TCC 2022

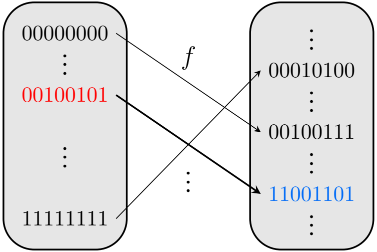
The Parallel Reversible Pebbling Game: Analyzing the Post-Quantum Security of iMHFs

Jeremiah Blocki, Blake Holman, and Seunghoon Lee

November 7, 2022

Motivation: Quantum Pre-Image Attack

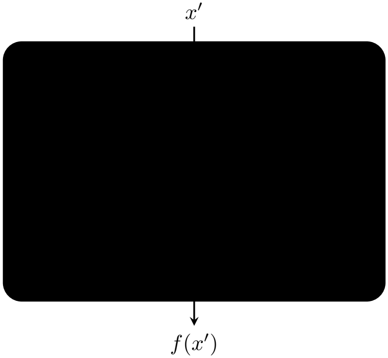
Problem. Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and a target output y, find an input $x \in \{0,1\}^n$ such that $y = f(x)$.



Find $x \in \{0,1\}^8$ s.t. $f(x) = 11001101$

Motivation: Quantum Pre-Image Attack

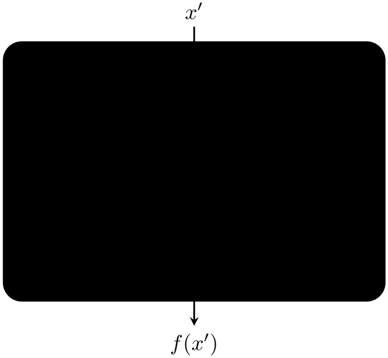
Problem. Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and a target output y, find an input $x \in \{0,1\}^n$ such that $y = f(x)$.



Find $x \in \{0,1\}^8$ s.t. $f(x) = 11001101$

Motivation: Quantum Pre-Image Attack

Problem. Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and a target output y, find an input $x \in \{0,1\}^n$ such that $y = f(x)$.

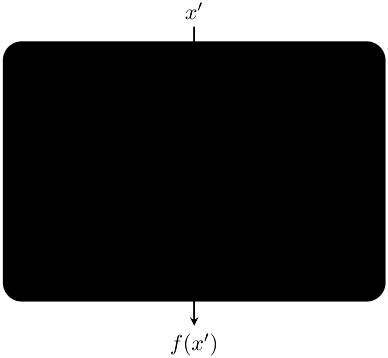


Find $x \in \{0,1\}^8$ s.t. $f(x) = 11001101$

- **Classical Unstructured Search:** need $\Omega(2^n)$ black-box queries to the function f

Motivation: Quantum Pre-Image Attack

Problem. Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and a target output y, find an input $x \in \{0,1\}^n$ such that $y = f(x)$.

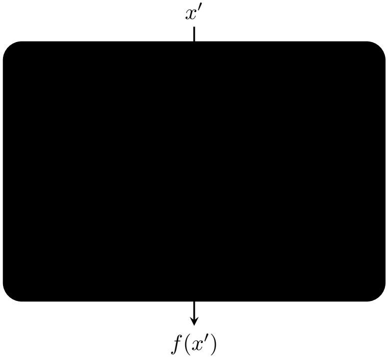


Find $x \in \{0,1\}^8$ s.t. $f(x) = 11001101$

- **Classical Unstructured Search:** need $\Omega(2^n)$ black-box queries to the function f
- **Quantum Computing:** *Grover's algorithm* [Gro96]
  - only requires $\mathcal{O}(2^{n/2})$ black-box queries to the function f
  - this is tight: *any* quantum algorithm using f as a black box must make $\Omega(2^{n/2})$ queries [BBBV97]

Motivation: Quantum Pre-Image Attack

Problem. Given a function $f : \{0,1\}^n \to \{0,1\}^n$ and a target output y, find an input $x \in \{0,1\}^n$ such that $y = f(x)$.



Find $x \in \{0,1\}^8$ s.t. $f(x) = 11001101$

- **Classical Unstructured Search:** need $\Omega(2^n)$ black-box queries to the function f
- **Quantum Computing:** *Grover's algorithm* [Gro96]
  - only requires $\mathcal{O}(2^{n/2})$ black-box queries to the function f
  - this is tight: *any* quantum algorithm using f as a black box must make $\Omega(2^{n/2})$ queries [BBBV97]
- What is the full cost of a quantum pre-image attack?

Motivation: Quantum Pre-Image Attack

The full cost of a quantum pre-image attack is defined as the *space-time cost* (ST-cost), i.e.,

$$(\text{space}(C)) \times (\text{time}(C)),$$

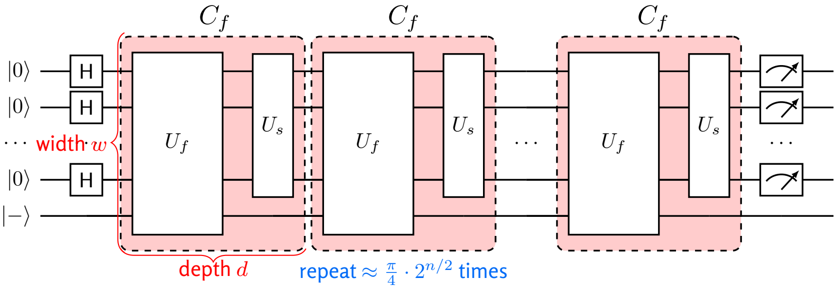
where C: a *quantum circuit* that computes the quantum pre-image attack.

Motivation: Quantum Pre-Image Attack

The full cost of a quantum pre-image attack is defined as the *space-time cost* (ST-cost), i.e.,

$$(\text{space}(C)) \times (\text{time}(C)),$$

where C: a *quantum circuit* that computes the quantum pre-image attack.



If we instantiate f with a quantum circuit C_f of width w and depth d using Grover's algorithm,

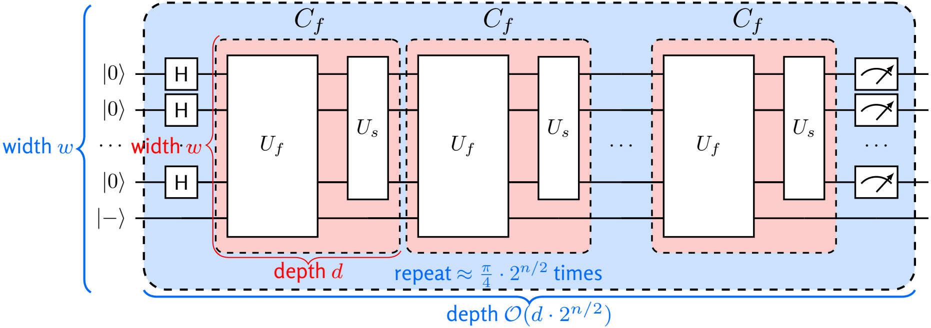
$$(\text{total ST-cost of the attack}) = \mathcal{O}(wd \cdot 2^{n/2}).$$

Motivation: Quantum Pre-Image Attack

The full cost of a quantum pre-image attack is defined as the *space-time cost* (ST-cost), i.e.,

$$(\text{space}(C)) \times (\text{time}(C)),$$

where C: a *quantum circuit* that computes the quantum pre-image attack.



If we instantiate f with a quantum circuit C_f of width w and depth d using Grover's algorithm,

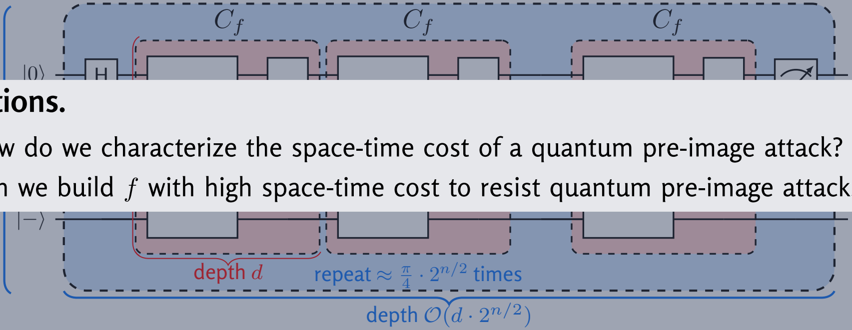
$$(\text{total ST-cost of the attack}) = \mathcal{O}(wd \cdot 2^{n/2}).$$

Motivation: Quantum Pre-Image Attack

The full cost of a quantum pre-image attack is defined as the *space-time cost* (ST-cost), i.e.,

$$(\text{space}(C)) \times (\text{time}(C)),$$

where C: a *quantum circuit* that computes the quantum pre-image attack.



Questions.

- How do we characterize the space-time cost of a quantum pre-image attack?
- Can we build f with high space-time cost to resist quantum pre-image attacks?

