Proof-of-work protocols for AI-generated solutions in mathematics

Erich L. Kaltofen[1,2][0000−0003−2739−3230]

[1] Department of Mathematics, North Carolina State University,
Raleigh, North Carolina 27695-8205, USA
kaltofen@ncsu.edu
https://kaltofen.math.ncsu.edu
[2] Department of Computer Science, Duke University,
Durham, North Carolina 27708-0129, USA

Extended Abstract

Proof-of-work protocols can produce for a massive cloud computation an additional certificate for the output, which allows the verification of the output with much less compute effort and possibly at a later time. The soundness of the protocols prevents the prover from giving a false output. Such protocols should be deployable when the prover is an AI-based solver with possibly incorrect substeps.

We discuss using AI-generated proofs by the example of computing the determinant of a very large, dense $n \times n$ matrix. For purpose of simplifying the discussion, we shall assume that the entries are constant integers, whose bit-size does not grow with the dimension n. We note that the discussed protocols could also account for the bit-lengths in the entries. The best proof-of-work protocol for the integer determinant with an $n\,\mathrm{polylog}(n)$ bit-sized proof and which is checkable in $n^2\,\mathrm{polylog}(n)$ bit-complexity uses interactivity [4] and Figure 1 below. Here $\mathrm{polylog}(n)$ stands for a function of order $O(\log(n)^k)$ for a constant exponent k. Removal of the interactions is possible by the cryptography-based Fiat-Shamir method [5].

Figure 1 gives the proof-of-work protocol, which is adapted from the first protocol in [4, Section 2] and [2, Section 4.4]. In Step 2 in Figure 1, the bound $2^{\mathrm{polylog}(n)}$ can be determined explicitly from the size bounds of the integers in $\mathcal{A}$ and the probability that $\Delta = \det(\mathcal{A})$ after all verifier checks succeed. The multiplicative pre-conditioner $\Gamma(t, s)$ guarantees that the Prover can achieve the relatively prime characteristic polynomials in Figure 1, Steps 5 and 6: $\mathrm{GCD}(c^B(\lambda), c^C(\lambda)) = 1$. Relative primeness is necessary for the soundness of the checks in Figure 1, Steps 8.

The proof for $\Delta = \det(\mathcal{A})$ in Figure 1, which are the prime p, the residues t, s, r_1, the polynomials $c^B(\lambda), c^C(\lambda)$ and the vector w, have a combined bit-size of $n\,\mathrm{polylog}(n)$. The protocol is high-level and does not specify how the Prover computes the integer Δ, the modular polynomials $c^B(\lambda)$, $c^C(\lambda)$ and the modular linear system solution vector w. Here we suppose that that output is computed by an AI-based solver. The protocol essentially detects incorrect outputs, as it would if the Prover had cheated and not computed the output properly.

Fig. 1. A simple integer determinant proof-of-work protocol

$$\mathcal{A} \in \mathbb{Z}^{n\times n} \text{ public}, \quad \det(\mathcal{A}) \neq 0, \quad \Gamma(\sigma,\tau) = \begin{bmatrix} \tau & -1 & 0 & \dots & 0 \\ 0 & \tau & -1 & \ddots & \vdots \\ \vdots & 0 & \ddots & \ddots & 0 \\ 0 & & \ddots & \tau & -1 \\ \sigma & 0 & \dots & 0 & \tau \end{bmatrix} \in \mathbb{Z}[\sigma,\tau]^{n\times n}$$

