On the Security of Proofs of Sequential Work in a Post-Quantum World

Jeremiah Blocki¹, Seunghoon Lee¹, Samson Zhou²

¹Department of Computer Science, Purdue University ²School of Computer Science, Carnegie Mellon University

July 28, 2021



Carnegie Mellon University

Conference on Information-Theoretic Cryptography (ITC) 2021



CS590 FINAL EXAM



Jeremiah Blocki, Seunghoon Lee, Samson Zhoi

On the Security of Proofs of Sequential Work in a Post-Quantum World

CS590 FINAL EX/ গত্যা Q Â 볶 ₹₩ Fr*

[CS59	0] 5 mins late - having internet issue
CG	$\begin{array}{c} \mbox{Cinseer Goodman} \\ \mbox{Tus $SR2021 9.09 PM} \\ \mbox{Tvs $Seurghoon Lee} \\ \hline \\ \hline \hline \\ \hline \\ \mbox{manual mathematical stress} \\ \mbox{Tyr B} \\ \mbox{Tyr B} \end{array} \qquad $
	Dear Professor, My name is Cinseer Goodman who is taking CS590 this semester. I hope this email finds you well. was not able to submit the find a xam to the server on time due to an unexpected internet connectivity loss. It just went back 5 minutes later so I send you the file via email. I promise I have not done any extra work after the exam time. I hope it works. Best, Cinseer Goodman Reply Forward



		[CS590] 5 mins late - having internet issue	
		Cinseer Goodman Tue 5/2/2021 9:05 PM To: Seunghoon Lee	
CS	S59	Final Exam - Internet Connectivity Issue	
	×	Liar King $rac{1}{2}$ 5 $rac{1}{3}$ $ ightarrow$ Tue 5/16/2021 9.45 AM $rac{1}{2}$ 5 $rac{1}{3}$ $ ightarrow$ $ ightarrow$ Tue Srunghoon Lee	
	5	answer_liar.pdf	g CS590 this semester. le server on time due to
ŝ,	(You might not believe this, but the internet went down during the final Exam since my cat accidentally chewed out my ethernet cable. I called maintenance, but the repair guy was assasinated on his way. Then the server tormado struck my town.	fter the exam time. I
	, . E,	I know it's been 2 weeks since the deadline, but this is the earliest I could send the answer to you. Please understand. I swear I haven't made any edits since the deadline. Kind regards,	
		Liar King	
		Reply Forward	



	[CS590] 5 mins late - havi	ing internet issue	
	CG Cinseer Goodman Tue 5/2/2021 9:05 PM To: Seunghoon Lee		
CS59 Final Exam - Internet Conr	nectivity Issue		
(CS590) Internet issue - for real!!	Ċ	$5 \stackrel{_{\scriptstyle \bullet}}{\longrightarrow} \cdots$	
Quantom Cheat Tue 5020201 11:30 PM To: Seunghoon Lee 5 Image: Seunghoon Lee answer_cheat.pdf Image: Sum Seunghoon Lee V	$\scriptstyle <\!$	g CS590 this semestr e server on time due ou the file via email. Ter the exam time. I able. on his way. e earliest I could	ər. to
Hello Professor, Please believe this, somehow my internet I swear I haven't touched the file after the Please receive my submission. I will upgrade my internet plan if I take you again. Beet	ne. went down!! deadline. ır course		
Quantom Cheat Reply Forward			

19



Inte	rnet pro	oblem				[CS590] 5 mins la	ate - having internet issue	
TS	[CS59	90] Help, inte CS590 fina	rnet issue!! I exam answer			CG Cinseer Go Tue 5/2/2021 To: Seunghoo	odman 9:05 PM m Lee	$c \rightarrow \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $
		cs590) internet went down		iternet Co	nnectivity Issue		
		EM FY	Fool Yoo Wed 5/32021 7:13 PM To: Seunghoon Lee answer_fool.pdf 2 MB Professor, Finally, I got my internet back. It is alread but please take my answer sheet. My mom thought I was playing a game a cable. I immediately called maintenance I can certainly prove that I haven't done a exam deadline. For real. Thank you for your consideration. Sincerely, Fool Yoo	$rightarrow 5 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	my interne le after the if I take yo	$0 {\twoheadrightarrow} \rightarrow \cdots$	ch sh sh → … down during the final internet cable, assinated on his way. this is the earliest I could ne.	g CS590 this semester. re server on time due to ou the file via email. ter the exam time. I
			Reply Forward					



(CS590) Help, internet issue!! ME CS590 final exam answer BM cs590 internet went down	Cinseer Gor Tue 5/2/2021 5 To: Seunghoo	odman :05 PM n Lee	
cs590 internet went down	ternet Connectivity Issue		
Fool Yoo Image: Source of the source of	→ wdents are ne truth?	So So So → Sown during the final atternet cable, assinated on his way. this is the earliest I could no.	g CS590 this semester. re server on time due to ou the file via email. ter the exam time. I

/19

What is a Proof of Sequential Work? (Informal)

A proof that a large amount (N) of sequential work was performed after a prover committed an initial message, e.g., the solution for the final exam



What is a Proof of Sequential Work? (Informal)

A proof that a large amount (N) of sequential work was performed after a prover committed an initial message, e.g., the solution for the final exam