If we instantiate f with a quantum circuit C_f of width w and depth d using Grover's algorithm,

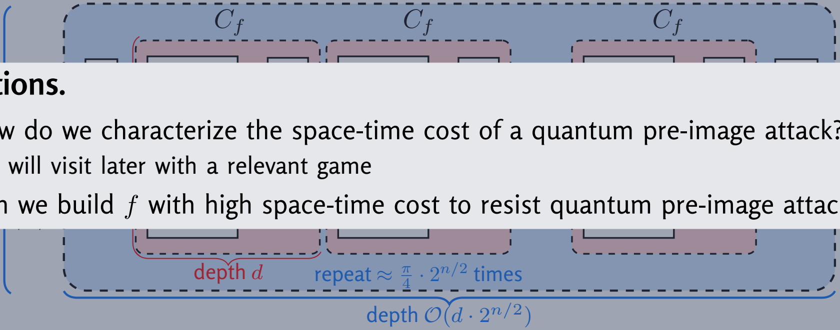
$$(\text{total ST-cost of the attack}) = \mathcal{O}(wd \cdot 2^{n/2}).$$

Motivation: Quantum Pre-Image Attack

The full cost of a quantum pre-image attack is defined as the *space-time cost* (ST-cost), i.e.,

$$(\text{space}(C)) \times (\text{time}(C)),$$

where C: a *quantum circuit* that computes the quantum pre-image attack.



Questions.

- How do we characterize the space-time cost of a quantum pre-image attack?
    - will visit later with a relevant game
- Can we build f with high space-time cost to resist quantum pre-image attacks?

If we instantiate f with a quantum circuit C_f of width w and depth d using Grover's algorithm,

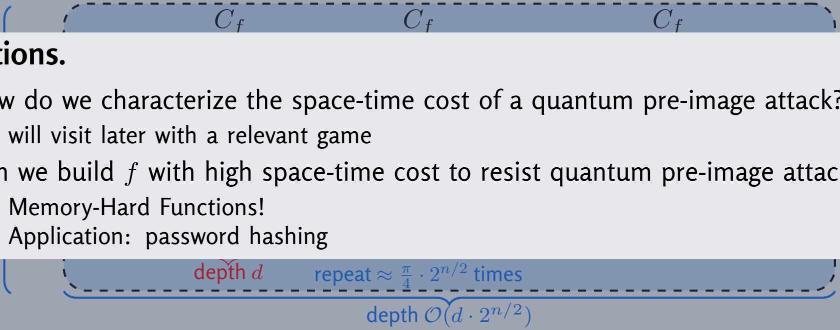
$$(\text{total ST-cost of the attack}) = \mathcal{O}(wd \cdot 2^{n/2}).$$

Motivation: Quantum Pre-Image Attack

The full cost of a quantum pre-image attack is defined as the *space-time cost* (ST-cost), i.e.,

$$(\text{space}(C)) \times (\text{time}(C)),$$

where C: a *quantum circuit* that computes the quantum pre-image attack.



Questions.

- How do we characterize the space-time cost of a quantum pre-image attack?
  - will visit later with a relevant game
- Can we build f with high space-time cost to resist quantum pre-image attacks?
  - Memory-Hard Functions!
  - Application: password hashing

If we instantiate f with a quantum circuit C_f of width w and depth d using Grover's algorithm,

$$(\text{total ST-cost of the attack}) = \mathcal{O}(wd \cdot 2^{n/2}).$$

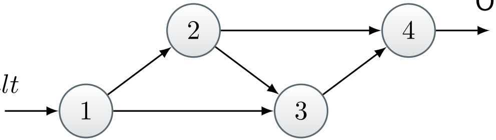
Data-Independent Memory-Hard Function (iMHF)

Definition. An iMHF $f_{G,H}$ is defined by

- $H : \{0,1\}^{2k} \rightarrow \{0,1\}^k$ (Random Oracle)
- A DAG G (encodes data-dependencies), with maximum indegree $\delta = \mathcal{O}(1)$

Output: $f_{G,H}(pwd, salt) = L_4$

Input: $pwd, salt$



$L_1 = H(pwd, salt)$ $L_3 = H(L_2, L_1)$

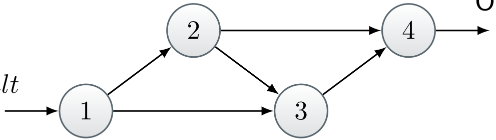
Data-Independent Memory-Hard Function (iMHF)

Definition. An iMHF $f_{G,H}$ is defined by

- $H : \{0,1\}^{2k} \to \{0,1\}^k$ (Random Oracle)
- A DAG G (encodes data-dependencies), with maximum indegree $\delta = \mathcal{O}(1)$

Output: $f_{G,H}(pwd, salt) = L_4$

Input: $pwd, salt$



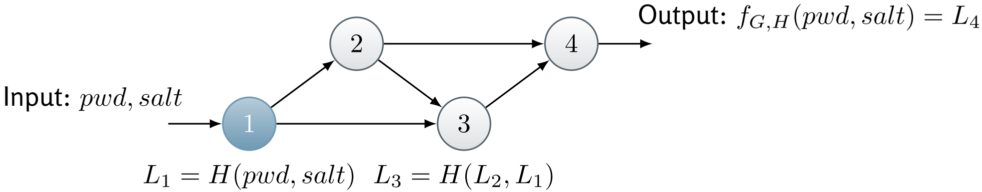
$L_1 = H(pwd, salt)$ $L_3 = H(L_2, L_1)$

- Classically, evaluating an iMHF: **the black pebbling game**
  - Rule 1: should **start with no pebbles** on the graph and **end with target nodes**
  - Rule 2: **all the parents** need to be previously pebbled to place a new pebble
  - Rule 3 **(sequential only):** can place **only one pebble** at each round

Data-Independent Memory-Hard Function (iMHF)

Definition. An iMHF $f_{G,H}$ is defined by

- $H : \{0,1\}^{2k} \to \{0,1\}^k$ (Random Oracle)
- A DAG G (encodes data-dependencies), with maximum indegree $\delta = \mathcal{O}(1)$

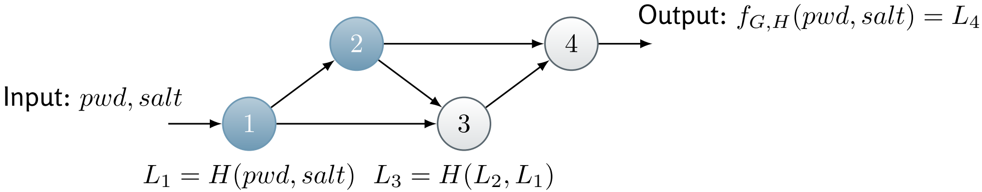


- Classically, evaluating an iMHF: **the black pebbling game**
  - Rule 1: should **start with no pebbles** on the graph and **end with target nodes**
  - Rule 2: **all the parents** need to be previously pebbled to place a new pebble
  - Rule 3 **(sequential only):** can place **only one pebble** at each round

Data-Independent Memory-Hard Function (iMHF)

Definition. An iMHF $f_{G,H}$ is defined by

- $H : \{0,1\}^{2k} \to \{0,1\}^k$ (Random Oracle)
- A DAG G (encodes data-dependencies), with maximum indegree $\delta = \mathcal{O}(1)$

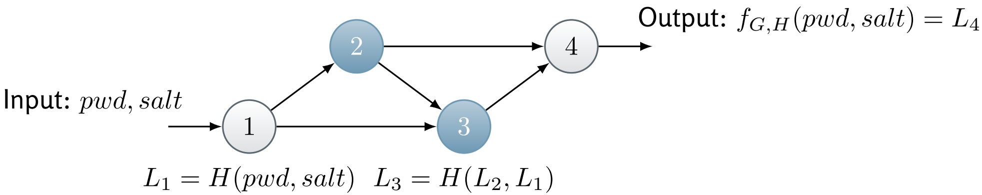


- Classically, evaluating an iMHF: **the black pebbling game**
  - Rule 1: should **start with no pebbles** on the graph and **end with target nodes**
  - Rule 2: **all the parents** need to be previously pebbled to place a new pebble
  - Rule 3 **(sequential only):** can place **only one pebble** at each round

Data-Independent Memory-Hard Function (iMHF)

Definition. An iMHF $f_{G,H}$ is defined by

- $H : \{0,1\}^{2k} \to \{0,1\}^k$ (Random Oracle)
- A DAG G (encodes data-dependencies), with maximum indegree $\delta = \mathcal{O}(1)$

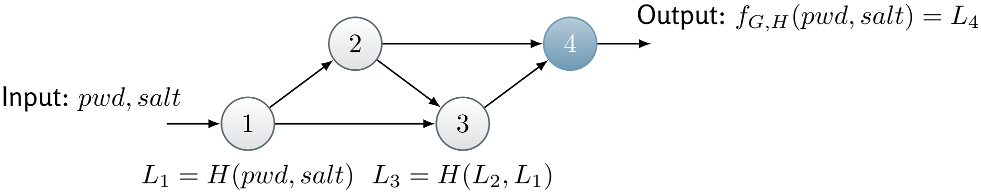


- Classically, evaluating an iMHF: **the black pebbling game**
  - Rule 1: should **start with no pebbles** on the graph and **end with target nodes**
  - Rule 2: **all the parents** need to be previously pebbled to place a new pebble
  - Rule 3 **(sequential only)**: can place **only one pebble** at each round

Data-Independent Memory-Hard Function (iMHF)

Definition. An iMHF $f_{G,H}$ is defined by

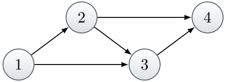
- $H : \{0,1\}^{2k} \to \{0,1\}^k$ (Random Oracle)
- A DAG G (encodes data-dependencies), with maximum indegree $\delta = \mathcal{O}(1)$



- Classically, evaluating an iMHF: **the black pebbling game**
  - Rule 1: should **start with no pebbles** on the graph and **end with target nodes**
  - Rule 2: **all the parents** need to be previously pebbled to place a new pebble
  - Rule 3 **(sequential only):** can place **only one pebble** at each round

Space-Time Complexity

In the Black Pebbling Game



A pebbling $P = (P_1 = \{1\}, P_2 = \{1,2\}, P_3 = \{2,3\}, P_4 = \{4\})$

Space-Time (ST) Complexity

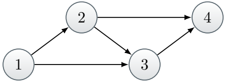
- $\mathsf{ST}(P) = (\text{time}) \times (\text{max space})$, and $\mathsf{ST}(G) = \min_P \mathsf{ST}(P)$
- For above example, we have

$$\mathsf{ST}(P) = 4 \times 2 = 8$$

Back to our first question:
Can we use black pebbling to analyze the space-time cost of a quantum circuit?

