each answer residue by a random quadratic residue modulo N with Jacobi symbol 1. This gives the promised honest-user version. In order to deal with dishonest users, the user can be required to prove in zero-knowledge that its queries validly correspond to some index i. Since the queries validity predicate is in NP, standard zero-knowledge proof techniques can be used to obtain multi-round SPIR schemes with a (theoretically insignificant) polynomial communication blowup. □

- The scheme $\mathcal{B}'_2$ may serve as basis for the recursion (in contrast to the original PIR setting, in which it was possible to use a trivial scheme $\mathcal{B}_1$ as basis).

Proof of Theorem 6. As in the proof of Theorem 4, we follow the recursive construction of [1]. Let $d = 2k-1$ and $\ell = n^{1/d}$. Suppose we have a $(k-1)$-database SPIR scheme $\mathcal{B}''_{k-1}$ of communication complexity $O(\log n \cdot n^{1/(2k-3)})$. In this case we make an additional assumption on $\mathcal{B}''_{k-1}$: we assume that the user is required to *commit* to the index being retrieved. This assumption is defined in the following: We say that a 1-round PIR scheme $\mathcal{P}$ satisfies the *strong data-privacy requirement* with parameter d', if the following conditions hold:

1. On a data string x of length $n' = \ell^{d'}$, the user sends special queries Q^0_m, Q^1_m, $m \leq 1 \leq d'$ (each of which is a $(\log_2 \ell)$-bit string); and
2. If a user (possibly a dishonest user) sends queries in which $Q^0_m \oplus Q^1_m = \mathrm{bin}(i'_m)$ for each $1 \leq m \leq d'$, then the answers reveal at most the bit $x_{(i'_1,\ldots,i'_{d'})}$.

Note that strong data privacy implies the usual data privacy. Also note that the scheme $\mathcal{B}''_2$ satisfies this stronger requirement with $d' = 3$, since in that scheme the user can only obtain information on the data bit corresponding to index components i_1, i_2, i_3 it shares between the two databases. Our additional assumption on $\mathcal{B}''_{k-1}$ (which will be carried on to $\mathcal{B}''_k$) is that it satisfies the strong data privacy requirement with $d' = 2(k-1)-1 = 2k-3$. The scheme $\mathcal{B}''_k$ proceeds as follows:

- The user sends the subcube $C_{0^d} = (S^0_1, \ldots, S^0_d)$ to $\mathcal{DB}_1$ and $C_{1^d} = (S^1_1, \ldots, S^1_d)$ to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$. Independently, the user shares *binary* representations of the index components i_m, $m = 1, 2, \ldots, d$ (as in the scheme $\mathcal{B}''_2$). In their answers, $\mathcal{DB}_1, \mathcal{DB}_2$ disclose each j-th bit of a PSM message string w_{e_m}, $1 \leq j \leq \ell, 1 \leq m \leq d$, subject to the condition $i_m = j$ (where e_m denotes the m-th unit vector of length d). In addition, $\mathcal{DB}_1$ sends the bit $w_{0^d} \oplus s$, where s is a shared random bit.
- For each $\sigma \in \{0,1\}^d$ such that weight$(\sigma) \geq 2$, the user and the databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ recursively invoke $\mathcal{B}''_{k-1}$ on the virtual data string w_σ defined in the following. Let $d' = d-2$ and $n' = \ell^{d'}$. Let m^σ_z, $1 \leq z \leq$ weight(σ), denote the position of the z-th zero in σ. With every σ such that weight$(\sigma) \geq 2$ and tuple $i' = (i'_1, \ldots, i'_{d'}) \in [\ell]^{d'}$ we associate a subcube $C^\sigma_{i'}$ (of the cube $[\ell]^d$), which is obtained from C_{1^d} by replacing each set S^1_z, $1 \leq z \leq$ weight(σ), with the set $S^1_z \oplus i_{m^\sigma_z}$. Each w_σ is defined to be the n'-bit string, whose i'-th bit is equal to the exclusive-or of data bits residing in the subcube $C^\sigma_{i'}$ together with the PSM random bit r_σ. In a recursive invocation of $\mathcal{B}''_{k-1}$ on the virtual data string w_σ, the user retrieves the bit whose index is represented by the d'-tuple $i'_\sigma = (i_{m^\sigma_1}, i_{m^\sigma_2}, i_{m^\sigma_p}, 1, \ldots, 1)$, where $p =$ weight(σ).
- The shared random bit s is disclosed subject to a conjunction of the following conditions:
  1. For every $3 \leq j \leq k$, the subcube sent to $\mathcal{DB}_j$ is equal to the subcube sent to $\mathcal{DB}_2$.
  2. The subcubes sent to $\mathcal{DB}_1, \mathcal{DB}_2$ are consistent with the components of the index i shared between these two databases (see $\mathcal{B}''_2$ for implementation details).
  3. For every $\sigma \in \{0,1\}^d$ such that weight$(\sigma) \geq 2$, the index i' shared by the user in the invocation of $\mathcal{B}''_{k-1}$ on w_σ (in accordance with the strong data privacy assumption made on $\mathcal{B}''_{k-1}$) is equal to i'_σ. This can be verified by comparing each component of i' with the corresponding component of i as shared by the user.

We start by analyzing the communication complexity. The combined size of all conditional disclosure formulas is $O(\ell \log \ell)$. Thus, the communication complexity is $c_k(n) = O(\ell \log \ell) + (2^d - d - 1) \cdot c_{k-1}(\ell^{d-2}) = O(\ell \log \ell) = O(\log n \cdot n^{1/(2k-1)})$.

The correctness and the user's privacy are straightforward to verify. It remains to show that the strong data privacy requirement also holds for $\mathcal{B}''_k$. We argue that if the user commits to an index $i = (i_1, \ldots, i_d)$ (by sharing its components between $\mathcal{DB}_1$ and $\mathcal{DB}_2$), then it can learn at most the bit x_i. As in the $\mathcal{B}'_k$ scheme, an *honest* user learns x_i alone. In order to learn something on other bits, a dishonest user must deviate from the scheme's specification either by sending to $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$ subcubes which don't meet the requirements imposed by i, or by trying to retrieve from the recursive invocations of $\mathcal{B}''_{k-1}$ different bits than those corresponding to i. The specified disclosure conditions and the strong data privacy assumption made on $\mathcal{B}''_{k-1}$ guarantee that in both of these cases, the user will learn nothing about s. By the use of the underlying PSM protocol and by the data-privacy of $\mathcal{B}''_{k-1}$, the answers to any choice of queries reveal at most the exclusive-or of 2^d bits (one selected from each list) together with s. Thus, learning nothing about s makes the user learn nothing at all. □

C Computational SPIR

In a computational setting, where requirements of strict equality between distributions are relaxed to computational indistinguishability requirements, our information-theoretic primitives may be replaced by more powerful cryptographic primitives. Since polynomial communication overhead is theoretically insignificant for current computational PIR schemes, we can obtain the following:

Theorem 8. *There exist a 1-round 2-database* computational *SPIR scheme of communication complexity $O(n^c)$ (for any $c > 0$, assuming the existence of a one-way function), and a multi-round 1-database computational SPIR scheme (alternatively, a 1-round 1-database computational honest-user-SPIR scheme) of communication complexity $O(n^c)$ (for any $c > 0$, relying on the quadratic residuosity assumption).*

Proof. In [12] it is shown that for every function in P there exists a *computational* PSM protocol of polynomial communication complexity, assuming the existence of a one-way function. The 2-database computational PIR scheme of [9] can have a communication complexity of $O(n^c)$, for any constant $c > 0$. Since both the reconstruction function and the validity of the user's queries in this scheme can be verified in polynomial time (uniformly in c), the techniques of Section 5 can be applied to obtain a 1-round SPIR scheme as required.

