HAL open science

# Essentially optimal interactive certificates in linear algebra

Jean-Guillaume Dumas, Erich Kaltofen

**To cite this version:**

Jean-Guillaume Dumas, Erich Kaltofen. Essentially optimal interactive certificates in linear algebra. ACM International Symposium on Symbolic and Algebraic Computation (ISSAC 2014), Jul 2014, Kobe, Japan. pp.146-153, 10.1145/2608628.2608644 . hal-00932846v4

**HAL Id: hal-00932846**

**https://hal.science/hal-00932846v4**

Submitted on 24 Dec 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Essentially Optimal Interactive Certificates in Linear Algebra*

Jean-Guillaume Dumas[†] Erich Kaltofen[‡]

December 24, 2019

**Abstract**

Certificates to a linear algebra computation are additional data structures for each output, which can be used by a—possibly randomized—verification algorithm that proves the correctness of each output. The certificates are essentially optimal if the time (and space) complexity of verification is essentially linear in the input size $N$, meaning $N$ times a factor $N^{o(1)}$, i.e., a factor $N^{\eta(N)}$ with $\lim_{N\to\infty} \eta(N) = 0$.

We give algorithms that compute essentially optimal certificates for the positive semidefiniteness, Frobenius form, characteristic and minimal polynomial of an $n \times n$ dense integer matrix $A$. Our certificates can be verified in Monte-Carlo bit complexity $(n^2 \log \|A\|)^{1+o(1)}$, where $\log \|A\|$ is the bit size of the integer entries, solving an open problem in [Kaltofen, Nehring, Saunders, Proc. ISSAC 2011] subject to computational hardness assumptions.

Second, we give algorithms that compute certificates for the rank of sparse or structured $n \times n$ matrices over an abstract field, whose Monte Carlo verification complexity is 2 matrix-times-vector products + $n^{1+o(1)}$ arithmetic operations in the field. For example, if the $n \times n$ input matrix is sparse with $n^{1+o(1)}$ non-zero entries, our rank certificate can be verified in $n^{1+o(1)}$ field operations.

This extends also to integer matrices with only an extra $\log \|A\|^{1+o(1)}$ factor.

All our certificates are based on interactive verification protocols with the interaction removed by a Fiat-Shamir identification heuristic. The validity of our verification procedure is subject to standard computational hardness assumptions from cryptography.

[†]Université de Grenoble. Laboratoire LJK, umr CNRS, INRIA, UJF, UPMF, GINP. 51, av. des Mathématiques, F38041 Grenoble, France. Jean-Guillaume.Dumas@imag.fr, ljk.imag.fr/membres/Jean-Guillaume.Dumas/

[‡]Department of Mathematics. North Carolina State University. Raleigh, NC 27695-8205, USA. kaltofen@math.ncsu.edu, www.kaltofen.us

*This research was supported in part by the Agence Nationale pour la Recherche under Grant ANR-11-BS02-013 HPAC (Dumas) and the National Science Foundation under Grant CCF-1115772 (Kaltofen).

# 1 Introduction

Suppose you want to externalize your computations to cloud services. Prior to payment of the services, it would be desirable to verify that the returned result has been correctly computed by the cloud servers. This model is economically viable only if the verification process requires less resources than the computation itself. It is therefore important to design *certificates* that can be used to verify a result at a lower cost than that of recomputing it.

For instance, a primality certificate is a formal proof that a number is prime. In [18], primality is assessed by presenting a primitive root and the factorization of $m-1$. The latter can be checked fast by remultiplying, and then primitivity is polynomially checkable.

In linear algebra our original motivation is related to sum-of-squares. By Artin's solution to Hilbert 17th Problem, any polynomial inequality $\forall \xi_1, \ldots, \xi_n \in \mathbb{R}, f(\xi_1, \ldots, \xi_n) \geq g(\xi_1, \ldots, \xi_n)$ can be proved by a fraction of sum-of-squares:

$$\exists u_i, v_j \in \mathbb{R}[x_1, \ldots, x_n], f - g = \left(\sum_{i=1}^{\ell} u_i^2\right) / \left(\sum_{j=1}^{m} v_j^2\right) \tag{1}$$

Such proofs can be used to establish global minimality for

$g = \inf_{\xi_v \in \mathbb{R}} f(\xi_1, \ldots, \xi_n)$ and constitute certificates in non-linear global optimization. A symmetric integer matrix $W \in \mathbb{SZ}^{n \times n}$ is positive semidefinite, denoted by $W \succeq 0$, if all its eigenvalues, which then must be real numbers, are non-negative. Then, a certificate for positive semidefiniteness of rational matrices constitutes, by its Cholesky factorizability, the final step in an exact rational sum-of-squares proof, namely

$$\begin{aligned} &\exists e \geq 0,\ W^{[1]} \succeq 0,\ W^{[2]} \succeq 0,\ W^{[2]} \neq \mathbf{0}: \\ &\qquad (f-g)(x_1, \ldots, x_n) \cdot (m_e(x_1, \ldots, x_n)^T W^{[2]} m_e(x_1, \ldots, x_n)) = \\ &\qquad\qquad m_d(x_1, \ldots, x_n)^T W^{[1]} m_d(x_1, \ldots, x_n), \end{aligned} \tag{2}$$

where the entries in the vectors $m_d, m_e$ are the terms occurring in $u_i, v_j$ in (1). In fact, (2) is the semidefinite program that one solves.

Thus arose the question how to give possibly probabilistically checkable certificates for linear algebra problems. In [15] the certificates are restricted to those that are checkable in essentially optimal time, that is, in bit complexity $(n^2 \log \|W\|)^{1+o(1)}$, where $\log \|W\|$ is the bit size of the entries in $W$. Quadratic time is feasible because a matrix multiplication $AB$ can be certified by the resulting product matrix $C$ via Rusin Freivalds's [9] (see also [16]) probabilistic check: check $A(Bv) = Cv$ for a random vector $v$.

Note that programs that check their results from [5] have the higher matrix-multiplication time complexity. In [15] a certificate for matrix rank was presented, based on Storjohann's Las Vegas rank algorithm [21], but matrix positive semidefiniteness remained open. Also the presented certificate for the rank did not take into account a possible structure in the matrix.

