocol execution to get extra knowledge. We can also say that in a semi-honest scenario, the adversary is *passive*: she follows the protocol, but outputs her entire view [15].

We now show the security of the protocol when the probability p is in the interval $\left(0, \frac{1}{2}\right)$. In the next section, we will show how to relax this requirement in order to deal with any probability p in the interval $(0, 1)$.

Theorem 1. *The protocol described in Section 3.1 securely computes 1-2 oblivious transfer with error probability ε when it is executed on a Z-channel with error probability p, where $0 < p < \frac{1}{2}$, and with the security parameter n satisfying:*

$$n > \max\left(\frac{-2\log(\varepsilon)}{(1-2p)^2}, \frac{\log\left(\frac{\varepsilon}{2}\right)}{\log\left(1-\frac{p}{2}\right)}\right) \quad . \tag{7}$$

Proof. We prove that our construction is secure by showing that each of the three security conditions for an oblivious transfer protocol is satisfied.

Correctness When transmitted through a binary Z-channel, each pair (c_i, c_i') is received correctly except with probability p, and Rachel is able to decide whether she received that pair correctly: the pair has been modified by the channel only if $c_i = c_i' = 0$.

The protocol is correct, that is, Rachel is able to build the sequence I_c correctly if at least $\frac{n}{2}$ pairs have been delivered without errors; we call this event $\mathsf{Correct}$. This $\mathsf{Correct}$ event happens with the following probability:

$$\Pr[\mathsf{Correct}] = \sum_{k=\frac{n}{2}}^{n} \binom{n}{k} (1-p)^k p^{n-k} \geq 1 - e^{-2n\left(\frac{1}{2}-p\right)^2} \quad , \tag{8}$$

where the inequality follows from the Chernoff bound.

Security for Sam The aim of a curious Rachel is to learn the value of b_{1-c}. Let us call this event $\mathsf{Success}$. Rachel has two ways of achieving it: by decoding e_{1-c} on the correct inputs (let us call this event $\mathsf{DecodeE}$), or by not doing so. So, we have that $\Pr[\mathsf{Success}] = \Pr[\mathsf{Success} \wedge \mathsf{DecodeE}] + \Pr[\mathsf{Success} \wedge \neg\mathsf{DecodeE}]$. The latter probability is upper-bounded by $\frac{1}{2}$, due to the properties of a universal hash function, while the former is upper-bounded by $\Pr[\mathsf{DecodeE}]$. In the following, we evaluate that probability.

Rachel needs to learn the value of all the bits in e_{1-c}. For each bit, she has two ways of doing so:

1. By computing the value. This is possible if $d_i \neq d'_i$ for the pair of bits r_i corresponding to the bit in e_{1-c} she wants to learn. We call this event NoAmbiguity.
2. By guessing the value. This is necessary if she cannot directly compute it, that is, if $d_i = d'_i = 0$ for the corresponding pair of bits. Let us call this event Ambiguity, and note that it is complementary to the event NoAmbiguity.

The first case happens with probability $(1-p)$. Therefore, Rachel will be able to decode the whole string e_{1-c} without any guessing with probability $(1-p)^n$.

In the second case, which happens with probability $\Pr[\mathsf{Ambiguity}] = 1 - \Pr[\mathsf{NoAmbiguity}]$, Rachel has no information about which pair (c_i, c'_i) of weight 1 was sent. Therefore Rachel has to guess the value of the bit by tossing a coin, with a probability to succeed equal to $\frac{1}{2}$.

Overall, for any bit pair r_i she will select the correct value for the bit in e_{1-c} with probability $\left(\frac{p}{2} + (1-p)\right) = 1 - \frac{p}{2}$. Since the hash function can be correctly evaluated only if all the guesses are correct, we have

$$\Pr[\mathsf{DecodeE}] = \left(1 - \frac{p}{2}\right)^n \quad . \tag{9}$$

Therefore we have

$$\Pr[\mathsf{Success}] \leq \frac{1}{2} + \left(1 - \frac{p}{2}\right)^n \quad . \tag{10}$$

Security for Rachel The Z-channel does not give any feedback to the sender as to what errors it introduces in a message transmitted through it. Since Rachel distinguishes the sets I_c and I_{1-c} from the bits flipped by the channel, from Sam's point of view the distribution of (I_0, I_1) is independent of c.

