A Note on Secure Computation of the Moore-Penrose Pseudoinverse and Its Application to Secure Linear Algebra

Ronald Cramer[1,2], Eike Kiltz[1*], and Carles Padró[3**]

[1] Cryptology and Information Security Research Theme
CWI Amsterdam
The Netherlands
{cramer,kiltz}@cwi.nl
[2] Mathematical Institute
Leiden University
The Netherlands
[3] Department of Applied Mathematics IV
Universitat Politècnica de Catalunya
Barcelona, Spain
cpadro@ma4.upc.edu

Abstract. This work deals with the communication complexity of secure multi-party protocols for linear algebra problems. In our model, complexity is measured in terms of the number of secure multiplications required and protocols terminate within a constant number of rounds of communication.

Previous work by Cramer and Damgård proposes secure protocols for solving systems $Ax = b$ of m linear equations in n variables over a finite field, with $m \leq n$. The complexity of those protocols is n^5.

We show a new upper bound of $m^4 + n^2m$ secure multiplications for this problem, which is clearly asymptotically smaller. Our main point, however, is that the advantage can be substantial *in case m is much smaller than n*. Indeed, if $m = \sqrt{n}$, for example, the complexity goes down from n^5 to $n^{2.5}$.

Our secure protocols rely on some recent advances concerning the computation of the Moore-Penrose pseudo-inverse of matrices over fields of positive characteristic. These computations are based on the evaluation of a certain characteristic polynomial, in combination with variations on a well-known technique due to Mulmuley that helps to control the effects of non-zero characteristic. We also introduce a new method for

[*] Supported by the research program Sentinels (http://www.sentinels.nl). Sentinels is being financed by Technology Foundation STW, the Netherlands Organization for Scientific Research (NWO), and the Dutch Ministry of Economic Affairs.

[**] Supported by the Spanish Ministry of Education and Science under projects TIC2003-00866 and TSI2006-02731. This work was done while this author was in a sabbatical stay at CWI, Amsterdam. This stay was funded by the *Secretaría de Estado de Educación y Universidades* of the Spanish Ministry of Education and Science.

secure polynomial evaluation that exploits properties of Chebychev polynomials, as well as a new secure protocol for computing the characteristic polynomial of a matrix based on Leverrier's lemma that exploits this new method.

1 Introduction

This paper deals with *secure multi-party computation* (MPC), that is, with the scenario in which n players want to compute an agreed function of their secret inputs in such a way that the correct result is obtained but no additional information about the inputs is released. These requirements should be achieved even in the presence of an *adversary* who is able to corrupt some players. The power of a *passive* adversary is limited to see all internal data of the corrupted adversaries, while an *active* one can control their behavior.

Multi-party computation protocols can be classified according to which model of communication is considered. In the *cryptographic* model, first considered in [21, 11], the adversary can see all messages in the network and the security must rely on some computational assumption. Unconditional security can be achieved if the existence of a private channel between every pair of participants is assumed. This is the *information-theoretic* model introduced in [5, 6]. It is well known that in both models any functionality can be securely evaluated — if evaluating the functionality is efficient, so is the secret multi-party protocol. However, generic solutions may need polynomial many rounds of communication between the participating players, whereas in practise one wants the round complexity to be as small as possible, preferably constant.

For conditionally secure multi-party protocols in the cryptographic model, every probabilistic polynomial-time functionality can be efficiently and privately evaluated in a constant number of communication rounds [22, 3]. The situation is completely different for unconditionally secure multi-party protocols in the information-theoretic model. Up to now it is not known yet which class of functions can be efficiently computed in a constant number of rounds. Some progress in the direction of solving that question was made in [1, 9, 12, 13, 2] but, for instance it is not even known if all functions in basic classes like NC can be securely evaluated in constant rounds.

For specific functions of interest from linear algebra, Cramer and Damgård [7] proposed constant round multi-party computation protocols in the information-theoretic model. Among their considered functions are the determinant, the characteristic polynomial, the rank of a matrix, and solving a linear system of equations. The advantage with the approach from [7] is that all protocols could be tailor-made to the nature of the specific problem and, in contrast to the generic solutions, did not have to rely on circuit-based secure gate evaluation techniques.

1.1 Our results

This work deals with the communication complexity of secure multi-party protocols for linear algebra problems. In our model, complexity is measured in terms of the number of secure multiplications required and protocols terminate within a constant number of rounds of communication.

Assuming a model in which constant round protocols for basic arithmetic operations are given as usual, previous work by Cramer and Damgård proposes secure protocols for solving systems $Ax = b$ of m linear equations in n variables over a finite field, with $m \leq n$. The complexity of those protocols is n^5. Since a solution in [7] could only be obtained for square matrices the general case of non-square matrices had to be reduced to solving linear systems for the case of an $n \times n$ matrix which is potentially huge compared to the original $m \times n$ matrix A. The protocol for the latter problem basically reduces to computing n times the characterstic polynomial which is shown in [7] to be securely computable in constant rounds and with (roughly) n^4 complexity (n^4 calls to the multiplication protocol). Therefore the overall complexity of the proposed protocol to solve the linear system $Ax = b$ is n^5.

We show a new upper bound of $m^4 + n^2m$ secure multiplications for this problem, which is clearly asymptotically smaller. Our main point, however, is that the advantage can be substantial *in case m is much smaller than n*. Indeed, if $m = \sqrt{n}$, for example, the complexity goes down from n^5 to $n^{2.5}$.

As a concrete motivating application we consider the secure multi-party variant of the travelling salesman problem from combinatorial optimization. Given a number of t cities and the costs of travelling from any city to any other city, what is the cheapest round-trip route that visits each city exactly once and then returns to the starting city?[4] In a multi-party scenario the participating players may want to keep the travelling cost between two cities belonging to "their territory" secret such that only the concrete round-trip is revealed to everybody. It is well known [20, Vol 2, Chap. 58.4] that this problem can be reduced using integer linear programming to simultaneously solving two systems of linear equations, each of size $m \times n$, where $n = 2t \cdot 2^m \approx 2^m$ and $m \leq t^2$ is the number of edges in the graph representing the cost-matrix between the t cities. Hence, in this (admittingly extreme) example complexity of our protocol is $\approx (2^m)^2$, compared to the $(2^m)^5$ protocol from [7].

Our secure protocols rely on some recent advances concerning the computation of the Moore-Penrose pseudo-inverse of matrices over fields of positive characteristic. These computations are based on the evaluation of a certain characteristic polynomial, in combination with variations on a well-known technique due to Mulmuley that helps to control the effects of non-zero characteristic. We also introduce a new method for secure polynomial evaluation that exploits properties of Chebychev polynomials, as well as a new secure protocol for computing the characteristic polynomial of a matrix based on Leverrier's lemma that exploits this new method. These techniques may be of separate interest, and are central to our claimed improvements.

Below we give a more detailed overview of the techniques used. If A is an $n \times m$ matrix over a field $\mathbb{K}$, a *pseudoinverse* of A is any $m \times n$ matrix B such

[4] Since the travelling salesman problem is known to be NP-complete, for the purpose of this motivating example one may think of a small amount of cities t.

that $ABA = A$ and $BAB = B$. Note that in case A is a non-singular square matrix then $B = A^{-1}$. A linear system of equations $Ax = b$ can be easily solved if a pseudoinverse of A is given. First of all, the system has a solution if and only if $ABb = b$. In this case, $x_0 = Bb$ is a particular solution and, since the columns of the matrix $I_m - BA$ span $\ker A$, the general solution of the system is obtained.

