Secure Games with Polynomial Expressions

Aggelos Kiayias[1] and Moti Yung[2]

[1] Graduate Center, CUNY, NY USA,
akiayias@gc.cuny.edu
[2] CertCo, NY USA
moti@cs.columbia.edu

Abstract. We present the first private information retrieval (PIR) scheme which is both, deterministically correct and has poly-logarithmic communication complexity. Our PIR protocol is symmetrically secure, and improves by a few orders of magnitude the known probabilistically correct poly-logarithmic scheme. This result is achieved as an application of our methodology which introduces a broad family of games, called Secure Games with Polynomial Expressions (SGPEs), that involve two interacting players: Alice and Bob. The objective of these games is the secure "interactive computation" of the value of a polynomial expression which is made up of polynomials and field elements that both players distributedly contribute to the game. The players wish to keep some or all the data (field elements and polynomials) they contribute to the game, secret and independently secure. We show that any SGPE can be played much more *efficiently* than by using generic methods, and so that no party reveals more than what it intends to. Besides the above mentioned PIR application, we present additional applications such as the "lists' intersection predicate" which is useful for secure conduct of e-commerce procedures, such as negotiation methods known as "settlement escrows" in the legal/ economics/ business literature.

1 Introduction

One of the most important results on the foundations of cryptography (suggested by Yao [Yao86], generalized to multi-party by Goldreich, Micali and Wigderson [GMW87], and characterized based on the Oblivious Transfer primitive by Kilian [Kil90]) is that given any polynomially computable function $f(x, y)$, it is possible for two parties, Alice (A for short) and Bob (B for short), to jointly compute $f(\alpha, \beta)$, with A contributing α and B contributing β, in such a way so that no party learns anything more than what can be deduced by the final output. The resulting protocols are relative to the size of the circuit that computes f that, even for simple functions, are considerably expensive to implement. Consequently, nowadays where distributed applications over the Internet are about to become a reality, it is worthwhile to seek special cases of useful function families that can accept more efficient protocol techniques (as advocated in [Gol97]).

In that spirit, Naor and Pinkas [NP99] introduced an efficient protocol for obliviously computing the value of a polynomial (Oblivious Polynomial Evalua-

F. Orejas, P.G. Spirakis, and J. van Leeuwen (Eds.): ICALP 2001, LNCS 2076, pp. 939–950, 2001.

tion, OPE). In their setting, B possesses a polynomial P, A has a value α and wishes to obliviously compute $P(\alpha)$.

In this paper, we further investigate possibilities for efficient solutions of new useful problems in the general area of secure function evaluation by introducing a family of protocols called *Secure Games with Polynomial Expressions* (SGPEs). The general idea of our approach is to consider the joint computation of a polynomial expression that is made up of secret polynomials owned by the two players (as well as non-secret components). Player A selects an input for the expression, and wishes to obtain the value of the expression on this input. Depending on the contribution of A to the expression we can categorize SGPEs to those that A contributes only field elements to the expression (type 1), and to those that A contributes also polynomials (type 2). An example of a type-1 SGPE is the Secure Multivariate Polynomial Evaluation (SMPE): B holds a secret multivariate polynomial P, and A wishes to obtain a point in the graph of P of her choice. A Secure Nested Polynomial Game (SNPG for short) is an example of SGPE of type-2: A holds a constant number of c secret polynomials $Q_2, \ldots, Q_c$ and wants to compute $P_c(Q_c(\ldots(P_2(Q_2(P_1(\alpha))))\ldots))$ for an α of her choice, where the polynomials $P_1, P_2, \ldots, P_c$ are contributed by B.

The security conditions that we consider, are the following: A does not want to reveal anything about the data she contributes to the game, and B does not want to disclose his data beyond what is trivially inferred from A's output. In addition to the above (traditional) conditions, both players wish that if the secrecy of some of their private data is compromised or the search space of some part of the data is small, this has no effect on the secrecy of the remaining inputs (this property can be called *secret independence*). More generally players wish that their data are secure even if they are not uniformly distributed over all possible inputs.

We present an efficient construction for SGPEs of type-1 and an efficient transformation of a type-2 game to a type-1 game. We get a protocol of two flows of communication, one of which is employing an implementation of a single t-out-of-n Oblivious Transfer over values of the proper field, where t and n are small polynomial functions (in the size of the polynomial expression used in the game). Our security assumption is coding theoretic and is related to the Polynomial Reconstruction Problem. In fact, one of the basic contributions of Naor and Pinkas [NP99] is the consideration of this problem as a hard problem to base protocol security on.

Using SMPE, we provide a new Private Information Retrieval scheme (PIR) with polylogarithmic communication complexity. Our scheme is the first direct polylogarithmic Symmetric PIR that is deterministically correct and is at least five orders of magnitude better, in the polylogarithmic sense, compared to the previous polylogarithmic PIR of [CMS99]. Our PIR protocol assures correct execution always in contrast with the [CMS99]-scheme, which is a probabilistically correct protocol that exhibits a trade-off between error probability and communication complexity.

