ON THE HARDNESS OF COMPUTING THE PERMANENT OF RANDOM MATRICES

Uriel Feige and Carsten Lund

Abstract. Extending a line of research initiated by Lipton, we study the complexity of computing the permanent of random n by n matrices with integer values between 0 and $p-1$, for any suitably large prime p. Previous to our work, it was shown hard to compute the permanent of half these matrices (by Gemmell and Sudan), and to enumerate for any matrix a polynomial number of options for its permanent (by Cai and Hemachandra, and by Toda). We show that unless the polynomial-time hierarchy collapses to its second level, no polynomial time algorithm can compute the permanent of every matrix with probability at least $13n^3/p$, nor can it compute the permanent of at least a $(49n^3/\sqrt{p})$-fraction of the matrices. As p may be exponential in n, these represent very low success probabilities for any efficient algorithm that attempts to compute the permanent. For 0/1 matrices, our results show that their permanents cannot be guessed with probability greater than $1/2^{n^{1-\epsilon}}$.

We also show that it is hard to get even partial information about the value of the permanent modulo p. For random matrices we show that *any* balanced polynomial-time 0/1 predicate (e.g., the least significant bit, the parity of all the bits, the quadratic residuosity character) cannot be guessed with probability significantly greater than 1/2 (unless the polynomial-time hierarchy collapses). This result extends to showing simultaneous hardness for linear size groups of bits.

Key words. Permanent, Computational Complexity, Heuristics, Interactive Proof Systems.

Subject classifications. 15A15, 68Q15

1. Introduction

Let $A = \{A_{ij}\}$ be an n by n matrix. Then *permanent* of A is defined as

$$perm(A) = \sum_{\sigma} \prod_{i=1}^{n} A_{i\sigma(i)},$$

where the summation is over all permutations of n elements. Valiant (1979) has shown that computing the permanent of 0/1 matrices is #P-hard (as hard as counting the number of satisfying assignments of a boolean formula). Thus computing the permanent on worst case inputs is hard (cannot be done in polynomial time), unless $P^{\#P} = P$ (and in particular, P=NP). The best known procedure for computing the permanent is due to Ryser (1963) and it takes $O(n^2 2^n)$ arithmetic operations.

Intuitively, there is no feature that distinguishes "worst case" inputs from "average" inputs in terms of computing the permanent. It seems unlikely that random matrices, as opposed to matrices that are obtained by Valiant's reduction, possess some special properties which make their permanents easy to compute. In this work we study the complexity of computing the permanent on random instances, and give evidence that supports the above view.

DEFINITION 1.1. *A random n by n matrix A is obtained by selecting each entry $A_{ij} \in [0, p-1]$ independently, uniformly at random, where p is some prime. Let $\mathcal{M}_{n,p}$ denote the set of all such matrices.*

Our hardness results hold even if we relax the computational problem and request only the permanent modulo the prime p. Note that if $p \geq n!$, this does not affect the complexity of computing the permanent of 0/1 matrices, since for these matrices modular reduction would never occur.

All arithmetic operations in this paper are performed modulo p (whenever it makes sense).

The key to relating average case complexity and worst case complexity is Lipton's surprising observation that the permanent is *randomly self-reducible* (Beaver & Feigenbaum 1990), (Lipton 1991). Lipton's procedure relates the permanent of an arbitrary matrix A to the permanent of $n+1$ random matrices in the following way:

Select a random matrix X. Compute the matrices

$$R_i = A + iX$$

for $1 \leq i \leq n+1$. There exists a degree n polynomial P such that for any i, $P(i) = perm(R_i)$, and furthermore, $P(0) = perm(A)$. Reconstruct P from the values of $perm(R_i)$ and compute $perm(A)$.

The above procedure implies that if the permanent can be computed efficiently and correctly on a fraction of $1-1/3n$ of the matrices, then a randomized procedure can compute it correctly everywhere (with high probability).

In Gemmell *et al.* (1991) and Gemmell & Sudan (1992), the theory of error correcting codes was used to obtain the following improvement: If the permanent can be computed efficiently and correctly on a $(1/2 + o(1))$-fraction of the random matrices, then a randomized procedure can compute it correctly everywhere.

Negative results of a different flavor were obtained by Cai & Hemachandra (1991), Toda (1990) and Amir *et al.* (1990). They considered deterministic m-*enumerators* (polynomial-time algorithms which for any input produce a list of m values, one of which is correct). Cai and Hemachandra and independently Toda concluded that if n^k-enumerators for the permanent exist for some constant k, then the permanent can be computed in deterministic polynomial time. Amir et al. showed that if #SAT (given a CNF formula compute the number of satisfying assignments) has $2^{n^{1-\epsilon}}$-enumerators then the polynomial-time hierarchy would collapse to the fourth level.

1.1. Interactive Proofs. Our hardness results are significantly stronger and rely on a slightly stronger complexity theoretic assumption. They relate the ability to efficiently compute the permanent with some low success probability to the existence of bounded-round interactive proof systems for the permanent.

DEFINITION 1.2. *Let V be a randomized polynomial time verifier which tries to check whether an input x belongs to a language L by exchanging messages with a computationally unbounded prover P. Then (V, P) is said to be an* interactive proof system *for L if the following holds for all x:*

1. *If $x \in L$, then $\exists P : Pr[V(x)$ accepts$] \geq 2/3$.*
2. *If $x \notin L$, then $\forall P : Pr[V(x)$ accepts$] \leq 1/3$.*

A bounded-round interactive proof system *is an interactive proof system in which the number of messages exchanged between the prover and the verifier is bounded by some fixed constant for all inputs. The class of languages that have a bounded-round interactive proof system is called AM.*

We say that (V, P) verifies a function f if (V, P) is an interactive proof system for the language consisting of all pairs of the form $(x, f(x))$.

(More information on interactive proofs, as well as explanation of the name AM, can be found in Babai & Moran (1988), Goldwasser *et al.* (1989).)

Lund *et al.* (1992) constructed an interactive proof system for the permanent modulo p. Their proof system (the LFKN protocol) requires polynomially many rounds. A bounded-round interactive proof system for the permanent modulo some odd prime would imply a major breakthrough in computational complexity theory, for the following reason. Valiant & Vazirani (1986) constructed a randomized reduction that reduces co-NP to computing the permanent modulo an odd prime. If co-NP has bounded-round interactive proof systems, then the polynomial-time hierarchy (PH) collapses to its second level (Boppana *et al.* 1987). Thus the existence of a bounded-round interactive proof system for the permanent implies that the polynomial-time hierarchy collapses to its second level.

PROPOSITION 1.3. *For some constant c, let $(p_1, p_2, \ldots)$ be a sequence of odd primes that such that $p_n \leq 2^{n^c}$ for every n. If there exist a bounded-round interactive proof system that for any n by n integer matrix computes the permanent modulo p_n then $PH = AM = \Sigma_2^P$.*

Some properties of bounded-round interactive proof systems that we use are the ability to perform randomized computations, the ability to perform NP computations, and the ability to lower bound the size of sparse subsets (see Goldwasser & Sipser (1989)).

1.2. Our results. We distinguish between three types of algorithms:

1. Randomized algorithms: Algorithms that for any input, have probability at least ϵ of giving the correct answer. Note that m-enumerators can be viewed as randomized algorithms with success probability $1/m$. We shall consider very low success probabilities, such as 2^{-n}.

2. Heuristics: Deterministic algorithms which succeed on an ϵ-fraction of the instances. The heuristic does not give any indication whether it is right or wrong. Hardness results for $\epsilon > 1/2$ are achieved through the works of Lipton (1991), Gemmell *et al.* (1991), Gemmell & Sudan (1992). Again, we shall consider exponentially small values for ϵ.

3. Randomized heuristics: Randomized algorithms that succeed with probability ϵ, where the probability is taken both over their coin tosses and over the choice of the random input. This generalizes the above two cases.

We will not require that the algorithms work for every pair (n, p), the only requirement is that the pairs for which the algorithms achieve their goals are dense in the following sense:

Definition 1.4. *A sequence* $((n_1, p_1), (n_2, p_2), \ldots)$ *of integer pairs is* polynomially-dense *if there exists a constant* c *such that for every* $n > 0$ *there exists an* i *such that* $n \le n_i \le n^c$ *and* $p_i \le 2^{n^c}$. *In all such sequences, we assume that the* p_i *are odd primes, for all* i.

Theorem 1.5. *Let* $((n_1, p_1), (n_2, p_2), \ldots)$ *be a sequence of integer pairs that is polynomially-dense. If a randomized polynomial time algorithm computes the permanent with probability at least* $13n_i^3/p_i$ *for all inputs from* $\mathcal{M}_{n_i p_i}$ *and for all* i, *then* $PH = AM = \Sigma_2^P$.

Note that for p exponential in n, Theorem 1.5 gives an exponentially small bound for the success probability of polynomial algorithms which try to guess the permanent. Theorem 1.5 also holds if the randomized algorithm only computes the permanent modulo p. For this function it is trivial to achieve success probability $1/p$.

Theorem 1.5 has implications for the hardness of computing the permanent of 0/1 matrices as well, due to a reduction between the two problems (Zanko 1991). If there exists a randomized polynomial time algorithm which computes the permanent of 0/1 matrices with probability $1/2^{n^{1-\epsilon}}$, for some $\epsilon > 0$, then $P^{\#P} = PH = AM$.

