Essentially Optimal Interactive Certificates in Linear Algebra

Erich L. Kaltofen
North Carolina State University
`google->kaltofen`

Joint with Jean-Guillaume Dumas
University of Grenoble, France

Sparse Matrix GL7d19

From K-Theory Conjectures [Elbaz-Vincent, Gangle, Soulé ’05]

1911130×1955309 matrix of rank 1033568

Computed by J.-G. Dumas with LinBox in 1050 CPU days

With Monte-Carlo randomized algorithm ...

Do you believe the rank?

Sparse Matrix GL7d19

From K-Theory Conjectures [Elbaz-Vincent, Gangle, Soulé '05]

1911130×1955309 matrix of rank 1033568

Computed by J.-G. Dumas with LinBox in 1050 CPU days

With Monte-Carlo randomized algorithm ...

Do you believe the rank?

We construct an easily checkable certificate

Kaltofen, Li, Yang, Zhi 2009

*"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 **valdidated by the algorithm** (adversary-verifier model)."*

Kaltofen, Li, Yang, Zhi 2009

*"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 **valdidated by the algorithm** (adversary-verifier model)."*

Fancy setting: Certificates are produced by the
"prover (Peggy)" in the Cloud,
which the "verifier (Victor)" client user checks

Kaltofen, Li, Yang, Zhi 2009

> *"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 **valdidated by the algorithm** (adversary-verifier model)."*

What it is NOT: programs that check their results [Blum et al.]

What it is: limiting prover power in delegated computation [Goldwasser et. al. 2008]: more on this later

Warm-up: Rusin Freivalds's 1979 Certificates

Let $A, B, C \in \mathbb{Z}^{n \times n}$

Certify $C = A \cdot B$ via a random vector $y \in S^n$, $S \subseteq \mathbb{Z}$

and check $Cy = A(By):$ Monte Carlo of $n^{2+o(1)}$ bit complexity

Warm-up: Rusin Freivalds's 1979 Certificates

Let $A, B, C \in \mathbb{Z}^{n\times n}$
Certify $C = A \cdot B$ via a random vector $y \in S^n$, $S \subseteq \mathbb{Z}$
and check $Cy = A(By)$: Monte Carlo of $n^{2+o(1)}$ bit complexity

Kimbrel and Sinha 1993: $O(\log n)$ random bits
Choose $y = [1, r, r^2, \ldots, r^{n-1}]^T$, $r \in S$
If $|S| \geq 2n$, then $\geq n$ of the r certify $C \neq A \cdot B$
Otherwise $(C - A \cdot B) \cdot (\text{non-sing. Vandermonde matrix}) = 0.$

Warm-up: Rusin Freivalds's 1979 Certificates

Let $A, B, C \in \mathbb{Z}^{n \times n}$
Certify $C = A \cdot B$ via a random vector $y \in S^n$, $S \subseteq \mathbb{Z}$
and check $Cy = A(By)$: Monte Carlo of $n^{2+o(1)}$ bit complexity

Kimbrel and Sinha 1993: $O(\log n)$ random bits
Choose $y = [1, r, r^2, \ldots, r^{n-1}]^T$, $r \in S$
If $|S| \geq 2n$, then $\geq n$ of the r certify $C \neq A \cdot B$
Otherwise $(C - A \cdot B) \cdot$ (non-sing. Vandermonde matrix) $= 0$.

Non-singularity certificate
Smallish prime p, $L, U \in \mathbb{Z}_p^{n \times n}$, P permut. matrix
Verify $(A \bmod p) \equiv LUP \pmod{p}$ as above

Warm-up continued: Singularity, Rank

Singularity certificate

$y \in \mathbb{Z}^n$ with $\log \|y\| = n^{1+o(1)}$ such that $Ay = 0$

Verify for smallish random prime p: $(A \bmod p)(y \bmod p) \equiv 0$

Note: $y \bmod p$ also takes $n^{2+o(1)}$ bit operations

Warm-up continued: Singularity, Rank

Singularity certificate

$y \in \mathbb{Z}^n$ with $\log \|y\| = n^{1+o(1)}$ such that $Ay = 0$

Verify for smallish random prime p: $(A \bmod p)(y \bmod p) \equiv 0$

Note: $y \bmod p$ also takes $n^{2+o(1)}$ bit operations

Rank certificate [Kaltofen, Nehring, Saunders 2011]

