Holographic SNARGs for P and Batch-NP from (Polynomially Hard) Learning with Errors

Susumu Kiyoshima

SNARG (Succinct Non-interactive ARGument)

SNARG (Succinct Non-interactive ARGument)

<u>statement: $x \in L$</u>

$$\begin{tikzcd} & G \arrow[dl, "\mathrm{crs}"'] \arrow[dr, "\mathrm{crs}"] & \\ P \arrow[rr, "\pi"] & & V \end{tikzcd}$$

SNARG (Succinct Non-interactive ARGument)

statement: $x \in L$

$$\begin{tikzcd} & G \arrow[dl, "\mathsf{crs}"'] \arrow[dr, "\mathsf{crs}"] & \\ P \arrow[rr, "\pi"] & & V \end{tikzcd}$$

- **Completeness:** $x \in L \Rightarrow$ honest P can convince V
- **Soundness:** $x \notin L \Rightarrow$ any PPT P^* cannot convince V

SNARG (Succinct Non-interactive ARGument)

statement: $x \in L$

$$
\begin{tikzcd}
 & G \arrow[dl, "\mathrm{crs}"'] \arrow[dr, "\mathrm{crs}"] & \\
P \arrow[rr, "\pi"] & & V
\end{tikzcd}
$$

- **Completeness:** $x \in L \Rightarrow$ honest P can convince V
- **Soundness:** $x \notin L \Rightarrow$ any PPT P^* cannot convince V
- **Succinctness:** proof length / verification time are very small

SNARG for Subsets of NP

- **SNARG for P**

- **SNARG for Batch-NP**

SNARG for Subsets of NP

▶ SNARG for P

- **statement**: x (true iff $M(x) = 1$ for a pre-determined poly-time TM M)

▶ SNARG for Batch-NP

SNARG for Subsets of NP

- **SNARG for P**
  - **statement**: x (true iff $M(x) = 1$ for a pre-determined poly-time TM M)
  - **succinctness**: verification time is $\mathrm{poly}(|x|, \log T)$, where $T := M$'s runtime
- **SNARG for Batch-NP**

SNARG for Subsets of NP

SNARG for P

- **statement**: x (true iff $M(x) = 1$ for a pre-determined poly-time TM M)
- **succinctness**: verification time is poly($|x|, \log T$), where $T := M$'s runtime

SNARG for Batch-NP

- **statement**: $(\mathrm{Ckt}, x_1, \ldots, x_k)$ (true iff $\forall i \in [k], \exists w_i$ s.t. $\mathrm{Ckt}(x_i, w_i) = 1$)

SNARG for Subsets of NP

SNARG for P

- **statement**: x (true iff $M(x) = 1$ for a pre-determined poly-time TM M)
- **succinctness**: verification time is $\mathsf{poly}(|x|, \log T)$, where $T := M$'s runtime

SNARG for Batch-NP

- **statement**: $(\mathrm{Ckt}, x_1, \ldots, x_k)$ (true iff $\forall i \in [k], \exists w_i$ s.t. $\mathrm{Ckt}(x_i, w_i) = 1$)
- **succinctness**: proof length is $\mathsf{poly}(|\mathrm{Ckt}|, \log k)$

SNARG for Subsets of NP

NTT

▶ SNARG for P

- **statement**: x (true iff $M(x) = 1$ for a pre-determined poly-time TM M)
- **succinctness**: verification time is poly$(|x|, \log T)$, where $T := M$'s runtime

▶ SNARG for Batch-NP

- **statement**: $(\mathrm{Ckt}, x_1, \ldots, x_k)$ (true iff $\forall i \in [k], \exists w_i$ s.t. $\mathrm{Ckt}(x_i, w_i) = 1$)
- **succinctness**: proof length is poly$(|\mathrm{Ckt}|, \log k)$

Both can be achieved under standard assumptions!