Space-Time Complexity

In the Black Pebbling Game



A pebbling $P = (P_1 = \{1\}, P_2 = \{1, 2\}, P_3 = \{2, 3\}, P_4 = \{4\})$

Space-Time (ST) Complexity

- $\mathsf{ST}(P) = (\text{time}) \times (\text{max space})$, and
  $\mathsf{ST}(G) = \min_P \mathsf{ST}(P)$
- For above example, we have

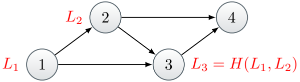
$$\mathsf{ST}(P) = 4 \times 2 = 8$$

Back to our first question:
Can we use black pebbling to analyze the space-time cost of a quantum circuit?

No!

Space-Time Complexity

In the Black Pebbling Game



A pebbling $P = (P_1 = \{1\}, P_2 = \{1,2\}, P_3 = \{2,3\}, P_4 = \{4\})$

Space-Time (ST) Complexity

- $\mathsf{ST}(P) = (\text{time}) \times (\text{max space})$, and $\mathsf{ST}(G) = \min_P \mathsf{ST}(P)$
- For above example, we have

$$\mathsf{ST}(P) = 4 \times 2 = 8$$

Back to our first question:
Can we use black pebbling to analyze the space-time cost of a quantum circuit?

No!

Why?

- Quantum circuits must be *reversible*
- $P_3 = \{2,3\} \to P_4 = \{4\}$: not a *reversible* transition
- **Quantum Uncomputation** in the QROM:

$$|x, y\rangle \xmapsto{H} |x, y \oplus H(x)\rangle$$

$$\begin{aligned}\therefore |(L_1, L_2), L_3\rangle &\xmapsto{H} |(L_1, L_2), L_3 \oplus H(L_1, L_2)\rangle \\ &= |(L_1, L_2), 0^k\rangle\end{aligned}$$

$\therefore$ to remove a pebble from node 3 using uncomputation, we need have needed pebbles on nodes 1 and 2

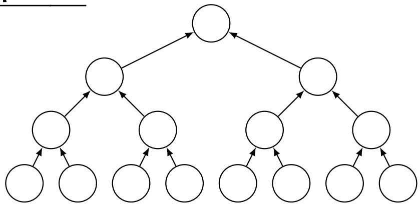
The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

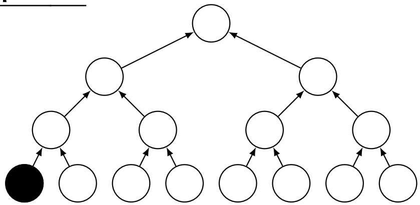
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

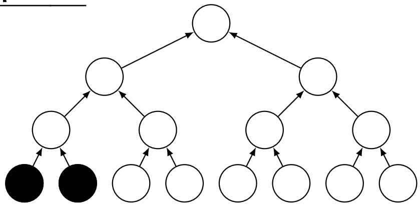
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Sequential

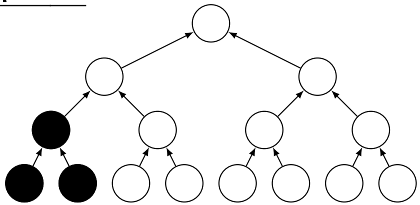


The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*

  ∵ the circuit can evaluate H in parallel

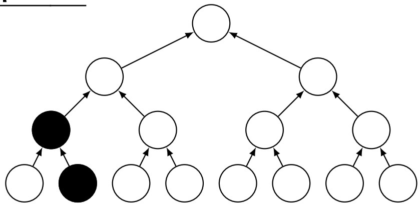
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

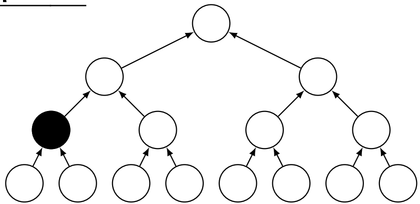
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

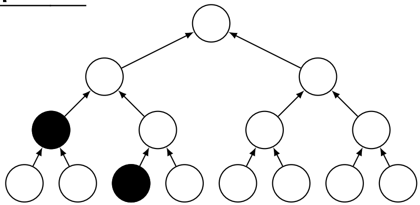
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

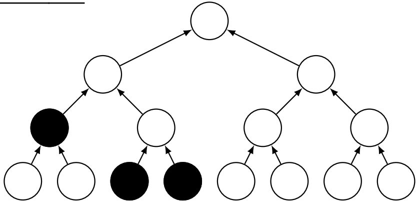
Sequential



The Sequential Reversible Pebbling Game

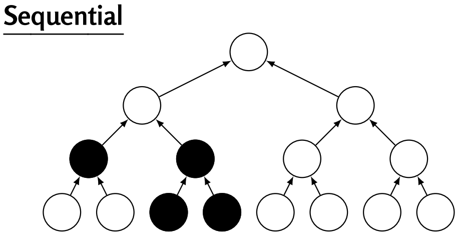
- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Sequential



The Sequential Reversible Pebbling Game

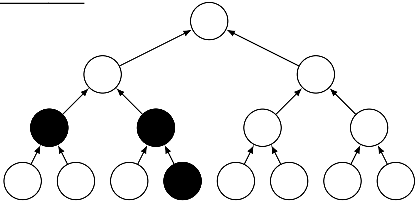
- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - ∵ the circuit can evaluate H in parallel

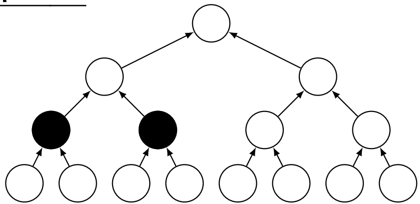
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

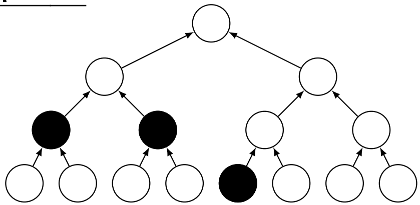
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

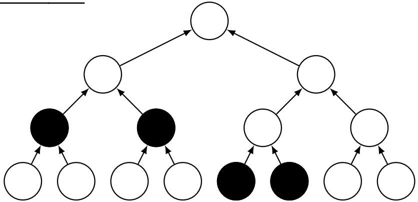
<u>**Sequential**</u>



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

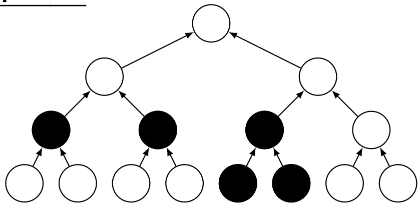
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

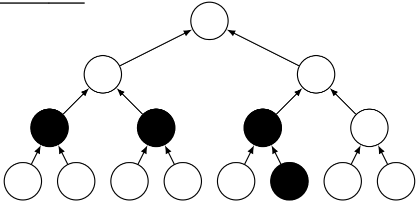
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

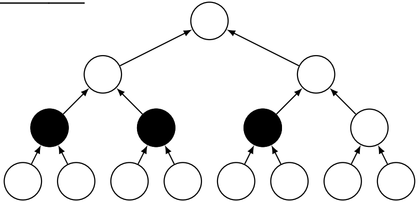
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

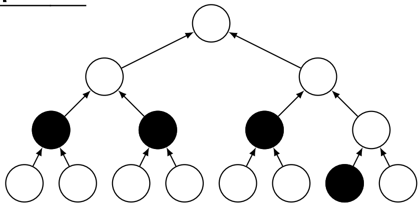
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

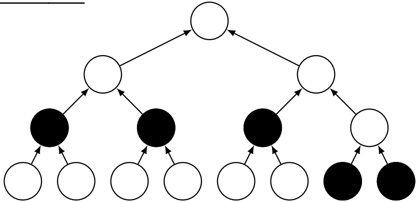
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

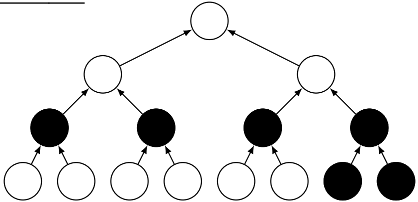
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

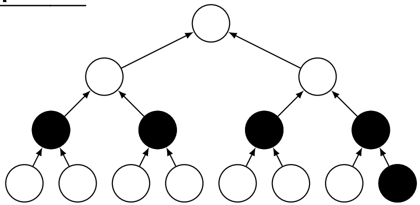
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

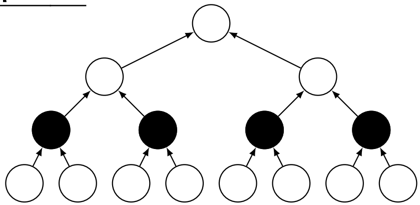
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- “Sequential” Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