List of $2n^{1+o(1)}$ smallish primes $p_i, L^{[i]}, U^{[i]}, P^{[i]}$ with

$$\mathrm{rank}(A) = \mathrm{rank}(U^{[i]}) \quad \text{and} \quad L^{[i]}AP^{[i]} \equiv U^{[i]} \pmod{p_i}$$

Verify for random j rank of $U^{[j]}$ and modular row echelon form

Note: only $n^{1+o(1)}$ bad p_i that can lie about the rank undetectably

But: certificate occupies $n^{3+o(1)}$ bit space

Bit complexity of the Determinant/Rank

ω: matrix-multiplication exponent: best $\omega = 2.372864$

$\log\|A\|$: bit-size of entries in $A \in \mathbb{Z}^{n\times n}$

Det with Chinese remaindering: $(n \cdot \log\|A\|)^{1+o(1)} \times n^{\omega}$

Monte-Carlo Rank = Rank modulo a random smallish prime

$$(n^2 \log\|A\| + n^{\omega} \operatorname{loglog}\|A\|)^{1+o(1)}$$

Monte-Carlo Rank $r = n^{2/\omega+o(1)}$: $(n^2 \log\|A\|)^{1+o(1)}$

[Essentially optimal!]

Las-Vegas Det + Rank: $(n^{\omega} \log\|A\|)^{1+o(1)}$

[Storjohann 2002, 2009]

Certificates for Det/Rank of $n^{2+o(1)}$ Bit Complexity [Kaltofen, Nehring, Saunders 2011]

Step 1: Run Storjohann's Las Vegas algorithms

Step 2: Record all random choices and intermediate results except in matrix multiplications

Step 3: For the matrix multiplications, record inputs and outputs

Certificates for Det/Rank of $n^{2+o(1)}$ Bit Complexity [Kaltofen, Nehring, Saunders 2011]

Step 1: Run Storjohann's Las Vegas algorithms

Step 2: Record all random choices and intermediate results except in matrix multiplications

Step 3: For the matrix multiplications, record inputs and outputs

Verfication: rerun Storjohann's algorithms, making the same random choices and instead of the matrix multiplications, verify the $AB = C$ by Freivalds's algorithm

It's like running the det/rank algorithms with a quadratic matrix multiplication procedure

Baby steps/giant steps bit complexity exponent η via block Wiedemann [Kaltofen&Villard 2001, 2004]

	ω	ζ	η	σ	τ
1	ω	ζ	$\omega + \frac{1-\zeta}{\omega^2-(2+\zeta)\omega+2}$	$1 - \frac{\omega-(1+\zeta)}{\omega^2-(2+\zeta)\omega+2}$	$\frac{\omega-2}{\omega^2-(2+\zeta)\omega+2}$
2	2.372864	0.3029805	2.694691*	0.506016	0.172158
3	ω	0	$\omega + \frac{1}{(\omega-1)^2+1}$	$1 - \frac{\omega-1}{(\omega-1)^2+1}$	$\frac{\omega-2}{(\omega-1)^2+1}$
4	3	0	$3+\frac{1}{5}$	$\frac{3}{5}$	$\frac{1}{5}$
5	$\log_2(7)$	0	3.041738	0.576388	0.189230
6	2.372864	0	2.719514	0.524070	0.129253
7	2	0	$2+\frac{1}{2}$	$\frac{1}{2}$	0

*Best known for MINPOLY, CHARPOLY, FROBENIUS, SMITH

—all Monte-Carlo

[Also best known for **algebraic division-free** Determinant]

Monte-Carlo Yields Cryptographically Strong Certificates [Kaltofen 2012]

Prevent cheating with "unlucky" random choices by fixing a pseudo-random bits generator and always seed it the same, dependent on input matrix: $s = \sum |a_{i,j}| \bmod 2^{64}$

At ISSAC 2014, we used a cryptographic hash value (random oracle function) $\mathrm{hash}(A)$

But:
With Kaltofen-Villard algorithms—rerun with Freivalds's matrix multiplication check—we get CHARPOLY/MINPOLY certificates of verification complexity $(n^{2.5} \log \|A\|)^{1+o(1)}$:
[Superlinear in input size!]

