Direct FSS Constructions For Branching Programs and More from PRGs with Encoded-Output Homomorphism

Elette Boyle Lisa Kohl Zhe Li Peter Scholl

Reichman University & NTT CWI CWI Aarhus University

December 11, 2024

Table of Contents

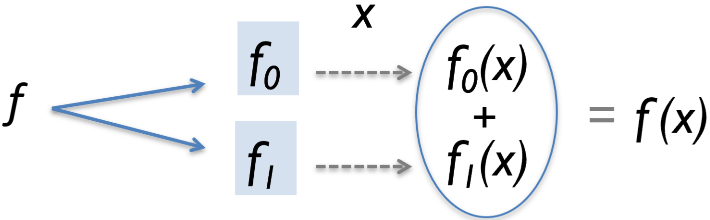
Table of Contents

Function Secret Sharing (FSS)

FSS is a secret sharing scheme for functions.

Function Secret Sharing (FSS)

FSS is a secret sharing scheme for functions.



Function Secret Sharing (FSS)

FSS is a secret sharing scheme for functions.

Definition (FSS)

Given $F := \{f\colon D \to R\}$, FSS consists of a pair of PPT algorithms $(\mathsf{Gen}, \mathsf{Eval})$:

- $(k_0, k_1) \leftarrow \mathsf{Gen}(f, 1^\lambda)$.
- $y_b \leftarrow \mathsf{Eval}(b, k_b, x)$.

Function Secret Sharing (FSS)

FSS is a secret sharing scheme for functions.

Definition (FSS)

Given $F := \{f : D \to R\}$, FSS consists of a pair of PPT algorithms $(\mathsf{Gen}, \mathsf{Eval})$:

- $(k_0, k_1) \leftarrow \mathsf{Gen}(f, 1^\lambda)$.
- $y_b \leftarrow \mathsf{Eval}(b, k_b, x)$.

Hiding: k_b alone hides f.

Correctness: $y_0 + y_1 = f(x)$ for all $x \in D$.

Compactness: $|k_b|$ scales with the description size of f

In this work, we focus on the two-party case.

Applications of FSS (Incomplete List)

- Mozillia+Prio: Private data collection [CB17] moz://a
- Writing PIR: Riposte [CBM15]
- SQL query: Splinter [WYG+17]
- ORAM: Floram [Ds17]
- Mixed-mode secure computation [BGI19, BCG+21]
- Private heavy hitters [BBC+21]
- Private set intersection (PSI) [GRS22, GRS23]

Previous Constructions of (Two-Party) FSS

- FSS from one-way functions [GI14, BGI15, BGI16b, BCG+21]
  - for point functions, interval functions, decision trees
- FSS from learning parity with noise [BCG+19, CM21, DIJL23]
  - for low-degree polynomials
- FSS from DDH [BGI16a, BCG+17, BGI+18], DCR [FGJS17, OSY21, RS21], LWE [BKS19, ACK23], class groups [ADOS22]
  - for branching programs
  - via homomorphic secret sharing and universal branching programs
- FSS from multi-key FHE [DHRW16]
  - for all function classes

Table of Contents

Our Results

1. PRG with Encoded-Output Homomorphism (EOH-PRG)
   - New abstraction of (relaxed) homomorphic PRG
   - Can be instantiated from LWE or DCR

Our Results

1. PRG with Encoded-Output Homomorphism (EOH-PRG)
   - New abstraction of (relaxed) homomorphic PRG
   - Can be instantiated from LWE or DCR
2. Direct FSS constructions from EOH-PRG
   - for bit-fixing predicates
   - for branching programs (without universal circuits)
   - for deterministic finite automata (DFA)

Our Results - Efficiency

		Assumption	Key Size	Run time (No. of Mul./ Exp.)
LWE	HSS [BKS19]	Ring-LWE	$4(n \cdot w) \cdot w\ell \log q$	$8w^2 \cdot \ell n \log n$
	EOH-PRG(Ours)	Ring-LWE	$2(n + w) \cdot w\ell \log p$	$2(1 + \lceil \frac{w}{n} \rceil) \cdot \ell n \log n$
DCR	HSS [OSY21]	DCR	$7w \cdot w\ell \log N^2$	$14w^2 \cdot \ell$
	EOH-PRG(Ours)	DCR	$2(w + 1) \cdot w\ell \log N^2$	$(3w + 2) \cdot \ell$