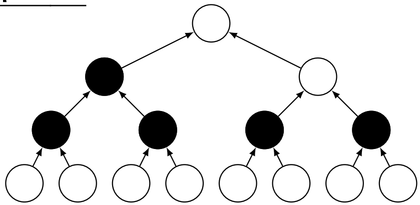
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- “Sequential” Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

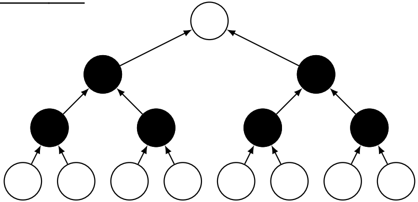
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

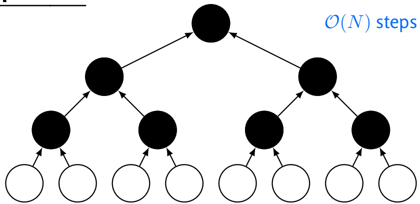
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

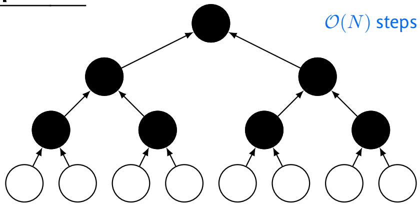
Sequential



The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Sequential



$\mathcal{O}(N)$ steps

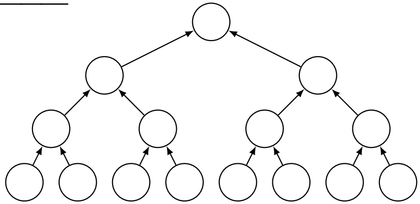
Sequential Reversible Pebbling:

$$\Rightarrow \text{(ST-Cost)} = \text{(space)} \times \text{(time)}$$
$$= 7 \times 23 = 161$$

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Parallel



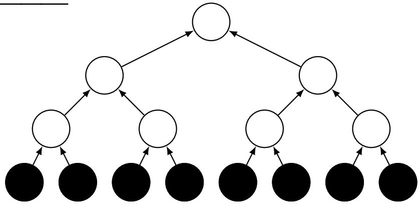
Sequential Reversible Pebbling:

$$\Rightarrow \text{(ST-Cost)} = \text{(space)} \times \text{(time)}$$
$$= 7 \times 23 = 161$$

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Parallel



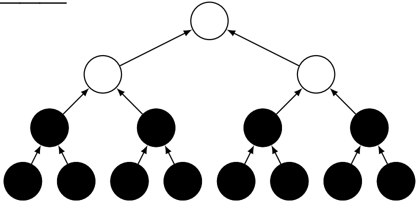
Sequential Reversible Pebbling:

$\Rightarrow$ (ST-Cost) = (space) $\times$ (time)
$= 7 \times 23 = 161$

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Parallel



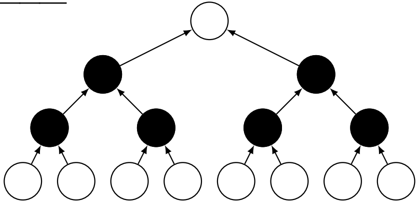
Sequential Reversible Pebbling:

$\Rightarrow$ (ST-Cost) $=$ (space) $\times$ (time)
$= 7 \times 23 = 161$

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Parallel



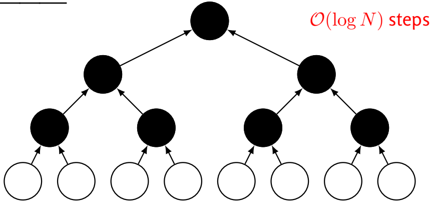
Sequential Reversible Pebbling:

$$\Rightarrow \text{(ST-Cost)} = \text{(space)} \times \text{(time)}$$
$$= 7 \times 23 = 161$$

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR+19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Parallel



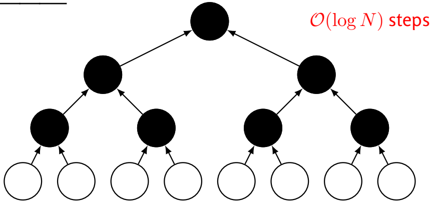
Sequential Reversible Pebbling:

$$\Rightarrow \text{(ST-Cost)} = \text{(space)} \times \text{(time)}$$
$$= 7 \times 23 = 161$$

The Sequential Reversible Pebbling Game

- Prior work [Ben89, LV96, Krá01, MSR$^+$19] introduced *sequential* reversible computation /reversible pebbling game
  - Added more constraints to capture reversible transitions by quantum uncomputation
  - Analyzed space-time tradeoffs in quantum computing
- "Sequential" Reversible Pebbling Game: *Still not suitable* for analyzing the *space-time cost of a quantum circuit*
  - $\because$ the circuit can evaluate H in parallel

Parallel



Sequential Reversible Pebbling:

$\Rightarrow$ (ST-Cost) $=$ (space) $\times$ (time) $= 7 \times 23 = 161$

Analyzing the Quantum Circuit:

$\Rightarrow$ (ST-Cost) $= 12 \times 4 = 48$

$\Rightarrow$ the time cost can be decreased from $\mathcal{O}(N)$ to $\mathcal{O}(\log N)$

Key Research Question

Back to Our First Question

How to characterize the space-time cost of a quantum circuit $C_{f_{G,H}}$ for a data-independent Memory-Hard Function $f_{G,H}$?

Key Research Question

Back to Our First Question

How to characterize the space-time cost of a quantum circuit $C_{f_{G,H}}$ for a data-independent Memory-Hard Function $f_{G,H}$?

Partial Answer: *The Parallel Reversible Pebbling Game*
& Study some attacks against iMHFs in this pebbling model

Definition: Parallel Reversible Pebbling Game

A *parallel reversible pebbling* $P = (P_0, \ldots, P_t)$ is a sequence of pebbling configurations with the conditions (same as classical):

1. start with no pebbles (i.e., $P_0 = \varnothing$) and end with target nodes T (i.e., $T \subseteq P_t$) $^{(\star)}$,
2. a new pebble can be added only if its parents were previously pebbled, and

the following *additional* conditions:

Condition 3. (Quantum No-Deletion)

a pebble *can be deleted* only if all of its parents were previously pebbled

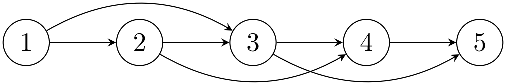
Condition 4. (Quantum Reversibility)

we *must keep the pebble* if a pebble was required to generate new pebbles (or delete pebbles)

$(\star)$ we can make this condition strict, i.e., $P_t = T$. See the paper for detail.

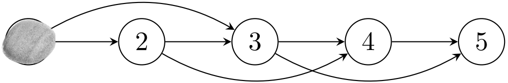
Example: A Parallel Pebbling

Classical vs. Reversible



Example: A Parallel Pebbling

Classical vs. Reversible

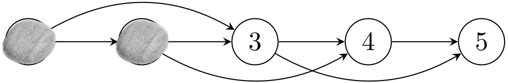


Classical

Round 1 ● ○ ○ ○ ○

Example: A Parallel Pebbling

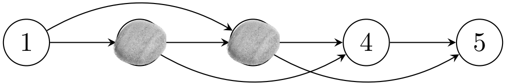
Classical vs. Reversible



	Classical				
Round 1	●	○	○	○	○
Round 2	●	●	○	○	○

Example: A Parallel Pebbling

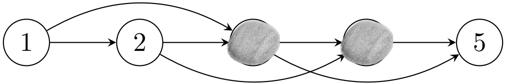
Classical vs. Reversible



	Classical				
Round 1	●	○	○	○	○
Round 2	●	●	○	○	○
Round 3	○	●	●	○	○

Example: A Parallel Pebbling

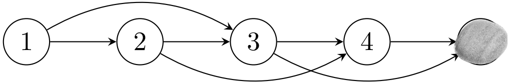
Classical vs. Reversible

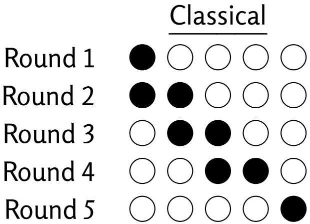


	Classical				
Round 1	●	○	○	○	○
Round 2	●	●	○	○	○
Round 3	○	●	●	○	○
Round 4	○	○	●	●	○

Example: A Parallel Pebbling

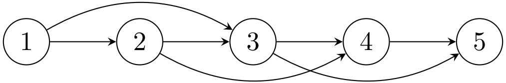
Classical vs. Reversible





Example: A Parallel Pebbling

Classical vs. Reversible



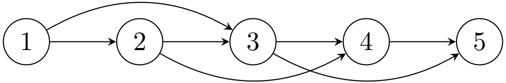
	Classical				
Round 1	●	○	○	○	○
Round 2	●	●	○	○	○
Round 3	○	●	●	○	○
Round 4	○	$\boxed{\circ}$	●	●	○
Round 5	○	$\boxed{\circ}$	$\boxed{\circ}$	○	●

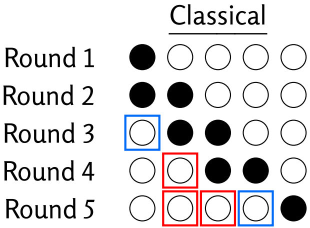
$\boxed{}$: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