Combining the two results of the correctness and security for Sam sections of the proof by extracting n in the two inequalities, and using the definition of prediction advantage, we obtain the two arguments of the maximum function in the theorem statement. □

We observe that the bounds provided here are also valid for similar channels on which OT is built through a simple reduction to the binary erasure channel (e.g., the *binary discrete-time delaying channel* considered in [13]).

4 Efficiency and Resistance to Unfair Adversaries

Our construction exhibits a particularly low sensitivity of the n parameter to the value of the probability p for a given security bound ε, in a wide range of values of p. We believe that this is a very useful feature of using a Z-channel for realizing OT, as the precise channel characteristics are often difficult to evaluate when communication is achieved between a sender and a receiver who do not trust each other.

A concrete depiction of the bounds of Theorem 1 is provided in Fig. 2. Our protocol is particularly efficient when the probability p is around 0.25 (the optimal value has little sensitivity to fluctuations of the ε parameter: it ranges from 0.2486 to 0.2462 when ε ranges from 10^{-6} to 10^{-15}). [1] Just 163 bit pairs are sufficient to guarantee a security of $\varepsilon = 10^{-9}$ at the optimal probability $p = 0.2473$. More importantly, the graph in Fig. 2 shows that the number of bit pairs n can be kept low for large ranges of p and reasonable values of ε. For instance, for $\varepsilon = 10^{-9}$, transmitting 250 pairs of bits on the Z-channel is enough to realize OT for $0.17 \leq p \leq 0.29$, and the even larger range of $0.04 \leq p \leq 0.40$ only requires $n = 1060$.

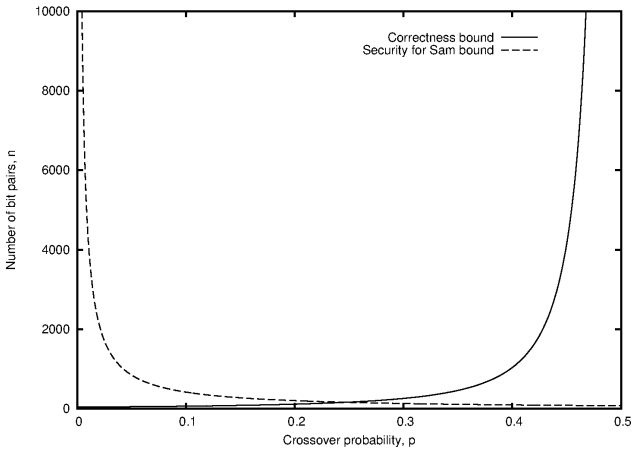

Fig. 2. n as a function of p for $\varepsilon = 10^{-9}$.

4.1 Unfair Z-channel

The large acceptable ranges discussed above also imply that our construction behaves considerably well when the adversary is given the advantage of choosing the channel probability within a certain range. Following the definition in [7], an unfair Z-channel with parameters γ and δ, where $\gamma, \delta \leq \frac{1}{2}$, is a Z-channel with an error probability $p \in [\gamma, \delta]$ that is set by the adversary but is not known by the honest players, who have access to the values of γ and δ only. Therefore, a protocol implementing OT over an unfair Z-channel must work for any p in this range.

