Parallel Kac's Walk Generates PRU

Chuhan Lu*, Minglong Qin†, Fang Song*, Penghui Yao§‡, Mingnan Zhao§

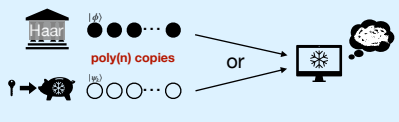
*Computer Science Department, Portland State University
†Centre for Quantum Technologies, National University of Singapore
§State Key Laboratory for Novel Software Technology, New Cornerstone Science Laboratory, Nanjing University
‡Hefei National Laboratory

1. Background

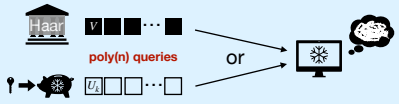
Quantum Pseudorandomness

[JLS17] defines PRS, PRU

Pseudorandom States (PRS)



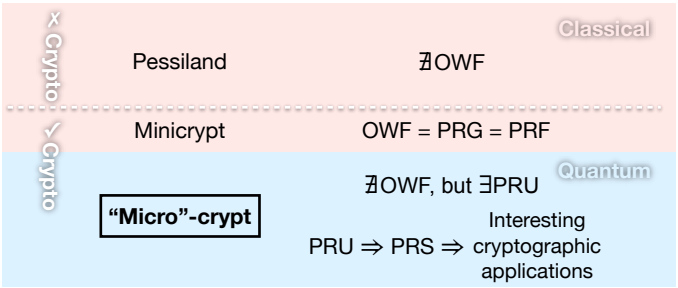
Pseudorandom Unitaries (PRU)



Classical Analogue

Pseudorandom string generator (PRG)

Pseudorandom function family (PRF)



Related Work on PRU

- **Non-adaptive security:**
  - PFC construction: PRPs + PRFs + Clifford gates [MPSY24]
  - A real-valued construction, secure on orthogonal input states [BM24]
  - Products of exponentiated sums of random permutations with random phases [CBBDX24]
- **Adaptive security:**
  - PFC: adaptive security and the strong security against inverse query [MH25]
  - Alternating pseudorandom phase unitaries and Hadamard gates [BHHP24]

2. Main Result

The distribution of the unitary corresponding to a $O(n)$-step random parallel Kac's walk is *computationally indistinguishable* from Haar distribution.

3. Key Idea

a. Parallel Kac's Walk (on $x \in \mathcal{S}(\mathbb{C}^{2^n})$) [LQSYZ24]

(1) Select a random perfect matching of $\{1,\cdots,2^n\}$

$$P = \{(i_1, j_1), \ldots, (i_{2^{n-1}}, j_{2^{n-1}})\}$$

(2) For each pair $(i_k, j_k) \in P$, sample three angles

$$\alpha_k,\ \beta_k,\ \theta_k \in [0, 2\pi)$$

(3) and set

$$\begin{bmatrix} x_{i_k} \\ x_{j_k} \end{bmatrix} \leftarrow \begin{bmatrix} e^{i\alpha_k}\cos\theta_k & -e^{i\beta_k}\sin\theta_k \\ e^{-i\beta_k}\sin\theta_k & e^{-i\alpha_k}\cos\theta_k \end{bmatrix} \begin{bmatrix} x_{i_k} \\ x_{j_k} \end{bmatrix}$$

Theorem[LQSYZ24]
$O(n)$ steps of parallel Kac's walk transform any input pure state to a family of pseudorandom states.

Random State Scrambler

b. Simulation via Quantum Circuit

- Sample a random permutation σ and a random function f
- Implement two unitary operators P_σ and H_f

$$P_\sigma = \sum_{x\in\{0,1\}^n} |\sigma(x)\rangle\langle x|$$

$|x_1\rangle$ and $|x_2\rangle$ are paired up iff $\sigma(x_1)$ and $\sigma(x_2)$ share the same suffix of length $n-1$.

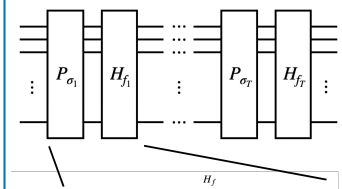
$$H_f = \sum_{y\in\{0,1\}^{n-1}} \begin{bmatrix} e^{i\alpha_y}\cos\theta_y & -e^{i\beta_y}\sin\theta_y \\ e^{-i\beta_y}\sin\theta_y & e^{-i\alpha_y}\cos\theta_y \end{bmatrix} \otimes |y\rangle\langle y|$$

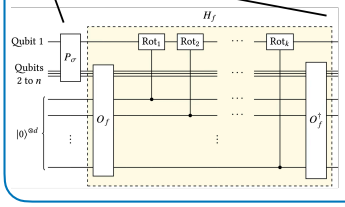
On the pair with suffix y, apply $U(\alpha_y, \beta_y, \theta_y)$.
$\alpha_y, \beta_y, \theta_y$ depend on both f and y.