Initial approach: iterative hash chain

$$\begin{split} & \textcircled{O}: \fbox{O} \rightarrow \mathcal{H}(\fbox{O}) \rightarrow \mathcal{H}^{2}(\fbox{O}) \rightarrow \mathcal{H}^{3}(\fbox{O}) \rightarrow \cdots \rightarrow \mathcal{H}^{N-1}(\fbox{O}) \rightarrow \mathcal{H}^{N}(\fbox{O}) \\ & \textcircled{O}: \fbox{O} \rightarrow \mathcal{H}(\fbox{O}) \rightarrow \mathcal{H}^{2}(\fbox{O}) \rightarrow \mathcal{H}^{3}(\textcircled{O}) \rightarrow \cdots \rightarrow \mathcal{H}^{N'-1}(\textcircled{O}) \rightarrow \mathcal{H}^{N'}(\textcircled{O}) \\ & \vdots \\ & \vdots \\ \end{split}$$



What is a Proof of Sequential Work? (Informal)

A proof that a large amount (N) of sequential work was performed after a prover committed an initial message, e.g., the solution for the final exam

Initial approach: iterative hash chain

$$\begin{array}{c} \textcircled{\odot}: \fbox{1} \mapsto \mathcal{H}(\fbox{1}) \mapsto \mathcal{H}^{2}(\fbox{1}) \mapsto \mathcal{H}^{3}(\fbox{1}) \mapsto \cdots \mapsto \mathcal{H}^{N-1}(\fbox{1}) \mapsto \mathcal{H}^{N}(\fbox{1}) \\ \textcircled{\odot}: \fbox{1} \mapsto \mathcal{H}(\fbox{1}) \mapsto \mathcal{H}^{2}(\Huge{C}) \mapsto \mathcal{H}^{3}(\Huge{C}) \mapsto \cdots \mapsto \mathcal{H}^{N'-1}(\Huge{C}) \mapsto \mathcal{H}^{N'}(\Huge{C}) \\ \vdots \end{array}$$

- Disadvantage: Instructor needs to recompute the whole thing
- Many late students? ightarrow insufficient computational resources to verify all solutions



What is a Proof of Sequential Work? (Informal)

A proof that a large amount (N) of sequential work was performed after a prover committed an initial message, e.g., the solution for the final exam

Initial approach: iterative hash chain

$$\begin{array}{c} \textcircled{\odot}: \fbox{1} \mapsto \mathcal{H}(\fbox{1}) \mapsto \mathcal{H}^{2}(\fbox{1}) \mapsto \mathcal{H}^{3}(\fbox{1}) \mapsto \cdots \mapsto \mathcal{H}^{N-1}(\fbox{1}) \mapsto \mathcal{H}^{N}(\fbox{1}) \\ \textcircled{\odot}: \fbox{1} \mapsto \mathcal{H}(\fbox{1}) \mapsto \mathcal{H}^{2}(\fbox{1}) \mapsto \mathcal{H}^{3}(\fbox{1}) \mapsto \cdots \mapsto \mathcal{H}^{N'-1}(\fbox{1}) \mapsto \mathcal{H}^{N'}(\fbox{1}) \\ \vdots \end{array}$$

- Disadvantage: Instructor needs to recompute the whole thing
- Many late students? \rightarrow insufficient computational resources to verify all solutions

Additional requirements:

• Efficiency: instructor (verifier) should be able to quickly verify the proofs (e.g., in time polylog N), and

What is a Proof of Sequential Work? (Informal)

A proof that a large amount (N) of sequential work was performed after a prover committed an initial message, e.g., the solution for the final exam

Initial approach: iterative hash chain

$$\begin{array}{l} \textcircled{O}: \fbox{I} \mapsto \mathcal{H}(\fbox{I}) \mapsto \mathcal{H}^{2}(\fbox{I}) \mapsto \mathcal{H}^{3}(\fbox{I}) \mapsto \cdots \mapsto \mathcal{H}^{N-1}(\fbox{I}) \mapsto \mathcal{H}^{N}(\fbox{I}) \\ \hline \textcircled{O}: \fbox{I} \mapsto \mathcal{H}(\fbox{I}) \mapsto \mathcal{H}^{2}(\fbox{I}) \mapsto \mathcal{H}^{3}(\fbox{I}) \mapsto \cdots \mapsto \mathcal{H}^{N'-1}(\Huge{I}) \mapsto \mathcal{H}^{N'}(\fbox{I}) \\ \hline \vdots \end{array}$$

- Disadvantage: Instructor needs to recompute the whole thing
- Many late students? \rightarrow insufficient computational resources to verify all solutions

Additional requirements:

- Efficiency: instructor (verifier) should be able to quickly verify the proofs (e.g., in time polylog N), and
- Soundness: students (prover) should *not* be able to produce a *valid* proof faster (than sequential time $\Omega(N)$, even if running in parallel).

Mahmoody et al. [MMV13]: the first theoretical construction of a PoSW

- Verifier time polylog N, and prover time $\Omega(N)$,
- Parallel cheating prover running in sequential time < N cannot fool the verifier, and
- Security proof in the classical ROM.



Mahmoody et al. [MMV13]: the first theoretical construction of a PoSW

- Verifier time polylog N, and prover time $\Omega(N)$,
- Parallel cheating prover running in sequential time < N cannot fool the verifier, and
- Security proof in the classical ROM.

Cohen and Pietrzak [CP18]: an improved & practical PoSW construction

- Modular security proof in the classical ROM:
 - $\circ~$ Any parallel cheating prover (for the PoSW) must produce a long $\mathcal{H}\text{-}sequence$ (whp), and
 - Any parallel cheating prover running in time < N cannot produce an \mathcal{H} -sequence of length N (whp).

Mahmoody et al. [MMV13]: the first theoretical construction of a PoSW

- Verifier time polylog N, and prover time $\Omega(N)$,
- Parallel cheating prover running in sequential time < N cannot fool the verifier, and
- Security proof in the classical ROM.

