Communication Complexity of Key Agreement on Limited Ranges

(extended abstract)

Jin-Yi Cai[1] Richard J. Lipton[2] Luc Longpré [3] Mitsunori Ogihara[4]
Kenneth W. Regan[5] D. Sivakumar[6]

May 1994

[1]Department of Computer Science, State Univ. of NY at Buffalo, 226 Bell Hall, Buffalo NY 14260-2000. Email: `cai@cs.buffalo.edu`. Supported in part by NSF Grants CCR-9057486 and CCR-9319093, and by an Alfred P. Sloan Fellowship.

[2]Department of Computer Science, Princeton University, Princeton, NJ 08544. E-mail: `rjl@cs.princeton.edu`. Supported in part by NSF Grant CCR-9304718.

[3]College of Computer Science, Cullinane Hall, Northeastern University, Boston, MA 02115. Email: `luc@ccs.northeastern.edu`. Research supported in part by NSF Grant CCR-9211174.

[4]Department of Computer Science, University of Rochester, Room 620, Computer Science Building, Rochester, NY 14627. E-mail: `ogihara@cs.rochester.edu`. Supported in part by the NSF under grant CCR-9002292 and the JSPS under grant NSF-INT-9116781/JSPS-ENG-207.

[5]Department of Computer Science, State Univ. of NY at Buffalo, 226 Bell Hall, Buffalo NY 14260-2000. Email: `regan@cs.buffalo.edu`.

[6]Department of Computer Science, State Univ. of NY at Buffalo, 226 Bell Hall, Buffalo NY 14260-2000. Email: `sivak-d@cs.buffalo.edu`.

Abstract

This paper studies a variation on classical key-agreement and consensus problems in which the set S of possible keys is the range of a random variable that can be sampled. We give tight upper and lower bounds of $\lceil \log_2 k \rceil$ bits on the communication complexity of agreement on some key in S, using a form of Sperner's Lemma, and give bounds on other problems. In the case where keys are generated by a probabilistic polynomial-time Turing machine, agreement is shown to be possible with zero communication if every fully polynomial-time approximation scheme (fpras) has a certain symmetry-breaking property.

1 Introduction

A fundamental problem in key agreement between two parties, commonly called "Alice" and "Bob," is for Alice to communicate some string w to Bob over an expensive, noisy, and/or insecure channel. Most work allows w to be any given string, making no assumptions about its source, or considers w to be uniformly generated among strings of some length n. We study cases in which w comes from a relatively small set $S \subseteq \{0,1\}^n$, namely the set of possible outputs of a random variable (figuratively, a key generator) G. Alice and Bob can gain information about S by interacting with G with the help of a private coin, and each can send messages to the other. For instance, G may generate large primes p satisfying certain requirements, and Alice and Bob wish to agree on some suitable p, while exchanging much fewer than $\log_2 p$ bits. Several kinds of problems we consider are:

1. *Any-key agreement.* Upon the end of their conversation, Alice commits to a string $s_A \in S$, Bob commits to $s_B \in S$, and they succeed if $s_A = s_B$.
2. *Selected-key agreement.* Alice commits to a designated key $w \in S$ before the conversation, and they succeed if after the conversation, Bob also commits to w.
3. *Subset agreement.* Upon the end of their conversation, Alice commits to a subset S_A of S of some pre-determined size m, Bob commits to S_B, and they succeed if $S_A = S_B$.
4. *Weak subset agreement.* Same as last item, but with success if $S_A \cap S_B \neq \emptyset$.

For a first motivation, suppose Alice and Bob are users at remote sites. There are many applications that call for generating one or a few primes of length n having special properties, and they want to make sure that with high probability they generate the same primes. The simple method where Alice generates a prime p and sends it to Bob costs n bits. Somewhat more economical is for Alice to send the *rank* of p in the set S of admissible primes, costing $\log_2 \|S\| < n$ bits. However, this requires both computing and inverting the *ranking function* of S as defined on all of $\{0,1\}^n$, which can be (NP-) hard even in cases where S has cardinality two (see [GS91]). Suppose, however, that the generator G makes S relatively small in size. This is not unreasonable: consider either the effect of the small seed space of a good pseudorandom number generator used as input to a prime-number generator, or the possibility that a third party supplying the generator to Alice and Bob could also supply a secret that restricts the range. Then Alice and Bob can gain enough useful information about S by interacting with G to send ranking information based on their samples of S. This has the advantage that so long as G itself does not fall into enemy hands, the information about rank within S itself gives away little information about the actual values in S.

When it is important to select one or m keys from S "at random," our intent is that Alice would build the designated key or keys, and then assist Bob in building the same ones, or at least one of them. This leads to the latter three problems. The first problem arises when any key will serve.