- (ℓ, w) length and width of the branching program
- For LWE, n secret dimension, p plaintext modulus and q ciphertext modulus
- For DCR, N RSA modulus

Table of Contents

Recall FSS for Point Functions [GI14, BGI15, BGI16b]

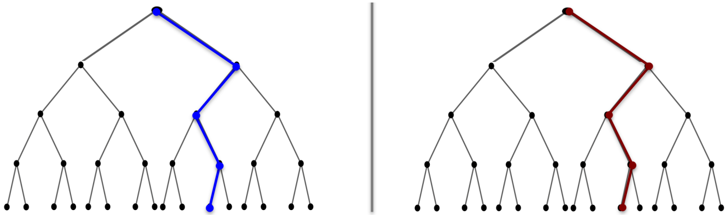
For $\alpha \in \{0,1\}^n, \beta \in R$, $f_{\alpha,\beta}\colon \{0,1\}^n \to R$ is defined as

$$f_{\alpha,\beta}(x) = \begin{cases} \beta, & \text{if } x = \alpha. \\ 0, & \text{otherwise.} \end{cases}$$

Recall FSS for Point Functions [GI14, BGI15, BGI16b]

For $\alpha \in \{0,1\}^n, \beta \in R$, $f_{\alpha,\beta}\colon \{0,1\}^n \to R$ is defined as

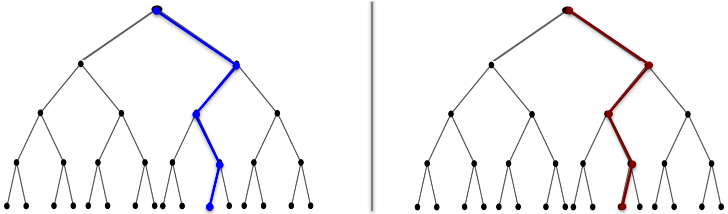
$$f_{\alpha,\beta}(x) = \begin{cases} \beta, & \text{if } x = \alpha. \\ 0, & \text{otherwise.} \end{cases}$$



Recall FSS for Point Functions [GI14, BGI15, BGI16b]

For $\alpha \in \{0,1\}^n, \beta \in R$, $f_{\alpha,\beta}\colon \{0,1\}^n \to R$ is defined as

$$f_{\alpha,\beta}(x) = \begin{cases} \beta, & \text{if } x = \alpha. \\ 0, & \text{otherwise.} \end{cases}$$



- The shares define two correlated GGM-like trees.
- $CW_i \leftarrow G(s_i) + \mathcal{E}(s_{i+1})$ w.r.t. α with $G(\cdot)$ PRG.
- For each node v, the two parties obtain shares of $(\mathbf{s}_i, 1) \in \{0,1\}^{\lambda+1}$ if v is specified by α and $(\mathbf{0}, 0)$ otherwise.

Towards FSS for Bit-Fixing Predicates

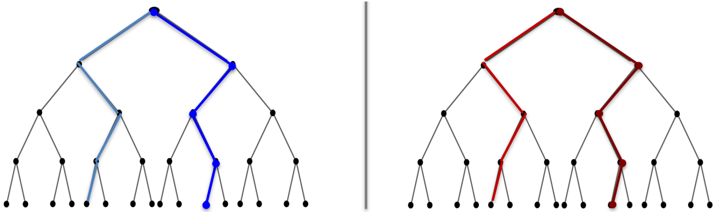
For $\alpha \in \{0,1,*\}^n, \beta \in \{0,1\}$, $f_{\alpha,\beta}\colon \{0,1\}^n \to \{0,1\}$ is defined as

$$f_\alpha(x) = \begin{cases} \beta, & \text{if } \wedge_{i\in[n]} (\alpha[i] = * \vee x[i] = \alpha[i]). \\ 0, & \text{otherwise.} \end{cases}$$

Towards FSS for Bit-Fixing Predicates

For $\alpha \in \{0,1,*\}^n, \beta \in \{0,1\}$, $f_{\alpha,\beta}\colon \{0,1\}^n \to \{0,1\}$ is defined as