Furthermore, even if the above randomized algorithm has access to an oracle in the polynomial-time hierarchy, then our techniques combined with the fact that $\#P$ is hard for the polynomial-time hierarchy (Toda 1989) would imply that the polynomial-time hierarchy collapses (to a level which depends on the complexity of the oracle). This negative result complements Stockmeyer's positive result (Stockmeyer 1983), that for any constant c there exists a procedure in the polynomial-time hierarchy that approximates the permanent to within a factor of $(1 + n^{-c})$.

For randomized heuristics, rather than algorithms, we prove a slightly weaker result:

Theorem 1.6. *Let $((n_1, p_1), (n_2, p_2), \ldots)$ be a sequence of integer pairs that is polynomially-dense.If a polynomial time randomized heuristic computes the permanent with probability at least $49n_i^3/\sqrt{p_i}$ for random inputs from $\mathcal{M}_{n_i p_i}$ for all i, then $PH = AM = \Sigma_2^P$.*

Having established (under the assumption that $PH \neq AM$) that computing the permanent is hard almost everywhere, we address the hardness of computing partial information about the permanent. We assume computation modulo p, and that p is superpolynomial in n. A question with cryptographic motivation is to identify balanced 0/1 predicates which are hard to compute (cf. hard bits for the discrete log, and for the RSA function (Blum & Micali 1984), (Alexi *et al.* 1984)). For the permanent, we are able to prove hardness results for arbitrary predicates.

Theorem 1.7. *Let $((n_1, p_1), (n_2, p_2), \ldots)$ be a sequence of integer pairs that is polynomially-dense, and $p_i = n_i^{\omega(1)}$. Let Ψ be a polynomial time computable balanced 0/1 predicate, where balanced mean that the predicate is 0 on roughly half the instances and 1 on the other half. If for some constant c and for all i, $\Psi(perm(M) \bmod p_i)$ can be computed with probability at least $1/2 + 1/n_i^c$ by a randomized polynomial time heuristic (where probability is taken over random coin tosses of the algorithm, and random choice of matrix M from $\mathcal{M}_{n_i p_i}$), then $PH = AM = \Sigma_2^P$.*

Remark 1.8. *The above does not hold for the permanent of integer matrices without modular reduction. The least significant bit can be computed in polynomial time (by computing the least significant bit of the determinant). Valiant shows that the k least significant bits of the permanent can be computed in time exponential in k, giving a polynomial time algorithm for constant k (Valiant 1979).*

Theorem 1.7 can be extended to balanced functions with range of size nearly $\sqrt{p}$. In particular this establishes the simultaneous hardness of a linear number of balanced predicates. For example, we show:

Theorem 1.9. *Let $((n_1, p_1), (n_2, p_2), \ldots)$ be a sequence of integer pairs that is polynomially-dense, and $p_i = n_i^{\omega(1)}$. For any constant c and $\epsilon > 0$, for any pair (n_i, p_i) and for any $k \leq |p_i|(1/2 - \epsilon)$ (where $|p_i|$ denotes the length of p_i), if a randomized polynomial time heuristic can compute the k least significant bits of the permanent modulo p_i with probability better than $2^{-k}(1 + 1/n_i^c)$, then $PH = AM = \Sigma_2^P$.*

Previously, there was only one other "natural" function which was known to hide a linear fraction of its bits (under suitable complexity assumptions) – the discrete log modulo a composite (Schrift & Shamir 1990).

The cryptographic significance of the hardness results for the permanent is questionable, since the permanent is not a one-way function (it has no "easy" direction).

Our negative results are achieved under the assumption that the permanent does not have bounded-round interactive proof systems. They make extensive use of the power of AM. We obtain less impressive results under the weaker assumption that $P^{\#P} \neq BPP$.

THEOREM 1.10. *If there exists a randomized polynomial time heuristic for computing the permanent modulo p (for $p > n^2$) which succeeds with probability at least $(1/2 - 1/n)$, then the permanent modulo p can be computed in probabilistic polynomial-time.*

Theorem 1.10 represents a small quantitative improvement over Gemmell & Sudan (1992), but a significant qualitative improvement, as we have to deal with heuristics which err on the majority of instances.

1.3. The general approach. The general setting is that we have an auxiliary procedure that gives us some limited information about the permanent function (e.g., it can compute the permanent of a small fraction of the matrices). Using this procedure we construct a bounded-round interactive proof system that verifies that the value of the permanent for an arbitrary matrix M is v, where M and v are given as inputs to the proof system. The basic bounded-round protocol in all these results is a parallel version of the LFKN protocol (Lund *et al.* 1992) for verifying the value of the permanent. (We note that Cai & Hemachandra (1991) also used the LFKN protocol to obtain their negative results about the permanent.)

In all our protocols, even if the auxiliary procedure is faulty (that is, gives incorrect information about the permanent), a cheating prover has only negligible probability of convincing the verifier to accept false inputs. Hence the verifier needs to trust only his own private randomness (rather than trust the auxiliary procedure, or the prover).

In Section 2 we review the LFKN protocol and outline our bounded-round version of the LFKN protocol. In Section 3 we prove Theorem 1.5, in Section 4 we prove Theorem 1.6, in Section 5 we prove theorems 1.7 and 1.9, and in Section 6 we prove Theorem 1.10. The sections 2 to 5 are best read in order, as each section extends the ideas presented in the previous ones.

1.4. Notation. Matrices are denoted by capital letters. If A denotes a matrix, then A_{ij} denotes A's entry in row i and column j, and A^{ij} denotes the cofactor obtained by crossing out row i and column j.

The cardinality of a set R is denoted by $|R|$, and the length (in binary notation) of an integer p is denoted by $|p|$.

We use the word *verify* to denote computations done with the aid of a prover, and the word *check* to denote computations done with the verifier's own resources, without interaction with a prover.

The reader should view p as a "large" (but arbitrary) prime. In particular, the results of Section 5 assume $p > n^{\omega(1)}$.

2. The LFKN protocol and a bounded-round version

Lund *et al.* (1992) constructed an interactive proof system for verification of the value of the permanent for any matrix. Their proof system alternates between two procedures. One procedure reduces the verification of the permanent of one n by n matrix A to the simultaneous verification of the permanents of n matrices, each of order $n-1$. This is done deterministically by using the well known relation:

$$perm(A) = \sum_{j=1}^{n} A_{1j} perm(A^{1j}).$$

The other procedure is a randomized method of reducing the verification of the permanents of n matrices of order $n-1$ to the verification of the permanent of one matrix of order $n-1$. Combining both procedures, one obtains a randomized method of reducing the verification of the permanent of one matrix of order n to the verification of the permanent of one matrix of order $n-1$. Using the above to successively decrease the order of the matrix whose permanent has to be verified, one is eventually left with a small enough matrix whose permanent can be checked in polynomial time.

We now describe the second of these procedures. Assume that the verifier has n pairs (B_1, v_1), (B_2, v_2), …, (B_n, v_n), where B_i is an n by n matrix and $v_i \in \mathbf{F}_p$. (For notational convenience, we treat each B_i as a matrix of order n rather than $n-1$. See also the technical remark at the end of this section.) The verifier has to accept if and only if for all i, $v_i = perm(B_i)$. The idea of the LFKN protocol is to look at the matrices as points in $\mathbf{F}_p^{n^2}$ and interpolate a curve through all the points. Define

$$C(x) = \sum_{i=1}^{n} L_i(x) B_i,$$

where L_i is the polynomial of degree $n-1$ such that

$$L_i(j) = \begin{cases} 1 & \text{if } j = i \\ 0 & \text{if } j \in \{1,2,\ldots,n\} - \{i\}. \end{cases}$$

Note that for $i \in \{1,2,\ldots,n\}, C(i) = B_i$. Furthermore the permanent function restricted to C is a polynomial of degree at most $n(n-1)$. The prover is required to send to the verifier this polynomial f, for which the verifier checks that $f(i) = v_i$ for all i. Thereafter the verifier chooses a random number α in $\mathbf{F}_p$ and it now verifies that the permanent of $C(\alpha)$ is $f(\alpha)$. The correctness of this reduction is based on the fact that two different polynomials of degree d agree on at most d points. If for some i, $perm(B_i) \neq v_i$, then the prover is forced to give the verifier an incorrect polynomial instead of $perm(C(x))$, and therefore with probability $1 - \frac{n(n-1)}{p}$ the permanent of $C(\alpha)$ is not $f(\alpha)$.

PROTOCOL 1.

P→V: $f(x)$, a polynomial of degree at most $n(n-1)$.
V: Checks that $\forall i \in \{1,2,\ldots,n\} : f(i) = v_i$.
V→P: α, chosen uniformly at random from $\mathbf{F}_p$.
V: Verifies that $perm(C(\alpha)) = f(\alpha)$.

It is essential for the correctness of the above protocol that P sends the polynomial $f(x)$ before V sends the random value α. If P knows α beforehand, then regardless of the values v_i claimed as the permanents of the matrices B_i, P can easily choose a polynomial f that agrees with these values and also satisfies $f(\alpha) = perm(C(\alpha))$. Thus the LFKN protocol is inherently sequential, taking n rounds of communication until the order of the matrices is sufficiently reduced. As pointed out in the introduction, no bounded-round protocol for the permanent exists, unless $PH = AM = \Sigma_2^P$.

In this paper we show the strong consequences that would follow from the existence of polynomial time procedures for computing the permanent on random instances, or computing partial information about the permanent. We show that these procedures (which we call *auxiliary procedures*) would enable us to construct a bounded-round version of the LFKN protocol. In this bounded-round protocol, V sends all the values of α (one for each order of matrices) in advance. To prevent P from cheating, clearly V must perform additional tests to the polynomial $f(x)$ that P sends. These tests involve the auxiliary procedure whose existence we assume. The verifier checks that the polynomial the prover is providing is consistent with the information that the auxiliary procedure is giving. With these additional constraints, even a cheating prover will only be able to choose the polynomial $f(x)$ from a small set of "plausible" polynomials. This will be enough for the verifier to succeed.

