Interactive proofs and Cryptography

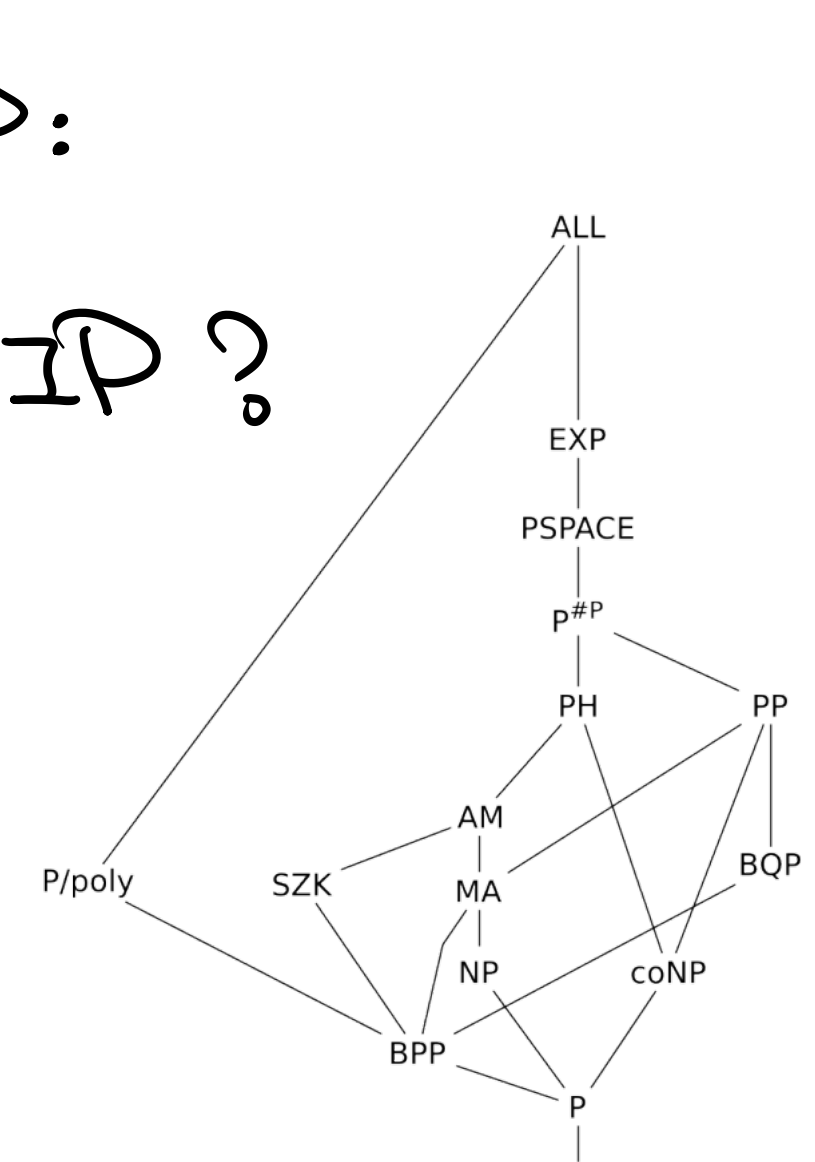
3/3/21 10:11 AM

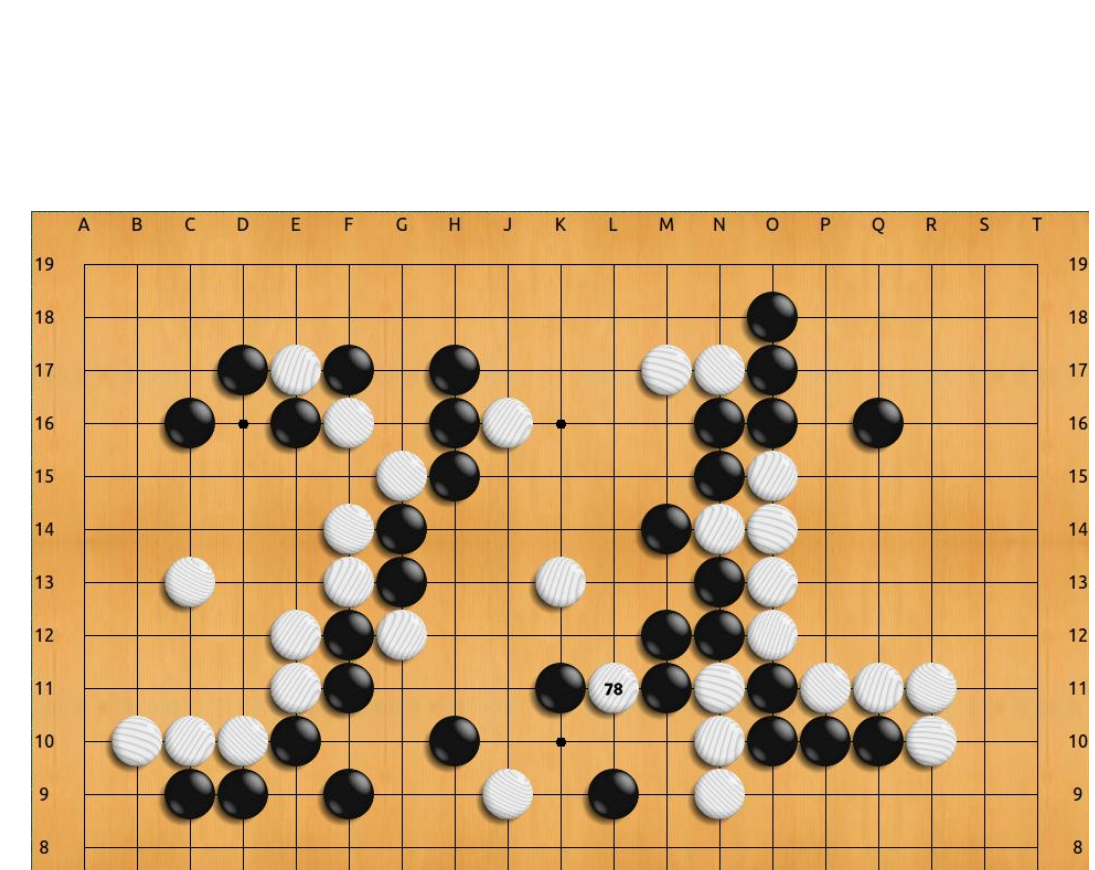
Last time, on IP:

Q. How strong is IP?

- $NP \subseteq IP$.
- $BPP \subseteq IP$.
- $IP \subseteq PSPACE$.

How to even prove $coNP \subseteq IP$?





How do we convince someone ϕ has no answer?

#SAT

- input: CNF ϕ and integer K.
- output: ϕ has exactly K sat. assignments.

Goal. Prove (to V) that ϕ has K answers.

Non-example 1.

Verifier ← $\phi(\alpha)$ $\forall \alpha \in \{0,1\}^n$ — Prover

2^n possible α.

Idea. Encode formulas as polynomials.

3CNF $\phi = \bigwedge C_j$ ⟷ polynomial $P_\phi = \prod P_j(x)$. (deg $3m$)

$C_j = (x_i \vee \overline{x_j} \vee x_k)$ ⟷ $P_j = 1 - (1-x_i)x_j(1-x_k)$ (deg 3.)

$P_\phi(a_1, \dots, a_n) = 1$ iff $\phi(a_1, \dots, a_n)$ sat.

$K_1(0)$: number, #sat. α to ϕ w/ $x_1 = 0$.