Second, our setting gives what appears to be a new twist on much-studied distributed-consensus problems. In a typical consensus problem involving k processors, each processor is given some value, and the goal is for the processors to agree on one of the values that were initially input. We can extend our setting from Alice and Bob to k parties, and the difference is that now the keys are given as outcomes of a random variable, rather than being pre-set. In this abstract we concentrate on the two-party case, without faulty parties, and state some results about the extension at the end. Our work shows that the problems are interesting even with only Alice and Bob, with some open questions. Like the methods for distributed-consensus lower bounds in [CHLT93, BG93, SZ93, HS93], our lower bounds use a form of *Sperner's Lemma*, but with the difference that the nodes in the simplicial complex used are labeled by random variables, rather than by processor IDs and features of local communication graphs (see [CHLT93] or "views" in [BG93]).

Our work also differs from that of Maurer [Mau91, Mau93] and Ahlswede and Csiszàr [AC93] in its emphasis on the *communication complexity* of the problems: Do restrictions on the size or structure of the key space help Alice and Bob to agree while exchanging substantially fewer than n bits? We treat questions of secrecy and channel noise as secondary, regarding them as factors that can make communication expensive. Orlitsky and others [OG90, Orl90, Orl91b, Orl91a, Orl92, NOS93] have studied communication complexity in settings where Alice observes a random variable X, Bob observes a random variable Y (often dependent on X), and the object is for them to exchange single outcomes each has seen. For example, Y may select two "teams" i and j from a "league" $S = \{1, \ldots, k\}$, while X reveals the winner to Alice. In [Orl90] it is shown that $\lceil \log_2 k \rceil$ bits are necessary and sufficient for a one-message protocol in which Alice tells Bob the winner. If two messages beginning with Bob are allowed, however, Bob can tell Alice the index l of the first bit where the binary representations of i and j differ, and Alice sends the lth bit of her number, making at most $\lceil \log\log k \rceil + 1$ bits; both of these protocols are error-free. Our setting has several differences: (i) the ability for Alice and Bob to take repeated samples of the random variable G, (ii) the lack of advance knowledge by Alice and Bob of the universe $S \subseteq \{0,1\}^n$, and (iii) the inherent role of randomness and error. The selected-key problem can be represented in their framework, with Y null, but even here there are differences in the representation and non-rankability of S. We show:

Theorem 1.1 *For any-key agreement with exponentially vanishing error, $\lceil \log_2 k \rceil$ bits are both sufficient and necessary, where $k = \|S\|$. The upper bound is obtained by a one-message protocol with runtime polynomial in n and k, while the identical lower bound holds for arbitrary protocols, even when Alice is computationally omnipotent and knows the distribution of G.*

This lower bound holds under the assumption that the source G is a "black box" random variable, one that Alice and Bob can interact with only by sampling. When G is a *feasible generator*, namely one computed by a probabilistic polynomial time Turing machine (PPTM), the question of lower bounds leads to the problem of whether every feasible generator has a "monic refinement," as defined in Section 3. This is the same as asking whether the multi-valued mapping from seeds to valid keys has a *single-valued refinement* (see [Sel91]) in the class called "BPPSV" by Grollmann and Selman [GS88]. In Section 3 we relate this further to questions about *fully polynomial randomized approximation schemes* (fpras) for functions in #P (see [KL83, JVV86, JS89]), noting that an fpras can always be "rounded" to take at most two values with high probability.

Theorem 1.2 *If ties between two values in an fpras can be broken, then every feasible generator with polynomial-time decidable key set has a monic refinement, and then any-key agreement for a PPTM can be done with no communication, regardless of the size of S.*

This sets up an interesting contrast between the sharp lower bounds in the case of arbitrary G, versus the problem that proving even a nonzero lower bound in the case of feasible G requires proving that P $\neq$ #P. Lipton [Lip94] observed a related contrast between arbitrary and feasible *channels* (formally, these are virtually the same as our generators) in coding theory, and noted that if there were a usable source in nature that is not feasible, then it could be used to violate the random-polynomial-time analogue of Church's Thesis, which is commonly believed.

2 Main Results

In this abstract we assume familiarity with interactive protocols, the formalization of an r-round *conversation* $(\alpha_1, \beta_1, \ldots, \alpha_r, \beta_r)$ between Alice and Bob (beginning with Alice, and with β_r possibly null), and unambiguous encodings of conversations by binary strings α. (Details may be found in [GMR89].) The case $r = 1$, β_r null is a *one-message protocol*. Each of Alice and Bob is allowed to interact privately with the random variable G by making "sample requests." Each sample request returns some string in $\{0,1\}^n$ according to the distribution of G, and costs n time units. Alice and Bob may also use their private coin in computations. Neither is allowed to see the other's coinflips or sample strings; i.e., there is no "common randomness." When G is fixed we write p_y as short for Prob$[G = y]$. The set $S = \{ y : p_y > 0 \}$ is called the *support set* of G, and the number k stands for an upper bound on its cardinality. We express time bounds in terms of n, k, and the *error tolerance* ϵ of the protocol on the problem at hand. The numbers n, k, and ϵ are known by both Alice and Bob. A function $\epsilon(n)$ that is $o(n^{-c})$ for every fixed $c > 0$ is said to be *negligible*, and if it is $o(2^{-cn})$ for some $c > 0$, then it is *exponentially vanishing*.

The basic method for the upper bounds is to form a "histogram" of the strings received in sample requests, and look for reproducible "gaps" in the observed frequencies.