(e.g., LWE, DLIN over bilinear maps, sub-exponential DDH)

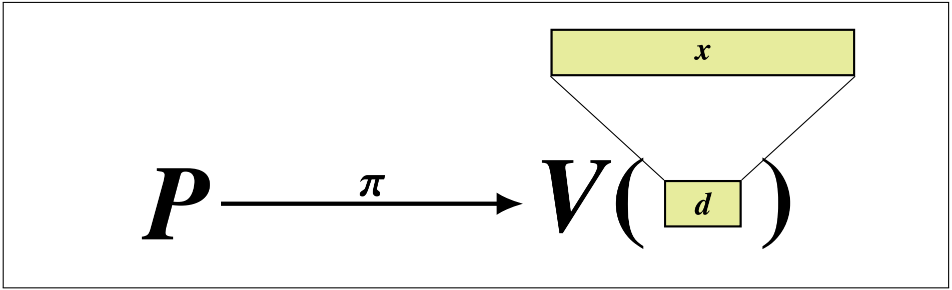
[CJJ21, KVZ21, HJKS22, WW22, CGJJZ23, ...]

Can Verification Time Be Sublinear in |Statement|?

- **Goal**: making verification time be sub-linear in $|x|$ (for the case of P) and sub-linear in $|x_1| + \cdots + |x_k|$ (for the case of Batch-NP)

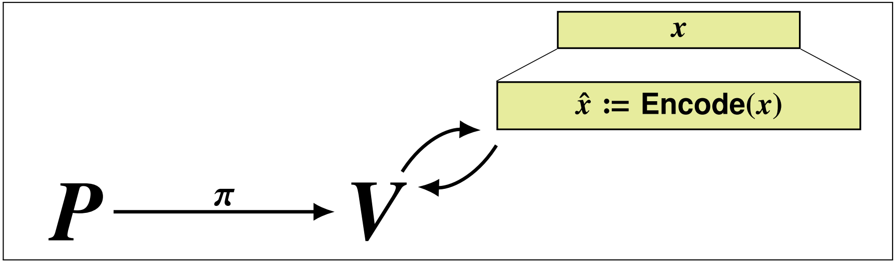
Can Verification Time Be Sublinear in |Statement|?

- **Goal**: making verification time be sub-linear in $|x|$ (for the case of P) and sub-linear in $|x_1| + \cdots + |x_k|$ (for the case of Batch-NP)
  - Possible if the statement is given to V in a pre-processed format (e.g., V is given a digest of x or $(x_1, \ldots, x_k)$ [KP16, CJJ21, KVZ21, DGKV22, ...])



Our Target: Holographic SNARG (1/2)

- Verification time is sub-linear in the statement length when V is given oracle access to an encoding of the statement



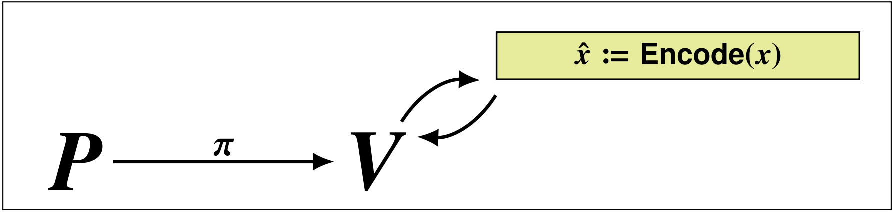
- **Related notions:**
  - Holographic PCPs [BFLS91], Holographic IOPs [CHMMVW20, COS20] , Holographic interactive proofs/arguments [GR17, BR22]

Our Target: Holographic SNARG (2/2)

- **Application:**
  - 2-round arguments of proximity [KR15], 3-round ZK arguments [BKP18, K22], probabilistically checkable arguments [BR22]

Our Target: Holographic SNARG (2/2)

- **Application:**
  - 2-round arguments of proximity [KR15], 3-round ZK arguments [BKP18, K22], probabilistically checkable arguments [BR22]
- **Why useful as building block?**
  - $\hat{x}$ has many nice proprieties (e.g., information theoretic & locally testable)
  - Verification of arbitrary computations is reduced to simple checks about $\hat{x}$



Our Results (1/2)

Theorem 1 (main result)

Two **holographic SNARGs** under the LWE assumption

1. For P, and verification time is poly$(\lambda, \log|x|, \log T)$
2. For Batch-NP, and verification time is poly$(\lambda, |\mathrm{Ckt}|, \log k)$

($\lambda :=$ security parameter)

(As in prior constructions, the encoding we use is low-degree encoding (LDE))

Our Results (1/2)

Theorem 1 (main result)

