comput. complex. **11** (2002), 158 – 170
1016-3328/02/040158–13
DOI 10.1007/s00037-002-0174-3

computational complexity

ON THE HARDNESS OF APPROXIMATING THE PERMANENT OF STRUCTURED MATRICES

Bruno Codenotti, Igor E. Shparlinski, and Arne Winterhof

Abstract. We show that for several natural classes of "structured" matrices, including symmetric, circulant, Hankel and Toeplitz matrices, approximating the permanent modulo a prime p is as hard as computing its exact value. Results of this kind are well known for arbitrary matrices. However the techniques used do not seem to apply to "structured" matrices. Our approach is based on recent advances in the hidden number problem introduced by Boneh and Venkatesan in 1996 combined with some bounds of exponential sums motivated by the Waring problem in finite fields.

Keywords. Approximation of the permanent, hidden number problem, exponential sums.
Subject classification. 11T23, 15A15, 68Q17.

1. Introduction

Given a matrix $X = (x_{ij})_{i,j=1}^n$ over a ring, we denote by $\operatorname{per} X$ its *permanent*, that is,

$$\operatorname{per} X = \sum_{\sigma \in \mathcal{S}_n} \prod_{i=1}^{n} x_{i\sigma(i)},$$

where $\mathcal{S}_n$ denotes the symmetric group acting on $\{1, \ldots, n\}$.

The permanent has attracted a lot of attention in mathematics and computer science (Minc 1982). It is well known that, unless a very strong complexity conjecture is false, the permanent is very hard to compute. In technical terms, the permanent is $\sharp\mathbf{P}$-*complete*. Thus in a number of papers various *approximability* and *non-approximability* properties of the permanent have been considered, taking into account randomized algorithms as well. In particular, it has been shown by Cai *et al.* (1999) that randomized polynomial time algorithms cannot compute the permanent correctly even on a very small fraction of the instances, unless $\mathbf{P}^{\sharp\mathbf{P}} = \mathbf{BPP}$. Recall that the class $\sharp\mathbf{P}$ is the class

of functions counting the number of accepting computations in a nondeterministic polynomial time Turing machine (see Valiant 1979a,b,c), while the class **BPP** is the analogue of the class **P** for probabilistic computations (with bounded error). The inapproximability results mentioned above apply to arbitrary matrices and take advantage of the random self-reducibility properties of the permanent (see Cai *et al.* 1999; Feige & Lund 1996/1997; Linial *et al.* 1998 and references therein). Such results should be contrasted with the randomized polynomial time approximation schemes, which apply to matrices with nonnegative entries (Jerrum *et al.* 2000).

On the other hand, despite a variety of results on permanents of structured matrices, little is known on the computational complexity of structured permanents, except for very special cases.

Here we propose an alternative approach which allows us

- to obtain conditional non-approximability results for symmetric matrices,
- to prove a connection between approximation and exact computation for circulant, Hankel and Toeplitz matrices, and, more generally, for all classes of matrices which are closed under scalar multiplication. For more on these matrices and on other interesting families of matrices closed under scalar multiplication, the reader is encouraged to see Pan (2001).

We recall that a square matrix $X = (x_{ij})_{i,j=1}^{n}$ is a *Hankel matrix* if its entries x_{ij} depend only on $i + j$, and a *Toeplitz matrix* if its entries x_{ij} depend only on $i - j$. *Circulant matrices* are a special family of Toeplitz matrices, whose entries x_{ij} depend only on $i - j \pmod{n}$.

For these classes of matrices over a finite field $\mathbb{F}_p$ of p elements, where p is prime, we prove that if computing the permanent is hard then approximating the permanent is hard as well.

More precisely, if $\mathcal{PER}_{\mathcal{M}_n,\mu}$ denotes an oracle which, given any matrix X over $\mathbb{F}_p$ from a family which is closed under scalar multiplication, outputs an approximation to $\operatorname{per} X$, we prove that there exists an efficient probabilistic algorithm which makes polynomially many calls to $\mathcal{PER}_{\mathcal{M}_n,\mu}$ and evaluates $\operatorname{per} X$ correctly with high probability. This reduction works for rather crude approximations to $\operatorname{per} X$.