Our secure MPC protocol to solve linear systems of equations applies the results and techniques from [10] about using of the Moore-Penrose pseudoinverse for solving linear systems of equations over arbitrary fields. Specifically, there is a polynomial that, evaluated on the Gram matrix $G = A^\top A$, (where $A^\top$ denotes the transpose of A) makes it possible to efficiently compute in MPC the Moore-Penrose pseudoinverse of A. The polynomial in turn is derived from the characteristic polynomial of $G = A^\top A$. Here our secure polynomial evaluation protocol based on Chebyshev polynomials can be used to perform the secure evaluation.

Nevertheless, the Moore-Penrose pseudoinverse of a matrix A exists if and only if the subspaces $\ker A$ and $\operatorname{Im} A$ have trivial intersection with their orthogonals, and unfortunately this may not be the case if the field has positive characteristic. This problem is solved by using some techniques from [10], based on results by Mulmuley [17]. Namely, there exists a random invertible diagonal matrix that, with high probability, transforms the matrix A into a matrix A' having the required properties on the subspaces $\ker A'$ and $\operatorname{Im} A'$.

Computing the Moore-Penrose pseudoinverse in particular involves secure evaluation of a public (or secret) polynomial in a secret field element (or a secret matrix). Motivated by this and maybe of independent interest, we present a constant round MPC protocol for the above task. If the field element (or the matrix) is guaranteed to be invertible this can be done using the well-known technique of unbounded fan-in multiplication [1]. In the general case of non-zero field elements a generic framework from [1] can be applied. However, the latter technique boosts the complexity of the resulting protocol from linear to quadratic in the degree d of the polynomial. One the other hand, if one admits some small probability of information leakage then the protocol can be made linear in d using certain randomization techniques.

We present an alternative protocol for the same task which is perfectly secure and has complexity linear in the degree d. The basic idea is explained in the following. Consider a matrix $M(x)$ whose entries are polynomials over a finite field $\mathbb{F}$ and such that $M(\alpha)$ is invertible for every $\alpha \in \mathbb{F}$. Specifically, we present a 2×2 matrix $M(x)$ such that in the top-left entry of $M(2x)M^{i-1}(x)$ we have the ith *Chebyshev polynomial* $T_i(x)$. Since the first $d+1$ Chebyshev polynomials $\{T_i(x)\}_{0 \leq i \leq d}$ form a basis of the polynomials of degree at most d, every polynomial of degree d is a linear combination the Chebyshev polynomials. Therefore we can securely compute, even if α may be zero, $F(\alpha)$ for every polynomial $F(x)$ with degree at most d by using the unbounded fan-in multiplication protocol to compute the needed powers of the matrix $M(\alpha)$.

1.2 Related Work

Nissim and Weinreb [19] also considered the problem of securely solving a set of linear equations in the *computational two-party model*, focusing on low (nearly optimal) communication complexity. Their protocol needs $O(n^{0.275})$ rounds of communication which was later improved to $O(\log n)$ [15].

2 Preliminaries

2.1 The Model

We assume that n parties are connected by perfectly secure channels in a synchronous network. Let $\mathbb{F}_p$ denote the finite field with p elements where p is a prime power. We will assume throughout that p is large enough because our protocols can only guarantee security with a probability $1 - O(n^2/p)$, where n is the maximum number of rows or columns in the matrices appearing in the linear systems of equations.

By $[a]$ we denote a secret sharing of $a \in \mathbb{F}_p$ over $\mathbb{F}_p$. We assume that the secret-sharing scheme allows to compute a sharing $[a+b]$ from $[a]$ and $[b]$ without communication, and that it allows to compute $[ab]$ from $a \in \mathbb{F}_p$ and $[b]$ without communication; we write

$$[a+b] \leftarrow [a] + [b] \text{ and } [ab] \leftarrow a[b]$$

for these operations. The secret-sharing scheme should of course also allow to take a sharing $[c]$ and reveal the value $c \in \mathbb{F}_p$ to all parties; We write $c \leftarrow \textsc{reveal}([c])$.

We also assume that the secret sharing scheme allows to compute a sharing $[ab]$ from $[a]$ and $[b]$ with unconditional security. We denote the multiplication protocol by MULT, and write

$$[ab] \leftarrow \textsc{mult}([a], [b]) \ .$$

We will express the protocols' round complexities as the number of sequential rounds of MULT invocations — and their communication complexities as the overall number of MULT invocations. I.e., if we first run a copies of MULT in parallel and then run b copies of MULT in parallel, then we say that we have round complexity 2 and communication complexity $a + b$. Note that standard linear (verifiable) secret-sharing schemes have efficient constant-rounds protocols for multiplication.

For a matrix $A \in \mathbb{F}_p^{n \times m} = (A_{ij})_{1 \le i \le n, 1 \le j \le m}$ we will use $[A] = ([A_{ij}])_{1 \le i \le n, 1 \le j \le m}$ for a sharing of a matrix. For multiplication of two matrices $A \in \mathbb{F}_p^{n \times k}$, $B \in \mathbb{F}_p^{k \times m}$ of matching dimensions we simply write $[C] \leftarrow \textsc{mult}([A], [B])$, where $C = AB \in \mathbb{F}_p^{n \times m}$. Matrix multiplication has to be understood componentwise and can be carried out in one round and nmk parallel invocations of the multiplication protocol.

For our protocols to be actively secure, the secret sharing scheme and the multiplication protocol should be actively secure. This in particular means that the adversary structure must be $Q2$. By the adversary structure we mean the set $\mathcal{A}$ of subsets $C \subset \{1, \ldots, n\}$ which the adversary might corrupt; It is $Q2$ if it holds for all $C \in \mathcal{A}$ that $\{1, \ldots, n\} \setminus C \notin \mathcal{A}$.

2.2 Some known techniques

The following known techniques will be of importance later on.

Random Elements. The parties can share a uniformly random, unknown field element. We write $[a] \leftarrow \text{RAN}()$. This is done by letting each party P_i deal a sharing $[a_i]$ of a uniformly random $a_i \in \mathbb{F}_p$. Then the parties compute the sharing $[a] = \sum_{i=1}^{n} [a_i]$. The communication complexity of this is given by n dealings, which we assume is upper bounded by the complexity of one invocation of the multiplication protocol.

If passive security is considered, this is trivially secure. If active security is considered and some party refuses to contribute with a dealing, the sum is just taken over the contributing parties. This means that the sum is at least taken over a_i for $i \in H$, where $H = \{1, \ldots, n\} \setminus C$ for some $C \in \mathcal{A}$. Since $\mathcal{A}$ is $Q2$ it follows that $H \notin \mathcal{A}$. So, at least one honest party will contribute to the sum, implying randomness and privacy of the sum.