Theorem 2.1 *Any-key agreement, for any G with range $S \subseteq \{0,1\}^n$ of size k, can be achieved with exponentially-vanishing error by a protocol in which Alice sends one message of at most $\lceil \log_2 k \rceil$ bits, and in which Alice and Bob run in time polynomial in n, k, and* $\log(1/\epsilon)$.

Proof Sketch. Let N be large enough so that for each element $y \in S$, the probability that the observed frequency of y in N independent trials belongs to the interval $[p_y - 1/k^2 \ldots p_y + 1/k^2]$ is greater than $1 - \epsilon/2$. N is bounded by a polynomial in k and $\log(1/\epsilon)$ independent of p_y. Alice makes N sample requests and observes some number $k' \leq k$ of distinct strings returned by G. Let $f_1, \ldots, f_{k'}$, in nonincreasing order, stand for the observed frequencies of the strings in her histogram, and for convenience put $f_{k'+1} = 0$. Then there exists some i, $1 \leq i \leq k'$, such that $f_i - f_{i+1} > 2/k^2$. This represents a "non-negligible gap" in her histogram. Alice chooses any such i (e.g., the least one) and sends i to Bob.

Then Bob makes N sample requests, and forms his own histogram with frequencies $f'_1, \ldots, f'_l$; here l may differ from k'. Then with error at most ϵ, the set S_A of the i most-frequent strings observed by Alice is the same as the set S_B observed by Bob. Thus if Alice commits to the *lexicographically* greatest member of S_A, and Bob likewise with S_B, they succeed with probability at least $1 - \epsilon$. $\square$

The case where all elements of S are equally likely makes $i = k$ and meets the stated upper bound $\lceil \log_2 k \rceil$ in the protocol. Note that on the promise that this is the case, or even that all elements of S occur with probability that is non-negligible in k, Alice and Bob can agree with *zero* communication by doing polynomially-many samples until, with high probability, each sees all k distinct elements. In general, Alice and Bob can try the strategy of doing some predetermined (or communicated) numbers of trials and committing to the lex greatest string each sees. This may fail when S has elements of probability $1/k^c$ for many different values of c. A different strategy is for Alice and Bob to choose their most-frequent elements, and this can be augmented with communication about lex high or low or otherwise "distinctive" elements among the frequent ones. With all of this latitude even for 0-bit and 1-bit protocols, it seems surprising that the upper bound in Theorem 2.1 cannot be lowered even by 1 bit. This is so even when Alice knows the *true* distribution of G and is computationally omnipotent! We first prove this in the case of a 1-message protocol.

Theorem 2.2 *The best success probability achieved by a one-message protocol, where Alice has u distinct messages available to her and Bob is polynomial-time, is bounded above by u/k.*

Proof. Let a PPTM B representing "Bob" be fixed. It is enough to consider random variables G whose range S is a subset of k elements $e_1, \ldots, e_k$ whose identities are known to both Alice and Bob in advance. The one important point is that if $e_i \notin S$, i.e. if $\mathrm{Prob}[G = e_i] = 0$, and Alice and Bob commit to e_i, they are considered *not* to succeed, in accordance with the stipulation that they agree on a member of S.

We furthermore consider random variables G with the property that for some $m > 0$, and all i, $\mathrm{Prob}[G = e_i]$ is a multiple of $1/2^m$. (We use m and 2^m this way for notational uniformity with the next section, and to relate Bob's time bound to m.) The space of all such random variables forms a *simplicial complex* S_k embedded in the nonnegative orthant of the $(k-1)$-dimensional hyperplane of points in $\mathbf{R}^k$ whose coordinates sum to 1. Two nodes in S_k are adjacent iff they differ by $1/2^m$ in two coordinates, and agree in all the others. The maximum clique size in this graph is k, and a k-clique is called a *simplex* in S_k. S_k has k-many extreme nodes G_i defined by $\mathrm{Prob}[G_i = e_i] = 1$, and the nodes G where $\mathrm{Prob}[G = e_i] = 0$ are said to form the *opposite facet* of G_i. Every interior node, i.e. where all elements have nonzero probability, has $k^2 - k$ neighbors, and belongs to $2k$-many simplexes.

Now we define a "coloring function" $C : S_k \to \{1, \ldots, k\}$ for all nodes G by: Take some $i \in S$ and message x_t $(1 \leq t \leq u)$ that maximizes the probability that Bob, given message x_t and sampling G, commits to i. (If i is not unique, any such i will do.) This probability is well-defined since there is no further interaction after x_t is transmitted, and a computationally omnipotent Alice can do no better than committing to i. Then define $C(G) = i$. This coloring satisfies the hypotheses of *Sperner's Lemma*, using the statement in [CHLT93], namely:

> Suppose for each i, $C(G_i) = i$, and for every node G in the opposite facet to i, $C(G) \neq i$. Then there exists at least one simplex whose k mutually-adjacent nodes are given k distinct colors by C.

