Practical Verifiable computing
exact linear algebra certificates

Jean-Guillaume Dumas

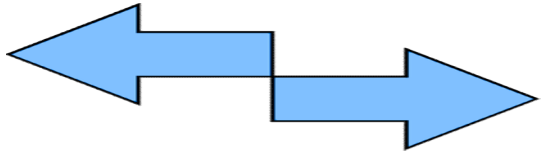
Laboratoire
Jean Kuntzmann
Informatique et Mathématiques Appliquées

Delegating computation

- Cloud computing
  - Businesses buy computing power from a service provider
    - No need to provision and maintain hardware
    - Pay for what you need, scalability
    - Small devices outsourcing complex computing problems to larger servers

  ⇨ Issue: correctness of result?

[blog.fi-xifi.eu]



[www.psdgraphics.com]

RSYSTEMS
powering innovation

We run clusters so you don't have to....

PENGUIN COMPUTING | ON DEMAND

SABALCORE
Where Data Becomes Discovery

Google Cloud Platform

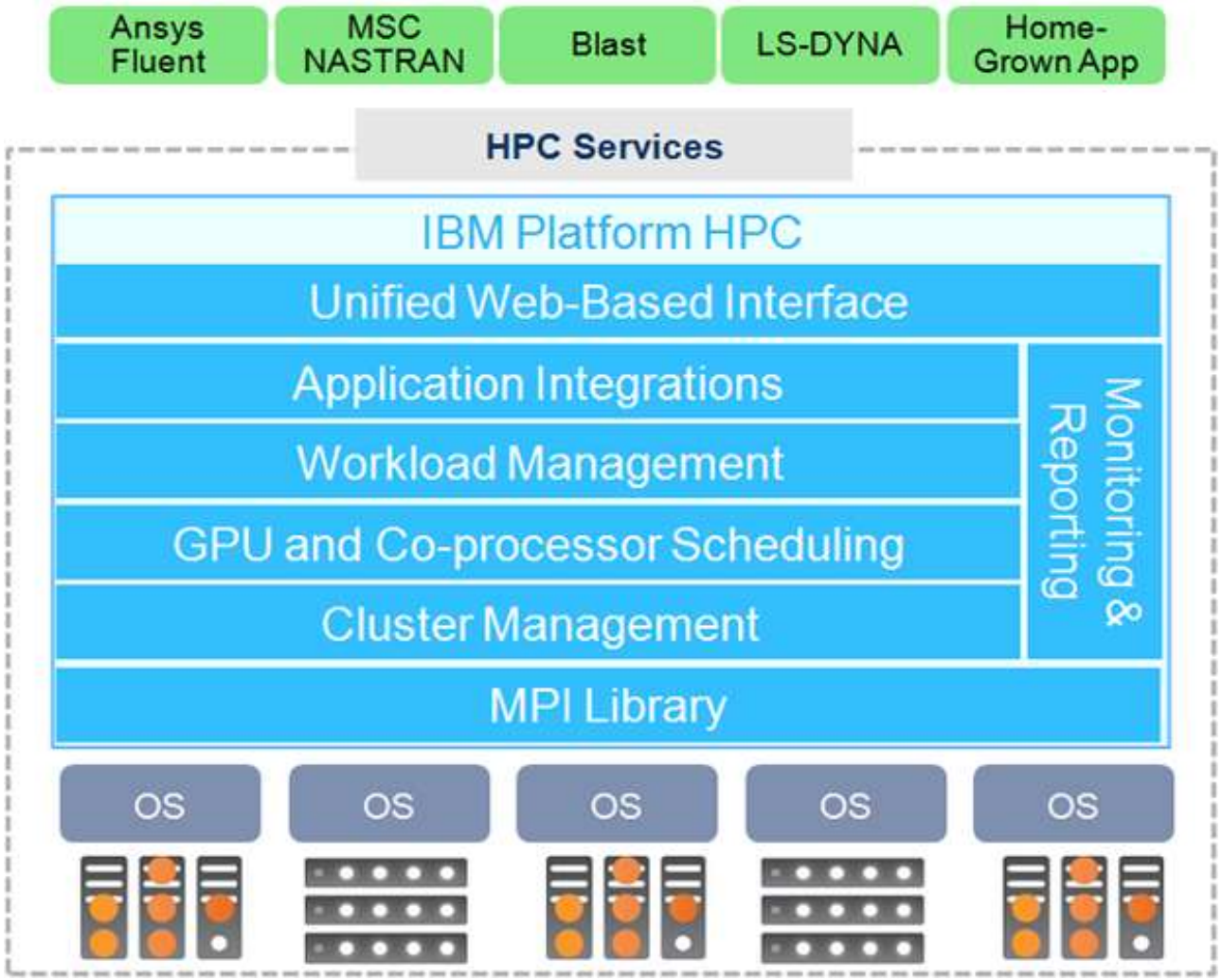
amazon web services

Microsoft Azure

ORACLE
grid engine

IBM Platform Computing

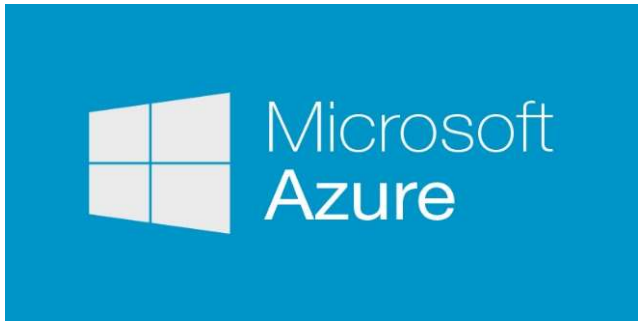
High-performance as a service



[http://www-03.ibm.com/systems/platformcomputing/products/hpc/]

Azure example fares

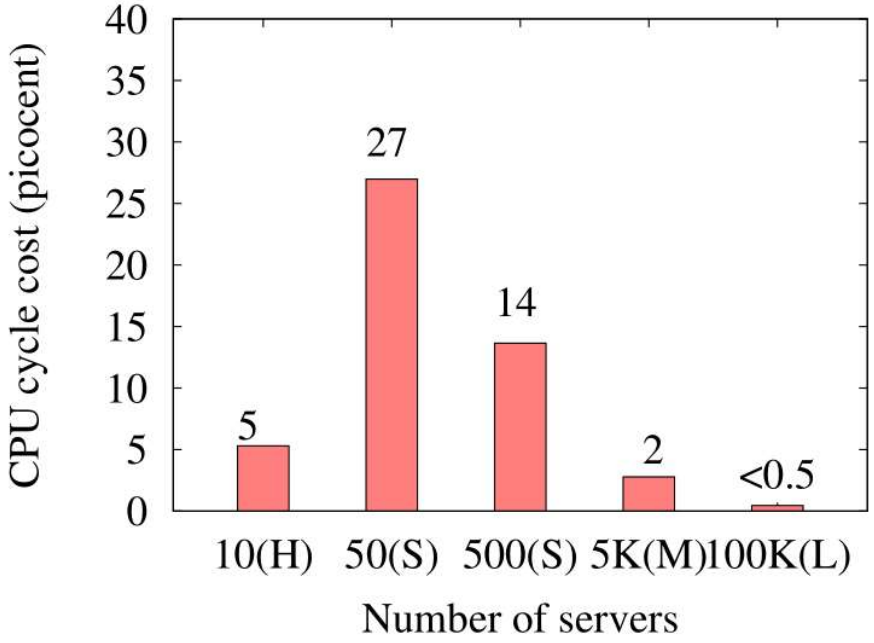
Cores	RAM	Disk Sizes	Price
1	**0.75** GB	**19** GB	$0.02/hour (~$15/month)
1	**1.75** GB	**224** GB	$0.08/hour (~$60/month)
2	**3.5** GB	**489** GB	$0.16/hour (~$119/month)
4	**7** GB	**999** GB	$0.32/hour (~$238/month)
8	**14** GB	**2,039** GB	$0.64/hour (~$476/month)