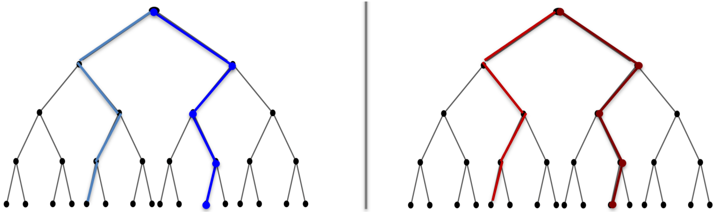
$$f_\alpha(x) = \begin{cases} \beta, & \text{if } \wedge_{i\in[n]} (\alpha[i] = * \vee x[i] = \alpha[i]). \\ 0, & \text{otherwise.} \end{cases}$$



Towards FSS for Bit-Fixing Predicates

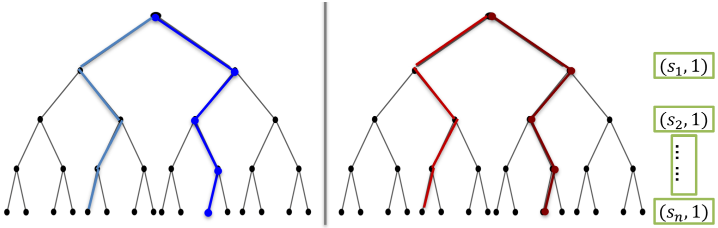
For $\alpha \in \{0,1,*\}^n, \beta \in \{0,1\}$, $f_{\alpha,\beta}\colon \{0,1\}^n \to \{0,1\}$ is defined as

$$f_\alpha(x) = \begin{cases} \beta, & \text{if } \wedge_{i\in[n]} (\alpha[i] = * \vee x[i] = \alpha[i]). \\ 0, & \text{otherwise.} \end{cases}$$

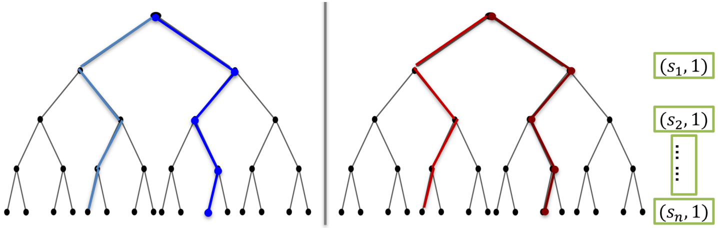


- $\lambda^{\omega(1)}$ matched evaluation paths.
- Key size scales with the number of matched paths.

How to compress the key size?

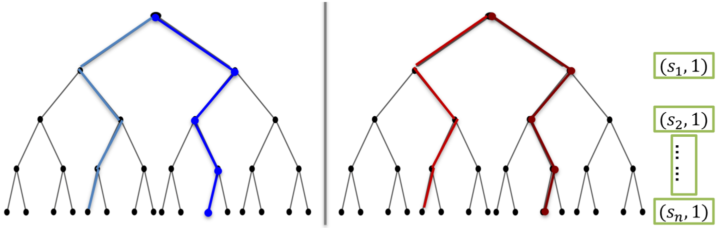


How to compress the key size?



1. Sample a random value for each level
2. For each level i node v, the two parties obtain shares of $(\mathbf{s}_i, 1) \in \{0,1\}^{\lambda+1}$ if v is specified by α and $(\mathbf{0}, 0)$ otherwise.

How to compress the key size?



1. Sample a random value for each level
2. For each level i node v, the two parties obtain shares of $(\mathbf{s}_i, 1) \in \{0,1\}^{\lambda+1}$ if v is specified by α and $(\mathbf{0}, 0)$ otherwise.

How to move from current level to next level while maintaining the invariant for each matched node?

Homomorphic PRG (A Little Technical)

Suppose we have a homomorphic PRG satisfying $G(s+t) = G(s) + G(t)$.

1. Assume $CW_i \leftarrow G(s_i) \oplus \mathcal{E}(s_{i+1})$.
2. Assume P_b has $s_{i,b}$ such that $s_{i,0} \oplus s_{i,1} = s_i$.
3. Then P_b obtains $\sigma_b \leftarrow b \cdot CW_i \oplus G(s_{i,b})$.
4. It holds that $\sigma_0 \oplus \sigma_1 = \mathcal{E}(s_{i+1})$.

Homomorphic PRG (A Little Technical)

Suppose we have a homomorphic PRG satisfying $G(s+t) = G(s) + G(t)$.