Prover	*Communication*	*Verifier*
1. $\Delta = \det(\mathcal{A}) \in \mathbb{Z}$	$\xrightarrow{\Delta}$	
2.	$\xleftarrow{p}$	random prime p with $p \leq 2^{\text{polylog}(n)}$
3. $A = (\mathcal{A} \bmod p) \in \mathbb{Z}_p^{n\times n}$		
Steps 4–8 verify $\det(A) \in \mathbb{Z}_p$ *and* $\Delta \equiv \det(A) \pmod p$		
4. $B \equiv A\,\Gamma(t,s)$, $t,s \in \mathbb{Z}_p$ with $t^n + s \not\equiv 0$	$\xrightarrow{t,\,s}$	Checks $t^n + s \equiv \det(\Gamma(t,s)) \not\equiv 0 \pmod p$
5. $c^B(\lambda) = \det(\lambda I_n - B) \in \mathbb{Z}_p[\lambda]$	$\xrightarrow{c^B}$	Note $c^B(0) \equiv (-1)^n \det(A\Gamma(t,s)) \pmod p$
6. $C = [b_{i,j}]_{1\leq i,j\leq n-1} \in \mathbb{Z}_p^{(n-1)\times(n-1)}$, $c^C(\lambda) = \det(\lambda I_{n-1} - C) \in \mathbb{Z}_p[\lambda]$, $t,s \in \mathbb{Z}_p$ also with $\text{GCD}(c^B, c^C) = 1$	$\xrightarrow{c^C}$	Checks $\text{GCD}(c^B(\lambda), c^C(\lambda)) = 1$ in $\mathbb{Z}_p[\lambda]$
7.	$\xleftarrow{r_1}$	$r_1 \in \mathbb{Z}_p$ random with $c^B(r_1) \not\equiv 0 \pmod p$
8. Computes $w \in \mathbb{Z}_p^n$ such that $(r_1 I_n - B)w \equiv e_n \equiv \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$	$\xrightarrow{w}$	Checks $(r_1 I_n - B)w \equiv e_n \pmod p$ and $w_n \equiv \dfrac{c^C(r_1)}{c^B(r_1)} \pmod p$ and $(-1)^n \dfrac{c^B(0)}{t^n + s} \equiv \Delta \pmod p$

A classical method for removing interaction in proof-of-work protocols is in [5]; see also [1]. For Fiat-Shamir'ing, the Prover also selects the random values for p and r_1 in the protocol of Figure 1, but after Δ and after the characteristic polynomials $c^B(\lambda), c^C(\lambda)$ have been computed. One possibility is to use hash values $p = H_1(\mathcal{A},\ \Delta)$ and $r_1 = H_2(\mathcal{A},\ \Delta,\ p,\ c^B,\ c^C)$, where H_1 and H_2 are cryptographically strong hash functions which the Verifier also can compute. For example, H_1 can be random bits of a cryptographically strong random bits generator, which is seeded with $\mathcal{A}$, Δ. The unpredictability of the hashed values/cryptographically strong random bits prevents the Prover, for example, of committing a false Δ value in Step 1 in Figure 1, which still satisfies $\Delta \equiv \det(A) \pmod p$. We note that iteration of the Fiat-Shamir method can lead to insecurity [6].

We do not know a non-interactive protocol for the integer determinant which is of n polylog(n) bit-size and which has an n^2 polylog(n) binary verification complexity. But we note that we have given an n^2 polylog(n) bit-sized non-interactive proof in [11]. The proof traces Arne Storjohann's Las Vegas randomized determinant algorithm [12]. Instead of doing the matrix multiplications, the verifier performs a Freivalds check of the products; see [9, Section 2.2] for Freivalds checking. Here, the prover only can use an AI-based matrix multiplication algorithm, which is much finer grain, and the proof is quadratic size.

We propose that AI-based provers should incorporate the Fiat-Shamir paradigm. Instead of cryptographically strong hash functions/random bits generators, one may use a random bits server in the cloud. The AI-based prover then commits parts of the proof to the cloud before receiving random bits. The Verifier then can check the time-stamps of each step in the interaction between the AI solver and the cloud server. With cloud Fiat-Shamir'ing services, interactive proof-of-work protocols such as the GKR protocol [7,8] or linear algebra protocols [2,3] can be deployed with AI-based solvers. In [2] we give an interactive protocol to verify a sum-of-squares proof [10] without fully computing the sum-of-squares.

Acknowledgment: I thank Lihong Zhi and the organizers of the session "Automating Mathematical Reasoning: Interactions Between Proof Assistants, Computer Algebra Systems (CAS) and Generative AI" at the International Congress on Mathematical Software 2026 for asking me to give a talk.

