On Robust Combiners for Private Information Retrieval and Other Primitives

Remo Meier and Bartosz Przydatek

Department of Computer Science, ETH Zurich
8092 Zurich, Switzerland
remmeier@student.ethz.ch, przydatek@inf.ethz.ch

Abstract. Let $\mathcal{A}$ and $\mathcal{B}$ denote cryptographic primitives. A (k, m)-*robust* $\mathcal{A}$*-to-*$\mathcal{B}$ *combiner* is a construction, which takes m implementations of primitive $\mathcal{A}$ as input, and yields an implementation of primitive $\mathcal{B}$, which is guaranteed to be secure as long as at least k input implementations are secure. The main motivation for such constructions is the tolerance against *wrong assumptions* on which the security of implementations is based. For example, a (1,2)-robust $\mathcal{A}$-to-$\mathcal{B}$ combiner yields a secure implementation of $\mathcal{B}$ even if an assumption underlying *one* of the input implementations of $\mathcal{A}$ turns out to be wrong.

In this work we study robust combiners for private information retrieval (PIR), oblivious transfer (OT), and bit commitment (BC). We propose a (1,2)-robust PIR-to-PIR combiner, and describe various optimizations based on properties of existing PIR protocols. The existence of simple PIR-to-PIR combiners is somewhat surprising, since OT, a very closely related primitive, seems difficult to combine (Harnik *et al.*, Eurocrypt'05). Furthermore, we present (1,2)-robust PIR-to-OT and PIR-to-BC combiners. To the best of our knowledge these are the first constructions of $\mathcal{A}$-to-$\mathcal{B}$ combiners with $\mathcal{A} \neq \mathcal{B}$. Such combiners, in addition to being interesting in their own right, offer insights into relationships between cryptographic primitives. In particular, our PIR-to-OT combiner together with the impossibility result for OT-combiners of Harnik *et al.* rule out certain types of reductions of PIR to OT. Finally, we suggest a more fine-grained approach to construction of robust combiners, which may lead to more efficient and practical combiners in many scenarios.

Keywords: robust combiners, cryptographic primitives, reductions, private information retrieval, oblivious transfer, bit commitment

1 Introduction

Consider a scenario when two implementations, I_1 and I_2, of some cryptographic primitive are given, e.g., two encryption schemes or two bit commitment schemes. Each implementation is based on some unproven computational assumption, α_1 resp. α_2, like for example the hardness of factoring integer numbers or the hardness of computing discrete logarithms. We would like to have an implementation I of the primitive, which is as secure as possible given the current state of knowledge. As it is often not clear, which of the assumptions α_1, α_2 is more likely to be correct, picking just one of the implementations does not work — we might bet on the wrong assumption! A better option would be to have an implementation which is guaranteed to be secure as long as *at least one* of the assumptions α_1, α_2 is correct. That is, given I_1 and I_2 we would like to construct an efficient implementation I, which is secure whenever at least one of the input implementations is. Such a construction is an example of a *(1,2)-robust combiner*, as it *combines* the input implementations and is *robust* against situations when one of the two inputs is insecure.

In general, robust combiners can use more than just two input schemes, and aim at providing a secure implementation of the output primitive assuming that sufficiently many of the candidates are secure. Moreover, the input candidates do not have to be necessarily implementing the same primitive, and the goal of a combiner may be a construction of a primitive different from the primitives given at the input. That is, a robust combiner can be viewed as a robust *reduction* of the output primitive to the input primitive(s).

The concept of robust combiners is actually not so new in cryptography and many techniques are known for combining cryptographic primitives to improve security guarantees, e.g., cascading of block ciphers. However, a more formal and rigorous study of combiners was initiated quite recently [Her05,HKN$^+$05].

Robust combiners for some primitives, like one-way functions or pseudorandom generators, are rather simple, while for others, e.g., for oblivious transfer (OT), the construction of combiners seems considerably harder. In particular, in a recent work Harnik *et al.* [HKN$^+$05] show that there exists no "transparent black-box" (1,2)-robust OT-combiner. Given the impossibility result for OT-combiners, it is interesting to investigate the existence of combiners for *single-database* private information retrieval (PIR), a primitive closely related, yet not known to be equivalent, to oblivious transfer. Potential PIR-combiners could lead to better understanding of relations between PIR, OT, and other primitives. Moreover, constructions of robust PIR-combiners are also of considerable practical interest, stemming from the fact that some of the most efficient PIR protocols are based on relatively new computational assumptions (e.g., [CMS99,KY01]), which are less studied and thus potentially more likely to be proved wrong.

Contributions. In this work we consider robust combiners for private information retrieval, bit commitment, and oblivious transfer. In particular, we present (1,2)-robust PIR-combiner, i.e. combiner which given two implementations of PIR yield an implementation of PIR which is secure if at least one of the input implementations is secure. We also describe various techniques and optimizations based on properties of existing PIR protocols, which yield PIR-combiners with better efficiency and applicability.