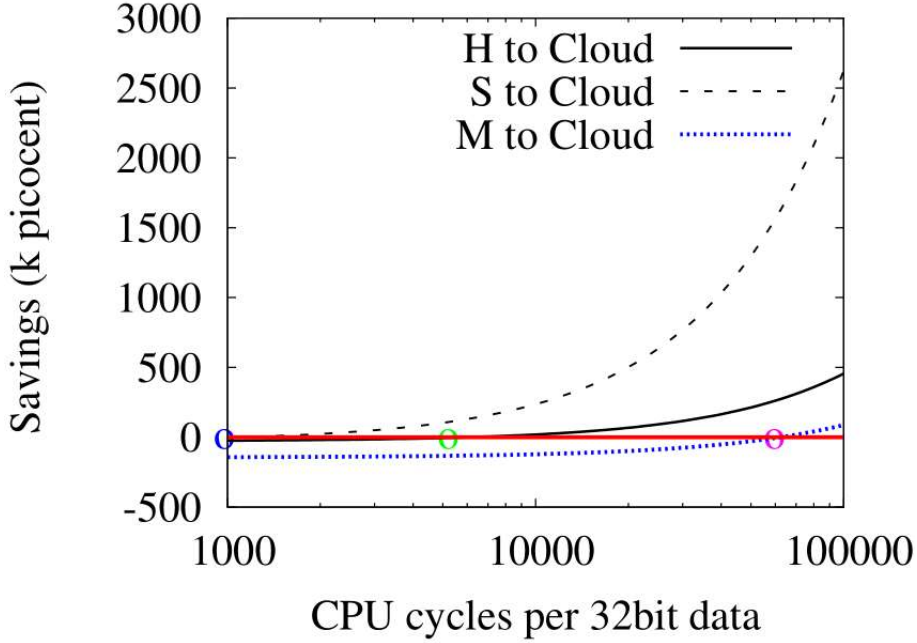


[https://azure.microsoft.com/en-us/pricing/details/cloud-services/]

To Cloud Or Not To Cloud? Musings On Costs and Viability

- *[Chen, Sion 2011]*
  - Home users (H), Small/Mid-size/Large Enterprises (S,M,L)





  - Savings = Cycles $\times$ ($Cost_{Local}$-$Cost_{Cloud}$) – DataTransfer

Contents

- Outsourcing
- Verifiable computing
- Certificates for Dense Matrices
- Certificates for Sparse Matrices

Contents

- Outsourcing
- Verifiable computing
  - Clouds offer no guarantee
  - Interactive certificates
  - Public/private verification
  - Probabilistic verification
- Certificates for Dense Matrices
- Certificates for Sparse Matrices

http://aws.amazon.com/agreement/ *[Thaler]*

Amazon EC2

- **10. Disclaimers: amazon elastic compute cloud**
- THE SERVICE OFFERINGS ARE PROVIDED "AS IS."

WE AND OUR AFFILIATES AND LICENSORS MAKE NO [...] WARRANTY THAT THE SERVICE OFFERINGS OR THIRD PARTY CONTENT WILL BE UNINTERRUPTED, ERROR FREE OR FREE OF HARMFUL COMPONENTS,

OR THAT ANY CONTENT, INCLUDING YOUR CONTENT OR THE THIRD PARTY CONTENT, WILL BE SECURE OR NOT OTHERWISE LOST OR DAMAGED.

https://cloud.google.com/terms/

- **12. Disclaimer: Google Compute Engine**

- NEITHER GOOGLE NOR ITS SUPPLIERS, WARRANTS THAT THE OPERATION OF THE SOFTWARE OR THE SERVICES WILL BE ERROR-FREE OR UNINTERRUPTED.
- NEITHER THE SOFTWARE NOR THE SERVICES ARE DESIGNED, MANUFACTURED, OR INTENDED FOR HIGH RISK ACTIVITIES.

Privately Verifiable (outsourced) computation

- Client (Verifier, Victor) sends
  - a function F and an input x to the server
- The Server (Prover, Peggy) returns
  - $y=F(x)$ and Π, a proof that y is correct

[blog.fi-xifi.eu]

F, x

y=F(x), proof Π

[www.psdgraphics.com]

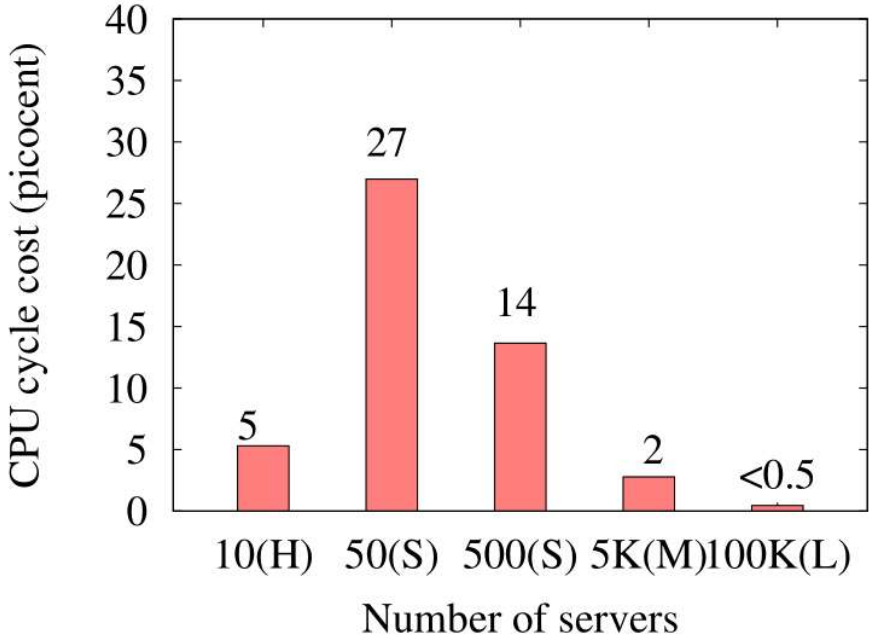
- Verifying Π, should take less time than computing $F(x)$

