MPC with Silent Preprocessing

or: Two-Round OT Extension from LPN

Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai,

Lisa Kohl, Peter Rindal, Peter Scholl

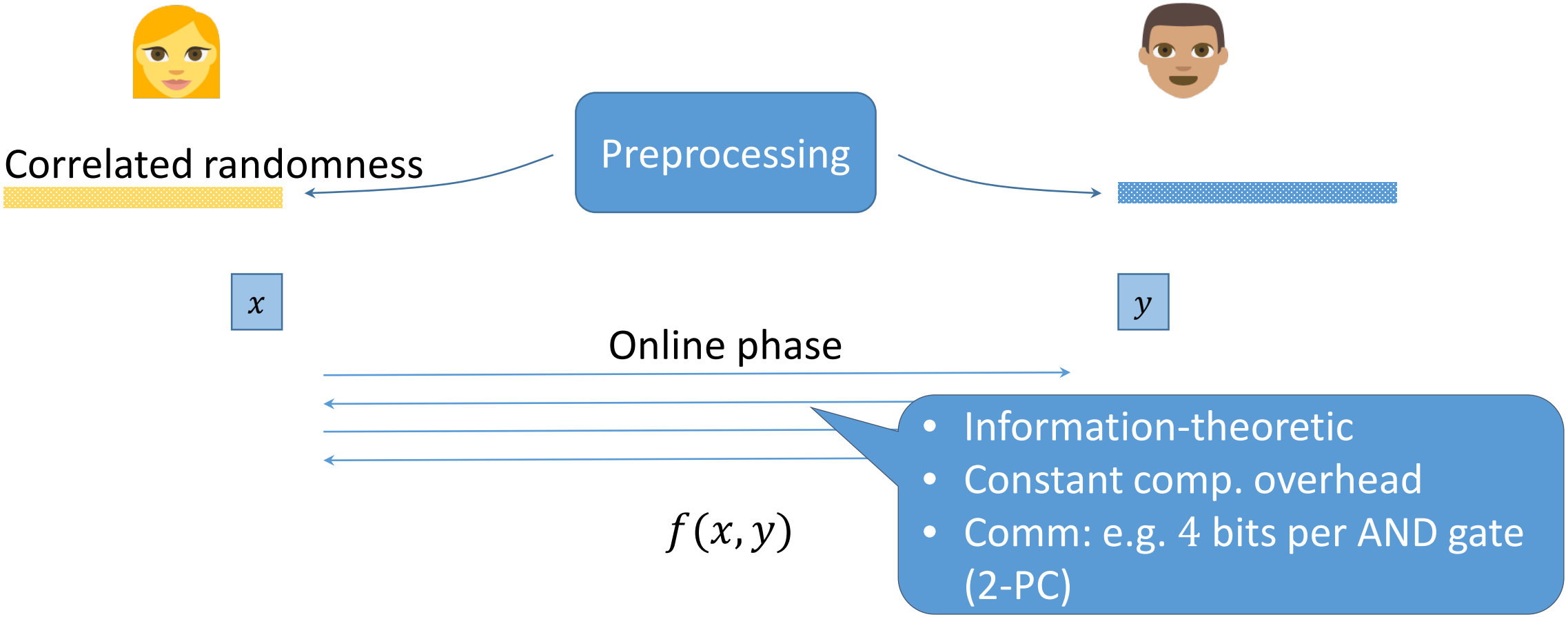
AARHUS UNIVERSITY