Therefore, one part in the "parallel" bounded-round protocol is the following:

PROTOCOL 2.

V: Chooses α uniformly at random from $\mathbf{F}_p$.
V→P: α.
(**V:** Verifies in parallel the value of $perm(C(\alpha))$.)
P→V: $f(x)$, a polynomial of degree at most $n(n-1)$.
V: Verifies that $\forall i \in \{1, 2, \ldots, n\} : f(i) = v_i$.
V: Tests the polynomial f, using the auxiliary procedure. The verifier may use the prover to help him test f, by running a test protocol.

In subsequent sections we complete the description of the protocol, by designing several possible test protocols for the polynomial f, each based on a different auxiliary procedure.

The following lemma states how good the testing of f has to be.

LEMMA 2.1. *If for every curve $C(x)$ in $\mathbf{F}_p^{n^2}$, where each coordinate function is a polynomial of degree at most $n-1$,*

1. *$perm|_C$ passes the test with probability at least $1 - \frac{1}{3n}$.*
2. *The number of polynomials that pass the test with probability at least $\frac{1}{3n}$ is at most $\frac{p}{3n^3}$.*

then protocol 2 is a bounded-round interactive proof system for the permanent modulo p.

PROOF. Similar to the proof of correctness for the LFKN protocol.

Consider first the case of a truthful prover, where for each curve $C(x)$, the corresponding polynomial $f(x)$ sent by P satisfies $f(x) = perm|_C$. The only reason V might reject, is if one of the polynomials sent by P fails the test, due to unlucky coin tosses of V. The probability that $perm|_C$ fails the test is at most $1/3n$ (by our assumption on the quality of the test), and P is required to send not more than n polynomials (by the properties of Protocol 2), and hence V accepts with probability at least $2/3$.

Consider now the case of a cheating prover, who claims an incorrect value for $perm(A)$. For each of the curves $C(x)$ sent by V, call a polynomial *plausible* if it passes the test with probability at least $1/3n$. By our assumption on the quality of the test, there are at most $\frac{p}{3n^3}$ plausible polynomials for each curve. Each incorrect plausible polynomial agrees with the true polynomial (representing the true values of the permanents along the curve) on at most $n(n-1)$ points. Since V picks a random point α from the p points on the curve, the probability that some incorrect plausible polynomial gives the true value of $perm(C(\alpha))$ is at most $(n-1)/3n^2$. Recall that altogether, V specifies at most n curves. Hence the probability that the prover can pass all verifications of the type $\forall i \in \{1, 2, \ldots, n\} : f(i) = v_i$ by sending only plausible polynomials is at most $(n-1)/3n$. On the other hand if P chooses to use a non-plausible polynomial then by definition the verifier will accept with probability at most $1/3n$. Hence altogether the verifier accepts an incorrect value with probability at most $(n-1)/3n + 1/3n = 1/3$. $\square$

Observe that in order to have error at most $1/3n$ in testing a polynomial (as required by Lemma 2.1), it suffices to devise a test that has error $1/3$. This follows from standard error reduction techniques (based on repeating the test $O(\log n)$ times in parallel, and accepting if a majority of the repetitions accept.). Hence w.l.o.g., we relax condition 2 of Lemma 2.1 to the following: *The number of polynomials that pass the test with probability at least* $\frac{1}{3}$ *is at most* $\frac{p}{3n^3}$.

Technical Remark: We assume the existence of an auxiliary procedure that operates on matrices of order n. The LFKN protocol successively reduces the order of matrices involved, possibly making the auxiliary procedure ineffective for the matrices obtained. This does not constitute a real problem, since any smaller matrix can be embedded in an n by n matrix (which otherwise contains only 1's along the diagonal and 0 elsewhere) in a way that preserves the value of the permanent.

3. Randomized Algorithms

In this section we will assume that an auxiliary procedure $\mathcal{P}$ has the property that it can guess the value of the permanent with probability at least $q = \frac{13n^3}{p}$ for all matrices. We will show that such a procedure makes it possible to construct a test that satisfies the conditions in Lemma 2.1.

The test will consist of picking 2 random points, x_1 and x_2, on the curve C, and for each of the points checking that the value that the prover claims is the permanent, is one of the values that $\mathcal{P}$ will guess with probability at least q. That is $Pr_r[\mathcal{P}(C(x), r) = f(x)] \geq q$, where r is $\mathcal{P}$'s random seed. This must hold if the prover does not lie.

If q^{-1} is polynomially bounded the verifier could estimate $Pr_r[\mathcal{P}(C(x), r) = f(x)]$ by sampling random values of r. However, for exponentially small q the random sampling approach is ineffective. We solve this problem by letting the prover help the verifier verify that the value $f(x)$ is among $\mathcal{P}(C(x))$'s frequently guessed values. It does so by using the Goldwasser-Sipser lower bound protocol (which we call the GS protocol), that proves that sets are large.

LEMMA 3.1. (GOLDWASSER & SIPSER (1989)) *Let r, a be integers and $L \in AM$. Then there exists a bounded round protocol (P, V) such that*

- *If $|\{x|x \in L$ and $|x| = r\}| \geq a$ then there exists a prover such that the verifier accepts with probability at least $1 - 2^{-n}$.*
- *If $|\{x|x \in L$ and $|x| = r\}| \leq a/2$ then for every prover the verifier accepts with probability at most 2^{-n}.*
- *V runs in time polynomial in r and the time of the verifier that recognizes L.*

(See Section A in the appendix for the proof.)

By using this test the prover is forced to give the verifier a polynomial that for at least half the points on C will give a permanent value that is among the one that $\mathcal{P}$ guesses frequently (See Lemma 3.3).

Now it only has to be shown that there is only a small ($< \frac{p}{3n^3}$, see Lemma 3.4) number of such polynomials. The proof of this uses the fact that two different polynomials of degree d agree on at most d points.

The test described above is the following:

PROTOCOL 3.

V: Chooses x_1, x_2 randomly from $\mathbf{F}_p$.

P$\leftrightarrow$V: Proves that for $i \in \{1, 2\}$, $|\{r \in R | \mathcal{P}(C(x_i), r) = f(x_i)\}| \geq q|R|$, where R is the set of random seeds for $\mathcal{P}$. This test uses the GS protocol (Lemma 3.1).

DEFINITION 3.2. *For a matrix M we call $v \in \mathbf{F}_p$ a* popular *value if the probability that $\mathcal{P}(M)$ outputs v is at least $q/2$.*

LEMMA 3.3. *If the value of f, for at least half the numbers in $\mathbf{F}_p$, is not one of the popular values of $\mathcal{P}$, then f fails the test with probability at least $\frac{2}{3}$.*

PROOF. If $f(x)$ is not a popular value of $\mathcal{P}$ for at least half the numbers x in $\mathbf{F}_p$, then the test will, with probability $3/4$, choose such a value. The GS protocol will discover that the prover is cheating with probability $1-2^{-n}$. Hence the probability of acceptance is at most

$$1/4 + 2^{-n} < 1/3,$$

for $n > 3$. □

The above lemma implies strong restrictions on the polynomials that the prover can send to the verifier in the parallel version of the LFKN protocol.

LEMMA 3.4. *If $q \geq \frac{13n^3}{p}$, then the number of $n(n-1)$ degree polynomials that pass the above test with probability at least $\frac{1}{3}$, is at most $\frac{p}{3n^3}$.*

PROOF. Using Lemma 3.3 we know that the prover is restricted to sending polynomials f that give popular values for at least half the matrices on C. Note that there are at most $2q^{-1}$ popular values for each M on C.

Let $f_1, f_2, \ldots, f_N$ be the set of such polynomials. Let

$$U = \{(x,v) | v \text{ is a popular value for } C(x)\}$$

and $F_i = \{(x, f_i(x)) | x \in \mathbf{F}_p\} \cap U$. Now observe that $|F_i| \geq p/2$ and for $i \neq j$, $|F_i \cap F_j| \leq n(n-1)$ since f_i, f_j are different $n(n-1)$ degree polynomials. Hence using inclusion-exclusion

$$\begin{aligned} |\bigcup_{i=1}^{N} F_i| &\geq \sum_{i=1}^{N} |F_i| - \sum_{i \neq j} |F_i \cap F_j| \\ &\geq Np/2 - \frac{N(N-1)n(n-1)}{2} \\ &> Np/2 - \frac{N^2 n^2}{2}. \end{aligned}$$

Since $\bigcup_{i=1}^{N} F_i \subset U$ we know that $|\bigcup_{i=1}^{N} F_i| \leq \frac{2p}{q}$. Letting $N = \frac{p}{3n^3}$ this implies that $q < \frac{36\,n^4}{p(3\,n-1)} < \frac{13n^3}{p}$ for $n > 4$. Thus if $q \geq \frac{13n^3}{p}$ then $N < \frac{p}{3n^3}$. □

THEOREM 3.5. *For every probabilistic polynomial time procedure $\mathcal{P}$ there exists a bounded-round interactive proof system (V,P) such that for any (n,p):*

- *If $\mathcal{P}$ guesses the permanent modulo p for all matrices in $\mathcal{M}_{n,p}$ with probability at least $q = \frac{13n^3}{p}$ then (V,P) verifies the permanent modulo p for all such matrices.*

- (V, P) *rejects (with probability at least* $\frac{2}{3}$*) all inputs* $((M,p),v)$ *with* $v \neq perm(M) \bmod p$*. This holds even if* $\mathcal{P}$ *is an arbitrary procedure that offers no advantage in guessing the permanent modulo* p*.*

PROOF. Follows from Lemma 2.1, Lemma 3.4. □

PROOF OF THEOREM 1.5. From Proposition 1.3 it is enough to construct a bounded-round interactive proof system that computes the permanent modulo an odd prime. Given a n by n matrix M the prover will first send the verifier the pair (n_i, p_i) for which the probabilistic algorithm works and where $n \leq n_i \leq n^c$ and $p_i \leq 2^{n^c}$. It follows from Theorem 3.5 that there exists a bounded-round interactive proof system for computing the permanent modulo p_i. □

3.1. Randomized algorithms for 0/1-permanents. Theorem 3.5 has the following corollary for guessing the value of the 0/1-permanent.