Furthermore, we construct $\mathcal{A}$-to-$\mathcal{B}$ combiners, i.e. "cross-primitive" combiners, which given multiple implementations of a primitive $\mathcal{A}$ yield an implementation of some other primitive $\mathcal{B}$, which is provably secure assuming that sufficiently many of the input implementations of $\mathcal{A}$ are secure. Specifically, we construct (1,2)-robust PIR-to-BC and PIR-to-OT combiners. To the best of our knowledge these are the first combiners of this type. While interesting in their own right, such combiners also offer insights into relationships and reductions between cryptographic primitives. In particular, our PIR-to-OT combiner together with the impossibility result of [HKN+05] rule out certain types of reductions of PIR to OT (cf. Corollary 4 in Section 4).

Finally, we suggest a more fine-grained approach to design of robust combiners. That is, we argue that in order to obtain combiners as efficient as possible, the constructions may take into account that some properties of the input candidates are proved to hold unconditionally, and hence cannot go wrong even if some computational assumption turns out to be wrong. Therefore, keeping in mind the original motivation for combiners, i.e. protection against wrong assumptions, a more fine-grained approach to design of robust combiners exploits the unconditionally secure properties and focuses on the properties which hold only under given assumptions. This change of focus yields sometimes immediately trivial constructions of combiners (as observed by Harnik *et al.* [HKN+05] for OT and BC), yet in many cases the resulting problems are still interesting and challenging (see Sections 2.3 and 5 for more details).

Related work. As mentioned above, a more rigorous study of robust combiners was initiated only recently, by Herzberg [Her05] and by Harnik *et al.* [HKN+05]. On the other hand, there are numerous implicit uses and constructions of combiners in the literature (e.g., [AB81,EG85,DK05,HL05]).

Private information retrieval was introduced by Chor *et al.* [CKGS98] and has been intensively studied since then. The original setting of PIR consisted of multiple non-communicating copies of the database and guaranteed *information-theoretic* privacy for the user. Later, Kushilevitz and Ostrovsky [KO97] gave the first solution to *single-database* PIR, in which the privacy of the user is based on a computational assumption. The first PIR protocol with communication complexity polylogarithmic in the size of the database was proposed by Cachin *et al.* [CMS99], and in recent years more efficient constructions have been proposed (e.g. [Cha04,Lip05]). For more information about PIR we refer to the survey by Gasarch [Gas04].

The relationships between PIR and other primitives have been studied intensively in the recent years. In particular, Beimel *et al.* [BIKM99] proved that any non-trivial single-database PIR implies one-way functions, and Di Crescenzo *et al.* [DMO00] showed that such a PIR implies oblivious transfer. Kushilevitz and Ostrovsky [KO00] demonstrated that one-way trapdoor *permutations* are sufficient for non-trivial single-database PIR. On the negative side, Fischlin [Fis02] showed that there is no black-box construction of one-round (i.e., two-message) PIR from one-to-one trapdoor *functions*.

Techniques similar to the ones employed in the proposed PIR-combiners were previously used by Di Crescenzo *et al.* [DIO01] in constructions of universal service-providers for PIR.

Organization. Section 2 contains notation, definitions of cryptographic primitives and of robust combiners, and some general remarks about combining PIR protocols. In Section 3 we describe proposed constructions of (1,2)-robust PIR-combiners. In Section 4 we turn to "cross-primitive" combiners and present PIR-to-BC and PIR-to-OT combiners. Finally, in Section 5 we discuss some general aspects of design of efficient combiners and point out some open problems.

2 Preliminaries

Notational conventions. If x is a bit-string, $|x|$ denotes its length, and we write $x\|y$ to denote the concatenation of the bit-strings x, y. For an integer n we write $[n]$ to denote the set $\{1, \ldots, n\}$. The parties participating in the protocols and the adversary are assumed to be probabilistic polynomial time Turing machines, (PPTMs).

2.1 Primitives

We review shortly the primitives relevant in this work. For more formal definitions we refer to the literature.

Private Information Retrieval is a protocol between two parties, a server holding an n-bit database $x = (x_1\| \ldots \|x_n)$, and a user holding an index $i \in [n]$. The protocol allows the user to retrieve bit x_i without revealing i to the server, i.e. it protects user's privacy. In this work we consider only single-database PIR. Of interest are only *non-trivial* protocols, in which the total *server-side* communication (i.e. communication from the server to the user) is less than n bits. Moreover, of special interest are 2-message protocols, in which only two messages are sent: a *query* from the user to the server and a *response* from the server to the user.

Oblivious Transfer [1] is a protocol between a sender holding two bits x_0 and x_1, and a receiver holding a choice-bit c. The protocol allows the receiver to get bit x_c so that the sender does not learn any information about receiver's choice c, and the receiver does not learn any information about bit x_{1-c}.

Bit Commitment is a two-phase protocol between two parties Alice and Bob. In the *commit* phase Alice commits to bit b without revealing it, by sending to Bob an "encrypted" representation e of b. Later, in the *decommit* phase, Alice sends to Bob a decommitment string d, allowing Bob to "open" e and obtain b. In addition to correctness, a bit commitment scheme must satisfy two properties: *hiding*, i.e. Bob does not learn the bit b before the decommit phase, and *binding*, i.e. Alice cannot come up with decommitment strings d, d' which lead to opening the commitment as different bits. We consider also *weak* bit commitment, i.e. BC with *weak binding* property: Alice might be able to cheat, but Bob catches her cheating with noticeable probability [BIKM99].