Cohen and Pietrzak [CP18]: an improved & practical PoSW construction

- Modular security proof in the classical ROM:
 - \circ Any parallel cheating prover (for the PoSW) must produce a long \mathcal{H} -sequence (whp), and
 - Any parallel cheating prover running in time < N cannot produce an $\frac{\mathcal{H}\text{-sequence}}{\mathcal{H}\text{-sequence}}$ of length N (whp).



Mahmoody et al. [MMV13]: the first theoretical construction of a PoSW

- Verifier time polylog N, and prover time $\Omega(N)$,
- Parallel cheating prover running in sequential time < N cannot fool the verifier, and
- Security proof in the classical ROM.

Cohen and Pietrzak [CP18]: an improved & practical PoSW construction --- Post-Quantum Security

- Modular security proof in the classical ROM:
 - $\circ~$ Any parallel cheating prover (for the PoSW) must produce a long $\mathcal H$ -sequence (whp), and
 - Any parallel cheating prover running in time < N cannot produce an $\frac{\mathcal{H}}{\mathcal{H}}$ -sequence of length N (whp).



Post-Quantum Security of the PoSW

Cohen and Pietrzak [CP18]:

- Any parallel cheating prover (for the PoSW) must produce a long $\mathcal{H}\text{-sequence},$
- Any parallel cheating prover running in time < N cannot produce an \mathcal{H} -sequence of length N.



Post-Quantum Security of the PoSW

Cohen and Pietrzak [CP18]:

- Any parallel cheating prover (for the PoSW) must produce a long $\mathcal{H}\text{-sequence},$
- Any parallel cheating prover running in time < N cannot produce an \mathcal{H} -sequence of length N.

Key Research Questions:

- Can a sequentially time-bounded parallel quantum attacker produce a long \mathcal{H} -sequence?
- Can a sequentially time-bounded parallel *quantum* attacker produce a valid PoSW?







Jeremiah Blocki, Seunghoon Lee, Samson Zhou

Post-Quantum Security of the PoSW

Cohen and Pietrzak [CP18]:

- Any parallel cheating prover (for the PoSW) must produce a long $\mathcal{H}\text{-}sequence,$
- Any parallel cheating prover running in time < N cannot produce an \mathcal{H} -sequence of length N.

Key Research Questions:

- Can a sequentially time-bounded parallel quantum attacker produce a long \mathcal{H} -sequence?
- Can a sequentially time-bounded parallel *quantum* attacker produce a valid PoSW?



Our Result. Hardness of Producing an H-Sequence/PoSW in a Quantum Setting

Theorem (informal)

A quantum adversary making at most $q \ll 2^{\lambda/3}$ queries over N-1 rounds outputs an \mathcal{H} -sequence of length N

 $(x_0, \ldots, x_N \text{ with } |x_i| \leq \delta \lambda \text{ where } \delta \geq 1)$ with negligible probability $\mathcal{O}\left(rac{q^3 \delta \lambda}{2^{\lambda}}
ight)$.

Our Result. Hardness of Producing an H-Sequence/PoSW in a Quantum Setting

Theorem (informal)

A quantum adversary making at most $q \ll 2^{\lambda/3}$ queries over N-1 rounds outputs an \mathcal{H} -sequence of length N

 $(x_0,\ldots,x_N \text{ with } |x_i| \leq \delta \lambda \text{ where } \delta \geq 1)$ with negligible probability $\mathcal{O}\left(\left. rac{q}{2\lambda} \right) \right)$

$$\mathcal{O}\left(rac{q^3\delta\lambda}{2^\lambda}
ight).$$

Theorem (informal)

Suppose \mathcal{A} makes at most $q \ll 2^{\lambda/\log N}$ quantum queries to the random oracle \mathcal{H} over at most $T = (1 - \alpha)N$ rounds. Then \mathcal{A} outputs a valid non-interactive PoSW with negligible probability $\mathcal{O}\left(q^2(1-\alpha)^{\frac{\lambda}{\log N}} + \frac{q^3\lambda\log N}{2^{\lambda}}\right)$.



Our Result. Hardness of Producing an H-Sequence/PoSW in a Quantum Setting

Theorem (informal)

A quantum adversary making at most $q \ll 2^{\lambda/3}$ queries over N - 1 rounds outputs an \mathcal{H} -sequence of length N

 $(x_0,\ldots,x_N \text{ with } |x_i| \leq \delta \lambda \text{ where } \delta \geq 1)$ with negligible probability $\mathcal{O}\left(rac{q^3\delta\lambda}{2^\lambda}
ight)$.

Theorem (informal)

Suppose \mathcal{A} makes at most $q \ll 2^{\lambda/\log N}$ quantum queries to the random oracle \mathcal{H} over at most $T = (1 - \alpha)N$ rounds. Then \mathcal{A} outputs a valid non-interactive PoSW with negligible probability $\mathcal{O}\left(q^2(1-\alpha)^{\frac{\lambda}{\log N}} + \frac{q^3\lambda\log N}{2^{\lambda}}\right)$.

- We give a direct proof for the non-interactive version of the PoSW from [CP18], and
- Our proofs are in the parallel quantum random oracle model (pqROM).

Our Result. Hardness of Producing an $\mathcal{H}\text{-}\mathsf{Sequence}/\mathsf{PoSW}$ in a Quantum Setting

Theorem (informal)

A quantum adversary making at most $q \ll 2^{\lambda/3}$ queries over N - 1 rounds outputs an \mathcal{H} -sequence of length N $(x_0, \ldots, x_N \text{ with } |x_i| \leq \delta \lambda$ where $\delta \geq 1$) with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.

