IACR Communications in Cryptology
ISSN 3006-5496, Vol. 1, No. 1, 25 pages.

https://doi.org/10.62056/avzojbkrz

Secure Multi-Party Linear Algebra with Perfect Correctness

Jules Maire and Damien Vergnaud

Sorbonne Université, CNRS, LIP6, Paris, France

Abstract. We present new secure multi-party computation protocols for linear algebra over a finite field, which improve the state-of-the-art in terms of security. We look at the case of *unconditional security with perfect correctness*, i.e., information-theoretic security without errors. We notably propose an expected constant-round protocol for solving systems of m linear equations in n variables over $\mathbb{F}_q$ with expected complexity $O(k(n^{2.5} + m^{2.5} + n^2 m^{0.5}))$ where $k > m(m+n)+1$ (complexity is measured in terms of the number of secure multiplications required). The previous proposals were not error-free: known protocols can indeed fail and thus reveal information with probability $\Omega(\mathsf{poly}(m)/q)$. Our protocols are simple and rely on existing computer-algebra techniques, notably the Preparata-Sarwate algorithm, a simple but poorly known "baby-step giant-step" method for computing the characteristic polynomial of a matrix, and techniques due to Mulmuley for error-free linear algebra in positive characteristic.

Keywords: Secure Multi-Party Computation · Linear Algebra · Preparata-Sarwate Algorithm · Moore-Penrose Pseudo-Inverse

1 Introduction

The importance of linear algebra computation over finite fields for a wild range of tasks is a well-established fact (*e.g.* for integer polynomial factorization, Gröbner basis computation, integer system solving, large integer factorization, discrete logarithms, error correcting codes,...). The concept of secure multi-party computation (MPC) was introduced by Yao [Yao86] and allows mutually distrusting parties to run joint computations without disclosing any participant's private inputs.

We present new MPC protocols for linear algebra computation over a finite field, improving state-of-the-art security. We notably propose efficient protocols for (matrix)-polynomial evaluation, determinant computation, and other linear algebra problems, particularly the computation of the characteristic polynomial which underlies many problems such as the resolution of linear systems of equations. We target protocols with everlasting security unconditionally, without relying on unproven intractability assumptions. There already exist numerous protocols in this setting, but we require in addition that in our protocols, (honest) parties always get a valid output and that protocols fail with a zero-error probability. Indeed, known protocols can fail and thus reveal information on the parties' inputs. Apart from being a natural goal, achieving unconditional security and perfect correctness provides important security advantages over protocols that have a negligible probability of failure. Indeed, it has been proven that every protocol that is perfectly secure in the stand-alone model is secure under concurrent general composition [KLR06]. Moreover, we target efficient multi-party protocols, meaning that our protocols enjoy both low communication and round complexity. Indeed, since in almost all systems, the time

E-mail: jules.maire@lip6.fr (Jules Maire), damien.vergnaud@lip6.fr (Damien Vergnaud)

Received: 2024-01-09 Accepted: 2024-03-05

spent on sending and receiving messages is large compared to local computation time, one tries to achieve an expected constant round complexity while keeping the communication complexity as low as possible.

1.1 Related work

General results in MPC do not yield efficient protocols for linear algebra with information-theoretic security. Cramer and Damgård [CD01] proposed the first efficient information-theoretically secure MPC protocols for solving linear systems, i.e. that achieve security even against computationally unbounded adversaries. The main focus of their proposal was to achieve constant-round complexity, and they notably proposed a constant-round protocol for solving m equations in n variables with communication complexity $\Omega(n^4)$ elements of the underlying finite field, which can be reduced to $O(n^3)$ with the subsequent matrix product protocol from [MW08]. In 2007, Cramer, Kiltz and Padró [CKP07] designed a constant-round protocol for solving m equations in n variables with communication complexity $O(n^4 + n^2\,m)$. Then, Mohassel and Weinreb [MW08] developed a constant-round protocol, with $O(t\,n^{2+1/t})$ communication for every constant $t \in \mathbb{N}$, for computing the rank and solving a shared linear system of equations. In all the protocols from [CD01, CKP07, MW08], non-zero error probabilities arise when the parties happen to select obliviously zeroes of "hidden" polynomials. This error probability is typically a polynomial in the dimensions of the matrix considered divided by the cardinal of the underlying finite field (e.g. $\Theta(m^2/q)$ in [CKP07, MW08] over a finite field $\mathbb{F}_q$). The authors of [CD01, CKP07, MW08] thus require to consider only large values of q, typically q superpolynomial in m to achieve protocols with negligible error probability (which may not be compatible with the problem being solved), or to instantiate the protocols in large enough extensions of the underlying finite field (which increase the computation and communication costs). Even in these cases, the error probability is not null and may reveal information on the parties' inputs. In [CD01], Cramer and Damgård mentioned a work in progress (co-authored with Daza) to achieve perfect security, but as far as we know, this work has not been published, and it remains an open problem to propose efficient constant-round protocols for linear algebra with perfect correctness and unconditional security.

Numerous protocols were also proposed for computationally secure MPC linear algebra. For instance, using the garbled circuit method of Yao [Yao86], one can get a constant-round two-party protocol for various linear algebraic problems. A protocol due to Nissim and Weinreb [NW06] was the first to improve the communication complexity to roughly $O(n^2)$ (for $n \times n$ matrices), but with a trade-off on a large $O(n^{0.275})$ round complexity. The protocols from [CD01, CKP07] can be readily adapted to the computationally-secure setting using linearly homomorphic encryption schemes, and the resulting schemes achieve similar complexities. Later, [KMWF07] achieved $O(n^2 \log n)$ communication complexity and $O(\log n)$ round complexity with an ingenuous concatenation idea to compute iterative powers of a matrix. This protocol to solve linear systems also has a non-zero error probability of $3n^2/q$ over a finite field of cardinal q. The protocols from [MW08] can also be adapted to the computational setting (with similar complexities and error probability). Finally, Bouman and de Vreede [BdV18] recently proposed two protocols based on (oblivious) Gaussian elimination with $O(n^3)$ computational complexity and $O(n)$ round complexity, and based on block-recursive matrix decomposition with $O(n^2)$ computational complexity and $O(n^{1.585})$ round complexity. Both protocols use a preconditioning method, and non-zero error probabilities also arise.

1.2 Our contributions

We present new secure multi-party computation protocols for linear algebra over a finite field with *unconditional security and perfect correctness*, i.e., information-theoretic without

error. These protocols rely on known techniques in computer algebra and their adaptation in an MPC setting is actually more routine than innovative. However, even if this paper assumes an expository character, we contend that describing these protocols is of interest to the community given the potential impact to design threshold cryptosystems. In particular, the NIST has recently initiated a process to solicit, evaluate, and standardize "threshold schemes, for a secure distribution of trust in the operation of cryptographic primitives". Variants of our protocols (with security against malicious adversaries) could find applications to design post-quantum threshold digital signatures. They can be used to distribute the signing algorithms used, for instance, in the code-based Wave signatures [DST19] or the multivariate unbalanced Oil and Vinegar signatures [KPG99, Beu21, BCH+23] (that both require a linear system solving in a small finite field).

Previous efficient proposals [CKP07, KMWF07, MW08] are based on the computation of a certain characteristic polynomial. Cramer, Kiltz, and Padró [CKP07] presented a protocol for secure polynomial evaluation of a shared polynomial of degree d at a shared point that runs in a constant number of rounds and $O(d)$ secure multiplications. However, the protocol leaks information about the input with probability $1/q$ over a finite field of cardinal q. Then, they developed a perfectly correct protocol with $O(d)$ secure multiplications using Chebyshev polynomials. As a first contribution, we propose simple secure protocols for polynomial evaluation of a shared polynomial of degree d at a shared point with perfect correctness that runs in $O(t)$ rounds and has communication complexity $O(t\, d^{1/t})$ for any parameter $t \in \mathbb{N}$. For polynomial evaluation of a shared polynomial at a shared $n \times n$ matrix, the complexity increases to $O(t\, n^2\, d^{1/t})$. These protocols are of independent interest and can be used for instance in the recent *Polymath* framework from Lu, Yu, Kate, and Maji [LYKM22] with round complexity (and therefore latency) for secure polynomial evaluations of scalars and matrices independent of the polynomial degree and matrix dimensions (and therefore for their interesting use cases of privacy-preserving evaluation of decision trees and privacy-preserving evaluation of Markov processes).

Using these tools, we propose an expected constant-round protocol for solving systems of m linear equations in n variables over $\mathbb{F}_q$ with expected complexity $O(k(n^{2.5} + m^{2.5} + n^2 m^{0.5}))$ (where complexity is measured in terms of the number of secure multiplications required) for $k > m(m+n)+1$ when the field characteristic is greater than n. This last condition can be removed via a work of Schönhage [Sch93] for securely computing the characteristic polynomial over a field of positive characteristic. This increases the cost of communication by a factor n.

As mentioned above, our protocols are simple and rely on existing computer algebra techniques. In particular, we make use of the Preparata-Sarwate algorithm [PS78]. It is a simple "baby-step giant-step" method for computing the characteristic polynomial, determinant, and adjugate of a $n \times n$ matrix using only ring operations together with exact divisions by small integers with complexity $O(n^{2.5})$. This algorithm is poorly known and has been rediscovered several times (see e.g. [Joh20]). We adapt this algorithm for secure MPC using classical techniques. The algorithm boils down to performing $O(\sqrt{n})$ multiplications of matrices, each naively requiring $O(n^2)$ operations in the ring. We take up a technique sketched without details in [MW08] allowing us to perform these multiplications at a communication cost similar to $O(n^2)$ secure multiplications (but always with a computational cost in $O(n^3)$).

Using this error-free protocol, we follow the blueprint from [CKP07] and obtain an error-free protocol for the computation of the Moore–Penrose pseudo-inverse of a matrix A over a finite field. This requires computing the characteristic polynomial of the so-called Gram matrix of A from which we can compute the Moore–Penrose pseudo-inverse *via* a technique due to Diaz-Toca, Gonzalez–Vega and Lombardi [DGL05] as an extension of the work of Mulmuley [Mul86] to achieve perfect correctness (and avoid the errors and possible leakage on the parties' inputs from [CKP07]).

2 Preliminaries

We denote by $\mathbb{F}_q$ a finite field with q elements, and by $GL_n(\mathbb{F}_q)$ the subset of non-singular matrices over $\mathbb{F}_q^{n\times n}$ which is a group for multiplication (with $n \in \mathbb{N}$).

Secure Multi-Party Computation (MPC). MPC deals with a set of parties who want to compute a public function of their secret inputs such that each party obtains the correct result but no additional information about the other parties' inputs. We consider the *honest but curious* model (or semi-honest setting), in which parties try to find out as much as possible about the other inputs despite following the protocol.

Security Model. Protocols can be categorized based on their security into two main types: those relying on computational hardness assumptions and those deemed *unconditionally secure* (information-theoretical). We focus on the second category: we construct secure protocols against adversaries with unlimited computing resources and time. Moreover, our protocols do not leak information with probability 1, and thus achieve *perfect correctness*. In the following, when we say that a protocol is secure, we mean that it is unconditionally secure.

Complexity Measures. Two measures of complexity are important for our protocols. The first one is the *communication complexity*, i.e., the total number of bits exchanged during the whole execution of the protocol. This complexity only depends on the number of secure multiplications that the protocol requires, since, for protocols relying on *linear secret sharings* (see below) only secure multiplications involve communication between parties. We consider a secure multiplication protocol that needs to communicate 2 field elements per invocation. Hence, to determine the communication complexity of a protocol, it is equivalent to computing the number of calls to the secure multiplication subprotocol. Fortunately, parties can batch some multiplications before interacting with others, decreasing the communication complexity. The second complexity measure is the *round complexity*, i.e., the number of sequential rounds of secure multiplication that the protocol invokes. In other words, it corresponds to the number of interactions during which each party is allowed to send one flow of messages to other parties.