[1] The version of oblivious transfer described here and used in this paper is more precisely denoted as *1-out-of-2 bit-OT* [EGL85]. There are several other versions of OT, e.g., *Rabin's OT*, *1-out-of-n bit-OT*, or *1-out-of-n string-OT*, but all are known to be equivalent [Rab81,Cré87,CK88].

2.2 Robust combiners

The following definition is a generalization of the definition of combiners given in [HKN+05].

Definition 1 ((k,m)-robust $\mathcal{A}$-to-$\mathcal{B}$ combiner). *Let $\mathcal{A}$ and $\mathcal{B}$ be cryptographic primitives. A (k,m)-*robust $\mathcal{A}$-to-$\mathcal{B}$ combiner *is a PPTM which gets m candidate schemes implementing $\mathcal{A}$ as inputs and implements $\mathcal{B}$ while satisfying the following two properties:*

1. *If at least k candidates securely implement $\mathcal{A}$, then the combiner securely implements $\mathcal{B}$.*
2. *The running time of the combiner is polynomial in the security parameter κ, in m, and in the lengths of the inputs to $\mathcal{B}$.*

An *$\mathcal{A}$-to-$\mathcal{A}$ combiner* is called an *$\mathcal{A}$-combiner*. For completeness we recall three definitions from [HKN+05], which will be useful in our constructions.

Definition 2 (Black-box combiner [HKN+05]). *A $(1,2)$-robust combiner is called a* black-box combiner *if the following two conditions hold:*

Black-Box Implementation: *The combiner is an oracle PPTM given access to the candidates via oracle calls to their implementation function.*

Black-Box Proof: *For every candidate there exists an oracle PPTM R^A (with access to A) such that if adversary A breaks the combiner, then R^A breaks the candidate.*

Definition 3 (Third-party black-box combiner [HKN+05]). *A* third-party black-box combiner *is a black-box combiner where the input candidates behave like trusted third parties. The candidates give no transcript to the players but rather take their inputs and return outputs.*

Definition 4 (Transparent black-box combiner [HKN+05]). *A* transparent black-box combiner *is a black-box combiner for an interactive primitive where every call to a candidate's next message function is followed by this message being sent to the other party.*

Note that the notion of *reduction* of primitive $\mathcal{B}$ to primitive $\mathcal{A}$, i.e. a construction of $\mathcal{B}$ from $\mathcal{A}$, can be viewed as a (1,1)-robust $\mathcal{A}$-to-$\mathcal{B}$ combiner. Therefore, the above definitions also include notions like a *transparent black-box reduction* or a *third-party black-box reduction*.

2.3 Remarks on combiners for PIR protocols

As pointed out by Harnik *et al.* [HKN+05], cryptographic primitives are mainly about security, while functionality issues are often straightforward. For example, a PIR protocol has to satisfy a security property, i.e. privacy of the user, and functionality properties: efficiency, completeness, and non-triviality. Usually[2] the privacy of the user is based on some cryptographic assumption, and the remaining properties hold unconditionally. Moreover, in some cases a possible way of dealing with unknown implementations of primitives is to test them for the desired functionality, hence, even if the candidate input primitives are given as black-boxes, one can test them before applying a combiner (for a more detailed discussion of these issues see Section 3.1 in [HKN+05]).

For the above reasons we assume that the PIR candidates used as input by the combiners are guaranteed to have the desired functionality (i.e. efficiency, completeness, and non-triviality), and that explicit bounds on running time and on communication complexity are given as parts of the input to the combiner. Thus, the task of a combiner is to protect against wrong computational assumptions. This approach is especially relevant in the context of private information retrieval, since some of the most efficient PIR protocols are based on new computational assumptions (e.g., [CMS99,KY01]), which are less studied and so potentially more likely to be broken (cf. recent attack of Bleichenbacher *et al.* [BKY03] on [KY01]).

3 PIR-combiners

In this section we assume that two (non-trivial) private information retrieval schemes are given, PIR_1 and PIR_2, where PIR_1 is a *two-message* PIR protocol with a query $q = Q_1(i)$ and a response $r = R_1(q, (x_1\|\ldots\|x_n))$, and where PIR_2 is an arbitrary (possibly multi-round) PIR protocol. We use $c_{s,1}(n)$ and $c_{s,2}(n)$ to denote the server-side communication complexities of the PIR-schemes, and $c_{u,1}(n)$ and $c_{u,2}(n)$ to denote the corresponding user-side complexities[3]. Without loss of generality we assume that these complexities give the exact number of communicated bits and are not just upper bounds.

First we describe a basic scheme for a (1,2)-robust PIR-combiner, and then present some variations of the scheme, resulting in better efficiency. Our constructions are black-box combiners, but not *transparent* black-box combiners because they require offline access to one of the candidates.