Theorem (informal)

Suppose \mathcal{A} makes at most $q \ll 2^{\lambda/\log N}$ quantum queries to the random oracle \mathcal{H} over at most $T = (1 - \alpha)N$ rounds. Then \mathcal{A} outputs a valid non-interactive PoSW with negligible probability $\mathcal{O}\left(q^2(1-\alpha)^{\frac{\lambda}{\log N}} + \frac{q^3\lambda\log N}{2^{\lambda}}\right)$.

- We give a direct proof for the non-interactive version of the PoSW from [CP18], and
- Our proofs are in the parallel quantum random oracle model (pqROM).

Concurrent/Subsequent Work.

• Chung et al. [CFHL21]: also gave a comparable security bounds for the PoSW in the pqROM















- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ~$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent



- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ~$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent





- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ~$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent



- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ~$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent



- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ~$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent



- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root e
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ~$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent



- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent
- Audit process: interactive or non-interactive (Fiat-Shamir)
The [CP18] Construction



- For all leaf nodes v, add an edge (u, v) for any u that is a left sibling of a node on the path from v to the root ε
- Each node has a label, a hash of its parents
- The label of root node forms a Merkle tree commitment of all the other nodes
 - $\circ\;$ Verifier can audit the prover by forcing the prover to open certain labels
 - Show that they are locally consistent
- Audit process: interactive or non-interactive (Fiat-Shamir)
- Any classical ROM attacker that produces a valid PoSW in time < N must produce a long $\mathcal H$ -sequence

ROM vs qROM [BDF $^+$ 11]

<Classical ROM>



<Quantum ROM>





ROM vs qROM [BDF $^+$ 11]

<Classical ROM>



 $\langle \text{Quantum ROM} \rangle$ $\sum_{x} \alpha_{x} |x\rangle$ $\sum_{x} \alpha_{x} |x, \mathcal{H}(x)\rangle$

- Security proofs are much more challenging in the qROM
 - Programmability & Extractability (ROM: ✔, qROM: ✗)
 - Recording quantum queries?



ROM vs qROM [BDF $^+$ 11]

 $(Classical ROM) \times (Quantum R$

- Security proofs are much more challenging in the qROM
 - Programmability & Extractability (ROM: ✔, qROM: ✗)
 - Recording quantum queries?
- Compressed Oracle Technique [Zha19]: change of view (compressed phase oracle (CPhsO))

$$egin{aligned} &|x,y
angle \otimes |\mathcal{H}
angle \mapsto |x,y\oplus\mathcal{H}(x)
angle \otimes |\mathcal{H}
angle \ &1 \ &1 \ &\|x,y
angle \otimes |\mathcal{H}
angle \mapsto (-1)^{y\cdot\mathcal{H}(x)}|x,y
angle \otimes |\mathcal{H}
angle \end{aligned}$$



19

A database
$$\mathcal{D} := \{(x_i, y_i), i \geq 1\}$$
, where $\mathcal{D}(x_i) = y_i$.

- Classical: databases of known I/O pairs & unknown I/O pairs don't appear
- Quantum: superposition over databases (known I/O pairs + indeterminates)



A database
$$\mathcal{D} := \{(x_i, y_i), i \geq 1\}$$
, where $\mathcal{D}(x_i) = y_i$.

How to view a random oracle?

- Classical: databases of known I/O pairs & unknown I/O pairs don't appear
- Quantum: superposition over databases (known I/O pairs + indeterminates)

After q queries,

The state can be viewed as

$$\sum_{x,y,z,\mathcal{D}} \alpha_{x,y,z,\mathcal{D}} | \begin{array}{c} x,y \\ \end{array}, \\ \end{array} \rangle \otimes | \begin{array}{c} | \begin{array}{c} \mathcal{D} \rangle \\ \end{array} \rangle,$$

where

A database
$$\mathcal{D} := \{(x_i, y_i), i \geq 1\}$$
, where $\mathcal{D}(x_i) = y_i$.

- Classical: databases of known I/O pairs & unknown I/O pairs don't appear
- Quantum: superposition over databases (known I/O pairs + indeterminates)





A database
$$\mathcal{D} := \{(x_i, y_i), i \geq 1\}$$
, where $\mathcal{D}(x_i) = y_i$.

- Classical: databases of known I/O pairs & unknown I/O pairs don't appear
- Quantum: superposition over databases (known I/O pairs + indeterminates)





A database
$$\mathcal{D} := \{(x_i, y_i), i \geq 1\}$$
, where $\mathcal{D}(x_i) = y_i$.

- Classical: databases of known I/O pairs & unknown I/O pairs don't appear
- Quantum: superposition over databases (known I/O pairs + indeterminates)







Example: Single Query (simplest case)

$$|x, y, z\rangle \otimes |\mathcal{D}\rangle \stackrel{\mathsf{CPhsO}}{\underset{(x,y) \not\in \mathcal{D}}{\mapsto}} |x, y, z\rangle \otimes \sum_{w} (-1)^{y \cdot w} |\mathcal{D} \cup (x, w)\rangle.$$

w ranges over



Example: Single Query (simplest case)

$$|x, y, z\rangle \otimes |\mathcal{D}\rangle \xrightarrow[(x, y) \notin \mathcal{D}]{} |x, y, z\rangle \otimes \sum_{w} (-1)^{y \cdot w} |\mathcal{D} \cup (x, w)\rangle.$$
all possible outputs of $\mathcal{H}(x)$.