SODA

TPMPC 2019, Bar-Ilan University

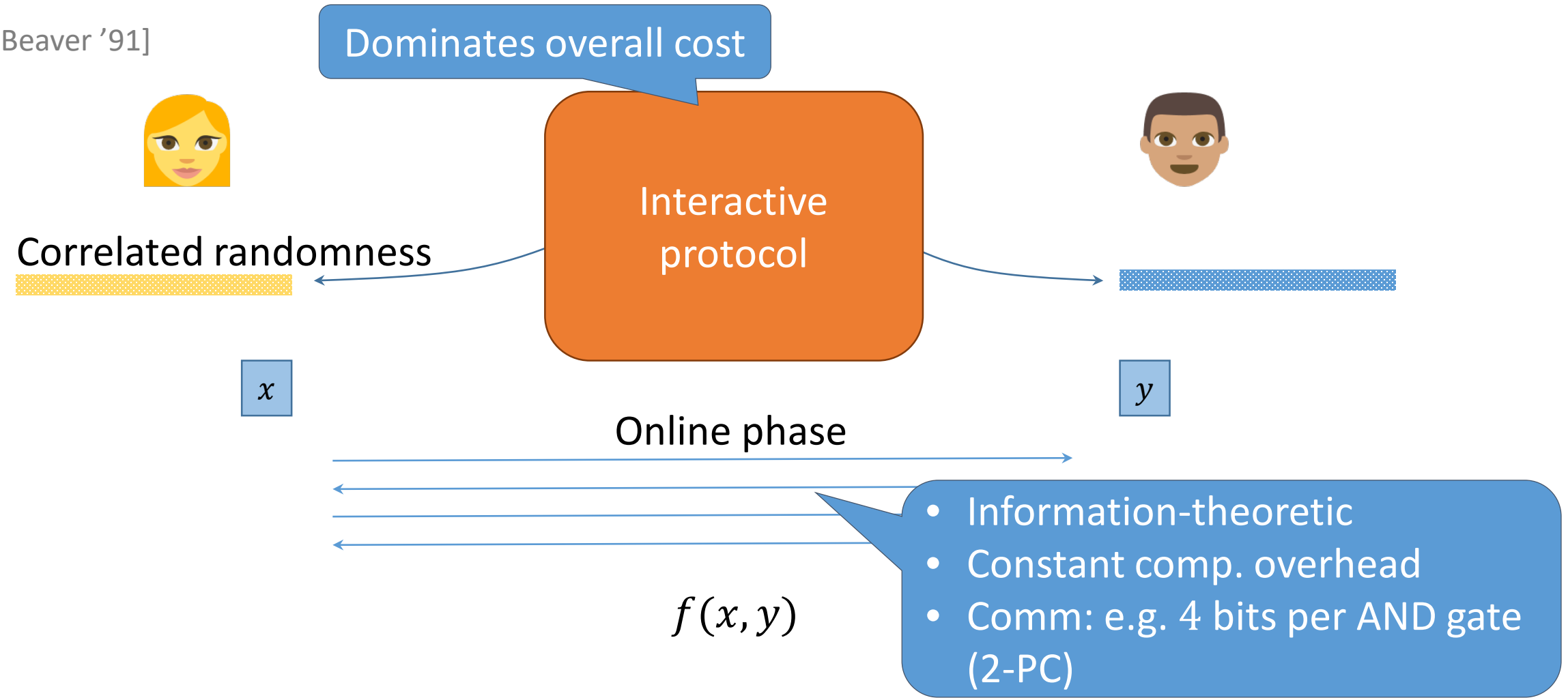
Secure Computation with Preprocessing

[Beaver '91]



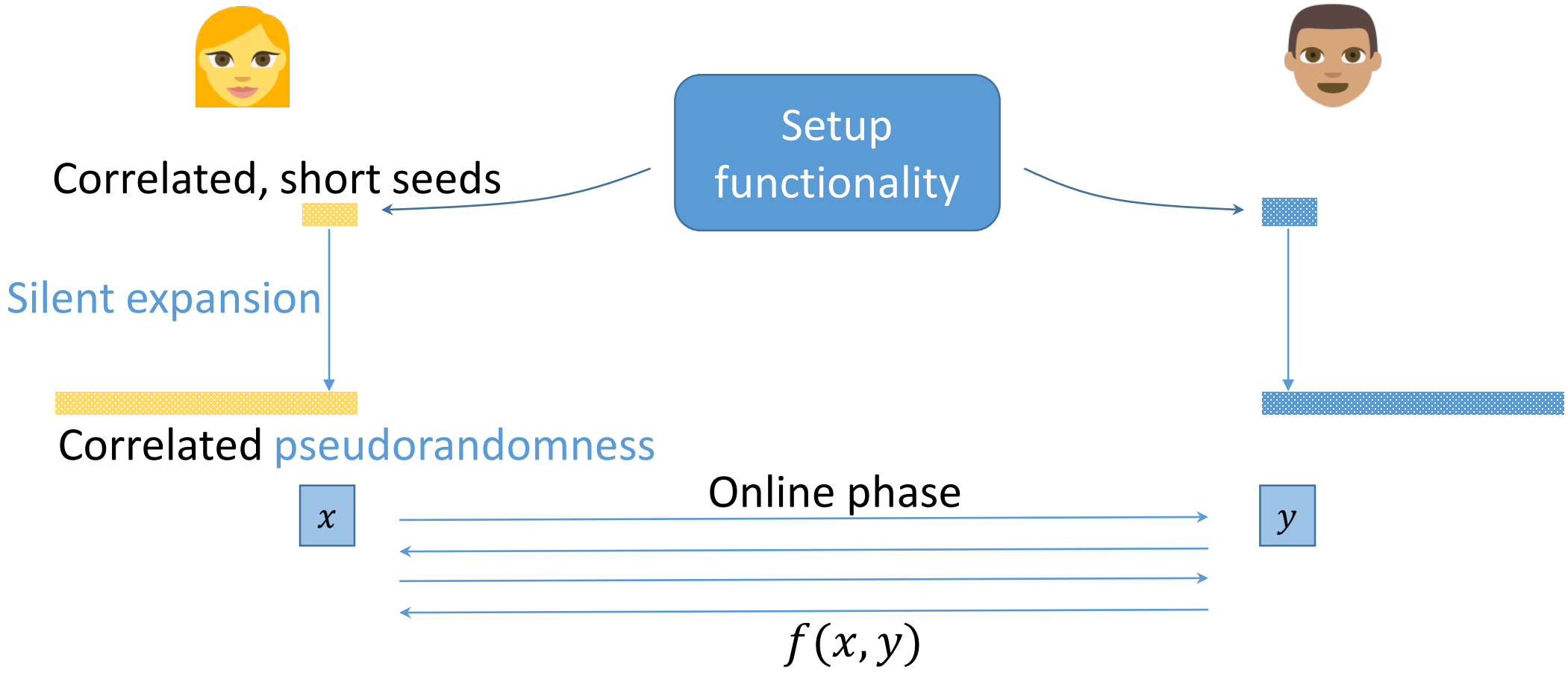
Secure Computation with Preprocessing

[Beaver ’91]