In the following we solve these two problems. In both cases, positive semidefiniteness and structured or blackbox matrices, our solution is to use either *interactive* certificates under the random oracle model, or heuristics under standard computational hardness assumptions from cryptography. Removing the cryptographic assumptions remains however a fundamental open problem. Providing certificates to other problems, such as the determinant or the minimal and characteristic polynomial of blackbox matrices, is also open.

In Section 2, we detail the different notions of certification that can be used and in particular the relaxation we make over the certificates of [15]: in the certificates presented here, we allow the verifier to provide the random bits used by the prover, in an interactive manner. We also present in this section the Fiat-Shamir derandomization heuristic that can turn any interactive certificate into a non-interactive heuristic certificate.

More precisely, the idea is to devise an interactive protocol for the random oracle model, and then to replace oracle accesses by the computation of an appropriately chosen function $h$ [7, 2].

Then we first present in Section 3 an interactive certificate for the Frobenius normal form that can be verified in $O(n^{2+o(1)}(\log \|A\|)^{1+o(1)})$ binary operations, as in [15], but our new certificate also occupies an optimal space of $O(n^{2+o(1)}(\log \|A\|)^{1+o(1)})$ bits. This is an order of magnitude improvement over [15, Theorem 4]. This certificate can then be used as in the latter paper to certify the minimal and characteristic polynomial as well as positive semidefiniteness, while keeping the lower memory requirements. In the same section we also present another, stand-alone, characteristic polynomial certificate, which can also be used for positive semidefiniteness, with slightly smaller random evaluation points.

Finally in Section 4 we present a new certificate for the rank of sparse or structured matrices.

The certificate combines an interactive certificate of non-singularity, giving a lower bound to the rank, with an interactive certificate for an upper bound to the rank. Overall the interactive certificate for the rank requires only $2\Omega + n^{1+o(1)}$ arithmetic operations over any coefficient domain, where $\Omega$ is the number of operations required to perform one matrix-times-vector product.

This certificate is then extended to work over the integers with only an extra $\log \|A\|^{1+o(1)}$ factor. For instance, if the matrix is sparse with only $n^{1+o(1)}$ nonzero elements, then the certificate verification is essentially linear.

## 2 Notions of certificate

The ideas in this paper arise from linear algebra, probabilistic algorithms, program testing and cryptography.

We will in particular combine:

- the notions of certificates for linear algebra of Kaltofen et al. [15], which extend the randomized algorithms of Freivalds [9],

and reduce the computation cost of the program checkers of Blum and Kannan [5],

- with probabilistic interactive proofs of Babai [1] and Goldwasser et al. [13],
- as well as Fiat-Shamir heuristic [7, 2] turning interactive certificates into non-interactive heuristics subject to computational hardness.

We first recall some of these notions and then define in Section 2.3 what we mean by perfectly complete, sound and efficient interactive certificates.

## 2.1 Arthur-Merlin interactive proof systems

A proof system usually has two parts, a theorem $T$ and a proof $\Pi$, and the validity of the proof can be checked by a verifier $V$. Now, an *interactive proof*, or a $\sum$*-protocol*, is a dialogue between a prover $P$ (or *Peggy* in the following) and a verifier $V$ (or *Victor* in the following), where $V$ can ask a series of questions, or challenges, $q_1$, $q_2$, ... and $P$ can respond alternatively with a series of strings $\pi_1$, $\pi_2$, ..., the responses, in order to prove the theorem $T$. The theorem is sometimes decomposed into two parts, the hypothesis, or input, $H$, and the commitment, $C$. Then the verifier can accept or reject the proof: $V(H, C, q_1, \pi_1, q_2, \pi_2, \ldots) \in \{\text{accept}, \text{reject}\}$.

To be useful, such proof systems should satisfy *completeness* (the prover can convince the verifier that a true statement is indeed true) and soundness (the prover cannot convince the verifier that a false statement is true). More precisely, the protocol is *complete* if the probability that a true statement is rejected by the verifier can be made arbitrarily small. Similarly, the protocol is *sound* if the probability that a false statement is accepted by the verifier can be made arbitrarily small. The completeness (resp. soundness) is *perfect* if accepted (resp. rejected) statement are always true (resp. false).

It turns out that interactive proofs with perfect completeness are as powerful as interactive proofs [10]. Thus in the following, as we want to prove correctness of a result more than proving knowledge of it, we will only use interactive proofs with perfect completeness.

On the one hand, if a protocol is both perfectly complete and perfectly sound then it is deterministic. On the other hand, if at least one of completeness and soundness is not perfect, then the proof is probabilistic and correspond to Monte Carlo algorithms (always fast, probably correct).

After submitting our paper to ISSAC, we learned of related work on certifying the evaluation of a Boolean or arithmetic circuit by multi-round interactive protocols GKR'08 [12] and Thaler'13 [22], where the verifier runs in essentially linear time in the input size and the prover is limited in compute power. Since our computational problems in linear algebra can be solved by a Boolean or arithmetic circuit of size $S(n)$, where $S(n)$ is the bit or arithmetic computation time, the GKR'08 and Thaler'13 protocols constitute an alternate (implicit) approach to certification of the output. Those protocols (for multiple rounds) have smaller communication cost, namely, linear in the depth of the circuit, than

ours, which can be linear or quadratic in the matrix dimension. But GKR'08 and Thaler'13 may require a more powerful prover: their provers may require a factor of $O(\log(S(n)))$ or a constant factor more time than $S(n)$. Several of our certificates are computed faster, in $S(n) + o(S(n))$ time, where $S(n)$ is the sequential time, e.g., our certificate for positive semidefiniteness. A technical condition for the GKR'08 and Thaler'13 verifier is that its complexity depends linearly on the depth of the circuit, so an algorithm of depth $\Omega(n^{2.1})$ would not satisfy our requirement of linear complexity for the verifier, but none of our algorithms have that much depth.

More significantly, our protocols and certificates are explicit: the verifier need not know the circuit for the computation, nor compute certain properties of it. Nevertheless, GKR'08 [12] and Thaler'13 [22] guarantee multi-round protocols, and motivate the search for explicit certificates.

In our explicit setting, we *require* that the verifier algorithm has lower computational complexity than any known algorithm computing the property.

