A Practical Approach to the Secure Computation of the Moore–Penrose Pseudoinverse over the Rationals[⋆]

Niek J. Bouman[1⋆⋆] and Niels de Vreede[2⋆⋆⋆]

[1] Roseman Labs, The Netherlands
[2] Technische Universiteit Eindhoven, The Netherlands
niek.bouman@rosemanlabs.com, n.d.vreede@tue.nl

Abstract. Solving linear systems of equations is a universal problem. In the context of secure multiparty computation (MPC), a method to solve such systems, especially for the case in which the rank of the system is unknown and should remain private, is an important building block.
We devise an efficient and *data-oblivious* algorithm (meaning that the algorithm's execution time and branching behavior are independent of all secrets) for solving a bounded integral linear system of unknown rank over the rational numbers via the Moore–Penrose pseudoinverse, using finite-field arithmetic. I.e., we compute the Moore–Penrose inverse over a finite field of sufficiently large order, so that we can recover the rational solution from the solution over the finite field. While we have designed the algorithm with an MPC context in mind, it could be valuable also in other contexts where data-obliviousness is required, like secure enclaves in CPUs.
Previous work by Cramer, Kiltz and Padró (*CRYPTO 2007*) proposes a constant-rounds protocol for computing the Moore–Penrose pseudoinverse over a finite field. The asymptotic complexity (counted as the number of secure multiplications) of their solution is $O(m^4 + n^2 m)$, where m and n, $m \leq n$, are the dimensions of the linear system. To reduce the number of secure multiplications, we sacrifice the constant-rounds property and propose a protocol for computing the Moore–Penrose pseudoinverse over the rational numbers in a linear number of rounds, requiring only $O(m^2 n)$ secure multiplications.
To obtain the common denominator of the pseudoinverse, required for constructing an integer-representation of the pseudoinverse, we generalize a result by Ben-Israel for computing the squared volume of a matrix. Also, we show how to precondition a symmetric matrix to achieve generic rank profile while preserving symmetry and being able to remove the preconditioner after it has served its purpose. These results may be of independent interest.

Keywords: secure multiparty computation · secure linear algebra · Moore–Penrose pseudoinverse · oblivious algorithms

[⋆] Full version; includes proofs omitted from the ACNS 2020 proceedings version

[⋆⋆] Work done while at TU Eindhoven, under support from H2020-EU SODA

[⋆⋆⋆] Supported by H2020-EU PRIViLEDGE

1 Introduction

Motivated by the goal of performing elementary statistical tasks such as linear regression *securely*, we revisit the topic of secure linear algebra. In this paper, "securely" refers to *secure multiparty computation* (MPC) [14], however, our results might be of use in other settings as well, for example, for mitigating certain side-channel attacks in trusted execution environments in CPUs.

Secure linear algebra goes back to the work of Cramer and Damgård [12], who proposed constant-rounds MPC protocols for various basic tasks in linear algebra. In that paper, as well as in later papers in the same line of work, like [15, 23, 27, 31], the focus is on linear algebra *over a finite field*.

Our goal is to obtain, in an "MPC-friendly" way, an (approximate) solution to a linear system *over the real numbers*. In this paper we choose to approximate real arithmetic by (exact) rational arithmetic, or, in fact, integer arithmetic, using appropriate scaling. Our main reason behind this choice is the close connection between the finite field $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$ (where p is prime) and integer arithmetic, since we target MPC schemes that offer finite-field arithmetic. Hence, the protocols that we propose in this paper will employ finite-field arithmetic *as a tool, rather than as a goal*. We note that there are various papers targeting the same problem that explore other choices, such as secure fixed-point arithmetic (see, e.g., [18, 29]) or secure floating-point arithmetic (e.g., [7]).

In an earlier joint work with Blom and Schoenmakers [6], we focused on the case of solving full-rank systems. In this paper, we focus on the more general case of solving linear systems whose rank is unknown. Also, we would like to obtain meaningful solutions in case the system is over- or underdetermined. The *Moore–Penrose pseudoinverse* gives natural solutions in both cases: in the overdetermined case, which is the relevant case for linear regression, it yields the least-squares solution; in the underdetermined case it gives the minimum-norm solution. Another application of the Moore–Penrose pseudoinverse is to compute the condition number of a matrix that is not, or not-necessarily, invertible.

Concretely, given a matrix A with integral elements of unknown rank, we propose a protocol for computing the Moore–Penrose pseudoinverse over the rational numbers in a linear number of rounds. The computational complexity, counted as the number of secure multiplications, is $O(m^2n)$, where m and n, $m \leq n$, are the dimensions of the system. In multiplicative-linear-secret-sharing-based MPC schemes, such as Shamir's scheme, we may count a secure inner product as a single secure multiplication; in that case the complexity reduces to $O(mn)$.

It should be rather easy to implement our protocol in any finite-field-based arithmetic secret-sharing MPC framework; beyond elementary finite-field arithmetic our protocol merely requires secure subprotocols for sampling (public) random elements, performing a zero test on a secret-shared field element, computing the reciprocal of a secret-shared field element, and computing the determinant of an invertible secret-shared matrix.

Circumventing Rational Reconstruction. It is well known that one can perform (bounded) rational arithmetic via arithmetic in $\mathbb{F}_p$, essentially as follows: (i) represent the rational inputs as finite-field elements, i.e., an input of the form a/b, for integers a and b and such that $|a|, |b| \leq \sqrt{p/2}$, is encoded as the element $x = a \cdot b^{-1} \in \mathbb{F}_p$, (ii) perform the computation in integer arithmetic modulo p, (iii) reconstruct the numerators and denominators of the results of the computation, elementwise, in the following manner. Let $y \in \mathbb{F}_p$ be an output of the computation, that corresponds to the fraction c/d for integers c and d. Then, if $|c|, |d| \leq \sqrt{p/2}$, we can uniquely reconstruct c and d from y by reducing the two-dimensional lattice basis $\{(p, 0), (y, 1)\}$ using the Lagrange–Gauss algorithm, in the sense that the reduced basis will contain the vector (c, d). This reconstruction procedure is known as *rational reconstruction* (see, e.g., [37]).

An important drawback of the use of rational reconstruction in our scenario is that we essentially would need to *double* the bit-length of the finite field modulus p to guarantee unique reconstruction, compared to a route without rational reconstruction (for more details, see Figure 1). Because arithmetic in a larger finite field is computationally more expensive, we would like to avoid the use of rational reconstruction.

In [6], a key trick for obtaining the inverse of an invertible integer matrix B over the rational numbers from the corresponding inverse over the finite field $\mathbb{F}_p$ *without requiring rational reconstruction*, was to form the integer-valued *adjugate matrix* by multiplying B^{-1} by $\det B$. In a similar spirit, we compute the pseudoinverse $A^\dagger$ over the finite field $\mathbb{F}_p$ and identify the conditions under which it corresponds to the pseudoinverse over the rational numbers. Essentially, this comes down to choosing p sufficiently large; see Section 4.2. We can then obtain an integer representation of the pseudoinverse by forming the pair $(dA^\dagger, d)$, where $dA^\dagger$ is an integer matrix containing the numerators of the pseudoinverse and d is the common denominator of the pseudoinverse, which coincides with the squared *volume* of A [4], which we write as $(\operatorname{vol} A)^2$. Figure 1 illustrates our approach and compares it to the alternative route of rational reconstruction.

Although taking the square of the volume is rather excessive in certain cases (for example, the magnitude of the common denominator of B^{-1}, for any *invertible* matrix B, equals $|\det B| = \operatorname{vol} B$), it is essentially the price we have to pay for not knowing whether we are dealing with such a special case.

Computing the Pseudoinverse and Its Common Denominator. To compute the Moore–Penrose pseudoinverse of A obliviously, we first compute a *reflexive generalized inverse* of the symmetric product $AA^\intercal AA^\intercal$ by means of block-recursive elimination. We then compute the Moore–Penrose pseudoinverse from this generalized inverse.