Using our construction for type-1 games we can solve a variety of other problems such as the "Lists' Intersection Predicate." In this problem, two agencies A and B, have two lists S_A, S_B respectively, and they want to check whether $S_A \cap S_B \neq \emptyset$. If this is the case, no agency wants to reveal any witness for this fact. This procedure enables negotiating parties to know that there is a common issue to be discussed without revealing mutual interests up front. This can be applied to solving (without any trusted party) the problem known as "settlement escrows." This procedure was originally proposed for pretrial negotiations (employing a trusted third party) in out of court legal settlements [GM95]. It allows two negotiating parties to figure out if their price ranges intersect and nothing more, in order to further continue with negotiating a deal. It can be applied to distributed decision-making in general e-commerce and business procedures (see [BN96]).Additional applications of our setting such as "Oblivious Negotiations" or "Oblivious Bargaining" will appear in the full version due to lack of space.

We note that trying to reduce our setting to OPE encounters a number of problems, mainly with respect to security, as the reduction fails to enforce secret-independence, a property that is necessary for the new applications. Ultimately secret-independence appears to require a stronger intractability assumption compared to the one needed for the security of OPE, which we formulate in this work. The OPE protocol has a two-flow structure for two layer computation (polynomial over data). We note that it is not at all obvious how to retain this protocol structure for our multi-layer setting, but, interestingly, we show it to be possible.

2 Preliminaries and Definitions

Let $\mathcal{P} := \{P_1, P_2, \ldots\}$ be a set of predicates, and $\mathcal{X} := \{x_1, x_2, \ldots\}$ a set of variables. An expression $\mathcal{E}$ is a rooted-DAG (direct acyclic graph) with all arcs directed towards the root specified as follows: each node is one of the following: P_i, $+$, $\cdot$, or a natural number. If a node is $+$ or $\cdot$ then it has two children, if a node is a number it has a single child; if a node is P_i then it has any non-zero number of children; each arc entering P_i is labeled by a non-zero natural number; The leaves of the DAG are selected from $\mathcal{X}$. The *value* of a path from a leaf to the root, is the product of all labels and number nodes that are in its course (and is set to 1, if there are no labels or number nodes). The *degree* of a variable is defined as the maximum path value taken over all paths from the variable node to the root. Let $\mathcal{E}$ be an expression, and let $P_1, \ldots, P_v$ denote its predicate nodes; if we map each predicate P_i to a polynomial with the same number of variables as the children of P_i and of the same degrees as the labels of its incoming arcs, an in-order traversal of $\mathcal{E}$ can be seen as a polynomial (interpreting each number node as exponentiation); we denote this polynomial by $\mathcal{E}(P_1, \ldots, P_v)$, and say that the polynomials $P_1, \ldots, P_v$ "fit into" $\mathcal{E}$.

If P is a predicate node we denote by $\text{label}(P, j)$ the label of the j-th incoming arc. Let $|\mathcal{E}|$ denote the size of the DAG (number of arcs). We define $\text{size}(\mathcal{E}) := |\mathcal{E}| + \sum_P \prod_j (\text{label}(P, j) + 1)$ where the sum is over all predicate nodes of $\mathcal{E}$. In order to store $\mathcal{E}(P_1, \ldots, P_v)$ we need $\text{size}(\mathcal{E})$ space. One of the reasons for introducing expressions instead of talking simply about polynomials is space: if $\mathrm{coef}(P)$ denotes the number of coefficients of a polynomial P, then it holds that $\mathrm{coef}(\mathcal{E}(P_1,\ldots,P_v))$ can be exponentially large compared to $\mathrm{size}(\mathcal{E})$. In order to compute a value of $\mathcal{E}(P_1,\ldots,P_v)$ using the expression representation we need $\mathcal{O}(\mathrm{size}(\mathcal{E}))$ field operations. If $\mathcal{E}$ is an expression, denote by $d_1,\ldots,d_r$ the degrees of its variables. For a fixed constant c, we say that an expression is c-bound if $\mathrm{lcm}(d_1,\ldots,d_r) = \mathcal{O}([\mathrm{size}(\mathcal{E})]^c)$.

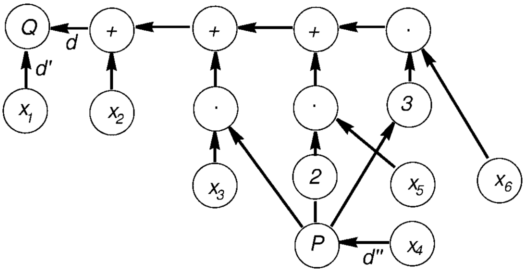

Fig. 1. Example of an expression that defines the polynomial $Q(x_1, x_2 + x_3P(x_4) + x5(P(x_4))^2 + x_6(P(x_4))^3)$, with $\mathrm{degree}(x_2) = \mathrm{degree}(x_3) = \mathrm{degree}(x_5) = \mathrm{degree}(x_6) = d$, $\mathrm{degree}(x_1) = d'$, $\mathrm{degree}(x_4) = 3dd''$.

For the following, fix a c-bound expression $\mathcal{E}$ with v predicates and r variables. Note that we restrict the applicability of our protocol to c-bound expressions. Although we do not rule out the existence of a construction for unbounded expressions, c-bound expressions are sufficient for all applications discussed here.