Monte-Carlo Yields Cryptographically Strong Certificates [Kaltofen 2012]

Prevent cheating with "unlucky" random choices by fixing a pseudo-random bits generator and always seed it the same, dependent on input matrix: $s = \sum |a_{i,j}| \bmod 2^{64}$

At ISSAC 2014, we used a cryptographic hash value (random oracle function) $\mathrm{hash}(A)$

But:
With Kaltofen-Villard algorithms—rerun with Freivalds's matrix multiplication check—we get CharPoly/MinPoly certificates of verification complexity $(n^{2.5} \log \|A\|)^{1+o(1)}$:

[Superlinear in input size!]

Our original motivation for certificates:
Certify a symmetric matrix positive semidefinite

[Kaltofen, Li, Yang, Zhi 2009]

Our motivation: sum-of-squares proofs in global optim.

For a real polynomial $f \in \mathbb{R}[X_1, \ldots, X_n]$:

$\boxed{f \succeq 0}$ (f is *positive semidefinite*)

$$\iff \forall \xi_1, \ldots, \xi_n \in \mathbb{R} \colon f(\xi_1, \ldots, \xi_i) \geq 0,$$

$\boxed{f \succ 0}$ (f is *positive definite*)

$$\iff \forall \xi_1, \ldots, \xi_n \in \mathbb{R} \colon f(\xi_1, \ldots, \xi_i) > 0.$$

Note: $\mu = \inf_{\xi \in \mathbb{R}} f(\xi) \Longrightarrow f - \mu \succeq 0$

Our motivation: sum-of-squares proofs in global optim.

For a real polynomial $f \in \mathbb{R}[X_1, \ldots, X_n]$:

$\boxed{f \succeq 0}$ (f is *positive semidefinite*)

$$\iff \forall \xi_1, \ldots, \xi_n \in \mathbb{R}\colon\ f(\xi_1, \ldots, \xi_i) \geq 0,$$

$\boxed{f \succ 0}$ (f is *positive definite*)

$$\iff \forall \xi_1, \ldots, \xi_n \in \mathbb{R}\colon\ f(\xi_1, \ldots, \xi_i) > 0.$$

Note: $\mu = \inf_{\xi \in \mathbb{R}} f(\xi) \implies f - \mu \succeq 0$

For a real **symmetric** matrix $W \in \mathbb{R}^{N \times N}$, all of whose eigenvalues are necessarily $\in \mathbb{R}$:

$\boxed{W \succeq 0}$ if W is positive semidefinite, i.e.,

all eigenvalues of W are ≥ 0;

$\boxed{W \succ 0}$ if W is positive definite, i.e.,

all eigenvalues of W are > 0 ($\implies W$ is nonsingular).

Our motivation: sum-of-squares proofs in global optim.

For a real polynomial $f \in \mathbb{R}[X_1, \ldots, X_n]$:

$\boxed{f \succeq 0}$ (f is *positive semidefinite*)

$$\iff \forall \xi_1, \ldots, \xi_n \in \mathbb{R}\colon\ f(\xi_1, \ldots, \xi_i) \geq 0,$$

$\boxed{f \succ 0}$ (f is *positive definite*)

$$\iff \forall \xi_1, \ldots, \xi_n \in \mathbb{R}\colon\ f(\xi_1, \ldots, \xi_i) > 0.$$

Note: $\mu = \inf_{\xi \in \mathbb{R}} f(\xi) \implies f - \mu \succeq 0$

For a real **symmetric** matrix $W \in \mathbb{R}^{N \times N}$, all of whose eigenvalues are necessarily $\in \mathbb{R}$:

$\boxed{W \succeq 0}$ if W is positive semidefinite, i.e.,

all eigenvalues of W are ≥ 0;

Note: $W \succeq 0 \iff \pm f^W(-x) = \prod_\alpha (x + \alpha)$ has coeff's ≥ 0

where f^W is $\textsc{CharPoly}/\textsc{MinPoly}(W)$

Certification of Lower Bounds

Emil Artin's 1927 Theorem (Hilbert's 17th Problem)

$$f \in \mathbb{Q}[X_1,\ldots,X_n]: \quad f \succeq 0$$

$$\Updownarrow$$

$$\exists u_i, v_j \in \mathbb{Q}[X_1,\ldots,X_n]: f(X_1,\ldots,X_n) = \frac{\sum_{i=1}^m u_i^2}{\sum_{j=1}^m v_j^2}$$

$$\Updownarrow$$

$$\exists \text{rational } W^{[1]} \succeq 0, W^{[2]} \succeq 0: f = \frac{m_d^T\, W^{[1]}\, m_d}{m_e^T\, W^{[2]}\, m_e}$$

with $m_d(X_1,\ldots,X_n)$, $m_e(X_1,\ldots,X_n)$ vectors of terms

$W \succeq 0$ (positive semidefinite)