1. Assume $CW_i \leftarrow G(s_i) \oplus \mathcal{E}(s_{i+1})$.
2. Assume P_b has $s_{i,b}$ such that $s_{i,0} \oplus s_{i,1} = s_i$.
3. Then P_b obtains $\sigma_b \leftarrow b \cdot CW_i \oplus G(s_{i,b})$.
4. It holds that $\sigma_0 \oplus \sigma_1 = \mathcal{E}(s_{i+1})$.

In our construction, $\sigma_b \leftarrow t_b \cdot CW_i \oplus G(s_{i,b})$, where $t_0 \oplus t_1 = 1$ if the node is in the path defined by α and $t_0 \oplus t_1 = 0$ otherwise.

Homomorphic PRG (A Little Technical)

Suppose we have a homomorphic PRG satisfying $G(s+t) = G(s) + G(t)$.

1. Assume $CW_i \leftarrow G(s_i) \oplus \mathcal{E}(s_{i+1})$.
2. Assume P_b has $s_{i,b}$ such that $s_{i,0} \oplus s_{i,1} = s_i$.
3. Then P_b obtains $\sigma_b \leftarrow b \cdot CW_i \oplus G(s_{i,b})$.
4. It holds that $\sigma_0 \oplus \sigma_1 = \mathcal{E}(s_{i+1})$.

In our construction, $\sigma_b \leftarrow t_b \cdot CW_i \oplus G(s_{i,b})$, where $t_0 \oplus t_1 = 1$ if the node is in the path defined by α and $t_0 \oplus t_1 = 0$ otherwise.

The invariant is preserved for each node.

Homomorphic PRG (A Little Technical)

Suppose we have a homomorphic PRG satisfying $G(s+t) = G(s) + G(t)$.

1. Assume $CW_i \leftarrow G(s_i) \oplus \mathcal{E}(s_{i+1})$.
2. Assume P_b has $s_{i,b}$ such that $s_{i,0} \oplus s_{i,1} = s_i$.
3. Then P_b obtains $\sigma_b \leftarrow b \cdot CW_i \oplus G(s_{i,b})$.
4. It holds that $\sigma_0 \oplus \sigma_1 = \mathcal{E}(s_{i+1})$.

In our construction, $\sigma_b \leftarrow t_b \cdot CW_i \oplus G(s_{i,b})$, where $t_0 \oplus t_1 = 1$ if the node is in the path defined by α and $t_0 \oplus t_1 = 0$ otherwise.

The invariant is preserved for each node.

Problem

Homomorphic PRG does not exist!

PRG with Encoded-Output Homomorphism(EOH-PRG)

Definition (EOH-PRG)

Given additive secret shares (s_0, s_1) of a seed s, and additive secret shares (y_0, y_1) of a blinded encoding $G(s) + \mathcal{E}(m)$, it holds that

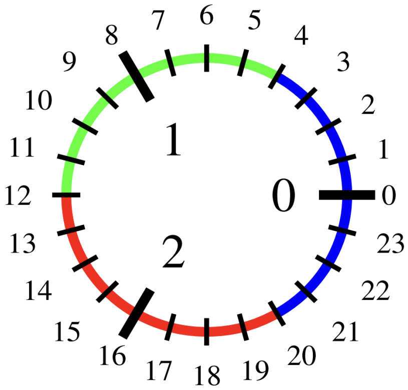
$$\mathsf{Conv}(y_0 - G(s_0)) - \mathsf{Conv}(y_1 - G(s_1)) = m$$

except with negligible probability over the random choice of the secret shares.

Then $(G, \mathsf{Conv}, \mathcal{E})$ is a EOH-PRG.