Since there are only u different messages Alice can send, at least k/u of these nodes are optimized by sending the same message x_t. Since these nodes represent random variables whose component probabilities differ by $1/2^m$ at most, Bob cannot statistically distinguish them in the polynomially-many trials he is allowed. Hence for any element e_j, the differences among these nodes G in the probability that Bob receiving x_t commits to e_j are at most $\delta(m)$, where $\delta(m)$ is exponentially vanishing in m. Since an optimal Alice commits to a different element at each node, there is one at which Bob is correct with probability at most $u/k + \delta(m)$. Letting $m \to \infty$ and exploiting metric completeness and closure yields the conclusion. ☐

If we restricted attention to r.v.'s G with range S of cardinality exactly k, then the argument trivially fails if Alice and Bob know S in advance, but it goes through for the case of arbitrary

$S \subseteq \{0,1\}^n$ if $k \ll 2^n$. The proof works even if Bob runs in sub-exponential time, so long as m can be chosen large enough (e.g. $m = \Theta(n)$) that Bob cannot distinguish the adjacent nodes in his time bound.

The reduction from multi-round protocols to one-message protocols can lower the success probability and blow up the running time by moderate amounts.

Theorem 2.3 *For every b-bit protocol (A, B) for any-key agreement that runs in time t and succeeds with probability p, and $\delta > 0$, there is a one-message protocol (A', B') that succeeds with probability $p - \delta$, and that runs in time linear in t, nearly-linear in 2^b, and polynomial in $1/\delta$.*

Proof Sketch. There are at most $N = 2^b$ possible conversations under the unambiguous encoding. For each i, $1 \leq i \leq N$, let q_i denote the probability that Alice and Bob have conversation α_i, and let p_i be the success probability conditioned on α_i occurring. Then $p = \sum_i p_i q_i$. Elementary calculation shows that there exists some i such that $p_i \geq p - \delta$ and $q_i \geq Q = \delta/N(1 - p + \delta)$.

Both A' and B' are given copies of the old Alice A and the old Bob B to simulate. A computationally-omnipotent A' who knows the distribution of G could calculate i herself and send α_i to B'. We observe that a polynomial-time A' can do almost as well: she can simulate enough runs of the (A, B) protocol so that with confidence at least $1-\delta/2$, she finds a conversation α_i that gives $q_i > Q/2$ and $p_i > p - 2\delta$. The number of runs needed is on the order of:

$$N \log N(1 - p + \delta)(p - \delta)(1 - p + \delta)\log^3(1/\delta)/\delta^3. \tag{1}$$

When B' receives α_i from A', he does multiple runs of the old (A, B) protocol until conversation α_i occurs, and chooses the same value B did in the first such run. With probability at least $1 - \delta/2$, this happens in the first $(1/q_i)\log_e(2/\delta)$ runs, so the time taken by the new Bob upon receiving a message from Alice can be bounded by a constant times $tN\log_2(1/\delta)\cdot(1-p)/\delta$, which is less than the time for A'. The conclusions follow. $\square$

Note that if p is close to 1, and δ is about $(1-p)/2$ in Equation 1, then then the number of trials needed works out roughly to $N\cdot(1/\delta)$. In any event, if we tolerate a falloff in the success probability of the form 1/polynomial, then A' and B' still run in polynomial time. When the new Alice is computationally omnipotent and knows the distribution, however, the hit on the running time is only a factor of $\log(1/\delta)$, which is polynomial even when $1 - p$ is exponentially vanishing. This suffices to prove the lower bound in the following stronger form of Theorem 1.1, while the proof of the upper bound is deferred to the next subsection.

Theorem 2.4 *For any integer $a \geq 0$, in order to attain success probability $1/2^a - 1/2^n$ for any-key agreement with a polynomial-time Bob, $\lceil log_2 k \rceil - a$ bits are both necessary and sufficient.* $\square$

2.1 Other Agreement Problems

Now we study the communication complexity of the other three problems in the Introduction, namely selected-key agreement, subset agreement, and weak subset agreement. Where the allowed error probability ϵ on the protocol is unstated, it is assumed to be exponentially vanishing.

Theorem 2.5 *Agreement on a selected key w can be achieved in expected polynomial time (in n, k, and $1/p_w$) by a one-message protocol that communicates at most $2\lceil \log_2 k \rceil$ bits.*

Proof Sketch. Let w denote the element that Alice wishes to communicate. Let c be such that $p_w > 1/k^c$. Assume further that Alice knows the value of c (if not, she can sample in increasing powers of k until she obtains a good estimate of p_w). Let N be large enough so that the probability that the observed frequency of w in N independent trials is in the interval $[p_w - 1/4k^2 \dots p_w + 1/4k^2]$ is greater than $\epsilon/2$. Alice makes N sample requests and chooses a "gap" of at least $1/k^d$ in her histogram, for some $d > c$. Note that such a gap must exist since $p_w > 1/k^d$. Let f_i and f_{i+1} denote the observed frequencies on either side of the gap, so that $f_i - f_{i+1} > 1/k^d$. Let S_A denote the elements whose observed frequencies are at least f_i. As before, Alice sends to Bob the index i of the gap. In addition, Alice sends the lexicographic rank of w in the set S_A.