Random Invertible Elements. Using [1] the parties can share a uniformly random, unknown, invertible field element along with a sharing of its inverse. We write $([a], [a^{-1}]) \leftarrow \text{RAN}^*()$ and it proceeds as follows: $[a] \leftarrow \text{RAN}()$ and $[b] \leftarrow \text{RAN}()$. $[c] = \text{MULT}([a], [b])$. $c \leftarrow \text{REVEAL}([c])$. If $c \notin \mathbb{F}_p^*$, then abort. Otherwise, proceed as follows: $[a^{-1}] \leftarrow c^{-1}[b]$. and output $([a], [a^{-1}])$.

The correctness is straightforward. As for privacy, if $c \in \mathbb{F}_p^*$, then (a, b) is a uniformly random element from $\mathbb{F}_p^* \times \mathbb{F}_p^*$ for which $ab = c$, and thus a is a uniformly random element in $\mathbb{F}_p^*$. If $c \notin \mathbb{F}_p^*$, then the algorithm aborts. This happens with probability less than $2/p$. The complexity is (at most) 2 rounds and 3 invocations of MULT.

Unbounded Fan-In Multiplication. Using the technique from [1] it is possible to do unbounded fan-in multiplication in constant rounds. For the special case where we compute all "prefix products" $\prod_{i=1}^{m} a_i$ $(m = 1, \ldots, \ell)$, we write

$$([a_1], \ldots, [(a_1 a_2 \cdots a_\ell)]) \leftarrow \text{MULT}^*([a_1], \ldots, [a_\ell]) \ .$$

In the following, we only need the case where we have inputs $[a_1], \ldots, [a_\ell]$, where $a_i \in \mathbb{F}_p^*$. For $1 \leq i_0 \leq i_1 \leq \ell$, let $a_{i_0,i_1} = \prod_{i=i_0}^{i_1} a_i$. We are often only interested in computing $a_{1,\ell}$, but the method allows to compute any other a_{i_0,i_1} at the cost of one extra multiplication. For the complexity analysis, let A denote the number of a_{i_0,i_1}'s which we want to compute.

First run RAN* ℓ+1 times in parallel, to generate $[b_0 \in_R \mathbb{F}_p^*], [b_1 \in_R \mathbb{F}_p^*], \ldots, [b_\ell \in_R \mathbb{F}_p^*]$, along with $[b_0^{-1}], [b_1^{-1}], \ldots, [b_\ell^{-1}]$, using 2 rounds and $3(\ell + 1)$ invocations of MULT. For simplicity we will use the estimate of 3ℓ invocations. Then for $i = 1, \ldots, \ell$ compute and reveal $[d_i] = \text{MULT}([b_{i-1}], [a_i], [b_i^{-1}])$, using 2 rounds and 2ℓ invocations of MULT. Now we have that $d_{i_0,i_1} = \prod_{i=i_0}^{i_1} d_i = b_{i_0-1}(\prod_{i=i_0}^{i_1} a_i)b_{i_1}^{-1} = b_{i_0-1}a_{i_0,i_1}b_{i_1}^{-1}$, so we can compute $[a_{i_0,i_1}] = d_{i_0,i_1}\text{MULT}([b_{i_0-1}^{-1}], [b_{i_1}])$, using 1 round and A invocations of MULT. The overall complexity is 5 rounds and $O(\ell+a)$ invocations of MULT.

The same concept generalizes to unbounded fan-in multiplication of matrices. Let shares $[M_i]$ of matrices $M_i \in \mathbb{F}_p^{m\times m}$ be given. Again we write

$$([M_1], \ldots, [(M_1 M_2 \cdots M_\ell)]) \leftarrow \text{MULT}^*([M_1], \ldots, [M_\ell]) \ .$$

for the special case where we compute all "prefix matrix products" $\prod_{i=1}^k M_i$ ($k = 1, \ldots, \ell$). The above protocol generalizes to the matrix case, where a random invertible field element now translates to a random invertible matrix. Random invertible matrices are created using the same the method as generating a shared random invertible field element.

Equality. We define the equality function $\delta : \mathbb{F}_p \to \mathbb{F}_p$ as $\delta(x) = 0$ if $x = 0$ and $\delta(x) = 1$ otherwise. Given a shared value $[x]$, there exists a protocol [8, 18] that computes, in a constant number of rounds and using $O(\log p)$ calls to the multiplication protocol MULT, shares $[\delta(x)]$. We write $[y] \leftarrow \text{EQ}([x])$.

3 Secure Polynomial Evaluation

In this section we are interested in the natural problem of secure polynomial evaluation: the players hold a public (shared) polynomial F of maximal public degree d and a shared field element x. The goal is to securely evaluate F in x, i.e. to compute shares $[F(x)]$.

Based on known techniques [1, 4] the latter shares can be computed in constant rounds and quadratic complexity, i.e. the protocol makes $O(d^2)$ calls to the multiplication protocol.

Surprisingly, as we will show in this section, Chebyshev polynomials of the first and the second kind can be used as a mathematical tool to bring the complexity of the above problem down to linear. We will first consider the simpler case where the polynomial $F(X)$ is publicly known and later reduce the case of a shared polynomial to the latter one.

3.1 Known Solution

First we present a naïve protocol based on known techniques with linear complexity. Unfortunately, as we will see, the protocol leaks information for the interesting case when the polynomial gets evaluated at zero.

The protocol is given a shared value $[x]$, where $x \in \mathbb{F}_p^*$ and a public polynomial $F(X) = \sum_{i=0}^{d} a_i X^i$. The protocol's task is to compute shares $[F(x)]$. First, it computes $([x], [x^2], \dots, [x^d]) \leftarrow \text{MULT}_p^*([x], \dots, [x])$ and then the share $[F(x)]$ can be computed without interaction as $[F(x)] \leftarrow a_0 + a_1[x] + \sum_{i=2}^{d} a_i[x^i]$. The complexity is constant rounds and $6d = O(d)$ invocations of the multiplication protocol MULT. Privacy follows since we assumed $x \in \mathbb{F}_p^*$ and hence we can apply the protocol MULT* securely. On the other hand, if $x \notin \mathbb{F}_p^*$ then this fact will leak throughout the application of protocol MULT*.

As already done in [1], using a technique from [4], the general case (where the input may be equal to zero) can be reduced to unbounded fan-in multiplication of non-singular 3×3 matrices as we will sketch now. Later we will give an alternative protocol for the same task with improved running time. The main result from [4] states that every algebraic formula Φ of depth l can be expressed as the product of $O(4^l)$ non-singular 3×3 matrices over $\mathbb{F}_p$ (in the sense that the value $\Phi(x)$ can be read from the right top corner of the matrix product). Since any polynomial $F(X)$ of degree d can be expressed as an algebraic formula of depth $\log d$, $F(X)$ can be expressed as the product of $O(d^2)$ such non-singular 3×3 matrices. The appearing matrices are either one of five (known) constant matrices or are the identity matrix with x in the right upper corner. Using an efficient constant round protocol for multiplying non-singular constant size matrices we imply that there there exists a protocol that privately computes shares $[F(x)]$, where x may equal to zero. The protocol runs in a constant number of rounds and $O(d^2)$ invocations of MULT.

