Strong Batching for Non-Interactive Statistical Zero-Knowledge

Mu Changrui (NUS), Shafik Nassar (UT Austin),
Ron D. Rothblum (Technion),
Prashant Nalini Vasudevan (NUS)

May 29, 2024

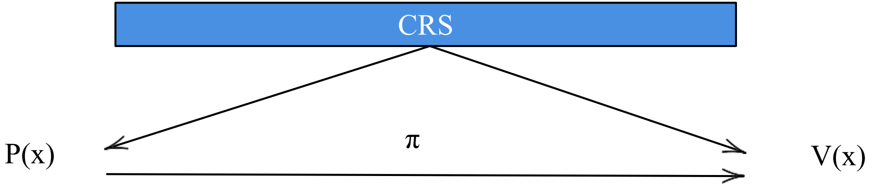
Non-Interactive Statistical Zero-Knowledge Proof

- Zero-knowledge proofs [GMR89] are amazing
  - Prove without revealing additional information beyond validity.

↓

- Non-interactive Zero-knowledge proofs [BFM88]
  - Common Random String (CRS model)
  - Non-interactive (Prover sends one message)

Non-Interactive Statistical Zero-Knowledge Proof



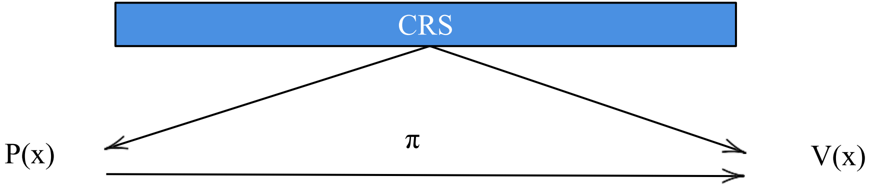
- Completeness: if x ∈ YES ⇒ Pr[V Accepts] $\geq 1 - \text{negl}$
- Soundness: if x ∈ NO ⇒ $\forall P^*$, Pr[V Accepts] $\leq \text{negl}$

- Zero-knowledge:

  ∃ PPT Sim s.t. for any x ∈ YES

  $$(\text{CRS}, \pi) \approx \text{Sim}(x)$$

Non-Interactive Statistical Zero-Knowledge Proof



- Completeness: if $x \in$ YES $\Rightarrow$ Pr[V Accepts] $\geq 1 - \text{negl}$
- Soundness: if $x \in$ NO $\Rightarrow \forall P^*$, Pr[V Accepts] $\leq \text{negl}$

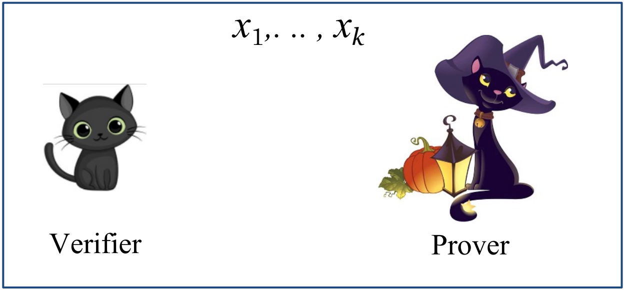
- Statistical Zero-knowledge:

  ∃ PPT Sim s.t. fo

  $$(\text{CRS}, \pi) \approx_s \text{Sim}(x)$$

NISZK: Problems that have NISZK protocol

Batch Verification: Check k instances



Check $x_1, \ldots, x_k$ are all YES instances

- Accept if $x_1, \ldots, x_k$ are all YES instances
- Reject if at least one x_i is NO instance

Batch Verification: Naive Solution

Verify $x_1, \ldots, x_k \in$ YES

m bits for x_1

$\longleftrightarrow$

m bits for x_2

$\longleftrightarrow$

...

m bits for x_k

$\longleftrightarrow$

Verifier

Prover

$k \cdot m$ communication

Non-Trivial Batch Verification

Communication and Round Complexity

Verify one instance:

- m Communication
- r Randomness (CRS)
- t Rounds

$\Rightarrow$

Verify k instances:

- less than $m \cdot k$ Communication
- less than $r \cdot k$ Randomness (CRS)
- less than $t \cdot k$ Rounds

Which classes of problems have non-trivial batching?