4. Construction

The $HP_{n,T}$ Distribution

- Repeat the parallel Kac's walk
  - Sample T random permutations $\sigma_1, \ldots, \sigma_T$, and T random functions $f_1, \ldots, f_T$,
  - Output the unitary $\prod_{i=1}^{T}(H_{f_i} \cdot P_{\sigma_i})$.

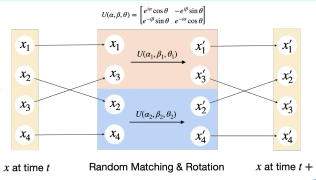




Theorem

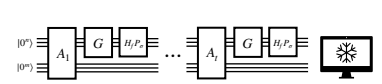
- For $T = 30n + 1$, $HP_{n,T}$ is computationally indistinguishable from Haar distribution.
- For $T = 60n + 1$, $HP_{n,T}$ is computationally indistinguishable from Haar distribution *even when adversaries are granted query access to the inverse unitary.*

Simulation



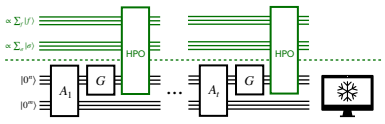
5. Proof Overview

Use the *path recording* technique [MH25].



Purification

1. Initialize two environment registers to the state $|\phi_{\varnothing}\rangle \propto \sum_{\sigma,f}|\sigma,f\rangle$
2. On each query, apply HPO
   $\mathrm{HPO}|x\rangle|\sigma\rangle|f\rangle := (H_f P_\sigma)|x\rangle|\sigma\rangle|f\rangle$



Connecting HPO and PR via Compress Isometry

Action of HPO:
$\mathrm{HPO}|x\rangle \otimes |\phi_R\rangle \longmapsto \sum_y |y\rangle \otimes |\phi_{R\cup\{x,y\}}\rangle$

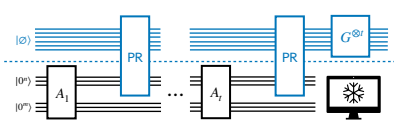
$\{|\phi_R\rangle\}$ are orthonormal vectors if $R \in \mathcal{R}^{\mathrm{DB}}$.

Compress $\sum_{R\in\mathcal{R}^{\mathrm{DB}}} |R\rangle\langle\phi_R|$

- $x_1, x_2 \in \{0,1\}^n$ are in *distinct blocks* if they have different suffix of length $n-1$.
- $R = \{(x_1, y_1), \ldots, (x_r, y_r)\} \in \mathcal{R}^{\mathrm{DB}}$ if all x in R are in distinct blocks and all y are in distinct blocks.
- Works when the environment registers are in *distinct blocks subspace*, i.e., $R \in \mathcal{R}^{\mathrm{DB}}$
- Random state scrambler G ensures this happens w.h.p.

Action of PR:
$\mathrm{PR}|x\rangle \otimes |R\rangle \longmapsto \sum_{y\notin R} |y\rangle \otimes |R \cup \{x,y\}\rangle$

$\{|R\rangle\}$ are orthonormal vectors.

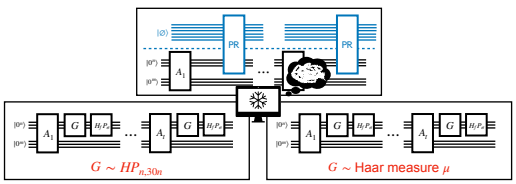


Right Invariance of PR

$$(\mathrm{PR}\cdot G\cdot A_t)\ \cdots\ (\mathrm{PR}\cdot G\cdot A_1)\ |0^{n+m}\rangle|\varnothing\rangle$$
$$= (G^{\otimes t})\cdot(\mathrm{PR}\cdot A_t)\ \cdots\ (\mathrm{PR}\cdot A_1)\ |0^{n+m}\rangle|\varnothing\rangle$$

Environment

As long as G is a random state scrambler,



6. Open Questions

- Can we build scalable PRUs from the parallel Kac's walk?
- Can we simplify our construction for near-term quantum technology (e.g., using the same random rotation or local permutation)?
- Can we find (cryptographic) applications for our PRU?

7. References

[MH25] How to construct random unitaries.
[MPSY24] Simple constructions of linear-depth t-designs and pseudorandom unitaries.
[LQSYZ24] Quantum pseudorandom scramblers.
[BM24] Real-valued somewhat-pseudorandom unitaries.
[CBBDX24] Efficient unitary designs and pseudorandom unitaries from permutations.
[BHHP24] Efficient quantum pseudorandomness from hamiltonian phase states.
[JLS17] Pseudorandom quantum states.