Regarding the common denominator, Springer computes $(\operatorname{vol} A)^2$ via an integer-preserving rank decomposition [36]. To circumvent the need for constructing such a rank decomposition, we seek a simpler alternative. Ben-Israel gives a method for computing $(\operatorname{vol} A)^2$ that requires an orthonormal basis for the left nullspace of A [4]. Although an *orthonormal* basis might not even exist over a finite field, we can easily construct a matrix K whose columns span the left

$$\begin{tikzcd}
A \in \mathbb{Z}^{m\times n} \arrow[r, "\bmod p"] \arrow[d, "\pi"] & \tilde{A} \in \mathbb{F}_p^{m\times n} \arrow[d, "\mathsf{Pseudoinverse}"] \\
A^\dagger \in \mathbb{Q}^{n\times m} \arrow[d, "d"] & \tilde{A}^\dagger \in \mathbb{F}_p^{n\times m} \arrow[d, "d"] \\
dA^\dagger \in \mathbb{Z}^{n\times m} & d\tilde{A}^\dagger \in \mathbb{F}_p^{n\times m} \arrow[l, "\mathrm{id}"]
\end{tikzcd}$$

(a) Our approach. The map d represents scalar multiplication by $d = (\operatorname{vol} A)^2$ and id represents the identity map. The solutions $dA^\dagger$ and $d\tilde{A}^\dagger$ coincide, provided that p is chosen large enough, i.e., according to Lemma 5.

$$\begin{tikzcd}
A \in \mathbb{Z}^{m\times n} \arrow[r, "\bmod q"] \arrow[d, "\pi"] & \tilde{A} \in \mathbb{F}_q^{m\times n} \arrow[d, "\mathsf{Pseudoinverse}"] \\
A^\dagger \in \mathbb{Q}^{n\times m} & \tilde{A}^\dagger \in \mathbb{F}_q^{n\times m} \arrow[l, "\nu"]
\end{tikzcd}$$

(b) Approach using rational reconstruction. The map ν represents the elementwise rational reconstruction procedure. All reconstructed fractions will be in lowest terms (numerator and denominator have no common nontrivial factors). There is, however, a price to be paid, in that $q \geq 2p^2$. Also, the map ν (the Lagrange–Gauss algorithm) is not "MPC-friendly".

Fig. 1: Comparison between our approach and the approach via rational reconstruction. In the diagrams, the map $\pi : \mathbb{Q}^{m\times n} \to \mathbb{Q}^{n\times m}, A \mapsto A^\dagger$ applies the Moore–Penrose inverse over the rationals.

nullspace of A. We generalize Ben-Israel's result so that we can compute $(\operatorname{vol} A)^2$ from A and K.

Preconditioning for Computing Pseudoinverses. As noted above, we will compute the Moore–Penrose inverse via a generalized inverse that is obtained using block-recursive elimination.

Deterministic elimination algorithms typically employ pivoting to avoid problems like division by zero. Pivoting involves searching for and applying suitable row and/or column swaps prior to each elimination step. In secure computation, however, we aim to avoid pivoting because searching for particular elements and applying data-dependent row and column swaps, *obliviously*, is expensive (in a computational- and round-complexity sense).

An MPC-friendly alternative is to transform the matrix to be eliminated into an equivalent matrix for which the elimination procedure will succeed *without any pivoting*; this approach is called *preconditioning.* In case of Gaussian elimination, for example, the condition of *generic rank profile*[3] guarantees that pivoting can be omitted. As we prove in this paper, generic rank profile is also a sufficient condition for correctness of the particular block-recursive elimination algorithm that we use.

When dealing with a square, full rank matrix B over a finite field $\mathbb{F}$ with large order, one way to achieve generic rank profile with high probability is by pre-multiplying B by a preconditioner matrix R that is chosen uniformly at random from the set of all invertible matrices having the same size as B. When

[3] A matrix A of rank r has *generic rank profile* if and only if all upper-left square submatrices of A up to dimension $r \times r$ are invertible.

computing the inverse of RB, we can apply the rule $(RB)^{-1} = B^{-1}R^{-1}$, which we will refer to as the *reverse order law* for matrix inversion, to show that the inverse of the preconditioner can easily be removed by post-multiplying by R. For a matrix A with arbitrary rank r, pre-multiplying by a randomly chosen invertible matrix R (of appropriate size) is not sufficient for achieving generic rank profile; we additionally need to mix A's columns by multiplying A by a preconditioner matrix from the right.

A major problem that arises when trying to remove a preconditioner when computing the pseudoinverse, is that the reverse order law for pseudoinverses does not hold in general [19, 20]. In particular, unfortunately, we have that $(LAR)^\dagger$ does not necessarily equal $R^\dagger A^\dagger L^\dagger$ for invertible preconditioner matrices L and R. Hence, we cannot simply extract $A^\dagger$ from $(LAR)^\dagger$ like we could do above for B^{-1}. We circumvent this problem by applying the preconditioner only to $AA^\mathsf{T}AA^\mathsf{T}$ and removing the preconditioner immediately after computing the reflexive generalized inverse, for which the reverse-order law does hold.

An additional constraint in our setting where we apply preconditioning to $AA^\mathsf{T}AA^\mathsf{T}$, rather than to A directly, is that the preconditioner should preserve symmetry, since the symmetry property enables significant computational savings during elimination. A preconditioner for this particular scenario seems to be lacking in the literature. We resolve this by proving that the preconditioner $X \mapsto UXU^\mathsf{T}$ for a uniformly random matrix U fulfills all our constraints.

Interestingly, and unlike Gaussian elimination, when working over the real or complex numbers, the particular block-recursive algorithm that we use for computing the reflexive generalized inverse does not even require its input to have generic rank profile, hence no preconditioning is needed in this case. Nonetheless, in fields with positive characteristic, the condition emerges from the phenomenon of self-orthogonality.

1.1 Related Work

Cramer, Kiltz and Padró [15] propose a constant-rounds protocol for securely computing the Moore–Penrose pseudoinverse over a finite field. Their approach is to first compute the characteristic polynomial of the Gram matrix $A^\mathsf{T}A$, from which they then compute the rank of A (via a technique by Mulmuley [28]) as well as the pseudoinverse of A (via the Cayley–Hamilton theorem).

An important theme in [15] is to ensure that A (and A^T) are *suitable*, which guarantees, informally speaking, that certain subspaces that are orthogonal over a field with characteristic zero, remain orthogonal over fields with positive characteristic. In our work, where we focus on the setting where the modulus (hence the field's characteristic) is chosen sufficiently large, existence of the pseudoinverse is guaranteed by a result in [2]. (We state this result in the next section.) Nonetheless, as described in the previous section, we do take special precautions, namely, applying preconditioning, to avoid problems related to working over a field with positive characteristic when computing a reflexive generalized inverse.

For an $m \times n$ matrix where $m \leq n$, the complexity (number of secure multiplications) of Cramer et al.'s solution is $O(m^4 + n^2 m)$. Our solution, albeit not constant-rounds, has complexity $O(m^2 n)$, and even $O(mn)$ when assuming availability of a "cheap inner product", where the hidden constants in the Big-Oh of our solution are single-digit integers. By "cheap inner product", we mean that an inner product between two vectors of the same but arbitrary length has the same communication and round complexity as a single secure multiplication. It is possible to perform multiplication of an $m \times \ell$ matrix by an $\ell \times n$ matrix using no more than mn "cheap inner products". Because the coefficients of the result matrix may all be mutually independent, it is reasonable to take the complexity of such a matrix product to be equal to mn.

We leave it to further research to compare the practical performance of our method to that of [15] in various application scenarios (i.e., various matrix-dimension regimes, network latency, bounded computational resources and storage space, etc.).

Relation to the LEU Decomposition. An earlier work by the authors [10] proposes to use Malaschonok's *LEU* decomposition [24] for solving linear systems of unknown rank in the context of secure computation. (Note that [10] does not deal with the problem of computing the Moore–Penrose pseudoinverse.) Our new protocol Pseudoinverse is superior to the *LEU*-decomposition-based protocol from [10]; in terms of round complexity, $O(m)$ versus $O(m^{1.59})$, as well as in terms of the asymptotic computational complexity, $O(m^2)$ versus $O(m^2 \log m)$ secure inner products for a square $m \times m$ matrix.