Bob samples until he sees all i elements promised by Alice and knows their frequencies with enough confidence to perceive the gap, and then deduces w from the rank information in Alice's message. □

(*Remarks:* If we want Bob to shut himself off in polynomial time in all possible computations, it seems we need Alice also to communicate c to Bob, taking an extra $\log c = \mathrm{loglog}(1/p_w)$ bits. When c is fixed; i.e., when the probability of the selected key is non-negligible, the time bounds are polynomial in k.)

This leaves an interesting question about the gulf between $2\lceil \log_2 k \rceil$ bits in the upper bound and the lower bound of $\lceil \log_2 k \rceil$ bits that carries over from Theorem 1.1. If Bob were able to compute the ranking function of S, as obtains in other cases of key-transfer we know in the literature, then clearly $\lceil \log_2 k \rceil$ bits would suffice. Our *point* is that when S is an arbitrary subset of $\{0, 1\}^n$ of moderate size ($\log_2 k < n/2$), we see no way for Alice to tell Bob what to look for without taking samples and communicating some robust feature of the results, in addition to the ranking information. We suspect that our upper bound is tight, but have not been able to prove it.

For the problem of subset agreement, let m denote the sizes of sets S_A and S_B that Alice and Bob must commit to. We first observe that m can be at most k', where $k' \leq k$ denotes the number of elements that occur with non-negligible probability.

Corollary 2.6 *For subset agreement in expected polynomial time,* $\lceil \log_2 k \rceil$ *bits are necessary and sufficient.*

Proof Sketch. The lower bound follows immediately from Theorem 2.2. The protocol used in the proof of Theorem 2.1 can be modified to work for the subset agreement case. Alice samples sufficiently many times, and chooses a gap i such that there are at least m elements with observed frequencies at least f_i. Such a gap must exist, since there are at least m elements with non-negligible probability. Alice then sends the number of elements above the "chosen gap." Finally Alice and Bob commit to the lexicographically largest m strings from this set. □

(*Remarks:* Again, if Bob is required always to shut himself off in a given time bound, we have Alice send an additional $\mathrm{loglog}(1/p_m)$ bits, where p_m is the frequency of the *mth* most likely element.)

Before proceeding to the problem of weak subset agreement, we prove the upper bound in Theorem 2.4, re-stated in the following way. Let $\ell = \lceil \log_2 k \rceil$.

Proposition 2.7 *With b bits of communication, Alice and Bob can achieve any-key agreement with success probability at least $1/2^{\ell-b} - 1/2^n$, in polynomial time.*

As in the proof of Theorem 2.1, Alice finds an index i, $1 \le i \le k'$, such that the observed frequencies f_i and f_{i+1} satisfy the "gap" requirement $f_i - f_{i+1} > 1/k^2$. Instead of sending i to Bob, Alice sends the *most significant* b bits of the binary representation of i. Bob "fills in" the least significant $\ell - b$ bits of the index i randomly. Clearly, the probability that Bob hits the index that Alice intended is at least $1/2^{\ell-b}$. □

Using similar ideas, we provide an upper bound for weak subset agreement.

Corollary 2.8 *Weak subset agreement can be achieved with $\lceil \log_2 k - \log_2 m \rceil$ bits.*

Proof. Alice chooses a "gap index" i and sends the binary string x that represents the most significant $\lceil \log_2 k - \log_2 m \rceil$ bits of i. Alice and Bob, respectively, initialize S_A and S_B to $\emptyset$. For each possible binary string y of $\lceil \log_2 m \rceil$ bits, Alice and Bob consider the index $j = xy$ obtained by concatenation. Let e^j_A and e^j_B, respectively, denote the lexicographically largest members of the set of elements that Alice and Bob observe to have frequencies at least f_j. Alice and Bob add e^j_A and e^j_B, respectively, to S_A and S_B. Clearly, S_A and S_B have m elements each. Moreover, both Alice and Bob must consider the index i that Alice picked initially; by the proof of Theorem 2.1, $e^i_A = e^i_B$ with very high probability, so $S_A \cap S_B \neq \emptyset$ with high probability. The running times are similar to those in Theorem 2.1. □

Next we prove that $\lceil \log_2 k - 2\log_2 m \rceil$ bits are necessary for weak-subset agreement. We leave open whether either of these bounds can be improved to meet the other.

Corollary 2.9 *Weak subset agreement requires at least $\lceil \log_2 k - 2\log_2 m \rceil$ bits.*

Proof. Let ϵ be an exponentially vanishing quantity. Suppose to the contrary that there is a protocol P that uses $\log_2 k - 2\log_2 m - 1$ bits of communication to succeed with probability $1-\epsilon$ on any random variable G. We show that Alice and Bob can use the protocol P to beat the lower bound of Theorem 2.2: Alice and Bob simply run protocol P to commit to sets S_A and S_B of size m, and then pick elements $s_A \in S_A$ and $s_B \in S_B$ uniformly at random. The probability that $s_A = s_B$ equals $1/m^2$, which is greater than the upper bound of $1/2m^2$ implied by Theorem 2.2, by a non-negligible quantity. □

3 Feasible Generators

In this section we remove the assumption that the generator G is a "black box," and instead suppose that it is modeled by a probabilistic polynomial-time Turing machine (PPTM) M_G. Without much loss of generality we may suppose that M_G has a binary fair coin, and makes m coinflips in any computation. Then M_G can be regarded as an ordinary deterministic Turing machine computing a function from $\{0,1\}^m$ to $\{0,1\}^n$, with uniform distribution over strings $u \in \{0,1\}^m$.