A type-1 SGPE is as follows: player B has v secret polynomials $P_1,\ldots,P_v$, player A has r secret values $\alpha_1,\ldots,\alpha_r \in \mathbb{F}$ and wants to obtain $\mathcal{E}(P_1,\ldots,P_v)(\alpha_1,\ldots,\alpha_r)$. Some of the polynomials of player B may be publicly known. If $v = 1$ and $\mathcal{E}(P) := P$, then the game is called "Secure Multivariate Polynomial Evaluation" (SMPE). A type-2 SGPE is defined similarly with the only difference that some of the $P_1,\ldots,P_v$ polynomials are contributed by A. When $\mathcal{E}$ has the form $P_c(Q_c(\ldots(P_2(Q_2(P_1(x))))\ldots))$ with the P_i contributed by B, and the Q_i contributed by A then we will call this game a "Secure Nested Polynomial Game" (SNPG). Our game schema involves two flows of information, from A to B and from B to A (this latter flow employs a t-out-of-n OT). Correctness and security requirements for both types of games are as follows:

Definition 1. *Let $\mathcal{E}$ be a c-bound expression with v predicates $P_1,\ldots,P_v$, and r variables. Let $\mathcal{H}_A,\mathcal{H}_B$ denote the sequence of secrets contributed by the two players to the expression. There are two probabilistic polynomial time (PPT) algorithms $\mathcal{A},\mathcal{B}$ and a deterministic algorithm $\mathcal{C}$ (parts of our protocol) so that: $\mathcal{C}(\mathcal{B}(\mathcal{A}(\mathcal{H}_A),\mathcal{H}_B)) = \mathcal{E}(P_1,\ldots,P_v)(\alpha_1,\ldots,\alpha_r)$ (independently of the coin tosses of $\mathcal{A},\mathcal{B}$). Informally, $\mathcal{A}$ is used by A to hide her secrets and give them to B; B uses $\mathcal{B}$ to hide his secrets and apply them over the secrets of A; $\mathcal{C}$ is used by A* *to reconstruct the output of the protocol from the reply of B (which is obtained through a t-out-of-n OT). The computation cost is polynomial in* $\mathrm{size}(\mathcal{E})$*.*

Security of A. *Informally, the security of A is established by showing that B cannot deduce anything meaningful out of the protocol transcript he receives. More formally, for all PPT* $\mathcal{B}'$ *playing B's part and all probability distributions* $\mathcal{D}_A$ *for* $\mathcal{H}_A$ *there is a PPT* $\mathcal{B}''$ *such that the following is negligible:*

$$| \,\mathbf{Prob}[Z = \mathcal{H}_A : Z \leftarrow \mathcal{B}'(\mathcal{A}(\mathcal{H}_A))] - \mathbf{Prob}[Z = \mathcal{H}_A : Z \leftarrow \mathcal{B}''] \,|$$

Security of B. *Informally, security of B can be claimed by comparing with the ideal implementation. Let* $\mathcal{I}(\mathcal{H}_A, \mathcal{H}_B)$ *denote the output of player A in the ideal implementation of the protocol. Also, let* $\mathcal{T}(\mathcal{H}_A, \mathcal{H}_B)$ *be the protocol transcript obtained by player A at the end of the protocol. We show that for any PPT* $\mathcal{A}'$ *and any* $\mathcal{H}_A$ *there is a PPT* $\mathcal{A}''$ *s.t.*

$$| \,\mathbf{Prob}[\mathcal{A}'(\mathcal{T}(\mathcal{H}_A, \mathcal{H}_B)) = 1] - \mathbf{Prob}[\mathcal{A}''(\mathcal{I}(\mathcal{H}_A, \mathcal{H}_B)) = 1] \,|$$

is negligible (the probability is taken over the internal coin tosses of $\mathcal{A}', \mathcal{A}''$ *and* $\mathcal{H}_B$ *is distributed according to some probability distribution* $\mathcal{D}_B$*).*

The security of the party A for the OPE protocol of [NP99] was based on the following problem, which is also related to the security of A in our construction:

Definition 2. Polynomial Reconstruction (PR). *Given n, k, t and the pairs* $\{\langle z_i, y_i\rangle\}_{i=1}^{n}$ *in* $\mathbb{F}^2$*, output all* $\langle p, I\rangle$ *such that* $p \in \mathbb{F}[x]$*,* $\mathrm{degree}(p) < k$*,* $I \subseteq \{1, \ldots, n\}$*,* $|I| = t$ *and* $\forall i \in I (p(z_i) = y_i)$*.*

PR is of prime interest in Coding Theory, since it corresponds to the decoding problem of Reed-Solomon codes. Translated in this context PR asks for all messages that agree with at least t positions of the received codeword. For a general treatment on the subject the interested reader is referred to [Ber68] or [MS77]. From the perspective of Reed-Solomon codes, we will further specialize definition 2 to require: (i) $k < n$ since k/n is the message rate of the code, and (ii) at least one solution $\langle p, I\rangle$ exists, since before the addition of the noise all pairs belong in the graph of some polynomial.