THEOREM 3.6. *If for some* $\epsilon > 0$*, there exists a polynomial time probabilistic procedure that for all 0/1-matrices correctly computes the permanent with probability at least* $q = \frac{1}{2^{n^{1-\epsilon}}}$*, then* $P^{\#P} = AM$*.*

PROOF. Follows from the fact that the problem of computing the permanent of 0/1-matrices is many-one complete for $\#P$ (Zanko 1991). More precisely, there exists a polynomial time computable function h that maps an n by n m-bit integer matrix M into an N by N 0/1-matrix, where $N = O(n^2 m)$, such that $perm(M) = perm(h(M))$. Setting $p > 2^{n^{3/\epsilon}}$, the proof follows from Theorem 3.5. □

Note the similar result by Amir *et al.* (1990), that #SAT does not have $2^{n^{1-\epsilon}}$-enumerators (n here is the input size, assuming some reasonable encoding scheme) unless $PH = \Sigma_4^P$.

4. Heuristics

In this section, we look at a weaker form of auxiliary procedure than a randomized algorithm. We prove that no *randomized heuristic* has success probability greater than $8\sqrt{\frac{n}{p}}$ of computing the permanent (modulo p) of a random n by n matrix, unless $PH = AM$. Recall that if $\mathcal{P}$ is a randomized heuristic, then its success probability q is defined as $q = \Pr[\mathcal{P}(X) = perm(X)]$, where the probability is taken over the random coin tosses of $\mathcal{P}$ and the random choice of the matrix X from the uniform distribution.

We have seen in the previous section that a randomized algorithm can be used in order to limit the number of polynomials f that the prover may send in reply to a curve. In this section, we use the randomized heuristic for the same purpose. The main problem we encounter is that the curves specified in the parallel version of the LFKN protocol cover only a very small fraction of the possible matrices, and we have no guarantee that the heuristic has a nontrivial success probability in guessing the permanents of this small number of matrices. In order to overcome this difficulty, we show how a randomized heuristic for the permanent can be transformed into a randomized algorithm, in the context of AM protocols. Once we obtain a randomized algorithm, we can use the results of Section 3.

We now describe how to transform the randomized heuristic into a randomized AM algorithm. For an arbitrary matrix A (on which the heuristic may be completely wrong), we use a randomized procedure to restrict the number of values that the prover can claim as $perm(A)$. A conceptually similar procedure in a related context was used by Gemmell & Sudan (1992).

Protocol 4.

V: Choose randomly two matrices X and Y, and define the curve

$$D(x) := \frac{(x-1)(x-2)}{2}A - x(x-2)X + \frac{x(x-1)}{2}Y.$$

V→P: $D(x)$.

P→V: $f'(x)$, a polynomial of degree at most $2n$.

P↔V: Proves using the GS protocol (Lemma 3.1) that $|\{(x,r) \in \mathbf{F}_p \times R | x \neq 0 \text{ and } \mathcal{P}(D(x), r) = f'(x)\}| \geq qp|R|/2$, where R is the set of random seeds for $\mathcal{P}$.

The fact that X and Y were chosen independently at random implies that all matrices on D (except for $D(0) = A$) are uniformly distributed pairwise independent random matrices. Recall that a degree 2 polynomial can be determined uniquely from its value on three points. In our case, $D(0) = A$ is one such point, $D(1) = X$ and $D(2) = Y$ are the other two points. An alternative equivalent way to choose D would have been by specifying two random matrices as its value at points i and j (rather than specifying $D(1)$ and $D(2)$). Hence there is a one-to-one correspondence between $(D(1), D(2))$ and $(D(i), D(j))$, implying that $D(i)$ and $D(j)$ are uniformly distributed pairwise independent random matrices.

LEMMA 4.1. *With probability at least* $1 - \frac{4}{(p-1)q}$ *(over the choice of* X *and* Y*), the auxiliary procedure* $\mathcal{P}$ *has success probability at least* $q/2$ *as a probabilistic heuristic on the set* $\{D(x)|x = 1, 2, \ldots, p-1\}$.

PROOF. Define the random variables $R_x :=$"the probability that $\mathcal{P}(D(x)) = perm(D(x))$," where probability is taken over choice of D (that is, choice of X and Y), choice of x, and the random seed of $\mathcal{P}$. For every $x \in \{1, 2, \ldots, p-1\}$, $E(R_x) = q$ and $\mathrm{Var}(R_x) \leq q(1-q)$. Define the random variable Z to be the probability that $\mathcal{P}$ is correct on a randomly chosen matrix from D. Hence $Z = \frac{1}{p-1}\sum_{x=1}^{p-1} R_x$ and $E(Z) = q$ and $\mathrm{Var}(Z) \leq \frac{q(1-q)}{p-1} < \frac{q}{p-1}$. Using the pairwise independence we get from Chebyshev's bound that

$$\Pr[|Z - q| \geq q/2] < \frac{4q}{(p-1)q^2} = \frac{4}{(p-1)q}.$$

□

Lemma 4.1 will be used in order to show that a truthful prover causes V to accept correct values of the permanent. The next Lemma shows that the behavior of cheating provers is restricted by Protocol 4.

LEMMA 4.2. *If* $q > 8\sqrt{n/p}$ *then the number of polynomials* f' *such that* f' *passes the test of Protocol 4 with probability at least* $\frac{1}{3}$ *is at most* $8/q$.

PROOF. Let $f_1, f_2, \ldots, f_N$ be the set of polynomials that pass the test with probability at least $\frac{1}{3}$. Choose (x, r) uniformly at random, where $x \in \mathbf{F}_p$ and r is a random seed for $\mathcal{P}$. Let E_i denote the event that $f_i(x) = \mathcal{P}(x, r)$. Note that regardless of the curve $D(x)$ that is chosen:

1. $\Pr[E_i] \geq q/4$. This follows from the second part of Lemma 3.1.
2. For $i \neq j$, $\Pr[E_i \cap E_j] \leq 2n/p$, since x has to be a root of $f_i(x) - f_j(x)$.

Hence

$$\begin{aligned} \Pr[\bigcup_{i=1}^{N} E_i] &\geq \sum_{i=1}^{N} \Pr[E_i] - \sum_{i \neq j} \Pr[E_i \cap E_j] \\ &\geq Nq/4 - \frac{N(N-1)2n}{2p}. \end{aligned}$$

Thus if $q > 8\sqrt{n/p}$ then $N < 8/q$, because for $N = 8/q$

$$\Pr[\bigcup_{i=1}^{N} E_i] \geq 2 - \frac{N^2 n}{p} > 1.$$

□

Hence for any matrix A, regardless of the curve $D(x)$ chosen, there are at most $8/q$ polynomials f' for which there is nonnegligible probability of passing the test in Protocol 4. Therefore the prover is restricted to using at most $8/q$ "popular" values for the permanent of A. This implies, by a proof similar to that of Lemma 3.4, that the number of f's that the prover can give in the parallel version of the LFKN protocol is at most $p/3n^3$ if $q \geq 49n^3/p$. (The main inequality being $8p/q > Np/2 - N^2n^2/2$.) This implies the following theorem.

THEOREM 4.3. *For every polynomial time randomized heuristic $\mathcal{P}$ there exist a bounded-round interactive proof system (V, P) such that for any (n, p):*

- *If $\mathcal{P}$ has success probability at least $q = \max(8\sqrt{\frac{n}{p}}, \frac{49n^3}{p})$ (over the choice of random matrix from $\mathcal{M}_{n,p}$ and random seed for $\mathcal{P}$) then (V, P) verifies the permanent modulo p for all such matrices.*
- *(V, P) rejects with high probability all inputs $((M, p), v)$ where $v \neq perm(M) \bmod p$.*

PROOF. Follows from Lemma 2.1 and Lemma 4.2 using an argument similar to the one in Lemma 3.4. □

PROOF OF THEOREM 1.6. Similar to the proof of Theorem 1.5 using Theorem 4.3 □

5. Partial Information

In this section we show that it is difficult to obtain even partial information about the permanent modulo p (for sufficiently large p). In particular, our results imply that for a random matrix M, and k almost half the length of p, a polynomial time algorithm cannot distinguish between the k least significant bits of the permanent and a random number of length k, unless the polynomial-time hierarchy collapses. It is interesting to note that without the reduction modulo p, the least significant bit of the permanent can be computed in polynomial time (by computing the least significant bit of the determinant).