References

1. Canetti, R., Chen, Y., Holmgren, J., Lombardi, A., Rothblum, G.N., Rothblum, R.D., Wichs, D.: Fiat-Shamir: from practice to theory. In: Charikar, M., Cohen, E. (eds.) Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019. pp. 1082–1090. ACM (2019), URL: https://ia.cr/2018/1004, https://doi.org/10.1145/3313276.3316380
2. Dumas, J.G., Kaltofen, E.: Essentially optimal interactive certificates in linear algebra. In: Nabeshima, K. (ed.) ISSAC 2014 Proc. 39th Internat. Symp. Symbolic Algebraic Comput. pp. 146–153. Association for Computing Machinery, New York, N. Y. (2014), URL: https://kaltofen.math.ncsu.edu/bibliography/14/DuKa14.pdf
3. Dumas, J.G., Kaltofen, E., Lucas, D., Pernet, C.: Elimination-based certificates for triangular equivalence and rank profiles. J. Symbolic Comput. **98**, 246–269 (May–June 2020), special Issue on ISSAC 2017; Mohab Safey El Din, Chee Yap editors. URL: https://kaltofen.math.ncsu.edu/bibliography/18/DKLP18.pdf, https://doi.org/10.1016/j.jsc.2019.07.013
4. Dumas, J.G., Kaltofen, E., Thomé, E., Villard, G.: Linear time interactive certificates for the minimal polynomial and the determinant of a sparse matrix. In: Rosenkranz, M. (ed.) ISSAC'16 Proc. 2016 ACM Internat. Symp. Symbolic Algebraic Comput. pp. 199–206. Association for Computing Machinery, New York, N. Y. (2016), URL: https://kaltofen.math.ncsu.edu/bibliography/16/DKTV16.pdf
5. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Odlyzko, A.M. (ed.) Advances in Cryptology -

CRYPTO'86. Lect. Notes Comput. Sci., vol. 263, pp. 186–194. Springer (11–15 Aug 1986), URL: https://link.springer.com/chapter/10.1007/0-387-34799-2_18
6. Goldwasser, S., Kalai, Y.T.: On the (in)security of the Fiat-Shamir paradigm. In: 44th Symposium on Foundations of Computer Science (FOCS 2003), 11-14 October 2003, Cambridge, MA, USA, Proceedings. pp. 102–113. IEEE Computer Society (2003), URL: https://eprint.iacr.org/2003/034.pdf, https://doi.org/10.1109/SFCS.2003.1238185
7. Goldwasser, S., Kalai, Y.T., Rothblum, G.N.: Delegating computation: Interactive proofs for muggles. J. ACM **62**(4), 27:1–27:64 (2015), URL: http://doi.acm.org/10.1145/2699436
8. Kaltofen, E.: The GKR protocol revisited: Nearly optimal prover-complexity for polynomial-time wiring algorithms and for primality testing in $n^{1/2+o(1)}$ rounds. In: Hashemi, A. (ed.) ISSAC '22 Proc. 2022 ACM Internat. Symp. Symbolic Algebraic Comput. pp. 177–186. Association for Computing Machinery, New York, N. Y. (2022), URL: https://kaltofen.math.ncsu.edu/bibliography/22/Ka22gkr.pdf, https://doi.org/10.1145/3476446.3536183
9. Kaltofen, E.: Encounters in symbolic computation: Ideas for the ages. In: Hauenstein, J., shin Lee, W., Chen, S. (eds.) ISSAC '24 Proc. 2024 ACM Internat. Symp. Symbolic Algebraic Comput. pp. 1–7. Association for Computing Machinery, New York, N. Y. (2024), URL: https://kaltofen.math.ncsu.edu/bibliography/24/Ka24encounters.pdf, https://doi.org/10.1145/3666000.3672619
10. Kaltofen, E., Li, B., Yang, Z., Zhi, L.: Exact certification in global polynomial optimization via sums-of-squares of rational functions with rational coefficients. J. Symbolic Comput. **47**(1), 1–15 (Jan 2012), in memory of Wenda Wu (1929–2009). URL: https://kaltofen.math.ncsu.edu/bibliography/09/KLYZ09.pdf
11. Kaltofen, E., Nehring, M., Saunders, B.D.: Quadratic-time certificates in linear algebra. In: Leykin, A. (ed.) Proc. 2011 Internat. Symp. Symbolic Algebraic Comput. ISSAC 2011. pp. 171–176. Association for Computing Machinery, New York, N. Y. (Jun 2011), URL: https://kaltofen.math.ncsu.edu/bibliography/11/KNS11.pdf
12. Storjohann, A.: The shifted number system for fast linear algebra on integer matrices. J. Complexity **21**(5), 609–650 (2005)