# Proof-of-work certificates that can be efficiently computed in the cloud

Jean-Guillaume Dumas

# Proof-of-work certificates that can be efficiently computed in the cloud

Jean-Guillaume Dumas*

June 28, 2018

**Abstract**

In an emerging computing paradigm, computational capabilities, from processing power to storage capacities, are offered to users over communication networks as a cloud-based service. There, demanding computations are outsourced in order to limit infrastructure costs.

The idea of verifiable computing is to associate a data structure, a *proof-of-work certificate*, to the result of the outsourced computation. This allows a verification algorithm to prove the validity of the result, faster than by recomputing it. We talk about a Prover (the server performing the computations) and a Verifier.

Goldwasser, Kalai and Rothblum gave in 2008 a generic method to verify any parallelizable computation, in almost linear time in the size of the, potentially structured, inputs and the result. However, the extra cost of the computations for the Prover (and therefore the extra cost to the customer), although only almost a constant factor of the overall work, is nonetheless prohibitive in practice.

Differently, we will here present problem-specific procedures in computer algebra, e.g. for exact linear algebra computations, that are Prover-optimal, that is that have much less financial overhead.

## 1 Introduction

In an emerging computing paradigm, computational capabilities, from processing power to storage capacities, are offered to users over communication networks as a service.

Many such outsourcing platforms are now well established, as Amazon web services (through the Elastic Compute Cloud), Microsoft Azure, IBM Platform Computing or Google cloud platform (via Google Compute Engine), as shown in Figure 1. None of these platforms, however, offer any guarantee whatsoever on the calculation: no guarantee that the result is correct, nor even that the computation has even effectively been done.

*Université Grenoble Alpes. Laboratoire Jean Kuntzmann, CNRS, UMR 5224. 700 avenue centrale, IMAG - CS 40700, 38058 Grenoble, cedex 9 France. Jean-Guillaume.Dumas@univ-grenoble-alpes.fr


Figure 1: Some outsourced computing services

## 1.1 Verifiable computing

This new paradigm holds enormous promise for increasing the utility of computationally weak devices. A natural approach is for weak devices to delegate expensive tasks, such as storing a large file or running a complex computation, to more powerful entities (say servers) connected to the same network. While the delegation approach seems promising, it raises an immediate concern: when and how can a weak device verify that a computational task was completed correctly? This practically motivated question touches on foundational questions in cryptography, coding theory, complexity theory, proofs and algorithms.

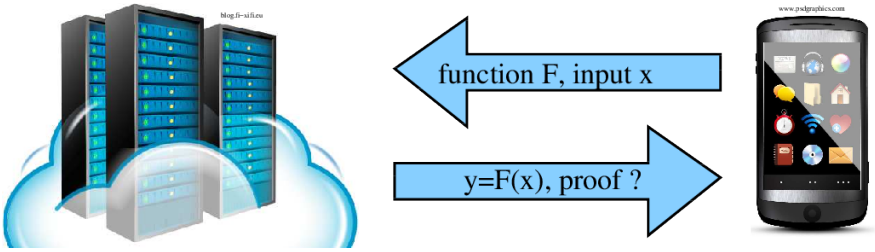



Figure 2: Verifying the computation should take less time than computing it

More generally, the question of verifying a result at a lower cost (time, memory) than that of recomputing it, as shown on Figure 2, is of paramount importance. Another example of application is for the extension of the trust about results computed via probabilistic or approximate algorithms. There the idea is to gain confidence into the correctness, but only at a cost negligible when compared to that of the computation.

## 1.2 Linear algebra, global optimization

For instance, GL7d19 is an $1\,911\,130 \times 1\,955\,309$ matrix whose rank $1\,033\,568$ was computed once in 2007 with a Monte-Carlo randomized algorithm [19].

This required 1050 CPU days of computation. We thus need publicly verifiable certificates to improve our confidence in computational results.

In linear algebra our original motivation is also 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 computer algebra step in an exact rational sum-of-squares proof, namely

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

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 [43]. Then, the client can verify the positiveness by checking Descartes' rule of sign on the *certified* characteristic polynomial of $W^{[1]}$ and $W^{[2]}$. Thus arose the question how to give possibly probabilistically checkable certificates for linear algebra problems.

## 1.3 Techniques

The tools used to provide such efficient *proof-of-work certificates* stem from programs that check their work [12], to proof of knowledge protocols [7], via error-correcting codes [42, 35] complexity theory [1] or secure multiparty protocols [17], and the interaction of these different methodologies is crucial.

Here we will thus follow this road-map:

- We recalled that global optimization can be reduced to linear algebra. Thereupon we will focus on certificates for linear algebra problems [43] in computer algebra, which extend the randomized algorithms of Freivalds [32].
- We combine those with probabilistic interactive proofs of Babai [5] and Goldwasser et al. [39],
- as well as Fiat-Shamir heuristic [29, 9] turning interactive certificates into non-interactive heuristics subject to computational hardness.
- Overall, we obtain problem-specific efficient certificates for dense, sparse, structured matrices with coefficients in fields or integral domains.

# 2 Interactive protocols, the PCP theorem and homomorphic encryption

## 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, in successive *rounds*, 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 [33]. Thus in the following, as we want to prove correctness of a result more than proving knowledge of it, we will mainly show interactive proofs with perfect completeness.