Two **holographic SNARGs** under the LWE assumption

1. For P, and verification time is poly($\lambda, \log|x|, \log T$)
2. For Batch-NP, and verification time is poly($\lambda, |\mathrm{Ckt}|, \log k$)

($\lambda :=$ security parameter)

(As in prior constructions, the encoding we use is low-degree encoding (LDE))

- **Prior constructions**: either in the designated-verifier setting [KRR22, BHK17] or under the sub-exponential hardness of LWE [K22]
- **Ours**: publicly verifiable and under the polynomial hardness of LWE ☺

Our Results (2/2)

Theorem 2 (application of our holographic SNARGs)

Public-coin 3-round ZK argument from slightly super-poly hardness of LWE and keyless multi-collision-resistant hash function

- Our holographic SNARGs + existing transformation [BKP18, K22]
- Prior to this result:
  - Private-coin: slightly super-poly hardness is sufficient for LWE [BKP18] ☺
  - Public-coin: sub-exponential hardness is required for LWE [K22] ☹

Our Results (2/2)

Theorem 2 (application of our holographic SNARGs)

Public-coin 3-round ZK argument from slightly super-poly hardness of LWE and keyless multi-collision-resistant hash function

- Our holographic SNARGs **+** existing transformation [BKP18, K22]
- Prior to this result:
  - Private-coin: slightly super-poly hardness is sufficient for LWE [BKP18] ☺
  - Public-coin: ~~sub-exponential hardness is required for LWE [K22] ☹~~
    slightly super-poly hardness is sufficient for LWE [this work] ☺

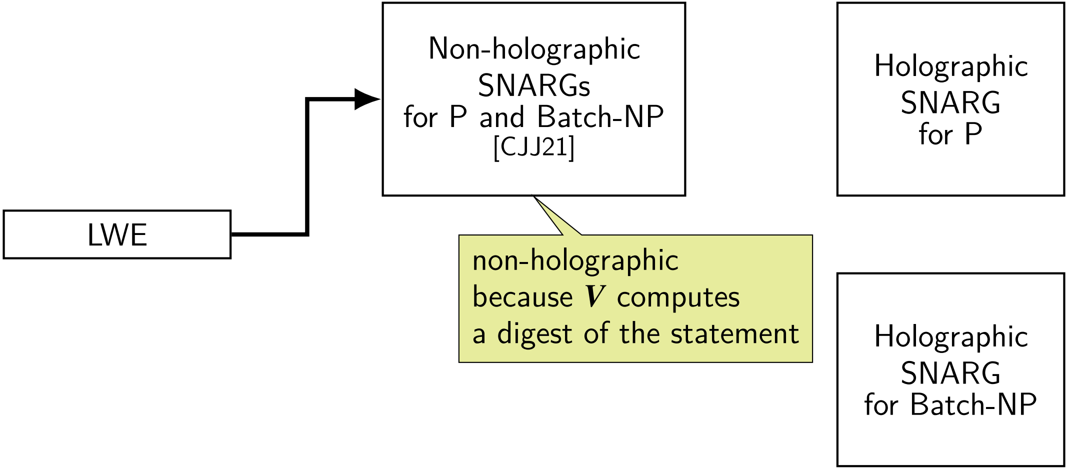
Technical Overview

Holographic SNARG for P

LWE

Holographic SNARG for Batch-NP

Technical Overview



Technical Overview

Non-holographic SNARGs for P and Batch-NP [CJJ21]