2.1 Definitions of our theoretical information model

The protocols that we present do not rely on any cryptographic assumption, except that the underlying secret sharing has to be unconditionally secure. Let $P_1, \ldots, P_k$ be k parties taking part in some MPC protocol. We use a linear secret sharing scheme to design secure MPC protocols to share values over a finite field $\mathbb{F}_q$.

Linear secret sharing scheme. A secret sharing scheme is a cryptographic primitive with a sharing and a reconstruction phase. The sharing phase allows a secret to be distributed among a group of parties (by some dealer). Once the secret has been distributed, each of the parties holds a share of the secret. On its own, this share does not reveal any information about the secret, unless combined with sufficient other shares of a subset of the participants (the reconstruction phase). One denotes by $[s] = ([s]_1, \ldots, [s]_k) \in \mathbb{F}_q^k$ a secret sharing of $s \in \mathbb{F}_q$ with $[s]_i$ the share of the party P_i (for $1 \leq i \leq k$). A secret sharing scheme is *linear* if the reconstruction function of the secret from the shares is a linear mapping. Due to the linearity of the secret sharing, given secret sharings $[a]$ and $[b]$ and a third field element $c \in \mathbb{F}_q$, parties can compute their share of the secret sharing $[a + cb]$ locally (i.e. without communication). Furthermore, we require our linear secret sharing scheme to be *multiplicative*. In a nutshell, this means that a party P_i can use their shares from $[a]$ and $[b]$ to locally compute a value c_i. Then, via some computation using the c_i's, parties communicate to realize a refreshing step. Based on a public reconstruction vector $\lambda \in \mathbb{F}_q^k$, the product ab can be reconstructed, hence leading to a secret sharing of

ab. Moreover, one can construct a multiplicative linear secret sharing scheme from any linear secret sharing scheme without loss of efficiency [CDM00].

We may use Shamir's secret sharing along with BGW protocol [BGW88] with quadratic communication (in the number of parties) per secure multiplication. But any other linearly homomorphic secret sharing with, for example, the Damgård-Nielsen multiplication protocol [DN07] yields a protocol with linear communication at the cost of preparing a pair of random double sharings for each multiplicative gate.

The sharing of a vector or more generally of a matrix is seen component-wisely. In the same spirit, the sharing of a polynomial $p(x) = \sum_{i=0}^{n} a_i x^i \in \mathbb{F}_q[x]$ is defined as $[p(x)]_\ell = \sum_{i=0}^{n} [a_i]_\ell x^i$, for $1 \leq \ell \leq k$.

A well-known multiplicative linear secret sharing scheme is the *Shamir secret sharing* based on polynomial interpolation over some finite field. Given a public set of k distinct non-zero field points $\{\alpha_1, \ldots, \alpha_k\} \in \mathbb{F}_q$, the share of P_i is $p(\alpha_i) := [s]_i$ (for $1 \leq i \leq k$), where $p(x) \in \mathbb{F}_q[x]$ is a random polynomial with constant term s and degree $d < k$. In particular, the field size has to be larger than the number of players. Any subset of $d+1$ shares enables recovery of the secret, however, a subset with less than $d+1$ shares does not reveal any information about the secret.

Field elements multiplication. Given a multiplicative secret sharing, one assumes that it has a secure MPC protocol Mult which computes the product of sharing of $a, b \in \mathbb{F}_q$ as

$$[ab] \leftarrow \mathsf{Mult}([a], [b])$$

with constant communication and round complexity. We detail this protocol in the proof of Theorem 2. For example, when dealing with the Shamir secret sharing, the BGW protocol [BGW88] straightly provides $\mathsf{Mult}([a], [b])$ (and requires $k^2 \log_2(q)$ bits to communicate where k is the number of parties, which corresponds to one round of dealing). Thus, the communication complexity, which usually corresponds to the number of bits exchanged during the protocol, can also be expressed in terms of the number of secure multiplications over the field (for every considered protocol, a factor k^2 is hidden in the communication complexity). As an example, if a protocol involves α secure multiplications in parallel and later in the protocol β secure multiplications in parallel, then the round complexity is at most 2 and the communication complexity is $\alpha + \beta$. Up to now, we have been looking at the communication complexity over $\mathbb{F}_q$, but at some point, we may need to work over a field extension. The complexity will still be stated over the base field (i.e. the number of secure multiplication over the base field).

Random element. To generate a sharing of a random value, each party P_i chooses at random a sharing $[r_i]$ of a random element $r_i \in \mathbb{F}_q$ and deals it with the other parties such that at the end P_i gets $\{[r_1]_i, \ldots, [r_k]_i\}$. This defines a share of the sharing $[r] = \sum_{i=1}^{k} [r_i]$ of the random element $r = \sum_{i=1}^{k} r_i \in \mathbb{F}_q$. The communication complexity is $k^2 \log q$ thus bounded by one invocation of the secure multiplication protocol.

Test to zero. Assume that each party owns a share of $a \in \mathbb{F}_q$, and suppose that they would like to compute a share of 1 if $a \neq 0$ or a share of 0 if $a = 0$. For this purpose, there exists a protocol from Damgård et al. [DFK+06] in a constant number of rounds and with $O(\log \log q)$ communication, improved by Nishide and Ohta [NO07] with a protocol in $O(1)$ communication complexity.

Matrix multiplication. Let $A \in \mathbb{F}^{m \times \ell}$ and $B \in \mathbb{F}^{\ell \times n}$ be shared matrices. A naive approach to compute $[AB]$ would be to compute it component-wisely with $n\ell m$ parallel invocation of Mult. The work [MW08] reduces this communication complexity from $O(m\ell n)$ to $O(mn)$. We immediately adapt this protocol for inner-product (with constant communication), and for matrix-vector multiplication with linear communication (in the size of the vector) and constant rounds.

Generation of random non-singular matrices. To generate a set of m random non-singular matrices, consider the following Protocol 1.

Protocol 1: Generation of random non-singular matrices (NonSingularMatrix)

Data: The target number m of drawn non-singular $n \times n$ matrices over $\mathbb{F}_q$, $\mathcal{I} \leftarrow \emptyset$, and a counter $\eta \leftarrow 1$.

Result: A set of m such random non-singular matrices.

1. Parties conjointly construct c couples of random shared matrices $\{R_i, S_i\}_{i\in\{1,\dots,c\}} \in \mathbb{F}_q^{n\times n} \times \mathbb{F}_q^{n\times n}$ in parallel, and securely multiply $R_iS_i := T_i$ with $O(n^2)$ communication and $O(1)$ rounds for each $i \in \{1, \dots, c\}$.
2. Parties reveal T_i and publicly check whether T_i is non-singular, for each $i \in \{1, \dots, c\}$. If it does, then R_i is non-singular and $\mathcal{I} \leftarrow \mathcal{I} \cup \{R_i\}$. If $\#\mathcal{I} \geq m$, go to 3. Otherwise, set $\mathcal{I} \leftarrow \emptyset$, $\eta \leftarrow \eta + 1$ and go back to 1.
3. For m matrices $R_i \in \mathcal{I}$, parties publicly inverse the corresponding T_i and multiply $[S_i]T_i^{-1} = [R_i^{-1}]$.

The security of this protocol relies on the security of previous protocols (generation of random elements, secure multiplication protocol) and on the fact that $GL_n(\mathbb{F}_q)$ is a group for multiplication. Indeed, we then get that $T_i = R_iS_i$ is a uniform random element of $GL_n(\mathbb{F}_q)$ and thus Step 2 of Protocol 1 reveals no information.

Lemma 1. *Protocol 1 to randomly generate m non-singular $n \times n$ matrices over $\mathbb{F}_q$ has expected $O(m\,n^2)$ communication and expected $O(1)$ rounds complexity.*

Proof. In the following, we pick $c = \Theta(m)$, thus the overall communication complexity is $O(\eta\, m\, n^2)$. Since Step 1 is realized in parallel, the overall round complexity is $O(\eta)$. Indeed, Step 1 can be realized via the *Random element* protocol from subsection 2.1 with $O(m\,n^2)$ communication and constant rounds. We show via the Chernoff Bound Theorem that the overall protocol has in fact expected $O(m\,n^2)$ communication and $O(1)$ rounds complexity. We apply the lower tail of the Chernoff Bounds Theorem (see Appendix 10). Consider η trials of $(1 + \frac{3}{q-1})m$ drawings of two random non-singular matrices over $\mathbb{F}_q$. The result of the j-th drawing of the i-th trial is modelled by the random variable $X_{i,j}$:

$$\begin{cases} X_{i,j} = 1 \text{ if both matrices are non-singular;} \\ X_{i,j} = 0 \text{ otherwise.} \end{cases}$$

Let $X = \sum_{i=1}^{\eta} X_i$ such that

$$\begin{cases} X_i = 1 \text{ if } \sum_{j=1}^{(1+\frac{3}{q-1})m} X_{i,j} \geq m; \\ X_i = 0 \text{ otherwise.} \end{cases}$$

One discusses about $\mathbb{P}[X_{i,j} = 1]$. From [Ran93], if $q > 2$,

$$\frac{GL_n(\mathbb{F}_q)}{M_n(\mathbb{F}_q)} = \frac{(q^n-1)(q^n-q)\dots(q^n-q^{n-1})}{q^{n^2}} > \frac{q-2}{q-1},$$

where $M_n(\mathbb{F}_q)$ is the monoid of $n \times n$ matrices over $\mathbb{F}_q$. Otherwise, the probability for a matrix to be invertible is at least $1/4$. See Appendix 8 for further details. Therefore, the probability that in a couple, both random matrices are non-singular is at least $\left(1 - \frac{1}{q-1}\right)^2 > 1 - \frac{2}{q-1}$ (at least $1/16$ over $\mathbb{F}_2$). If one draws $(1 + \frac{3}{q-1})m$ couples (in step 1

of Protocol 1), then the expected number of couples with both non-singular matrices is at least

$$\left(1-\frac{2}{q-1}\right)\left(1+\frac{3}{q-1}\right)m > m.$$

By defining $\mu := \mathbb{E}[X] = \eta\,\mathbb{E}[X_i]$, one has that $\mu > \eta\,m$. Since the $X_{i,j}$'s are independent random variables, so are the X_i's. X_i is a characteristic function for the number of pairs of random non-singular matrices among $(1+\frac{3}{q-1})m$ drawings being less than m or not during the i-th trial. Therefore, X is a sum of independent Bernoulli trials, thus we can apply Chernoff Bounds Theorem (see Appendix 10), and the lower tail gives us

$$\mathbb{P}(X \leq (1-\delta)\mu) \leq e^{-\mu\delta^2/2} \text{ for all } 0 < \delta < 1.$$

Since $\mu > \eta\,m \geq \eta$, one has:

$$\mathbb{P}(X \leq (1-\delta)\eta) \leq e^{-\mu\delta^2/2} \text{ for all } 0 < \delta < 1.$$

Let $\epsilon \in]0,1]$ and choose δ such that $e^{-\frac{\delta^2}{4(1-\delta)}} < \epsilon$. As long as $\eta \geq \frac{1}{2(1-\delta)}$,

$$\begin{aligned} e^{-\frac{\delta^2}{4(1-\delta)}} < \epsilon &\implies e^{-\frac{\eta\,\delta^2}{2}} < \epsilon \\ &\implies e^{-\frac{\mu\,\delta^2}{2}} < \epsilon \\ &\implies \mathbb{P}(X \leq (1-\delta)\eta) < \epsilon \\ &\implies \mathbb{P}(X \leq 1/2) < \epsilon \\ &\implies \mathbb{P}(X = 0) < \epsilon. \end{aligned}$$

Therefore for every ϵ as small as possible, there exists a constant (which depends on δ) such that the probability that the counter η in Protocol 1 exceeds this constant is bounded by ϵ. □

Matrix inversion. Let $A \in \mathbb{F}^{n\times n}$ be a shared non-singular matrix. The inversion protocol from Bar-Ilan and Beaver [BB89] works as follows: parties generate a shared random non-singular matrix $R \in \mathbb{F}^{n\times n}$ with Protocol 1, and securely compute AR with (expected) $O(n^2)$ communication. They reveal and invert it publicly. Then they locally compute $[R](AR)^{-1} = [A^{-1}]$. Since $GL_n(\mathbb{F})$ is a group for multiplication, AR is a uniform random element of $GL_n(\mathbb{F})$ and thus reveals no information. With Lemma 1, this protocol has expected $O(n^2)$ communication and $O(1)$ rounds complexity.