The class of problems solvable by an interactive proof system (IP) is equal to the class PSPACE [55] and a probabilistically checkable proof, $\text{PCP}[r(n), \pi(n)]$, for an input of length $n$, is a type of proof that can be checked by a randomized algorithm using a bounded amount of randomness $r(n)$ and reading a bounded number of bits of the proof $\pi(n)$. For instance, $\text{PCP}[O(\log n), O(1)]$=NP [6, 3].

In general, interactive protocols encompass many kinds of proofs and Prover and Verifier settings. One can think of the difficulty of integer factorization versus that of re-multiplying found factors. Another example could be satisfiability checking, where the solver has to explore the state space, while verifying a variable assignment or a conflict clause could be much simpler [2]. In computer algebra, the Prover can be a probabilistic algorithm or a symbolic-numeric program, where the Verifier would perform the checks exactly or symbolically; further, computer algebra systems could perform a complex calculations where an interactive theorem prover (or proof assistant like Isabel-HOL or Coq) only has to a posteriori formally verify the certificate [16, 15].

Table 1 gives more examples of such settings.

| Prover | Verifier |
|---|---|
| Computer Scientist | Mathematician |
| Computer Algebra system | Formal proof assistant |
| Cloud | User |
| Server | Client |
| Cellphone | Trusted platform module |

Table 1: Examples of Prover/Verifier settings

## 2.2 Goldwasser et al. prover efficient interactive certificates

Now, efficient protocols (interactive proofs between a *Prover*, responsible for the computation, and a *Verifier*, to be convinced) can be designed for delegating computational tasks.

Recently, generic protocols, mixing zero-sum checks [45] and probabilistically checkable proofs, have been designed by teams around Shafi Goldwasser at the MIT or Harvard, for circuits with polylogarithmic depth [38, 57], namely for problems that can be efficiently solved on a parallel computer (in the NC or AC complexity class). These results have also been extended to any structured inputs (any polynomial-time-uniform polylog-depth Boolean circuits in the sense of Beame's et al, [8], division circuits) [23].

The resulting protocols are interactive and there is a trade-off between the number of interactive rounds, the volume of communication and the computational cost [50]; the cost for the verifier being usually only roughly proportional to the input size.

These protocols can, e.g., certify that two supersparse polynomials are relatively prime in verifier cost which is polylog time (and rounds) in the degree.

The produced certificates, in analogy to processor-efficient parallel algorithms, are Prover-efficient: if the cost to compute the result by the best known algorithm is $T(n)$ for a size $n$ problem, then the cost to produce the result together with the verifiable certificate is $T(n)^{1+o(1)}$.

Those techniques can however produce a non negligible practical overhead for the Prover and are restricted to certain classes of circuits.

## 2.3 Parno et al. homomorphic solutions

Another approach as been developed by Gentry et al., at Microsoft and IBM research, it is Pinocchio. It solves a broader range of problems, to the cost of using relatively inefficient homomorphic routines [48] in an amortized way.

The idea is that the Prover should run the program (or at least part of the program twice), once normally on the input, and once homomorphically on an encrypted version of the input. The Verifier will then verify the consistency between the normal output and the encrypted one. Usually the Verifier is required to run the algorithm at least once for a given size or structure of the input but

can reuse this for multiple inputs.

This generic procedure can be applied on specific linear algebra or polynomial problems [31, 60, 28, 25], or on generic quadratic arithmetic programs [48]. There, fully homomorphic encryption can be used [36] or sometimes just pairings [48] and/or cryptographic hashes [30].

Here also the Prover can be efficient, but subject in practice to the overhead of homomorphic computations.

## 2.4 Public verification, delegatability and zero-knowledge

Interactive certificates require some exchanges between the Prover and the Verifier. With such a protocol, the Verifier can be *privately* convinced that the computation of the Prover produced the correct answer. This does not mean that other people would be convinced be the transcript of their exchange: the Prover and Verifier could be in cahoots and the supposedly random challenges carefully crafted.

Fiat-Shamir heuristic [29, 9] can thus turn interactive certificates into non-interactive heuristics subject to computational hardness: the random challenges are replaced by cryptographic hashes of all previous data and exchanges. Crafting such values would then reduce to being able to forge cryptographic fingerprints [20, § 4.5].

Further, more properties could be sought for such protocols, such *privacy* of data and/or computations. In this setting, a publicly verifiable computation scheme can also be four algorithms (*KeyGen*, *ProbGen*, *Compute*, *Verify*), where *KeyGen* is some (amortized) preparation of the data, *ProbGen* is the preparation of the input, *Compute* is the work of the *Prover* and *Verify* is the work of the *Verifier* [49]. Usually the Verifier also executes *KeyGen* and *ProbGen* but in a more general setting these can be performed by different entities (respectively called a *Preparator* and a *Trustee*).

This allows to define several adversary models but usually the protocols are secure against a *malicious Prover only* (that is the Client must trust both the Preparator and the Trustee).

One can also further impose that there is no interaction between the Client and the Trustee after the Client has sent his input to the Server. Publicly verifiable protocols with this property are said to be *publicly delegatable* [60, 28, 25].