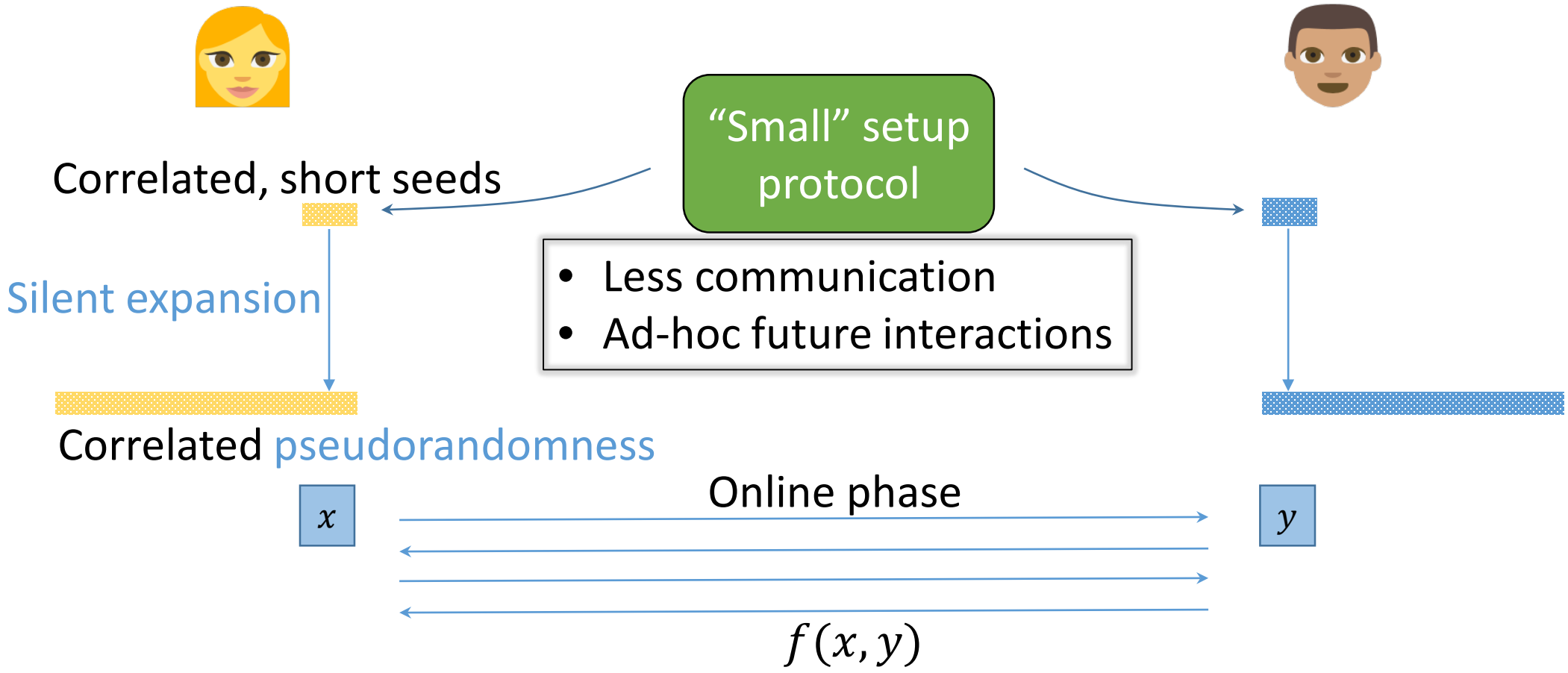
Secure Computation with Silent Preprocessing

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



Secure Computation with Silent Preprocessing

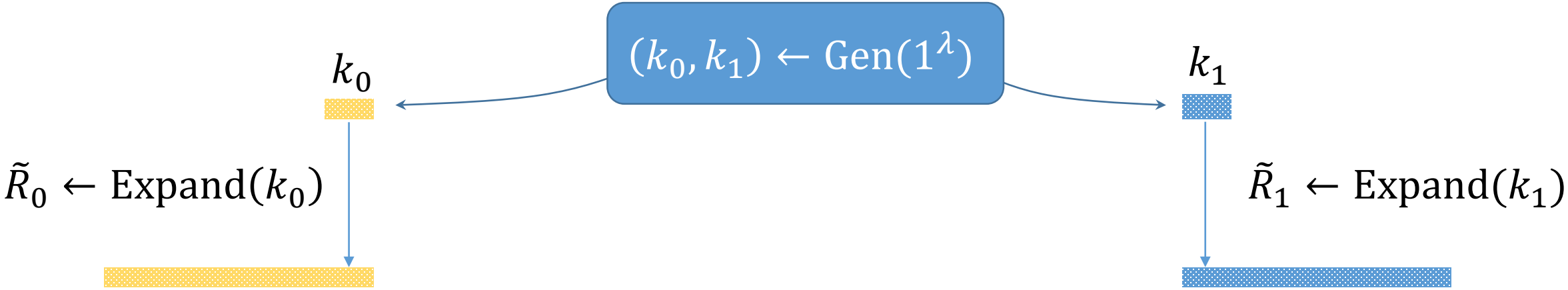
[BCGI 18, BCGIK**S** 19]