2.2 Power of a matrix

Let $A \in \mathbb{F}_q^{n\times n}$. The following protocol to compute the sharing of powers of A is based on a variant of an iterative products method developed in [BB89] and works as follows. Generate two non-singular shared random matrices M and N in $\mathbb{F}_q^{n\times n}$. Following the above discussion, parties securely compute $[M^{-1}]$ and $[N^{-1}]$. Then they securely compute $[N_1] = [AN^{-1}]$ and $[N_2] = [NAM^{-1}]$, reveal it and multiply them publicly as N_1N_2. They deduce $[A^2] = N_1N_2[M]$ by local computation. This reasoning can be iterated to get a higher shared power of A. Hence, to compute the first m shared power of A, one protocol needs to generate $2m$ non-singular matrices (e.g. with Protocol 1).

One presents the following protocol to compute the first m powers of a matrix $A \in \mathbb{F}_q^{n\times n}$:

$$\{[A^2], \ldots, [A^m]\} \leftarrow \mathsf{Power}([A], m).$$

This protocol 2, for computing the first m powers of a matrix $A \in \mathbb{F}_q^{n\times n}$, has expected $O(m\,n^2)$ communication and $O(1)$ rounds complexity. Step 1 yields to expected $O(m\,n^2)$

Protocol 2: Basic Power Computation Protocol (Power)

Data: $[A]$ with $A \in \mathbb{F}_q^{n\times n}, m \in \mathbb{N}$
Result: $[A^2], \ldots, [A^m]$

1. Parties call to Protocol 1 to get $2m$ non-singular matrices $\{[R_1], \ldots, [R_m]\}$ and $\{[S_1], \ldots, [S_m]\}$, so that they can securely compute $\{[R_1^{-1}], \ldots, [R_m^{-1}]\}$;
2. They securely compute shares of $R_{i-1}A := M_i$ for every $2 \le i \le m$;
3. They securely compute shares of $M_i R_i^{-1} := N_i$ for every $2 \le i \le m$ and $AR_1^{-1} = N_1$;
4. They reveal all the N_i's and compute in the clear $P_j = \prod_{i=1}^{j} N_i$ for every $1 \le j \le m$;
5. They multiply $P_j[R_j] = [A^j]$ for every $1 \le j \le m$.

communication and constant round with Lemma 1. Step 2 and 3 both implies $O(n^2)$ communication for each $2 \le i \le m$ and thus can be done with $O(m\,n^2)$ communication and $O(1)$ rounds (in parallel). Step 4 leads to $O(m\,n^2)$ communication and $O(1)$ rounds (in parallel) for broadcasting the shares. Step 6 is local.

However, this protocol is not perfectly correct because of step 4: if the matrix A is singular, then the protocol reveals its determinant. One will explain later how to manage this leakage by developing and proving a secure protocol.

2.3 Managing the error-probability

The Power protocol to compute powers of $A \in \mathbb{F}^{n\times n}$ in sharing needs, after drawing two random non-singular matrices $M, N \in \mathbb{F}^{n\times n}$ in sharing, to reveal $[MAN^{-1}]$ and so if its determinant is 0 then one learns information about the determinant of A. To overcome this problem, one increases the size of the matrix A by 2 by adding identity block matrices:

$$A^+ = \begin{pmatrix} A & -I_n \\ I_n & 0 \end{pmatrix} \in \mathbb{F}^{2n\times 2n}, \tag{1}$$

so that $\det(A^+) = 1$ regardless the invertibility of A.

Once shared powers of A^+ are computed, parties would like to get shared powers of A. For $0 \le i \le m$, from sharing of $I_{2n}, A^+, {A^+}^2, \ldots, {A^+}^i$, one can deduce sharing of A^i using linear combinations (via linearity of the secret sharing). Indeed, the $n \times n$ top-left block of ${A^+}^i$ denoted by ${A^+}^i_1$ equals

$${A^+}^i_1 = \sum_{j=0}^{\lfloor i/2 \rfloor} \alpha_{i,i-2j} A^{i-2j}$$

with some $\alpha_{i,j} \in \mathbb{F}$. This leads to the following triangular invertible linear system

$$\begin{pmatrix} \alpha_{0,0} & & \\ \vdots & \ddots & \\ \alpha_{m,0} & \cdots & \alpha_{m,m} \end{pmatrix} \begin{pmatrix} I_n \\ A \\ A^2 \\ \vdots \\ A^m \end{pmatrix} = \begin{pmatrix} I_n \\ {A^+}^1_1 \\ {A^+}^2_1 \\ \vdots \\ {A^+}^m_1 \end{pmatrix}, \tag{2}$$

where

$$\alpha_{i,i} = 1, \alpha_{i,i-1} = 0,$$

$$\alpha_{i,0} = \mathbb{1}_{\{i \equiv 0 \bmod 2\}}(i) \text{ for } 0 \leq i \leq m,$$

and

$$\alpha_{i,j} = \alpha_{i-1,j-1} + \alpha_{i-2,j} \text{ for } 2 \leq i \leq m, 1 \leq j \leq m.$$

In the following, we call $\mathcal{A}_m$ the matrix of this system.

Remark 1. Note that the recurrence relationship defining Chebyshev polynomials appears in the inverse of the matrix $\mathcal{A}_m$ in Equation 2, these same polynomials used by [CKP07].

The above discussion yields to the following secure protocol SecPower with the same complexity as Power i.e. expected constant round and $O(m\,n^2)$ communication to securely compute the first m powers of a matrix $A \in \mathbb{F}^{n \times n}$:

$$\{[A^2], \ldots, [A^m]\} \leftarrow \mathsf{SecPower}([A], m).$$

Protocol 3: Secure Power Computation (SecPower)

Data: $[A] \in \mathbb{F}^{n \times n}$, $m \in \mathbb{N}$
Result: $[A^2], \ldots, [A^m]$

1. Parties randomly compute sharing of 0 and 1 to get $[A^+]$ as defined in Equation 1;
2. They invoke $\mathsf{Power}([A^+], m)$;
3. They compute in the clear the matrix of the linear system 2, invert it publicly to locally get shares of the solution of this system.

Theorem 1. *Let $A \in \mathbb{F}^{n \times n}$, and $m \in \mathbb{N}$. Then, SecPower is a secure protocol with perfect correctness for computing the first m power of A with expected $O(m\,n^2)$ communication and $O(1)$ rounds complexity.*

Proof. The correctness relies on the one from Power and of the linear system 2. The round complexity is the same as Power (step 1 and 2 have constant and expected constant rounds in parallel). Step 3 contains no interaction between parties, thus the communication complexity is dominated by step 2, which is doubled compared to Power. We show that the protocol does not leak information about A. The crucial step is the fifth in Power during which values are revealed. Firstly, we call Power with the non-singular matrix A^+, thus the discussion in 2.3 ensures us that no information about the determinant of A leaks in step 5 of Power when A is singular. Moreover, keeping notation from Power, for every $2 \leq i \leq m$ the couple (R_{i-1}, R_i^{-1}) is hiding A^+ because they are elements of $GL_{2n}(\mathbb{F})$, a group under multiplication, thus $N_i = R_{i-1}A^+R_i$ is a random element of $GL_{2n}(\mathbb{F})$. So, revealing N_i does not leak any information about A. At the end, each party has a share for each power of A, thus we shall check that it does not give additional information on A. In the last step of Power, shares of A^j are constructed as $P_j[R_j] = [A^j]$. We have seen in the proof of Power that the N_i's and so the $P_j = \prod_{i=1}^{j} N_i$'s are random elements of $GL_{2n}(\mathbb{F})$, hence sharing of powers of A are independent of each other and of the sharing of A.

□

2.4 Secure matrix multiplication

With the naive approach based on the usual matrix multiplication, one would have a $O(n^3)$ communication complexity (and constant rounds) protocol for secure matrix multiplication. One uses [MW08] for an efficient protocol with $O(n^2)$ communication.

Consider the Shamir secret sharing scheme to illustrate the need of a conjoint refreshing/resharing step. Let $a, b \in \mathbb{F}$ with $|\mathbb{F}| = q$ encoded by $f_a(x)$ and $f_b(x)$ via Shamir secret sharing, where f_a and f_b are two random rational polynomials of degree ℓ over $\mathbb{F}$ with ℓ smaller than k the number of parties. Note that the constant term of $g(x) := f_a(x)f_b(x)$ is ab. Ben-Or et al. [BGW88] noticed two problems with using $g(x)$ to encode the product ab. The first one is that the degree of $g(x)$ is 2ℓ. As long as $q > 2\ell$ interpolation is possible, however, the degree raises during each multiplication, and this limits the number of multiplication that can be handled. The second problem comes from the fact that $g(x)$ is not a random polynomial of degree 2ℓ: for example $g(x)$ is never irreducible. To overcome these two problems, we randomize the coefficients of $g(x)$ and reduce its degree while keeping the constant coefficient unchanged as a refreshing step. Mohassel and Weinreb [MW08] took into account these remarks when stating the following theorem.

Theorem 2. *Let two shared matrices A and B where $A \in \mathbb{F}^{m\times n}, B \in \mathbb{F}^{n\times j}$. Then there exists a secure MPC protocol for computing a secret sharing of the product AB with a constant number of rounds and $O(jm)$ communication.*

One takes advantage to give more details about this theorem, in particular, one proves the following corollary. For the sake of completeness, this corollary states a more general statement by adapting the algorithm of Mohassel and Weinreb to securely compute the inner product, where $\langle\cdot,\cdot\rangle$ denotes the inner product.

We introduce the following protocol InnerProd, and for $a, b \in \mathbb{F}^n$ one writes

$$[\langle a, b\rangle] \leftarrow \mathsf{InnerProd}([a], [b]).$$

Protocol 4: Secure Inner Product Protocol (InnerProd)

Data: $[a], [b] \in \mathbb{F}^n$
Result: $[\langle a, b\rangle]$

1. Each party P_t locally computes $\tilde{c}_t = \sum_{i=1}^{n} [a_i]_t [b_i]_t$ for $1 \le t \le k$;
2. Each party P_t shares $\tilde{c}_t$ in $[\tilde{c}_t]_1, \ldots, [\tilde{c}_t]_k$ and sends $[\tilde{c}_t]_h$ to P_h for $1 \le h \le k$;
3. Each party P_t deduces a share of the secret by computing $c_t = \sum_{h=1}^{k} \lambda_h [\tilde{c}_t]_h$, where $(\lambda_1, \ldots, \lambda_k)$ is the public recombination vector for the sharing scheme.