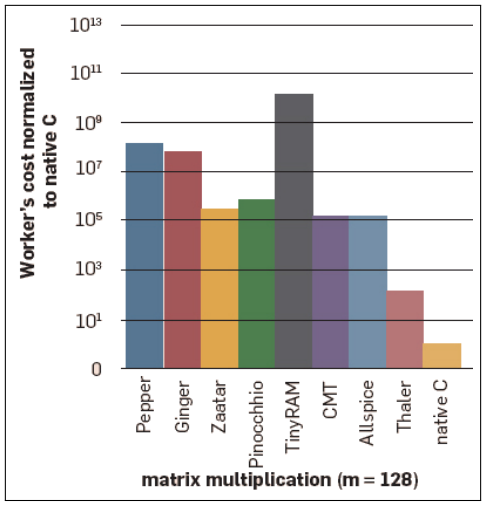
Then, some different properties of the protocol could be desirable, such as not disclosing the result but instead just providing a *proof-of-work*. This results in general in *zero-knowledge* protocols over confidential data, such as cryptocurrency transactions, as in, e.g., [39], with recent efficient implementations [13, 10, 11, 14].

## 2.5 Problem-specific efficient certificates

Differently, dedicated certificates (data structures and algorithms that are verifiable a posteriori, without interaction) have also been developed, e.g., in com-

puter algebra for exact linear algebra [32, 43, 20, 22, 24], even for problems that are not structured. There the certificate constitute a proof of correctness of a result, not of a computation, and can thus also **detect bugs** in the implementations.

Moreover, problem-specific certificates can gain crucial logarithmic factors for the verifier and allow for optimal prover computational time, see Figure 3.




| $2048 \times 2048$ | Thaler[57] | Ad-hoc [32] |
|---|---|---|
| Server time | 18.23s | 0.65s |
| Certificate overhead | 0.13s | 0.00s |
| Client time | 2.89s | 0.01s |

Figure 3: Generic protocols [58] versus dedicated protocols for matrix multiplication

For this, the main difficulty is to be able to design verification algorithms for a problem that are completely orthogonal to the computational algorithms solving it, while remaining checkable in time and space not much larger than the input.

# 3 Prover-optimal certificates in linear algebra

We show in this section, that such problem-specific certificates are attainable in linear algebra, where we allow certificates that are validated by Monte Carlo randomized algorithms.

### 3.1 Freivalds zero equivalence of matrix expressions

The seminal certificate in linear algebra is due to Rūsiņš Freivalds [32]: quadratic time is feasible because a matrix multiplication $AB$ can be certified by the resulting product matrix $C$ via the probabilistic projection to matrix-vector products (see also [44] who reduced the requirements to only $O(\log(n))$ random bits), shown in Protocol 1.

| *Prover* | *Communication* | *Verifier* |
|---|---|---|
| | $\mathbf{A} \in \mathbb{F}^{m\times k}$, $\mathbf{B} \in \mathbb{F}^{k\times n}$ | |
| Compute $\mathbf{C} = \mathbf{A}\cdot\mathbf{B}$ | $\xrightarrow{\mathbf{C}}$ | $r \xleftarrow{\$} S \subseteq \mathbb{F}$<br>Form $\vec{v} = [1, r, r^2, \ldots, r^{n-1}]^T$<br>$\mathbf{A}\,(\mathbf{B}\vec{v}) - \mathbf{C}\vec{v} \stackrel{?}{=} \vec{0}$ |

**Protocol 1.** Matrix multiplication certificate [44].

In Protocol 1, we give the variant of [44] that requires $\log(n)$ random bits, but works over sufficiently large coefficient domains, as its soundness is $1 - \frac{|S|}{n}$ by the DeMillo-Lipton/Schwartz/Zippel lemma [18, 61, 53]. Freivalds original version randomly selects a zero-one vector instead. This requires $n$ random bits instead but applies to any ring and gives a soundness larger than $\frac{1}{2}$.

In both cases it is sufficient to repeat the test several times to achieve any desired probability.

### 3.2 Reductions to matrix multiplication

With a certificate for matrix expressions, then one can certify **any algorithm that reduces to matrix multiplication**: the Prover records all the intermediate matrix products and sends them to the Verifier who reruns the same algorithm but checks the matrix products instead of computing them [43], as shown in Protocol 2.

| *Prover* | *Communication* | *Verifier* |
|---|---|---|
| Runs the algorithm | All intermediate matrix products $\longrightarrow$ | Runs the algorithm but replace each matrix products by Freivalds' checks |

**Protocol 2.** Certificates with reduction to matrix multiplication [43, § 5].

Overall, the communications and Verifier computational cost are given by taking $\omega = 2$ in the Prover's complexity bounds (with potential additional logarithmic factors due to summations). Further, *t*he production of the certificate

has no computational overhead for the Prover: it only adds the communication of the intermediate matrix products.

For instance, Storjohann's Las Vegas rank algorithm of integer matrices [56] becomes a non-interactive/non-cryptographic Monte Carlo checkable proof-of-work certificate that has soft-linear time communication and verifier bit complexity in the number of input bits!

### 3.3 Sparse or structured matrices

When the matrices are sparse or present some structure, quadratic run time and/or quadratic communications might be overkill for the Verifier. There it is better if his communications and computational cost is of the form $\mu(A)+n^{1+o(1)}$ where $\mu(A)$ is the number of operations to perform a matrix-vector products. This scheme is thus also interesting if the considered matrix is only given as a blackbox [40].

In that vein, we now have certificates for :

- **non-singularity**, Protocol 3;
- **an upper bound to the rank**, Protocol 4 (if elimination on the input matrix is possible for the Prover then a variant without preconditioners can be used [26, 24]);
- the **rank**, combining Protocols 3 and 4;
- the **minimal polynomial**, using Protocol 5 (where $f_u^{A,v}$ is the monic scalar minimal generating polynomial of the sequence $u^Tv, \ldots, u^TA^iv$, $\rho_u^{A,v}$ is such that $\rho_u^{A,v} = f_u^{A,v} \cdot G$ with $G$ the generating function of the latter sequence, for random vectors $u$ and $v$, chosen by the Verifier [41, Theorem 5]);
- the **determinant**, Protocol 6, which randomness could be reduced from $O(n)$ to a constant number of field elements [21, § 7].