If we admit some small probability of information leakage we can get a $O(d)$ protocol for the same task as follows. First choose a random field element $[c] \leftarrow$ RAN() and compute the share $[x+c]$. Then compute $([x+c], [(x+c)^2], \dots, [(x+c)^d]) \leftarrow \text{MULT}_p^*([x+c], \dots, [x+c])$. This step is secure as long as $x + c \neq 0$ which happens with probability $1 - 1/p$ (over all coin tosses of the RAN protocol). Then open the share $[c]$ to obtain the field element c. Since the polynomials $(x+c)^i$ $(0 \leq i \leq d)$ form a basis for all polynomials of degree at most d we can compute $[F(x)]$ without interaction using $F(x) = \sum_{i=0}^{d} \lambda_i (x+c)^i$, where the coefficients λ_i can be computed by the players using only public information (including the value c). The protocol runs in a constant number of rounds and $O(d)$ invocations of MULT. However, it leaks information about x with probability $1/p$. In the remainder of this section we will develop a perfectly secure protocol in $O(d)$ invocations of MULT.

3.2 Chebyshev Polynomials

We use Chebyshev polynomials of the first kind which satisfy the linear recurrence

$$T_d(x) = 2xT_{d-1}(x) - T_{d-2}(x), \quad d \geq 2$$

with starting values $T_0(x) = 1$ and $T_1(x) = x$, and Chebyshev polynomials of the second kind

$$U_d(x) = 2xU_{d-1}(x) - U_{d-2}(x), \quad d \geq 2$$

with starting values $U_0(x) = 1$ and $U_1(x) = 2x$. For notational convenience we also set $T_d(x) = U_d(x) = 0$ for any $d < 0$. It is well known that the Chebyshev polynomials $T_i(x)$, $0 \leq i \leq d$ form a basis for all polynomials of degree at most d. I.e., there exist coefficients $\lambda_i \in \mathbb{F}_p$ such that every polynomial F of degree at most d given in its monomial representation $F(x) = \sum_{i=0}^{d} a_i x^i$ can be expressed in the Chebyshev basis as

$$F(x) = \sum_{i=0}^{d} \lambda_i T_i(x) \; . \tag{1}$$

The coefficients λ_i only depend on the polynomial F, but not on x. (All λ_i's can be computed from the a_i's in $O(d^2 \log^2 p)$ bit operations using, for instance, the recursive formulas from [16].)

For $x \in \mathbb{F}_p$ define the 2×2 matrix $M(x)$ over $\mathbb{F}_p$ as

$$M(x) = \begin{pmatrix} x & -1 \\ 1 & 0 \end{pmatrix},$$

and note that since $\det(M(x)) = 1$, the matrix $M(x)$ is always non-singular.

Claim. The following identity holds for any integer $d \geq 1$:

$$M(x)M^{d-1}(2x) = \begin{pmatrix} T_d(x) & -T_{d-1}(x) \\ U_{d-1}(x) & -U_{d-2}(x) \end{pmatrix}. \tag{2}$$

We quickly prove the claim by induction over d. For $d = 1$ (2) is correct by definition. Now assume (2) holds for an integer $d \geq 1$. Then we have

$$\begin{aligned} M(x)M^{d}(2x) = M(x)M^{d-1}(2x) \cdot M(2x) &= \begin{pmatrix} T_d(x) & -T_{d-1}(x) \\ U_{d-1}(x) & -U_{d-2}(x) \end{pmatrix} \cdot \begin{pmatrix} 2x & -1 \\ 1 & 0 \end{pmatrix}, \\ &= \begin{pmatrix} 2T_d(x) \cdot x - T_{d-1} & -T_d(x) \\ 2U_{d-1}(x) \cdot x - U_{d-2}(x) & -U_{d-1}(x) \end{pmatrix}. \end{aligned}$$

This shows (2) for $d + 1$.

3.3 Perfectly secure polynomial evaluation of a shared field element

We now come to our improvement over the protocols from Section 3.1. We design an alternative protocol to evaluate a polynomial in a share with running time linear in the degree d (instead of quadratic). The protocol does not leak any information about the shared secret x. Using the results on the Chebychev polynomials from the last section a protocol to securely evaluate a given public polynomial $F \in \mathbb{F}_p[X]$ of degree d in a share $[x]$ is as follows: The players first locally create matrix-shares $[M(x)]$ and $[M(2x)]$ from the share $[x]$. Then they compute (component-wise and in parallel) matrix-shares $[M(x)M^{i-1}(2x)]$ for $1 \leq i \leq d$ by

$$([M(x)M(2x)], \ldots, [M(x)M^{d-1}(2x)]) \leftarrow \textsc{mult}^*([M(x)], [M(2x)], \ldots, [M(2x)]).$$

Security is granted since $M(x)$ and $M(2x)$ are both non-singular. By Eq. (2), the share $[T_i(x)]$ can now be read in the upper left corner of the resulting matrices.

Once we are given shares of the Chebychev basis $\{T_i(x)\}_{1\leq i\leq d}$ we can evaluate any given polynomial F of maximal degree d without interaction by computing $[F(x)] = \sum_{i=0}^{d} \lambda_i [T_i(x)]$. Here λ_i are the coefficients from (1) that are computed by each player in a precomputation phase. This leads to the following:

Proposition 1. *Let a set of ℓ public polynomials $F_i \in \mathbb{F}_p[X]$ be given, all of maximal degree d. There exists a multi-party protocol that, given shares $[x]$ (for any $x \in \mathbb{F}_p$, possibly $x = 0$), computes all shares $([F_1(x)], \ldots, [F_\ell(x)])$. The protocol runs in constant rounds and $O(d)$ applications of the multiplication protocol.*

It is easy to see that the given techniques can be extended to evaluate a shared value x in a *shared* polynomial F, i.e. the shared F is given by shares of its coefficients $[a_i]$, $1 \leq i \leq d$. The protocol first computes shares $[x^i]$ for $1 \leq i \leq d$ with the methods from Proposition 1 (here the ith polynomial $F_i(X)$ is defined as $F_i(X) = X^i$). Then the polynomial F can be securely evaluated in x by first computing all shares $[a_i x^i]$ using d parallel applications of the multiplication protocol and finally summing the products all up.

Theorem 1. *Let a shared polynomial $[F(X)]$ of maximal degree d (i.e., shared field elements $[a_0], \ldots, [a_d]$ such that $F(X) = \sum_{i=0}^{d} [a_i] X^i$) and a shared field element $[x]$ (for any $x \in \mathbb{F}_p$, possibly $x = 0$) be given. There exists a perfectly secure multi-party protocol that computes shares $[F(x)]$ in constant rounds and $O(d)$ applications of the multiplication protocol.*

3.4 Perfectly secure polynomial evaluation of a shared matrix

In this section we generalize the results from the last sections to the case of evaluating a shared matrix in a known/shared polynomial. Let a share $[A]$ of a matrix $A \in \mathbb{F}_p^{m\times m}$ be given, together with a public polynomial $F(x)$ of degree d. We want to give a multi-party protocol that computed shares $[F(A)]$. With known techniques, similar to the finite field case from Section 3.1 this can be carried out using $O(d^2 m^3)$ applications of the multiplication protocol.