DEFINITION 5.1. *A function $f : \mathcal{Z}_p \longrightarrow \mathcal{Z}_Q$ is δ-*flat *if for any $y \in \mathcal{Z}_Q$, $|f^{-1}(y)| \leq (1+\delta)\frac{p}{Q}$. That is, no output value appears with frequency which is more than $(1+\delta)$ times the average frequency. A function $f : \mathcal{Z}_p \longrightarrow \mathcal{Z}_Q$ is* flat *if for any $y \in \mathcal{Z}_Q$, $|f^{-1}(y)| \leq \frac{p}{Q} + 1$ (corresponding to $\delta \leq Q/p$).*

For $Q = 2$, flat functions correspond to balanced binary predicates, such as the least significant bit of input x, the parity of its bits, or its Jacobi symbol. Examples of δ-flat functions (for various values of δ) are individual bits of the binary representation of x. Higher order bits may require larger values of δ. For larger values of Q, a typical flat function is $x \ mod \ Q$.

5.1. Intuitive introduction. We outline how one may prove that an arbitrary balanced predicate Ψ about the permanent is hard to guess. We build on the ideas of the previous sections, including the bounded-round version of the LFKN protocol and pairwise independent curves.

As in Section 4, for an arbitrary matrix A, we want to restrict the number of values that the prover can claim as its permanent. This is done by passing a pairwise independent curve $C(x)$ through the given matrix, and requesting the prover to send a low degree polynomial $c(x)$ such that $c(x) = perm(C(x))$. For a random matrix on this curve (random value of x), we check that $\Psi(c(x))$ agrees with the value that our auxiliary algorithm is giving us on $C(x)$. If the success rate of this check is not significantly higher than random, we reject the polynomial that the prover sent.

The above idea does not suffice for arbitrary predicates. Consider for example the quadratic residuosity predicate χ, which has a multiplicative property. Let a be an arbitrary quadratic residue. Then for any matrix A, the prover can falsely claim $a \cdot perm(A)$ as its permanent. As the polynomial $c(x)$ the prover may later send $a \cdot perm(C(x))$, without affecting the value of χ. Thus the prover has a choice of $(p-1)/2$ values that it can claim as the permanent of A.

To defeat this sort of cheating, we let the pairwise independent curve $C(x)$ pass through one matrix whose permanent is known (the identity matrix). Now it is easy to reject the polynomials that are constant multiples of $perm(C(x))$. But how can we prove that for arbitrary predicates Ψ, there is no other way in which a cheating prover can choose wrong polynomials that cleverly use the properties of Ψ to avoid rejection? For this we devise an additional procedure of randomizing the value of the permanent.

Note that if the first row of a matrix M is multiplied by a constant α, then its permanent is changed by a multiplicative factor of α. Moreover, if β is added to the entry M_{11}, then the value of the permanent is changed by an additive amount of $\beta perm(M^{11})$ (recall that M^{11} is the $(1,1)$ cofactor of M). If the prover claims a wrong value v as $perm(M)$, then multiplying the first row of M by a randomly chosen α, and adding a randomly chosen β to M_{11}, will send v and $perm(M)$ to two pairwise independent values (assuming that

$perm(M)$ and $perm(M^{11})$ are nonzero), ruining any correlation of the values of Ψ that the prover may have tried to exploit.

To compute the value to which v is sent, the verifier needs to know $perm(M^{11})$. Thus we request the prover to send $c^{11}(x)$ as well, where $c^{11}(x) = perm(C^{11}(x))$ (i.e., the polynomial which gives the permanents of the appropriate cofactors). Cheating provers gain an extra degree of freedom in cheating, by sending a wrong polynomial as $c^{11}(x)$. In what follows, we add some missing pieces to our construction, and complete the proof of correctness. An important part of the proof is showing that with probability $1-O(n^2/p)$, for pairwise independent curves $C(x)$, $gcd(perm(C(x)), perm(C^{11}(x))) = 1$.

5.2. The full proof. We will prove Theorem 1.7 and Theorem 1.9 by proving the following generalization:

THEOREM 5.2. *Let $Q \leq \frac{\sqrt{p}}{n^{\omega(1)}\log p}$. Let $\Psi : \mathcal{Z}_p \rightarrow \mathcal{Z}_Q$ be a flat function, that can be computed in polynomial time. (Pedantically, Ψ is a family of functions, one for each prime p.) Let $((n_1, p_1), (n_2, p_2), \ldots)$ be a sequence of integer pairs that is polynomially-dense, and $p_i = n_i^{\omega(1)}$. If for some constant c and for all i, there exists a random polynomial time heuristic $\mathcal{P}$ that guesses $\Psi(perm(M) \bmod p_i)$ with probability at least $\frac{1}{Q}(1+\epsilon)$ (over $\mathcal{P}$'s own randomness, and the choice of a random matrix M from $\mathcal{M}_{n_i p_i}$, for $\epsilon = 1/n_i^c$), then $PH = AM = \Sigma_2^P$.*

PROOF. We show that any such algorithm $\mathcal{P}$ can be used to restrict for any matrix M the number of possible values that the prover can claim as its permanent. With our assumption on Q, the restriction allows us to apply Theorem 3.5. In what follows we will assume that $\mathcal{P}$ is deterministic. We can assume so without loss of generality since we can let the prover give the verifier a "favorable" seed to be used as the random input to $\mathcal{P}$, such that $\mathcal{P}$ with that random seed will succeed with probability at least $\frac{1}{Q}(1+\epsilon)$, where the probability is taken over the choice of the random input matrices alone. From our proof it will follow that sending a "bad" seed will not help a cheating prover to convince the verifier of a false statement.

For a matrix M, recall that M_{ij} denotes its entry in row i column j, and M^{ij} denotes the cofactor resulting from deleting row i and column j.

Given a matrix M and $\alpha, \beta \in \mathbf{F}_p$, define $M[\alpha, \beta]$ as the outcome of multiplying its first column by α, and then adding β to the (1,1) entry. Note that $perm(M[\alpha, \beta]) = \alpha perm(M) + \beta perm(M^{11})$. Define an equivalence relation $\sim$ by $M \sim M'$ if and only if there exists some $\alpha \in \mathbf{F}_p - \{0\}, \beta \in \mathbf{F}_p$, such that $M' = M[\alpha, \beta]$. Note that if the permanents of M and M^{11} are known then the permanents of all matrices in M's equivalence class can be computed in polynomial time.

Let $\tilde{M}$ denote the equivalence class containing M. A matrix M is *typical* if for some $i \geq 2$, $M_{i1} \neq 0$. The fraction of matrices that are not typical is negligible, p^{-n+1}. If M is typical then $|\tilde{M}| = p(p-1)$, and $\tilde{M}$ is also considered typical.

First let us figure out how good our heuristic will be on these typical equivalence classes. We say that $\tilde{M}$ is *r-good* if it is typical and $\mathcal{P}$ is correct on a $\frac{1}{Q}(1+r)$ fraction of the matrices in $\tilde{M}$. Let $\gamma(r)$ be the fraction of r-good equivalence classes. To obtain the best theorem, we select r that maximizes the product $r\gamma(r)$. Specifically, we require the prover to send to the verifier an $r \geq \epsilon/2$ that satisfies $r\gamma(r) \geq \frac{\epsilon}{4(\lceil \log Q + \log \epsilon^{-1} \rceil + 1)}$. The following lemma states that such an r always exists, and thus a truthful prover can carry out this part of the protocol.

LEMMA 5.3. *There exists an* $i \in \{0, 1, \ldots, \lceil \log Q + \log \epsilon^{-1} \rceil\}$ *such that*

$$2^{i-1}\epsilon\gamma(2^{i-1}\epsilon) \geq \frac{\epsilon}{4(\lceil \log Q + \log \epsilon^{-1} \rceil + 1)}.$$

PROOF. Let $\gamma_i' = \gamma(2^{i-1}\epsilon) - \gamma(2^i\epsilon)$. An averaging argument shows that

$$\sum_{i=0}^{\lceil \log Q + \log \epsilon^{-1} \rceil} \frac{2^i\epsilon}{Q}\gamma_i' \geq \frac{\epsilon}{2Q}.$$

Hence there exists an i such that

$$2^i\gamma_i' \geq \frac{1}{2(\lceil \log Q + \log \epsilon^{-1} \rceil + 1)}.$$

□

For any pair $(m, m') \in \mathbf{F}_p^2$, let $\overline{(m, m')}$ denote the $p(p-1)$ dimensional vector $(\Psi(m), \Psi(2m), \ldots, \Psi(m+m'), \Psi(2m+m'), \ldots, \Psi((p-1)m + (p-1)m'))$ over $\mathcal{Z}_Q$.

Given a matrix A we are going to perform the following test to restrict the number of *nonzero* values that the prover can claim for the permanent of A. This test can then be used to construct a bounded-round protocol for the permanent as in Section 4. The verifier chooses two random matrices R and S. The verifier constructs the curve $C(x)$ such that $C(0) = A, C(1) = R, C(2) = S$ and $C(3) = I$, where I is the identity matrix. That is:

$$\begin{aligned} C(x) &= -\frac{(x-1)(x-2)(x-3)}{6}A + \frac{x(x-2)(x-3)}{2}R \\ &\quad - \frac{x(x-1)(x-3)}{2}S + \frac{x(x-1)(x-2)}{6}I. \end{aligned}$$

The prover is requested to send the verifier two polynomials: $c(x) = perm(C(x))$ and $c^{11}(x) = perm(C^{11}(x))$. The verifier accepts $c(0)$ as a plausible value for the permanent of A if all the following conditions hold:

1. c, c^{11} are non-zero polynomials.
2. $c(3) = c^{11}(3) = 1$.
3. $\gcd(c(x), c^{11}(x)) = 1$.
4. For at least an $\gamma/2$ fraction of the points on $C(x)$, the vectors $\overline{\mathcal{P}(x)}$ and $\overline{(c(x), c^{11}(x))}$ agree on at least $(1+r/2)p(p-1)/Q$ coordinates, where $\overline{\mathcal{P}(x)}$ is the vector $(\mathcal{P}(C(x)[1,0]), \mathcal{P}(C(x)[2,0]), \ldots, \mathcal{P}(C(x)[p-1,p-1]))$ and where $\gamma r = \frac{\epsilon}{4(\lceil \log Q + \log \epsilon^{-1} \rceil + 1)}$.

