Efficient Pseudorandom Correlation Generators From Ring-LPN

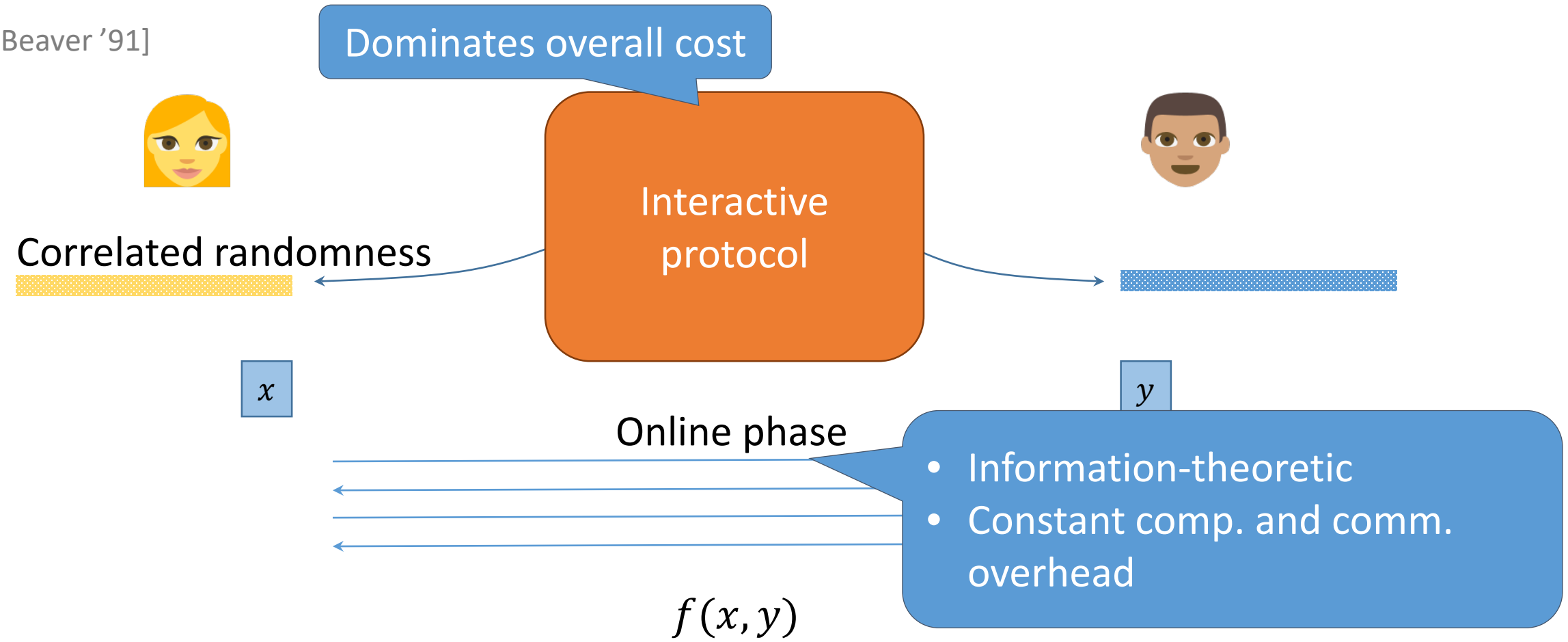
Crypto 2020

Elette Boyle (IDC Herzliya)
Geoffroy Couteau (IRIF)
Niv Gilboa (Ben-Gurion University)
Yuval Ishai (Technion)
Lisa Kohl (Technion)
Peter Scholl (Aarhus University)