Additionally, properties of the given matrices can also sometimes be discovered at low cost: whether the blackbox is a **band matrix**, has a **low displacement rank**, has a few or many **nilpotent blocks** or **invariant factors** [27]. Similarly, the existence of a **triangular one sided equivalence**, as well as the **rank profiles** can also be certified without sending an explicit factorization to the Verifier [24]. For the latter, the price to pay is to require a linear number of rounds.

### 3.4 Integer or polynomial matrices

Over an integral domain, the verification procedure can be performed via a randomly chosen modular projection. If there are sufficiently many *small* maximal ideals, then one can uniformly chose one at random and then ask for a certification of the result in the associated quotient field as shown in Protocol 7.

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

**Protocol 3.** Blackbox interactive certificate of non-singularity [20]

| *Prover* | | *Verifier* |
|---|---|---|
| | $A \in \mathbb{F}^{m\times n}$ | $S \subset \mathbb{F}$ |
| $rank(A) \leq r$ | $1:\ r \longrightarrow$ | $r \stackrel{?}{<} \min\{m,n\}$ |
| | $\longleftarrow 2:\ U,V$ | $U \in \mathbb{B}_S^{m\times m}, V \in \mathbb{B}_S^{n\times n}$ |
| | | preconditioners of size $n^{1+o(1)}$ |
| $w \in \mathbb{F}^{r+1} \neq 0$ | $3:\ w \longrightarrow$ | $w \stackrel{?}{\neq} 0$ |
| | | $[I_{r+1}\|0]\, UAV \begin{bmatrix} I_{r+1} \\ 0 \end{bmatrix} w \stackrel{?}{=} 0$ |

**Protocol 4.** Blackbox upper bound to the rank certificate [20]

| *Prover* | *Communication* | *Verifier* |
|---|---|---|
| $H(\lambda) = f_u^{A,v}(\lambda)$, | | |
| $h(\lambda) = \rho_u^{A,v}(\lambda)$. | $\xrightarrow{H,h}$ | |
| $\phi, \psi \in \mathbb{F}[\lambda]$ with | | |
| $\phi f_u^{A,v} + \psi \rho_u^{A,v} = 1$, | $\xrightarrow{\phi,\psi}$ | $\deg(\phi) \stackrel{?}{\leq} \deg(h) - 1$, |
| | | $\deg(\psi) \stackrel{?}{\leq} \deg(H) - 1$. |
| | | Random $r_0 \in S \subseteq \mathbb{F}$. |
| | | Checks $\mathrm{GCD}(H(\lambda), h(\lambda)) = 1$ by $\phi(r_0)H(r_0) + \psi(r_0)h(r_0) \stackrel{?}{=} 1$. |
| Computes $w$ such that | $\xleftarrow{r_1}$ | Random $r_1 \in S \subseteq \mathbb{F}$. |
| $(r_1 I_n - A)w = v$. | $\xrightarrow{w}$ | Checks $(r_1 I_n - A)w \stackrel{?}{=} v$ and $(u^T w)H(r_1) \stackrel{?}{=} h(r_1)$. |
| | | Returns $f_u^{A,v}(\lambda) = H(\lambda)$. |

**Protocol 5.** Certificate for $f_u^{A,v}$ [22]

| *Prover* | *Communication* | *Verifier* |
|---|---|---|
| 1. Form $B = DA$ with $D \in S^n \subseteq \mathbb{F}^{*n}$ and $u, v \in S^n$, s.t. $\deg(f_u^{B,v}) = n$. | $\xrightarrow{D,u,v}$ | |
| | Protocol 5 | |
| 2. | $\xrightarrow{H,h,\phi,\psi}$ | Checks: |
| 3. | $\xleftarrow{r_1}$ | $\deg(H) \stackrel{?}{=} n$, |
| 4. | $\xrightarrow{w}$ | $H \stackrel{?}{=} f_u^{B,v}$, w.h.p. |
| 5. | | Returns $\dfrac{f_u^{B,v}(0)}{\det(D)}$. |

**Protocol 6.** Determinant certificate for a non-singular blackbox [22]

| | *Prover* | *Communication* | *Verifier* |
|---|---|---|---|
| *Commitment* | Result $r \in \mathbb{R}$ | $\xrightarrow{r}$ | |
| *Challenge* | | $\xleftarrow{\mathcal{I}}$ | $\mathcal{I} \xleftarrow{\$}$ maximal ideals |
| *Response* | Result $x \in \mathbb{R}/\mathcal{I}$ with field certificate $\mathcal{C}_{\mathbb{R}/I}$ | $\xrightarrow{x, \mathcal{C}_{\mathbb{R}/I}}$ | $x \stackrel{?}{\equiv} r \mod \mathcal{I}$ and $\mathcal{C}_{\mathbb{R}/I}(x) \stackrel{?}{=}$ valid |

**Protocol 7.** Certification in a quotient field [20, § 3.2 and § 4.4].

For instance this gives very efficient certificates for polynomial or integer/rational matrices, provided that one has a bound on the degree or the magnitude of the coefficients:

- For **integral matrices**, if the true result $v$ is bounded in magnitude, then only a finite number of prime numbers will divide the difference between the commitment $r$ and the result. Therefore the result can be checked over a *small* prime field [20, Theorem 5].
- For **polynomial matrices**, if the true $v(X)$ result's degree is bounded, then only a finite number of evaluation points can be roots of the difference polynomial between the committed one $r(X)$ and the result. Therefore the result can be checked in the ground field at a *small* evaluation point [20, Theorem 2].