LEMMA 5.4. *The above tests can be performed by bounded-round protocols with exponentially small error probability.*

PROOF. (SKETCH) The first three tests can be carried out in polynomial time. The fourth test uses the Goldwasser-Sipser protocol for proving that sets are large. The GS protocol is used in two different ways. First it is used to demonstrate that at least a $\gamma/2$ fraction of the points on $C(x)$ are "good" (i.e., the vectors $\overline{\mathcal{P}(x)}$ and $\overline{(c(x), c^{11}(x))}$ agree on at least $(1 + r/2)p(p-1)/Q$ coordinates) by selecting a good point. Then for the good point x, the GS protocol is used again, this time to show that the vectors $\overline{\mathcal{P}(x)}$ and $\overline{(c(x), c^{11}(x))}$ agree on at least $(1 + r/2)p(p-1)/Q$ coordinates. □

LEMMA 5.5. *For a fraction of $1 - 1/n^{\omega(n)}$ of the possible choices of R and S, for correct respective $c(x)$ and $c^{11}(x)$, the verifier accepts with probability $1 - 2^{-n}$, where the probability is taken over random choices of the verifier in the GS protocol.*

PROOF. The correctness of test 2 follows because $C(3)$ is the identity matrix. Therefore neither c nor c^{11} are the zero polynomial, implying the correctness of test 1. The correctness of test 4 follows because the matrices on curve $C(x)$ are pairwise independent, and thus they form a sample space that reflects the averages of the whole space of matrices (the proof is similar to that of Lemma 4.1).

The proof that test 3 is successful is technical and we prove it in Section B of the Appendix. □

We have shown that for arbitrary matrices, the true value of the permanent is among the values that a truthful prover can claim as their permanent. We now restrict the number of false values that a cheating prover can claim for the permanent. We always allow a prover to claim that the permanent of a matrix is zero. To restrict the number of nonzero values that a cheating prover may claim (in Lemma 5.9, to follow), we follow the intuition of Section 5.1 (in Lemma 5.6 below), and combine it with a standard result in coding theory (Lemma 5.7).

LEMMA 5.6. *Let* $(c(x), c^{11}(x)) \neq (c'(x), c'^{11}(x)) \in \mathbf{F}_p[x]^2$ *be two pairs of nonzero polynomials such that* $\gcd(c(x), c^{11}(x)) = 1$, $\gcd(c'(x), c'^{11}(x)) = 1$, $c(3) = c'(3) = 1$, *and* d *is a common bound on the degrees of the polynomials. Then the number of coordinates in which* $\overline{(c(x), c^{11}(x))}$ *and* $\overline{(c'(x), c'^{11}(x))}$ *differ is at least* $p^2(Q-1)/Q - 2p$, *for all but at most* $2d$ *possible values of* x.

PROOF. Consider any $x \in \mathbf{F}_p$ satisfying

$$\begin{vmatrix} c(x) & c^{11}(x) \\ c'(x) & c'^{11}(x) \end{vmatrix} \neq 0$$

Then for any $a, b \in \mathbf{F}_p$ there exist $\alpha, \beta \in \mathbf{F}_p$ such that

$$\begin{aligned} \alpha c(x) + \beta c^{11}(x) &= a \\ \alpha c'(x) + \beta c'^{11}(x) &= b. \end{aligned}$$

Since Ψ is assumed to be flat, the vectors $\overline{(c(x), c^{11}(x))}$ and $\overline{(c'(x), c'^{11}(x))}$ agree on at most $(p-1)(p/Q+1)$ of their coordinates.

Hence, it remains to show that $c(x)c'^{11}(x) = c'(x)c^{11}(x)$ for at most $2d$ values of x. By the bound d on the degree, it suffices to show that $c(x)c'^{11}(x)$ and $c'(x)c^{11}(x)$ differ as polynomials. First note that if $c(x) = c'(x)$ then $c^{11}(x) = c'^{11}(x)$ since $c(x)$ is nonzero. Hence we can assume that $c(x) \neq c'(x)$. Since $c(3) = c'(3)$ we know that some irreducible factor $f(x)$ divides $c(x)$ and $c'(x)$ a different number of times. Assume without loss of generality that $(f(x))^i | \, c(x)$, but $(f(x))^i \not| \, c'(x)$ for some $i \geq 1$. Hence $(f(x))^i | \, c(x)c'^{11}(x)$ but $(f(x))^i \not| \, c'(x)c^{11}(x)$, since $f(x) \not| \, c^{11}(x)$. $\square$

To make use of Lemma 5.6 we need the following technical lemma. Let n and s be positive integers. Consider vectors of length n over the integers in the range $[0, s]$. The *distance* between two vectors is the number of coordinates in which they differ. The *weight* of a vector is the number of non-zero entries in the vector.

Lemma 5.7. *Let $T = T(n, d, w, s)$ denote the maximum number of vectors over $[0, s]$ of length n, each with weight at most w, and any two of which are distance at least d apart. Then*

$$T \leq \frac{dns}{dns - 2nws + w^2(s+1)} \tag{5.1}$$

if $dns - 2nws + w^2(s+1) > 0$.

Proof. The proof is a generalization of a proof given in MacWilliams & Sloane (1981) for the binary case ($s = 1$).

Consider T vectors satisfying the conditions of the lemma. Denote them by x_1, x_2,...,x_T. Let $D(x_i, x_j)$ denote the distance between x_i and x_j. Let $\delta = \sum_{i<j} D(x_i, x_j)$. Considering each pair of vectors separately, and then summing over their coordinates, we obtain a lower bound on δ:

$$\delta \geq \binom{T}{2} d = dT(T-1)/2. \tag{5.2}$$

To upper bound δ, we consider each coordinate separately, and sum over all vectors. The total number of entries in all vectors is nT. The total number of nonzero entries is at most wT. To maximize the count of δ, the number of 0 entries has to be the minimum possible, and they have to be distributed evenly among all coordinates. Likewise, all nonzero entries along any coordinate have to be distributed as evenly as possible among the s possible values $[1, s]$. Thus we obtain:

$$\begin{aligned} \delta &\leq n\left(\frac{n-w}{n}T\frac{w}{n}T + \binom{s}{2}\left(\frac{w}{ns}T\right)^2\right) \\ &= T^2\left(\frac{(n-w)w}{n} + \frac{w^2(s-1)}{2ns}\right). \end{aligned} \tag{5.3}$$

From inequalities 5.2 and 5.3 it follows after some manipulations:

$$T(dns - 2(n-w)ws - w^2(s-1)) \leq dns.$$

The proof of the lemma follows. □

We will use Lemma 5.7 with the following parameters. For a point on a curve $C(x)$, consider those vectors $\overline{(c(x), c^{11}(x))}$ that agree well with $\overline{\mathcal{P}(C(x))}$ (pass test 4 above). The length of the vectors (denoted n in the lemma) is $p(p-1)$, which is the size of an equivalence class. The size of the alphabet $(s+1)$ is Q. The distance between any two vectors (d) is at least $p(\frac{ps}{s+1}-2)$ (see Lemma 5.6). Now modify each such vector by subtracting from it (coordinatewise, where computations are performed modulo Q) the vector $\overline{\mathcal{P}(C(x))}$. Hence we now consider the vectors $\overline{(c(x), c^{11}(x))} - \overline{\mathcal{P}(C(x))}$. (These vectors are well defined, by our assumption that $\mathcal{P}$ is deterministic – see beginning of proof of Theorem 5.2.) The weight of each vector (w) is at most $p(p-1)\frac{s-r/2}{s+1}$, and the other parameters (n, s, d) remain unchanged.

COROLLARY 5.8. *For the above parameters ($d = p(\frac{ps}{s+1}-2)$, $n = p(p-1)$, $w = p(p-1)\frac{s-r/2}{s+1}$, and $s = Q-1$), T of Lemma 5.7 satisfies $T \le 5Q^2/r^2$.*

PROOF. We first compute the value of the denominator D of Equation 5.1.

$$\begin{aligned} D &= p\left(\frac{ps}{s+1}-2\right)p(p-1)s \\ &\quad -2(p(p-1))^2\frac{s-r/2}{s+1}s+\left(p(p-1)\frac{s-r/2}{s+1}\right)^2(s+1) \\ &= \frac{p^2(p-1)(r^2p/4-s^2-2s-r^2/4)}{s+1}. \end{aligned}$$

Using that $s = Q-1$ we obtain that $r^2p/4-s^2-2s-r^2/4 > (p-1)r^2/4-Q^2$, which is at least $(p-1)r^2/5$ for large n since $Q^2 \le p/n^{\omega(n)}$ and $r^2 \ge \epsilon^2/4 \ge 1/n^{O(n)}$. Thus D is positive for large n and is lower bounded by $\frac{p^2(p-1)^2r^2}{5s}$. The numerator of Equation 5.1 is smaller than $p^2(p-1)^2s$, and so $T < 5s^2/r^2$. □

To bound the number of nonzero values that a prover may claim as the permanent of matrix A, it is sufficient to prove the following:

LEMMA 5.9. *Let A be an arbitrary matrix. Let R and S be random matrices. Let m denote the number of pairs of polynomials $(c(x), c^{11}(x))$ that satisfy tests 1 to 4 above. Then $m < \frac{11Q^2}{\gamma r^2} < p/n^{\omega(1)}$.*