2 Preliminaries

Secret Sharing and Secure Computation. Let $\mathbb{F}_p = \mathbb{Z}/p\mathbb{Z}$, where p is prime. We use $\mathbb{F}$ to denote an arbitrary field. We assume the use of an MPC protocol based on arithmetic secret-sharing over $\mathbb{F}_p$. Our protocols will inherit the security properties (passive vs. active) from the underlying MPC protocol and of the subprotocols invoked by our protocol. The notation $[\![x]\!]$ represents an element $x \in \mathbb{F}_p$ that is secret-shared among the parties in the MPC protocol. Notation for secure arithmetic then follows naturally, for example, $[\![c]\!] \leftarrow [\![a]\!] + [\![b]\!]$ describes the addition of a and b where the result is stored in a new secret-shared element c, and $[\![d]\!] \leftarrow [\![a]\!][\![b]\!]$ describes an invocation of the multiplication protocol to securely compute the product of a and b and store the result in d. For arbitrary integer matrices A and B, the notation $[\![A]\!]$ expresses that all elements of A are secret-shared over $\mathbb{F}_p$, and $[\![A]\!] + [\![B]\!]$ and $[\![A]\!][\![B]\!]$ represent secure matrix addition (which coincides with elementwise addition) and secure matrix multiplication, respectively. Our protocols assume the availability of subprotocols for securely sampling private as well as public random field elements (e.g., [13]), denoted as $[\![a]\!] \overset{\$}{\leftarrow} \mathbb{F}_p$ and $a \overset{\$}{\leftarrow} \mathbb{F}_p$ respectively, for securely inverting a field element (see [3]), and for performing a secure zero test [16,30]. The latter two are denoted as protocols Reciprocal and IsZero, respectively. We require protocol Reciprocal to be secure for all nonzero inputs (i.e., the protocol is allowed to leak information when run on a secret share of zero). Protocol IsZero returns $[\![1]\!]$ if its argument equals zero and returns $[\![0]\!]$ otherwise.

Generalized Inverses. A *generalized inverse* of a matrix A is a matrix X associated to A that exists for a class of matrices larger than the class of invertible matrices, shares some properties with the ordinary inverse, and reduces to the ordinary inverse when A is non-singular. In this paper, we classify generalized inverses using the following four properties, also known as the *Penrose equations*:

$$(1)\, AXA = A, \quad (2)\, XAX = X, \quad (3)\, (AX)^{\mathsf{T}} = AX, \quad (4)\, (XA)^{\mathsf{T}} = XA.$$

The matrix X that satisfies all four Penrose equations for a given matrix A is called the *Moore–Penrose pseudoinverse*, or simply *pseudoinverse* of A, which we denote as $A^{\dagger}$. The Moore–Penrose inverse of A over $\mathbb{F}$ exists if and only if $\operatorname{rank}(AA^{\mathsf{T}}) = \operatorname{rank}(A^{\mathsf{T}}A) = \operatorname{rank} A$ [32, Thm 1], and if it exists it is unique. We will also focus on generalized inverses of A which only satisfy equations (1) and (2); such generalized inverses are called *reflexive generalized inverses* and we denote any reflexive generalized inverse of A by A^{-}. Note that reflexive generalized inverses are not necessarily unique. For an extensive treatment of generalized inverses, the reader is referred to [5].

For a square matrix A partitioned as

$$A = \begin{pmatrix} E & F \\ G & H \end{pmatrix} \tag{5}$$

such that E is square, A/E denotes the *generalized Schur complement*

$$A/E = H - GE^{-}F.$$

Submatrices, Their Determinants and Rank Properties. For any $n \in \mathbb{N}$, we write $[n]$ for the set $\{1, \ldots, n\}$. For any $m \times n$ matrix A and index sets $\mathcal{I} \subset [m]$ and $\mathcal{J} \subset [n]$, $[A]_{\mathcal{I},\mathcal{J}}$ denotes the determinant of the submatrix of A obtained by selecting all rows in $\mathcal{I}$ and all columns in $\mathcal{J}$. Furthermore, $A_{[k]}$ denotes the leading principal submatrix of order k, i.e., the matrix obtained by taking the first k rows and first k columns of A, and we use $[A]_k$ as shorthand for $[A]_{[k],[k]}$, i.e., the leading principal minor of order k. Thus, it holds that $\det A_{[k]} = [A]_k$.

Let A be a matrix of rank r. We say that a matrix A has *generic rank profile* [21] if for all $k \in [r]$, it holds that A's leading principal minor of order k is nonzero.

Let A be partitioned as in (5). If $\det E \neq 0$, then *Schur's determinant formula* asserts that

$$\det A = \det(E)\det(A/E) = \det(E)\det(H - GE^{-1}F).$$

A direct consequence of [25, Thm 19] is that

$$\operatorname{rank} A \geq \operatorname{rank} E + \operatorname{rank}(A/E).$$

Hence, if A has generic rank profile and E has at least dimension $r \times r$ where $r = \operatorname{rank} A$, then A/E is the null matrix.

The Volume of a Matrix. For any matrix A with rank r and nonzero singular values $\sigma_1, \ldots, \sigma_r$, its *volume* is defined as $\operatorname{vol} A = \prod_{i=1}^{r} \sigma_i$. Note that this definition implies that we define the volume of the zero matrix to be one, which will be convenient for our purpose but deviates from Ben-Israel's definition of matrix volume for this special case [4]. A matrix over an integral domain has a pseudoinverse if and only if its squared volume is a unit (i.e., an invertible element) of the integral domain [2]. The fact that, for any matrix $A \in \mathbb{R}^{m \times n}$, the singular values of AA^{T} are the squares of the singular values of A leads to the following equation:

$$\operatorname{vol}(AA^{\mathsf{T}}) = (\operatorname{vol} A)^2, \tag{6}$$

which holds over an arbitrary field. In case A is a square nonsingular matrix, i.e., $m = n$ and $\det A \neq 0$, its volume coincides with the absolute value of its determinant:

$$\operatorname{vol} A = |\det A|. \tag{7}$$

Combining the two preceding equations gives

$$(\operatorname{vol} A)^2 = \det(AA^{\mathsf{T}}), \tag{8}$$

in the case that $\operatorname{rank} A = m$.

3 Block-Recursive Elimination

In this section we present ObliviousRGInverse, our oblivious protocol for computing a reflexive generalized inverse of any symmetric matrix over $\mathbb{F}_p$ that has generic rank profile. Although we could easily devise a protocol that also works for non-symmetric matrices, we deliberately restrict to symmetric matrices, for the following two reasons: (i) by doing so, we achieve a significant computational saving (essentially a factor of two); and (ii) for our application we anyway only need to compute a reflexive generalized inverse of a symmetric matrix.

First, we define the *extended reciprocal* of an element $c \in \mathbb{F}$ as zero if $c = 0$ and c^{-1}, i.e., the (ordinary) reciprocal, otherwise. Note that the reflexive generalized inverse of a 1×1 matrix is equal to the 1×1 matrix containing the extended reciprocal of its only coefficient. ScalarRGInverse is a secure protocol for computing the extended reciprocal.

Protocol 1 ScalarRGInverse($[\![a]\!]$)

```
$[\![z]\!] \leftarrow$ IsZero($[\![a]\!]$)
return Reciprocal($[\![a + z]\!]$) $-$ $[\![z]\!]$
```

ObliviousRGInverse is given as Protocol 2. On line 4, the partitioning is done such that E and G are square and their dimensions differ by at most one. We remark that the side notes with label "symmetric" in ObliviousRGInverse

Protocol 2 $\mathsf{ObliviousRGInverse}([\![A]\!])$

```
if $n = 1$ then
    return $\mathsf{ScalarRGInverse}([\![a_{1,1}]\!])$
else
    $\begin{pmatrix} [\![E]\!] & [\![F]\!] \\ [\![F^\mathsf{T}]\!] & [\![G]\!] \end{pmatrix} \leftarrow [\![A]\!]$    ▷ split as evenly as possible
    $[\![X]\!] \leftarrow \mathsf{ObliviousRGInverse}([\![E]\!])$
    $[\![XF]\!] \leftarrow [\![X]\!][\![F]\!]$
    $[\![G - F^\mathsf{T}XF]\!] \leftarrow [\![G]\!] - [\![F^\mathsf{T}]\!][\![XF]\!]$    ▷ symmetric
    $[\![Y]\!] \leftarrow \mathsf{ObliviousRGInverse}([\![G - F^\mathsf{T}XF]\!])$
    $[\![XFY]\!] \leftarrow [\![XF]\!][\![Y]\!]$
    $[\![X + XFYF^\mathsf{T}X]\!] \leftarrow [\![X]\!] + [\![XFY]\!][\![XF]\!]^\mathsf{T}$    ▷ symmetric
    return $\begin{pmatrix} [\![X + XFYF^\mathsf{T}X]\!] & -[\![XFY]\!] \\ -[\![XFY]\!]^\mathsf{T} & [\![Y]\!] \end{pmatrix}$
```

indicate that the resulting matrix is symmetric, which is to be exploited in an implementation.

