List-Decoding of Linear Functions and Analysis of a 2-Round Zero Knowledge Argument

Cynthia Dwork, Microsoft

Ronen Shaltiel, Weizmann

Adam Smith, MIT

Luca Trevisan, Berkeley

De-randomization and 2-Round Zero Knowledge

Cynthia Dwork, Microsoft

Ronen Shaltiel, Weizmann

Adam Smith, MIT

Luca Trevisan, Berkeley

This paper

- Dwork-Stockmeyer: 2-round ZK in non-standard model
- This paper: understand hardness assumption
  - Weaker assumptions (worst-case hardness)
  - Uniform protocols
  - Simpler proofs
- Main tools:
  - List-decoding results for code of all **linear** functions $\{0,1\}^k \to \{0,1\}^k$
- New use of de-randomization in cryptography […,Lu’02,…,BOV’03,…]

Zero-Knowledge Arguments [GMR,BCC]

- Interactively prove a statement without leaking any extra information
- Extensively studied
- Building block for other protocols
- **Round complexity**: number of messages

Standard Computational Model

Honest Prover & Verifier are PPT (prob poly-time)

Cheating Prover & Verifier need super-poly time

- 4 rounds… possible [FS]
- 3 rounds… open ?
- 2 rounds… impossible [GO]

Dwork-Stockmeyer: 2-round ZK

- Different model (following [DN,DNS,…]):
  - fixed polynomial bound on **prover's** resources (space,time)
  - Verifier & simulator are PPT
- 2 round argument for NP:

$$\mathbf{P} \xleftarrow{\text{challenge}} \mathbf{V}$$
$$\mathbf{P} \xrightarrow{\text{response}} \mathbf{V}$$

- Example: D.S. protocol with linear functions:
  - Honest prover needs $O^*(k)$ space and time
  - Cheating prover needs k^2 **space at runtime**
- Tradeoff: physical understanding vs. efficiency

De-randomization and 2-round ZK

- Ronen S: "There must be an extractor there."
- Average-case hardness via list-decoding
  - Better reductions
  - Uniform protocols
  - Simpler proofs
- New facts about linear functions

Outline

- Basic idea behind DS protocol
- Our Goal:
  - linear functions hard for resources < k^2
- List-Decoding Functions
- Combinatorial result: advice-bounded provers
- Complexity-theoretic result: small circuit provers

Dwork-Stockmeyer: 2-round ZK

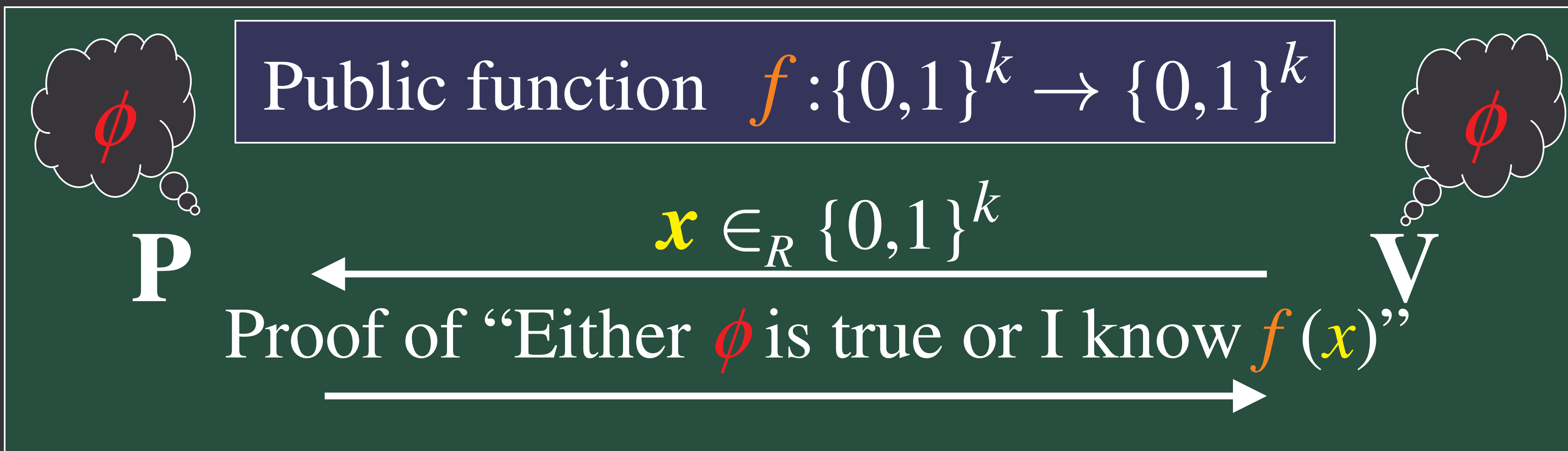
Public function $f:\{0,1\}^k \to \{0,1\}^k$