Analogously to Section 3.2, for an $m \times m$ matrix A we define the $2m \times 2m$ matrix $M(A)$ over $\mathbb{F}_p$ as

$$M(A) = \begin{pmatrix} A & -I_m \\ I_m & 0 \end{pmatrix},$$

where I_m is the $m \times m$ identity matrix. We note that since $\det(M(A)) = 1$, $M(A)$ is non-singular for each $A \in \mathbb{F}_p^{m\times m}$, including the special case of singular A. Then again the following identity is easy to show by induction over d:

$$M(A)M^{d-1}(2A) = \begin{pmatrix} T_d(A) & -T_{d-1}(A) \\ U_{d-1}(A) & -U_{d-2}(A) \end{pmatrix}.$$

Proposition 2. *Let a set of ℓ public polynomials $F_i \in \mathbb{F}_p[X]$ of maximal degree d and a shared $m \times m$ matrix $[A]$ be given. There exists a perfectly secure multi-party protocol that computes all shares $([F_1(A)], \dots, [F_l(A)])$ in constant rounds and $O(dm^3)$ applications of the multiplication protocol.*

Theorem 2. *Let a shared polynomial $[F(x)]$ of maximal degree d and a shared $m \times m$ matrix $[A]$ be given. There exists a perfectly secure multi-party protocol that computes shares $[F(x)]$ in constant rounds and $O(dm^3)$ applications of the multiplication protocol.*

4 Solving Linear Systems of Equations

In this section we provide the necessary mathematical framework for understanding our algorithm. In particular, we present here the probabilistic algorithm to solve linear systems of equations that will be implemented in Section 5 in a secure multi-party computation protocol. This algorithm is based on the methods presented in [10]. Specifically, we solve the linear system of equations $Ax = b$ by computing the Moore-Penrose pseudoinverse of the matrix A. Since we are dealing with finite fields, we have to use the results by Mulmuley [17] to avoid that certain subspaces have nontrivial intersection with their orthogonals.

4.1 Computing the rank of a matrix

Let $\mathbb{K}$ be a field. For every pair of vectors $u, v \in \mathbb{K}^n$, we notate $\langle u, v \rangle$ for the usual scalar product $\langle u, v \rangle = \sum_{i=1}^{n} u^i v^i$. If $V \subset K^n$ is a subspace, we notate $V^\perp = \{u \in \mathbb{K}^n : \langle u, v \rangle = 0 \text{ for every } v \in V\}$. Clearly, $\dim V^\perp = n - \dim V$. It is well known that $V^\perp \cap V = \{0\}$ if $\mathbb{K} = \mathbb{Q}$ or $\mathbb{K} = \mathbb{R}$. This does not hold in general if $\mathbb{K}$ has positive characteristic.

If A is an $n \times m$ matrix over the field $\mathbb{K}$, the *Gram matrix* of A is defined by $G(A) = A^\top A$, where $A^\top$ denotes the transpose of A. For every $i = 1, \dots, m$, we take the vector $u_i \in \mathbb{K}^n$ corresponding to the i-th column of A. Then, the entries of the Gram matrix are the scalar products of these vectors, that is, $G(A) = (\langle u_i, u_j \rangle)_{1 \leq i,j \leq m}$.

Consider the vector spaces $E = \mathbb{K}^m$ and $F = \mathbb{K}^n$ and let A be an $n \times m$ matrix over $\mathbb{K}$ representing a linear mapping $A \colon E \to F$. Then, the transpose matrix $A^\top$ corresponds to a linear mapping $A^\top \colon F \to E$ such that $\langle Ax, y \rangle = \langle x, A^\top y \rangle$ for every pair of vectors $x \in E$ and $y \in F$. Then, $\ker A^\top = (\operatorname{Im} A)^\perp$ and $\operatorname{Im} A^\top = (\ker A)^\perp$. The terminology we introduce in the following definition will simplify the presentation.

Definition 1. *A subspace $V \subset \mathbb{K}^n$ is said to be* suitable *if $V^\perp \cap V = \{0\}$. We say that a matrix A is* suitable *if $\operatorname{Im} A$ is a suitable subspace, that is, if $(\operatorname{Im} A)^\perp \cap \operatorname{Im} A = \{0\}$.*

Lemma 1. *Let A be an $n \times m$ matrix over $\mathbb{K}$ and let $G = A^\top A$ be its Gram matrix. Then A is a suitable matrix if and only if $\ker G = \ker A$.*

Proof. Observe that $\ker A \subset \ker G$. If A is suitable and $x \in \ker G$, then $Ax \in \operatorname{Im} A \cap \ker A^\top = \operatorname{Im} A \cap (\operatorname{Im} A)^\perp = \{0\}$. Conversely, if $\ker A = \ker G$ and $y = Ax \in \operatorname{Im} A \cap (\operatorname{Im} A)^\perp$, then $x \in \ker G = \ker A$, and hence $y = 0$.

Lemma 2. *Let A be an $n \times m$ matrix over $\mathbb{K}$ and suppose that A and $A^\top$ are both suitable matrices. Let $G = A^\top A$ and $H = AA^\top$ be the Gram matrices of A and $A^\top$, respectively. Then G and H are suitable matrices.*

Proof. Since G is a symmetric matrix, $\operatorname{Im} G = (\ker G)^\perp$. By applying Lemma 1, $\operatorname{Im} G = (\ker G)^\perp = (\ker A)^\perp = \operatorname{Im} A^\top$. Since $A^\top$ is suitable, $(\operatorname{Im} G)^\perp \cap \operatorname{Im} G = \{0\}$. Symmetrically, H is suitable as well.

Lemma 3. *Let G be a symmetric $m \times m$ matrix and assume G is suitable. Consider $P(X) = \det(XI_m - G) = X^m + a_1X^{m-1} + \cdots + a_{m-1}X + a_m$, the characteristic polynomial of G. Then $\operatorname{rank} G = \max\{i : a_i \neq 0\}$.*

Proof. From Lemma 1, $\ker G^2 = \ker G$. If $r = \max\{i : a_i \neq 0\}$, the characteristic polynomial of G is of the form $P(X) = X^{m-r}Q(X)$ with $Q(0) \neq 0$. Then $\dim\ker G = \dim\ker G^{m-r} = m - r$, and hence $\operatorname{rank} G = r$.

From the previous lemmas the rank of the matrix A can be found by computing the characteristic polynomial of the Gram matrix $G(A) = A^\top A$. Nevertheless, we need that both A and $A^\top$ are suitable matrices. If we are dealing with a field with positive characteristic we cannot be sure that this is the case. We avoid this problem by applying a random transformation to the matrix A that, with high probability, produces a matrix with the same rank and verifying that property. This can be done by using Theorem 3, due to Mulmuley [17], and Propositions 3 and 4.

Theorem 3. *Consider the field $\mathbb{K}(x)$, a transcendental extension of the field $\mathbb{K}$, and the diagonal matrix $D_x = \operatorname{diag}(1, x, \ldots, x^{n-1})$, which defines a linear mapping $D_x \colon K(x)^n \to K(x)^n$. Then for every subspace $V \subset \mathbb{K}$, the subspace $V_x = D_x(V') \subset \mathbb{K}(x)^n$, where $V' \subset \mathbb{K}(x)^n$ is the natural extension of V, is suitable. As a consequence, for every $n \times m$ matrix A over the field $\mathbb{K}$, the matrix D_xA (over the field $\mathbb{K}(x)$) is suitable.*

The proofs of the next two propositions will be given in the full version.