This approach certainly applies to general matrices as well, although in this case Theorems 1.7 and 1.9 of Feige & Lund (1996/1997) give a much stronger result. However, the method of proof does not apply to structured matrices. Indeed, the transformation described in the proof of Theorem 5.2 of Feige & Lund (1996/1997) does not preserve structural properties as being symmetric or Toeplitz.

Our method takes advantage of recent advances in the *hidden number problem*, a problem introduced by Boneh & Venkatesan (1996, 1997). The approach of Boneh & Venkatesan (1996, 1997) (which is based on lattice reduction algorithms) combined with exponential sum techniques has led to a number of results in cryptography and complexity theory (González Vasco & Shparlinski 2001, 2002; Li *et al.* 2002; Mahassni *et al.* 2001; Nguyen & Shparlinski 2002, 2003; Shparlinski 2001a,b, 2002; Shparlinski & Winterhof 2003).

Here we show that the above combination of two celebrated techniques, lattice reduction and bounds of exponential sums, can be applied to studying the permanent.

For integers s and $p \ge 1$ we denote by $\lfloor s \rfloor_p$ the remainder of s on division by p.

For an integer p and a real $\eta > 0$ we denote by $\mathrm{APPROX}_{\eta,p}(t)$ any integer u which satisfies the inequality

$$| \lfloor t \rfloor_p - u| < \frac{p}{2^{\eta+1}}. \tag{1.1}$$

Thus, roughly speaking, if η is an integer, then $\mathrm{APPROX}_{\eta,p}(t)$ is an integer having about η most significant bits same as $\lfloor t \rfloor_p$. However, this definition is more flexible and better suited to our purposes. In particular we remark that η in inequality (1.1) need not be an integer.

We always assume that the field $\mathbb{F}_p$ consists of elements $\{0, \ldots, p-1\}$, so that we can apply $\mathrm{APPROX}_{\eta,p}$ to elements of $\mathbb{F}_p$.

Using the above notation, we can formulate the hidden number problem as follows:

> Let $\alpha \in \mathbb{F}_p$. Assuming we have access to values $\mathrm{APPROX}_{\eta,p}(\alpha t)$, for some $\eta > 0$ and for many known random values $t \in \mathbb{F}_p^*$, recover the number α.

It is clear that the only case of interest occurs when $\eta < \log p$. In Boneh & Venkatesan (1996) a polynomial time algorithm has been given which recovers α for $\eta \sim \log^{1/2} p$. However it has turned out that for many applications the property that t is randomly selected from $\mathbb{F}_p^*$ is too restrictive (see González Vasco & Shparlinski 2001, 2002; Li *et al.* 2002; Mahassni *et al.* 2001; Nguyen & Shparlinski 2002, 2003; Shparlinski 2001a,b, 2002; Shparlinski & Winterhof 2003). For those applications one has rather to study the case when t is selected at random from a certain sequence $\mathcal{T}$ of elements from $\mathbb{F}_p$. The above papers show that the uniformity of distribution properties of $\mathcal{T}$ plays a crucial role and thus exponential sums have been brought into the problem. However,

for some sequences, for example, for very small multiplicative subgroups of $\mathbb{F}_p^*$ such uniformity results are not available. To deal with this case, in Shparlinski & Winterhof (2003) a modification of the basic algorithm has been proposed which replaces the sequence $\mathcal{T}$ with the sequence of all k-sums of the elements of this sequence. This change usually amplifies the uniformity of distribution properties up to the required level. In Shparlinski & Winterhof (2003) this approach has been applied to proving a bit security result for the Diffie–Hellman scheme. Here we use the same approach to study permanents.

Throughout the paper $\log x$ always denotes the binary logarithm of $x > 0$ and the constants in the 'O'-symbols may occasionally, where obvious, depend on a small positive parameter ε and are absolute otherwise. We always assume that p is a prime number with $p \geq 5$, thus the expressions $\log\log p$ and $\log\log\log p$ are defined (and positive).