Moreover, we want the complexity of the prover algorithm as close as possible to the best known algorithm computing the property without certificates.

## 2.2 Certificates in linear algebra

For Blum and Kannan [5] a program checker for a program $P$ is itself a program $C$. For any instance $I$ on which program $P$ is run, $C$ is run subsequently. $C$ either certifies that the program $P$ is correct on $I$ or declares $P$ to be buggy. There, the programs can be rerun on modified inputs, as in their matrix rank check, and thus might require more time to check their work than to do the work itself.

On the contrary, in [14, 15], a certificate for a problem that is given by input/output specifications is an input-dependent data structure and an algorithm that computes from that input and its certificate the specified output, and that has lower computational complexity than any known algorithm that does the same when only receiving the input. Correctness of the data structure is not assumed but validated by the algorithm. With respect to interactive proofs, the input/output is related to the property to be proven together with the commitment. However, as no interaction is possible between the prover and the verifier, this amounts to using a single round protocol where the prover sends only the commitment and then the verifier accepts it or not.

In this paper we use a modified version where we allow interactive exchanges between the prover and the verifier but preserve the requirements on lower total complexity for the verifier. Moreover, we then can convert back these two-rounds protocol into one round protocols via Fiat-Shamir heuristic: hash the input and commitment with an unpredictable and universal hash function (such as a cryptographic hash function), to simulate the random challenges proposed by the verifier.

It turns out that it seems easier to design certificates that are interactive than to design directly single round certificates. This could be related to the power

of the interactive proof system complexity class (IP) and the probabilistically checkable proofs (PCP).

## 2.3 Interactive certificates

There exists ways to reduce interactive proofs with $k$ rounds to interactive proofs with perfect completeness and 2 rounds, by increasing the verifier's complexity by time exponential in $k$. Here we will limit ourselves to 2 rounds for our definition of interactive certificates.

More precisely, in the following we use interactive certificates of a given property, mainly as *two-rounds probabilistic $\sum$-protocols with perfect completeness*:

1. The prover of a property sends a commitment to the verifier.
2. The verifier sends back a (randomly sampled) challenge, potentially depending on both the property and the commitment.
3. The prover completes the protocol with a response convincing the verifier of the property.

In order to become an *interactive certificate*, this two round $\sum$-protocol should then satisfy soundness, perfect completeness and efficiency as follows:

i. The protocol is *perfectly complete*: a true statement will always be accepted by the verifier.

ii. The protocol is *sound*: the probability that a false statement will be accepted by the verifier can be made arbitrarily small.

iii. The protocol is *efficient*: the verifier has lower computational complexity than any known algorithm that computes the true statement when only receiving the input.

The interactive certificate can also be said to be *essentially optimal* when the verifier needs only time and space complexity of the same order of magnitude as the size of the input and output to verify the latter.

With this relaxed model, we are able in the following to improve on some space complexities for integer linear algebra problems and also on time complexities for some problems over generic domains, like the rank of blackbox matrices.

## 2.4 Fiat-Shamir derandomization into a single heuristic round

In a practical perspective (say when using a compiled library, rather than an interpreter; or when posting the certificate in question) it is not always possible for the verifier (a user wanting a result) to interact with the prover (the program).

Then, there is always the possibility to transform an interactive certificate into a non-interactive heuristic. Here we use the strong Fiat-Shamir heuristic [7, 2, 3], where the random challenge message of the verifier is replaced by a cryptographic hash of the property *and* the commitment message. In practice, the cryptographic hash can be used as a seed for a pseudo-randomly generated sequence that the prover can generate a priori. For an a posteriori verification, the verifier decides whether to accept or not the certificate, as in two rounds interactive protocols, but has also to check that the challenge used by the prover has really been generated using the input and commitment as seeds.

In this setting, breaking the protocol is somewhat equivalent to breaking the cryptographic hash function: finding a combination of input and false commitment that will be accepted by the verifier relates to knowing in advance some parts of the output of the hash function. See for instance Section 4.5 where breaking the protocol is equivalent to predicting the value of some bits in a hash, and that can for instance be used to factor integers if Blumb-Blum-Shub hash function is used.

Note that it is important to use the *strong* heuristic that uses a combination of both the input and the commitment for the hashing. See for instance Section 3.2 where we need the result itself to be part of the seed in order to obtain a correct certificate.

# 3 Reducing space with respect to non-interactive certificates over the integers

## 3.1 Interactive certificate for residue systems

In [15, Theorem 5], the given certificates for the rank and determinant of an integer matrix are essentially optimal whereas the certificates for the Frobenius normal form (without transformation matrices), the characteristic and minimal polynomial and positive semidefiniteness are not: they require residue systems that occupies cubic bit space whereas the input and results occupy only a quadratic number of bits.

Those residue systems allow the verifier to check an integer matrix factorization ($A = LU$ for Gaussian elimination or $A = SFS^{-1}$ for the Frobenius form) where the resulting factors are in general of cubic size (quadratic number of entries but each one with linear magnitude) via Freivalds' certificate. The trick is to store these factorizations modulo many distinct primes. Then if the integer matrix factorization is not correct it means that $A - LU$ (resp. $A - SFS^{-1}$) is non zero. Therefore, from a bound on the maximal possible size of this difference (roughly cubic), it cannot be zero modulo a large number of primes. Consequently, if the set of distinct primes is larger than the bound, selecting a random prime $p$ in the set and checking whether $A - LU$ (resp. $A - SFS^{-1}$) is zero modulo $p$ would reveal the false statement with high probability.

Our idea here is to use several rounds interactive certificates: instead of storing the factorizations modulo many distinct primes, just compute them on

demand of the verifier. The verifier has just to select random primes and the prover will respond with the factorization modulo these primes.