When $t \geq \frac{n+k}{2}$ then PR has only one solution and it can be found with the algorithm of Berlekamp and Welch [BW86] ($\frac{n+k}{2}$ is the error-correction bound of Reed-Solomon codes). The problem has been investigated further for smaller values of t ([Sud97,GS98,GSR95]) and it is believed that PR is hard when $t < \sqrt{kn}$ (the best algorithm known, by Guruswami and Sudan [GS98], finds all solutions when $t \geq \sqrt{kn}$). From a cryptographic perspective we are more interested in the hardness of PR on the average. It is easy to see that PR on the average (termed also noisy PR) has only one solution $\langle p, I\rangle$ such that $|I| = t$ with very high probability. It is believed that the noisy PR is not easier than the PR (see [NP99]). This is because given an instance of the PR it is possible to randomize the polynomial (but it is not known how to randomize the noise). PR was further investigated in [BN00].

Our proof of security (for player A) is based on the following problem, which we call Multisample Polynomial Reconstruction (MPR):

Definition 3. MPR. *Given n, k, t, r, and the distinct tuples $\{\langle z_i, y_{i,1}, \ldots, y_{i,r}\rangle\}_{i=1}^{n}$ so that each $\{\langle z_i, y_{i,\ell}\rangle\}_{i=1}^{n}$ is a noisy PR instance with parameters n, k, t and solution $\langle p_\ell, I\rangle$, find $\langle p_1, \ldots, p_r, I\rangle$.*

MPR appears to be not much easier than PR, a fact that is justified similarly to the case of PR vs. noisy PR: given an instance of PR it is possible to randomize the polynomial r times and get a version of MPR, but as before it is not apparent how to replicate and randomize the noise. We will formulate this as a complexity assumption:

Complexity Assumption. For any r there are n, k, t polynomially related parameters so that any probabilistic algorithm solving the MPR has negligible success probability in n.

Solving MPR either involves using techniques against a specific noisy PR instance that is included in the MPR instance (since the recovery of some $\langle p_\ell, I\rangle$ immediately implies the recovery of $\langle p_1, \ldots, p_r, I\rangle$) or in a more direct fashion trying to take advantage of the relation between the noisy PR instances included in the MPR instance. The best algorithm for solving PR is [GS98], which succeeds when $t \geq \sqrt{kn}$. Solving MPR directly has been discussed recently in [KY01] and succeeds for choices of $r > \frac{n}{t}$. As a result the current state of the art suggests that MPR is hard when $ct < \sqrt{kn}$ and $c'r < \frac{n}{t}$ for some $c, c' > 1$.

To complete this section, let us comment briefly on the relation of SGPEs and OPE. In particular if it is possible to simulate a SGPE using OPE; there are two possibilities: (1) if only univariate polynomials appear in the expression the two players can use many individual OPEs to obtain intermediate results and finally player A will compose the final output. Nevertheless to conform to the security requirements randomization of the partial results is necessary something that appears to be hard unless the expression degenerates to an affine transformation. (2) The case of multivariate polynomial evaluation e.g. $P(\alpha, \beta)$, it can be performed by OPE as follows: A sends to B random $s(x), s'(x) \in \mathbb{F}[x]$ s.t. $s(x_0) = \alpha$ and $s'(x_0) = \beta$ (x_0 is kept secret by A); A and B engage in OPE so that A obtains $P(s(x_0), s'(x_0))$. This approach has the deficiency that the values contributed by A are not “independently secure”, i.e. partial knowledge of some of the values (or a small search-space for one of the values) can lead to the recovery of all secret input of A with non-negligible probability.

3 SPGEs of Type 1

In the following construction, a t-out-of n OT protocol is used as a primitive.

- *Protocol parameter*: a c-bound expression $\mathcal{E}$ of v predicates.
- *Input of B*: Polynomials $P_1, \ldots, P_v$ that fit into $\mathcal{E}$.
- *Input of A*: r elements of $\mathbb{F}$, $\alpha_1, \ldots, \alpha_r$.
- *Output of A*: $\mathcal{E}(P_1, \ldots, P_v)(\alpha_1, \ldots, \alpha_r)$.
- *Security parameters*: n, l.
- Let $P(x_1, \ldots, x_r) := \mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$, and denote by d_ℓ the degree of x_ℓ in P. Set $d := lr\mathrm{lcm}(d_1, \ldots, d_r)$, and $k := \min_\ell \frac{d}{rd_\ell} + 1$.