The proof for the other part of the theorem is outlined in [17]; for the sake of completeness, we include here a more detailed description. In the 1-database scheme of [17], the database's answers (both direct ones and those simulated via recursion) consist of k-bit residues modulo some composite number $N = pq$. It can be readily verified that if an honest user only learns the *quadratic character* of each such residue, then the answers reveal (information-theoretically) only the single bit x_i. Thus in the honest-user case, any extra information can be eliminated by having the database multiply

[5] M. Blum, P. Feldman, and S. Micali. Non-interactive zero-knowledge and its applications. In *Proc. of 20th STOC*, pages 103–112, 1988.
[6] G. Brassard, C. Crépeau, and J.-M. Robert. Information theoretic reductions among disclosure problems. In *Proc. of 18th STOC*, pages 168–173, 1986.
[7] G. Brassard, C. Crépeau, and M. Santha. Oblivious transfers and intersecting codes. In *IEEE Transaction on Information Theory, special issue on coding and complexity*. November 1996.
[8] D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols (extended abstract). In *Proc. of 20th STOC*, pages 11–19, 1988.
[9] B. Chor and N. Gilboa. Computationally private information retrieval. In *Proc. of 29th STOC*, pages 304–313, 1997.
[10] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *Proc. of 36th FOCS*, pages 41–50, 1995.
[11] C. Crépeau and J. Kilian. Achieving oblivious transfer using weakened security assumptions (extended abstract). In *Proc. of 29th FOCS*, pages 42–52, 1988.
[12] U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation (extended abstract). In *Proc. of 26th STOC*, pages 554–563, 1994.
[13] Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *Proc. of 5th ISTCS*, pages 174–183, 1997.
[14] M. Ito, A. Saito, and T. Nishizeki. Secret sharing schemes realizing general access structures. In *Proc. IEEE Global Telecommunication Conf., Globecom 87*, pages 99–102, 1987.
[15] M. Karchmer and A. Wigderson. On span programs. In *Proc. of 8th IEEE Structure in Complexity Theory*, pages 102–111, 1993.
[16] J. Kilian. Founding cryptography on oblivious transfer. In *Proc. of 20th STOC*, pages 20–31, 1988.
[17] E. Kushilevitz and R. Ostrovsky. Single-database computationally private information retrieval. In *Proc. of 38th FOCS*, 1997.
[18] R. Ostrovsky and V. Shoup. Private information storage. In *Proc. of 29th STOC*, pages 294–303, 1997.
[19] A. Shamir. How to share a secret. *Commun. ACM*, 22(6):612–613, June 1979.
[20] S. Skyum and L. Valiant. A complexity theory based on Boolean algebra. *JACM*, 32:484–502, 1985.

A Necessity of Shared Randomness

In this section we show that the addition of a shared randomness resource to the basic PIR setting is in a sense minimal.

Suppose we allow the databases to use *private* randomness in answering the user's queries, but we still do not allow them to interact without the mediation of the user (and in particular we do not allow them to share a random string unknown to the user). We argue that in this setting, (information-theoretic) SPIR cannot be implemented at all, even when the user is honest.

Claim 3. *There exists no (multi-round) k-database SPIR scheme without interaction between the databases, even if the databases are allowed to hold* private, *independent random inputs, and the user is honest.*

Proof. The strong privacy requirement implies that any single database $\mathcal{DB}_j$ cannot respond to the user's queries in a way that depends on the data string x. Formally, at any round the distribution of the database's answer given previous communication is the same under every x. For otherwise, this answer distribution must either not follow from x_i alone, thus violating the data's privacy, or alternatively reveal the index i on which it depends, thus violating the user's privacy. By independence of private random inputs held by different databases, this implies that the *joint* distribution of their answers, given previous communication, is independent of x. Fixing an index i, it follows by induction on the number of rounds that for any $w > 0$ the accumulated communication in the first w rounds is distributed independently of x. This implies that the user's output cannot depend on the value of x_i, contradicting the correctness requirement. □

B Proofs

Proof of Theorem 4. We show how to adapt the proof of Theorem 3 to the k-database generalization given in [1]. Let $d = 2k - 1$ and $\ell = n^{1/d}$. In the scheme $\mathcal{B}_k$, the k databases (denoted $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$) jointly emulate the 2^d databases of the d-dimensional elementary cube scheme. The scheme proceeds as follows. The user sends to $\mathcal{DB}_1$ the subcube C_{0^d} as in the elementary cube scheme, and sends to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ the subcube C_{1^d}. In its answers, $\mathcal{DB}_1$ emulates all databases $\mathcal{DB}_\sigma$ of the original scheme such that $\sigma \in \{0,1\}^d$ is at Hamming distance at most 1 from 0^d, similarly to the way such an emulation is done in the scheme $\mathcal{B}_2$. Simultaneously, the remaining databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ jointly emulate the remaining databases of the original schemes, namely all $\mathcal{DB}_\sigma$ such that σ contains at least two 1's. This is done using a constant number $(2^d - d - 1)$ of recursive invocations of the scheme $\mathcal{B}_{k-1}$ between the user and $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$. In each such invocation the user retrieves a single bit b_σ from a virtual data string, whose entries correspond to the different subcubes possibly sent to $\mathcal{DB}_\sigma$ in the elementary scheme (i.e., each bit of the virtual data strings equals the exclusive-or of data bits residing in such a potential subcube). By xoring $d+1$ bits selected from the answers of $\mathcal{DB}_1$ together with the $2^d - d - 1$ bits retrieved by the recursive invocations of $\mathcal{B}_{k-1}$, the user reconstructs x_i.

It is not hard to see that the technique we used for converting $\mathcal{B}_2$ into an honest-user-SPIR scheme $\mathcal{B}'_2$ of the same asymptotic complexity can be carried on recursively. Suppose we have a $(k-1)$-database honest-user-SPIR scheme $\mathcal{B}'_{k-1}$ of communication complexity $O(n^{1/(2k-3)})$. We can obtain a recursive definition of $\mathcal{B}'_k$ (in terms of $\mathcal{B}'_{k-1}$) from the recursive definition of $\mathcal{B}_k$, similarly to the way $\mathcal{B}'_2$ was obtained from $\mathcal{B}_2$. Omitting details which should be clear by now, $\mathcal{B}'_k$ proceeds as follows:

- The user sends the subcube C_{0^d} to $\mathcal{DB}_1$ and the subcube C_{1^d} to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$. Independently, the user shares all characteristic vectors χ_{i_m}, $m = 1, \ldots, d$, between $\mathcal{DB}_1$ and $\mathcal{DB}_2$. In their answers, $\mathcal{DB}_1, \mathcal{DB}_2$ disclose each bit of the PSM message strings w_σ held by $\mathcal{DB}_1$, subject to an appropriate condition on corresponding bits of the shares they received (as in the scheme $\mathcal{B}'_2$).
- The user and databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ recursively invoke $\mathcal{B}'_{k-1}$, as in the definition of $\mathcal{B}_k$, except that the virtual data strings a_σ used in $\mathcal{B}_k$ are replaced by the corresponding PSM message strings w_σ. I.e., every bit of a_σ (from which the bit b_σ is retrieved) is xored with the same shared random bit r_σ.

□

Remarks:

- The above scheme requires only one round of interaction. Since the conditional disclosure of each bit requires only a constant amount of communication, the communication complexity satisfies: $c_k(n) \leq (d+1) \cdot O(\ell) + (2^d - d - 1) \cdot c_{k-1}(\ell^{d-2}) = O(n^{1/(2k-1)})$.
- Databases $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$ do not have to *directly* apply the conditional disclosure primitive; the recursive use of $\mathcal{B}'_{k-1}$ "takes care" of exposing only a single bit b_σ per invocation.

Theorem 6. *For every* constant $k \geq 2$ *there exists a* k*-database SPIR scheme,* $\mathcal{B}''_k$*, of communication complexity* $O(\log n \cdot n^{1/(2k-1)})$.

5.2 A Polynomial Interpolation Scheme

The polynomial interpolation PIR scheme for $k = \log_2 n + 1$ databases from [10] (see also [2]) can be transformed into a SPIR scheme with the same number of databases, and with a *constant* factor of communication and randomness overhead.

Theorem 7. *There exists a* $\lceil \log_2 n + 1 \rceil$*-database SPIR scheme of communication complexity* $O(\log^2 n \cdot \log\log n)$.

Proof. We start by describing the elementary polynomial interpolation scheme. Suppose $n = 2^s$, and let $k = s + 1$ be the number of databases. Let $GF(q)$ be a finite field with at least $k + 1$ elements, and α_j, $1 \leq j \leq k$, be distinct, nonzero elements of $GF(q)$. With every index $i \in [n]$ we associate an s-tuple $\vec{i} = (i_1, i_2, \ldots, i_s) \in \{0,1\}^s$, corresponding to the binary representation of i. For each data string $x \in \{0,1\}^n$, let $p_x(y_1, \ldots, y_s)$ denote a multivariate degree-s polynomial such that $p_x(\vec{i}) = x_i$ for every $i \in [n]$ (such p_x can be easily constructed). The user picks a random s-tuple $\vec{c} = (c_1, \ldots, c_s) \in GF(q)^s$, and sends to each database $\mathcal{DB}_j$, $1 \leq j \leq k$, the query $\vec{u}^j = \alpha_j \cdot \vec{c} + \vec{i}$.[8] Each database $\mathcal{DB}_j$ replies with a single field element $a_j \stackrel{\triangle}{=} p_x(\vec{u}^j)$. The user reconstructs x_i by interpolation: if p' is the unique degree-s univariate polynomial (over $GF(q)$) such that $p'(\alpha_j) = a_j$ for every $1 \leq j \leq k$, then $x_i = p'(0)$. The communication complexity of this scheme is $O(\log^2 n \log\log n)$.