Pseudorandom Correlation Generators

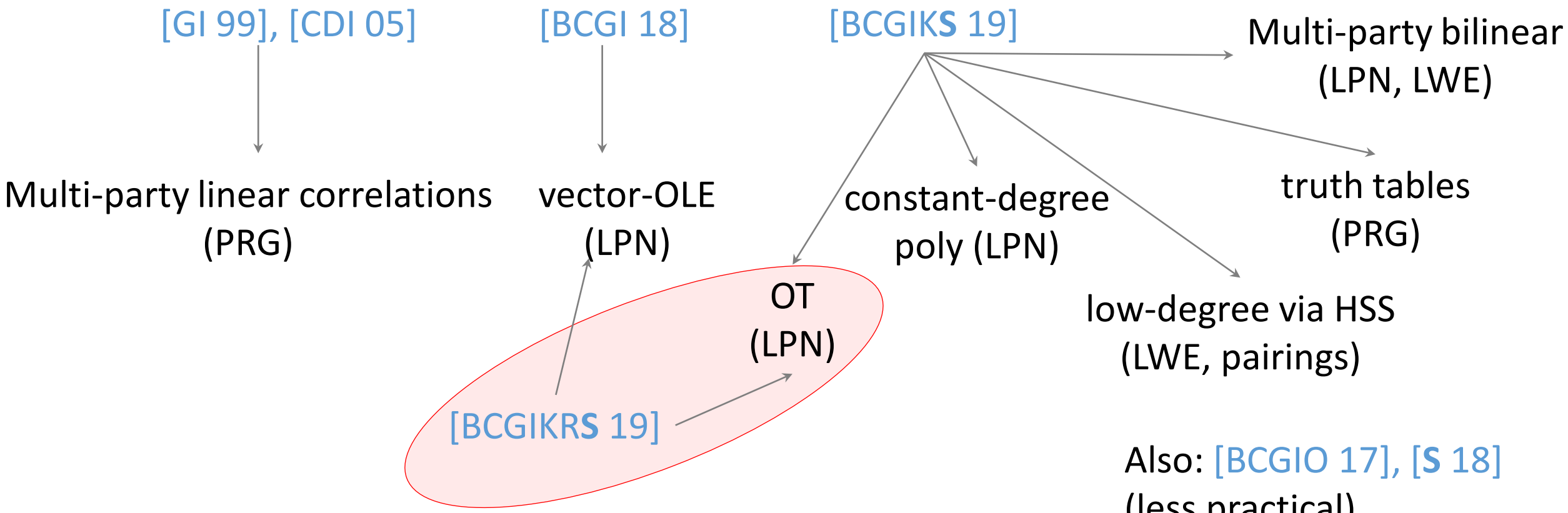
[BCGI 18, BCGIK**S** 19]

- Target correlation: (R_0, R_1)
  - E.g. random OT $((b, m_b), (m_0, m_1))$
- Algorithms Gen, Expand:



Security: $\approx (R_0, R_1)$ (in some sense)