Holographic SNARG for P

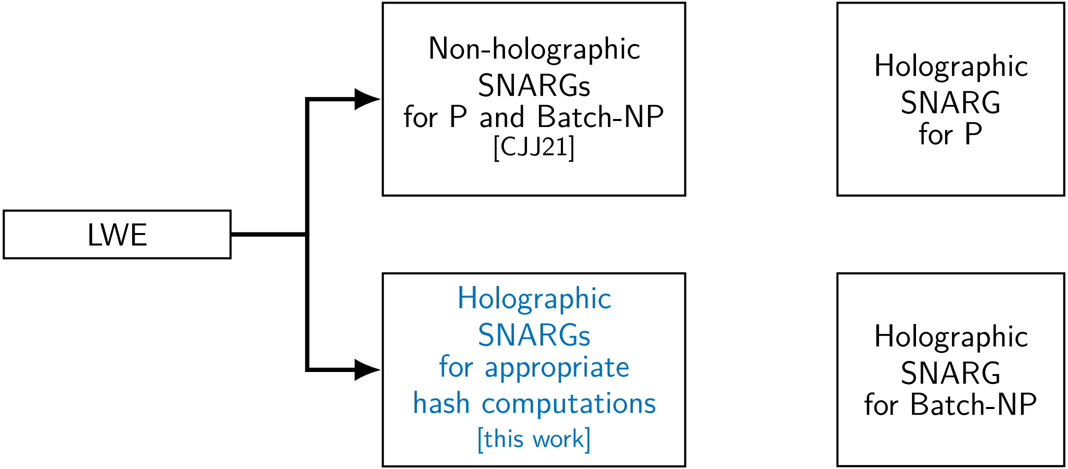
LWE

non-holographic because V computes a digest of the statement

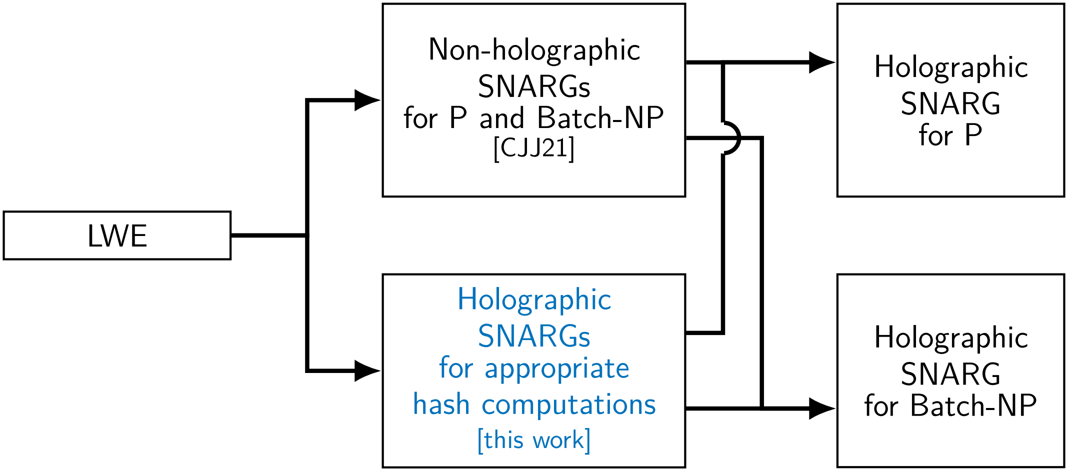
Holographic SNARG for Batch-NP

How about delegating the computation of the digest to P?

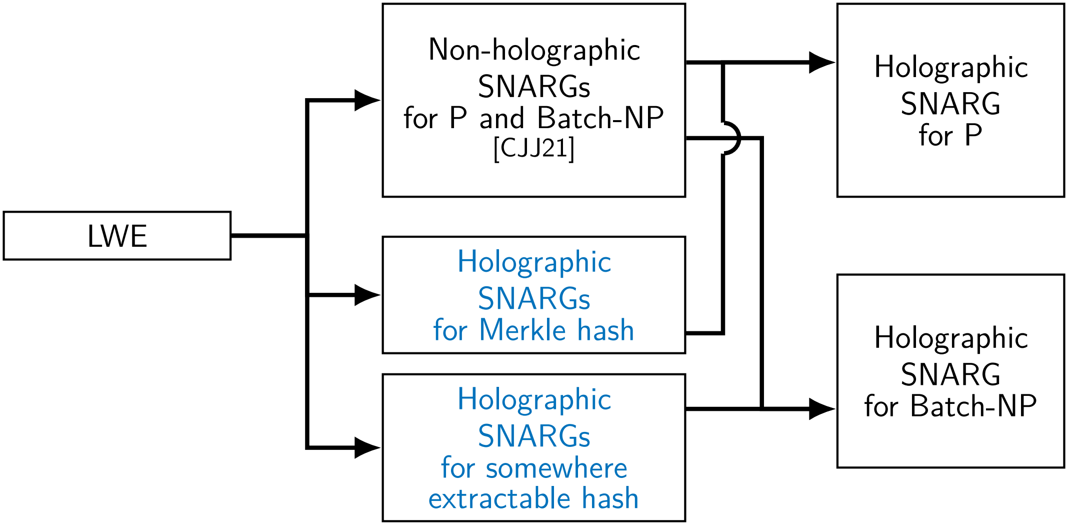
Technical Overview



Technical Overview



Technical Overview



Technical Overview

LWE

Non-holographic SNARGs for P and Batch-NP [CJJ21]