As noted in Subsection 4.2, the linearity of the reconstruction function allows to obtain a PSM-based honest-user-SPIR scheme with the same communication complexity. To prevent a dishonest user from obtaining any illegitimate information on x, we require the user to prove that its queries are consistent with some $\vec{i} \in \{0,1\}^s$ and $\vec{c} \in GF(q)^s$. Such a proof will consist of sharing each entry of $\vec{c}$ and $\vec{i}$, and its validation will consist of verifying that $\vec{i} \in \{0,1\}^s$ and that $\vec{u}^j = \alpha_j \cdot \vec{c} + \vec{i}$ for each $1 \leq j \leq k$.

We start by the following observation, which also yields a slight improvement to the original PIR scheme described above. Note that the user reconstructs x_i by computing some *fixed* linear combination over $GF(q)$ of the k field elements replied by the databases. Thus, as a first step, we can let each database multiply its original answer by the corresponding coefficient, so that reconstruction will consist of computing the *sum* of all answers over $GF(q)$. Then, if q is chosen to be a power of 2 ($q = 2^{\lceil \log_2(k+1) \rceil}$ suffices), it is enough for the databases to reply only with the "least significant bit" of each answer, and for the user to reconstruct x_i by taking the exclusive-or of the k answer bits. From now on we refer to this modified scheme. The SPIR scheme is formally described as follows:

QUERIES: The user sends to each database $\mathcal{DB}_j$ a query $\vec{u}^j$ as in the original scheme. In addition, the user picks random tuples $\vec{i}^0, \vec{i}^1, \vec{c}^0, \vec{c}^1 \in GF(q)^s$ such that $\vec{i}^0 + \vec{i}^1 = \vec{i}$ and $\vec{c}^0 + \vec{c}^1 = \vec{c}$, and sends $\vec{i}^0, \vec{c}^0$ to $\mathcal{DB}_1$ and $\vec{i}^1, \vec{c}^1$ to each of $\mathcal{DB}_2, \ldots, \mathcal{DB}_k$.

ANSWERS: Let $r_1, r_2, \ldots, r_k$ be independent, random bits (included in the databases' shared randomness), and let r denote their exclusive-or. Each database $\mathcal{DB}_j$ replies with $a'_j \stackrel{\triangle}{=} a_j \oplus r_j$, where a_j is its answer according to the modified scheme. In addition, the databases use their shared randomness to independently disclose the bit r, subject to the following conditions: (1) for every $3 \leq j \leq k$, the shares of $\vec{i}$ and $\vec{c}$ sent to $\mathcal{DB}_j$ must be identical to those sent to $\mathcal{DB}_2$; (2) for every $1 \leq m \leq s$, either $i^0_m + i^1_m = 0$ or $i^0_m + i^1_m = 1$ (where i^b_m denotes the m-th entry of the b-th share of $\vec{i}$); and finally (3) for every $1 \leq j \leq k$ and $1 \leq m \leq s$, it is required that $\alpha_j(c^0_m + c^1_m) + (i^0_m + i^1_m) = u^j_m$. Note that each of the atomic conditions can be expressed as equality between two elements of $GF(q)$, known to two different databases. For instance, if $j > 1$ then verifying the condition $\alpha_j(c^0_m + c^1_m) + (i^0_m + i^1_m) = u^j_m$ is equivalent to comparing $\alpha_j c^0_m + i^0_m$, which is known to $\mathcal{DB}_1$, and $u^j_m - \alpha_j c^1_m - i^1_m$, which is known to $\mathcal{DB}_j$. Thus each such atomic condition can be verified by a Boolean formula of size $O(\log s)$, and the whole disclosure condition by a formula of size $O(s^2 \log s)$. It follows that conditionally disclosing the bit r requires $O(s^2 \log s) = O(\log^2 n \cdot \log\log n)$ communication bits.

RECONSTRUCTION: The user reconstructs r, and computes x_i as the exclusive-or of $a'_1, \ldots, a'_k$ and r.

The correctness and the user's privacy of the original scheme are clearly maintained. To see the data privacy of this scheme, consider two possible cases. If the user's queries are valid, then the tuple $(a'_1, a'_2, \ldots, a'_k, r)$ is uniformly distributed among all $(k+1)$-tuples over $GF(2)$ which add up to x_i, implying that the answer distribution depends only on x_i. Otherwise, the user obtains no information on r, and consequently $a'_1, \ldots, a'_k$ (which are uniformly and independently distributed over $GF(2)$) are independent of the conditional disclosure messages. It follows that in the latter case the user learns nothing at all. The communication complexity is dominated by the conditional disclosure of r, which requires $O(\log^2 n \cdot \log\log n)$ communication bits. □

[8] Throughout this proof, all additions and multiplications are over $GF(q)$.

6 Final Remarks

The techniques we have used here can be applied to obtain SPIR analogues of other PIR-related results which were not discussed in this work. In particular, efficient SPIR schemes for retrieving *blocks* of data, and schemes with higher degree of user privacy (i.e., privacy with respect to any coalition of at most t databases) may be obtained by applying similar techniques to schemes from [10].

Acknowledgments

We are grateful to Shafi Goldwasser for many enlightening discussions. We would also like to thank Madhu Sudan for helpful comments on an earlier version of this paper.

References

[1] A. Ambainis. Upper bound on the communication complexity of private information retrieval. In *Proc. of 24th ICALP, to appear*, 1997.

[2] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *STACS*, 1990.

[3] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *Proc. of 20th STOC*, pages 1–10, 1988.

[4] J. Benaloh and J. Leichter. Generalized secret sharing and monotone functions. In *Proc. of CRYPTO '88*, pages 27–35, 1990.

PSM protocol for XOR, the joint distribution of these 8 bits depends only on their exclusive-or, which equals x_i. Finally, by the independence of the PSM randomness and the conditional disclosure randomness, the user's view given its queries depends only on x_i. (This argument may be formalized by describing a simulator which, given an honest user's queries and a value for x_i, produces the answers distribution).

Since the condition for disclosing each of the $O(n^{1/3})$ bits of the strings w_j is expressed by a Boolean formula of a constant size, it follows from Corollary 1 that all such bits can be conditionally disclosed with a total communication cost of $O(n^{1/3})$ bits. Altogether, the communication complexity of the scheme is $O(n^{1/3})$, as required. □

The next theorem, whose proof appears in Appendix B, generalizes Theorem 3 to any number of databases $k \geq 2$.

Theorem 4. *For every* constant $k \geq 2$ *there exists a k-database honest-user-SPIR scheme, $\mathcal{B}'_k$, of communication complexity $O(n^{1/(2k-1)})$.*

Note that the methodology of using conditional disclosure of secrets on top of a PSM protocol may be useful for handling *any* scheme whose reconstruction depends only on a proper subset of the answer bits. In particular, this applies to schemes obtained from the generic communication balancing techniques of [10].

5 Specific SPIR Schemes with Respect to Dishonest Users

In the previous section we were concerned with an honest but curious user. In this section we construct SPIR schemes which guarantee data privacy with respect to *dishonest* users as well. The following example demonstrates the undesired extra power granted to a dishonest user in ordinary PIR schemes and in the honest-user-SPIR schemes constructed above.

Example 2. *Consider the scheme $\mathcal{B}_2$. By sending the subcube $C = (\{i_1\}, \{i_2\}, \{i_3\})$ (which is a legitimate query) to the 1st database, its answers alone reveal about $3n^{1/3}$* physical *bits of data. Namely, the data bits revealed are all those whose Hamming distance from (i_1, i_2, i_3) is at most 1. Although an honest user might also be lucky enough to learn that many physical data bits, this occurs with only a negligible probability. Moreover, even in the scheme $\mathcal{B}'_2$ (which maintains data privacy with probability 1 for an honest user), a dishonest user may similarly obtain $O(n^{1/3})$ physical data bits, by sharing the all-ones vector (instead of a characteristic vector χ_{i_m}).*

Generally, a dishonest user can cheat in two ways in our honest-user-SPIR protocols: in the sharing of its index, and by sending invalid queries in the original PIR scheme (which may make the output of the reconstruction function depend on more than one bit of data, even when the input to the function is successfully hidden). In principle, every honest-user-SPIR scheme can be made resilient to dishonest users by filtering every original answer bit using the conditional disclosure primitive, where the condition tests for the validity of all the user's queries. However, the complexity of disclosing each answer bit subject to a full validity test will usually be prohibitive. We significantly reduce this overhead by requiring the user to send a "proof" for the validity of its queries (which may be viewed as switching to *nondeterministic* verification), and by letting different answer bits be disclosed subject to different conditions. The next subsections present specific transformations from PIR schemes into corresponding SPIR schemes. All transformations involve at most an $O(\log n)$ multiplicative communication overhead.