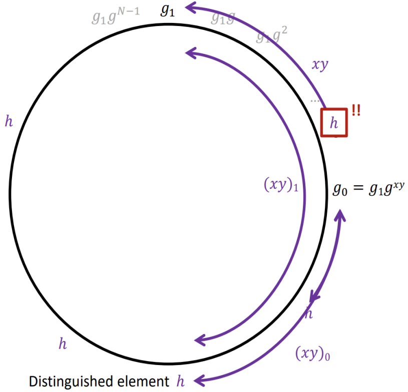
EOH-PRG Instantiatings from LWE or DCR

LWE
Distributed rounding [BKS19]



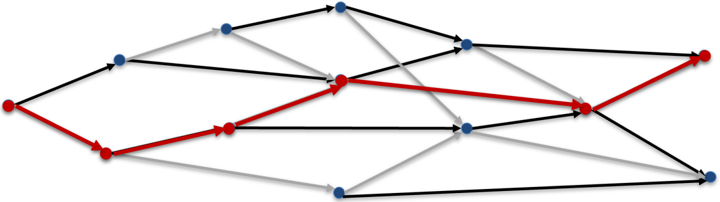
Lifting

DCR
DDLog [OSY21, RS21]



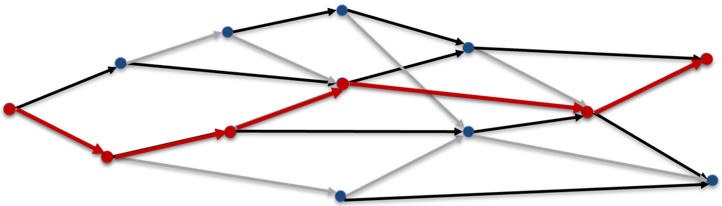
Lifting

FSS for Branching Programs



- Reduce $\lambda^{\omega(1)}$ evaluation paths to $O(w)$ nodes of each level of BP.

FSS for Branching Programs



- Reduce $\lambda^{\omega(1)}$ evaluation paths to $O(w)$ nodes of each level of BP.
- Viewing a DFA as a BP jumping to the same level together with KDM security, we obtain FSS for DFAs.

Summary

- New EOH-PRG abstraction.
- FSS for bit-fixing predicates from EOH-PRG.
- FSS for branching programs from EOH-PRG without universal transformations.
- FSS for DFAs from KDM secure EOH-PRG.
    - It is not clear how to achieve it from HSS for BPs.

Summary

- New EOH-PRG abstraction.
- FSS for bit-fixing predicates from EOH-PRG.
- FSS for branching programs from EOH-PRG without universal transformations.
- FSS for DFAs from KDM secure EOH-PRG.
  - It is not clear how to achieve it from HSS for BPs.

The Key Takeaway

An ideal cryptographic primitive, even if purely hypothetical, such as a homomorphic PRG, has the potential to significantly simplify the overall construction.


Thank you

References I

Thomas Attema, Pedro Capitão, and Lisa Kohl.
On homomorphic secret sharing from polynomial-modulus LWE.
In *PKC 2023, Part II*, LNCS, pages 3–32, May 2023.

Damiano Abram, Ivan Damgård, Claudio Orlandi, and Peter Scholl.
An algebraic framework for silent preprocessing with trustless setup and active security.
In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part IV*, volume 13510 of *LNCS*, pages 421–452, August 2022.

Dan Boneh, Elette Boyle, Henry Corrigan-Gibbs, Niv Gilboa, and Yuval Ishai.
Lightweight techniques for private heavy hitters.
In *2021 IEEE Symposium on Security and Privacy*, pages 762–776. IEEE Computer Society Press, May 2021.

References II

Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, and Michele Orrù.
Homomorphic secret sharing: Optimizations and applications.
In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 2105–2122. ACM Press, October / November 2017.

Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl.
Efficient pseudorandom correlation generators: Silent OT extension and more.
In Alexandra Boldyreva and Daniele Micciancio, editors, *CRYPTO 2019, Part III*, volume 11694 of *LNCS*, pages 489–518, August 2019.

References III

Elette Boyle, Nishanth Chandran, Niv Gilboa, Divya Gupta, Yuval Ishai, Nishant Kumar, and Mayank Rathee.
Function secret sharing for mixed-mode and fixed-point secure computation.
In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 871–900, October 2021.

Elette Boyle, Niv Gilboa, and Yuval Ishai.
Function secret sharing.
In Elisabeth Oswald and Marc Fischlin, editors, *EUROCRYPT 2015, Part II*, volume 9057 of *LNCS*, pages 337–367, April 2015.

References IV

Elette Boyle, Niv Gilboa, and Yuval Ishai.
Breaking the circuit size barrier for secure computation under DDH.
In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part I*, volume 9814 of *LNCS*, pages 509–539, August 2016.