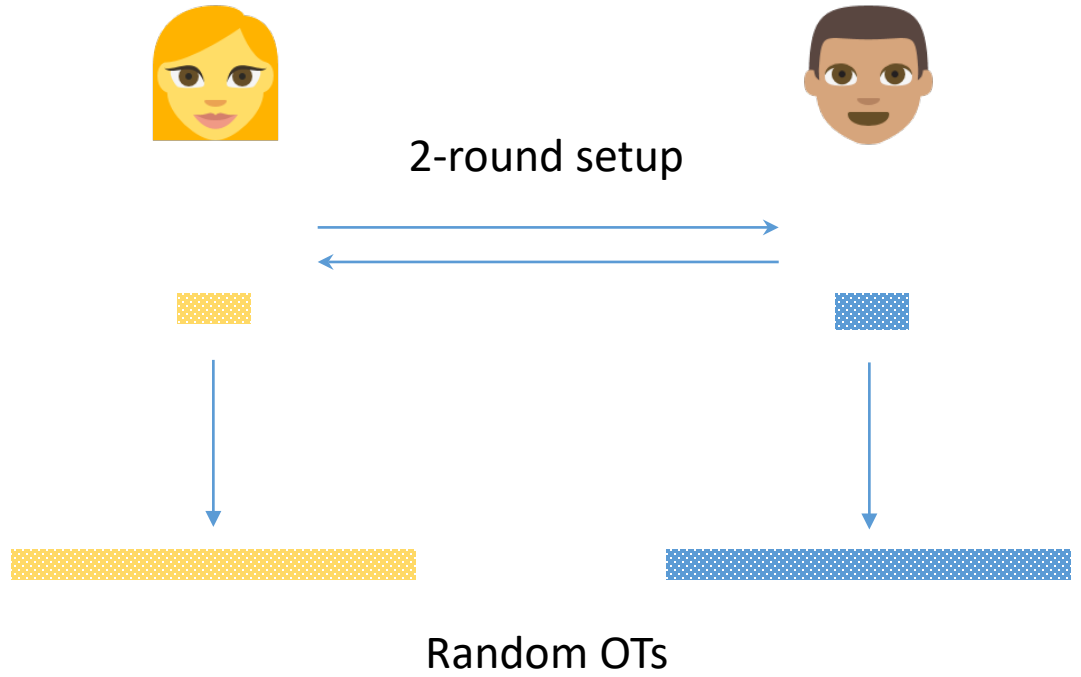
Constructions of PCGs



Silent OT Extension: PCG for random OT from LPN

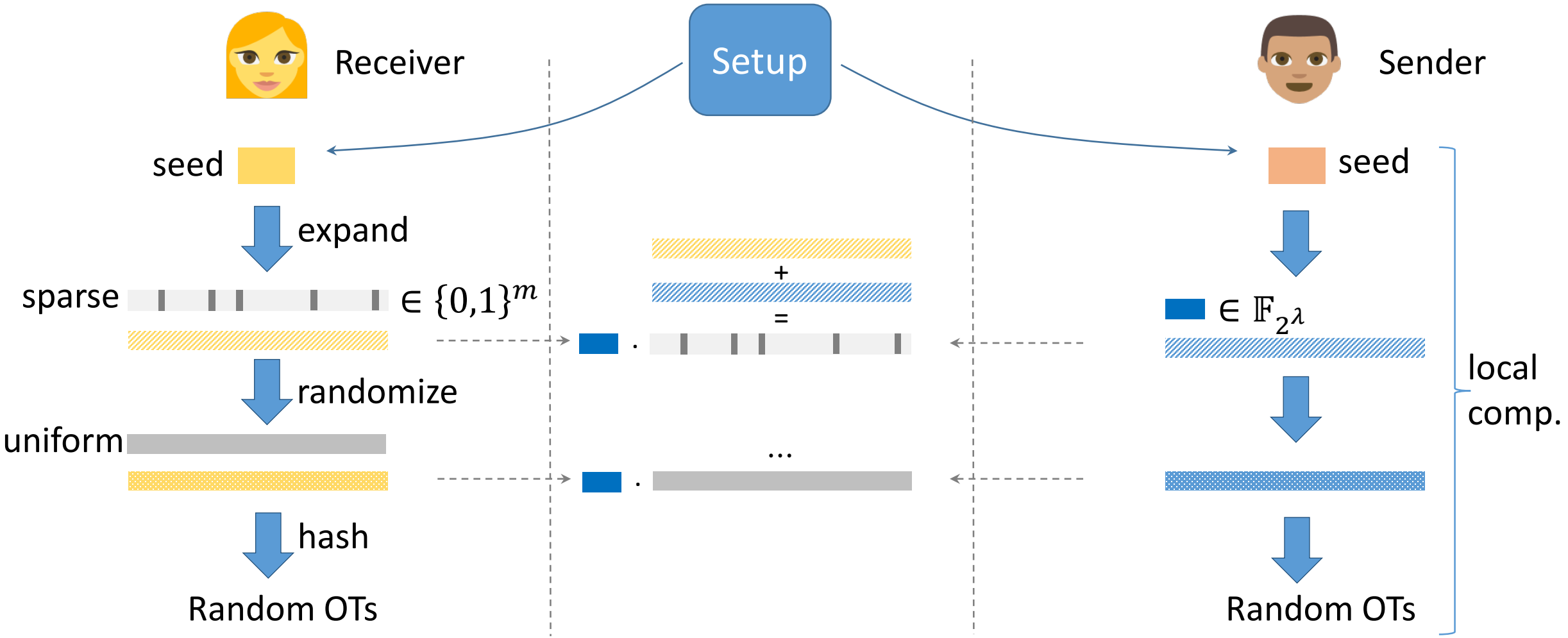
[BCGIK**S** 19, BCGIKR**S** 19]

- Two-round OT extension
  - Useful for non-interactive secure computation [IKOPS 11, AMPR 15, MR 17]
  - Previously only w/ [Beaver 96]
  - Inherently non-black box [GMMM 18]
- Malicious security
  - 2 rounds with Fiat-Shamir
- Faster than [IKNP 03] in WAN:
  - Almost no communication: only pay |input|
  - 3.6 million OT/s



E.g. Semi-honest 2-PC w/ **4.2 bits per AND gate**, **30x less** than [DKSSZZ17]

Silent OT Extension: Overview



Main tool: puncturable PRF

- PRF $F : \{0,1\}^\lambda \times \{1, \dots, N\} \to \{0,1\}^\lambda$

- $k \leftarrow \mathrm{Gen}(1^\lambda)$
  - Master key: allows evaluating $F(k, x)$ for all x
- $k^* \leftarrow \mathrm{Punc}(k, \alpha)$
  - Punctured key: can evaluate at all points except for $x = \alpha$

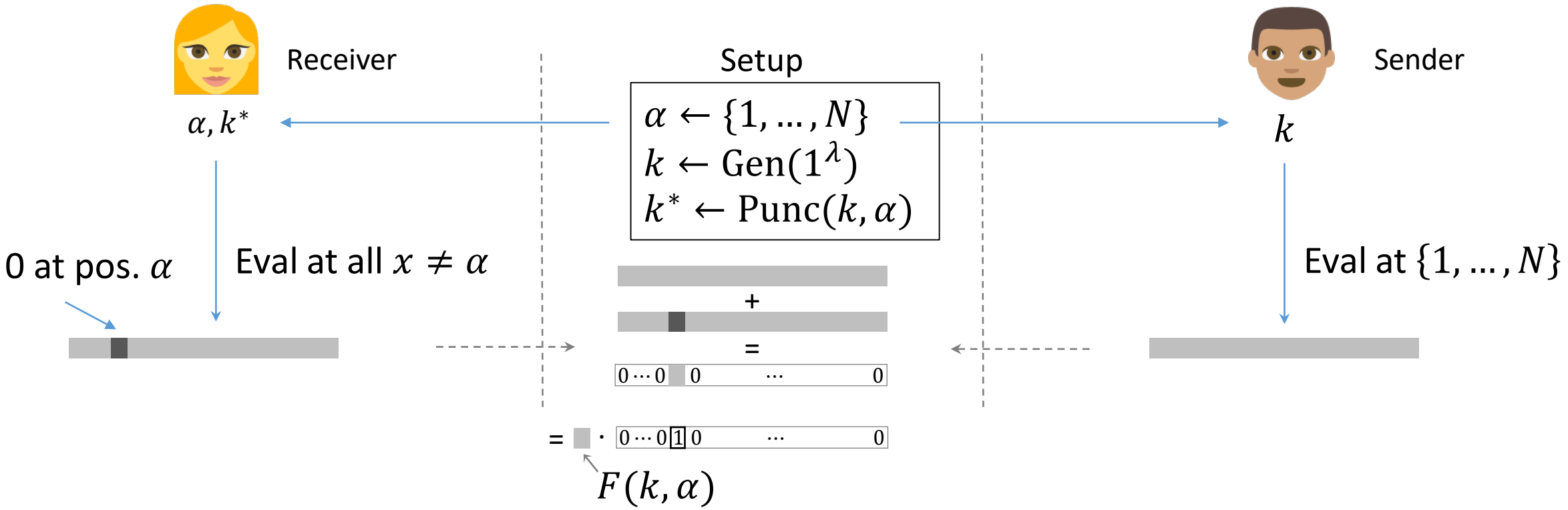
- Security: $F(k, \alpha)$ is pseudorandom, given k^*