5.1 Cube Schemes

In this subsection we construct a k-database SPIR scheme of complexity $O(\log n \cdot n^{1/(2k-1)})$. As in the previous section, we first address the 2-database case.

Theorem 5. *There exists a 2-database SPIR scheme, $\mathcal{B}''_2$, of communication complexity $O(\log n \cdot n^{1/3})$.*

Proof. Assume that $\ell = n^{1/3}$ is a power of 2. For every $j \in [\ell]$, let $\mathrm{bin}(j)$ denote the $(\log_2 \ell)$-bit binary representation of j. The scheme $\mathcal{B}''_2$ proceeds as follows:

QUERIES: The user sends to $\mathcal{DB}_{000}$ the subcube $C_{000} = (S^0_1, S^0_2, S^0_3)$ and to $\mathcal{DB}_{111}$ the subcube $C_{111} = (S^1_1, S^1_2, S^1_3)$, as in the scheme $\mathcal{B}_2$. In addition, the user independently shares *binary* representations of the index components i_m, $m = 1, 2, 3$. This is done by picking random $(\log_2 \ell)$-bit strings r^0_m, r^1_m such that $r^0_m \oplus r^1_m = \mathrm{bin}(i_m)$, and sending the strings r^0_m to $\mathcal{DB}_{000}$ and the strings r^1_m to $\mathcal{DB}_{111}$.

ANSWERS: Let w_σ denote the PSM message strings as in the scheme $\mathcal{B}'_2$, and let s be a shared random bit. The databases use their shared randomness to independently disclose to the user each of the following bits:

1. The bit s is disclosed subject to the condition $\bigwedge_{m=1}^{3}(S^0_m \oplus S^1_m = \{r^0_m \oplus r^1_m\})$ (which validates the user's queries). This condition can be verified by a Boolean formula of size $O(\ell \log \ell)$.[7]
2. The bits $w_{000} \oplus s$ and w_{111} are sent in a plain form. The former bit may be viewed as the PSM message corresponding to the input bit $b_{000} \oplus s$.
3. Each bit of w_σ ($\sigma \neq 000, 111$) is disclosed subject to the appropriate comparison of its position to the shared index components; e.g., the j-th bits of w_{001} and w_{110}, $1 \leq j \leq \ell$, are disclosed subject to the condition $r^0_3 \oplus r^1_3 = \mathrm{bin}(j)$. Each such condition can be verified by a Boolean formula of size $O(\log \ell)$.

RECONSTRUCTION: The honest user can reconstruct s and the 8 bits corresponding to the index i. The retrieved bit x_i is equal to the exclusive-or of these 9 bits.

The correctness and the user's privacy in the above scheme are easy to verify. The scheme's data privacy, relative to *any* user, follows from the following observations:

- From each of the 6 ℓ-bit strings w_σ the user can obtain only a *single physical bit* of w_σ, corresponding to the appropriate shared index component. By the use of an underlying PSM protocol for XOR, this means that the user can only learn $(s \oplus b_{000}) \oplus b_{111} \oplus b$, where b is the exclusive-or of the 6 selected bits.
- If the user's queries are inconsistent, i.e. the relation between the two subcubes does not match the shared index components, then the user obtains no information on s, and hence (by the previous observation) no information at all. On the other hand, if the user's queries are consistent with some index i, then it learns only s and $s \oplus x_i$, which is equivalent to learning x_i.

From the sizes of the Boolean formulas used as disclosure conditions it follows that the scheme meets the specified complexity bound. □

Theorem 5 is generalized by the following theorem, whose proof appears in Appendix B.

[7] A condition of the form $S \oplus S' = \{r \oplus r'\}$ can be expressed as a conjunction of ℓ conditions of the form $((\chi_S)_j \oplus (\chi_{S'})_j = 1) \leftrightarrow (r \oplus r' = \mathrm{bin}(j))$, where each such smaller condition can be easily verified by a formula of size $O(\log \ell)$.

honest-user-SPIR scheme $\mathcal{S}$ whose communication complexity is $(\alpha_k(n), c_k(\beta_k(n)))$.

Proof. A scheme $\mathcal{S}$ of the specified complexity can be obtained from $\mathcal{P}$ as follows. The databases compute their answers as in $\mathcal{P}$, but instead of sending back their answers, they use their shared randomness to simulate the PSM computation of the reconstruction function, f. The correctness and privacy of $\mathcal{S}$ follow from the correctness and privacy of $\mathcal{P}$ and of the PSM protocol for f. □

Whether this approach will be useful for our reduction or not depends on the PSM complexity of the reconstruction function f of the PIR scheme.. In particular, to maintain the same communication complexity, we need a PSM solution for f that will be linear (or "almost" linear) in the size of its input.

Luckily, the reconstruction functions in some currently known PIR schemes have the lowest PSM complexity possible, allowing their transformation into honest-user-SPIR schemes with the same communication complexity. This is illustrated by the following example.

Example 1. ***PSM-based honest-user-SPIR scheme for the basic d-dimensional cube scheme.*** *Consider the basic d-dimensional cube scheme from Subsection 4.1. This scheme does not maintain data privacy, since the user learns the exclusive-or of k different subsets of data bits. In this case, the extra information can be eliminated by applying the following simple modification to the scheme. Instead of sending the original answer b_σ, each database $\mathcal{DB}_\sigma$ will send a* masked *answer $b_\sigma \oplus r_\sigma$, where $r = r_{0\cdots00} r_{0\cdots01} \cdots r_{1\cdots10}$ is a $(k-1)$-bit* shared *random string, and $r_{1\cdots11}$ is computed as the exclusive-or of the bits of r. (Alternatively, $r_{0\cdots00}, r_{0\cdots01}, \ldots, r_{1\cdots11}$ can be randomly chosen from the k-tuples whose bits XOR to 0). Under the modified scheme, an honest user's view is uniformly distributed among all k-tuples whose XOR is $\bigoplus_{\sigma\in\{0,1\}^d} b_\sigma$, which by the scheme's correctness equals the physical bit x_i.*[6]

By an easy generalization of Example 1, any PIR scheme with a *linear* reconstruction function can be transformed into an honest-user-SPIR scheme of the same communication cost. This is the case with the polynomial-interpolation schemes (for a $O(\log n)$ number of databases) of [10, 2], for which (dishonest-user) SPIR counterparts of the same asymptotic complexity will be given in Subsection 5.2. Note, however, that in order to use a PSM protocol for evaluating the reconstruction function, this function must depend only on the answers computed by the databases. This is *not* the case with the scheme $\mathcal{B}_2$, for instance, where reconstruction heavily depends on the index i held by the user. However, the scheme $\mathcal{B}_2$ (or any other PIR scheme) can be easily augmented to satisfy this requirement, e.g. by letting the user share the index i between two databases, so that its privacy is not violated, and concatenating these shares to the answers; however, the reconstruction function in the resultant scheme is not linear. Since even some of the simplest Boolean functions (including functions with linear formula size) are not known to have linear PSM complexity, using a PSM protocol for computing non-linear reconstruction functions may result in considerable communication overhead. In the next subsection we show how conditional disclosure of secrets may be used together with PSM protocols, in order to solve this problem for schemes of a certain structure, and in particular for the k-database scheme $\mathcal{B}_k$.

[6] Notice that even in this simple case, a dishonest user may obtain information that doesn't follow from any physical bit of data. Specifically, by choosing the cubes C_σ not according to the scheme's specification, a user may obtain the exclusive-or of different (possibly large) subsets of data bits of his choice. Such a dishonest user is treated in Section 5.

4.3 SPIR Schemes Based on Conditional Disclosure of Secrets and PSM

In this section an honest-user-SPIR scheme is constructed from the k-database $\mathcal{B}_k$ scheme of [1], by combining the use of a PSM protocol with the use of conditional disclosure of secrets. As a basis for a recursive construction, we initially show that the 2-database scheme $\mathcal{B}_2$ can be transformed into an honest-user-SPIR scheme $\mathcal{B}'_2$ of the same asymptotic complexity.

Theorem 3. *There exists a 2-database honest-user-SPIR scheme, $\mathcal{B}'_2$, of communication complexity $O(n^{1/3})$.*

Proof. Recall the reconstruction function of $\mathcal{B}_2$, which may be viewed as a two-stage procedure: (1) the user selects a *single* bit from each of 8 answer strings, depending only on the index i; and (2) the user exclusive-ors the 8 bits it has selected to obtain x_i. Thus, if we let the honest user learn only the exclusive-or of the 8 bits corresponding to i, the data privacy requirement will be met. This can be achieved by using the conditional disclosure of secrets primitive *on top* of a PSM protocol computing the exclusive-or of 8 bits. The scheme $\mathcal{B}'_2$, an honest-user-SPIR version of $\mathcal{B}_2$, proceeds as follows:

QUERIES: The user sends the subcubes C_{000} to $\mathcal{DB}_{000}$ and C_{111} to $\mathcal{DB}_{111}$, as in the scheme $\mathcal{B}_2$. In addition, the user independently shares the characteristic vectors χ_{i_m}, $m = 1, 2, 3$, among the two databases. This is done by picking random ℓ-bit strings r^0_m, r^1_m such that $r^0_m \oplus r^1_m = \chi_{i_m}$ and sending the three strings r^0_m to $\mathcal{DB}_{000}$ and the three strings r^1_m to $\mathcal{DB}_{111}$.