The latter results allows, for instance, to certify the global optimization problems of Section 1.2.

This is illustrated in Figure 4, where many of the reductions presented here are recalled.

## 3.5 Non-interactive certificates

The certificates in Sections 3.1 and 3.2 are non-interactive: all the communications can be recorded and publicly verified later.

On the contrary the certificates of Sections 3.3, 3.4 are interactive: the Verifier chooses some random bits during the computation of the certificate. Non-interactivity can be recovered via Fiat-Shamir scheme: any random bits are generated by cryptographic hashes of the inputs and all the previous intermediate commitments. Soundness is then subject to standard cryptographic assumptions.

For sparse or structured problems fewer results exists without this assumption, or with worse complexity bounds:

- For the minimal polynomial (scalar or matrix) or the determinant, non-interactive certificates exists, but with communications and computational cost $O(n\sqrt{\mu(A)})$ instead of $\mu(A) + n^{1+o(1)}$ [21].

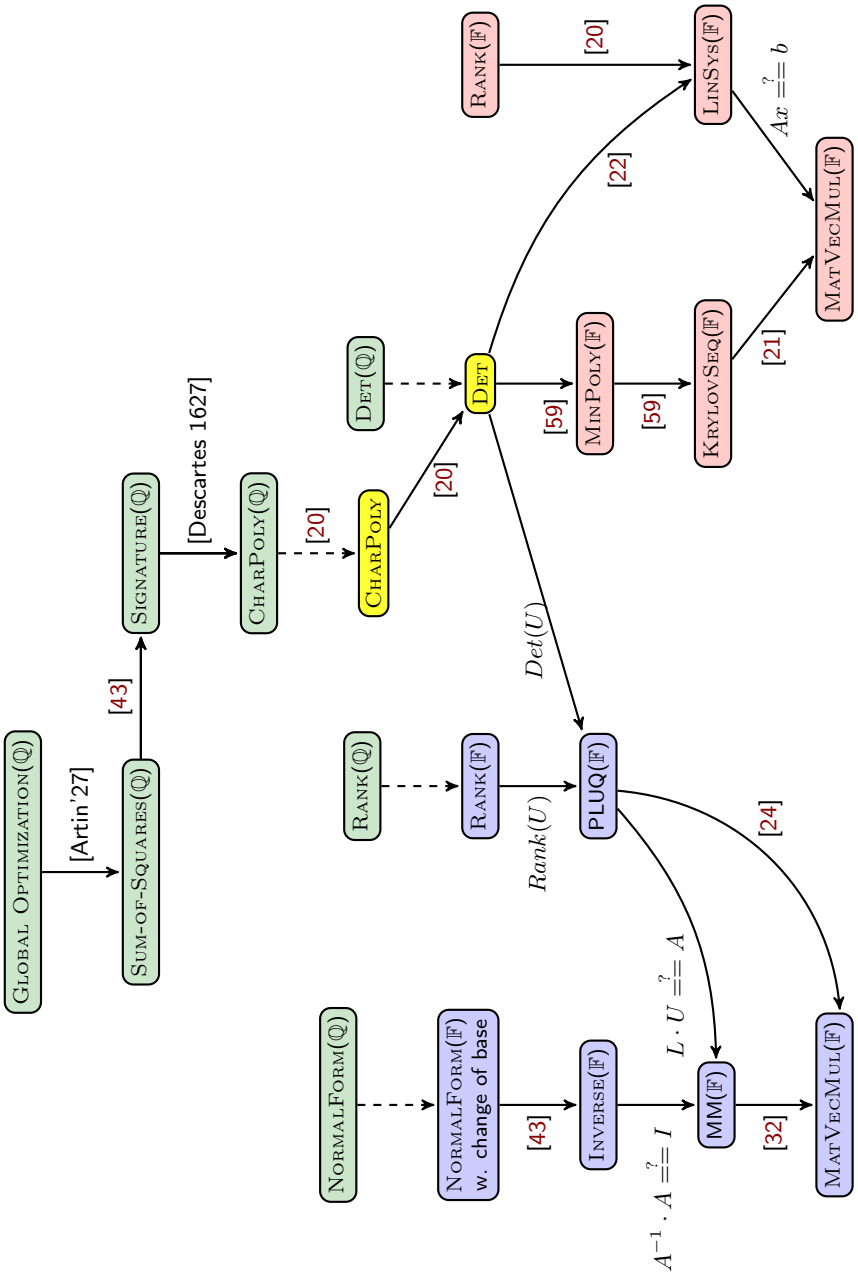



Figure 4: Global optimization via problem-specific interactive certificates: dense (purple) or sparse (red) algebraic problems, as well as over the reals (green) or oblivious (yellow).

- Non-interactive certificates can also verify polynomial minimal approximant bases in $O(mD + m^\omega)$, where $D$ is the sum of the column degrees of the output [37].