Step 1. A generates r instances of the noisy PR, $\{\langle z_i, y_{i,\ell}\rangle\}_{i=1}^n$ with solution $\langle p_\ell, I\rangle$, such that $p_\ell(0) := \alpha_\ell$, degree$(p_\ell) = k-1$, $z_i \neq 0$ and $z_i \neq z_j$ for all i, j, $j \neq i$. Then, A, forms the $(r+1)$-tuples $\{\langle z_i, y_{i,1}, \ldots, y_{i,r}\rangle\}_{i=1}^n$, and she sends them to B.
Step 2. B hides P in a random polynomial Q: Let $C, C' \in \mathbb{F}[x]$ be random polynomials of degree d such that $C(x) = C'(x) = 0$. Define a polynomial $Q \in \mathbb{F}[x_0, x_1, \ldots, x_r]$ as follows: $Q(x_0, \ldots, x_r) = P(x_1, \ldots, x_r) + C(x_0) + x_1^{d_1} \ldots x_r^{d_r} C'(x_0)$. The storage space needed for Q is size$(\mathcal{E})$ + $2d$. Computing a value of Q requires $\mathcal{O}(\text{size}(\mathcal{E}) + d)$ field operations. For each tuple $(z_i, y_{i,1}, \ldots, y_{i,r})$ B computes the value $Q(z_i, y_{i,1}, \ldots, y_{i,r})$. Note that the polynomial $R(x) := Q(x, p_1(x), \ldots, p_r(x))$ on 0 gives $R(0) = P(\alpha_1, \ldots, \alpha_r)$. The degree of R is $d_R = d + d_1 d_{p_1} + \ldots + d_r d_{p_r} \leq 2d$. Therefore, if A learns $t := 2d+1$ values of R, she can interpolate it and compute $R(0)$.
Step 3. A and B engage in a t-out-of n OT in which A chooses to learn the values $Q(z_i, p_1(z_i), \ldots, p_r(z_i))$. Now A knows $2d+1$ values of the polynomial R and can interpolate it to compute $R(0) = P(\alpha_1, \ldots, \alpha_r)$.
Implementation and Complexity. Clearly, A can compute $P(\alpha_1, \ldots, \alpha_r)$ for any $\alpha_1, \ldots, \alpha_r$ of her choice. The time-complexity of the protocol is $\mathcal{O}(rn + d\log^2 d + f_A(t,n))$ for player A and $\mathcal{O}(nd + n\text{size}(\mathcal{E}) + f_B(t,n))$ for player B, where $f_A(t,n), f_B(t,n)$ denotes the running time of the t-out-of n OT protocol for each player respectively. The communication complexity is $\mathcal{O}(rn + c(t,n))$ where $c(t,n)$ is the communication complexity of the t-out-of n OT. Regarding the security parameters, in section 3.1 we show that $n = \mathcal{O}(rd + d^2/l)$ is sufficient; l relates to the value k and is chosen so that k is large enough so that player B is not be able to find $p_1, \ldots, p_r$ by brute-force in $\min\{\binom{n}{t}, \binom{n}{k}\}$ steps. We point here that if the expression $\mathcal{E}$ is 0-bound, then the complexity of player A does not depend on the size of the expression. For a t-out-of-n OT protocol the reader is referred to e.g. [NP99] where t-out-of n OT is efficiently and unconditionally reduced to 1-out-of-2 OT.

3.1 Security of A

The security of A is based on the hardness of MPR as the following theorem reveals:

Theorem 1. *If* B *breaks the security of* A *in our protocol then, assuming that the underlying t-out-of-n OT is secure,* MPR *is polynomial time for parameters* $n, k-1 := \min_\ell \frac{d}{rd_\ell}, t := 2d+1, r$.

By a suitably large choice of the security parameter n we can enforce the security of A under the MPR-complexity-assumption (provided that the security parameter l is large enough to withstand a brute-force attack – see previous section). Both $ct < \sqrt{kn}$ and $c'r < \frac{n}{t}$ should be satisfied (the parameters c, c' allow for small improvements to the results against MPR). It is easy to see that it suffices to select $n = \mathcal{O}(rd + d^2/l)$.

3.2 Security of B

The security of player B is established by showing that the output of player A out of a protocol execution (the protocol transcript obtained by A) is essentially identical to what she gets in an ideal implementation. This holds true independently of A's behavior. In an ideal implementation, A gives to a trusted third party C all information send to B in step 1 of the protocol together with the randomness she used — note that this reveals her secret values $\alpha_1, \ldots, \alpha_r$. Player B gives to C its secret input $P_1, \ldots, P_v$. In turn, C returns to A, either a value of $\mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$ or a linear combination of some values of $\mathcal{E}(P_1, \ldots, P_v)(x_1, \ldots, x_r)$ (the exact formulation is given in the full version).

Lemma 1. *There is a PPT $\mathcal{G}$ that given the output of the ideal implementation of the protocol for player A, and all information available to player A, generates a protocol transcript that is statistically indistinguishable from legitimate protocol transcripts generated during normal operation, under the assumption that t-out-of-n OT can be implemented ideally.*

Theorem 2. *Our construction is secure with respect to player B under the assumption that the underlying t-out-of-n OT is secure.*

4 SGPEs of Type 2

In this section we present a transformation of type-2 games to type-1 games. First we deal with SNPGs: we will consider only the two round case and it will become clear how to generalize to any constant number of rounds. Suppose B possesses the secret polynomials $P_2, P_1 \in \mathbb{F}[x]$ and A the secret polynomial $Q_2 \in \mathbb{F}[x]$ of degree δ (known to B). A wants to compute $P_2(Q_2(P_1(\alpha)))$ for an α of her choice. B defines the expression $\mathcal{E}(P_1, P_2)(x_0, \ldots, x_\delta, x) = P_2(x_0 + x_1 P_1(x) + \ldots x_\delta (P_1(x))^\delta)$.