Corollary 1. *Given two shared vectors a and b in $\mathbb{F}^n$, InnerProd is a secure protocol for computing a secret sharing of the inner product $\langle a, b\rangle$ with a constant number of rounds and $O(1)$ communication.*

Proof. Protocol InnerProd has communication complexity in $O(1)$: the only communication occurs during step 2 where each party sends a share of its secret to other parties. The correctness of InnerProd follows from the proof of Corollary 1. We prove the security of InnerProd). Parties may learn information about a or b during the second step when they communicate with each other. From step 1, each party P_t computes a secret $\tilde{c}_t$ and sends to the other a share of it. This sharing created by P_t is independent of all the other sharings created by the other parties. Thus, P_t does not learn more information about a or b by receiving shares of other parties. Let's prove the correctness of Protocol 2.4 when

dealing with the Shamir secret sharing scheme. Let $a, b \in \mathbb{F}_q^n$, and define $f_{a_j}(x)$ and $f_{b_j}(x)$ rational polynomials of degree ℓ respectively encoding a_j and b_j for $1 \leq j \leq n$ via the Shamir secret sharing. Assume that $k \geq 2\ell + 1$ where k is the number of parties. Consider that party P_i holds the values $f_{a_j}(i)$ and $f_{b_j}(i)$ for every $j \leq n$. Define

$$f_{a_j b_j}(x) := f_{a_j}(x) f_{b_j}(x) = \alpha_{2\ell}^j x^{2\ell} + \cdots + \alpha_1^j x + a_j b_j$$

for every $j \leq n$, and $f_{\langle a,b \rangle}(x) := \sum_{j=1}^n f_{a_j b_j}(x)$. Thus

$$f_{\langle a,b \rangle}(i) = \sum_{j=1}^n f_{a_j b_j}(i) = \sum_{j=1}^n f_{a_j}(i) f_{b_j}(i)$$

for $1 \leq i \leq k$. This yields the following Vandermonde linear system:

$$A \begin{pmatrix} \langle a, b \rangle \\ \sum_{i=1}^n \alpha_1^i \\ \vdots \\ \sum_{i=1}^n \alpha_{2\ell}^i \end{pmatrix} = \begin{pmatrix} f_{\langle a,b \rangle}(1) \\ f_{\langle a,b \rangle}(2) \\ \vdots \\ f_{\langle a,b \rangle}(2\ell+1) \end{pmatrix},$$

where the ij-th coefficient of the $(2\ell+1) \times (2\ell+1)$ matrix A is i^{j-1}. Clearly, A is invertible and we denote the first row of its inverse as $(\lambda_1, \ldots, \lambda_{2\ell+1})$, the fixed recombination vector. Then

$$\langle a, b \rangle = \sum_{i=1}^{2\ell+1} \lambda_i f_{\langle a,b \rangle}(i).$$

Now one refreshes: P_i shares the value $f_{\langle a,b \rangle}(i)$ by choosing a random polynomial $g_i(x)$ of degree ℓ such that $g_i(0) = f_{\langle a,b \rangle}(i)$. They give the value $g_i(j)$ to P_j for every party. Therefore, each party P_j can compute their share of $\langle a, b \rangle$ via the polynomial $G(x) = \sum_i \lambda_i g_i(x)$ of degree ℓ: they locally compute the linear combination $G(j) = \sum_i \lambda_i g_i(j)$. $G(1), \ldots, G(k)$ determine $G(0) = \langle a, b \rangle$ (via polynomial interpolation). □

Remark 2. Note that the proof of Corollary 1 can be adapted to prove Theorem 2. We can derive a protocol for secure matrix multiplication by invoking n^2 times in parallel the protocol InnerProd. This is the algorithm of Mohassel and Weinreb [MW08], where each party P_t first locally compute $[A]_t[B]_t$ before resharing this matrix with other parties, whence $O(n^2)$ communication. Note also that secure matrix-vector multiplication can be computed with $O(n)$ communication complexity via n invocation of InnerProd.

Now, we state a simple but important fact about InnerProd which will be mainly used in the following to batch communication:

$$\begin{aligned} &\mathsf{InnerProd}([a], [b]) + \mathsf{InnerProd}([c], [d]) \\ &= \mathsf{InnerProd}(\begin{pmatrix} [a] \\ [c] \end{pmatrix}, \begin{pmatrix} [b] \\ [d] \end{pmatrix}). \end{aligned} \tag{3}$$

In particular, we can apply this batching for the sum of matrix products, i.e. if we are dealing with $n \times n$ matrices the secure computation of the sum implies $O(n^2)$ communication.

2.5 Moore-Penrose pseudo-inverse

The existence of solution(s) of a linear system $Ax = b$ over a field $\mathbb{K}$ only depends on the rank of the concatenated matrix $A \| b$. The rank of A can be computed via the characteristic polynomial of $G := A^T A$, the so-called *Gram matrix* of A. To apply this, we have to

avoid non-trivial intersection between some subspaces and their orthogonals. However, it may appear over fields with positive characteristic as explained in [CKP07]. For this purpose, we will use a work of Mulmuley [Mul86] for computing rank over arbitrary field along with the paper from Diaz-Toca, Gonzalez–Vega and Lombardi [DGL05] for solving linear system. A previous work of [CKP07] gives a probabilistic algorithm without perfect correctness. By taking a sufficient large extension, we can get rid of this probability.

In the following, let $\mathbb{K}$ be a field. Let $V \subset \mathbb{K}^i$ be a subspace, then one defines the set $V^\perp := \{u \in \mathbb{K}^i \mid \langle u, v\rangle = 0 \ \forall v \in \mathbb{K}^i\}$. For $\mathbb{K} = \mathbb{C}$ or $\mathbb{R}$, the subspace $V^\perp \cap V$ is trivial. However, this result is not true over a field of positive characteristic.

The next lemma characterizes the rank of a matrix via the coefficients of the characteristic polynomial of its Gram-matrix. In the following, one identifies a matrix $A \in \mathbb{K}^{n\times n}$ as a linear endomorphism of $\mathbb{K}^n$.

Lemma 2. *Let $A \in \mathbb{K}^{m\times n}$, and G its Gram-matrix. Assume that $(Im\, A)^\perp \cap Im\, A = \{0\}$ and $(Im\, A^T)^\perp \cap Im\, A^T = \{0\}$. Consider $\chi_G(x) = \sum_{i=1}^{m} a_i x^{m-i} + x^m$ the characteristic polynomial of G. Then $rank\, A = max_{1\le i\le m}\{i \mid a_i \neq 0\}$.*

Proof. [Mul86, CKP07] □

For a general matrix over a finite field, the conditions on the vector spaces intersection do not always hold. However, there exists a matrix that always satisfies these conditions and has the same rank as the initial matrix. We use a work of Mulmuley [Mul86] that proposes a technique to compute the rank over an arbitrary field. He considered the transcendental field extension $\mathbb{K}(x)$. Over this extension, for every matrix $A \in \mathbb{K}^{m\times n}$, the matrix $\mathrm{diag}(1, x, \ldots, x^{m-1})A := DA \in \mathbb{K}^{m\times n}(x)$ satisfy $(\mathrm{Im}\, DA)^\perp \cap \mathrm{Im}\, DA = \{0\}$ and then lemma 2 holds for DA. This yields a method for computing the rank of A (with perfect correctness) since it's equal to the rank of DA.

We now introduce the notion of pseudo-inverse. A pseudo-inverse of a matrix $A \in \mathbb{K}^{m\times n}$ is a matrix $X \in \mathbb{K}^{n\times m}$ that exists for a class of matrices larger than the class of non-singular matrices, and reduces to the classical inverse when A is non-singular. In this paper, we consider the class of Moore-Penrose pseudo-inverse determined by the following four properties, also known as the Penrose equations [Pen55]:

$$AXA = A,\ XAX = X,\ (AX)^T = AX,\ (XA)^T = XA.$$

In this case, we denote X by $A^\dagger$. $A^\dagger$ exists if and only if $\mathrm{rank}(AA^T) = \mathrm{rank}(A^TA) = \mathrm{rank}(A)$, and existence implies uniqueness.

Our motivation for studying this pseudo inverse is the following: let $b \in \mathbb{K}^n$ then the system $Ax = b$ has at least one solution if and only if $AA^\dagger b = b$. In this case, all the solutions are given by

$$x = A^\dagger b + (I_n - A^\dagger A)v$$

with arbitrary $v \in \mathbb{K}^n$. If a solution exists, then either the solution is unique when $A^\dagger A$ has full column rank (i.e. $A^\dagger A = I_n$) or there exists infinitely many solutions when $A^\dagger A$ does not have full column rank. We mention other motivations such as the least squares problem or even the minimum norm problem for a linear system.

3 (Matrix)-Polynomial Evaluation

Based on Paterson and Stockmeyer work [PS73], a preliminary protocol can be unrolled for evaluating a public polynomial of degree d into a shared element running with a constant number of rounds and $O(\sqrt{d})$ communication. This protocol leaks information on the secret with probability $1/q$. Later, Cramer, Kiltz and Padro [CKP07] presented a secure

and perfectly correct protocol with constant rounds and $O(d)$ communication. One starts by adapting the protocol following the technique [PS73] to get a protocol with expected $O(t)$ rounds and $O(t\,d^{1/t})$ communication for any $t \in \mathbb{N}$. Then one uses [CKP07] and an idea of Bar-Ilan and Beaver to obtain a perfectly correct protocol with the same expected complexity. For any $t \in \mathbb{N}$, we obtain the following result for the evaluation of a degree-t polynomial with a $n \times n$ matrix:

Table 1: Complexity and correctess of $n \times n$ matrix secure evaluation of degree-t polynomial

	round complexity	comm. complexity	correctness
based on [PS73]	$O(1)$	$O(n^3\sqrt{d})$	$1/q$
[CKP07]	$O(1)$	$O(n^3 d)$	*perfect*
Our work	expected $O(t)$	expected $O(n^2\,t\,d^{1/t})$	*perfect*

The technique of [PS73] can be applied to [CKP07], leading to a communication complexity in $O(n^3\sqrt{d})$. When the polynomial is also shared, then [CKP07] has also $O(n^3 d)$ communication complexity since their protocol for iterative powers requires $O(n^3 d)$ communication. The generalization of our work to shared polynomials does not increase the communication complexity.

Firstly, we will detail our non-perfectly-correct protocol. Let $s \in \mathbb{F}$ shared via the Shamir secret sharing scheme, p a public polynomial of degree d, and $t \in \mathbb{N}$. If the polynomial was shared, then our construction could be easily adapted. We could directly apply $\mathsf{SecPower}$ to get a protocol with $O(d)$ communication, but we follow another path to get a better complexity. If one computes shares of power of s via Power, one would leak information when $s = 0$ (i.e., when s is not invertible since one applies the iterative products' method from Bar-Ilan Beaver, see the discussion in 2.3). That's why one apply Power to $[s + r]$ with a random $r \in \mathbb{F}_q$ to randomize the input and reduce its probability to be non-zero to $1/q$. For this purpose, parties conjointly generate a random element $r \in \mathbb{F}^*$ and reveal r. Let $m = \lceil d^{1/t} \rceil$.