It is easy to see that protocol ObliviousRGInverse is oblivious: it only branches on the dimensions of the matrix, which are considered public, and otherwise only performs elementary arithmetic operations, and calls to secure subprotocols (including recursive calls to itself).

3.1 Correctness Analysis

Rohde [35] shows that a reflexive generalized inverse A^- of a symmetric, positive-semidefinite matrix *over the real numbers*[4]

$$A = \begin{pmatrix} E & F \\ F^\mathsf{T} & G \end{pmatrix} \tag{9}$$

can be expressed in Banachiewicz–Schur form as

$$A^- = \begin{pmatrix} E^- + E^-FS^-F^\mathsf{T}E^- & -E^-FS^- \\ -S^-F^\mathsf{T}E^- & S^- \end{pmatrix}, \tag{10}$$

where E^- is a reflexive generalized inverse of E and S^- is a reflexive generalized inverse of $S = G - F^\mathsf{T}E^-F$. This form allows for a block-recursive algorithm for computing the reflexive generalized inverse over the real numbers. As proved by Marsaglia and Styan, the correctness of Rohde's result *over an arbitrary field* depends on the following additional condition.

Lemma 1 ([26], statement tailored to our needs). *Over an arbitrary field, Equation* (10) *is a reflexive generalized inverse of A if and only if*

$$\operatorname{rank} A = \operatorname{rank} E + \operatorname{rank} S, \tag{11}$$

[4] Rohde [35] actually shows his result for complex matrices, but for our purposes it is more convenient to state his result for real matrices.

or, equivalently, the following three conditions are satisfied simultaneously

$$\begin{cases} (I - EE^-)F(I - S^-S) = 0 & (12) \\ (I - SS^-)F^\mathsf{T}(I - E^-E) = 0 & (13) \\ (I - EE^-)FS^-F^\mathsf{T}(I - E^-E) = 0, & (14) \end{cases}$$

where E^- and S^- are reflexive generalized inverses of E and $S = A/E$ respectively.

Lemma 2. *Over an arbitrary field, a sufficient condition for Equation* (10) *to be a reflexive generalized inverse of a symmetric matrix A is that A has generic rank profile.*

Proof. We partition A as in Equation (9) arbitrarily but such that E is square. Now we can make a case distinction on E: (i) E is invertible. Then E^- coincides with the ordinary inverse and it immediately follows that $(I - EE^-) = (I - E^-E) = 0$, thus satisfying (12)–(14) from Lemma 1.

(ii) E is not invertible. Since A has generic rank profile, it then immediately follows that $\operatorname{rank} A = \operatorname{rank} E$ and furthermore that $\operatorname{rank} S = 0$, thus satisfying (11). □

Lemma 3. *For any $m \times n$ matrix A over an arbitrary field, any k such that $A_{[k]}$ is invertible, and any i such that $0 \leq i \leq \min(m, n) - k$ it holds that*

$$A_{[k+i]}/A_{[k]} = (A/A_{[k]})_{[i]}.$$

Proof. Let

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix},$$

where $A_{11} = A_{[k]}$ is an invertible $k \times k$ matrix and A_{22} is an $i \times i$ matrix. Then

$$\begin{aligned} (A/A_{[k]})_{[i]} &= \left(\begin{pmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{pmatrix} - \begin{pmatrix} A_{21} \\ A_{31} \end{pmatrix} A_{11}^{-1} \begin{pmatrix} A_{12} & A_{13} \end{pmatrix} \right)_{[i]} \\ &= A_{22} - A_{21}A_{11}^{-1}A_{12} \\ &= A_{[k+i]}/A_{[k]}. \end{aligned}$$

□

Corollary 1. *Protocol* ObliviousRGInverse*, when run on a symmetric matrix A over $\mathbb{F}_p$ having generic rank profile, correctly computes a reflexive generalized inverse.*

Proof. For the base case, we have already argued correctness of the extended reciprocal near the beginning of Section 3. For the recursive step applied to A, note that for an arbitrary partitioning but such that E is a $k \times k$ matrix for some integer k, it is easy to see that E is symmetric and has generic rank profile. Correctness then follows from Lemma 2.

We prove that S is symmetric and has generic rank profile by distinguishing two cases. If E is not invertible, then $\operatorname{rank} A = \operatorname{rank} E$ and S is necessarily the (square) null matrix, which is symmetric and has generic rank profile. Otherwise, E is invertible and $S = A/E = G - F^T E^{-1} F$, which is clearly symmetric. For generic rank profile, we can apply Schur's determinant formula to the leading principal minors of A: for any i such that $0 \leq i \leq \operatorname{rank} A - k$ we have $0 \neq \det(A_{[k+i]}) = \det(E)\det(A_{[k+i]}/E)$. Then, applying Lemma 3 gives $\det(A_{[k+i]}/E) = \det((A/E)_{[i]}) \neq 0$, i.e., A/E has generic rank profile. In both cases, correctness now follows from Lemma 2. □

Remark 1. We have proved that generic rank profile is sufficient for correctness—we did not prove that this condition is necessary. This leaves open the possibility that a weaker condition on the input matrix (weaker than generic rank profile) would suffice for correctness of ObliviousRGInverse. In the next section we will compute $(AA^{\mathsf{T}}AA^{\mathsf{T}})^{-}$, from which we construct $A^{\dagger}$. To ensure the correctness of ObliviousRGInverse we will actually randomize its input, $AA^{\mathsf{T}}AA^{\mathsf{T}}$, so that it has generic rank profile with high probability and then undo the randomization on the result. One might raise the question whether choosing the modulus p large enough to guarantee the existence of $A^{\dagger}$, could immediately guarantee correctness of ObliviousRGInverse without requiring $AA^{\mathsf{T}}AA^{\mathsf{T}}$ to have generic rank profile. We do not address this question, as our randomization technique suffices and introduces only minimal overhead.

3.2 Complexity Analysis

We first state the complexity (number of secure operations) of protocol ObliviousRGInverse when run on a square matrix whose dimensions are a power of two.

Proposition 1. *Protocol* ObliviousRGInverse*, when run on an $m \times m$ matrix over $\mathbb{F}_p$, where $m = 2^k$ for integer k, requires $\frac{3}{2}m(m-1) + \frac{1}{2}m\log_2 m$ secure inner products and m invocations of* ScalarRGInverse*.*

Proof. Correctness of Proposition 1 is easily proved using induction on k. In the base case, $k = 0$, protocol ObliviousRGInverse simply invokes ScalarRGInverse once.

As induction hypothesis, suppose the proposition holds for some $m = 2^k$, where k is integer. Then protocol ObliviousRGInverse, when run on a $2m \times 2m$ matrix over $\mathbb{F}_p$, performs 2 invocations of OblivousRGInverse on $m \times m$ matrices, which requires $3m(m-1) + m\log_2 m$ secure inner products and $2m$ invocations of ScalarRGInverse per the induction hypothesis. The protocol further performs four matrix-matrix products of $m \times m$ matrices. Two of these products are symmetric, so these products can be performed using $3m^2 + m$ secure inner products. The total number of secure inner products required is therefore equal to $6m^2-2m+m\log_2 m = 6m^2-3m+m\log_2(2m) = \frac{3}{2}(2m)(2m-1)+\frac{1}{2}(2m)\log_2(2m)$ and the number of invocations of ScalarRGInverse is $2m$. □

If the dimensions of the matrix, m, are not a power of two, it is not always possible to divide the matrix evenly in step 4 of the protocol. In these cases the number of secure inner products required is slightly greater than the number stated in Proposition 1. For general dimensions, we prove the following proposition. We note that this bound is not tight.

Proposition 2. *Protocol* ObliviousRGInverse*, when run on an $m \times m$ matrix over $\mathbb{F}_p$, requires fewer than $\frac{3}{2}m(m-1) + m\log_2 m$ secure inner products and exactly m invocations of* ScalarRGInverse*.*

We also express the complexity of protocol ObliviousRGInverse in terms of elementary secure multiplications, for MPC schemes for which the "cheap inner product" is not available. Note that the bound given here is exact if we assume the naïve algorithm for matrix multiplication. A more advanced algorithm would result in sub-cubic, but still super-quadratic complexity.

Proposition 3. *Protocol* ObliviousRGInverse*, when run on an $m \times m$ matrix over $\mathbb{F}_p$, requires at most $\frac{1}{2}m^3 + \frac{1}{2}m^2 - m$ secure multiplications and exactly m invocations of* ScalarRGInverse*.*

The proofs of Proposition 2 and 3 can be found in the appendix.