The concept of unfair Z-channel gives the graph in Fig. 2 a new meaning: the area comprised between the two functions is also the largest acceptable range for

[1] Note that these values are actually as precise as the Chernoff bound that we used in the proof on Theorem 1.

a given n, where the security for Sam bound is equivalent to the minimum γ, and the correctness bound is the maximum δ. This result is particularly interesting if compared to the best known achievable ranges for the unfair BSC (UNC) [6]. While in the case of the UNC oblivious transfer can not be achieved as soon as the difference between δ and γ becomes too large, namely if $\delta \geq 2\gamma\,(1-\gamma)$, in the unfair Z-channel OT is possible for any fixed range. The largest possible range for the UNC has a width of 0.125, with $\gamma = 0.25$ and $\delta = 0.375$, but such an interval can not be achieved by any of the current protocols. As we have seen before, for the relatively small $n = 1060$ our construction offers security for a maximum range as wide as 0.36.

4.2 Efficiency

For input messages longer than one bit, the efficiency of the protocol proposed in Section 3.1 can be increased by using privacy amplification techniques (see Crépeau [3] for instance). Following the terminology of Imai et al. [9], we can observe that, by letting n grow to infinity on a Z-channel with probability $p = \frac{1}{2}$, the I_{1-c} sequence will only contain indices for which the corresponding (c_i, c'_i) pair is unknown from r_i. As a result, it is possible to realize the oblivious transfer of messages of $\frac{n}{2}$ bits by exchanging $2n$ bits on the Z-channel, achieving an oblivious transfer rate of $\frac{1}{4}$.

This rate is however an asymptotic notion, and is only useful for the exchange of longer messages. We now investigate improvements on the efficiency of our protocol for concrete parameters (i.e., concrete values of ε and n).

Usually, it is not possible to choose the characteristics of the Z-channel that one uses. This suggests the use of coding strategies that could let the parties emulate a Z-channel with probability $p \approx 0.25$ from a Z-channel exhibiting a very different crossover probability.

Emulating a Z channel with lower crossover probability. Suppose first that a Z-channel with crossover probability $p \gg 0.25$ is available (including the case where $p > 0.5$, as long as $p < 1$). Using a simple repetition code is actually enough to emulate a Z-channel with lower crossover probability.

Suppose indeed that, instead of selecting c_i and c'_i as bits, we select them as sequences of m repeated bits. A sequence of m zeros will always be delivered as it is, while a sequence of m ones can be delivered as an arbitrary bit string of length m. However, as long as that bit string contains a single 1, the receiver can be sure that a sequence of m "1" was actually sent. So, ambiguity happens only if all the m "1" are flipped during transmission. This happens with probability p^m, which is always smaller than p for $m > 1$.

As a result, a m repetition code allows emulating a Z-channel with crossover probability p^m from a Z-channel with crossover probability p. This also provides a way to use our protocol on a Z-channel with $p > \frac{1}{2}$.

Emulating a Z channel with larger crossover probability. Suppose now that a Z-channel with crossover probability $p \ll 0.25$ is available. We can then decide to

encode each "1" bit to be sent on the Z-channel as a sequence of l "1", and each "0" bit as a sequence of $(l-1)$ "1" with a "0" placed in an arbitrary position unknown to the receiver. Now, a "1" will only be recognized as it if none of its l bits is modified while going through the channel. This happens with probability $(1-p)^l$, which allows emulating a Z-channel with probability $1-(1-p)^l$, which is larger than p for any $l > 1$.

These two codes can always be combined in order to emulate a Z-channel with the desired crossover probability with arbitrary precision. For instance, we could choose to encode a "1" as a sequence of $m \cdot l$ "1", and a "0" as a sequence of $l-1$ blocks of m "1" among which a block of m "0" is inserted, which would provide a Z-channel with crossover probability $1-(1-p^m)^l$.

These two codes however come at the price of each time increasing the number of bits to be transmitted on the Z-channel by a factor $m \cdot l$, and one may wonder whether this is actually compensated by the possibility to use a lower number n of pairs. This actually becomes the case when the probability p leaves the efficient range mentioned above.

Let us consider for instance a Z-channel with crossover probability $p = 0.4$. A number of pairs $n = 1037$ is needed to reach a security margin $\varepsilon = 10^{-9}$ with our basic protocol. However, applying our first code with $m = 2$ allows emulating a Z-channel with crossover probability $p^2 = 0.16$, for which a security parameter $n = 246$ is needed, improving the efficiency of our protocol by a factor larger than 2.