Elette Boyle, Niv Gilboa, and Yuval Ishai.
Function secret sharing: Improvements and extensions.
In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *ACM CCS 2016*, pages 1292–1303. ACM Press, October 2016.

Elette Boyle, Niv Gilboa, Yuval Ishai, Huijia Lin, and Stefano Tessaro.
Foundations of homomorphic secret sharing.
In *9th Innovations in Theoretical Computer Science, ITCS 2018*, page 21. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2018.

References V

Elette Boyle, Niv Gilboa, and Yuval Ishai.
Secure computation with preprocessing via function secret sharing.
In Dennis Hofheinz and Alon Rosen, editors, *TCC 2019, Part I*, volume 11891 of *LNCS*, pages 341–371, December 2019.

Elette Boyle, Lisa Kohl, and Peter Scholl.
Homomorphic secret sharing from lattices without FHE.
In Yuval Ishai and Vincent Rijmen, editors, *EUROCRYPT 2019, Part II*, volume 11477 of *LNCS*, pages 3–33, May 2019.

Henry Corrigan-Gibbs and Dan Boneh.
Prio: Private, robust, and scalable computation of aggregate statistics.
In *NSDI*, pages 259–282. USENIX Association, 2017.

References VI

Henry Corrigan-Gibbs, Dan Boneh, and David Mazières.
Riposte: An anonymous messaging system handling millions of users.
In *2015 IEEE Symposium on Security and Privacy*, pages 321–338.
IEEE Computer Society Press, May 2015.

Geoffroy Couteau and Pierre Meyer.
Breaking the circuit size barrier for secure computation under quasi-polynomial LPN.
In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part II*, volume 12697 of *LNCS*, pages 842–870, October 2021.

Yevgeniy Dodis, Shai Halevi, Ron D. Rothblum, and Daniel Wichs.
Spooky encryption and its applications.
In Matthew Robshaw and Jonathan Katz, editors, *CRYPTO 2016, Part III*, volume 9816 of *LNCS*, pages 93–122, August 2016.

References VII

- Quang Dao, Yuval Ishai, Aayush Jain, and Huijia Lin.
  Multi-party homomorphic secret sharing and sublinear MPC from sparse LPN.
  In *CRYPTO 2023, Part II*, LNCS, pages 315–348, August 2023.

- Jack Doerner and abhi shelat.
  Scaling ORAM for secure computation.
  In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *ACM CCS 2017*, pages 523–535. ACM Press, October / November 2017.

References VIII

Nelly Fazio, Rosario Gennaro, Tahereh Jafarikhah, and William E Skeith.
Homomorphic secret sharing from paillier encryption.
In *Provable Security: 11th International Conference, ProvSec 2017, Xi'an, China, October 23-25, 2017, Proceedings 11*, pages 381–399. Springer, 2017.

Niv Gilboa and Yuval Ishai.
Distributed point functions and their applications.
In Phong Q. Nguyen and Elisabeth Oswald, editors, *EUROCRYPT 2014*, volume 8441 of *LNCS*, pages 640–658, May 2014.

References IX

Gayathri Garimella, Mike Rosulek, and Jaspal Singh.
Structure-aware private set intersection, with applications to fuzzy matching.
In Yevgeniy Dodis and Thomas Shrimpton, editors, *CRYPTO 2022, Part I*, volume 13507 of *LNCS*, pages 323–352, August 2022.

Gayathri Garimella, Mike Rosulek, and Jaspal Singh.
Malicious secure, structure-aware private set intersection.
In *CRYPTO 2023, Part I*, LNCS, pages 577–610, August 2023.

Claudio Orlandi, Peter Scholl, and Sophia Yakoubov.
The rise of paillier: Homomorphic secret sharing and public-key silent OT.
In Anne Canteaut and François-Xavier Standaert, editors, *EUROCRYPT 2021, Part I*, volume 12696 of *LNCS*, pages 678–708, October 2021.

References X

Lawrence Roy and Jaspal Singh.
Large message homomorphic secret sharing from DCR and applications.
In Tal Malkin and Chris Peikert, editors, *CRYPTO 2021, Part III*, volume 12827 of *LNCS*, pages 687–717, Virtual Event, August 2021.

Frank Wang, Catherine Yun, Shafi Goldwasser, Vinod Vaikuntanathan, and Matei Zaharia.
Splinter: Practical private queries on public data.
In *NSDI 2017*, 2017.