4 Computing the Moore–Penrose Pseudoinverse

We will compute the Moore–Penrose pseudoinverse using a formula (see, e.g., [34, p. 207]) that computes $A^\dagger$ in terms of a reflexive generalized inverse:

$$A^\dagger = A^\mathsf{T}(AA^\mathsf{T}AA^\mathsf{T})^- AA^\mathsf{T}. \tag{15}$$

Before proposing our protocol Pseudoinverse, we deal with three remaining questions, namely how to compute the common denominator, how to choose an appropriate modulus, and how to reliably compute $(AA^\mathsf{T}AA^\mathsf{T})^-$, as $AA^\mathsf{T}AA^\mathsf{T}$ does not necessarily have generic rank profile, which is required by protocol ObliviousRGInverse for correctness.

4.1 Computing the Common Denominator

Over the rational numbers, a common denominator d such that $dA^\dagger$ is integer-valued if A is integer-valued is $d = (\operatorname{vol} A)^2$ [36, Satz 10]. The squared volume is minimal in the sense that there exist matrices for which it is the smallest possible common denominator.

If we would have an *orthonormal* basis for the left or right nullspace of A, then we could use [4, Thm. (4.1)] to compute $(\operatorname{vol} A)^2$ directly. An orthonormal basis does not necessarily exist over an arbitrary field. Instead, we generalize [4, Thm. (4.1)] by relaxing the requirements on the nullspace basis.

Lemma 4. *Let $A \in \mathbb{F}^{m\times k}$ be a matrix of rank r. Let $B \in \mathbb{F}^{m\times \ell}$ be a matrix of rank $m-r$ such that its columns are orthogonal to the columns of A, i.e., $B^{\mathsf{T}}A = 0$. Then,*

$$\det(AA^{\mathsf{T}} + BB^{\mathsf{T}}) = (\operatorname{vol} A)^2(\operatorname{vol} B)^2.$$

Proof. Note that $AA^{\mathsf{T}} + BB^{\mathsf{T}} = \begin{pmatrix} A & B \end{pmatrix}\begin{pmatrix} A & B \end{pmatrix}^{\mathsf{T}}$. Because the columns of A are orthogonal to those of B, the matrix $\begin{pmatrix} A & B \end{pmatrix}$ has rank $r + (m-r) = m$ and hence

$$\det(\begin{pmatrix} A & B \end{pmatrix}\begin{pmatrix} A & B \end{pmatrix}^{\mathsf{T}}) = (\operatorname{vol}\begin{pmatrix} A & B \end{pmatrix})^2 = (\operatorname{vol} A)^2(\operatorname{vol} B)^2,$$

where the first equality holds by equation (8), and the second equality is [4, Example 5.1]. □

Theorem 1. *Let $A \in \mathbb{F}^{m\times n}$ be a matrix of rank r. Let $K = I - AA^{\dagger} \in \mathbb{F}^{m\times m}$. Then,*

$$(\operatorname{vol} A)^2 = \det(AA^{\mathsf{T}} + K).$$

Proof. By property (3) of the pseudoinverse, we have that $K = K^{\mathsf{T}}$. This fact, and property (1) of the pseudoinverse imply that $KK^{\mathsf{T}} = KK = K$ and $K^{\mathsf{T}}A = 0$, i.e., K is idempotent and its columns are orthogonal to the columns of A.

Combining equation (6) with the fact that K is idempotent and symmetric gives us that $\operatorname{vol} K = \operatorname{vol}(KK^{\mathsf{T}}) = (\operatorname{vol} K)^2$. Since the volume of a matrix is nonzero, we conclude that $\operatorname{vol} K = 1$.

Orthogonality of the columns of K and A implies that $\operatorname{rank} K \le m - r$ and

$$\operatorname{rank} K = \operatorname{rank}(I - AA^{\mathsf{T}}) \ge \operatorname{rank} I - \operatorname{rank}(AA^{\mathsf{T}}) = m - r$$

follows from subadditivity of matrix rank. Applying Lemma 4 gives us

$$\det(AA^{\mathsf{T}} + K) = \det(AA^{\mathsf{T}} + KK^{\mathsf{T}}) = (\operatorname{vol} A)^2(\operatorname{vol} K)^2 = (\operatorname{vol} A)^2. \qquad \square$$

4.2 Bound on the Modulus

Springer [36] has proved the following upper bound on the magnitudes of the numerators and the common denominator of the pseudoinverse. Choosing p larger than twice this bound will guarantee that: (i) $d = (\operatorname{vol} A)^2$ is an invertible element in $\mathbb{F}_p$, which is a necessary and sufficient condition for existence of $A^{\dagger}$ over $\mathbb{F}_p$ [2] (see also Section 2), and (ii) that the pair $(dA^{\dagger}, d)$ over $\mathbb{F}_p$ coincides with $(dA^{\dagger}, d)$ over $\mathbb{Z}$ (see Lemma 5 below), and (iii) that the product $AA^{\mathsf{T}}AA^{\mathsf{T}}$ occurring in Equation (15) has the same rank as A (which we will need in Theorem 2, and note that (iii) is implied by applying (i) to the upcoming Proposition 4).

Lemma 5 ([36, Satz 12]). *Let $N_0 = (\operatorname{vol} A)^2$ and $Z_0 = (z_{ij}) \in \mathbb{Z}^{m\times n}$ be an integer matrix of rank r such that $A^{\dagger} = \frac{1}{N_0}Z_0$. Let $\mu = \min(m, n)$. Then,*

$$\max(|N_0|, \max_{i,j}|z_{ij}|) \le \max\left(\frac{\|A\|_F^{2r}}{r^r}, \frac{\|A\|_F^{2r-1}}{\sqrt{r^r(r-1)^{r-1}}}\right), \tag{16}$$

and

$$\max(|N_0|, \max_{i,j} |z_{ij}|) \leq \max\left(\frac{\|A\|_F^{2\mu}}{\mu^\mu}, \frac{\|A\|_F^{2\mu-1}}{\sqrt{\mu^\mu(\mu-1)^{\mu-1}}}\right), \tag{17}$$

where $\|A\|_F = \sqrt{\sum_{ij} |a_{ij}|^2}$ *is the Frobenius norm of* A*.*

Remark 2. In a setting in which the rank r is unknown, one would use (17).

For our construction, we further require that

$$\operatorname{rank}(AA^\mathsf{T}AA^\mathsf{T}) = \operatorname{rank} A. \tag{18}$$

This requirement holds unconditionally over fields of characteristic zero, but not necessarily over finite fields. Nonetheless, as we show below, it turns out that existence of the Moore–Penrose inverse already implies (18).

Proposition 4. *Let* A *be an arbitrary matrix over* $\mathbb{F}$*. The Moore–Penrose inverse of* A *exists if and only if*

$$\operatorname{rank}(AA^\mathsf{T}AA^\mathsf{T}) = \operatorname{rank} A.$$

Proof. Recall from Section 2 that the Moore–Penrose inverse exists over $\mathbb{F}$ if and only if $\operatorname{rank}(AA^\mathsf{T}) = \operatorname{rank}(A^\mathsf{T}A) = \operatorname{rank} A$. Note that

$$\operatorname{rank}(AA^\mathsf{T}AA^\mathsf{T}) = \operatorname{rank} A \implies \operatorname{rank} A = \operatorname{rank}(AA^\mathsf{T}) = \operatorname{rank}(A^\mathsf{T}A),$$

so we only have to prove the converse.

Let $A = VW$ be a rank decomposition of A, i.e., V and W have full column-rank and full row-rank, respectively. Over an arbitrary field, a rank decomposition exists but is not necessarily unique; see, e.g., [33]. Then,

$$\operatorname{rank} A = \operatorname{rank}(AA^\mathsf{T}) = \operatorname{rank}(VWW^\mathsf{T}V^\mathsf{T}) \implies \operatorname{rank}(WW^\mathsf{T}) \geq \operatorname{rank} A,$$

and similarly,

$$\operatorname{rank} A = \operatorname{rank}(A^\mathsf{T}A) = \operatorname{rank}(W^\mathsf{T}V^\mathsf{T}VW) \implies \operatorname{rank}(V^\mathsf{T}V) \geq \operatorname{rank} A.$$

Also note that both WW^T and $V^\mathsf{T}V$ have dimension $r \times r$ with $r = \operatorname{rank} A$, therefore, they are invertible. We now write $AA^\mathsf{T}AA^\mathsf{T}$ in terms of V and W, and multiply by V^T from the left and by V from the right, by which we obtain:

$$V^\mathsf{T}AA^\mathsf{T}AA^\mathsf{T}V = (V^\mathsf{T}V)(WW^\mathsf{T})(V^\mathsf{T}V)(WW^\mathsf{T})(V^\mathsf{T}V),$$

the rank of which bounds $\operatorname{rank}(AA^\mathsf{T}AA^\mathsf{T})$ from below.