In general, for a Z-channel with probability p, we have a minimum number of bit pairs n for a given security bound ε. It is convenient to use an m-l-coding strategy if the number of bit pairs needed after applying the coding scheme is

$$n' < \frac{n}{m \cdot l} \ . \tag{11}$$

Since we have, for a given security bound ε, an optimal probability p_{opt} and the respective minimum number of bit pairs n_{opt}, the combinations of m and l that could provide for a convenient m-l-coding scheme are those for which $n > n_{\text{opt}} \cdot m \cdot l$, from which we have $m \cdot l < \frac{n}{n_{\text{opt}}}$. By simply iterating through the limited number of possibilities, we can find the best n'. Then, subtracting this n' to n, we get the measure of the improvement we get by using the associated l-m-coding scheme, calculated in number of bit pairs. For probabilities $p > 0.37$, the gain quickly becomes consistent. There is also a (very) modest gain for low probabilities, namely for $0 < p \leq 0.002$, where the number of saved packets reaches, for example, the value of 51 for $p = 0.0001$.

A different strategy to increase the efficiency of the protocol without using coding schemes is to modify the size of the index sets I_0 and I_1. This strategy is effective when the observed probability of the channel is higher than the optimal value. By reducing the size of the sets I_0 and I_1, we are able to tolerate a larger number of errors introduced by the channels, effectively moving the correctness bound to the right of the graph. However, we are at the same time increasing the minimum number of errors needed to guarantee the security for Sam, moving

the relative bound to the right too. In practice, if $r = |I| - \left|\overline{I}\right|$ is the difference between the cardinality of regular set I, $\frac{n}{2}$, and that of the reduced set of size $\left|\overline{I}\right|$, we have a probability that Rachel will be able to correctly decode the selected bit (correctness) $\sum_{k=\frac{n}{2}-r}^{n} \binom{n}{k} (1-p)^k p^{n-k}$. Using Hoeffding's inequality we can bound that probability to $\leq \exp\left(-2n\left(\frac{1}{2} - p - \frac{r}{n}\right)^2\right)$. Her probability of being able to correctly decode both bits is instead $\leq \frac{1}{2} + \left(\frac{1+p}{2}\right)^{n-2r}$.

5 Conclusion

In this paper we consider the use of the Z-channel for implementing oblivious transfer. This simple communication channel models the functioning of various real-life communication methods, ranging from optical to covert communication. The Z-channel features an unusual property for a noisy channel: only part of the information sent through it is affected by the noise, so the receiver can always interpret correctly some of the output symbols. This particular characteristic eliminates the need for error correction codes, which, in other constructions, limit considerably the range of error probabilities acceptable for secure computation.

Our construction follows the common strategy of constructing some form of erasure channel, which is then repeatedly used to implement OT. Thanks to an efficient reduction of the Z-channel to a binary erasure channel, the protocol exhibits a low sensitivity to the channel characteristics. The parties are less constrained by exact knowledge of the channel statistics, and for a very large range of error probabilities, $0.17 < p < 0.29$, a total of 500 bits transmitted through the channel is sufficient to guarantee a security of $\varepsilon = 10^{-9}$ (Fig. 2). This is particularly useful when confronted to unfair adversaries, who are able to select the channel probability within a certain range $[\gamma, \delta]$. Over a Z-channel, security against unfair behavior is possible for any fixed range, while the unfair BSC introduced by Damgård et al. [7, 6] can not achieve OT as soon as the range is larger than $\delta \geq 2\gamma(1-\gamma)$, with a maximum possible width of 0.125. A total of 2120 bits transmitted over a Z-channel is instead sufficient to guarantee security for any probability in the range $[0.04, 0.4]$, for $\varepsilon = 10^{-9}$.