AARHUS
UNIVERSITY

Secure Computation with Preprocessing

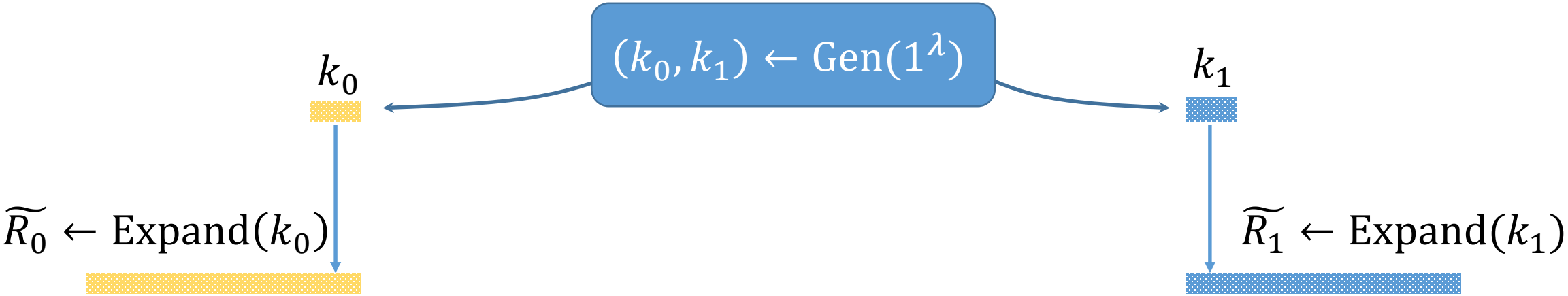
[Beaver '91]



Pseudorandom Correlation Generators (PCGs)

[BCGI 18, BCGIK**S** 19]

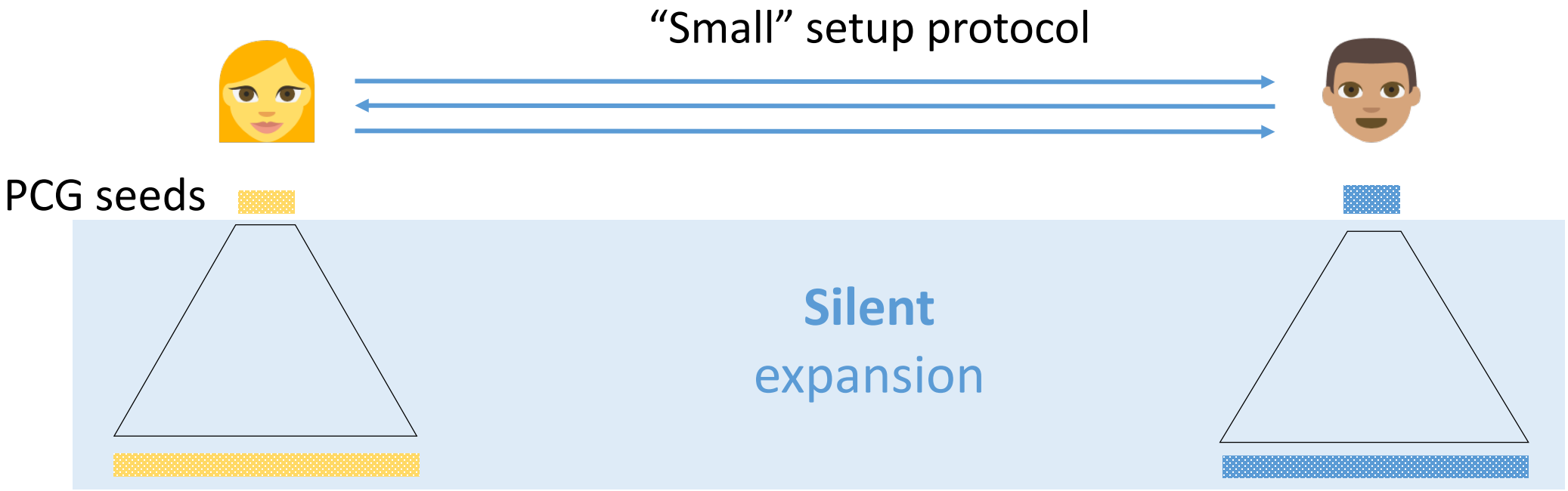
Target correlation: (R_0, R_1)



Correctness: $(\widetilde{R_0}, \widetilde{R_1}) \approx (R_0, R_1)$

Security: $(k_0, \widetilde{R_1}) \approx (k_0, R_1 \mid \widetilde{R_0})$

PCG + Setup Protocol ⇒ Silent Preprocessing



Correlated *pseudorandomness*

Previous PCG Constructions

- From Learning With Errors:
  - Arbitrary additive correlations via homomorphic secret sharing [BGI 16a, DHRW 16]
  - Uses FHE, expensive