**Theorem 1.** *Let $A \in \mathbb{Z}^{n\times n}$. There exists an* interactive certificate *for the Frobenius normal form, the characteristic or minimal polynomial of $A$. The interactive certificate can be verified in $n^{2+o(1)}(\log \|A\|)^{1+o(1)}$ bit operations and occupies $n^{2+o(1)}(\log \|A\|)^{1+o(1)}$ bit space.*

*Proof.* Use the same algorithm as in [15, Theorem 4] but replacing the random choice by the verifier of a given tuple $(p, S_p, F_p, T_p)$ (where $T_p \equiv S_p^{-1} \mod p$) by the choice of a random prime $p$ by the verifier and a response of a corresponding $(S_p, F_p, T_p)$ modulo $p$ by the prover. □

**Corollary 1.** *There exists a* non-interactive heuristic certificate *for the Frobenius normal form, the characteristic or minimal polynomial that occupy the same space and can be verified in the same time.*

*Proof.* We use Fiat-Shamir. The prover:

1. computes the integer Frobenius normal form $F$ (or the characteristic or minimal polynomial) over the integers;
2. then he chooses a cryptographic hash function and a pseudo-random prime generator;
3. he computes the hash of the input matrix together with the result;
4. this hash is used as a seed for the pseudo-random prime generator to generate one (or a constant number of) prime number(s);
5. the prover finally produces the Frobenius normal form and the change of basis modulo that prime(s).

The certificate is then composed of the input, the output, the hash function, the pseudo-random prime generator, the generated prime numbers and the associated triples $(S_p, F_p, T_p)$.

The verifier then:

1. checks that the hash function and the pseudo-random prime generator are well-known, cryptographically secure, functions;
2. checks that he can recover the primes via hashing the combination of the input and the output;
3. and verifies the zero equivalence modulo $p$ of $(F - F_p) \mod p$, $(S_pT_p - I) \mod p$ and $(S_pF_pT_p - A) \mod p$.

□

## 3.2 Direct interactive certificate for the characteristic polynomial and positive definiteness of integer matrices

In [15], the certificate for characteristic polynomial occupies roughly $n^{3+o(1)}$ bit space as it requires the Frobenius matrix normal form with a similarity residue system with primes bounded by $O(n(\log(n) + \log \|A\|))$.

As shown in Theorem 1, with an interactive certificate and a random oracle for the choice of prime numbers of the latter size, this yields an interactive certificate with only $n^{2+o(1)}$ bit space requirements.

We propose in the following Figure 1 a simpler certificate, still relying on the determinant certificate, but with evaluation points bounded only by $O(n)$. This gives a similar but smaller $o(1)$ factor in the complexity.

| | *Peggy* | | *Victor* |
|---|---|---|---|
| *Input* | | $A \in \mathbb{Z}^{n\times n}$ | |
| *Commitment* | $g \in \mathbb{Z}[X] = charpoly_A$ | $\xrightarrow{1:\ g(X)}$ | $degree(g) \stackrel{?}{=} n$ |
| *Challenge* | | $\xleftarrow{2:\ \lambda}$ | $\lambda \in \mathbb{Z}$ |
| *Response* | $\delta \in \mathbb{Z} = det(\lambda I - A)$ | $\xrightarrow{3:\ \delta}$ | $\delta \stackrel{?}{=} g(\lambda)$ |
| | $C : Cert(\delta = det(\lambda I - A))$ | $\xrightarrow{4:\ C}$ | $\delta \stackrel{?}{=} det(\lambda I - A)$ |

Figure 1: Interactive certificate for the characteristic polynomial

**Theorem 2.** *For $A \in \mathbb{Z}^{n\times n}$, the interactive certificate of Figure 1 for the characteristic polynomial is sound, perfectly complete and the number of operations performed by the verifier, as well as the bit space required to store this certificate, is bounded by $n^{2+o(1)}(\log \|A\|)^{1+o(1)}$.*

*Proof.* For the determinant certificate we use [15, Theorem 5] whose complexity matches that of the present theorem.

If Peggy is honest then the definition of the characteristic polynomial yields $charpoly_A = det(XI - A)$ and thus the protocol is perfectly complete.

If Peggy is dishonest then $g - charpoly_A$ being of degree at most $n$, it has at most $n$ roots. Thus if Victor samples random elements among the first say $cn$ integers, *after* the commitment $g$, the probability that Victor accepts the certificate is less than $1/c$. If the protocol is repeated $k$ times with independent draws of $\lambda$, then the probability that Victor accepts it $k$ times is lower than $\left(\frac{1}{c}\right)^k$ and therefore the protocol is sound.

For the complexity, one chooses a constant $c > 2$ so that $\lambda$ has $O(\log(n))$ bits. Thus $\delta$, as the determinant of $\lambda I - A$, is bounded by Hadamard's bound

to $O(n \log(\|A\| + n))$ bits. With Horner evaluation and Chinese remaindering, the check $g(\lambda) \stackrel{?}{=} \delta$ can thus be performed in less than $O(n^2 \log(\|A\| + n))$ operations. This is within the announced bound. □

**Corollary 2.** *Let $A$ be an $n \times n$ symmetric matrix having minors bound $H_A$ of bit length $\log_2(H_A) = n^{1+o(1)}$. The signature of $A$ can be verified by an interactive certificate in $n^{2+o(1)}$ binary operations with a $n^{2+o(1)}$ bit space characteristic polynomial certificate. Thus the same certificate serves for positive or negative definiteness or semidefiniteness.*

*Proof.* We just use the certificate of [15, Corollary 1] but replace their characteristic polynomial certificate by the interactive one of Figure 1 and Theorem 2. □

# 4 Interactive certificate for the rank of sparse matrices

Now we turn to matrices over any domain and count arithmetic operations instead of bit complexity. That is to say we consider that the four arithmetic operations over the domain, plus say equality testing, random sampling of one element, etc., are all counted as 1 operation.

For the sake of simplicity we will use the notation $\mathbb{F}$ as for finite fields but the results are valid over any abstract field, provided that the random sampling is done on a finite subset $S$ of the domain.

We improve on $O(n^2)$ certificates for the rank (given with say an $LU$ factorization), when the matrix is sparse, structured or given as blackbox. That is to say when the product of the matrix by a vector requires strictly less arithmetic operations than what would be required if the matrix was dense.

If the matrix is given as a blackbox, then the only possible operation with the matrix is the latter matrix-times-vector product.

For a matrix of rank $r$, if $\Omega$ is the cost of one of those matrix-times-vector product, the blackbox certificates of [19] would also require $O(nr)$ extra arithmetic operations and at least $O(r)$ extra matrix-times-vector products for a total of $O(r\Omega + nr)$ arithmetic operations.