– cannot remove pebble since not all parents were pebbled

Example: A Parallel Pebbling

Classical vs. Reversible





: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

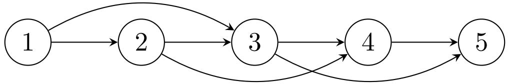
– cannot remove pebble since not all parents were pebbled

: illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

– must keep pebbles since they were required to place a new pebble

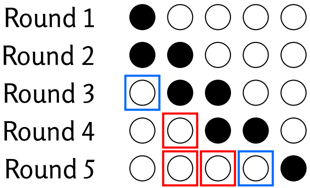
Example: A Parallel Pebbling

Classical vs. Reversible



Classical

Reversible



: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

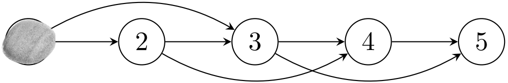
– cannot remove pebble since not all parents were pebbled

: illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

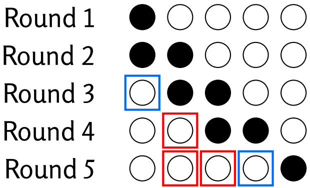
– must keep pebbles since they were required to place a new pebble

Example: A Parallel Pebbling

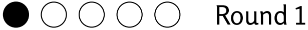
Classical vs. Reversible



Classical



Reversible



: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

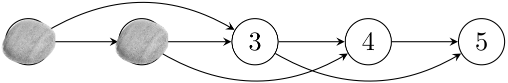
– cannot remove pebble since not all parents were pebbled

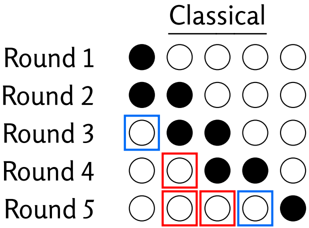
: illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

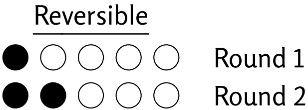
– must keep pebbles since they were required to place a new pebble

Example: A Parallel Pebbling

Classical vs. Reversible







: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

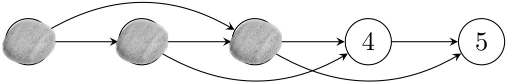
– cannot remove pebble since not all parents were pebbled

: illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

– must keep pebbles since they were required to place a new pebble

Example: A Parallel Pebbling

Classical vs. Reversible



Classical

Round 1	●	○	○	○	○
Round 2	●	●	○	○	○
Round 3	○	●	●	○	○
Round 4	○	○	●	●	○
Round 5	○	○	○	○	●

Reversible

●	○	○	○	○	Round 1
●	●	○	○	○	Round 2
●	●	●	○	○	Round 3

: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

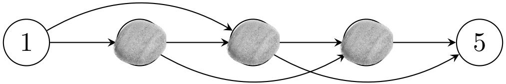
– cannot remove pebble since not all parents were pebbled

: illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

– must keep pebbles since they were required to place a new pebble

Example: A Parallel Pebbling

Classical vs. Reversible



Classical

Round 1	●	○	○	○	○
Round 2	●	●	○	○	○
Round 3	○	●	●	○	○
Round 4	○	○	●	●	○
Round 5	○	○	○	○	●

Reversible

●	○	○	○	○	Round 1
●	●	○	○	○	Round 2
●	●	●	○	○	Round 3
○	●	●	●	○	Round 4

□ : illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

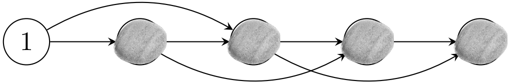
– cannot remove pebble since not all parents were pebbled

□ : illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

– must keep pebbles since they were required to place a new pebble

Example: A Parallel Pebbling

Classical vs. Reversible



Classical

Round 1	●	○	○	○	○
Round 2	●	●	○	○	○
Round 3	○	●	●	○	○
Round 4	○	○	●	●	○
Round 5	○	○	○	○	●

Reversible

●	○	○	○	○	Round 1
●	●	○	○	○	Round 2
●	●	●	○	○	Round 3
○	●	●	●	○	Round 4
○	●	●	●	●	Round 5

: illegal in a reversible pebbling by Condition 3 (Quantum No-Deletion)

– cannot remove pebble since not all parents were pebbled

: illegal in a reversible pebbling by Condition 4 (Quantum Reversibility)

– must keep pebbles since they were required to place a new pebble

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

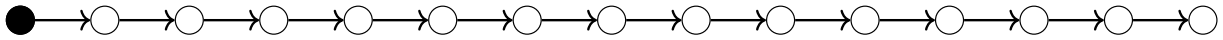
Parallel Classical Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling



Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

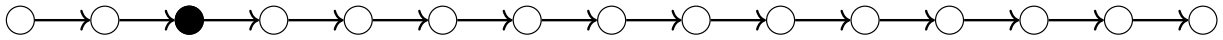
Parallel Classical Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

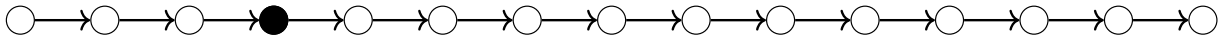


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

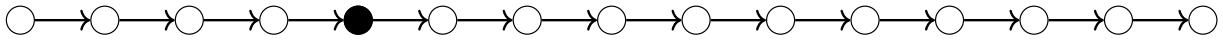


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

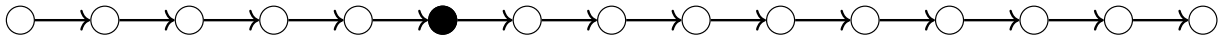


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

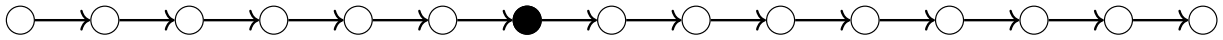


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling



Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

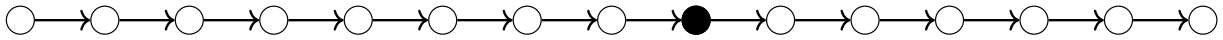
Parallel Classical Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
    - can be characterized as a line graph
    - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

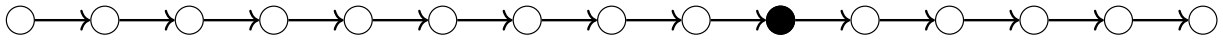


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

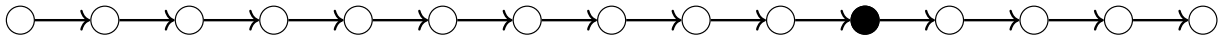


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

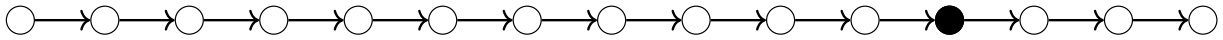


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
    - can be characterized as a line graph
    - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

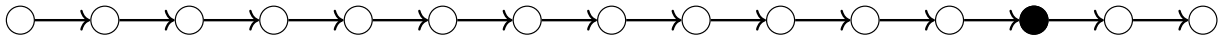


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

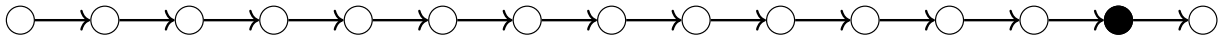


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling

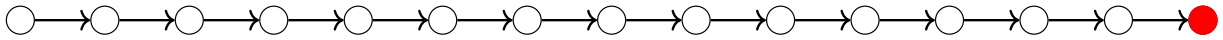


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Classical Pebbling



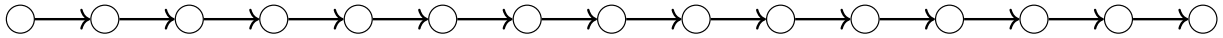
(ST-Cost) $= N$

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

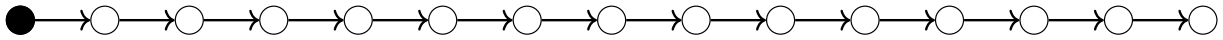


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling



Reversible Pebbling Attack 1

Attack on a Line Graph

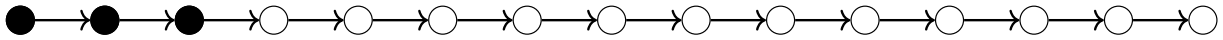
- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

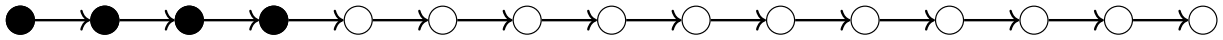
- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling 

Reversible Pebbling Attack 1

Attack on a Line Graph

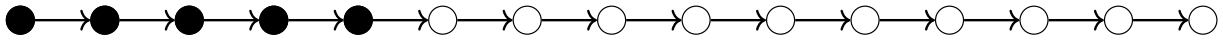
- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling 