2. Hidden number problem and Waring problem in finite fields

We recall that the *discrepancy* $\mathcal{D}(\Omega)$ of a sequence $\Omega = (\omega_\nu)_{\nu=0}^{N-1}$ of N elements of the interval $[0,1]$ is defined as

$$\mathcal{D}(\Omega) = \sup_{J \subseteq [0,1]} \left| \frac{A(J,N)}{N} - |J| \right|,$$

where the supremum is extended over all subintervals J of $[0,1]$, $|J|$ is the length of J, and $A(J,N)$ denotes the number of points ω_ν in J, for $0 \leq \nu \leq N-1$. For our purposes we also need the following definition. We say that a finite sequence $\mathcal{T}$ of elements of $\mathbb{F}_p$ is Δ-*homogeneously distributed modulo* p if for any $a \in \mathbb{F}_p^*$ the discrepancy of the sequence $(\lfloor at \rfloor_p / p)_{t \in \mathcal{T}}$ is at most Δ.

Our principal tool is the following statement which is Lemma 4 of Nguyen & Shparlinski (2002) and which is a generalization of Theorem 1 of Boneh & Venkatesan (1996). The proof makes use of an approximation algorithm for the *closest vector problem* in a lattice and follows the same lines as the proof of Theorem 1 of Boneh & Venkatesan (1996).

LEMMA 2.1. *Let $\omega > 0$ be an arbitrary absolute constant. For a prime p, define*

$$\eta = \left\lceil \omega \left(\frac{\log p \log\log\log p}{\log\log p} \right)^{1/2} \right\rceil, \qquad d = \left\lceil \frac{3 \log p}{\eta} \right\rceil.$$

Let $\mathcal{T}$ be a sequence of elements of $\mathbb{F}_p$, $2^{-\eta}$-homogeneously distributed modulo p. There exists a probabilistic polynomial time algorithm $\mathcal{A}$ such that for

any $\alpha \in \mathbb{F}_p$, given as input the prime p, d elements $t_1, \ldots, t_d \in \mathcal{T}$, and d integers

$$u_i = \mathrm{APPROX}_{\eta,p}(\alpha t_i), \qquad i = 1, \ldots, d,$$

for sufficiently large p, its output satisfies

$$\Pr[\mathcal{A}(p, t_1, \ldots, t_d; u_1, \ldots, u_d) = \alpha] \ge 1 - p^{-1},$$

where the probability is taken over all $t_1, \ldots, t_d$ chosen uniformly and independently at random from the elements of $\mathcal{T}$ and over all random choices of the algorithm $\mathcal{A}$.

For a prime p and $z \in \mathbb{F}_p$ we use the notation

$$\mathbf{e}_p(z) = \exp(2\pi i z/p).$$

Thus, we see that in order to use Lemma 2.1 we need to establish a certain uniformity of distribution property of the sequence $\mathcal{T}$ which naturally leads to a problem of estimating exponential sums with elements of our sequence $\mathcal{T}$.

By Theorem 1 of Konyagin (1992) we have the following bound (see also Cochrane *et al.* 2003; Konyagin 2002; Konyagin & Shparlinski 1999).

LEMMA 2.2. *For any $0 < \varepsilon < 1$ there exists a constant $c(\varepsilon) > 0$ such that for any integer n with*

$$n \le \frac{p(\log\log p)^{1-\varepsilon}}{\log p}$$

we have the bound

$$\max_{a \in \mathbb{F}_p^*} \left| \sum_{\lambda \in \mathbb{F}_p^*} \mathbf{e}_p(a\lambda^n) \right| \le p\left(1 - \frac{c(\varepsilon)}{(\log p)^{1+\varepsilon}}\right).$$

Unfortunately the bound of Lemma 2.2 is too weak to be applied directly. We however apply it to k-sums of nth powers of elements of $\mathbb{F}_p$.

For integers $k, n \ge 1$ and an element $t \in \mathbb{F}_p$ we denote by $N_{k,n,p}(t)$ the number of solutions of the equation

$$\lambda_1^n + \cdots + \lambda_k^n = t, \qquad \lambda_1, \ldots, \lambda_k \in \mathbb{F}_p^*.$$

The problem of finding the smallest possible value of k for which the congruence (or in more traditional settings the corresponding equation over $\mathbb{Z}$) has a solution for any t is known as the *Waring problem*. However for our purposes just solvability is not enough. Rather we need an asymptotic formula for the number of solutions.

We show that Lemma 2.2 can be used to prove that for reasonably small k, $N_{k,n,p}(t)$ is close to its expected value.