ANSWERS: Each of the two databases computes 4 answer strings as in the $\mathcal{B}_2$ scheme. Denote by a_σ the answer string emulating $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^3$. The databases treat each bit of a string a_σ as an input to a PSM protocol computing the XOR of 8 bits, and using their shared randomness they compute the PSM message sent for each such bit. Under the simple PSM protocol for XOR (see Example 1), each such message consists of a single bit. Let w_σ denote the string obtained by replacing each bit from a_σ by its corresponding PSM message bit. In this case, w_σ is obtained by XOR-ing every bit of a_σ with the same random bit r_σ, where the bits $\{r_\sigma\}$ are 8 random bits whose XOR is 0. Finally, for every $\sigma \in \{0,1\}^3$ and $1 \le j \le |w_\sigma|$, the databases use their shared randomness to disclose to the user the j-th bit of w_σ, $(w_\sigma)_j$, subject to an appropriate condition. For $\sigma = 100, 011$ the condition is $(r^0_1)_j \oplus (r^1_1)_j = 1$, for $\sigma = 010, 101$ it is $(r^0_2)_j \oplus (r^1_2)_j = 1$, and for $\sigma = 001, 110$ it is $(r^0_3)_j \oplus (r^1_3)_j = 1$. The single bits w_{000}, w_{111} can be sent in a plain form.

RECONSTRUCTION: The user reconstructs the eight PSM message bits corresponding to the index i (using the reconstruction function of the conditional disclosure protocol), and computes their exclusive-or to obtain x_i.

The correctness of the above scheme and the user's privacy are easy to verify. We now show that the scheme meets the data privacy requirement with respect to an honest user. By the use of conditional disclosure of secrets, the answer strings disclose to the user only 8 bits: a single bit from each w_j, whose position corresponds to the appropriate characteristic vector shared by the user. By the privacy of the underlying

4 Specific SPIR Schemes with Respect to Honest Users

In this section we construct honest-user-SPIR schemes which perform as well as their PIR counterparts, up to a multiplicative constant, both in terms of communication and randomness. Then, the honest-user version is used to construct SPIR schemes for dishonest users, with at most a logarithmic complexity overhead. Our reductions rely on two primitives: private simultaneous messages protocols (described below), and conditional disclosure of secrets (above). To save space, we only refer to the communication complexity in the statement and proofs of the results; it may be readily verified that the same upper bounds apply to randomness complexity (both user's randomness and shared randomness) as well.

Since our schemes rely on specific PIR schemes, we first give a brief overview of some of their details which are important to our constructions.

4.1 The PIR schemes

We now review an elementary PIR scheme from [10], denoted *basic cube scheme*. This scheme is the basis for the scheme $\mathcal{B}_2$ from [10], also described below, which in turn serves as basis for the recursive scheme $\mathcal{B}_k$ from [1].

Basic d-dimensional Cube Scheme: This is a PIR scheme for $k = 2^d$ databases. Assume wlog that the database size is $n = \ell^d$, where ℓ is an integer. The index set $[n]$ can then be identified with the d-dimensional cube $[\ell]^d$, in which each index $i \in [n]$ can be naturally identified with a d-tuple $(i_1, \ldots, i_d)$. A d-dimensional *subcube* is a subset $S_1 \times \cdots \times S_d$ of the d-dimensional cube, where each S_i is a subset of $[\ell]$. Such a subcube is represented by the d-tuple $C = (S_1, \ldots, S_d)$. The $k(= 2^d)$ databases will be indexed by all binary strings of length d. The scheme proceeds as follows.

QUERIES: The user picks a random subcube $C = (S_1^0, \ldots, S_d^0)$, where $S_1^0, \ldots, S_d^0$ are independent, random subsets of $[\ell]$. Let $S_m^1 = S_m^0 \oplus i_m$ $(1 \le m \le d)$. For each $\sigma = \sigma_1\sigma_2\cdots\sigma_d \in \{0,1\}^d$, the user sends to database $\mathcal{DB}_\sigma$ the subcube $C_\sigma = (S_1^{\sigma_1}, \ldots, S_d^{\sigma_d})$, where each set $S_m^{\sigma_m}$ is represented by its characteristic ℓ-bit string.

ANSWERS: Each database $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^d$, computes the exclusive-or of the data bits residing in the subcube C_σ, and sends the resultant bit b_σ to the user.

RECONSTRUCTION: The user computes x_i as the exclusive-or of the k bits b_σ it has received.

The scheme's correctness follows from the fact that every bit in x except x_i appears in an even number of subcubes C_σ, $\sigma \in \{0,1\}^d$, and x_i appears in exactly one such subcube (see [10] for details). The communication complexity of this 2^d-database scheme is $n^{1/d}$, much worse than the following scheme $\mathcal{B}_2$ and its generalization $\mathcal{B}_k$.

The scheme $\mathcal{B}_2$: This scheme may be regarded as a 2-database implementation of the 8-database 3-dimensional cube scheme described above. Let $\ell = n^{1/3}$, and let $i = (i_1, i_2, i_3)$ be the index of the data bit being retrieved. Each of the two databases $\mathcal{DB}_{000}$ and $\mathcal{DB}_{111}$ emulates the 4 databases $\mathcal{DB}_\sigma$, $\sigma \in \{0,1\}^3$, such that the Hamming distance of σ from its index is at most 1. This is done in the following way. The user sends to $\mathcal{DB}_{000}$ the subcube $C_{000} = (S_1^0, S_2^0, S_3^0)$ and to $\mathcal{DB}_{111}$ the subcube $C_{111} = (S_1^1, S_2^1, S_3^1)$ as in the basic cube scheme. We would like the answers of each of the two databases to include the 4 answer bits of the 4 databases it emulates. To this end, $\mathcal{DB}_{000}$ replies with its own answer bit b_{000} along with 3 ℓ-bit long strings, each of which contains the answer bit of one of the other databases it emulates. For instance, the i_1'-th bit of the string emulating $\mathcal{DB}_{100}$ is obtained by computing the exclusive-or of all data bits residing in the subcube $(S_1^0 \oplus i_1', S_2^0, S_3^0)$, implying that the i_1-th bit in this string is equal to b_{100}. Symmetrically, $\mathcal{DB}_{111}$ sends the single bit b_{111} along with 3 ℓ-bit long strings, each of which corresponds to the subcubes obtained from C_{111} by "masking" the set S_m^1 with all ℓ possible values of i_m'. Altogether, the user receives 8 answer strings $a_\sigma, \sigma \in \{0,1\}^3$, six of which contain ℓ bits each, and the other two (namely, a_{000} and a_{111}) contain single bits. In each of the ℓ-bit long strings, the index of the required answer bit b_σ is either i_1 (for $\sigma = 100, 011$), i_2 ($\sigma = 010, 101$), or i_3 ($\sigma = 001, 110$). Since the user can locate all 8 bits b_σ, $\sigma \in \{0,1\}^3$, in the answer strings, it can reconstruct x_i by computing their exclusive-or.

4.2 SPIR Schemes Based on Private Simultaneous Messages (PSM) Protocols

In the PIR model, every scheme must reveal to the honest user some extra information, which does not follow from x_i, so that data privacy is not maintained (see Claim 3, Appendix A). In this section we identify the source of information leakage to an honest user, and fix it using the following idea. Consider any 1-round PIR scheme. The user, depending on the index i, produces k queries $q_1, \ldots, q_k$ (one for each database); it sends each of them to the corresponding database and in response receives k answer strings $a_1, \ldots, a_k$. Then, the user applies a *reconstruction function* f to obtain the desired bit x_i. Our idea is that it might be possible for the user to compute the output of f without actually getting the answers $a_1, \ldots, a_k$, which discloses other information as well, but rather some other messages $m_1, \ldots, m_k$ that keep the privacy of the string x.

Precisely this idea is captured by the model of private computation introduced by [12] and further studied in [13]. We call this model the *Private Simultaneous Messages (PSM)* model. In this model there are k players, each player P_j holding a private input string y_j, and an external referee called Carol. All players have access to a shared random input, which is unknown to Carol. Based on its private input y_j and the shared random input, each player P_j simultaneously sends a single message to Carol. From the messages she received, Carol should be able to compute some predetermined function $f(y_1, \ldots, y_k)$ of the inputs, but should obtain no additional information on the input other than what follows from the value of f.