3.1 The basic scheme

Our basic PIR-combiner works as follows: to retrieve the i-th bit from a database $x = (x_1\|\ldots\|x_n)$, the database first defines n auxiliary databases $y_1, ..., y_n$, where y_j is just a copy of x rotated by $(j-1)$ positions, i.e.

$$y_j = (x_j\|\ldots\|x_{n-1}\|x_n\|x_1\|\ldots\|x_{j-1}).$$

The user picks a random $t \in [n]$ and sends to the server PIR_1-query $q = Q_1(t)$. For each database y_j, $j \in [n]$, the server computes the corresponding response

[2] We are not aware of any (single-database) PIR protocol not conforming to this characterization.

[3] In particular, $c_{s,1}(n) = |R_1(q, (x_1, \ldots, x_n))|$ and $c_{u,1} = |Q_1(i)|$

SERVER'S INPUT: n-bit string $x = (x_1\|\dots\|x_n)$
USER'S INPUT: $i \in [n]$
INPUT PIR PROTOCOLS:
  PIR_1: 2-message, with query $Q_1(j)$ and response $R_1(Q_1(j), x)$
  PIR_2: arbitrary

phase I:

1. server defines n databases y_j for each $j \in [n]$ as:
   $y_j = (x_j\|\dots\|x_{n-1}\|x_n\|x_1\|\dots\|x_{j-1})$
2. user picks a random $t \in [n]$ and sends PIR_1-query $q = Q_1(t)$ to the server
3. server computes n PIR_1-responses $r_j = R_1(q, y_j)$ for each $j \in [n]$

phase II:

1. user computes $k = ((i - t) \mod n) + 1$
2. user retrieves from server response r_k bit-by-bit, by running $|r_k|$ instances of PIR_2 with $x' = (r_1\|r_2\|\dots\|r_n)$ as server's input
3. user computes x_i from the PIR_1-response r_k

Fig. 1: The basic (1,2)-robust PIR-combiner.

$r_j = R_1(q, y_j)$, but instead of sending the responses back to the user, he stores them in a new database[4] $x' = (r_1\|\dots\|r_n)$. Note that the new database x' contains a PIR_1-response for each bit x_j of the original database x, but with the positions rotated by $(t-1)$. Finally the user retrieves bit-by-bit the response r_k for $k = ((i-t) \mod n)+1$, by running $c_{s,1}(n)$ instances of PIR_2, and computes x_i from r_k. Figure 1 presents the combiner in more detail, and the following theorem summarizes the properties of the combiner.

Theorem 1. *There exists a black-box (1,2)-robust PIR-combiner for input candidates PIR_1 and PIR_2, where PIR_1 is a 2-message protocol, and where the candidates' server-side communication complexities satisfy*

$$c_{s,1}(n) \cdot c_{s,2}\left(n \cdot c_{s,1}(n)\right) < n \; . \tag{1}$$

The user-side communication of the resulting PIR scheme is equals

$$c_{u,1}(n) + c_{s,1}(n) \cdot c_{u,2}\left(n \cdot c_{s,1}(n)\right) \; .$$

Proof. *(sketch)* Consider the construction presented in Figure 1. It is clear that this construction is efficient. It remains to show that the resulting protocol is a secure, non-trivial PIR if at least one of the input candidates is a secure PIR. Let $\overline{\mathsf{PIR}}$ denote the PIR protocol resulting from the combiner. If PIR_1 is insecure, then the server learns the random index t. However, as in this case PIR_2 remains secure, the server obtains no information about index k when the user retrieves the response r_k, and so does not gain information about the index i.

[4] A similar technique was used in [DIO01] for universal service-providers for PIR.

On the other hand, if PIR_2 is insecure, then the server learns index k, but as PIR_1 is now secure, the server gets no information about t. Since t is randomly chosen, knowledge of k does not give any information about index i.

Finally, we argue the non-triviality condition: it is easy to verify that the server-side communication of $\overline{\mathsf{PIR}}$ is

$$c_s(n) = c_{s,1}(n) \cdot c_{s,2}\left(n \cdot c_{s,1}(n)\right) \ ,$$

and the user-side communication is

$$c_u(n) = c_{u,1}(n) + c_{s,1}(n) \cdot c_{u,2}\left(n \cdot c_{s,1}(n)\right) \ .$$

Thus if $c_s(n) < n$ holds, i.e. if condition (1) is satisfied, then $\overline{\mathsf{PIR}}$ is non-trivial. □

3.2 PIR-combiners with lower communication

The basic combiner presented in the previous section is conceptually simple and works well for a wide range of candidate PIR-protocols, but leaves some space for improvements. In this section we describe some variations and optimizations of this basic combiner, which yield significant improvements in communication efficiency of the resulting PIR schemes. This results in combiners applicable to a wider range of input candidates.

First we describe how to reduce the cost of querying x' by using several databases in parallel. Then we discuss possible improvements in situations when the candidates return entire blocks of several bits instead of single bits.

Reducing overall communication. In the second phase of the basic scheme the user retrieves r_k bit-by-bit by running $|r_k| = c_{s,1}(n)$ instances of PIR_2 with server's input x' of length $n \cdot c_{s,1}(n)$. An alternative way of retrieving r_k is the following: we arrange all responses $r_1, \ldots, r_n$ into $l = |r_k|$ databases $x'_1, \ldots, x'_l$, each of length n, where x'_j contains the j-th bits of all responses $r_1, \ldots, r_n$. Then the user obtains r_k by retrieving the k-th bits from the databases $x'_1, \ldots, x'_l$. That is, user and server run $|r_k|$ instances of PIR_2, where in the j-th instance server's input is x'_j and user's input k. Thus we obtain the following corollary.