In the following we show that it is possible to reduce the time and space complexity bounds of verifying certificates for the rank of blackbox matrices to only $2\Omega + n^{1+o(1)}$ arithmetic operations. This is essentially optimal, e.g. for sparse matrices, as reading and storing a matrix of dimensions $n \times n$ should also require $O(\Omega + n)$ operations.

We then extend this result, but in bit complexity, to also certify the rank of integer matrices with essentially the same optimal complexity bounds.

We proceed in two steps. First we certify that there exists an $r \times r$ nonsingular minor in the matrix. Second, we precondition the matrix so that it is of generic rank profile and exhibit a vector in the null-space of the leading $(r+1) \times (r+1)$ minor of the preconditioned matrix.

## 4.1 Certifying non-singularity

| | *Peggy* | | *Victor* |
|---|---|---|---|
| *Input* | | $A \in \mathbb{F}^{n\times n}$ | |
| *Commitment* | | $1:$ non-singular $\longrightarrow$ | |
| *Challenge* | | $\longleftarrow$ $2:\ b$ | $b \in S^n \subset \mathbb{F}^n$ |
| *Response* | $w \in \mathbb{F}^n$ | $3:\ w$ $\longrightarrow$ | $Aw \stackrel{?}{=} b$ |

Figure 2: Blackbox interactive certificate of non-singularity

**Theorem 3.** *Let $S$ be a finite subset of $\mathbb{F}$ with at least two distinct elements. For $A \in \mathbb{F}^{n\times n}$, whose matrix-times-vector products costs $\Omega$ operations in $\mathbb{F}$, the interactive certificate of Figure 2 for non-singularity is sound, perfectly complete and the number of arithmetic operations performed by the verifier is bounded by $\Omega + n^{1+o(1)}$.*

*Proof.* If Peggy is honest, then she can solve the system with an invertible matrix and provide $w = A^{-1}b$ to Victor. Therefore the protocol is perfectly complete.

If Peggy is dishonest, then it means that $A$ is singular. Therefore, it means that the rank of $A$ is at most $n-1$.

We use, e.g., Gaussian elimination to get $A = PLUQ$, where $P$ and $Q$ are permutation matrices, $L$ is unit invertible lower triangular and $U$ is upper triangular. As the rank of $A$ is at most $n-1$, $U$ is of the form $\left[\begin{smallmatrix} U_1 & U_2 \\ 0 & 0 \end{smallmatrix}\right]$ where $U_1 \in \mathbb{F}^{(n-1)\times(n-1)}$ is upper triangular. One then sees that making the system inconsistent is equivalent to setting to zero at least the last entry of $L^{-1}P^{-1}b$ in $\mathbb{F}^n$.

Thus, with probability at least $1 - 1/|S|$, the challenge vector proposed by Victor makes the system inconsistent*. In the latter case, Peggy will never be able to find a solution to the system.

Thus Victor can accept the certificate of Peggy only when he has randomly found a consistent vector. The probability that this happens $k$ times with $k$ independent selections of $b$ is bounded by $\frac{1}{|S|^k}$. Therefore, when the matrix is singular, Victor can accept repeated applications of the protocol only with negligible probability and the protocol is sound.

*Alternatively, one can get the same result with the Schwartz-Zippel lemma applied to the linear function whose kernel is the nullspace of $A$ as in e.g. [11, Theorem 2.2].

For the complexity, Victor needs to perform one matrix-times-vector product with $A$, of arithmetic complexity $\Omega$. Victor also needs to produce a random vector of size $n$ of elements in $S$ and perform a vector equality comparison.

□

## 4.2 Certifying an upper bound for the rank

For an upper bound, we precondition $A \in \mathbb{F}^{m\times n}$ of rank $r$ so that the leading $r\times r$ minor of the preconditioned matrix is non-zero and then present a non-zero vector in the nullspace of the $r+1$ leading minor. We use the butterfly probabilistic preconditioners of [6, Theorem 6.3] that can precondition an $n\times n$ matrix of rank $r$ so that the first $r$ rows of the preconditioned matrix become linearly independent with high probability. We denote by $\mathbb{B}_S^{n\times n}$ the set of such butterfly networks composed by less than $n(\log_2(n))$ switches of the form $\begin{bmatrix} 1 & \alpha \\ 1 & 1+\alpha \end{bmatrix}$, for $\alpha \in S \subset \mathbb{F}$. Choosing a random butterfly reduces to choosing an element $\alpha$, a row index and a column index for each of its switches.

[6, Theorem 6.3] works for square matrices but can easily be extended to work for rectangular matrices as follows.

**Lemma 1.** *Let $\mathbb{F}$ be a field and $S$ be a finite subset of $\mathbb{F}$. Let $A$ be an $m\times n$ matrix over $\mathbb{F}$ with $r$ linearly independent rows. Then a butterfly preconditioner $U \in \mathbb{B}_S^{m\times m}$ will make the first $r$ rows of $UA$ linearly independent with probability not less than $1-\frac{r\lceil\log_2(m)\rceil}{|S|}$.*

*Proof.* From [6, Theorem 6.2], we know that a depth $\lceil\log_2(m)\rceil$ butterfly network can switch any $r \le m$ indices into the continuous block $1, 2, \ldots, r$. The proof of [6, Theorem 6.3] uses the butterfly to permute and combine the rows of $A$ independently of its number of columns. Therefore $UA$ will have its first $r$ rows linearly independent with probability not less than $1-\frac{r\lceil\log_2(m)\rceil}{|S|}$. □

We can thus now turn to our interactive certificate of an upper bound to the rank.

| | *Peggy* | | *Victor* |
|---|---|---|---|
| *Input* | | $A \in \mathbb{F}^{m\times n}$ | |
| *Commitment* | $rank(A) \le r < \min\{m,n\}$ | $\xrightarrow{1:\ r}$ | |
| *Challenge* | | $\xleftarrow{2:\ U,V}$ | $U \in \mathbb{B}_S^{m\times m}, V \in \mathbb{B}_S^{n\times n}, S \subset \mathbb{F}$ |
| *Response* | $w \in \mathbb{F}^{r+1} \neq 0$ | $\xrightarrow{3:\ w}$ | $w \overset{?}{\neq} 0$ |
| | | | $[I_{r+1}\vert 0]\,UAV \begin{bmatrix} I_{r+1} \\ 0 \end{bmatrix} w \overset{?}{=} 0$ |

Figure 3: Blackbox interactive certificate for an upper bound to the rank

**Theorem 4.** *Let $A \in \mathbb{F}^{m\times n}$, whose matrix-times-vector products costs $\Omega$ operations in $\mathbb{F}$ and let $S$ be a finite subset of $\mathbb{F}$ with $|S| > 2\min\{m,n\}(\lceil\log_2(m)\rceil + \lceil\log_2(n)\rceil)$. The interactive certificate of Figure 3 proving an upper bound for the rank of $A$ is sound, perfectly complete and the number of arithmetic operations performed by the verifier is bounded by*

$\Omega + (m+n)^{1+o(1)}$.