Which class of problems have non-trivial batching?

- Batching for **IP** via IP = PSPACE [LFKN92; Sha92]
  - Lose efficiency of prover.
  - Lose zero-knowledge.

Which class of problems have non-trivial batching?

- Batching for **IP** via IP = PSPACE [LFKN92; Sha92]
  - Lose efficiency of prover.
  - Lose zero-knowledge.
- Preserve prover efficiency.
  - Batching for UP with efficient prover [RRR16; RRR18; RR20]
  - Batching for **NP** with computational soundness [BHK17; CJJ21a; CJJ21b,...]

Which class of problems have non-trivial batching?

- Batching for **IP** via IP = PSPACE [LFKN92; Sha92]
  - Lose efficiency of prover.
  - Lose zero-knowledge.
- Preserve prover efficiency.
  - Batching for UP with efficient prover [RRR16; RRR18; RR20]
  - Batching for **NP** with computational soundness [BHK17; CJJ21a; CJJ21b,…]

This work: Preserve zero-knowledge.

Batching NISZK

Set Problems that have SZK (interactive)

Main Question: suppose $\Pi \in$ SZK, can we verify that $x_1, \ldots, x_k \in$ YES(Π) with **non-trivial batching** in **zero-knowledge**?

- [KRRSV20][KRV21]:

  $\Pi \in$ NISZK $\Rightarrow$ Batching SZK (interactive), $k + poly(n)$ communication

- Our result:

  $\Pi \in$ NISZK $\Rightarrow$ Batching NISZK (non-interactive),

  $poly(\log k, n)$ communication and CRS length

Our Result

Main Theorem:

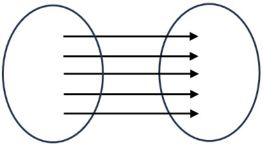
Every problem $\Pi \in$ NISZK has a non-interactive-SZK batch verification protocol with $poly(n, \log k)$ communication and CRS length.

Overview

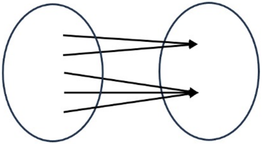
- Background and Bottlenecks.
- Our Solution:
  - Key Observation.
  - New Protocol.
- Open Questions.

Warm-up: Batching for Permutation (*PERM*)

- Input: length-preserving circuit $C : \{0,1\}^n \rightarrow \{0,1\}^n$



YES case: C defines a Permutation.

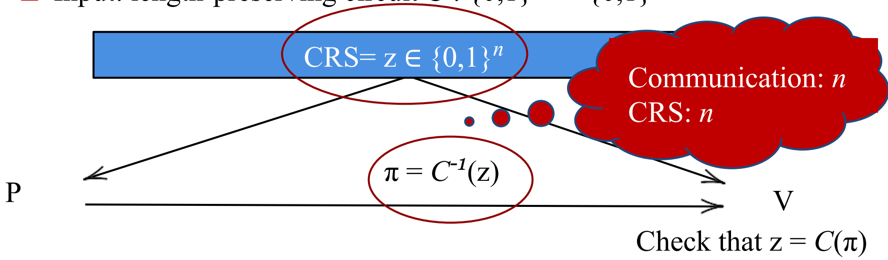


NO case: every image has at least two preimages.

PERM has NISZK protocol.