$\Longleftrightarrow W = PL\,D\,L^T P^T$, D diagonal, $D_{i,i} \geq 0$ (Cholesky)

Theodore Motzkin's 1967 Polynomial

$$(3 \text{ arithm. mean} - 3 \text{ geom. mean})(x^4y^2, x^2y^4, z^6)$$
$$= x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$$

is positive semidefinite (AGM inequality) but **not** a sum-of-squares.

However,

$$(x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2)(x^2 + y^2 + z^2) =$$
$$(z^4 - x^2y^2)^2 + 3\left(xyz^2 - \frac{xy^3}{2} - \frac{x^3y}{2}\right)^2 + \left(\frac{xy^3}{2} - \frac{x^3y}{2}\right)^2$$
$$+ (xz^3 - xy^2z)^2 + (yz^3 - x^2yz)^2$$

Theodore Motzkin's 1967 Polynomial

$$(3 \text{ arithm. mean} - 3 \text{ geom. mean})(x^4y^2, x^2y^4, z^6)$$
$$= x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2$$

is positive semidefinite (AGM inequality) but **not** a sum-of-squares.

However,

$$(x^4y^2 + x^2y^4 + z^6 - 3x^2y^2z^2)(x^2 + z^2) =$$
$$\left(z^4 - x^2y^2\right)^2 + \left(xyz^2 - x^3y\right)^2 + \left(xz^3 - xy^2z\right)^2$$

Proof of Knowledge 2 Rounds Protocol [Chaum, Evertse, van de Graaf, Peralta 1986]

Public p, g where g is primitive root modulo prime p

Given x, computing $v = (g^x \bmod p)$ is easy,
but given v, x cannot be computed fast [Discrete Log Problem]

Proof of Knowledge 2 Rounds Protocol [Chaum, Evertse, van de Graaf, Peralta 1986]

Public p, g where g is primitive root modulo prime p

Prover "Peggy" must convince Verifier "Victor"
that she has ID x without revealing x

Prover	*Commun.*	*Verifier*
Chooses $r \in \mathbb{Z}_{p-1}$ randomly		
$v = (g^x \bmod p)$,		
$h = (g^r \bmod p)$	$\xrightarrow{v, h}$	
	"commits"	
	$\xleftarrow{c}$	Chooses $c \in \mathbb{Z}_{p-1}$ randomly
$s = (r + cx) \bmod (p-1)$	$\xrightarrow{s}$	Checks $g^s \equiv hv^c \pmod{p}$

Non-Interactive Proof of Knowledge Protocol [Fiat, Shamir 1986]

Public p, g where g is primitive root modulo prime p

Prover "Peggy" must convince Verifier "Victor"
that she has ID x without revealing x

Prover	*Commun.*	*Verifier*
Chooses $r \in \mathbb{Z}_{p-1}$ randomly		
$v = (g^x \bmod p)$,		
$h = (g^r \bmod p)$	$\xrightarrow{v,h}$	
$c = \text{hash}(p, g, v, h)$	$\xrightarrow{c}$	
$s = (r + cx) \bmod (p-1)$	$\xrightarrow{s}$	Checks $c = \text{hash}(p, g, v, h)$
		Checks $g^s \equiv hv^c \pmod{p}$

Matrix Rank Certificate As Interactive Protocol

Prover "Peggy" must convince Verifier "Victor"
that $r = \mathrm{rank}(A), A \in \mathbb{Z}^{n \times n}$

Prover	*Commun.*	*Verifier*
	$\xleftarrow{\quad p \quad}$	Chooses smallish p randomly
Computes LUP factorization of A modulo p	$\xrightarrow{L^{[p]}, U^{[p]}, P}$	Checks $L^{[p]}AP \equiv U^{[p]} \pmod{p}$ via Freivalds algorithm Then $r = \mathrm{rank}(U^{[p]})$ w.h.p