$$\mathbf{P} \xleftarrow{\text{challenge}} \mathbf{V}$$

$$\mathbf{P} \xrightarrow{\text{response}} \mathbf{V}$$

Limited resources

Dwork-Stockmeyer: 2-round ZK



- Proof takes only $O^*(k)$ bits
- Cheating prover must compute $f(x)$ on the fly
- Soundness $\Leftrightarrow$ f is hard on average for **P**
- Hardness not enough…

Proof efficiency

Public function $f:\{0,1\}^k \to \{0,1\}^k$

P $\xleftarrow{\quad x \in_R \{0,1\}^k \quad}$ **V**

Proof of "Either ϕ is true or I know $f(x)$" $\longrightarrow$

- For proof to be easy:
  - f is **linear**

$$\begin{pmatrix} f(x) \end{pmatrix} = \underset{k \times k}{\begin{pmatrix} \mathrm{M}_f \end{pmatrix}} \begin{pmatrix} x \end{pmatrix}$$

Our Goal: Hard Linear Functions

$$\begin{array}{c} \\ \begin{pmatrix} f(x) \end{pmatrix} \end{array} = \begin{array}{c} k \times k \\ \begin{pmatrix} \mathbf{M}_f \end{pmatrix} \end{array} \begin{array}{c} \\ \begin{pmatrix} x \end{pmatrix} \end{array}$$

- Hard for prover*: $\mathrm{Prob}_x[\ \mathbf{P}(x) = f(x)\ ] \leq \varepsilon$
- Always easy with k^2 space
  - We want hardness for $< k^2$ resources (e.g. $k^{3/2}$)
- Two models:
  - – Advice-bounded prover: cannot store all of M_f
  - – Time-bounded prover: circuit size $< k^2$

Results

- Advice-bounded provers
  - Random function hard for prover with advice < k^2 bits
  - Simpler proof of DS result
- Time-bounded provers
  - Security under worst-case hardness assumption
  - Assume: $\exists$ h $\in$ DTIME($2^{O(n)}$)
    worst-case hard for MAM-circuits of size $2^{n(\frac{1}{2} + \gamma)}$
  - Uniform protocol secure against prover with size $k^{1+2\gamma}$

Outline

- Basic idea behind DS protocol
- Our Goal:
  - linear functions hard for resources < k^2
- List-Decoding Functions
- Combinatorial result: advice-bounded provers
- Complexity-theoretic result: small circuit provers

Linear Functions as Codewords

	Coding space	Distance
Usual notion	Strings Σ^N	Hamming
De-randomization	Functions $\{0,1\}^k \to \{0,1\}^k$	$\text{Prob}_x[f(x) \neq g(x)]$

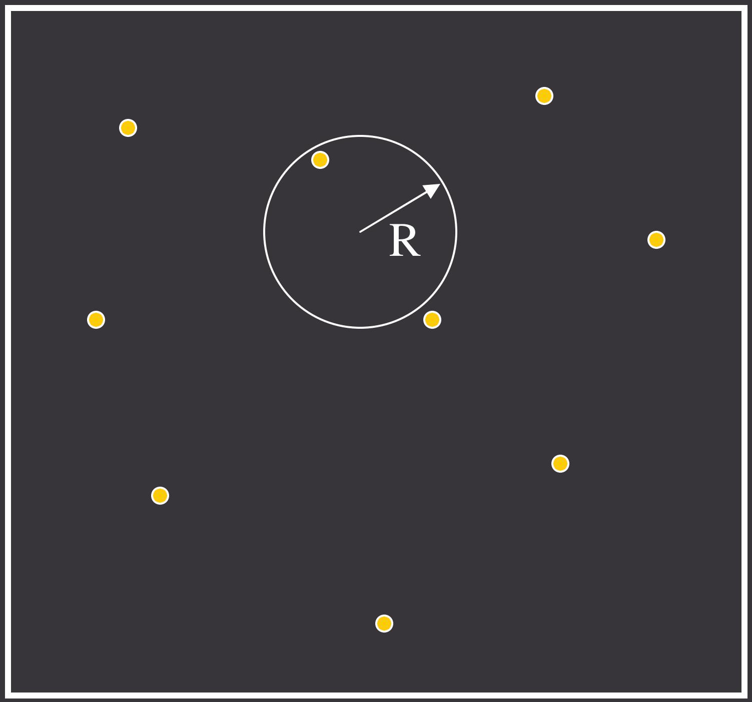
- Conceptually different
- Technically identical