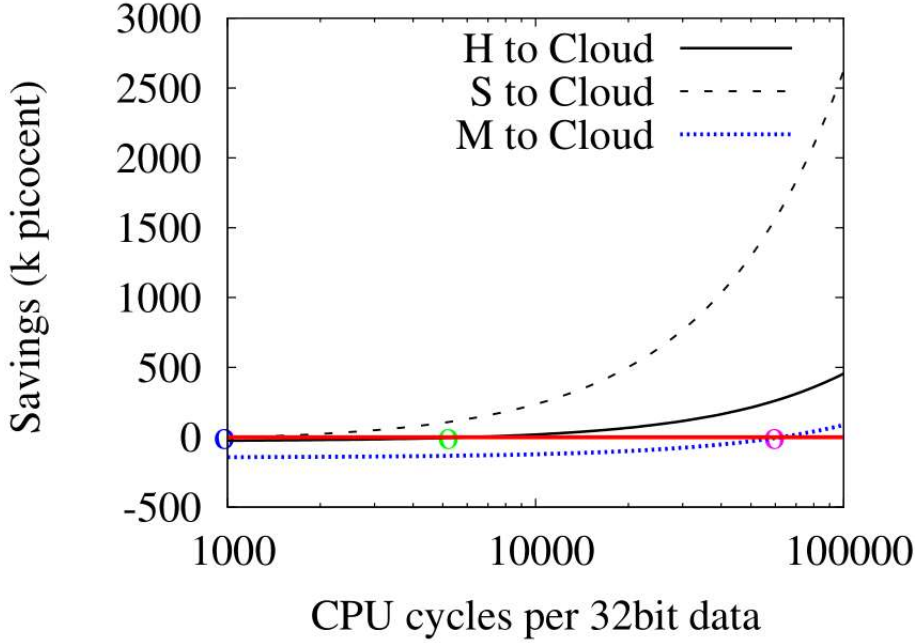
Goals of verifiable computation

- Provide user with guarantee of correctness without requiring to perform full computation
  - Ideally not much more than reading input/output
- Minimize extra effort required for cloud to provide correctness guarantee
  - Ideally not much more than just solve the problem
- Achieve protocols:
  - Secure against malicious clouds
  - Lightweight in benign settings

To Cloud Or Not To Cloud? Viability of verifiability

- *[Chen, Sion 2011]*
  - Savings = Cycles × ($Cost_{Local}$-$Cost_{Cloud}$) – DataTransfer





- Verifiability

$$Cycles \times Cost_{Local} \geq Cycles_{Verifier} \times Cost_{Local} + Cycles_{Prover} \times Cost_{Cloud} + DataTransfer$$

Approaches

1. Strong assumptions on the cloud
   - Replication: majority of responses have to be correct
   - Trusted hardware

2. Minimal assumptions
   - Interactive proofs:
     - Generic approaches certifying the algorithm (if in NC)
       [Goldwasser et al.' 08 … Thaler et al.'13]
     - Ad-hoc approaches certifying the result
   - Amortized systems (homomorphic cryptography) *[Gentry et al.'13]*

3. Using 2 or more clouds
   - Refereed games: 1 cloud has to be honest
   - Multi-prover interactive proofs: non-communicating clouds

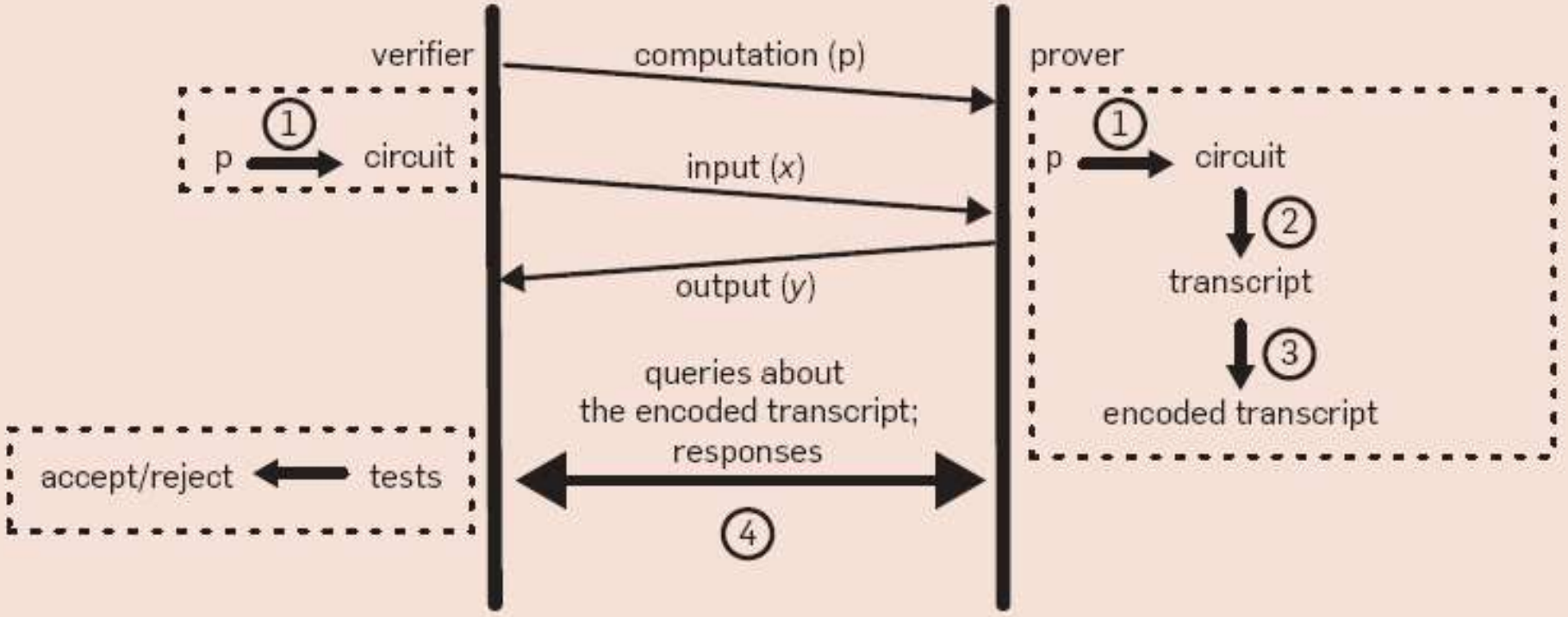
Private verifiability in interactive proofs

- Prover P, Peggy
- Verifier V, Victor

- Peggy solves problem, tells Victor the answer
  - Peggy and Victor have a conversation
  - Peggy's goal: convince Victor of the correctness of her answer