LEMMA 2.3. *For any $0 < \varepsilon < 1$ there exists a constant $C(\varepsilon) > 0$ such that for any integer n with*

$$n \le \frac{p(\log\log p)^{1-\varepsilon}}{\log p}$$

the bound

$$\max_{t\in\mathbb{F}_p}\left|N_{k,n,p}(t) - \frac{(p-1)^k}{p}\right| \le \frac{(p-1)^k}{p^2}$$

holds for any integer $k \ge C(\varepsilon)(\log p)^{2+\varepsilon}$ and sufficiently large p.

PROOF. The well known identity (see for example Lidl & Niederreiter 1997, Chapter 5.1)

$$\sum_{a\in\mathbb{F}_p} \mathbf{e}_p(au) = \begin{cases} 0 & \text{if } u \in \mathbb{F}_p^*, \\ p & \text{if } u = 0, \end{cases}$$

implies that

$$\begin{aligned} N_{k,n,p}(t) &= \sum_{\lambda_1,\ldots,\lambda_k\in\mathbb{F}_p^*} \frac{1}{p}\sum_{a\in\mathbb{F}_p} \mathbf{e}_p(a(\lambda_1^n + \cdots + \lambda_k^n - t)) \\ &= \frac{1}{p}\sum_{a\in\mathbb{F}_p} \mathbf{e}_p(-at)\Big(\sum_{\lambda\in\mathbb{F}_p^*} \mathbf{e}_p(a\lambda^n)\Big)^k. \end{aligned}$$

Separating the term $(p-1)^k/p$, corresponding to $a = 0$, and applying Lemma 2.2 to other terms, we obtain

$$\max_{t\in\mathbb{F}_p}\left|N_{k,n,p}(t) - \frac{(p-1)^k}{p}\right| \le (p-1)^k\left(1 + \frac{1}{p-1}\right)^{k-1}\left(1 - \frac{c(\varepsilon)}{(\log p)^{1+\varepsilon}}\right)^k$$

and the desired result follows. □

Assume that for $\alpha \in \mathbb{F}_p^*$ and an integer $n \ge 1$ we are given an oracle $\mathcal{HNP}_{n,\mu,p}$ such that for every $\lambda \in \mathbb{F}_p^*$, it returns $\mathrm{APPROX}_{\mu,p}(\alpha\lambda^n)$.

LEMMA 2.4. *Let $\vartheta > 0$ be an arbitrary absolute constant and let*

$$\mu = \vartheta\left(\frac{\log p \log\log\log p}{\log\log p}\right)^{1/2}.$$

There exists a polynomial time probabilistic algorithm which, for any $0 < \varepsilon < 1$ and any integer n with

$$n \le \frac{p(\log\log p)^{1-\varepsilon}}{\log p},$$

makes $O(\mu^{-1}(\log p)^{3+\varepsilon})$ calls of the oracle $\mathcal{HNP}_{n,\mu,p}$ and then recovers α with probability at least $1 + O(2^{-\mu/2})$.

PROOF. Take $C(\varepsilon)$ from Lemma 2.3, and set

$$k = \lceil C(\varepsilon)(\log p)^{2+\varepsilon})\rceil, \quad \eta = 2\mu/3, \quad d = \lceil 3\eta^{-1}\log p\rceil.$$

Then by Lemma 2.3 the sequence

$$\mathcal{T} = (\lambda_1^n + \cdots + \lambda_k^n \mid \lambda_1, \ldots, \lambda_k \in \mathbb{F}_p^*)$$

of k-sums of nth powers of elements of $\mathbb{F}_p^*$ is $2^{-\eta}$-homogeneously distributed modulo p. Now we call the oracle $\mathcal{HNP}_{n,\mu,p}$ for dk uniformly and independently at random chosen

$$\lambda_{11}, \ldots, \lambda_{1k}, \ldots, \lambda_{d1}, \ldots, \lambda_{dk} \in \mathbb{F}_p^*$$

and get integers u_{hj} with

$$|\lfloor \alpha\lambda_{hj}^n \rfloor_p - u_{hj}| < p/2^{\mu+1}, \quad h = 1, \ldots, d,\ j = 1, \ldots, k.$$

For $h = 1, \ldots, d$ we put

$$v_h = \sum_{j=1}^{k} \left\lfloor \alpha\lambda_{hj}^n \right\rfloor_p, \quad t_h = \left\lfloor \sum_{j=1}^{k} \lambda_{hj}^n \right\rfloor_p, \quad u_h = \sum_{j=1}^{k} u_{hj}$$

(where we used addition over $\mathbb{Z}$).