$g \to \quad (\, g(0\ldots00),\ g(0\ldots01),\ g(0\ldots10),\ \ldots,\ g(1\ldots11)\,)$

vector with entries in $\Sigma = \{0,1\}^k$

List-Decodable Codes

- Codewords are functions $\{0,1\}^k \rightarrow \{0,1\}^k$
- Distance(f,g) = $\Pr_x[f(x) \neq g(x)]$

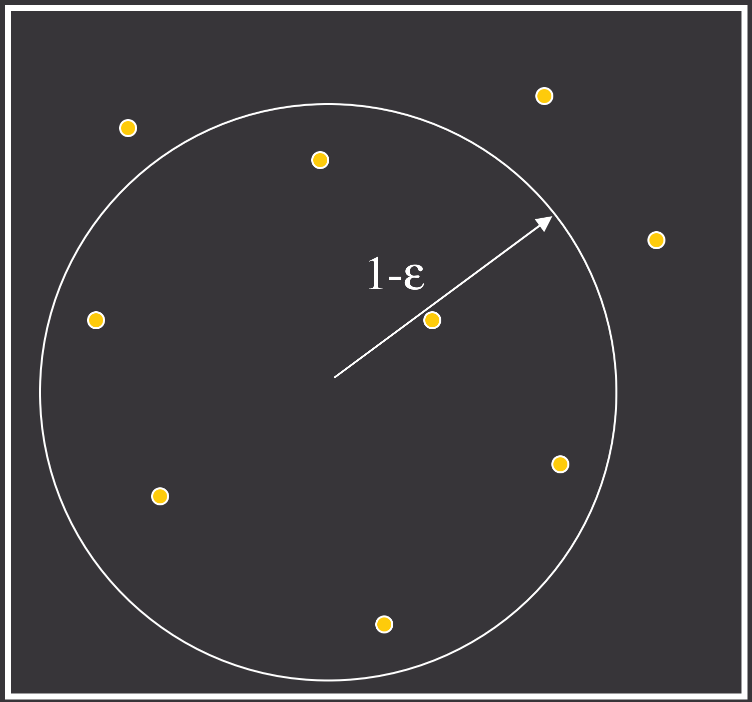


Error-Correcting Code:

Every ball of radius R contains at most one point

List-Decodable Codes

- Codewords are functions $\{0,1\}^k \rightarrow \{0,1\}^k$
- Distance(f,g) = $\Pr_x [f(x) \neq g(x)]$



Error-Correcting Code:
Every ball of radius R contains at most one point

List-Decodable Code:
Every ball of radius 1- ε contains at most $t(\varepsilon)$ points

Why List Decodability?

- Fix $g:\{0,1\}^k \to \{0,1\}^k$

Q: How many $k \times k$ matrices **M** such that

$$\Pr_x[\ g(x) = \mathbf{M}.x\ ] \geq \varepsilon\ ?$$

A: $(1/\varepsilon)^{2k}$ = small polynomial number of matrices

- Fix prover **P** who wants to cheat

Q: How many functions f such that

P can cheat w. prob. $\geq \varepsilon$?

- Same question! (almost… **P** can be randomized)

Advice-Bounded Provers

List-decodable codes give incompressible functions

- Suppose that prover's advice is at most $A < k^2$ bits
  - As much pre-processing as desired
  - Only keeps A bits about f (e.g. smart card)
- How many f s.t. $\exists$ prover who cheats w. prob. $\geq \varepsilon$?
  - Each prover can cheat for $(1/\varepsilon)^{2k}$ linear functions*
  - Prover described by advice: 2^A possible provers
  - Describe any "cheatable" f using $A + 2\,k \log(1/\varepsilon)$ bits
  - As long as $A < k^2 - 2\,k \log(1/\varepsilon) - 100$ bits,
    Prob. that random function is "cheatable" at most 2^{-100}

Proving List-Decodability

- Fix $g:\{0,1\}^k \rightarrow \{0,1\}^k$

Q: How many $k \times k$ matrices **M** such that

$$\Pr_x[\ g(x) = \mathbf{M}.x\ ] \geq \varepsilon\ ?$$

A: $(1/\varepsilon)^{2k}$ = small polynomial number of matrices

- Usual proof technique (Johnson bound) fails
- Problem: min. distance of code is ½
  - (Flip one bit in a matrix)
- We want list-decoding radius 1-ε.