For the baby steps, parties first compute the sharings $[s + r], \ldots, [(s + r)^m]$ by invoking $\mathsf{Power}([s + r], m)$. Then they locally compute $[s^2], \ldots, [s^m]$ via the commutativity of $\mathbb{F}_q$ and the linearity of the secret sharing scheme. This is done recursively using Newton binomial

$$[(s + r)^i] = \sum_{k=0}^{i} [\binom{i}{k} s^k r^{i-k}] = \sum_{k=0}^{i} \binom{i}{k} r^{i-k} [s^k].$$

For the giant steps, parties proceed in expected $O(t)$ rounds of communication. Once they know $[s^{m^j}]$ for $1 \leq j \leq t - 1$, they can invoke $\mathsf{Power}([s^{m^j} + r], m)$ with an expected constant number of rounds to get $[(s^{m^j} + r)^2], \ldots, [(s^{m^j} + r)^m]$ and locally deduce $[s^{2m^j}], \ldots, [s^{m^{j+1}}]$ via Newton binomial as in baby-step:

$$[(s^{m^j} + r)^i] = \sum_{k=0}^{i} \binom{k}{i} r^{i-k} [s^{km^j}]. \tag{4}$$

This procedure leads to an expected number of $O(t)$ rounds and expected $O(t \cdot d^{1/t})$ communication. This step is secure as long as $s + r \not\equiv 0 \bmod q$ which happens with probability $1 - 1/q$.

Note that a degree-d polynomial $p(x)$ can be decomposed as follows:

$$p(x) = \sum_{i=0}^{m^{t-1}} x^{im} p_i(x), \tag{5}$$

where p_i are public polynomials of degree at most $m - 1$. Let's denote $p_i(x) = a_{im} + a_{im+1}x + \cdots + a_{(i+1)m-1}x^{m-1}$. Therefore parties can invoke $\mathsf{InnerProd}([a], [b])$ with

$$a = (a_0, a_1 s, a_2 s^2, \ldots, a_{m^t} s^{m^t}) \tag{6}$$

$$b = (\underbrace{1, \ldots, 1}_{m \text{ times}}, \underbrace{s^m, \ldots, s^m}_{m \text{ times}}, \underbrace{s^{2m}, \ldots, s^{2m}}_{m \text{ times}}, \ldots, \underbrace{s^{m^t - m}, \ldots, s^{m^t - m}}_{m \text{ times}}, s^{m^t}), \tag{7}$$

since $[a]$ and $[b]$ can been locally deduced from previously computation. In particular, when iterative powers of s have been computed, this evaluation requires $O(1)$ calls to the secure multiplication protocol.

Remark 3. If the polynomial $p(x)$ was shared between the parties, before calling to $\mathsf{InnerProd}([a],[b])$, we should compute $[a]$ which would add d calls to the secure multiplication protocol and lead to $O(n^2 d)$ communication. However, the protocol $\mathsf{InnerProd}$ can be generalized to sum with three terms $\sum_{i=0}^{d}[a_i][s^i][b]_i$ as long as the number of participants is larger than 3ℓ, where ℓ is the degree of the encoding polynomials for the Shamir secret sharing scheme.

3.1 Secure polynomial evaluation

In the following and for the sake of generality, one considers the more general case of matrix-polynomial evaluation. Let $p(x) \in \mathbb{F}_q[x]$ be a rational polynomial of degree d with public (shared) coefficients and a shared matrix $[A] \in \mathbb{F}_q^{n\times n}$.

One modifies the previous approach to get a perfectly correct protocol with an expected $O(t)$ number of rounds and expected $O(n^2 t d^{1/t})$ communication for computing $[p(A)] = \sum_{i=0}^{m^{t-1}} x^{im} p_i(x)$ following the decomposition from Equation 5, with $m = \lceil \sqrt[t]{d} \rceil$ and $p_i(x)$ public (shared) polynomial of degree at most $m-1$.

Protocol 5: Secure Polynomial Evaluation ($\mathsf{PolyEval}$)

Data: $[A] \in \mathbb{F}_q^{n\times n}$, $t \in \mathbb{N}$, $m = \lceil d^{1/t} \rceil$, and $p \in \mathbb{F}_q[x]$ of degree d
Result: $[p(A)]$
For $0 \le i \le t-1$:

1. Parties invoke $\mathsf{SecPower}([A^{m^i}], m)$ and derive a sharing of $A^{2m^i}, A^{3m^i}, \ldots, A^{m^{i+1}}$ with Equation 4;
2. For each $[A^{im}][p_i(A)]$, they call to $\mathsf{InnerProd}([a],[b]) = [p(A)]$ with $[a]$ and $[b]$ from Equations 6 and 7.

If p is shared among the parties, then step 2 is modified as explained in Remark 3: each inner product is substituted by a sum of three terms.

Theorem 3. *Let $A \in \mathbb{F}_q^{n\times n}$ be a shared matrix, $p \in \mathbb{F}[x]$ a public polynomial of degree d, and $t \in \mathbb{N}$. Then $\mathsf{PolyEval}$ is a secure with perfect correctness protocol to evaluate $p(A)$ with expected $O(t)$ rounds and $O(n^2 t d^{1/t})$ communication.*

Proof. The correctness of Protocol 5 relies on the correctness of $\mathsf{SecPower}$. We now prove the complexity of Protocol 5. The first step of the protocol can be done with expected $O(t)$ rounds and $O(n^2 t d^{1/t})$ communication: each invocation of $\mathsf{SecPower}$ requires $O(n^2 d^{1/t})$ communication. The step 2 is done with $O(n^2)$ secure multiplication using the batching property 2. Finally, we prove the security of Protocol 5. First, notice that parties never reconstruct a secret value. For the first three steps, we have seen that $\mathsf{SecPower}$ is secure and that knowing shares of different powers of A does not reveal information about A (see proof of security of $\mathsf{SecPower}$). The security of the last step follows from the one of $\mathsf{InnerProd}$. □

4 Computation of the Characteristic Polynomial

Leverrier's Lemma and Preparata-Sarwate Algorithm. In 1840, Le Verrier published a method to compute the characteristic polynomial $\chi_A(X) = X^n + \sum_{i=1}^{n} X^{n-i} d_i$ of $A \in \mathbb{F}_q^{n\times n}$ summarized in the following lemma. It was redeveloped by many authors including Faddeev. We use this latter version. Define by recurrence the following sequence of matrices $(A_i)_{0\le i\le n-1} \in$

$\mathbb{F}_q^{n\times n}$:

$$A_0 = A;$$
$$A_i = A\left(A_{i-1} - \frac{1}{i}\mathrm{tr}(A_{i-1})I_n\right) \text{ for } 1 \le i \le n-1.$$

Then it holds that $d_i = -\frac{1}{i}\mathrm{tr}(A_{i-1})$ for $1 \le i \le n$ where tr is the trace operator.

Lemma 3. ***(Leverrier's Lemma).*** *The coefficients $d_1, d_2, \cdots, d_n$ of the characteristic polynomial of the matrix $A \in \mathbb{F}_q^{n\times n}$ satisfy*

$$\begin{pmatrix} 1 & & & & \\ t_1 & 2 & & & \\ t_2 & t_1 & 3 & & \\ \vdots & \vdots & \vdots & \ddots & \\ t_{n-1} & t_{n-2} & t_{n-3} & t_1 & n \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{pmatrix} = - \begin{pmatrix} t_1 \\ t_2 \\ t_3 \\ \vdots \\ t_n \end{pmatrix} \tag{8}$$

where $t_j := tr(A^j)$.

Note that if $\mathbb{F}_q$ has characteristic greater than n, then the matrix is guaranteed to be non-singular since its determinant is $\prod_{i=1}^{n} i \not\equiv 0 \bmod q$.

Preparata and Sarwate [PS78] introduced a new idea improving Leverrier and Faddeev's method. It relies on the computation of the trace of the product AB for $A = (a_{i,j})_{1\le i,j\le n}, B = (b_{i,j})_{1\le i,j\le n} \in \mathbb{F}^{n\times n}$ via

$$\mathrm{tr}(AB) = \sum_{1\le k,l\le n} a_{k,l}b_{l,k} = \sum_{1\le k\le n} \langle a_k^T, b^k\rangle \tag{9}$$

with a_k the k-th line of A and b^k the k-th column of B. Thus, the complete matrix product AB is not necessary. Hence computing $[tr(AB)]$ only requires one invocation of InnerProd, since the n inner products from Equation 9 can be batched via Equation 3. This yields to $O(1)$ secure matrix multiplication. Moreover, in order to apply Leverrier's Lemma, Preparata and Sarwate used a baby-step giant-step approach: one only needs to precompute $A^2, A^3, \ldots, A^m$ and $A^{2m}, A^{3m}, \ldots, A^{m^2}$ for $m = \lceil\sqrt{n}\rceil$. Indeed, to get $t_2, t_3, \ldots, t_n$, we can compute $t_{i+jm} = \mathrm{tr}(A^{i+jm}) = \mathrm{tr}(A^i A^{jm})$ with $O(1)$ communication. See Appendix 7 for the complete protocol.

4.1 Secure protocol for the characteristic polynomial

Let $A \in \mathbb{F}_q^{n\times n}$ be a shared matrix, $\mathbb{F}_q$ of characteristic greater than n (denoted as $\mathrm{char}(\mathbb{F}_q)$ in the following), $m = \lceil\sqrt{n}\rceil$, and A^+ the augmented matrix defined in Equation 1. Then, based on previous secure protocols and on the above discussion on characteristic polynomial computation, we propose the following protocol.

Theorem 4. *Let $A \in \mathbb{F}_q^{n\times n}$ be a shared matrix with $char(\mathbb{F}_q) > n$. Then PolyChar is a secure protocol with perfect correctness with expected $O(n^{2.5})$ communication complexity and $O(1)$ rounds.*

To compare with the work [CKP07] who achieved a protocol with small error probability $O(n^2/q)$ and $O(m^4 + m^2 n)$ multiplications.

Proof. The correctness and security of Protocol 6 follows from the one of SecPower, InnerProd, and the linear system 8. About this linear system, we have seen in *Matrix inversion* 2.1 that our inversion protocol is secure, and the security of the matrix-vector product relies on the one of InnerProd. Moreover, notice that parties never reconstruct a secret value, and knowing shares of different powers of A does not reveal information about A (see proof of security of SecPower).

Complexity of step 1 and 2 is equivalent to the one of SecPower, with expected $O(mn^2) = O(n^{2.5})$ communication and constant rounds. During step 3, for each trace, parties can compute a sharing of it by invoking InnerProd with Equation 9 and the batching Equation 3. This yields to $O(1)$ communication complexity in constant rounds for each trace, thus an overall of $O(n)$ communication. Since the computation of all the traces is done in parallel, it yields to an overall constant number of rounds. In step 4, they securely compute the inverse of the Toeplitz matrix via *Matrix inversion* 2.1 ($O(n^2)$ communication) and then apply the secure matrix-vector multiplication as in Remark 2 with $O(n)$ communication (and a constant number of rounds). □

Protocol 6: Secure Computation of the Characteristic Polynomial (PolyChar)

Data: $[A]$ with $A \in \mathbb{F}_q^{n\times n}$, $\text{char}(\mathbb{F}_q) > n$, and $m = \lceil\sqrt{n}\rceil$
Result: $[\chi_A(x)]$ the shared characteristic polynomial of A

1. Baby-Step: Parties invoke SecPower$([A], m)$;
2. Giant-Step: Once having shares of A^m, parties invoke SecPower$([A^m], m)$. Let $B = A^m$ for the sake of clarity;
3. Parties compute their share of each trace (in parallel) by invoking once InnerProd for each trace via
$$[\text{tr}(A^{i+mj})] = [\text{tr}(A^i B^j)] = \sum_{k,l} [a^i_{k,l}][b^j_{l,k}]$$
with $0 \le i \le m-1$, $0 \le j \le m$, a^i and b^j respective coefficients of A^i and B^j;
4. Parties securely resolve the linear system 8 by securely computing the inverse of the Toeplitz matrix and by doing the secure matrix-vector multiplication invoking n times InnerProd.