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

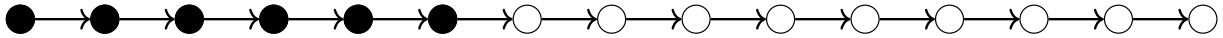
Naïve Reversible Pebbling 

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

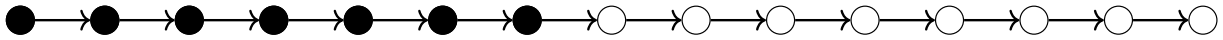


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling



Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Naïve Reversible Pebbling

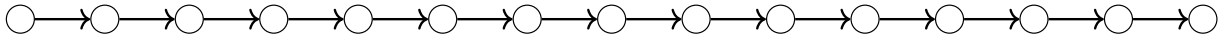
$(\text{ST-Cost}) = N^2$

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

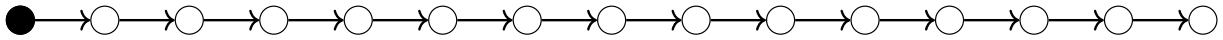


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

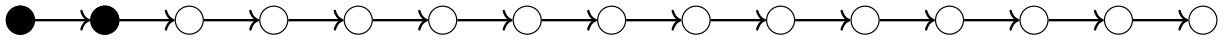


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

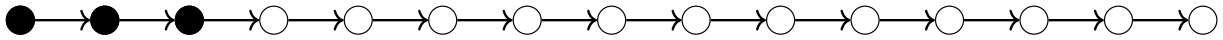


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

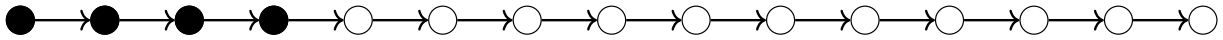


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

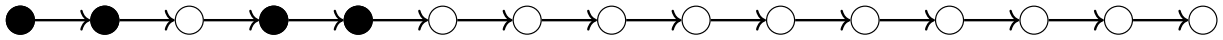


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

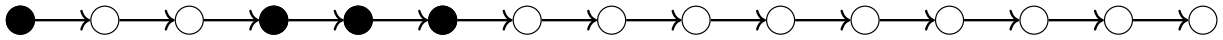


Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

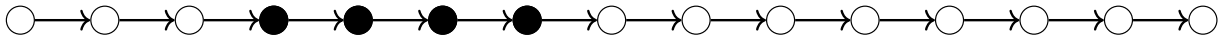
Parallel Reversible Pebbling [Our Work]



Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work] 

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

Reversible Pebbling Attack 1

Attack on a Line Graph

- **PBKDF2** [Kal00] and **BCRYPT** [PM99]: widely deployed hash functions
  - can be characterized as a line graph
  - Are they resistant to quantum pre-image attacks?

Parallel Reversible Pebbling [Our Work]

$\|$: parallel, $\leftrightarrows$: reversible

Our Result. For a line graph L_N with N nodes, we have $\mathsf{ST}^{\|,\leftrightarrows}(L_N) = \mathcal{O}\left(N^{1+\frac{2}{\sqrt{\log N}}}\right)$.

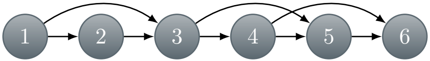
- We modified Li and Vitányi's (sequential) strategy [LV96]
- A similar (sequential) argument was implicitly assumed by Bennett [Ben89] but was not formalized as a reversible pebbling strategy

Reversible Pebbling Attack 2

Attack on Any (e, d)-Reducible DAGs

Definition. A DAG $G = (V, E)$ is *(e, d)-reducible* if there exists a *depth-reducing set* $S \subseteq V$ of size $|S| \leq e$ such that the longest path in $G - S$ has length $\leq d$.

Example. $(2, 2)$-reducible graph

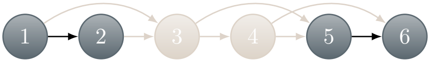


Reversible Pebbling Attack 2

Attack on Any (e, d)-Reducible DAGs

Definition. A DAG $G = (V, E)$ is *(e, d)-reducible* if there exists a *depth-reducing set* $S \subseteq V$ of size $|S| \leq e$ such that the longest path in $G - S$ has length $\leq d$.

Example. $(2, 2)$-reducible graph

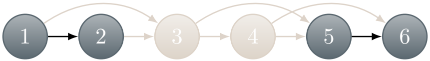


Reversible Pebbling Attack 2

Attack on Any (e, d)-Reducible DAGs

Definition. A DAG $G = (V, E)$ is *(e, d)-reducible* if there exists a *depth-reducing set* $S \subseteq V$ of size $|S| \leq e$ such that the longest path in $G - S$ has length $\leq d$.

Example. $(2, 2)$-reducible graph



Our Result. If G is (e, d)-reducible, then $\mathsf{ST}^{\|,\Leftrightarrow}(G) = \mathcal{O}(Ne + Nd2^d)$.

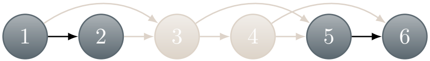
- It becomes useful when $e \ll N$ and $d \ll \log N$ (which implies $\mathsf{ST}^{\|,\Leftrightarrow}(G) \ll \mathcal{O}(N^2)$)

Reversible Pebbling Attack 2

Attack on Any (e, d)-Reducible DAGs

Definition. A DAG $G = (V, E)$ is (e, d)*-reducible* if there exists a *depth-reducing set* $S \subseteq V$ of size $|S| \leq e$ such that the longest path in $G - S$ has length $\leq d$.

Example. $(2, 2)$-reducible graph



Our Result. If G is (e, d)-reducible, then $\mathsf{ST}^{\parallel, \Longleftrightarrow}(G) = \mathcal{O}(Ne + Nd2^d)$.

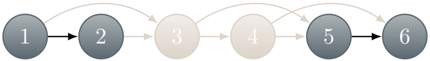
- It becomes useful when $e \ll N$ and $d \ll \log N$ (which implies $\mathsf{ST}^{\parallel, \Longleftrightarrow}(G) \ll \mathcal{O}(N^2)$)
- **Argon2i-A/B**: winner of the password hashing competition/standardized

Reversible Pebbling Attack 2

Attack on Any (e, d)-Reducible DAGs

Definition. A DAG $G = (V, E)$ is *(e, d)-reducible* if there exists a *depth-reducing set* $S \subseteq V$ of size $|S| \leq e$ such that the longest path in $G - S$ has length $\leq d$.

Example. $(2, 2)$-reducible graph

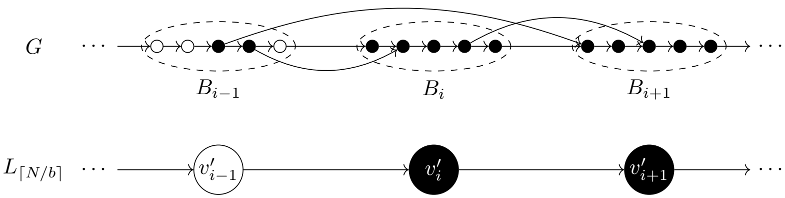


Our Result. If G is (e, d)-reducible, then $\mathsf{ST}^{\|,\leftrightarrow}(G) = \mathcal{O}(Ne + Nd2^d)$.

- It becomes useful when $e \ll N$ and $d \ll \log N$ (which implies $\mathsf{ST}^{\|,\leftrightarrow}(G) \ll \mathcal{O}(N^2)$)
- **Argon2i-A/B**: winner of the password hashing competition/standardized
- Using this result, we have $\mathsf{ST}^{\|,\leftrightarrow}(\text{Argon2i-A}) = \mathcal{O}(N^2 \log\log N / \sqrt{\log N})$ and $\mathsf{ST}^{\|,\leftrightarrow}(\text{Argon2i-B}) = \mathcal{O}(N^2 / \sqrt[3]{\log N})$

Reversible Pebbling Attack 3

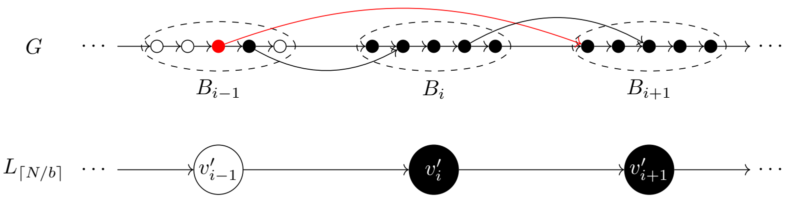
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G

Reversible Pebbling Attack 3

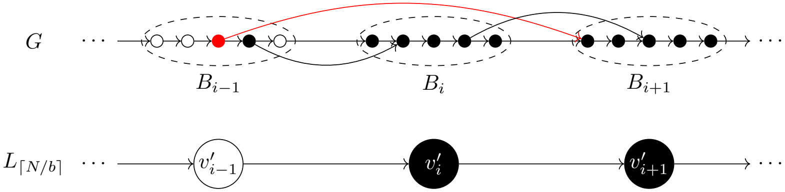
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block

Reversible Pebbling Attack 3

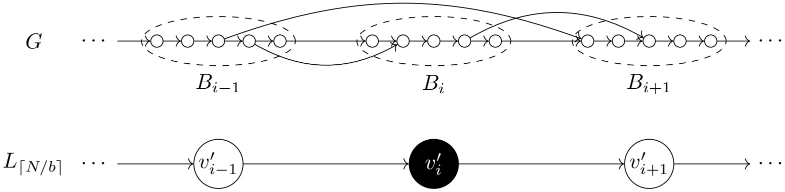
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

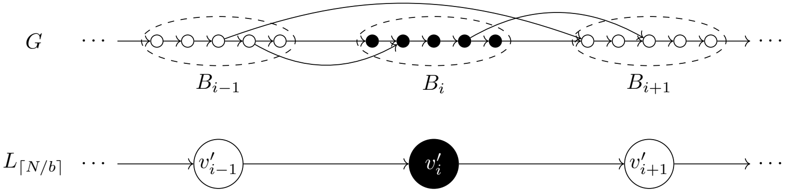
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