Proof* that list size is $(1/\varepsilon)^{2k+1}$

- Meshulam, Shpilka: $\exists$ subspace V of matrices s.t.

  $\forall M, M' \in V$, $\Pr_x[M.x \neq M'.x] \geq 1-\varepsilon^2$
- Dimension(V) = $k^2 - 2k\log(1/\varepsilon)$
- Apply Johnson bound to V:

  Ball of radius $1-\varepsilon$ contains $O(1/\varepsilon)$ elements of V
- V has $(1/\varepsilon)^{2k}$ cosets, each with min. distance $1-\varepsilon^2$
- Ball of radius $1-\varepsilon$ contains $1/\varepsilon$ from each coset
- Total number of functions is $(1/\varepsilon)^{2k+1}$

Advice-Bounded Provers

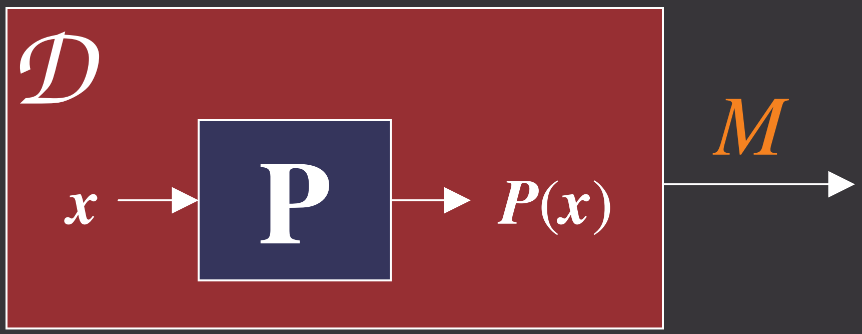
- Linear functions form a list-decodable code
- Random matrix is secure against advice-bounded provers
- Resulting protocol is non-uniform
  - Different matrix for every setting of k
  - No compact description of matrix
- Uniform protocol?
  - No! Advice-bounded prover has time to reconstruct the whole matrix

Outline

- Basic idea behind DS protocol
- Our Goal:
  - linear functions hard for resources < k^2
- List-Decoding Functions
- Combinatorial result: advice-bounded provers
- Complexity-theoretic result: small circuit provers

A Basic Decoder

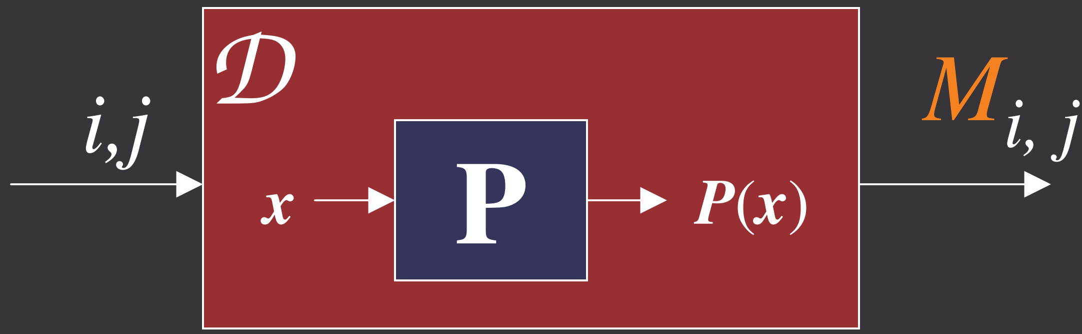
- Suppose $\Pr_x[\mathbf{P}(x) = M.x\ ] \geq \varepsilon$
- Then $\mathcal{D}^{\mathbf{P}}() = M$



time $(\mathcal{D}^{\mathbf{P}}) > k^2$

A Better Decoder: Output 1 bit

- Suppose $\Pr_x[\mathbf{P}(x) = M.x\ ] \geq \varepsilon$
- Then $\mathcal{D}^{\mathbf{P}}(i, j) = M_{i, j}$

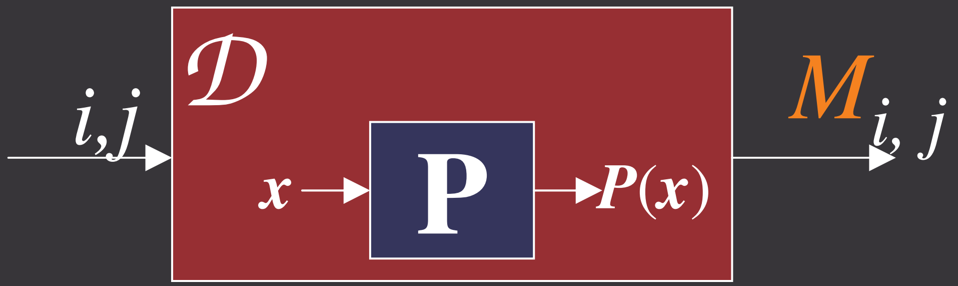


time ($\mathcal{D}^{\mathbf{P}}$) can be very low… $O($ time$(\mathbf{P})\ k^{\delta})$

Why does this help?