$$K_i(x_1, \dots, x_i) := \sum_{a_{i+1}, \dots, a_n \in \{0,1\}} P_\phi(x_1, \dots, x_i, a_{i+1}, \dots, a_n)$$ (deg $3m$)

#sat. assignments of ϕ w/ first i var. set to $x_1 \dots x_i$

$$K_0 = \sum_{a_1, \dots, a_n \in \{0,1\}^n} P_\phi(a)$$ #sat. assignments to ϕ.

$K_n(x_1, \dots, x_n) = P_\phi(x_1, \dots, x_n)$

$$K_i(x_1, \dots, x_i) = K_{i+1}(x_1, \dots, x_i, 0) + K_{i+1}(x_1, \dots, x_i, 1)$$

$K_1(0)$ = $K_2(0, 0)$ + $K_2(0, 1)$

Non-example 2.

A. $\forall i \in [n]$. $\tilde{K}_{i-1}(x_1, \dots, x_{i-1}) = \tilde{K}_i(x_1 \dots x_{i-1}, 0) + \tilde{K}_i(x_1, \dots, x_{i-1}, 1)$?

Verifier ← $\tilde{K}_0, \tilde{K}_1(x_1), \dots, \tilde{K}_n(x_1, \dots, x_n)$ (n+1 polynomials) — Prover

Verifier: $\phi(x)$, K

Problem: $\tilde{K}_i$: describe all coeff. deg $\leq 3m$ $\Rightarrow$ #terms $= (3m)^n \sim \binom{3m}{n}$

B. $\tilde{K}_n(x) = \phi(x)$? $\tilde{K}_0 = K$

Idea. Fingerprinting!