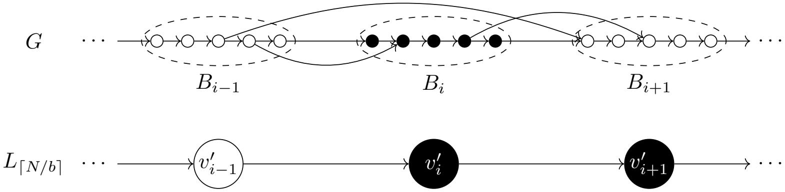
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

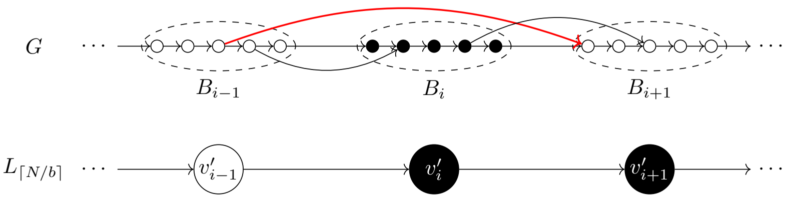
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

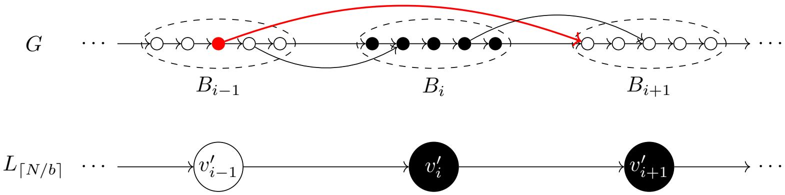
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

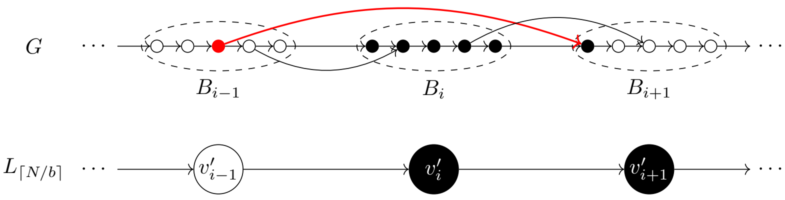
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

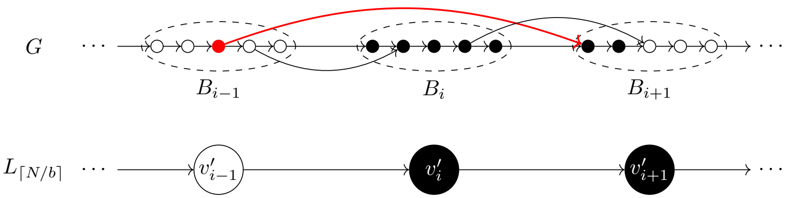
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

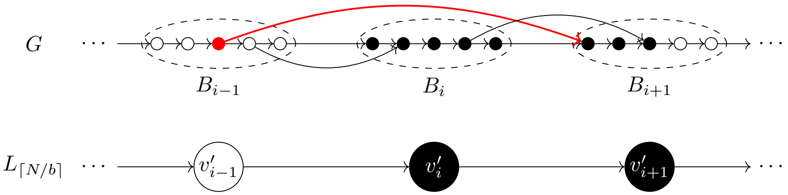
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

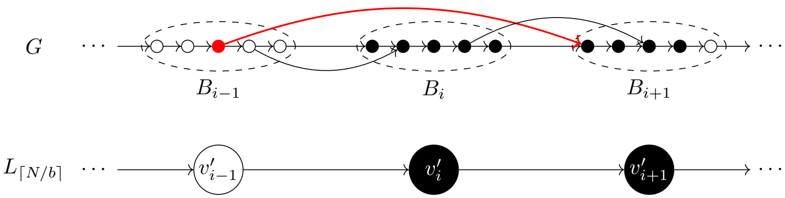
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

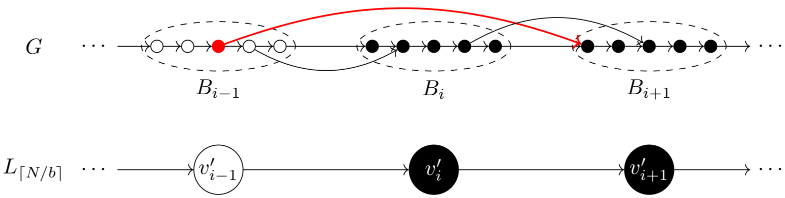
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

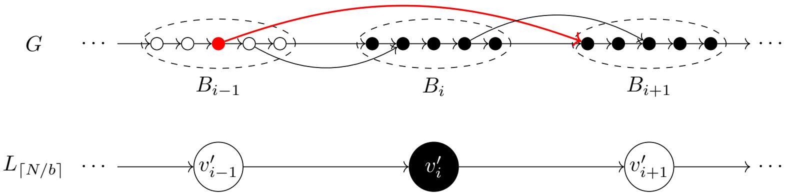
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

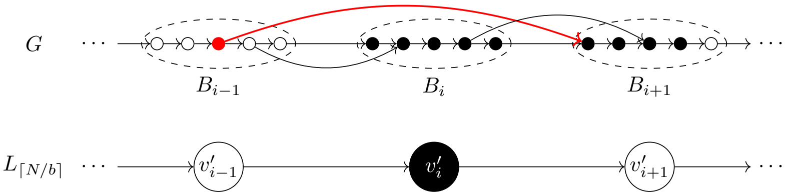
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

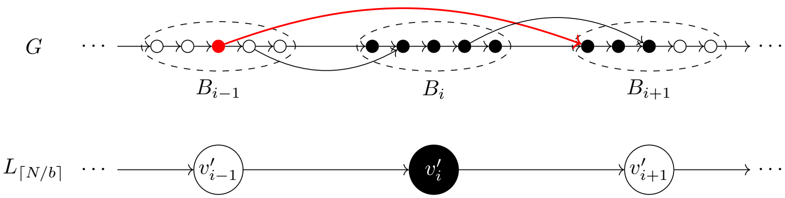
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

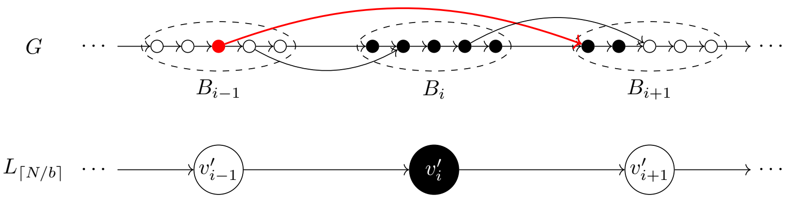
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

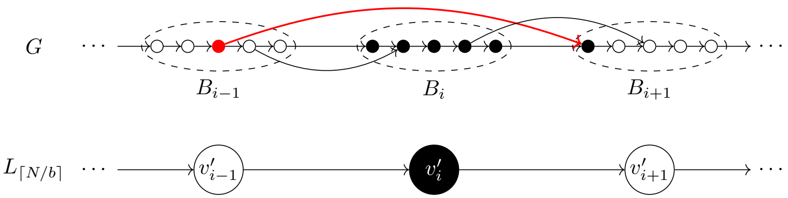
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

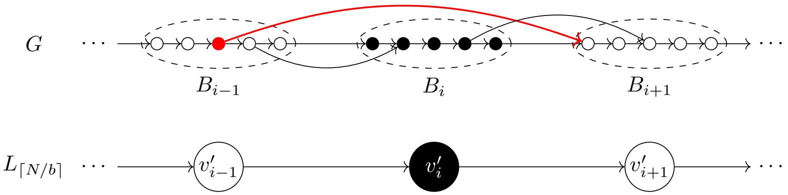
Using an Induced Line Graph



- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Reversible Pebbling Attack 3

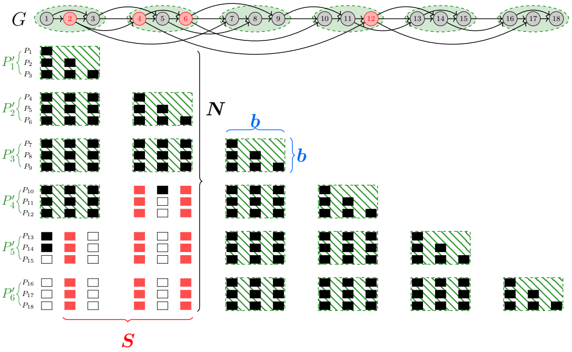
Using an Induced Line Graph

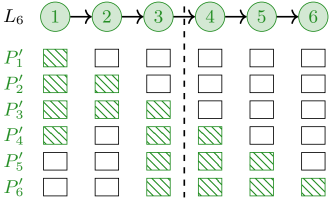


- Given a graph G, split into blocks of size b and create a line graph $L_{\lceil N/b \rceil}$ of size $\lceil N/b \rceil$
- Transform the reversible pebbling of $L_{\lceil N/b \rceil}$ to the original graph G
- The red node above is called a *skip node* as the edge *skips over* the next block
- **Key Intuition:** If we keep those skip nodes, then we can easily transform a reversible pebbling of $L_{\lceil N/b \rceil}$ to a reversible pebbling of G

Example

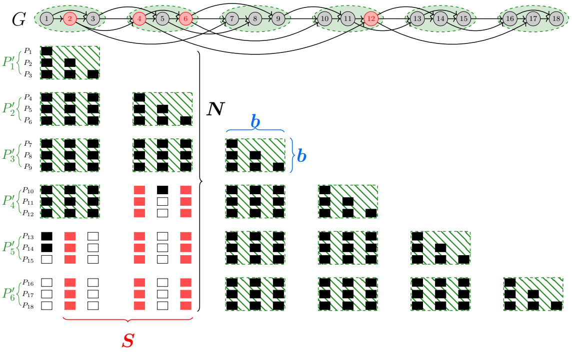
Attack Using an Induced Line Graph

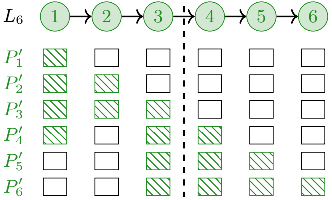




Example

Attack Using an Induced Line Graph





Our Result.