- [BCGI 18, BCGIKS 19] use a DPF
- For OT/VOLE, PPRF is enough

Simple tree-based construction from a PRG: $|k| = \lambda, \quad |k^*| = \lambda \cdot \log N$

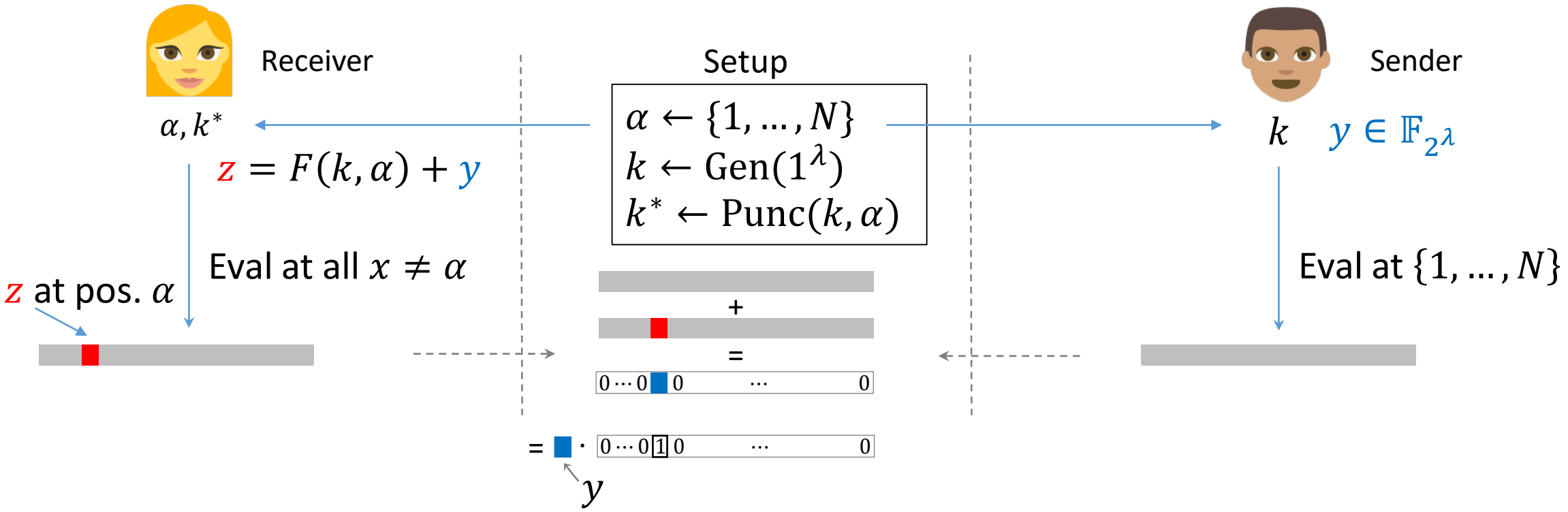
[BW13], [BGI 13], [KPTZ 13]

Key observation: puncturable PRF compresses unit vector products



- Shares compressed from $\lambda \cdot N$ to $\approx \lambda \cdot \log N$ bits

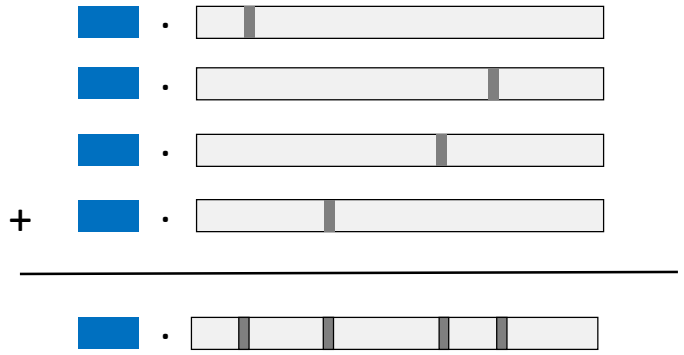
Key observation: puncturable PRF compresses unit vector products



- Shares compressed from $\lambda \cdot N$ to $\approx \lambda \cdot \log N$ bits
- Can tweak to multiply by arbitrary $y \in \mathbb{F}_{2^\lambda}$