Thus, $\operatorname{rank}(AA^\mathsf{T}AA^\mathsf{T}) = \operatorname{rank} A$, if and only if $\operatorname{rank} A = \operatorname{rank}(AA^\mathsf{T}) = \operatorname{rank}(A^\mathsf{T}A)$. □

4.3 Symmetric Preconditioning

A *preconditioner* is a mapping $A \mapsto h(A)$ for matrices A from a given class, where the goal is to achieve a certain property, either with certainty or with high probability. This property is typically an input condition from some computational technique. For a more elaborate and formal introduction into preconditioning we refer to [11]. Here, we restrict to preconditioners for achieving *generic rank profile* for symmetric matrices of the form $A = BB^\mathsf{T}$ over an arbitrary field of positive characteristic.

To ensure correctness of protocol ObliviousRGInverse, we need a preconditioner with the following three properties:

(i) achieves generic rank profile with high probability;
(ii) preserves symmetry, i.e., $h(A)$ is symmetric;
(iii) is *removable*. Informally speaking, this means that the preconditioner can be efficiently removed once "it has done its job". Formally, a preconditioner is removable with respect to computing a reflexive generalized inverse if there exists an efficiently computable mapping g such that $g(h(A)^-) \in \mathcal{A}^-$, where $\mathcal{A}^-$ denotes the set of reflexive generalized inverses of A.

Although several preconditioners for achieving generic rank profile have been proposed in the literature, we are not aware of an existing result that covers all of the above properties simultaneously. For example, the Toeplitz preconditioner by Kaltofen and Saunders [22] fails to satisfy (ii), and the diagonal preconditioner proposed in [17] (combined with a suitable linear-independence preconditioner, see [11]) fails to satisfy (iii).

In this section we will show that for a symmetric matrix A, the preconditioner $h(A) = UAU^\mathsf{T}$ with U a uniformly random (invertible) matrix is sufficient for satisfying (i)–(iii). It is easy to see that (ii) holds. We prove property (i) in Theorem 2 and (iii) in Lemma 8.

Lemma 6 (Schwartz–Zippel). *Let $g \in \mathbb{F}[x_1, \dots, x_n]$ be a nonzero polynomial of total degree $d \geq 0$ over a field $\mathbb{F}$. Let $\mathcal{S} \subseteq \mathbb{F}$ and let $\alpha_1, \dots, \alpha_n$ be chosen independently and uniformly at random from $\mathcal{S}$. Then,*

$$\Pr[g(\alpha_1, \dots, \alpha_n) = 0] \leq \frac{d}{|\mathcal{S}|}.$$

Lemma 7 (See, e.g., [8, Lem. 2-(iii)]). *The probability that a uniformly random matrix $U \in \mathbb{F}^{m \times m}$ is invertible equals*

$$\Pr(\det U \neq 0) = \prod_{k=1}^{m} \left(1 - |\mathbb{F}|^{-k}\right).$$

Theorem 2. *Let $A \in \mathbb{F}^{m \times n}$ be arbitrary, let r be the rank of A and let AA^T have the same rank as A. Let $U \in \mathbb{F}^{m \times m}$ be chosen uniformly at random. Then, the probability that U is invertible and $UAA^\mathsf{T}U^\mathsf{T}$ has generic rank profile is*

$$\Pr_U \left(\det U \neq 0 \wedge [UAA^\mathsf{T}U^\mathsf{T}]_k \neq 0 \quad \forall k \in [r]\right) > 1 - \frac{r(r+1)+2}{|\mathbb{F}|}.$$

Proof. We view $U = (u_{i,j})$ as a polynomial matrix with $u_{i,j}$ as indeterminates. For every $1 \leq k \leq r$, we apply the Cauchy–Binet formula to obtain an expression for the leading principal minor of order k of the matrix $UAA^{\mathsf{T}}U^{\mathsf{T}}$, which is a polynomial in the variables $u_{i,j}$, where we let $\mathcal{K} = [k]$,

$$\begin{aligned} f_k(u_{1,1}, \dots, u_{i,j}, \dots, u_{m,m}) &= [UAA^{\mathsf{T}}U^{\mathsf{T}}]_{\mathcal{K},\mathcal{K}} \\ &= \sum_{\substack{\mathcal{I} \subset [m] \\ |\mathcal{I}|=k}} [UA]_{\mathcal{K},\mathcal{I}} [A^{\mathsf{T}}U^{\mathsf{T}}]_{\mathcal{I},\mathcal{K}} = \sum_{\substack{\mathcal{I} \subset [m] \\ |\mathcal{I}|=k}} \big([UA]_{\mathcal{K},\mathcal{I}}\big)^2 \\ &= \sum_{\substack{\mathcal{I} \subset [m] \\ |\mathcal{I}|=k}} \Big(\sum_{\substack{\mathcal{J} \subset [m] \\ |\mathcal{J}|=k}} [U]_{\mathcal{K},\mathcal{J}} [A]_{\mathcal{J},\mathcal{I}} \Big)^2. \end{aligned}$$

It follows immediately from the structure of this formula that the total degree of f_k is $2k$.

Let us now prove that none of the polynomials f_k for all $1 \leq k \leq r$ is equal to the zero polynomial. Because AA^{T} is symmetric, there exists an invertible matrix $S = (s_{i,j})$ such that $SAA^{\mathsf{T}}S^{\mathsf{T}} = \mathit{\Lambda}$ where $\mathit{\Lambda} = \operatorname{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0)$ with $\lambda_i \neq 0$ for all $1 \leq i \leq r$ [1, Thm. 6]. Hence,

$$f_k(s_{1,1}, \dots, s_{i,j}, \dots, s_{m,m}) = \prod_{i=1}^{k} \lambda_i \neq 0 \qquad \forall k \in [r].$$

The Schwartz–Zippel lemma asserts that $\Pr[f_k(U_{1,1}, \dots, U_{m,m}) = 0] \leq \frac{2k}{|\mathbb{F}|}$, where the $U_{i,j}$ represent the elements of U when viewed as (uniformly random and independent) random variables. Hence, by applying the union bound over k we obtain

$$\Pr[f_1(U) \neq 0 \wedge \dots \wedge f_r(U) \neq 0] \geq 1 - \frac{\sum_{k=1}^{r} 2k}{|\mathbb{F}|} = 1 - \frac{r(r+1)}{|\mathbb{F}|}.$$

Combining this bound with that of Lemma 7 gives

$$\begin{aligned} &\Pr_U\big(\det U \neq 0 \wedge [UAA^{\mathsf{T}}U^{\mathsf{T}}]_k \neq 0 \quad \forall k \in [r]\big) \\ &\quad \geq \prod_{k=1}^{m} \big(1 - |\mathbb{F}|^{-k}\big) - \frac{r(r+1)}{|\mathbb{F}|} > \frac{|\mathbb{F}|-2}{|\mathbb{F}|-1} - \frac{r(r+1)}{|\mathbb{F}|} > 1 - \frac{r(r+1)+2}{|\mathbb{F}|}, \end{aligned}$$

where we used that

$$\begin{aligned} \prod_{k=1}^{m} (1 - x_k) > \prod_{k=1}^{\infty} (1 - x_k) &= 1 - x_1 - x_2(1 - x_1) - x_3(1 - x_1)(1 - x_2) - \dots \\ &> 1 - \sum_{k=1}^{\infty} x_k. \end{aligned}$$

With $x_k = |\mathbb{F}|^{-k}$, we get that $\prod_{k=1}^{m} \big(1 - |\mathbb{F}|^{-k}\big) > 1 - (|\mathbb{F}| - 1)^{-1}$. □

We now prove that the preconditioner $h(A) = UAU^\mathsf{T}$ with invertible U is removable.