Corollary 1. *There exists a black-box (1,2)-robust PIR-combiner for input candidates* PIR_1 *and* PIR_2*, where* PIR_1 *is a 2-message protocol, and where the candidates' server-side communication complexities satisfy*

$$c_{s,1}(n) \cdot c_{s,2}(n) < n \ . \tag{2}$$

The user-side communication of the resulting PIR scheme is equals

$$c_{u,1}(n) + c_{s,1}(n) \cdot c_{u,2}(n) \ . \tag{3}$$

Note that if PIR_2 is also a 2-message PIR protocol, then only *one query* must be sent in the second phase of the combiner (for which $c_{s,1}(n)$ PIR_2-responses will be sent), thus reducing the user-side communication of the resulting PIR scheme even further, to merely

$$c_{u,1}(n) + c_{u,2}(n) \ .$$

Further optimizations and variations. If PIR_2 retrieves entire blocks rather than single bits (for example, the basic PIR protocol of [KO97] does exactly that), than the retrieval of r_k can be substantially sped-up, as it can proceed block-by-block rather than bit-by-bit. Moreover, if $|r_k|$ is not larger than the size of blocks retrieved by PIR_2, than just one execution of PIR_2 is sufficient.

Corollary 2. *There exists a black-box (1,2)-robust PIR-combiner for input candidates* PIR_1 *and* PIR_2*, where* PIR_1 *is a 2-message protocol,* PIR_2 *retrieves blocks of size at least* $c_{s,1}(n)$*, and where the candidates' server-side communication complexities satisfy*

$$c_{s,2}(n \cdot c_{s,1}(n)) < n \ .$$

The user-side communication of the resulting PIR scheme is equals

$$c_{u,1}(n) + c_{u,2}(n \cdot c_{s,1}(n)) \ .$$

Another simple optimization is possible when PIR_1 supports block-wise retrieval, i.e., when each PIR_1-response r_j allows retrieval of ℓ-bit blocks. In such a case it is sufficient to store in x' a subset of n/ℓ responses, so that the corresponding blocks cover the entire database — then in phase II user simply retrieves the block containing the desired bit x_i.

Finally, when the user-side communication of the candidate PIRs is much higher than the server-side communication, it is possible to balance the load better between the two parties with the so called *balancing technique*, which was introduced in the context of information-theoretic PIR [CKGS98], and which can be viewed as a simulation of block-wise retrieval: server partitions the database to u databases of size n/u. User provides then a single query for some index $j \in [n/u]$, which is answered for each of u databases, yielding a block of u bits.

Clearly, one can use multiple optimizations together (if applicable) to obtain the most efficient construction for the given candidate PIR protocols.

4 Combining PIR protocols to other primitives

The research on robust combiners so far focused mainly on finding ways of combining candidate instances of a given primitive $\mathcal{A}$ to yield a secure instance of $\mathcal{A}$. In this section we describe robust combiners of a more general type, combining instances of primitive $\mathcal{A}$ to an instance of primitive $\mathcal{B}$. Such combiners can be viewed as a combination of robust combiners and reductions between primitives in one construction.

First we consider the problem of combining PIR protocols to obtain a bit commitment scheme, and present a third-party black-box (1,2)-robust PIR-to-BC combiner. Then we turn to combining PIR protocols to oblivious transfer, and present a black-box (1,2)-robust PIR-to-OT combiner. The existence of such a combiner is somewhat surprising, given the impossibility result of [HKN+05] and the fact that PIR and OT are very closely related.

ALICE'S INPUT: bit b
BOB'S INPUT: (none)
INPUT PIR PROTOCOLS: PIR_1, PIR_2

commit phase:

1. Alice picks two independent, uniformly random strings $x, y \in_R \{0,1\}^n$
   Bob picks two independent, uniformly random indices $i_1, i_2 \in_R [n]$
2. Alice and Bob execute PIR protocols $\mathsf{PIR}_1(x; i_1)$ and $\mathsf{PIR}_2(x; i_2)$, with Alice as server with database x, and Bob retrieving bit with index i_1 resp. i_2.
3. Alice sends to Bob y and $c = b \oplus \mathrm{IP}(x, y)$
   (where IP denotes inner product over GF(2))

decommit phase:

1. Alice sends to Bob x as decommitment string
2. Bob verifies that x is consistent with the bits retrieved during commit phase and computes $b = c \oplus \mathrm{IP}(x, y)$

Fig. 2: A (1,2)-robust PIR-to-(weak)BC combiner.

4.1 PIR-to-BC combiner

It is well-known that single-database PIR implies one-way functions [BIKM99], which in turn are sufficient to construct computationally hiding and statistically binding bit commitments schemes [Nao91]. It follows immediately that there exists a generic combiner going through these reductions and an OWF-combiner. However, such a combiner is quite inefficient, and it is not a third-party black-box combiner.