# 4 Some open problems

We conclude this survey with some open problems in the area of problem specific linear algebra certificates:

- **Sparse Smith form**: for dense matrices, one can interactively certify any normal form via a Freivalds certificate on a randomly chosen modular factorization. With sparse matrices, even the modular projection of the change of base can be too large. In that setting extending protocols for the rank or the determinant to deal with the Smith form should be possible.
- **Non integral domains certificates**: more generally, how to efficiently certify some properties when there is no quotients or if those properties do not carry over (e.g., Smith form)?
- We have defined certificates resisting a malicious server with unbounded power. This is error detection with unbounded number of errors. Thus the question of the complexity of **problem specific unbounded error correction** also arises. This path again was first taken for matrix multiplication [35] and was recently extended to the matrix inverse [51].

# Acknowledgment

I thank Brice Boyer , Pascal Lafourcade , Shafi Goldwasser , Erich Kaltofen , Julio López Fenner , David Lucas , Vincent Neiger , Jean-Baptiste Orfila , Clément Pernet , Maxime Puys , Jean-Louis Roch , Dan Roche , Guy Rothblum , Justin Thaler and an anonymous referee for their helpful comments.

# References

[1] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):2:1–2:54, February 2009. doi:10.1145/1490270.1490272.

[2] Erika Ábrahám, John Abbott, Bernd Becker, Anna Maria Bigatti, Martin Brain, Bruno Buchberger, Alessandro Cimatti, James H. Davenport, Matthew England, Pascal Fontaine, Stephen Forrest, Alberto Griggio, Daniel Kroening, Werner M. Seiler, and Thomas Sturm. SC$^2$: Satisfiability checking meets symbolic computation. In Michael Kohlhase, Moa Johansson, Bruce R. Miller, Leonardo de de Moura, and Frank Wm. Tompa, editors, *Intelligent Computer Mathematics (CICM)*, volume 9791

of *Lecture Notes in Computer Science*, pages 28–43. Springer, 2016. URL: https://members.loria.fr/PFontaine/Abraham1.pdf.

[3] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs; A new characterization of NP. In *33rd Annual Symposium on Foundations of Computer Science*, pages 2–13, Pittsburgh, Pennsylvania, 24–27 October 1992. IEEE.

[4] Carlos Arreche, editor. *ISSAC'2018, Proceedings of the 2018 ACM International Symposium on Symbolic and Algebraic Computation, New York, USA*. ACM Press, New York, July 2018.

[5] László Babai. Trading group theory for randomness. In Sedgewick [54], pages 421–429. URL: http://dx.doi.org/10.1145/22145.22192.

[6] Laszlo Babai, Lance Fortnow, and Carsten Lund. Nondeterministic exponential time has two-prover interactive protocols. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, volume 1, pages 16–25, October 1990. doi:10.1109/FSCS.1990.89520.

[7] Endre Bangerter, Jan Camenisch, and Ueli M. Maurer. Efficient proofs of knowledge of discrete logarithms and representations in groups with hidden order. In *Proceedings of the 8th international conference on Theory and Practice in Public Key Cryptography*, PKC'05, pages 154–171, Berlin, Heidelberg, 2005. Springer-Verlag. doi:10.1007/978-3-540-30580-4_11.

[8] Paul W. Beame, Stephen A. Cook, and H. James Hoover. Log depth circuits for division and related problems. *SIAM J. Comput.*, 15:994–1003, 1986. doi:10.1137/0215070.

[9] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In Victoria Ashby, editor, *Proceedings of the 1st ACM Conference on Computer and Communications Security*, pages 62–73, Fairfax, Virginia, November 1993. ACM Press. URL: http://www-cse.ucsd.edu/users/mihir/papers/ro.pdf.

[10] Eli Ben-Sasson, Iddo Bentov, Alessandro Chiesa, Ariel Gabizon, Daniel Genkin, Matan Hamilis, Evgenya Pergament, Michael Riabzev, Mark Silberstein, Eran Tromer, and Madars Virza. Computational integrity with a public random string from quasi-linear pcps. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*, volume 10212 of *Lecture Notes in Computer Science*, pages 551–579, 2017. URL: https://doi.org/10.1007/978-3-319-56617-7, doi:10.1007/978-3-319-56617-7_19.

[11] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. Cryptology

ePrint Archive, Report 2018/046, 2018. https://eprint.iacr.org/2018/046.

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

[13] Jonathan Bootle, Andrea Cerulli, Pyrros Chaidos, Jens Groth, and Christophe Petit. Efficient zero-knowledge arguments for arithmetic circuits in the discrete log setting. In Marc Fischlin and Jean-Sébastien Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part II*, volume 9666 of *Lecture Notes in Computer Science*, pages 327–357. Springer, 2016. URL: https://doi.org/10.1007/978-3-662-49896-5, doi:10.1007/978-3-662-49896-5_12.

[14] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 319–338, 2018. doi:10.1109/SP.2018.00020.

[15] Cristian S. Calude and Declan Thompson. Incompleteness, undecidability and automated proofs. In Vladimir P. Gerdt, Wolfram Koepf, Werner M. Seiler, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, pages 134–155, Cham, 2016. Springer International Publishing.

[16] Frédéric Chyzak, Assia Mahboubi, Thomas Sibut-Pinote, and Enrico Tassi. A Computer-Algebra-Based Formal Proof of the Irrationality of $\zeta(3)$. In *ITP - 5th International Conference on Interactive Theorem Proving*, Vienna, Austria, 2014. URL: https://hal.inria.fr/hal-00984057.

[17] Ronald Cramer, Ivan Damgård, and Jesper B. Nielsen. Multiparty computation from threshold homomorphic encryption. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings*, pages 280–300, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg. doi:10.1007/3-540-44987-6_18.

[18] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Letters*, 7(4):193–195, June 1978. doi:10.1016/0020-0190(78)90067-4.