- Requirements
  1. Completeness: an honest P can convince V to accept
  2. Soundness: V will catch lying P with high probability
     - Secure even if P is computationally unbounded

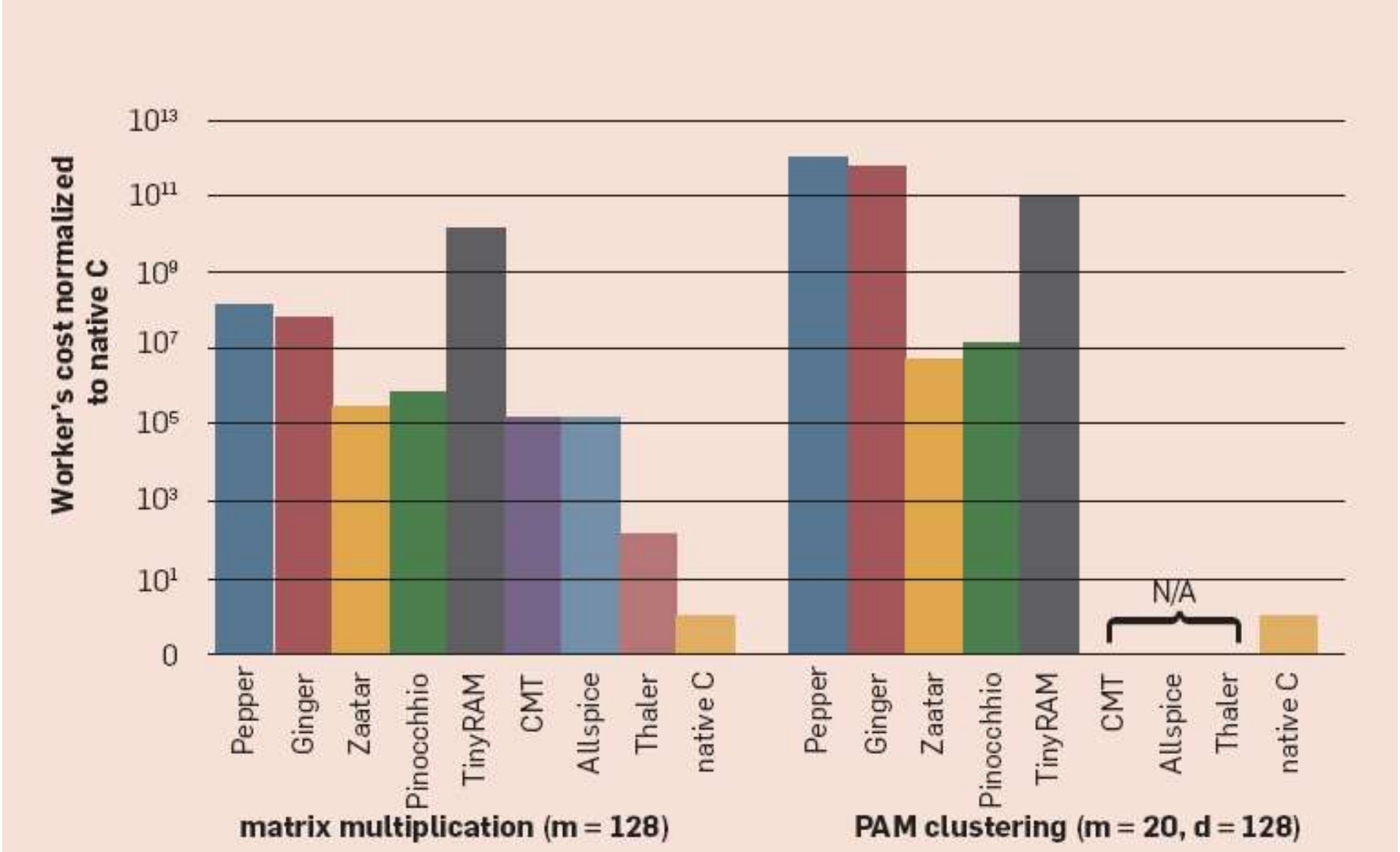
A framework for generic verifications



Framework in which a verifier can check that, for a computation *p* and desired input *x*, the prover's purported output *y* is correct. Step 1: The verifier and prover compile *p*, which is expressed in a high-level language (for example, C) into a Boolean circuit, $\mathcal{C}$. Step 2: the prover executes the computation, obtaining a transcript for the execution of $\mathcal{C}$ on *x*. Step 3: the prover encodes the transcript, to make it suitable for efficient querying by the verifier. Step 4: the verifier probabilistically queries the encoded transcript; the structure of this step varies among the protocols (for example, in some of the works,[7,36] explicit queries are established before the protocol begins, and this step requires sending only the prover's responses).

[Walfish-Blumberg CACM2015]

A framework for generic verifications



[Walfish-Blumberg CACM2015]

Interactive protocol for problems in NC

[Goldwasser, Kalai, Rothblum 2008]

- Construction based on Prob. Checkable Proofs (PCP)
- log-space uniform Boolean circuits $\mathcal{C}_N$ with N inputs
  - Prover
    - Compresses levels of the evaluated circuit by a linear form
    - Complexity: size($\mathcal{C}_N$)$^{\mathcal{O}(1)}$ (sometimes $\mathcal{O}$(size($\mathcal{C}_N$)) *[Thaler 2012]*)
  - Verifier
    - performs a single Boolean zero-sum check on the levels
    - Complexity: (N+depth($\mathcal{C}_N$))·log(N+size($\mathcal{C}_N$))$^{\mathcal{O}(1)}$
- Our ad-hoc certificates are instead
  - Independent of the computation ⇨ expose bugs in $\mathcal{C}_N$
  - Optimal prover complexity: best(N) + ∘(best(N))
  - Essentially optimal verifier complexity: $N^{1+\circ(1)}$

Public/Private verifiability

- Private verifiability
  - Client only has to be convinced
  - ⇨ Through the conversation
- Public verifiability
  - Publication of the conversation is not sufficient
  - 💣 Server and Client could be in cahoots
  - ✚ Must convince also external, independent, a posteriori, verifiers
- In some cases, automatic transform private → public
  - *[Fiat-Shamir 1986]*
  - △ Requires cryptographic hardness assumptions

Public verifiability: Sparse matrix GL7d19

- *[Elbaz-Vincent, Gangl, Soulé 2005]*
  - K-theory conjectures ⇦ ranks of boundary matrices
- GL7d19: $1\,911\,130 \times 1\,955\,309$ matrix
  - 1050 CPU days: rank is $1\,033\,568$

👍 Computed once in 2010 with LinBox …

😐 With a Monte-Carlo randomized algorithm …

⇨ … do you believe that this rank is correct?