Kaltofen's 2012 Matrix Rank Certificate
As Fiat-Shamir Non-Interactive Protocol

Prover "Peggy" must convince Verifier "Victor"
that $r = \mathrm{rank}(A), A \in \mathbb{Z}^{n\times n}$

Prover	*Commun.*	*Verifier*
$p = \mathrm{hash}(A)$	$\xrightarrow{p}$	
Computes LUP factorization of A modulo p	$\xrightarrow{L^{[p]}, U^{[p]}, P}$	Checks $p = \mathrm{hash}(A)$
		Checks $L^{[p]}AP \equiv U^{[p]} \pmod{p}$ via Freivalds's algorithm
		Then $r = \mathrm{rank}(U^{[p]})$ w.h.p

Dumas's & Kaltofen's 2014 CharPoly Certificate As 2 Round Interactive Protocol

Prover "Peggy" must convince Verifier "Victor"
that $c^A(\lambda) = \det(\lambda I - A), A \in \mathbb{Z}^{n\times n}$

Prover	*Commun.*	*Verifier*
$c^A(\lambda) = \det(\lambda I - A)$	$\xrightarrow{c^A(\lambda)}$ "commits"	
		p a smallish random prime
	$\xleftarrow{p,r}$	r a smallish random integer
Computes LUP factor. of $rI - A$ modulo p "Certificate for $\det(rI - A)$"	$\xrightarrow{L^{[p]}, U^{[p]}, P}$	Checks $L^{[p]}(rI - A)P \equiv U^{[p]} \pmod{p}$ by Freivalds's alg. Checks $\det(U^{[p]}) \equiv c^A(r) \pmod{p}$

At Last: Dumas's & Kaltofen's 2014 CharPoly Certificate As Non-Interactive Protocol With Optimal Verifier

Prover "Peggy" must convince Verifier "Victor"
that $c^A(\lambda) = \det(\lambda I - A), A \in \mathbb{Z}^{n\times n}$

Prover	*Commun.*	*Verifier*
$c^A(\lambda) = \det(\lambda I - A)$	$\xrightarrow{c^A(\lambda)}$	
$p, r = \text{hash}(A, c^A)$	$\xrightarrow{p,r}$	
Computes LUP factor.		
of $rI - A$ modulo p	$\xrightarrow{L^{[p]}, U^{[p]}, P}$	Checks $p, r = \text{hash}(A, c^A)$
		Checks $L^{[p]}(rI - A)P$
		$\equiv U^{[p]} \pmod{p}$ by Freivalds's alg.
		Checks $\det(U^{[p]}) \equiv c^A(r) \pmod{p}$

At Last: Dumas's & Kaltofen's 2014 CharPoly Certificate As Non-Interactive Protocol With Optimal Verifier

Prover "Peggy" must convince Verifier "Victor"
that $c^A(\lambda) = \det(\lambda I - A), A \in \mathbb{Z}^{n\times n}$

Prover	*Commun.*	*Verifier*
$c^A(\lambda) = \det(\lambda I - A)$	$\xrightarrow{c^A(\lambda)}$	
$p, r = \text{hash}(A, c^A)$	$\xrightarrow{p, r}$	
Computes LUP factor. of $rI - A$ modulo p	$\xrightarrow{L^{[p]}, U^{[p]}, P}$	Checks $p, r = \text{hash}(A, c^A)$ Checks $L^{[p]}(rI - A)P \equiv U^{[p]} \pmod{p}$ by Freivalds's alg. Checks $\det(U^{[p]}) \equiv c^A(r) \pmod{p}$

Verification complexity: $(n^2 \log \|A\|)^{1+o(1)}$

Dumas's & Kaltofen's 2014 Sparse Matrix Rank Certificate As Interactive Protocol

Prover "Peggy" must convince Verifier "Victor"
that $r = \mathrm{rank}(A), A \in \mathbb{Z}^{n\times n}$, A has $n^{1+o(1)}$ non-zero entries