$\mathsf{ST}^{\|,\Leftrightarrow}(G) = \mathcal{O}\left(SN + b^2 \cdot \mathsf{ST}^{\|,\Leftrightarrow}(L_{\lceil N/b \rceil})\right)$, where S = (# skip nodes) and $b > 0$: block size

iMHF Example: DRSample

Attack Using an Induced Line Graph

- **DRSample** [ABH17]: a practical iMHF candidate with stronger *classical* memory-hardness
- For DRSample, we showed that (whp) the number of *skip nodes* is at most

$$(\text{\# skip nodes}) = \mathcal{O}\left(\frac{N \log\log N}{\log N}\right),$$

when we set the block size $b = \mathcal{O}(N/\log^2 N)$.

$$\Rightarrow \mathsf{ST}^{\parallel,\Leftrightarrow}(\mathsf{DRSample}) = \mathcal{O}\left(\frac{N^2 \log\log N}{\log N}\right).$$

iMHF Example: DRSample

Attack Using an Induced Line Graph

- **DRSample** [ABH17]: a practical iMHF candidate with stronger *classical* memory-hardness
- For DRSample, we showed that (whp) the number of *skip nodes* is at most

$$(\text{\# skip nodes}) = \mathcal{O}\left(\frac{N \log\log N}{\log N}\right),$$

when we set the block size $b = \mathcal{O}(N/\log^2 N)$.

$$\Rightarrow \mathsf{ST}^{\parallel,\leftrightarrow}(\text{DRSample}) = \mathcal{O}\left(\frac{N^2 \log\log N}{\log N}\right).$$

- **Note.** DRSample admits a more efficient reversible pebbling attack than Argon2i-A/B

cf.) $\mathsf{ST}^{\parallel,\leftrightarrow}(\text{Argon2i-A}) = \mathcal{O}\left(\frac{N^2 \log\log N}{\sqrt{\log N}}\right)$ and $\mathsf{ST}^{\parallel,\leftrightarrow}(\text{Argon2i-B}) = \mathcal{O}\left(\frac{N^2}{\sqrt[3]{\log N}}\right)$

Other Results

Parallel Amortized Space-Time Cost

The attacks so far:

- We considered running a single instance of Grover's search

Other Results

Parallel Amortized Space-Time Cost

The attacks so far:

- We considered running a single instance of Grover's search
- What if the attacker runs *multiple instances* of Grover's algorithm in parallel?

$\Rightarrow$ can "amortize" space usage over multiple inputs

$\therefore$ Amortized Space-Time Complexity (aST) for parallel reversible pebblings also matters!
(:= the sum of the number of pebbles used in each round)

Other Results

Parallel Amortized Space-Time Cost

The attacks so far:

- We considered running a single instance of Grover's search
- What if the attacker runs *multiple instances* of Grover's algorithm in parallel?

$\Rightarrow$ can "amortize" space usage over multiple inputs

$\therefore$ Amortized Space-Time Complexity (aST) for parallel reversible pebblings also matters!
(:= the sum of the number of pebbles used in each round)

Our Result: We extend the (non-reversible) Alwen and Blocki's attack [AB16]

Theorem. If G is (e, d)-reducible with N nodes with indegree δ, then

$$\mathsf{aST}^{\parallel,\leftrightarrow}(G) \leq \min_{g \geq d} \left\{ 2N \left(\frac{2Nd}{g} + e + (\delta + 1)g \right) + N + \frac{2Nd}{g} \right\}.$$

- **Corollary:** $\mathsf{aST}^{\parallel,\leftrightarrow}(\text{Argon2-A}) = \mathcal{O}(N^{1.75} \log N)$ and $\mathsf{aST}^{\parallel,\leftrightarrow}(\text{Argon2-B}) = \mathcal{O}(N^{1.8})$.

Conclusion

- We introduced the parallel reversible pebbling game, and
- We use this game to analyze the reversible space-time complexity of a line graph and data-independent Memory-Hard Functions such as Argon2i-A/B and DRSample
- We also give a reversible pebbling attack with low reversible cumulative pebbling cost by extending [AB16] attack

Conclusion

- We introduced the parallel reversible pebbling game, and
- We use this game to analyze the reversible space-time complexity of a line graph and data-independent Memory-Hard Functions such as Argon2i-A/B and DRSample
- We also give a reversible pebbling attack with low reversible cumulative pebbling cost by extending [AB16] attack

Open Questions

- Asymptotically stronger reversible pebbling attacks for iMHFs?
  - Can we extend the recursive pebbling attack [ABP17] to the reversible setting?
- Is there a DAG with constant indegree having (parallel) reversible ST-cost $\Omega(N^2)$?
  - Candidate: DRS+BRG [BHK$^+$19], none of our attacks performed well against DRS+BRG
- Can we come up with a reversible pebbling reduction in the parallel quantum random oracle model?
  - We only showed that efficient reversible pebbling attacks yield efficient quantum pre-image attacks, but not the reverse direction

Conclusion

- We introduced the parallel reversible pebbling game, and
- We use this game to analyze the reversible space-time complexity of a line graph and data-independent Memory-Hard Functions such as Argon2i-A/B and DRSample
- We also give a reversible pebbling attack with low reversible cumulative pebbling cost by extending [AB16] attack

Open Questions

Questions?

- Asymptotically stronger reversible pebbling attacks for iMHFs?
  - Can we extend the recursive pebbling attack [ABP17] to the reversible setting?
- Is there a DAG with constant indegree having (parallel) reversible ST-cost $\Omega(N^2)$?
  - Candidate: DRS+BRG [BHK$^+$19], none of our attacks performed well against DRS+BRG
- Can we come up with a reversible pebbling reduction in the parallel quantum random oracle m
  - We only showed that efficient reversible pebbling attacks yield efficient quantum pre-imag attacks, but not the reverse direction

References I

Joël Alwen and Jeremiah Blocki, *Efficiently computing data-independent memory-hard functions*, CRYPTO 2016, Part II (Matthew Robshaw and Jonathan Katz, eds.), LNCS, vol. 9815, Springer, Heidelberg, August 2016, pp. 241–271.

Joël Alwen, Jeremiah Blocki, and Ben Harsha, *Practical graphs for optimal side-channel resistant memory-hard functions*, ACM CCS 2017 (Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, eds.), ACM Press, October / November 2017, pp. 1001–1017.

Joël Alwen, Jeremiah Blocki, and Krzysztof Pietrzak, *Depth-robust graphs and their cumulative memory complexity*, EUROCRYPT 2017, Part III (Jean-Sébastien Coron and Jesper Buus Nielsen, eds.), LNCS, vol. 10212, Springer, Heidelberg, April / May 2017, pp. 3–32.

Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh V. Vazirani, *Strengths and weaknesses of quantum computing*, SIAM J. Comput. **26** (1997), no. 5, 1510–1523.

Charles H. Bennett, *Time/space trade-offs for reversible computation*, SIAM J. Comput. **18** (1989), no. 4, 766–776.

Jeremiah Blocki, Benjamin Harsha, Siteng Kang, Seunghoon Lee, Lu Xing, and Samson Zhou, *Data-independent memory hard functions: New attacks and stronger constructions*, CRYPTO 2019, Part II (Alexandra Boldyreva and Daniele Micciancio, eds.), LNCS, vol. 11693, Springer, Heidelberg, August 2019, pp. 573–607.

References II

Lov K. Grover, *A fast quantum mechanical algorithm for database search*, 28th ACM STOC, ACM Press, May 1996, pp. 212–219.

Burt Kaliski, *PKCS #5: Password-Based Cryptography Specification Version 2.0*, RFC 2898, RSA Laboratories, September 2000.

Richard Král'ovič, *Time and space complexity of reversible pebbling*, SOFSEM 2001: Theory and Practice of Informatics (Berlin, Heidelberg) (Leszek Pacholski and Peter Ružička, eds.), Springer Berlin Heidelberg, 2001, pp. 292–303.

Ming Li and Paul Vitányi, *Reversibility and adiabatic computation: Trading time and space for energy*, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **452** (1996), no. 1947, 769–789.

Giulia Meuli, Mathias Soeken, Martin Roetteler, Nikolaj Bjorner, and Giovanni De Micheli, *Reversible pebbling game for quantum memory management*, 2019 Design, Automation Test in Europe Conference Exhibition (DATE), 2019, pp. 288–291.

Niels Provos and David Mazières, *A future-adaptive password scheme*, Proceedings of the Annual Conference on USENIX Annual Technical Conference (USA), ATEC '99, USENIX Association, 1999, p. 32.