⇨ We construct an easily checkable certificate (public verifiability)

Verification of linear system solving (LinSys)

- Publicly & deterministically verifiable Victor ask for the solution to A . ? = b
  - Peggy answers with the vector x
  - Anybody can check whether Ax =?= b
- Computation costs $\mathcal{O}(n^3)$ (or $\mathcal{O}(n^\omega)$, with *[…LeGall'14]*)
- Communication is $\mathcal{O}(n)$
- Verification costs $\mathcal{O}(n^2)$

Probabilistic verification

[Zippel-Schwarz 1979]

- 2 polynomials f, g with $d°(f) \leq d°(g) \leq n$
  - Check equality of f and g?
  - (g-f) has at most n roots
  - Randomly select $\lambda \in \mathbb{S}$
  - If $g \neq f$ then $\mathcal{P}(g(\lambda)-f(\lambda) = 0) < 1-n/|\mathbb{S}|$

[Freivalds 1979]

- 3 matrices A, B, C of dimensions $m \times k$, $k \times n$, $m \times n$
  - Check equality of AB and C?
  - Randomly select $v \in \mathbb{F}^n$
  - If $AB \neq C$ then $\mathcal{P}(A(Bv)-Cv = 0) < 1-1/|\mathbb{F}|$

Verifiability in practice?

4096x4096	MatMul [Thaler 2012]	MatMul [FFlas-FFpack]	LinSys [FFlas-Ffpack]
Server time	364.61s	5.01s	4.08s
+certificate overhead	0.49s	0.00s	0.00s
Client time	9.86s	[Freivalds] 0.05s	[Freivalds] 0.02s

- Goldwasser et al.: linear time verifiers do exist
  - ☺ Faster generic approach to date …
  - ☹ Prover/Verifier time prohibitive, even with model restrictions
- Ad-hoc approach:
  - 👍 Reduce to MatMul/LinSolve …

Contents

- Outsourcing
- Verifiable computing
- Certificates for Dense Matrices
  - RANK
  - Reductions
  - Hilbert, Artin, Global optimization
  - CHARPOLY$_{\mathbb{Z}}$
- Certificates for Sparse Matrices

Certifying the rank of dense matrices over a field

- *à la [Rūsiņš Freivalds 1979]*
  - Prover: exhibits P, L, U, Q
    - complexity $2/3\, n^3$ (or $\mathcal{O}(n^\omega)$)
  - Verifier: Probabilistic check that A == PLUQ
    - Check permutation and triangular matrices
    - Check rank of U in linear time
    - Random projection vector v
      - check $A \cdot v - P \cdot (L \cdot (U \cdot (Q \cdot v))) == 0$

☺ Overall Verifier Monte-Carlo complexity: $\mathcal{O}(n^2)$

Non-singularity certificate of dense matrices over $\mathbb{Z}$

- *à la [Rūsiņš Freivalds 1979]*
  - Prover
    - Exhibits P, L, U, Q ; all invertible
    - Exhibits smallish prime p
  - Verifier
    - Random vector v
      - checks $A \cdot v - P \cdot (L \cdot (U \cdot (Q \cdot v))) \equiv 0 \bmod p$

☺ Overall verifier Monte-Carlo bit complexity $n^{2+\circ(1)}$

△ Rank of singular matrix?

△ Prime p is chosen by Peggy,

△ Victor does not know whether p preserves the rank or not …

Interactive
RANK certificate of dense matrices over $\mathbb{Z}$

1. Verifier
   - Randomly chooses smallish prime p
2. Prover
   - Exhibits P, L, U, Q s.t. rank(A)=rank(U) mod p
3. Verifier
   - Random v and $A \cdot v - P \cdot (L \cdot (U \cdot (Q \cdot v))) \equiv 0 \bmod p$

☺ Prover cannot choose a bad prime and time is optimal
☺ Verifier time is essentially optimal (better constant factor)

△ **Certificate is not checkable a posteriori anymore**

Bit complexity	**Prover**	**Communications**	**Verifier**
RANK, DET	Best known $n^{\omega+\circ(1)}$	$n^{2+\circ(1)}$	$n^{2+\circ(1)}$

Fiat-Shamir derandomization (random oracle model)
RANK certificate of dense matrices over $\mathbb{Z}$

1. Prover
   - Computes p=NextPrime(CryptographicHASH(A))
   - Exhibits P, L, U, Q s.t. rank(A)=rank(U) mod p
2. Verifier
   - Checks p=NextPrime(CryptographicHASH(A))
   - Random v and $A \cdot v - P \cdot (L \cdot (U \cdot (Q \cdot v))) \equiv 0 \bmod p$

☺ **Certificate is now checkable a posteriori**

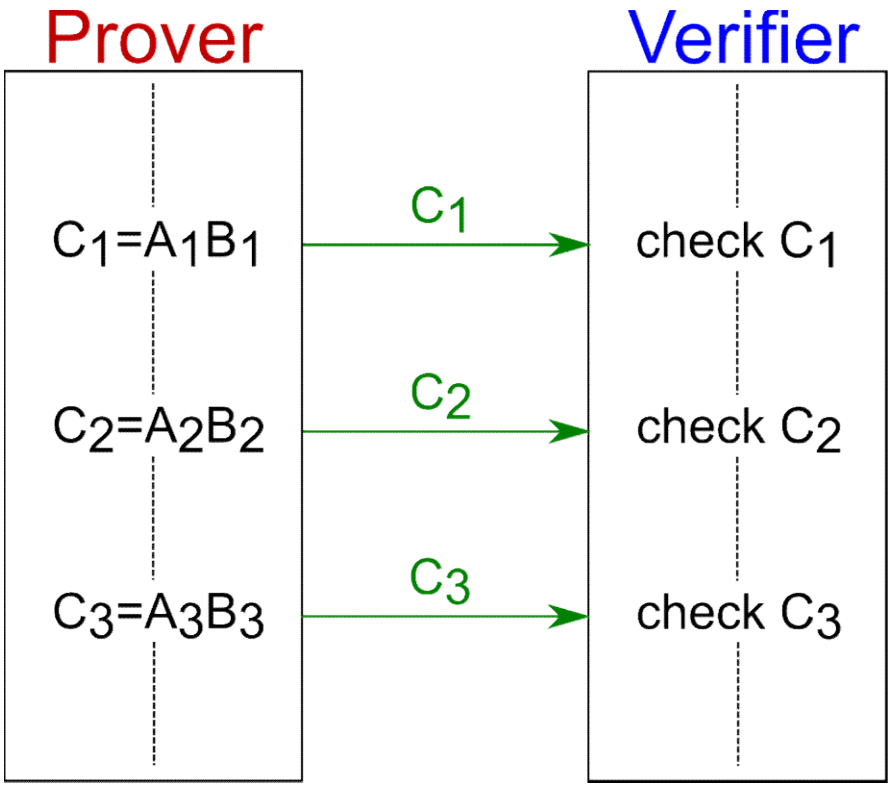
Bit complexity	Prover	Communications	Verifier
RANK, DET	Best known $n^{\omega+\circ(1)}$	$n^{2+\circ(1)}$	$n^{2+\circ(1)}$

Ad-hoc certificates

- *"Mathematics is the art of reducing any problem to linear algebra". ---* William Stein