In order to talk about asymptotic complexity in a uniform manner, we give n to M_G on its input tape, and we also suppose that $m(n) = n^{O(1)}$. For a useful extension of generality, we allow the input tape of M_G to hold an *argument string* $x \in \{0,1\}^n$. Then M_G represents an *ensemble* of random variables G_x, with valid key-sets S_x. Formally, for any language S, let $S_x := \{ y : \langle x, y\rangle \in S \}$, where $\langle , \rangle$ is some fixed feasible pairing function.

Definition 3.1. A *feasible generator* consists of a set S and a PPTM M, such that for all arguments x, M makes $m(|x|)$ coinflips and $\text{Prob}_u[M(x,u) \in S_x] > 3/4$.

Note that it is not the case that every random seed u leads to a valid key, but the probability of failure is not too large. If S is polynomial-time decidable, we can equivalently suppose $M(x,u) = \perp$ if $M(x,u) \notin S_x$. Two examples where as yet no deterministic polynomial-time algorithm is known are generating certified primes or normal bases for finite fields (the latter is known for fixed characteristic [Len91]). The existing generators have the above properties: not every run gives a certified prime or a normal basis, though since the certificates or normality can be checked in polynomial time, invalid outputs can be discarded. In general we allow that S may not admit deterministic polynomial-time generation in the sense of Sanchis and Fulk [SF90].

In the primes and GF(2) cases, the argument x is used only for its length n. Jerrum, Valiant, and Vazirani [JVV86] considered generators in which x stands for a graph, and S_x is e.g. the set of perfect matchings in x. They were interested in generating elements with uniform or nearly-uniform distribution on S_x, as holds for normal bases [vzGG90]. We pose intuitively the opposite question: can the distribution on S_x be heavily biased in favor of one element, or a small number of elements?

Definition 3.2. A feasible generator (M,S) has a *monic refinement* M' if for all arguments x, there exists a single $y \in S_x$ such that $\text{Prob}_u[M'(x,u) = y] > 3/4$.

Here the "3/4" is amplifiable by majority vote to give exponentially-vanishing error in polynomially-many trials. The function mapping x to y then belongs to the class BPPSV defined by Grollmann and Selman [GS88]. Having a monic refinement is different from the notion of probabilistically "isolating a unique element" in Chari, Rohatgi, and Srinivasan [CRS93]. They use the method from [MVV87] of assigning random weights to edges so that with high probability, there is a unique minimum-weight perfect matching (when one exists at all), but different random weightings can yield different matchings.

Now we reconsider the problems of Section 1 when the generator is feasible, and when Alice and Bob *share* the argument string x. This models the situation of two remote users working on the same problem who want to generate the same primes or normal bases without common randomness or heavy communication. In these examples the size k of the key sets S is exponential in n, and so the sampling methods for the upper bounds in the last section take too long. Instead:

Proposition 3.1 *Alice and Bob can solve any-key agreement with no communication (and success probability 3/4) for a feasible generator iff it has a monic refinement.* □

We show, however, that the question of monic refinements is hard even when $k = 2$, using a natural class of PPTMs. A function $f : \Sigma^* \to \mathbf{N}$ is said to have a *fully polynomial time randomized approximation scheme* (fpras) [KL83, JVV86] if there is a PPTM M such that for all $x \in \Sigma^*$ and $\epsilon > 0$ (where we suppose $\epsilon = 1/c$ for some integer c):

$$\Pr_u\left[\frac{f(x)}{(1+\epsilon)} \;\le\; M(\langle x, 0^c\rangle, u) \;\le\; f(x)(1+\epsilon)\right] \;>\; 3/4. \tag{2}$$

Jerrum and Sinclair [JS89] showed that the permanent function for "dense" 0-1 matrices, which is still #P-complete, has an fpras.

Note that M is multi-valued. We observe that the approximation can be done by a total function which is at most 2-valued. The "3/4" here and in (2) can be amplified to give exponentially vanishing error.