Hardness-Randomness Paradigm

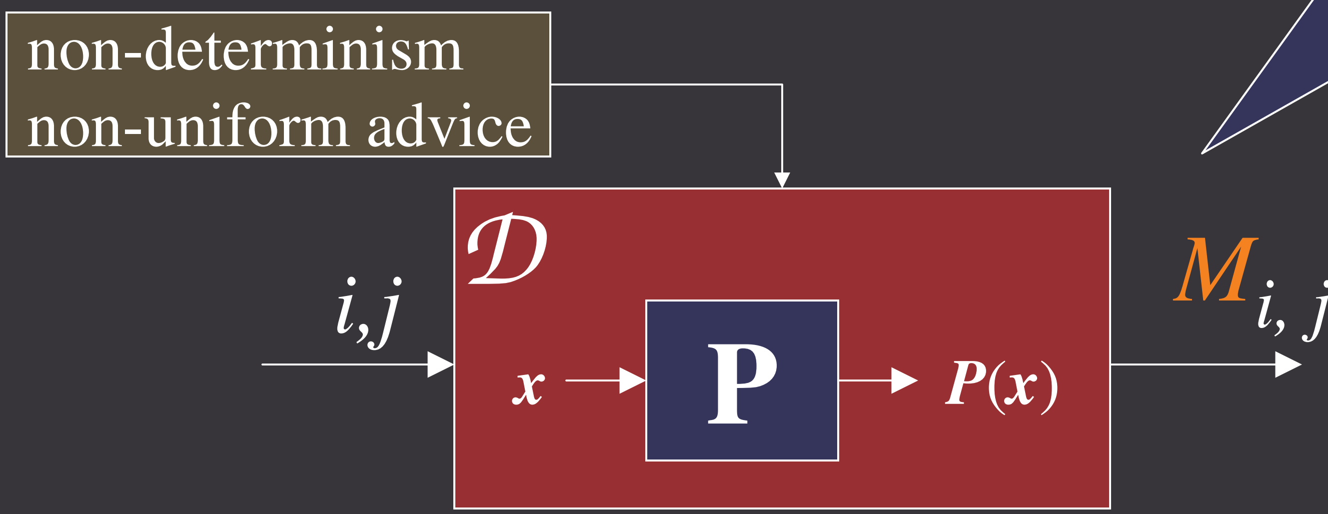
- Suppose $h: \{0,1\}^{2 \log k} \rightarrow \{0,1\}$
  is hard for circuits of size $k^{3/2}$ (note: k^2 is trivial)
- Use M = TT(h)
  TT(h) = (h(0….00), h(0…01), h(0…10), … , h(1…11))
- **P** cheats in time < $k^{3/2-\delta}$
  $\Rightarrow$ $\mathcal{D}$ computes $M_{i,j} = h(i, j)$ in time < $k^{3/2}$



Our Decoder: Uses Extra Help

- Suppose $\Pr_x[\mathbf{P}(x) = M.x\,] \geq \varepsilon$
- Then $\mathcal{D}^{\mathbf{P}}(i, j) = M_{i,\,j}$

Need to assume hardness for non-deterministic circuits (MAM)



time ($\mathcal{D}^{\mathbf{P}}$) can be very low… $O($ time$(\mathbf{P})\ k^{\delta})$

Results

- Connection to list-decoding (standard)
- Advice-bounded provers
  - Random function hard for prover with advice < k^2 bits
  - Simpler proof of DS result
- Time-bounded provers
  - Assume: $\exists$ h $\in$ DTIME($2^{O(n)}$)
    worst-case hard for MAM-circuits of size $2^{n(\frac{1}{2} + \gamma)}$
  - Uniform protocol secure against prover with size $k^{1+2\gamma}$

Conclusions

- Better understanding of DS model & protocol
- Open questions

1. Better decoding → nicer assumptions
2. Increase to arbitrary polynomial gap
   - Possible if one assumes completely malleable encryption
3. Other uses of de-randomization in crypto

Questions?