Note that for sufficiently large p,

$$|v_h - u_h| < kp/2^{\mu+1} \le p/2^{\eta+1}.$$

Next, we have

$$v_h - u_h - \lfloor v_h \rfloor_p + \lfloor u_h \rfloor_p = \nu p \quad \text{with } \nu \in \{-1, 0, 1\}.$$

If $\nu = 1$ then we obtain

$$\lfloor v_h \rfloor_p - \lfloor u_h \rfloor_p + p = |v_h - u_h| < p/2^{\eta+1},$$

which is only possible if $\lfloor v_h \rfloor_p > p - p/2^{\eta+1}$. If $\nu = -1$ then we obtain

$$\lfloor u_h \rfloor_p - \lfloor v_h \rfloor_p + p = |v_h - u_h| < p/2^{\eta+1}$$

which is only possible if $\lfloor v_h \rfloor_p < p/2^{\eta+1}$. If $\nu = 0$ then we obtain

$$|\,\lfloor \alpha t_h \rfloor_p - \lfloor u_h \rfloor_p\,| = |\,\lfloor v_h \rfloor_p - \lfloor u_h \rfloor_p\,| = |v_h - u_h| < p/2^{\eta+1}.$$

By Lemma 2.3, the probability that $p/2^{\eta+1} \le \lfloor v_h \rfloor_p \le p - p/2^{\eta+1}$ for all $h = 1, \ldots, d$ is $1 + O(d2^{-\eta})$. Now the algorithm of Lemma 2.1 yields the correct α with probability at least $1 + O(d2^{-\eta} + p^{-1}) = 1 + O(2^{-\mu/2})$ if p is sufficiently large. □

3. Main result

In this section, we exploit the advances in the hidden number problem to prove our main results. Roughly speaking, we show that approximating the permanent of matrices which belong to a class which is closed under scalar multiplication is as hard as computing it. We then use the known fact that the permanent of general matrices is hard to compute (unless $\mathbf{P}^{\sharp\mathbf{P}} = \mathbf{BPP}$) to prove a hardness result for symmetric matrices.

In the following, we give some definitions, and we state and prove these results.

We say that a class $\mathcal{M}_n$ of $n \times n$ matrices with entries from $\mathbb{F}_p$ is *closed under scalar multiplication* if for any $X = (x_{ij})_{i,j=1}^n \in \mathcal{M}_n$ and any $\lambda \in \mathbb{F}_p^*$ we also have $X_\lambda = (\lambda x_{ij})_{i,j=1}^n \in \mathcal{M}_n$.

Recall that $\mathcal{PER}_{\mathcal{M}_n,\mu}$ denotes an oracle which, given any $X \in \mathcal{M}_n$, outputs $\mathrm{APPROX}_{\mu,p}(\operatorname{per} X)$.

THEOREM 3.1. *Let $\vartheta > 0$ and $\varepsilon > 0$ be arbitrary constants. Then for any class $\mathcal{M}_n$ of matrices over $\mathbb{F}_p$ which is closed under scalar multiplication, of size*

$$n \le \frac{p(\log\log p)^{1-\varepsilon}}{\log p},$$

there exists a probabilistic algorithm running in time polynomial in n and $\log p$ which for any $X \in \mathcal{M}_n$ makes polynomially many calls to the oracle $\mathcal{PER}_{\mathcal{M}_n,\mu}$ with

$$\mu = \left\lceil \vartheta \left(\frac{\log p \log\log\log p}{\log\log p} \right)^{1/2} \right\rceil$$

and evaluates $\operatorname{per} X$ correctly with probability at least $1 + O(2^{-\mu/2})$.