Lemma 8. *Let A be a matrix over $\mathbb{F}$ and let $\mathcal{A}^-$ denote the set of reflexive generalized inverses of A. Let U be an invertible matrix over $\mathbb{F}$ and let $Y = (UAU^\mathsf{T})^-$ be a reflexive generalized inverse of UAU^T. Then, $U^\mathsf{T}YU \in \mathcal{A}^-$.*

Proof. Given the Penrose equations (1) and (2) for Y, we need to show that the Penrose equations (1) and (2) hold for A^-. Since U is invertible,

$$A(U^\mathsf{T}YU)A = U^{-1}(UAU^\mathsf{T})Y(UAU^\mathsf{T})(U^\mathsf{T})^{-1} = U^{-1}(UAU^\mathsf{T})(U^\mathsf{T})^{-1} = A.$$

Furthermore,

$$(U^\mathsf{T}YU)A(U^\mathsf{T}YU) = U^\mathsf{T}Y(UAU^\mathsf{T})YU = U^\mathsf{T}YU. \qquad \square$$

4.4 Construction

Our protocol Pseudoinverse, on input of a secret-shared matrix $\llbracket A \rrbracket \in \mathbb{F}_p^{m\times n}$, computes the pair $(\llbracket A^\dagger \rrbracket, \llbracket (\operatorname{vol} A)^2 \rrbracket)$ and is given as Protocol 3. Protocol Pseudoinverse makes use of a secure subprotocol Determinant for computing the determinant of an invertible matrix in $\mathbb{F}_p^{m\times m}$ in secret-shared form. A possible instantiation of Determinant can be found in [12], where it is called protocol Π_0. See also [6], which slightly modifies this protocol to reduce its randomness complexity.

Protocol 3 Pseudoinverse($\llbracket A \rrbracket$)

```
if $m > n$ then
    return Pseudoinverse($\llbracket A \rrbracket^\mathsf{T}$)$^\mathsf{T}$
$\llbracket AA^\mathsf{T} \rrbracket \leftarrow \llbracket A \rrbracket \llbracket A \rrbracket^\mathsf{T}$                                  ▷ symmetric
$\llbracket AA^\mathsf{T}AA^\mathsf{T} \rrbracket \leftarrow \llbracket AA^\mathsf{T} \rrbracket \llbracket AA^\mathsf{T} \rrbracket$                     ▷ symmetric
$U \xleftarrow{\$} \mathbb{F}_p^{m\times m}$
$\llbracket X \rrbracket \leftarrow U^\mathsf{T}$ObliviousRGInverse$(U\llbracket AA^\mathsf{T}AA^\mathsf{T} \rrbracket U^\mathsf{T})U$
$\llbracket XAA^\mathsf{T} \rrbracket \leftarrow \llbracket X \rrbracket \llbracket AA^\mathsf{T} \rrbracket$
$\llbracket A^\dagger \rrbracket \leftarrow \llbracket A^\mathsf{T} \rrbracket \llbracket XAA^\mathsf{T} \rrbracket$
$\llbracket K \rrbracket \leftarrow I - \llbracket AA^\mathsf{T} \rrbracket \llbracket XAA^\mathsf{T} \rrbracket$            ▷ symmetric; in parallel with $\llbracket A^\dagger \rrbracket$
$\llbracket d \rrbracket \leftarrow$ Determinant$(\llbracket AA^\mathsf{T} \rrbracket + \llbracket K \rrbracket)$
return $(\llbracket A^\dagger \rrbracket, \llbracket d \rrbracket)$
```

We note that the rank of A is given by $\operatorname{Tr}(AA^\dagger)$ [9]. It can be computed obliviously in Pseudoinverse as $\llbracket r \rrbracket = m - \operatorname{Tr}(\llbracket K \rrbracket)$.

Corollary 2. *Protocol* Pseudoinverse*, when run on an arbitrary $m \times n$ matrix over $\mathbb{F}_p$, correctly computes the Moore–Penrose pseudoinverse with probability at least*

$$\Pr(\mathit{success}) \geq \left[1 - \frac{m(m+1)+2}{|\mathbb{F}|}\right] \cdot P_{\mathsf{Determinant}},$$

where $P_{\mathsf{Determinant}}$ denotes the success probability of protocol Determinant*.*

4.5 Complexity Analysis

Proposition 5. *Protocol* Pseudoinverse*, when run on an arbitrary $m \times n$ matrix over $\mathbb{F}_p$, requires $mn + \frac{5}{2}m^2 + \frac{3}{2}m$ secure inner products (or: $m^2n + \frac{5}{2}m^3 + \frac{3}{2}m^2$ secure multiplications), one invocation of protocol* Determinant *on a symmetric $m \times m$ matrix and one invocation of* ObliviousRGInverse *on a symmetric $m \times m$ matrix.*

Protocol Determinant, instantiated as in [6], when invoked on a $m \times m$ matrix, requires secure sampling of m^2 random elements, and performing $2m^2 + m - 1$ secure inner products (or: $\frac{4}{3}m^3 + \frac{2}{3}m - 1$ secure multiplications) and m^2 open operations.

The field inversion technique from Bar-Ilan and Beaver [3] requires secure sampling of one random element and one secure multiply-and-open operation.

Subprotocol IsZero can be instantiated with the probabilistic secure zero test from Nishide and Ohta [30]. This secure zero test is constant round and requires 2κ secure multiplications, 4κ secure multiply-and-open operations and secure sampling of 5κ random elements, where κ is a security parameter and the protocol may fail with probability $2^{-\kappa} + 1/p$.

Corollary 3. *Protocol* Pseudoinverse*, when run on an arbitrary $m \times n$ matrix over $\mathbb{F}_p$, with protocol* Determinant *instantiated as in [6], requires in total $nm + 6m^2 + o(m^2)$ secure inner products (or: $nm^2 + \frac{13}{3}m^3 + o(m^3)$ secure multiplications), m^2 public random elements, m^2 private random elements, m^2 openings, m secure zero tests and m secure field inversions.*

If protocols IsZero *and* Reciprocal *are instantiated as the probabilistic zero test from [30] and as in [3], respectively, the m secure zero tests and field inversions require $O(\kappa m)$ secure multiplications, random elements and openings.*

Remark 3. It is straightforward to adapt Protocol 3 such that in line 8 it computes the vector $A^\dagger \boldsymbol{b}$ instead of the matrix $A^\dagger$, i.e., directly solving the linear system $A\boldsymbol{x} = \boldsymbol{b}$ for the vector $\boldsymbol{x}$. By replacing line 8, in case $m \leq n$, with the two lines $[\![XAA^\mathsf{T}\boldsymbol{b}]\!] \leftarrow [\![XAA^\mathsf{T}]\!][\![\boldsymbol{b}]\!]$ and $[\![A^\dagger\boldsymbol{b}]\!] \leftarrow [\![A^\mathsf{T}]\!][\![XAA^\mathsf{T}\boldsymbol{b}]\!]$, one can avoid the matrix-matrix product that gives rise to the mn term. Namely, the complexity (number of secure inner products) becomes $O(n + m^2)$. If $m > n$, then we would transpose the system to be solved: $\boldsymbol{x}^\mathsf{T}A^\mathsf{T} = \boldsymbol{b}^\mathsf{T}$. In this case, line 8 would be replaced by two secure products in which the matrix is multiplied from the *left* by the vector and this would result in a complexity of $O(n^2)$ secure inner products. Note, however, that this adaptation imposes an additional constraint on the size of the modulus; the field should now be large enough to uniquely represent the coefficients of the vector $dA^\dagger\boldsymbol{b}$.

Acknowledgements

We would like to thank Berry Schoenmakers for interesting discussions and valuable feedback.