The *PSM complexity* of f is the number of communication bits needed to privately compute the function f in such a way. We denote this complexity by $c_k(n)$, where n is the total number of input bits held by the k players. In [12, 13] several upper bounds on PSM complexity are obtained. These include an $O(k \cdot S(n)^2)$ bound for any Boolean function with a deterministic branching program of size $S(n)$ (for any partition of the n bits among the players), which in particular implies the same upper bound for every function with formula size $S(n)$.

We now give a general claim, which in principle allows to apply PSM solutions to any PIR scheme.

Claim 2. *Suppose $\mathcal{P}$ is a 1-round k-database PIR scheme with communication complexity $(\alpha_k(n), \beta_k(n))$, such that the reconstruction function f depends only on the answers, and has PSM complexity of $c_k(n)$. Then there exists a 1-round*

3.2 Conditional Disclosure of Secrets

In this section a new cryptographic primitive – conditional disclosure of secrets – is described, and an implementation of it using generalized secret sharing is proposed. This primitive is then used in the next subsection to obtain a general reduction to SPIR for *any* user behavior. It is also used in the other reductions in subsequent sections, and may be of independent interest as a building block in the design of more general cryptographic protocols.

The model of *conditional disclosure of secrets* consists of: the "condition" – a fixed Boolean function $h : \{0,1\}^n \to \{0,1\}$ for some n; k players $P_1, \ldots, P_k$; and an external party Carol. Carol holds an input string $y \in \{0,1\}^n$, which is also partitioned between the k players in an arbitrary way (i.e. each player holds some fixed subset of the bits of y). All k players have access to a shared random string, which is hidden from Carol. Finally, a secret input s (single bit unless otherwise mentioned) is known to at least one of the players, but is unknown to Carol. Based on its share of y and on the shared randomness, each player P_j simultaneously sends a message to Carol, such that: (1) if $h(y) = 1$, then Carol is able to reconstruct the secret s from her input y and the messages she received; and (2) if $h(y) = 0$, then Carol obtains no information (in the information-theoretic sense) about s.

The following claim and its corollary show how to implement conditional disclosure of secrets by reducing it to generalized secret sharing [4, 14] relative to a corresponding access structure.[3]

Claim 1. *Let $h_M : \{0,1\}^m \to \{0,1\}$ be a monotone Boolean function, and let $h : \{0,1\}^n \to \{0,1\}$ be a* projection *of h_M; that is, $h(y_1, \ldots, y_n) = h_M(g_1, \ldots, g_m)$, where each g_i is a function of a single variable y_j. Let $\mathcal{S}$ be a generalized secret sharing scheme realizing the access structure h_M, in which the total share size is c, and let s denote a secret (from the secret domain of $\mathcal{S}$) which is known to all k players. Then, there exists a protocol $\mathcal{P}$ for disclosing s subject to the condition h, whose communication complexity is bounded by c.*

Proof. Let $P'_1, \ldots, P'_m$ denote the participants in the generalized secret sharing scheme $\mathcal{S}$. The conditional disclosure protocol $\mathcal{P}$ proceeds as follows. Each player, holding input bits $(y_j)_{j \in T}$ for some $T \subseteq [n]$, uses the shared random input to compute (as specified by $\mathcal{S}$) the shares of all participants P'_i such that: (1) g_i is a function of some y_j that the player holds, *and* (2) $g_i(y_j) = 1$. All such shares are simultaneously sent to Carol, who uses her input y to determine whether $h_M(\vec{g}(y)) = 1$, and if so reconstructs the secret from the shares she received using the reconstruction function of $\mathcal{S}$.

The privacy and correctness of the conditional disclosure protocol follow from observing that the m-bit characteristic vector z of the shares received by Carol satisfies $z = \vec{g}(y)$, and so $h_M(z) = h_M(\vec{g}(y)) = h(y)$. Indeed, if $h(y) = 0$ then $h_M(z) = 0$, which by the privacy of $\mathcal{S}$ implies that the shares received by Carol give her no information about the secret. If $h(y) = 1$ then $h_M(z) = 1$, which by the correctness of $\mathcal{S}$ implies that Carol can reconstruct the secret. □

Corollary 1. *Suppose $h : \{0,1\}^n \to \{0,1\}$ has a Boolean formula of size $S(n)$,[4] and let s denote a secret bit known to at least one player. Then there exists a protocol for disclosing s subject to the condition h, whose total communication complexity is $S(n) + 1$.*

Proof. The players first conditionally disclose a shared random bit r, known to *all* of them, subject to the condition h. This is done as follows. Let H be a Boolean (AND-OR) formula for h over the literals $\{y_1, \overline{y}_1, \ldots, y_n, \overline{y}_n\}$, whose size is $S(n)$. Replacing each negative literal $\overline{y}_j$ with a positive literal w_j, we obtain a *monotone* Boolean formula H_M of size $S(n)$ computing a monotone function $h_M(y_1, \ldots, y_n, w_1, \ldots, w_n)$. Note that h is a projection of h_M, since $h(y_1, \ldots, y_n) = h_M(y_1, \ldots, y_n, \overline{y}_1, \ldots, \overline{y}_n)$. Using the generalized secret sharing scheme of [4] (or [15]), in which the total share size can be bounded by the size of a monotone Boolean formula representing the access structure, it follows from Claim 1 that the players can disclose the bit r subject to the condition h using $S(n)$ communication. Finally, a single player holding s simultaneously sends the bit $s \oplus r$ to Carol. Clearly, if Carol can reconstruct r then she can also reconstruct s, and if she obtains no information on r then she can obtain no information on s, and the corollary follows. □

We note that the number of shared random bits used in the conditional disclosure protocol with a condition of size $S(n)$ is no larger than $S(n)$.

3.3 A General Reduction with Respect to Dishonest Users

Using conditional disclosure of secrets, the following theorem gives a general reduction for a dishonest user.

Theorem 2. *Let $\mathcal{P}$ be any 1-round k-database PIR scheme with communication complexity $(\alpha_k(n), \beta_k(n))$. Then, there exists a 1-round, $(k+1)$-database SPIR scheme $\mathcal{S}_\mathcal{P}$ with complexity $(\alpha_k(n) + (k+1)\lceil \log_2 n \rceil, O(\beta_k(n) \log n))$.*

Proof. As noted in Subsection 3.1, the general reduction of Theorem 1 satisfies data privacy as long as the user sends to every $\mathcal{DB}_j$ the same shift s_j. Thus, to prevent the user from obtaining any information which does not follow from such legal queries, each database $\mathcal{DB}_j$, $2 \le j \le k$, will disclose each answer bit subject to the condition $s_j = s_1$ (where s_j is the $\lceil \log_2 n \rceil$-bit shift sent to $\mathcal{DB}_j$). Since each such condition can be verified by a formula of size $O(\log n)$, the scheme meets the specified complexity requirement.[5] □

Remark. Notice that even if the original PIR scheme $\mathcal{P}$ is t-private for some $t > 1$ (i.e., the user's privacy is maintained with respect to any coalition of at most t databases), the resultant SPIR scheme $\mathcal{S}_\mathcal{P}$ will still only be 1-private. Generalizing the construction of $\mathcal{S}_\mathcal{P}$, a *t-private* SPIR scheme $\mathcal{S}^t_\mathcal{P}$ can be obtained from any t-private PIR scheme $\mathcal{P}$ as follows. Instead of directly asking $\mathcal{DB}_0$ for the $(i+s)$-th bit of the shared random string r, the user retrieves this bit by recursively invoking the $(t-1)$-private SPIR scheme $\mathcal{S}^{t-1}_\mathcal{P}$ with a "fresh" set of databases. Note that as a basis $\mathcal{S}^0_\mathcal{P}$ for this recursion, we may take the trivial 1-database scheme in which the user explicitly asks for the desired index. Further details are omitted from this version.

In subsequent sections we present SPIR schemes which rely on specific structural properties of some underlying PIR schemes, and exploit them to outperform the above general transformation. In particular, they use less shared randomness, and do not require an auxiliary database.

[3] The problem of generalized secret sharing extends the usual notion of t-out-of-m secret sharing [19] in the following way. Instead of allowing every subset of the m participants whose size is at least t to reconstruct the secret, the collection of such qualified sets is specified by an *access structure*, which naturally corresponds to a monotone Boolean function $h_M : \{0,1\}^m \to \{0,1\}$ (i.e., $S \subseteq [m]$ is qualified iff $h_M(\chi_S) = 1$). See [4, 14] for a formal definition.

[4] Using the secret sharing scheme of [15], this can be generalized to any function h with a *span program* of size $S(n)$.