- From Learning Parity with Noise (LPN):
  - Vector-OLE [BCGI 18]
  - OT [BCGIK**S**19]
  - OLE, multiplication triples [BCGIK**S**19]

Good concrete efficiency
[SGRR 19, BCGIKR**S** 19, YWLZW 20]

Impractical $O(N^2)$ cost

- Also:
  - Linear correlations [GI99, CDI05], truth tables [BCGIK**S**19]

This work

- Efficient PCGs for OLE/multiplication triples from the ring-LPN assumption
  - Silently expand N OLEs in $\tilde{O}(N)$ time
  - Extensions: authenticated triples, multi-party, other bilinear correlations

- Actively secure setup: silent preprocessing for SPDZ (two parties)
  - Concretely efficient: ≈1MB seeds, runtimes comparable to Overdrive [KPR 18]

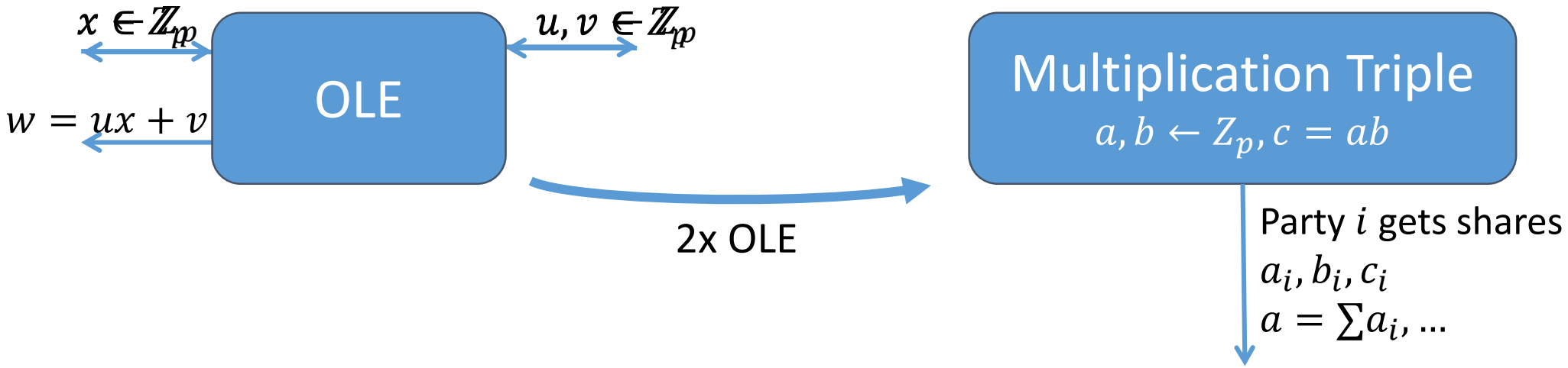
- Other highlights:
  - LPN → ring-LPN to avoid $O(N^2)$ blowup
  - "Bootstrapping" as an optimization

  } inspired by FHE [Gen 09, BV 11]

  - Security analysis of arithmetic variants of ring-LPN over $Z_p[X]/f(X)$

Main goals: Oblivious linear function evaluation and multiplication triples

[Beaver 91]



Authenticated triples: [BDOZ11, DPSZ12]

- Parties also get shares of $a\Delta, b\Delta, c\Delta$

Background: Distributed Point Functions (DPF)

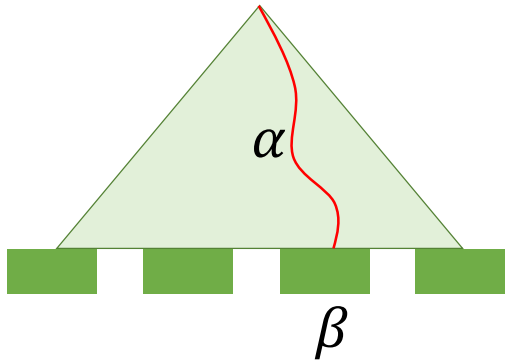
Point function $f_{\alpha,\beta}: \{1, \dots, N\} \to \{0,1\}^{\lambda}$