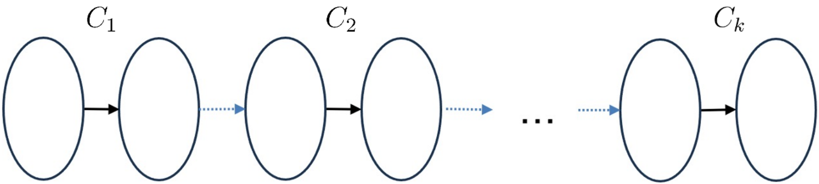
Warm-up: NISZK for Permutation (*PERM*)

- Input: length-preserving circuit $C : \{0,1\}^n \to \{0,1\}^n$



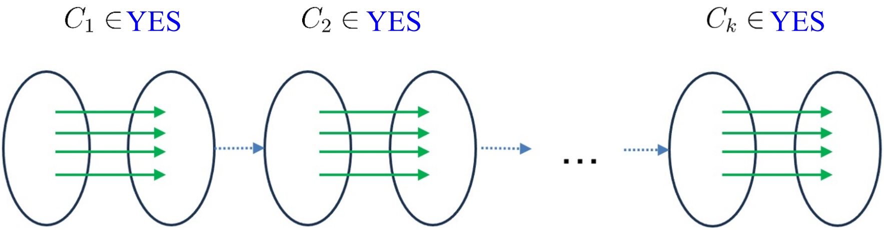
- Completeness: perfect!
- Soundness: NO case, random z doesn't have a preimage with probability at least ½.
- ZK: simulator samples x and output (crs = C(x), π = x).
  - Perfect Zero-Knowledge.

NISZK Batching for PERM



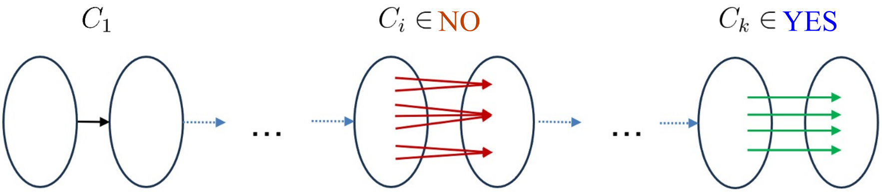
$$\bar{C}_k = C_k \circ \cdots \circ C_2 \circ C_1 \in \text{PERM}$$

Yes Cases: NISZK Batching for PERM



$$\bar{C}_k = C_k \circ \cdots \circ C_2 \circ C_1 \in \text{YES}(PERM)$$

No Cases: NISZK Batch Verification for PERM



$$\bar{C}_k = C_k \circ \cdots \circ C_2 \circ C_1 \in \text{NO(PERM)}$$

Batching NISZK for PERM

Are we done? :) thank you!

PERM is not known to be **NISZK-hard**

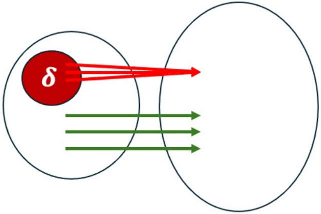
- CRS : [CRS for NISZK of PERM]
- Protocol:
  1. Construct $\bar{C}_k = C_k \circ \cdots \circ C_2 \circ C_1$
  2. Runs NISZK protocol for one instance of PERM

Communication: *n*

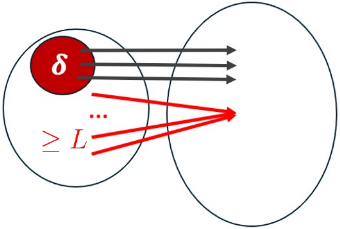
CRS Length: *n*

NISZK-Complete: Approximate Injectivity ($AI_{\delta,L}$)