To check if $K_i(x_1, \dots, x_i) = \tilde{K}_i(x_1, \dots, x_i)$, sufficient to check

$(K_i - \tilde{K}_i)(a_1, \dots, a_{i-1}, x_i) \neq 0$ for random $a_j \in [1..p]$ (prime)

(choose $p \sim 2^n$ s.t. $K_0 = K_0 \bmod p$)

SZ lemma ⇒ error pr. $\leq 3m/2^n \ll 1/\text{poly}\,n$.

- $\forall i \in [n]$. $\tilde{K}_{i-1}(r_1, \dots, r_{i-1}) = \tilde{K}_i(r_1 \dots r_{i-1}, 0) + \tilde{K}_i(r_1, \dots, r_{i-1}, 1)$?
- $\tilde{K}_n(r) = \phi(r)$?

Verifier ← $\tilde{K}_0$, prime p — Prover
Verifier ← $\tilde{K}_1(x_1)$
Verifier → r_1
Verifier ← $\tilde{K}_2(r_1, x_2)$ ($3m+1$ coeff.)
Verifier → r_2
⋮
Verifier ← $\tilde{K}_n(r_1, \dots, r_n)$

Analysis. If Prover sent fake $\tilde{K}_i(r, x_i)$

$\tilde{K}_i \neq K_i$ w.p. $\geq 1 - 1/\text{poly}\,n$

Verifier caught cheating w.p. $\geq (1 - 1/\text{poly}\,n)^n$

Conclusion.

- To utilize randomness, encode problem as low-deg. polynomial.
- To check if $\#\phi = 0$, compute $\#\phi$ directly.
- interaction + randomness is powerful!

Cryptography.

Alice (plaintext x) — key — Bob

Ciphertext: $y = E_{key}(x)$ →

Eve

Bob: $x = D_{key}(y)$

- RSA.
- Rot 13
- Substitution cipher

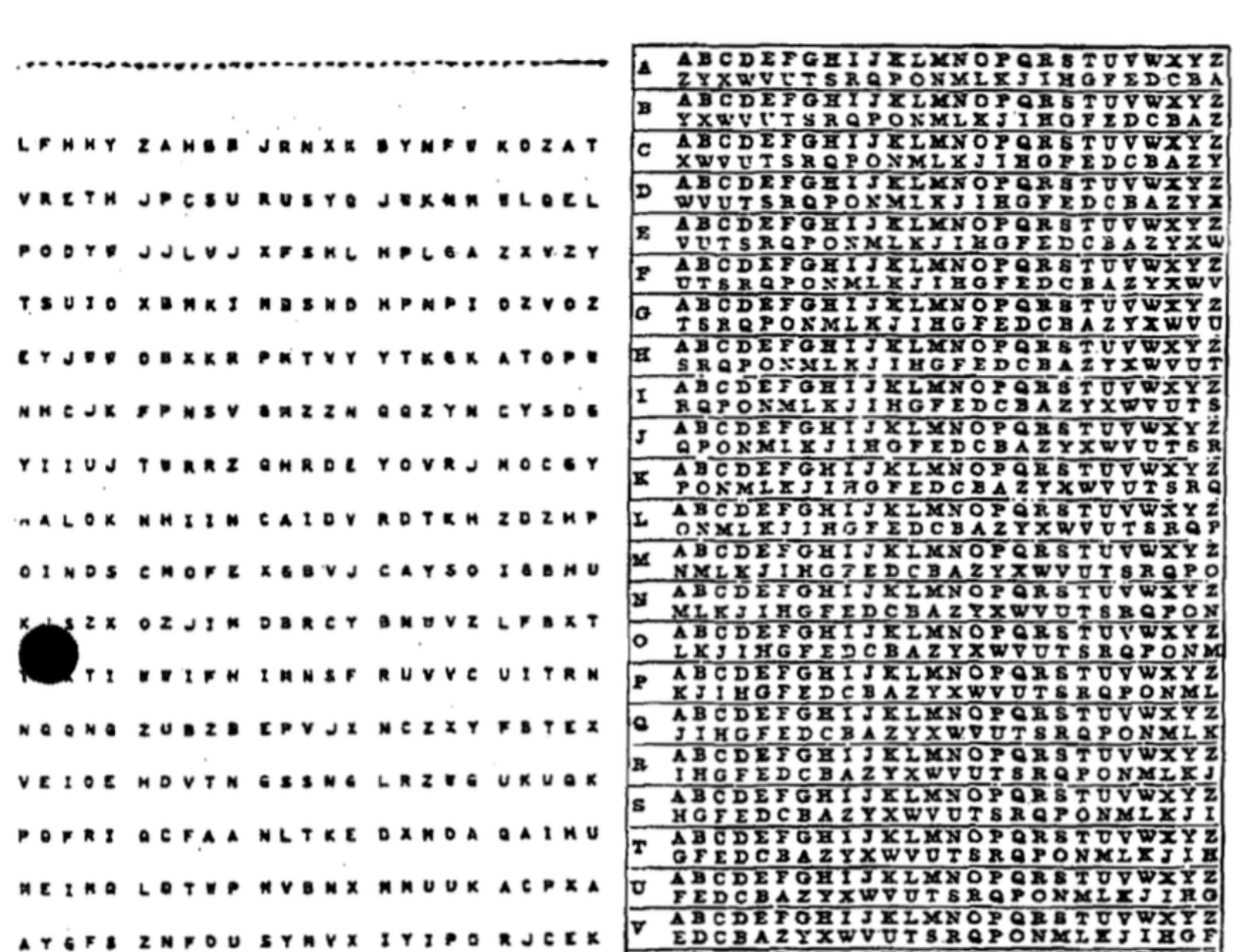
Jul qvq gur puvpxra pebff gur ebnq? (c c ncross)