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

Distributed Point Function:

$$\text{Gen}(\alpha, \beta) \to (k_0, k_1)$$

$$\text{Eval}(k_0, x) \oplus \text{Eval}(k_1, x) = f_{\alpha,\beta}(x)$$



- Efficient tree-based constructions from PRG [GI 14, BGI 15, BGI 16b]
- Efficient distributed setup protocol [Ds 17]

Warm-up: PCG for unit vector from DPF

$\text{Gen}(\alpha, \beta)$

k_0

k_1

Eval at all $x \in \{1, \ldots, N\}$

y_0

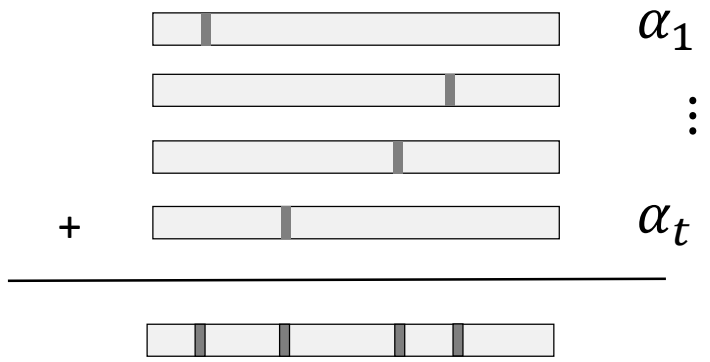
$y_0 \oplus y_1 = (00 \cdots \beta \cdots 00)$

y_1

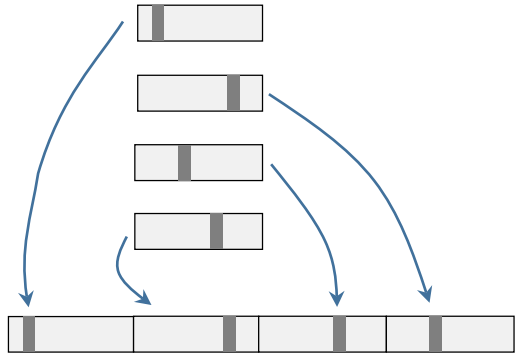
Warm-up: from unit vector to sparse vector

Sparsity t: use t DPF instances

Approach 1: addition



Approach 2: concatenation

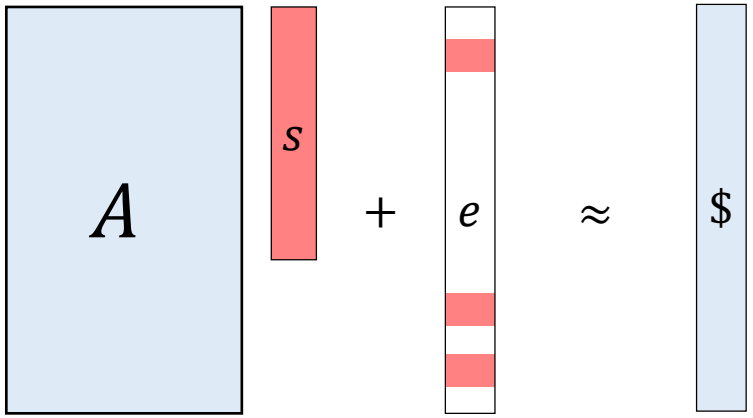


More efficient

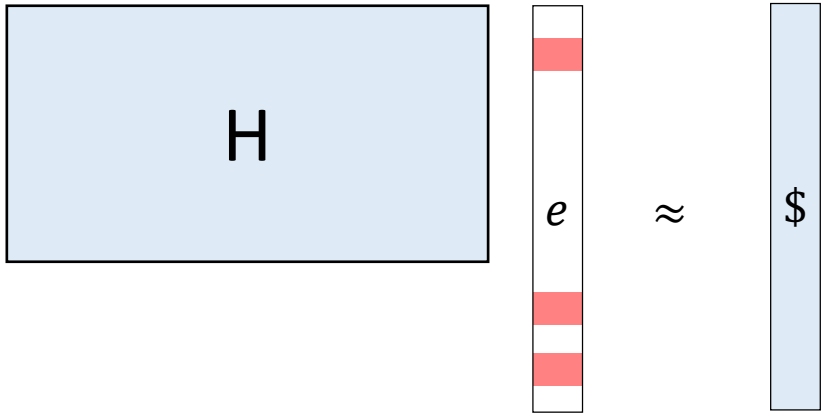
Requires regular structure

Learning Parity with Noise assumption

Primal LPN



Dual LPN (or syndrome decoding)