PROOF. Given $X \in \mathcal{M}_n$, let us select $\lambda \in \mathbb{F}_p^*$ uniformly at random, compute X_λ and use the oracle $\mathcal{PER}_{\mathcal{M}_n,k}$ with input X_λ to evaluate

$$\mathrm{APPROX}_{\eta,p}(\operatorname{per} X_\lambda) = \mathrm{APPROX}_{\eta,p}(\lambda^n \operatorname{per} X).$$

By Lemma 2.4, repeating this procedure $O(\mu^{-1}(\log p)^{3+\varepsilon})$ times, we obtain the desired result. □

COROLLARY 3.2. *For any constants $\delta > 0$ and $A \ge 1$ there is a constant $C > 0$ depending only on δ and A such that the following statement holds true. If for some constant $\vartheta > 0$ there is a probabilistic polynomial time algorithm*

which, with exponentially small failure probability, achieves an approximation of $\mathrm{APPROX}_{\eta,p}(\operatorname{per} S)$ *with*

$$\eta = \left\lceil \vartheta \left(\frac{\log p \log\log\log p}{\log\log p} \right)^{1/2} \right\rceil$$

to the permanent of an $n \times n$ *symmetric matrix* S *over* $\mathbb{F}_p$ *for at least* $\delta Q \log^{-A} Q$ *primes* p *of the interval* $[Q, 2Q]$ *with some* Q *satisfying*

$$Q \ge Cn \log^A n \quad \textit{and} \quad \log Q = n^{O(1)}$$

and returning an error message for other primes, then $\mathbf{P}^{\sharp\mathbf{P}} = \mathbf{BPP}$.

Proof. We show that the above algorithm can be transformed to a probabilistic algorithm to compute the permanent of symmetric $n \times n$ binary matrices (that is, matrices with $0, 1$-entries). The latter problem is $\sharp\mathbf{P}$-complete, as follows from the easy reduction, mapping any arbitrary $n \times n$ binary matrix X to the $2n \times 2n$ symmetric matrix

$$S = \begin{bmatrix} 0 & X \\ X^T & 0 \end{bmatrix},$$

whose permanent is the square of the permanent of X, $\operatorname{per} S = (\operatorname{per} X)^2$.

Given any symmetric $n \times n$ binary matrix S, it is obvious that $0 \le \operatorname{per} S \le n!$.

Let us set

$$\ell = \left\lceil \frac{n \log n}{\log Q} \right\rceil$$

and let us choose ℓ primes in the interval $[Q, 2Q]$ for which there exists an algorithm $\mathcal{A}$ in the condition of the theorem.

We can build a set $\mathcal{L}$ of such primes iteratively by just selecting random integers in the interval $[Q, 2Q]$ and testing them for primality by the algorithm of Agrawal *et al.* (2002) (one can also use one of any of the polynomial time probabilistic primality tests, see Bach & Shallit 1996; Crandall & Pomerance 2001), and then whether they are already on the list and whether they are such that the algorithm $\mathcal{A}$ works for them.

We remark that because of the conditions on Q we have $\ell < 0.5\delta Q \log^{-A} Q$ for an appropriately chosen constant $C > 0$. Thus at each of the ℓ steps we have at least $\delta Q \log^{-A} Q - \ell > 0.5\delta Q \log^{-A} Q$ "suitable" primes. Hence each step takes only $(\log Q)^{O(1)} = n^{O(1)}$ binary operations.

We also see that each $p \in \mathcal{L}$ satisfies $n = O(Q \log^{-1} Q) = O(p \log^{-1} p)$, so Theorem 3.1 applies. Therefore the approximation algorithm $\mathcal{A}$ can be transformed into a probabilistic algorithm to compute $\lfloor \operatorname{per} S \rfloor_p$ for each $p \in \mathcal{L}$. Thus using the Chinese Remainder Theorem, in time polynomial in ℓ and $\log Q$, and thus in n, we can compute $\operatorname{per} S$ modulo

$$\prod_{p \in \mathcal{L}} p > Q^\ell \geq n! \geq \operatorname{per} S$$

and hence find the actual value of $\operatorname{per} S$.