Holographic SNARGs for Merkle hash

Holographic SNARGs for somewhere extractable hash

Key Points:

1. **weak soundness**
(guaranteeing that extractable value and input string are consistent)

2. **recursive construction**
(each recursion depends only on a small part of the encoding of statement)

Technical Overview

LWE

Holographic SNARGs for Merkle hash

Holographic SNARGs for somewhere extractable hash

Key Point:
construction relies on simple structure of Merkle hash

Key Points:

1. **weak soundness**
(guaranteeing that extractable value and input string are consistent)

2. **recursive construction**
(each recursion depends only on a small part of the encoding of statement)

Conclusion

- **Main Result:**
  - **Holographic SNARGs for P and Batch-NP from LWE**
- **Application:**
  - **Public-coin 3-round ZK from weaker assumptions**

    (closing the gap between public-coin 3-round ZK and private-coin one)

Conclusion

- **Main Result:**
  - **Holographic SNARGs for P and Batch-NP from LWE**
- **Application:**
  - **Public-coin 3-round ZK from weaker assumptions**

    (closing the gap between public-coin 3-round ZK and private-coin one)

Thank you!

Reference

- [BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. STOC 1991.
- [BHK17] Zvika Brakerski, Justin Holmgren, and Yael Tauman Kalai. Non-interactive delegation and batch NP verification from standard computational assumptions. STOC 2017.
- [BKP18] Nir Bitansky, Yael Tauman Kalai, and Omer Paneth. Multi-collision resistance: a paradigm for keyless hash functions. STOC 2018.
- [BR22] Liron Bronfman and Ron D. Rothblum. PCPs and Instance Compression from a Cryptographic Lens. ITCS 2022.
- [CGJJZ23] Arka Rai Choudhuri, Sanjam Garg, Abhishek Jain, Zhengzhong Jin, Jiaheng Zhang . Correlation Intractability and SNARGs from Sub-exponential DDH. CRYPTO 2023.
- [CHMMVW20] Alessandro Chiesa, Yuncong Hu, Mary Maller, Pratyush Mishra, Psi Vesely, and Nicholas P. Ward. Marlin: Preprocessing zkSNARKs with universal and updatable SRS. EUROCRYPT 2020.
- [CJJ21] Arka Rai Choudhuri, Abhishek Jain, and Zhengzhong Jin. SNARGs for P from LWE. FOCS 2021.
- [COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. EUROCRYPT 2020.
- [DGKV22] Lalita Devadas, Rishab Goyal, Yael Kalai, and Vinod Vaikuntanathan. Rate-1 non-interactive arguments for batch-NP and applications. FOCS 2022.
- [GR17] Tom Gur and Ron D. Rothblum. A hierarchy theorem for interactive proofs of proximity. ITCS 2017.
- [HJKS22] James Hulett, Ruta Jawale, Dakshita Khurana, and Akshayaram Srinivasan. SNARGs for P from sub-exponential DDH and QR. EUROCRYPT 2022.
- [K22] Susumu Kiyoshima. Public-coin 3-round zero-knowledge from learning with errors and keyless multi-collision-resistant hash. CRYPTO 2022.
- [KP16] Yael Tauman Kalai and Omer Paneth. Delegating RAM computations. TCC 2016-B.
- [KR15] Yael Tauman Kalai and Ron D. Rothblum. Arguments of proximity. CRYPTO 2015.
- [KRR22] Yael Tauman Kalai, Ran Raz, and Ron D Rothblum. How to Delegate Computations: The Power of No-Signaling Proofs. Journal of the ACM, 2022.
- [KVZ21] Yael Tauman Kalai, Vinod Vaikuntanathan, and Rachel Yun Zhang. Somewhere statistical soundness, post-quantum security, and SNARGs. TCC 2021.
- [WW22] Brent Waters and David J. Wu. Batch arguments for NP and more from standard bilinear group assumptions. CRYPTO 2022.