A, s random over $\mathrm{Z_p}$ $(p \geq 2)$

$HW(e)$ small

Variants:

- Structured A, H
- Structured e

Starting Point:
PCG for tensor product from [BCGIKS 19]

Target correlation:

$\boldsymbol{x}_0 \in Z_p^N, Z_0 \in Z_p^{N \times N}$

$\boldsymbol{x}_1 \in Z_p^N, Z_1 \in Z_p^{N \times N}$

Such that

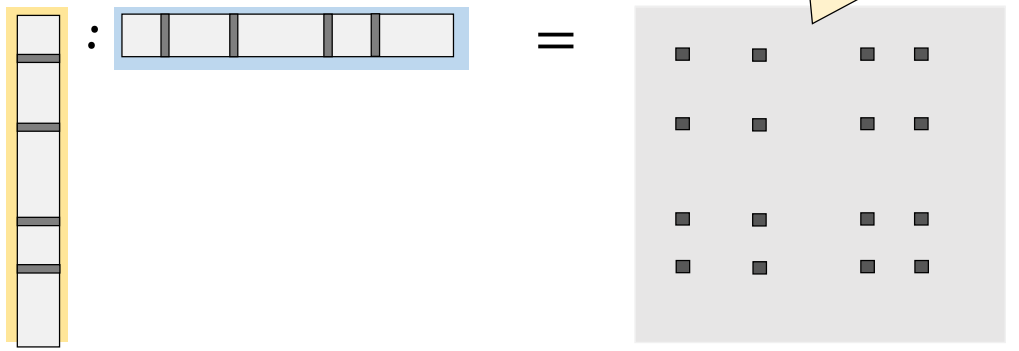
$$Z_0 + Z_1 = \boldsymbol{x}_0 \otimes \boldsymbol{x}_1$$

Note:

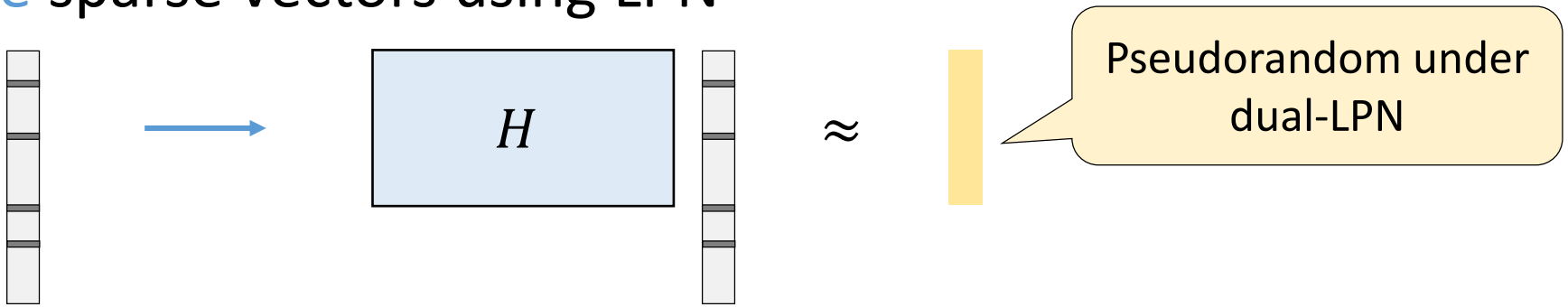
- Tensor product implies N OLEs
- $\Omega(N^2)$ cost

PCG for tensor product from [BCGIKS 19]
Main ideas:

1) Tensor product preserves sparsity

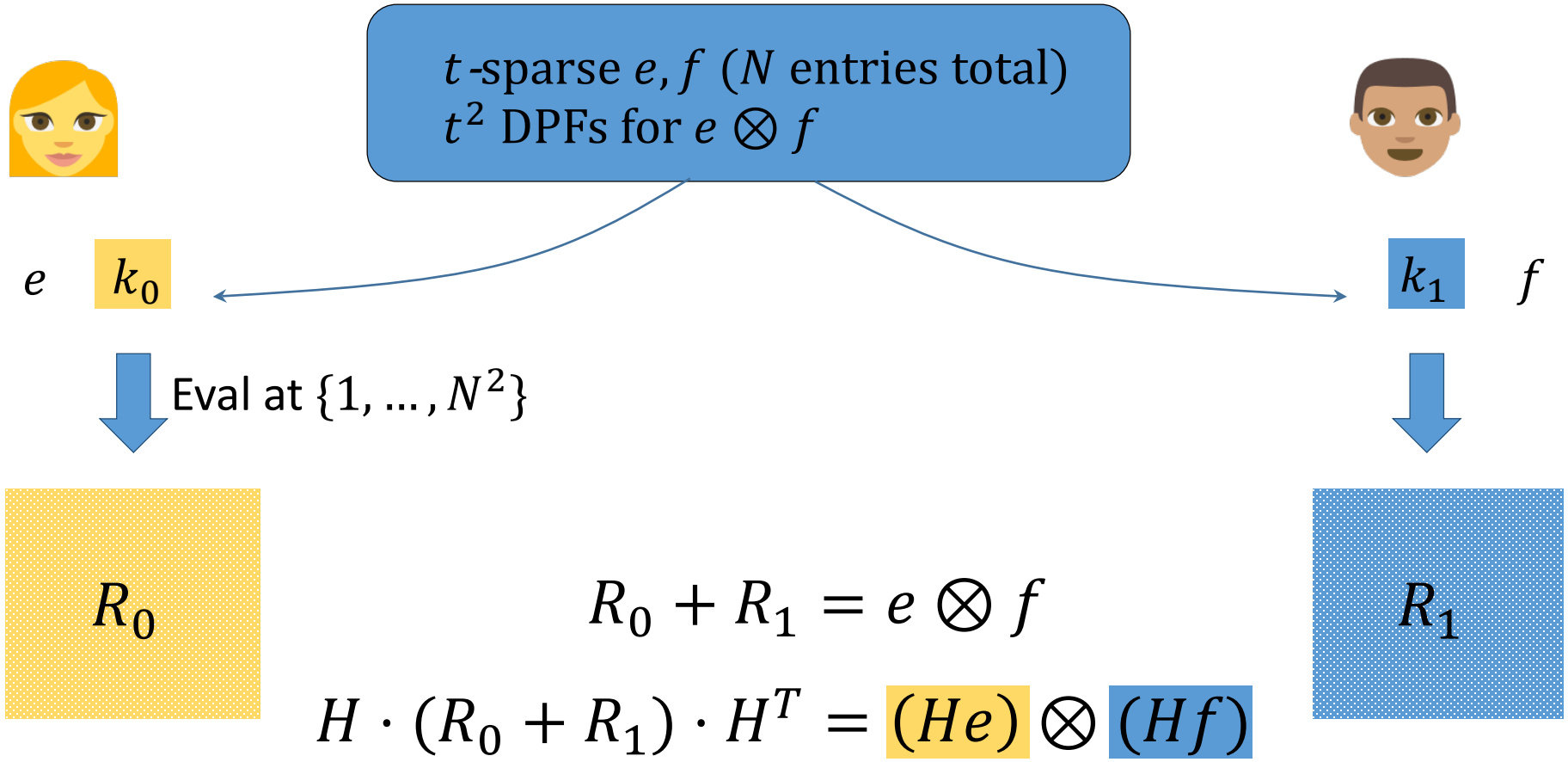


2) Can randomize sparse vectors using LPN



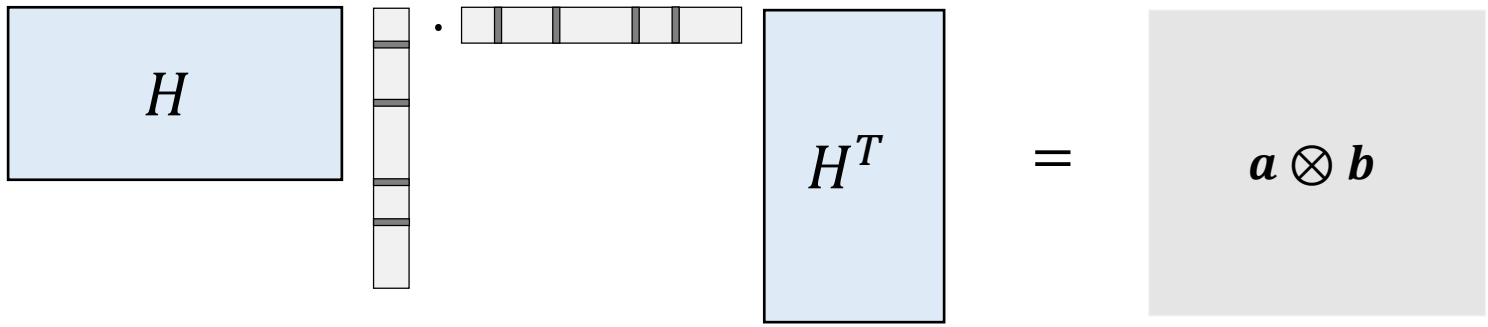
PCG for tensor product from [BCGIKS 19]

Construction:



Improving efficiency of PCG for OLE

- **Problem:** compute entire tensor product, only need diagonals



- **Q:** Can we find an H that allows computing diagonals directly?
  - Yes, with ~~Vandermonde matrix~~ Not secure!
  - Yes, with ring-LPN

Arithmetic ring-LPN assumption

$R_p = Z_p[X]/(X^N + 1)$, N a power of two

$$(a, a \cdot e + e') \approx (a, \$)$$

$a \leftarrow R_p, e, e'$ sparse in R_p

Reducible variant: $X^N + 1$ splits into linear factors mod p

$\Rightarrow R_p \cong Z_p^N$

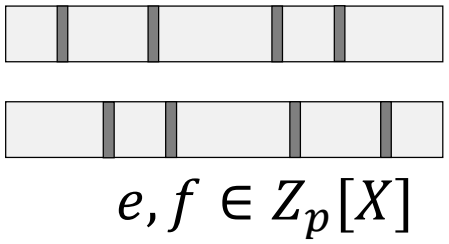
$\Rightarrow$ triple in $R_p \Leftrightarrow N$ triples in Z_p

Security:

- Reducible: seems slightly weaker than irreducible (or standard LPN)
- Attacker can exploit sparse factors $f_i \mid (X^N + 1)$
  - "Dimension reduction" attack (tradeoff: increases noise rate)