*Proof.* If Peggy is honest this means that the rank of $A$ is upper bounded by $r < \min\{m,n\}$. Thus the rank of $M = [I_{r+1}|0]\,UAV \begin{bmatrix} I_{r+1} \\ 0 \end{bmatrix} \in \mathbb{F}^{(r+1)\times(r+1)}$ is also upper bounded by $r$. Therefore, there exist at least one non-zero vector $w$ in the nullspace of $M$. Hence Peggy can produce it and the protocol is perfectly complete.

If Peggy is dishonest, this means that the rank of $A$ is at least $r+1$.

Now, from Lemma 1, the butterfly preconditioner $U \in \mathbb{B}_S^{m\times m}$ will make the first $r+1$ rows of $UA$ linearly dependent with probability less than $\frac{(r+1)\lceil\log_2(m)\rceil}{|S|}$. Similarly the butterfly preconditioner $V \in \mathbb{B}_S^{n\times n}$ will make the first $r+1$ columns of $AV$ linearly dependent with probability less than $\frac{(r+1)\lceil\log_2(n)\rceil}{|S|}$. Overall the $(r+1)\times(r+1)$ leading principal minor of $UAV$ will be non-zero with probability at least $1 - \frac{(r+1)(\lceil\log_2(m)\rceil+\lceil\log_2(n)\rceil)}{|S|}$. The latter is greater than $1 - \frac{\min\{m,n\}(\lceil\log_2(m)\rceil+\lceil\log_2(n)\rceil)}{|S|} \geq \frac{1}{2}$. In this case the minor is invertible and Peggy will never be able to produce a non-zero vector in its kernel. The only possibility for Victor to accept the certificate is thus that the leading minor is zero and the probability that this happens $k$ times with $k$ independent selections of $U$ and $V$ is thus bounded by $\frac{1}{2^k}$. Thus Victor can accept repeated applications of the protocol only with negligible probability and the protocol is sound.

For the complexity, [6, Theorem 6.2] gives us that butterflies of respective sizes $m\lceil\log_2(m)\rceil/2$ and $n\lceil\log_2(n)\rceil/2$ are sufficient.

Victor thus needs to produce $(m\lceil\log_2(m)\rceil/2 + n\lceil\log_2(n)\rceil/2)$ random elements in $S$ and also $O(\log_2(m)+\log_2(n))$ random bits for the row and columns indices. Then the successive applications of $U$, $A$ and $V$ to a vector cost no more than $3m\lceil\log_2(m)\rceil/2 + \Omega + 3n\lceil\log_2(n)\rceil/2$ arithmetic operations. □

**Remark 1.** Over small fields,

it might not be possible to find a sufficiently large subset $S$. Then one can use extension fields or change the preconditioners. For instance, $\tilde{U} \in \mathbb{W}^{(r+1)\times m}$, and respectively $\tilde{V} \in \mathbb{W}^{n\times(r+1)}$, can be taken as sparse matrix preconditioners, as in [23] (see also [6, Corollary 7.3]), and replace respectively $[I_{r+1}|0]\,U$ and $V \begin{bmatrix} I_{r+1} \\ 0 \end{bmatrix}$. They are randomly sampled with the Wiedemann distribution, denoted by $\mathbb{W}$, and have thus not more than $2n(2+\log_2(n))^2$ non zero entries with probability at least 1/8, [23, Theorem 1]. □

## 4.3 Blackbox interactive certificate for the rank

Now we can propose a complete certificate for the rank.

If the matrix is full rank, then it is sufficient to produce a certificate for a maximal lower bound. Otherwise, it will use a non-singularity certificate on a sub-matrix of dimension $r \times r$ together with an upper bound certificate: for a matrix $A \in \mathbb{F}^{m\times n}$,

1. Peggy computes $I^{(r)} \in \mathbb{F}^{r\times m}$ (resp. $J^{(r)} \in \mathbb{F}^{n\times r}$) a concatenation of $r$ distinct row canonical vectors (resp. $r$ distinct column canonical vectors) such that $I^{(r)}AJ^{(r)}$ is a non-singular submatrix of $A$; and produces the non-singularity certificate of Figure 2 on the latter. This provides a certified lower bound for the rank of $A$.

2. Peggy provides the certificate for an upper bound $r$ of the rank of $A$ of Figure 3.

3. With certified lower bound $r$ and upper bound $r$, the rank is certified.

Using Theorems 3 and 4, we have proven:

**Corollary 3.** *Let $A \in \mathbb{F}^{m\times n}$, whose matrix-vector products costs $\Omega$ operations in $\mathbb{F}$ and let $S$ be a finite subset of $\mathbb{F}$ with $|S| > 2\min\{m,n\}(\lceil\log_2(m)\rceil + \lceil\log_2(n)\rceil)$. The above $\sum$-protocol provides an interactive certificate for the rank of $A$. This interactive certificate is sound, perfectly complete and the number of arithmetic operations performed by the verifier is bounded by*
$2\Omega + (m+n)^{1+o(1)}$.

## 4.4 Blackbox interactive certificate for the rank of integer matrices

This rank certificate can also be used for integer matrices, at roughly the same cost: just use the sparse certificate modulo a randomly chosen prime $p$.