Proposition 3.2 *Let f have an fpras. Then there is a p-machine M' such that for all $x \in \Sigma^*$ and $c > 0$, there are two values y_1, y_2 such that $f(x)/(1+\epsilon) \leq y_1 \leq y_2 \leq f(x)(1+\epsilon)$ and $\Pr_r[M'(x, 0^c) \in \{ y_1, y_2 \}] > 3/4$.*

The proof idea is to let $a = M(x,u)$ and round a off to the nearest of appropriately-chosen gridpoints. However, if the true value of $f(x)$ is midway between gridpoints, then we may expect "equal scatter" between the two values, with no non-negligible advantage for either. If instead we always "round down," then we have a similar situation when $f(x)$ is close to a gridpoint. We call the problem of whether M' can be made single-valued the "symmetry-breaking problem for fpras." We first show:

Theorem 3.3 *If every 2-valued feasible generator has a monic refinement, then all feasible generators with $S \in$ P have monic refinements.*

The proof is not so simple as for analogous results about NP-machines in [Sel91, HNOS93]. One attempt is to let M be given, and by analogy with the next-ID function of an NP-machine, define $g(x,v) \mapsto b$ if $(\exists u \sqsupseteq vb)\, M(x,u) \in S$. (Here $b \in \{ 0, 1 \}$.) However, v might be a node in the tree of M with very few valid outputs below it, and hence the valid outputs of g may not have high enough probability. A second attempt is to define $g(x,v) \mapsto 1$ if $\Pr_{u \sqsupseteq v}[M(x,u) \in S_x] \geq 1/4$, and $g(x,v) \mapsto 0$ if $\Pr_{u \sqsupseteq v}[M(x,u) \in S_x] \leq 3/4$. Then g does meet the requirements of two-valuedness and high probability, so by hypothesis there is a total single-valued restriction g' and an M' which computes it with high probability. However, depth-first backtrack search on '1' values of g' might take exponential time. Our proof modifies the second attempt to make the search halt in expected polynomial time, using a trick analogous to the "method of conditional probabilities" in [AS92].

Proof Sketch. Given f and the PPTM M, let $q(n) = 2p(n) + 5$. For all a, $0 \leq a \leq p(n)+1$, and all $v \in \{ 0, 1 \}^{<p(n)}$, define

$$g(x,v) \mapsto a \quad \text{if} \quad \Pr_{u \sqsupseteq v}[M(x,u) \in S_x] \in [\frac{2a}{q(n)} \ \ldots \ \frac{2a+3}{q(n)}].$$

This covers $[0 \ldots 1]$ with $p(n)+1$ intervals so that adjacent intervals overlap, but no point is in more than two intervals and there is a large gap between every second interval. Then g is total. Since $graph(f) \in$ P, one can estimate $\Pr_{u \sqsupseteq v}[M(x,u) \in S_x]$ to within an additive term of $1/p(n)^2$ with high probability by taking polynomially many trials. Hence g meets the requirements of two-valuedness and high probability. By hypothesis, g has a monic refinement g'. The probability of error in g' can be made exponentially vanishing in polynomial time, so that with high probability, a search which requests polynomially many values of g' never obtains an erroneous one. The conclusion follows from the observation that if $g'(x,v) = a$, then at least one child w of v has $g'(x,w) \geq a - 1$. The root has value $g'(x,\lambda) = p(n) + 1$. Hence the path which takes the left child iff its value is at most one less than the current node hits the bottom before the probability reaches zero. $\square$

The attempt to do the left-leaning path directly with g again runs into symmetry-breaking problems if the value $g(x,w)$ of the left child of v is in the overlap between "one less" and 'two less." Now we observe:

Theorem 3.4 *If the symmetry-breaking problem can be solved for fpras, then every feasible generator M with $S \in P$ has a monic refinement.*

Proof Sketch. Let M be given, and with reference to the last proof, define

$$h(x,v) = 2^{p(n)} + 2^{|v|} \cdot \|\{\, u \in \{\,0,1\,\}^{p(n)} : u \sqsupseteq v \ \wedge \ M(x,u) \in S_x \,\}\|.$$

Then $h \in \#\mathrm{P}$. We claim that thanks to the padding term $2^{p(n)}$, h has an fpras computable by sampling polynomially many values as in the previous proof. (Before, g only estimated the number of witnesses below node u additively, not up to a multiplicative factor of $(1+\epsilon)$, and might give zero if the number were small, but now that the numbers are between $2^{p(n)}$ and $2^{p(n)+1}$, the factor pulls off a large interval.) Taking $\epsilon \approx 1/p(n)$ makes it possible to cover $[2^{p(n)} \ldots 2^{p(n)+1}]$ by $p(n)+1$ overlapping intervals of roughly equal size whose endpoints are powers of $(1+\epsilon)$. Symmetry breaking for the fpras allows monic selection of these endpoints, which then plays the role of g' in the previous proof. $\square$

The question of whether the lower bounds on any-key agreement in Section 2 carry over to the case of feasible generators motivates the following two hypotheses, intending also that $S \in \mathrm{P}$.

(1) There exist feasible generators (M,S) that have no monic refinement.

(2) (stronger) There exist feasible generators (M,S) such that for every PPTM M', if M satisfies $(\forall^\infty x)(\exists y \in S_x)\,\mathrm{Prob}_u[M'(x,u) = y] \geq 1/k + \epsilon(n)$, where $k = \|S_x\|$, then $\epsilon(n)$ is exponentially vanishing.

We find it strange that we have been unable to show that the failure of the weaker hypothesis (a) causes any "drastic" collapse of complexity classes, even of BPP into RP or ZPP, or relating to knowledge complexity (e.g. in [GOP94]). Moreover, there seems to be no straightforward connection between hypothesis (2) and the hypothesis that good pseudorandom number generators exist. These become interesting problems to study, and seem to be fairly natural and important.