Remark 4. Protocol 6 can be generalized to finite fields of any characteristic by using in step 4 the method from Schönhage mentioned in Appendix 9 instead of Leverrier's Lemma. The generalized protocol yields to $O(n^{3.5})$ communication.

5 Computation of Moore-Penrose Pseudo-Inverse

We have already discussed solving a shared non-singular linear system with $O(n^2)$ secure multiplications (see *Matrix inversion* 2.1). Thus, our protocol PolyChar along with Cayley-Hamilton is not an improvement in terms of complexity. However, for a matrix $A \in \mathbb{F}_q^{m\times n}$ of rank r, the generalized Moore-Penrose pseudo-inverse (see subsection 2.5 and [Pen55]) can be obtained through the characteristic polynomial of its generalized Gram-matrix. We first work over a finite field $\mathbb{F}_q$ with $\text{char}(\mathbb{F}_q) > m$. A generalization over finite field of any positive characteristic follows from a work from Schönhage [Sch93] (see Appendix 9 and Remark 4). Our work is based on a paper from Mulmuley [Mul86] which was later expanded by Diaz-Toca et al. [DGL05]. The latter started by working over the real or complex field. With the purpose of generalizing their result for an arbitrary field, they followed the idea of Mulmuley and introduced a parameter t to work over a transcendental field extension for the reason mentioned in subsesction 2.5. We make the choice to work over a sufficient large extension instead of a transcendental extension. The following table compares our result with previous works with $t \in \mathbb{N}, k \in \mathbb{N}$ such that $k > r(n+m-2r)+1$:

Table 2: Complexity and correctness of Moore-Penrose pseudo-inverse secure computation

	round complexity	comm. complexity	correctness
[CD01]+[MW08]	$O(1)$	$O(n^3)$	$\Theta(\texttt{poly}(n)/q)$
[CKP07]	$O(1)$	$O(n^4 + mn^2)$	$\Theta(\texttt{poly}(n)/q)$
[MW08]	$O(t)$	$O(tn^{2+1/t})$	$\Theta(\texttt{poly}(n)/q)$
Our work	expected $O(1)$	expected $O(k(n^{2.5} + m^{2.5} + n^2 m^{0.5}))$	*perfect*

Let $k \in \mathbb{N}$ such that $k > r(n+m-2r)+1$, and let $p(x) \in \mathbb{F}_q[x]$ be irreducible of degree k. Let $\mathbb{F}$ be the splitting field of p and ζ a root of p. By taking the intersection of all sub-extensions of $\mathbb{F}/\mathbb{F}_q$ containing ζ, one can consider $\mathbb{F}[x]/(p(x)) \simeq \mathbb{F}_q(\zeta) \simeq \mathbb{F}_{q^k}$ which is the smallest field extension containing ζ. However, one has to reveal k and by choosing it in this way one reveals information

about the rank of A. Thus, one chooses k such that $k > m(n+m)+1 > r(n+m-2r)+1$. The first step consists of detailing the complexity of the secure matrix multiplication over $\mathbb{F}_q(\zeta)$. We define by $\mathrm{diag}(u_1, ..., u_\ell) \in \mathbb{F}_q^{\ell\times\ell}$ the diagonal matrix of size ℓ with coefficient $u_1, ..., u_\ell$.

Lemma 4. *Let $A \in \mathbb{F}_q(\zeta)^{m\times n}$ and $B \in \mathbb{F}_q(\zeta)^{n\times m}$ be two shared matrices, then a sharing of the product AB can be securely computed with expected constant round and $O(km^2)$ communication.*

Proof. Over $\mathbb{F}_q(\zeta) \simeq \mathbb{F}_{q^k}$, the sharing computation of AB requires $O(m^2\log_2(q^k)) = O(km^2\log_2(q))$ bits communication (with InnerProd over $\mathbb{F}_{q^k}$), which corresponds to $O(km^2)$ call to the secure multiplication protocol (defined over the base field $\mathbb{F}_q$) or in other words $O(km^2)$ communication. A more detailed proof would be to decompose A and B as

$$A = \sum_{i=0}^{k-1} A_i\zeta^i,\ B = \sum_{i=0}^{k-1} B_i\zeta^i \tag{10}$$

where $A_0, \ldots, A_{k-1} \in \mathbb{F}_q^{m\times n}$ and $B_0, \ldots, B_{k-1} \in \mathbb{F}_q^{n\times m}$. Then

$$AB = \sum_{v=0}^{k-1}\zeta^v \sum_{y=0}^{k-1} \sum_{\substack{0\le z\le k-1\\ y+z\equiv v \bmod k}} A_yB_z := \sum_{j=0}^{k-1}\zeta^j\beta_j,$$

with $\beta_0, \ldots, \beta_{k-1} \in \mathbb{F}_q^{m\times m}$. The double sum β_j can be securely computed with $O(n^2)$ communication via InnerProd and Equation 3 where the batching is applied to the sum of matrix product. Finally, we get an overall communication complexity of $O(km^2)$. □

Remark 5. As a direct implication of the Lemma 4, the communication complexity of SecPower, PolyEval, and PolyChar are all affected by a factor k when working over an extension of degree k.

Let $Q_n = \mathrm{diag}(1, \zeta, \zeta^2, \ldots, \zeta^{n-1}) \in \mathbb{F}_q(\zeta)^{n\times n}$ and $Q_m = \mathrm{diag}(1, \zeta, \zeta^2, \ldots, \zeta^{m-1}) \in \mathbb{F}_q(\zeta)^{m\times m}$, and define

$$A^0 := Q_n^{-1}A^TQ_m \in \mathbb{F}_q(\zeta)^{n\times m}.$$

Consider the generalized Gram-matrix $G := AA^0 \in \mathbb{F}_q(\zeta)^{m\times m}$ and its characteristic polynomial $\chi_G(x) \in \mathbb{F}_q(\zeta)[x]$. One has that $\chi_G(x) = (-1)^m x^m\eta(1/x)$ where $\eta(x) = \det(I_m + xG) = 1 + a_1(\zeta)x + \cdots + a_m(\zeta)x^m$, with $a_0 = 1$ and $a_2(x), \ldots, a_m(x)$ are Laurent polynomials. Given the previous notations, we are able to introduce the result from Diaz-Toca et al. [DGL05] that generalize the work due to Decell [jD65] for fields of positive characteristic to compute the generalized Moore-Penrose pseudo-inverse introduced in subsection 2.5.

Theorem 5. *(generalized Moore-Penrose pseudo-inverse). Let r be the rank of $A \in \mathbb{F}_q(\zeta)^{m\times n}$. Then the Moore-Penrose pseudo-inverse of A is given by*

$$A^\dagger = a_r^{-1}(a_{r-1}I_n - a_{r-2}A^0A + \cdots + (-1)^{r-1}(A^0A)^{r-1})A^0.$$

This yields to the following protocol to securely compute $A^\dagger$.

Theorem 6. *Let $\mathbb{F}_q$ be a finite field, $A \in \mathbb{F}_q^{m\times n}$ of rank r and consider the field extension $\mathbb{F}_{q^k}$ where $k > m(n+m)+1$. Then Protocol 7 is a secure protocol with perfect correctness to compute the Moore-Penrose pseudo-inverse with expected constant round and $O(k(n^{2.5} + m^{2.5} + n^2m^{0.5}))$ communication as long as $\mathrm{char}(\mathbb{F}_q) > n$.*

Proof. Correctness of Protocol 7 follows from Theorem 5, the correctness of InnerProd (when using Lemma 4), PolyChar, SecPower, PolyEval and *Test to zero* protocol from [NO07]. The security of genInverse relies on the security of these subprotocols.

Let's prove the complexity. The first step requires $O(k(n^2 + m^2))$ communication thanks to Lemma 4 and the fact that Q_n and Q_m are public (so we use the linearity of the sharing). Theorem 4 with Remark 5 implies that step 2 requires $O(km^{2.5})$ communication. Still with Remark 5, step 3 yields to $O(kn^{2.5})$ communication, and step 4 to $O(kn^2m^{1/2})$ (Theorem 3 with $t = 2$). □

Protocol 7: Secure Computation of the generalized Moore-Penrose inverse (genInverse)

Data: $[A]$ with $A \in \mathbb{F}_q(\zeta)^{m\times n}$ of rank r
Result: $[A^\dagger]$

1. Parties locally compute a sharing of A^0 and then securely and conjointly compute a sharing of $G = AA^0$ and of A^0A with Lemma 4;
2. Parties invoke PolyChar($[G]$);
3. Parties invoke SecPower($[A^0A], \sqrt{n}$) and then SecPower($[(A^0A)^{\sqrt{n}}], \sqrt{n}$);
4. Parties compute a sharing of $p_r(A^0A) := \sum_{i=0}^{r-1}(-1)^i a_{r-1-i}(A^0A)^i$ with PolyEval where p_r is shared;
5. Parties securely compute $[a_r^{-1}]$;
6. Parties compute $[a_r^{-1}][p_r(A^0A)][A^0] = [A^\dagger]$ with InnerProd.

We give additional details on the different step of Protocol 7. In particular, we detail the invocation of PolyChar over the extension field. In step 1 of Polychar, the augmented matrix G^+ is obviously defined over the extension. SecPower invokes Power where shared non-singular matrices are drawn at random over the field extension. One refers to the discussion 2.3 about the probability of drawing a non-singular matrix over $\mathbb{F}_{p^k}$. The security is preserved because $GL_{2m}(\mathbb{F}_p(\zeta))$ is a group for multiplication. The last step of SecPower yields shared powers of G from shared powers of G^+ by solving the linear system 2: the public matrix $\mathcal{A}_{\sqrt{m}}$ is inverted, and the parties locally compute

$$\sum_{i=0}^{k-1} \zeta^i \mathcal{A}^{-1}_{\sqrt{m}} \left[\begin{pmatrix} I_m \\ {G^+}_{1,i} \\ {G^+}^2_{1,i} \\ \vdots \\ {G^+}^{\sqrt{m}}_{1,i} \end{pmatrix} \right] = \left[\begin{pmatrix} I_m \\ G \\ G^2 \\ \vdots \\ G^{\sqrt{m}} \end{pmatrix} \right],$$

where ${G^+}^j_{1,i} \in \mathbb{F}_p^{m\times m}$ is the $m \times m$ top-left block of G^{+j}_i which is the $(i+1)$-th term in the decomposition of G^{+j} as a polynomial in ζ, where G^{+j} is the j-th power of G^+. This is realized with expected constant rounds and $O(km^{2.5})$ communication. Take a look at the third step of PolyChar. For every $0 \le i \le \sqrt{m}-1$ and $0 \le j \le \sqrt{m}$, we consider the polynomial in ζ $G^{i+j\sqrt{m}} = \sum_{v=0}^{k-1} \zeta^v G_v^{i+j\sqrt{m}}$ (with all the $G_v \in \mathbb{F}_q^{2m\times 2m}$). Define $H := G^{\sqrt{m}}$, then via the linearity of the trace and of the secret sharing

$$[\mathrm{tr}(G^{i+j\sqrt{m}})] = [\mathrm{tr}(G^iH^j)] = \sum_{v=0}^{k-1}\zeta^v \sum_{y=0}^{k-1} \sum_{\substack{0\le z\le k-1 \\ y+z\equiv v \bmod k}} \left[\mathrm{tr}\left(G_y^i H_z^j\right)\right].$$

We have seen that a trace over $\mathbb{F}_q$ can be computed with one invocation of InnerProd, and by batching these two double sums of inner products with Remark 2, this yields to $O(k)$ communication for trace of each power of G and an overall of $O(km)$ communication (in parallel). Finally, they securely solve the linear system 8 by adapting the Bar-Ilan and Beaver's protocol for secure *Matrix inversion* in subsection 2.1 with $O(km^2)$ communication. The matrix-vector multiplication is also adapted and requires $O(km)$ communication.