- *[Kaltofen, Nehrig, Saunders 2011]*
  - Reductions to MATMUL
  - Prover
    - Sends all intermediate MATMUL
  - Verifier reruns algorithm
    - [Freivalds] check of intermediate MATMUL

👍 Like Verifier has an n^2 MATMUL



Prover	Communications	Verifier
$\mathcal{O}(n^\omega)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$

Artin's solution to Hilbert 17th Problem

- Exact certification of global optimality
  - Prove: polynomial inequality $\forall \xi_1,\ldots,\xi_n \quad f(\xi_1,\ldots,\xi_n) \geq g(\xi_1,\ldots,\xi_n)$?
  - via SOS: $\exists u_i, v_j \in \mathbb{R}[x_1,\ldots,x_n]$, $\quad (f-g) = (\sum_{i=1}^{k} u_i^2)/(\sum_{j=1}^{m} v_j^2)$
  - $\exists \mu, \nu \in \mathbb{R}[x_1,\ldots,x_n]$,

$$(f-g) \cdot (\mu(x_1,\ldots,x_n)^T \mathbf{W_2}\ \mu(x_1,\ldots,x_n)) = (\nu(x_1,\ldots,x_n)^T \mathbf{W_1}\ \nu(x_1,\ldots,x_n))$$

  - $\mathbf{W_1}, \mathbf{W_2} \neq 0 \in \mathbb{SZ}^{n \times n}$ symmetric positive semi-definite
  - Entries in vectors μ, ν in are precisely terms occurring in u_i, v_j

- Verifier
  - Checks Descartes' rule of sign on certified CHARPOLYS of $\mathbf{W_1}, \mathbf{W_2}$
  - Checks remultiplication of u_i, v_j is (f-g)

CHARPOLY?

- *[Kaltofen-Villard'04]* integer characteristic polynomial
  - Best bit complexity bound exponent: $\omega+(1-\zeta)/(\omega^2-(2+\zeta)\omega+2)$
  - $\omega=2.373, \zeta=0.303$: CHARPOLY exponent is 2.695
  - $\omega=3$, $\zeta=0$: CHARPOLY exponent is 3.2
  - $\omega=2$, $\zeta=0$: CHARPOLY exponent is 2.5

⇨ *[KNS'11]* CHARPOLY certificate verification in $n^{2.5+\circ(1)}$

Bit complexity	**Prover**	**Communications**	**Verifier**
CHARPOLY *[KV]*	$n^{\omega+(1-\zeta)/(...)}$	$n^{2.5+\circ(1)}$	$n^{2.5+\circ(1)}$
CHARPOLY *[KNS]*	$n^{\omega+1+\circ(1)}$	$n^{3+\circ(1)}$	$n^{2+\circ(1)}$

Reducing CHARPOLY verifier to interactive DET verifier

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

[D., Kaltofen 2014]

Bit complexity	Prover	Communications	Verifier
CHARPOLY	$n^{\omega+(1-\zeta)/(...)}$	$n^{2+\circ(1)}$	$n^{2+\circ(1)}$

Derandomized CHARPOLY verifier reduced to DET verifier

Peggy	$A \in \mathbb{Z}^{n \times n}$	*Victor*
$g \in \mathbb{Z}[X] = charpoly_A$	$\xrightarrow{g(X)}$	
$\lambda = HASH(A, g)$	$\xrightarrow{\lambda}$	
$\delta \in \mathbb{Z} = det(\lambda I - A)$	$\xrightarrow{\delta}$	
$C_{Det} : Cert(\delta = det(\lambda I - A))$	$\xrightarrow{C_{Det}}$	$degree(g) \stackrel{?}{=} n$
		$\lambda \stackrel{?}{=} HASH(A, g)$
		$\delta \stackrel{?}{=} det(\lambda I - A)$
		$\delta \stackrel{?}{=} g(\lambda)$

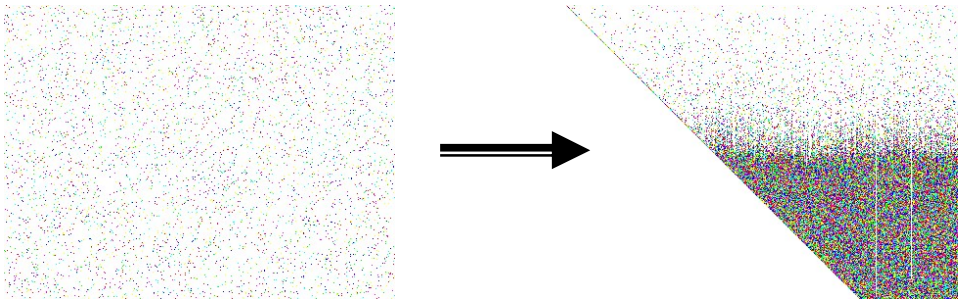
Bit complexity	Prover	Communications	Verifier
CHARPOLY	$n^{\omega+(1-\zeta)/(...)}$	$n^{2+\circ(1)}$	$n^{2+\circ(1)}$

Contents

- Outsourcing
- Verifiable computing
- Certificates for Dense Matrices
- Certificates for Sparse Matrices
  - RANK
  - DETERMINANT
  - MINPOLY, CHARPOLY, ...