Efficient PCG for multiplication triples from ring-LPN

- **Gen:** Distribute shares of sparse e, e', f, f', and $(e, e') \otimes (f, f')$



- **Expand:**
  - Obtain product as

$$(ae + e') \cdot (af + f') \mod (X^N + 1)$$

Linear in $(e, e') \otimes (f, f')$

  - Locally "unpack" into Z_p

Efficient PCG for multiplication triples from ring-LPN

Reduce to $O(Nt)$ with regular errors

- **Cost**: for N ...ples in Z_p
  - $O(Nt^2)$ PRG + $O(N \log N)$ arithmetic operations
  - Seed size: $O(t^2 \lambda \log N)$ bits
- Authenticated triples (almost) for free
  - Just multiply with MAC key
- Extensions
  - Inner products, matrix triples, degree-2 correlations (less efficient)
  - Multi-party (unauthenticated)

Distributed setup protocol for triples

Main challenge: setup DPF keys for $e \cdot f$ in R_p, with malicious security

1) Sparse polynomial multiplication $e \cdot f$

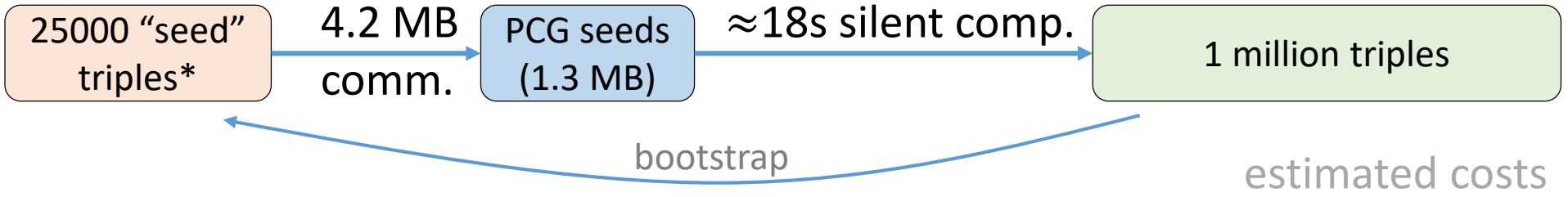
- t^2 mults in $Z_p \Rightarrow$ additive shares of non-zero values
- t^2 binary addition circuits $\Rightarrow$ XOR shares of non-zero positions