In this section we present a more efficient, third-party black-box PIR-to-BC combiner, which is basically a slight variation of the reduction of bit commitment to private information retrieval due to Beimel *et al.* [BIKM99]. In contrast to the generic combiner described above, the BC-scheme resulting from the proposed combiner is statistically hiding and computationally binding. We describe only a construction for *weak* bit commitment, which can then be strengthened by using multiple independent commitments to the same bit [BIKM99]. A detailed description of the combiner is presented in Figure 2.

Theorem 2. *There exists a third-party black-box (1,2)-robust PIR-to-BC combiner yielding a statistically hiding BC, for input candidates* PIR_1 *and* PIR_2 *with server-side communication complexities satisfying*

$$c_{s,1}(n) + c_{s,2}(n) \leq n/2 \; . \tag{4}$$

Proof. *(sketch)* As mentioned above, it is sufficient to show a combiner from PIR to *weak* bit commitment. Consider the construction presented in Fig. 2. The correctness and the efficiency of the scheme are straightforward. The hiding property follows from the bound on server-side communication complexities (4) and from high communication complexity of the inner product IP (see [BIKM99] for details). The weak binding property follows from the assumption that at least one of the PIR protocols is secure for the receiver, hence at least one of the indices i_1, i_2 remains unknown to Alice. Finally, it is straightforward to verify that this is a third-party black-box combiner. $\square$

Obviously, the bound $n/2$ in (4) is not tight. Since our focus in this work is on existence of efficient combiners, and since many practical PIR protocols have polylogarithmic communication bounds (which clearly satisfy (4)), we do not attempt to optimize this bound. Moreover, the PIR-to-OT combiner presented in the next section implies an alternative, efficient $(1, 2)$-robust PIR-to-BC combiner (cf. Corollary 3).

4.2 PIR-to-OT combiner

The PIR-to-BC combiner presented in the previous section can be viewed as a variation of the general approach to construct (1,2)-robust PIR-to-BC combiners: first use a construction of unconditionally hiding BC from a single-database PIR to obtain BC_1 resp. BC_2, and then combine the two BC protocols using the fact that both are unconditionally secure for Alice and at most one not binding (if the corresponding PIR protocol is insecure). As we show in this section, a similar approach works for PIR-to-OT combiners.[5] That is, our proposed PIR-to-OT combiner first constructs OT protocols OT_1 and OT_2 based on candidates PIR_1 resp. PIR_2, and then combines OT_1 and OT_2 using the fact that both these protocols are unconditionally secure for the sender.

For completeness, Figure 4 in the appendix presents the construction of OT (unconditionally secure for the sender) based on single-database PIR [DMO00]. Using this construction, our proposed (1,2)-robust PIR-to-OT combiner works as follows: given two PIR protocols, PIR_1 and PIR_2, we use the reduction from Fig. 4 to obtain OT protocols OT_1 and OT_2, respectively. Now, as both resulting OT's are unconditionally secure for the sender, we can combine them by using a combiner which guarantees the privacy of the receiver as long as at least one of the two input OT's is secure. For this purpose we use the combiner[6] $R(\cdot, \cdot)$ from [HKN$^+$05]. Figure 3 presents the proposed PIR-to-OT combiner in full detail, and we obtain the following theorem.

Theorem 3. *There exists a black-box (1,2)-robust PIR-to-OT combiner.*

Proof. *(sketch)* Consider the construction in Figure 3, and let $\overline{\mathsf{OT}}$ denote the resulting OT protocol. Since *any* non-trivial single-database PIR implies OT with

[5] Note that unlike in the case of PIR-to-BC combiners, it is unclear whether there exists (1,2)-robust PIR-to-OT combiner based on combiners for one-way functions: while it is known that non-trivial PIR implies one-way functions [BIKM99], it is unlikely that OT can be constructed from one-way functions only [IR89] (the most general assumptions known to be sufficient for OT are the existence of *enhanced* [EGL85,Gol04] or *dense* [Hai04] one-way trapdoor permutations).

[6] This combiner was used in [HKN$^+$05] to construct a (2,3)-robust OT-combiner.

SENDER'S INPUT: two bits b_0, b_1
RECEIVER'S INPUT: choice bit c
INPUT PIR PROTOCOLS: PIR_1, PIR_2

1. parties construct OT protocols OT_1, OT_2 from PIR_1, PIR_2, respectively (using the construction from Fig. 4)
2. parties combine OT_1, OT_2, using combiner $R(\mathsf{OT}_1,\mathsf{OT}_2)(b_0, b_1; c)$ [HKN+05]:
   (i) sender picks a random bit r,
   receiver picks random bits c_1, c_2 s.t. $c_1 \oplus c_2 = c$
   (ii) parties run $\mathsf{OT}_1(r, r \oplus b_0 \oplus b_1; c_1)$ and $\mathsf{OT}_2(r \oplus b_0, r \oplus b_1; c_2)$
   (iii) receiver outputs the XOR of the two bits received in OT_1 and OT_2

Fig. 3: A (1,2)-robust PIR-to-OT combiner.

unconditional security for the sender [DMO00], OT_1 and OT_2 are well defined, and it is easy to verify the correctness of $\overline{\mathsf{OT}}$. Moreover, unconditional security of the sender in OT_1 and OT_2 means that the receiver obtains information about at most one of $(r, r \oplus b_0 \oplus b_1)$ and about at most one of $(r \oplus b_0, r \oplus b_1)$. This implies unconditional security of the sender in $\overline{\mathsf{OT}}$. The security of the receiver in $\overline{\mathsf{OT}}$ follows from the assumption that at least one of the input PIR protocols is secure. More precisely, security of at least one PIR implies security of at least one of OT_1, OT_2, hence, at least one of c_1, c_2 remains hidden from the sender, and consequently the sender obtains no information about c. Finally, it is easy to verify that this is a black-box combiner. □

Since the OT protocol resulting from the above combiner is unconditionally secure for the sender, we can use it to construct a statistically hiding BC scheme, hence we get the following corollary.