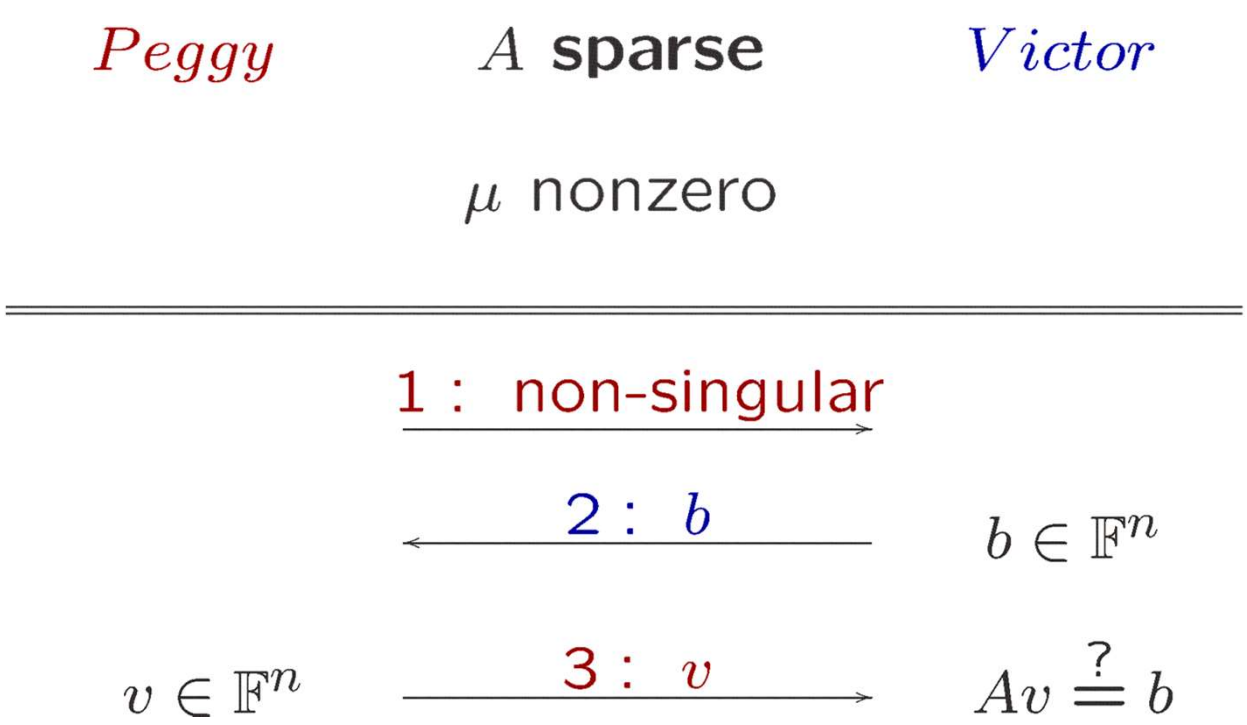
Sparse Matrices

- Matrix factorization are not viable anymore
  - Ex: P,L,U,Q



- Instead, matrix-vector product only is allowed
  - Blackbox model: 1 m-v costs μ operations

Linear certificate for non-singularity of sparse matrices over $\mathbb{F}$



- Soundness: suppose A is singular, then

	Prover	Communications	Verifier
SPARSE NONSING.	Best known	μ+2n	μ+n

Breaking random oracle and integer factorization

- Fiat-Shamir heuristic with public hash function
  - Prover
    - Compute b = Hash(A) = Blum-Blum-Shub$_N$(A)
    - Solve Av=b, return v
  - Verifier
    - Compute b = Hash(A) = Blum-Blum-Shub$_N$(A)
    - Check Av =?= b

- If the matrix is singular, to break certificate
  - Prover need to predict first entry of P^{-1} b is 0
  - She can thus predict bits of b=Blum-Blum-Shub$_N$(A)
  - She can thus factor N …

Interactive certified upper bound to the rank

	Peggy		*Victor*
Input		$A \in \mathbb{F}^{m\times n}$	
Commitment	$rank(A) \leq r < \min\{m,n\}$	$\xrightarrow{1:\ r}$	
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	$w \in \mathbb{F}^{r+1} \neq 0$	$\xrightarrow{3:\ w}$	$w \overset{?}{\neq} 0$
			$\left[I_{r+1}\middle\vert 0\right] UAV \begin{bmatrix} I_{r+1} \\ 0 \end{bmatrix} w \overset{?}{=} 0$

- Precondition the matrix
  - U, V structured and fast to apply
  - then UAV has generic rank profile …
  - … and (r+1)×(r+1) zero principal minor

Essentially optimal interactive certificate for the rank of sparse matrices

- Prover

  Input: A and U,V s.t. UAV is generic rank profile

  1. Certificate: Non-singularity of leading $(U\ A\ V)_{r\times r}$
     ⇨Solve this r×r system with any right-hand side
  2. Certificate: singularity of leading $(U\ A\ V)_{(r+1)\times(r+1)}$
     ⇨Produce a (r+1) non-zero vector in the nullspace

- Verifier
  - 2 matrix-vector products 2μ
  - 2 products with structured U, V $n^{1+\circ(1)}$
  - 2 vector equality tests $2n$

[D., Kaltofen 2014]

If $\mu=n^{1+\circ(1)}$	**Prover**	**Communications**	**Verifier**
SPARSE RANK	$\mathbf{n^{2+\circ(1)}}$	**5n**	$\mathbf{n^{1+\circ(1)}}$

Extension to SPARSE RANK over $\mathbb{Z}$

Peggy $A \in \mathbb{Z}^{m \times n}$ *Victor*

$r = rank(A)$

$\xrightarrow{1 : \ r}$

$\xleftarrow{2 : \ p, b, U, V}$

p prime $\mathcal{O}\left((m+n)^{1+\circ(1)}\right)$

$U \in \mathbb{B}_S^{m \times m}, V \in \mathbb{B}_S^{n \times n}, b \in \mathbb{Z}/p\mathbb{Z}^r$

rank certif. mod p $\xrightarrow{3 : \ v, w}$ NonSing & UpperRank mod p

$Ax \in \mathbb{Z}/p\mathbb{Z}$ via $(Ax \in \mathbb{Z})$ mod p

If $\mu = n^{1+\circ(1)}$	Prover	Communications	Verifier
$\mathbb{Z}$ SPARSE RANK	$n^{2+\circ(1)}$	$n^{1+\circ(1)} \cdot \log\|\|A\|\|^{1+\circ(1)}$	$n^{1+\circ(1)} \cdot \log\|\|A\|\|^{1+\circ(1)}$

Direct SPARSE DETERMINANT certificate

Prover	*Communications*	*Verifier*
$B = A\,\Gamma(t,s)$	$\xrightarrow{t,s}$	Checks $t^n + s \neq 0$
$c^B(\lambda) = \det(\lambda I_n - B)$	$\xrightarrow{c^B(\lambda)}$	
$C = [b_{i,j}]_{1 \leq i,j \leq n-1}$ $c^C(\lambda) = \det(\lambda I_{n-1} - C)$	$\xrightarrow{c^C(\lambda)}$	
$h^B, h^C \in \mathbb{F}[\lambda]$: $h^B c^B + h^C c^C = 1$ $t, s \in \mathbb{F}$ also s.t. $\mathrm{GCD}(c^B(\lambda), c^C(\lambda)) = 1$	$\xrightarrow{h^B, h^C}$	Checks $\mathrm{GCD}(c^B(\lambda), c^C(\lambda)) = 1$ in $\mathbb{F}[\lambda]$ by checking at α $h^B(\alpha)c^B(\alpha) + h^C(\alpha)c^C(\alpha) = 1$
Computes u such that	$\xleftarrow{r}$	$r \in S \subseteq \mathbb{F}$ a random element
$(rI_n - B)u = e^{[n]} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$	$\xrightarrow{u}$	Checks $(rI_n - B)u = e^{[n]}$ $u_n c^B(r) = c^C(r)$ Computes $\det(A) = c^B(0)/(t^n + s)$