2) DPF setup protocol

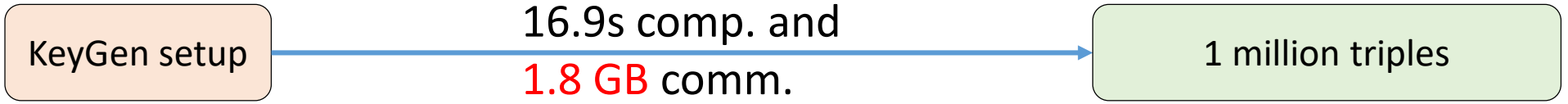
- Malicious secure variant of [Doerner-shelat 17]
- Compute on authenticated inputs + check correctness of outputs
- Allows selective failure attacks – sender can guess 1 bit of LPN error

Silent SPDZ triples: concrete estimates

To produce authenticated triples in Z_p, $\log p \approx 128$, with malicious security:



For comparison, Overdrive [KPR18]:



* plus some OTs, VOLE…

Summary

- PCGs for OLE and multiplication triples are practical using ring-LPN
  - Silent preprocessing compares favourably with recent SPDZ protocols

- Open problems:
  - Extend to triples over Z_{2^k} or Z_2
  - Improved matrix triples, higher-degree correlations
  - Further study of ring-LPN variants

References

[BCGI 18] *Compressing Vector-OLE*
Boyle, Couteau, Gilboa, Ishai (CCS 2018)

[BCGIKS 19] *Efficient Pseudorandom Correlation Generators: Silent OT Extension and More*
Boyle, Couteau, Gilboa, Ishai, Kohl, Scholl (CRYPTO 2019)

[BCGIKRS 19] Efficient Two-Round OT Extension and Silent Non-Interactive Secure Computation
Boyle, Couteau, Gilboa, Ishai, Kohl, Rindal, Scholl (CCS 2019)

[BDOZ 11] *Semi-homomorphic encryption and multiparty computation*
Bendlin, Damgård, Orlandi, Zakarias (CRYPTO 2011)

[BGI 16a] *Breaking the circuit size barrier for secure computation under DDH*
Boyle, Gilboa, Ishai (CRYPTO 2016)

[BGI 16b] *Function secret sharing: Improvements and extensions*
Boyle, Gilboa, Ishai (ACM CCS 16)

[CDI 05] *Share Conversion, Pseudorandom Secret-Sharing and Applications to Secure Computation*
Cramer, Damgård, Ishai (TCC 2005)

[DHRW 16] *Spooky Encryption and its Applications*
Dodis, Halevi, Rothblum, Wichs (CRYPTO 2016)

[DPSZ 12] *Multiparty computation from somewhat homomorphic encryption*
Damgård, Pastro, Smart, Zakarias (CRYPTO 2012)

[Ds 17] *Scaling ORAM for secure computation*
Doerner, shelat (CCS 2017)

[GI 99] *Compressing Cryptographic Resources*
Gilboa, Ishai (CRYPTO 1999)

[GI 14] *Distributed point functions and their applications*
Gilboa, Ishai (EUROCRYPT 2014)

[KPR 18] *Overdrive: Making SPDZ Great Again*
Keller, Pastro, Rotaru (EUROCRYPT 2018)

[SGRR 19] *Distributed Vector-OLE: Improved Constructions and Implementation*
Schoppmann, Gascon, Reichert, Raykova (CCS 2019)

[YWLZW 20] *Ferret: Fast Extension for coRRElated oT with small communication*
Yang, Weng, Lan, Zheng, Wang (CCS 2020)