From weight-1 vectors to weight-t vectors

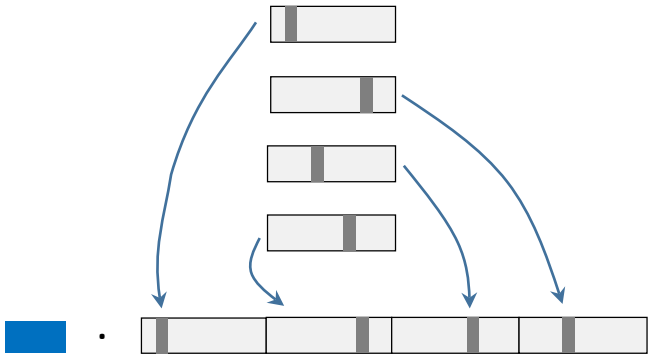
Approach 1: addition



Weight e.g. $t = 4$

Expansion cost: $O(t \cdot N)$ (naïve)
($O(N)$ possible)

Approach 2: concatenation

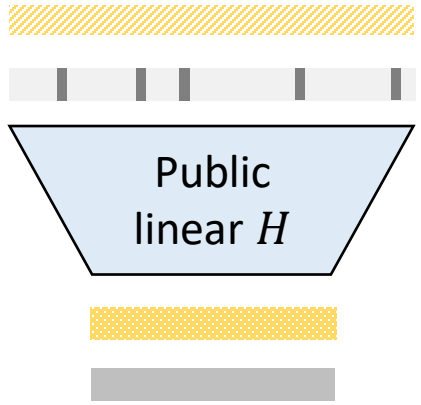


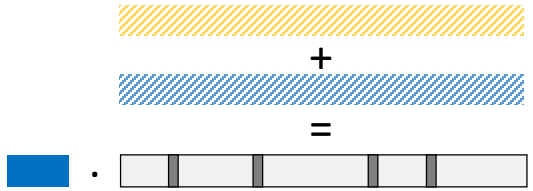
$$O\left(t \cdot \frac{N}{t}\right) = O(N)$$

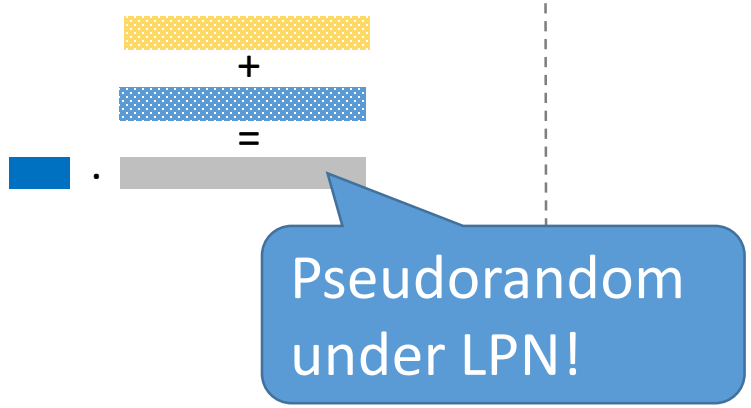
Note: regular error pattern

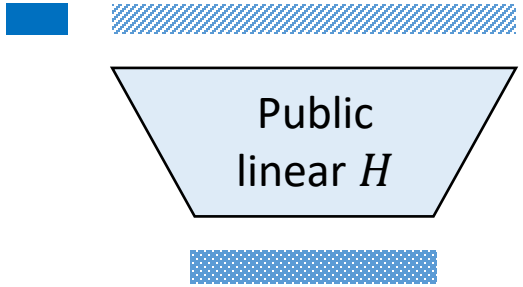
From sparse to pseudorandom products

- Recall: have shares
- Want: uniform vector

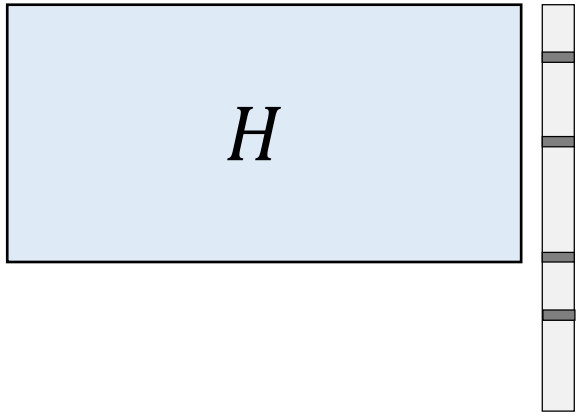








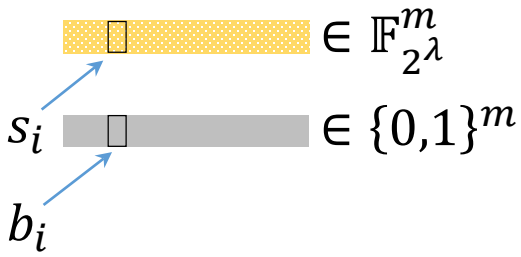
Our LPN setting

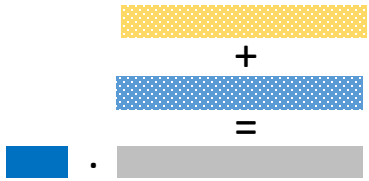


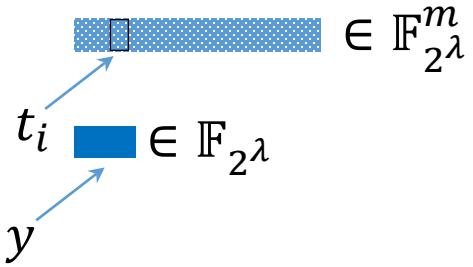
is pseudorandom

- "Dual" version of LPN over $\mathbb{F}_2$
  - ➢ Equivalent to LPN for dual code H^*
- Main challenge: H is big (up to 10 million)
  - ➢ Use quasi-cyclic codes
    - o $\tilde{O}(N)$ (polynomial multiplication over $\mathbb{F}_2$)
- Security:
  - ➢ Similar to PQ cryptosystems BIKE, HQC [ABB+19, AAB+19]
  - ➢ No better with regular errors [AFS 03, BLPS 11, HOSS 18, BLMZ 19]
  - ➢ Noise rate need not imply PKE or OT