[5] For this special case of conditional disclosure of secrets, where the condition tests for *equality* between two strings, more direct and slightly more efficient solutions can be applied (e.g., via universal hashing). However, in subsequent sections the conditional disclosure primitive will be used to realize more complex conditions.

2 Notation and Definitions

The following notations and conventions are used throughout. By $[\ell]$ we denote the set $\{1, 2, \ldots, \ell\}$. For any sets $S, S' \subseteq [\ell]$, we let $S \oplus S'$ denote the symmetric difference between S and S' (i.e., $S \oplus S' = (S \setminus S') \cup (S' \setminus S)$), and χ_S denote the *characteristic vector* of S: an ℓ-bit binary string whose j-th bit is equal to 1 iff $j \in S$. To simplify notation, $S \oplus j$ and χ_j are used instead of $S \oplus \{j\}$ and $\chi_{\{j\}}$, respectively. For any $\sigma \in \{0,1\}^d$, weight(σ) denotes the number of nonzero entries in σ. By default, whenever referring to a *random* choice of an element from a finite domain A, the associated distribution is uniform over A, and is independent of all other random choices.

The following notations and definitions are specifically related to PIR and SPIR schemes. We let k denote the number of databases, $\mathcal{DB}_j$ denote the j-th database, x denote an n-bit data string which is held by all k databases, and i denote the position (also called *index*) of a data bit which the user wants to retrieve.

A *PIR* scheme is a randomized protocol, where in each round *queries* are sent from the user to each database, and *answers* are sent from each database to the user. At the end of the execution, the user applies some *reconstruction* function to the answers (possibly depending also on i and the user's private coin-tosses) to obtain the desired data bit x_i. Any (1-private, information-theoretic) PIR scheme must satisfy the following *user privacy* requirement: under any two indices i, i', the communication seen by any *single*[2] database is identically distributed.

A *SPIR* scheme is a PIR scheme in a setting which allows databases to access a *shared* random string not known to the user, such that in every invocation the user cannot obtain any information which doesn't follow from a single physical bit of data. Formally, this *data privacy* condition requires that for *any* behavior of a user (possibly a dishonest one, not following the scheme's specification) there exists an index i, such that the distribution of communication between the user and the (honest) databases is independent of x given x_i, i.e. is the same under any two data strings x, x' such that $x_i = x'_i$. An *honest-user-SPIR* scheme is a PIR scheme that satisfies the data-privacy requirement with respect to an *honest (but curious)* user, which follows the scheme's specification, but may try to deduce extra information from the communication.

By default, the terms "PIR scheme" and "SPIR scheme" refer to *1-round*, information-theoretically private schemes. Complexity is measured, by default, in terms of communication. The communication complexity of a k-database scheme will be denoted $(\alpha_k(n), \beta_k(n))$, where $\alpha_k(n)$ is the total number of query bits and $\beta_k(n)$ is the total number of answer bits on a data string of size n.

Finally, we use $\mathcal{B}_2$ to denote the 2-database covering-codes scheme from [10] (described in Subsection 4.1), and $\mathcal{B}_k$ to denote its k-database generalization from [1]. These schemes are the most efficient of their kind known to date.

3 A General Reduction from SPIR to PIR

In this section we present a construction for a SPIR scheme using *any* PIR scheme as a black box. This construction introduces a single auxiliary database, a logarithmic factor in communication complexity, and linear shared randomness. The auxiliary database *need not* hold a copy of the data x; it only needs to have access to the shared random string.

[2] A generalization to t-privacy is straightforward; however, all cited PIR works concentrate mainly on the case of 1-privacy.

More specifically, we present two general reductions. The first is with respect to an honest user and costs only an *additive* logarithmic factor in communication complexity. The second is a generalization of the first, and deals with *any* user, possibly dishonest. It is done by utilizing a new cryptographic primitive, denoted "conditional disclosure of secrets", which will also be extensively used in later sections. We note that both reductions (theorems 1 and 2) are stated and proved for a single round PIR, but can be generalized to apply to PIR schemes with any number of rounds.

3.1 A General Reduction with Respect to Honest Users

Theorem 1. *Let $\mathcal{P}$ be any 1-round k-database PIR scheme with communication complexity $(\alpha_k(n), \beta_k(n))$. Then, there exists a 1-round $(k+1)$-database honest-user-SPIR scheme $\mathcal{S}_\mathcal{P}$ with communication complexity $(\alpha_k(n)+(k+1)\lceil \log_2 n \rceil, \beta_k(n)+1)$.*

Proof. To simplify notation, assume that the index i is taken from the set $Z_n = \{0, 1, \ldots, n-1\}$ (rather than from $[n]$). In the scheme $\mathcal{S}_\mathcal{P}$, the k databases $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$, of the original scheme $\mathcal{P}$, and an auxiliary database, $\mathcal{DB}_0$, share a random string $r \in \{0,1\}^n$. The scheme $\mathcal{S}_\mathcal{P}$ proceeds as follows:

QUERIES: In addition to its queries as specified by $\mathcal{P}$, the user independently picks a random $s \in Z_n$; it sends the original queries along with s to $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$, and the shifted index $i' \triangleq (i+s) \bmod n$ to $\mathcal{DB}_0$.

ANSWERS: Databases $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$ send answers by simulating $\mathcal{P}$ on the virtual data string $x' \triangleq x \oplus (r << s)$ (where $\oplus$ denotes bitwise exclusive-or and $r << s$ denotes a cyclic shift of the random string r by s places to the left). The auxiliary database $\mathcal{DB}_0$ replies with the single bit $r_{i'}$.

RECONSTRUCTION: The user reconstructs x_i by computing the exclusive-or of the bit $b_\mathcal{P}$ reconstructed from the answers of $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$ according to $\mathcal{P}$, and the bit $r_{i'}$.

By the correctness of $\mathcal{P}$, reconstruction yields $b_\mathcal{P} \oplus r_{i'} = x'_i \oplus r_{i'} = (x_i \oplus r_{i'}) \oplus r_{i'} = x_i$, which proves the correctness of $\mathcal{S}_\mathcal{P}$. The user's privacy follows from the privacy of $\mathcal{P}$, and from the fact that each of the additional queries s and i' is uniformly distributed in Z_n, independently of the $\mathcal{P}$-queries. Finally, to see that an honest user learns only a single physical bit of data, fix any pair of i, s and consider the joint distribution of $(x', r_{i'})$ (where the probability is taken over the choice of r). This distribution is uniform on the set $\{(y,b) : y \in \{0,1\}^n, b \in \{0,1\}, y_i \oplus b = x_i\}$, thus depending only on x_i. Since the answers of $\mathcal{DB}_1, \ldots, \mathcal{DB}_k$ given x' can be simulated by the user, the joint distribution of all answers given any valid choice of queries depends only on x_i. □

Note that in the above scheme $\mathcal{S}_\mathcal{P}$, a dishonest user can send invalid $\mathcal{P}$-queries, or send different shifts s_j to different databases. However, only the latter dishonest behavior could potentially give the user more information about the data. In other words, if the user sends the same shifts to all databases, then data privacy will always be maintained, regardless of the validity of the other queries. Thus, to extend this scheme for a dishonest user, it would suffice to have the databases (each of which sees only a single s_j) send their answers disguised so that the user learns the answers only if the condition $s_1 = \ldots = s_k$ is satisfied. To this end, we use the primitive of "conditional disclosure of secrets", introduced in the following subsection.

theoretic setting (compare with [17]).

Realizing SPIR also involves a modification to the previous multi-database model. This is because information-theoretic SPIR schemes, regardless of their complexity, cannot possibly be achieved in the original PIR setting, in which the databases do not interact with each other at all (see Appendix A). We thus use a minimal extension of the original setting: continue to disallow direct interaction between the databases, but grant them access to a *shared* random string, unknown to the user. A similar kind of extension has been studied before in the contexts of private computation [12, 13], non-interactive zero-knowledge [5] and other scenarios. Here, this extension is particularly natural, since even in the basic PIR setting databases are required to maintain *identical* copies of the same data string. Thus, the data string may be viewed as consisting of two parts: a deterministic one, containing ordinary data, and a random one, containing the shared random string. This becomes more feasible if one is willing to settle for *computational* privacy of the data (while maintaining the information-theoretic privacy of the user); sharing a short random seed allows generating longer shared pseudo-random strings "on the fly", without extra communication.

Our Results

We construct efficient SPIR schemes, with sublinear communication complexity, which may be even further improved if better PIR schemes are designed. More precisely, we present transformations from PIR schemes to SPIR schemes, preserving the user privacy and guaranteeing data privacy as well, with a small penalty in the communication complexity. We give two types of reductions.