If $Q_2(x) = a_0 + a_1 x + \ldots a_\delta x^\delta$ then A, using the type-1 protocol, can compute the value $\mathcal{E}(P_2, P_1)(a_0, \ldots, a_\delta, \alpha)$ for an α of her choice. Now by the definition: $\mathcal{E}(P_2, P_1)(a_0, \ldots, a_\delta, \alpha) = P_2(a_0 + a_1 P_1(\alpha) + \ldots a_\delta (P_1(\alpha))^\delta) = P_2(Q_2(P_1(\alpha)))$.

The case of any type-2 game can be sketched as follows: player A should obtain $\mathcal{E}(P_1, \ldots, P_v, Q_1, \ldots, Q_{v'})(\alpha_1, \ldots, \alpha_r)$ where the polynomials $P_1, \ldots, P_v$ are contributed by B, and the values $\alpha_1, \ldots, \alpha_r$, and polynomials $Q_1, \ldots, Q_{v'}$ are contributed by A. For simplicity we assume that the polynomials Q_i are univariate. Let the degree of Q_i be δ_i. B substitutes in the expression $\mathcal{E}$ each occurrence of $Q_i(V)$ with $x_0 + x_1 V + \ldots x_{\delta_i} V^{\delta_i}$ for all $i = 1, \ldots, v'$. The resulting expression is $\mathcal{E}'$. Note that $\mathcal{E}'$ is independent of the sequence of the substitutions (each substitution works on a disjoint portion of the DAG). It is not hard to show that $|\mathcal{E}'| = \mathcal{O}(\text{size}(\mathcal{E}))$, and consequently $\text{size}(\mathcal{E}') = \mathcal{O}(\text{size}(\mathcal{E}))$. Note also that if $\mathcal{E}$ is c-bound then $\mathcal{E}'$ is also c-bound. By engaging in type-1 game with $\mathcal{E}'$, player A can "plug-in" all the coefficients of her polynomials, along with the values $\alpha_1, \ldots, \alpha_r$, and therefore the type-2 game transforms to a type-1 game.

Theorem 3. *The correctness and security of our construction for type-1 games, implies the correctness and security of the type-2 protocol described above.*

We note that in general, SNPGs are not produced by c-bound expressions. An expression for an SNPG is c-bound only if the number of polynomials contributed by both players is constant (constant nesting).

5 Private Information Retrieval

In Private Information Retrieval (PIR for short), the database prober, wants to obtain a bit or an object of her choice from a database of size N, without revealing her choice to the database moderator. The problem was introduced in [CGKS95]. A PIR can be seen as a 1-out-of-N OT with the additional restriction that we are interested in achieving prober time complexity which is sublinear in N, and more specifically sublinear communication complexity. Note that in a PIR scheme the security of the database is not enforced; something that happens in a Symmetric PIR – SPIR for short, [GIKM98]. Communication complexity of $\mathcal{O}(N^{1/k})$ in [CGKS95] (replication of the databases), and later $\mathcal{O}(N^c)$ in [CG97] (computational setting – cPIR) was shown. In [KO97] replication was dropped as a requirement (for the computational setting), and in [CMS99] the first cPIR with polylogarithmic communication complexity was presented. For any cPIR, it seems inevitable that the communication complexity is polynomial in some security parameter l. A polylogarithmic PIR has communication complexity of $\mathcal{O}(\text{polylog}(N))$, therefore it is meaningful to require $l = \mathcal{O}(\text{polylog}(N))$ also. In [CMS99], communication is polynomial in the security parameter l and the moderator can break the security of the prober by an $\mathcal{O}(2^{cl})$ computation (provided that the underlying security assumption is true); therefore by choosing $l = \Omega(\delta^2)$, where $\delta = \log N$, breaking the security of the prober becomes super-polynomial in N. Here, we present the first direct (computational) SPIR protocol that has polylogarithmic communication complexity. We achieve substantial improvements compared to the result of [CMS99]:

1. The correctness of our PIR protocol is deterministic, rather than probabilistic as in [CMS99]. Note that in the [CMS99]-PIR reducing the error probability results in asymptotic increase of the communication complexity.

2. The communication complexity of our SPIR protocol is $\mathcal{O}(hl^2\delta^3)$, where h denotes the number of bits that are required to store a single object of the database. The choice $l = \Omega(\delta)$ is sufficient in order to ensure that the moderator needs to spend super-polynomial time in N for the search. If we set $l := \delta$ the communication complexity of our scheme is $\mathcal{O}(h\delta^5)$. The communication of the [CMS99]-PIR is $\mathcal{O}(hl^f)$ where $f \geq 5$ and depends on the underlying Φ-Hiding Assumption (its second constant). If $f = 5$, and $l = \delta^2$ then the communication of the PIR is $\mathcal{O}(h\delta^{10})$. Increasing l to achieve stronger security for the prober, yields larger asymptotic speed-up for our PIR-scheme.