PROOF. There are at most $(3n+(3n-3))\binom{m}{2} < 3nm^2$ points on the curve $C(x)$ that are exceptional with respect to some pair of pairs of polynomials (the determinant described in Lemma 5.6 is 0). For any other point and any pair of pairs of polynomials, the determinant described in Lemma 5.6 is nonzero, hence only $5Q^2/r^2$ of the pairs of polynomials can pass test 4 (by Corollary 5.8). Thus the total count of points (including multiplicities) that pass test 4 for some pair of the m pairs is at most $5pQ^2/r^2 + 3nm^2$. On the other hand, since each pair must succeed on at least $p\gamma/2$ points, we obtain that:

$$\frac{mp\gamma}{2} < \frac{5pQ^2}{r^2} + 3nm^2.$$

The above inequality is violated for $m = \frac{11Q^2}{\gamma r^2}$. Let $m = \frac{11Q^2}{\gamma r^2}$ then we get that

$$\frac{pQ^2}{2r^2} < \frac{3\ 11^2 Q^4 n}{\gamma^2 r^4},$$

which implies for some constant c that

$$Q > c\sqrt{p/n}r\gamma \geq \frac{c\sqrt{p/n}\,\epsilon}{4(\lceil \log Q + \log \epsilon^{-1} \rceil + 1)} > \frac{\sqrt{p}}{n^{O(1)} \log p},$$

since $1/\epsilon = n^{O(1)}$. This in turn contradict our choice of $Q = \frac{\sqrt{p}}{n^{\omega(1)} \log p}$. □

We can now complete the proof of Theorem 5.2 by applying Theorem 3.5. □

The proof of Theorem 5.2 can be modified so that it applies to δ-flat functions, for $\delta \leq 1/n^{\omega(1)}$. The statement of Lemma 5.6 would then be changed so that the respective vectors differ on $p^2(Q-1-\delta)/Q-p$ coordinates, rather than $p^2(Q-1)/Q-2p$. This gives weaker values for the parameter d in Corollary 5.8, but the proof can be made to go through. Further details are omitted.

6. Heuristics with success probability $\frac{1}{2} - \frac{1}{n}$

In this section we discuss the case where we have a much better heuristic than in section 4. We assume that the heuristic has success probability $q = \frac{1}{2} - \frac{1}{n}$. Here we obtain a better conclusion, that $P^{\#P} = BPP$. This result improves on previous results in Gemmell *et al.* (1991), Gemmell & Sudan (1992). These results showed that the existence of a randomized heuristic with success probability $\frac{1}{2} + \frac{1}{n^k}$, for some constant k, would imply that $P^{\#P} = BPP$. Their approach relied only on the fact that the permanent is a multilinear polynomial. Our Lemma 6.2 (to follow) is based on these earlier results of Gemmell & Sudan (1992), combined with an observation of Sigal Ar. For our case where the success probability of the heuristic is below half, we augment this approach by a procedure of resolving which of two candidate values for the permanent is the correct value.

In Section 4 the prover had two major tasks:

1. To supply the verifier with polynomials that agree with the permanents of matrices on a given curve.
2. To help lower bound the size of sets that are too small to be sampled efficiently.

In this section, due to the large value of q, the verifier will replace the prover in both the above tasks. Replacing the prover in the first of the above tasks is based on a procedure of Berlekamp & Welch (1991). For the second task, the relevant sets will not be sparse any more, and the verifier can lower bound their size by sampling.

THEOREM 6.1. *If there exists a polynomial time randomized heuristic that for all n and p has success probability at least $q = \frac{1}{2} - \frac{1}{n}$ then $P^{\#P} = BPP$.*

PROOF. The probabilistic polynomial-time procedure has two parts. The first part will allow the procedure to guess the permanent of fixed matrices with probability $\frac{1}{2} - \frac{n^2}{2(p-1)}$ and the second part uses the LFKN protocol to weed out the correct value.

LEMMA 6.2. *If there exists a polynomial time randomized heuristic $\mathcal{P}$, that has success probability at least $q = \frac{1}{2} - \frac{1}{n}$ for random matrices from $\mathcal{M}_{n,p}$, then there exist a polynomial time randomized procedure that for all matrices in $\mathcal{M}_{n,p}$ guesses the permanent with probability at least $\frac{1}{2} - \frac{n^2}{2(p-1)}$.*

PROOF. Given a matrix A, the probabilistic polynomial time procedure M chooses a curve $D(x)$ as in Protocol 4. With probability at least $1 - \frac{n^2}{4(p-1)}$ (over the choice of D), the probabilistic heuristic $\mathcal{P}$ is correct on D with probability at least $\frac{1}{2} - \frac{2}{n}$. For such curves D, there will be at most two *exceptional* polynomials that agree with the heuristic with probability at least $\frac{1}{2} - \frac{2}{n}$. (The proof of this fact repeats arguments that appear in Section 4, and is omitted.) Hence if M could find these two polynomials, it could guess randomly which one is the correct one and therefore guess the value of the permanent of A with probability at least $1/2$.

The following procedure can be used to find an exceptional polynomial. Choose $m = n^2$ random points $x_1, x_2, \ldots, x_m$ on the curve D, and obtain the $\mathcal{P}$'s answers $y_1, y_2, \ldots, y_m$. With some small, but constant probability more than $(m + 2n + 1)/2$ of these answers agree with the exceptional polynomial. The reason is that we are performing m pairwise independent Bernoulli trials with probability of success at least $\frac{1}{2} - \frac{2}{n}$. Hence, the expected number of successes is at least $n^2/2 - 2n$ and the standard deviation is at least $\frac{2n}{5}$, if $n \geq 20$. So we are looking for an event that is $\frac{25}{4}$ standard deviations away from the expected value. This will happen with some small, but constant, probability. (The procedure can be repeated to boost up this probability.)

Hence we are left with the following problem:

PROBLEM 6.3.
GIVEN : *m pairs of points $(x_i, y_i) \in \mathbf{F}_p \times \mathbf{F}_p$, such that there exists a polynomial f of degree at most d satisfying $y_i = f(x_i)$ for at least $\frac{m+d+1}{2}$ values of i,*
QUESTION : *Find f*

Berlekamp & Welch (1991) showed how to solve this problem in deterministic polynomial time. □

LEMMA 6.4. *If there exist a polynomial time randomized algorithm that for all matrices guesses the permanent with probability at least $\frac{1}{2} - \frac{1}{n}$ then $P^{\#P} = BPP$.*

PROOF. Basically, this is a simplified version of Theorem 3.5, and the proof is omitted. The conclusion $P^{\#P} = BPP$ follows from taking $p \geq n!$ (the problem of computing the permanent modulo $p \geq n!$ is $\#P$-complete). □

The proof of Theorem 6.1 follows from the above two lemmas. □

A similar result for heuristics with success probability $1/\text{poly}(n)$ would follow if some polynomial time procedure could find the corresponding exceptional polynomials.

Acknowledgements

This work was motivated by discussions with Sigal Ar and Avi Wigderson. Thanks to Don Coppersmith and Joan Feigenbaum for helpful discussions.

References

W. ALEXI, B. CHOR, O. GOLDREICH, AND C. SCHNORR, RSA/Rabin bits are $1/2 + 1/poly(logN)$ secure. In *Proc. of the 25th IEEE Symp. on Foundations of Computer Science*, 1984, 449–457.

A. AMIR, R. BEIGEL, AND W. I. GASARCH, Some connections between bounded query classes and non-uniform complexity. In *Proc. of the 5th Conference on Structure in Complexity Theory*, 1990, 232–243.

L. Babai and S. Moran, Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *J. of Computer and System Sciences* **36**(2) (1988), 254–276.

D. Beaver and J. Feigenbaum, Hiding instances in multioracle queries. In *Proc. 7th Symp. on Theoretical Aspects of Comp. Sci.* LNCS 415, 1990, 37–48.

E. Berlekamp and L. Welch, Error correction of algebreaic block codes., 1991. US Patent Number 4,633,470.

M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. on Computing* **13**(4) (1984), 850–864.

R. Boppana, J. Hastad, and S. Zachos, Does co-NP have short interactive proofs? *Information Processing Letters* **25**(2) (1987), 127–132.

J. Cai and L. Hemachandra, A Note on Enumerative Counting. *Information Processing Letters* **38**(4) (1991), 215–219.

P. Gemmell and M. Sudan, Highly resilient correctors for polynomials. *Information Processing Letters* **43** (1992), 169–174.

P. Gemmell, R. Lipton, R. Rubinfeld, M. Sudan, and A. Wigderson, Self-testing/correcting for polynomials and for approximate functions. In *Proc. of the 23rd ACM Symp. on the Theory of Computing*, 1991, 32–42.

S. Goldwasser and M. Sipser, Private coins versus public coins in interactive proof systems. In *Randomness and Computation*, ed. S. Micali, vol. 5 of *Advances in Computing Research*. JAI Press, 1989, 73–90.

S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof-systems. *SIAM J. on Computing* **18**(1) (1989), 186–208.

R. Lipton, New directions in testing. In *Distributed Computing and Cryptography*, ed. J. Feigenbaum and M. Merritt, vol. 2 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 191–202. American Mathematical Society, 1991.

C. Lund, L. Fortnow, H. Karloff, and N. Nisan, Algebraic methods for interactive proof systems. *J. of the ACM* **39**(4) (1992), 859–868.

F. MacWilliams and N. Sloane, *The Theory of Error-Correcting Codes*. North-Holland, 1981.

H. J. Ryser, *Combinatorial Mathematics*. Carus Math. Monograph no. 14. 1963.

A. Schrift and A. Shamir, The discrete log is very discreet. In *Proc. of the 22nd ACM Symp. on the Theory of Computing*, 1990, 405–415.