• w ranges over all possible outputs of $\mathcal{H}(x)$. –



Example: Single Query (simplest case)

$$|x,y,z
angle\otimes|\mathcal{D}
angle \stackrel{ ext{CPhsO}}{\stackrel{(x,y)
ot\in\mathcal{D}}{\rightarrow}}|x,y,z
angle\otimes\sum_{w}(-1)^{y\cdot w}|\mathcal{D}\cup(x,w)
angle.$$

• w ranges over all possible outputs of $\mathcal{H}(x)$. —

Example: Parallel Query (simplest case)

$$ert (x_1, y_1), \dots, (x_k, y_k), z
angle \otimes ert \mathcal{D}
angle \ \stackrel{ ext{CPhsO}^k}{\longmapsto} ert (x_1, y_1), \dots, (x_k, y_k) ert, z
angle \otimes \sum_{oldsymbol{w_1}, \dots, oldsymbol{w_k}} (-1)^{\sum_{i=1}^k x_i \cdot w_i} ert \mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}
angle,$$

where

Example: Single Query (simplest case)

$$|x,y,z\rangle\otimes |\mathcal{D}
angle \stackrel{ ext{CPhsO}}{\stackrel{(x,y)
ot\in\mathcal{D}}{\rightarrow}} |x,y,z\rangle\otimes \sum_{w}(-1)^{y\cdot w}|\mathcal{D}\cup(x,w)
angle.$$

• w ranges over all possible outputs of $\mathcal{H}(x)$. —

Example: Parallel Query (simplest case)

$$|(x_{1}, y_{1}), \dots, (x_{k}, y_{k}), z\rangle \otimes |\mathcal{D}\rangle$$

$$\xrightarrow{\text{CPhsO}^{k}} |(x_{1}, y_{1}), \dots, (x_{k}, y_{k}), z\rangle \otimes \sum_{w_{1}, \dots, w_{k}} (-1)^{\sum_{i=1}^{k} x_{i} \cdot w_{i}} |\mathcal{D} \cup \{(x_{1}, w_{1}), \dots, (x_{k}, w_{k})\}\rangle,$$
where
$$\bullet \text{ simplest case: } (x_{1}, y_{1}), \dots, (x_{k}, y_{k}) \notin \mathcal{D} \text{ and all } (x_{i}, y_{i})$$
's are distinct, and

V

Example: Single Query (simplest case)

$$|x,y,z
angle\otimes |\mathcal{D}
angle \xrightarrow{ ext{CPhsO}} |x,y,z
angle\otimes \sum_{(x,y)
otin \mathcal{D}} (-1)^{y\cdot w} |\mathcal{D}\cup(x,w)
angle.$$

• w ranges over all possible outputs of $\mathcal{H}(x)$. —

Example: Parallel Query (simplest case)

$$|(x_{1}, y_{1}), \dots, (x_{k}, y_{k}), z\rangle \otimes |\mathcal{D}\rangle$$

$$\xrightarrow{\text{CPhsO}^{k}} |(x_{1}, y_{1}), \dots, (x_{k}, y_{k}), z\rangle \otimes \sum_{w_{1}, \dots, w_{k}} (-1)^{\sum_{i=1}^{k} x_{i} \cdot w_{i}} |\mathcal{D} \cup \{(x_{1}, w_{1}), \dots, (x_{k}, w_{k})\}\rangle,$$
where
$$\text{simplest case: } (x_{1}, y_{1}), \dots, (x_{k}, y_{k}) \notin \mathcal{D} \text{ and all } (x_{i}, y_{i})$$
's are distinct, and
$$\text{w}_{i}$$
's range over all possible outputs of $\mathcal{H}(x_{i})$'s for each i .

V

Notations

• Given a database $\mathcal{D} = \{(x_1, y_1), \dots, (x_q, y_q)\}$, define a directed graph $G_{\mathcal{D}}$ on q nodes $(v_{x_1}, \dots, v_{x_q})$ such that:



- $\mathsf{PATH}_s := \{\mathcal{D} : G_\mathcal{D} \text{ contains a path of length } s\}$ (set of databases), and
- $\widetilde{\mathsf{PATH}}_s := \{ | (x_1, y_1), \dots, (x_k, y_k), z \rangle \otimes | \mathcal{D} \rangle : \mathcal{D} \in \mathsf{PATH}_s \}$ (set of basis states).



Notations

• Given a database $\mathcal{D} = \{(x_1, y_1), \dots, (x_q, y_q)\}$, define a directed graph $G_{\mathcal{D}}$ on q nodes $(v_{x_1}, \dots, v_{x_q})$ such that:



- PATH_s := { \mathcal{D} : $G_{\mathcal{D}}$ contains a path of length s} (set of databases), and
- $\widetilde{\mathsf{PATH}}_s := \{ | (x_1, y_1), \dots, (x_k, y_k), z \rangle \otimes | \mathcal{D} \rangle : \mathcal{D} \in \mathsf{PATH}_s \}$ (set of basis states).





Lemma

$$|\varphi\rangle$$
: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Interpretation/Intuition:

• $L_2(|\varphi\rangle, \widetilde{\mathsf{PATH}}_s)$: 2-norm of the projection of $|\varphi\rangle$ onto $\widetilde{\mathsf{PATH}}_s$, i.e.,

$$|arphi
angle = \sum_X lpha_X |X
angle \quad \Rightarrow \quad L_2(|arphi
angle, \widetilde{\mathsf{PATH}}_s) = \sqrt{\sum_{|X
angle\in\widetilde{\mathsf{PATH}}_s} |lpha_X|^2}.$$

• If we start with the state that is nearly orthogonal to $\widetilde{\mathsf{PATH}}_{s}$, then after applying the oracle CPhsO^k, the resulting state is also nearly orthogonal to $\widetilde{\mathsf{PATH}}_{s+1}$.