The efficiency of the construction can be further increased by using a combination of two coding schemes, presented in the last section of the paper. A combined m-l-coding strategy, where m and l are the parameters of the schemes, can reduce any Z-channel with probability p to a Z-channel with error probability $1 - (1-p^m)^l$. This allows the use of channels whose error probability is greater than 0.5. Moreover, when confronted with a channel with a probability distant from the optimal range, the parties can decide to opt for the use of a coding strategy in order to increase the efficiency of the protocol. The factor by which the communication over a Z-channel is increased $(m \cdot l)$ is widely compensated by the reduced repetition number n that the more efficient probability allows for. This is especially evident for any $p > 0.37$.

Acknowledgments

This research work was supported by the SCOOP Action de Recherche Concertées. Olivier Pereira is a Research Associate of the F.R.S.-FNRS.

References

1. Baumert, L.D., McEliece, R.J., Rumsey, H.: Coding for optical channels. In: JPL Deep Space Network Progress Report. vol. 42–49, pp. 70–77 (1978)
2. Crépeau, C.: Equivalence between two flavours of oblivious transfers. In: Pomerance, C. (ed.) CRYPTO. Lecture Notes in Computer Science, vol. 293, pp. 350–354. Springer (1987)
3. Crépeau, C.: Efficient cryptographic protocols based on noisy channels. In: EUROCRYPT. pp. 306–317 (1997)
4. Crépeau, C., Kilian, J.: Achieving oblivious transfer using weakened security assumptions (extended abstract). In: FOCS. pp. 42–52. IEEE (1988)
5. Crépeau, C., Morozov, K., Wolf, S.: Efficient unconditional oblivious transfer from almost any noisy channel. In: Blundo, C., Cimato, S. (eds.) SCN. Lecture Notes in Computer Science, vol. 3352, pp. 47–59. Springer (2004)
6. Damgård, I., Fehr, S., Morozov, K., Salvail, L.: Unfair noisy channels and oblivious transfer. In: Naor, M. (ed.) TCC. Lecture Notes in Computer Science, vol. 2951, pp. 355–373. Springer (2004)
7. Damgård, I., Kilian, J., Salvail, L.: On the (im)possibility of basing oblivious transfer and bit commitment on weakened security assumptions. In: EUROCRYPT. pp. 56–73 (1999)
8. Golomb, S.W.: The limiting behavior of the z-channel. IEEE Transactions on Information Theory 26(3), 372–372 (1980)
9. Imai, H., Morozov, K., Nascimento, A.: On the oblivious transfer capacity of the erasure channel. In: Proceedings of 2006 International Symposium on Information Theory (ISIT). pp. 1428–1431. IEEE (2006)
10. Kilian, J.: Founding cryptography on oblivious transfer. In: STOC. pp. 20–31. ACM (1988)
11. Moskowitz, I.S., Greenwald, S.J., Kang, M.H.: An analysis of the timed z-channel. IEEE Transactions on Information Theory 44(7), 3162–3168 (1998)
12. Nascimento, A.C.A., Winter, A.: On the oblivious-transfer capacity of noisy resources. IEEE Transactions on Information Theory 54(6), 2572–2581 (2008)
13. Palmieri, P., Pereira, O.: Building oblivious transfer on channel delays. In: Lai, X., Yung, M., Lin, D. (eds.) INSCRYPT. Lecture Notes in Computer Science, vol. 6584, pp. 125–138. Springer (2010)
14. Rabin, M.O.: How to exchange secrets by oblivious transfer. Technical Report TR-81, Aiken Computation Laboratory, Harvard University (1981), manuscript
15. Wullschleger, J.: Oblivious-transfer amplification. In: Naor, M. (ed.) EUROCRYPT. Lecture Notes in Computer Science, vol. 4515, pp. 555–572. Springer (2007)
16. Wullschleger, J.: Oblivious transfer from weak noisy channels. In: Reingold, O. (ed.) TCC. Lecture Notes in Computer Science, vol. 5444, pp. 332–349. Springer (2009)