[19] Jean-Guillaume Dumas, Pascal Giorgi, Philippe Elbaz-Vincent, and Anna Urbańska. Parallel computation of the rank of large sparse matrices from algebraic k-theory. In Marc Moreno-Maza and Stephen Watt, editors, *PASCO'07, Proceedings of the 3rd ACM International Workshop on*

*Parallel Symbolic Computation*, pages 43–52. Waterloo University, Ontario, Canada, July 2007. URL: http://hal.archives-ouvertes.fr/hal-00142141.

[20] Jean-Guillaume Dumas and Erich Kaltofen. Essentially optimal interactive certificates in linear algebra. In Nabeshima [46], pages 146–153. URL: http://hal.archives-ouvertes.fr/hal-00932846, doi:10.1145/2608628.2608644.

[21] Jean-Guillaume Dumas, Erich Kaltofen, and Emmanuel Thomé. Interactive certificate for the verification of Wiedemann's Krylov sequence: application to the certification of the determinant, the minimal and the characteristic polynomials of sparse matrices. Technical report, IMAG-hal-01171249 arXiv cs.SC/1507.01083, January 2016. URL: http://hal.archives-ouvertes.fr/hal-01171249.

[22] Jean-Guillaume Dumas, Erich Kaltofen, Emmanuel Thomé, and Gilles Villard. Linear time interactive certificates for the minimal polynomial and the determinant of a sparse matrix. In Gao [34], pages 199–206. URL: http://hal.archives-ouvertes.fr/hal-01266041, doi:10.1145/2930889.2930908.

[23] Jean-Guillaume Dumas, Erich Kaltofen, Gilles Villard, and Lihong Zhi. Polynomial time interactive proofs for linear algebra with exponential matrix dimensions and scalars given by polynomial time circuits. In Safey El Din [52], pages 125–132. URL: http://ljk.imag.fr/membres/Jean-Guillaume.Dumas/Publications/DKVZ17.pdf, doi:10.1145/3087604.3087640.

[24] Jean-Guillaume Dumas, David Lucas, and Clément Pernet. Certificates for triangular equivalence and rank profiles. In Safey El Din [52], pages 133–140. URL: http://hal.archives-ouvertes.fr/hal-01466093, doi:10.1145/3087604.3087609.

[25] Jean-Guillaume Dumas and Vincent Zucca. Prover efficient public verification of dense or sparse/structured matrix-vector multiplication. In Josef Pieprzyk and Suriadi Suriadi, editors, *ACISP 2017, 22nd Australasian Conference on Information Security and Privacy*, volume 10343 of *Lecture Notes in Computer Science*, pages 115–134. Springer, July 2017. URL: http://hal.archives-ouvertes.fr/hal-01503870.

[26] Wayne Eberly. A new interactive certificate for matrix rank. Technical Report 2015-1078-11, University of Calgary, June 2015. URL: http://prism.ucalgary.ca/bitstream/1880/50543/1/2015-1078-11.pdf.

[27] Wayne Eberly. Selecting algorithms for black box matrices: checking for matrix properties that can simplify computations. In Gao [34].

[28] Kaoutar Elkhiyaoui, Melek Önen, Monir Azraoui, and Refik Molva. Efficient techniques for publicly verifiable delegation of computation. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '16, pages 119–128, New York, NY, USA, 2016. ACM. doi:10.1145/2897845.2897910.

[29] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In A. M. Odlyzko, editor, *Advances in Cryptology - CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987, 11–15 August 1986. URL: http://www.cs.rit.edu/~jjk8346/FiatShamir.pdf.

[30] Dario Fiore, Cédric Fournet, Esha Ghosh, Markulf Kohlweiss, Olga Ohrimenko, and Bryan Parno. Hash first, argue later: Adaptive verifiable computations on outsourced data. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1304–1316. ACM, 2016. URL: http://doi.acm.org/10.1145/2976749.2978368, doi:10.1145/2976749.2978368.

[31] Dario Fiore and Rosario Gennaro. Publicly verifiable delegation of large polynomials and matrix computations, with applications. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, pages 501–512, New York, NY, USA, 2012. ACM. doi:10.1145/2382196.2382250.

[32] Rūsiņš Freivalds. Fast probabilistic algorithms. In J. Bečvář, editor, *Mathematical Foundations of Computer Science 1979*, volume 74 of *Lecture Notes in Computer Science*, pages 57–69, Olomouc, Czechoslovakia, September 1979. Springer-Verlag. doi:10.1007/3-540-09526-8_5.

[33] Martin Furer, Oded Goldreich, Yishay Mansour, Michael Sipser, and Stathis Zachos. On completeness and soundness in interactive proof systems. In Silvio Micali, editor, *Randomness and Computation*, volume 5, pages 429–442. Advances in Computing Research, JAI Press, Greenwich, Connecticut, 1989. URL: http://www.wisdom.weizmann.ac.il/~oded/PS/fgmsz.ps.

[34] Xiao-Shan Gao, editor. *ISSAC'2016, Proceedings of the 2016 ACM International Symposium on Symbolic and Algebraic Computation, Waterloo, Canada*. ACM Press, New York, July 2016.

[35] Leszek Gąsieniec, Christos Levcopoulos, Andrzej Lingas, Rasmus Pagh, and Takeshi Tokuyama. Efficiently correcting matrix products. *Algorithmica*, pages 1–16, 2016. doi:10.1007/s00453-016-0202-3.