Lemma

 $|\varphi\rangle$: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Basic proof idea: split the states into good and bad part. (in this talk, suppose that $x_1, \ldots, x_k \notin D$ and all distinct for simplicity)

 $|arphi
angle = |(x_1,y_1),\ldots,(x_k,y_k),z
angle \otimes |\mathcal{D}
angle$



Lemma

$$|\varphi\rangle$$
: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Basic proof idea: split the states into good and bad part. (in this talk, suppose that $x_1, \ldots, x_k \notin D$ and all distinct for simplicity)

$$egin{aligned} &|arphi
angle = |(x_1,y_1),\ldots,(x_k,y_k),z
angle \otimes |\mathcal{D}
angle \ &\stackrel{ ext{CPhsO}^k}{\longmapsto} |arphi'
angle = |(x_1,y_1),\ldots,(x_k,y_k),z
angle \otimes \sum_{w_1,\ldots,w_k} (-1)^{\sum y_iw_i} |\mathcal{D}\cup\{(x_1,w_1),\ldots,(x_k,w_k)\}
angle \end{aligned}$$



Lemma

$$|\varphi\rangle$$
: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Basic proof idea: split the states into good and bad part. (in this talk, suppose that $x_1, \ldots, x_k \notin D$ and all distinct for simplicity)

$$\begin{aligned} |\varphi\rangle &= |(x_1, y_1), \dots, (x_k, y_k), z\rangle \otimes |\mathcal{D}\rangle \\ & \stackrel{\text{CPhsO}^k}{\longmapsto} |\varphi'\rangle = |(x_1, y_1), \dots, (x_k, y_k), z\rangle \otimes \sum_{\substack{w_1, \dots, w_k \\ \in \text{GOOD}}} (-1)^{\sum y_i w_i} |\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}\rangle \\ & + \sum_{\substack{w_1, \dots, w_k \\ \in \text{GOOD}}} (-1)^{\sum y_i w_i} |\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}\rangle \\ & + \sum_{\substack{w_1, \dots, w_k \\ \in \text{BAD}}} (-1)^{\sum y_i w_i} |\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}\rangle \end{aligned}$$



Lemma

$$|\varphi\rangle$$
: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Basic proof idea: split the states into good and bad part. (in this talk, suppose that $x_1, \ldots, x_k \notin D$ and all distinct for simplicity)

$$\begin{aligned} |\varphi\rangle &= |(x_1, y_1), \dots, (x_k, y_k), z\rangle \otimes |\mathcal{D}\rangle \\ & \stackrel{\text{CPhsO}^k}{\longmapsto} |\varphi'\rangle = |(x_1, y_1), \dots, (x_k, y_k), z\rangle \otimes \sum_{\substack{w_1, \dots, w_k \\ \in \text{GOOD}}} (-1)^{\sum y_i w_i} |\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}\rangle \\ + \left| \begin{array}{c} |(x_1, y_1), \dots, (x_k, y_k), z\rangle \\ \otimes \sum_{\substack{w_1, \dots, w_k \\ \in \text{GOOD}}} (-1)^{\sum y_i w_i} |\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}\rangle \\ \end{array} \right| \\ + \left| \begin{array}{c} |(x_1, y_1), \dots, (x_k, y_k), z\rangle \\ \otimes \sum_{\substack{w_1, \dots, w_k \\ \in \text{BAD}}} (-1)^{\sum y_i w_i} |\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\}\rangle \\ \end{array} \right| \\ \end{aligned}$$

BAD: $\mathcal{D} \not\in \mathsf{PATH}_s$ but $\mathcal{D} \cup \{(x_1, w_1), \dots, (x_k, w_k)\} \in \mathsf{PATH}_{s+1}$



Lemma

$$|\varphi\rangle$$
: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Proof by example:





Lemma

$$|\varphi\rangle$$
: an initial state, and let $|\varphi'\rangle = \mathsf{CPhsO}^{k}|\varphi\rangle$. Then $L_{2}(|\varphi'\rangle, \widetilde{\mathsf{PATH}}_{s+1}) - L_{2}(|\varphi\rangle, \widetilde{\mathsf{PATH}}_{s}) \leq \frac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$.

Proof by example:



19

Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_1 = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (00001, w) \} \}$



Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_1 = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (00001, w) \} \}$





Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_1 = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (00001, w) \} \}$





Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_1 = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (00001, w) \} \}$





Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_{1} = \{(10101, 0001), (00011, 0010), (00010, 0110), (01101, 0000), (11110, 0011), (01011, 1101)\} \cup \{(00001, w)\}$





Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_1 = \{(10101, 0001), (00011, 0010), (00010, 0110), (01101, 0000), (11110, 0011), (01011, 1101)\} \cup \{(00001, w)\}$





Suppose we have one query: $x_7 = 00001$ (where $x_7 \notin D$). Then the updated database is:

 $\mathcal{D}_{1} = \{(10101, 0001), (00011, 0010), (00010, 0110), (01101, 0000), (11110, 0011), (01011, 1101)\} \cup \{(00001, w)\}$





Jeremiah Blocki, Seunghoon Lee, Samson Zhou

For a parallel query: x_7, \ldots, x_{k+6} (where $x_7, \ldots, x_{k+6} \notin D$ and all x_i 's are distinct for simplicity),

 $\mathcal{D}_{k} = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (x_{7}, w_{7}), \dots, (x_{k+6}, w_{k+6}) \} \}$