The assertion now follows by applying a result by Cai *et al.* (1999), who have proved that the existence of a probabilistic algorithm correctly computing the permanent of a matrix for any inverse polynomial fraction of all inputs implies the unlikely collapse $\mathbf{P}^{\sharp\mathbf{P}} = \mathbf{BPP}$. □

4. Remarks

Note that, although the traditional measure for the size of an $n \times n$ matrix X over $\mathbb{F}_p$ is about $n^2 \log p$, some matrices admit a much shorter description. For example, an *s-sparse circulant matrix*, with only s non-zero entries per row, can be described by only $O(s \log np)$ bits. For such matrices it is enough to specify s pairs (m_ν, x_ν), $\nu = 1, \ldots, s$, where m_ν, $1 \leq m_\nu \leq n$, is the position of the νth non-zero entry $x_\nu \in \mathbb{F}_p$ in the first row. In this case, provided that the oracle $\mathcal{PER}_{\mathcal{M}_{n,k}}$ accepts such a description, the algorithm of Theorem 3.1 becomes polynomial in $s \log np$.

Using this setting, one can consider an analogue of Theorem 3.1 for the determinant as well. Indeed, although the determinant is an "easy" function for dense matrices, it is not clear whether for s-sparse circulants it can be computed in time polynomial in $s \log np$. Moreover, an analogue of Theorem 3.1 and its modification for matrices with "short description" holds for the much wider class of matrix functions known as *immanants*, whose complexity has been studied, for example, by Bürgisser (2000a,b). Immanants are expressions of the form

$$\operatorname{imm}_\chi X = \sum_{\sigma \in \mathcal{S}_n} \chi(\sigma) \prod_{i=1}^{n} x_{i,\sigma(i)},$$

where $\chi : \mathcal{S}_n \to \mathbb{C}$ is an irreducible character of the symmetric group $\mathcal{S}_n$. The trivial character $\chi(\sigma) = 1$ corresponds to the permanent, the alternating character $\chi(\sigma) = \operatorname{sign} \sigma$ corresponds to the determinant. Because for any character χ we have $\operatorname{imm}_\chi X_\lambda = \lambda^n \operatorname{imm}_\chi X$, the result of Theorem 3.1 holds for $\operatorname{imm}_\chi X$ instead of just $\operatorname{per} X$ without any other changes.

Our approach can also be used to prove the hardness of modular approximation of several other polynomial functions, such as cycle format polynomials and the factor polynomials (see Section 3.3 of Bürgisser 2000a). In fact, our technique applies without any changes to any function F on matrices for which $F(X_\lambda) = \lambda^m F(X)$ with some integer $m \geq 1$ (depending only on the dimension n of X).

Acknowledgements

The authors would like to thank Jin-Yi Cai for his interest and for helpful comments. The second author was supported in part by ARC grant DP0211459. The third author was supported in part by DSTA grant R-394-000-011-422.

References

M. Agrawal, N. Kayal & N. Saxena (2002). PRIMES is in **P**. Preprint, 9 pp.

E. Bach & J. Shallit (1996). *Algorithmic Number Theory*. MIT Press.

D. Boneh & R. Venkatesan (1996). Hardness of computing the most significant bits of secret keys in Diffie–Hellman and related schemes. In Lecture Notes in Comput. Sci. 1109, Springer, Berlin, 129–142.

D. Boneh & R. Venkatesan (1997). Rounding in lattices and its cryptographic applications. In *Proc. 8th Annual ACM-SIAM Sympos. on Discrete Algorithms*, ACM, New York, 675–681.

P. Bürgisser (2000a). *Completeness and Reduction in Algebraic Complexity Theory*. Springer, Berlin.

P. Bürgisser (2000b). The computational complexity of immanants. *SIAM J. Comput.* **30**, 1023–1040.

J.-Y. Cai, A. Pavan & D. Sivakumar (1999). On the hardness of permanent. In Lecture Notes in Comput. Sci. 1563, Springer, Berlin, 90–99.

T. Cochrane, C. Pinner & J. Rosenhouse (2003). Bounds on exponential sums and the polynomial Waring's problem mod p. *J. London Math. Soc.* **67**, 319–336.

R. Crandall & C. Pomerance (2001). *Prime Numbers: A Computational Perspective*. Springer, Berlin.

U. Feige & C. Lund (1996/1997). On the hardness of computing the permanent of random matrices. *Comput. Complexity* **6**, 101–132.

M. I. González Vasco & I. E. Shparlinski (2001). On the security of Diffie–Hellman bits. In *Proc. Workshop on Cryptography and Computational Number Theory* (Singapore, 1999), Birkhäuser, 257–268.