Input: circuit $C : \{0,1\}^n \rightarrow \{0,1\}^t$ $t \geq n$



C is YES($AI_{\delta,L}$) if it is injective on all but δ-fraction of inputs

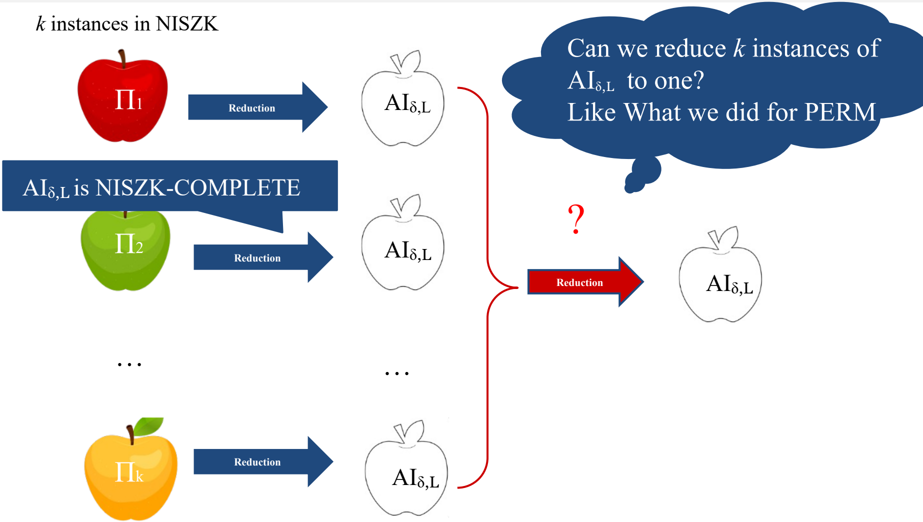


C is NO($AI_{\delta,L}$) if it is L-to-1 on all but δ-fraction of inputs

[KRRSV20]: $AI_{\delta,L}$ is NISZK-complete for $L(n) < 2^{n^{0.1}}$, $\delta > 2^{-n^{0.1}}$

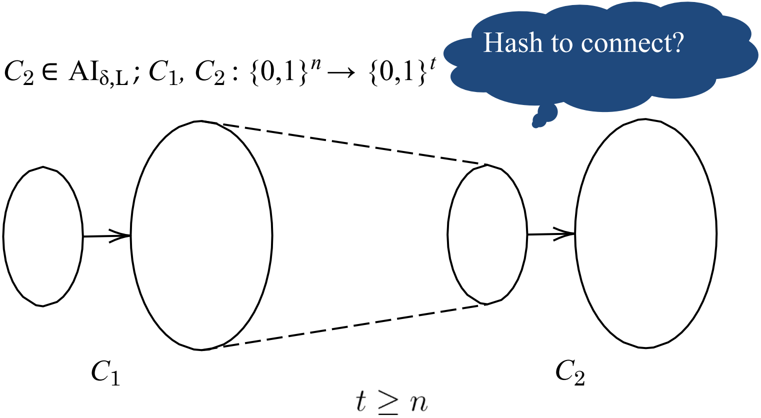
Distinguish *almost injective* from *very non-injective*

Bird's Eye View



Batching $\mathrm{AI}_{\delta,\mathrm{L}}$

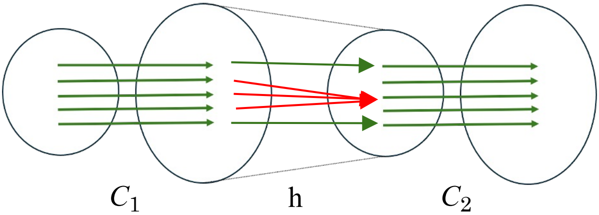
$C_1, C_2 \in \mathrm{AI}_{\delta,\mathrm{L}}$; $C_1, C_2 : \{0,1\}^n \to \{0,1\}^t$



Circuit is not length-preserving

Batching $\mathrm{AI}_{\delta,\mathrm{L}}$

C_1 , $C_2 \in$ YES($\mathrm{AI}_{0,\mathrm{L}}$)

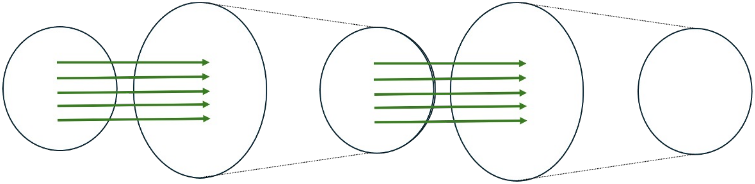


Compose with hash.