A General Reduction We show that using *any* PIR scheme it is possible to construct a SPIR scheme with the same number of rounds, a logarithmic factor in communication complexity, and linear shared randomness. The resultant SPIR scheme requires the use of an additional auxiliary database, which does *not* need to hold the original data (only the shared random string). That is, we achieve:

- $(k+1)$-database SPIR scheme of communication complexity $O(C(n)\log n)$, for any k-database PIR scheme of complexity $C(n)$.

However, the additional database requirement may be costly. In particular, it does not allow to obtain an information theoretic sublinear SPIR solution for the 2-database case. This case is important, since 2 is the minimal number of databases required for such a solution to exist. Indeed, via more specific reductions we manage to avoid the additional database, and in particular obtain a good solution for the 2-database case. Moreover, this next type of reductions is also better in terms of the amount of shared randomness needed.

Specific Reductions We present reductions which exploit specific structural properties of existing PIR schemes to transform them into SPIR schemes which use the same number of databases as the underlying PIR scheme, communication complexity which is at most a multiplicative logarithmic factor over the PIR scheme, and shared randomness of the same order of magnitude as the communication complexity. In particular, extending schemes from [10, 1, 9] we obtain:

- k-database SPIR scheme of complexity $O(\log n \cdot n^{1/(2k-1)})$ for any constant $k \geq 2$.
- $O(\log n)$-database SPIR scheme of complexity $O(\log^2 n \cdot \log\log n)$.
- 2-database computational SPIR scheme of communication complexity $O(n^c)$, for every constant $c > 0$ (assuming the existence of one-way functions).

Our schemes maintain the general paradigm of existing PIR schemes: all databases hold an *identical* copy of x, and all protocols use a *single* queries-answers round. A 1-database computational SPIR of complexity $O(n^c)$, for every constant $c > 0$ (assuming hardness of quadratic residuosity) is also possible, using general zero-knowledge techniques and a multi round protocol (see [17]).

An important observation is that the SPIR problem may be viewed as a distributed version of the cryptographic problem of $\binom{n}{1}$-Oblivious-Transfer (OT) [6, 7].[1] Thus, the results presented in our paper immediately give the first 1-round distributed implementations of $\binom{n}{1}$-OT with information theoretic security and sublinear communication complexity.

Finally, in all the schemes presented in this paper, the multiplicative logarithmic factor in communication complexity can be eliminated for the case of an *honest but curious* user (i.e., a user which follows the prescribed scheme, but may try to deduce extra information from the communication). All our schemes are first presented for this case (honest user), and then generalized to handle any user.

Previous Work

Private information retrieval (with information theoretic user privacy) was introduced in [10], where the schemes achieve communication complexity of $O(n^{1/3})$ bits with 2 databases; $O(n^{1/k})$ bits with $k \geq 3$ databases; and $O(\log^2 n \log\log n)$ bits with $k = O(\log n)$ databases. In [1] the k-database upper bound is improved to $O(n^{1/(2k-1)})$. The *computational* counterpart of PIR (i.e., schemes where the user privacy is only with respect to polynomial-time databases, relying on certain intractability assumptions) were considered in [9, 17]; they show how to obtain schemes with communication complexity $O(n^c)$ (for any constant $c > 0$) for $k=2$ databases [9], assuming the existence of one-way functions (see also [18]) and even for the case of a single database [17] (under the quadratic residuosity assumption). None of these works consider data privacy, with the exception of [17], who show how to add data privacy to their computational PIR scheme using general zero-knowledge techniques and a multi-round protocol.

Organization: In Section 2 we introduce notations and basic definitions. In Section 3 we show a general transformation of PIR schemes into SPIR schemes, including the introduction of "conditional disclosure of secrets" in Subsection 3.2. The following sections present specific schemes, which outperform the ones obtained by applying the general transformation. Section 4 includes SPIR schemes which rely on the user being honest. In Section 5 we present schemes which keep the data private from *any*, possibly dishonest, user (with a minor extra communication cost). Section 6 describes some possible generalizations of our results. Appendix A contains a proof of the impossibility of SPIR in the usual PIR setting (without interaction between the databases or shared randomness). Appendix B contains proofs which were deferred from the main body of text. Finally, Appendix C addresses the problem of SPIR in the computational setting.

[1] An $\binom{n}{1}$-OT protocol (also denoted "all or nothing disclosure of secrets") allows Bob to secretly choose one of n secret bits held by Alice, in a way that at the end of the protocol Bob learns only a single bit of his choice, and Alice learns nothing about Bob's choice.

Protecting Data Privacy in Private Information Retrieval Schemes

Yael Gertner* Yuval Ishai † Eyal Kushilevitz‡ Tal Malkin§

Abstract

Private Information Retrieval (PIR) schemes allow a user to retrieve the i-th bit of a data string x, replicated in $k \geq 2$ databases (in the information-theoretic setting) or $k \geq 1$ databases (in the computational setting), while keeping the value of i private. The main cost measure for such a scheme is its communication complexity.

In this paper we introduce a model of *Symmetrically-Private Information Retrieval (SPIR)*, where the privacy of the *data*, as well as of the user, is guaranteed. That is, in every invocation of a SPIR protocol, the user learns only a single (physical) bit of x and no other information about the data. Currently known PIR schemes fail to meet this goal. We show how to transform PIR schemes into SPIR schemes (with information theoretic privacy), paying at most a logarithmic factor in communication complexity. To this end, we introduce and utilize a new cryptographic primitive, denoted *conditional disclosure of secrets*, which we believe may be a useful building block for the design of more general cryptographic protocols. In particular, we get a k-database SPIR scheme of complexity $O(\log n \cdot n^{1/(2k-1)})$ for every constant $k \geq 2$, and an $O(\log n)$-database SPIR scheme of complexity $O(\log^2 n \cdot \log\log n)$. All our schemes require only a single round of interaction, and are resilient to any dishonest behavior of the user.

Our results also yield the first 1-round distributed implementations of $\binom{n}{1}$-OT (oblivious transfer) with information-theoretic security and sublinear communication complexity.

1 Introduction

Private Information Retrieval (PIR) schemes allow a user to retrieve information from a database while maintaining his query private. In this model, the database is viewed as an n-bit string x out of which the user retrieves the i-th bit x_i, while giving the database no information about i. The main cost measure for such schemes is their communication complexity. The notion of PIR was introduced in [10], where it was shown that if there is only one copy of the database available then $\Omega(n)$ bits of communication are needed (for information-theoretic user privacy). However, if there are $k \geq 2$ non-communicating copies of the database, then there are solutions with much better (sublinear) communication complexity.

In this paper, we introduce the stronger model of *Symmetrically Private Information Retrieval (SPIR)*, where privacy of the *database*, as well as of the user, is guaranteed. That is, every invocation of SPIR, in addition to maintaining the user's privacy, prevents the user (even a dishonest one) from obtaining any information other than a single *physical* bit of the data. Data privacy is a natural and crucial requirement in many settings. For example, consider a commercial database which sells information, such as stock information, to users, charging by the amount of data that the user retrieved. Here, both user privacy and database privacy are essential.

The original PIR model was only concerned with user privacy, without requiring any protection of data privacy. Indeed, previous PIR schemes allow the user to obtain other physical bits of the database (i.e., x_j for $j \neq i$) or other information such as the exclusive-or of certain subsets of the bits of x. A good example of this is a single invocation of the best 2-database information theoretic scheme currently known [10] from which a user can systematically retrieve $\Theta(n^{1/3})$ *physical* bits of data (see Section 5, Example 2).

To efficiently realize SPIR schemes, we introduce and utilize a new cryptographic primitive, denoted *"conditional disclosure of secrets"*, which may also be of independent interest as a building block for designing more general cryptographic protocols.

This work is mostly concerned with the information theoretic setting, which requires multiple databases in order to obtain sublinear communication complexity. However, our techniques can be applied to computational schemes as well (as discussed in Appendix C). The information theoretic setting is more challenging in the sense that efficiency is measured in a very precise way, and does not depend on an implicit polynomially-flexible security parameter. Additionally, standard cryptographic tools used for solving similar problems in more general protocols cannot be used here: general compilers of unconditionally secure multi-party computation (cf. [3, 8]) are too expensive in terms of communication and rounds complexity, and general zero-knowledge proof methodology cannot be used at all in the information

*Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA. e-mail: ygertner@saul.cis.upenn.edu. Supported by DARPA grant DABT63-96-C-0018.

†Department of Computer Science, Technion, Haifa 32000, Israel. e-mail: yuvali@cs.technion.ac.il

‡Department of Computer Science, Technion, Haifa 32000, Israel. e-mail: eyalk@cs.technion.ac.il URL: http://www.cs.technion.ac.il/~eyalk

§Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139. e-mail: tal@theory.lcs.mit.edu. Supported by DARPA grant DABT63-96-C-0018.