	Prover	Communications	Verifier
SPARSE DET	**3W(n)**	μ+**6n**	μ+**19n**

Family of certificates for SPARSE MINPOLY

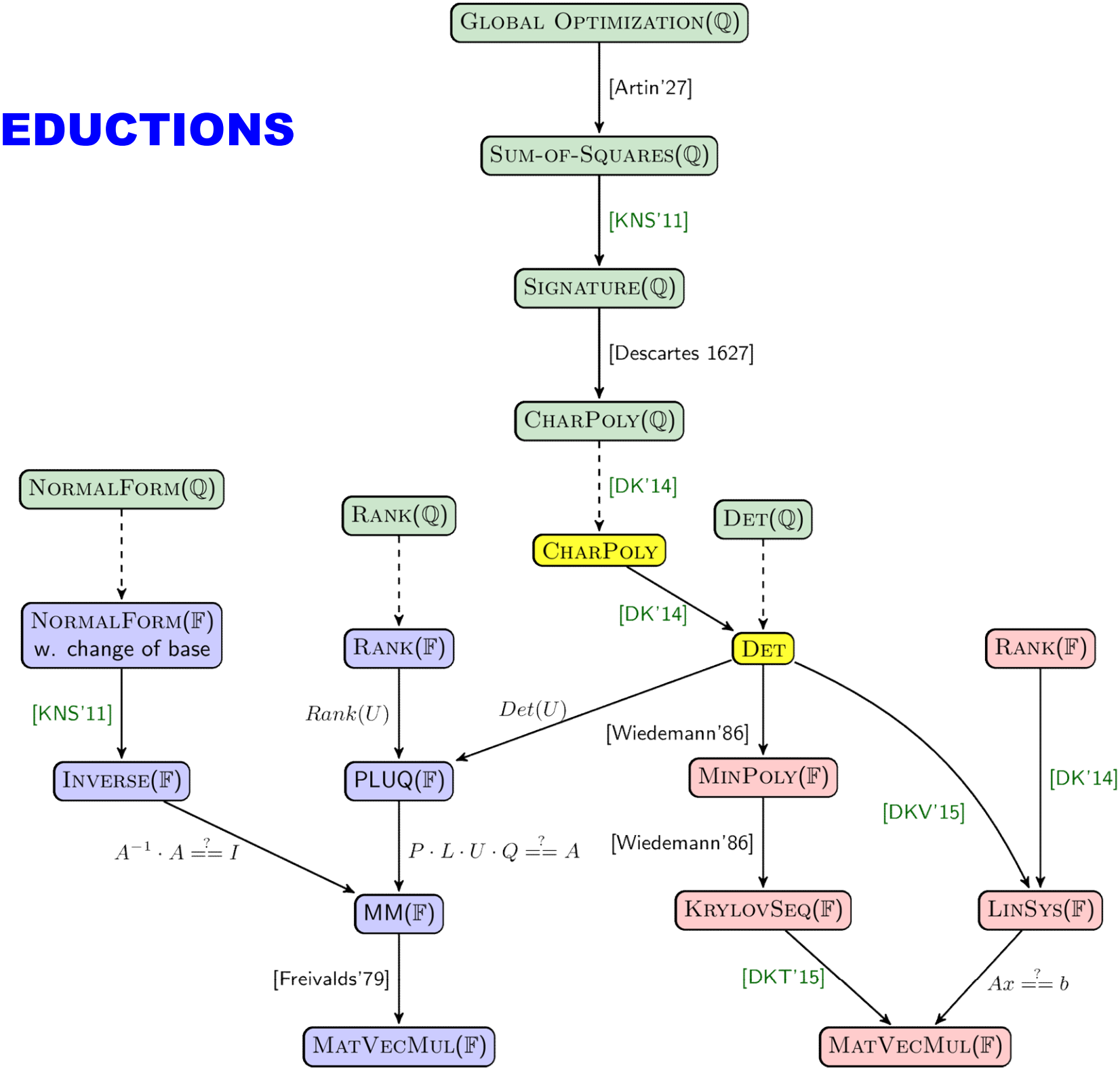
[D., Kaltofen, Thomé, Villard 2015]

Prover	Communications	Verifier
W(n)=$\mathcal{O}(\mu n)$	$\mathcal{O}(n\sqrt{\mu})$	$\mathcal{O}(n\sqrt{\mu})$
W(n)+$\mathcal{O}(\mu\sqrt{n})$	$\mathcal{O}(n\sqrt{n})$	$2\mu + \mathcal{O}(n\sqrt{n})$
W(n)+$\mathcal{O}(\mu n^{2/3})$	$\mathcal{O}(n^{1+1/3})$	$4\mu + \mathcal{O}(n^{1+1/3})$
W(n)+$o(W(n))$	$\mathcal{O}(n^{1+1/\ell})$	$2^{\ell}\mu + \mathcal{O}(n^{1+1/\ell})$
2W(n)	$\mathcal{O}(n)$	$\mu + \mathcal{O}(n)$

Contents

- Outsourcing
- Verifiable computing
- Certificates for Dense Matrices
- Certificates for Sparse Matrices
- Conclusion

Reductions


Global Optimization(ℚ)
[Artin'27]
Sum-of-Squares(ℚ)
[KNS'11]
Signature(ℚ)
[Descartes 1627]
CharPoly(ℚ)
[DK'14]
NormalForm(ℚ)
Rank(ℚ)
CharPoly
Det(ℚ)
[DK'14]
NormalForm(𝔽) w. change of base
Rank(𝔽)
Det
Rank(𝔽)
[KNS'11]
Rank(U)
Det(U)
[Wiedemann'86]
[DK'14]
Inverse(𝔽)
PLUQ(𝔽)
MinPoly(𝔽)
[DKV'15]
A^{-1} · A =? I
P · L · U · Q =? A
[Wiedemann'86]
MM(𝔽)
KrylovSeq(𝔽)
LinSys(𝔽)
[Freivalds'79]
[DKT'15]
Ax =? b
MatVecMul(𝔽)
MatVecMul(𝔽)

Open problems

- Sparse normal forms
  - Linear time SPARSE verifier for SMITHFORM?
  - ⇨ normal form certificates in the sparse case?
  - ⇨ do not compute change of base matrices …
- Remove cryptographic computational hardness assumption
  - For now, only $n^{1.5+o(1)}$ SPARSE DET verifier

References

- **To cloud or not to cloud?: musings on costs and viability**. Yao Chen, Radu Sion. *2nd ACM Symposium on Cloud Computing. No. 29, 2011. DOI>10.1145/2038916.2038945*

- **Time-Optimal Interactive Proofs for Circuit Evaluation.** JustinThaler. *Advances in Cryptology (CRYPTO'13).* 71--89. *http://people.seas.harvard.edu/~jthaler/ThalerCrypto.pdf*

- **Essentially optimal interactive certificates in linear algebra.** Jean-Guillaume Dumas, Erich Kaltofen. *39th International Symposium on Symbolic and Algebraic Computation, pages 146-153, 2014. DOI>10.1145/2608628.2608644*

- **Verifying Computations without Reexecuting Them.** Michael Walfish, Andrew J. Blumberg. *Communications of the ACM, Vol. 58 No. 2, pages 74-84, 2015. DOI>10.1145/2641562.*

- **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**. Jean-Guillaume Dumas, Erich Kaltofen, Emmanuel Thomé. *2015, arXiv: cs.SC/1507.01083.*