M. I. González Vasco & I. E. Shparlinski (2002). Security of the most significant bits of the Shamir message passing scheme. *Math. Comp.* **71**, 333–342.

M. Jerrum, A. Sinclair & E. Vigoda (2000). A polynomial-time algorithm for the permanent of a matrix with non-negative entries. *Electronic Colloq. on Comput. Compl.* **TR2000-079**, 1–22.

S. V. Konyagin (1992). On estimates of Gaussian sums and the Waring problem modulo a prime. *Trudy Mat. Inst. Akad. Nauk SSSR* **198**, 111–124 (in Russian); translation in *Proc. Steklov Inst. Math.* **1** (1994), 105–117.

S. V. Konyagin (2002). Bounds of exponential sums over subgroups and Gauss sums. Preprint (in Russian), 25 pp.

S. V. Konyagin & I. Shparlinski (1999). *Character Sums with Exponential Functions and their Applications*. Cambridge Univ. Press, Cambridge.

W.-C. W. Li, M. Näslund & I. E. Shparlinski (2002). The hidden number problem with the trace and bit security of XTR and LUC. In Lecture Notes in Comput. Sci. 2442, Springer, Berlin, 433–448.

R. Lidl & H. Niederreiter (1997). *Finite Fields*. Cambridge Univ. Press, Cambridge.

N. Linial, A. Samorodinski & A. Wigderson (1998). A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. In *Proc. 30th ACM Sympos. on Theory of Comp.*, 644–652.

E. El Mahassni, P. Q. Nguyen & I. E. Shparlinski (2001). The insecurity of Nyberg–Rueppel and other DSA-like signature schemes with partially known nonces. In Lecture Notes in Comput. Sci. 2146, Springer, Berlin, 97–109.

H. Minc (1982). *Permanents*. Encyclopedia Math. Appl. 6, Addison-Wesley.

P. Q. Nguyen & I. E. Shparlinski (2002). The insecurity of the Digital Signature Algorithm with partially known nonces. *J. Cryptology* **15**, 151–176.

P. Q. Nguyen & I. E. Shparlinski (2003). The insecurity of the elliptic curve Digital Signature Algorithm with partially known nonces. *Des. Codes Cryptogr.* **30**, 207–217.

V. Pan (2001). *Structured Matrices and Polynomials*. Birkhäuser and Springer.

I. E. Shparlinski (2001a). On the generalised hidden number problem and bit security of XTR. In Lecture Notes in Comput. Sci. 2227, Springer, Berlin, 268–277.

I. E. Shparlinski (2001b). Sparse polynomial approximation in finite fields. In *Proc. 33rd ACM Sympos. on Theory of Comput.* (Crete), 209–215.

I. E. Shparlinski (2002). Security of most significant bits of g^{x^2}. *Inform. Proc. Lett.* **83**, 109–113.

I. E. Shparlinski & A. Winterhof (2003). Hidden number problem in small subgroups. Cryptology ePrint Arch. Report 2003/049, 1–12.

L. G. Valiant (1979a). Completeness classes in algebra. In *Proc 11th ACM Sympos. on the Theory of Comput.*, 249–261.

L. G. Valiant (1979b). The complexity of computing the permanent. *Theoret. Comput. Sci.* **8**, 189–201.

L. G. Valiant (1979c). The complexity of enumeration and reliability problems. *SIAM J. Comput.* **8**, 410–421.

Manuscript received 4 August 2002

Bruno Codenotti
Department of Computer Science
University of Iowa
14 MacLean Hall
Iowa City, IA 52240, U.S.A.
(on leave from IIT-CNR, Pisa, Italy)

Other address of Bruno Codenotti:
Department of Computer Science
University of Chicago
Ryerson Hall, 1100E 58th Street
Chicago, IL 60637, U.S.A.
codenott@cs.uchicago.edu

Igor E. Shparlinski
Department of Computing
Macquarie University
Sydney, NSW 2109, Australia
igor@ics.mq.edu.au

Arne Winterhof
Temasek Laboratories
National University of Singapore
10 Kent Ridge Crescent
Singapore 119260, Singapore
tslwa@nus.edu.sg