Corollary 3. *There exists a black-box (1,2)-robust PIR-to-BC combiner yielding a statistically hiding BC.*

Furthermore, recall that a reduction of a primitive $\mathcal{B}$ to a primitive $\mathcal{A}$ can be viewed as a (1,1)-robust $\mathcal{A}$-to-$\mathcal{B}$ combiner, hence a notion of *transparent black-box reduction* is well-defined. Note also that in the case of honest-but-curious parties or low-communication PIR-candidates, the PIR-to-OT combiner resulting from the proof of Theorem 3 is even a *third-party* black-box combiner (cf. Appendix and [DMO00]). Therefore, a combination of Theorem 3 with the impossibility result for (1,2)-robust transparent black-box OT-combiners [HKN+05] leads to the following corollary, which rules out certain types of reductions of PIR to OT.

Corollary 4. *There exists no transparent black-box reduction of single-database private information retrieval to oblivious transfer, even for honest-but-curious parties.*

5 Conclusions and open problems

We have presented constructions of (1,2)-robust PIR-combiners, and also "cross-primitive" combiners: PIR-to-BC and PIR-to-OT. The existence of simple and efficient PIR-combiners is somewhat surprising given the impossibility result for OT-combiners. Moreover, a closer look at the PIR-to-BC and PIR-to-OT combiners reveals a common theme — we use a reduction of the target primitive to the input primitive, and exploit additional security properties guaranteed by the reduction to obtain an efficient combiner. It seems that such a fine-grained approach to the design of combiners, i.e. taking explicitly into account that *some* properties of the candidates hold unconditionally can yield more efficient, practical combiners. Indeed, as pointed out by Harnik *et al.* [HKN$^+$05], if the security of one of the parties is guaranteed, then constructing (1,2)-robust combiners for commitments is easy. The same observation holds for OT: it is easy to construct a (1,2)-robust OT-combiner if two candidate OTs are unconditionally secure for the sender (or receiver). Of course, while such combiners are very simple and efficient, they have somewhat limited applicability, as they require more knowledge about the input candidates. But given the apparent difficulty of efficient *general* (1,2)-robust combiners for primitives like OT or BC, a possible approach to obtain more practical combiners might be to consider constructions for "mixed" candidates, e.g., combiners that combine an unconditionally hiding bit commitment with an unconditionally binding one.

While the basic PIR-combiner we propose is applicable to many PIR protocols described in the literature, it is not universal in the sense that it does not work for *any non-trivial* PIR schemes — the combiners requires one *two-message* PIR and some bounds on communication complexities. It would be interesting to either find a universal combiner that does not need such assumptions or to further optimize the current combiner while maintaining its applicability.

An intermediate step towards universal (1,2)-robust PIR-combiners might be a construction of an universal (2,3)-robust PIR-combiner. Oblivious transfer and bit commitment, primitives considered to be hard to combine with (1,2)-robust combiners, do have very efficient universal (2,3)-robust combiners.

With regard to "cross-primitive" combiners, we have argued that there exists a PIR-to-BC combiner which yields statistically hiding bit commitment, and that there exists one yielding statistically binding bit commitment. However, for the later only an inefficient, generic construction via combiner for one-way functions is known. It would be interesting to find a more efficient, direct PIR-to-BC combiner yielding a statistically binding bit commitment scheme.

Acknowledgements. We would like to thank anonymous referees for useful comments and for pointing out the reference [DIO01].

References

[AB81] C. A. Asmuth and G. R. Blakely. An efficient algorithm for constructing a cryptosystem which is harder to break than two other cryptosystems. *Computers and Mathematics with Applications*, 7:447–450, 1981.

[BIKM99] A. Beimel, Y. Ishai, E. Kushilevitz, and T. Malkin. One-way functions are essential for single-server private information retrieval. In *Proc. ACM STOC*, pages 89–98, 1999.

[BKY03] D. Bleichenbacher, A. Kiayias, and M. Yung. Decoding of interleaved Reed-Solomon codes over noisy data. In *Proc. ICALP 2003*, pages 97–108, 2003.

[Cha04] Y.-C. Chang. Single database private information retrieval with logarithmic communication. In *Proc. Information Security and Privacy: 9th Australasian Conference, ACISP 2004*, pages 50–61, 2004.

[CK88] C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *Proc. IEEE FOCS '88*, pages 42–52, 1988.

[CKGS98] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

[CMS99] C. Cachin, S. Micali, and M. Stadler. Computationally private information retrieval with polylogarithmic communication. In *Proc. Eurocrypt '99*, pages 402–414, 1999.