4 Conclusion

An interesting aspect of our matching lower and upper bounds in Theorem 1.1 is that while the former holds for arbitrarily-powerful Alice, the latter can be achieved by polynomial-time computation. The upper bounds also work in the worst case over all possible distributions of G. This leaves open the possibility that when G has the structure of a feasible generator, or when the distribution of keys has some smoothness properties, better upper bounds on both time and communication can be obtained. The upper bounds also carry over to k-party environments in which $k-1$ fault-free processors play the role of "Bob." Our results and their technical content promise interesting and important ramifications related to other recent work cited in the Introduction.

References

[AC93] R. Ahlswede and I. Csiszàr. Common randomness in information theory and cryptography—part I: Secret sharing. *IEEE Trans. Info. Thy.*, 39:1121–1132, 1993.

[AS92] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley, 1992. With an appendix by P. Erdős.

[BG93] E. Borowsky and E. Gafni. Generalized FLP impossibility result for t-resilient asynchronous computations. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 91–100, 1993.

[CHLT93] S. Chaudhuri, M. Herlihy, N. Lynch, and M. Tuttle. A tight lower bound for k-set agreement. In *Proc. 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 206–215, 1993.

[CRS93] S. Chari, P. Rohatgi, and A. Srinivasan. Randomness-optimal unique element isolation, with applications to perfect matching and related problems. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 458–467, 1993.

[GMR89] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM J. Comput.*, 18:186–208, 1989.

[GOP94] O. Goldreich, R. Ostrovsky, and E. Petrank. Computational complexity and knowledge complexity. In *Proc. 26th Annual ACM Symposium on the Theory of Computing*, 1994. to appear.

[GS88] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comput.*, 17:309–335, 1988.

[GS91] A. Goldberg and M. Sipser. Compression and ranking. *SIAM J. Comput.*, 20, 1991.

[HNOS93] L. Hemaspaandra, A. Naik, M. Ogiwara, and A. Selman. Computing solutions uniquely collapses the polynomial hierarchy. Technical Report CS-TR 93-28, Computer Science Dept., SUNY at Buffalo, August 1993.

[HS93] M. Herlihy and N. Shavit. The asynchronous computability theorem for t-resilient tasks. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 111–120, 1993.

[JS89] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM J. Comput.*, 18:1149–1178, 1989.

[JVV86] M. Jerrum, L. Valiant, and V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theor. Comp. Sci.*, 43:169–188, 1986.

[KL83] R. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Proc. 24th Annual IEEE Symposium on Foundations of Computer Science*, pages 56–64, 1983.

[Len91] H.W. Lenstra. Finding isomorphisms between finite fields. *Mathematics of Computation*, 56:329–347, 1991.

[Lip94] R. Lipton. A new approach to information theory. In *Proc. 11th Annual Symposium on Theoretical Aspects of Computer Science*, volume 775 of *Lect. Notes in Comp. Sci.*, pages 699–708. Springer Verlag, 1994.

[Mau91] U. Maurer. Perfect cryptographic security from partially independent channels. In *Proc. 23rd Annual ACM Symposium on the Theory of Computing*, pages 561–572, 1991.

[Mau93] U. Maurer. Secret key agreement by public discussion from common information. *IEEE Trans. Info. Thy.*, 39:733–742, 1993.

[MVV87] K. Mulmuley, U. Vazirani, and V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7:105–113, 1987.

[NOS93] M. Naor, A. Orlitsky, and P. Shor. Three results on interactive communication. *IEEE Trans. Info. Thy.*, 39:1608–1615, 1993.

[OG90] A. Orlitsky and A. El Gamal. Average and randomized communication complexity. *IEEE Trans. Info. Thy.*, 36:3–16, 1990.

[Orl90] A. Orlitsky. Worst-case interactive communication I: Two messages are almost optimal. *IEEE Trans. Info. Thy.*, 36:1111–1126, 1990.

[Orl91a] A. Orlitsky. Interactive communication: balanced distributions, correlated files, and average-case complexity. In *Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 228–238, 1991.

[Orl91b] A. Orlitsky. Worst-case interactive communication II: Two messages are not optimal. *IEEE Trans. Info. Thy.*, 37:995–1005, 1991.

[Orl92] A. Orlitsky. Average-case interactive communication. *IEEE Trans. Info. Thy.*, 38:1534–1547, 1992.

[Sel91] A. Selman. A taxonomy of complexity classes of functions. Technical Report 91–12, Dept. of Comp. Sci., SUNY / Buffalo, 1991. Revised June 1992, to appear in *J. Comp. Sys. Sci.*

[SF90] L. Sanchis and M. Fulk. On the efficient generation of language instances. *SIAM J. Comput.*, 19:281–296, 1990.

[SZ93] M. Saks and F. Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. In *Proc. 25th Annual ACM Symposium on the Theory of Computing*, pages 101–110, 1993.

[vzGG90] J. von zur Gathen and M. Giesbrecht. Constructing normal bases in finite fields. *J. Symb. Comput.*, 10:547–570, 1990.