From secret-shared products to random OT







$b_i = 1$:
$\Rightarrow s_i + t_i = y$
$b_i = 0$:
$\Rightarrow s_i + t_i = 0$

$\Rightarrow$ OTs on correlated sender strings $(t_i, t_i + y)$

Break correlation with hash function [IKNP 03]

Useful for e.g. TinyOT, garbled circuits

2-round setup and 2-round OT extension

- 2-round punctured PRF setup from any 2-round OT
  - For tree-based PPRF (à la black-box [Ds 17] for DPF)
  - $\log N$ parallel OTs (unlike [Ds 17])

- Two-round OT extension on chosen inputs
  - Still need to convert random → chosen messages
  - Can run conversion in parallel with setup

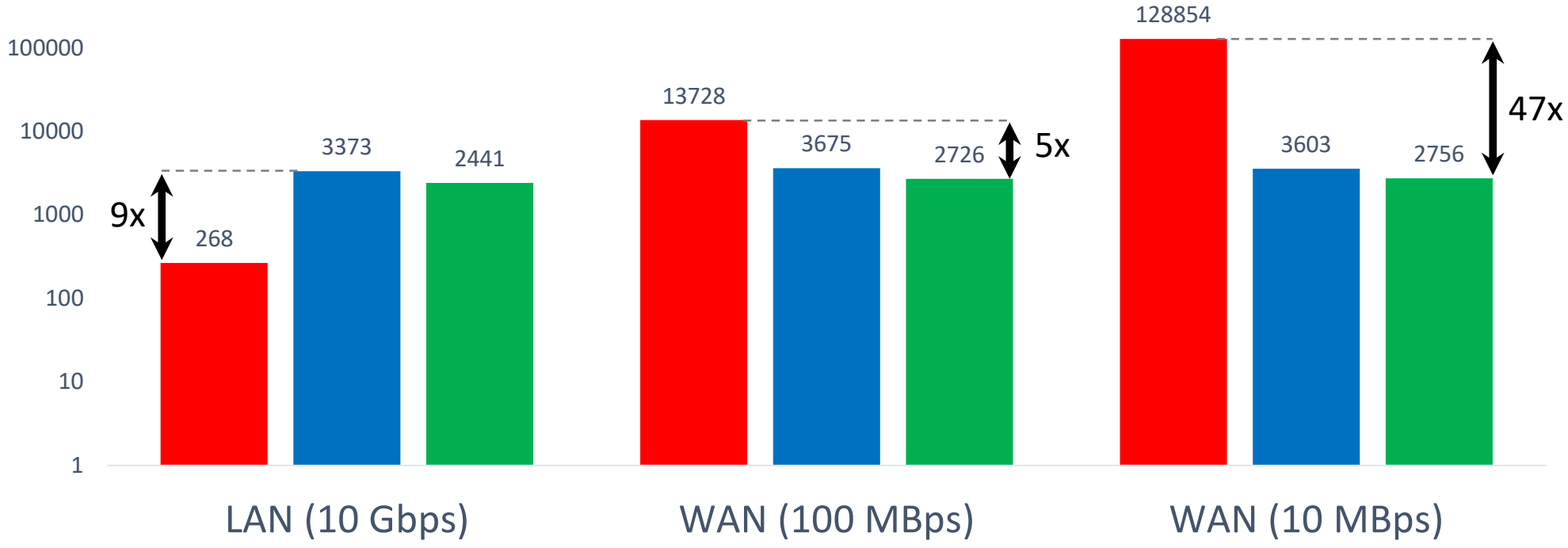
- Malicious security at 10-20% overhead
  - LPN with 1-bit leakage

Concrete parameters for silent OT extension

128-bit security

# output OTs	LPN noise weight	Seed size	Setup comm.	Bits sent per OT
10^4	126	18 kB	58 kB	46.2
10^5	120	23 kB	83 kB	6.7
10^6	118	30 kB	99 kB	0.3
10^7	116	35 kB	127 kB	0.1

Runtimes (ms) for 10 million random OTs



IKNP vs 2-round silent vs 3-round silent

Hybrid: IKNP for ≈1500 base OTs

Total comm: 160 MB vs 145 kB vs 127 kB

Conclusion

- 2-round, silent OT extension
  - Comm: |input| bits
    (previously: λ + |input| bits per OT)

- Broader perspective:
  - Pseudorandom Correlation Generators
  - Silent preprocessing for MPC

- Many future directions:
  - OT: faster codes, incremental output
  - PCGs + setup for complex correlations:
    - OLE, Beaver triples, truth tables, random bits ($\mathbb{Z}_p$), garbled circuits…

Thank you!

Efficient Pseudorandom Correlation Generators: Silent OT Extension and More

Boyle, Couteau, Gilboa, Ishai, Kohl, Scholl

https://ia.cr/2019/129

Two-Round OT Extension and Silent Non-Interactive Secure Computation

BCGIKS + Rindal

Paper: coming soon!

Code: https://github.com/osu-crypto/libOTe (prototype, not yet in master branch)