[36] Craig Gentry, Jens Groth, Yuval Ishai, Chris Peikert, Amit Sahai, and Adam Smith. Using fully homomorphic hybrid encryption to minimize

non-interative zero-knowledge proofs. *Journal of Cryptology*, pages 1–24, 2014. doi:10.1007/s00145-014-9184-y.

[37] Pascal Giorgi and Vincent Neiger. Certification of minimal approximant bases. In Arreche [4].

[38] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: interactive proofs for muggles. In Cynthia Dwork, editor, *STOC'2008, Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada*, pages 113–122. ACM Press, May 2008. URL: http://research.microsoft.com/en-us/um/people/yael/publications/2008-delegatingcomputation.pdf, doi:10.1145/1374376.1374396.

[39] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems. In Sedgewick [54], pages 291–304. doi:10.1145/22145.22178.

[40] E. Kaltofen and B. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/90/KaTr90.pdf.

[41] Erich Kaltofen. Analysis of Coppersmith's block Wiedemann algorithm for the parallel solution of sparse linear systems. *Mathematics of Computation*, 64(210):777–806, April 1995. doi:10.2307/2153451.

[42] Erich Kaltofen and Clément Pernet. Sparse polynomial interpolation codes and their decoding beyond half the minimum distance. In Nabeshima [46]. URL: http://arxiv.org/abs/1403.3594.

[43] Erich L. Kaltofen, Michael Nehring, and B. David Saunders. Quadratic-time certificates in linear algebra. In Anton Leykin, editor, *ISSAC'2011, Proceedings of the 2011 ACM International Symposium on Symbolic and Algebraic Computation, San Jose, California, USA*, pages 171–176. ACM Press, New York, June 2011. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/11/KNS11.pdf.

[44] Tracy Kimbrel and Rakesh Kumar Sinha. 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):107–110, February 1993. URL: ftp://trout.cs.washington.edu/tr/1991/08/UW-CSE-91-08-06.pdf.

[45] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, October 1992. doi:10.1145/146585.146605.

[46] Katsusuke Nabeshima, editor. *ISSAC'2014, Proceedings of the 2014 ACM International Symposium on Symbolic and Algebraic Computation, Kobe, Japan*. ACM Press, New York, July 2014.

[47] Edward W. Ng, editor. *EUROSAM '79, International Symposium on Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*. Springer, 1979. doi:10.1007/3-540-09519-5.

[48] Bryan Parno, Jon Howell, Craig Gentry, and Mariana Raykova. Pinocchio: Nearly practical verifiable computation. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, SP '13, pages 238–252, Washington, DC, USA, 2013. IEEE Computer Society. doi:10.1109/SP.2013.47.

[49] Bryan Parno, Mariana Raykova, and Vinod Vaikuntanathan. How to delegate and verify in public: Verifiable computation from attribute-based encryption. In Ronald Cramer, editor, *Theory of Cryptography: 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings*, pages 422–439, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. doi:10.1007/978-3-642-28914-9_24.

[50] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Constant-round interactive proofs for delegating computation. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 49–62. ACM, 2016. URL: http://dl.acm.org/citation.cfm?id=2897518, doi:10.1145/2897518.2897652.

[51] Daniel Roche. Error correction in fast matrix multiplication and inverse. In Arreche [4].

[52] Mohab Safey El Din, editor. *ISSAC'2017, Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation, Kaiserslautern, Deutschland*. ACM Press, New York, July 2017.

[53] Jacob T. Schwartz. Probabilistic algorithms for verification of polynomial identities. In Ng [47], pages 200–215. doi:10.1007/3-540-09519-5_72.

[54] Robert Sedgewick, editor. *STOC '85, ACM Symposium on Theory of Computing, Providence, Rhode Island, USA*. ACM Press, New York, May 1985.

[55] Adi Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, October 1992. doi:10.1145/146585.146609.

[56] Arne Storjohann. Integer matrix rank certification. In John P. May, editor, *ISSAC'2009, Proceedings of the 2009 ACM International Symposium on Symbolic and Algebraic Computation, Seoul, Korea*, pages 333–340. ACM Press, New York, July 2009. URL: https://cs.uwaterloo.ca/~astorjoh/issac09.pdf.

[57] Justin Thaler. Time-optimal interactive proofs for circuit evaluation. In Ran Canetti and JuanA. Garay, editors, *Advances in Cryptology - CRYPTO'13*, volume 8043 of *Lecture Notes in Computer Science*, pages 71–89. Springer Berlin Heidelberg, 2013. doi:10.1007/978-3-642-40084-1_5.

[58] Michael Walfish and Andrew J. Blumberg. Verifying computations without reexecuting them. *Commun. ACM*, 58(2):74–84, January 2015. doi:10.1145/2641562.

[59] Douglas H. Wiedemann. Solving sparse linear equations over finite fields. *IEEE Transactions on Information Theory*, 32(1):54–62, January 1986. doi:10.1109/TIT.1986.1057137.

[60] Yihua Zhang and Marina Blanton. Efficient secure and verifiable outsourcing of matrix multiplications. In Sherman S.M. Chow, Jan Camenisch, Lucas C.K. Hui, and Siu Ming Yiu, editors, *Information Security*, volume 8783 of *Lecture Notes in Computer Science*, pages 158–178. Springer International Publishing, 2014. doi:10.1007/978-3-319-13257-0_10.

[61] Richard Zippel. Probabilistic algorithms for sparse polynomials. In Ng [47], pages 216–226. doi:10.1007/3-540-09519-5_73.