[Cré87] C. Crépeau. Equivalence between two flavours of oblivious transfers. In *Proc. Crypto '87*, pages 350–354, 1987.

[DIO01] G. Di Crescenzo, Y. Ishai, and R. Ostrovsky. Universal service-providers for private information retrieval. *Journal of Cryptology*, 14(1):37–74, 2001.

[DK05] Y. Dodis and J. Katz. Chosen-ciphertext security of multiple encryption. In *Proc. TCC '05*, pages 188–209, 2005.

[DMO00] G. Di Crescenzo, T. Malkin, and R. Ostrovsky. Single database private information retrieval implies oblivious transfer. In *Proc. Eurocrypt '00*, pages 122–138, 2000.

[EG85] S. Even and O. Goldreich. On the power of cascade ciphers. *ACM Trans. Comput. Syst.*, 3(2):108–116, 1985.

[EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, 1985.

[Fis02] M. Fischlin. On the impossibility of constructing non-interactive statistically-secret protocols from any trapdoor one-way function. In *Proc. CT-RSA*, pages 79–95, 2002.

[Gas04] W. I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004.

[Gol04] O. Goldreich. *The Foundations of Cryptography*, volume II, Basic Applications. Cambridge University Press, 2004.

[Hai04] I. Haitner. Implementing oblivious transfer using collection of dense trapdoor permutations. In *Proc. TCC'04*, pages 394–409, 2004.

[Her05] A. Herzberg. On tolerant cryptographic constructions. In *CT-RSA*, pages 172–190, 2005. full version on Cryptology ePrint Archive, eprint.iacr.org/2002/135.

[HKN+05] D. Harnik, J. Kilian, M. Naor, O. Reingold, and A. Rosen. On robust combiners for oblivious transfer and other primitives. In *Proc. Eurocrypt '05*, pages 96–113, 2005.

[HL05] S. Hohenberger and A. Lysyanskaya. How to securely outsource cryptographic computations. In *Proc. TCC '05*, pages 264–282, 2005.

[IR89] R. Impagliazzo and S. Rudich. Limits on the provable consequences of one-way permutations. In *Proc. ACM STOC*, pages 44–61, 1989.

[KO97] E. Kushilevitz and R. Ostrovsky. Replication is not needed: Single database, computationally-private information retrieval. In *Proc. IEEE FOCS '00*, pages 364–373, 1997.

[KO00] E. Kushilevitz and R. Ostrovsky. One-way trapdoor permutations are sufficient for non-trivial single-server private information retrieval. In *Proc. Eurocrypt '00*, pages 104–121, 2000.

[KY01] A. Kiayias and M. Yung. Secure games with polynomial expressions. In *Proc. ICALP 2001*, pages 939–950, 2001.

[Lip05] H. Lipmaa. An oblivious transfer protocol with log-squared communication. In *Proc. Information Security, 8th International Conference, ISC 2005*, pages 314–328, 2005.

[Nao91] M. Naor. Bit commitment using pseudorandomness. *J. Cryptology*, 4(2):151–158, 1991.

[Rab81] M. O. Rabin. How to exchange secrets by oblivious transfer., 1981. Tech. Memo TR-81, Aiken Computation Laboratory, available at eprint.iacr.org/2005/187.

Appendix

For completeness, in Figure 4 we present the construction of OT (unconditionally secure for the sender) based on single-database PIR, due to Di Crescenzo *et al.* [DMO00]. This construction is used in our (1,2)-robust PIR-to-OT combiner (see Section 4.2).

Note that in this protocol the privacy of sender holds only against *honest-but-curious* receiver. It can however be transformed into a protocol resilient against arbitrary (possibly dishonest) parties [DMO00].

SENDER'S INPUT: two bits b_0, b_1
RECEIVER'S INPUT: choice bit c
COMMON INPUTS: PIR protocol, security param. κ, a param. m polynomial in κ

1. Sender and Receiver invoke m executions of PIR, with Sender as server and Receiver as user:
   for each execution $j \in [m]$ they pick independent, uniformly random inputs: Sender a string $x^j \in_R \{0,1\}^\kappa$, Receiver an index $i^j \in_R [\kappa]$
2. Receiver sets $(i_c^1, ..., i_c^m) := (i^1, ..., i^m)$, and picks random $(i_{1-c}^1, ..., i_{1-c}^m) \in [\kappa]^m$
3. Receiver sends $(i_0^1, ..., i_0^m)$ and $(i_1^1, ..., i_1^m)$ to Sender
4. Sender computes
   $$z_0 := b_0 \oplus x^1(i_0^1) \oplus ... \oplus x^m(i_0^m)$$
   and
   $$z_1 := b_1 \oplus x^1(i_1^1) \oplus ... \oplus x^m(i_1^m)$$
   (where $x^j(i)$ denotes the i-th bit of string x^j), and sends z_0, z_1 to Receiver
5. Receiver computes his output $b_c := z_c \oplus x^1(i^1) \oplus ... \oplus x^m(i^m)$

Fig. 4: Construction of (honest receiver) OT from single-database PIR [DMO00].