References

1. Albert, A.A.: Symmetric and alternate matrices in an arbitrary field, I. Transactions of the American Mathematical Society **43**(3), 386–436 (1938)
2. Bapat, R.B., Rao, K.P.S.B., Prasad, K.M.: Generalized inverses over integral domains. Linear Algebra and its Applications **140**, 181–196 (1990)
3. Bar-Ilan, J., Beaver, D.: Non-cryptographic fault-tolerant computing in constant number of rounds of interaction. In: Proc. 8th Symp. on Princip. of Distr. Comp. pp. 201–209. ACM, NY (1989)
4. Ben-Israel, A.: A volume associated with $m \times n$ matrices. Linear Algebra and its Applications **167**, 87–111 (1992)
5. Ben-Israel, A., Greville, T.N.E.: Generalized Inverses - Theory and Applications. CMS Books in Mathematics, Springer (2003)
6. Blom, F., Bouman, N.J., Schoenmakers, B., de Vreede, N.: Efficient secure ridge regression from randomized Gaussian elimination. Cryptology ePrint Archive, Report 2019/773 (2019)
7. Bogdanov, D., Kamm, L., Laur, S., Sokk, V.: Rmind: A tool for cryptographically secure statistical analysis. IEEE Trans. Dependable Sec. Comput. **15**(3), 481–495 (2018)
8. Borodin, A., von zur Gathen, J., Hopcroft, J.: Fast parallel matrix and GCD computations. Information and Control **52**(3), 241–256 (1982)
9. Boullion, T.L., Odell, P.L.: Generalized Inverse Matrices. Wiley (1971)
10. Bouman, N.J., de Vreede, N.: New protocols for secure linear algebra: Pivoting-free elimination and fast block-recursive matrix decomposition. Cryptology ePrint Archive, Report 2018/703 (2018)
11. Chen, L., Eberly, W., Kaltofen, E., Saunders, B.D., Turner, W.J., Villard, G.: Efficient matrix preconditioners for black box linear algebra. Linear Algebra Its Appl. **343–344**, 119–146 (2002)
12. Cramer, R.J.F., Damgård, I.B.: Secure distributed linear algebra in a constant number of rounds. In: Proc. CRYPTO 2001, Santa Barbara, USA. pp. 119–136. Springer (2001)
13. Cramer, R.J.F., Damgård, I.B., Ishai, Y.: Share conversion, pseudorandom secret-sharing and applications to secure computation. In: Proc. TCC 2005. LNCS, vol. 3378, pp. 342–362. Springer (2005)
14. Cramer, R.J.F., Damgård, I.B., Nielsen, J.B.: Secure multiparty computation and secret sharing: An information theoretic approach. Cambridge University Press (2015)
15. Cramer, R.J.F., Kiltz, E., Padró, C.: A note on secure computation of the Moore–Penrose pseudoinverse and its application to secure linear algebra. In: Proc. CRYPTO 2007. vol. 4622, pp. 613–630. Springer (2007)
16. Damgård, I.B., Fitzi, M., Kiltz, E., Nielsen, J.B., Toft, T.: Unconditionally secure constant-rounds multi-party computation for equality, comparison, bits and exponentiation. In: Proc. TCC 2006. LNCS, vol. 3876, pp. 285–304. Springer (2006)
17. Eberly, W., Kaltofen, E.: On randomized Lanczos algorithms. In: Proc. ISSAC '97. pp. 176–183. ACM (1997)
18. Gascón, A., Schoppmann, P., Balle, B., Raykova, M., Doerner, J., Zahur, S., Evans, D.: Privacy-preserving distributed linear regression on high-dimensional data. PoPETs **2017**(4), 345–364 (2017)
19. Greville, T.: Note on the generalized inverse of a matrix product. SIAM Review **8**(4), 518–521 (1966)

20. Hartwig, R.E.: The reverse order law revisited. Linear Algebra and its Applications **76**, 241–246 (1986)
21. Kaltofen, E., Lobo, A.: On rank properties of Toeplitz matrices over finite fields. In: Proc. ISSAC '96. pp. 241–249. ACM (1996)
22. Kaltofen, E., Saunders, B.D.: On Wiedemann's method of solving sparse linear systems. In: Proc. 9th Int. Symp. AAECC. LNCS, vol. 539, pp. 29–38. Springer (1991)
23. Kiltz, E., Mohassel, P., Weinreb, E., Franklin, M.: Secure linear algebra using linearly recurrent sequences. In: Proc. TCC 2007. LNCS, vol. 4392, pp. 291–310. Springer (2007)
24. Malaschonok, G.: Fast generalized Bruhat decomposition. In: International Workshop on Computer Algebra in Scientific Computing. pp. 194–202. Springer (2010)
25. Marsaglia, G., Styan, G.P.H.: Equalities and inequalities for ranks of matrices. Linear and Multilinear Algebra **2**(3), 269–292 (1974)
26. Marsaglia, G., Styan, G.P.H.: Rank conditions for generalized inverses of partitioned matrices. Sankhyā: The Indian Journal of Statistics, Series A pp. 437–442 (1974)
27. Mohassel, P., Weinreb, E.: Efficient secure linear algebra in the presence of covert or computationally unbounded adversaries. In: Proc. CRYPTO 2008. LNCS, vol. 5157, pp. 481–496. Springer (2008)
28. Mulmuley, K.: A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. Combinatorica **7**(1), 101–104 (1987)
29. Nikolaenko, V., Weinsberg, U., Ioannidis, S., Joye, M., Boneh, D., Taft, N.: Privacy-preserving ridge regression on hundreds of millions of records. In: Proc. 2013 IEEE Symp. on Security and Privacy. pp. 334–348. IEEE (2013)
30. Nishide, T., Ohta, K.: Multiparty computation for interval, equality, and comparison without bit-decomposition protocol. In: Proc. PKC 2007. LNCS, vol. 4450, pp. 343–360. Springer (2007)
31. Nissim, K., Weinreb, E.: Communication efficient secure linear algebra. In: Proc. TCC 2006. LNCS, vol. 3876, pp. 522–541. Springer (2006)
32. Pearl, M.H.: Generalized inverses of matrices with entries taken from an arbitrary field. Linear Algebra and its Applications **1**(4), 571–587 (1968)
33. Rao, C.R.: Linear Statistical Inference and its Applications. Wiley (1973)
34. Rao, C.R., Mitra, S.K.: Generalized inverse of matrices and its applications. Wiley (1971)
35. Rohde, C.A.: Generalized inverses of partitioned matrices. Journal of the Society for Industrial and Applied Mathematics **13**(4), 1033–1035 (1965)
36. Springer, J.: Die exakte Berechnung der Moore–Penrose-Inversen einer Matrix durch Residuenarithmetik. Zeitschrift für Angewandte Mathematik und Mechanik **63**(3), 203–210 (1983)
37. Wang, P.S.: A *p*-adic algorithm for univariate partial fractions. In: Proc. SYMSAC '81. pp. 212–217. ACM (1981)

A Proofs of Proposition 2 and 3

The proof is by complete induction. It is clear that the number of invocations of ScalarRGInverse required when ObliviousRGInverse is run on an $m \times m$ matrix is equal to m. Let $D(m)$ denote the number of secure inner products required to run protocol ObliviousRGInverse on an $m \times m$ matrix. Similarly, let $M(m)$ be the number of secure multiplications required, in case a "cheap inner product" is not available. Then, we have to show that

$$D(m) < \frac{3}{2}m(m-1) + m\log_2 m; \text{ and} \tag{19}$$

$$M(m) \leq \frac{1}{2}m^3 + \frac{1}{2}m^2 - m. \tag{20}$$

Inspection of the protocol shows that

$$\begin{aligned} D(1) &= 0; \\ D(2k) &= 2D(k) + 3k^2 + k; \text{ and} \quad (21) \\ D(2k+1) &= D(k) + D(k+1) + 3k^2 + 4k + 1, \quad (22) \end{aligned}$$

where we distinguish between even ($m = 2k$) and odd ($m = 2k+1$) dimensions. Similarly, for $M(m)$, we have

$$\begin{aligned} M(1) &= 0; \\ M(2k) &\leq 2M(k) + 3k^3 + k^2; \text{ and} \quad (23) \\ M(2k+1) &\leq M(k) + M(k+1) + 3k^3 + \frac{11}{2}k^2 + \frac{5}{2}k. \quad (24) \end{aligned}$$

The inequalities in (23) and (24) can be replaced with equalities in case the naïve algorithm for matrix multiplication is used.

In the base case $m = 1$, the propositions clearly hold. Assume the propositions hold for all $m' < m$. Then for odd $m = 2k+1$ substitution of (19) in (22) yields

$$\begin{aligned} D(2k+1) &< \frac{3}{2}m(m-1) + k + 1 + k\log_2 k + (k+1)\log_2(k+1) \\ &= \frac{3}{2}m(m-1) + \log_2\left(k^k(k+1)^{k+1}2^{k+1}\right) \\ &< \frac{3}{2}m(m-1) + (2k+1)\log_2(2k+1), \end{aligned}$$

where the last inequality follows from monotonicity of the logarithm and from the following inequality:

$$\begin{aligned} 2^{k+1}k^k(k+1)^{k+1} &= 2^{1-k}(2k+1-1)^k(2k+1+1)^k(k+1) \\ &= 2^{1-k}((2k+1)^2-1)^k(k+1) \\ &< 2^{1-k}(2k+1)^{2k}(k+1) \\ &< (2k+1)^{2k+1}. \end{aligned}$$

For the case of even m, Proposition 2 follows immediately by substituting equation (19) in (21). Similarly, Proposition 3 follows from the substitution of (20) in (23) and (24).