Proposition 3. *Let $\mathbb{K}$ be a finite field with $|\mathbb{K}| = p$. Consider the vector space $F = \mathbb{K}^n$ and a subspace $V \subset F$. For every $\alpha \in \mathbb{K}$, we consider the diagonal matrix $D_\alpha = \operatorname{diag}(1, \alpha, \alpha^2, \ldots, \alpha^{n-1})$. If an invertible element $\alpha \in \mathbb{K}^*$ is chosen uniformly at random, then the probability that the subspace $V_\alpha = D_\alpha(V) \subset F$ is suitable is at least $1 - 2n(n-1)/p$.*

Proposition 4. *Let $\mathbb{K}$ be a finite field with $|\mathbb{K}| = p$ and let A be an $n \times m$ matrix over the field $\mathbb{K}$. For every $\alpha \in \mathbb{K}$, take the diagonal matrices $D_{n,\alpha} = \operatorname{diag}(1, \alpha, \ldots, \alpha^{n-1})$ and $D_{m,\alpha} = \operatorname{diag}(1, \alpha, \ldots, \alpha^{m-1})$, and the matrix $A_\alpha = D_{n,\alpha}AD_{m,\alpha}$. Then the probability that both A_α and $A_\alpha^\top$ are suitable matrices if an invertible element $\alpha \in \mathbb{K}^*$ is chosen uniformly at random is at least $1 - (2/p)(n(n-1) + m(m-1))$.*

4.2 Moore-Penrose Pseudoinverse

Consider the vector spaces $E = \mathbb{K}^m$ and $F = \mathbb{K}^n$ and let A be an $n \times m$ matrix over $\mathbb{K}$ representing a linear mapping $A\colon E \to F$. A *pseudoinverse* of A is any $m \times n$ matrix $B\colon F \to E$ such that $ABA = A$ and $BAB = B$. Given two subspaces $V, W \subset E$, the notation $E = V \oplus W$ means that E is the *direct sum* of V and W, that is, $E = V + W$ and $V \cap W = \{0\}$.

There can exist many different pseudoinverses of a matrix. If B is a pseudoinverse of A, then $E = \operatorname{Im} B \oplus \ker A$ and $F = \operatorname{Im} A \oplus \ker B$. Moreover, for every pair of subspaces $V_1 \subset E$ and $V_2 \subset F$ such that $E = V_1 \oplus \ker A$ and $F = \operatorname{Im} A \oplus V_2$, there exists a unique pseudoinverse B of A such that $V_1 = \operatorname{Im} B$ and $V_2 = \ker B$. Finally, there is at most one pseudoinverse B of A such that AB and BA are symmetric matrices. This is the only pseudoinverse with $\operatorname{Im} B = (\ker A)^\perp$ and $\ker B = (\operatorname{Im} A)^\perp$. Of course, such a pseudoinverse exists if and only if $\ker A \subset E$ and $\operatorname{Im} A \subset F$ are suitable subspaces.

Definition 2. Let A be an $n \times m$ matrix corresponding to a linear mapping $A\colon E \to F$ such that $\ker A \cap (\ker A)^\perp = \{0\}$ and $\operatorname{Im} A \cap (\operatorname{Im} A)^\perp = \{0\}$, that is, A and $A^\top$ are suitable matrices. The *Moore-Penrose pseudoinverse* $A^\dagger$ of A is the unique pseudoinverse of A with $\operatorname{Im} A^\dagger = (\ker A)^\perp$ and $\ker A^\dagger = (\operatorname{Im} A)^\perp$. Actually, the Moore-Penrose pseudoinverse of A can be defined too as the unique $m \times n$ matrix $A^\dagger\colon F \to E$ such that $AA^\dagger A = A$, and $A^\dagger A A^\dagger = A^\dagger$, and $AA^\dagger$ and $A^\dagger A$ are symmetric matrices.

Observe that the Moore-Penrose pseudoinverse of A can be defined only if A and $A^\top$ are suitable matrices. Assume that we are in this situation. We consider $G = A^\top A$ and $H = AA^\top$, the Gram matrices of A and $A^\top$. From Lemma 2, G and H are suitable matrices with $\ker G = \ker A$ and $\ker H = \ker A^\top$.

We present next a useful expression for $A^\dagger$ in terms of the characteristic polynomial of H or the one of G. Let $f_0\colon \operatorname{Im} A^\top \to \operatorname{Im} A$ be the linear mapping obtained from the restriction of $A\colon E \to F$ to $\operatorname{Im} A^\top$ and let $\pi\colon F \to \operatorname{Im} A$ be the orthogonal projection over $\operatorname{Im} A$. It is not difficult to check that f_0 is invertible and that $A^\dagger = f_0^{-1}\pi$. Consider $r = \operatorname{rank} A = \operatorname{rank} A^\top = \operatorname{rank} G = \operatorname{rank} H$. From Lemma 3, the characteristic polynomial of H is of the form $\det(XI_n - H) = X^n + a_1 X^{n-1} + \cdots + a_r X^{n-r}$ with $a_r \neq 0$. Moreover, the characteristic polynomial of G has the same coefficients as the one of H, that is, $\det(XI_m - G) = X^m + a_1 X^{m-1} + \cdots + a_r X^{m-r}$. Consider a vector $y \in F$ and take $z = H^r y + a_1 H^{r-1} y + \cdots + a_{r-1} H y + a_r y$. By applying Cayley-Hamilton and taking into account that $\ker H^2 = \ker H$, we get that $z \in \ker H$. Then, $y = a_r^{-1} z - a_r^{-1}(H^r y + a_1 H^{r-1} y + \cdots + a_{r-1} H y) = z_1 + z_2$ with $z_1 = a_r^{-1} z \in \ker H = \ker A^\top$ and $z_2 \in \operatorname{Im} H = \operatorname{Im} A$. Therefore, the orthogonal projection of $y \in F$ on $\operatorname{Im} A$ is $\pi(y) = -a_r^{-1}(H^r y + a_1 H^{r-1} y + \cdots + a_{r-1} H y)$. Now, taking into account that $f_0^{-1} A A^\top = A^\top$, we get that

$$A^\dagger = f_0^{-1}\pi = \ -a_r^{-1} f_0^{-1}(H^r + a_1 H^{r-1} + \cdots + a_{r-1} H) =$$

$$= -a_r^{-1} A^\top (H^{r-1} + a_1 H^{r-2} + \cdots + a_{r-1} I_n) = -a_r^{-1}(G^{r-1} + a_1 G^{r-2} + \cdots + a_{r-1} I_m) A^\top .$$

The Moore-Penrose pseudoinverse can be used to solve a linear system of equations of the form $Ax = b$, but we need that both A and $A^\top$ are suitable matrices. Nevertheless, by using Proposition 4, we can apply a random transformation to the matrix A to obtain a matrix A_α verifying this property with high probability.

4.3 The algorithm

Given the theoretical results from the preceding sections, we extract the following probabilistic algorithm for solving linear systems of equations.