- ❑ Injectivity will not maintain even after one composition:(
  - [KRRSV20][KRV21]: resolve collision through interaction (Linearly dependent on k; and is interactive).

This work: Collision Probability is preserved!

Preserving Collision Probability

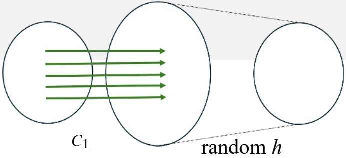


- *The distribution defined by* C_1 *has low collision probability:*

$$cp(C_1) = \Pr_{x_1, x_2 \leftarrow \{0,1\}^n}[C_1(x_1) = C_1(x_2)] = \frac{1}{2^n}$$

- *Will* $cp(h \circ C_1)$ *be much larger?*

Preserving Collision Probability



Bound $\mathop{E}_{h}\left[cp\left(h\circ C_1\right)\right]$

$$= \mathop{E}_{h}\left[\mathop{Pr}_{x_1,x_2}\left(h\circ C_1(x_1) = h\circ C_1(x_2)\right)\right]$$

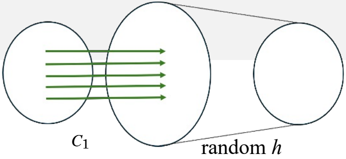
$$\le \underbrace{\mathop{E}_{h}\left[\mathop{Pr}_{x_1,x_2}\left(C_1(x_1) = C_1(x_2)\right)\right]}_{\text{Collision from } C_1} + \underbrace{\mathop{E}_{h}\left[\mathop{Pr}_{x_1,x_2}\left[h\circ C_1(x_1) = h\circ C_1(x_2) \middle| C_1(x_1) \ne C_1(x_2)\right]\right]}_{\text{Collision from } h}$$

$\mathrm{cp}(C_1)$

$\frac{1}{2^n}$, because h is random

Expected cp grows only by a tiny amount

Preserving Collision Probability



$$\mathop{E}_{h}\left[cp\left(h \circ C_1\right)\right] \le cp\left(C_1\right) + \frac{1}{2^n}$$

Similarly, bound

$$\mathop{Var}_{h}\left[cp\left(h \circ C_1\right)\right] \le O\left(\frac{cp\left(C_1\right)^2}{2^n}\right)$$

Apply Chebyshev

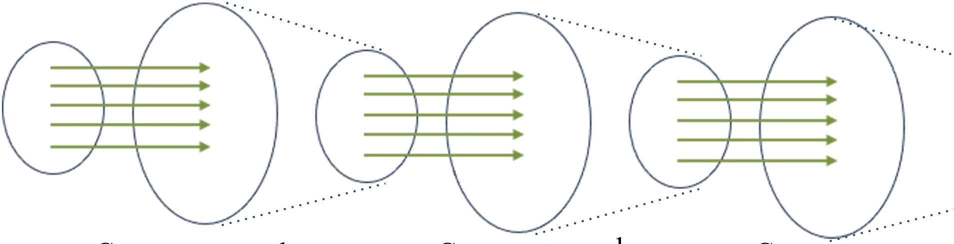
$$cp(h \circ C_1) \le cp(C_1) + \frac{2}{2^n}$$

with probability $1 - O(\frac{1}{2^n})$

Collision probability only increases slightly w.h.p

Preserving Collision Probability

$$\bar{C}_k \equiv \left(h_k \circ C_k \circ \cdots \circ h_1 \circ C_1 \right)$$