Let $H = \min\{\sqrt{n}^m ||A||_\infty^m, \sqrt{m}^n ||A||_\infty^n\}$ be Hadamard's bound for the invariant factors of $A$. Let $h = \log_2(H)$, there cannot be more than $h$ primes reducing the rank. Therefore, the protocol is as follows:

1. Peggy produces the rank $r$ of $A$.

2. Victor selects $p$ randomly from a set with, say, $c \cdot h$ primes, for a constant $c > 2$ (if the rank is correct then the probability that the rank of $A$ will be reduced modulo $p$ is then less than $1/2$).

3. Peggy produces the sparse rank certificate of Corollary 3 over the field of integers modulo $p$, that is, two vectors $w_1 \in \mathbb{Z}/p\mathbb{Z}^n$ and $w_2 \in \mathbb{Z}/p\mathbb{Z}^{r+1}$.

4. Victor checks the certificate of the rank modulo $p$ (to apply $A$ in $\mathbb{Z}/p\mathbb{Z}$, Victor applies $A$ over $\mathbb{Z}$ and then reduces the resulting vector modulo $p$).

**Theorem 5.** *Let $A \in \mathbb{Z}^{m\times n}$, whose matrix-times-vector products costs $\Omega$ arithmetic operations in $\mathbb{Z}$.*

*Let $\mu = \max\{\log \|A\|, \log(m+n) + \log(\frac{1}{2}\log(m+n) + \log\|A\|)\}$. The above $\sum$-protocol provides an interactive certificate for the rank of $A$. This interactive certificate is sound, perfectly complete and the number of bit operations performed by the verifier is bounded by $\left(2\Omega + (m+n)^{1+o(1)}\right)\mu^{1+o(1)}$*

*Proof.* From the prime number theorem, we know that the $h$-th prime number is $O(h\log(h))$. Therefore it is possible to sample $c\cdot h$ distinct primes of magnitude bounded by $O(h\log(h))$. Then, the cost of fast arithmetic modulo any of these primes can be bounded by $\eta = O(\log(h)^{1+o(1)})$

bit operations. Then $\mu^{1+o(1)} \geq \max\{\log\|A\|, \eta\}^{1+o(1)}$ and applying $A$ to a vector in $\mathbb{Z}/p\mathbb{Z}$ costs no more than $\Omega\mu^{1+o(1)}$ binary operations. Reducing the coefficients of the resulting vector modulo $p$ then costs $(\log(m) + \log\|A\| + \eta)^{1+o(1)}$, that is not more than the matrix-times-vector product.

From Corollary 3, the cost of the certificate becomes $2\Omega\mu^{1+o(1)} + (n+m)\eta^{1+o(1)}$.

Completeness is ensured from Corollary 3 and soundness by the fact that the number of primes reducing the rank is less than $h$. Indeed, if Victor then samples from $c\cdot h$ primes with $c > 2$, he can accept $k$ repeated wrong ranks with probability bounded by $\frac{1}{2^k}$. □

Note that if the matrix is sparse with $(m+n)^{1+o(1)}$ elements, then $\Omega$ is $(m+n)^{1+o(1)}$ and the overall cost of the rank interactive certificate over the integers remains essentially linear in the input size $(m+n)\log\|A\|$.

## 4.5 Reducing breaking the random oracle to factorization

Now we look at the derandomization of the previous certificates using the strong Fiat-Shamir heuristic, see Section 2.4, where the random challenge messages of the verifier are replaced by a cryptographic hash of the property and the commitment messages.

First, it is proven in [17] that this methodology always produces digital signature schemes that are provably secure against chosen message attacks in the "Random Oracle Model" – when the hash function is modeled by a random oracle. In other words, it is equivalent for a dishonest Peggy to e.g. produce consistent systems for singular matrices or to break the random oracle.

Second, we can, e.g., use the Blumb-Blum-Shub perfect random generator [4]: it transforms a seed $x_0$ (for us the matrix) into a bit string $b_1, \ldots, b_k$ with $b_i = x_i \mod 2$; $x_{i+1} = x_i^e \mod N$ for some RSA public key $(e, N)$. It is for instance shown in [8] that knowing a number of $b_i$'s polynomial in the size of $N$, say a number $B_N = O((\log N)^\gamma)$ bits, enables one to factor it. Now, we show next that forging a consistency certificate $Ax \stackrel{?}{=} b$ is equivalent to predicting the value of at least one bit of the random right-hand side vector $b$.

**Lemma 2.** *Forging the non-singularity certificate fixes at least one bit of the random right-hand side vector.*

*Proof.* A singular matrix $A \in \mathbb{F}^{n\times n}$ has rank at most $n-1$. Write it as $A = P\left[\begin{smallmatrix} 0 & 0 \\ & L \end{smallmatrix}\right]UQ$ where $P$ and $Q$ are permutation matrices, $L \in \mathbb{F}^{(n-1)\times(n-1)}$ is lower triangular and $U$ is unit invertible upper triangular. Then if $Aw = b$, it means that $\left[\begin{smallmatrix} 0 & 0 \\ & L \end{smallmatrix}\right]z = P^{-1}b$ for $z = UQw$. Therefore, the first entry of $P^{-1}b$ must be zero. □

Therefore, we fix the RSA modulus $N$ and require as a certificate that the consistency check is repeated $B_N$ times. When the protocol is repeated with Fiat-Shamir derandomization, we use as successive random vectors $b$, the hash of

the input and the previous iteration. If Peggy can find a matrix $A$ of dimension $n$ (polynomial in the size of $N$) for which she can forge the $B_N$ repeated applications of the certificate, then

she can predict $B_N$ bits of the Blumb-Blum-Shub hashes in $O(B_N(\Omega + n^{1+o(1)}))$ operations. Thus Peggy can factor $N$ in polynomial time.

## Acknowledgments

We thank Brice Boyer , Shafi Goldwasser , Clément Pernet , Jean-Louis Roch , Guy Rothblum , Justin Thaler and the referees for their helpful comments.

## References

[1] Babai, L. Trading group theory for randomness. In Sedgewick [20], pp. 421–429. http://dx.doi.org/10.1145/22145.22192.

[2] Bellare, M., and Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (Fairfax, Virginia, Nov. 1993), V. Ashby, Ed., ACM Press, pp. 62–73. http://www-cse.ucsd.edu/users/mihir/papers/ro.pdf.