Prover	*Commun.*	*Verifier*
		p a smallish random prime
		a random $b \in \mathbb{Z}_p^n$
	$\xleftarrow{p,b,T^{[1]},T^{[2]}}$	$T^{[1]}, T^{[2]} \in \mathbb{Z}_p$ 2 random Toeplitz matrices
Computes x	$\xrightarrow[\text{“Certificate for non-singularity”}]{x \in \mathbb{Z}_p^r \text{ s.t. } (T^{[1]}AT^{[2]})_{1\ldots r,1\ldots r}\, x \equiv b_{1\ldots r}}$	
Computes w	$\xrightarrow[\text{“Certificate for singularity”}]{0 \neq w \in \mathbb{Z}_p^{r+1} \text{ s.t. } (T^{[1]}AT^{[2]})_{1\ldots r+1,1\ldots r+1}\, w \equiv 0}$	

Dumas's & Kaltofen's 2014 Sparse Matrix Rank Certificate As Interactive Protocol

Prover "Peggy" must convince Verifier "Victor"
that $r = \text{rank}(A), A \in \mathbb{Z}^{n\times n}$, A has $n^{1+o(1)}$ non-zero entries

Prover	*Commun.*	*Verifier*
		p a smallish random prime
		a random $b \in \mathbb{Z}_p^n$
	$\xleftarrow{p, b, T^{[1]}, T^{[2]}}$	$T^{[1]}, T^{[2]} \in \mathbb{Z}_p$ 2 random Toeplitz matrices
Computes x	$\xrightarrow{x \in \mathbb{Z}_p^r \text{ s.t. } (T^{[1]} A T^{[2]})_{1\ldots r, 1\ldots r}\, x \equiv b_{1\ldots r}}$ "Certificate for non-singularity"	
Computes w	$\xrightarrow{0 \neq w \in \mathbb{Z}_p^{r+1} \text{ s.t. } (T^{[1]} A T^{[2]})_{1\ldots r+1, 1\ldots r+1}\, w \equiv 0}$ "Certificate for singularity"	

Verification complexity: $(n \log \|A\|)^{1+o(1)}$

Commun. complexity: $(n \log\log \|A\|)^{1+o(1)}$

Goldwasser, Kalai, Rothblum 2008: Delegating Computation: Interactive Proofs for “Muggles”

Theorem: *Let* C_N *be a family of log-space uniform Boolean circuit with* N *inputs. Then any computation has a randomized interactive proof protocol with*

Verifier complexity: $(N + \mathrm{depth}(C_N)) \times (\log N)^{O(1)}$

Prover complexity: $\mathrm{size}(C_N)^{O(1)}$

Goldwasser, Kalai, Rothblum 2008: Delegating Computation: Interactive Proofs for "Muggles"

Theorem: *Let C_N be a family of log-space uniform Boolean circuit with N inputs. Then any computation has a randomized interactive proof protocol with*

Verifier complexity: $(N + \mathrm{depth}(C_N)) \times (\log N)^{O(1)}$

Prover complexity: $\mathrm{size}(C_N)^{O(1)}$

Construction based on Probabil. Checkable Proofs (PCP): Prover compresses levels of the evaluated circuit by a linear form, the verifier performs a single Boolean operation on the levels

Goldwasser, Kalai, Rothblum 2008: Delegating Computation: Interactive Proofs for "Muggles"

Theorem: *Let C_N be a family of log-space uniform Boolean circuit with N inputs. Then any computation has a randomized interactive proof protocol with*

Verifier complexity: $(N + \mathrm{depth}(C_N)) \times (\log N)^{O(1)}$

Prover complexity: $\mathrm{size}(C_N)^{O(1)}$

Construction based on Probabil. Checkable Proofs (PCP):
Prover compresses levels of the evaluated circuit by a linear form, the verifier performs a single Boolean operation on the levels

Thaler 2012: Better prover complexity: $O(\mathrm{size}(C_N))$

Our customized certificates are:
– Independent of the circuits that compute them: expose bugs in C_N
– Optimal prover complexity: $\mathrm{size}(C_N) + o(\mathrm{size}(C_N))$

Open Problems

Certificates for Smith Normal Form of Dense Integer Matrices

Certificates for Determinant of a Sparse Matrix

And, of course,
Remove the cryptographic assumptions for CHARPOLY

Open Problems

Certificates for Smith Normal Form of Dense Integer Matrices

Certificates for Determinant of a Sparse Matrix

And, of course,
Remove the cryptographic assumptions for CHARPOLY

Right now, applies crypto to symbolic computation, not the other way around, unlike Groebner breaking encryption schemes

Thank you!

"End Key" wrong!