Algorithm Linsolve.

```
The input is $A, y, m, n$, where $A \in \mathbb{F}_p^{n\times m}$, $y \in \mathbb{F}_p^n$, and $m \leq n$
The output is $x$ such that $Ax = y$ and a bit $s$ indicating if the system is solvable.
  1. Pick random $\alpha \overset{R}{\leftarrow} \mathbb{F}_p$ and create the $n \times n$ matrix $D_{n,\alpha} = \mathrm{diag}(1, \alpha, \ldots, \alpha^{n-1})$
     and the $m \times m$ matrix $D_{m,\alpha} = \mathrm{diag}(1, \alpha, \ldots, \alpha^{m-1})$.
  2. Compute $A_\alpha \leftarrow D_{n,\alpha} A D_{m,\alpha}$ and $y_\alpha \leftarrow D_{n,\alpha} y$.
  3. Compute $G \leftarrow A_\alpha^\top A_\alpha \in \mathbb{F}^{m\times m}$ //$G$ is a symmetric $m \times m$ matrix
  4. Compute the coefficients $(a_1, \ldots, a_m)$ of the characteristic polynomial
     of $G$
  5. Compute the rank $r$ of $G$
  6. Compute $A_\alpha^\dagger \leftarrow -a_r^{-1}(G^{r-1} + a_1 G^{r-2} + \cdots + a_{r-1} I_m) A_\alpha^\top$
  7. Check if $A_\alpha A_\alpha^\dagger y_\alpha = y_\alpha$. If not, the system has no solution, and the bit
     $s = 0$ is returned.
  8. If the system has a solution return $s = 1$ and $x \leftarrow D_{m,\alpha} A_\alpha^\dagger y_\alpha$.
```

Correctness of the algorithm is stated in the next lemma.

Lemma 4. *Let $A \in \mathbb{F}_p^{n\times m}$, $y \in \mathbb{F}_p^n$, and $m \leq n$. Suppose that $y \in \operatorname{Im} A$, that is, that the system has a solution. Let x be the output of the randomized algorithm Linsolve applied to A, y, m, n. Then, with probability at least $1 - (2/p)(n(n-1) + m(m-1))$, we have $Ax = y$.*

Proof. Clearly, $Ax = D_{n,\alpha}^{-1} A_\alpha D_{m,\alpha}^{-1} D_{m,\alpha} A_\alpha^\dagger y_\alpha = D_{n,\alpha}^{-1} A_\alpha A_\alpha^\dagger y_\alpha = D_{n,\alpha}^{-1} y_\alpha = y$.

Until now we assumed $m \leq n$. If $n \leq m$, we should adapt the algorithm by using $H = A_\alpha A_\alpha^\top$ instead of G to obtain $A_\alpha^\dagger$. Since the obtained solution depends on α, a random solution x_0 of the linear system of equations $Ax = y$ is obtained but, clearly, the probability distribution is not uniform on the set of all possible solutions. If we want the output of the algorithm to be uniformly distributed among all possible solutions of the system, we can take a random vector $z \in \mathbb{F}_p^m$ and compute $x_1 = x_0 + D_{m,\alpha}(I_m - A_\alpha^\dagger A_\alpha)z$. Finally, observe that by picking at random an $m \times m$ invertible matrix M and computing $D_{m,\alpha}(I_m - A_\alpha^\dagger A_\alpha)M$, we get a random element among all $m \times m$ matrices whose columns span $\ker A$.

5 The Secure Multi-Party Protocols

Theorem 4. *Let shares $[A]$ of an $n \times m$ matrix and shares $[y]$ of an n-dimensional vector be given. There exists a multi-party protocol that, with probability at least $1 - O(n^2/p)$, securely computes shares $[x]$ of a solution to the system of linear equations $Ax = y$ and shares $[s]$ of a bit indication if the system is solvable. The protocol runs in constant rounds and uses $O(m^4 + m^2n + m \log p)$ applications of the multiplication protocol.*

We remark that the above protocol can easily be extended to yield shares of a uniform solution of the system.

Theorem 5. *Assume $\mathbb{F}_p$ has characteristic at least m. Let shares $[A]$ of an $n \times m$ matrix be given. There exists a multi-party protocol that, with probability at least $1-O(n^2/p)$, securely computes shares $([a_1], \ldots, [a_n])$ of the characteristic polynomial of A. The protocol runs in constant rounds and uses $O(m^4 + m^2n)$ applications of the multiplication protocol.*

Proof of Theorem 4 (sketch). We show how to securely implement each step of the protocol LINSOLVE from Section 4.3 within the given complexity bounds. We remark that, as a by-result, we also get efficient constant-round protocols for securely computing the characteristic polynomial and the rank of a given shared matrix. Details of the protocol are given in the full version of the paper. Instead we give some intuition and mention the main techniques used.

For the first two steps the players jointly agree on a common public value α. Since α is public, for computing shares of the appearing matrices there is no further interaction needed. Computing shares of the characteristic polynomial in Step 4 is done with the protocol from Theorem 5. In Step 5, shares of the rank need to be computed that, by Lemma 3, can be derived from the characteristic polynomial. Here we have to use several sequential applications of the equality protocol EQ to finally compute the rank in unary representation. Step 6 computes shares of the Moore-Penrose pseudoinverse. Note that the formula to compute $A_\alpha^\dagger$ *explicitly* depends on the rank r of matrix G. Since we do not know r in the clear we need to develop a technique to obliviously evaluate the matrix G in the correct polynomial. A first approach is to evaluate $A_\alpha^\dagger$ for all the possible values of the rank $r \in \{1, \ldots, m\}$ and then sum the resulting matrices weighted with the respective bit indicating if the summation index equals the rank. Note that shares of the latter bits are known from the last step. However, the naive complexity of this approach is m^5. Using certain linearities in the coefficients of the sums of the above polynomials we develop an alternative approach to obtain the necessary complexity $O(m^4)$. Efficiency of this step heavily relies on our efficient polynomial evaluation protocols proposed in Section 3. The rest of the steps are more or less easy to implement. We mention that the complexity of the protocol is dominated by Steps 4 and 6 ($O(m^4)$), Step 5 ($O(m \cdot \log p)$ for in total $O(m)$ applications of EQ), and computing two products of an $m \times n$ with an $m \times m$ matrix ($O(m^2n)$) in Steps 3 and 6. Security of the protocol follows by the security of the sub-protocols used.

Proof of Theorem 5 (sketch). We assume we are given shares of a symmetric square $m \times m$ matrix, if not apply the first three steps of the Linsolve protocol using $O(m^2 n)$ multiplications. Due to [7] there already exists a constant-round protocol for computing shares of the characteristic polynomial. We present an alternative and much simpler protocol based on Leverrier's Lemma (see Lemma 5) which basically says that the coefficients of the characteristic polynomial can be retrieved by inverting a certain non-singular lower-triangular matrix S, where each entry below the diagonal is the trace of the powers G^i of the matrix G. Leverrier's lemma is obtained by combining Newton's identities with the fact that these traces correspond to sums of powers of the characteristic roots.

Computing shares of all the m powers G^i of G can be done using the protocol from Proposition 2 in $O(m^4)$ applications of the multiplication protocol. All the traces of G^i can be locally computed by the players and assembled into the $m \times m$ matrix S. Finally the players compute the inverse of the non-singular matrix S using the protocol INV which enables them to compute the coefficients of the characteristic polynomial. The total complexity of the protocol is $O(m^4 + m^2 n)$ applications of the multiplication protocol and it runs in constant rounds. More details will be given in Appendix A. Security of the protocol follows by the security of the sub-protocols used.