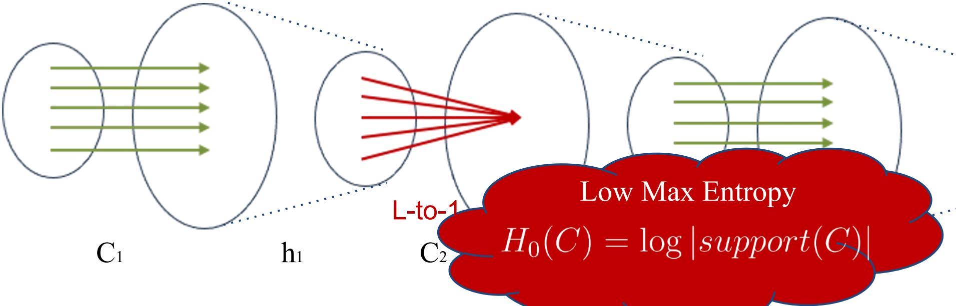


By induction, $cp(\bar{C}_k) \leq \dfrac{2k+1}{2^n}$ with probability $1 - O\left(\dfrac{k^3}{2^n}\right)$

High Collision Entropy $H_2(C) = -\log cp(C)$

No Case

$$\overline{C}_k \equiv \left(h_k \circ C_k \circ \cdots \circ h_1 \circ C_1 \right)$$



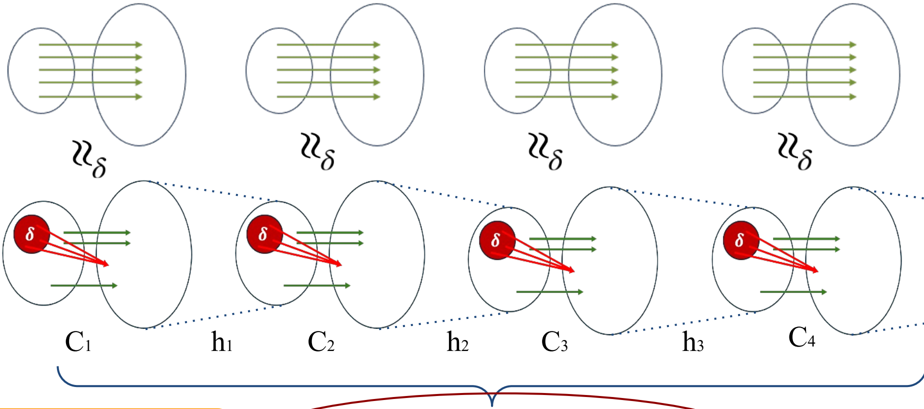
$$cp\left(C_2 \circ h_1 \circ C_1 \right) > \frac{L}{2^n},\ L \in O\left(2^{n^{0.1}} \right)$$

$$support\left(\overline{C}_k \right) \leq support\left(C_2 \circ h_1 \circ C_1 \right) \leq \frac{2^n}{L}$$

Hash Composition

$C_1, \ldots, C_k$	All Injective	Exists L-to-1
$\bar{C}_k$	Circuit with high collision entropy	low max entropy (small support)

Approximate Injectivity



$$\overline{C}_k \equiv \left(h_k \circ C_k \circ \cdots \circ h_1 \circ C_1 \right)$$

Close to Circuit with High Entropy

Summary: AI Hash Composition

NISZK-Complete Smooth Entropy Approximation (SEA)

	All YES	Exists NO
$\bar{C}_k$	close to circuit with high collision entropy	low max entropy (small su
(C_k)	close to uniform	far from uniform

Leftover Hash Lemma

NISZK-Complete SDU [GSV99]

Batching NISZK

CRS: $\Omega(k)$:(
Derandomization

k independent hash functions from 4-wise independent family.