53‡‡†305))6*;4826)4‡.)4‡);80
6*;48†8¶60))85;1‡(;:‡*8†83(88)
5*†;46(;88*96*?;8)*‡(;485);5*†
2:*‡(;4956*2(5*-4)8¶8*;40692
85);)6†8)4‡‡;1(‡9;48081;8:8‡1
;48†85;4)485†528806*81(‡9;48
;(88;4(‡?34;48)4‡;161;:188;‡?;

A good glass in the bishop's hostel in the devil's seat
forty-one degrees and thirteen minutes northeast and by north
main branch seventh limb east side
shoot from the left eye of the death's-head
a bee line from the tree through the shot fifty feet out.

Q. Safest crypto? Random string.

One-time Pad. [Miller 1882] [Vernam 1917]



DIANA by NSA.

E. German IA (HVA)

Perfectly safe! modulo

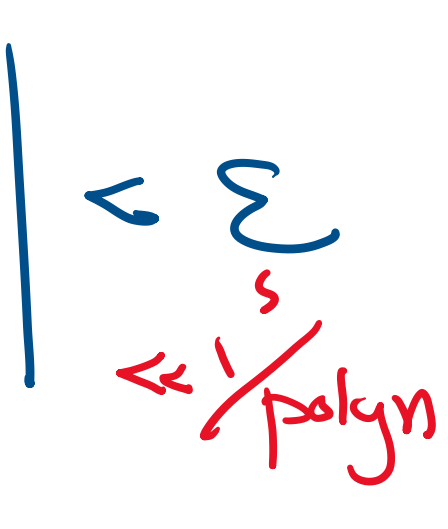
- Truely random
- One-time use
- Sharing pads/key

$A \oplus R \oplus B \oplus R = A \oplus B$

- Use crypto to play head-or-tail over mail!

Crypto-secure.

Pseudorandom generator



$G : \{0,1\}^\ell \to \{0,1\}^{S(\ell)}$ ~ poly ℓ

(1) stretching: s ℓ bits ⟶ $G(s)$ $S(\ell)$ bits.

(2) pseudorandom: $G(s)$ "looks" random to ANY "human" E

$$\left| \Pr_{s \sim U_\ell}[E(G(s)) \text{ accepts}] - \Pr_{r \sim U_{S(\ell)}}[E(r) \text{ accepts}] \right| < \varepsilon \ll 1/\text{poly}\,n$$

$\forall E : \{0,1\}^{S(\ell)} \to \{acc, rej\}$ ~~efficient~~ BPP

(3) efficient: G runs in $O(\text{poly}\ \ell)$ time. (~~$S(\ell)$~~)

Observation. Enough to stretch one bit! $s \to s+1$

Thm. $G(x, s) = h(x) \circ s \circ x \oplus s$ is secure PRG. [Goldreich-Levin '89]

if h is hard to invert. one-way fcn.

Commitment. [Naor '91]

Throw coin. Choose random seed s n-bits

Alice ← R poly n-bits — Bob

Alice → $m = PRG(s)$ if head; $m = PRG(s) \oplus R$ if tail → Bob

Alice ← guess g — Bob

Alice → reveal s → Bob

Alice: PRG, g Bob: PRG, $PRG(s) \oplus m$

- For Alice to cheat. need s' s.t. $PRG(s') = PRG(s) \oplus R$