J. T. Schwartz, Probabilistic algorithms for verification of polynomial identities. *J. of the ACM* **27** (1980), 701–717.

L. Stockmeyer, The complexity of approximate counting. In *Proc. of the 21st ACM Symp. on the Theory of Computing*, 1983, 118–126.

S. Toda, On the computational power of PP and $\oplus$P. In *Proc. of the 30th IEEE Symp. on Foundations of Computer Science*, 1989, 514–519.

S. Toda, 1990. Personal communication in Cai & Hemachandra (1991).

L. Valiant, The complexity of computing the permanent. *Theoretical Computer Science* **8** (1979), 189–201.

L. Valiant and V. Vazirani, NP is as easy as detecting unique solutions. *Theoretical Computer Science* **47**(1) (1986), 85–93.

V. Zanko, $\#P$-completeness via many-one reductions. *International Journal of Foundations of Computer Science* (1991).

A. The Goldwasser-Sipser lower bound protocol

We present the Goldwasser-Sipser lower bound protocol, and prove Lemma 3.1.

Let $S \subset \{0,1\}^r$ and let b be an integer.

Protocol 5.

V: Choose a random linear function $h : \{0,1\}^r \to \{0,1\}^b$ and a random $z \in \{0,1\}^b$.

V→P: h, z.

P→V: x.

V: Accepts if and only if $x \in S$ and $h(x) = z$.

Lemma A.1. (Goldwasser & Sipser (1989)) *Let $S \subset \{0,1\}^r$ and let b be an integer.*

- *If $|S| \geq 2^b/2$ then there exists a prover such that the verifier accepts with probability at least $1/4$.*
- *If $|S| \leq 2^b/8$ then for every prover the verifier accepts with probability at most $1/8$.*

Proof of Lemma 3.1. Let $S = \{(x_1, x_2, x_3) | x \in S \text{ and } |x| = r \text{ for } i = 1, 2, 3\}$ and $b = \lfloor 3 \log a \rfloor + 1$. Then if $|\{x | x \in L \text{ and } |x| = r\}| \geq a$ then $|S| \geq a^3 \geq 2^b/2$ and if $|\{x | x \in L \text{ and } |x| = r\}| \leq a/2$ then $|S| \leq a^3/8 \leq 2^b/8$. Thus using the GS protocol in parallel $O(n)$ times and accepting if at least $1/6$ of the subprotocols accept results in a protocol with exponentially low error probability. $\square$

For any fixed $\epsilon > 0$ a bounded-round protocol that distinguishes with exponentially small error-probability between the case when $|\{x | x \in L \text{ and } |x| = r\}| \geq a$ and the case when $|\{x | x \in L \text{ and } |x| = r\}| \leq a(1 - \epsilon)$ can be obtained in a similar way.

B. Proof of correctness of test 3

We assume the same notation as in Section 5.2. In order to prove that the honest prover's answers pass test 3 we need the following technical lemma.

Lemma B.1. *With probability at least* $1 - (3n^2 - n)/2p$

$$\gcd(perm(C^{11}(x)), perm(C^{12}(x))) = x^i (x-2)^j (x-3)^k$$

for some nonnegative integers i, j, k.

Proof. We denote the multiplier of R by $t(x)$ $(= \frac{x(x-2)(x-3)}{2})$. We express the ij'th entry in $C(x)$ as $(p_{ij}(x) + R_{ij} t(x))$, where $p_{ij}(x)$ is a polynomial independent of R_{ij}. Let $c^{11}(x) = perm(C^{11}(x))$ and $c^{12}(x) = perm(C^{12}(x))$.

The proof is by induction on n. The base case is where $n = 2$, i.e.,

$$C(x) = \begin{pmatrix} p_{11}(x) + t(x)R_{11} & p_{12}(x) + t(x)R_{12} \\ p_{21}(x) + t(x)R_{21} & p_{22}(x) + t(x)R_{22} \end{pmatrix}.$$

Hence $c^{11}(x) = p_{22}(x) + t(x)R_{22}$ and $c^{12}(x) = p_{21}(x) + t(x)R_{21}$. Thus with probability at most $1/p$ (over the choice of R_{21}), $c^{12}(x) = 0$, hence we can assume that c^{12} is not the zero polynomial. Let F be some extension field where $c^{12}(x)$ factors completely. Let y in F be some root of $c^{12}(x)$. Note that y does not depend on R_{22}. In order for y to be a root of $c^{11}(x)$, the following must hold:

$$p_{22}(y) + t(y)R_{22} = 0.$$

If y is not a root of $t(x)$ we get that

$$R_{22} = -p_{22}(y)/t(y),$$

which happens only with probability at most $1/p$.

Since there are at most 3 different roots of c^{12} there are at most 3 different values of y and we get that $\gcd(c^{11}(x), c^{12}(x)) \neq x^i(x-2)^j(x-3)^k$ for some i,j,k with probability at most $4/p$.

For the inductive step, to avoid excessive use of indices, we let $B(x)$ denote $C^{11}(x)$. Let $b(x)$ denote the permanent of the corresponding matrices, and let b^{ij} denote the permanents of the corresponding cofactors of $B(x)$. We can assume the induction hypothesis on $b^{11}(x)$ and $b^{12}(x)$. Furthermore, by symmetry we may assume (with probability at least $1-(3(n-1)^2-n+1)/2p$) that $\gcd(b^{11}(x), b^{21}(x)) = x^i(x-2)^j(x-3)^k$ for some i,j,k. We prove that with probability at most $(3(n-1)+1)/p$ (over the choice of R_{22} and R_{32}), $\gcd(c^{11}(x), c^{12}(x)) \neq x^{i'}(x-2)^{j'}(x-3)^{k'}$ for some i', j', k'.

Note that

$$\begin{aligned} c^{11}(x) &= \sum_{i=2}^{n}(p_{i2}(x)+t(x)R_{i2})b^{(i-1)1}(x) \\ &= f(x)+t(x)(R_{22}b^{11}(x)+R_{32}b^{21}(x)), \end{aligned}$$

where $f(x)$ is some polynomial independent of R_{22} and R_{32}.

Note that $c^{12}(x) = g(x) + R_{21}b^{11}(x)$ and that $b^{11}(x) \neq 0$. Therefore c^{12} equals the all zero polynomial with probability at most $1/p$. Hence let us assume that $c^{12}(x) \neq 0$.

Let F be some extension field where $c^{12}(x)$ factors completely. Let y in F be some root of $c^{12}(x)$. Note that y is independent of R_{22} and R_{32}. Assume that y is not a root of $t(x)$. We have to show that with high probability y is not a root of $c^{11}(x)$.

Assume that $c^{11}(y) = 0$. This implies

$$f(y)+t(y)(R_{22}b^{11}(y)+R_{32}b^{21}(y)) = 0.$$

Since $\gcd(b^{11}(x), b^{21}(x)) = x^i(x-2)^j(x-3)^k$ we get that either $b^{11}(y) \neq 0$ or $b^{21}(y) \neq 0$. Assume without loss of generality that $b^{11}(y) \neq 0$. Hence

$$R_{22} = -(f(y)+t(y)R_{32}b^{21}(y))/(t(y)b^{11}(y)).$$

which happens with probability at most $1/p$. Since there are at most $3(n-1)$ roots of c^{12} we obtain the conclusion with probability $\frac{3(n-1)+1}{p}$ when we assume the induction hypothesis on $b^{11}(x)$ and $b^{12}(x)$. □

Lemma B.2. *If $perm(A) \neq 0$ then with probability (over R and S) at least $1-(3n^2+7n-6)/2p$*

$$\gcd(perm(C(x), perm(C^{11}(x))) = 1.$$

Proof. We first show that with probability at least $1-(3n^2+5n-6)/2p$

$$\gcd(perm(C(x)), perm(C^{11}(x))) = x^i(x-2)^j(x-3)^k$$

for some i, j, k, using the randomness in R.

By Lemma B.1, with probability at least $1-(3n^2-n)/2p$, only roots of $t(x)$ can be common roots of $c^{11}(x)$ and $c^{12}(x)$. Using the notation defined in the proof of Lemma B.1, express $c(x) = f(x) + R_{12}t(x)c^{12}(x)$, where $f(x)$ is independent of R_{12}. Consider y, an arbitrary root of $c^{11}(x)$ that is not a root of $t(x)$ (if no such root exists then we are done). Note that y is independent of R_{12} and that there are at most $3(n-1)$ such roots. In order for $c(y)=0$, we need $R_{12} = -f(y)/t(y)c^{12}(y)$, which happens with probability at most $1/p$.

To complete the proof, we have to show that with high probability, roots of $t(x)$ are not roots of $c(x)$. Observe that neither 0 nor 3 can be roots of $c(x)$, since both A and I have nonzero permanents. Likewise, 2 is a root of $c(x)$ only if $perm(S)=0$, and since S is random, this happens with probability at most n/p (follows from the fact that the permanent is a multi-variate polynomial of degree n and that such a polynomial can only be zero with probability n/p for a randomly chosen S (Schwartz 1980)).

Thus the probability that $\gcd(perm(C(x), perm(C^{11}(x))) \neq 1$ is bounded by $(3n^2+5n-6)/2p + n/p = (3n^2+7n-6)/2p$. $\square$

Manuscript received August 10, 1994

Uriel Feige
Department of Applied Math.
The Weizmann Institute
Rehovot 76100, Israel
feige@wisdom.weizmann.ac.il

Carsten Lund
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974-0636
lund@research.att.com