For a parallel query: x_7, \ldots, x_{k+6} (where $x_7, \ldots, x_{k+6} \notin D$ and all x_i 's are distinct for simplicity),

 $\mathcal{D}_{k} = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (x_{7}, w_{7}), \dots, (x_{k+6}, w_{k+6}) \} \}$



 $\Rightarrow \mathcal{D} \not\in \mathsf{PATH}_4 \text{ but} \\ \mathcal{D}_k \in \mathsf{PATH}_5!!$

For a parallel query: x_7, \ldots, x_{k+6} (where $x_7, \ldots, x_{k+6} \notin D$ and all x_i 's are distinct for simplicity),

 $\mathcal{D}_{k} = \{ (10101, 0001), (00011, 0010), (00010, 0110) \\, (01101, 0000), (11110, 0011), (01011, 1101) \} \cup \{ (x_{7}, w_{7}), \dots, (x_{k+6}, w_{k+6}) \} \}$



Key observation: The fraction of such w_7, \ldots, w_{k+6} 's is negligibly small! $((q+k)\delta\lambda)$ out of 2^{λ} for each w_i)

¹⁶/₁₉

We have shown: k parallel queries in a single round,

$$L_2(|arphi'
angle,\widetilde{\mathsf{PATH}}_{s+1})-L_2(|arphi
angle,\widetilde{\mathsf{PATH}}_s)\leq rac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$$



We have shown: k parallel queries in a single round,

$$L_2(|arphi'
angle,\widetilde{\mathsf{PATH}}_{s+1}) - L_2(|arphi
angle,\widetilde{\mathsf{PATH}}_s) \leq rac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$$

<u>Throughout N - 1 rounds</u>: k_i parallel queries in each round, with $\sum_{i=1}^{N-1} k_i \leq q$.


Proof Ideas: Hardness of Producing an \mathcal{H} -sequence in a Quantum Setting

We have shown: k parallel queries in a single round,

$$L_2(|arphi'
angle,\widetilde{\mathsf{PATH}}_{s+1}) - L_2(|arphi
angle,\widetilde{\mathsf{PATH}}_s) \leq rac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$$

<u>Throughout N - 1 rounds</u>: k_i parallel queries in each round, with $\sum_{i=1}^{N-1} k_i \leq q$.

• By triangle inequality,

$$L_2(|\varphi_{N-1}\rangle, \widetilde{\mathsf{PATH}}_N) \leq \sum_{i=1}^{N-1} \frac{4k_i \sqrt{2q\delta\lambda}}{2^{\lambda/2}} = \mathcal{O}\left(rac{\sqrt{q^3\delta\lambda}}{2^{\lambda/2}}
ight).$$

 $\Rightarrow \mathcal{A}$ measures a database in PATH_N with probability at most $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$,

i.e., \mathcal{A} can produce an \mathcal{H} -sequence of length N with only negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.



Proof Ideas: Hardness of Producing an \mathcal{H} -sequence in a Quantum Setting

We have shown: k parallel queries in a single round,

$$L_2(|arphi'
angle,\widetilde{\mathsf{PATH}}_{s+1}) - L_2(|arphi
angle,\widetilde{\mathsf{PATH}}_s) \leq rac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$$

Throughout N - 1 rounds: k_i parallel queries in each round, with $\sum_{i=1}^{N-1} k_i \leq q$.

• By triangle inequality,

$$L_2(|arphi_{N-1}
angle,\widetilde{\mathsf{PATH}}_N) \leq \sum_{i=1}^{N-1} rac{4k_i\sqrt{2q\delta\lambda}}{2^{\lambda/2}} = \mathcal{O}\left(rac{\sqrt{q^3\delta\lambda}}{2^{\lambda/2}}
ight).$$

 $\Rightarrow \mathcal{A}$ measures a database in PATH_N with probability at most $\mathcal{O}\left(rac{q^3\delta\lambda}{2^{\lambda}}
ight)$,

i.e., \mathcal{A} can produce an \mathcal{H} -sequence of length N with only negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.

• We only considered the simplest case in this talk - see the paper for the full security proof!

Proof Ideas: Hardness of Producing an \mathcal{H} -sequence in a Quantum Setting

We have shown: k parallel queries in a single round,

$$L_2(|arphi'
angle,\widetilde{\mathsf{PATH}}_{s+1}) - L_2(|arphi
angle,\widetilde{\mathsf{PATH}}_s) \leq rac{4k\sqrt{(q+k)\delta\lambda}}{2^{\lambda/2}}$$

<u>Throughout N - 1 rounds</u>: k_i parallel queries in each round, with $\sum_{i=1}^{N-1} k_i \leq q$.

• By triangle inequality,

$$L_2(|\varphi_{N-1}\rangle, \widetilde{\mathsf{PATH}}_N) \leq \sum_{i=1}^{N-1} \frac{4k_i \sqrt{2q\delta\lambda}}{2^{\lambda/2}} = \mathcal{O}\left(rac{\sqrt{q^3\delta\lambda}}{2^{\lambda/2}}
ight).$$

 $\Rightarrow \mathcal{A}$ measures a database in PATH_N with probability at most $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$,

i.e., \mathcal{A} can produce an \mathcal{H} -sequence of length N with only negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.

• We only considered the simplest case in this talk - see the paper for the full security proof!

Security of a non-interactive PoSW: similar argument using the result above - details in the paper (https://arxiv.org/pdf/2006.10972.pdf)

Takeaways.

• PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.

Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the <code>pqROM</code> making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an

 $\mathcal H$ -sequence of length N except with negligible probability $\mathcal O\left(rac{q^3\delta\lambda}{2^\lambda}
ight)$.

Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the pqROM making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an \mathcal{H} -sequence of length N except with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.
- Any attacker in the pqROM making $q \ll 2^{\lambda/\log N}$ total queries in sequential time $T = (1 \alpha)N$ cannot produce a valid non-interactive PoSW ([CP18] construction) except with negligible probability

$$\mathcal{O}\left(q^2(1-lpha)^{rac{\lambda}{\log N}}+rac{q^3\lambda\log N}{2^\lambda}
ight)$$

Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the pgROM making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an \mathcal{H} -sequence of length N except with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.
- Any attacker in the pqROM making $q \ll 2^{\lambda/\log N}$ total queries in sequential time $T = (1 \alpha)N$ cannot produce a valid non-interactive PoSW ([CP18] construction) except with negligible probability

$$\mathcal{O}\left(q^2(1-lpha)rac{\lambda}{\log N}+rac{q^3\lambda\log N}{2^\lambda}
ight).$$

Open Questions.

`-----... • Can we tighten the security bound from q^3 to q^2 ?



Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the pqROM making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an \mathcal{H} -sequence of length N except with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2\lambda}\right)$.
- Any attacker in the pqROM making $q \ll 2^{\lambda/\log N}$ total queries in sequential time $T = (1 \alpha)N$ cannot produce a valid non-interactive PoSW ([CP18] construction) except with negligible probability

$$\mathcal{O}\left(q^2(1-lpha)^{rac{\lambda}{\log N}}+rac{q^3\lambda\log N}{2^\lambda}
ight)$$

Open Questions.

• Can we tighten the security bound from q^3 to q^2 ?

• Establishing security for larger q: can we extract more than $\lambda/\log N$ challenges from a single RO output?

Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the pqROM making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an \mathcal{H} -sequence of length N except with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.
- Any attacker in the pqROM making $q \ll 2^{\lambda/\log N}$ total queries in sequential time $T = (1 \alpha)N$ cannot produce a valid non-interactive PoSW ([CP18] construction) except with negligible probability

$$\mathcal{O}\left(q^2(1-lpha)^{rac{\lambda}{\log N}}+rac{q^3\lambda\log N}{2^\lambda}
ight)$$
 .

Open Questions.

- Can we tighten the security bound from q^3 to q^2 ?
- Establishing security for larger q: can we extract more than $\lambda/\log N$ challenges from a single RO output?
- Can we prove for an interactive PoSW?



Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the pqROM making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an \mathcal{H} -sequence of length N except with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.
- Any attacker in the pqROM making $q \ll 2^{\lambda/\log N}$ total queries in sequential time $T = (1 \alpha)N$ cannot produce a valid non-interactive PoSW ([CP18] construction) except with negligible probability

$$\mathcal{O}\left(q^2(1-lpha)^{rac{\lambda}{\log N}}+rac{q^3\lambda\log N}{2^\lambda}
ight)$$
 .

Open Questions.

- Can we tighten the security bound from q^3 to q^2 ?
- Establishing security for larger q: can we extract more than $\lambda/\log N$ challenges from a single RO output?
- Can we prove for an interactive PoSW?
- Can we extend our security proof for other PoSW constructions?

Takeaways.

- PoSW allows a prover to convince a resource-bounded verifier that the prover invested a substantial amount of sequential time to produce a valid proof.
- Any attacker in the pqROM making $q \ll 2^{\lambda/3}$ total queries in N-1 sequential rounds cannot find an \mathcal{H} -sequence of length N except with negligible probability $\mathcal{O}\left(\frac{q^3\delta\lambda}{2^{\lambda}}\right)$.
- Any attacker in the pqROM making $q \ll 2^{\lambda/\log N}$ total queries in sequential time $T = (1 \alpha)N$ cannot produce a valid non-interactive PoSW ([CP18] construction) except with negligible probability

$$\mathcal{O}\left(q^2(1-lpha)^{rac{\lambda}{\log N}}+rac{q^3\lambda\log N}{2^\lambda}
ight)$$
 .

Open Questions.

- Can we tighten the security bound from q^3 to q^2 ?
- Establishing security for larger q: can we extract more than $\lambda/\log N$ challenges from a single RO output?
- Can we prove for an interactive PoSW?
- Can we extend our security proof for other PoSW constructions?
- Can techniques extend to other primitives, e.g., Proofs of Space, Memory-Hard Functions, etc.?

References I

- Dan Boneh, Özgür Dagdelen, Marc Fischlin, Anja Lehmann, Christian Schaffner, and Mark Zhandry, Random oracles in a quantum world, ASIACRYPT 2011 (Dong Hoon Lee and Xiaoyun Wang, eds.), LNCS, vol. 7073, Springer, Heidelberg, December 2011, pp. 41–69.
- Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao, On the compressed-oracle technique, and post-quantum security of proofs of sequential work, 2021.
- Bram Cohen and Krzysztof Pietrzak, *Simple proofs of sequential work*, EUROCRYPT 2018, Part II (Jesper Buus Nielsen and Vincent Rijmen, eds.), LNCS, vol. 10821, Springer, Heidelberg, April / May 2018, pp. 451–467.
- Mohammad Mahmoody, Tal Moran, and Salil P. Vadhan, *Publicly verifiable proofs of sequential work*, ITCS 2013 (Robert D. Kleinberg, ed.), ACM, January 2013, pp. 373–388.
- Mark Zhandry, *How to record quantum queries, and applications to quantum indifferentiability*, CRYPTO 2019, Part II (Alexandra Boldyreva and Daniele Micciancio, eds.), LNCS, vol. 11693, Springer, Heidelberg, August 2019, pp. 239–268.