The time-complexity of the two parties is low: $\mathcal{O}(l^2\delta^3)$ for the prober, and $\mathcal{O}(Nl\delta^2)$ for the moderator; our construction requires an $\mathcal{O}(N^2)$ pre-processing stage by the database moderator that needs to be performed only once, prior to servicing any number of requests.

Note that we deal directly with a database containing words rather than bits. Let Δ be a database consisting of $N := 2^{\delta}$ words, $\Delta := \{w_0, \ldots, w_{N-1}\}$. Define a polynomial $P(x_1, \ldots, x_\delta) := \sum_{j_1,\ldots,j_\delta} a_{j_1 j_2 \ldots j_\delta} x_1^{j_1} x_2^{j_2} \ldots x_\delta^{j_\delta}$, where each $j_\ell \in \{0, 1\}$, $\ell = 1, \ldots, \delta$. Let $v(j_1, \ldots, j_\delta) := j_\delta + 2j_{\delta-1} + \ldots + 2^{\delta-1} j_1$. We write $\langle j_1, \ldots, j_\delta \rangle \prec \langle i_1, \ldots, i_\delta \rangle$ to denote the coordinate-wise ordering of bit-strings. The coefficient $a_{j_1 j_2 \ldots j_\delta}$ of P is defined recursively as follows: $a_{j_1 j_2 \ldots j_\delta} := w_{v(j_1,j_2,\ldots,j_\delta)} - \sum_{\langle j'_1, j'_2, \ldots, j'_\delta \rangle \prec \langle j_1, j_2, \ldots, j_\delta \rangle} a_{j'_1 j'_2 \ldots j'_\delta}$ (note that $a_{00\ldots0} := w_0$).

In our PIR protocol, A plays the role of the prober and player B is the moderator of the database. B prepares P during a pre-processing stage. A wants to obtain the word w_q of the database. Let $\langle j_1, \ldots, j_\delta \rangle$ be the binary representation of q. By using the type-1 protocol for SMPE, A obtains the value $P(j_1, \ldots, j_\delta)$ which is equal to w_q.

Theorem 4. *The scheme above is a deterministically correct SPIR scheme with polylogarithmic communication complexity.*

We point out that the multivariate polynomial setting is suitable for PIR, since only in such a polynomial it is possible to directly store "exponentially" many words while at the same time keeping the degree logarithmically small w.r.t. the number of coefficients. This is what allows the complexity of the prober to be sublinear in the database size, as the prober has to spend polynomial time in the degree of the polynomial.

6 Lists' Intersection Predicate

The List Intersection Problem was introduced and solved in [NP99]: two agencies holding two lists, jointly compute their intersection, without revealing any elements not common to both lists. Here we consider a different setting for this problem where even the common part needs to remain secret. More specifically, the two agencies have a number of lists and want to *check* whether there exist any common items in these lists; if this is the case no party should get any information about these elements. This makes it possible for two parties to discover whether they are holding the same elements *without* revealing them if this is the case.

Assume that B has a collection of sets $S_B^1, \ldots, S_B^v$ and A has a collection of sets $S_A^1, \ldots, S_A^v$. A wants to compute the truth-value of the following predicate:

$$(S_A^1 \cap S_B^1 \neq \emptyset) \wedge (S_A^2 \cap S_B^2 \neq \emptyset) \wedge \ldots \wedge (S_A^v \cap S_B^v \neq \emptyset)$$

B agrees that A can learn the truth-value of the predicate however he does not want to let A find out anything more (e.g. in case $S_A^1 \cap S_B^1 \neq \emptyset$, A should not find a witness for this fact). For simplicity, we assume that $\forall j, |S_B^j| = L$. Let $S_A^1 = \{\alpha_1, \ldots, \alpha_{k_1}\}$ and for $j = 2, \ldots, v$, $S_A^j = \{\alpha_{k_{j-1}} + 1, \ldots, \alpha_{k_j}\}$. Let $k :=$

$k_v = \sum_{j=1}^{v} |S_A^j|$. B computes k polynomials p_i such that $s \in S_B^j$ iff $(p_i(s) = 0) \wedge (k_{j-1} \leq i \leq k_j)$. The degree of each p_i is L, (note that given S_B^j, there are $|\mathbb{F}|-1$ possible choices for each p_i). We define the following expression: $P(x_1, \ldots, x_r) := \mathcal{E}(p_1, \ldots, p_k)(x_1, \ldots, x_k) = \sum_{j=1}^{v} \prod_{i=k_{j-1}+1}^{k_j} p_i(x_i)$ (where $k_0 := 0$). Note that $\mathcal{E}$ is 1-bound. Following the type-1 protocol, A securely computes $P(\alpha_1, \ldots, \alpha_k)$. The lists' intersection predicate is $[P(\alpha_1, \ldots, \alpha_r) = 0]$.