Step 4 requires developing a protocol for securely computing the rank of a shared matrix. By defining $g_i := (-1)^m a_i(\zeta)$, one has that $\chi_G(X) = \sum_{i=0}^m g_i X^{m-i}$ with $g_0 = 1$. From each sharing of g_i for $i \in \{0,\dots,m\}$, the parties can compute a sharing of h_i defined as:

$$h_i = \begin{cases} 1 & \text{if } g_i \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

This can be done using the protocol *Test to zero* 2.1 in parallel with $O(1)$ rounds and $O(k)$ invocations of the secure multiplication protocol for each coefficient (and thus an overall of $O(1)$ rounds and $O(mk)$ communication). Let define

$$P_j(X) := \sum_{i=0}^{j-1} a_{j-1-i} X^i$$

for $j \in \{2, \ldots, m\}$. Parties can locally compute a sharing of $P_j(X)$ for each j. Then, one can readily see that the Moore-Penrose pseudo-inverse is equal to

$$a_r^{-1} \beta_m(A^0 A) A^0 = a_r^{-1} P_r(A^0 A) A^0$$

where one defines recursively

$$\begin{aligned} \beta_1(X) &= h_1 \\ \beta_i(X) &= h_i P_i(X) + (1 - h_i)\beta_{i-1}(X) \quad \text{for } 2 \leq i \leq m. \end{aligned}$$

Indeed, if $h_i = 1$, we obtain $\beta_i = P_i$ and if $h_i = 0$, we obtain $\beta_i = \beta_{i-1}$. One can expand the following expression as the algebraic expression:

$$h_m P_m(X) + (1 - h_m)\left[h_{m-1} P_{m-1}(X) + (1 - h_{m-1})\left(\ldots\right)\right]$$

and by expanding it one can see that it can be expressed as a linear combination of elements of the form

$$\begin{aligned} (1 - h_m), \\ (1 - h_m)(1 - h_{m-1}), \ldots, \\ (1 - h_m)(1 - h_{m-1})(1 - h_{m-2}) \ldots (1 - h_2) \end{aligned}$$

and of the form

$$\begin{aligned} (1 - h_m) h_{m-1} P_{m-1}(X), \\ (1 - h_m)(1 - h_{m-1}) h_{m-2} P_{m-2}(X), \ldots, \\ (1 - h_m)(1 - h_{m-1}) \ldots (1 - h_2) h_1 P_1(X). \end{aligned}$$

Using an iterated products' protocol largely inspired by SecPower with the same complexity, the parties can compute shared values of the first three lines with $O(1)$ rounds invoking $O(m)$ times the secure multiplication protocol. When this is done, they can compute sharing of elements of the last three lines with $O(1)$ rounds invoking $O(m)$ times the secure multiplication protocol. Then parties can compute $[\beta_m(A^0 A)] = [P_r(A^0 A)]$ with the baby-step giant-step method. Indeed, each P_j can be computed as follows:

$$P_j(A^0 A) = \sum_{i=0}^{j-1} a_{j-1-i} (A^0 A)^i = \sum_{i=0}^{\tilde{j}} (A^0 A)^{i\tilde{j}} Q_i(A^0 A),$$

where $\tilde{j} = \lceil \sqrt{j-1} \rceil$ and each Q_i is a polynomial of degree at most $\tilde{j} - 1$. These latter polynomials can be deduced locally: $Q_{\tilde{j}}(X) = a_0 X^{\tilde{j}}$ and $Q_i(X) = \sum_{s=0}^{\tilde{j}-1} a_{i\tilde{j}+s-j+1} X^s$ for $i \neq \tilde{j}$. Once sharing of successive powers of A have been computed, a sharing of $Q_i(A^0 A)$ can be deduced with $O(kn^2)$ communication using Lemma 4 and batching the sum. Finally, $[P_r(A^0 A)]$ follows with $O(kn^2)$ communication by applying the same idea for the sum of product of the form $[(A^0 A)^{i\tilde{j}}][Q_i(A^0 A)]$.

Step 5 also requires the protocol for securely computing the rank of a shared matrix. But first, we need to check that a_r is invertible. By definition,

$$a_r := a_r(\zeta) = \zeta^{-r(n-r)} \sum_{l=0}^{r(m+n-2r)} a_{r,l} \zeta^l$$

where $a_{k,l}$ are called the generalized Gram coefficients. By the choice of k and of ζ of degree k, one knows that $a_r \neq 0$. Using the same procedure as one used to compute $[\beta_m(A^0A)] = [P_r(A^0A)]$, one defines

$$\begin{aligned}\alpha_1(X) &= h_1\\ \alpha_i(X) &= h_i a_i(X) + (1-h_i)\alpha_{i-1}(X) \quad \text{for } 2 \leq i \leq m.\end{aligned}$$

And a_r is securely inverted with $O(mk)$ secure multiplications.

Finally, we can extend this result to fields of any positive characteristic by adapting the work of Schönhage [Sch93] (see Appendix 9). When $n \geq m$, this yields to a protocol with $O(n^{5.5})$ communication.

References

[BB89] Judit Bar-Ilan and Donald Beaver. Non-cryptographic fault-tolerant computing in constant number of rounds of interaction. In Piotr Rudnicki, editor, *Proceedings of the Eighth Annual ACM Symposium on Principles of Distributed Computing, Edmonton, Alberta, Canada, August 14-16, 1989*, pages 201–209. ACM, 1989. doi:10.1145/72981.72995.

[BCH+23] Ward Beullens, Ming-Shing Chen, Shih-Hao Hung, Matthias J. Kannwischer, Bo-Yuan Peng, Cheng-Jhih Shih, and Bo-Yin Yang. Oil and vinegar: Modern parameters and implementations. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2023(3):321–365, 2023. URL: https://doi.org/10.46586/tches.v2023.i3.321-365, doi:10.46586/TCHES.V2023.I3.321-365.

[BdV18] Niek J. Bouman and Niels de Vreede. New protocols for secure linear algebra: Pivoting-free elimination and fast block-recursive matrix decomposition. Cryptology ePrint Archive, Report 2018/703, 2018. https://eprint.iacr.org/2018/703.

[Beu21] Ward Beullens. MAYO: practical post-quantum signatures from oil-and-vinegar maps. In Riham AlTawy and Andreas Hülsing, editors, *Selected Areas in Cryptography - 28th International Conference, SAC 2021, Virtual Event, September 29 - October 1, 2021, Revised Selected Papers*, volume 13203 of *Lecture Notes in Computer Science*, pages 355–376. Springer, 2021. doi:10.1007/978-3-030-99277-4_17.

[BGW88] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation (extended abstract). In *20th Annual ACM Symposium on Theory of Computing*, pages 1–10, Chicago, IL, USA, May 2–4, 1988. ACM Press. doi:10.1145/62212.62213.

[CD01] Ronald Cramer and Ivan Damgård. Secure distributed linear algebra in a constant number of rounds. In Joe Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Computer Science*, pages 119–136, Santa Barbara, CA, USA, August 19–23, 2001. Springer, Heidelberg, Germany. doi:10.1007/3-540-44647-8_7.

[CDM00] Ronald Cramer, Ivan Damgård, and Ueli M. Maurer. General secure multi-party computation from any linear secret-sharing scheme. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 316–334. Springer, 2000. doi:10.1007/3-540-45539-6_22.

[CKP07] Ronald Cramer, Eike Kiltz, and Carles Padró. A note on secure computation of the Moore-Penrose pseudoinverse and its application to secure linear algebra. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 613–630, Santa Barbara, CA, USA, August 19–23, 2007. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-74143-5_34.

[DFK+06] Ivan Damgård, Matthias Fitzi, Eike Kiltz, Jesper Buus Nielsen, and Tomas Toft. Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 285–304, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany. doi:10.1007/11681878_15.

[DGL05] Gema M. Diaz-Toca, Laureano González-Vega, and Henri Lombardi. Generalizing cramer's rule: Solving uniformly linear systems of equations. *SIAM J. Matrix Anal. Appl.*, 27(3):621–637, 2005. doi:10.1137/S0895479802418860.

[DN07] Ivan Damgård and Jesper Buus Nielsen. Scalable and unconditionally secure multiparty computation. In Alfred Menezes, editor, *Advances in Cryptology - CRYPTO 2007, 27th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2007, Proceedings*, volume 4622 of *Lecture Notes in Computer Science*, pages 572–590. Springer, 2007. doi:10.1007/978-3-540-74143-5_32.

[DST19] Thomas Debris-Alazard, Nicolas Sendrier, and Jean-Pierre Tillich. Wave: A new family of trapdoor one-way preimage sampleable functions based on codes. In Steven D. Galbraith and Shiho Moriai, editors, *Advances in Cryptology – ASIACRYPT 2019, Part I*, volume 11921 of *Lecture Notes in Computer Science*, pages 21–51, Kobe, Japan, December 8–12, 2019. Springer, Heidelberg, Germany. doi:10.1007/978-3-030-34578-5_2.

[jD65] H. P. jun. Decell. An application of the Cayley-Hamilton theorem to generalized matrix inversion. *SIAM Rev.*, 7:526–528, 1965. URL: https://doi.org/10.1137/1007108.

[Joh20] Fredrik Johansson. On a fast and nearly division-free algorithm for the characteristic polynomial. *CoRR*, abs/2011.12573, 2020. URL: https://arxiv.org/abs/2011.12573, arXiv:2011.12573.

[KLR06] Eyal Kushilevitz, Yehuda Lindell, and Tal Rabin. Information-theoretically secure protocols and security under composition. In Jon M. Kleinberg, editor, *38th Annual ACM Symposium on Theory of Computing*, pages 109–118, Seattle, WA, USA, May 21–23, 2006. ACM Press. doi:10.1145/1132516.1132532.

[KMWF07] Eike Kiltz, Payman Mohassel, Enav Weinreb, and Matthew K. Franklin. Secure linear algebra using linearly recurrent sequences. In Salil P. Vadhan, editor, *TCC 2007: 4th Theory of Cryptography Conference*, volume 4392 of *Lecture Notes in Computer Science*, pages 291–310, Amsterdam, The Netherlands, February 21–24, 2007. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-70936-7_16.

[KPG99] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced Oil and Vinegar signature schemes. In Jacques Stern, editor, *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 206–222, Prague, Czech Republic, May 2–6, 1999. Springer, Heidelberg, Germany. doi:10.1007/3-540-48910-X_15.

[LYKM22] Donghang Lu, Albert Yu, Aniket Kate, and Hemanta K. Maji. Polymath: Low-latency MPC via secure polynomial evaluations and its applications. *Proc. Priv. Enhancing Technol.*, 2022(1):396–416, 2022. doi:10.2478/popets-2022-0020.

[Mul86] Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 338–339. ACM, 1986. doi:10.1145/12130.12164.