- *CRS* : $[h_1,\ldots,h_k]$, [CRS for NISZK of SEA]
- Protocol:
  - Prover and verifier construct $\bar{C}_k$ with $[h_1,\ldots, h_k]$
  - Prover NISZK protocol for SEA on $\bar{C}_k$

Communication: *poly(n, log k),*

CRS Length: $k \cdot poly(n)$

Derandomizing the Hash Functions

$$\bar{C}_k \equiv \bar{C}_{h_1,\ldots,h_k} \equiv (h_k \circ C_k \circ \cdots \circ h_1 \circ C_1)$$

$cp(\bar{C}_{h_1,\ldots,h_k}(x))$ can be computed using a read-once branching program (width 2^{2n}; depth k) that takes h_i as the randomness in layer i

- Using Nisan ε-PRG [Nis92]

$$h_1, \ldots, h_k \leftarrow H : \; cp(\bar{C}_{h_1,\ldots,h_k})$$

$$\approx_\varepsilon$$

$$h_1, \ldots, h_k \leftarrow PRG(s) : \; cp(\bar{C}_{h_1,\ldots,h_k})$$

- Choose $\varepsilon = 2^{(-\Omega(n))}$
- Seed length = *poly(n, log k)*

Batching NISZK

- CRS : [Seed for *PRG*], [CRS for NISZK of SEA]
- Protocol:
  1. Construct $\bar{C}_k$ with [Seed for *PRG*]
  2. Runs NISZK protocol for SEA on $\bar{C}_k$

Communication: *poly(n, log k)*

CRS Length: *poly(n, log k)*

Summary and Open Problems

Main Theorem:

Every problem $\Pi \in$ NISZK has a non-interactive-SZK batch protocol with *poly*(n, log k) communication and CRS length for $k \in O(2^{n^{0.01}})$

- Open problems:
  - Batch verification for SZK
  - Batch verification NISZK $\cap$ NP with efficient prover
  - [KRV24] Doubly Efficient NISZK Batching for NISZK $\cap$ UP
  - $O(m)+ polylog(n, k)$ communication? where m is commu for one instance.
  - Efficient Batching for More Expressive Policies (beyond conjunction)?

Q&A

Thank You

[BBKLP23] Zvika Brakerski et al. dzSNARGs for Monotone Policy Batch NPǴ. In: Advances in Cryptology - CRYPTO 2023 - 43rd Annual Vol. 14082. Lecture Notes in Computer Science. Springer, 2023, pp. 252–283. DOI: 10.1007/978-3-031-38545-2_9.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. dzNon-Interactive Zero-Knowledge and Its Applications (Extended Abstract)Ǵ. In: Proceedings of the 20th Annual ACM Symposium on T ACM, 1988, pp. 103–112. DOI: 10.1145/62212.62222.

[BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. dzNon-interactive delegation and batch NP verification from standard computational assumptionsǴ. In:

Proceedings of the 49th Annual ACM SIGACT Sympos
ACM, 2017, pp. 474–482. DOI: 10.1145/3055399.3055497.

[CGJJZ23] Arka Rai Choudhuri et al. dzCorrelation Intractability and SNARGs from Sub-exponential DDHǴ. In: Advances in Cryptology - CRYPTO 2023 - 43rd Annual Vol. 14084. Lecture Notes in Computer Science. Springer, 2023, pp. 635–668. DOI: 10.1007/978-3-031-38551-3_20.

[CJJ21a] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. dzNon-interactive Batch Arguments for NP from Standard AssumptionsǴ. In:

Springer, 2021, pp. 394–423. DOI: 10.1007/978-3-030-84259-8_14.

[CJJ21b] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. dzSNARGs for P from LWEĠ. In: 62nd IEEE Annual Symposium on Foundations of Com IEEE, 2021, pp. 68–79. DOI: 10.1109/FOCS52979.2021.00016.

[DGKV22] Lalita Devadas et al. dzRate-1 Non-Interactive Arguments for Batch-NP and ApplicationsĠ. In: 63rd IEEE Annual Symposium on Foundations of Comp IEEE, 2022, pp. 1057–1068. DOI: 10.1109/FOCS54457.2022.00103.