[3] Bernhard, D., Pereira, O., and Warinschi, B. How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios. In *Advances in Cryptology - ASIACRYPT'12* (2012), X. Wang and K. Sako, Eds., vol. 7658 of *Lecture Notes in Computer Science*, Springer, pp. 626–643. http://www.uclouvain.be/crypto/services/download/publications.pdf.87e67d05ee05000b.6d616

[4] Blum, L., Blum, M., and Shub, M. Comparison of two pseudo-random number generators. In *Advances in Cryptology—CRYPTO'82* (23–25 Aug. 1982), D. Chaum, R. L. Rivest, and A. T. Sherman, Eds., Plenum Press, New York and London, 1983, pp. 61–78. http://dx.doi.org/10.1007/978-1-4757-0602-4_6.

[5] Blum, M., and Kannan, S. Designing programs that check their work. *Journal of the ACM 42*, 1 (Jan. 1995), 269–291. http://www.icsi.berkeley.edu/pubs/techreports/tr-88-009.pdf.

[6] Chen, L., Eberly, W., Kaltofen, E. L., Saunders, B. D., Turner, W. J., and Villard, G. Efficient matrix preconditioners for black box linear algebra. *Linear Algebra and its Applications 343-344* (2002), 119–146. http://www.math.ncsu.edu/~kaltofen/bibliography/02/CEKSTV02.pdf.

[7] Fiat, A., and Shamir, A. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology—CRYPTO'86* (11–15 Aug. 1986), A. M. Odlyzko, Ed., vol. 263 of *Lecture Notes in Computer Science*, Springer-Verlag, 1987, pp. 186–194. http://www.cs.rit.edu/~jjk8346/FiatShamir.pdf.

[8] Fischlin, R., and Schnorr, C. P. Stronger security proofs for RSA and Rabin bits. In *Advances in Cryptology - EUROCRYPT'97* (1997), vol. 1233 of *Lecture Notes in Computer Science*, pp. 267–279. http://dx.doi.org/10.1007/3-540-69053-0_19.

[9] Freivalds, R. Fast probabilistic algorithms. In *Mathematical Foundations of Computer Science 1979* (Olomouc, Czechoslovakia, Sept. 1979), J. Bečvář, Ed., vol. 74 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 57–69. http://dx.doi.org/10.1007/3-540-09526-8_5.

[10] Furer, M., Goldreich, O., Mansour, Y., Sipser, M., and Zachos, S. On completeness and soundness in interactive proof systems. In *Randomness and Computation*, S. Micali, Ed., vol. 5. Advances in Computing Research, JAI Press, Greenwich, Connecticut, 1989, pp. 429–442. http://www.wisdom.weizmann.ac.il/~oded/PS/fgmsz.ps.

[11] Giesbrecht, M. W., Lobo, A., and Saunders, B. D. Certifying inconsistency of sparse linear systems. In *Proceedings of the 1998 ACM International Symposium on Symbolic and Algebraic Computation, Rostock, Germany* (July 1998), V. W. Barry Trager, Ed., ACM Press, New York, pp. 113–119.

[12] Goldwasser, S., Kalai, Y. T., and Rothblum, G. N. Delegating computation: interactive proofs for muggles. In *STOC'2008, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada* (May 2008), C. Dwork, Ed., ACM Press, pp. 113–122.

[13] Goldwasser, S., Micali, S., and Rackoff, C. The knowledge complexity of interactive proof-systems. In Sedgewick [20], pp. 291–304. http://dx.doi.org/10.1145/22145.22178.

[14] Kaltofen, E. L., Li, B., Yang, Z., and Zhi, L. Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients.

*Journal of Symbolic Computation 47*, 1 (Jan. 2012), 1–15. http://www.math.ncsu.edu/~kaltofen/bibliography/09/KLYZ09.pdf.

[15] Kaltofen, E. L., Nehring, M., and Saunders, B. D. Quadratic-time certificates in linear algebra. In *ISSAC'2011, Proceedings of the 2011 ACM International Symposium on Symbolic and Algebraic Computation, San Jose, California, USA* (June 2011), A. Leykin, Ed., ACM Press, New York, pp. 171–176. http://www.math.ncsu.edu/~kaltofen/bibliography/11/KNS11.pdf.

[16] Kimbrel, T., and Sinha, R. K. A probabilistic algorithm for verifying matrix products using $O(n^2)$ time and $\log_2 n + O(1)$ random bits. *Information Processing Letters 45*, 2 (Feb. 1993), 107–110. ftp://trout.cs.washington.edu/tr/1991/08/UW-CSE-91-08-06.pdf.

[17] Pointcheval, D., and Stern, J. Security proofs for signature schemes. In *Advances in Cryptology—EUROCRYPT'96* (12–16 May 1996), U. Maurer, Ed., vol. 1070 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 387–398. http://www.di.ens.fr/~pointche/Documents/Papers/1996_eurocrypt.pdf.

[18] Pratt, V. R. Every prime has a succinct certificate. *SIAM Journal on Computing 4*, 3 (Sept. 1975), 214–220. http://boole.stanford.edu/pub/SucCert.pdf.

[19] Saunders, B. D., Storjohann, A., and Villard, G. Matrix rank certification. *Electronic Journal of Linear Algebra 11* (2004), 16–23. http://perso.ens-lyon.fr/gilles.villard/BIBLIOGRAPHIE/PDF/rank-certif.pdf.

[20] Sedgewick, R., Ed. *STOC '85, ACM Symposium on Theory of Computing, Providence, Rhode Island, USA* (May 1985), ACM Press, New York.

[21] Storjohann, A. Integer matrix rank certification. In *ISSAC'2009, Proceedings of the 2009 ACM International Symposium on Symbolic and Algebraic Computation, Seoul, Korea* (July 2009), J. P. May, Ed., pp. 333–340. https://cs.uwaterloo.ca/~astorjoh/issac09.pdf.

[22] Thaler, J. Time-optimal interactive proofs for circuit evaluation. In *Advances in Cryptology - CRYPTO'13* (2013), R. Canetti and J. Garay, Eds., vol. 8043 of *Lecture Notes in Computer Science*, Springer Berlin Heidelberg, pp. 71–89.

[23] Wiedemann, D. H. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory 32*, 1 (Jan. 1986), 54–62. http://dx.doi.org/10.1109/TIT.1986.1057137.