Theorem 5. *The above scheme computes the lists' intersection predicate with error probability at most* $1/|\mathbb{F}|$ *(error probability 0, in the case* $v = 1$*).*

Note two interesting special cases: (1) when v=1, A merely checks whether $S_A \cap S_B \neq \emptyset$. If they are disjoint A does not gain any additional information, and if they have common elements A does not obtain a witness. (2) If $\forall j \, (|S_A^j| = 1) \wedge (S_B^j = S_B)$, A checks whether $S_A \subseteq S_B$ (where $S_A := \cup_j S_A^j$)). If this is not the case A does not gain any information about S_B.

An application of the above is the Settlement Escrows Problem [GM95, BN96]: A (the buyer) and B (the seller) negotiate in some fixed price range $[1, \ldots, N]$. B will accept any offer over p_B and A will give at most p_A. A and B wish to know whether p_A and p_B "cross" i.e. $p_A \geq p_B$. Traditionally this problem is solved by revealing the prices to a third party (escrow). Using the lists' intersection predicate scheme twice with $v = 1$, and $S_A = \{1, \ldots, p_A\}$ and $S_B = \{p_B, \ldots, N\}$ each player checks whether there is a cross, without a third party.

7 Other Applications

SGPEs can capture a variety of other "oblivious" interactions between two players. In the full version we present more applications of our construction for type-1 and type-2 games such as *Oblivious Negotiations*, *Oblivious Bargaining*, *Committing to Large Files* and *Oblivious Scoring.*

Acknowledgments. We would like to thank Dan Boneh and Yuval Ishai for their helpful suggestions.

References

[Ber68] Elwyn R. Berlekamp, *Algebraic Coding Theory.* McGraw-Hill, 1968.

[BW86] Elwyn R. Berlekamp and L. Welch, *Error Correction of Algebraic Block Codes.* U.S. Patent, Number 4,633,470 1986.

[BN00] Daniel Bleichenbacher and Phong Nguyen, *Noisy Polynomial Interpolation and Noisy Chinese Remaindering.* In the Proceedings of EUROCRYPT2000, Lecture Notes in Computer Science, Springer, 2000.

[BN96] Adam M. Brandeburger and Barry J. Nalebuff, *Co-opetition*, Doubleday Publications, 1996.

[CMS99] Christian Cachin, Silvio Micali, and Markus Stadler, *Computationally Private Information Retrieval with Polylogarithmic Communication*, In the Proceedings of EUROCRYPT '99, Lecture Notes in Computer Science, Springer, 1999.

[CG97] Benny Chor and Niv Gilboa, *Computationally Private Information Retrieval*, In the Proceedings of the 29th ACM Symposium on the Theory of Computing, 1997.

[CGKS95] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan, *Private Information Retrieval*, In the Proceedings of the 36th Annual Symposium on Foundations of Computer Science, 1995.

[GIKM98] Yael Gertner, Yuval Ishai, Eyal Kushilevitz, and Tal Malkin, *Protecting Data Privacy in Private Information Retrieval Schemes*, In the Proceedings of the 30th ACM Symposium on the Theory of Computing, 1998.

[GM95] Robert H. Gertner and Geoffrey P. Miller, *Settlement Escrows*, Journal of Legal Studies, Vol. 24, pp.87-122, 1995.

[Gol97] S. Goldwasser, *Multi-party computations: Past and present.* In PODC'97, pages 1–6. invited talk.

[GS98] Venkatesan Guruswami and Madhu Sudan, *Improved Decoding of Reed-Solomon and Algebraic-Geometric Codes.* In the Proceedings of the 39th Annual Symposium on Foundations of Computer Science, 1998.

[GMW87] Oded Goldreich, Silvio Micali and Avi Wigderson, *How to Play any Mental Game — A Completeness Theorem for Protocols with Honest Majority.* In the Proceedings of the 19th ACM Symposium on the Theory of Computing, 1987.

[GSR95] Oded Goldreich, Madhu Sudan and Ronitt Rubinfeld, *Learning Polynomials with Queries: The Highly Noisy Case.* In the Proceedings of the 36th Annual Symposium on Foundations of Computer Science, 1995.

[KY01] Aggelos Kiayias and Moti Yung, *Computationally Perfect Symmetric Encryption based on Polynomial Reconstruction*, manuscript, 2001.

[Kil90] Joe Kilian, *Use of Randomness in Algorithms and Protocols.* MIT Press, Cambridge, Massachusetts 1990.

[KO97] Eyal Kushilevitz amd Rafail Ostrovsky, *Replication is not Needed: Single Database, Computationally-Private Information Retrieval*, In the Proceedings of the 38th Annual Symposium on Foundations of Computer Science, 1997.

[MS77] F. J. MacWilliams and N. Sloane, *The Theory of Error Correcting Codes.* North Holland, Amsterdam, 1977.

[NP99] Moni Naor and Benny Pinkas, *Oblivious Transfer and Polynomial Evaluation.* In the Proceedings of the 31th ACM Symposium on the Theory of Computing, 1999.

[Sud97] Madhu Sudan, *Decoding of Reed Solomon Codes beyond the Error-Correction Bound.* Journal of Complexity 13(1), pp. 180–193, 1997.

[Yao86] Andrew C. Yao, *How to Generate and Exchange Secrets.* In the Proceedings of the 27th Annual Symposium on Foundations of Computer Science, 1986.