[GHKS23] Aarushi Goel et al. dzSpeed-Stacking: Fast Sub- linear Zero-Knowledge Proofs for DisjunctionsĠ. In:

Springer, 2023, pp. 347–378. DOI: 10.1007/978-3-031-30617-4_12.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. dzThe Knowledge Complexity of Interactive Proof SystemsǴ. In: SIAM J. Comput. 18.1 (1989), pp. 186–208. DOI: 10.1137/0218012.

[GSV99] Oded Goldreich, Amit Sahai, and Salil Vadhan. dzCan Statistical Zero Knowledge Be Made Non-interactive? or On the Relationship of SZK and NISZKǴ. In: Advances in Cryptology — CRYPTO' 99. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 467–484. ISBN: 978-3-540-48405-9.

[KLVW23] Yael Kalai et al. dzBoosting Batch Arguments and RAM DelegationǴ. In: Proceedings of the 55th Annual ACM Symposium on T

ACM, 2023, pp. 1545–1552. DOI: 10.1145/3564246.3585200.

[KRRSV20] Inbar Kaslasi et al. dzBatch Verification for Statistical Zero Knowledge ProofsǴ. In: Theory of Cryptography - 18th International Conferenc Vol. 12551. Lecture Notes in Computer Science. Springer, 2020, pp. 139–167. DOI: 10.1007/978-3-030-64378-2_6.

[KRV21] Inbar Kaslasi, Ron D. Rothblum, and Prashant Nalini Vasudevan. dzPublic-Coin Statistical Zero-Knowledge Batch Verification Against Malicious VerifiersǴ. In: Advances in Cryptology - EUROCRYPT 2021 - 40th A Vol. 12698. Lecture Notes in Computer Science. Springer, 2021, pp. 219–246. DOI: 10.1007/978-3-030-77883-5_8.

[KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. dzSomewhere Statistical Soundness, Post-Quantum Security, and SNARGsǴ. In: Theory of Cryptography - 19th International Conferenc Vol. 13042. Lecture Notes in Computer Science. Springer, 2021, pp. 330–368. DOI: 10.1007/978-3-030-90459-3_12.

[LFKN92] Carsten Lund et al. dzAlgebraic Methods for Interactive Proof SystemsǴ. In: J. ACM 39.4 (1992), pp. 859–868. DOI: 10.1145/146585.146605.

[NWW23] Shafik Nassar, Brent Waters, and David J. Wu. dzMonotone Policy BARGs from BARGs and Additively Homomorphic EncryptionǴ. In: IACR Cryptol. ePrint Arch. (2023).

[PP22] Omer Paneth and Rafael Pass. dzIncrementally Verifiable Computation via Rate-1 Batch ArgumentsǴ. In: 63rd IEEE Annual Symposium on Foundations of Comp

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. Comb. 12(4):449–461, 1992.

,

[RR20] Guy N. Rothblum and Ron D. Rothblum. dzBatch Verification and Proofs of Proximity with Polylog OverheadǴ. In: Theory of Cryptography - 18th International Conferenc

[RRR16] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. dzConstant-round interactive proofs for delegating computationǴ. In: Proceedings of the Forty-Eighth Annual ACM Symposiu STOC '16. Cambridge, MA, USA: Association for Computing Machinery, 2016, pp. 49–62. ISBN: 9781450341325. DOI: 10.1145/2897518.2897652.

[RRR18] Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. dzEfficient Batch Verification for UPǴ. In: 33rd Computational Complexity Conference, CCC 2018 Vol. 102. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, 22:1–22:23. DOI: 10.4230/LIPIcs.CCC.2018.22.

[Sha92] Adi Shamir. dzIP = PSPACEǴ. In: J. ACM 39.4 (1992), pp. 869–877. DOI: 10.1145/146585.146609.

[WW22] Brent Waters and David J. Wu. dzBatch Arguments for NP and More from Standard Bilinear Group AssumptionsǴ. In: Advances in Cryptology - CRYPTO 2022 - 42nd Annua Vol. 13508. Lecture Notes in Computer Science. Springer, 2022, pp. 433–463. DOI: 10.1007/978-3-031-15979-4_15.