[MW08] Payman Mohassel and Enav Weinreb. Efficient secure linear algebra in the presence of covert or computationally unbounded adversaries. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, volume 5157 of *Lecture Notes in Computer Science*, pages 481–496, Santa Barbara, CA, USA, August 17–21, 2008. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-85174-5_27.

[NO07] Takashi Nishide and Kazuo Ohta. Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In Tatsuaki Okamoto and Xiaoyun

Wang, editors, *PKC 2007: 10th International Conference on Theory and Practice of Public Key Cryptography*, volume 4450 of *Lecture Notes in Computer Science*, pages 343–360, Beijing, China, April 16–20, 2007. Springer, Heidelberg, Germany. doi:10.1007/978-3-540-71677-8_23.

[NW06] Kobbi Nissim and Enav Weinreb. Communication efficient secure linear algebra. In Shai Halevi and Tal Rabin, editors, *TCC 2006: 3rd Theory of Cryptography Conference*, volume 3876 of *Lecture Notes in Computer Science*, pages 522–541, New York, NY, USA, March 4–7, 2006. Springer, Heidelberg, Germany. doi:10.1007/11681878_27.

[Pen55] R. Penrose. A generalized inverse for matrices. *Proc. Camb. Philos. Soc.*, 51:406–413, 1955. URL: https://doi.org/10.1017/S0305004100030401.

[PS73] Mike Paterson and Larry J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2(1):60–66, 1973. doi:10.1137/0202007.

[PS78] Franco P. Preparata and Dilip V. Sarwate. An improved parallel processor bound in fast matrix inversion. *Inf. Process. Lett.*, 7(3):148–150, 1978. doi:10.1016/0020-0190(78)90079-0.

[Ran93] Dana Randall. Efficient generation of random nonsingular matrices. *Random Struct. Algorithms*, 4(1):111–118, 1993. doi:10.1002/rsa.3240040108.

[Sch93] Arnold Schönhage. Fast parallel computation of characteristic polynomials by leverrier's power sum method adapted to fields of finite characteristic. In Andrzej Lingas, Rolf G. Karlsson, and Svante Carlsson, editors, *Automata, Languages and Programming, 20nd International Colloquium, ICALP93, Lund, Sweden, July 5-9, 1993, Proceedings*, volume 700 of *Lecture Notes in Computer Science*, pages 410–417. Springer, 1993. doi:10.1007/3-540-56939-1_90.

[Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *27th Annual Symposium on Foundations of Computer Science*, pages 162–167, Toronto, Ontario, Canada, October 27–29, 1986. IEEE Computer Society Press. doi:10.1109/SFCS.1986.25.

Supplementary Material

6 Secure Computation of the Rank

As an additional application, one presents a secure protocol computing the rank of a shared matrix with perfect correctness. Let $\mathbb{F}_q$ be a finite field with characteristic greater than n, an extension degree k chosen as in Section 5, and let $A \in \mathbb{F}_q^{m\times n}$. One works over $\mathbb{F}_{q^k} \simeq \mathbb{F}_q(\zeta)$ where ζ of degree k is defined as in Section 5. Let $A^0 := Q_n^{-1}A^tQ_m \in \mathbb{F}_{q^k}$ with $Q_n = \mathrm{diag}(1, \zeta, \zeta^2, \ldots, \zeta^{n-1})$ and $Q_m = \mathrm{diag}(1, \zeta, \zeta^2, \ldots, \zeta^{m-1})$.

Parties can securely compute the generalized Gram matrix $G = AA^0 \in \mathbb{F}_{q^k}^{m\times m}$ with expected $O(1)$ rounds and $O(km^2)$ communication. Let $\chi_G(z)$ be the characteristic polynomial of G. Parties can compute it with $O(km^{2.5})$ communication by invoking PolyChar. Recall that $\chi_G(z) = (-1)^m z^m q(1/z)$ where $q(z) = \det(I_m + zG) = 1 + a_1(\zeta)z + \cdots + a_m(\zeta)z^m$.

At this point, we can characterize the rank of A by adapting [Mul86].

Lemma 5. *(generalized Gram conditions for the rank). The rank of A is equal to r if and only if $a_k(\zeta) = 0$ for every $k > r$ and $a_r(\zeta) \neq 0$.*

Define $g_i := (-1)^m a_i(\zeta)$, then $\chi_G(X) = \sum_{i=0}^{m} g_i X^{m-i}$ with $g_0 = 1$. It now remains to compute the rank of A as the largest integer $k \in \{0, \ldots, m\}$ such that $g_k \neq 0$. In order to do so, from each sharing of g_i for $i \in \{0, \ldots, m\}$, the parties will compute a sharing of h_i for $i \in \{0, \ldots, m\}$ defined as:

$$h_i = \begin{cases} 1 & \text{if } g_i \neq 0 \\ 0 & \text{otherwise.} \end{cases}$$

One has seen that this can be done in parallel in $O(1)$ rounds and $O(mk)$ invocations of the secure multiplication protocol.

Following a previous reasoning, the rank is equal to α_m where we define recursively

$$\begin{aligned} \alpha_1 &= h_1 \\ \alpha_i &= h_i i + (1 - h_i)\alpha_{i-1} \quad \text{for } i \in \{2, \ldots, m\}. \end{aligned}$$

Again, one can expand the following expression as the algebraic expression:

$$h_m m + (1 - h_m)\left[h_{m-1}(m-1) + (1 - h_{m-1})\left(\ldots\right)\right] \tag{11}$$

and by expanding it one can see that it can be expressed as a linear combination of elements of the form

$$\begin{aligned} &(1 - h_m), \\ &(1 - h_m)(1 - h_{m-1}), \ldots, \\ &(1 - h_m)(1 - h_{m-1})(1 - h_{m-2})\ldots(1 - h_2), \end{aligned}$$

and of the form

$$\begin{aligned} &(1 - h_m)h_{m-1}, \\ &(1 - h_m)(1 - h_{m-1})h_{m-2}, \ldots, \\ &(1 - h_m)(1 - h_{m-1})\ldots(1 - h_2)h_1. \end{aligned}$$

Using an iterated products' method as in SecPower, the parties can compute shared values of the first three lines in $O(1)$ rounds and invoking $O(m)$ times the secure multiplication protocol. When this is done, they can compute sharings of elements of the last three lines again in $O(1)$ rounds and invoking $O(m)$ times the secure multiplication protocol.

Eventually, they can all compute locally a sharing of the expression given in (11) which is the rank of the matrix A.

Theorem 7. *Let $\mathbb{F}_q$ be a finite field with characteristic greater than n, and consider $k > m(m+n)+1$, and $A \in \mathbb{F}^{m\times n}$ of rank r. Then there exists a secure protocol with perfect correctness computing the rank in expected constant round protocol with expected communication complexity in $O(km^{2.5} + mk)$.*

A generalization over finite fields of any characteristic is possible using Remark 4 and Appendix 9 leading to $O(km^{3.5} + mk)$ communication.

7 Secure Trace Computation of a Matrix-Product

We propose a slight adaptation of InnerProd to securely compute the trace of a product of shared matrices. It is based on Equation 9, and yields to a protocol with $O(1)$ communication.

Protocol 8: Secure MPC for the trace of a product

Data: For $1 \leq t \leq k$, party P_t holds the shares $[A]_t = ([a_{i,j}]_t)_{1 \leq i,j \leq n}$ and $[B]_t = ([b_{i,j}]_t)_{1 \leq i,j \leq n}$

Result: Shares $[c]_1, \ldots, [c]_k$ determining the trace of AB

1. Each party P_t locally computes $\tilde{c}_t = \sum_{k,l} [a_{k,l}]_t [b_{l,k}]_t$;
2. Each party P_t re-shares $\tilde{c}_t$, resulting in shares $[\tilde{c}_t]_1, \ldots, [\tilde{c}_t]_k$, and sends $[\tilde{c}_t]_h$ to party P_h for $1 \leq h \leq k$;
3. Each party P_t recombines their share by computing $[c]_t = \sum_{h=1}^{k} \lambda_h [\tilde{c}_h]_t$, where $(\lambda_1, \ldots, \lambda_k)$ is the fixed recombination vector for the secret sharing scheme $[\cdot]$.

Complexity in communication is constant since parties communicate only during step 2 and send one element to each other party. Complexity and security proofs are identical to these for InnerProd.

8 Probability of Singularity

Assume that $|\mathbb{F}| = p > 2$. We follow [Ran93] to derive the probability to draw a non-singular $n \times n$ matrix over $\mathbb{F}$:

$$\begin{aligned} \frac{GL_n(\mathbb{F})}{M_n(\mathbb{F})} &= \frac{(p^n - 1)(p^n - p)...(p^n - p^{n-1})}{p^{n^2}} \\ &= \frac{\prod_{i=0}^{n-1}(p^n - p^i)}{p^{n^2}}. \end{aligned}$$

By developing the first terms of the previous product, one can show that the number of non-singular matrices is greater than

$$\begin{aligned} p^{n^2}\left(1 - \sum_{i=0}^{n-1} \frac{p^i}{p^n}\right) &= p^{n^2}\left(1 - \frac{p^n - 1}{p^n(p-1)}\right) \\ &> p^{n^2}\left(\frac{p-2}{p-1}\right). \end{aligned}$$

Hence, the probability to draw a non-singular matrix is greater than $\frac{p-2}{p-1}$.

If $|\mathbb{F}| = 2$, then the probability for a matrix to be non-singular is at least $1/4$. Indeed, the number of non-singular matrices over $\mathbb{F}_2$ is greater than

$$\begin{aligned} \prod_{i=0}^{n-1}(2^n - 2^i) &= 2^{n-1}\left(\prod_{i=0}^{n-2}(2^n - 2^i)\right) \\ &> 2^{n-1}2^{n(n-1)}\left(1 - \sum_{i=0}^{n-2}\frac{2^i}{2^n}\right) \\ &> \frac{2^{n^2}}{4}. \end{aligned}$$

9 Computation of Characteristic Polynomials over Field of any Characteristic

Let A be a $n \times n$ matrix over $\mathbb{F}$ of characteristic p. The condition on the characteristic of the field in PolyChar comes from the determinant of the system 8: for a field of characteristic p, the inversion failed as soon as $n \geq p$ since $n!$ has to be invertible in the field. Schönhage [Sch93] adapted Le Verrier's power sum method to compute the characteristic polynomial over a finite field $\mathbb{F}$ of any characteristic p. For this purpose, he worked over a transcendental extension $\mathbb{F}' = \mathbb{F}[y]/(y^{n+1})$. As we have seen in Lemma 4, the communication complexity for the computation of the characteristic polynomial will increase by a factor one. Indeed, each element $a \in \mathbb{F}'$ can be decomposed as $a = \sum_{i=0}^{n} a_i y^i$ with all the a_i's in $\mathbb{F}$. This leads to a $O(n^{3.5})$ communication complexity.

10 Chernoff Bounds Theorem

We state the theorem for the case of a sum of independent Bernoulli trials.

Theorem 8. *(Chernoff Bounds Theorem). Let $X = \sum_{i=1}^{n} X_i$, where $X_i = 1$ with probability p_i and $X_i = 0$ with probability $1 - p_i$, and all X_i are independent. Let $\mu = \mathbb{E}(X) = \sum_{i=1}^{n} p_i$. Then*

1. *Upper Tail: $\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2}{2+\delta}\mu}$ for all $\delta > 0$.*
2. *Lower Tail: $\mathbb{P}[X \leq (1-\delta)\mu] \leq e^{-\mu\delta^2/2}$ for all $0 < \delta < 1$.*