Acknowledgments

The authors would like to thank an anonymous referee from CRYPTO 2006 who proposed the "small error protocol" for secure polynomial evaluation from Section 3.1.

References

1. J. Bar-Ilan and D. Beaver. Non-cryptographic fault-tolerant computing in a constant number of rounds interaction. In *8th ACM PODC*, pages 201–209, Edmonton, Alberta, Canada, August 14–16, 1989.
2. D. Beaver. Minimal latency secure function evaluation. In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 335–350, Bruges, Belgium, May 14–18, 2000. Springer-Verlag.
3. D. Beaver, S. Micali, and P. Rogaway. The round complexity of secure protocols. In *22nd ACM STOC*, pages 503–513, Baltimore, Maryland, USA, May 14–16, 1990. ACM Press.
4. M. Ben-Or and R. Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992.
5. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computations. In *20th ACM STOC*, pages 1–10, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.
6. D. Chaum, C. Crépeau, and I. Damgård. Multiparty unconditionally secure protocols. In *20th ACM STOC*, pages 11–19, Chicago, Illinois, USA, May 2–4, 1988. ACM Press.

7. R. Cramer and I. Damgård. Secure distributed linear algebra in a constant number of rounds. In *CRYPTO 2001*, volume 2139 of *LNCS*, pages 119–136, Santa Barbara, CA, USA, August 19–23, 2001. Springer-Verlag.
8. I. Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen, and T. Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In *TCC 2006*, volume 3876 of LNCS, pages 285–304. Springer-Verlag.
9. U. Feige, J. Kilian, and M. Naor. A minimal model for secure computation. In *26th ACM STOC*, pages 554–563, Montréal, Québec, Canada, May 23–25, 1994. ACM Press.
10. H. Lombardi G. M. Diaz-Toca, L. Gonzalez-Vega. Generalizing cramer's rule: Solving uniformly linear systems of equations. *SIAM J. Matrix Anal. Appl.*, 27:621–637, 2005.
11. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *19th ACM STOC*, pages 218–229, New York City,, New York, USA, May 25–27, 1987. ACM Press.
12. Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. In *Proc. 5th Israel Symposium on Theoretical Comp. Sc. ISTCS*, pages 174–183, 1997.
13. Y. Ishai and E. Kushilevitz. Randomizing polynomials: A new paradigm for round-efficient secure computation. In *41st FOCS*, Las Vegas, Nevada, USA, November 12–14, 2000. IEEE Computer Society Press.
14. J. Jájá. *An Introduction to Parallel Algorithms.* Eddison-Wesley, 1992.
15. E. Kiltz, P. Mohassel, E. Weinreb, and M. Franklin. Secure linear algebra using linearly recurrent sequences. In *TCC 2007*, volume 4392 of *LNCS*, pages 291–310. Springer-Verlag.
16. F. T. Krogh. Efficient algorithms for polynomial interpolation and numerical differentiationi. *Math. Comput.*, 24:185–190, 1970.
17. K. Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Combinatorica*, 7:101–104, 1987.
18. T. Nishide and K. Ohta. Multiparty Computation for Interval, Equality, and Comparison without Bit-Decomposition Protocol. In *PKC 2007*, volume 4450 of *LNCS*, pages 343–360. Springer-Verlag.
19. K. Nissim and E. Weinreb. Communication efficient secure linear algebra. In *TCC 2006*, LNCS. Springer-Verlag, Berlin, Germany, 2006.
20. A. Schrijver. *Combinatorial Optimization - Polyhedra and Efficiency.* Springer-Verlag, Berlin, 2003.
21. A. Yao. Protocols for secure computation. In *23rd FOCS*, pages 160–164, Chicago, Illinois, November 3–5, 1982. IEEE Computer Society Press.
22. A. Yao. How to generate and exchange secrets. In *27th FOCS*, pages 162–167, Toronto, Ontario, Canada, October 27–29, 1986. IEEE Computer Society Press.

A Protocol for the Characteristic Polynomial

We assume we are given shares of a symmetric square $m \times m$ matrix (possibly singular), if not apply the first three steps of the Linsolve protocol using $O(m^2n)$ multiplications. We want to compute shares $([a_1], \dots, [a_m])$ of the characteristic polynomial of G. With the techniques of Cramer and Damgård [7] this can be reduced to computing m times (in parallel) the determinant of a non-singular matrix and applying polynomial interpolation to reconstruct the coefficients. Since securely computing the determinant can essentially be done by multiplying two shared $m \times m$ matrices, which can be carried out in constant rounds and using $O(m^3)$ applications of the multiplication protocol, the whole protocol runs in constant rounds and $O(m^4)$ applications of the multiplication protocol. We write

$$([a_1], \ldots, [a_m]) \leftarrow \text{CHARPOLY}([G]).$$

We now describe an alternative and more simple approach with roughly the same complexity based on Leverrier's Lemma [14, Chapter 8]. For this technique to work we will have to assume that the finite field's characteristic is at least m. Efficiency of this approach depends on the new secure polynomial evaluation technique from Section 3. We note that the use of Leverrier's Lemma in that context was first proposed by M. Rabin in [7]. Our algorithm retrieves the coefficients of the characteristic polynomial by inverting a certain lower-triangular matrix, where each entry below the diagonal is the trace of the powers G^i of the matrix G. The following lemma is obtained by combining Newton's identities with the fact that these traces correspond to sums of powers of the characteristic roots.

Lemma 5 (Leverrier's Lemma). *The coefficients $a_1, a_2, \ldots, a_m$ of the characteristic polynomial of a matrix G satisfy*

$$S \cdot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_{m-1} \\ a_m \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_{m-1} \\ s_m \end{pmatrix}, \text{ where } S = \begin{bmatrix} 1 & 0 & 0 & \ldots & 0 & 0 \\ s_1 & 2 & 0 & \ldots & 0 & 0 \\ s_2 & s_1 & 3 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ s_{m-2} & s_{m-3} & s_{m-4} & \ldots & m-1 & 0 \\ s_{m-1} & s_{m-2} & s_{m-3} & \cdots & s_1 & m \end{bmatrix},$$

and $s_i = \mathrm{tr}(G^i) = \sum_{j=1}^{m} G_{jj}^i$ *for all* $1 \leq i \leq m$.

Based on Leverrier's Lemma we can securely compute shares of the characteristic polynomial as follows: First the players compute shares of all the powers of G using the protocol from Proposition 2 and then they locally compute shares of the traces $[s_i]$. Then they apply the matrix inversion protocol to compute $[S^{-1}] \leftarrow \text{INV}([S])$, where S is the matrix from Lemma 5. Finally they calculate shares of the matrix-vector product $S^{-1} \cdot (s_1, s_2, \ldots, s_m)^\top$ to obtain shares of the characteristic polynomial. (Note that the matrix S is guaranteed to be non-singular since for it's determinant we have $\det(S) = \prod_{i=1}^{m} i$ which is non-zero by our assumption that $\mathbb{F}_p$ has characteristic at least m.) Using the protocol explained in Proposition 2 shares of all powers $G, G^2, \ldots, G^m$ can be computed in constant rounds and $O(m \cdot m^3) = O(m^4)$ applications of the multiplication protocol. The protocol is secure with probability at least $1 - O(m^2/p)$. This proves Theorem 5.